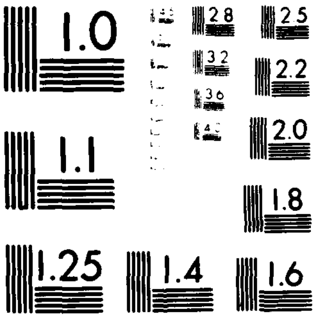
AD-A097 415 OHIO STATE UNIV COLUMBUS ELECTROSCIENCE LAB F/G 20/4
NUMERICAL ELECTROMAGNETIC CODE (NEC)-REFLECTOR ANTENNA CODE: PA--ETC(U)
SEP 79 S H LEE, R C RUDDUCK N00123-76-C-1371
UNCLASSIFIED ESL-784508-16 NL

1 OF 3



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

# NUMERICAL ELECTROMAGNETIC CODE (NEC) - REFLECTOR ANTENNA CODE
## PART II: CODE MANUAL

S. H. Lee and R. C. Rudduck

The Ohio State University
**ElectroScience Laboratory**
Department of Electrical Engineering
Columbus, Ohio 43212

Technical Report 784508-15
September 1979

[illegible]

# NOTICES

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| | |
|---|---|
| **REPORT DOCUMENTATION PAGE** | **READ INSTRUCTIONS BEFORE COMPLETING FORM** |
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO.<br>ADA097 45 |
| | 3. ~~RECIPIENT'S CATALOG NUMBER~~<br>14 ESL-784508-1 |
| 4. ~~TITLE (and Subtitle)~~<br>6 NUMERICAL ELECTROMAGNETIC CODE (NEC) - REFLECTOR ANTENNA CODE: PART ~~II~~, CODE MANUAL. | 5. ~~TYPE OF REPORT & PERIOD COVERED~~<br>9 Technical Report |
| | 6. ~~PERFORMING ORG. REPORT NUMBER~~<br>ESL 4508-16 (784508) |
| 7. AUTHOR(s)<br>10 S.H. Lee [redacted] R.C. Rudduck | 8. CONTRACT OR GRANT NUMBER(s)<br>15 N00123-76-C-1371 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering<br>Columbus, Ohio 43212 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>Project N00953/6/009121 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Naval Regional Procurement Office<br>Long Beach, California 90822 | 12. ~~REPORT DATE~~<br>11 Sep~~tember 19~~79 |
| | 13. NUMBER OF PAGES<br>206 12 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | 15. SECURITY CLASS. (of this report) |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | |
| 18. SUPPLEMENTARY NOTES | |

DTIC
ELECTE
APR 7 1981
S D
C

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

| | |
|---|---|
| Computer Code | Reflector Antenna |
| Algorithm | Aperture Integration |
| Geometrical Theory of Diffraction | Near Field |
| Far Field Pattern | |

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

> The code manual documents a detailed explanation of the Numerical Electromagnetic Code - Reflector Antenna Code by which the near field and far field of a typical Navy reflector antenna can be calculated. One important feature of the code is the capability for a general reflector rim shape. Another important feature is the capability to input a practically arbitrary volumetric feed pattern. Only the class of parabolic surfaces was implemented in the computer code.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

40

> The theoretical approach for computing the fields of the general reflector is based on a combination of the Geometrical Theory of Diffraction (GTD) and Aperture Integration (AI) techniques. Typically, AI is used to compute the main beam and near sidelobes; GTD is used to compute the wide-angle sidelobes and the backlobes.

The theoretical background on which the computer algorithms are based is described along with descriptions of the main program and the various subroutines. For each subsection of the main program and subroutine, the purpose and method are included, accompanied by a flow diagram, a key variable list and a listing of the code.

## TABLE OF CONTENTS

# I. INTRODUCTION

This code manual describes the Numerical Electromagnetic Code - Reflector Antenna Code by which the near field and far field of a typical Navy reflector antenna can be calculated. One important feature of the code is the capability for a general reflector rim shape. Another important feature is the capability to input a practically arbitrary volumetric feed pattern.

Since many Navy reflector antennas have parabolic surfaces, only the class of parabolic surfaces was implemented in the computer code. The geometry of the reflector rim is treated as piece-wise linear. The code for the reflector geometry is flexible enough to include offset fed reflectors and general reflector rim shapes such as elliptical and rectangular with chopped corners.

The theoretical approach for computing the fields of the general reflector is based on a combination of the Geometrical Theory of Diffraction (GTD) and Aperture Integration (AI) techniques. Typically, AI is used to compute the main beam and near sidelobes; GTD is used to compute the wide-angle sidelobes and the backlobes. To implement the computer algorithms based on these theories, efficient ways were developed to handle calculations involving the feed pattern, the aperture field and the far field pattern computation.

Sampled data from each measured feed pattern cut is input and stored in the code. Linear interpolation is then used to obtain a piece-wise linear representation of the input pattern cut. The feed patterns in planes other than those corresponding to the input pattern cuts also are calculated by linear interpolation. This method provides a computationally efficient way of calculating the aperture field without requiring large amounts of computer storage for the measured feed pattern. Only relatively few data points need to be stored for essentially complete feed pattern information. Furthermore, the piece-wise linear method has the advantages of flexibility and simplicity for general feed patterns. No cut-and-try procedures are needed; the sample feed values can be obtained directly from measured feed pattern data.

The aperture fields are calculated and stored on the principal grid for use in the aperture integration. The principal grid values are used for all output pattern cuts. The aperture fields are calculated at points off the principal grid by using linear interpolation from the principal grid. This is more efficient than calculating the aperture fields from the feed pattern for each rotated grid that is used for off-principal plane cuts.

The aperture integration uses an approach of overlapping subapertures which allows a piece-wise linear representation for the aperture distribution. Thus variations in the aperture fields can be represented with relatively few subapertures. Furthermore, the subapertures can be electrically large; thus minimizing the computer storage and also the amount of numerical integration required. For far field computations, a rotating grid method is employed in that the y-integrations are carried out for each column of the aperture and each one-dimensional integration result is stored. The stored values for the y-integration are then used for each pattern angle in the plane perpendicular to the y-axis; thus the efficiency approaches that of a one-dimensional integration. Even though the integration grid must be rotated to obtain the pattern in other planes, the required grid rotation is computationally much faster than the numerous two-dimensional integrations that would otherwise be required.

The GTD and AI approaches used for the reflector code have a basic limitation on the minimum size reflector that can be modeled. This limitation is probably on the order of 1λ to 3λ for the reflector diameter. However, virtually all practical reflector antennas exceed 3λ diameter. There is no limitation on the maximum size of the reflector for the basic analysis.

This code manual documents the detailed explanation of this code except the input data section which is described in the User's Manual [1]. The theoretical background on which the computer algorithms are based is discussed in Section II. Section III consists of the actual code descriptions of the main program and the various subroutines. For each subsection of the main program and subroutine, the purpose and method are included, accompanied by a flow diagram, a key variable list and a listing of the code.

## II. BACKGROUND

### A. Aperture Integration

For aperture fields with arbitrary polarization having both x and y components, the near field can be expressed as

$$\overline{E} = \frac{jk}{2\pi} \iint [\overline{F}_x E_x^a + \overline{F}_y E_y^a] \frac{e^{-jks}}{s} \, dx \, dy$$

where $\overline{F}_x$ and $\overline{F}_y$ are the modified vector element patterns associated with two Huygen's sources (crossed electric and magnetic dipoles)[2] each having its electric field vector parallel to the X- and Y-axis, respectively. These vector element patterns are expressed by

$$\overline{F}_x = [\hat{\theta} \cos\phi - \hat{\phi} \sin\phi] \cos\left(\frac{\theta}{2}\right)$$

$$\overline{F}_y = [\hat{\theta} \sin\phi + \hat{\phi} \cos\phi] \cos\left(\frac{\theta}{2}\right)$$

The aperture integration is performed over the portion of the aperture plane inside the reflector rim. For near field computations, a rectangular grid size ($D_X$ and $D_Y$) is chosen so that the aperture can be divided into a principal rectangular grid as shown in Fig. 1. Using the approach of overlapping subapertures, the aperture is treated as a collection of overlapping subapertures. Each subaperture is rectangular in shape and consists of four adjacent grid rectangles. The aperture distribution for each subaperture is triangular. The use of overlapping, rectangular subapertures with triangular distributions permits a piecewise linear approximation to the overall aperture distribution of the reflector. Furthermore, the grid spacings $D_X$ and $D_Y$ can be electrically large, i.e., several wavelengths in size. This further minimizes the computation time. Thus the aperture integration results in a sum of the pattern functions of the rectangular subapertures weighted by the aperture field $E^a$ and their respective areas. For far field computations, the rectangular grid is rotated to form a non-orthogonal rotating grid in which the y-axis is rotated an angle $\phi$ from the principal Y-axis.

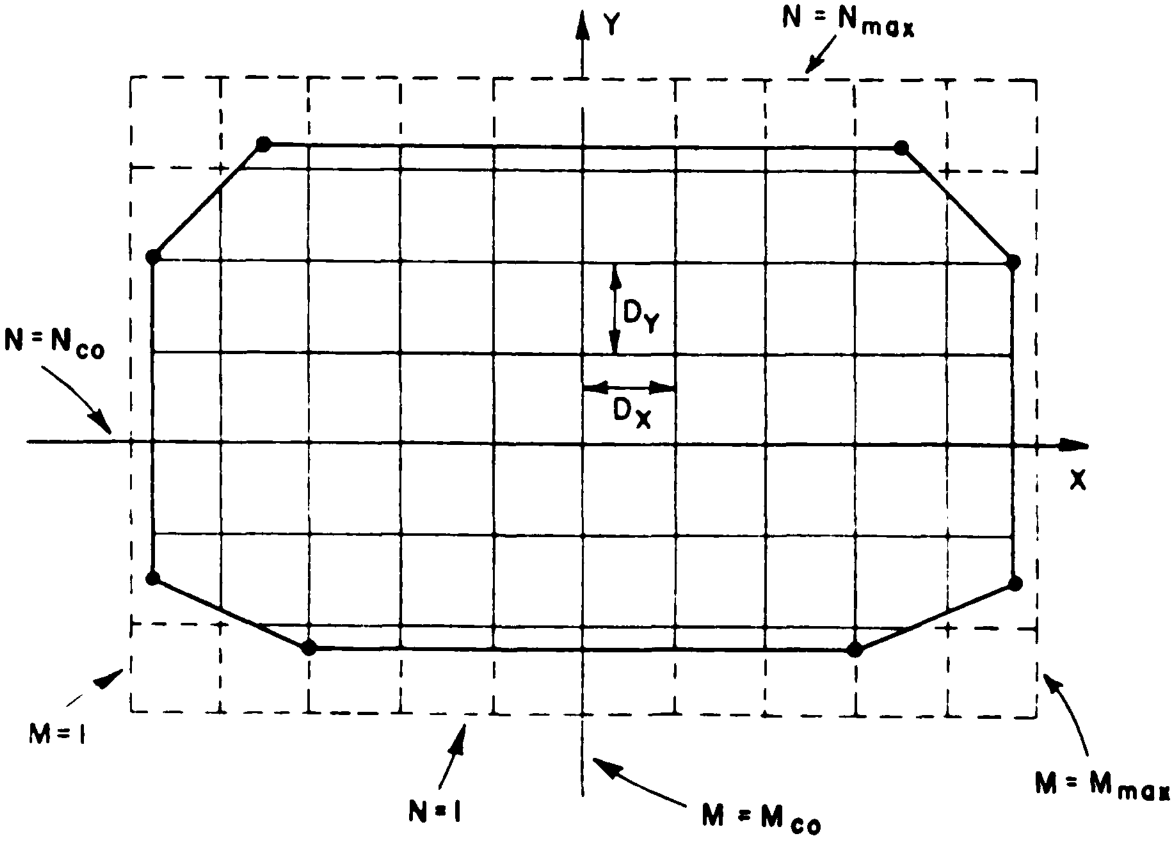


Figure 1. Geometry for principal rectangular grid.

Thus the y-integrations are independent of $\theta$ and can be stored. Consequently, the far field pattern in the plane perpendicular to the y-axis is reduced to a one-dimensional integration; this provides greatly improved efficiency over the many two-dimensional integrations that would otherwise be required. Detailed implementation of these integration techniques are given in the related sections.

## B. GTD

This section summarizes the GTD analysis. For further detail, see the section describing the subroutine GTD.

The GTD analysis of the reflector is similar to that of diffraction by a flat plate[3,4], except that the curvature of the reflector surface must be taken into account. It was found that the reflector rim must be subdivided into nearly straight segments. A suitable criterion is that each segment of the reflector rim be small enough that the focus lies in the far field of the rim segment.

The GTD method used in the reflector code increments around the rim and determines whether a diffraction occurs for each linear rim segment. This is done by comparing the diffraction angle with the bounds on the permissible range of angles. If the diffraction for that segment is not significant, the code checks the next rim segment. If the diffraction is significant, the diffraction point and the vector for the incident ray from the feed are calculated. This procedure is the same as that used for the flat plate scattering code except that the geometry information associated with the parabolic reflector surface is changed.

Once the diffraction point $X_D$ is located, the diffraction angles $\beta_0$ and $\phi$ are defined in the edge fixed coordinate system at the diffraction point. The three orthogonal unit vectors associated with this system on each segment of the reflector rim are the edge unit vector $\hat{V}$, the unit normal vector $\hat{VN}$ which is given by

$$\hat{VN} = -\hat{\rho} \sin \frac{\psi}{2} + \hat{z} \cos \frac{\psi}{2}$$

where

$$\hat{\rho} = \hat{X} \cos\phi + \hat{Y} \sin\phi$$

and the unit binormal vector $\hat{VP}=\hat{VN}x\hat{V}$ as shown in Fig. 2.

The incident angles $\beta_0'$ and $\phi'$ and the diffraction angles $\beta_0$ and $\phi$ and the associated unit vectors $\hat{\beta}_0'$, $\hat{\phi}'$, $\hat{\beta}_0$ and $\hat{\phi}$ which define the ray fixed coordinate system are determined using the incident ray unit

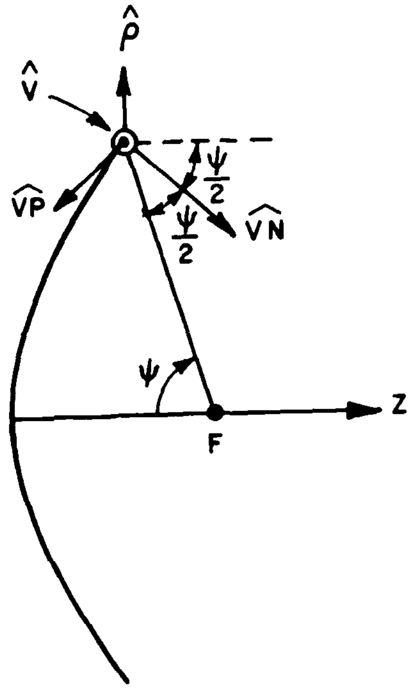


Figure 2. Unit vectors associated with the reflector rim.

vector $\hat{VI}$, the diffracted ray unit vector $\hat{d}$ and the unit vectors in the edge fixed system as given by

$$\beta_o' = \beta_o = \sin^{-1}|\hat{d}\times\hat{V}| \quad ,$$

$$\phi' = \tan^{-1}\left(\frac{-\hat{VI}\cdot\hat{VN}}{-\hat{VI}\cdot\hat{VP}}\right) \quad ,$$

$$\phi = \tan^{-1}\left(\frac{\hat{d}\cdot\hat{VN}}{\hat{d}\cdot\hat{VP}}\right) \quad ,$$

$$\hat{\phi}' = -\hat{VP}\sin\phi' + \hat{VN}\cos\phi'$$

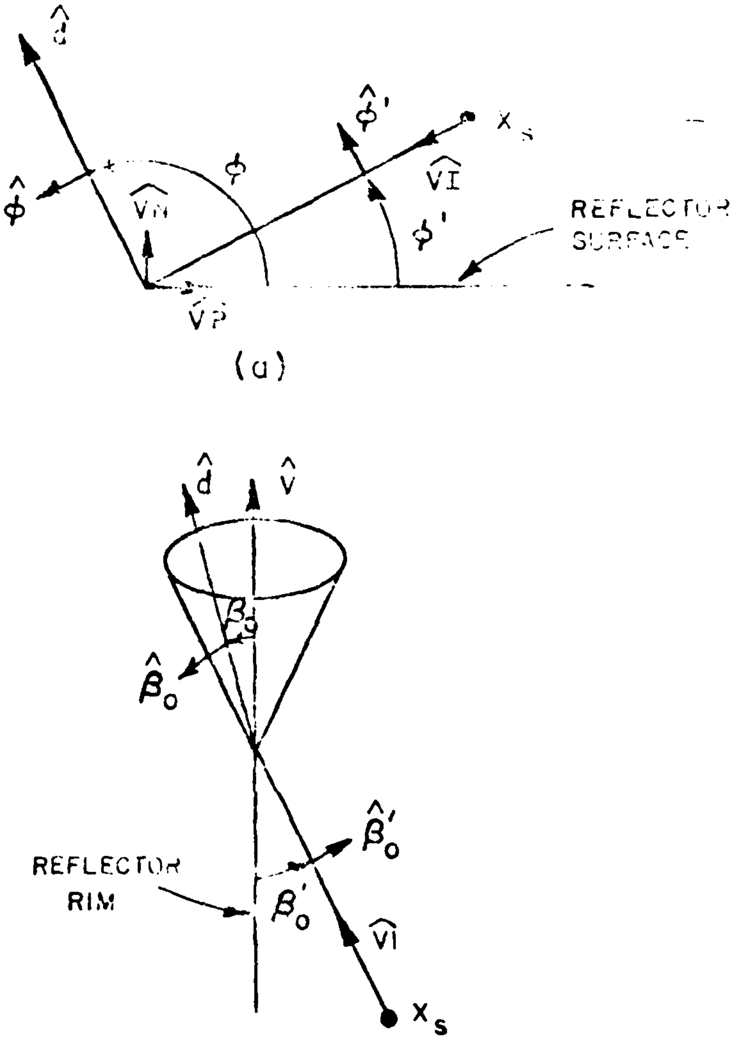
$$\hat{\phi} = -\hat{VP}\sin\phi + \hat{VN}\cos\phi$$

$$\hat{\beta}_o' = \hat{\phi}' \times \hat{VI}$$

and

$$\hat{\beta}_o = \hat{\phi} \times \hat{d}$$

as illustrated in Fig. 3.



(a)

(b)

Figure 3a,b. Geometry for three dimensional diffraction of a half plane.

Thus the edge diffracted field from each segment, expressed in parallel and perpendicular components referred to the ray fixed system, is given by[5,6]

$$E_{\parallel}^{d}(S) = -E_{\parallel}^{i}(X_D)\; D_s(L)\; A(S)\; e^{-jks}$$

$$E_{\perp}^{d}(S) = -E_{\perp}^{i}(X_D)\; D_h(L)\; A(S)\; e^{-jks}$$

where

$$D_{s,h} = \frac{e^{-j\frac{\pi}{4}}}{2\sqrt{2\pi k}\,\sin\beta_o}\left[\frac{F[kLa(\beta^-)]}{\cos\frac{\beta^-}{2}} \mp \frac{F[kL(\beta^+)]}{\cos\frac{\beta^+}{2}}\right] \quad ,$$

$$\beta^{\mp} = \phi \mp \phi' \quad ,$$

$$a = 2\cos^2\!\left(\frac{\beta}{2}\right) \quad ,$$

$$F(X) = 2j\,|\sqrt{X}|\,e^{jX}\int_{|\sqrt{X}|}^{\infty} e^{-j\tau^2}\,d\tau \text{ is the transition function,}$$

$$\begin{cases} A(S) = \sqrt{\dfrac{S'}{S(S+S')}} & \\ & \text{for near field,} \\ L = \dfrac{SS'}{S+S'}\,\sin^2\beta_o & \end{cases}$$

and

$$\begin{cases} A(S) = \dfrac{\sqrt{S'}}{S} & \\ & \text{for far field} \quad . \\ L = S'\,\sin^2\beta_o & \end{cases}$$

The slope diffracted fields are calculated in a similar way except that the slope diffraction coefficients $\partial D_s/\partial\phi'$ and $\partial D_h/\partial\phi'$ and the slope $\partial E^i/\partial n$ of the incident field at the edge are used. Thus the respective parallel and perpendicular components of the slope diffracted field are given by

$$E_{\parallel}^{sd}(S) = \frac{1}{jk\sin\beta_o}\,\frac{\partial E_{\parallel}^{i}(X_D)}{\partial n}\,\frac{\partial D_s(L)}{\partial\phi'}\,A(S)e^{-jks}$$

$$E_{\perp}^{sd}(S) = \frac{1}{jk\sin\beta_o}\,\frac{\partial E_{\perp}^{i}(X_D)}{\partial n}\,\frac{\partial D_h(L)}{\partial\phi'}\,A(S)e^{-jks}$$

where

$$\frac{\partial D_{s,h}}{\partial\phi'} = j\sqrt{\frac{k}{2\pi}}\,\frac{e^{-j\frac{\pi}{4}}L}{\sin\beta_o}\left\{\sin\left(\frac{\beta^-}{2}\right)[1-F[kLa(\beta^-)]] \pm \sin\left(\frac{\beta^+}{2}\right)[1-F[kLa(\beta^+)]]\right\} .$$

Since each rim segment is small, the diffractions from its two endpoints are significant. These diffractions are calculated by using the corner diffraction analysis developed by Burnside, et al[7]. The corner diffraction compensates for the discontinuity which occurs when the diffraction point moves off of the rim segment. The corner diffraction field is given by[7]

$$\begin{Bmatrix} E_{\parallel}^{c} \\ E_{\perp}^{c} \end{Bmatrix} = \begin{Bmatrix} IZ_o \\ MY_o \end{Bmatrix} \frac{\sin\beta_c\, e^{-j\frac{\pi}{4}}}{2\pi(\cos\beta_{oc}+\cos\beta_c)}\, F|kL_c a(\beta_{oc}+\beta_c)|\, \frac{e^{-jks_c}}{\sqrt{s_c}}\,\frac{e^{-jks_s}}{s_s}$$

where

$$\begin{Bmatrix} I \\ M \end{Bmatrix} = -\begin{Bmatrix} E_{\parallel}^{i}(X_D) \\ E_{\perp}^{i}(X_D) \end{Bmatrix} \begin{Bmatrix} C_s(X_D)Y_o \\ C_h(X_D)Z_o \end{Bmatrix} \sqrt{s'}\; e^{jks'}$$

and

$$C_{s,h}(X_D) = \frac{-e^{-j\frac{\pi}{4}}}{2\sqrt{2\pi k}\,\sin\beta_0} \left\{ \left|\frac{F|kLa(\beta^-)|}{\cos\frac{\beta^-}{2}}\right| F\left[\frac{La(\beta^-)}{kL_c a(\beta_{0c}+\beta_c)}\right] \right.$$

$$\left. \mp \left|\frac{F|kLa(\beta^+)|}{\cos\frac{\beta^+}{2}}\right| F\left[\frac{La(\beta^+)}{kL_c a(\beta_{0c}+\beta_c)}\right] \right\}$$

where

$L_c = S_c$ for far field

and

$L_c = \frac{S_c S_s}{S_c + S_s}$ for near field .

The other variables associated with geometry are shown in Fig. 4.

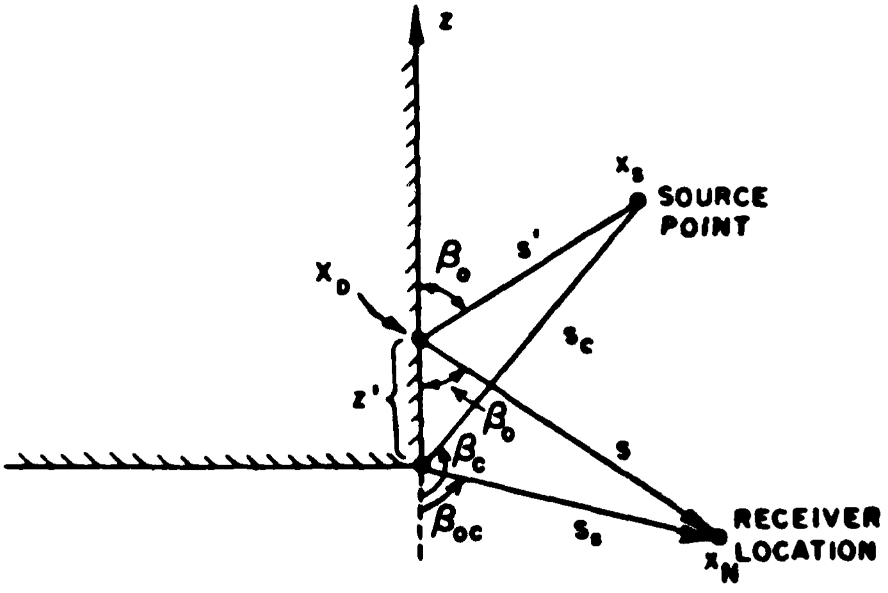


Figure 4. Geometry for corner diffraction problem.

For near field calculations, the geometrical optics reflected field must also be included in the total field if the observation point is inside the projected aperture. Since the reflected fields from a parabolic reflector are those of a plane wave with its wavefront parallel to the aperture plane, the magnitude of the reflected fields can be calculated from the aperture field and adding the appropriate phase term.

### C. OUTPUT From the Code

If the field point is in the spillover region, the feed spillover field is calculated and added to the total field from the reflector as calculated by either AI or GTD.

For far field calculations or for near field calculations with constant range, the total field is converted to principal and cross polarized components as referred to the polarization of the field components from a Huygen's source. For near field calculations with constant z, the field is still expressed in rectangular components.

Far field calculations can be made with or without the $e^{-jkR}/R$ range factor and this is controlled by the input logical variable LRANG. If the range factor is suppressed (LRANG=false) the dB output of the code is expressed as antenna gain relative to isotropic.

For far field calculations including the range factor (LRANG=true) or for near field calculations the output is expressed as the electric field relative to the field level of the feed along its axis and at a range equal to the focal distance of the reflector. In cases for which the feed axis is aligned with the reflector axis (zero feed tilt angle) this field reference is the aperture field at the center of the aperture. Thus, the power density (based on free space impedance) for these cases can be calculated from

$$S = \frac{P_T |E|^2}{F^2 P_{rad}}$$

where

$|E|$ = magnitude output of the code

$P_T$ = transmitter power (radiated)

$F$ = focal length of the reflector

$P_{rad}$ = relative power radiated by the feed (see Section 1)

The information for F and $P_{rad}$ are included in the variable

$$\mathrm{REFDB} = 10 \log \frac{4\pi\lambda^2}{F^2 P_{rad}}$$

This variable is used to calculate far field gain and is given as output from the code. Thus the power density in dB relative to 1 Watt/meter$^2$ (assuming $P_T$ is watts and $\lambda$ is meters) is given by

$$S_{dB} = 20 \log|E| + \mathrm{REFDB} + 10 \log \frac{P_T}{4\pi\lambda^2}$$

Power density calculations can be used for radiation hazard predictions or for calculating coupling in EMI predictions.

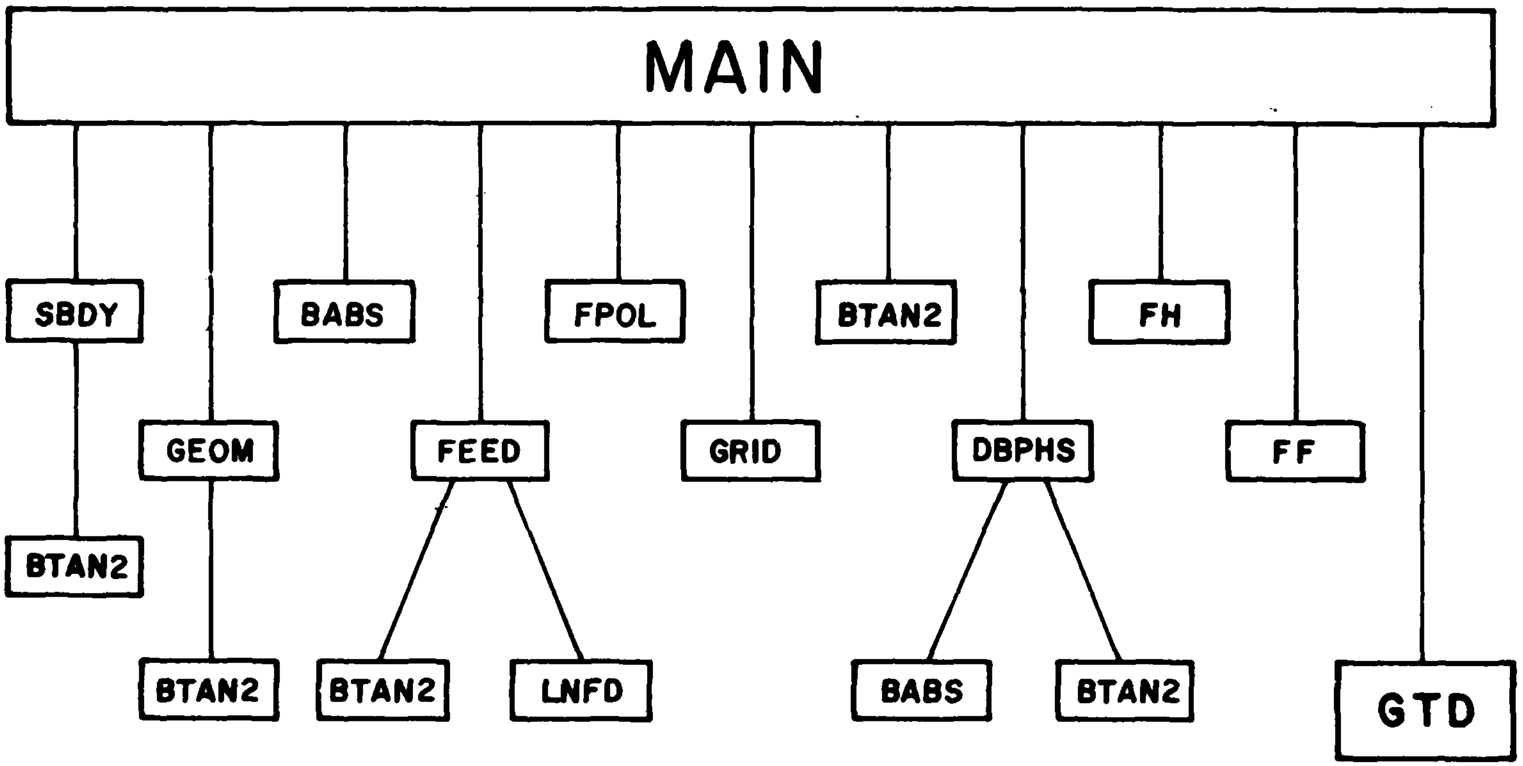
## III. CODE DESCRIPTION

This computer code calculates both far field and near field patterns of reflector antennas with general rim shapes and arbitrary feed patterns. It uses a combination of Aperture Integration (AI) and the Geometrical Theory of Diffraction (GTD) techniques.

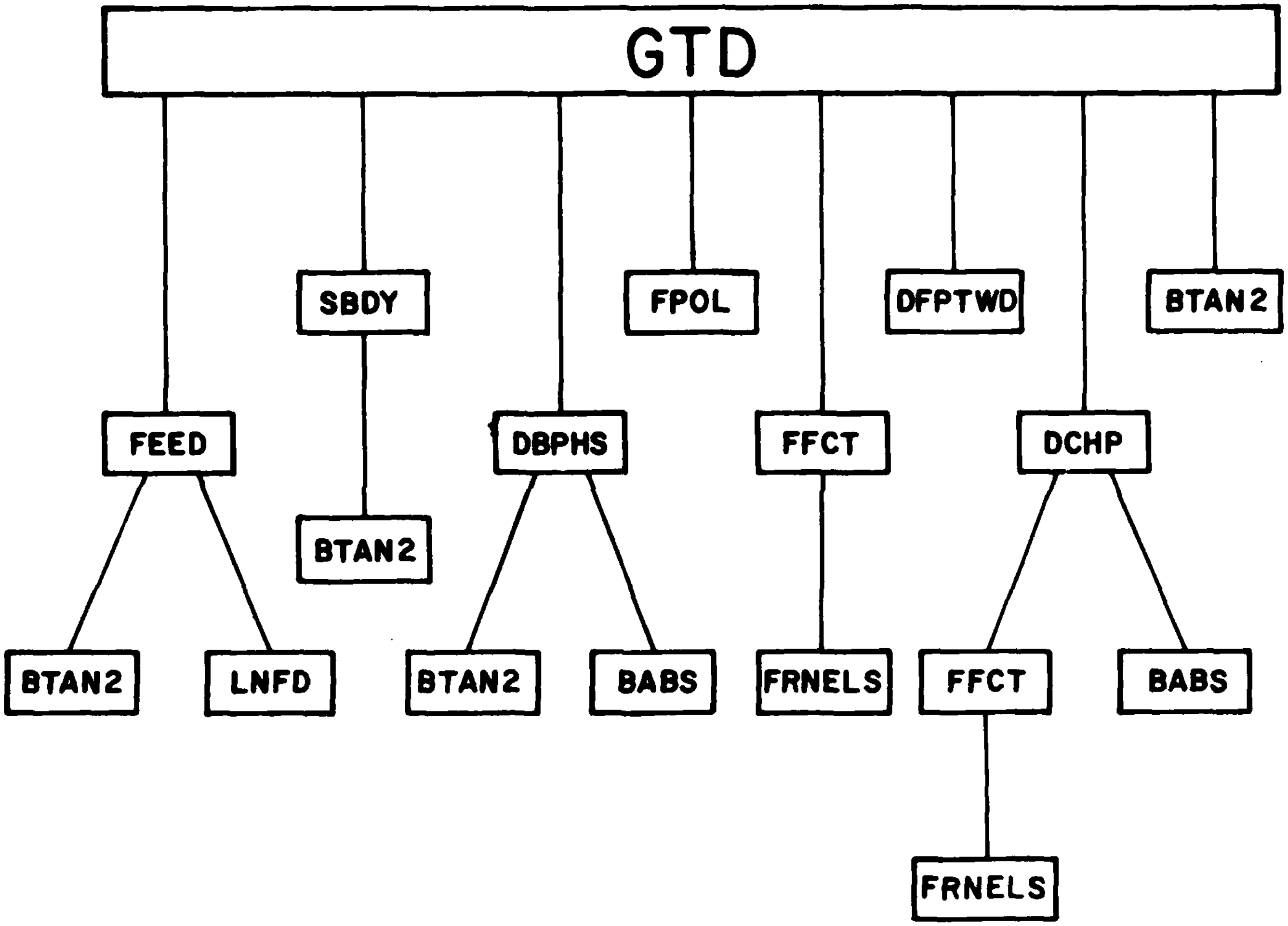
This code is divided into two parts. The first part consists of various command words which read all the input data. The details of this command word system is explained in the User's Manual[1] and thus is not repeated here.

The rest of this code belongs to the content of the XQ command which performs the unit conversion of input data and all the computations to get the far field or near field results. Various subroutines are called during the execution of this program and are described in Part B of this chapter. In the main program, some of the sections which need more detailed explanation are separated as subsections which are actually expansions of their corresponding blocks in the flow diagram of the main program.

