EMPRJ.DOC

(last update May 2, 2000)

CONTENTS

A. INTRODUCTION
B. PROGRAM INPUT
C. EXAMPLE EMPRJ BATCH JOB
D. FINAL NOTES
E. FLOW CHART FOR EMPRJ

A. INTRODUCTION

EMPRJ produces a series of projection images from a 3D MAP (PIF format) (similar to the function of option 'X' of the old EMMAP program). EMPRJ computes projected views of the 3D MAP in one of three ways. Projected views can be computed in the orientations specified in data PARAMETER files, or they can be computed at specified intervals (DANG) that cover an entire one-half asymmetric unit (as specified by parameter SYM), or they can be computed at random orientations within the whole asymmetric unit. It is easy to create a variety of PARAMETER files and edit them as you see fit to create different sets of projection images.

EMPRJ outputs the projection data in either of two PIF data formats. The default (outPifMode=0) outputs the model projection images in the PIF format for PRJ-type data (PIFMODE=21) as is used in the EMPFT program. An alternative (outPifMode=1) generates a standard byte-image type PIF format file.

You also have the option (via CTFMODE) to impose a CTF or inverse CTF onto the projection images. This is useful for manipulating and generating test data. The program RobEM can also be used to accomplish this task.

NOTES:

EMPRJ currently ONLY works with cubic 3D MAP data (i.e. where NCOL=NROW=NSEC).

We eventually hope to fold this routine into ROBEM so projections of the 3D MAP can be compared directly alongside corresponding 'raw' data images. In addition, a program like ICO_TOOLS might be set up to allow users a simple means to create a variety of parameter files for generating sets of projection images.

B. PROGRAM INPUT

1.   BIN, outPifMode, SYM, DANG, CTFMODE, NVIEWS (3I,F,2I)
2.   PIXSIZE, UNITS, VOLTS, AMP_FAC, dF_MAJ, dF_MIN, ANG_MAJ, Cs,
       Cc, BETA, dE, TEMPFAC, WIENER (F,I,11F)
3.   3D MAP input filename (A)
4.   PIF format PRJs output filename (A)
5-N. Input/output parameter filename(s) (A)

1. BIN, outPifMode, SYM, DANG, CTFMODE, NVIEWS (3I,F,2I)

   BIN = 1 Binfactor of 1 (no data compression)
       = 2 Binfactor of 2 (compress 3D data by 8X; 2D data by 4X)
           The current limit on BIN is 2 and the DEFAULT is 1
           (i.e. no data compression)

   outPifMode = 0 for I*4 PRJ type output
              = 1 for byte IMG type output

   SYM =   1 for no symmetry (like a ribosome)
       = 2-31 for n-fold cyclic symmetry (about z-axis)
       = 532 for 532 icosahedral symmetry (DEFAULT)
       = 222 for 222 dihedral symmetry
       =   32 for 32 dihedral symmetry
       = 422 for 422 dihedral symmetry
       =   52 for 52 dihedral symmetry
       = 622 for 622 dihedral symmetry
       =   72 for 72 dihedral symmetry
       = 822 for 822 dihedral symmetry
       =   92 for 92 dihedral symmetry

   SYM specifies the point group symmetry for computing the
projection images within one-half (SYM > 0) or the whole (SYM < 0)
asymmetric unit. Note: Set SYM = 0 (DEFAULT) when you want to use
PARAM files to specify the projection angles. Set SYM < 0 to
trigger the program to produce a set of images with random θ and ϕ
orientation angles (constrained to be within the asymmetric unit
defined by |SYM|).

   DANG specifies the step size (in degrees: DEFAULT = 1.0) between
projection views in the θ and ϕ directions. Note: This parameter
is only used by EMPRJ when SYM > 0 or when SYM < 0 and NVIEWS =
0. The step size in the θ direction remains constant (=DANG) but
it varies in the ϕ direction from a smallest value (=DANG) when
θ=90° (at the 'equator') and *increases* thereafter for progressively
smaller values of θ. Varying the ϕ step size assures uniform
sampling of the ASU in regions where θ is <90°. The step size in
the ϕ direction is given by the formula:

$$DANG/\sin\theta$$

If the ϕ step size was not varied as given in the above formula, the grid sampling would be much too fine near the 'poles'. When θ approaches $0°$ or $180°$ (N and S poles), the program adjusts the ϕ step size to maintain even sampling. For example, at $\theta = 30°$, the ϕ step size will be twice DANG (*i.e.* $2.0°$ if DANG is $1.0°$, since $\{1.0/\sin(30°)\}=2.0$).

