Version 2.5, Revised May, 1996

Translation Function

Program TF

Liang Tong[†]

INTRODUCTION

Translation searches are employed to determine the location of molecules (with known orientations) in a crystal unit cell. The search is normally formulated as a minimization or maximization of certain indicators by varying the positional parameters of the molecules in the unit cell. Most indicators involve a comparison between the observed and the calculated structure factor amplitudes. For example, the difference between the observed and the calculated structure factor amplitudes (or the R factor) can be minimized. Alternatively, the correlation between the observed and the calculated structure factor amplitudes can be maximized.

I. THE R FACTOR AND THE CORRELATION COEFFICIENT

The crystallographic R factor is often used as an indicator in translation searches. It is a measure of the percentage difference between the observed and the calculated structure factor amplitudes,

$$R_F = 100 \times \frac{\sum\limits_h \left| F_h^{obs} - k_F F_h^{calc} \right|}{\sum\limits_h F_h^{obs}}. \tag{1}$$

A similar R factor can be defined based on the square of the structure factor amplitudes, *i.e.*, an R factor based on intensity,

$$R_I = 100 \times \frac{\sum\limits_h \left| I_h^{obs} - k_I I_h^{calc} \right|}{2 \sum\limits_h I_h^{obs}}. \tag{2}$$

In the equations above k_F and k_I are scale factors that bring the observed and the calculated structure factors to the same level,

$$k_F = \frac{\sum\limits_h F_h^{obs}}{\sum\limits_h F_h^{calc}}.$$

The scale factors are calculated in shells of equal reciprocal volume. This will compensate for any differences in the temperature factors between the observed and the calculated structure factor amplitudes.

[†] Please send comments and bug reports to : b4w@cc.purdue.edu.
Reference: L. Tong *J. Appl. Cryst.* **26**, 748–751, (1993).

The R factor values are very sensitive to the position and the orientation of the molecules in the unit cell. The correlation coefficients between the observed and the calculated structure factors are somewhat less sensitive. Hence they may allow for larger errors in the reflection data as well as in the orientational parameters of the molecules. As is the case with the R factors, the correlation coefficients can be defined based on the amplitude or the intensity of the reflections.

$$\begin{aligned} CC_F &= \frac{\sum\limits_h (F_h^{obs} - < F_h^{obs} >)(F_h^{calc} - < F_c^{calc} >)}{\sqrt{\sum\limits_h (F_h^{obs} - < F_h^{obs} >)^2 \sum\limits_h (F_h^{calc} - < F_h^{calc} >)^2}} \\ &= \frac{\sum\limits_h F_h^{obs} F_h^{calc} - \sum\limits_h F_h^{obs} \sum\limits_h F_h^{calc} / N}{\sqrt{\left[\sum\limits_h (F_h^{obs})^2 - \big(\sum\limits_h F_h^{obs}\big)^2 / N\right]\left[\sum\limits_h (F_h^{calc})^2 - \big(\sum\limits_h F_h^{calc}\big)^2 / N\right]}} \end{aligned} \tag{3}$$

$$CC_I = \frac{\sum\limits_h I_h^{obs} I_h^{calc} - \sum\limits_h I_h^{obs} \sum\limits_h I_h^{calc} / N}{\sqrt{\left[\sum\limits_h (I_h^{obs})^2 - \big(\sum\limits_h I_h^{obs}\big)^2 / N\right]\left[\sum\limits_h (I_h^{calc})^2 - \big(\sum\limits_h I_h^{calc}\big)^2 / N\right]}}. \tag{4}$$

In the equations N is the number of reflections that are used in the calculation. Unlike the R factors, the correlation coefficients do not depend on the overall scale factor between the observed and the calculated structure factors.

The region of the unit cell that should be covered for a translation search with the above indicators does not normally correspond to the asymmetric unit of the space group. Because the molecule to be searched has a determined orientation, it can only reside in one of the asymmetric units in the unit cell. Lacking a knowledge as to which asymmetric unit the molecule is occupying, the entire unit cell would need to be searched. However, most space groups have alternative origins, which means the position of a molecule in the unit cell can only be determined to within certain sets of translations (the Cheshire group). For example, in space group $P2_12_12_1$, there are eight alternative origins, at $(0,0,0)$, $(1/2,0,0)$, $(0,1/2,0)$, $(0,0,1/2)$, $(1/2,1/2,0)$, $(1/2,0,1/2)$, $(0,1/2,1/2)$, and $(1/2,1/2,1/2)$. This implies the region that should be searched to locate a molecule need only be 1/8 of the volume of the unit cell (for example, $[1/2,1/2,1/2]$). Once the first molecule has been located, the origin is determined as well. The search for subsequent molecules will need to cover the entire unit cell.

II. CALCULATION OF STRUCTURE FACTORS

In order to calculate R factors and correlation coefficients for a translation search, structure factors need to be calculated for molecules at different positions in the unit cell. The orientations of the molecules have been determined beforehand with rotation searches (*e.g.*, with the program GLRF). Therefore, only the positional parameters of the molecules need to be varied.

The structure factor equation can be written as a double summation — first over the atoms in one asymmetric unit of the unit cell and then over all the asymmetric units,

$$F_h^{calc} = \sum_n \sum_j f_j e^{2\pi i h([T_n]x_j + t_n)}, \tag{5}$$

where j goes over all the atoms in the asymmetric unit and n goes over all the asymmetric units. The nth crystallographic symmetry operator is given by

$$x' = [T_n]x + t_n,$$

where $[T_n]$ is the rotational component and t_n the translational component of the symmetry operator.

For simplicity, consider the case where there is only one molecule in the asymmetric unit. In the translation search, the molecule will be placed at different positions in the unit cell,

$$x_j = x_j^0 + x_0, \tag{6}$$

where x_0 is a translation vector which relates the molecule located at x_j with that located at x_j^0. Substituting x_j in Eq. (6) into Eq. (5),

$$\begin{aligned} F_h^{calc} &= \sum_n \sum_j f_j e^{2\pi i h\left([T_n](x_j^0 + x_0) + t_n\right)} \\ &= \sum_n \sum_j f_j e^{2\pi i h([T_n]x_j^0 + t_n)} e^{2\pi i h[T_n]x_0} \\ &= \sum_n e^{2\pi i h[T_n]x_0} \sum_j f_j e^{2\pi i h([T_n]x_j^0 + t_n)} \\ &= \sum_n f_{h,n} e^{2\pi i h[T_n]x_0}, \end{aligned} \tag{7}$$

where $f_{h,n}$ is the structure factor calculated based on the nth symmetry-related molecule only,

$$f_{h,n} = \sum_j f_j e^{2\pi i h([T_n]x_j^0 + t_n)}. \tag{8}$$

Therefore, the summation over the atoms in the structure factor calculation, a rather time-consuming process, need to be performed only once, for the molecule at a reference position (x_j^0). Subsequent structure factor calculation after translation of the molecule from the reference position is no longer dependent on the number of atoms present in the unit cell (Eq. (7)). The reference position is usually chosen with the center of the molecule at $(0, 0, 0)$. Then the translation vector that is determined from the translation searches will define the center of the molecule in the unit cell.