The linkage of the subroutines to the main program is shown in the following flow charts. All GTD calculations are controlled by the subroutine GTD. The linkage of the subroutines to the subroutine GTD is shown in the second flow chart that follows:



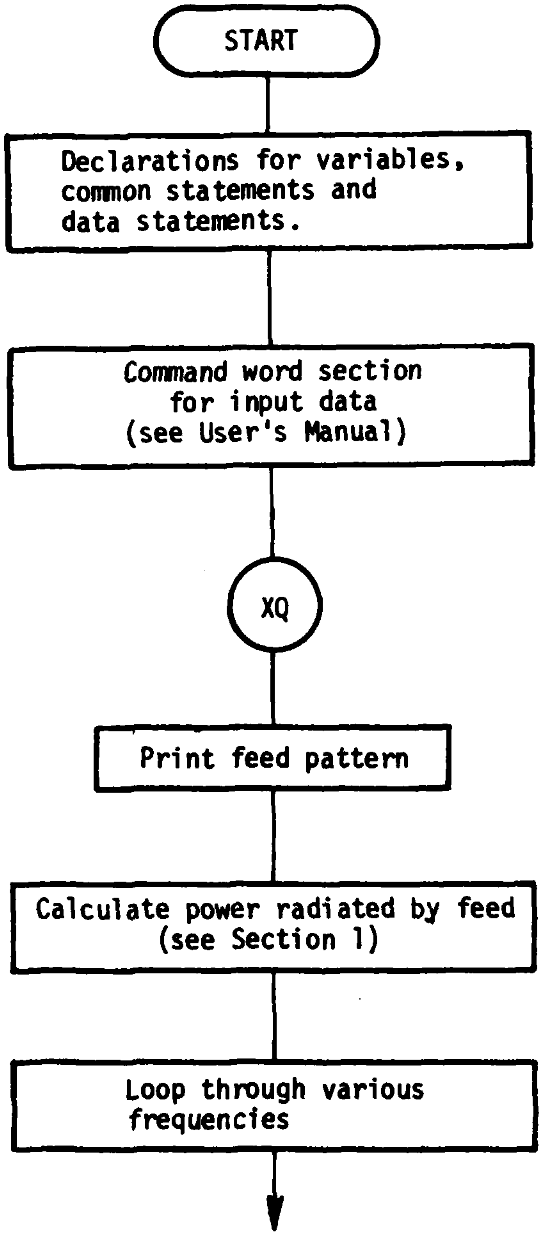
Subroutine Linkage Chart I

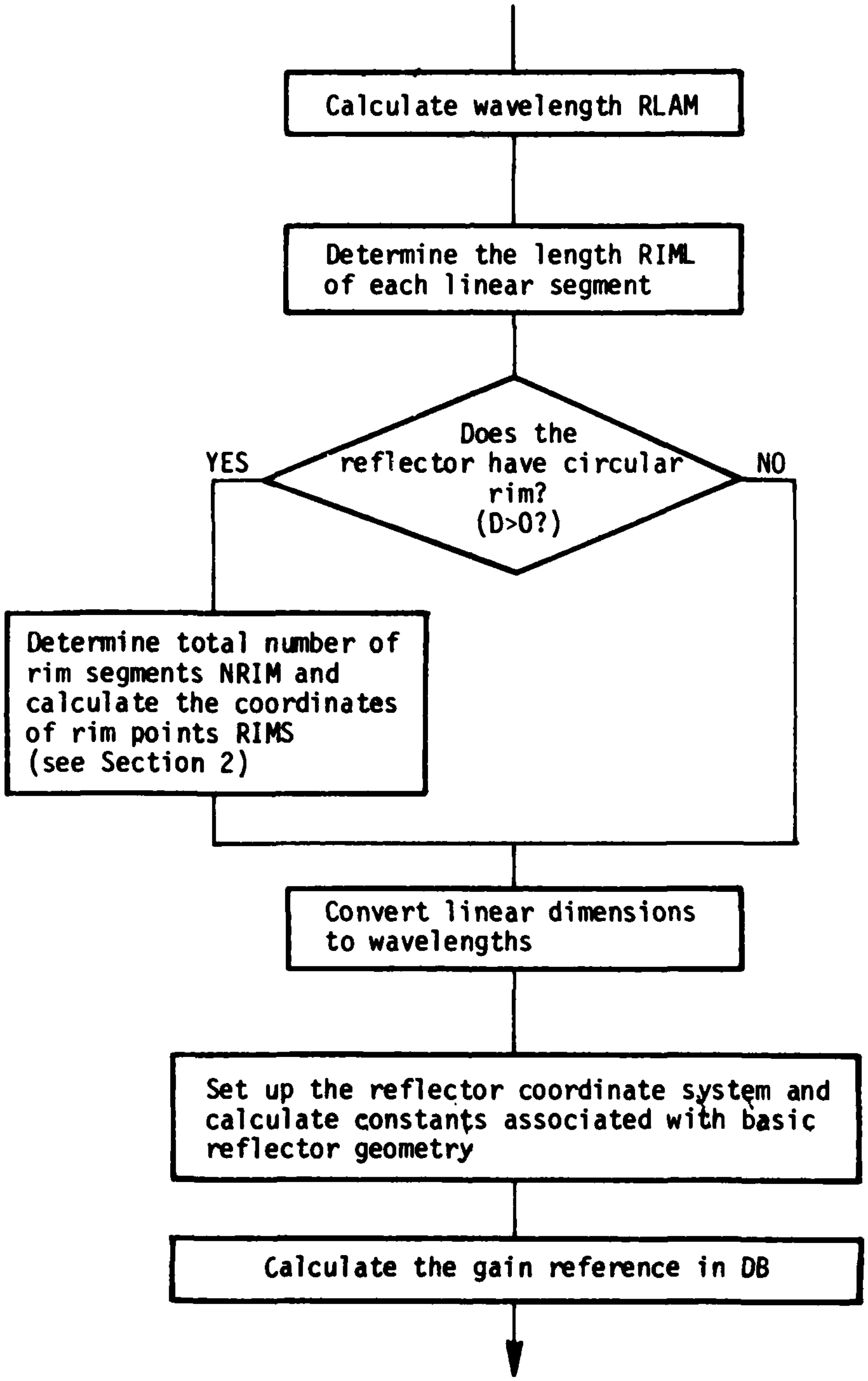


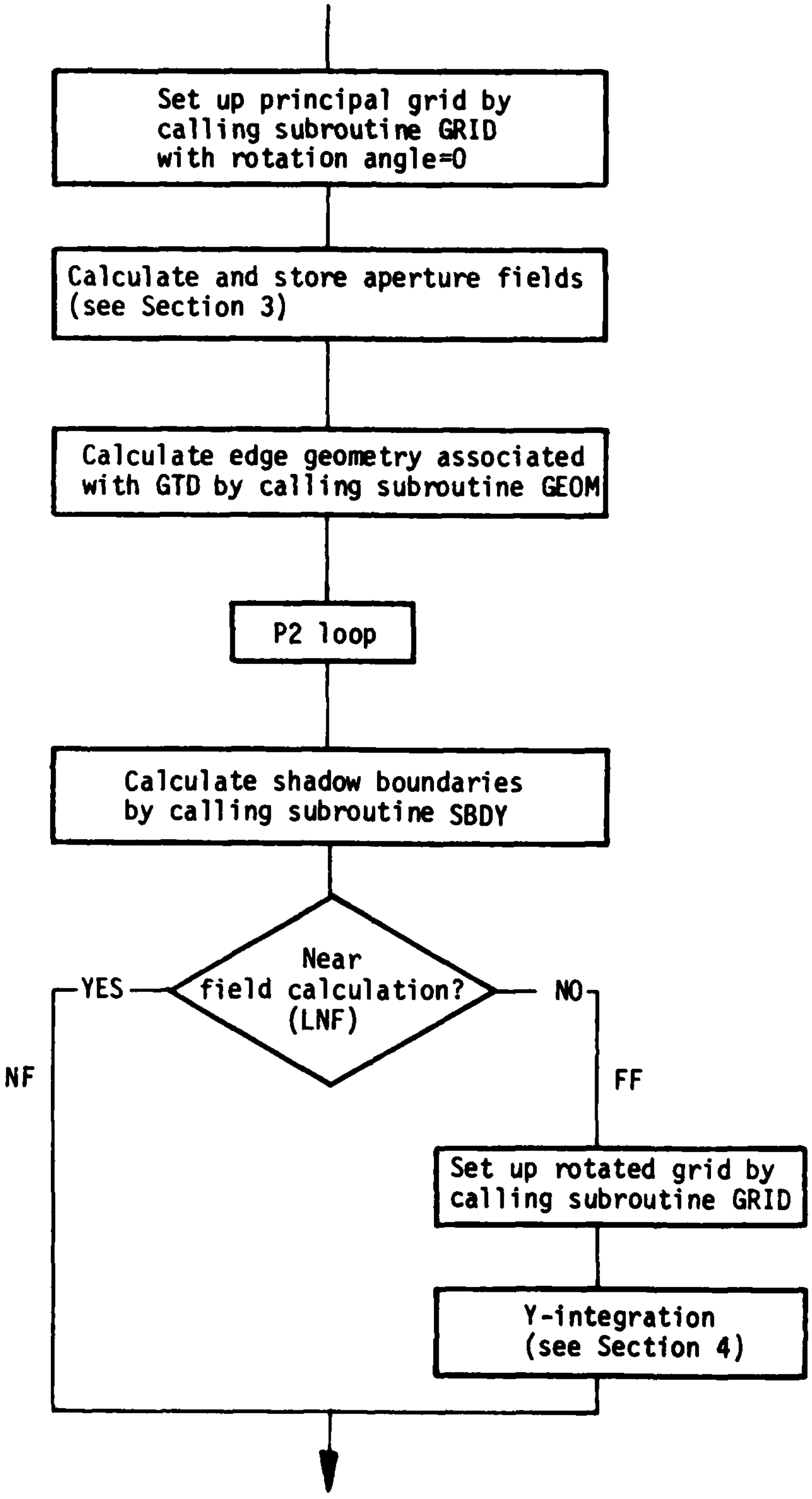
Subroutine Linkage Chart II

## A. MAIN PROGRAM

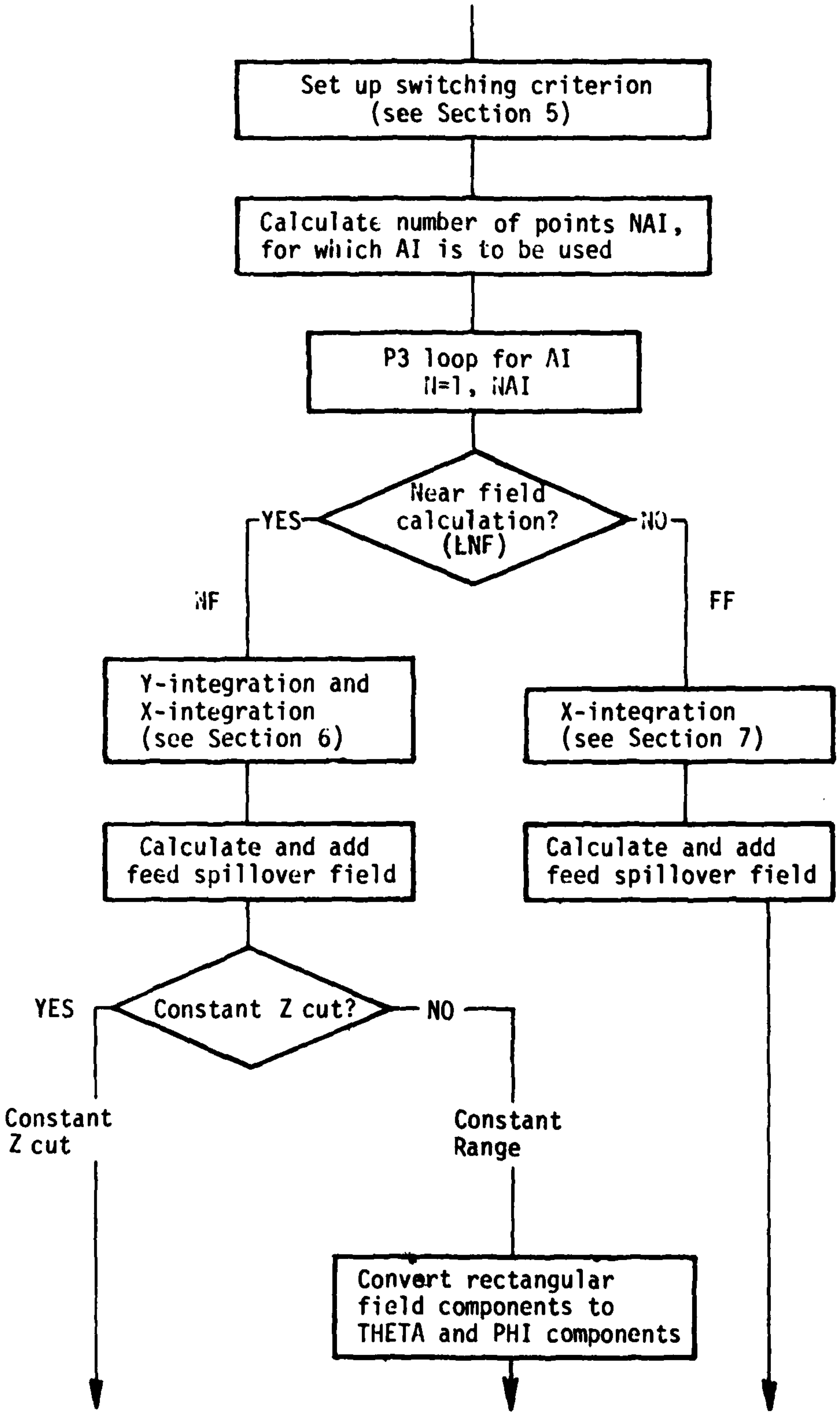
### FLOW DIAGRAM

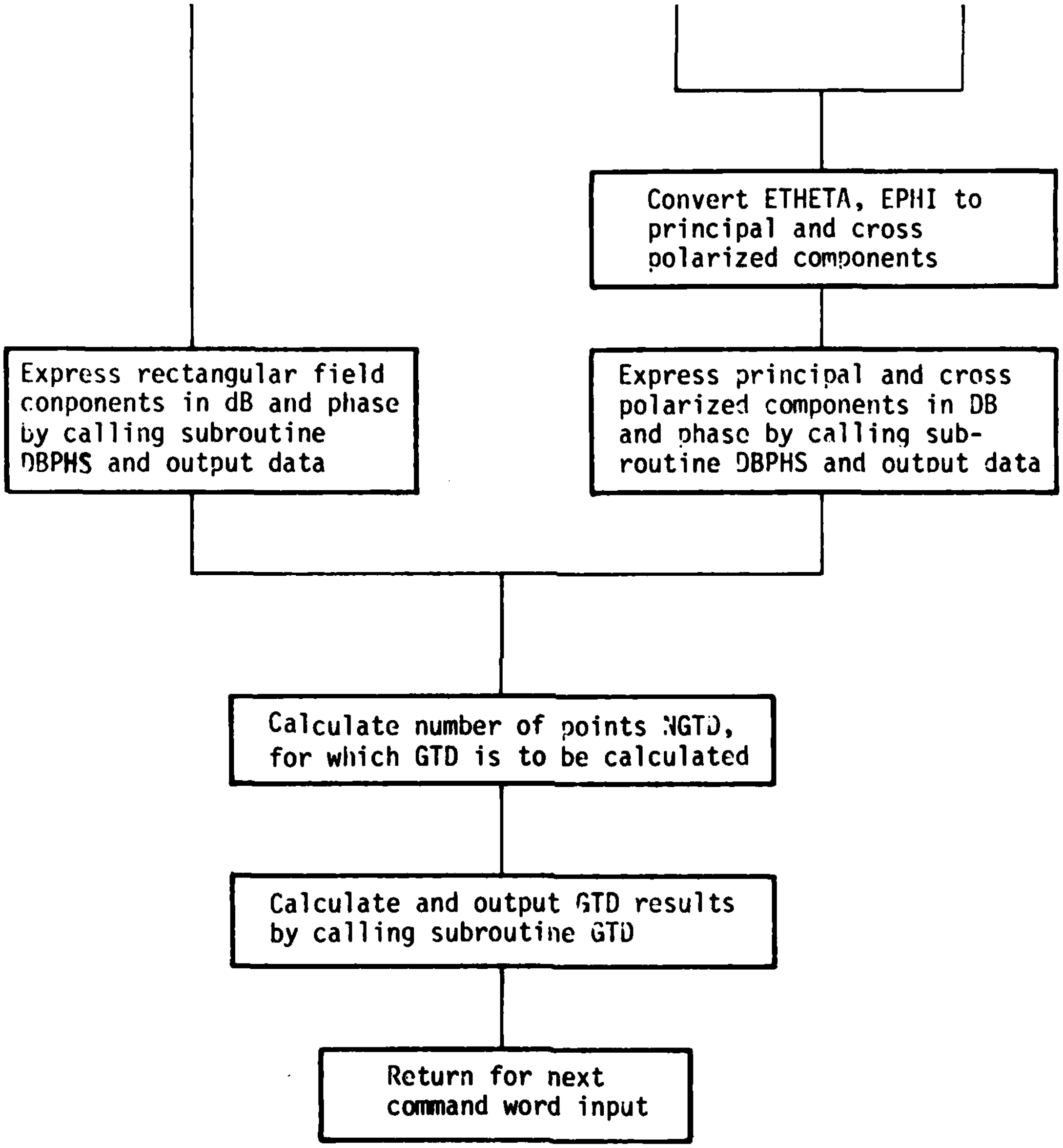



Calculate wavelength RLAM
Determine the length RIML of each linear segment
Does the reflector have circular rim? (D>0?)
YES
NO
Determine total number of rim segments NRIM and calculate the coordinates of rim points RIMS (see Section 2)
Convert linear dimensions to wavelengths
Set up the reflector coordinate system and calculate constants associated with basic reflector geometry
Calculate the gain reference in DB



Set up principal grid by calling subroutine GRID with rotation angle=0
Calculate and store aperture fields (see Section 3)
Calculate edge geometry associated with GTD by calling subroutine GEOM
P2 loop
Calculate shadow boundaries by calling subroutine SBDY
Near field calculation? (LNF)
YES
NO
NF
FF
Set up rotated grid by calling subroutine GRID
Y-integration (see Section 4)


Set up switching criterion (see Section 5)
Calculate number of points NAI, for which AI is to be used
P3 loop for AI N=1, NAI
Near field calculation? (LNF)
YES
NO
NF
FF
Y-integration and X-integration (see Section 6)
X-integration (see Section 7)
Calculate and add feed spillover field
Calculate and add feed spillover field
Constant Z cut?
YES
NO
Constant Z cut
Constant Range
Convert rectangular field components to THETA and PHI components


Convert ETHETA, EPHI to principal and cross polarized components
Express rectangular field conponents in dB and phase by calling subroutine DBPHS and output data
Express principal and cross polarized components in DB and phase by calling subroutine DBPHS and output data
Calculate number of points NGTD, for which GTD is to be calculated
Calculate and output GTD results by calling subroutine GTD
Return for next command word input

CODE LISTING

```
C
C  * FAR AND NEAR FIELD PATTERN FOR PARABOLIC REFLECTOR ANTENNA *
C     * WITH GENERAL RIM SHAPE AND ARBITRARY FEED PATTERN *
C
      DIMENSION JL(50),JU(50),YLR(50),YUR(50),QYL(50),QYU(50),XN(3)
      DIMENSION RHOS(2),CLRIM(67,2),CURIM(67,2),RIMS(67,2)
      DIMENSION RIM(67,2),VI(3),VIM(67,3),RMM(67),DELX(2)
      DIMENSION ANPF(10),AP2(10),FREQ(10),PHIN(15),PSIO(15),
     1FP(15,15),PX(15,15),AEX(15),CAN(15)
      DIMENSION NSNS(10)
      COMPLEX EDX,EDY,EDZ,EDT,EDP,FIP,EIT,EIX,EIY,EIZ,PHEI,YEXP
      COMPLEX YSUM(3,50),EA(2,50,50),E(2,50),CJ,FH,FHXP,FHXM,
     2FXP,FXM,YMR,YML,ERX,ERY,XEXP,EXPL,FXPI,EXPR,EXPM,EA1,EA2,
     3YSLX,YSLY,YSLZ,YSMX,YSMY,YSMZ,YSUX,YSUY,YSUZ,PHSEA,EAL,EAU,
     4SUMLX,SUMLY,SUMMX,SUMMY,SUMRX,SUMRY,SUMX,SUMY,SUMZ,TMX,TMT
      COMPLEX CX,CY,FFXN,FHYM,FHYP,FYM,FYP,RFCT
      LOGICAL LLFD,LAI,LFEED,LGTD
      LOGICAL LSLOPE,LCORNR,LOUT,LWFD,LRESET,LWRITE,LPLT
      LOGICAL LDEBUG,LTEST,LWYSUM,LDEAS,LDB,LCP,LNF,LRANG
      COMMON/LOGDIF/LSLOPE,LCORNR,LNF,LRANG
      COMMON /GRID1/GRIDX,GRIDY,EA
      COMMON /GRID2/CJ,CLRIM,CURIM,RIM,PG,XMIN,XMAX,YMIN,YMAX,
     2NLRIM,NURIM,GRDX,GRDY,ACOSP,TANP,PCHG,MAXO,NRIM
      COMMON /GEOM1/X(67,3),V(67,3),MRIM
      COMMON /GEOM2/VP(67,3),VN(67,3),BD(67,2),VMAG(67),RMC(67),
     2VIC(67,3),XM(67,3)
      COMMON /DIM/MDRIM
      COMMON /SORINF/XS(3)
      COMMON /DIR/DX(3),EIX,EIY,EIZ
      COMMON /NF/RFCT,XOO(3),PHIE,P2,RR
      COMMON /GTDD/LFEED,LOUT,LCP,LWRITE,COSPT,SINPT,REF,TEM2
      COMMON /FED/N2,PHIN,PX,FP,LDB,NCK,NPHI,NPW,AEX,CAN,PSIO,PSIT
      COMMON /COMP/CX,CY,GF,PHP,PHO,KX,KY,ISYM,SINTL,COSTL
      COMMON /PIS/PI,TPI,DPR
      COMMON /PREV/IPR,PREP,PREX,PRES
      COMMON /TEST/LDEBUG,LTEST,NTEST
      COMMON /FOCAL/F,ZOP
      COMMON /RFBDY/RHOS
      COMMON /BDY2/TH1,TH2,THEB
      COMMON /REFL/D,RO,ICQ,JCO
      COMMON /OUT/NW
      DIMENSION IR(24),IT(14)
      DIMENSION LABEL(2,3),UNIT(3)
      DATA UNIT/1.,.3048,0.0254/
      DATA LABEL/18HMETERSFEET  INCHES/
      DATA IT/42HDC:TO:FD:FQ:NF:NX:LP:PP:CM:CE:TL:PZ:XO:EN:/
      DATA DEL/0.01/
      DATA C/3.D8/
      NTEST=0
      TEM2=0.70710678
```

```
      CJ=(0.,1.)
      MDPAT=361
      MDEA=50
      MDFP=15
      MDRIM=64
C!!!
C!!!  DEFAULT DATA    !!!
C!!!
C------
      WRITE(6,3002)
      WRITE(6,3006)
      WRITE(6,3610)
 3010 FORMAT(2H *,T20,'DEFAULT DATA',T79,1H*)
      WRITE(6,3006)
      WRITE(6,3006)
 3000 CONTINUE
C------     NX:    COMMAND    -------------
      LRESET=.TRUE.
      LDEBUG=.FALSE.
      LTEST=.FALSE.
      LWYSUM=.FALSE.
      LOUT=.FALSE.
      LWFD=.FALSE.
      LSLOPE=.TRUE.
      LCORNR=.TRUE.
      LAI=.TRUE.
      LFEED=.TRUE.
      LGTD=.FALSE.
      ZX=0.
      THETAX=0.
      LLFD=.FALSE.
      LCP=.FALSE.
      LDB=.TRUE.
      ISYM=1
      TAU=90.
      NPHI=2
      PHIN(1)=0.
      PHIN(2)=90.
      NPW=1
      AEX(1)=5.
      AEX(2)=6.
      CAN(1)=0.09
      CAN(2)=0.1
      PSIO(1)=120.
      PSIO(2)=140.
      IUNIT=3
      F=8.
      GRIDX=0.6
      GRIDY=0.6
      D=24.
```

```
      PSIT=0.
      YC=0.
      NFRQ=1
      FREQ(1)=11.
      IP2=1
      AP2(1)=0.
      AP3I=0.
      AP3F=90.
      ADP3=5.
      LWRITE=.TRUE.
      LPLT=.TRUE.
      INPF=0
      XOO(1)=0.
      XOO(2)=0.
      XOO(3)=0.
      PHIE=0.
      RANG=10000.
      LRANG=.FALSE.
      LNF=.FALSE.
      GO TO 3100
 3000 CONTINUE
      LRESET=.FALSE.
      WRITE(6,3006)
 3006 FORMAT(1X,1H*,76X,1H*)
      WRITE(6,3006)
      WRITE(6,3005)
 3005 FORMAT(1X,26(3H***))
C!!!  READ IN VARIOUS COMMAND OPTIONS.
 2999 READ(5,3001,END=3004)(IR(I),I=1,24)
 3001 FORMAT(24A3)
 3059 WRITE(6,3002)
 3002 FORMAT(///1X,26(3H***))
      WRITE(6,3006)
      WRITE(6,3003)(IR(I),I=1,24)
 3003 FORMAT(1X,1H*,2X,24A3,2X,1H*)
      IF(IR(1).EQ.IT(9).OR.IR(1).EQ.IT(10))GO TO 3900
      WRITE(6,3006)
      WRITE(6,3006)
C!!!
C!!!  CHECK AGAINST STORED OPTIONS
C!!!
C!!!  DG (IT(1)) : DISH GEOMETRY INPUT
C!!!  TO (IT(2)) : TEST DATA GENERATION OPTION.
C!!!  FD (IT(3)) : FEED PATTERN DEFINED
C!!!  FQ (IT(4)) : FREQUENCY RANGE DEFINED
C!!!  NF (IT(5)) : NEAR FIELD
C!!!  NX (IT(6)) : RESET DEFAULT DATA
C!!!  LP (IT(7)) : LINE PRINTER LISTING OF RESULTS
C!!!  PP (IT(8)) : PEN PLOT OF RESULTS
C!!!  CM (IT(9)) : COMMENT CARD
```

```
C!!!   CE (IT(10)) : END OF COMMENT INFORMATION
C!!!   TL (IT(11)) : FEED TILT ANGLE AND APERTURE CENTER
C!!!   PZ (IT(12)) : PHI PATTERN CUTS DEFINED
C!!!   XQ (IT(13)) : EXECUTE PROGRAM
C!!!   EN (IT(14)) : END PROGRAM
C!!!
       IF (IR(1).EQ.IT(1)) GO TO 3100
       IF (IR(1).EQ.IT(2)) GO TO 3200
       IF (IR(1).EQ.IT(3)) GO TO 3300
       IF (IR(1).EQ.IT(4)) GO TO 3400
       IF (IR(1).EQ.IT(5)) GO TO 3500
       IF (IR(1).EQ.IT(6)) GO TO 3600
       IF (IR(1).EQ.IT(7)) GO TO 3700
       IF (IR(1).EQ.IT(8)) GO TO 3800
       IF (IR(1).EQ.IT(11)) GO TO 4000
       IF (IR(1).EQ.IT(12)) GO TO 4100
       IF (IR(1).EQ.IT(13)) GO TO 4300
       IF (IR(1).EQ.IT(14)) GO TO 3004
       WRITE(6,3021)
 3021  FORMAT(' ***   PROGRAM ABORTS!!! COMMAND INPUT IS NOT PART OF
 ',
      1' STORED COMMAND LIST   ***')
 3004  CALL EXIT
C------
 3100  CONTINUE
C------   OG:   COMMAND   --------------
C$$$
C$$$   IUNIT=UNITS USED TO INPUT THE FOLLOWING LINEAR DIMENSIONS
C$$$         1=DIMENSIONS INPUT IN METERS
C$$$         2=DIMENSIONS INPUT IN FEET
C$$$         3=DIMENSIONS INPUT IN INCHES
C$$$
C$$$
C$$$   F=FOCAL DISTANCE OF THE PARABOLA
C$$$
C$$$   GRIDX=GRID SIZE IN X-DIRECTION USED IN APERTURE INTEGRATION
C$$$
C$$$   GRIDY=GRID SIZE IN Y-DIRECTION USED IN APERTURE INTEGRATION
C$$$
C$$$   D=DIAMETER OF REFLECTOR. IF INPUT GREATER THAN ZERO ASSUMED
C$$$     CIRCULAR AND CODE GENERATES THE RIM POINTS. IF LESS THAN ZE
 RO
C$$$     RIM DATA INPUT WITH FOLLOWING READ STATEMENT
C$$$
C$$$   NOTE: ALL ABOVE DATA INPUT IN UNITS SPECIFIED BY IUNIT
C$$$
       IF(.NOT.LRESET)READ (5,-) IUNIT,F,GRIDX,GRIDY,D
       WRITE (6,3101) (LABEL(N,IUNIT),N=1,2)
 3101  FORMAT (2H *,'  LINEAR DIMENSION INPUTS ARE IN ',2A3,T79,1H*)
       WRITE (6,3006)
       UNITC=UNIT(IUNIT)
       IF (D.LE.0.) GO TO 3104
```

```
      WRITE (6,3102) D
3102  FORMAT(2H *,T8,'CIRCULAR REFLECTOR WITH APERTURE DIAMETER =',
     2F9.2,T79,1H*)
      WRITE(6,3006)
      GO TO 3112
C$$$
C$$$  IF DIAMETER OF DISH IS DEFINED NEGATIVE ABOVE,THEN INPUT RIM
C$$$  POINTS DIRECTLY
C$$$
C$$$  NRIM=NUMBER OF RIM POINTS INPUT
C$$$
C$$$  RIM(NE,1)=X-POSITION OF THE NE-TH RIM POINT
C$$$  RIM(NE,2)=Y-POSITION OF THE NE-TH RIM POINT
C$$$
3104  IF (D.LE.0.AND.(.NOT.LRESET)) READ (5,-) NRIM,((RIM(NE,N),N=1
 ,2),
     1NE=1,NRIM)
      WRITE (6,3106)
3106  FORMAT(2H *,T10,'COORDINATES OF RIM POINTS IN METERS',T79,1H*
 ,
     2/2H *,T20,'RIM POINT',9X,'X',14X,'Y',T79,1H*)
      WRITE(6,3006)
      DO 3110 NE=1,NRIM
      WRITE (6,3108) NE,(RIM(NE,N),N=1,2)
      RIMS(NE,1)=RIM(NE,1)*UNITO
      RIMS(NE,2)=RIM(NE,2)*UNITO
3108  FORMAT(2H *,T20,I5,2F15.2,T79,1H*)
3110  CONTINUE
3112  WRITE (6,3006)
      WRITE (6,3115) F,GRIDX,GRIDY
 3115 FORMAT(2H *,T10,'FOCAL DISTANCE=',F9.2,T35,'GRIDX=',F7.3,5X,
     2'GRIDY =',F7.3,T79,1H*)
      FOCUS=F*UNITO
      GRX=GRIDX*UNITO
      GRY=GRIDY*UNITO
      A=0.5*D*UNITO
      IF(LRESET)GO TO 3300
      GO TO 3600
C------
3200  CONTINUE
C-----   IO:   COMMAND   ------------
C$$$
C$$$  LDEBUG=DEBUG DATA OUTPUT ON LINE PRINTER(TRUE OR FALSE)
C$$$
C$$$  LTEST=TEST DATA TO INSURE PROGRAM OPERATION(TRUE OR FALSE)
C$$$
C$$$  LWYSUM=WRITE  YSUM DATA ON LINE PRINTER(TRUE OR FALSE)
C$$$
C$$$  LOUT=OUTPUT MAIN PROGRAM DATA ON LINE PRINTER(TRUE OR FALSE)
C$$$
C$$$  LWFD=OUTPUT FEED PATTERN DATA ON LINE PRINTER(TRUE OR FALSE)
C$$$
```

```
      READ (5,-) LDEBUG,LTEST,LWYSUM,LOUT,LWFD
      WRITE(6,3201)LDEBUG,LTEST,LWYSUM,LOUT,LWFD
 3201 FORMAT(2H *,5X,'LDEBUG= ',L2,5X,'LTEST= ',L2,5X,'LWYSUM=',L2,
     15X,'LOUT =',L2,5X,'LWFD =',L2,T79,1H*)
      WRITE(6,3006)
C$$$
C$$$  LSLOPE=SLOPE DIFFRACTED FIELD DESIRED (T OR F)
C$$$
C$$$  LCORNR=CORNER DIFFRACTED FIELD DESIRED (T OR F)
C$$$
      READ(5,-)LSLOPE,LCORNR
      WRITE(6,3202)LSLOPE,LCORNR
 3202 FORMAT(2H *,5X,'LSLOPE= ',L2,5X,'LCORNR= ',L2,5X,
     1T79,1H*)
C$$$
C$$$  LAI=APERTURE INTEGRATION SOLUTION INCLUDED (TRUE OR FALSE)
C$$$
C$$$  LFEED=FEED SPILLOVER INCLUDED IN SOLUTION (TRUE OR FALSE)
C$$$
C$$$  LGTD=GTD INCLUDED IN SOLUTION (TRUE OR FALSE)
C$$$
C$$$  THETAX=PATTERN SWITCHING ANGLE FROM AI TO GTD
C$$$
C$$$  ZX=STARTING CRITERION FOR USING AI IN NEAR FIELD CALCULATION
C$$$
      READ(5,-)LAI,LFEED,LGTD,THETAX,ZXP
      WRITE (6,3006)
      WRITE (6,3204) LAI,LFEED,LGTD
 3204 FORMAT(2H *,5X,'LAI =',L2,8X,'LFEED =',L2,6X,'LGTD =',
     2L2,T79,1H*)
      WRITE (6,3006)
      WRITE (6,3206) THETAX,ZXP
 3206 FORMAT (2H *,5X,'THETAX =',F5.2,5X,'ZX =',F10.3,T79,1H*)
      ZXP2=ZXP*UNITO
      GO TO 3000
C------
 3300 CONTINUE
C------   FD:   COMMAND   --------------
      KX=0
      KY=0
      CX=TEM2+CJ*0.
      CY=CJ*TEM2
C$$$
C$$$  LLFD=INPUT FEED PATTERN IN TERMS OF LINEAR DATA POINTS
C$$$       IF .TRUE. OR ANALYTIC FUNCTION IF .FALSE.
C$$$
C$$$  LCP=FEED IS CIRCULARLY POLARIZED (TRUE OR FALSE)
C$$$
C$$$  LDB=FEED DATA INPUT IN DB.,IF LDB=.TRUE.
C$$$      LINEAR FEED DATA INPUT, IF LDB=.FALSE.
```

```
C$$$
C$$$   COEFFICIENTS OF THE FEED PATTERN
C$$$
C$$$      ISYM=0     NO SYMMETRY
C$$$      ISYM=1     EVEN SYMMETRY W.R.T. X AND Y AXIS
C$$$      ISYM=-1    ODD SYMMETRY W.R.T. X AND Y AXIS
C$$$      ISYM=2     EVEN SYMMETRY W.P.T. X AXIS
C$$$      ISYM=-2    ODD SYMMETRY W.R.T. X AXIS
C$$$      ISYM=3     EVEN SYMMETRY W.R.T. Y AXIS
C$$$      ISYM=-3    ODD SYMMETRY W.R.T. Y AXIS
C$$$
C$$$
C$$$  PSIT=TILT ANGLE OF FEED RELATIVE TO -Z AXIS IN THE YZ PLANE.
C$$$       NORMALLY ZERO ;HOWEVER USEFUL FOR OFFSET REFLECTOR
C$$$
C$$$  TAU=LINEAR POLARIZATION ANGLE RELATIVE TO X-AXIS OF FEED
C$$$
      IF(.NOT.LRESET)READ(5,-)LLFD,LCP,LDB,ISYM,TAU
      NCK=2
      IF(LLFD)NCK=0
      IF (LCP) WRITE (6,3301)
 3301 FORMAT(2H *,T8,'CIRCULARLY POLARIZED FEED',T79,1H*)
      WRITE(6,3006)
      WRITE(6,3302)ISYM
3302  FORMAT(2H *,T8,'FEED PATTERN SYMMETRY GIVEN BY;ISYM=',I2,
     1T79,1H*)
      WRITE(6,3006)
C$$$
C$$$  NPHI=NUMBER OF INPUT FEED PATTERN CUTS
C$$$
C$$$  PHIN(N)=PHI ANGLE OF N-TH INPUT PATTERN CUT
C$$$
      IF(.NOT.LRESET)READ (5,-) NPHI,(PHIN(N),N=1,NPHI)
      IF (LCP) GO TO 3305
      WRITE (6,3303) TAU
 3303 FORMAT(2H *,T8,'LINEARLY POLARIZED FEED',T79,1H*,/2H *,T79,
     21H*,/2H *,T10,'POLARIZED ANGLE =',F7.2,T79,1H*)
      WRITE(6,3006)
      TAUR=TAU/DPR
      SINTU=SIN(TAUR)
      COSTU=COS(TAUR)
      CX=COSTU+CJ*0.
      CY=SINTU+CJ*0.
3305  CONTINUE
      IF(LDB)WRITE(6,4002)
4002  FORMAT(2H *,T10,'FEED DATA INPUT IN DB.',T79,1H*)
      IF(.NOT.LDB)WRITE(6,4003)
4003  FORMAT(2H *,T10,'LINEAR FEED DATA INPUT',T79,1H*)
      WRITE(6,3006)
      IF (DABS(CX).GT.1.D-5) KX=1
```

```
      IF (BABS(CY).GT.1.D-5) KY=1
      WRITE(6,3006)
      PN1=PHIN(1)
      PNN=PHIN(NPHI)
      IB=IABS(ISYM)
      PHQ=90.
      PHP=0.
C!!!  CHECK INITIAL AND FINAL INPUT PHIN
      IF (IB.EQ.0.AND.PN1.NE.-180.) GO TO 285
      IF (IB.EQ.1.AND.(PN1.NE.0..OR.PNN.NE.90.)) GO TO 285
      IF (IB.EQ.2.AND.(PN1.NE.0..OR.PNN.NE.180.)) GO TO 285
      IF (IB.EQ.3.AND.(PN1.NE.-90..OR.PNN.NE.90.)) GO TO 285
      IF (LLFD) GO TO 3315
C$$$
C$$$  ANALYTIC FEED PATTERN INPUT(LLDF=.FALSE.)
C$$$
C$$$  NPW=COSINE RAISED TO THIS POWER
C$$$
C$$$  AEX=EXPONENTIAL FACTOR TO CONTROL SIDE LOBE LEVEL
C$$$
C$$$  CAN=CONSTANT TERM TO APPROXIMATE FAR OUT SECTION OF FEED
C$$$  PATTERN
C$$$
C$$$  PSIO(N)=ANGLE TO CONTROL THE ZERO ASSOCIATED WITH COSINE
C$$$          FOR THE N-TH PHI INPUT FEED PATTERN CUT
C$$$
C$$$  NOTE: FEED=CEXP(-AEX*(PSI/PSIO)**2)*COS(.5*PI(PSI/PSIO))**NPW
 +CAN
C$$$
      IF(.NOT.LRESET)READ (5,-)NPW,(AEX(N),CAN(N),PSIO(N),N=1,NPHI)
      WRITE (6,3308) NPW
 3308 FORMAT (2H *,T12,5HNPW =,I2,T79,1H*,/2H *,T16,'N',T26,
     1'PHIN(N)',6X,'PSIO(N)',9X,'AEX(N)',7X,'CAN(N)',T79,1H*)
      DO 3312 N=1,NPHI
      WRITE (6,3310) N,PHIN(N),PSIO(N),AEX(N),CAN(N)
 3310 FORMAT(2H *,T15,I2,3F14.1,F13.2,T79,1H*)
 3312 CONTINUE
      GO TO 3000
 3315 N1=0
C$$$
C$$$  LINEAR FEED PATTERN INPUT(LLFD=.TRUE.)
C$$$
C$$$  N2=MAXIMUM NUMBER OF FEED PATTERN POINTS TO BE READ FOR
C$$$     ALL INPUT PHI ANGLES
C$$$
      IF(.NOT.LRESET)READ (5,-) N2
      WRITE (6,3318) N2
 3318 FORMAT(2H *,T10,'MAXIMUM NUMBER OF FEED POINTS=',I2,T79,1H*)
      WRITE(6,3006)
      IF (N2.GT.MDFP) GO TO 272
      NPP=NPHI+1
```

```
      IF (KY.EQ.0) WRITE (6,3320)
      IF (KX.EQ.0) WRITE (6,3322)
 3320 FORMAT(2H *,T8,'X-ORIENTED DIPOLE FEED',T79,1H*)
 3322 FORMAT(2H *,T8,'Y-ORIENTED DIPOLE FEED',T79,1H*)
      WRITE(6,3006)
      DO 3340 NP=1,NPHI
      WRITE (6,3325) NP,PHIN(NP)
3325  FORMAT(2H *,T8,5HPHIN(,I1,3H) =,F6.1,T79,1H*)
      WRITE(6,3006)
      WRITE (6,3326)
3326  FORMAT(2H *,T10,27HPIECEWISE LINEAR FEED INPUT,T79,1H*,/
     22H *,T16,3HPSI,T31,1HF,12X,5HF(DB),T79,1H*)
      WRITE(6,3006)
      DO 3340 K=1,N2
C$$$
C$$$  PSIX=K-TH PSI PATTERN ANGLE OF INPUT FEED POINT
C$$$
C$$$  FN=PATTERN VALUE IN DB.
C$$$
      IF(.NOT.LRESET)READ (5,-) PSIX,FN
      PX(NP,K)=PSIX
      FP(NP,K)=FN
      IF (NP.GT.1) GO TO 3328
      PX(NPP,K)=PSIX
      FP(NPP,K)=FN
3328  IF (LDB) GO TO 3330
      AFN=ABS(FN)
      IF (AFN.LT.1.E-5) FDB=-500.
      IF (AFN.GE.1.E-5) FDB=20.*ALOG10(AFN)
      GO TO 3332
3330  FDB=FN
      FN=10.**(FDB/20.)
3332  WRITE (6,3334) PSIX,FN,FDB
3334  FORMAT(2H *,T10,F10.2,F15.4,F13.2,T79,1H*)
 3340 CONTINUE
      GO TO 3600
C------
3400  CONTINUE
C------    FQ:   COMMAND   -------------
C$$$
C$$$  NFRQ=NUMBER OF FREQUENCIES CONSIDERED IN COMPUTATION
C$$$
C$$$  FREQ(I)=I-TH FREQUENCY IN GIGAHERTZ
C$$$
      READ(5,-)NFRQ,(FREQ(I),I=1,NFRQ)
      WRITE(6,3401)NFRQ,(FREQ(I),I=1,NFRQ)
3401  FORMAT(2H *,' FOR THIS GEOMETRY,THERE WILL BE ',I3,' FREQUENC
  IES'
     1,' CONSIDERED AS FOLLOWS:',T79,1H*,/2H *,10(F6.2,','),
     2T79,1H*)
      GO TO 3600
```

```
C------
3500  CONTINUE
C---    NF:   COMMAND   --------------
      READ (5,-) LNF,LRANG
      WRITE (6,3006)
      IF (LNF) GO TO 3505
      WRITE (6,3501)
3501  FORMAT (2H *,T5,'FAR FIELD PATTERN WILL BE CALCULATED',T79,1H
*)
      IF (.NOT.LRANG) GO TO 3000
      READ (5,-) RANG
      WRITE (6,3502) RANG
3502  FORMAT (2H *,T10,'WITH RANGE =',F10.2,T79,1H*)
      RANG=RANG*UNITO
      GO TO 3000
3505  WRITE (6,3506)
3506  FORMAT (2H *,T5,'NEAR FIELD PATTERN WILL BE CALCULATED',T79,1
H*)
      READ (5,-) PHIE,(XOO(I),I=1,3)
      WRITE (6,3507) PHIE,(XOO(I),I=1,3)
3507  FORMAT (2H *,T10,'IN PHIE =',F7.2,' DEGREE CUT, AND ORIGIN AT
 (',
     23(F6.2,','),' )',T79,1H*)
      X01=XOO(1)*UNITO
      X02=XOO(2)*UNITO
      X03=XOO(3)*UNITO
      IF (LRANG) WRITE (6,3508)
3508  FORMAT (2H *,T10,'WITH CONTANT RANGE',T79,1H*)
      IF (.NOT.LRANG) WRITE (6,3509)
3509  FORMAT (2H *,T10,'WITH CONTANT Z CUT',T79,1H*)
      WRITE (6,3006)
      GO TO 3000
C------
3700  CONTINUE
C---    LP:   COMMAND   --------------
C$$$
C$$$  SET WRITE OUTPUT FLAG SO DATA WRITTEN OUT ON LINE PRINTER.
C$$$
      WRITE(6,3701)
3701  FORMAT(2H *,5X,' DATA WILL BE OUTPUT ON LINE PRINTER !!!',
     1T79,1H*)
      LWRITE=.TRUE.
      GO TO 3000
C------
3800  CONTINUE
C---    PP:   COMMAND   --------------
C$$$
C$$$  SET FLAG SUCH THAT THE DATA WILL BE PEN PLOTTED
C$$$  IN RECTANGULAR FORM
C$$$
      READ (5,-) LPLT,INPF
      IF (LPLT) WRITE(6,3802)
3802  FORMAT(2H *,5X,'DATA WILL BE OUTPUT TO PEN PLOTTER !!!',T79,1
H*)
```

```
      IF (.NOT.LPLT) WRITE (6,3804)
 3804 FORMAT (2H *,5X,'NO PLOT OUTPUT GENERATED',T79,1H*)
      IF (INPF.EQ.0) GO TO 3000
      IF (INPF.GT.0) READ (5,-) (ANPF(L),L=1,INPF)
      IF (INPF.LT.0)READ (5,-) ANPF(1),ANPF(2)
      GO TO 3000
C------
 3900 CONTINUE
C------    CM:    OR    CE:    COMMAND    --------------
C$$$
C$$$  READ "CM:" CARDS AS COMMENTS UNTIL A "CE:" CARD IS INPUT WHIC
H
C$$$  ENDS COMMENTS
C$$$
 3999 IF(IR(1).EQ.IT(10))GO TO 3000
      READ(5,3001,END=3004)(IR(I),I=1,24)
      WRITE(6,3003)(IR(I),I=1,24)
      IF(IR(1).EQ.IT(9).OR.IR(1).EQ.IT(10))GO TO 3999
      GO TO 3059
C------
 4000 CONTINUE
C------    TL:    COMMAND    --------------
C$$$
C$$$  PSIT=TILT ANGLE OF FEED RELATIVE TO -Z AXIS IN THE YZ PLANE.
C$$$        NORMALLY ZERO ;HOWEVER USEFUL FOR OFFSET REFLECTOR
C$$$
C$$$  YC=Y-COORDINATE OF APERTURE CENTER
C$$$
      READ (5,-) PSIT,YC
      WRITE (6,4001) PSIT,YC
 4001 FORMAT (2H *,T10,'FEED AXIS TILT ANGLE =',F8.2,T79,1H*,/2H *,
     2T79,1H*,/2H *,T10,'APERTURE CENTER AT (0.,',F8.3,')',T79,1H*
)
      YCM=YC*UNITO
      WRITE (6,3006)
      GO TO 3000
C------
 4100 CONTINUE
C------    PZ:    COMMAND    --------------
C$$$
C$$$  IP2=ABSOLUTE VALUE OF NUMBER OF PATTERN CUTS DESIRED.
C$$$       IF IP2 LESS THAN ZERO ,THEN INPUT ANP SUCH THAT
C$$$       P2=AP2(1)+(N-1)*AP2(2) FOR N=0 TO N=/IP2/-1.
C$$$       IF IP2 GREATER THAN ZERO ,THEN IP2UT THE DESIRED
C$$$       PATTERN CUT VALUES INPUT DIRECTLY
C$$$
      READ(5,-)IP2
      NP2=IABS(IP2)
      WRITE(6,4101)NP2
 4101 FORMAT(2H *,' USING THE PRESENT GEOMETRY,THERE WILL BE ',I3,
     1' PATTERN CUTS COMPUTED',T79,1H*)
      WRITE(6,3006)
      WRITE(6,3006)
```

```
C$$$
C$$$  ANP=INFORMATION ASSOCIATED WITH THE DESIRED CUTS. THE
C$$$  DEFINITION OF THIS ARANGAY IS GIVED IN THE PREVIOUS COMMENTS
C$$$
      IF (IP2.GT.0) READ (5,-) (AP2(L),L=1,NP2)
      IF(IP2.GT.0)WRITE(6,4102)(AP2(L),L=1,NP2)
4102  FORMAT(2H *,' SINCE IP2 IS POSITIVE THE FOLLOWING CUTS WILL'
     1,' BE COMPUTED:',T79,1H*,/2H *,8(F6.1,',','),T79,1H*)
      IF (IP2.LT.0) READ (5,-) (AP2(L),L=1,2)
      IF(IP2.LT.0)WRITE(6,4103)AP2(1),AP2(2)
4103  FORMAT(2H *,' PATTERN CUTS WILL BE COMPUTED STARTING AT P2='
     1,F6.1,' AND INCREMENTED BY',F6.1,T79,1H*)
      WRITE(6,3006)
C$$$
C$$$  ADP3=INCREMENTS IN PATTERN VALUES FOR EACH CUT
C$$$
C$$$  AP3I=INITIAL THETA ANGLE OR RHO FOR EACH COMPUTED PATTERN CUT
C$$$
C$$$  AP3F=FINAL THETA ANGLE OR RHO FOR PATTERN CUT
C$$$
      READ(5,-)AP3I,AP3F,ADP3
      WRITE (6,4104) AP3I,AP3F
4104  FORMAT(2H *,5X,'AP3I =',F7.2,5X,'AP3F =',F7.2,T79,1H*)
      WRITE (6,3006)
      WRITE(6,4105)ADP3
4105  FORMAT(2H *,'  FOR EACH CUT THE PATTERN WILL BE COMPUTED EACH
  '
     1,F6.1,' DEGREES THETA OR',T79,1H*,/2H *,T79,1H*,/2H *,
     2'  INPUT UNIT IN RHO',T79,1H*)
      IF (.NOT.LNF) GO TO 4106
4106  WRITE(6,3006)
      GO TO 3000
C------
4300  CONTINUE
C------   XO:   COMMAND   ------------------------------
      RFCT=(1.,0.)
      SINTL=SIN(PSIT/DPR)
      COSTL=COS(PSIT/DPR)
      IF (.NOT.LWFD) GO TO 70
C
C     *** PRINT FEED PATTERN ***
C
      IF (NCK.EQ.0) DPSI=5.
      IF (NCK.EQ.2) DPSI=PSIO(1)/10.
      PHIP=0.
      DO 60 NP=1,2
      WRITE (6,60) PHIP
 60   FORMAT(/2H F,T5,6HPHIP =,F7.2,1X,20H DEGREE FEED PATTERN,T79,1
  1F
     2,/2H F,11X,3HPSI,T29,2HGX,11X,2HGY,T79,1HF)
      WRITE(6,3006)
```

```
      PSI=0.
      DO 65 I=1,19
      CALL FEED(PSI,PHIP,PSA,PHCAM)
      GX=BABS(GF*CX)
      GY=BABS(GF*CY)
      IF (.NOT.LDB) GO TO 62
      IF (GX.LT.1.E-5) GX=-100.
      IF (GY.LT.1.E-5) GY=-100.
      IF (GX.GE.1.E-5) GX=20.*ALOG10(GX)
      IF (GY.GE.1.E-5) GY=20.*ALOG10(GY)
   62 CONTINUE
      WRITE (NL,64) PSI,GX,GY
   64 FORMAT(2H F,T10,F7.2,5X,2F12.4,T79,1HF)
      PSI=PSI+DPSI
   65 CONTINUE
      PHIP=PHIP+90.
   66 CONTINUE
    70 TEMP=PSIT
      PSIT=0.
C
C     *** PLOT FEED PATTERN ***
C
      IF (INPF.EQ.0) GO TO 80
      NNF=IABS(INPF)
      LAB=0
      DO 76 MF=1,NNF
      IF (INPF.GT.0) PHIP=ANPF(MF)
      IF (INPF.LT.0) PHIP=ANPF(1)+(MF-1)*ANPF(2)
      PHIPF=PHIP
      PSI=0.
      DO 75 I=1,51
      PSIF=ABS(PSI)
      IF (PSI.LT.0.) PHIPF=PHIP-180.
       CALL FEED(PSIF,PHIPF,PSA,PHCAM)
      AGX=BABS(GF*CX)
      AGY=BABS(GF*CY)
      IF (AGX.LT.1.E-5) GXDB=-100.
      IF (AGY.LT.1.E-5) GYDB=-100.
      IF (AGX.GE.1.E-5) GXDB=20.*ALOG10(AGX)
      IF (AGY.GE.1.E-5) GYDB=20.*ALOG10(AGY)
      IF (KX.EQ.1) PLTF=GXDB
      IF (KY.EQ.1) PLTF=GYDB
      PSI=PSI+1.
   75 CONTINUE
   76 CONTINUE
      LAB=0.
      IF ((.NOT.LAI).AND.(.NOT.LCTP)) GO TO 3000
```

