Version 1.0, Revised December, 1990

Reciprocal Space Molecular Replacement

Program GHKL

Liang Tong†

INTRODUCTION

Consider the presence of a molecule in two different crystal forms, h (x) and p (y). The structure factors for the p crystal can be calculated based on those (amplitudes and phases) for the h crystal, as derived by P. Main & M. G. Rossmann, *Acta Cryst.* **21**, 67–72, (1966).

The molecule in the p crystal is related to the molecule in the h crystal by the following equation,

$$\begin{aligned} y &= [C]x + d \\ &= [\alpha_p][\rho][\beta_h]x + d \end{aligned}$$

where $[\rho]$ is a rotation matrix that brings the molecule in the h crystal to the same orientation as the molecule in the p crystal, and d is the translational element. The above equation can be inverted to write x as a function of y,

$$x = [C]^{-1}y - [C]^{-1}d.$$

If the center of the molecule in the h cell is s_h, the center of the molecule in the p cell will be given by

$$s_{p,1} = [C]s_h + d.$$

The centers of the symmetry-related molecules will be given by

$$s_{p,n} = [T_n]s_{p,1} + t_n$$

where $[T_n]$ and t_n are the rotational and the translational components of the symmetry operation, respectively.

The derivation of the molecular replacement equation is based on the equality of electron density for the molecule in the h and the p cells, *i.e.*, in the following derivation,

$$\begin{aligned} \rho_1(y) &= \rho(x) \\ &= \sum_h F_h e^{-2\pi i h x} \\ &= \sum_h F_h e^{-2\pi i h([C]^{-1}y - [C]^{-1}d)}. \end{aligned}$$

† Please send comments and bug reports to : b4w@mace.cc.purdue.edu.

$$
\begin{aligned}
F_p &= \int_{\substack{unit\\cell}} \rho(y) e^{2\pi i p y} dy \\
&= \sum_{n=1}^{N} \int_{\substack{unit\\cell}} \rho_n(y) e^{2\pi i p y} dy \\
&= \sum_{n=1}^{N} \int_{\substack{\Omega\\(|y-S_{p,1}|\leq R)}} \rho_1(y) e^{2\pi i p([T_n]y+t_n)} dy \\
&= \sum_{n=1}^{N} \int_{\Omega} \left[\sum_h F_h e^{-2\pi i h([C]^{-1}y-[C]^{-1}d)} \right] d^{2\pi i p([T_n]y+t_n)} dy \\
&= \sum_{n=1}^{N} \sum_h F_h e^{2\pi i(h[C]^{-1}d+pt_n)} \int_{\Omega} e^{2\pi i(p[T_n]-h[C]^{-1})y} dy \\
&= \sum_{n=1}^{N} \sum_h F_h e^{2\pi i(h[C]^{-1}d+pt_n)} G_{hpn} e^{2\pi i(p[T_n]-h[C]^{-1})s_{p,1}} \\
&= \sum_{n=1}^{N} \sum_h F_h e^{2\pi i(h[C]^{-1}d+pt_n)} G_{hpn} e^{2\pi i(p[T_n]-h[C]^{-1})([C]s_h+d)} \\
&= \sum_{n=1}^{N} \sum_h F_h G_{hpn} e^{2\pi i h[C]^{-1}d+2\pi i p t_n+2\pi i p[T_n][C]s_h+2\pi i p[T_n]d-2\pi i h s_h-2\pi i h[C]^{-1}d} \\
&= \sum_{n=1}^{N} e^{2\pi i p(t_n+[T_n][C]s_h+[T_n]d)} \sum_h F_h G_{hpn} e^{-2\pi i h s_h} \\
&= \sum_{n=1}^{N} e^{2\pi i p(t_n+[T_n]s_{p,1})} \sum_h F_h G_{hpn} e^{-2\pi i h s_h} \\
&= \sum_{n=1}^{N} e^{2\pi i p s_{p,n}} \sum_h F_h G_{hpn} e^{-2\pi i h s_h}
\end{aligned}
$$

The presence of the G function (G_{hpn}) suggests that the summation over h can be limited to those values which give small differences for $p[T_n] - h[C]^{-1}$.

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. 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 five logical I/O units, labelled by the variables LIN, LOG, LPRT, LOBS, and LSCRCH, which are initialized to be 5, 6, 3, 1, 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 using the PRINt command. The reflection data (if any) is read from unit LOBS. This unit is also used internally to read and/or write translation function map files. 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 characters). Variables whose names begin with the letter 'Q' are logical variables. A 'True' of 'False' input is expected for these variables. The defaults (if any) of the variables are given in square brackets. These default values will be used by the program if the command is omitted from input, or if a value of 0 is input to the command.

