CHAPTER 5

SYMMETRY AVERAGING REFINEMENT

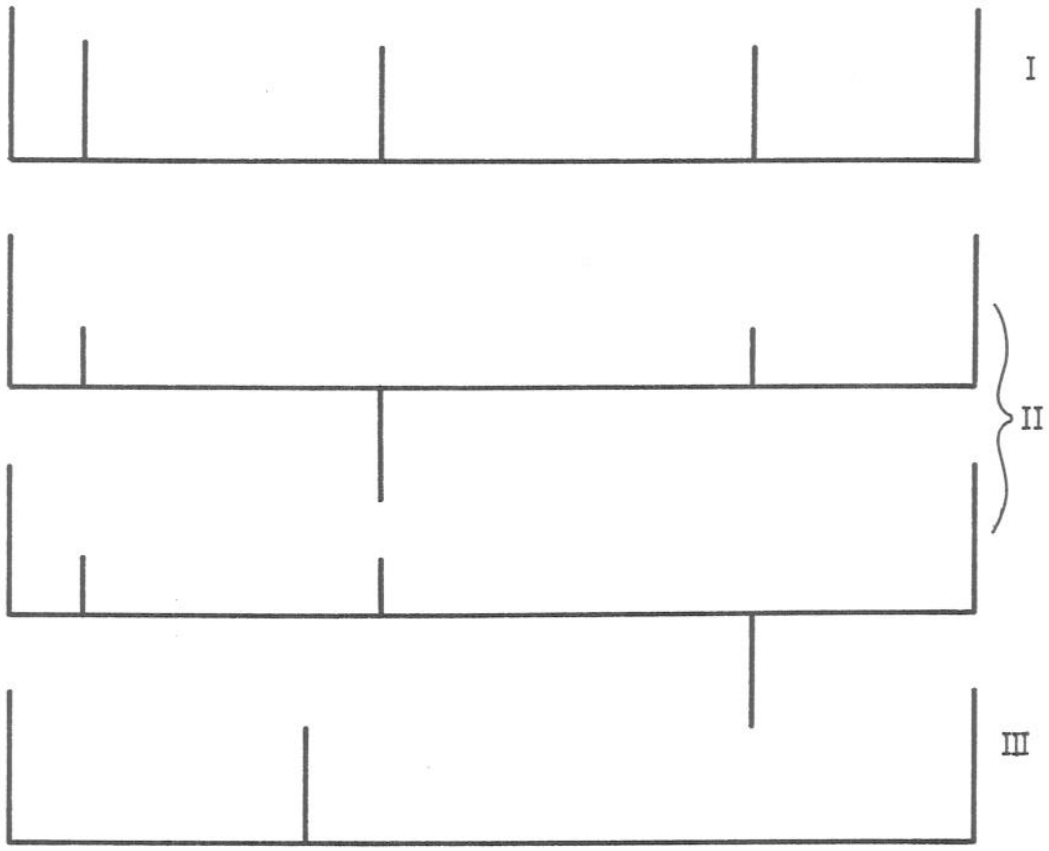
5.1 Introduction.

The principles of the exploitation of non-crystallographic symmetry have been laid out by Crowther (1967, 1969) and Bricogne (1974). Their conclusion applies to crystals which have more than one identical electron density motif in their asymmetric unit, either by virtue of the multimeric nature of the structural units making up the crystal or by the way these pack together. The existence of multimeric structures is widespread in biology for the reasons discussed by Monod, Wyman and Changeux (1965). Such non-crystallographic symmetry implies a very powerful constraint on the possible phases that could be associated with a measured set of structure factors. The constraint is so great in fact that in principle there is only one unique phase set which is consistent with a complete set of F's describing a structure with a non-crystallographic symmetry of order three or greater. To understand why this is true more clearly we will consider a detailed example.

5.1.1 Theoretical Background.

Figure 5.1(a) shows a one dimensional structure consisting of 26 discrete density points. The boundaries

(a) A one dimensional 'structure' represented by 26 discrete density points. The Shannon interpolation function is drawn through these.



(b) A complete set of density eigenfunctions of the non-crystallographic symmetry projection operator, B.

Figure 5.1. Demonstration of the non-crystallographic symmetry constraint in direct space.

are periodic so that the structure is a one dimensional crystal. The density is band-limited to 26 terms in reciprocal space so that exact interpolation is possible for all positions by means of the Shannon formula (Shannon, 1949). The structure also has an exact threefold non-crystallographic symmetry, shown here for simplicity with integer displacements between the motifs, but which is completely generalisable.

We are interested in the effects of the "averaging/truncation operator", B, which acts here by replacing every density point inside each motif by the average of the densities for the corresponding positions in all three, and every density outside the motifs by zero. This operator is a projection operator, as can be seen by its action upon itself,

$$B^2 = B \qquad (5.1).$$

The number of degrees of freedom in the domain of this operator is 26, being one for each density point in the structure, but in its range there are only 7, one for each density point in the motif. This reduction of dimensionality between the domain space and the range space is a fundamental characteristic of projection operators.

Figure 5.1(b) shows diagrammatically one complete set of orthogonal eigenfunctions of the operator B; there exist many such representations as it is highly degenerate.

The set contains 7 functions of type I with unit eigenvalue (one for each of the positions inside the motif), 14 of type II with zero eigenvalue (two for each position, mutually orthogonal to those of type I) and 5 of type III also with zero eigenvalue. The allowed eigenvalues of 1 and 0 are a direct consequence of equation 5.1,

$$Bu = bu$$

$$B^2u = b^2u = bu$$

$$b^2 = b$$

$$b = 0 \text{ or } 1 \qquad \text{Q.E.D.}$$

If any structure is represented in the space spanned by these 26 functions (as a set of 26 coefficients) then its symmetry-averaged image is very easy to calculate, being the sum of the first 7 terms only. Thus the B projection is a filtering process which removes all components of a structure not possessing the required symmetry.

A parallel situation exists in reciprocal space: discrete Fourier transformation of the structure produces a set of 26 complex structure factors (with Friedel symmetry, halving the number of independent parameters). The B projector becomes,

$$H = DBD^{-1} ,$$

where D is the discrete Fourier transform operator. H is also a projection operator,

$$H^2 = DBD^{-1}DBD^{-1}$$
$$= DBBD^{-1}$$
$$= DBD^{-1}$$
$$= H \ .$$

An eigenfunction u_B of B with eigenvalue b satisfies,

$$BU_B = bu_B \ .$$

This transforms to u_H, which is an eigenfunction of H with the same eigenvalue,

$$u_H = DU_B$$
$$Hu_H = DBD^{-1}Du_B$$
$$= DBu_B$$
$$= Dbu_B$$
$$= bDu_B$$
$$= bu_H \ .$$

The operator H, therefore, acting upon a general set of structure factors produces a filtered set, representing a structure with the required symmetry. The dimensionality argument holds for H as it did for B, so the number of degrees of freedom in the structure factors of a structure which has such a symmetry is reduced from 26 (26 complex numbers with Friedel symmetry) to 7 in the example used.

The fundamental crystallographic problem to be solved is the calculation of phases for the measured amplitudes of the structure factors. When enough non-

crystallographic symmetry is present, the problem can become overdetermined: in the example there are 7 degrees of freedom and 13 observable (independent) structure factor amplitudes; this is in principle solvable.