**Power Radiated by feed (Section 1)**

```
C
C        *** FREQUENCY LCOP ***
C
         DO 270 IQ=1,NFRQ
         WRITE(6,3006)
         WRITE (6,96) FREQ(IQ)
   96    FORMAT(2H *,T10,'FREQUENCY =',F10.3,' GHZ',T79,1H*)
         WRITE(6,3006)
         RLAM=1.D-9*C/FREQ(IQ)
         WRITE (6,97) RLAM
   97    FORMAT (2H *,T10,'WAVELENGTH =',F12.6,' METERS',T79,1H*,/2H *,
        2T10,'* THE FOLLOWING DIMENSION UNITS ARE IN WAVELENGTHS *',
        3T79,1H*)
         WRITE (6,3006)
         IF (.NOT.LNF) GO TO 98
         X00(1)=X01/RLAM
         X00(2)=X02/RLAM
         X00(3)=X03/RLAM
   98    RR=RANG/RLAM
         IF (LRANG) RFCT=CEXP(-CJ*TPI*RR)/RR
         D=2.*A/RLAM
         YC=YCM/RLAM
         F=FOCUS/RLAM
         ZX=ZXP2/RLAM
         RIML=SQRT(F/2.)
         IF (LNF) RIML=RIML/2.
         IF(D.LE.0.)GO TO 104
```

**Rim point calculation for circular aperture (Section 2)**

```
 104   WRITE(6,3006)
       WRITE (6,105)
  105 FORMAT(2H *,T10,'COORDINATES OF RIM POINTS (WAVELENGTHS)',T7
    9,
     21H*,/2H *,T20,'RIM POINT',9X,'X',14X,'Y',T79,1H*)
       RMAX=0.
       DO 108 NE=1,NRIM
       DO 106 N=1,2
  106  RIM(NE,N)=RIMS(NE,N)/RLAM
       RHOSO=RIM(NE,1)**2+RIM(NE,2)**2
       IF (RHOSO.GT.RMAX) RMAX=RHOSO
  108  WRITE (6,3108) NE,(RIM(NE,N),N=1,2)
       WRITE(6,3006)
       GRIDX=GRX/RLAM
       GRIDY=GRY/RLAM
       WRITE (6,109) F,GRIDX,GRIDY
  109  FORMAT (2H *,T12,'FOCAL DISTANCE =',F9.2,T79,1H*,/2H *,T79,
      21H*,/2H *,T12,'GRIDX =',F7.2,5X,'GRIDY =',F7.2,T79,1H*,)
       WRITE (6,3006)
       ZOP=RMAX/(4.*F)
       ZO=F-ZOP
       RO=SQRT(RMAX+ZO**2)
       XS(1)=0.
       XS(2)=0.
       XS(3)=ZO
       IF (LTEST) WRITE (6,110) ZOP,ZO,RO
 110   FORMAT(2H D,T10,4HZOP=,F9.2,5X,3HZO=,F9.2,5X,3HRO=,F9.3,T79,1H
    D)
       REFDB=10.*ALOG10(2.*TPI/(F*F*PRAD))
       REF=REFDB
       IF (LRANG) REF=0.
       WRITE(6,3006)
       WRITE (6,111) REF
  111  FORMAT(2H *,T12,'REF =',F10.3,T79,1H*)
       WRITE(6,3006)
       PHSEA=CEXP(-CJ*TPI*RO)
       P3I=AP3I
       P3F=AP3F
       DP3=ADP3
       IF (.NOT.LNF.OR.LRANG) GO TO 112
       P3I=AP3I*UNITO/RLAM
       P3F=AP3F*UNITO/RLAM
       DP3=ADP3*UNITO/RLAM
  112  NT=(P3F-P3I)/DP3+1.1
C
C      *** ALL UNITS ARE IN WAVELENGTHS FROM HERE ON ***
C
       IF (LPLT) WRITE (2) LNF,LRANG,LAI,LGTD
       IF (LPLT) WRITE (2) P3I,P3F,DP3,NP2,PHIE,F,D,GRIDX,YC,RR,PSIT
C
C      *** SET UP PRINCIPAL GRID ***
```

```
C
        CALL GRID(0.,ICO,IMO)
        FJC=-YMIN/GRDY+DEL
        JCO=FJC+1
        IF (FJC.LT.-1.) JCO=JCO-1
        FJ=YMAX/GRDY+DEL
        JMO=FJ+JCO
        IF (LTEST) WRITE (6,113) ICO,IMO,JCO,JMO
  113   FORMAT(2H D,T10,4HICO=,I3,5X,4HIMO=,I3,T79,1HD,/2H D,T10,4HJ
 CO=,
       2,I3,5X,5HJMO =,I3,T79,1HT)
        IC=ICO
        JC=JCO
        IMAX=IMO
        MMAX=IMO+2
        NMAX=JMO+2
        MAXO=MMAX
        IF (NMAX.GT.MAXO) MAXO=NMAX
        MNO=(MAXO+1)/2
        AX=XMAX-XMIN
        BY=YMAX-YMIN
        IF (MAXO.LE.MDEA) GO TO 116
        MAX=MAXO
  114   WRITE(6,3006)
        WRITE (6,115) MAX
  115   FORMAT(2H E,T10,'MAX =',I3,T79,1HE)
        WRITE(6,3006)
        GO TO 272
  116   CONTINUE
C
        IF (LTEST.OR.LDEBUG) NTEST=1
        IF (LTEST.OR.LDEBUG) WRITE (6,117)
  117   FORMAT(2H T,T10,'TESTING APERTURE FIELDS',T79,1HT)
```

**Aperture field calculations (Section 3)**

```
C
        CALL GEOM(MRIM,RIML,RIM)
        IF (MRIM .LE.MDRIM) GO TO 123
        WRITE(6,3006)
        WRITE (6,122) MRIM
  122   FORMAT(2H E,T10,'MRIM =',I2,T79,1HE)
        WRITE(6,3006)
        GO TO 272
  123   CONTINUE
```

```
C
      IF ((.NOT.LNF).AND.(LRANG)) WRITE (6,126) RR
  126 FORMAT (/2H *,T10,'** CONSTANT RANGE R =',F10.2,' **',T79,1H*
     ,/)
C
C     *** P2 LOOP ***
C
      NP2=IABS(IP2)
      DO 270 MP=1,NP2
      IF (IP2.GT.0) P2=AP2(MP)
      IF (IP2.LT.0) P2=AP2(1)+(MP-1)*AP2(2)
      WRITE(6,3006)
      IF (LNF) PHI=PHIE
      IF (.NOT.LNF) PHI=P2
      WRITE (6,130) PHI
  130 FORMAT(2H *,T5,5HPHI =,F8.2,T79,1H*)
      IF (PHI.GT.180.) PHI=PHI-360.
      IF (PHI.GT.180..OR.PHI.LT.-180.) WRITE (6,131)
  131 FORMAT (2H E,T10,'***ERROR : INVALID PHI FOR SUBROUTINE SBDY'
     2,T79,1HE)
      IF (.NOT.LNF) GO TO 137
  132 P2=P2*UNITO/RLAM
      WRITE (6,3006)
      IF (LRANG) WRITE (6,135) P2
  135 FORMAT (2H *,T10,'NEAR FIELD WITH CONSTANT RANGE R =',F10.2,
     2T79,1H*)
      IF (.NOT.LRANG) WRITE (6,136) P2
  136 FORMAT (2H *,T10,'NEAR FIELD OBSERVATION PLANE AT Z =',F10.2,
     2T79,1H*)
  137 WRITE(6,3006)
      IF (LPLT) WRITE (2) P2
      PHIR=PHI/DPR
      COSP=COS(PHIR)
      IF (ABS(COSP).LT.1.D-5) COSP=0.
      SINP=SIN(PHIR)
      TH1=180.
      TH2=180.
      THEB=PI/2.
C
C     *** CALCULATE SHADOW BOUNDARIES ***
C
      CALL SBDY(MRIM,X,XS,PHI,TH1,TH2,THEB)
      WRITE (6,138) TH1,TH2
  138 FORMAT (2H F,T10,'TH1 =',F8.2,5X,'TH2 =',F8.2,T79,1HF)
      WRITE(6,3006)
      IF (LNF.AND..NOT.LRANG) WRITE (6,139)
  139 FORMAT (2H W,T31,'EX',27X,'EY',27X,'EZ',//T7,3HRHO,6X
     2,3(5X,3HMAG,7X,2HDB,7X,5HPHASE))
      IF(.NOT.LNF.OR.LRANG)WRITE (6,1391)
 1391 FORMAT(2H W,T27,'PRINCIPAL POL',T55,'CROSS POL',T79
     2,1HW,/2H W,T79,1HW,/2H W,T6,5HTHETA,5X,
```

```
     32(5X,3HMAG,7X,2HDB,7X,5HPHASE),T79,1HW,/2H *,T79,1H*)
      PPT=(PHI-TAU)/DPR
      SINPT=SIN(PPT)
      COSPT=COS(PPT)
      IF(.NOT.LAI)GO TO 180
      IF (LNF) GO TO 148
C
C     *** SET UP ROTATED GRID ***
C
      PHIG=PHI
      CALL GRID(PHIG,IC,IMAX)
      IF (IMAX.GE.3) GO TO 142
      WRITE(6,3006)
      WRITE (6,140)
  140 FORMAT(2H E,T5,28H* ERROR : IMAX LESS THAN 3 *,T79,1HE)
      WRITE(6,3006)
      GO TO 3000
  142 IF (PCHG) 143,144,144
  143 JC=ICO
      JMAX=IMO
      ICOP=JCO
      GO TO 145
  144 JC=JCO
      JMAX=JMO
      ICOP=ICO
  145 IF (LTEST) WRITE (6,146) IC,IMAX,JC,JIMAX
  146 FORMAT(2H D,T5,3HIC=,I3,5X,5HIMAX=,I3,T79,1HD/2H D,T5,3HJC=,I
 3
     2,5X,6HJMAX =,I3,,T79,1HD)
  148 IF (LNF.AND.(MP.GT.1)) GO TO 173
      K=0
      L=0
      MC=IC+1
      MAX=IMAX+2
      MIX=MAX-1
      IF (MAX.GT.MDEA) GO TO 114
```

## Y integration for far field (Section 4)

```
C
  173 DXL=(1-IC)*GRDX-XMIN
      QXL=DXL/GRDX
      DXR=XMAX-(IMAX-IC)*GRDX
      QXR=DXR/GRDX
      IF (LTEST) WRITE (6,175) DXL,DXR
  175 FORMAT(2H D,T5,5HDXL =,F6.2,5X,5HDXR =,F6.2,T79,1HD)
```

## Switching criterion (Section 5)

```
C
C      *** P3 LOOP ***
C
  182 S=RR
      IF (.NOT.LNF) GO TO 185
      SINPE=SIN(PHIE/DPR)
      COSPE=COS(PHIE/DPR)
      IF (.NOT.LRANG) ZE=P2
      IF (LRANG) RE=P2
  185 P3=P31
      IF (LTEST) WRITE (6,186) THETAX,NAI
  186 FORMAT(2H D,T10,'THETAX =' F7.2,5X,'NAI =',I5,T79,1HD)
      DO 250 N=1,NAI
      THER=P3/DPR
      IF (.NOT.LNF) GO TO 196
C
C      *** NEAR FIELD COORDINATE CONVERSION ***
C
      IF (.NOT.LRANG) GO TO 190
      THE=P3/DPR
      SINTE=SIN(THE)
      COSTE=COS(THE)
  188 XN(1)=XCO(1)+RE*SINTE*COSPE
      XN(2)=XCO(2)+RE*SINTE*SINPE
      XN(3)=XCO(3)+RE*COSTE
      IF (XN(1).NE.0..OR.XN(2).NE.0.) GO TO 194
      SINTE=SINTE+0.001
      GO TO 188
  190 ZL=P3
  191 XN(1)=XCO(1)+ZL*COSPE
      XN(2)=XCO(2)+ZL*SINPE
      XN(3)=ZE
      IF (XN(1).NE.0..OR.XN(2).NE.0.) GO  TO 194
      ZL=ZL+0.001
      GO TO 191
  194 PHIR=ATAN2(XN(2),XN(1))
      SINP=SIN(PHIR)
      COSP=COS(PHIR)
      RR=SQRT(XN(1)*XN(1)+XN(2)*XN(2)+XN(3)*XN(3))
      IF (LTEST) WRITE (6,195) XN(1),XN(2),XN(3)
  195 FORMAT (T20,6F12.5)
      COST=XN(3)/RR
      THER=ACOS(COST)
      THETA=THER*DPR
      GO TO 200
```

```
  196 THETA=P3
      THER=THETA/DPR
C
  200 SINT=SIN(THER)
      COST=COS(THER)
      EDX=(0.,0.)
      EDY=(0.,0.)
      EDZ=(0.,0.)
      IF (.NOT.LNF) GO TO 227
```

Aperture integration for near field
(Section 6)

```
C
C     *** SPILLOVER FIELDS FOR NEAR FIELD ***
C
      X1=XN(1)-XS(1)
      X2=XN(2)-XS(2)
      X3=XN(3)-XS(3)
      RHO=SQRT(X1*X1+X2*X2)
      PHIPR=BTAN2(X2,X1)
      PHIP=PHIPR*DPR
      PSI=BTAN2(RHO,-X3)*DPR
      RS=SQRT(RHO*RHO+X3*X3)
      PHEI=CEXP(-CJ*TPI*RS)*F/RS
      CALL FEED(PSI,PHIP,PSA,PHGAM)
      CALL FPOL(EIX,EIY,EIZ,PSA,PHGAM)
      EIX=EIX*PHEI
      EIY=EIY*PHEI
      EIZ=EIZ*PHEI
      IF (LOUT) WRITE (6,222) EIX,EIY,EIZ
  222 FORMAT(2H O,T15,5HEIX =,2E10.4,5X,5HEIY =,2E10.4,5X,5HEIZ =,
     22E10.4,T79,1HO)
C
C     ***OUTPUT RECTANGULAR COMPONENTS FOR CONSTANT Z NEAR FIELD ***
C
      EDX=EDX+EIX
      EDY=EDY+EIY
      EDZ=EDZ+EIZ
  224 IF (LRANG) GO TO 225
      CALL DBPHS(AEDX,EDX,0.)
      CALL DBPHS(AEDY,EDY,0.)
      CALL DBPHS(AEDZ,EDZ,0.)
      IF(LWRITE)WRITE (6,226) P3,AEDX,EDX,AEDY,EDY,AEDZ,EDZ
  226 FORMAT(2H W,T5,F6.1,4X,,3(E10.3,2F10.2))
      PLT=AEDZ
```

```
      GO TO 249
  225 EDT=COST*(COSP*EDX+SINP*EDY)-SINT*EDZ
      EDP=-SINP*EDX+COSP*EDY
      GO TO 242
```

## X-integration for far field (Section 7)

```
C
C     *** SPILLOVER FIELDS FOR FAR FIELD ***
C
      PHIP=PHI
      PSI=180.-THETA
      PSIR=PSI/DPR
      SINS=SIN(PSIR)
      COSS=COS(PSIR)
      CALL FEED(PSI,PHIP,PSA,PHGAM)
      CALL FPCL(EIX,EIY,EIZ,PSA,PHGAM)
      EIT=-COSS*COSP*EIX-COSS*SINP*EIY-SINS*EIZ
      EIP=-SINP*EIX+COSP*EIY
      PHEI=CEXP(CJ*TPI*ZO*COST)*F*RFCT
      EIT=EIT*PHEI
      EIP=EIP*PHEI
      EDT=EDT+EIT
      EDP=EDP+EIP
C
C     *** PRINCIPAL AND CROSS POLARIZED COMPONENTS FOR FAR FIELD ***
C                    AND CONSTANT RANGE NEAR FIELD
C
  242 IF (LTEST) WRITE (6,243) ACOSP,COSP,SINP,EDT,EDP
  243 FORMAT (T10,3F12.4,/T10,'EDT =',2F12.5,5X,'EDP =',2F12.5,/)
      TMT=EDT
      EDT=COSPT*EDT-SINPT*EDP
      EDP=SINPT*TMT+COSPT*EDP
      IF (.NOT.LCP) GO TO 245
      TMT=EDT
      EDT=TEM2*(EDT-CJ*EDP)
      EDP=TEM2*(TMT+CJ*EDP)
  245 CALL DBPHS(AEDT,EDT,REF)
      CALL DBPHS(AEDP,EDP,REF)
      IF (LTEST) WRITE (6,195) PHI,TAU,COSPT,SINPT
      IF(LWRITE)WRITE (6,248) P3,AEDT,EDT,AEDP,EDP
  248 FORMAT(2H W,T5,F6.1,4X,,2(E10.3,2F10.2),T79,1HW)
      PLT=REAL(EDT)
  249 CONTINUE
```

```
      IF (LPLT) WRITE (2) PLT
      P3=P3+DP3
  250 CONTINUE
      NGTD=NT-NAI
      IF (NGTD.LE.0) GO TO 270
      THI2=NAI*DP3+P3I
  255 CONTINUE
      IF (LTEST) WRITE (6,260) NT,NAI,NGTD,THI2
 260  FORMAT(2H D,T10,3I10,3F10.3,T79,1HD)
      IF(.NOT.LGTD)GO TO 270
      IF(LTEST.OR.LDEBUG)NTEST=2
C     CALL GTD(THI2,NGTD,NAI,DP3)
 270  CONTINUE
      GO TO 3000
  272 WRITE (6,275)
  275 FORMAT(2H E,T8,'** ERROR : DECLARED DIMENSION EXCEEDED',T79,
1HE)
      GO TO 3000
 285  WRITE (6,290)
  290 FORMAT(2H E,T10,'***ERROR : INPUT PHIN(1) OR PHIN(NPHI) '
     2'INCORRECT ***',T79,1HE)
      GO TO 3000
      END
```

## SECTION 1. RELATIVE POWER RADIATED BY FEED

### PURPOSE

To calculate the relative power radiated by feed using the input feed pattern data

### METHOD

The relative radiated power from the feed is given by

$$P_{rad} = \int_0^{2\pi}\int_0^{\pi} g^2(\psi,\phi)\sin\psi d\psi d\phi$$

where

$$g(\psi,\phi) = \frac{\phi-\phi_Q}{\phi_P-\phi_Q}\; g_P(\psi) + \frac{\phi_P-\phi}{\phi_P-\phi_Q}\; g_Q(\psi)$$

is obtained by linearly interpolating the input feed pattern $g_P$ and $g_Q$ in the two planes $\phi_P$ and $\phi_Q$ adjacent to $\phi$.

Since the feed pattern $g(\psi,\phi)$ is piecewise linear between two input PHI planes, the integration over $\phi$ reduces to a sum of integrals as

$$SPHI = \sum_1^{NM} \int_{\phi_Q}^{\phi_P}\int_0^{\pi} g^2(\psi,\phi)\sin\psi d\psi d\phi$$

where

$$\begin{cases} NM = NPHI & \text{if } IB=0 \text{ (no symmetry)} \\ NM = NPHI - 1 & \text{if } IB\neq 0 \text{ (with symmetry)} \end{cases}$$

and $\phi_P$ and $\phi_Q$ represent the upper and lower bound for each subregion. NPHI is the number of input feed cuts and IB is the absolute value of the symmetry index.

The integration over $\phi$ can be carried out analytically which gives

$$\text{SPHI} = \sum_{1}^{NM} \frac{\phi_P - \phi_Q}{3}\ \text{SPSI}$$

where

$$\text{SPSI} = \int_0^{\pi} (g_P^2 + g_Q^2 + g_P g_Q) \sin\psi d\psi$$

which is carried out numerically by using the trapezoidal rule.

Then the relative radiated power is given by

$$P_{rad} = \begin{cases} \text{SPHI} & \text{IB=0} \\ \text{2SPHI} & \text{IB=2 or 3} \\ \text{4SPHI} & \text{IB=1} \end{cases}$$

The relative power radiated is used for the purpose of calculating the far field results in antenna gain when the range factor $e^{-jkR}/R$ is suppressed. This is done through the variable REFDB as given by

$$\text{REFDB} = 10 \log \frac{4\pi\lambda^2}{F^2 P_{rad}}$$

which is calculated later in the main program and used as input to the subroutine DBPHS.

## KEY VARIABLES

| | | |
|---|---|---|
| DELI | | Angular increments for numerical integration over $\psi$ |
| GFP | $(g_P(\psi))$ | Calculated feed value in the plane $\phi_P$ at angle $\psi$ |
| GFQ | $(g_Q(\psi))$ | Calculated feed value in the plane $\phi_Q$ at angle $\psi$ |
| IB | | Absolute value of the symmetry index (see User's Manual) |
| NM | | Number of integration regions over $\phi$ |
| NPHI | | Number of input feed cuts |
| PHIP | $(\phi_P)$ | Upper input PHI cut adjacent to $\phi$ |
| PHIQ | $(\phi_Q)$ | Lower input PHI cut adjacent to $\phi$ |
| PRAD | $(P_{rad})$ | Power radiated from feed |
| PSI | $(\psi)$ | Theta coordinate angle of the observation direction referred to the feed axis |
| SPHI | | Sum of numerical integration over $\phi$ |
| SPSI | | Sum of numerical integration over $\psi$ |

CODE LISTING

```
C
C     *** CALCULATE POWER RADIATED BY FEED ***
C
 80   IN=181
      DELI=PI/(IN-1)
      SPHI=0.
      IF (IB.EQ.0) NM=NPHI
      IF (IB.NE.0) NM=NPHI-1
      IF(LTEST.OR.LDEBUG)NTEST=1
      DO 94 NQ=1,NM
      NP=NQ+1
      IF (NP.GT.NPHI) NP=1
      PHIQ=PHIN(NQ)+0.001
      PHIP=PHIN(NP)-0.001
      PSIR=0.
      SPSI=0.
      DO 92 I=1,IN
      IF(I.EQ.4)NTEST=0
      PSI=PSIR*DPR
      CALL FEED(PSI,PHIQ,PSA,PHGAM)
      GFQ=GF
      CALL FEED(PSI,PHIP,PSA,PHGAM)
      GFP=GF
      FI=GFQ*GFQ+GFP*GFP+GFQ*GFP
      IF (I.EQ.1.OR.I.EQ.IN) FI=FI/2.
      SPSI=SPSI+FI*DELI*SIN(PSIR)
      PSIR=PSIR+DELI
      IF(NTEST.EQ.1)WRITE(6,90)PSIR,SPSI
90    FORMAT(2H *,T12,'PSIR=',F7.2,5X,'SPSI=',F10.3,T79,1H*)
 92   CONTINUE
      NTEST=0
      DPHI=(PHIP-PHIQ)/DPR
      IF (DPHI.LT.0.) DPHI=DPHI+TPI
      SPHI=SPHI+SPSI*DPHI/3.
 94   CONTINUE
      PRAD=SPHI
      IF (IB.EQ.2.OR.IB.EQ.3) PRAD=2.*PRAD
      IF (IB.EQ.1) PRAD=4.*PRAD
      WRITE (6,95) PRAD
 95   FORMAT(2H *,T10,6HPRAD =,E10.3,,T79,1H*)
      PSIT=TEMP
```

## SECTION 2. RIM POINT CALCULATION FOR CIRCULAR APERTURE

### PURPOSE

To calculate the rim point coordinates of a circular aperture.