DANG influences the number of projection views generated, and hence, determines the size of then output file and run-tie for the program. The following table illustrates how many projection views are generated as a function of SYM and DANG.

Table: Number of views per ASU as a function of particle symmetry and orientation search angle increment ($\Delta°$)

	Symmetry			
$\Delta°$	532	52	5	1
3.00	52	266	519	2,353
2.00	96	573	1,127	5,253
1.00	382	2,173	4,309	20,809
0.50	1,448	8,471	16,869	82,869
0.25	5,653	33,439	66,733	330,751
0.10	34,427	207,358	414,353	2,064,448

CTFMODE is a switch used to signify whether projection images are made with or without a microscope contrast transfer function being imposed. Note that CTFMODE=4-6 is useful for creating test images from a set of 'perfect' projection images, as might be generated, for example, from X-ray atomic coordinates.

CTFMODE	Action taken
0	No CTF modification made to the projection image data.
1	Apply inverse CTF (flip phases and correct amplitudes)
2	Apply inverse CTF (flip phases only)
3	Apply inverse CTF (correct amplitudes only)
4	Apply forward CTF (phases and amplitudes affected)
5	Apply forward CTF (flip phases only)
6	Apply forward CTF (modify amplitudes only)

NVIEWS is used to specify the number of random projection views to generate. This program variable is only used if SYM < 0. If SYM < 0 and NVIEWS is set = 0, then the number of random projection views calculated will be the same as that determined by the DANG and |SYM| values as given in the Table above.

2. PIXSIZE, UNITS, VOLTS, AMP_FAC, dF_MAJ, dF_MIN, ANG_MAJ, Cs, Cc, BETA, dE, TEMPFAC, WIENER (F,I,11F)

The prompt for these parameters is only used only when 0<CTFMODE<7.

```
  PIXSIZE  = pixel size for the image data specified in any units
             (e.g. Å, nanometers, pixels, etc.) as long as you
             make sure to specify the correct value for UNITS.

  UNITS    = specifies the units assigned to PIXSIZE.  UNITS is
             defined as follows:

           = 0 assumes PIXSIZE is given in dimensionless pixels

           = 1 assumes PIXSIZE is given in Ångstroms

           = 2 assumes PIXSIZE is given in nanometers

  VOLTS    = microscope voltage (in volts).

  AMP_FAC  = amplitude factor. For cryoEM data, a DEFAULT value of
             0.07 is reasonable for 100kV electrons (or 0.05 for
             200kV electrons), and anything above 0.15 might be
             suspicious.

  dF_MAJ   = defocus value (μm) along the major axis of
             astigmatism.  Note: Positive values are used to
             designate underfocus.

  dF_MIN   = defocus value (μm) along the minor axis of
             astigmatism.

  ANG_MAJ  = angle between major axis and X-direction in FFT,
             measured positive in a counter-clockwise direction
             (X-axis = 0.0)

  Cs       = spherical aberration coefficient (mm).

  Cc       = chromatic aberration coefficient (mm).

  BETA     = spatial coherence (milliradians).

  dE       = energy spread (eV).

  TEMPFAC  = specifies the inverse temperature factor to be used
             only when applying the inverse CTF (CTFMODE = 1,2,
             or  3).   Values  in  the  range  of  -100  to  -500
             Angstroms  square  are  typical.   TEMPFAC  should
             normally be set = 0.0.

  WIENER   = the 'fudge' factor that keeps the inverse CTF from
             magnifying the noise in the data by a value greater
             than  1.0/WIENER.   WIENER  is  typically  set  to  0.2
             (DEFAULT).
```

3. 3D MAP input filename (A FORMAT)

   Enter the name of the MAP file that contains the 3D model used
to generate the projection images.

4. PIF format PRJs output filename (A FORMAT)

   Enter name of PIF format output file for storing the generated
set of projection images (DEFAULT = EMPRJ.PIF). The projection
images are stored either as INT*4 PRJ data format (when SYM≤0) or
as byte-IMG box format data (SYM>0).

