TITLE:	**LFI Beams Delivery: Format Specifications**

DOC. TYPE:	TECHNICAL NOTE		
PROJECT REF.:	PL-LFI-PST-TN-044	**PAGE:**	I of V, 9
ISSUE/REV.:	1.0	**DATE:**	July 2003

Prepared by	M. SANDRI LFI Project System Team	Date: Signature:	July 15th, 2003 Maura Sandri
Checked by	C. BURIGANA F. VILLA LFI Project System Team	Date: Signature:	July 15th, 2003 Carlo Burigana
Agreed by	C. BUTLER LFI Program Manager	Date: Signature:	July 15th, 2003 R.C. Butler
Approved by	N. MANDOLESI LFI Principal Investigator	Date: Signature:	July 15th, 2003

DISTRIBUTION LIST

Recipient	Company / Institute	E-mail address	Sent
M. BERSANELLI	Univ. Di Milano – Milano	bersanelli@uni.mi.astro.it	Yes
C. BURIGANA	IASF/CNR – Sezione di Bologna	burigana@bo.iasf.cnr.it	Yes
R. C. BUTLER	IASF/CNR – Sezione di Bologna	butler@bo.iasf.cnr.it	Yes
N. LAMA	OAT – Trieste	lama@ts.astro.it	Yes
N. MANDOLESI	IASF/CNR – Sezione di Bologna	mandolesi@bo.iasf.cnr.it	Yes
F. PASIAN	OAT – Trieste	pasian@ts.astro.it	Yes
M. SANDRI	IASF/CNR – Sezione di Bologna	sandri@bo.iasf.cnr.it	Yes
F. VILLA	IASF/CNR – Sezione di Bologna	villa@bo.iasf.cnr.it	Yes
LFISPCC		lfispcc@bo.iasf.cnr.it	Yes

CHANGE RECORD

Issue	Date	Sheet	Description of Change	Release
1.0	July 15TH, 2003	All	First Issue of Document	===

USEFUL ACRONYMS

Acronym	Description
GO	Geometrical Optics
GTD	Geometrical Theory of Diffraction
PO	Physical Optics
PTD	Physical Theory of Diffraction
MrGTD	Multi Reflector GTD
RDP	Reference Detector Plane
LOS	Line Of Sight
ET	Edge Taper
FM	Flight Model
QM	Qualification Model

TABLE OF CONTENTS

1 Introduction

This note is to describe the data format in which LFI beams are delivered from the LFI System Team to the DPC. Essentially, the format is the GRASP8 one [1]. In this framework not all possible values for the involved GRASP8 variables are explained. Only the useful values for PLANCK/LFI beam simulations are discussed. In the original GRASP8 ASCII output file (**.grd** or **.cut**, as explained in the sections 2 and 3) a header is added, in which the relevant beam simulation characteristics are reported.

LFI main beams are computed in spherical UV–grids whereas intermediate and full beams are computed in spherical polar cuts. Each beam, both UV–grid or cuts, is computed with respect its own coordinate system – identified by the three angles θ_{MB}, ϕ_{MB}, ψ_{MB}, in the LOS frame [2] – in which the beam power peak falls in the centre of the UV– grid ($U_p = V_p = 0$) or at $\theta = 0$ for each cut [3].

Far fields are computed in the co– and x– polar basis according to the Ludwig's third definition [4]. A detailed description of simulation methods applied to the PLANCK/LFI case is reported in [5], whereas the transformation rules between the LOS frame and the beam frame are explicitly given in [6].

2 FIELD DATA IN RECTANGULAR UV– GRID

This format is used for storing the main beam field values in a rectangular UV– grid. The file extension is **grd**. Data points are located on a sphere and the direction to a field point **r** is related to the polar angles θ and ϕ by

$$\mathbf{r} = \mathbf{x}\sin\theta\cos\phi + \mathbf{y}\sin\theta\sin\phi + \mathbf{z}\cos\theta.$$