Equation (7) can be generalized to allow for the presence of other molecules that are to remain stationary during the translation search,

$$F_h^{calc} = A_h + \sum_n f_{h,n} e^{2\pi i h[T_n]x_0}, \tag{9}$$

where A_h is the contribution from the stationary molecules. This formulation is useful if there are more than one molecules in the asymmetric unit. The position of one of the molecules can be determined first and the model is then input as a stationary molecule for the search of the position of the next molecule.

III. PATTERSON CORRELATION TRANSLATION FUNCTION

Besides the R factor and the correlation coefficient, another commonly-used translation search indicator is the correlation between the observed and the calculated Patterson maps. Rotation functions are based on the overlap of only a subset of the interatomic vectors in the Patterson map, *i.e.* only the self-vectors within each crystallographically-unique molecule. The correct orientation and position of a molecular structure in the crystal unit cell should lead to the maximal overlap of both the self and the cross vectors, *i.e.* maximal overlap between the observed and the calculated Patterson maps throughout the entire unit cell,

$$\begin{aligned}
PC &= \int_\Omega P^{obs}(u) P^{calc}(u) du \\
&= \int_\Omega \sum_h (F_h^{obs})^2 e^{-2\pi i h u} \sum_p (F_p^{calc})^2 e^{-2\pi i p u} du \\
&= \sum_h \sum_p (F_h^{obs})^2 (F_p^{calc})^2 \int_\Omega e^{-2\pi i (h+p) u} du \\
&= \sum_h (F_h^{obs})^2 (F_h^{calc})^2. \qquad (10)
\end{aligned}$$

Note that the integral $\int_\Omega e^{-2\pi i(h+p)u} du$ is 1 when $h+p=0$ and 0 otherwise.

The calculated structure factor is a function of the translation vector x_0 (Eq. (9)). Combining Eqs. (9) and (10),

$$\begin{aligned}
PC(x_0) &= \sum_h (F_h^{obs})^2 (A_h + \sum_n f_{h,n} e^{2\pi i h [T_n] x_0})(A_h^* + \sum_n f_{h,n}^* e^{-2\pi i h [T_n] x_0}) \\
&= \sum_h (F_h^{obs})^2 \big(A_h^2 + A_h \sum_n f_{h,n}^* e^{-2\pi i h [T_n] x_0} + A_h^* \sum_n f_{h,n} e^{2\pi i h [T_n] x_0} \\
&\quad + \sum_m \sum_n f_{h,m} f_{h,n}^* e^{-2\pi i h ([T_n]-[T_m]) x_0} \big) \\
&= \sum_h (F_h^{obs})^2 A_h^2 + \sum_h \sum_n (F_h^{obs})^2 f_{h,n}^2 \\
&\quad + \sum_h \sum_n (F_h^{obs})^2 A_h f_{h,n}^* e^{-2\pi i (h[T_n]) x_0} \\
&\quad + \sum_h \sum_n (F_h^{obs})^2 A_h^* f_{h,n} e^{-2\pi i (-h[T_n]) x_0} \\
&\quad + \sum_h \sum_m \sum_{n \neq m} (F_h^{obs})^2 f_{h,m} f_{h,n}^* e^{-2\pi i [h([T_n]-[T_m])] x_0} \qquad (11) \\
&= \sum_h \sum_n (F_h^{obs})^2 A_h f_{h,n}^* e^{-2\pi i (h[T_n]) x_0} \\
&\quad + \sum_h \sum_m \sum_{n < m} (F_h^{obs})^2 f_{h,m} f_{h,n}^* e^{-2\pi i [h([T_n]-[T_m])] x_0}. \qquad (12)
\end{aligned}$$

The first two terms in Eq. (11) contribute a constant to the correlation and hence are omitted from further calculation. It can be seen from Eq. (11) that reflections (h, k, ℓ) and ($-h$, $-k$, $-\ell$) contribute equally to the correlation. The translation function is therefore a Fourier transform, with $h[T_n]$ and $h([T_n]-[T_m])$ as the indices and the coefficients of this transform are defined by Eq. (12).

Sometimes it may be desirable to use a subset of the symmetry-related molecules in the translation function calculation. For example, in space group $P4_12_12$, molecules at (x, y, z) and $(-x, -y, 1/2+z)$ can be used in the calculation. The translation function Fourier transform will have 'reflections' with indices $(2h, 2k, 0)$, hence only the $z = 0$ section need to be calculated. More importantly, the resolution of the translation function transform in this case is twice that of the reflection data used in the structure factor calculation.

The intra-molecular vectors can be removed from the observed Patterson map by subtracting the structure factors calculated from individual symmetry-related molecules (Eq. (8)) from the observed structure factors,

$$(F_h^{obs})^2 - k\sum_n f_{h,n}^2,$$

where k is a weighting factor on the intra-molecular vectors.

It has been suggested that only the phases, rather than both the phases and the amplitudes, of the calculated structure factors be used in the translation function calculation (Eq. (12)).

IV. ELECTRON DENSITY CORRELATION TRANSLATION FUNCTION

If an atomic model needs to be placed in an electron density map that has been obtained through other methods (*e.g.*, the MIR method or partial model phases), the electron density correlation translation function (also known as the phased translation function) can be used. It is defined as the product of the observed and the calculated electron density functions integrated over the unit cell. The derivation of the function is similar to that of the Patterson correlation translation function.

$$\begin{aligned} EDC(x_0) &= \int_\Omega \rho^{obs}(x)\rho^{calc}(x)dx \\ &= \int_\Omega \sum_h F_h^{obs} e^{-2\pi i h x} \sum_p F_p^{calc} e^{-2\pi i p x} dx \\ &= \sum_h F_h^{obs} (F_h^{calc})^* \\ &= \sum_h F_h^{obs} \big(A_h^* + \sum_n f_{h,n}^* e^{-2\pi i h [T_n] x_0}\big) \\ &= \sum_h \sum_n F_h^{obs} f_{h,n}^* e^{-2\pi i (h[T_n]) x_0}. \end{aligned} \tag{13}$$

Therefore the electron density correlation translation function is also a Fourier transform.

DESCRIPTION OF INPUT COMMANDS

All input commands to the program are keyword-based and free-formatted. The input parser converts lower case characters to upper case so the program commands are not case dependent (though the cases for the title and the file names are kept by the program). Each input line can contain at most 80 characters. The following characters are recognized as delimiters between words – a space, a comma, a tab, and an equal sign. (More delimiters can be implemented by inserting them in the array SPACER in subroutine PARSER, and change the variable NSP at the same time.) Comments in the input can be introduced by using the COMMent command or using the special character "!" — in the input parser, all characters in the input line beginning at the "!" are ignored.