5-N. Input/output parameter filename(s) (A FORMAT)

   If SYM = 0, enter the name or names of one or more PARAM files
(use separate line for each filename) from which the orientation
angles are obtained. The first line of each PARAM file NORMALLY
contains the name of the IMAGE file which contains byte-packed raw
image data stored in PIF format. The next line of each PARAM file
NORMALLY contains the values of: PIXSIZE, VOLTAGE, AMPFAC,
FOCUS_MAJ, FOCUS_MIN, ANG_DELF, and Cs. The remaining lines in
each PARAM file contain the usual set of THETA,PHI,OMEGA,ORIG_X,
ORIG_Y and MAG_FAC values for data in the raw image file.

   Note: for this particular program, the first two lines of each
PARAM file are meaningless, so the lines MAY be blank but two
blank lines MUST be present. If they are not present, then EMPRJ
will NOT compute the first two projections because the program
ignores the first two lines of each PARAM file.

   For |SYM| > 0,the program produces a new PARAM file with two
dummy lines at the start followed by lines containing the THETA,
PHI, OMEGA, ORIG_X, ORIG_Y, etc. values for the model projections.

C. EXAMPLE EMPRJ BATCH COMMAND PROCEDURE

```
$ SET DEF DEXTRO3:[TSB.V.HSV.TEST]
$ RUN DEXTRO3:[TSB.EXE]EMPRJ.EXE
2, 0, 0, 0.0, 0, 0
HSV.PIFMAP
HSV.PRJS
HSV.DAT
$ EXIT
```

D. FINAL NOTES

1. DEXTRO3:[TSB.FOR]EMPRJ.BCH is an example BATCH command file used to run EMPRJ.

2. Use BIN=2 if speed is important to you (i.e. you want a quick look at a series of projection images). For fine detail, use BIN=1, which will produce projection images of exactly the same size as your raw data images (only true if your 3D MAP also has the same pixel size as the image data).

E. PROGRAM FLOW CHART

```
*******************
*      MAIN        *
*   (EMPRJ.FOR)    *
*******************
      *
      *          |- GET_NVIEWS — GET_T1T2P1 !
      *          |- PIF_OPEN !
      *          |- PIF_READ_GH - differentEndian_() !
      *- INFO -|- PIF_READ_DH ----------------------|- differentEndian_() !
      *          |- PIF_INIT_HEAD !                 |- convertBackFloat_() !
      *          |- PIF_WRITE_GH - differentEndian_() !
      *          |- PIF_WRITE_DH -|- differentEndian_() !
      *                           |- convertFloat_() !
      *
      *- GETMEM - MALLOC !
      *
      *                        |- GET_T1T2P1 !
      *- GETTPO — MODEL_TPS —|- SEED !          |- TO_NFOLD !
      *                        |- TO_ASYM_UNIT -|- RETURN_TO_UNIT_TRIANGLE**
      *                                         |- TO_N22 !
      *
      *          |- PIF_READ_WMAPI2 - differentEndian_() !
      *          |- PIF_CLOSE !
      *          |- MAP_CLEAR !  |- PIRADDEG !
      *          |               |- MAP_CLEAR !
      *          |- MAP_PRJ -----|- MAP_PRJ_XZ --- MAP_CLEAR !
      *- PRJS -|               |- MAP_PRJ_AXIS - MAP_CLEAR !
      *          |               |- MAP_PRJ_ALL -- MAP_CLEAR !
      *          |
      *          |               |- PIF_WRITE_MAP_FLTINT - differentEndian_() !
      *          |- PRJS_STORE -|- PIF_WRITE_PRJI1 - PIF_WRITE_BYTE_IMAGE !
      *                          |- MAP_SYM**
      *- FREEMEM — FREE !
      #


=================================================================================
|
|            |- MAP_SYM_CAVG - MAP_STATS — MAP$STATS !
|            |- MAP_SYM_RAVG -|- MAP_SYM_GRID !
|- MAP_SYM -|- COPY_R4        |- MAP_STATS - MAP$STATS !
|            |- MAP_STATS - MAP$STATS !
|
=================================================================================
|
|                          |- POLAR_TO_COSINES - NORM !
|- RETURN_TO_UNIT_TRIANGLE -|- EQUIVALENT_VIEW -|- CROWTHER_TO_MATRIX !
|                                               |- MATMUL !
|                                               |- MATRIX_TO_CROWTHER !
|
=================================================================================
```