These conclusions have been completely generalised by the work of Bricogne (1974) to the three dimensional situation with arbitrary position and orientation of the motifs, which may also occupy more than one crystal. The critical parameter, that is a measure of the amount of redundancy of information, is the ratio of the number of independent density points inside the motif (7 in the example) to the number inside the the whole crystallographic asymmetric unit of the crystal (26 in the example). This is the same as the ratio of the volume of the motif ('U') to that of the asymmetric unit ('V'). The latter terminology will be used henceforth. When $U/V < 0.5$ the crystallographic problem is overdetermined.

5.1.2 The Refinement of Phases.

Let f_i represent a vector of all the structure factors at their current stage of refinement, containing the measured amplitude and current phase information. Consider now the effect of the projector H:

$$f'_i = Hf_i$$

This projects f_i onto the space spanned only by eigenfunc-

tions of H with unit eigenvalue, since all components with zero eigenvalue vanish. Thus f'_i contains only the components of f_i consistent with the non-crystallographic symmetry.

Since the final structure is assumed to possess the exact non-crystallographic symmetry, it is reasonable to expect that f'_i is a closer approximation to the true set of structure factors than f_i. However the projection operation alters the amplitudes as well as the phases, in general diminishing the former; since the final structure factors must agree with the observations, this should also be used as a constraint. Crowther (1967) suggested an iterative procedure for approaching the true set of structure factors:

$$f'_i = Hf_i \tag{5.3a}$$

$$(f_{i+1})_h = F_h^{obs} \frac{(f'_i)_h}{|(f'_i)_h|} \tag{5.3b}$$

where each F_h^{obs} is an individual measured structure factor amplitude. Equation 5.3(b) can be considered either as a rescaling of the f'_i to the observed lengths or else as a transfer of phase from the f'_i to the F_h^{obs}; the product of one such cycle is a vector f_{i+1}, with the observed amplitudes and refined phases, which can then be recycled until convergence. It is clear that the converged 'correct' set of structure factors is left untouched by both steps, but it is not clear that there is only one such solution. If there

were only one solution, it would be possible to start with a completely random set of phases and refine them to the correct answer; instead, it is recommended that the refinement start with a set of phases closer to the solution than random, either derived from isomorphous replacement or based on a model of the structure.

If it is assumed that there is a single solution then it is possible to prove geometric convergence towards it in the case that $U/V < 0.5$ (Bricogne, 1974).

5.2 Computational Application of Phase Refinement.

Bricogne (1974) discusses the relative merits of direct application of equation 5.3(a) in reciprocal space and its direct space equivalent, as implied in equation 5.2. The conclusion is that the computational advantage of the direct space method is enormous in spite of the addition of the two discrete Fourier transformations bracketing the projection itself. The crystallographic fast Fourier transform ("FFT") made widely available by Ten Eyck (1973) and based on the algorithm of Cooley and Tukey (1965) had made this possible. A computationally efficient scheme for performing the averaging/truncation operation ('B') has been described in detail by Bricogne (1976). If N is the number of structure factors in the calculation, direct multiplication by the H matrix would require N^2 complex multiplication operations, whereas the total of all the steps in the direct

space route requires a number proportional to $N \log_2 N$. Such operations (both the FFT and conventional 'fast' sort procedures scale as N log N). Even though the constant of proportionality is of order 10^2 to 10^3 for a calculation of adequate precision, the advantage of the second method in calculations with $N > 10^4$ is obvious.

Since the most general forms of non-crystallographic symmetry involve completely arbitrary transformations to relate the various motifs (unlike the simple example above), some interpolation method must be used to calculate the density between grid points for purposes of averaging. Shannon's formula (1949) gives an exact evaluation but requires the summation of contributions from every grid point for every interpolation; this immediately returns us to an N^2 problem, so is not suitable. A reasonable approximation to Shannon's formula is to perform eight point <u>linear</u> interpolation,

$$\begin{aligned}\rho(x,y,z) = {} & xyz\,\rho_{111} + xy(1-z)\,\rho_{110} \\ & + x(1-y)z\,\rho_{101} + x(1-y)(1-z)\,\rho_{100} \\ & + (1-x)yz\,\rho_{011} + (1-x)y(1-z)\,\rho_{010} \\ & + (1-x)(1-y)z\,\rho_{001} + (1-x)(1-y)(1-z)\,\rho_{000}\end{aligned}$$

where the ρ_{ijk} represent the densities at the corners of the unit cube surrounding the point (x,y,z). This has the advantage that all the interpolations, once correctly sorted, can be performed during a single pass through the electron density map while holding only two sections of it

in the computer memory at a time. It has the disadvantage that the interpolation error can be fairly large unless the grid of the map is much finer than the wavelength of its highest order Fourier terms. Bricogne (1976) has calculated the interpolation error as a function of oversampling ratio and geometric redundancy, concluding that for 5% precision a grid about 2.5 times finer than the limiting resolution wavelength is usually required. This adds to the size of the calculation, but in a way that is proportional to N log N, not N^2.

Bricogne's (1976) method of performing the averaging projection, B, in direct space consists of 5 steps:

i) Generation of pairs of indices, the first denoting the spatial coordinates of the point in the unaveraged map for which the density is to be interpolated, the second being the grid coordinates of the destination in the motif. An 'envelope' describing the boundaries of the motif is required. Such indices are needed for every grid point inside the envelope and for every one of the transformations to the various motif positions in the unaveraged map. Space group specific routines 'SPGRP1' (referring to the unaveraged map) and 'SPGRP2' (referring to the averaged one) are required to describe the crystallographic symmetry (if any) of the two maps.

ii) The sorting of these indices in the direction of sectioning of the unaveraged map. In this way, all of the interpolations needed between any two sections of the

map are clustered together and in the same sequence as the sections. For the standard Ten Eyck (1973) map format, this direction corresponds to the y coordinate.

iii) Interpolation of the map by the eight point linear method. The output of this step is a list of destination coordinates and associated electron density values.

iv) The sorting of these according to which section of the final map they belong.

v) Reconstitution, section by section, of the final map from the interpolated density contributions. The empty map is preset to a constant value calculated to represent the solvent regions, being places with no contributions, outside all of the envelopes.

Only the last three steps are required for <u>every</u> cycle of refinement, since the same set of sorted indices can be used each time.

There are two ways of reproducing an entire electron density map from identical motifs. A second step, very similar to the averaging one can be used to expand one image of the averaged motif into all of the desired positions; this requires a second step of interpolation as the grid points of one image of the motif do not usually correspond to grid points in the final map. Alternatively, all positions of the averaged motif can be generated simultaneously by using an envelope which contains multiple copies of that of the individual. If the order of the non-crystallographic

symmetry is m fold, then the second method requires the generation and handling of m^2 indices per final motif grid point, while the first requires $(\alpha^3+1)m$, where α is the degree of oversampling. This latter result follows from the fact that the intermediate grid that contains a single averaged motif must also be oversampled, as it is interpolated in the second stage. For the recommended value of $\alpha = 2.5$, the first method has the advantage when $m > 16$. The expanded TBSV case has $m = 30$, so the double interpolation method is the correct choice. This has the additional advantage that the intermediate grid can be chosen to be the natural coordinate frame of the virus particle itself, for easy comparison with the native structure. It also allows a finely sampled version of the map to be stored more compactly as the intermediate form, which occupies about a tenth of the space of the unaveraged one.