Presently, the program uses six logical I/O units, labelled by the variables LIN, LOG, LPRT, LOBS, LMAP and LSCRCH, which are initialized to be 5, 6, 3, 1, 4 and 2, respectively. (The default values of most of the input variables are initialized in subroutine INITSS). All input commands are read from unit LIN. The output of the program will be written to unit LPRT. A file name can be specified for this print file by the PRINt command. The reflection data (if any) is read from unit LOBS. The translation function map is written to and read from unit LMAP. LSCRCH is a scratch file and is used for echoing the input commands at the end of the output file. Program messages are written to unit LOG. They are informational messages, warning error messages (*Warning*), or fatal error messages (*Fatal*). Each message is preceded by the name of the subroutine that produced it.

What follows is a description of all the input commands that are currently supported by the program. A new command can be incorporated by inserting the command name in array COMMND, in subroutine INITSS, and by inserting a segment in subroutine INTPRT that defines the input parameters (if any) of the command. The program identifies each command by its first four letters, although more can be input for readability. In the following listing, the four-letter command keyword will be given in **bold face**, the names of input variables associated with the command will be given in UPPER case. The expected length of the character variables are also given (for example, ORDOR*5 means ORDOR is a character variable with 5 letters). Variables whose names begin with the letter 'Q' are logical variables. A 'True' or 'False' input is expected for these variables. The defaults (if any) of the variables are given in square brackets. These default values will generally be used by the program if the command is omitted from input, or if a value of 0 is input to the command.

The input commands can be divided into two categories. The first category consists of those that are carried out immediately. These commands are marked by a "*" in the following write-up. **The input order of these commands is therefore very important.** Commands in the second category are executed only after a 'STOP' or an end-of-file is encountered on unit LIN. Their input orders are therefore not important.

I. GENERAL INPUT COMMANDS

COMMent none [none]

This can be used to incorporate comments in the command input file. The entire input line is simply ignored by the program. The other way to introduce comments is through the use of the special character "!" (see above). The difference between the two is that comments incorporated by this command will be echoed at the end of the program print file.

***PRIN**t-File PRTFIL*72 [TF.PRT]

This opens the output file of the program (associated with the logical unit LPRT). To make sure the output from all the commands are written to this file, this command should be specified at the beginning of the program input.

The LOG unit is meant to be associated with the system output (terminal screen in interactive mode and job log file in batch mode).

The program will open all the output files with the status UNKNOWN. Existing files with the same names will be over-written.

***STOP** none [none]

This means the program should stop reading from the command input file and start the calculation. Otherwise, the program will read until the end-of-file on unit LIN before initiating the calculation.

***TITL**e ATITLE*132 [PROGRAM TF]

This provides a title for the current run of the program. It will be placed at the beginning of each output page. The program will insert the current date and time at the beginning of ATITLE (characters 2-18), and the program will append a version number identifier at the end of ATITLE (characters 111-130).

II. DEFINITION OF A FEW CONVENTIONS

***EULE**r-Angle EULER*3 [ZYZ]

This specifies the definition convention of Eulerian angles. Two conventions are currently supported — ZXZ and ZYZ, corresponding to rotating the molecule around the Cartesian Z axis (θ_3), then around the X (and Y, respectively) axis (θ_2), and finally around the Z axis (θ_1). ZXZ is the convention described in M. G. Rossmann & D. M. Blow *Acta. Cryst.* **15** 24, (1962). ZYZ is used in program MERLOT (P. M. D. Fitzgerald, *J. Appl. Cryst.* **21** 273, (1988)). The angles should be input to the program in the following order — $\theta_1, \theta_2, \theta_3$, and the program outputs the angles in the same order.

WARNING: The matrix used in this program is the transpose of the matrix as printed in Table 1 of the paper by Rossmann & Blow. Therefore, if you are inputing a set of angles from other programs, be sure the programs are using the matrices the same way! The same goes for the command POLAr (see below).

***POLA**r-Angle POLAR*3 [XZK]

This specifies the polar angle definition convention. Two conventions are currently supported – XYK and XZK. In both cases, ϕ is the angle from the Cartesian X axis. In convention XYK (as defined in Rossmann & Blow), ψ is the angle from the Y axis. In convention XZK (as in MERLOT), ψ is the angle from the Z axis. κ is the rotation around the axis defined by ϕ and ψ. The angles should be input to the program in the following order – ϕ, ψ, κ, and the program outputs the angles in the same order.

The XZK convention should be used for most calculations. The XYK convention can be used for monoclinic space groups in the *b*-unique setting.

***ORTH**ogonalization ORDOR*5 [AXABZ]

This specifies the orthogonalization convention that should be used by the program. Three conventions are currently supported – BYBCX, CZBCX, and AXABZ. In convention BYBCX, the real space b axis coincides with the Cartesian Y axis, and $b \times c$ (or a^*) coincides with the X axis (this is first defined in the paper by Rossmann & Blow). In convention CZBCX, the real space c axis coincides with the Cartesian Z axis, and $b \times c$ (or a^*) coincides with the X axis (this is IPER=1 in MERLOT). In convention AXABZ, the real space a axis coincides with the Cartesian X axis, and $a \times b$ (or c^*) coincides with the Z axis (this is used in FRODO's helper SAM, PROTEIN, and X-Plor. This is also option NCODE=1 in program ALMN). Other conventions can be incorporated as well, by inserting the codes in subroutine GTOMDM, which calculates the orthogonalization matrix from the cell parameters. The deorthogonalization matrix is calculated as the inverse of the orthogonalization matrix.

The AXABZ convention should be used for most calculations. The BYBCX convention was designed for monoclinic space groups in the *b*-unique setting to align the crystal b axis along the Cartesian Y axis. The AXABZ convention will maintain this alignment and should therefore be used for monoclinic space groups as well.

III. COMMANDS THAT DEFINE THE SEARCH ATOMIC MODEL.

***MODE**l-cell (CELL(i, 2), i=1, 6) [1, 1, 1, 90, 90, 90]

This defines the unit cell in which the model is placed. This is used only if deorthogonalized atomic coordinates (see command COOR TYpe) are being input. The input coordinates will be orthogonalized according to the current ORDOR (see command ORTHogonalization) and then deorthogonalized (with the same ORDOR) into the crystal unit cell. The program internally saves fractional deorthogonalized coordinates in the crystal cell (as defined by command CELL). Therefore, the dimensions of the crystal unit cell must be defined before the input of atomic coordinates.

***COOR**dinate OPTION*2 [none]

This command defines the handling of the atomic coordinates. OPTION is a two-letter keyword that defines the action that should be taken. The currently-supported options are listed alphabetically below. The program can store a maximum of MAXCOR molecules. Most of the commands below apply only to molecule #1, which can then be STored or REtrieved.