### METHOD

For circular reflectors, the rim points are not input as they are for noncircular rim shapes. Instead, the diameter of the aperture (D) and the y-coordinate of aperture center ($Y_c$) are input.

The coordinates of the rim points of a circular aperture are then calculated as follows:

First an approximate value of the number of rim points is estimated by

$$\mathrm{NRIM}_{(\mathrm{APP})} = \mathrm{Int}(\pi D/\mathrm{RIML})$$

where

Int(X) means the integer value of X,

D is the diameter of the aperture, and

RIML is the reference length of each rim segment.

Then the actual value of NRIM is obtained by adjusting the estimated value such that NRIM is a multiple of 4 and is given by

$$\mathrm{NRIM} = 4\left(\mathrm{Int}\left(\frac{\mathrm{NRIM}_{(\mathrm{APP})}+2}{4}\right)\right) \quad .$$

This adjustment is done for the purpose of having symmetrical rim segments.

To maintain approximately the same aperture area as the original aperture, the polar distance of each rim point is determined by taking the average polar distance to the corners of an inscribed regular polygon and a circumscribed regular polygon to the original circle, thus the polar distance

$$AA = \frac{a}{2}\left(1 + \frac{1}{\cos\left(\frac{\Delta\phi}{2}\right)}\right)$$

where a is the radius of the circular aperture and

$$\Delta\phi = \frac{2\pi}{NRIM}$$

The rim points are then calculated by

$$X_{RIM} = AA \cos\phi_{en}$$

$$Y_{RIM} = AA \sin\phi_{en} + Y_C$$

where

$$\phi_{en} = \left(n - \frac{1}{2}\right)\Delta\phi \qquad n=1,2,3,\cdots,NRIM$$

and $Y_C$ is the y-coordinate of the aperture center.

KEY VARIABLES

| | | |
|---|---|---|
| A | ($a$) | Radius of the circular aperture |
| AA | | Polar distance of each rim point |
| D | ($D$) | Diameter of the circular aperture |
| DELP | ($\Delta\phi$) | Sector angle associated with each rim segment |
| NRIM | | Number of rim segments |
| PHE | ($\phi_{en}$) | Polar angle of a rim point |
| RIML | | Reference rim segment length |
| RIMS | | X and Y coordinates of rim point ME |
| YCM | ($Y_C$) | Y-coordinate of the center of aperture |

```
C
C      *** CIRCULAR RIM SECTION ***
C
       NRIM=PI*D/RIML
       NRIM=4*((NRIM+2)/4)
       IF(NRIM.LT.16)NRIM=16
       IF (NRIM.GT.MDRIM) NRIM=MDRIM
       WRITE (6,100) NRIM
  100  FORMAT(2H *,T10,'NUMBER OF RIM SEGMENTS=',I3,T79,1H*)
       WRITE(6,3006)
       RIML=-1.
       DELP=2.*PI/NRIM
       PHE=0.5*DELP
C!!!   USE THE AVERAGE RADIUS TO COMPUTE RIM POINTS FOR CIRCULAR REF
L.
       AA=0.5*A*(1.+1./COS(PHE))
       DO 102 NE=1,NRIM
       RIMS(NE,1)=AA*COS(PHE)
       RIMS(NE,2)=AA*SIN(PHE)+YCM
       PHE=PHE+DELP
  102  CONTINUE
       WRITE(6,3006)
       IF (D.GT.0.) WRITE (6,103) D,YC
 103   FORMAT(2H *,T10,'APERTURE DIAMETER =',F9.2,' WAVELENGTHS',T79
,
      21H*,/2H *,T79,1H*,/2H *,T10,'APERTURE CENTER AT (0.,',F7.2,'
)',
      3T79,1H*)
```

## SECTION 3. APERTURE FIELDS

### PURPOSE

To calculate and store the aperture fields on the principal rectangular grid.

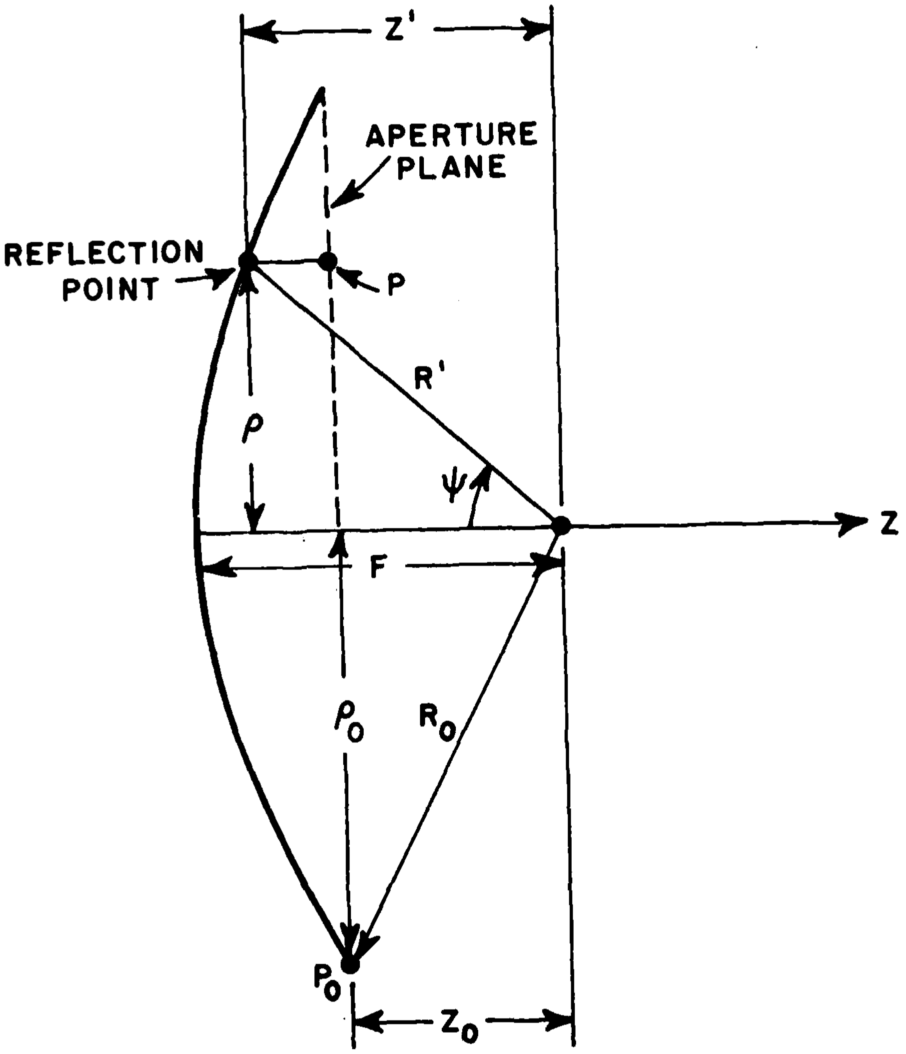


Figure 1. Coordinate system for the aperture field.

## METHOD

The coordinate information for the aperture field is defined by the point of reflection on the reflector surface with coordinates X,Y as shown in Fig. 1. Thus

$$\phi' = \tan^{-1} \frac{Y}{X}$$

$$\rho = \sqrt{X^2+Y^2}$$

$$Z' = F - \frac{\rho^2}{4F}$$

$$R' = \sqrt{\rho^2 + Z'^2}$$

and

$$\psi = \tan^{-1} \frac{\rho}{Z'}$$

Let the vector incident feed pattern in the direction $(\psi,\phi)$ be of the form as

$$\overline{f^i} = \hat{\theta}\, f_\theta + \hat{\phi}'\, f_\phi$$

where $f_\theta$ and $f_\phi$ are the feed pattern values calculated in subroutines FEED and FPOL. The reflected field pattern from the parabolic surface is given by

$$\overline{f^r} = \hat{\rho}\, f_\theta - \hat{\phi}'\, f_\phi \quad .$$

Its corresponding rectangular components can be expressed as

$$f_x^r = \cos\phi' f_\theta + \sin\phi' f_\phi$$

$$f_y^r = \sin\phi' f_\theta - \cos\phi' f_\phi$$

The aperture plane is defined as the plane perpendicular to the Z-axis and passing through the rim point $P_0(X_0,Y_0,0)$ with the greatest distance

$\rho_o$ from the Z-axis. The aperture field at the point P(X,Y,0) on the aperture plane is given by

$$E_x^a = F f_x^r \frac{e^{-jkR_o}}{R'}$$

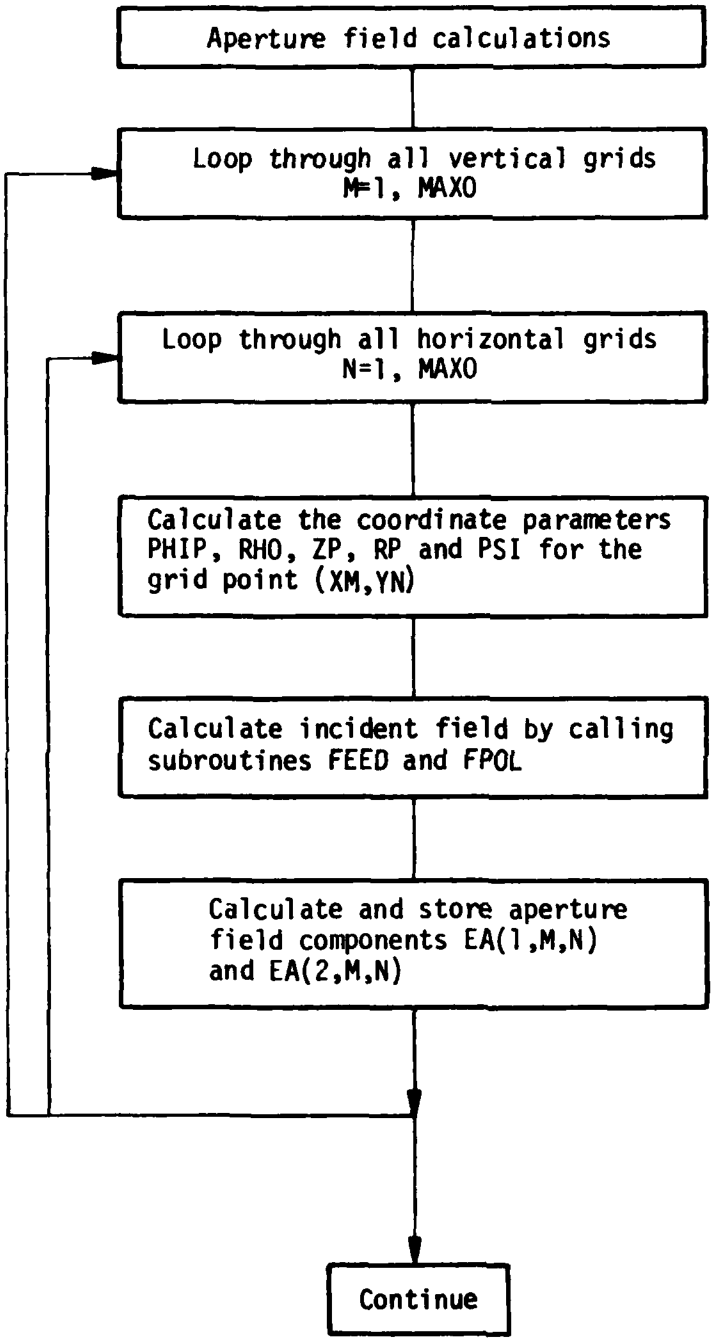
and

$$E_y^a = F f_y^r \frac{e^{-jkR_o}}{R'}$$

where F is the focal distance.

For any grid point $(X_M, Y_N)$, the X and Y components, $E_x^a$ and $E_y^a$ of the aperture field are calculated by the above two equations. By looping through all the horizontal and vertical grid lines, a two dimensional array of the aperture fields is set and stored.

FLOW DIAGRAM



## KEY VARIABLES

| | | |
|---|---|---|
| EA(1,M,N) | ($E_x^a$) | X component of the aperture field at grid point (XM,YN) |
| EA(2,M,N) | ($E_y^a$) | Y component of the aperture field at grid point (XM,YN) |
| EIP | ($f_\phi$) | PHI component of feed pattern |
| EIT | ($f_\theta$) | THETA component of feed pattern |
| EIX | | X component of feed pattern |
| EIY | | Y component of feed pattern |
| EIZ | | Z component of feed pattern |
| ERX | ($f_x^r$) | X component of the reflected field pattern |
| ERY | ($f_y^r$) | Y component of the reflected field pattern |
| M | | Index of vertical grid line |
| N | | Index of horizontal grid line |
| MAXO | | Maximum number of horizontal and Vertical grid line |
| PHSEA | ($e^{-jkR_o}$) | Phase factor of the aperture field |
| PHIP | ($\phi'$) | PHI coordinate of grid point (XM,YN) |
| PSI | ($\psi$) | THETA coordinate of grid point (XM,YN) measured from the negative Z-axis |
| RP | ($R'$) | The distance from the focal point to the reflection point |
| XX | ($X$) | X-coordinate of the reflection point |
| YY | ($Y$) | Y-coordinate of the reflection point |
| ZP | ($Z'$) | The projected distance of RP on the Z-axis |

## CODE LISTING

```
C
C      *** CALCULATE APERTURE FIELDS ***
C
       DO 120 M=1,MAXO
       I=M-1
       XX=(I-ICO)*GRDX
       IF (LNF.AND.(M.EQ.1)) XX=XMIN
       IF (LNF.AND.(M.EQ.(MAX+1))) XX=XMAX
       DO 120 N=1,MAXO
       J=N-1
       YY=(J-JCO)*GRDY
       PHIPR=BTAN2(YY,XX)
       PHIP=PHIPR*DPR
       RHO=SQRT(XX*XX+YY*YY)
       ROSQ=RHO*RHO
       ZP=F-ROSQ/(4.*F)
       RP=SQRT(ROSQ+ZP**2)
       PSIR=BTAN2(RHO,ZP)
       PSI=PSIR*DPR
       CALL FEED(PSI,PHIP,PSA,PHGAM)
       CALL FPOL(EIX,EIY,EIZ,PSA,PHGAM)
       SINPP=SIN(PHIPR)
       COSPP=COS(PHIPR)
       SINS=SIN(PSIR)
       COSS=COS(PSIR)
       EIT=-COSS*COSPP*EIX-COSS*SINPP*EIY-SINS*EIZ
       EIP=-SINPP*EIX+COSPP*EIY
       NTEST=0
       ERX=COSPP*EIT+SINPP*EIP
       ERY=SINPP*EIT-COSPP*EIP
       EA(1,M,N)=F*ERX/RP*PHSEA
       EA1=EA(1,M,N)
       CALL DBPHS(AE1,EA1,0.)
       EA(2,M,N)=F*ERY/RP*PHSEA
       EA2=EA(2,M,N)
       CALL DBPHS(AE2,EA2,0.)
       IF (.NOT.LDEBUG) GO TO 120
       IF (M.LE.MNO.AND.N.LE.MNO) WRITE (6,118) M,N,EA1,EA2
   118 FORMAT(2H D,T15,2I5,4F10.2,T79,1HD)
  120  CONTINUE
 121   CONTINUE
```

## SECTION 4. Y-INTEGRATION FOR FAR FIELD

### PURPOSE

To numerically integrate the aperture fields along the rotated $\phi$-grid lines.

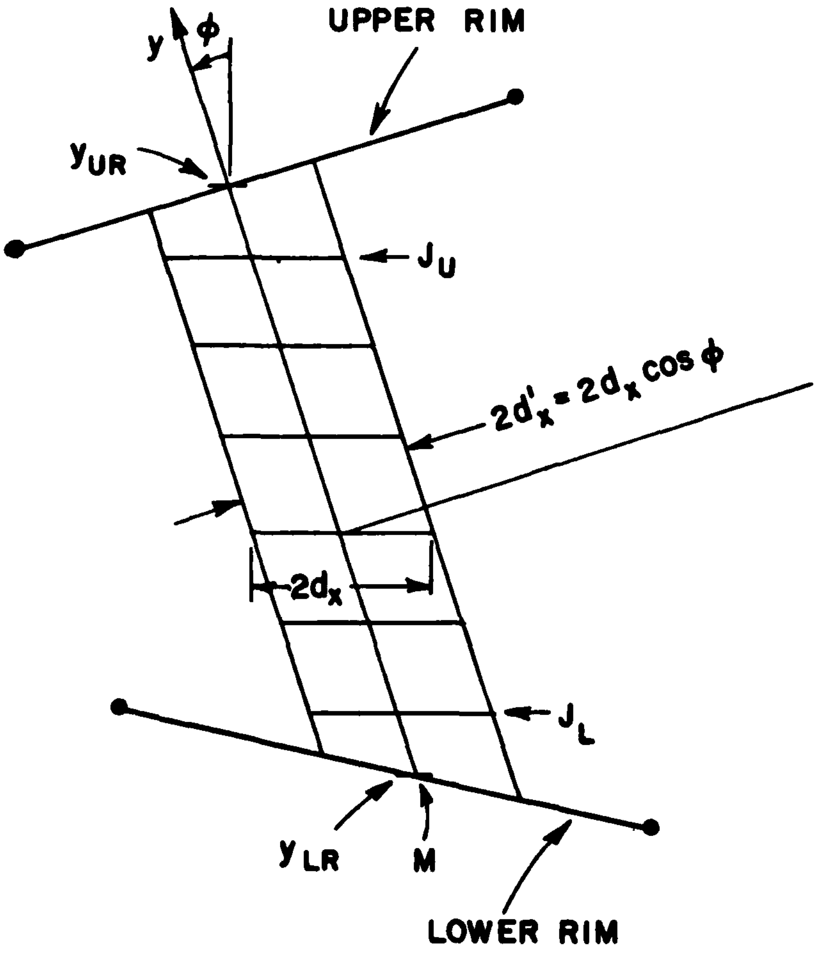


Figure 1. Geometry of y-integration for far field.

## METHOD

The y-integration along the rotated grid line (M) as shown in Fig. 1 is represented by

$$Y_{SUM}(M) = \int_{y_{LR}}^{y_{UR}} E^a \, dy \tag{1}$$

where $y_{LR}$ and $y_{UR}$ correspond to the intersections of the grid line (M) with the lower and upper rims, respectively, as shown in Fig. 1 and $E^a$ is the aperture field distribution along the grid line.

To determine these intersection points, the x-coordinate of the grid line (M) is compared with those of the rim points until a rim segment is found such that the x-coordinate of the grid line (M) is in between those of the two rim points of that segment. Then $y_{LR}$ or $y_{UR}$ is obtained by solving for the intersection point of the rim segment and the grid line (M).

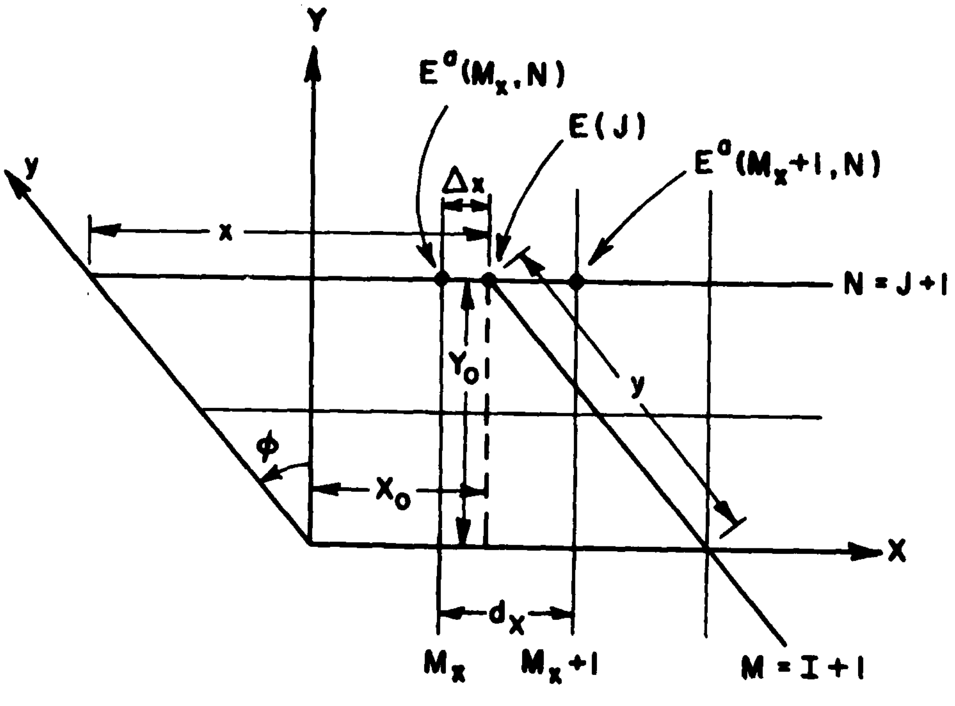


Figure 2. Geometry for interpolation of aperture field.

In order to carry out the y-integration in Equation (1) the aperture field $E^a$ is calculated by interpolation from its stored values corresponding to points on the principal rectangular grid. The geometry for interpolation is shown in Fig. 2 where the rotated grid line I=M-1 intersects the horizontal grid line N=J+1 at the point with principal grid coordinates $(X_0,Y_0)$. The principal grid coordinates are related to the rotated grid coordinates by

$$Y_0 = (J-J_c)d_y \cos\phi \tag{2}$$

and

$$X_0 = x - Y_0 \tan\phi \tag{3}$$

where

$$x = (I-I_c)d_x \quad .$$

The principal grid coordinates are then used to determine the integer value $M_x$ for the nearest principal vertical grid line to the left of the point $(X_0,Y_0)$ with aperture field E(J) as shown in Fig. 2. Thus, interpolation yields the aperture field at the point on the rotated grid as

$$E(J) = \left(1 - \frac{\Delta x}{d_x}\right) E^a(M_x,N) + \frac{\Delta x}{d_x} E^a(M_x+1,N) \tag{4}$$

where $\Delta x$ is the displacement of the aperture field point from the vertical grid line $(M_x)$.

Let $J_L$ and $J_U$ represents the indices of the lower and upper grid lines closest to the intersection $y_{LR}$ and $y_{UR}$ inside the aperture respectively. If $J_U - J_L \leq 1$, Equation (1) is approximated by

$$Y_{SUM}(M) = (y_{UR}-y_{LR})E(J_U) \quad . \tag{5}$$

If $J_U - J_L > 1$, the y-integration is divided into three parts as shown in Fig. 3. Using the subaperture method, the middle part $Y_{SM}$ is given by

$$Y_{SM} = \sum_{J_L+1}^{J_U-1} E(J) \tag{6}$$

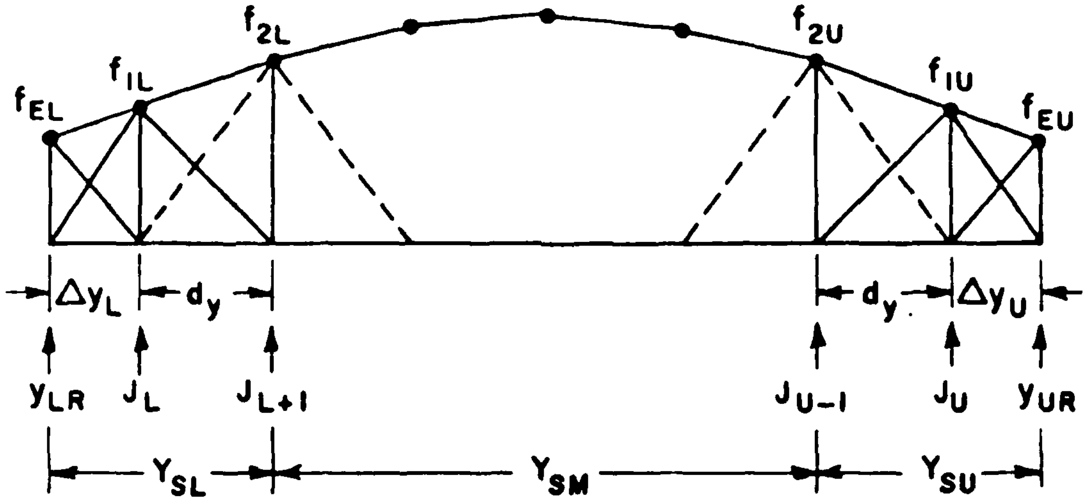


Figure 3. The *y*-integration parts for far field.

The lower part of the y-integration consists of the contribution $Y_{SL}$ from $y_{LR}$ to the $(J_L+1)$ grid line and the upper part consists of the contribution $Y_{SU}$ from the $(J_U-1)$ grid line to $y_{UR}$

The contributions $Y_{SU}$ from the upper part is given in terms of the aperture field values $f_{1U} = E(J_U)$ and $f_{EU} = E(y=y_{UR})$ as shown in Fig. 3. The edge value $f_E = f_{EU}$ is calculated by linear extrapolation from

$$f_E = f_1 + (f_1 - f_2)\,\frac{\Delta y}{dy} \tag{7}$$

The contribution $Y_{SL}$ from the lower part is obtained in a similar way. Thus both contributions can be represented by

$$Y_S dy = \frac{1}{2}\,dy\ f_1 + \frac{1}{2}\,\Delta y\ f_1 + \frac{1}{2}\,\Delta y\ f_E \quad . \tag{8}$$

Substituting Equation (7) into Equation (8) and simplifying terms yields

$$Y_{SL} = \frac{1}{2}\left[\left(1 + \frac{\Delta y_L}{dy}\right)^2 E(J_L) - \left(\frac{\Delta y_L}{dy}\right)^2 E(J_L+1)\right] \tag{9}$$

and

$$Y_{SU} = \frac{1}{2}\left[\left(1 + \frac{\Delta y_U}{dy}\right)^2 E(J_U) - \left(\frac{\Delta y_U}{dy}\right)^2 E(J_U-1)\right] \tag{10}$$

Thus the y-integration of Equation (1) can be calculated from

$$Y_{SUM}(M) = (Y_{SL}+Y_{SM}+Y_{SU})d_y \tag{11}$$

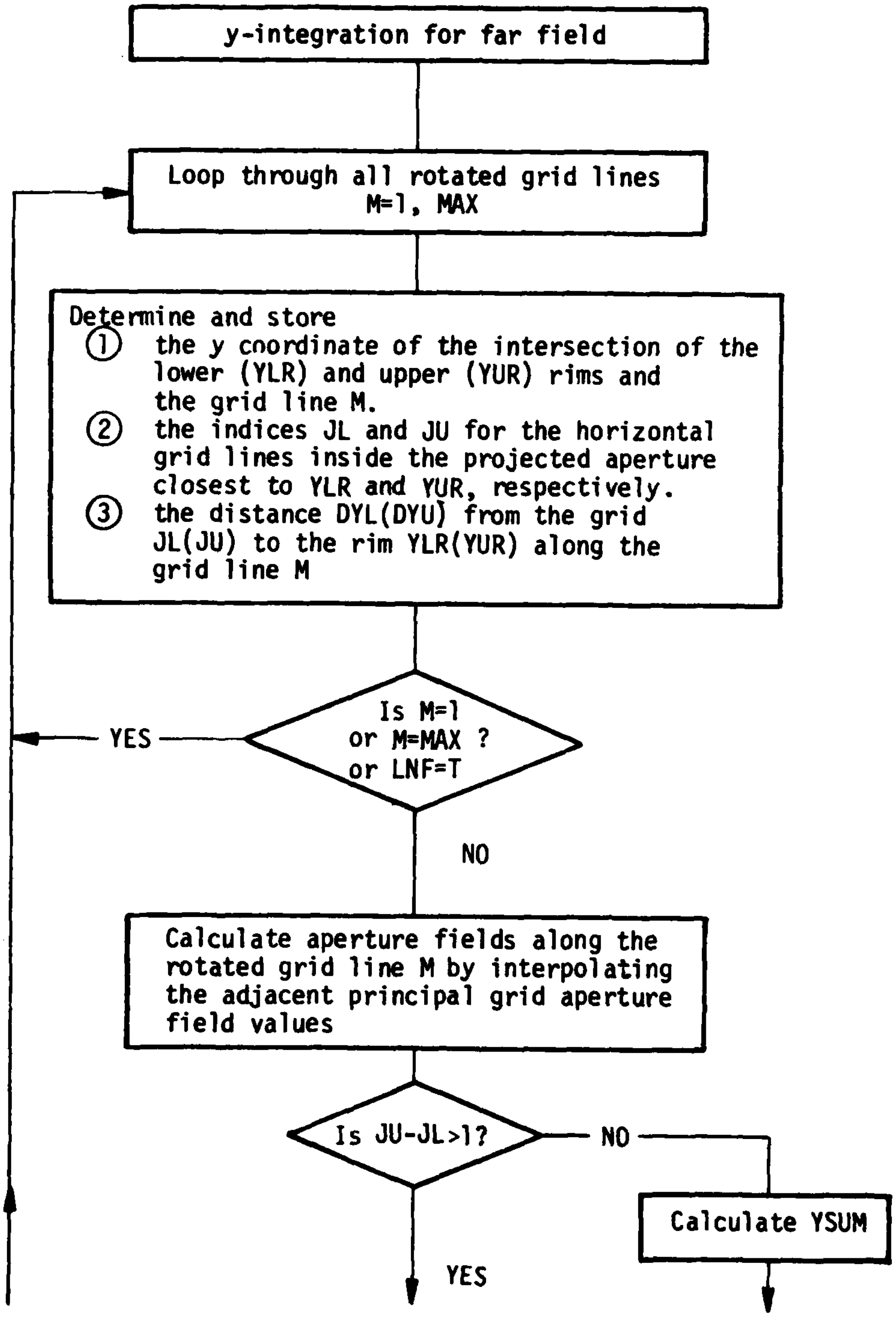
In general, the aprture field can be decomposed into x and y components. Thus two y-integration sums $Y_{SUM}(1,M)$ and $Y_{SUM}(2,M)$ are obtained by carrying the y-integration for each component respectively, i.e.,

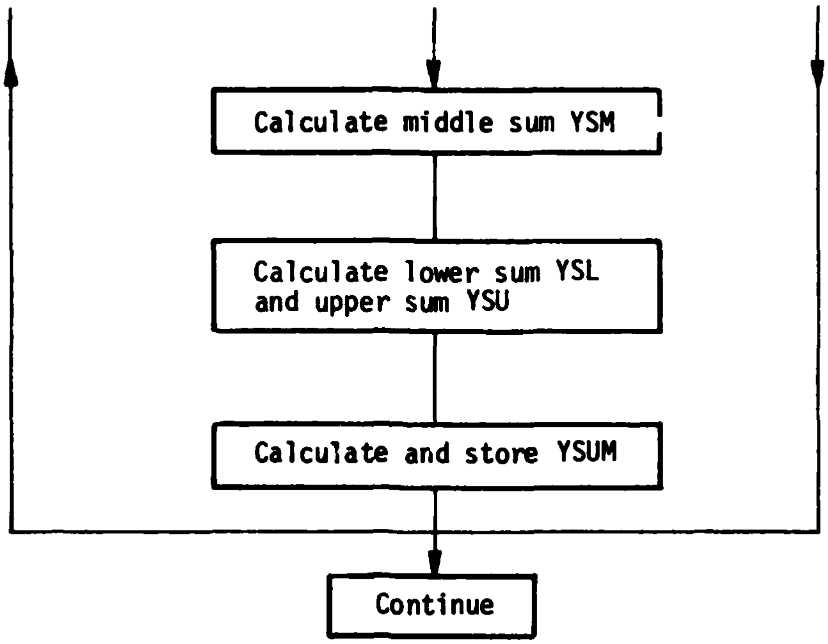
$$Y_{SUM}(1,M) = \int_{y_{LR}}^{y_{UR}} E_x^a\, dy \tag{12}$$

and

$$Y_{SUM}(2,M) = \int_{y_{LR}}^{y_{UR}} E_y^a\, dy \tag{13}$$

FLOW DIAGRAM




Calculate middle sum YSM
Calculate lower sum YSL and upper sum YSU
Calculate and store YSUM
Continue

KEY VARIABLES

| | | |
|---|---|---|
| DYL | ($\Delta Y_L$) | The distance from the horizontal grid line JL to the lower rim along the rotated grid line M |
| DYU | ($\Delta Y_U$) | The distance from the horizontal grid line JU to the upper rim along the rotated grid line M |
| E(1,J) | | X component of the interpolated aperture field on the rotated grid |
| E(2,J) | | Y component of the interpolated aperture field on the rotated grid |
| EA(1,M,N) | ($E_x^a$) | X component of the aperture field at grid point (XM,YN) |
| EA(2,M,N) | ($E_y^a$) | Y component of the aperture field at grid point (XM,YN) |
| GRDX | ($d_x$) | Grid size along the X-axis |
| GRDY | ($d_y$) | Grid size along the rotated Y-axis |
| GRIDY | ($D_x$) | Grid size along the principal Y-axis |
| IC | | Vertical grid line index of the origin of the reflector coordinate system |
| JC | | Horizontal grid line index of the origin of the reflector coordinate system |
| JLO | ($J_L$) | Index for the horizontal grid line inside the projected aperture closest to the lower intersection point on the grid line M |
| JUO | ($J_U$) | Index for the horizontal grid line inside the projected aperture closest to the upper intersection point on the grid line M |
| MAX | | Maximum number of rotated grid lines |
| MX | | Index of vertical principal grid line |
| QDX | | Normalized distance from the integration point to the vertical grid line M |
| QYL | | Normalized distance from the lower rim to the grid line JL |

| | | |
|---|---|---|
| QYU | | Normalized distance from the upper rim to the grid line JU |
| XO | ($X_o$) | X-coordinate of the integration point in the principal grid system |
| XX | (X) | X-coordinate of the integration point in the rotated gird system |
| YLR | ($y_{LR}$) | Y-coordinate of the intersection of the grid line M and the lower rim |
| YO | ($Y_o$) | Y-coordinate of the integration point in the principal grid system |
| YSLX | | X-component of the lower sum of the Y-integration |
| YSLY | | Y-component of the lower sum of the Y-integration |
| YSMX | | X-component of the middle sum of the Y-integration |
| YSMY | | Y-component of the middle sum of the Y-integration |
| YSUX | | X-component of the upper sum of the Y-integration |
| YSUY | | Y-component of the upper sum of the Y-integration |
| YSUM(1,M) | | X-component of the total sum of the Y-integration for grid line M |
| YSUM(2,M) | | Y-component of the total sum of the Y-integration for grid line M |
| YUR | ($y_{UR}$) | Y-coordinate of the intersection of the grid line M and the upper rim |

```
C
C      *** Y INTEGRATION FOR FAR FIELD ***
C
      DO 172 M=1,MAX
      I=M-1
      XX=(I-IC)*GRDX
      IF (M.EQ.1) XX=XMIN
      IF (M.EQ.MAX) XX=XMAX
      IF (M.EQ.1) GO TO 150
      IF (XX.LE.XLKP) GO TO 152
  150 K=K+1
      IF (K.GE.NLRIM) GO TO 153
      XLK=CLRIM(K,1)
      XLKP=CLRIM(K+1,1)
      IF (XX.GT.XLKP) GO TO 150
      YLK=CLRIM(K,2)
      YLKP=CLRIM(K+1,2)
      IF (LDEBUG) WRITE (6,152) K,XLK,XLKP,YLK,YLKP
  152 FORMAT(2H D,T5,3HK =,I2,5X,5HXLK =,F6.2,5X,6HXLKP =,F6.2,
     25X,5HYLK =,F6.2,5X,6HYLKP =,F6.2,T79,1HD)
      TEMP=(YLKP-YLK)/(XLKP-XLK)
  153 YLR(M)=YLK+TEMP*(XX-XLK)
      JL(M)=(YLR(M)/GRDY)+JC+0.99
      IF (M.EQ.1) GO TO 154
      IF (XX.LE.XUKP) GO TO 158
  154 L=L+1
      IF (L.GE.NURIM) GO TO 158
      XUK=CURIM(L,1)
      XUKP=CURIM(L+1,1)
      IF (XX.GT.XUKP) GO TO 154
      YUK=CURIM(L,2)
      YUKP=CURIM(L+1,2)
      IF (LDEBUG) WRITE (6[illegible] L,XUK,XUKP,YUK,YUKP
  156 FORMAT(2H D,T5,3HL[illegible],5HXUK =,F6.2,5X,6HXUKP =,F6.2,
     25X,5HYUK =,F6.2,5X,6[illegible]F6.2,T79,1HD)
      TENP=(YUKP-YUK)/(XUKP-XU[illegible]
  158 YUR(M)=YUK+TENP*(XX-XUK)
      JU(M)=(YUR(M)/GRDY)+JC+0.01
      IF (JU(M).LT.JL(M)) JU(M)=JL(M)
      IF (LDEBUG) WRITE (6,160) JL(M),JU(M),YLR(M),YUR(M)
  160 FORMAT(2H D,T12,'JL=',I2,' JU=',I2,' YLR=',F8.2,' YUR=',F8.2,
     1T79,1HD)
      DYL=(JL(M)-JC)*GRDY-YLR(M)
      OYL(M)=DYL/GRDY
      DYU=YUR(M)-(JU(M)-JC)*GRDY
      OYU(M)=DYU/GRDY
      IF (LDEBUG) WRITE (6,909) JL(M),JU(M),DYL,DYU,YLR(M)
     2,YUR(M),OYL(M),OYU(M)
  909 FORMAT (2I5,8F10.4)
```

```
      IF ((M.EQ.1.OR.M.EQ.MAX).OR.LNF) GO TO 172
      GDY=GRDY*ACOSP
      JLO=JL(M)
      JUO=JU(M)
      DO 165 J=JLO,JUO
      IF (NCK.NE.1) GO TO 162
      E(1,J)=(0.,0.)
      E(2,J)=(1.,0.)
      GO TO 165
  162 YO=(J-JC)*GDY
      XX=(I-IC)*GRDX
      XO=XX-YO*TANP
      FIX=XO/GRDX+ICOP
      IX=FIX
      MX=IX+1
      ODX=FIX-IX
      N=J+1
      E(1,J)=EA(1,MX,N)*(1.-ODX)+EA(1,MX+1,N)*ODX
      E(2,J)=EA(2,MX,N)*(1.-ODX)+EA(2,MX+1,N)*ODX
      IF (LDEBUG) WRITE (6,164) J,E(1,J),E(2,J),ODX
  164 FORMAT(2H D,I10,5F12.4,T79,1HD)
  165 CONTINUE
      IF (JUO-JLO.GT.1) GO TO 168
      YSUM(1,M)=(YUR(M)-YLR(M))*E(1,JUO)
      YSUM(2,M)=(YUR(M)-YLR(M))*E(2,JUO)
      IF (LWYSUM) WRITE (6,166) M,YSUM(1,M),YSUM(2,M)
  166 FORMAT (2H D,I5,8E10.3)
      GO TO 172
  168 CONTINUE
C
C         *** CALCULATE YSM ***
C
      JF=JUO-1
      JI=JLO+1
      KM=JF-JI+1
      YSMX=(0.,0.)
      YSMY=(0.,0.)
      DO 170 KJ=1,KM
      J=KJ+JI-1
      YSMX=YSMX+E(1,J)
      YSMY=YSMY+E(2,J)
  170 CONTINUE
C
C         *** CALCULATE YSL AND YSU ***
C
      YSLX=(E(1,JLO)*(OYL(M)+1.)**2-E(1,JLO+1)*OYL(M)**2)/2.
      YSLY=(E(2,JLO)*(OYL(M)+1.)**2-E(2,JLO+1)*OYL(M)**2)/2.
      YSUX=(E(1,JUO)*(OYU(M)+1.)**2-E(1,JUO-1)*OYU(M)**2)/2.
      YSUY=(E(2,JUO)*(OYU(M)+1.)**2-E(2,JUO-1)*OYU(M)**2)/2.
      YSUM(1,M)=(YSLX+YSMX+YSUX)*GRDY
      YSUM(2,M)=(YSLY+YSMY+YSUY)*GRDY
      IF (LWYSUM) WRITE (6,166) M,YSLX,YSMX,YSUX,YSUM(1,M)
      IF (LWYSUM) WRITE (6,166) M,YSLY,YSMY,YSUY,YSUM(2,M)
  172 CONTINUE
C
```

## SECTION 5. SWITCHING CRITERION FOR AI AND GTD

### PURPOSE

To calculate the switching criterion between AI and GTD in the near field or far field computation.

