NATIONAL RADIO ASTRONOMY OBSERVATORY

POST OFFICE BOX 2 GREEN BANK, WEST VIRGINIA 24944
TELEPHONE 304 456 2011 TWX 710 938 1530

ELECTRONICS DIVISION TECHNICAL NOTE NO. 121

Title: Program for Computing Focal Plane Field Distribution from Far Field Antenna Pattern

Author(s): J. Richard Fisher

Date: July 26, 1984

DISTRIBUTION:

GB	CV	TU	VLA
GB Library	CV Library	Library Downtown	VLA Library
W. Brundage	H. Hvatum	Library Mountain	P. Napier
R. Weimer	M. Balister	J. Payne	J. Campbell
D. Schiebel	S. Weinreb		
E. Childers	C. Burgess		
S. Srikanth			
C. Brockway			
J. Coe			
G. Behrens			
R. Mauzy			
R. Norrod			
R. Bradley			
R. Fisher			
F. Crews			
B. Peery			
R. Lacasse			

OPERATED BY ASSOCIATED UNIVERSITIES, INC., UNDER CONTRACT WITH THE NATIONAL SCIENCE FOUNDATION

PROGRAM FOR COMPUTING FOCAL PLANE FIELD DISTRIBUTION FROM FAR FIELD ANTENNA PATTERN

J. Richard Fisher

A Fortran program for computing the relative amplitude and phase distribution of the electric field in the vicinity of the phase center of a far field antenna pattern is now working on the Green Bank MASSCOMP computer. Program input consists of a one-dimensional set of phase and amplitude values as a function of polar angle which describe the far field pattern plus a few other parameters such as wavelength and offset of the computed field plane from the far field phase reference point. This will be explained in connection with Table 1. A two-dimensional pattern is created by sweeping the far field pattern around its polar axis.

The program writes the two-dimensional focal plane field distribution to a UNIX file called 'splita.dat', and two orthogonal cross sections through the center of this plane are sent to the CRT terminal. A second program is available for computing the ratio of field power within a specified diameter circle to the power in the whole computed plane. The far field to focal plane field conversion is done with a two-dimensional Fourier transform.

The program may be run by typing

r datafile <CR>

where datafile is the name of the file containing the formatted input data, e.g., sl15k+.dat in the example in Table 1. The command r is a UNIX shell script which invokes the main program "splitsize" and all of its attendant subroutines. All of these files are in the rick/Splitsubs directory. The power ratio program requires the simple command

p <CR>

after which you will be asked to specify the circle diameter for which the ratio is to be calculated. The command p is also a shell script which runs the main program called "powerratio".

The source code for "splitsize" is fairly heavily commented so it should not be too difficult to follow the steps of its operation from the listing at the end of this memo. You are welcome to copy and modify this program as you like, but please do not change the original or its subroutines. Two-dimensional input data could be relatively easily accommodated by bypassing the one-dimensional interpolation and rotation routines.

The format for the input file is as follows (Table 1):

The first line is an alphanumeric identifier of any form up to 80 characters.

The second line is self-explanatory except for the fact that the numeric quantity must not start any further to the left than shown, i.e., do not abbreviate the label since those character locations are ignored by the input routine. This is also true for lines three and four. Following the equal sign the format is F10.7 with the explicit decimal taking precedence.

Line three is the linear offset along the far field pattern axis of the plane in which the field distribution is to be computed.

Negative offsets are in the direction of the beam pattern. For very large offsets the size of the transform array must be increased from 64 x 64 to prevent undersampling of the induced phase curvature. With the relatively narrow input beam pattern in Table 1 offsets of a couple of hundred wavelengths and more required a larger transform. Wider beams will produce errors at smaller offsets. Real running time for the program is on the order of two minutes with a relatively busy MASSCOMP using a 64 x 64 transform. This does not increase terribly with a 128 x 128 array. About half the time is taken up with the large output to disk.

Line four is half of the angular range of the far field pattern used in the transform. Specifying 30^{o} means that the transform input is +- 30^{o} from the pattern axis. This value must be less than the largest angle in the input pattern. With 30^{o} the sample spacing near the focal plane is one wavelength. Smaller pattern ranges give larger spacings.

Line five is a character string which controls the disposition of the input pattern. Averaging of two sides of an input pattern or averaging E and H plane patterns can be performed if desired. The possibilities are given in the "params" routine listing. A subroutine was written to handle different E and H plane patterns (ellippat), but this has yet to be implemented in the main program.

Line six is a header label ignored by the input routine.

Line seven and beyond is the far field pattern with one point per line in the format 3F16.7. The points need not be evenly spaced nor in numerical order of angle. Portions of the pattern with quickly changing values may need more frequent sampling. The E plane pattern is terminated with an angle value greater than 200 after which an H plane pattern may be specified. The input routine can take 50 points in each plane, but there is no fundamental reason why this could not be increased.