5.3 Application to Expanded TBSV.

The overall scheme that was used for phase refinement of the expanded TBSV data is shown in figure 5.2. It uses a double interpolation in the form of steps 'AVERAGE' and 'EXPAND', which perform the 30:1 symmetry averaging and the 1:30 reconstruction operations respectively. The intermediate 'U' grid between these steps contains a single averaged motif of the viral shell, which is the triangle pyramid shaped icosahedral asymmetric unit suitably truncated on the

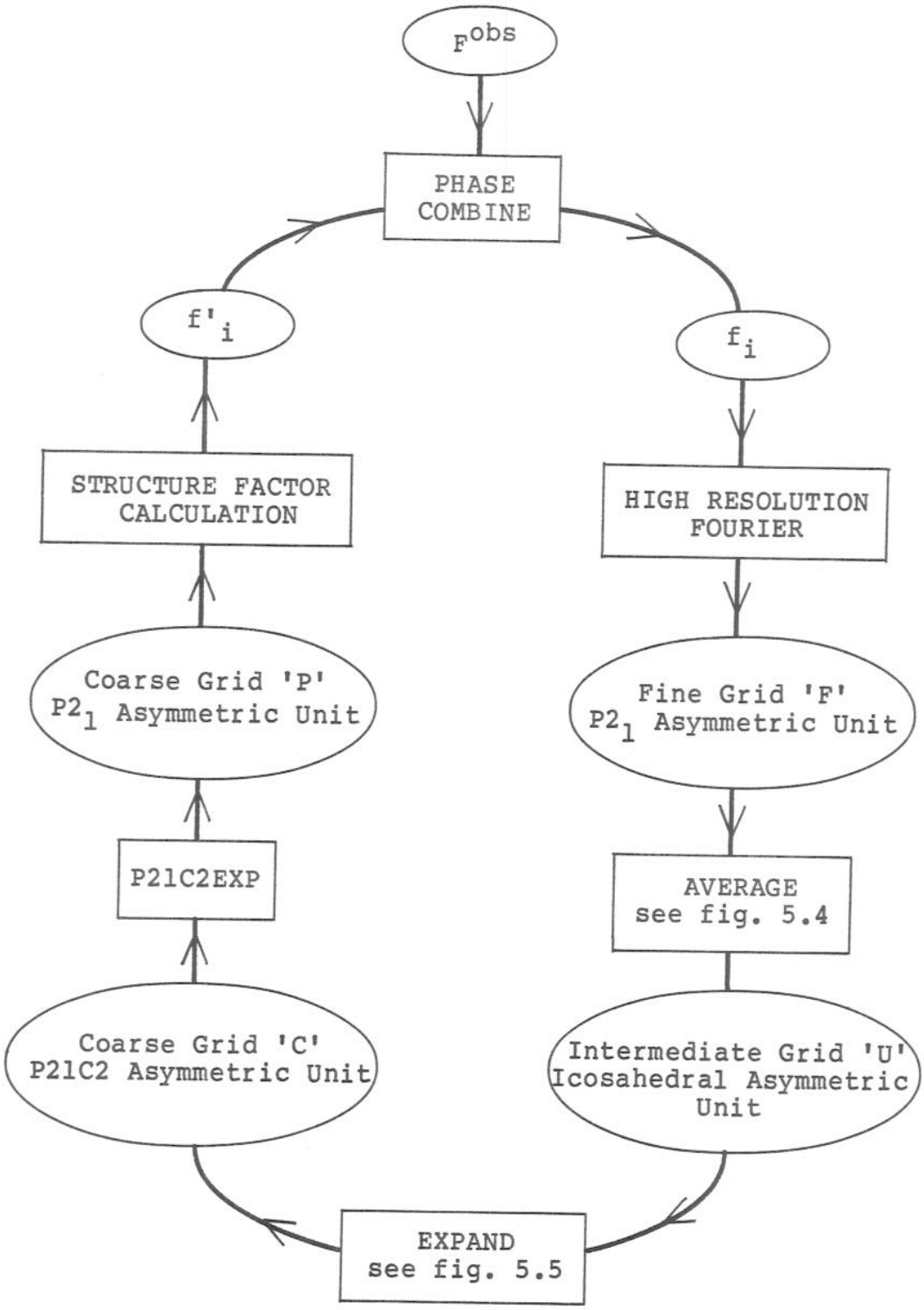


Figure 5.2. Grand scheme for symmetry-averaging refinement of the expanded TBSV phases.

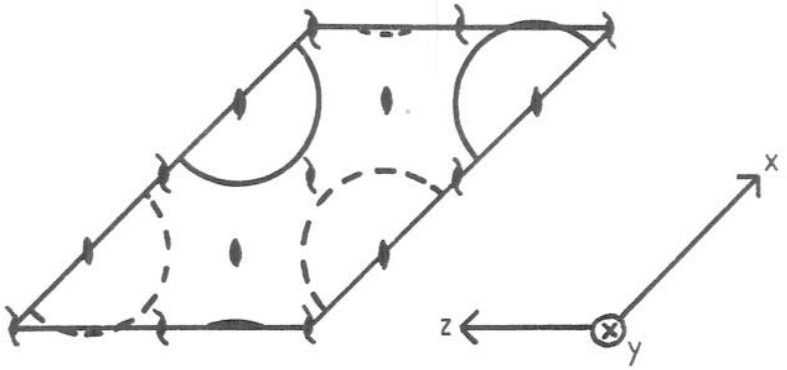
outer surface (see discussion of envelope below); because this is an isolated object in space, the box that contains it is not of critical shape, so is chosen to be a rectangular prism. The grid is cubic and so the fineness is specified by its grid spacing in Ångstroms. All the remaining grids are crystallographic and their specification is by the number of divisions along each edge of the unit cell. The relation of each of these grids to the packing of the virus particles in the crystal is shown in figure 5.3.

5.3.1 Space Groups and Grid Sizes.

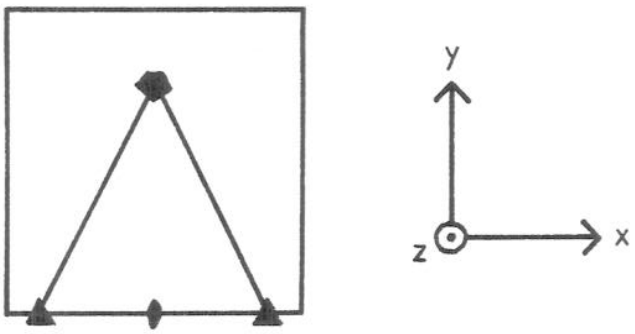
All Fourier calculations were done in the space group $P2_1$ (a subgroup of C2) for which a crystallographic FFT was available. The asymmetric unit in $P2_1$ is half of the unit cell; by convention the unique volume is defined as,

$$0 < x < 1$$
$$0 < y < 0.5$$
$$0 < z < 1 .$$

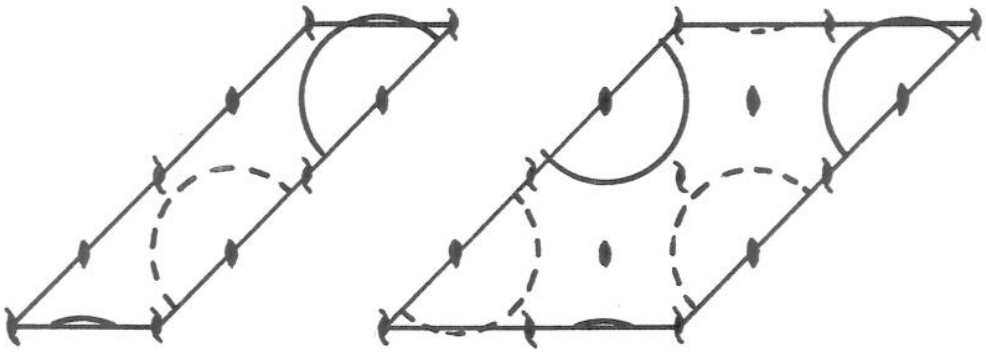
The space group C2 has a quarter of the unit cell as the asymmetric unit, so the $P2_1$ unique volume contains <u>two</u> C2 asymmetric units. This causes no problem during the 'AVERAGE' step, since the 'F' grid contains two identical images of the unaveraged virus. However, thirty fold expansion of the averaged motif produces only half an asymmetric unit of the $P2_1$ coarse grid map. The full $P2_1$ asymmetric unit could



(a) Fine grid 'F'. Contains one $P2_1$ asymmetric unit of the crystal.