ADd-residue-number — N.

This specifies a number by which the residue numbers in molecule #1 should be incremented. The residues must be numbered either numerically or with one heading or trailing letter. For example, adding 300 to the residue numbers 225, 225A, and A225 will produce 525, 525A, and A525, respectively.

APpend — N.

This will append the information present for molecule #1 to molecule N. Molecule #1 is then initialized. The flag (see option FLag) for the new molecule will be taken from that for molecule N. The center of the new molecule will be calculated over all the atoms.

BIncrement — BINCRM.

This specifies an increment value to the atomic temperature factors as they are read in. BINCRM defaults to 0.

BOverall — BOVRAL.

This defines an overall value that replaces the atomic temperature factor values as they are read in. BOVRAL defaults to 0, meaning the original B values from the input file will be kept.

BZero — BZERO.

This defines a temperature factor value for those atoms with an input temperature factor of 0. BZERO defaults to 30.0Å^2.

CEnter — (CENTER(i), i=1, 3), KORTYP*2.

This recenters molecule #1 to the specified position. If no center position is given in the command, the program will place the center at (0, 0, 0). The interpretation of the values of CENTER depends on the parameter KORTYP, or if it is not given, CORTYP (option TYpe).

CHain — CHNNAM*1.

This assigns a chain name to the atoms in molecule #1.

COpy — M, N.

This copies the information for molecule number M to number N.

FCalc — CALFIL*72, CALFMT*40.

This specifies that the calculated structure factors for molecule #1 should be read from file CALFIL, with format CALFMT. The default is that the structure factors will be calculated within the program. The file should contain h, k, l, F and ϕ (in degrees). The default for CALFMT is (3I4, 2F8.2). If this option has been selected, the molecule must be defined by specifying a flag (FLag option), the center (CEnter option), and possibly the size (SIze option). There should not be any rotation or translation operations on this molecule. There should not be any atomic coordinates input for this molecule.

If the molecule is 'T' flagged (see option FLag), the structure factors should be calculated for the molecule in the correct orientation and position in the crystal cell and need only cover one asymmetric unit. The program will transform the indices to the 'correct' asymmetric unit or expand them to $P1$ if necessary.

If the molecule is 'R' flagged, the structure factors should be calculated for the molecule in the correct orientation in a $P1$ cell with the same cell dimensions as those for the crystal cell. The structure factors should therefore cover a hemisphere. The center of the molecule can be chosen arbitrarily, though it is oftentimes most convenient to place it at $(0, 0, 0)$.

WARNING: If calculated structure factors are being input for more than one molecule, be sure that they are on the same scale.

FLag — CORFLG*1.

This assigns a flag to molecule #1. CORFLG='T' defines a molecule with known positional (and orientational) parameters. CORFLG='R' defines a molecule with known orientational parameters only, the positional parameters of which is to be determined. Currently, only one molecule can have the flag 'R'.

FOrmat — FMTTYP*1, CORFMT*40.

This specifies the coordinate format type for the input/output coordinate file. Supported types are – 'P'DB for Protein Data Bank, 'W'H for Wayne Hendrickson, 'D'iamond, and 'U'ser-specified format. If user-specified format has bee selected, CORFMT should be input corresponding to the following input list – RESTYP*4, RESNAM*4, ATMNAM*4, X, Y, Z, B. Otherwise, CORFMT is not necessary.

INput — CORFIL*72.

This inputs the atomic coordinates in file CORFIL to molecule #1. The crystal cell parameters (see command CELL) must be defined before the input of any atomic coordinates. If there are atoms already present in the molecule, the input coordinates will be appended to the existing ones. The format of the file is specified by FMTTYP and CORFMT (option FOrmat). The coordinate type is defined by CORTYP (option TYpe). The occupancies are assumed to be 1. The chain name and insertion code fields in the PDB file are supported. The coordinates thus input can be COpied to or STored in other molecules. The program can save a maximum of MAXCOR sets of coordinates.

OUtput — OUTFIL*70.

This outputs the atomic coordinates in molecule #1 to file OUTFIL. The format of the file is specified by FMTTYP and CORFMT. The coordinate type is defined by CORTYP.

PEmute — L, M, N.

The coordinates will be permuted for input or output. The COORdinate TYpe must be 'OA' as the permutation is applied to orthogonal Ångstrom coordinates. For example, specifying '2 3 1' will make the new X equal the old Y, the new Y equal the old Z, and the new Z equal the old X. The permuation, if any, should be defined before the 'COOR INput' or the 'COOR OUtput' command.

The default is no permutation will be performed.

REtrieve — N.

This retrieves the atomic parameters stored in molecule number N to number 1. Molecule number N is then initialized.

ROtate — (ROTANG(i), i=1, 3), ANGTYP*1, (ROTCEN(i), i=1, 3).

This will rotate the atoms in molecule #1 around a certain position by the specified angles. ANGTYP specifies that ROTANG should be interpreted as either 'E'ulerian or 'P'olar angles. The rotation will be carried out around the center ROTCEN. If ROTCEN has been omitted, rotation will be carried out around the center-of-mass of the molecule. The interpretation of ROTCEN depends on the current definition of CORTYP (see option TYpe). Internally, the stored fractional coordinates are orthogonalized and deorthogonalized as specified with the ORDOR parameter.

SAlt-bridge — DSALT.

Interactions between a pair of charged residues (Asp, Glu, Lys and Arg), and their nearest charged or polar residue neighbors, will be identified. The distance cut-off, default 5Å, can be input. If the space group symmetry and the cell dimensions are input, the program will check the symmetry-related residues as well.

SIze — RADIUS.

This defines the radius (in Å) of molecule #1. If the radius of a molecule is not defined with this command, the program will set it to SIZMOL (see command SIZE). Therefore, the molecular sizes defined with this command takes precendence over that defined with the SIZE command. Note that this radius should be chosen rather conservatively, to avoid allowed grid points being skipped over with this criterion.

STore — N.

This stores the atomic parameters in molecule #1 in number N. Molecule #1 is then initialized. The program can save a maximum of MAXCOR sets of coordinates.

Most of the operations (options ADd, CEnter, CHain, FCalc, FLag, INput, OUtput, ROtate, SIze, and TRanslate) apply only to molecule #1. These operations are sufficient to completely define a molecule. If more than one molecules are involved in the calculation, each molecule should be individually defined and the information STored into another molecule with this option.

SWap — M, N.

This swaps the information for molecule number M and number N.

TRanslate — (VECTOR(i), i=1, 3), KORTYP*2.

Atoms in molecule #1 will be translated by the specified vector. The type of the vector is given by KORTYP. If KORTYP is not given, the current CORTYP (see option TYpe) will be used. Note that KORTYP as defined here will be re-initialized once the translation operation is complete.