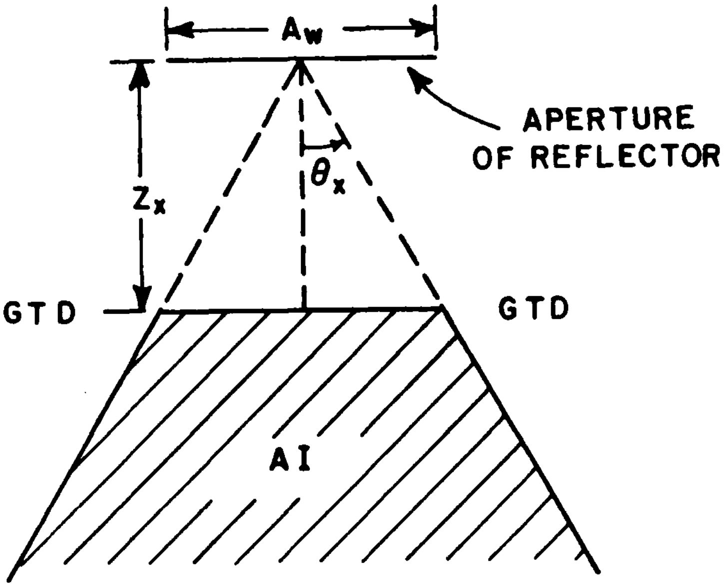


Figure 1. Geometry for switching criterion between AI and GTD.

### METHOD

The angle criterion which is used for the near field as well as the far field, is defined as

$$\theta_x = \sin^{-1}\left(\frac{1}{\sqrt{Aw}}\right)$$

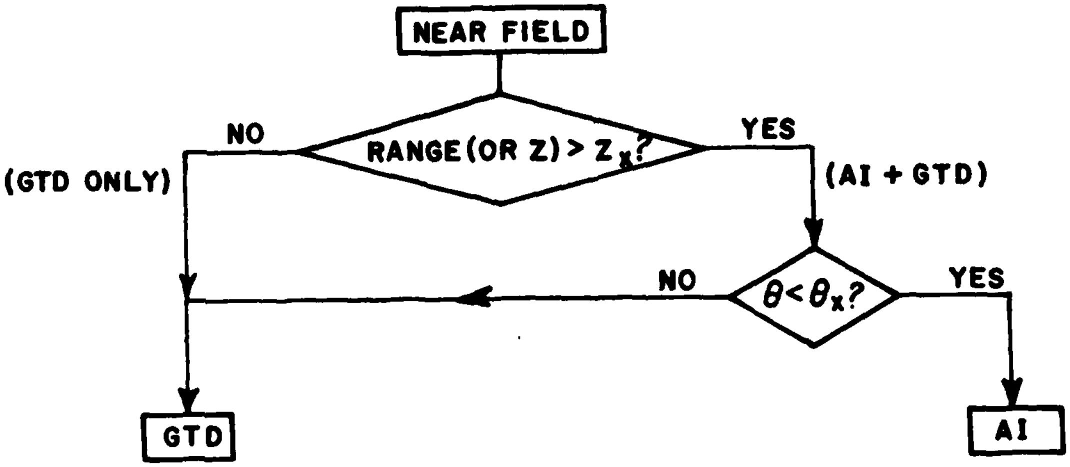
where Aw is the aperture width in the specific pattern cut. Thus AI is used when $0<\theta<\theta_x$ and GTD is used when $\theta \geq \theta_x$.

The range criterion is used solely for the near field and is defined by

$$Z_x = \frac{Aw}{2\tan\theta_x}$$

as shown in Fig. 1.

The usage of the above criteria for near field computation is summarized in the flow diagram as shown below.



KEY VARIABLES

| | | |
|---|---|---|
| AW | $(A_w)$ | Aperture width in the specific pattern cut |
| P3X | | Variable representing the switching criterion |
| THETAX | $(\theta_X)$ | Angle criterion in degrees |
| THEX | $(\theta_X)$ | Angle criterion in radians |
| ZX | $(Z_X)$ | Range criterion |

```
C
C      *** SET UP SWITCHING CRITERION ***
C
       P3X=P3F
       IF (.NOT.LGTD) GO TO 179
       THEX=THETAX/DPR
       TANX=TAN(THEX)
       IF (ZX.GT.0..AND.THETAX.GT.0.) GO TO 177
       AW=RHOS(1)-RHOS(2)
       THEX=ASIN(1./SQRT(AW))
       TANX=TAN(THEX)
       IF (LNF.AND.(TANX.NE.0.)) ZX=0.5*AW/TANX
       THETAX=THEX*DPR
  177  P3X=THETAX
       IF (.NOT.LNF) GO TO 179
       IF (P2.LT.ZX) GO TO 180
       IF (LRANG) GO TO 179
       P3X=P2*TANX
  179  NAI=(P3X-P3I)/DP3+1.1
       IF (NT.LT.NAI) NAI=NT
       IF (NAI.GT.0) GO TO 182
 180   NAI=0
       NGTD=NT
       THI2=P3I
       GO TO 255
```

## SECTION 6. APERTURE INTEGRATION FOR NEAR FIELD

### PURPOSE

To numerically integrate the aperture fields for near field calculations and express the field in rectangular components.

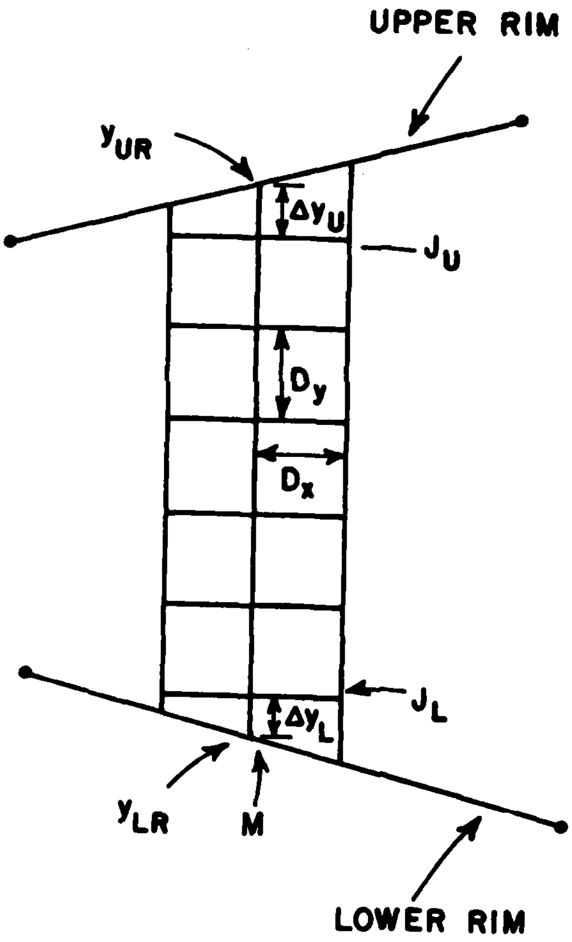


Figure 1. Geometry of y-integration for near field.

## METHOD

For aperture fields with arbitrary polarization having both x and y components, the near field can be expressed as

$$\bar{E} = \frac{jk}{2\pi} \iint \left[ \bar{F}_x \, E^a_x + \bar{F}_y \, E^a_y \right] \frac{e^{-jks}}{s} \, dx \, dy \qquad (1)$$

where $\bar{F}_x$ and $\bar{F}_y$ are the vector element patterns for the respective x and y components $(E^a_x, E^a_y)$ of the aperture field.

By integrating numerically Equation (1) can be expressed in a sum of series form as

$$\bar{E} = \frac{j}{\lambda} \sum_M \sum_N \left[ \bar{F}_{XMN} E^a_{XMN} + \bar{F}_{YMN} E^a_{YMN} \right] F_{RS} \frac{e^{-jks}}{s} \qquad (2)$$

where $\bar{F}_{XMN}$ and $\bar{F}_{YMN}$ are the vector element patterns of the equivalent aperture currents. These are assumed to radiate the same polarization as a Huygen's source and thus the vector element patterns are expressed in rectangular coordinates as

$$\begin{aligned}
\bar{F}_{XMN} &= \{\hat{x}[1+(\cos\theta_{MN} - 1)\cos^2\phi_{MN}] \\
&\quad + \hat{y}(\cos\theta_{MN} - 1)\sin\phi_{MN}\cos\phi_{MN} \\
&\quad - \hat{z}\,\sin\theta_{MN}\cos\phi_{MN}\}\,\cos\left(\frac{\theta_{MN}}{2}\right) \\
&= \{\hat{x}\,C_{xx} + \hat{y}\,C_{xy} - \hat{z}\,\sin\theta_{MN}\cos\phi_{MN}\}\mathrm{ELPAT}
\end{aligned} \qquad (3)$$

$$\begin{aligned}
\bar{F}_{YMN} &= \left\{ \hat{x}(\cos\theta_{MN} - 1)\sin\phi_{MN}\cos\phi_{MN} \right. \\
&\quad + \hat{y}[1+(\cos\theta_{MN} - 1)\sin^2\phi_{MN}] \\
&\quad \left. - \hat{z}\,\sin\theta_{MN}\sin\phi_{MN} \right\} \cos\left(\frac{\theta_{MN}}{2}\right) \\
&= \{\hat{x}\,C_{xy} + \hat{y}\,C_{yy} - \hat{z}\,\sin\theta_{MN}\sin\phi_{MN}\}\mathrm{ELPAT} \quad .
\end{aligned} \qquad (4)$$

The fields $E^a_{XMN} = E^a(1,M,N)$ and $E^a_{YMN} = E^a(2,M,N)$ are the X and Y components of the aperture field sampled at the points $(X_M, Y_N)$ on the principal grid. The basic pattern $F_{RS}$ of each rectangular subaperture is given by

$$F_{RS} = D_X D_Y F_{XN} F_{YN} \tag{5}$$

where $F_{XN}$ and $F_{YN}$ are the horizontal and vertical element patterns of each rectangular subaperture. The typical element patterns for a basic subaperture with full triangular distribution are given by

$$F_{XN} = \left( \frac{\sin \frac{\phi_x}{2}}{\frac{\phi_x}{2}} \right)^2 \tag{6}$$

$$F_{YN} = \left( \frac{\sin \frac{\phi_y}{2}}{\frac{\phi_y}{2}} \right)^2 \tag{7}$$

where

$$\phi_x = k\, D_X \sin\theta_{MN} \cos\phi_{MN} \tag{8}$$

and

$$\phi_y = k\, D_Y \sin\theta_{MN} \sin\phi_{MN} \tag{9}$$

The angles $\theta_{MN}$ and $\phi_{MN}$ are the polar coordinate angles to the near field point $(X,Y,Z)$ as referred to the aperture point $(X_M,Y_N)$. The distance S in Equation (2) is given by

$$S = \sqrt{(X-X_M)^2+(Y-Y_N)^2+Z^2} \quad . \tag{10}$$

The summations over N in Equation (2) are performed over the vertical grid lines; a typical vertical grid line is shown in Fig. 1. The y-integrations given by $Y_{SUM}(M)$ are calculated in a similar way as that for the far field (see Section 4) as expressed by

$$Y_{SUM}(M) = (y_{SL}+y_{SM}+y_{SU})D_Y \tag{11}$$

for each rectangular component of the near field. However, the vector element patterns in Equations (3) and (4) for the equivalent aperture currents and the element pattern functions $F_{RS}$ in Equation (5) for the rectangular subaperture must be included for the near field. For subapertures near the rim, the element patterns $F_{XN}$ and $F_{YN}$ in Equation (5) are expressed by the pattern of a half triangular distribution (see Section 6).

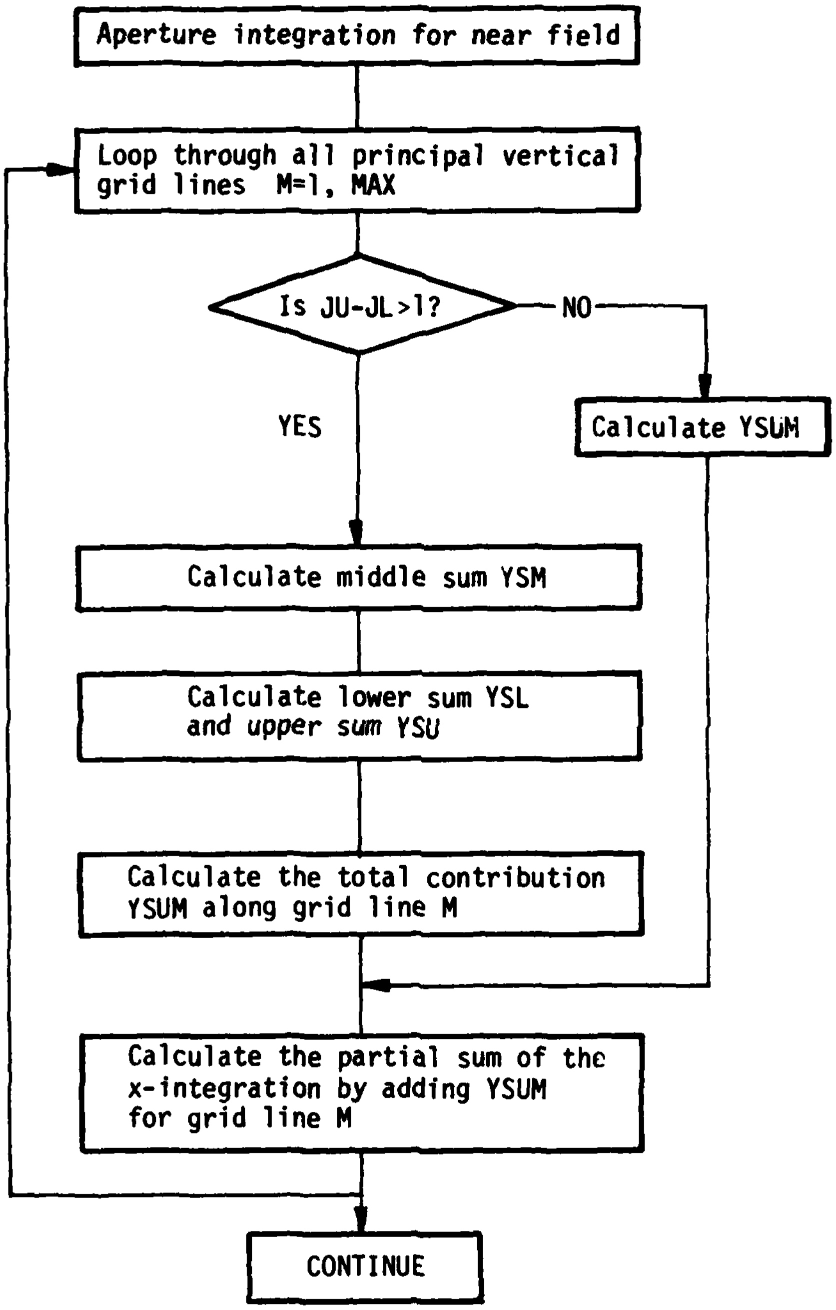
The summation over M in Equation (2), i.e., the x-integration part, is just a simple sum of $Y_{SUM}$'s as

$$\mathrm{SUM} = D_X \sum_{M=1}^{MAX} Y_{SUM}(M) \tag{12}$$

for each rectangular component of the near field. Then the near field at point (x,y,z) is obtained by

$$\bar{E} = \frac{j}{\lambda} \left( \mathrm{SUM}_x \hat{x} + \mathrm{SUM}_y \hat{y} + \mathrm{SUM}_z \hat{z} \right) \quad . \tag{13}$$

FLOW DIAGRAM



KEY VARIABLES

| | | |
|---|---|---|
| CXX | | X component of a X-oriented Huygen's source |
| CXY | | X component of a Y-oriented Huygen's source or Y component of an X-oriented Huygen's source |
| CYY | | Y component of a Y-oriented Huygen's source |
| DPX | $(\phi_x)$ | Horizontal phase argument of a basic subaperture |
| DPXL | | Horizontal phase argument of a subaperture at the left edge |
| DPXR | | Horizontal phase argument of a subaperture at the right edge |
| DPY | $(\phi_y)$ | Vertical phase argument of a basic subaperture |
| DPYL | | Vertical phase argument of a subaperture at the lower edge |
| DPYU | | Vertical phase argument of a subaperture at the upper edge |
| EA(1,M,N) | $(E^a_x)$ | X component of the aperture field at grid point (XM,YN) |
| EA(2,M,N) | $(E^a_y)$ | Y component of the aperture field at grid point (XM,YN) |
| EAL | | Interpolated aperture field at the lower rim point along grid line M |
| EAU | | Interpolated aperture field at the upper rim point along grid line M |
| EDX | | X component of the computed near field |
| EDY | | Y component of the computed near field |
| EDZ | | Z component of the computer near field |
| ELPAT | | Element pattern function for equivalent aperture current |
| EXPI | | Phase term for the leftmost grid point inside the aperture |
| EXPL | | Phase term for the leftmost rim point |

| | | |
|---|---|---|
| EXPM | | Phase term for the rightmost grid point inside the aperture |
| EXPR | | Phase term for the rightmost rim point |
| FFXN | ($F_{XN}$) | Horizontal pattern function for a rectangular subaperture |
| FFY | | Vertical pattern function for a basic rectangular subaperture |
| FHYM | | Vertical pattern function for a basic rectangular subaperture with a half triangular distribution (negative argument) |
| FHYP | | Vertical pattern function for a basic rectangular subaperture with a half triangular distribution (positive argument) |
| FYM | | Vertical pattern function for a rectangular subaperture at the edge with a half triangular distribution (negative argument) |
| FYP | | Vertical pattern function for a rectangular subaperture at the edge with a half triangular distribution (positive argument) |
| GRIDX | ($D_X$) | Horizontal grid size in the principal grid system |
| GRIDY | ($D_Y$) | Vertical grid size in the principal grid system |
| JC | | Horizontal grid line index of the origin of the reflector coordinate system |
| JLO | ($J_L$) | Index for the horizontal grid line inside the projected aperture closest to the lower intersection point on the grid line M |
| JUO | ($J_U$) | Index for the horizontal grid line inside the projected aperture closest to the upper intersection point on the grid line M |
| MAX | ($M_{MAX}$) | Maximum number of vertical grid lines |
| MC | | Vertical grid line index M of the origin of the reflector coordinate system |
| PHIRN | ($\phi_{MN}$) | PHI coordinate angle of the near field point XN as referred to the aperture point (XM,YN) |

| | | |
|---|---|---|
| QDX | | Normalized distance from the integration point to the vertical grid line M |
| QXL | | Normalized distance from the leftmost rim point to the first vertical grid line inside the aperture |
| QXR | | Normalized distance from the rightmost rim point to the last vertical grid line inside the aperture |
| QYL | | Normalized distance from the lower rim to the grid line JL |
| QYU | | Normalized distance from the upper rim to the grid line JU |
| S | | Distance from a grid point not adjacent to the rim to the near field point XN |
| S1 | | Distance from the intersection point along the grid line M to the near field point XN |
| SUMX | | X component of the x-integration sum |
| SUMY | | Y component of the x-integration sum |
| SUMZ | | Z component of the x-integration sum |
| THERN | $(\theta_{MN})$ | Theta coordinate angle of the near field point XN as referred to the aperture point (XM.YN) |
| XN | | Coordinates of near field point |
| XP | | X-coordinate of the near field point XN as referred to the aperture point (XM,YN) |
| YEXP | | Phase term for near field integration for the grid point (XM,YN) |
| YLR | $(y_{LR})$ | Y-coordinate of the intersection of the grid line M and the lower rim |
| YP | | Y-coordinate of the near field point XN as referred to the aperture point (XM,YN) |
| YSLX | | X component of the lower sum of the Y-integration |

|  |  |  |
|---|---|---|
| YSLY |  | Y component of the lower sum of the Y-integration |
| YSLZ |  | Z component of the lower sum of the Y-integration |
| YSMX |  | X component of the middle sum of the Y-integration |
| YSMY |  | Y component of the middle sum of the Y-integration |
| YSMZ |  | Z component of the middle sum of the Y-integration |
| YSUX |  | X component of the upper sum of the Y-integration |
| YSUY |  | Y component of the upper sum of the Y-integration |
| YSUZ |  | Z component of the upper sum of the Y-integration |
| YSUM(1,M) |  | X component of the total sum of the Y-integration for grid line M |
| YSUM(2,M) |  | Y component of the total sum of the Y-integration for grid line M |
| YSUM(3,M) |  | Z component of the total sum of the Y-integration for grid line M |
| YUR | ($y_{UR}$) | The Y-coordinate of the intersection of the grid line M and the upper rim |

## CODE LISTING

```
C
C     ***Y INTEGRATION FOR NEAR FIELD ***
C
      SUMX=(0.,0.)
      SUMY=(0.,0.)
      SUMZ=(0.,0.)
      DO 220 M=1,MAX
      NPRI=0
      IF (LDEBUG.AND.(M.EQ.MNO)) NPRI=1
      XP=XN(1)-(M-MC)*GRDX
      IF (M.EQ.1) XP=XN(1)-XMIN
      IF (M.EQ.MAX) XP=XN(1)-XMAX
      RP=XP*XP+XN(3)*XN(3)
      JLO=JL(M)
      JUO=JU(M)
      IF (JUO-JLO.GT.1) GO TO 206
      YP=XN(2)-YLR(M)
      PHIRN=BTAN2(YP,XP)
      SINPN=SIN(PHIRN)
      COSPN=COS(PHIRN)
      THERN=BTAN2(SQRT(XP*XP+YP*YP),XN(3))
      SINTN=SIN(THERN)
      COSTN=COS(THERN)
      S=SQRT(YP*YP+RP)
      EXPL=CEXP(-CJ*TPI*S)
      YSUM(1,M)=(YUR(M)-YLR(M))*EA(1,M,JUO)*EXPL/S
      YSUM(2,M)=(YUR(M)-YLR(M))*EA(2,M,JUO)*EXPL/S
      ELPAT=COS(THERN/2.)
      DUMY=COSTN-1.
      CXX=1.+COSPN*COSPN*DUMY
      CXY=SINPN*COSPN*DUMY
      CYY=1.+SINPN*SINPN*DUMY
      YSUM(3,M)=-SINTN*(YSUM(1,M)*COSPN+YSUM(2,M)*SINPN)*ELPAT
      TMX=YSUM(1,M)
      YSUM(1,M)=(CXX*TMX+CXY*YSUM(2,M))*ELPAT
      YSUM(2,M)=(CXY*TMX+CYY*YSUM(2,M))*ELPAT
      IF (LWYSUM) WRITE (6,166) M,YSUM(1,M),YSUM(2,M),YSUM(3,M)
      GO TO 220
C
C         * CALCULATE YSM *
C
  206 JF=JUO-1
      JI=JLO+1
      KM=JF-JI+1
      YSMX=(0.,0.)
      YSMY=(0.,0.)
      YSMZ=(0.,0.)
      DO 208 KJ=1,KM
      J=KJ+JI-1
```

```
      NJ=J+1
      YP=XN(2)-(J-JC)*GRDY
      PHIRN=BTAN2(YP,XP)
      SINPN=SIN(PHIRN)
      COSPN=COS(PHIRN)
      THERN=BTAN2(SQRT(XP*XP+YP*YP),XN(3))
      SINTN=SIN(THERN)
      COSTN=COS(THERN)
      DPX=TPI*COSPN*SINTN*GRDX
      FFXN=FF(DPX)+CJ*0.
      DPXL=DPX*QXL
      DPXR=DPX*QXR
      IF (M.EQ.1) FFXN=QXL*FH(DPXL)
      IF (M.EQ.2) FFXN=FH(DPX)+QXL*FH(-DPXL)
      IF (M.EQ.MIX) FFXN=FH(-DPX)+QXR*FH(DPXR)
      IF (M.EQ.MAX) FFXN=QXR*FH(-DPXR)
      DPY=TPI*SINPN*SINTN*GRDY
      FFY=FF(DPY)
      S=SQRT(YP*YP+RP)
      YEXP=CEXP(-CJ*TPI*S)*FFXN*FFY
      ELPAT=COS(THERN/2.)
      DUMY=COSTN-1.
      CXX=1.+COSPN*COSPN*DUMY
      CXY=SINPN*COSPN*DUMY
      CYY=1.+SINPN*SINPN*DUMY
      YSMX=YSMX+(CXX*EA(1,M,NJ)+CXY*EA(2,M,NJ))*YEXP*ELPAT/S
      YSMY=YSMY+(CXY*EA(1,M,NJ)+CYY*EA(2,M,NJ))*YEXP*ELPAT/S
      YSMZ=YSMZ-SINTN*(EA(1,M,NJ)*COSPN+EA(2,M,NJ)*SINPN)*YEXP*ELPAT/S
  208 CONTINUE
C
C         * CALCULATE YSL *
C
      YP=XN(2)-YLR(M)
      S1=SQRT(YP*YP+RP)
      EXPL=CEXP(-CJ*TPI*S1)
      YP=XN(2)-(JLO-JC)*GRDY
      S2=SQRT(YP*YP+RP)
      EXPI=CEXP(-CJ*TPI*S2)
      PHIRN=BTAN2(YP,XP)
      SINPN=SIN(PHIRN)
      COSPN=COS(PHIRN)
      THERN=BTAN2(SQRT(XP*XP+YP*YP),XN(3))
      SINTN=SIN(THERN)
      COSTN=COS(THERN)
      DPX=TPI*COSPN*SINTN*GRDX
      FFXN=FF(DPX)+CJ*0.
      DPXL=DPX*QXL
      DPXR=DPX*QXR
      IF (M.EQ.1) FFXN=QXL*FH(DPXL)
      IF (M.EQ.2) FFXN=FH(DPX)+QXL*FH(-DPXL)
      IF (M.EQ.MIX) FFXN=FH(-DPX)+QXR*FH(DPXR)
```

```
      IF (M.EQ.MAX) FFXN=QXR*FH(-DPXR)
      DPY=TPI*SINPN*SINTN*GRDY
      FHYP=FH(DPY)
      DPYL=DPY*QYL(M)
      FYM=FH(-DPYL)
      FYP=FH(DPYL)
      EXPL=EXPL*FYP*QYL(M)
      EXPI=EXPI*(FYM*QYL(M)+FHYP)
      EAL=EA(1,M,JLO+1)*(QYL(M)+1.)-EA(1,M,JLO+2)*QYL(M)
      IF (NPRI.EQ.1) WRITE (6,-) EA(1,M,JLO+1),EA(1,M,JLO+2),EAL
      YSLX=FFXN*(EAL*EXPL/S1+EA(1,M,JLO+1)*EXPI/S2)
      EAL=EA(2,M,JLO+1)*(QYL(M)+1.)-EA(2,M,JLO+2)*QYL(M)
      IF (NPRI.EQ.1) WRITE (6,909) JLO,JI,EA(2,M,JI),EA(2,M,JLO+2)
     2,EAL
      YSLY=FFXN*(EAL*EXPL/S1+EA(2,M,JLO+1)*EXPI/S2)
      ELPAT=COS(THERN/2.)
      DUMY=COSTN-1.
      CXX=1.+COSPN*COSPN*DUMY
      CXY=SINPN*COSPN*DUMY
      CYY=1.+SINPN*SINPN*DUMY
      YSLZ=-SINTN*(YSLX*COSPN+YSLY*SINPN)*ELPAT
      TMX=YSLX
      YSLX=(CXX*YSLX+CXY*YSLY)*ELPAT
      YSLY=(CXY*TMX+CYY*YSLY)*ELPAT
C
C          * CALCULATE YSU *
C
      YP=XN(2)-YUR(M)
      S1=SQRT(YP*YP+RP)
      EXPR=CEXP(-CJ*TPI*S1)
      YP=XN(2)-(JUO-JC)*GRDY
      S2=SQRT(YP*YP+RP)
      EXPM=CEXP(-CJ*TPI*S2)
      PHIRN=BTAN2(YP,XP)
      SINPN=SIN(PHIRN)
      COSPN=COS(PHIRN)
      THERN=BTAN2(SQRT(XP*XP+YP*YP),XN(3))
      SINTN=SIN(THERN)
      COSTN=COS(THERN)
      DPX=TPI*COSPN*SINTN*GRDX
      FFXN=FF(DPX)+CJ*0.
      DPXL=DPX*QXL
      DPXR=DPX*QXR
      IF (M.EQ.1) FFXN=QXL*FH(DPXL)
      IF (M.EQ.2) FFXN=FH(DPX)+QXL*FH(-DPXL)
      IF (M.EQ.MIX) FFXN=FH(-DPX)+QXR*FH(DPXR)
      IF (M.EQ.MAX) FFXN=QXR*FH(-DPXR)
      DPY=TPI*SINPN*SINTN*GRDY
      FHYM=FH(-DPY)
      DPYU=DPY*QYU(M)
      FYM=FH(-DPYU)
```

```
      FYP=FH(DPYU)
      EXPM=EXPM*(FYP*QYU(M)+FHYM)
      EXPR=EXPR*FYM*QYU(M)
      EAU=EA(1,M,JUO+1)*(QYU(M)+1.)-EA(1,M,JUO)*QYU(M)
      IF (NPRI.EQ.1) WRITE (6,-) EA(1,M,JUO+1),EA(1,M,JUO),EAU
      YSUX=FFXN*(EAU*EXPR/S1+EA(1,M,JUO+1)*EXPM/S2)
      EAU=EA(2,M,JUO+1)*(QYU(M)+1.)-EA(2,M,JUO)*QYU(M)
      IF (NPRI.EQ.1) WRITE (6,909) JUO,JF,EA(2,M,JUO+1),EA(2,M,JUO)
     2,EAU
      IF (NPRI.EQ.1) WRITE (6,-) QYL(M),QYU(M),YLR(M),YUR(M)
      YSUY=FFXN*(EAU*EXPR/S1+EA(2,M,JUO+1)*EXPM/S2)
      ELPAT=COS(THERN/2.)
      DUMY=COSTN-1.
      CXX=1.+COSPN*COSPN*DUMY
      CXY=SINPN*COSPN*DUMY
      CYY=1.+SINPN*SINPN*DUMY
      YSUZ=-SINTN*(YSUX*COSPN+YSUY*SINPN)*ELPAT
      TMX=YSUX
      YSUX=(CXX*YSUX+CXY*YSUY)*ELPAT
      YSUY=(CXY*TMX+CYY*YSUY)*ELPAT
      YSUM(1,M)=(YSLX+YSMX+YSUX)*GRDY
      YSUM(2,M)=(YSLY+YSMY+YSUY)*GRDY
      YSUM(3,M)=(YSLZ+YSMZ+YSUZ)*GRDY
      IF (LWYSUM) WRITE (6,166) M,YSLX,YSMX,YSUX,YSUM(1,M)
      IF (LWYSUM) WRITE (6,166) M,YSLY,YSMY,YSUY,YSUM(2,M)
      IF (LWYSUM) WRITE (6,166) M,YSLZ,YSMZ,YSUZ,YSUM(3,M)
C
C     *** X INTEGRATION FOR NEAR FIELD ***
C
      SUMX=SUMX+YSUM(1,M)*GRDX
      SUMY=SUMY+YSUM(2,M)*GRDX
      SUMZ=SUMZ+YSUM(3,M)*GRDX
      IF (LWYSUM) WRITE (6,166) M,SUMX,SUMY,SUMZ
  220 CONTINUE
      EDX=CJ*SUMX
      EDY=CJ*SUMY
      EDZ=CJ*SUMZ
      IF (LTEST) WRITE (6,195) EDX,EDY,EDZ
      IF (.NOT.LFEED) GO TO 224
```

## SECTION 7. X-INTEGRATION FOR FAR FIELD

### PURPOSE

To numerically integrate the y-integration sums along the horizontal grid line and obtain the final far field pattern.

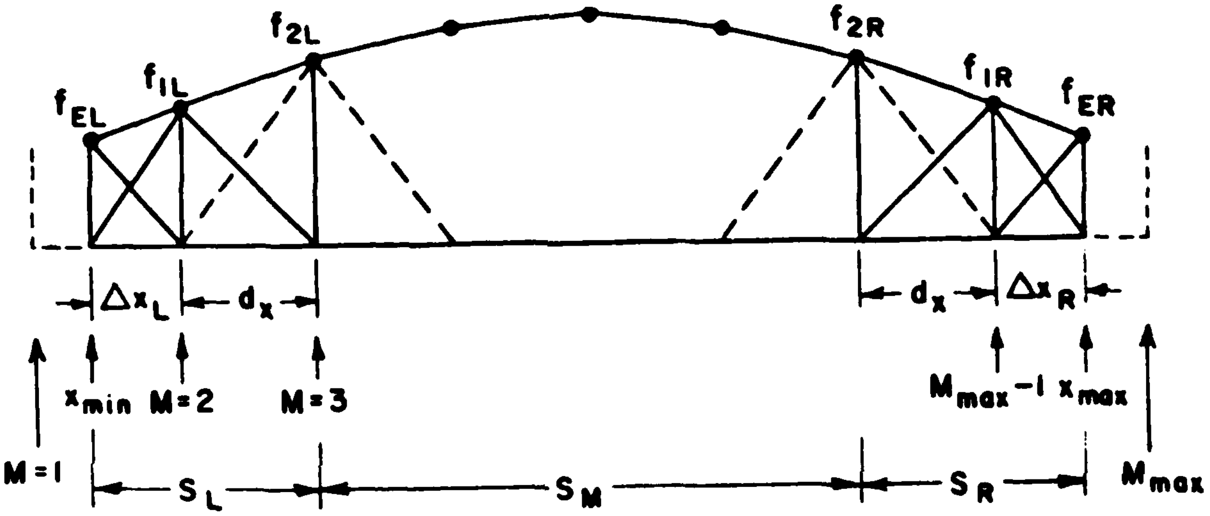


Figure 1. The x-integration parts

### METHOD

Using the result of y-integration, the scalar radiation integral for the far field pattern reduces to

$$E = \frac{jk}{2\pi} \int_{x_{min}}^{x_{max}} Y_{SUM}\, e^{jkx\sin\theta\cos\phi}\, dx \quad .$$

The x-integration is divided into three parts in a similar way as was the y-integration. The middle part consists of the basic subapertures (subapertures with full grid size) with full triangular distribution as shown by the dashed lines in Fig. 1. The expression for the distribution of a basic subaperture is given by

$$f_F(x) = 1 - \frac{|x-x_0|}{d_x}$$

for $|x-x_0|<d_x$, as shown in Fig. 2a. The resulting far field pattern for the basic subaperture, i.e., element pattern, is given by

$$F_{SF}(\theta,\phi) = d_x \cos\phi \; F_F(\phi_x)$$

where

$$F_F(\phi_x) = \left[\frac{\sin\left(\frac{\phi_x}{2}\right)}{\left(\frac{\phi_x}{2}\right)}\right]^2$$

and the argument

$$\phi_x = k \, d_x \sin\theta\cos\phi \quad .$$

Thus the result for the middle part of the x-integration is simply the sum of the product of the y-integration sum and the phase exponential for each subaperture multiplied by its element pattern as given by

$$S_M = d_x \, F_F(\phi_x) \sum_{M=3}^{M_{max}-2} Y_{sum}(M) \, e^{j(I-I_c)\phi_x}$$

where $I=M-1$, $I_c$ is the I index for the origin, and $M_{max}$ is the maximum number of rotated grid lines.