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.

PRINt-File PRTFIL*40 [GHKL.PRT]

This opens the output file of the program (associated with the logical unit LPRT). 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 attempt five times to open the scratch file, each time with a different name. The file names are of the form GHKL*.TMP. This should make it possible to run more than one jobs in the same directory on a UNIX system.

RESOlution DMAX, DMIN [10.0, 3.0]

This specifies the resolution range for the reflections that will be used in the calculation.

STOP none [none]

The program willstop reading from the command input file and start the actual calculation. Otherwise, the program will read until the end-of-file on unit LIN before initiating the calculation.

TITLe ATITLE*132 [PROGRAM GHKL]

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

EULEr-Angle EULER*3 [ZXZ]

This specifies the definition convention of Eulerian angles. Two conventions are currently supported — ZXZ and ZYZ, corresponding to rotation around the Cartesian Z axis (θ_1), then around the new X (and Y, respectively) axis (θ_2), and finally around the new Z axis (θ_3). 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).

POLAr-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 mentioned 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.

ORTHogonalization ORDOR*5 [BYBCX]

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.

III. COMMANDS THAT DEFINE THE MODEL CELL.

MODEl-cell OPTION*4 [none]

This command defines the parameters for the model unit cell (the h crystal in the derivation above). OPTION is a four-letter keyword that defines the action that should be taken. The options that are currently supported are given below.

BOXSize — (IBXSIZ(i), i=1, 3) [5, 5, 5]

This specifies the size of the interpolation box for the model cell. The maximum size is given by the parameter MAXBOX.

CELL-parameters — (CELL(i, 1), i=1, 6) [90, 90, 90, 90, 90, 90]

This defines the unit cell parameters of the model cell. The default

CENTer — (CENTER(i, 1), i=1, 3) [0, 0, 0]

This specifies the center of the molecule in the model cell. The coordinates must be in fractional deorthogonalized units.

FOBS-data — REFFIL(1)*40, REFFMT(1)*40 [FOBS.DAT, (3I4, 2F8.2)]

This defines the name of the file containing the structure factors for the model cell. The file is expected to contain for each reflection h, k, l, F, and ϕ (in degrees) values. The format of the file can also be specified with this option.

FOMWeighting — QFOMWT [F]

This specifies whether figure-of-merit weighting should be carried out. If set to true, the program expects to read from the structure factor file h, k, l, F, m, and ϕ values.

LATTice-type — LATICE [P]

This defines the lattice type of the model unit cell. Supported types are *P, A, B, C, F, I, R*.

RADIus — RADIUS [20]

This specifies the radius of integration, in Å.

ROTAtion — (ROTANG(i), i=1, 3), ROTTYP*1 [0, 0, 0, E]

This defines the rotational relationship between the molecule in the model cell and that in the crystal cell. The molecule in the model cell must be rotated by the specified set of angles to match the orientation of the molecule in the crystal cell. The angle type can be either Eulerian or polar. The angle convention is defined by POLAR or EULER.

SYMMetry — [none]

This defines the symmetry of the model cell, in the same format as the International Tables. The identity operation is assumed and need not be given. The program will expand the model cell reflection data to $P1$.

III. COMMANDS THAT DEFINE THE CRYSTAL CELL.

CRYStal-cell OPTION*4 [none]

This command defines the parameters for the crystal unit cell (the p cell in the derivation above). OPTION is a four-letter keyword that defines the action that should be taken. The options that are currently supported are given below. Options that parallel those for the model cell will not be further explained.

CELL-parameters — (CELL(i, 2), i=1, 6) [90, 90, 90, 90, 90, 90]

CENTer — (CENTER(i, 2), i=1, 3) [0, 0, 0]

FCALc-data — REFFIL(3)*40, REFFMT(3)*40 [none, (3I4, 3F8.2)]

If a file name has been specified, the program will output the calculated structure factors (h, k, l, F_{obs}, F_{calc}, and ϕ) to the file. Default is that the calculated structure factors will not be output. The F_{obs} value will be output only if observed structure factor amplitudes have been supplied (see option FOBS-data). If F_{obs} data are not provided, the program will automatically generate a list of reflections in the asymmetric unit of the crystal cell.

FOBS-data – REFFIL(2)*40, REFFMT(2)*40 [FOBS.DAT, (3i4, 2F8.2)]

If a set of F_{obs} data is provided, the program will calculate the phase angles for these reflections based on the model cell.

LATTice-type — LATICE [P]

SYMMetry — [none]

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.

Parameter	Limit
MAXH	100
MAXK	100
MAXL	100
MAXREF	50000
MAXG	1000
MAXBOX	9