TYpe – CORTYP*2.

This specifies the type of the input coordinates. Supported types are 'DF' for deorthogonalized fractional, 'DA' for deorthogolized Ångstrom, and 'OA' for orthogonalized Ångstrom coordinates.

IV. COMMANDS THAT DEFINE THE CRYSTAL

***CELL**-Parameters (CELL(i, 1), i=1, 6) [1.0, 1.0, 1.0, 90.0, 90.0, 90.0]

This inputs the crystal unit cell parameters. The parameters can be either in real or reciprocal space. The angles can be either in degrees or in cosines.

SYMMetry [none]

This inputs the space group symmetry operators of the crystal, in the same format as the International Table. Positions related by centering should be input by the LATTice command. Positions related by inversion or mirror planes are not supported by the program. If no symmetry card is given, the program defaults to $P1$. The identity operation is assumed and need not be specified.

The space group symmetry can also be specified by giving the symbol of the space group (for example P212121) or the number of the space group (for example 19). For monoclinic space groups, only the *b*-unique setting is supported. For rhombohedral space groups, only the hexagonal setting is supported. If you want the other settings, input the symmetry operators explicitly.

LATTice-type LATICE*1 [P]

This specifies the lattice centering type. Supported types are P, A, B, C, F, I, and R. For example, if I centering is desired, input ‘LATT I’.

FOBS-File OBSFIL*72 [FOBS.DAT]

This specifies the name of the formatted file holding the crystal reflection data. The file should contain h, k, l, F, and optionally σ_F for each reflection. If option EOEC is chosen from command FUNCtion, then phase angles (in degrees) are also expected to be input. The format of the file is specified by the command FORMat. The file should be terminated by the end of file, not by an end-of-file record. The maximum number of reflections that can be saved by the program is given by the MAXREF parameter.

FORMat OBSFMT*40 [(3i4, 2f8.2)]

This specifies the format of the reflection file, for the following input list – h, k, l, F, σ_F. The enclosing parentheses in the format are optional. Note that a format must be specified for σ_F even if the file does not contain σ_F values.

FREJection SIGCUT, FCUTLO, FCUTHI [2.0, 1.0, 0.0]

This specifies the cut-off criteria for screening the reflection data. Reflections with F(input) < SIGCUT × σ_F(input), or with F(input) < FCUTLO, or with F(input) > FCUTHI will be rejected from the calculation. If FCUTHI is 0, no high cut-off will be carried out. F(input) and σ_F(input) are the values as read from the input reflection file. The program will print out how many reflections are removed by each of the above criteria, so reasonable inputs can be selected accordingly.

POWEr POWER [2.0]

The intensity of reflections will be calculated as F(input)**POWER. Therefore, if intensity data is being input, set POWER to 1.0.

V. STRUCTURE FACTOR CALCULATION

ATOM-type ATMTYP*4, (ATOMFF(i), i=1, 9) [none]

This defines the atomic scattering factors, as defined in Table 6.1.1.4 of Volume C of the International Tables. A question mark ‘?’ can be used as a wild card in ATMTYP and will match to any letter or digit. The scattering factors for the following atoms have been included and should not be specified — C??? for carbon, N??? for nitrogen, O??? for oxygen, S??? for sulfur, P??? for phosphorus, MG?? for magnesium, CA+? for calcium, CU?? for copper, F??? for florine, FE?? for iron, CL?? for chlorine, BR?? for bromine, and I??? for iodine. Atom scattering types that are specified by the user are appended to those already supported by the program. The program checks the scattering types of atoms starting from the last atom type that is defined. Therefore, atoms having names ‘FE ’ and ‘F   ’ can be assigned different scattering types. Atoms that do not have identifiable scattering types will be ignored from the structure factor calculation. The maximum number of scattering types is given by the parameter MAXTYP.

DIREct-summation QDRECT [F]

If QDRECT is true, the structure factors will be calculated by direct summation over all the atoms. If QDRECT is false, then structure factors will be calculated by the fast Fourier transform (FFT) method. As the FFT method is much faster than direct summation, it should be used in most cases. The program performs the FFT in memory, and error messages will be issued if any of the array limits are exceeded. The FFT routines are courtesy of Dr. L. F. Ten Eyck and Dr. D. McRee.

TABLe-lookup QTABLE [T]

This is relevant only if structure factors are being calculated by direct summation. If QTABLE is true, lookup tables will be generated for ‘sin’, ‘cos’, and ‘exp’ functions to speed up the calculation.

VI. SPECIFICATION OF SEARCH PARAMETERS

FUNCtion OPTION*4 []

This specifies what type of search or calculation the program should carry out. OPTION is a four-letter keyword that defines the specific functions.

ANOMalous — QANOM.

This specifies whether the difference coefficients are anomalous differences. The default for QANOM is false. This is used only for the DIFFerence option, below.

CONTact —

This sets QCONT to true and specifies that the contacts between molecules with 'T' flags should be checked. Default value for QCONT is false. The program first selects the C_α atoms from molecules with 'T' flags based on NCAMOD (see option DCUToff) and collects all the C_α atom pairs that are separated by less than DCACUT. Then the 50 closest contacts will be listed. If the 'T' flagged molecules are assigned different molecular numbers, contacts between these 'T' flagged molecules themselves will be checked out as well.

CUTOff — CUTOFF.

This defines the selection of 'large terms'. These strong reflections will be used in the search calculations. Reflections with I > CUTOFF × I(average) will be saved as the large terms, where I(average) is calculated in NSHELL shells of equal reciprocal volume. The suggested CUTOFF value is 0.5, which is also the default.

DCUToff — DCACUT, NCACUT, NCAMOD.

DCACUT (in Å, default 3.0) specifies the contact distance allowed between two C_α atoms. NCACUT (default 8) specifies how many such C_α–C_α contacts are allowed between pairs of molecules. NCAMOD (default 1) specifies that every NCAMOD'th C_α atom will be used in the calculation. These values will affect the options CONTact, PACKing, RFMAp, RFPEak, and RFPC of the command FUNCtion. To speed up calculation for the latter three cases, DCACUT should be set to 2 and NCACUT should be set to 5 or less. For nucleic acid structures, the P, N1, C4 and N9 atoms of each nucleotide are saved for packing checks.

DIFFerence — DIFFIL*72, DIFFMT*40.

This specifies the name for the file containing the difference coefficients, which could be either isomorphous or anomalous differences (see option ANOMalous). This file is used only when combined molecular replacement calculations are being performed. The program will calculate, for each possible solution, the correlation coefficient, the R factor, and the rms value for the highest peak in the difference map. The solutions can be sorted on either of the three indications (see option RFSOrt). Note that the program will only look for peaks, so please ensure that the signs of the differences are correct, F(derivative) - F(native) for isomorphous and F(+) - F(-) for anomalous differences (do NOT use the absolute values!).