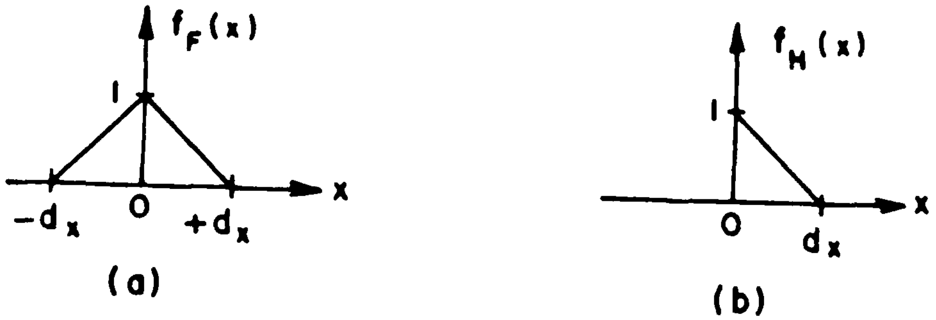


Figure 2. Triangular subaperture distributions (full and half).

The contributions from the left and right parts of the x-integration are treated in the same way as for the lower and upper parts of the y-integration, except that the element patterns are calculated separately for each of the three subapertures near each of the left and right edges of the reflector rim. Each of these subapertures has a half-triangular distribution as shown in Fig. 2b. The element patterns for these subapertures can be represented by

$$F_H(\phi_x) = \frac{1-e^{j\phi_x}}{(\phi_x)^2} + \frac{j}{\phi_x} \quad .$$

The contribution $S_L$ from the left part is given in terms of the y-integration sums $f_{1L}$ and $f_{EL}$ as shown in Fig. 1. The edge value $f_{EL}$ is obtained by extrapolation using Equation (15) with $f_{1L}=Y_{sum}(2)$ and $f_{2L}=Y_{sum}(3)$, thus

$$f_{EL} = Y_{sum}(x_{min}) = Y_{sum}(2)\left(1 + \frac{\Delta x_L}{d_x}\right) - Y_{sum}(3)\,\frac{\Delta x_L}{d_x}$$

where $\Delta x_L$ is the distance between $x_{min}$ and the M=2 grid line.

Consequently, the contributions from the three subapertures of the left part are given by

$$S_L = f_{EL}\, e^{jkx_{min}\sin\theta\cos\phi}\, F_H(+\phi_{xL})\Delta x_L$$

$$+ f_{1L}\, e^{j(1-I_c)\phi_x}\,[F_H(-\phi_{xL})\Delta x_L + F_H(+\phi_x)d_x]$$

where

$f_{1L} = Y_{SUM}(2)$ and

$\phi_{xL} = k\,\Delta x_L\,\sin\theta\cos\phi$

is the argument for the element patterns $F_H(\pm\phi_{xL})$ of the two subapertures with width $\Delta x_L$.

Similarly, the value $f_{ER}$ for the y-integration at the right edge of the reflector rim is given by

$$f_{ER} = Y_{sum}(x_{max}) = Y_{sum}(M_{max}-1)\left(1 + \frac{\Delta x_R}{d_x}\right) - Y_{sum}(M_{max}-2)\,\frac{\Delta x_R}{d_x} \quad .$$

The contributions of the three subapertures of the right part can be obtained as

$$S_R = f_{ER}\, e^{jkx_{max}\sin\theta\cos\phi}\, F_H(-\phi_{xR})\Delta x_R$$

$$+ f_{1R}\, e^{j(I_{max}-I_c)\phi_x}\, [F_H(+\phi_{xR})\Delta x_R + F_H(-\phi_x)d_x]$$

where

$$f_{1R} = Y_{sum}(M_{max}-1) \text{ and}$$

$$\phi_{XR} = k\ \Delta x_R\ \sin\theta\cos\phi \quad .$$

Finally, the resulting far field pattern function as calculated by the rotating grid method is obtained by adding up the partial sums.

$$SUM = (S_L + S_M + S_R)\ \cos\left(\frac{\theta}{2}\right)$$

where $\cos(\theta/2)$ is the element pattern factor of the equivalent aperture currents.

Since the aperture field has both x and y components, the far field pattern associated with these two orthogonal aperture field components are calculated by the above equation and represented by SUMx and SUMy respectively. Each element of the aperture is assumed to radiate the same polarization as a Huygen's source, thus the spherical components of the far field pattern are given by

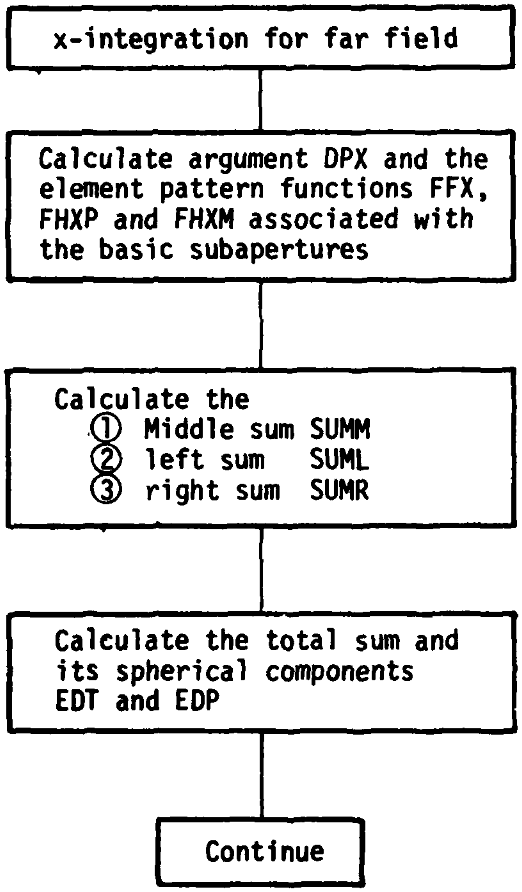
$$E_\theta^d = j(\cos\phi\cdot SUM_x+\sin\phi\cdot SUM_y)|\cos\phi|$$

and

$$E_\phi^d = -j(\sin\phi\cdot SUM_x-\cos\phi\cdot SUM_y)|\cos\phi|$$

where $|\cos\phi|$ is the correction factor for the enlarged grid size due to grid rotation.

FLOW DIAGRAM



KEY VARIABLES

| | | |
|---|---|---|
| ACOSP | $\|\cos\phi\|$ | Absolute value of $\cos\phi$ |
| DPX | $(\phi_x)$ | Phase argument of a basic subaperture |
| DPXL | $(\phi_{xL})$ | Phase argument of a subaperture at the left edge |
| DPXR | $(\phi_{xR})$ | Phase argument of a subaperture at the right edge |
| EDP | $(E_\phi^d)$ | PHI component of the radiation field |
| EDT | $(E_\theta^d)$ | THETA components of the radiation field |
| ELPAT | | Element pattern function for equivalent aperture current |
| EXPI | | Phase term for the leftmost grid point inside the aperture |
| EXPL | | Phase term for the leftmost rim point |
| EXPM | | Phase term for the rightmost grid point inside the aperture |
| EXPR | | Phase term for the rightmost rim point |
| FFX | $(F_F(\phi_x))$ | Horizontal pattern function for a basic subaperture with a full triangular distribution |
| FHXM | | Horizontal pattern function for a basic subaperture with a half triangular distribution (negative argument) |
| FHXP | | Horizontal pattern function for a basic subaperture with a half triangular distribution (positive argument) |
| FXM | | Horizontal pattern function for a subaperture at the edge with a half triangular distribution (negative argument) |
| FYP | | Vertical pattern function for a subaperture at the edge with a half triangular distribution (positive argument) |
| GRDX | | Horizontal grid size |

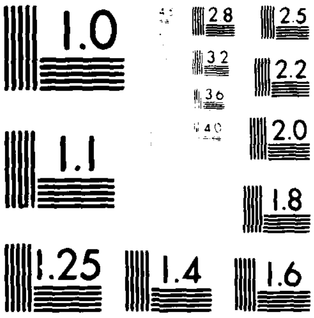
| | | |
|---|---|---|
| MAX | $M_{max}$ | Maximum number of rotated grid lines |
| PG | | Variable used for phase argument DPX (calculated in subroutine GRID) |
| PHSX | | Phase path of an integration grid point on the aperture |
| PHXL | | Phase path of the |
| PHXR | | Phase path of the rightmost rim point |
| QXL | | Normalized distance from the leftmost rim point to the first vertical grid line inside the aperture |
| QXR | | Normalized distance from the rightmost rim point to the last vertical grid line inside the aperture |
| RFCT | $\left(\frac{e^{-jkR}}{R}\right)$ | Range factor (used if LRANG is true) |
| SUMLX | | X-component of the left x-integration sum |
| SUMLY | | Y-component of the left x-integration sum |
| SUMMX | | X-component of the middle x-integration sum |
| SUMMY | | Y-component of the middle x-integration sum |
| SUMRX | | X-component of the right x-integration sum |
| SUMRY | | Y-component of the right x-integration sum |
| SUMX | | X-component of the total x-integration sum |
| SUMY | | Y-component of the total x-integration sum |
| XEXP | | Phase term for an integration grid point |
| YML | | Interpolated YSUM value at the left edge |
| YMR | | Interpolated YSUM value at the right edge |
| YSUM(1,M) | | X-component of the total sum of the Y-integration for grid line M |
| YSUM(2,M) | | Y-component of the total sum of the Y-integration for grid line M |

```
C
C     ***** X INTEGRATION FOR FAR FIELD *****
C
  227 DPX=PG*SINT
      FFX=FF(DPX)
      FHXP=FH(DPX)
      FHXM=FH(-DPX)
      IF (LTEST) WRITE (6,228) DPX,FFX,FHXP,FHXM
  228 FORMAT(2H D,T10,'DPX =',F7.4,5X,5F10.5,T79,1HD)
C
C           * MIDDLE SUM *
C
      SUMMX=(0.,0.)
      SUMMY=(0.,0.)
      DO 230 M=1,MAX
      IF (M.LE.2.OR.M.GE.MIX) GO TO 230
      I=M-1
      IX=I-IC
      PHSX=IX*DPX
      XEXP=CEXP(CJ*PHSX)
      SUMMX=SUMMX+YSUM(1,M)*XEXP*FFX*GRDX
      SUMMY=SUMMY+YSUM(2,M)*XEXP*FFX*GRDX
      IF (LWYSUM) WRITE (6,166) M,SUMMX,SUMMY
  230 CONTINUE
C
C           * LEFT SUM *
C
      PHXL=(XMIN/GRDX)*DPX
      DPXL=QXL*DPX
      FXP=FH(DPXL)
      FXM=FH(-DPXL)
      EXPL=CEXP(CJ*PHXL)
      EXPI=CEXP(CJ*(1-IC)*DPX)
      YML=YSUM(1,2)*(QXL+1.)-YSUM(1,3)*QXL
      SUMLX=YML*EXPL*FXP*DXL+YSUM(1,2)*EXPI*(FXM*DXL+FHXP*GRDX)
      YML=YSUM(2,2)*(QXL+1.)-YSUM(2,3)*QXL
      SUMLY=YML*EXPL*FXP*DXL+YSUM(2,2)*EXPI*(FXM*DXL+FHXP*GRDX)
C
C           * RIGHT SUM *
C
      PHXR=(XMAX/GRDX)*DPX
      DPXR=QXR*DPX
      FXP=FH(DPXR)
      FXM=FH(-DPXR)
      EXPR=CEXP(CJ*PHXR)
      EXPM=CEXP(CJ*(IMAX-IC)*DPX)
      YMR=YSUM(1,MIX)*(QXR+1.)-YSUM(1,IMAX)*QXR
      SUMRX=YMR*EXPR*FXM*DXR+YSUM(1,MIX)*EXPM*(FXP*DXR+FHXM*GRDX)
      YMR=YSUM(2,MIX)*(QXR+1.)-YSUM(2,IMAX)*QXR
```

```
      SUMRY=YAR*EXPR*FXM*DXR+YSUM(2,MIX)*EXPM*(FXP*DXR+FHXM*GRDX)
      IF (LWYSUM) WRITE (6,166) N,SUMLY,SUMRY
      ELPAT=COS(THER/2.)
      SUMX=ELPAT*(SUMLX+SUMMX+SUMRX)
      SUMY=ELPAT*(SUMLY+SUMMY+SUMRY)
      IF (LTEST) WRITE (6,195) SUMX,SUMY
      EDT=CJ*(COSP*SUMX+SINP*SUMY)*RFCT*ACOSP
      EDP=CJ*(-SINP*SUMX+COSP*SUMY)*RFCT*ACOSP
      IF (LTEST) WRITE (6,195) COSP,SINP,ACOSP
      IF (LTEST) WRITE (6,235) N,EDT,EDP
  235 FORMAT(2H T,I5,4E11.3,T79,1HT)
      IF (.NOT.LFEED) GO TO 242
```

AD-A097 415 OHIO STATE UNIV COLUMBUS ELECTROSCIENCE LAB F/G 20/4
NUMERICAL ELECTROMAGNETIC CODE (NEC)-REFLECTOR ANTENNA CODE: PA--ETC(U)
SEP 79 S H LEE, R C RUDDUCK N00123-76-C-1371
UNCLASSIFIED ESL-784508-16 NL

2 of 3



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

## B. SUBROUTINES

### SUBROUTINE BABS

### PURPOSE

This function computes the absolute value of a complex argument. It is similar to CABS, except it avoids run time errors when the real part and imaginary part of the argument are zero.

### METHOD

The system function CABS is used unless the absolute value of the real part and the imaginary part of the argument are close to zero, in which case a very small value is returned.

### KEY VARIABLES

| | |
|---|---|
| X | Absolute value of the real part of Z |
| Y | Absolute value of the imaginary part of z |
| Z | The complex argument |

### CODE LISTING

```
      FUNCTION BABS(Z)
C!!!
C!!!  THIS ROUTINE IS USED TO GIVE COMPLEX ABSOLUTE VALUES. IT IS
C!!!  USED RATHER STANDARD ROUTINES TO AVOID EXECUTION ERRORS.
C!!!
      COMPLEX Z
      X=ABS(REAL(Z))
      Y=ABS(AIMAG(Z))
      IF(X.LT.1.E-20.AND.Y.LT.1.E-20) GO TO 10
      BABS=CABS(Z)
      RETURN
 10   BABS=1.E-20
      RETURN
      END
```

## SUBROUTINE BTAN2

### PURPOSE

This function computes the two argument arctangent function. It is similar to ATAN2, except it avoids run time errors when the second argument is zero.

### METHOD

The system function ATAN2(Y,X) is used to return the angle in radians, whose sine is Y and cosine is X unless the second argument or both of the arguments are zero. If the second argument is zero, either $\pi/2$ or $-\pi/2$ is returned depending on the sign of the first argument. If both arguments are zero, a zero value is returned.

### KEY VARIABLES

| | |
|---|---|
| X | Second argument, which is the cosine of the angle to be computed |
| Y | First argument, which is the sine of the angle to be computed |

### CODE LISTING

```
      FUNCTION BTAN2(Y,X)
C!!!
C!!!  THIS ROUTINE IS USED TO COMPUTE THE ARCTANGENT. IT IS SIMILAR
C!!!  TO ATAN2 EXCEPT IT AVOIDS THE RUN TIME ERRORS.
C!!!
      COMMON/PIS/PI,TPI,DPR
      IF(ABS(X).GT.1.E-20) GO TO 50
      IF(ABS(Y).GT.1.E-20) GO TO 10
      BTAN2=0.
      RETURN
   10 BTAN2=PI/2.
      IF(Y.LT.0.) BTAN2=-BTAN2
      RETURN
   50 BTAN2=ATAN2(Y,X)
      RETURN
      END
```

## SUBROUTINE DBPHS

### PURPOSE

To calculate the normalized power level in dB and the phase of a complex field value.

### METHOD

The power of a complex field value E expressed in dB is given by

$$DB = 20 \log_{10}|E| + REF$$

and the phase of E by

$$\phi = \tan^{-1}\left(\frac{Im(E)}{Re(E)}\right)$$

where Re(E) and Im(E) are the real and imaginary part of E.

For far field calculations without the range factor $e^{-jkR}/R$ (LRANG =false). The output of the code is expressed as antenna gain relative to isotropic. In this case, the value of REF is set equal to REFDB which is calculated in the main program using the information of relative power radiated by feed (see Section 1 of the main program).

For far field calculations including the range factor or for near field calculations the value of REF is set to zero. The value of REF is summarized in the table below.

TABLE FOR REF VALUE

| INPUT VARIABLE LRANG | FAR FIELD (LNF=false) | NEAR FIELD (LNF=true) |
|---|---|---|
| True | REF = 0 | REF = 0 |
| False | REF = REFDB | REF = 0 |

FLOW DIAGRAM

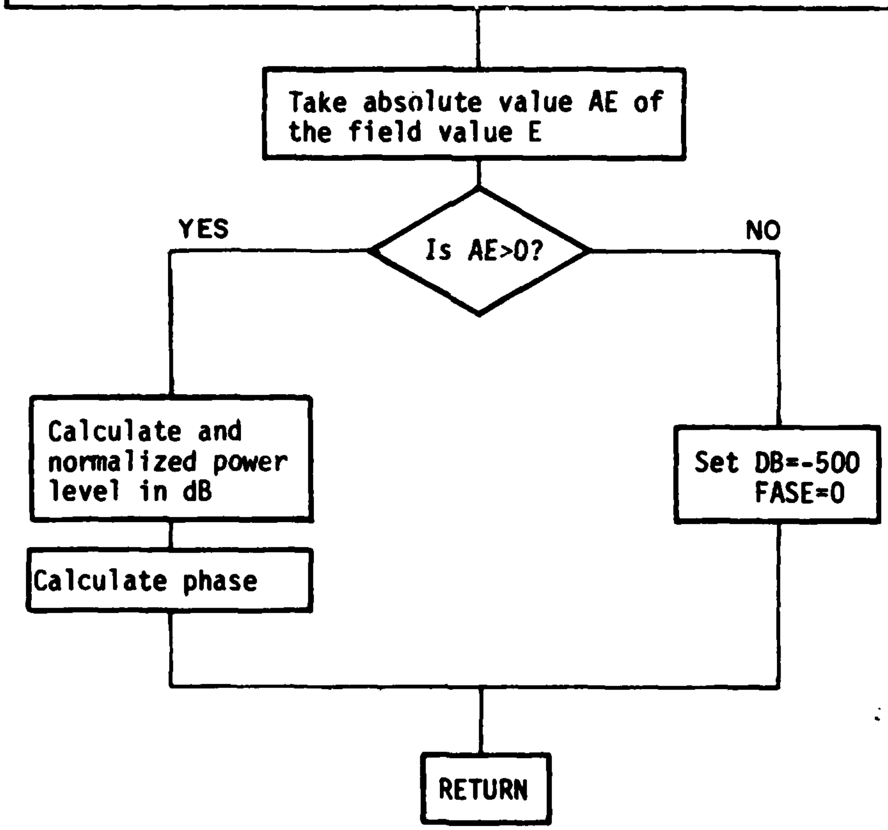
DBPHS(AE,E,REF)

INPUT VARIABLES

| | |
|---|---|
| E | Complex field value |
| REF | Reference power level |

OUTPUT VARIABLES

| | |
|---|---|
| AE | Absolute value of the field value |
| E | Normalized power level and phase of the field value |



KEY VARIABLES

| | |
|---|---|
| DB | Normalized power level of complex field value E |
| FASE (ϕ) | Phase of the complex field value E |

CODE LISTING

```
        SUBROUTINE DBPHS(AE,E,REF)
        COMPLEX E
        COMMON /PIS/PI,TPI,DPR
        AE=BABS(E)
        IF (AE.GT.0.) GO TO 10
        DB=-500.
        FASE=0.
        GO TO 20
   10   DB=20.*ALOG10(AE)+REF
        FASE=BTAN2(AIMAG(E),REAL(E))*DPR
   20   E=CMPLX(DB,FASE)
        RETURN
        END
```

## SUBROUTINE DCHP

## PURPOSE

To calculate the edge diffraction coefficients, the slope diffraction coefficients of a half plane, the corner diffraction coefficients and the slope corner diffraction coefficients for a plate.

## METHOD

Using the wedge diffraction coefficient formulation [5,6], the edge diffraction coefficients for a half plane can be expressed by

$$D_{s,h}(\beta,\beta_0) = DI^- \mp DI^+$$

where

$$DI^- = \frac{-e^{-j\frac{\pi}{4}}}{2\sqrt{2\pi k}\,\sin\beta_0}\;\frac{F[kLa(\beta^-)]}{\cos\frac{\beta^-}{2}}$$

$$DI^+ = \frac{-e^{-j\frac{\pi}{2}}}{2\sqrt{2\pi k}\,\sin\beta_0}\;\frac{F[kL(\beta^+)]}{\cos\frac{\beta^+}{2}}$$

$$\beta^{\mp} = \phi \mp \phi' \quad ,$$

$$a = 2\cos^2\left(\frac{\beta}{2}\right) \quad ,$$

L is the distance parameter,

$F(X) = 2j|\sqrt{X}|e^{jX}\int_{|\sqrt{X}|}^{\infty} e^{-j\tau^2}\,d\tau$ is the transition function,

$\beta_0$ is the diffracted cone angle and

$\phi$ and $\phi'$ are the diffraction angles for the diffracted field and incident field, respectively.

The slope diffraction coefficient for a half plane is given by

$$\frac{\partial D_{s,h}}{\partial \phi'} = DPI^{-} \pm DPI^{+}$$

where

$$DPI^{-} = j\sqrt{\frac{k}{2\pi}}\,\frac{e^{-j\frac{\pi}{4}}L}{\sin\beta_o}\,\sin\left(\frac{\beta^{-}}{2}\right)\,[1-F[kLa(\beta^{-})]]$$

and

$$DPI^{+} = j\sqrt{\frac{k}{2\pi}}\,\frac{e^{-j\frac{\pi}{4}}L}{\sin\beta_o}\,\sin\left(\frac{\beta^{+}}{2}\right)\,[1-F[kLa(\beta^{+})]] \quad .$$

The corner diffraction fields from a corner of a plate (see Fig. 1) can be represented by[7]

$$\left\{\begin{matrix} E^c_{\parallel} \\ E^c_{\perp} \end{matrix}\right\} = CORN \left\{\begin{matrix} C_s E^i_{\parallel} \\ C_h E^i_{\perp} \end{matrix}\right\}$$

where

$$CORN = -\,\frac{\sin\beta_c\, e^{-j\frac{\pi}{4}}}{2\pi(\cos\beta_{oc}+\cos\beta_c)}\; F|kL_c a(\beta_{oc}+\beta_c)|\; \sqrt{\frac{s'}{s_c}}\; e^{-jk(s_c-s')}\;\frac{e^{-jks}}{s}$$

$$C_{s,h} = DI^{-} \times AFC^{-} \mp DI^{+} \times AFC^{+}$$

and

$$AFC^{\mp} = \left| F\left[\frac{La(\beta^{\mp})}{kL_c a(\beta_{oc}+\beta_c)}\right]\right|$$ is the heuristic function .

Note that the angles $\beta_{oc}$, $\beta_c$ (see Fig. 1) and the corner distance parameter

$$L_c = \frac{s_c s}{s_c + s}$$

are calculated in the subroutine GTD.

In the code the diffracted fields from both corners ME and ME+1 of a rim segment ME are combined in the following way

$$\begin{Bmatrix} E_{\parallel}^{c} \\ E_{\perp}^{c} \end{Bmatrix} = - \begin{Bmatrix} B_s E_{\parallel}^{i} \\ B_h E_{\perp}^{i} \end{Bmatrix}$$

where the coefficients $B_s$ and $B_h$ are given by

$$B_{s,h} = DI^{-} \times CC^{-} \mp DI^{+} \times CC^{+}$$

and

$$CC^{\mp} = CORN_{ME} \times AFC_{ME}^{\mp} + CORN_{ME+1} \times AFC_{ME+1}^{\mp} \quad ,$$

Similarly, the coefficients for slope corner diffraction are given by

$$\frac{\partial B_{s,h}}{\partial \phi'} = \frac{\partial D_{s,h}}{\partial \phi'} \times (CORN_{ME} + CORN_{ME+1}) \quad .$$

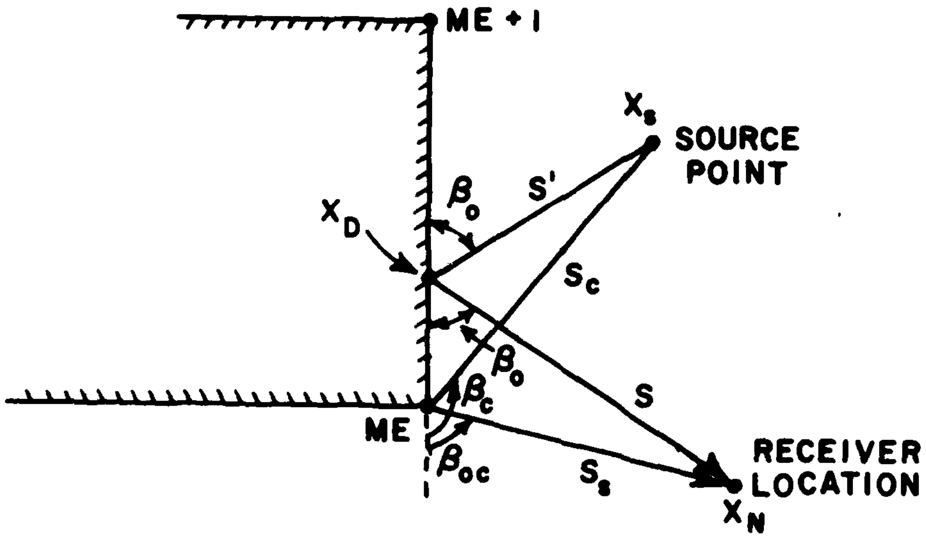


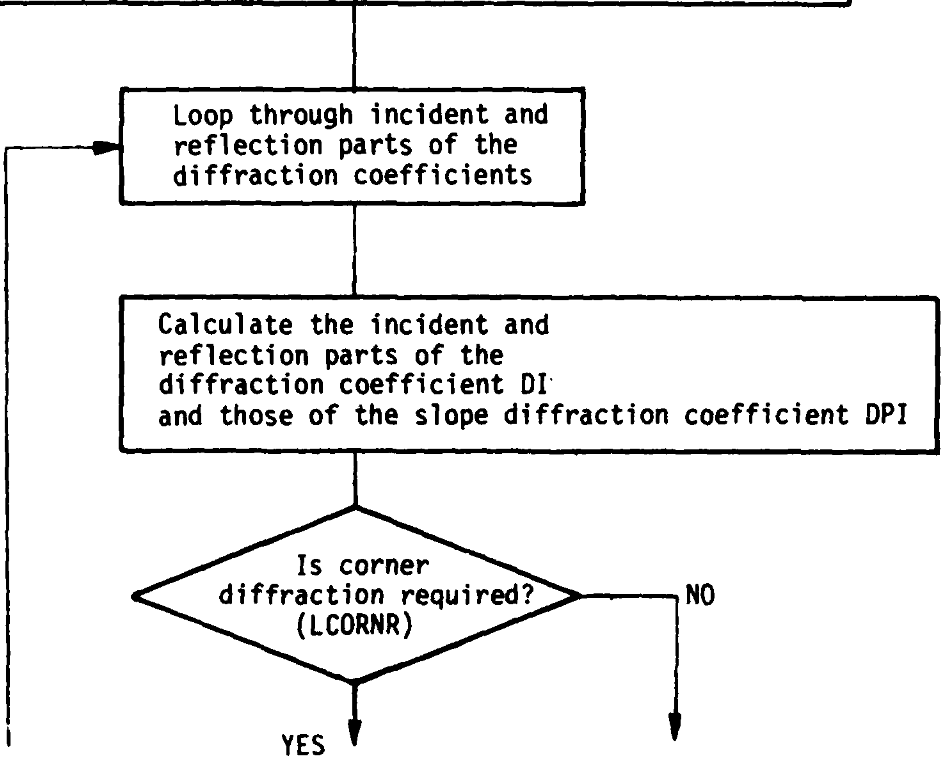
Figure 1. Geometry for corner diffraction problem.

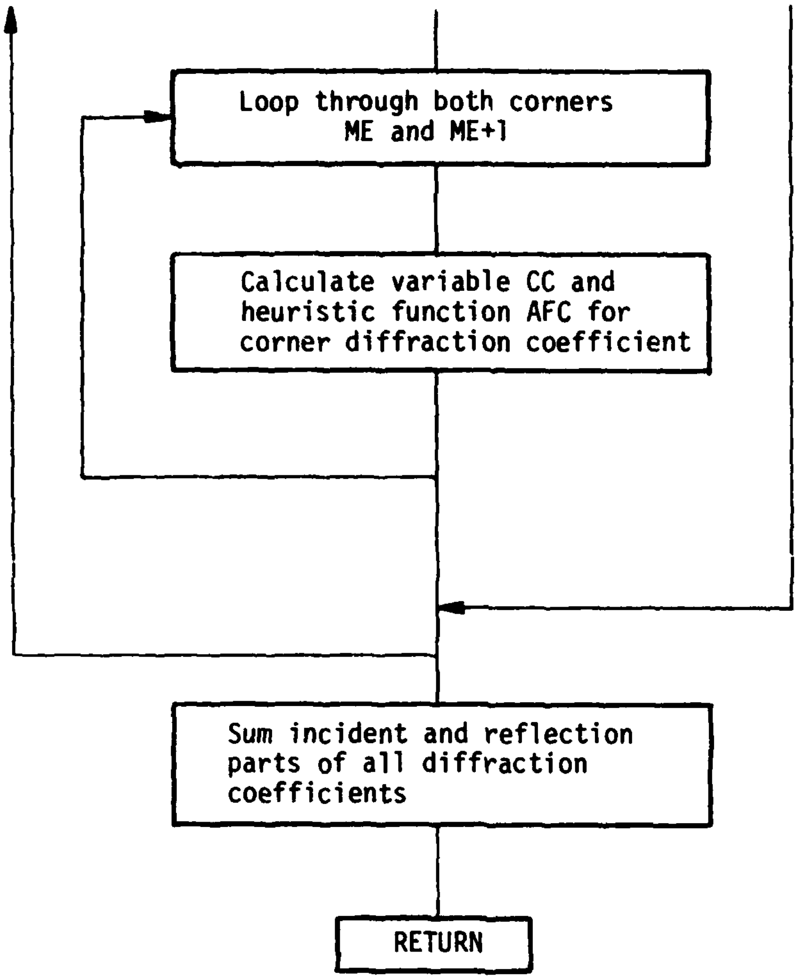
FLOW DIAGRAM

DCHP(DEL,CORN,R,PS,PSO,SBO)

INPUT VARIABLES

| | | |
|---|---|---|
| DEL | | Variable representing the value of $kL_c a(\beta_{oc}+\beta_c)$ |
| CORN | | Variable representing the part of the corner diffraction coefficient exclusive of $C_{s,h}$ |
| R | (L) | Distance parameter |
| PS | ($\phi$) | Diffraction angle for diffracted field |
| PSO | ($\phi'$) | Diffraction angle for incident field |
| SBO | | Sine of the diffracted cone angle $\beta_o$ |




Loop through both corners ME and ME+1
Calculate variable CC and heuristic function AFC for corner diffraction coefficient
Sum incident and reflection parts of all diffraction coefficients
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| A | $(a)$ | Angular separation parameter | |
| AFC | | Heuristic function for corner diffraction | |
| ANG | | Variable for PS±PSO in radians | |
| ARG | | Argument for AFC | |
| BET | $(\beta)$ | Variable for PS±PSO in degrees | |
| BH | $(B_h)$ | Hard corner diffraction coefficient | (0) |
| BPH | $\left(\frac{\partial B_h}{\partial \phi'}\right)$ | Hard slope corner diffraction coefficient | (0) |
| BPS | $\left(\frac{\partial B_s}{\partial \phi'}\right)$ | Soft slope corner diffraction coefficient | (0) |
| BS | $(B_s)$ | Soft corner diffraction coefficient | (0) |
| CC | | Variable representing the combined effect for both corners ME and ME+1 in the corner diffraction | |
| DI | | The incident or reflection part of the edge diffraction coefficient | |
| DH | $(D_h)$ | Hard diffraction coefficient for edge | (0) |
| DPH | $\left(\frac{\partial D_h}{\partial \phi'}\right)$ | Hard slope diffraction coefficient for edge | (0) |
| DPS | $\left(\frac{\partial D_s}{\partial \phi'}\right)$ | Soft slope diffraction coefficient for edge | (0) |
| DS | $(D_s)$ | Soft diffraction coefficient for edge | (0) |
| FA | | Transition function for edge diffraction | |
| FFCT | | Transition function (see section on FFCT) | |
| LCORNR | | Logical variable for corner diffraction (see User's Manual) | (I) |
| LSLOPE | | Logical variable for slope diffraction (see User's Manual) | (I) |

|  |  |
|---|---|
| SL | Value for DI near the shadow boundaries |
| TERM | Temporary coefficient for DS,DH |
| TERMP | Temporary coefficient for DPS,DPH |

CODE LISTING

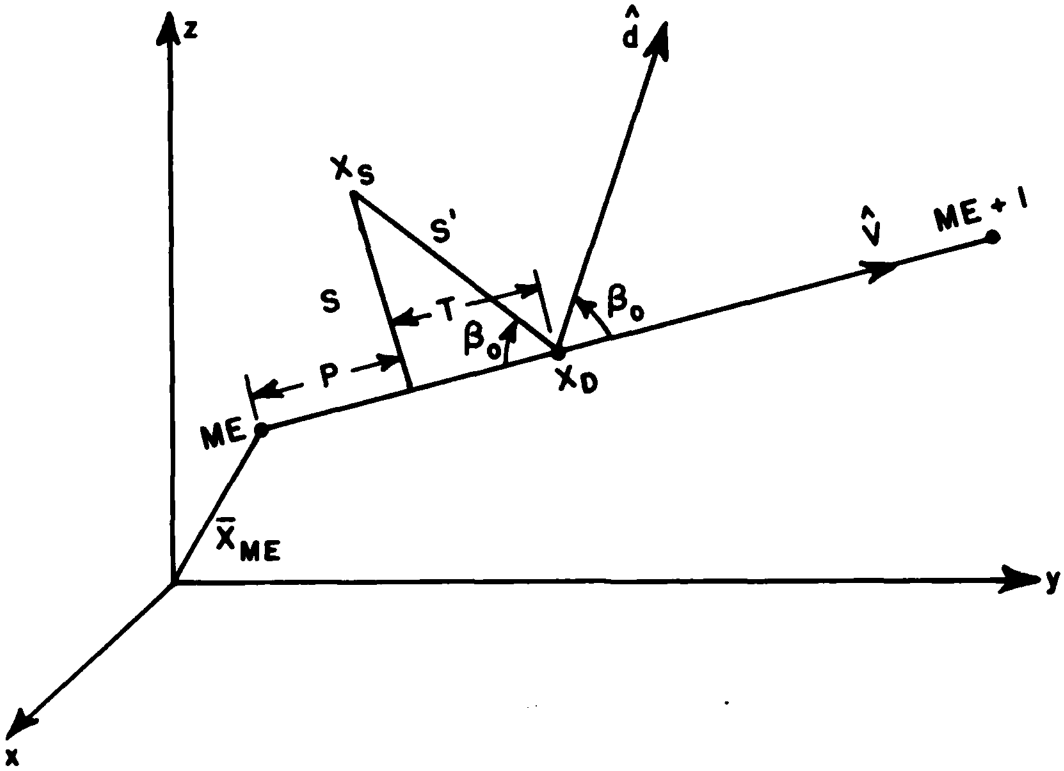
```
      SUBROUTINE DCHP(DEL,CORN,R,PS,PSO,SBO)
      DIMENSION DEL(2)
      COMPLEX CORN(2),DI(2),DPI(2),CC(2),TERM,TERMP,FFCT,FA,TOP
      COMPLEX CJ,DS,DH,DPS,DPH,BS,BH,BPS,BPH,CIN,CIP,CCP
      COMMON /DSC/DS,DH,DPS,DPH,BS,BH,BPS,BPH
      COMMON /PIS/PI,TPI,DPR
      COMMON /LOGDIF/LSLOPE,LCORNR,LNF,LRANG
      COMMON /TEST/LDEBUG,LTEST,NTEST
      COMMON /TOPD/TOP
      COMMON /OUT/NW
      LOGICAL LCORNR,LSLOPE,LDEBUG,LTEST
      IF (LDEBUG) WRITE (NW,2) R,PS,PSO,SBO,LOG,DEL(1),DEL(2)
    2 FORMAT (T5,'DEBUGGING SUBROUTINE DCHP',//4F10.2,L5,2F10.4)
      CJ=(0.,1.)
      TERM=TOP/(2.*TPI*SBO)
      TERMP=CJ*2.*TPI*TERM*R
      IF (LDEBUG) WRITE (NW,3) TERM,TERMP
    3 FORMAT (T10,6HTERM =,2F10.4,5X,7HTERMP =,2F10.4)
      SL=0.5*SQRT(R)/SBO
      IF (DEL(1).LT.1.D-20) DEL(1)=1.D-20
      IF (DEL(2).LT.1.D-20) DEL(2)=1.D-20
      BET=PS-PSO
      DO 20 N=1,2
      ANG=BET/DPR
      SB=SIN(ANG/2.)
      CB=COS(ANG/2.)
      A=2.*CB*CB
      IF (LDEBUG) WRITE (NW,4) BET,CB,A
    4 FORMAT (T10,5HBET =,F8.2,5X,4HCB =,F10.6,5X,3HA =,F10.6)
      X=TPI*ABS(R*A)
      IF ((LSLOPE).OR.(X.LE.10.)) FA=FFCT(X)
      IF (X.GT.10.) FA=1.+CJ/(2.*X)-3./(4.*X*X)
      IF (LDEBUG) WRITE (NW,7) X,FA
    7 FORMAT (T10,3HX =,F10.4,5X,4HFA =,2F10.6)
      IF (A.GT.1.D-20) GO TO 5
      DI(N)=-SL+CJ*0.
      GO TO 8
    5 CONTINUE
```

```
      DI(N)=TERM*FA/CB
    8 CONTINUE
      DPI(N)=TERMP*SB*(1.-FA)
      IF (LDEBUG) WRITE (NW,9) DI(N),DPI(N)
    9 FORMAT (T10,4HDI =,2F10.5,5X,5HDPI =,2F10.5)
      IF (.NOT.LCORNR) GO TO 15
      CC(N)=(0.,0.)
      DO 12 I=1,2
      ARG=R*A/DEL(I)
      AFC=1.
      IF (ARG.LE.10.) AFC=BABS(FFCT(ARG))
      CC(N)=CC(N)+CORN(I)*AFC
      IF (LDEBUG) WRITE (NW,11) ARG,AFC,CC(N),CORN(1),CORN(2)
   11 FORMAT (T5,5HARG =,F10.4,5X,5HAFC =,7F10.6)
   12 CONTINUE
   15 BET=PS+PSO
   20 CONTINUE
      DS=DI(1)-DI(2)
      DH=DI(1)+DI(2)
      DPS=DPI(1)+DPI(2)
      DPH=DPI(1)-DPI(2)
      IF (.NOT.LCORNR) RETURN
      CIN=DI(1)*CC(1)
      CIP=DI(2)*CC(2)
      BS=CIN-CIP
      BH=CIN+CIP
      CCP=CORN(1)+CORN(2)
      BPS=DPS*CCP
      BPH=DPH*CCP
      RETURN
      END
```