(b) Intermediate grid 'U'. Contains one icosahedral asymmetric unit of the virus (indicated by the triangle) in its local orthogonal frame inside a rectangular box.



(c) Coarse grids 'C' (left) and 'P' (right). 'C' contains one C2P21 asymmetric unit. This is duplicated to fill 'P', which contains the coarse grid $P2_1$ asymmetric unit.

Figure 5.3. The grids used for symmetry-averaging calculations with expanded TBSV. See table 5.1 for the values of the grid spacings. The closed circles represent virus particles in the zero section; open circles represent particles at y = 1/2.

be generated in one step by using a 60 fold expansion, but this is rather extravagant. Instead, the 'C' map is calculated first, which contains a single 'P21C2' (space group C2 with the origin shifted to lie on the 2_1 axis) asymmetric unit. This is then doubled by expanding the map, section by section, about the 2-fold axis to produce the 'P' map suitable for Fourier transformation. The program to do this was a corrected version of 'P21C2EXP' that had been used in a similar analysis of the Satellite Tobacco Necrosis Virus (Unge, et. al., 1980).

Two sets of grids sizes were selected, one for working at low resolution (12 Å ngstrom), the other at high (8 Å ngstrom). These are given in table 5.1, along with the estimated interpolation error based on Bricogne's (1976) formula. Consideration was given to the fact that the FFT is more efficient when the number of cell divisions has small prime factors.

5.3.2 Choice of Parameters for 'AVERAGE'.

All of the space group specific parts of the Bricogne (1976) averaging scheme described above lie within the first operation, corresponding to the program GENERATE. This is a general program with 11 possible modes of operation to handle various situations including envelope transportations (see below), handling of proper and improper symmetry as well as options of imposing space group con-

	12 Ångstrom	8 Ångstrom
Fine Grid 'F'	270x216x192	450x360x324
Average spacing	2.01Å	1.20Å
Intermediate Grid 'U'	2.0Å	1.4Å
Coarse Grid 'P' ('C')	96x72x64	150x120x108
Average spacing	5.90Å	3.60Å
Mean oversampling ratio	3.0Å	3.3Å
Expected r.m.s. error for interpolation.	9.8%	8.0%

Table 5.1. Grid spacings for phase refinement calculations at high and low resolution. Interpolation error estimates are calculated according to Bricogne (1976).

straints on either the source or destination indices. It requires a large amount of input information, which is used at the beginning of execution of the program to calculate the list of m transformations which act upon the integer grid coordinates of the points inside the envelope to produce the real grid coordinates for the interpolations. The program then proceeds rapidly through the envelope, calculating m interpolation positions for every point, folding these appropriately and writing out the desired list of indices. The transformations are calculated by the equations (Bricogne, 1975),

$$T_i = G_1 QPU_i G_2 \qquad (5.4a)$$

$$t_i = G_1 (QPu_i + d) \qquad (5.4b)$$

T_i are the m matrix parts of the transformation which act upon integer grid points in the final map to give real grid positions in the initial one.

t_i are the vector parts of the transformations which are added to the results of the multiplications by T_i.

G_2 is a diagonal matrix containing the three grid spacings of the final grid, thereby converting the grid coodinates to Ångstroms.

U_i are the set of m transformation matrices defined in the coordinate frame of the <u>final</u> map. They transform the final Ångstrom position(s) onto the initial one(s), also in Ångstroms.

u_i are the set of vector displacements associated with the U_i, also in Ångstroms.

P is a matrix defining the orientation of the final coordinate frame relative to an internal orthogonal one, again mapping Ångstroms to Ångstroms. The space group information of the final grid is contained here.

Q is a matrix mapping the internal orthogonal Ångstrom frame onto the crystallographic frame of the initial grid. In the crystallographic frame the coordinates are fractions of the unit cell edge. This defines the space group of the initial grid.

d is the position of the origin of the initial map in the crystallographic frame.

G_1 is a diagonal matrix containing the number of cell divisions of the initial map. This converts the fractional (crystallographic) coordinates in this map to real grid coordinates.

The details of the box marked 'AVERAGE' in the grand scheme are shown in figure 5.4. The choice of envelope will be explained later; it is a little larger than the triangular pyramid of the icosahedral asymmetric unit that becomes filled with the intermediate averaged map of a single motif. Mode 2 of GENERATE is used here: this imposes no space group constraints on the indices of the final map, and so does not require a SPGRP2. The standard $P2_1$ version of SPGRP1 is used to fold the indices for interpolation in the fine grid map. The input parameters to the program are