ECEC —

This sets QECEC to true and specifies the program will carry out a phased translation function, similar to EOEC. The difference is that here the phase information will be taken from molecules with T flags.

EOEC — (MOLCUL(i), i=1, NSYM)

This sets QEOEC to true and specifies that electron density correlation translation function coefficients will be calculated by the program. Default value for QEOEC is false. A set of phase angles must have been input in the reflection file (see command FOBS). The correlation will be calculated based on space-group symmetry-related molecule numbers as specified by MOLCUL. The identity symmetry operation is always assigned the number 1. If no molecule numbers are specified, then all the symmetry-related molecules will be used in the calculation. The translation function coefficients will be written to file EOEC.DAT, which will contain h, k, l, F, ϕ. The maximum F value is scaled to 1400. The reflections can be input to a Fourier summation program. The resulting peak in the map defines the vector by which the input search model with the 'R' flag should be translated. The program will also attempt to carry out the Fourier transform directly (see option POPC below for more information).

F1F1 —

This sets QF1F1 to true and specifies that correlation coefficients based on amplitude will be calculated during the search. Default value for QF1F1 is false.

F2F2 —

This sets QF2F2 to true and specifies that correlation coefficients based on intensity will be calculated during the search. Default value for QF2F2 is false.

FOFC —

This sets QFOFC to true and specifies that R factors based on amplitude will be calculated during the search. Default value for QFOFC is false.

HAND — QHAND.

This is applicable only if the EOEC option is used. This causes the hand of the reflection phases to be flipped upon input. This is useful when MIR phases of unknown hand are used for EOEC translation function. In this case, two separate calculations need to be done, one in each hand.

IOIC —

This sets QIOIC to true and specifies that R factors based on intensity will be calculated during the search. Default value for QIOIC is false.

PACKing-calculation —

This sets to QPACK to true and specifies that the packing of the molecule in the unit cell should be explored. Default value for QPACK is false. The program represents the molecule as a sphere with radius SIZMOL (see command SIZE). Then it places the sphere on the grid defined by the SLIMits commands. A grid point where the sphere comes into contact with a symmetry-related one will be flagged by '*'. The program will output to the print file a map showing the results of this packing search. The section of the map can be defined by the SECTion command, and the program will provide reasonable defaults otherwise. If atomic coordinates for molecules with 'T' flags are available, the program will also check the packing based on the C_α molecules. It will first select the C_α atoms based NCAMOD. Then grid positions where any two molecules in the unit cell have more than NCACUT C_α–C_α pairs separated by less than DCACUT will be flagged by '#'.

PHASe-only —

This sets QPHASE to true and specifies that only the phase angles from the model, rather than both the phases and the amplitudes, will be used in the POPC function. Default value for QPHASE is false, meaning both phases and amplitudes will be used.

POPC — (MOLCUL(i), i=1, NSYM)

This sets QPOPC to true and specifies that Patterson correlation translation function will be calculated by the program. Default value for QPOPC is false. The correlation will be calculated based on space-group symmetry-related molecule numbers as specified by MOLCUL. The identity symmetry operation is always assigned the number 1. If no molecule numbers are specified, then all the symmetry-related molecules will be used in the calculation. The translation function coefficients will be written to file POPC.DAT, which will contain h, k, l, F, ϕ. The maximum F value is scaled to 1400. The reflections can be input to a Fourier summation program. Due to the possible doubling of the reflection indices, the translation function should be sampled on a grid 1/6 of the maximum resolution of the reflection data. The resulting peak in the map defines the vector by which the input search model with the 'R' flag should be translated.

The program will also try to carry out the Fourier transform directly. A warning will be issued if array limits are exceeded and the program can not perform the transform. As with the R factor and the correlation coefficient searches, a peak search is done through the transform map. The map can be written to a file (command MAPFile), which can be used for re-contouring the map later. The map can also be contoured (command CNTFile). In this case, the desired sectioning of the contour should be specified (command SECTion). The contour level (command CNTLevel) can be set at 50, 100, 5, as the program will scale the maximum of the transform to 100. The contour sections (command CNTSection) can be left as default, which will lead to the contouring of 5 sections containing the top peaks.

For each of the peaks in the POPC translation function map, the program will calculate the correlation coefficient and the R factor (on amplitudes) and will also check the packing of the resulting structure in the crystal unit cell. The large-term cutoff should therefore be set at 0.5 or so for the POPC calculations.

Due to its faster speed as compared to R factor and correlation coefficient searches, the POPC option is preferred for doing translation searches.

RFCEnter — X, Y, Z.

This specifies the position around which the atomic coordinates will be rotated for the RFMAp and RFPEak options below. The COORdinate TYpe needs to defined for the input position. The default is that the rotation will be around the center of gravity of the molecule.

RFFIle — RFFILE*72.

The name of the file that contains the rotation function information is specified by this command. The content of the file depends on which of the RFMAp, RFPEak, RFPC options is used (see below).

RFMAp — RFCUT, RFCUT2, NRFPK

The rotation function map from GLRF is read by this program. If NRFPK is 0 (which is the default), all the grid points in the map that have values greater than RFCUT times the maximum (if RFCUT is less than 1) or RFCUT times the map standard deviation (if RFCUT is greater than 1) and less than RFCUT2 will be used for translation function calculation. If NRFPK is not 0, the program will do a peak search through the rotation function map, with cut-off defined by RFCUT and RFCUT2, and uses the top NRFPK peaks to do the translation search. The default for RFCUT is 0.8, and that for RFCUT2 is 1, meaning all angles (or peaks, depending on NRFPK) greater than 0.8 times the rotation function maximum will be used in the translation function. The maximum number of RF angles that can be saved is given by the MAXPKS parameter.

In this and the RFPEak, RFPC options, the program will first do a POPC search (see option POPC above) for each rotation angle. The top peaks in this POPC map are examined (see option RFSAve below). The region of the unit cell that needs to be covered for this peak search is defined automatically by the program. (The POPC function will always cover the entire unit cell). Possible solution(s) will be kept based on those peaks that do not have packing problems (see options DCUToff and RFSAve). A correlation coefficient (F1F1) and a R factor (FOFC) is calculated for each solution. The results are sorted on the correlation coefficient or the R factor values, depending on NRFSRT (see option RFSOrt below). If RFMAp or RFPEak has been selected, the solutions are examined first to removed close neighbors. For this check, the translation vectors of two solutions must be within 2Å and the orientations must be within 6°.

RFPEak — .

The program will do translation searches for rotation angles that are supplied in the file RFFILE. The file should contain first a few lines that define the angles, specifically SANGle, EULEr or POLAr, ORTHogonalization. The last line before the peak listing should be STOP. The peaks should be listed in free format. Comment lines (by the command COMMent or the ! character) can be inserted anywhere in the file.

RFPC — .