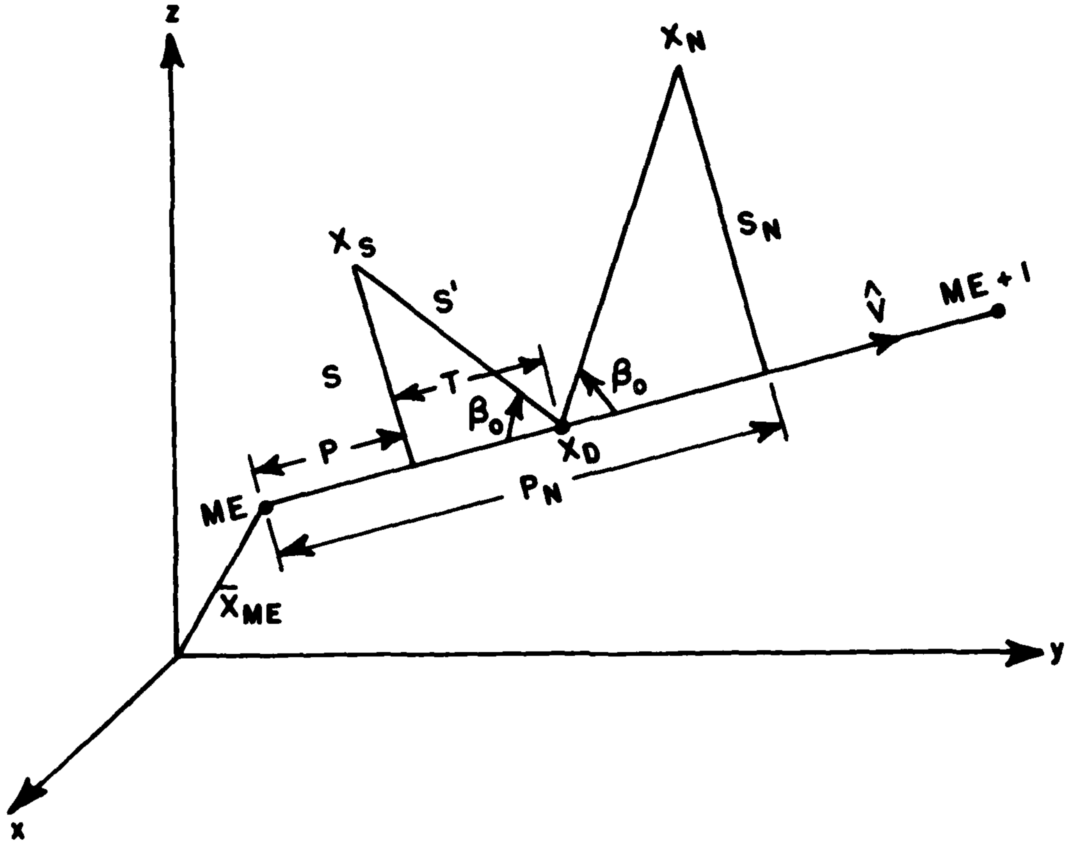
SUBROUTINE DFPTWD

PURPOSE

To determine the diffraction point on a rim segment for either a far field or near field point and determine the incident ray unit vector.



(a) far field



(b) Near field

Figure 1. Geometry for locating diffraction point $X_D$ on edge ME.

METHOD

The coordinates of the diffraction point $X_D$ are determined by solving a similar triangle system. For edge ME,

$$P = (\bar{X}_S - \bar{X}) \cdot \hat{V}$$

and

$$S = |\bar{X}_S - \bar{X}_{ME} - P\hat{V}|$$

For far field (see Fig. 1a), since $\cos\beta_o = \hat{d}\cdot\hat{V}$ then

$$\cot\beta_o = \frac{\hat{d}\cdot\hat{V}}{\sqrt{1-(\hat{d}\cdot\hat{V})^2}}$$

and $T = S\cot\beta_o$ .

For near field (see Fig. 1b)

$$P_N = (\overline{X}_N-\overline{X}_{ME})\cdot\hat{V}$$

$$S_N = |\overline{X}_N-\overline{X}_{ME}-P_N\hat{V}|$$

and

$$T = (P_N-P)S/(S+S_N) \quad .$$

Thus the coordinates of the diffraction point are determined by the vector

$$\overline{X}_D = \overline{X}_{ME} + (P+T)\hat{V} \quad .$$

The distance S' from the source $X_S$ to the diffraction point $X_D$ is determined from

$$\overline{VI} = \overline{X}_D-\overline{X}_S$$

and

$$S' = |\overline{VI}| \quad .$$

The unit vector for the incident ray is then obtained by normalizing the above vector

$$\hat{VI} = \frac{\overline{VI}}{S'}$$

FLOW DIAGRAM

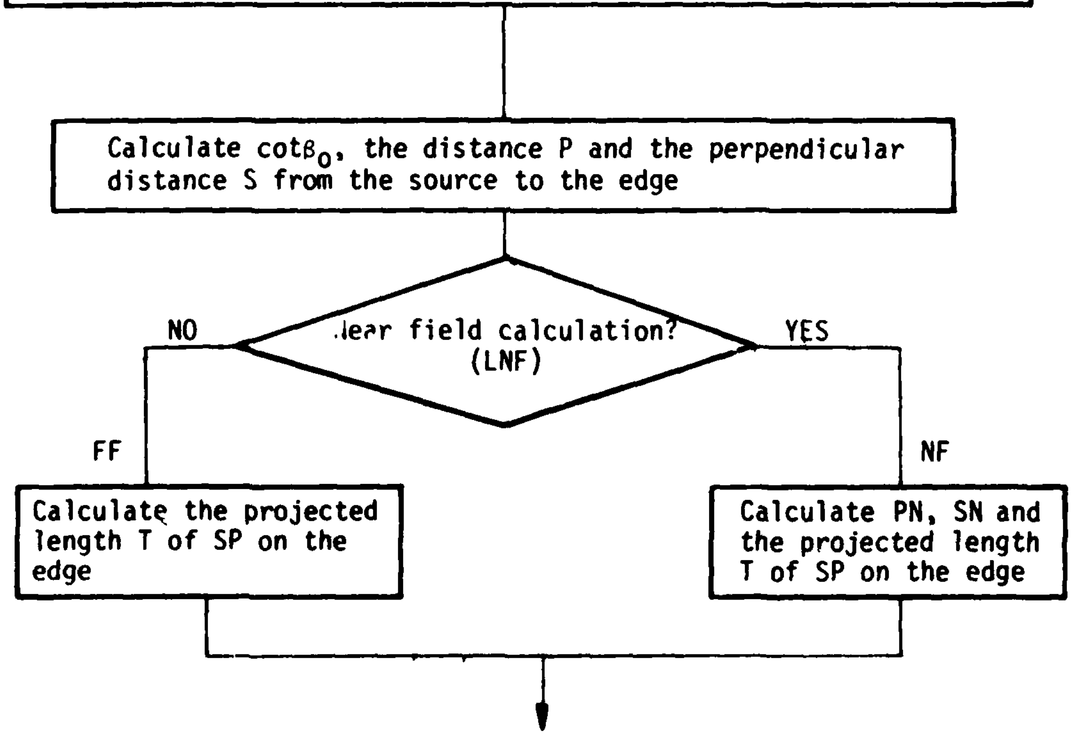
DFPTWD(XS,XN,DV,VI,SP,XD,ME)

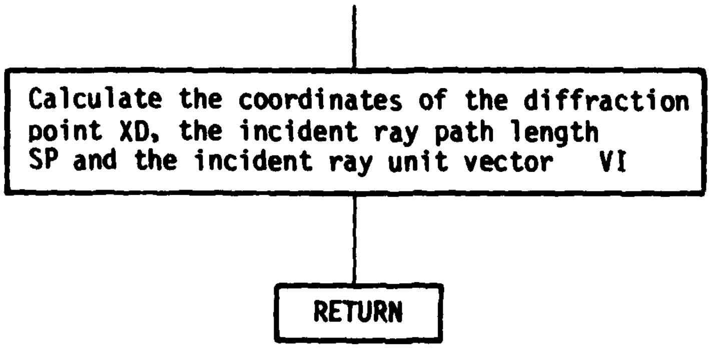
INPUT VARIABLES

| | |
|---|---|
| XS | Source location in rectangular coordinate system |
| XN | Near field point coordinates |
| DV | Inner product of the diffraction vector and the edge vector |
| ME | Rim segment index |

OUTPUT VARIABLES

| | |
|---|---|
| VI | Incident ray unit vector |
| SP | Incident ray path length |
| XD | Coordinates of the diffraction point |




Calculate the coordinates of the diffraction point XD, the incident ray path length SP and the incident ray unit vector VI
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| CTB | | Cotangent of diffraction angle $\beta_0$ | |
| LNF | | Logical variable to determine whether near field or far field is calculated | (I) |
| P | | Distance from the source to the corner ME projected on the edge | |
| PN | ($P_N$) | Distance from the near field point XN to the corner ME projected on the edge | |
| S | | Perpendicular distance from the source to the edge | |
| SN | ($S_N$) | Perpendicular distance from the near field point to the edge | |
| SP | ($S'$) | Incident ray path length | |
| T | | Projected length of SP on the edge | |
| X | ($X_{ME}$) | Rectangular coordinate components of the rim point ME | (I) |

CODE LISTING

```
      SUBROUTINE DFPTWD(XS,XN,DV,VI,SP,XD,ME)
C!!!
C!!!  DETERMINATION OF THE DIFFRACTION POINT
C!!!
      DIMENSION XS(3),XN(3),XD(3),VI(3)
      LOGICAL LSLOPE,LCORNR,LNF,LRANG
      COMMON /GEOM1/X(67,3),V(67,3),MRIM
      COMMON /LOGDIF/LSLOPE,LCORNR,LNF,LRANG
      CTB=DV/SQRT(1.-DV*DV)
      P=0.
      PN=0.
      DO 10 N=1,3
      IF (LNF) PN=PN+(XN(N)-X(ME,N))*V(ME,N)
   10 P=P+(XS(N)-X(ME,N))*V(ME,N)
      S=0.
      SN=0.
      DO 20 N=1,3
      SY=XN(N)-X(ME,N)-PN*V(ME,N)
      SX=XS(N)-X(ME,N)-P*V(ME,N)
      SN=SN+SY*SY
   20 S=S+SX*SX
      S=SQRT(S)
      SN=SQRT(SN)
      T=S*CTB
      IF (LNF) T=(PN-P)*S/(S+SN)
      DO 30 N=1,3
   30 XD(N)=X(ME,N)+(P+T)*V(ME,N)
      SP=0.
      DO 40 N=1,3
      VI(N)=XD(N)-XS(N)
   40 SP=SP+VI(N)*VI(N)
      SP=SQRT(SP)
      DO 50 N=1,3
   50 VI(N)=VI(N)/SP
      RETURN
      END
```

## SUBROUTINE FEED

### PURPOSE

To determine the magnitude of the feed pattern in any given direction as referred to the reflector coordinate system.

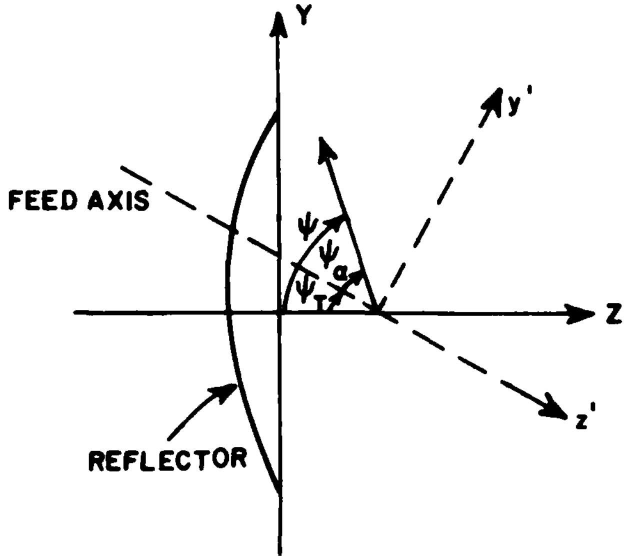


Figure 1. Geometry of the feed coordinate system.

### METHOD

For a given direction $(\psi,\phi)$ in the reflector coordinate system the transformation from $(\psi,\phi)$ to $(\psi_\alpha,\phi_\gamma)$ in the feed coordinate is given by

$$\psi_\alpha = \cos^{-1}(\sin\psi_T \sin\psi\sin\phi + \cos\psi_T \cos\psi)$$

$$\phi_\gamma = \tan^{-1} \frac{\cos\psi_T\sin\psi\sin\phi - \sin\psi_T\cos\psi}{\sin\psi\cos\phi}$$

where $\psi = \pi - \theta$ and $\psi_T$ is the feed tilt angle (in the YZ plane).

Various symmetry options are available to reduce the amount of input data required for symmetrical feed patterns, as shown in Table 1.

TABLE 1

| \|ISYM\| | SYMMETRY | $\phi_X$ | LIMITS FOR $\phi_X$ |
|---|---|---|---|
| 0 | NO | $\phi$ | $-\pi < \phi_X \leq \pi$ |
| 1 | x- and y-axis | $\|\phi\|$ or $\pm\pi-\phi$ | $0 \leq \phi_X \leq \frac{\pi}{2}$ |
| 2 | x-axis | $\|\phi\|$ | $0 \leq \phi_X \leq \pi$ |
| 3 | y axis | $\pm\pi-\phi$ | $-\frac{\pi}{2} \leq \phi_X \leq \frac{\pi}{2}$ |

To find the feed pattern value at $(\psi_\alpha,\phi_\chi)$, $\phi_\chi$ is adjusted and is represented by $\phi_X$ according to the symmetry index ISYM. Then the two PHI cuts $\phi_P$, $\phi_Q$ of the input feed pattern adjacent to $\phi_X$ are determined by comparison. The feed pattern values $g_P$ and $g_Q$ are calculated at the angle $\psi=\psi_\alpha$ in the planes $\phi_P$ and $\phi_Q$, respectively, by using either linear interpolation between stored values or an analytic pattern function.

The linear interpolation is performed by calling subroutine LNFD.

The analytic pattern is constructed by

$$\begin{cases} g_n = C\, e^{-A\left(\frac{\psi}{\psi_o}\right)^2} \sin^N\left(\frac{\pi\psi}{2\psi_o}\right) & \text{if ISYM} < 0 \text{ (odd symmetry)} \\ g_n = \dfrac{e^{-A\left(\frac{\psi}{\psi_o}\right)^2} \cos^N\left(\frac{\pi\psi}{2\psi_o}\right) + C}{1 + C} & \text{if ISYM} \geq 0 \text{ (even symmetry)} \end{cases}$$

$$\text{for } \psi \leq \psi_L$$

$$g_n = g_n(\psi_L)\left(1 - \frac{\psi-\psi_L}{\psi_L}\right) \qquad \text{for } \psi_L < \psi < 2\psi_L$$

and

$$g_n = 0 \qquad \text{for } 2\psi_L \leq \psi$$

where

$$\psi_L = \sqrt{\frac{3}{A}\,\psi_o}$$

is an empirical cutoff criterion and N, C, A and $\psi_o$ are input parameters to control the feed pattern.

Finally, the magnitude of the feed pattern value $g_f$ at $(\psi_\alpha, \phi_\gamma)$ is obtained by interpolating $g_P$ and $g_Q$ as follows:

$$g_f = d_{PQ}\, g_P + (1 - d_{PQ}) g_Q$$

where

$$d_{PQ} = \frac{\phi \;\; - \phi_Q}{\phi_P - \phi_Q}$$

## FLOW DIAGRAM

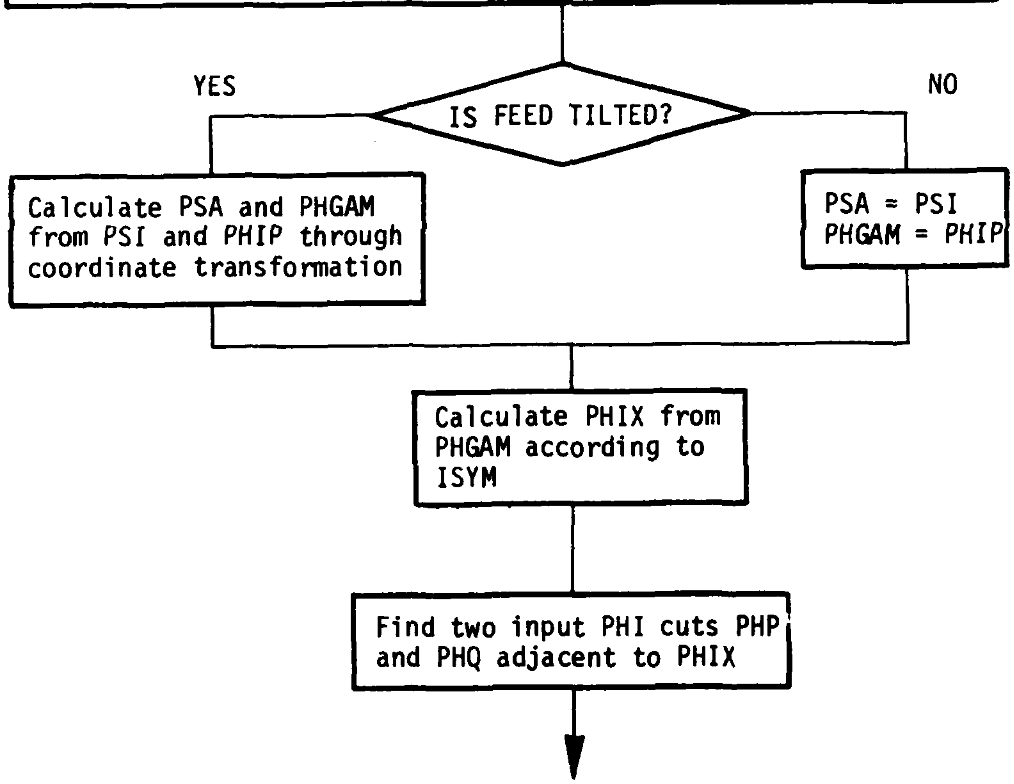
FEED (PSI,,PHIP,PSA,PHGAM)

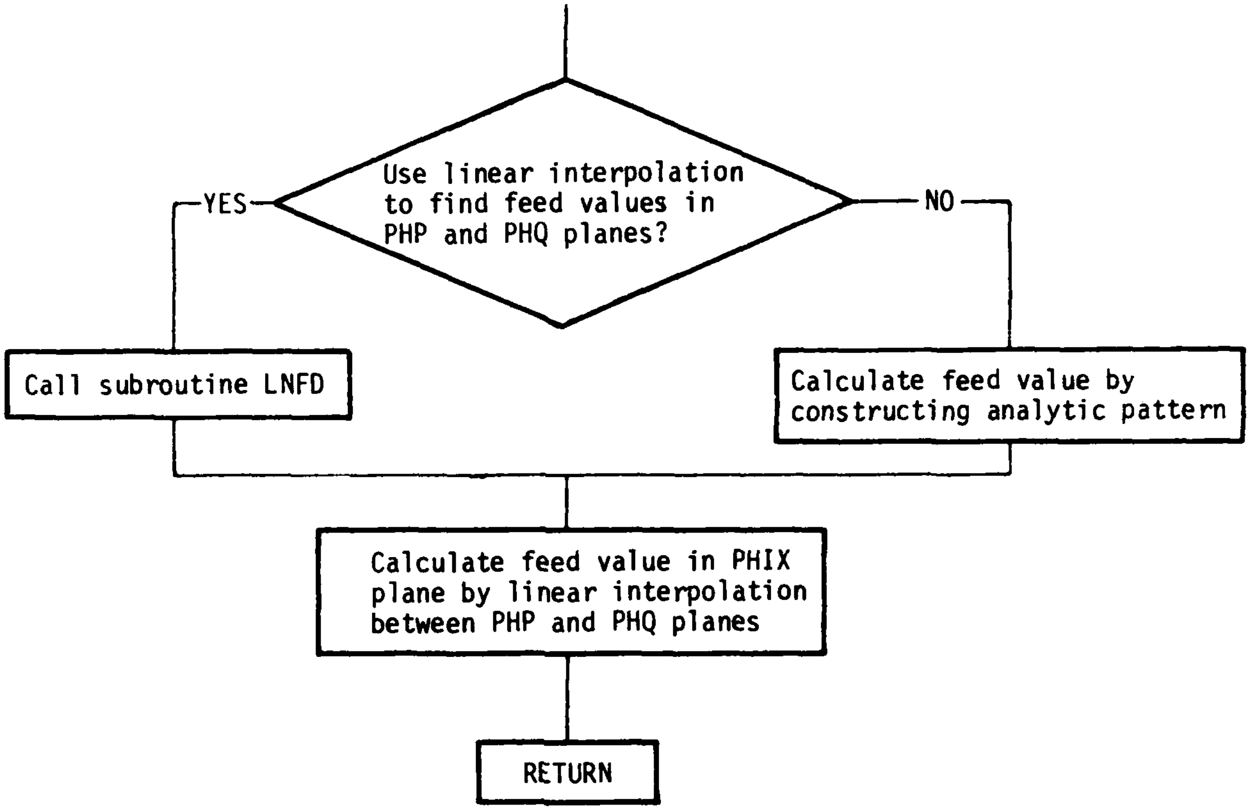
INPUT VARIABLES

| | | |
|---|---|---|
| PSI | ($\psi$) | Theta coordinate of the observation direction measured from the negative Z-axis. |
| PHIP | ($\phi'$) | Phi corrdinate of the observation direction. |

OUTPUT VARIABLES

| | | |
|---|---|---|
| PSA | ($\psi_\alpha$) | Theta coordinate of the observation direction measured from the feed axis. |
| PHGAM | ($\phi_\gamma$) | Phi coordinate of the observation direction referred to the tilted feed system. |




Use linear interpolation to find feed values in PHP and PHQ planes?
YES
NO
Call subroutine LNFD
Calculate feed value by constructing analytic pattern
Calculate feed value in PHIX plane by linear interpolation between PHP and PHQ planes
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| FP | | Input feed pattern data for linear interpolation | (I) |
| GF | ($g_f$) | Feed pattern value at (PSA, PHGAM) | (O) |
| GP | ($g_p$) | Feed pattern value calculated at PSA in PHP cut | |
| GQ | ($g_q$) | Feed pattern value calculated at PSA in PHQ cut | |
| ISYM | | Symmetry index for input feed pattern | |
| PHIN | | Input feed pattern cut angle | (I) |
| PHIX | ($\phi_X$) | Adjusted PHGAM angle according to ISYM | |
| PHP | ($\phi_P$) | Upper input PHI cut adjacent to PHIX | |
| PHQ | ($\phi_Q$) | Lower input PHI cut adjacent to PHIX | |
| PSIL | ($\psi_L$) | Cutoff criterion for analytic pattern | |
| PSIO | ($\psi_O$) | Input parameter to control the feed pattern | (I) |
| PSIT | ($\psi_T$) | Feed tilt angle | (I) |
| PX | | Input feed pattern angle | (I) |

CODE LISTING

```
      SUBROUTINE FEED(PSI,PHIP,PSA,PHGAM)
      DIMENSION PHIN(15),FP1(15),FP(15,15),AEX(15),CAN(15),
     1PX1(15),PX(15,15),GN(2),FP2(15),PX2(15),PSIO(15)
      COMPLEX CX,CY
      LOGICAL LDEBUG,LTEST,LPSL,LDB
      COMMON /FED/N2,PHIN,PX,FP,LDB,NCK,NPHI,NPW,AEX,CAN,PSIO,PSIT
      COMMON /COMP/CX,CY,GF,PHP,PHO,KX,KY,ISYM,SINT,COST
      COMMON /PIS/PI,TPI,DPR
      COMMON /PREV/IPR,PREP,PREX,PRES
      COMMON /TEST/LDEBUG,LTEST,NTEST
      COMMON /OUT/NW
      IF (NTEST.EQ.1) WRITE (NW,8) PSI,PHIP
    8 FORMAT (//T8,'DEBUGGING SUBROUTINE FEED',/T12,'PSI =',
     2F7.2,5X,'PHIP =',F7.2,/)
      PHGAM=PHIP
      PSA=PSI
      IF (PSIT.EQ.0.) GO TO 10
      PSIR=PSA/DPR
      PHIPR=PHIP/DPR
      SINS=SIN(PSIR)
      COSS=COS(PSIR)
      SINP=SIN(PHIPR)
      COSP=COS(PHIPR)
      PSA=ACOS(SINT*SINS*SINP+COST*COSS)*DPR
      TEMP=COST*SINS*SINP-SINT*COSS
      IF ((ABS(TEMP)+ABS(COSP)).LT.0.0001) TEMP=0.0001
      PHGAM=ATAN2(TEMP,SINS*COSP)*DPR
   10 CONTINUE
    5 FORMAT (3F12.4)
      IF (ISYM.EQ.IPR.AND.PHGAM.EQ.PREP) GO TO 15
      PHIX=PHGAM
      IB=IABS(ISYM)
      IF (IB.EQ.0) GO TO 15
      IF (IB.EQ.3) GO TO 12
      PHIX=ABS(PHIX)
      IF (IB.EQ.2) GO TO 15
   12 IF (PHIX.GT.90.) PHIX=180.-PHIX
      IF (PHIX.LT.-90.) PHIX=-180.-PHIX
   15 CONTINUE
      IF (NTEST.EQ.2.OR.NTEST.EQ.1) WRITE (NW,18) PHGAM,PSA,PHIX
   18 FORMAT (/T12,'PHGAM =',F7.2,' PSA =',F7.2,' PHIX =',F7.2)
      SINPX=SIN(PHIX/DPR)
      IF (PHIX.GE.PHO.AND.PHIX.LE.PHP) GO TO 30
      DO 20 NP=2,NPHI
      IF (PHIX.LE.PHIN(NP)) GO TO 22
   20 CONTINUE
      NP=NPHI+1
      PHIN(NP)=PHIN(1)+360.
      PSIO(NP)=PSIO(1)
```

```
 22   NQ=NP-1
      PHQ=PHIN(NQ)
      PHP=PHIN(NP)
      IF (NTEST.EQ.1) WRITE (NW,25) PHQ,PHP
 25   FORMAT (T12,'PHQ =',F7.2,5X,'PHP =',F7.2)
      IF (NCK.EQ.2) GO TO 32
 27   DO 28 K=1,N2
      PX1(K)=PX(NQ,K)
      FP1(K)=FP(NQ,K)
      PX2(K)=PX(NP,K)
      FP2(K)=FP(NP,K)
 28   CONTINUE
 30   IF (NCK.EQ.2) GO TO 32
      DPSI=ABS(PSA-PRES)
      IF (DPSI.LT.0.1.AND.PHIX.EQ.PREX) GO TO 39
      CALL LNFD(PX1,FP1,PSA,N2,GQ,LDB)
      CALL LNFD(PX2,FP2,PSA,N2,GP,LDB)
      IF (NTEST.EQ.1) WRITE (6,5) PSA,GQ,GP
      IF (.NOT.LDB) GO TO 39
      GQ=10.**(GQ/20.)
      GP=10.**(GP/20.)
      GO TO 39
 32   LPSL=.FALSE.
      SLOPE=0.
      PSY=PSA
      DO 38 N=1,2
      NN=NQ+N-1
      IF (ISYM.LT.0) GO TO 33
      IF (NPW.NE.1.OR.CAN(NN).LT.0..OR.AEX(NN).LT.3.) GO TO 33
      PSL=SQRT(3./AEX(NN))*PSIO(NN)
      DPSL=PSA-PSL
      IF (DPSL.LE.0.) GO TO 33
      PSY=PSL
      LPSL=.TRUE.
      IF (DPSL.LT.PSL) GO TO 33
      GN(N)=0.
      GO TO 38
 33   QX=PSY/PSIO(NN)
      ARG=0.5*PI*QX
      ARGEX=AEX(NN)*QX*QX
      IF (ARGEX.LT.20.) EXPN=EXP(-ARGEX)
      IF (ARGEX.GE.20.) EXPN=0.
      IF (NTEST.EQ.1) WRITE (NW,35) ARG,AEX(NN),CAN(NN),EXPN
 35   FORMAT (/T12,'ARG =',F9.3,5X,'AEX(N) =',F9.3,5X,'CAN(N) =',
     2F9.3,5X,'EXPN =',F9.3,/)
      IF (ISYM.GE.0) GO TO 37
      GN(N)=CAN(NN)*EXPN*SIN(ARG)**NPW
      GO TO 38
```

```
   37 GN(N)=EXPN*COS(ARG)**NPW
      GN(N)=(GN(N)+CAN(NN))/(1.+CAN(NN))
      IF (LPSL) SLOPE=-GN(N)/PSL
      GN(N)=GN(N)+SLOPE*DPSL
   38 CONTINUE
      GO=GN(1)
      GP=GN(2)
   39 DPO=(PHIX-PHO)/(PHP-PHO)
      GF=GP*DPO+GO*(1-DPO)
      IF ((NTEST.EQ.1).OR.(NTEST.EQ.2)) WRITE (NW,50) GF
   50 FORMAT (/T10,'GF =',F10.4)
      PREX=PHIX
      PREP=PHGAM
      PRES=PSA
      IPR=ISYM
      RETURN
      END
```

## FUNCTION FF

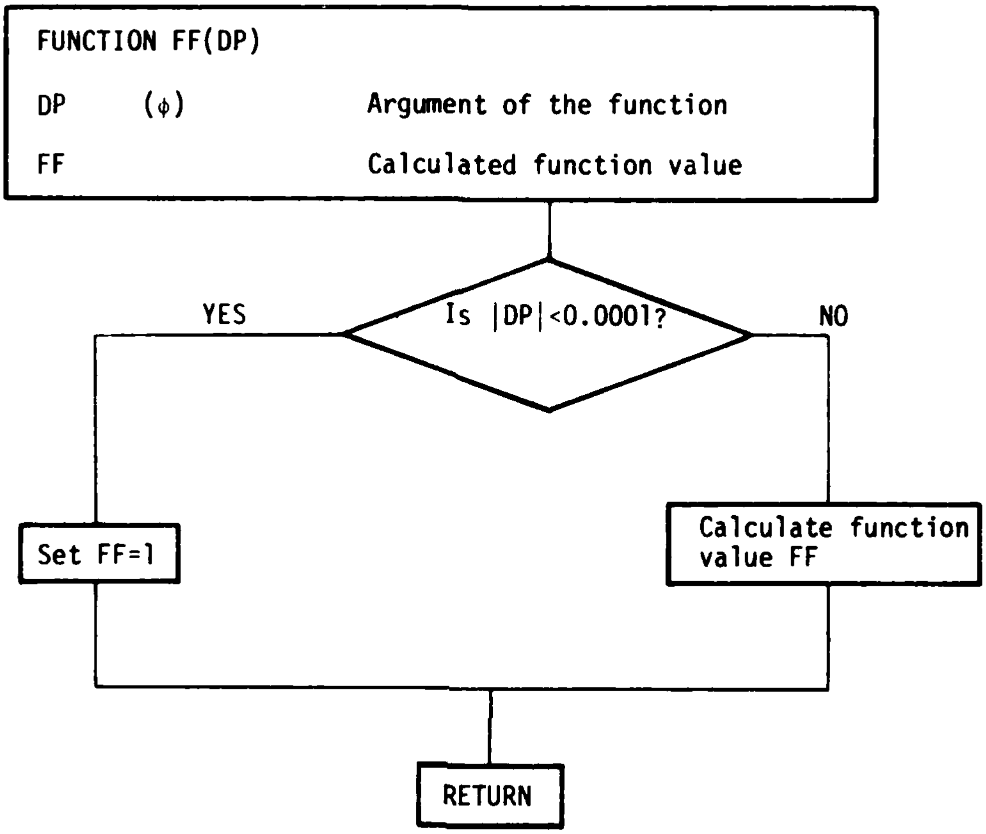
## PURPOSE

To calculate the element pattern function of a rectangular subaperture with full triangular distribution.

## METHOD

$$F_F(\phi) = \left(\frac{\sin \frac{\phi}{2}}{\frac{\phi}{2}}\right)^2$$

FLOW DIAGRAM



CODE LISTING

```
      FUNCTION FF(DP)
      IF (ABS(DP).LT.0.0001) GO TO 11
      X=DP/2.
      TEMP=SIN(X)/X
      FF=TEMP*TEMP
      GO TO 12
   11 FF=1.
   12 RETURN
      END
```

## FUNCTION FFCT

### PURPOSE

The purpose of this function is to determine the transition function for the edge and corner diffraction coefficients.