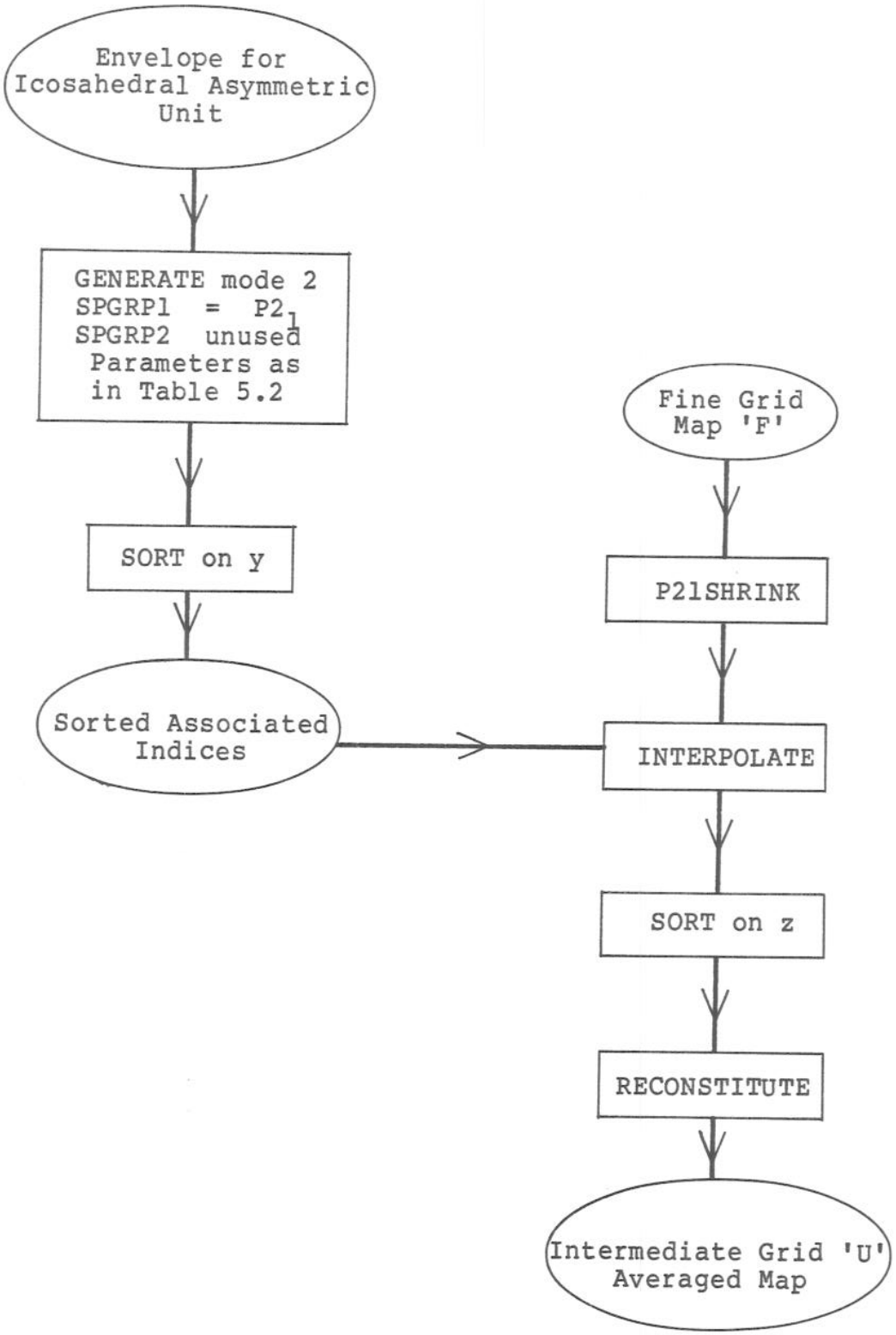


Figure 5.4. Flow diagram for the first interpolation step, AVERAGE. Mode 2 of GENERATE calculates indices for a mapping, while folding the initial indices according to a space group, but not the final ones.

$$G_1 = \begin{pmatrix} 270 & & \\ & 216 & \\ & & 192 \end{pmatrix} \qquad G_2 = \begin{pmatrix} 2.0 & & \\ & 2.0 & \\ & & 2.0 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1/a & & -\cot(\beta)/a \\ & 1/b & \\ & & 1/c \sin(\beta) \end{pmatrix} \qquad P = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}$$

$$d = (-1/4, 0, 0)$$

$$U_{2i+3j+5k} = \begin{pmatrix} \cos\theta & & -\sin\theta \\ & 1 & \\ \sin\theta & & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & & \\ & -1 & \\ & & -1 \end{pmatrix}^i \begin{pmatrix} & & 1 \\ 1 & & \\ & 1 & \end{pmatrix}^j$$

$$x \begin{pmatrix} .309 & -.809 & .500 \\ .809 & .500 & .309 \\ -.500 & .309 & .809 \end{pmatrix}^k \qquad i = 0,1, \quad j = 0,1,2, \quad k = 0,..4$$

$$u_i = (0, 0, 0)$$

Table 5.2. Transformation matrices input to GENERATE to achieve the 30 fold averaging step 'AVERAGE'. Values are for 12Å limiting resolution calculations. G_1 and G_2 are modified for the 8Å calculation (see table 5.1).

summarised in table 5.2. Q, G_1 and d describe the $P2_1$ space group of the unaveraged map with a particle centred at x = -1/4 (see figure 5.3), while G_2 and P describe the cubic intermediate grid. The 30 matrices U_i are generated with a separate program according to the formula in the table: all combinations of icosahedral 5-fold, 3-fold and 2-fold (about a <u>different</u> axis from the crystallographic b axis) rotations are premultiplied by a rotation about the y axis of angle $-\theta$. In this way the axes of the intermediate grid are the natural ones of the virus itself, and the packing into the crystal with a skewed orientation is accommodated.

The remaining steps of figure 5.4 correspond to the normal procedure described above. The step 'P21SHRINK' performs two functions (Bricogne, 1977):

i) The boundaries of the fine grid map are enlarged to ensure that there are eight points of density available around every position needed for interpolation. This involves enlarging every section by one grid point in each direction and reproducing the bottom section on the top of the map.

ii) The number of bytes used to store each density value is reduced from four to two by suitably scaling the whole map and converting the densities to integers. This relieves the pressure on the computer's disk storage resources during the most demanding step of all, the interpolation.

5.3.3 Choice of Parameters for 'EXPAND'

The reproducing of the averaged motif thirty times to fill up the P21C2 unit cell is achieved by the box marked 'EXPAND' in the grand scheme of figure 5.2. The details of this are summarised in figure 5.5 and table 5.3. The envelope describes half a virus particle on the coarse grid in such a way that the operation of the b axis 2-fold generates the full particle, yin/yang fashion. This envelope describes the complete protein portion of the P21C2 asymmetric unit, but is not folded into the crystallographically defined asymmetric unit (as in figure 5.3, for instance). The folding is the purpose of the SPGRP2 routine in the GENERATE step of 'EXPAND' and is a modified version of the standard library P21C2 routine. The changes made were to compensate for the corrections to P21C2EXP and to introduce the origin shift to x = -1/4. The latter could be achieved within the normal scope of GENERATE, but not as easily.

GENERATE mode 5 is used here because space group folding is required for both the initial and final maps. The folding in the initial map does not impose a <u>space</u> group symmetry, but instead the 532 <u>point</u> group symmetry of the virus; it must fold the coordinates of any point in space onto its point group related image in the standard icosahedral asymmetric unit. An efficient routine to do this was written in the format of a SPGRP1.