The program will do translation searches using the atomic coordinates after PC refinement. The file RFFILE now contains the coordinates in PDB format, after the PC refinement with the program GLRF. The various molecules in the file, corresponding to different orientations of the search model, are separated with the TER keyword.

RFSAve — NRPK, QSVALL.

This specifies how many peaks in the POPC translation function should be saved for each set of rotation angles. The program will then check packing for each of these peaks. Correlation coefficient (F1F1) and R factor (FOFC) will be calculated for those peaks with good packing. The one with the highest correlation coefficient will be saved. Default for NRPK is MAXRPK. If QSVALL is set to true (which defaults to false), all the positions with good packing will be saved.

RFSOrt — NRFSRT

This specifies whether the solutions will be sorted on the correlation coefficient (NRFSRT=1) or the

R factor (NRFSRT=2) or the difference map peak height if supplied (NRFSRT=3). The default for NRFSRT is 1.

SELF-vector-removal — SELFWT.

This specifies a weight for the intra-molecular vectors in calculating the Patterson correlation translation function coefficients. The SELFWT is applied after the calculated structure factors have been scaled to the observed ones, in shells of equal reciprocal volume. Default for SELFWT is 0, meaning the self-vectors will not be removed.

SFCAlculation — OBSNEW*72.

This sets the variable QSFCA to true and specifies that structure factors should be calculated based on molecules with 'T' and/or 'R' flags. Default value for QSFCA is false. A file name can be given for OBSNEW, which will contain the calculated structure factors. The default for OBSNEW is FCALC.DAT. If a set of reflections have not been input through the command FOBS, the program will generate a list of indices from the asymmetric unit of the reciprocal lattice and the program will output h, k, l, F_{calc} and ϕ_{calc} in the format of (3i4, 2f8.2). If structure factor amplitudes have been input with the FOBS command, the program will output $h, k, l, F_{obs}, F_{calc}$ and ϕ_{calc} in the format (3i4, 3f8.2). The program will also calculate the distribution of the scale factors, R factors and correlation coefficients versus resolution and the reflection index parity (h, k, l even or odd).

For the molecule with 'R' flag, structure factors are calculated for the molecule at a specific location. If atomic coordinates are input for the molecule, the structure factors are calculated for the current location of the coordinates. If structure factors are input for the molecule (through the COORdinate FCalc-file command), the structure factors will be modified by the translation vector that has been specified for the molecule. A COORdinate CEnter command must be given to specify the molecular center corresponding to the input set of structure factors. Then a translation vector or a new center can be specified with the COORdinate TRanslate or the COORdinate CEnter command.

SHELl — NSHELL.

The reflection data will be divided into NSHELL shells of equal reciprocal volume, and an average intensity will be calculated for each shell. This will be used for the calculation of resolution-dependent scale factors between observed and calculated structure factors for OPTIONs FOFC and IOIC. This will also be used in the selection of large terms. The default value for NSHELL is 8. The maximum number of shells is specified by the parameter MAXSHL.

MAPFile	MAPFIL*72	[none]

If this command is specified, the program will dump the map values from F1F1, F2F2, FOFC, IOIC, POPC, and EOEC functions as a formatted file. Default is the map values will not be dumped. This file may be useful for later processing with other programs (see the command PEAK). To maintain precision, a factor of 10 is multiplied to the function values before output.

PEAK-Cutoff	PKCUT, NLIST, MAPFIL*72	[5.0, 50]

The program will perform a peak search in the resulting maps from F1F1, F2F2, FOFC, IOIC, POPC, and EOEC functions. In this peak search, grid points whose values are less than PKCUT × σ (where σ is the standard deviation of the map values) over the map average will be skipped. NLIST (maximum 200) peaks will be printed at the end of the peak search. The maximum number of peaks that can be saved is given by the MAXPKS parameter.

If a file name has been specified for MAPFIL, the program will read the map values from that file and perform a peak search. In this case, no translation search will be carried out.

RESOlution-limits DMAX, DMIN [10.0, 6.0]

This specifies the resolution limits for reflection data that will be used in the search. The maximum Miller index is given by the parameters MAXH, MAXK, MAXL, respectively. The program will abort if either of them is exceeded.

SECTion SECSHN*3 []

The sectioning of the map in the output map file (see command MAPFile) and the contour plot (see command CNTFile) is defined by this command. The first field specifies the map sections, the second field specifies which direction will be output across the page, and the third field specifies which direction will be output down the page. For example, specifying '321' would mean *c* section, *b* across, and *a* down the page. The second and/or the third field can be omitted and the program will provide reasonable defaults. If the command is omitted, the program will choose the direction with the smallest number of grid points as the section direction. The direction with the largest number of grid points will be output across the page.

SIZE SIZMOL [30.0]

This specifies the radius of molecule, in Å. This will be used in the R-factor and correlation coefficient searches to exclude positions that are forbidden from packing considerations. This will also be used if option PACKing is employed in command FUNCtion (see below). Note that molecular size can also be defined with the COOR command. SIZMOL will be used by the program if the size has not been defined with COOR SIze.

If a negative SIZMOL value has been input, the program will not do a packing check during the search. This could prove useful when the molecule is located on a crystallographic symmetry axis.

SLIMits IAXIS, (SLIMTS(i, IAXIS), i=1, 3) [none]

This specifies the limits for the current search. Three separate SLIMits commands need to be given to define the limits in all three directions. IAXIS can assume values of 1, 2, and 3, corresponding to fractional translations along *a*, *b*, and *c* axes, respectively. The limits are specified as — beginning value, ending value, and increment, in fractional deorthogonalized units. The search region defined by this command is required by the FUNCtion options F1F1, F2F2, FOFC, IOIC, and PACKing. The entire unit cell is covered for FUNCtion options POPC and EOEC.

If the beginning value for IAXIS=2 is given a value that is greater than 10, the program will automatically assign the search limits based on the crystal symmetry. The increment values are assigned to be 1/4 of the highest resolution. Therefore, to ask the program to automatically assign the search limits, enter a command line 'slimits 2 10 10 0'.

VII. CONTOUR PLOTS

CNTFile CNTFIL*72 [none]

If a file name has been specified by this command, the program will produce contour plots for the translation function map. Default is that the contour plot will not be created.

If an one-dimensional search has been carried out (*i.e.*, only one coordinate was allowed to vary), the translation function values will be plotted as a function of the varying coordinate. In this case, the CNTLevel and CNTSection commands are ignored by the program.

The maximum number of points in each contour trace is given by the parameter MAXCVS. If the program is stopped because this limit is exceeded, either recompile the program with a larger MAXCVS or raise the starting contour level.