TABLE 1

Jul 16 14:43 1984 s115K+.dat Page 1

High band ellipsoid 1.15 cm, K-band 134-164 degrees
Wavelength (cm)=1.15
Phase Center Offset (cm)=-70.0
Pattern Range (deg)=30.
Positive E Plane only

Angle (deg)	Amplitude (dB)	Phase (deg)	
0.0	2.016	41.7	E-plane
1.0	1.587	41.9	
2.0	0.303	42.0	
3.0	-1.436	41.9	
4.0	-3.435	42.9	
5.0	-5.515	46.0	
6.0	-7.673	52.3	
7.0	-10.180	60.5	
8.0	-12.778	71.0	
9.0	-15.648	83.3	
10.0	-18.151	98.2	
11.0	-20.633	113.4	
12.0	-23.15	133.9	
13.0	-23.97	156.0	
14.0	-27.51	-151.0	
15.0	-39.92	-145.5	
16.0	-36.75	-154.1	
17.0	-36.20	-124.7	
18.0	-37.41	-9.0	
19.0	-34.33	120.2	
20.0	-44.78	-73.9	
21.0	-40.08	128.4	
22.0	-37.61	15.7	
23.0	-35.38	-149.7	
24.0	-43.13	127.4	
25.0	-30.97	-4.0	
26.0	-51.70	-19.7	
27.0	-32.94	-79.9	
28.0	-33.54	-174.4	
29.0	-42.32	144.3	
30.01	-36.91	-121.1	