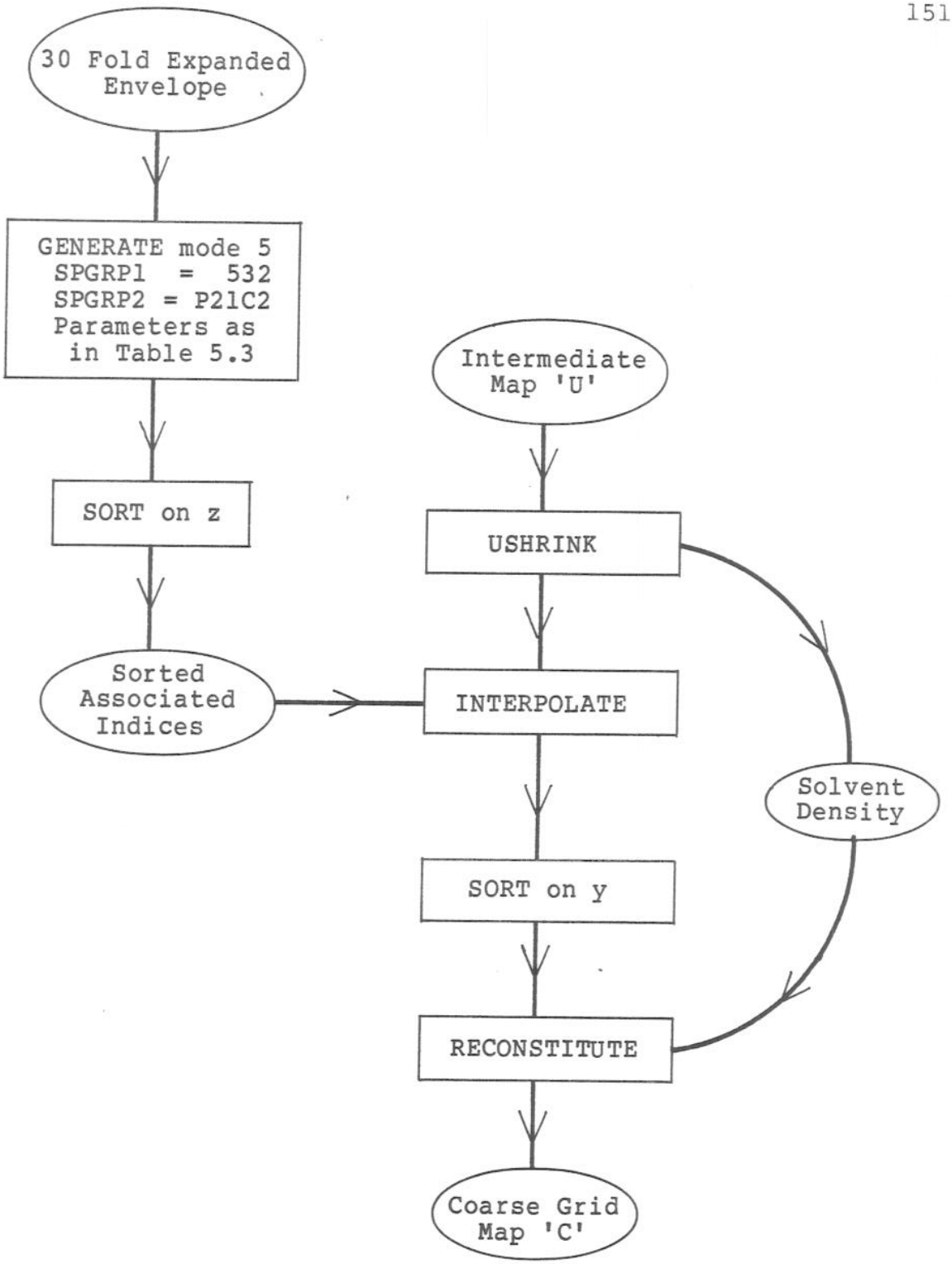


Figure 5.5. Flow diagram for the second interpolation step, EXPAND.

$$G_1' = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix} \qquad G_2' = \begin{pmatrix} a/96 & & \\ & b/72 & \\ & & c/64 \end{pmatrix}$$

$$Q' = \begin{pmatrix} 1/2.0 & & \\ & 1/2.0 & \\ & & 1/2.0 \end{pmatrix} \qquad P' = \begin{pmatrix} 1 & & \cos(\beta) \\ & 1 & \\ & & \sin(\beta) \end{pmatrix}$$

$$d = (0, 0, 0)$$

$$U' = \begin{pmatrix} 1 & & -\cot(\beta) \\ & 1 & \\ & & \sec(\beta) \end{pmatrix} \begin{pmatrix} \cos(-\theta) & & -\sin(-\theta) \\ & 1 & \\ \sin(-\theta) & & \cos(-\theta) \end{pmatrix} \begin{pmatrix} 1 & & \cos(\beta) \\ & 1 & \\ & & \sin(\beta) \end{pmatrix}$$

$$u = (0, 0, 0)$$

Table 5.3. Transformation matrices input to GENERATE to achieve the 1:30 virus expansion step, 'EXPAND', in figure 5.5. Values are for 12Å limiting resolution calculations. Q' and G_2' are modified for the 8Å calculation (see table 5.1).

Since no averaging is needed in the 'EXPAND' step, only one U_i matrix, U', is used. However, since this matrix is defined in the monoclinic coordinate frame of the final grid, it is not a simple rotation matrix. To obtain the matrix U' in the final coordinate frame, consider that the combined effect of

$$P' \; U'$$

where P' is the matrix relating the final coordinate frame to the intermediate one (see equation 5.4a), is to be the same as

$$R^{-1}(\theta) \; P'$$

where $R(\theta)$ is the standard rotation of angle θ about the y axis, defined in an <u>orthogonal</u> (Ångstrom) coordinate frame.

Since $$R^{-1}(\theta) = R(-\theta)$$

$$R(-\theta) \; P' = P' \; U'$$

$$\Rightarrow \quad U' = P'^{-1} \; R(-\theta) \; P'$$

G_1' and Q' define the cubic intermediate grid which is the initial grid for this step. The final grid is defined by the matrix P' and its spacing by G_2'.

Since the output of 'EXPAND' is to be used for the calculation of structure factors, it is important to treat the solvent regions correctly. If a step in the density arose between the filled protein region and the unfilled

space outside it, this could lead to errors in the structure factors and to Fourier ripples in the later maps. Rather than follow some elaborate procedure to average the entire solvent region, its level was estimated from the outermost skin of the averaged motif: the average of the electron densities in the cells at the greatest z values inside the envelope of the intermediate grid map (about 100 points, each averaged from 30 locations in the fine grid map) was calculated during the step 'USHRINK'. This step was otherwise analogous to 'P21SHRINK' described above. This solvent density became an input parameter to the reconstitution step (see figure 5.2) and was used to fill the empty cells in the final map. The solvent density calculated this way was usually negative and of about 2% of the highest peak of the map in magnitude.

As a check of all the parameters in the 'AVERAGE' and 'EXPAND' steps, the expression,

$$X = G_1 QPU_i G_2 G_1' Q' P' U' G_2'$$

can be evaluated algebraically. This represents the matrix part of the combined effect of the two steps. There should be one matrix U_i for which X is a simple scaling operation between the coarse grid and the fine one. This turned out to be the case for U_1. As an additional check, the overall grid to grid transformations that are printed out by GENERATE for the two steps were multiplied together numerically and found to produce the same diagonal matrix, again for

i = 1.

The ultimate test is to process actual maps. The fake-map program (chapter 4) was used to calculate a map containing only one atom in each icosahedral asymmetric unit of the virus. The intermediate grid map after averaging was found to contain a single atom in the correct place and the coarse grid final map was compared with the initial one: both had the same spherical shell of atoms, but the interpolation errors could be distinguished as background noise between the atoms of the latter. For the test with the grid sizes in the 12 Å ngstrom column of table 5.1, the r.m.s. interpolation noise level was about 1% of the height of the atomic peak.

5.3.4 Phase Combination.

The phase refinement cycle of figure 5.2 is completed by the program 'COMBINE', which was used exactly as described by Bricogne (1976). Since no heavy atom starting phases were used, all observed structure factors were assigned unit figures of merit, and the calculated phases were simply transferred to them directly. The program performs Wilson scaling of the calculated and observed reflections and produces a detailed report of the comparison between these.

It has been shown (unpublished) that the convergence of the refinement would be faster if coefficients of

$2F^{obs} - F^{calc}$ were used instead of F^{obs} in the calculation of the unaveraged map at each stage, but this was not used at the time of the original calculations.

5.4 Envelope Considerations.

The envelope is responsible for defining the boundary of the molecule within the crystal. It is conveniently represented in the computer as a map of 1-byte logical variables in the same format as electron density maps (Ten Eyck, 1973). Each cell is either 'True' or 'False' corresponding to points inside and outside the envelope respectively.

There are two ways that errors in the envelope are likely to bias the structure:

i) If the envelope is too small, regions of density will be systematically left out of the calculation on every cycle of phase refinement. In principle, the boundary should be at least half a wavelength (at the resolution limit) larger than the molecule itself to avoid the truncation leading to Fourier ripples in the density just inside the boundary.

ii) If the envelope is too large, on the other hand, there will be regions of overlap with itself, once folded into the unit cell of the crystal. Grid points in such regions will have two (or more) sources of density from different parts of the envelope. Only one of these can correspond to actual protein density, so this will

become artificially diluted with solvent density when the average is calculated. The effect is also to systematically remove density from the expanded map which will bias the final structure. A secondary effect of an oversized envelope is to increase the redundancy index U/V, thereby reducing the rate of convergence and increasing, slightly, the expected error in the derived phases.