CNTLevel (CNTLVL(i), i=1, 3), LINTYP [none]

This command specifies the beginning, ending, and increment for the contour level. LINTYP specifies the line type to be used for the contours in this level. This command can be repeated to obtain different contour levels. The contour levels can be specified as absolute values or in terms of the σ value of the map. In the latter case, the beginning contour level must be specified as less than 5 and the ending contour level must be less than 10. The increment must be less than 2. For example, specifying 'CNTLevel 1 5 0.5' means the contouring should begin at 1σ above the map average and end at 5σ above the map average, with increment of 0.5σ. The maximum number of contour levels that can be specified is given by the MAXLVL parameter.

Since only one set of contour levels is read by the program, only one of the search functions (F1F1, F2F2, FOFC, IOIC, POPC, EOEC) should be turned on during the search. Alternatively, the user can edit the resulting translation function map and leave only those values corresponding to one of the functions.

Currently 7 different line types are supported. LINTYP=1 is a solid line 0.1 point wide (there are 72 points to an inch). LINTYP=2 is a solid line 0.5 point wide. LINTYP=3 is a solid line 1 point wide. LINTYP=4 is a dashed line (2 on 2 off) 0.1 point wide. LINTYP=5 is a dashed line 0.5 point wide. LINTYP=6 is a dashed line (4 on 1 off) 0.1 point wide. LINTYP=7 is a dashed line 0.5 point wide.

Since it is difficult to predict the optimal contour levels beforehand, it is suggested that the translation function values be written out with the MAPFile command. Then contour level(s) can be chosen and the translation function map values can be read in with the command PEAK.

CNTRegion X1, X2, Y1, Y2 []

This defines the region in each section that should be plotted. X1 and X2 defines the beginning and ending values, in fractional or grid units, for the axis that is plotted across the page. Y1 and Y2 defines the values for the axis down the page. The default is that the entire section will be plotted. As translation functions generally contain one peak in a very large area, the search results can be written out with MAPFile. Then a smaller region around the peaks of interest can be plotted by using the command PEAK to read in the map file and this command to define the region of interest.

CNTSection ICNTSB, ICNTSE [0, 0]

This specifies the section range for which the contour plot should be created. The numbering of the sections starts with the first section being number 1. The default is that all the sections will be plotted if there are 5 or less sections in the map. If there are more than 5 sections, the program will plot a maximum of 5 sections containing the top peaks in the map. If no peaks are found in the map, the first 5 sections of the map will be plotted.

VIII. STORAGE LIMITS

Storage limits for different parameters are defined in a PARAMETER statement in the Fortran INCLUDE file MAIN.CMN. The program will need to be recompiled if any of these limits is exceeded. Note that the parameter MAXGRD should always be greater than or equal to the parameter MAXREF.

Parameter	Limit
MAXH	250
MAXK	250
MAXL	250
MAXATM	16000
MAXREF	150000
MAXGRD	1000000
MAXFFT	4×MAXGRD
MAXTYP	20
MAXSHL	10
MAXPKS	1000
MAXRPK	10
MAXLVL	5
MAXLIN	7

Appendix A.
New Features

New Features for Version 2.6 — July, 1996

- Use isomorphous or anomalous difference map peak as a criterion in the combined MR search. New options DIFFerence and ANOMalous to the command FUNCtion.

New Features for Version 2.5 — April, 1996

- Calculate correlation coefficient and R factor values for each solution from POPC option. And check packing as well.
- New option ECEC, for doing phased translation function (EOEC) using phases from molecules with T flags.
- New option HAND for command FUNCtion, for changing the hand of the phases for the EOEC option.
- New SWap option for command COORdinate.
- Specify the coordinate type on the same line for the CEnter and TRanslate options of COORdinate.
- The R factor on I is divided by 2 to place it on the same scale as that on F.

New Features for Version 2.4 — February, 1996

- The region of unit cell to search for TF solutions can now be assigned automatically by the program. This affects F1F1, F2F2, FOFC, IOIC, RFMAp, RFPEak, and RFPC searches. For POPC and EOEC searches, the entire unit cell will still be searched.
- An R factor is now also calculated for each of the solutions from the RFMAp, RFPEak, RFPC searches. The solutions can be sorted either on the correlation coefficient or the R factor.
- The program has been successfully compiled and run on a Cray J-90. Small modifications in routine NUPAGE are needed to get the current time.

New Features for Version 2.3 — December, 1995

- For TF based on the RF results (RFMAp, RFPEak, RFPC), the program will save the top few peaks (rather than the highest peak only) in the POPC map and then check packing for each of the peaks. The ones that have good packing will be considered as possible solutions. A new option RFSAve is implemented. The input to RFMAp is slightly different.
- The atom C1' from each nucleotide is also included for packing check purpose.
- Various bug fixes.

- Updated the documentation.

New Features for Version 2.2 — July, 1995

- The rotation function results can be directly input to the TF program. Four new options (RFFIle, RFMAp, RFPEak, RFPC) have been implemented for the FUNCtion command.
- The program will automatically do a packing check for the peaks in the translation function.
- Modified the packing check to make it run faster.
- The program will select both Ca and P atoms for packing check.
- Modified the coordinate I/O routines to fully support the PDB format.
- Various bug fixes.
- Updated the documentation.

New Features for Version 2.1 — May, 1995

- Structure factor calculation by FFT is now supported.
- FFT calculations for the POPC and EOEC translation functions are now available in the program. The program will peak search, dump, and contour the functions. The map input/output routines are updated to reflect this change.
- Scratch file is now opened with status SCRATCH. All other output files are opened with status UNKNOWN. This will overwrite old files.
- Provided labeling of the axes in the contour plots. Allow the plotting of a small region of the map (new command CNTRegion).
- Scattering factors for most common atom types are provided as default. Unknown atoms are ignored.
- A new scheme is devised for the storage of the structure factors for the R molecule. This eliminated the need to keep two separate versions (high and low symmetry) of the program.
- The chain and insertion code columns of the PDB file are now supported (new COORdinate CHain command). Only PDB records starting with ATOM and HETA are kept. Hydrogen atoms are removed.
- Various bug fixes.
- Updated the documentation and designed several example input files.

Appendix B.

Phase Relationship among Symmetry-related Structure Factors

The following relationships among F_h and $f_{h,n}$ are used by the program.

$$\begin{aligned}
F_{h[T_n]} &= \sum_j f_j e^{2\pi i h[T_n]x_j} \\
&= \sum_j f_j e^{2\pi i h([T_n]x_j + t_n - t_n)} \\
&= e^{-2\pi i h t_n} \sum_j f_j e^{2\pi i h([T_n]x_j + t_n)} \\
&= e^{-2\pi i h t_n} F_h
\end{aligned}$$

$$\begin{aligned}
f_{h,n} &= \sum_j f_j e^{2\pi i h([T_n]x_j^0 + t_n)} \\
&= e^{2\pi i h t_n} \sum_j f_j e^{2\pi i (h[T_n])x_j^0} \\
&= e^{2\pi i h t_n} f_{h[T_n],1}
\end{aligned}$$