The field points are parametrised by two variables which are generally denoted X and Y and which run from XMIN to XMAX and from YMIN to YMAX, respectively. The file is organised so that X is varying faster than Y. The variables X and Y should be considered as general names for the actual variables which are ϕ and θ, in this case. The file format is:

```
Line  Contents Format

0 HEADER (characters)
    BEAM TYPE - Main Beam
    FEED NUMBER - From 18 to 27
    FEED TYPE - GAU (Gaussian) or SWE (Spherical Wave Expansion) or TAB
      (tabulated pattern)
    FEED POLARIZATION - X or Y
    FEED DESIGN - QM (Qualification Model) or FM (Flight Model) or ID (from 00
      to 99). In all cases the Edge Taper is specified between brackets
    U V - power peak location with respect to the LOS
    THETA PHI PSI - angles defining the MB frame with respect to the LOS,
      according to the GRASP8 convention
    THETA PHI PSI - angles defining the MB frame with respect to the LOS,
      according to the LFI flight simulator convention
    SIMULATION METHODS - GO/GTD+PO or FPO
    SOFTWARE - GRASP8
    ORIGINATOR - Institution and Name
    DATE - dd/mm/yy


1 TEXT (characters)
```

TEXT - Line with comments

A line containing ++++ as the first 4 characters ends these two sections.

2 **KTYPE** (integer)
KTYPE = 1 - standard format for 2D grid

3 **NSET**, **ICOMP**, **NCOMP**, **IGRID** (4 integers)
NSET = 1 - Number of field sets or beams
ICOMP = 3 - Linear co and cx components (Ludwig's third definition)
NCOMP = 2 - The file contains two field components for each point
IGRID = 1 - uv-grid type: (X,Y) = (u,v) where u and v are the two first coordinates of the unit vector to the field point. Hence, **r**(u,v,$\sqrt{(1-u^2-v^2)}$). u and v are related to the spherical angles by u = $\sin\theta\cdot\cos\phi$ and u = $\sin\theta\cdot\sin\phi$

4 **IX**, **IY** (2 integers)
IX = 0 - Beam centre in u
IY = 0 - Beam centre in v

5 **XS**, **YS**, **XE**, **YE** (4 real numbers)
Limits of 2D grid. The grid points (X,Y) run through the values
X = XCEN + XS +DX*(I-1) and Y = YCEN + YS +DY*(J-1) where
DX = (XE-XS)/(NX-1), DY = (YE-YS)/(NY-1) and XCEN = DX*IX, YCEN = DY*IY .
The number of grid values NX and NY and the range of the index I and J are defined in the following lines.

6 **NX**, **NY**, **KLIMIT** (3 integers)
NX - Number of columns
NY - Number of rows
KLIMIT = 0 - Each row contains data for all NX columns

7 **F1**, **F2** (4 real numbers on each line)
F1,F2 - Complex field with two components. Since ICOMP = 3, F1 and F2 contain the real and imaginary parts of the field in linear scale.

-----end of data file-----

2.1 Example of a GRD file

The first part of a **grd** file storing a LFI main beam is reported, as example.

```
Main Beam
27
SWE
X
FM (ET 30dB@22°)
-0.06789 0.03369
4.3466 153.6074 -22.5
4.3466 153.6074 337.5
GO/GTD + PO
GRASP8
IASF/CNR Sezione di Bologna (Italy) - M.Sandri
10/04/03
Main Beam LFI27 SWE X-POL FM
++++
```

```
   1
   1    3    2    1
   0    0
  -0.260000000E-01  -0.260000000E-01   0.260000000E-01   0.260000000E-01
 301 301    0
   0.266168536E+00  -0.188242461E+00  -0.118328029E+00   0.804270343E-01
   0.250234160E+00  -0.183469660E+00  -0.121881117E+00   0.906006049E-01
   0.232528869E+00  -0.178177050E+00  -0.124944119E+00   0.100390631E+00
   ...............  ................  ................   ...............
[cut of 90596 lines]
```

3 FIELD DATA IN POLAR CUTS

This format is used for storing the intermediate and far beam field values in spherical cuts. The file extension is **cut**. A cut consists of the records 1, 2 and 3 as described below. Since more than one cut is contained in a file all lines (except the header) are repeated for each cut.

