

EMSF3DBT.DOC

(last update by TSB, November 3, 1998)

CONTENTS

- A. INTRODUCTION
- B. PROGRAM INPUT
- C. FLOW CHART FOR EMSF3DBT PROGRAM

A. INTRODUCTION

EMSF3DBT produces 3D MAP data from 3D structure factors (generated, for example, with EMMAP3DT). EMSF3DBT is mainly used to create a highly magnified portion of a 3D MAP so ROBEM can produce "publication quality" shaded-surface renderings. Such maps earn the "publication quality" rating because voxels in the new 3D MAP are much smaller than in the original 3D MAP. *Note: RWA currently has a beta version of this routine working in RobEM. When you first enter RobEM, go to the main pulldown menu Option and then click the button labeled "Calculate structure factors for incoming maps". EXIT and proceed as normal and in Display you can change the pixel to any size (within reason of course!).*

EMSF3DBT evolved from a crystallographic Fourier summation program, FOURIER, which I believe was written in the 1960's by R. J. Dellaca of the University of Canterbury, Christchurch, N.Z. Except for some 'hard core' subroutines, the program has been revised and rewritten numerous times to work on various machines. EMSF3DBT is intended for use on VAX and Unix computers and is specifically tailored for electron crystallography applications. The program FOURIER was designed to compute several useful crystallographic functions, including difference Fouriers and Patterson maps. EMSF3DBT **only** computes a Fourier summation. Other options from FOURIER are **not** available in EMSF3DBT, although interested parties may have the original code if they so desire (and I can locate it!).

The latest version of EMSF3DBT has done away with the infamous CONTROL data file (*.HED), which was used to store input parameters. All input is now entered interactively or via a BATCH command file, as you wish. Though the input to EMSF3DBT at first glance appears daunting, the program is normally run with a standard set of simple input parameters. **MOST** of the program input, especially line 2, and lines 5 through 11, can be entered by typing a <CR> and accepting the DEFAULT values.

B. PROGRAM INPUT

1. Filename for 3D structure factor data input (A)
2. RES_HI (F)
3. Filename for 3D MAP data output (A)
4. 3D MAP output file header (80A1)
5. ITRANS, NEQUIV, SCALEF (2I,F)
6. ((TS(n),FS(n,m),m=1,3),n=1,3) (12F)
7. U MATRIX (9F) (Only ITRANS=1)
8. P1,P2,P3,D,CELL(3) (7F) (Only ITRANS=2)
9. XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,DX,DY,DZ (9F) (Only ITRANS=2)
10. NREF_DUMP (I)
11. XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX (6F) (Only ITRANS.LE.1)
12. DX,DY,DZ (3F) (Only ITRANS.LE.1)
13. NORIEN (I) (Only ITRANS.LE.1)

Detailed descriptions of program input:

1. Filename for 3D structure factor data input (A FORMAT)

Enter the name for the file that contains the input 3D structure factor data. SF data are currently stored in BINARY or ASCII format. EMPROGS.DOC describes the SF data and how they are stored.

2. RES_HI (F)

RES_HI defines the highest resolution structure factor to use in computing the new 3D MAP. The default is to read and use all the SF data stored in the file. RES_HI **must be** specified in the same units (Å, nanometers, pixels, yards, etc.) as was used to create the SF data, otherwise strange results can occur. If the resolution of the input data is **lower** than your request, then RES_HI will DEFAULT to the lower value (*i.e.* you can't make something out of nothing!).

3. Filename for 3D MAP data output (A FORMAT)

Enter a filename for storing the new 3D MAP.

4. 3D MAP output file header (80A1 FORMAT)

Enter any header information (up to 80 characters) you like here. This information is stored in the header record of the PIF format 3D MAP data file.

5. ITRANS, NEQUIV, SCALEF (2I,F)

DEFAULTS: 0, 1, 1.0

ITRANS specifies the orientation of the sections. The DEFAULT (used 99.99% of the time) is to compute sections parallel to the unit cell axes:

- = 0 Sections calculated parallel to unit cell axes
- = 1 Sections calculated parallel to a general plane, the U matrix of which is supplied
- = 2 Sections calculated parallel to a general plane, the equation of which is supplied

NEQUIV specifies the number of symmetry cards (input line #6). This is the number of equivalent positions in the space group EXCLUDING those related by a center of symmetry or by a unit cell centering condition. Since P1 space group is used 99.99% of the time, NEQUIV is normally set = 1.

SCALEF is used to scale the height of density peaks if the input data are not on an absolute scale. For example, for a C-centered cell, SCALEF = 2.0. If the input data are doubled, for example with data sets containing Friedel pairs, SCALE = 0.5. For P1 data, SCALEF is normally set to the DEFAULT value of 1.0.

6. ((TS(n),FS(n,m),m=1,3),n=1,3) (12F) (one for each of the NEQUIV positions)

DEFAULT = 0.,1.,0.,0., 0.,0.,1.,0., 0.,0.,0.,1.

Each of the NEQUIV input lines contains a [3x3] rotation matrix [FS] and a [3x1] translation vector [TS] such that:

$$P' = [FS]*P + [TS],$$

where P' and P are the transformed and initial positions, respectively.

Take for example, (-x, x-y, 1/2+z):

$$[TS] = \begin{vmatrix} 0 \\ 0 \\ 0.5 \end{vmatrix} \quad \text{and} \quad [FS] = \begin{vmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

and the data card should appear: ((TS(n),FS(n,m),m=1,3),n=1,3)

The above example would be:

0., -1., 0., 0., 0., 1., -1., 0., 0.5, 0., 0., 1.