Of these two effects, the first is by far the more serious in the expanded TBSV case because it directly and completely removes density from the averaged map. The second effect applies to regions of particle to particle contact in the reexpanded map. Corresponding to the crystallographic asymmetric unit, there are four such regions, none of which is exactly identical. If density of the expanded map becomes halved at each contact, the effect, when averaged, is only a reduction of 1/60th. Even if the four locations coincided in the averaged icosahedral asymmetric unit the effect would be at the 7% level and so would not lead to serious bias. The effect of a large envelope on U/V is minor since there is so much redundancy to spare.

The original native TBSV envelope was available, but as this was based upon the 5.5 Ångstrom structure (Winkler, et. al., 1977) and not the more recent information in the form of atomic coordinates, it was decided to create a new one based upon the latter. Moreover, a model of the expansion could be more easily incorporated as a modified

set of atomic parameters than as a set of transformations acting on an existing envelope. The definition of the shape of the envelope of the crystallographic asymmetric unit could be based upon the boundaries of the domains within it, but since there is very little empty space between the close-packed domains up to a radius of 145 Ångstroms in the native structure, this region may as well be considered solid, and a geometric boundary used instead. In the expanded structure, this region can be extended further without causing overlaps. A 60 Ångstrom radial cutoff on the viral interior allows room for any additional ordering to be visible immediately beneath the protein shell. The envelope outside 150 Ångstroms was then derived by constructing a 5 Ångstrom sphere around every pseudo atom position in a coordinate list supplied by one of the virus model expansion programs ('SWELL') described in chapter 4.

As described above, the program GENERATE can be used to transport envelopes from one frame to another. To do this, the program writes out a single list of grid indices in the destination coordinate frame by applying its grid transformation (defined with the same set of matrices described above) to every point inside the input envelope. After sorting, these are reconstituted into the final envelope by the program 'RECNV6'. For every set of fractional grid indices received by RECNV6, the surrounding eight integer grid points are set 'True', defining points within the final envelope. If this final envelope is then

used to create a map and the initial envelope is used for interpolation of this map, then all the density points needed are guaranteed to have been filled beforehand, without any wastage. This is the purpose of the envelope transport facility. The 'growing' property of envelopes during transport is a necessary feature of the scheme: if an envelope on grid A is transported to grid B and then back to A again, it will be larger by at least one grid point in all directions and maybe more, depending on the relative coarseness of the two grids.

Because of this envelope growing phenomenon, it is necessary to prepare the envelopes for a multiple stage interpolation in the opposite order from which they are to be used. In particular, the icosahedral asymmetric unit envelope that is calculated for expanded TBSV must be transported twice, first into the coarse grid 30-fold expanded envelope needed by 'EXPAND' (figure 5.5), then back onto the cubic intermediate 'U' grid, that it originally came from, for 'AVERAGE'. Only this way will all the images of the points inside the coarse grid envelope be completely surrounded in density in the 'U' map during 'EXPAND'. The whole procedure for the generation of envelopes for expanded TBSV is summarised in figure 5.6.

The second of the two envelope transport steps is straightforward, as it corresponds exactly to the step 'EXPAND' detailed above. GENERATE mode -3 is used since a

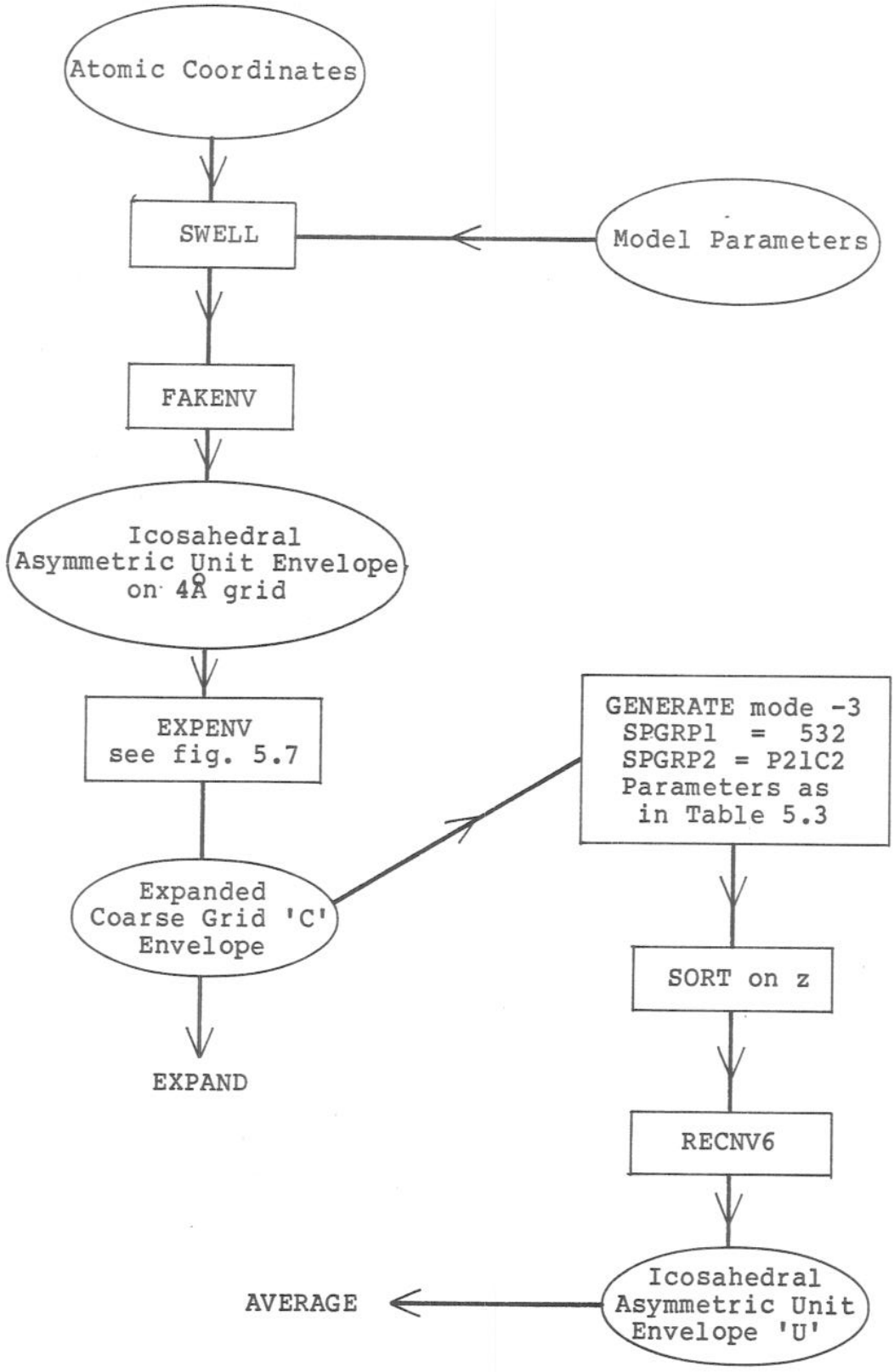


Figure 5.6. Flow diagram for the generation of envelopes. GENERATE mode -3 is for transport of envelopes with folding into the asymmetric unit of SPGRP1.

single set of indices is required and the envelope must be folded into the icosahedral asymmetric unit. The first step, called 'EXPENV', though, is a new one and is detailed in figure 5.7 and table 5.4. It is similar in function to 'AVERAGE' as it starts with an envelope in the icosahedral asymmetric unit and maps this onto a $P2_1$ monoclinic grid, but the latter is the coarse 'C' grid here, not the fine 'F' one. Moreover, the envelope is not folded into a crystallographic unit cell here but left isolated in space as a half-shell of the virus, as explained above. For this reason, GENERATE mode 3 is used with the $P2_1$ version of SPGRP1. The input parameters in table 5.4 apply to the coarse 'C' grid (G_1, Q and d) and the intermediate 'U' grid (G_2 and P). The U_i matrices for 'EXPENV' are the same as those used with 'AVERAGE' as the same mapping is being performed, only with a different grid spacing.