999.

```
      PROGRAM SPLITSIZE
*
*        This program computes the power distribution in a plane near the
*   140-foot cassegrain focus starting with the subreflector illumination
*   pattern after scattering from the ellipsoidal refocusing reflector above
*   the cassegrain feeds.
*
      IMPLICIT LOGICAL(A-Z)
*
      CHARACTER Ident*80
*
      INTEGER I,J,Aveparam,NEplane,NHplane,Xformsize,Gridsize
*
      REAL Wavelength,PhCtrOffset,PatRange,Eplane(3,100),Hplane(3,100),
     x     Grid(3,513),Tgrid(3,513),Fpscale(257),Ampang(2,64,64)
*
      COMPLEX Fftarray(64,64)
*
      EQUIVALENCE (Ampang,Fftarray)
*
*       Read the input parameters and the far-field illumination pattern.
*
      CALL PARAMS(Ident,Wavelength,PhCtrOffset,PatRange,Aveparam,Eplane,
     x            Hplane,NEplane,NHplane)
*
*         Hardwire the transform array size
*
      Xformsize=64
*
*         Set size of one-dimensional grid.
*
      Gridsize=256
*
*       Ident is a character string identifier label.
*       PhCtrOffset is the displacement of the computed field plane from the
*         far-field phase reference point.
*       PatRange is the polar angle in degrees of the extreme far-field data
*         point to be used in the focal field calculation.  The far-field
*         pattern is forced to be symmetric about the axis.
*       Aveparam designates the averaging to be performed on the input
*         far-field pattern.  See the listing for subroutine PARAMS.
*       Eplane and Hplane are the input far-field patterns whose data may be
*         in any order and at any uneven spacing so long as it extends beyond
*         PatRange.  Eplane(1, ) is the polar angle, Eplane(2, ) is the
*         amplitude in dB, and Eplane(3, ) is the phase in degrees between
*         -180 and +180.
*       NEplane and NHplane are the numbers of input data points in the
*         far-field patterns.
*
*       Now sort the data arrays.
*
      IF(NEplane.GT.1) THEN
        CALL SORTARRAY(Eplane,NEplane)
      END IF
      IF(NHplane.GT.1) THEN
        CALL SORTARRAY(Hplane,NHplane)
```

```
      END IF
*
*       Interpolate into evenly spaced arrays and average arrays as requested.
*
      IF(Aveparam.EQ.1) THEN
        CALL GRID1D(Eplane,Grid,NEplane,2*Gridsize+1,-PatRange,PatRange)
        CALL PLMINAVG(Grid,2*Gridsize+1)
      END IF
      IF(Aveparam.EQ.2) THEN
        CALL GRID1D(Hplane,Grid,NHplane,2*Gridsize+1,-PatRange,PatRange)
        CALL PLMINAVG(Grid,2*Gridsize+1)
      END IF
      IF(Aveparam.EQ.3) THEN
        CALL GRID1D(Eplane,Grid,NEplane,Gridsize+1,0.,PatRange)
      END IF
      IF(Aveparam.EQ.4) THEN
        CALL GRID1D(Hplane,Grid,NHplane,Gridsize+1,0.,PatRange)
      END IF
      IF(Aveparam.EQ.5) THEN
        CALL GRID1D(Eplane,Grid,NEplane,2*Gridsize+1,-PatRange,PatRange)
        CALL GRID1D(Hplane,Tgrid,NHplane,2*Gridsize+1,-PatRange,PatRange
     x              )
        CALL EHAVG(Grid,Tgrid,2*Gridsize+1)
        CALL PLMINAVG(Grid,2*Gridsize+1)
      END IF
      IF(Aveparam.EQ.6) THEN
        CALL GRID1D(Eplane,Grid,NEplane,Gridsize+1,0.,PatRange)
        CALL GRID1D(Hplane,Tgrid,NHplane,Gridsize+1,0.,PatRange)
        CALL EHAVG(Grid,Tgrid,Gridsize+1)
      END IF
*
*         Shift the field computation plane the specified distance from the
*       far-field reference phase point.
*
      CALL PHCTRMOD(Grid,Gridsize+1,Wavelength,PhCtrOffset)
*
*         Compute the linear scale of the computed field plane.
*
      CALL FOCSCALE(Fpscale,Wavelength,PatRange,Xformsize)
*
*         Map the one-dimensional far-field array into two dimensions by
*       rotating it around its low end.
*
      CALL CIRCPAT(Grid,Ampang,Gridsize+1,Xformsize,64)
*
*         Add 180 degrees of phase to every other data point in the
*       two-dimensional array to move the computed field pattern to the
*       center of the output plane.
*
      CALL MODULATOR(Ampang,Xformsize,64)
*
*         Change array from (dB,angle) to complex voltages.
*
      CALL DBTOCOMPLEX(Ampang,Fftarray,Xformsize,64)
*
*         Transform the far-field pattern to focal plane fields.
```

```
*
      CALL CFFT2(Fftarray,Xformsize,Xformsize,64,+1)
*
*        Print cross sections of the results through the center of the plane.
*
      CALL PRINTXY(Fpscale,Fftarray,Xformsize,64)
*
*        Save data for pattern integration filter
*
      OPEN(UNIT=7,FILE='splita.dat',STATUS='NEW')
      WRITE(7,'(I10)') Xformsize
      WRITE(7,'(5F10.4)') (Fpscale(I),I=1,Xformsize)
      WRITE(7,'(6F10.5)') ((Fftarray(I,J),I=1,Xformsize),
     x                          J=1,Xformsize)
      CLOSE (7)
      END
```

```
      SUBROUTINE PARAMS(Ident,Wavelength,PhCtrOffset,PatRange,Aveparam,
     :                  Eplane,Hplane,NEplane,NHplane)
*
*       This subroutine interprets the input file for the FFT from a beam
*  pattern to a field pattern near the focal plane.
*
      IMPLICIT LOGICAL(A-Z)
*
      INTEGER Aveparam,NEplane,NHplane,I,J
*
      REAL Wavelength,PhCtrOffset,PatRange,Eplane(3,50),Hplane(3,50)
*
      CHARACTER Ident*80,Alpha*30
*
      READ(*,'(80A)') Ident
      READ(*,'(16X,F10.7)') Wavelength
      READ(*,'(25X,F10.6)') PhCtrOffset
      READ(*,'(20X,F10.7)') PatRange
      READ(*,'(30A)') Alpha
*            Change the AVEPARAM phrase to all capital letters.
      CALL CAPS(Alpha)
*
*            Possible phrases for the averaging control input:
*              Aveparam=1 "Average E plane"
*                       2 "Average H plane"
*                       3 "Positive E plane"
*                       4 "Positive H plane"
*                       5 "Average E and H plane"
*                       6 "Average positive E an H"
*
      Aveparam=0
*            Check for the word "POSITIVE".
      IF(INDEX(Alpha,'POSIT').NE.0) THEN
*            Check for the word "AVERAGE".
        IF(INDEX(Alpha,'AVER').NE.0) THEN
          Aveparam=6
        ELSE
*            Check for the word " H" for H plane.
          IF(INDEX(Alpha,' H').NE.0) THEN
            Aveparam=4
          ELSE
            Aveparam=3
          END IF
        END IF
      ELSE
*            Check for the phrase "E AND H" for E and H planes
        IF(INDEX(Alpha,'E AND H').NE.0) THEN
          Aveparam=5
        ELSE
*            Check for the word " H" for H plane
          IF(INDEX(Alpha,' H').NE.0) THEN
            Aveparam=2
          ELSE
            Aveparam=1
          END IF
        END IF
```

```
      END IF
*           Skip the next card which is a title card
      READ(*,'(A)') Alpha
      DO 1 I=1,50
        READ(*,'(3F16.7)',END=2) (Eplane(J,I),J=1,3)
        IF(Eplane(1,I).GT.200.) THEN
          GOTO 5
        END IF
    1 CONTINUE
    2 NEplane=I-1
      RETURN
    5 NEplane=I-1
      DO 3 I=1,50
        READ(*,'(3F16.7)',END=4) (Hplane(J,I),J=1,3)
        IF(Hplane(1,I).GT.200.) THEN
          GOTO 4
        END IF
    3 CONTINUE
    4 NHplane=I-1
      RETURN
      END
```