Hence, the DEFAULT corresponds to the simplest case: (x,y,z)

7. U MATRIX (9F) (Only for ITRANS = 1)

Supply transformation matrix, U (U11,U12,U13,...U33), where

$$\begin{vmatrix} a' \\ b' \\ c' \end{vmatrix} = U * \begin{vmatrix} a \\ b \\ c \end{vmatrix}$$

8. P1, P2, P3, D, CELL(1-3) (7F) (Only for ITRANS = 2)

Supply the general equation of the plane as:

$$P1*X + P2*Y + P3*Z = D,$$

where D is the distance (in Å, nms, etc.) of the plane from the origin of the unit cell.

For this option, the new axis X' is set normal to the plane, Y' is set parallel to the plane and to the projection of the original unit cell vector which has the largest component on this plane (*i.e.* the unit cell axis which has the minimum P value). Z' completes a right-handed orthogonal system.

In this instance the program automatically sets NORIEN = 100. The length of each new axis is set to the maximum of four quantities:

$$|2*XMAX(I)|, |2*XMIN(I)|, 10.0, CELL(I)$$

The longer the axis, the greater the accuracy, *i.e.* unless the elements of U are integers, errors will be introduced when the indices are transformed by the U matrix. For a section through the plane, set XMIN = XMAX = |D|.

The U matrix is printed out as:

$$NEWA1 = U11*A1 + U12*A2 + U13*A3 \text{ etc.}$$

The determinant of the U matrix is equal to the ratio of the volumes of the transformed and original unit cells.

9. XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX, DX, DY, DZ (9F) (Only for ITRANS = 2)

The limit and increment information (in Å, nms, etc.) refer to the NEW TRANSFORMED unit cell.

10. NREF_DUMP (I)

NREF_DUMP specifies the number of reflections for which the equivalent forms and Fourier coefficients are to be listed. This

is normally used to check the calculation. The DEFAULT setting (NREF_DUMP = 0) bypasses this option.

11. XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX (6F)
DEFAULT: 0.0,1.0, 0.0,1.0, 0.0,1.0

These specify the limits along the X-, Y-, and Z-axes in **fractional** coordinates. If, for example you only wish to examine the central portion of the top half of the original MAP, values like the following might be entered:

0.25,0.75, 0.25,0.75, 0.5,1.0

12. DX,DY,DZ (3F) (Only for ITRANS.LE.1)
DEFAULTS:1.0,1.0,1.0

These specify the intervals (*i.e.* voxel dimensions in Å, nms, etc.) between the sampling points parallel to the X, Y and Z axes respectively. **WARNING:** Use dimensions in the same units as those used to specify the unit cell axis dimensions (as stored in one of the header records of the SF data file). If for example, the original 3D MAP was based on $(10\text{Å})^3$ pixels, you might want to produce a magnified portion of the MAP with $(2.5\text{Å})^3$ voxels. Be aware that if you attempt to create the entire 3D MAP with $(2.5\text{Å})^3$ voxels, it will take up 64 times more memory and/or disk space! If you make $(1.0\text{Å})^3$ voxels, it would be 1000 times larger!!! Hence, the smaller you make the voxels in the MAP, the smaller the region you are realistically able to view with programs like ROBEM.

If DY or DZ are not specified, they DEFAULT to the value of DX. Hence, if you only need to enter a value for DX if you want DY and DZ to be the same (saves a little typing!).

13. NORIEN (I) (Only for ITRANS.LE.1)

NORIEN defines the orientation of the sections as follows:

= 100 Calculate sections along the X-axis
 10 Calculate sections along the Y-axis
 1 Calculate sections along the Z-axis

The following points should be noted:

- (a) For each value of NORIEN, the orientation of the remaining two axes is as follows:

NORIEN	SECTION	ACROSS PAGE	DOWN PAGE
100	X	Z	Y

C. FLOW CHART FOR EMSF3DBT PROGRAM

```

*
*
*- SF_OPEN3 - SF$OPEN - | - STRING_UPPER - STRING_LEN !
*                       | - REAL_TO_REC !
*                       | - FILE_CHECK !
*
*- SF3DBT_GETMEM1 - MALLOC !
*
*- SF_FILL2 !
*
*- FREE !
*
*           | - PIRADDEG_D !
*- SF3DBT INFO - | - SF3DBT_STOAA !
*               | - SF3DBT_M13 !
*               | - SF3DBT_ROT !
*
*           | - SF3DBT_M13 !
*           | - SF3DBT_VMV !
*- SF3DBT_TRANS - | - SF3DBT_GP1 ----- | - SF3DBT_VM !
*                 | - SF3DBT_M13 !       | - SF3DBT_NORM - SF3BDT_VMV !
*                 | - SF3DBT_CROSS !
*- SF3DBT_GETMEM2 - MALLOC !
*
*           | - SF3DBT_CHECKUP !
*           | - SF3DBT_ROT !
*- SF3DBT_RIP - | - SF3DBT_ROT_I4 !
*               | - SF3DBT_NUCALC - IFIXR !
*               | - SF3DBT_ROT_I2 !
*
*- SF3DBT_FOURIER - SF3DBT_GETARGS !
*
*           | - PIF_INIT_HEAD !
*           | - PIF_OPEN !
*- SF3DBT_OUT - | - PIF_WRITE_GH - differentEndian !
*               | - PIF_WRITE_DH ----- | - differentEndian !
*               | - PIF_CLOSE !           | - convertBackFloat !

```