5.5 Computational Details.

Some of the relevant execution times and storage requirements for the phase refinement of expanded TBSV are quoted in table 5.5. Execution times vary a lot from installation to installation, so some details of the facilities used will be given to put these figures in perspective. Calculations were performed on a VAX 11/780 running VMS versions 2.1 and 2.2 (Digital Equipment Corporation, Maynard, Mass.). Three megabytes of semiconductor ('core') memory

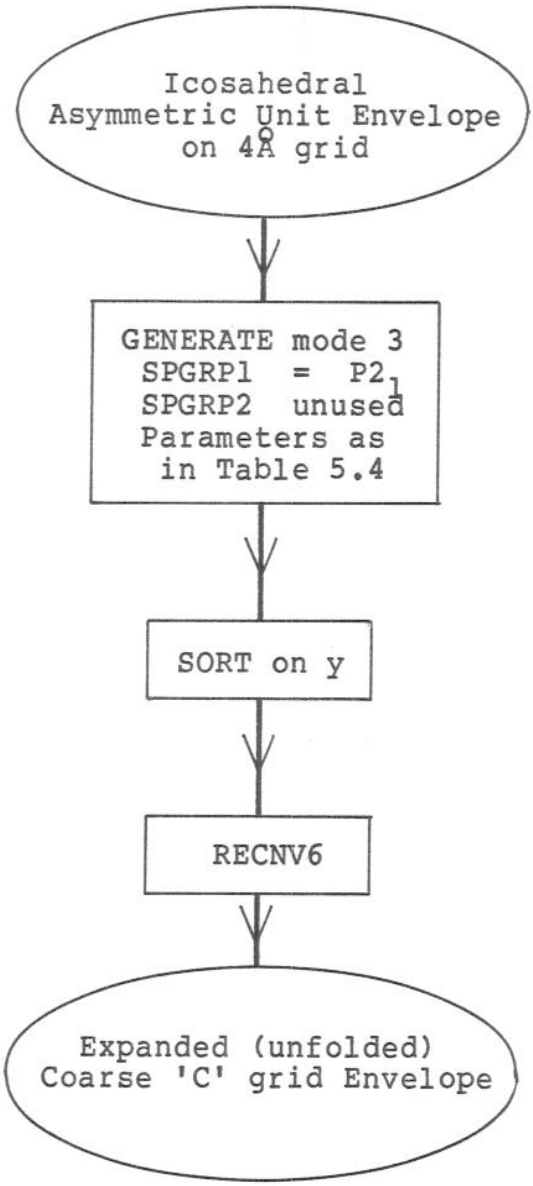


Figure 5.7. Flow diagram for the envelope expansion step, 'EXPENV'. Mode 3 of GENERATE is for envelope transport without folding.

$$G_1 = \begin{pmatrix} 96 & & \\ & 72 & \\ & & 64 \end{pmatrix} \qquad G_2 = \begin{pmatrix} 4.0 & & \\ & 4.0 & \\ & & 4.0 \end{pmatrix}$$

$$Q = \begin{pmatrix} 1/a & -\cot(\beta)/a & \\ & 1/b & \\ & & 1/c\ \sin(\beta) \end{pmatrix} \qquad P = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 1 \end{pmatrix}$$

$$d = (0, 0, 0)$$

$$U_{2i+3j+5k} = \begin{pmatrix} \cos\theta & -\sin\theta \\ & 1 \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & & \\ & -1 & \\ & & -1 \end{pmatrix}^i \begin{pmatrix} & & 1 \\ 1 & & \\ & 1 & \end{pmatrix}^j$$

$$x \begin{pmatrix} .309 & -.809 & .500 \\ .809 & .500 & .309 \\ -.500 & .309 & .809 \end{pmatrix}^k \qquad i = 0,1, \quad j = 0,1,2, \quad k = 0,..4$$

$$u_i = (0, 0, 0)$$

Table 5.4. Transformation matrices input to GENERATE to achieve the envelope expansion step, 'EXPENV', diagrammed in figure 5.7. Values are for 12Å limiting resolution calculations. G_1 and G_2 are modified for the 8Å calculation (see table 5.1).

were available and the system was configured so that relatively few processes were active at any one time, thus allowing up to 750 kilobytes of core to be accessed by a single process. The efficiency of an FFT program is dramatically improved when an entire section of the map can be held in core, since access to its elements occurs in an essentially random order, and large quantities of page 'swapping' would otherwise be needed. One section of the largest FFT performed was 450 x 324 elements, occupying 570 kilobytes of core, so this condition could be indeed fulfilled. Similarly, the interpolation is considerably more efficient when two sections can be held in core, but since the elements are 2-byte integers instead of 4-byte real variables, this criterion is the same as for the FFT. It is hard to estimate the additional time that would be required for the calculation in a smaller installation, but experience suggests at least a factor of two.

Direct access storage was provided by two 300 megabyte disk drives. Of this space, about half is maintained as 'scratch' storage which can become available to a single process on a temporary basis. The two most critical points of the refinement scheme (figure 5.2) from the point of view of storage are,

i) the fine grid interpolation. For this, the 'shrunk' map, as well as input and output indices must be on line simultaneously.

ii) the sorting of the associated indices. The 'workspace' needed for the sort is about three times the size of the output file, so the total requirement is equal to the size of the input file plus four times the size of the output file. By performing the sort in two passes, putting out half of the indices each time, the peak requirement can be reduced.

The times in table 5.5 required for sorting of indices were achieved by the BSORT program (C. Steele, unpublished), which made a more than ten fold improvement over the standard VAX SORT procedure for the largest sorts. The method uses a core-intensive bin sort (see chapter 4) rather than the traditional 'fast' sort (optimal sort/merge), taking advantage of the fact that there are relatively few possible values of the sort variable. The time required for such a sort scales in proportion to the number of records, whereas the 'fast' sort scales as N log N.

	12 Å ngstrom	8 Å ngstrom
Total time for cycle (without GENERATEs)	60 min	194 min
1st GENERATE and sort	17 min	62 min
Fine grid FFT	25 min	90 min
Peak on-line storage requirement	95 MB	144 MB
Size of 'F' map	22 MB	75 MB
Size of shrunk map	11 MB	38 MB
Associated indices	19 MB	48 MB
Associated densities	11 MB	29 MB
Size of 'U' map	1 MB	3 MB

Table 5.5. Critical computational parameters. All times are given in cpu minutes, the real time that would be taken, in principle, if no other processes were active. All storage sizes are in megabytes (MB), where 1 MB = 1,048,576 bytes, or 8,388,608 bits.