Line Contents Format

0 **HEADER** (characters)
- BEAM TYPE - Intermediate Beam or Full Beam
- FEED NUMBER - From 18 to 27
- FEED TYPE - GAU (Gaussian), SWE (Spherical Wave Expansion), TAB (tabulated pattern)
- FEED POLARIZATION - X or Y
- FEED DESIGN - QM (Qualification Model) or FM (Flight Model) or ID (from 00 to 99). In all cases the Edge Taper is specified between brackets
- U V - power peak location with respect to the LOS
- THETA PHI PSI - angles defining the MB frame with respect to the LOS, according to the GRASP8 convention
- THETA PHI PSI - angles defining the MB frame with respect to the LOS, according to the LFI flight simulator convention
- SIMULATION METHODS - MrGTD or FPO
- SOFTWARE - GRASP8
- ORIGINATOR - Institution and Name
- DATE - dd/mm/yy

The following lines are repeated for each cut stored in the CUT file.

1 **TEXT** (characters)
- TEXT - Line with comments

2 **V_INI, V_INC, V_NUM, C, ICOMP, ICUT, NCOMP** (2 real numbers, 1 integer, 1 real number, 3 integers)
- V_INI - Initial value of θ
- V_INC - Increment in θ
- V_NUM - Number of values in cut
- C - Constant angle (ϕ in degrees)
- ICOMP = 3 - Linear co and cx components (Ludwig's third definition)
- ICUT = 1 - standard polar cut where ϕ is fixed (C) and θ is the varying (V_)
- NCOMP = 2 - The file contains two field components for each point

3 **(F1(I),F2(I), I = 1, V_NUM)** (4 real numbers on each line)

```
F1,F2 - Complex arrays containing the two components of the field for the
  I'th data point. V = V_INI + V_INC*(I-1)

                  -----end of data file-----
```

3.1 Example of a CUT File

The first part of the first three cuts included in a CUT file storing a LFI full beam are reported, as example.

```
Full Beam
27
SWE
X
FM (ET 30dB@22°)
-0.06789 0.03369
4.3466 153.6074 -22.5
4.3466 153.6074 337.5
MrGTD
GRASP8
IASF/CNR Sezione di Bologna (Italy) - M.Sandri
16/04/03
Full Beam LFI27 SWE X-POL FM
++++
  -0.180000000E+03   0.200000000E+01  181   0.000000000E+00    3    1    2
   0.255416704E-02  -0.450266894E-02  -0.989735312E-03   0.469252023E-03
   0.500849070E-02  -0.434351666E-03   0.122292814E-03   0.690814680E-04
   ...............  ................  ................   ...............
[cut of 179 lines]
Full Beam LFI27 SWE X-POL FM
  -0.180000000E+03   0.200000000E+01  181   0.200000000E+01    3    1    2
   0.265234509E-02  -0.452665368E-02  -0.684405339E-03  -0.602623128E-04
   0.500660719E-02  -0.859806843E-03   0.113539829E-02  -0.684379231E-04
   ...............  ................  ................   ...............
[cut of 179 lines]
Full Beam LFI27 SWE X-POL FM
  -0.180000000E+03   0.200000000E+01  181   0.400000000E+01    3    1    2
   0.265234509E-02  -0.452665368E-02  -0.684405339E-03  -0.602623128E-04
   0.451212091E-02  -0.115271066E-02   0.116218865E-02  -0.314941344E-03
   ...............  ................  ................   ...............
[cut]
```

4 FILE NAME SPECIFICATION

In order to univocally identify a beam file by its name a standard labelling is adopted. File names meet the follows rule:

"beam type" + **"feed label"** + **"feed type"** + **"feed polarization"** + **"feed design"**

where,

- beam type - MB (Main Beam) or IB (Intermediate Beam) or FB (Full Beam)
- feed label - LFI# (#=18÷27)
- feed type - GAU (Gaussian) or SWE (Spherical Wave Expansion) or TAB (tabulated pattern)
- feed polarization - X or Y
- feed design - QM (Qualification Model) or FM (Flight Model) or ID (from 00 to 99)

For example, this is the complete set of simulations done for the LFI27 at 30 GHz:

- MB_LFI27_SWE_X_FM.grd
- IB_LFI27_SWE_X_FM.cut
- FB_LFI27_SWE_X_FM.cut

while files corresponding to analogous simulations done for the LFI23 at 70 GHz are:

- MB_LFI23_TAB_Y_QM.cut
- IB_LFI23_TAB_Y_QM.cut
- FB_LFI23_TAB_Y_QM.cut

5 REFERENCES

[1] K. Pontoppidan, *Technical Description of GRASP8*, TICRA, Doc.No.S-894-02, 1999.
[2] FIRST/PLANCK Project, *PLANCK Telescope Design Specification*, SCI-PT-RS-07024.
[3] M. Sandri and F. Villa, *PLANCK/LFI: Main Beam Locations and Polarization Alignment for the LFI baseline FPU*, PL-LFI-ST-TN-027, 2001.
[4] C. Ludwig, *The Definition of Cross Polarization*, IEEE Transactions on Antennas and Propagation, pp.116-119, Jan 1973.
[5] M. Sandri, F. Villa, R. Nesti, C. Burigana, M. Bersanelli, N. Mandolesi, *Trade-off between angular resolution and straylight contamination in CMB anisotropy experiments. Paper I. Pattern Simulations*, submitted to A&A, 2003.
[6] C. Burigana, M. Sandri, F. Villa, D. Maino, R. Paladini, C. Baccigalupi, M. Bersanelli, N. Mandolesi, *Trade-off between angular resolution and straylight contamination in CMB anisotropy experiments. Paper II. Straylight evaluation*, submitted to A&A, astro-ph/0303645, 2003.

6 APPENDIX

6.1 F90 routine for reading GRD files

Example of routine for reading a grid file generated by GRASP8 [1], written in F90.

```
SUBROUTINE READ_FIELD_GRID
! Example of reading field data from a 2D-grid file
! declarations
REAL :: &
XS, YS, XE, YE ,& ! Grid limits
A1, A2, A3, A4, A5, A6 ! Temporary variables
INTEGER :: &
UNIT_NO=15, & ! Unit no of open input file
IER , & ! Error code
KTYPE , & ! Type of file format
NSET , & ! Number of data sets
ICOMP , & ! Control of field components
NCOMP , & ! Number of field components
IGRID , & ! Control of grid type
I_SET , & ! Index of data set
NX, NY , & ! Number of grid points
KLIMIT , & ! Limits of 2D grid
IS, IN, IE, & ! Limits and number of points in row
I, J ! Loop index
CHARACTER(LEN=72) :: &
MESS , & ! message
TEXT , & ! File header
FILE_FORM ! Format of file
COMPLEX :: AJ = (0.,1.)
COMPLEX, POINTER, SAVE :: &
F1(:,:), & ! Array for storing first component of field data
F2(:,:), & ! Array for storing second component of field data
F3(:,:) ! Array for storing third component of field data
INTEGER, POINTER, SAVE :: &
I_X(:), I_Y(:) ! Beam centers
LOGICAL :: INIT=.TRUE. ! Initialisation flag
! Initialisation
IF (INIT) THEN
NULLIFY( &
I_X, I_Y, &
F1, F2, F3)
INIT = .FALSE.
ENDIF

IF (FILE_FORM == 'formatted') THEN
! Formatted input
DO
READ(UNIT_NO, '(A)', ERR=9990, END=9990) TEXT
IF (TEXT(1:4) == '++++') EXIT
END DO
READ(UNIT_NO, *, ERR=9990, END=9990) KTYPE
READ(UNIT_NO, *, ERR=9990, END=9990) NSET, ICOMP, NCOMP, IGRID
ELSE
! Unformatted data
DO
READ(UNIT_NO, ERR=9990, END=9990) TEXT
IF (TEXT(1:4) == '++++') EXIT
ENDDO
READ(UNIT_NO, ERR=9990, END=9990) KTYPE
READ(UNIT_NO, ERR=9990, END=9990) NSET, ICOMP, NCOMP, IGRID
ENDIF
! storage allocation
IF (ASSOCIATED(I_X)) DEALLOCATE(I_X)
IF (ASSOCIATED(I_Y)) DEALLOCATE(I_Y)
ALLOCATE(I_X(NSET),I_Y(NSET))
```

```
IF (FILE_FORM == 'formatted') THEN
! Formatted input
DO I = 1, NSET
READ(UNIT_NO, *, ERR=9990, END=9990) I_X(I), I_Y(I)
END DO
ELSE
! Unformatted data
DO I = 1, NSET
READ(UNIT_NO, ERR=9990, END=9990) I_X(I), I_Y(I)
END DO
END IF
! Loop through data sets. The following lines are read NSET times
sets_loop: &
DO I_SET = 1, NSET
IF (FILE_FORM == 'formatted') THEN
! Formatted input
READ(UNIT_NO, *, ERR=9990, END=9990) XS, YS, XE, YE
READ(UNIT_NO, *, ERR=9990, END=9990) NX, NY, KLIMIT
ELSE
! Unformatted data
READ(UNIT_NO, ERR=9990, END=9990) XS, YS, XE, YE
READ(UNIT_NO, ERR=9990, END=9990) NX, NY, KLIMIT
END IF
! storage allocation
IF (ASSOCIATED(F1)) DEALLOCATE(F1)
IF (ASSOCIATED(F2)) DEALLOCATE(F2)
IF (ASSOCIATED(F3)) DEALLOCATE(F3)
ALLOCATE(F1(NX,NY), F2(NX,NY), F3(NX,NY), STAT=IER)
IF (IER /= 0) THEN
MESS = 'Could not allocate enough memory'; GO TO 9998
END IF
! Loop through columns
column_loop: &
DO J = 1, NY
IF (KLIMIT == 1) THEN
IF (FILE_FORM == 'formatted') THEN
! Formatted input
READ(UNIT_NO, *, ERR=9990, END=9990) IS, IN
ELSE
! Unformatted data
READ(UNIT_NO, ERR=9990, END=9990) IS, IN
END IF
IE = IS + NX - 1 ELSE
IS = 1
IE = NX
END IF
(FILE_FORM == 'formatted') THEN
! Formatted input
IF (NCOMP == 2) THEN
DO I = IS, IE
READ(UNIT_NO, *, ERR=9990, END=9990) A1, A2, A3, A4
F1(I,J) = A1+AJ*A2; F2(I,J) = A3+AJ*A4
END DO
ELSE
DO I = IS, IE
READ(UNIT_NO, *, ERR=9990, END=9990) A1, A2, A3, A4, A5, A6
F1(I,J)=A1+AJ*A2; F2(I,J)=A3+AJ*A4; F3(I,J)=A5+AJ*A6
END DO
END IF
ELSE
! unformatted
IF (NCOMP == 2) THEN
READ(UNIT_NO, ERR=9990, END=9990) &
(F1(I,J), F2(I,J), I=IS,IE)
ELSE
READ(UNIT_NO, ERR=9990, END=9990) &
(F1(I,J), F2(I,J), F3(I,J), I=IS,IE)
END IF
END IF
```

```
ENDDO column_loop
ENDDO sets_loop
GO TO 9999
! Errors
9990 CONTINUE; MESS='Error in field data file'
9998 CONTINUE; IER=1
CALL OM_ERROR('READ_FIELD_GRID' ,'FATAL' , MESS)
9999 CONTINUE
END SUBROUTINE READ_FIELD_GRID
```

6.2 C routine for reading GRD files

Example of routine for reading a LFI main beam stored in a grid file, written in C.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void main(int argc, char *argv[])
{
  int i,j;
  char str[80]="\0";
  int KTYPE;
  int NSET,KCOMP,NCOMP,KGRID;
  int IX,IY;
  int NX,NY,KLIMIT;
  double XS,YS,XE,YE;
  double XR,XI,YR,YI;
  char grdfile[80];
  FILE *grd;

  if (argc != 2) {
    printf("Insert filename.grd \n");
    exit(1);
  }

  strcpy(grdfile,argv[1]);
  grd=fopen(grdfile,"r");

  while (strstr(str,"++++") == NULL) {
    fgets(str,80,grd);
  }

  fscanf(grd,"%d",&KTYPE);
  fscanf(grd,"%d %d %d %d",&NSET,&KCOMP,&NCOMP,&KGRID);
  fscanf(grd,"%d %d",&IX,&IY);
  fscanf(grd,"%lf %lf %lf %lf",&XS,&YS,&XE,&YE);
  fscanf(grd,"%d %d %d",&NX,&NY,&KLIMIT);

  for(j=0;j<NY;j++) {
    for(i=0;i<NX;i++) {
      fscanf(grd,"%lf %lf %lf %lf",&XR,&XI,&YR,&YI);
    }
  }

  fclose(grd);
}
```