### METHOD

The transition function for the edge and corner diffraction coefficients is given by[5]:

$$\mathrm{FFCT}(x) = 2j|\sqrt{x}|\; e^{jx} \int_{|\sqrt{x}|} e^{-j\tau^2}\, d\tau .$$

This can also be written as

$$\mathrm{FFCT}(x) = j\sqrt{2\pi|x|}\; e^{jx}\left[ (0.5-j0.5) - \left( C\left(\sqrt{\frac{2|x|}{\pi}}\right) - jS\left(\sqrt{\frac{2|x|}{\pi}}\right)\right]$$

where

$$\int_0^{\alpha} e^{-j\frac{\pi}{2}t^2}\, dt = C(\alpha) - jS(\alpha).$$

### KEY VARIABLES

| | |
|---|---|
| CFR | Real part of Fresnel integral |
| DEL | Argument of transition function |
| FFCT | Transition function |
| S | Argument of Fresnel integral |
| SDEL | SQRT(ABS(DEL)) |
| SFR | Imaginary part of Fresnel integral |

CODE LISTING

```
      COMPLEX FUNCTION FFCT(DEL)
C!!!
C!!!  DETERMINES THE TRANSITION FUNCTION RESULT FOR
C!!!  CORNER DIFFRACTED FIELD
C!!!
      COMMON/PIS/PI,TPI,DPR
      COMMON /OUT/NW
      SDEL=SQRT(ABS(DEL))
      S=SQRT(2./PI)*SDEL
      CALL FRNELS(CFR,SFR,S)
      FFCT=CMPLX(0.5-CFR,SFR-0.5)
      FFCT=SQRT(TPI)*SDEL*FFCT*CEXP(CMPLX(0.,DEL+PI/2.))
      RETURN
      END
```

## FUNCTION FH

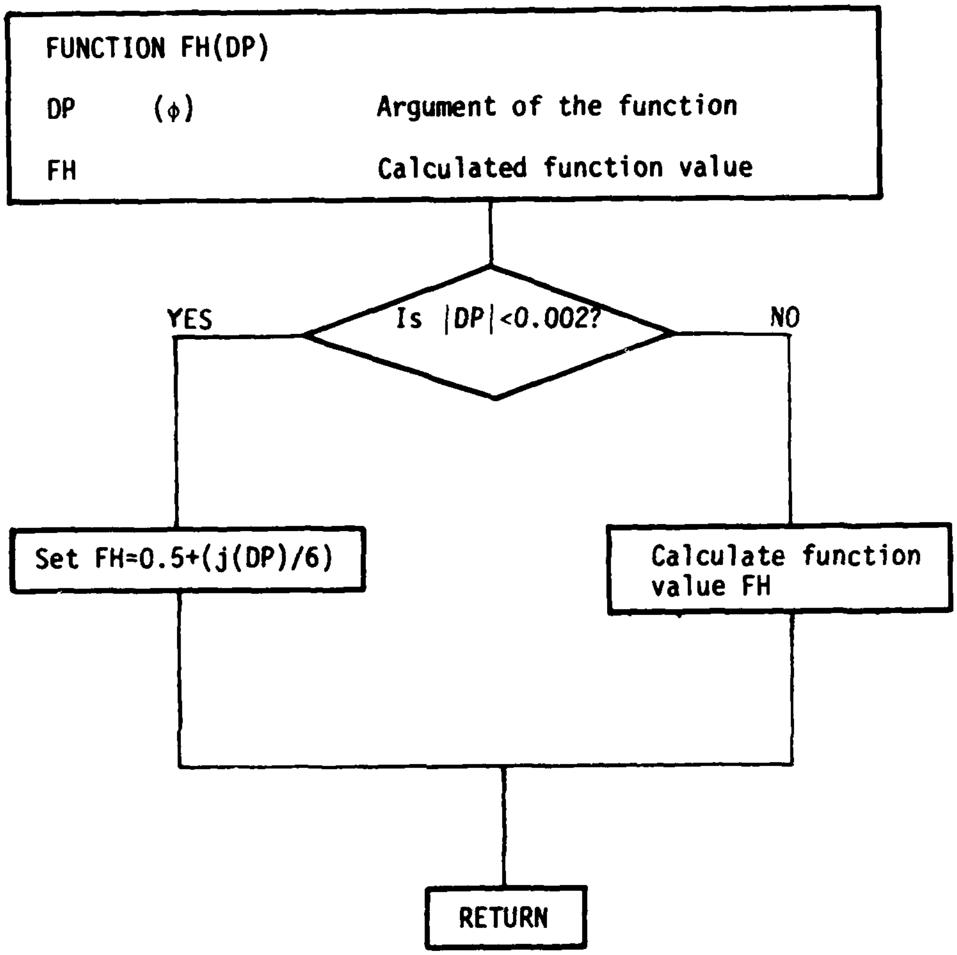
## PURPOSE

To calculate the element pattern function of a rectangular subaperture with half triangular distribution.

## METHOD

$$F_H(\phi) = \frac{1-e^{j\phi}}{(\phi)^2} + \frac{j}{\phi}$$

FLOW DIAGRAM



CODE LISTING

```
      COMPLEX FUNCTION FH(DP)
      COMPLEX CJ,CDP,TEMP
      CJ=(0.,1.)
      CDP=CJ*DP
      IF (ABS(DP).LT.0.002) GO TO 21
      TEMP=(1.-CEXP(CDP))/(DP*DP)
      FH=TEMP-(1./CDP)
      GO TO 22
   21 FH=0.5+CDP/6.
   22 RETURN
      END
```

## SUBROUTINE FPOL

### PURPOSE

To calculate the rectangular vector components of the E-field of the feed as referred to the reflector coordinate system.

### METHOD

The two linear polarization components of the feed pattern of an arbitrarily oriented (in the x-y plane) Huygen's source (crossed electric and magnetic dipoles [2]) are given by

$$f_x = C_x \cdot \phi_s \cdot g_f$$

$$f_x = C_y \cdot \phi_s \cdot g_f$$

where $C_x$ and $C_y$ are polarization parameters expressed as

$$\begin{cases} C_x = \cos(\tau) \\ C_y = \sin(\tau) \end{cases} \qquad \text{for linearly polarized feed with polarization angle} = \tau$$

or

$$\begin{cases} C_x = \dfrac{1}{\sqrt{2}} \\ C_y = \dfrac{j}{\sqrt{2}} \end{cases} . \qquad \text{for circularly polarized feed} \cdot$$

$\phi_s$ is the phase of excitation as given by

$$\phi_s = 1 \qquad \text{ISYM} \geq 0 \text{ (even symmetry)}$$

$$\left.\begin{array}{ll} \phi_s = e^{j\phi} & \text{ISYM} = -1 \\ \phi_s = \dfrac{\sin\phi}{|\sin\phi|} & \text{ISYM} = -2 \\ \phi_s = \dfrac{\cos\phi}{|\cos\phi|} & \text{ISYM} = -3 \end{array}\right\} \text{ (odd symmetry)} .$$

and $g_f$ is the magnitude of the feed pattern which is calculated by the subroutine FEED.

The spherical vector components of the feed pattern are obtained by

$$E^i_{\theta_\alpha} = -\cos\phi_\gamma f_x - \sin\phi_\gamma f_y$$

$$E^i_{\phi_\gamma} = -\sin\phi_\gamma f_x + \cos\phi_\gamma f_y$$

where $\theta_\alpha$ and $\phi_\gamma$ are the spherical coordinate angles in the feed coordinate system.

The rectangular components are calculated by

$$E^i_x = -\cos\psi_\alpha \cos\phi_\gamma E^i_{\theta_\alpha} - \sin\phi_\gamma E^i_{\phi_\gamma}$$

$$E^i_y = -\cos\psi_\alpha \sin\phi_\gamma E^i_{\theta_\alpha} + \cos\phi_\gamma E^i_{\phi_\gamma}$$

and

$$E^i_z = -\sin\psi_\alpha E^i_{\theta_\alpha} \quad .$$

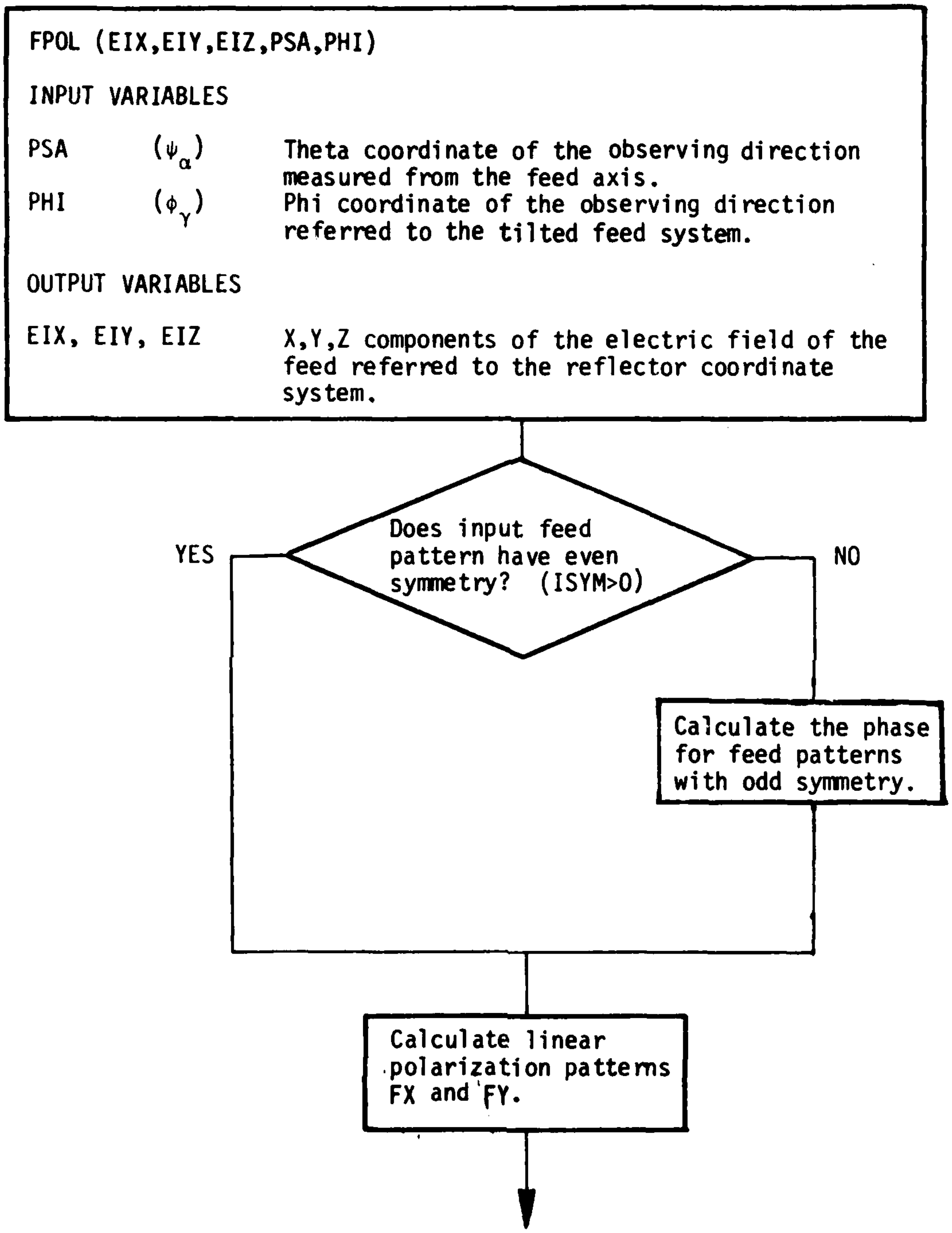
The electric field vector is then transformed from the tilted feed coordinate system to the reflector coordinate system as follows:

$$E^i_x = E^i_x$$

$$E^i_y = \cos\psi_T E^i_y - \sin\psi_T E^i_z$$

$$E^i_z = \sin\psi_T E^i_y + \cos\psi_T E^i_z$$

FLOW DIAGRAM



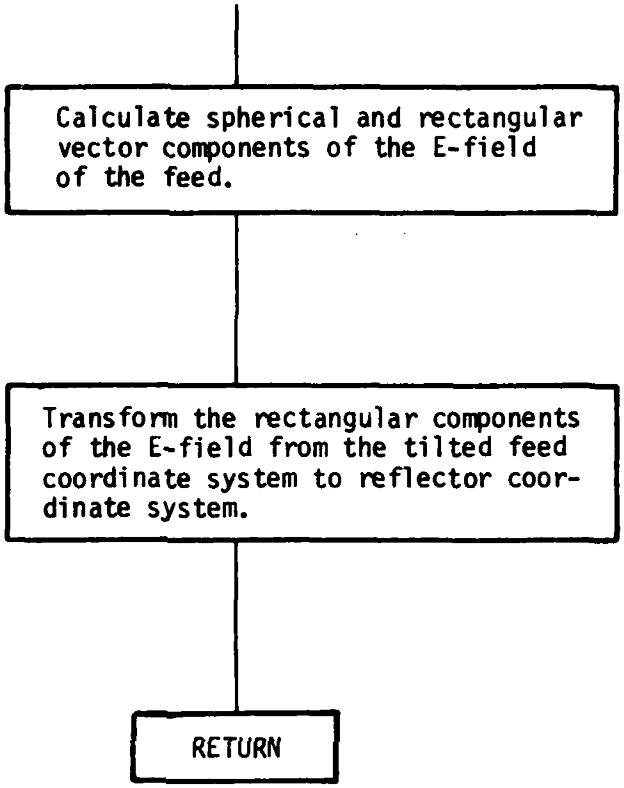
FPOL (EIX,EIY,EIZ,PSA,PHI)

INPUT VARIABLES

| | | |
|---|---|---|
| PSA | ($\psi_\alpha$) | Theta coordinate of the observing direction measured from the feed axis. |
| PHI | ($\phi_\gamma$) | Phi coordinate of the observing direction referred to the tilted feed system. |

OUTPUT VARIABLES

| | |
|---|---|
| EIX, EIY, EIZ | X,Y,Z components of the electric field of the feed referred to the reflector coordinate system. |


Calculate spherical and rectangular vector components of the E-field of the feed.
Transform the rectangular components of the E-field from the tilted feed coordinate system to reflector coordinate system.
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| CX | $(C_x)$ | Polarization parameter for x-polarized feed. | (I) |
| CY | $(C_y)$ | Polarization parameter for y-polarized feed. | (I) |
| EIP | $(E^i_{\phi_\gamma})$ | PHI component of the electric field in the feed coordinate system. | |
| EIT | $(E^i_{\theta_\alpha})$ | THETA component of the electric field in the reflector coordinate system. | |
| FX | $(f_x)$ | x-polarized feed pattern. | |
| FY | $(f_y)$ | y-polarized feed pattern. | |
| GF | $(g_f)$ | Feed pattern value calculated by subroutine FEED | (I) |
| PHASE | $(\phi_s)$ | Phase of feed pattern | |

CODE LISTING

```
      SUBROUTINE FPOL(EIX,EIY,EIZ,PSA,PHI)
      COMPLEX CJ,EIX,EIY,EIZ,EIP,EIT,PHASE,FX,FY,CX,CY
      COMPLEX TEMP
      COMMON /PIS/PI,TPI,DPR
      COMMON /COMP/CX,CY,GF,PHP,PHQ,KX,KY,ISYM,SINT,COST
      COMMON/TEST/LDEBUG,LTEST,NTEST
      LOGICAL LDEBUG,LTEST
      CJ=(0.,1.)
      PSI=PSA
      IF (NTEST.GT.0) WRITE (6,1) PSI,PHI,SINT,COST
    1 FORMAT (/T10,'DEBUGGING FPOL SUBROUTINE',/T15,4F10.3)
      IF (ABS(PSI-90.).LT.0.0001) PSI=89.9
      PSIR=PSI/DPR
      PHIR=PHI/DPR
      SINS=SIN(PSIR)
      COSS=COS(PSIR)
      SINP=SIN(PHIR)
      COSP=COS(PHIR)
      PHASE=(1.,0.)
      IF (ISYM.GE.0) GO TO 8
      IF (ISYM+2) 4,2,6
```

```
    2 REL=SINP/ABS(SINP)
      PHASE=REL+CJ*0.
      GO TO 8
    4 REL=COSP/ABS(COSP)
      PHASE=REL+CJ*0.
      GO TO 8
    6 PHASE=CEXP(CJ*PHIR)
    8 FX=CX*PHASE*GF
      FY=CY*PHASE*GF
      EIX=(0.,0.)
      EIY=(0.,0.)
      EIZ=(0.,0.)
      EIT=(0.,0.)
      EIP=(0.,0.)
      IF (KX.EQ.0) GO TO 10
      EIT=-COSP*FX
      EIP=-SINP*FX
   10 IF (KY.EQ.0) GO TO 20
      EIT=EIT-SINP*FY
      EIP=EIP+COSP*FY
   20 CONTINUE
      EIX=-COSP*COSS*EIT-SINP*EIP
      EIY=-SINP*COSS*EIT+COSP*EIP
      EIZ=-SINS*EIT
      IF (SINT.LT.0.01) GO TO 25
      TEMP=COST*EIY-SINT*EIZ
      EIZ=SINT*EIY+COST*EIZ
      EIY=TEMP
   25 CONTINUE
      IF (NTEST.EQ.0) GO TO 40
      WRITE (6,30) PSI,PHI,EIX,EIY,EIZ
   30 FORMAT (/T10,'PSA =',F8.2,5X,'PHGAM =',F8.2,/3(T10,2F10.4,/))
   40 RETURN
      END
```

## FUNCTION FRNELS

## PURPOSE

To compute the Fresnel integral,

$$f(x_s) = \int_0^{x_s} e^{-j\pi/2\, u^2}\, du = C(x_s) - j\, S(x_s).$$

## METHOD

The integral is evaluated using an approximation by J. Boersma[8]. The integral

$$f(x) = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}}\, dt$$

is approximated as follows:

$$\text{for } 0 \leq x \leq 4 \quad f(x) = \epsilon^{-jx} \sqrt{\frac{x}{4}} \sum_{n=0}^{11} (a_n + jb_n) \left(\frac{x}{4}\right)^n$$

$$\text{for } x \geq 4 \quad f(x) = \frac{1-j}{2} + e^{-jx} \sqrt{\frac{4}{x}} \sum_{n=0}^{11} (c_n + jd_n) \left(\frac{4}{x}\right)^n$$

(the constants $a_n$, $b_n$, $c_n$ and $d_n$ are provided by Boresma and are defined in data statements in the subroutine).

Note that by performing a change of variable, the integral to be solved becomes of the form of the integral which Boersma solved;

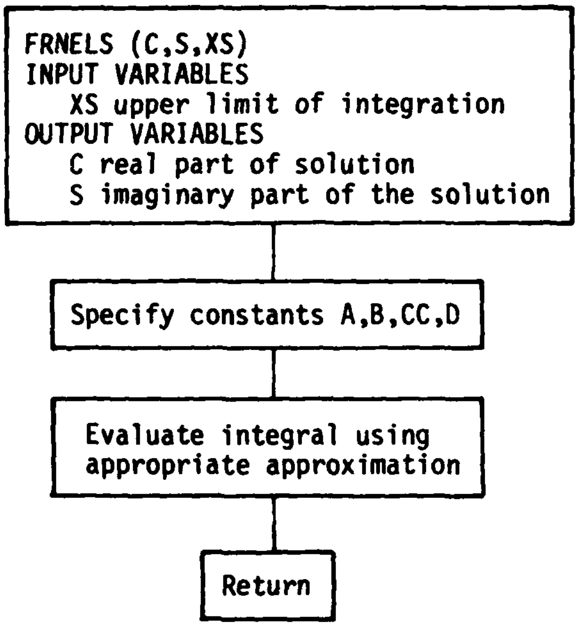
$$t = \frac{\pi}{2}\, u^2.$$

By applying this change of variable, we get

$$f(x_s) = \int_0^{x_s} e^{-j\frac{\pi}{2}u^2}\, du = \int_0^x \frac{e^{-jt}}{\sqrt{2\pi t}}\, dt$$

where $x = \frac{\pi}{2}\, x_s^2$.

FLOW DIAGRAM



KEY VARIABLE

| | |
|---|---|
| A<br>B<br>CC<br>D | Constants used in evaluating integral |
| FI | Imaginary component of summation function |
| FR | Real component of summation function |

CODE LISTING

```
      SUBROUTINE FRNELS(C,S,XS)
C!!!
C!!! THIS IS THE FRESNEL INTEGRAL SUBROUTINE WHERE THE INTEGRAL IS FROM
C!!! U=0 TO XS, THE INTEGRAND IS EXP(-J*PI/2.*U*U),AND THE OUTPUT IS
C!!!  C(XS)-J*S(XS).
C!!!
      LOGICAL LDEBUG,LTEST
      COMMON/TEST/LDEBUG,LTEST,NTEST
      COMMON/PIS/PI,TPI,DPR
      COMMON /OUT/NW
      DIMENSION A(12),B(12),CC(12),D(12)
      DATA A/1.595769140,-0.000001702,-6.808568854,-0.000576361,6.920691
     &902,-0.016898657,-3.050485660,-0.075752419,0.850663781,-0.02563904
     &1,-0.150230960,0.034404779/
      DATA B/-0.000000033,4.255387524,-0.000092810,-7.780020400,-0.00952
     &0895,5.075161298,-0.138341947,-1.363729124,-0.403349276,0.70222201
     &6,-0.216195929,0.019547031/
      DATA CC/0.,-0.024933975,0.000003936,0.005770956,0.000689892,-0.009
     &497136,0.011948809,-0.006748873,0.000246420,0.002102967,-0.0012179
     &30,0.000233939/
      DATA D/0.199471140,0.000000023,-0.009351341,0.000023006,0.00485146
     &6,0.001903218,-0.017122914,0.029064067,-0.027928955,0.016497308,-0
     &.005598515,0.000838386/
      IF(XS.LE.0.0) GO TO 414
      X=XS
      X = PI*X*X/2.0
      FR=0.0
      FI=0.0
      K=13
      IF(X-4.0) 10,40,40
   10 Y=X/4.0
   20 K=K-1
      FR=(FR+A(K))*Y
      FI=(FI+B(K))*Y
      IF(K-2) 30,30,20
   30 FR=FR+A(1)
      FI=FI+B(1)
      C=(FR*COS(X)+FI*SIN(X))*SQRT(Y)
      S=(FR*SIN(X)-FI*COS(X))*SQRT(Y)
      GO TO 1
   40 Y=4.0/X
   50 K=K-1
      FR=(FR+CC(K))*Y
      FI=(FI+D(K))*Y
      IF(K-2) 60,60,50
   60 FR=FR+CC(1)
      FI=FI+D(1)
```

```
      C=0.5+(FR*COS(X)+FI*SIN(X))*SQRT(Y)
      S=0.5+(FR*SIN(X)-FI*COS(X))*SQRT(Y)
      GO TO 1
  414 C=-0.0
      S=-0.0
    1 IF (.NOT.LTEST) GO TO 2
      WRITE (NW,3)
    3 FORMAT (/,' TESTING FRNELS SUBROUTINE')
      WRITE (NW,*) C,S,XS
    2 RETURN
      END
```

## SUBROUTINE GEOM

### PURPOSE

To approximate the reflector rim by straight segments and to calculate the unit vectors for each segment. Also the permissible range for the diffraction angle $\beta_0$ for each rim segment is determined.

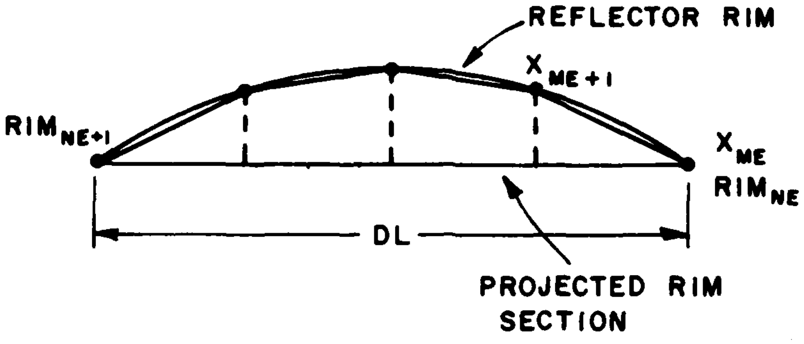


Figure 1. Illustration of subdivision of a reflector rim into straight segments

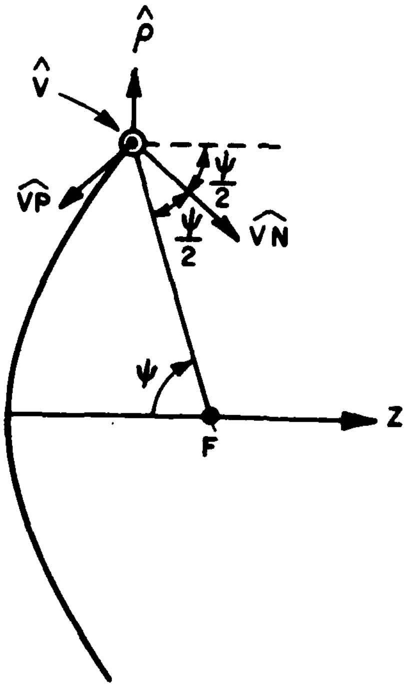


Figure 2. Unit vectors associated with the reflector rim.

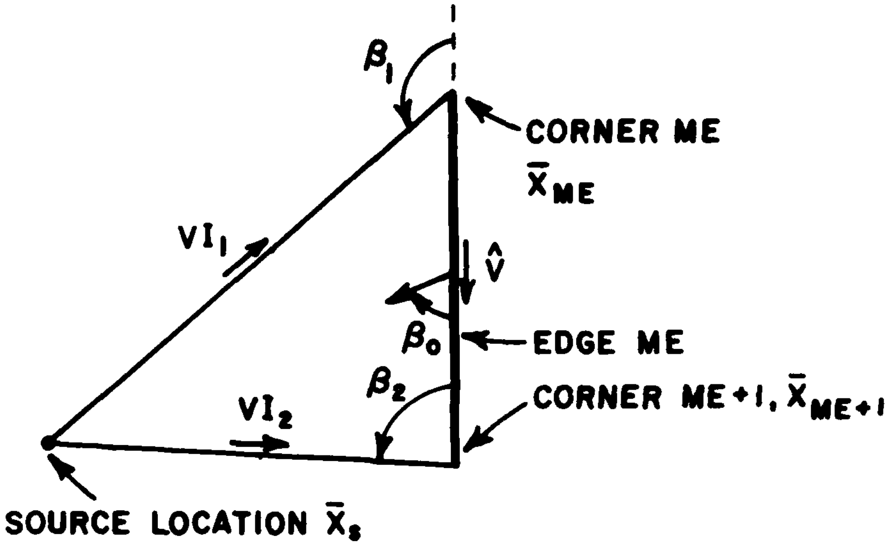


Figure 3. Geometry for determining diffraction angle range.

## METHOD

### a). Subdividing the reflector rim into straight segments

To ensure the focus of the parabola lies in the far field of the reflector rim, the section of the reflector rim between each pair of input rim points $RIM_{NE}$ and $RIM_{NE+1}$ is subdivided into K straight segments. The integer K is obtained by the formula

$$K = \text{Int}\left(\frac{DL}{RIML} + 1\right)$$

where DL is the length of the projected rim section on the aperture plane and RIML is the approximate length of a straight segment which is defined in the main program.

The coordinates of the new rim points, as shown in Fig. 1, are calculated by

$$X(ME+L,N) = RIM(NE,N) + L \times DEL(N)$$

where

L=I-1, I=1,2,----K is the number of segments in rim section NE

DEL(N) is the length of each rim segment and

N=1,2 representing the X and Y components respectively.

The Z coordinate of the rim point ME is given by

$$X(ME,3) = \frac{X(ME,1)^2+X(ME,2)^2}{4F} - Z'$$

where F is the focal distance and Z' is the coordinate of the vertex of the parabolic reflector.

b). *The unit vectors*

The edge unit vectors are found by

$$\hat{V}_{ME} = \frac{\bar{X}_{ME+1}-\bar{X}_{ME}}{|\bar{X}_{ME+1}-\bar{X}_{ME}|}$$

The unit normals are determined by considering that the normal vector of each rim edge is also normal to the parabolic surface for the limiting case. Since the diffraction point is not determined until all the unit vectors of that edge are found, the normal at the midpoint of an edge is used to approximate that at the diffraction point. Thus, as shown in Fig. 2

$$\widehat{VN}_{ME} = -\hat{\rho}\sin\frac{\psi}{2} + \hat{z}\cos\frac{\psi}{2}$$

where

$$\hat{\rho} = \hat{x}\cos\phi + \hat{y}\sin\phi$$

Note that $\psi$ and $\phi$ are the spherical coordinates of the midpoint with respect to the source point $X_S$ and are given by

$$\psi = \tan^{-1}\frac{\sqrt{VIM_x^2+VIM_y^2}}{(-VIM_z)}$$

and

$$\phi = \tan^{-1}\left(\frac{VIM_y}{VIM_x}\right)$$

where

$$\overline{VIM} = \frac{\overline{X}_{ME+1}+\overline{X}_{ME}}{2} - \overline{X}_S$$

The unit binormals are obtained by

$$\hat{VP}_{ME} = \hat{VN}_{ME} \times \hat{V}_{ME} \quad .$$

c). The permissible range for the diffraction angle.

The law of diffraction dictates that diffraction from a plate edge is possible when

$$\cos\beta_2 \leq \cos\beta_0 \leq \cos\beta_1 \quad ,$$

where $\beta_0$ is the angle that the incident and diffracted rays make with the edge (see Fig. 3). $\beta_1$ and $\beta_2$ are diffraction angle limits and are defined in terms of their cosines as:

$$BD(ME,1) = \cos\beta_1 = \hat{VI}_1 \cdot \hat{V}$$

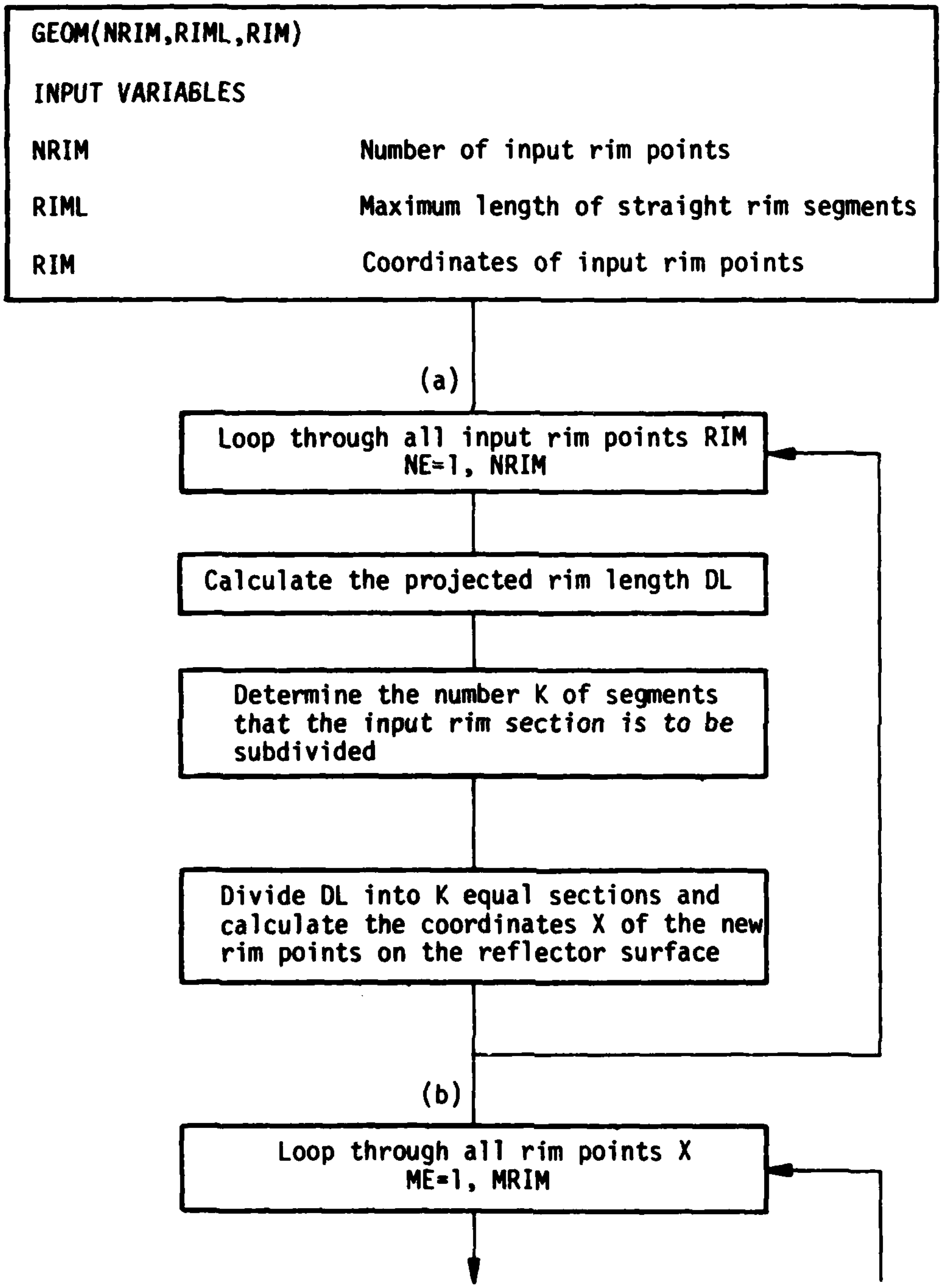
$$BD(ME,2) = \cos\beta_2 = \hat{VI}_2 \cdot \hat{V} \quad ,$$

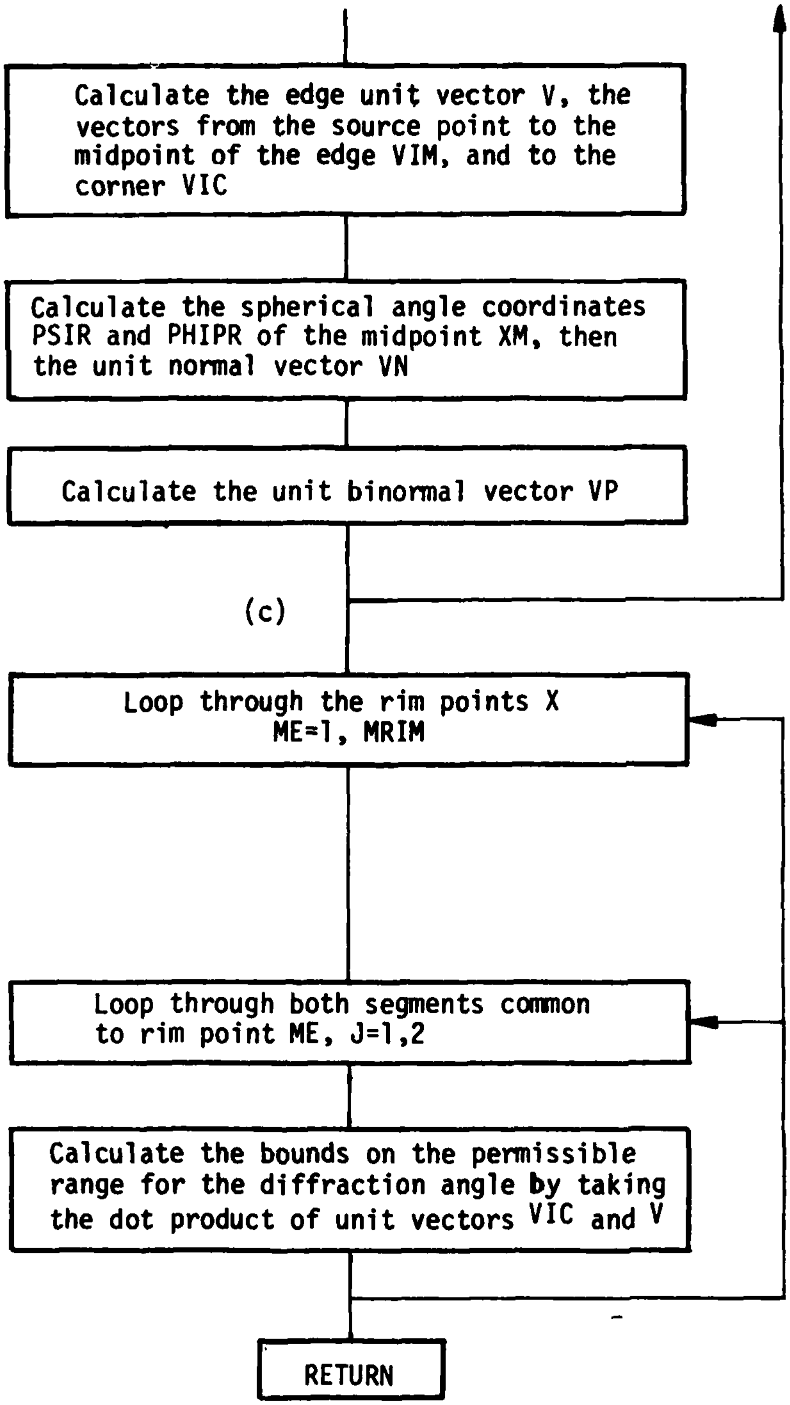
where

$$\hat{VI}_1 = \frac{\overline{X}_{ME}-\overline{X}_S}{|\overline{X}_{ME}-\overline{X}_S|}$$

$$\hat{VI}_2 = \frac{\overline{X}_{ME+1}-\overline{X}_S}{|\overline{X}_{ME+1}\ \overline{X}_S|}$$

FLOW DIAGRAM




Calculate the edge unit vector V, the vectors from the source point to the midpoint of the edge VIM, and to the corner VIC
Calculate the spherical angle coordinates PSIR and PHIPR of the midpoint XM, then the unit normal vector VN
Calculate the unit binormal vector VP
(c)
Loop through the rim points X ME=1, MRIM
Loop through both segments common to rim point ME, J=1,2
Calculate the bounds on the permissible range for the diffraction angle by taking the dot product of unit vectors VIC and V
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| BD | | Bounds for diffraction angle | (0) |
| DEL | | x and y components of the subdivided segment length | |
| DL | | Projected rim segment | |
| DSQ | | Square of DL | |
| K | | Number of straight segments into which an input rim section is subdivided | |
| ME | | Loop index of subdivided rim points X | |
| MRIM | | Total number of subdivided rim points | |
| NE | | Loop index of input rim points RIM | |
| NRIM | | Number of input rim points | (I) |
| PHIPR | ($\phi$) | PHI coordinate angle of the midpoint XM with respect to the source point in radians | |
| PSIR | ($\psi$) | THETA coordinate angle of the midpoint XM with respect to the source point in radians | |
| RMC | | Incident ray path length at the corner | (0) |
| RMM | | Incident ray path length at the midpoint XM of a straight edge | |
| THNR | $\left(\frac{\psi}{2}\right)$ | Half angle of PSIR | |
| V | | X, Y and Z components of the edge unit vector | (0) |
| VIC | | Stored X, Y and Z components of the incident ray vector at the corner for all edges | (0) |
| VIM | | X, Y and Z components of the incident ray vector at the midpoint | |
| VN | | X, Y and Z components of the unit normal vector | (0) |
| VP | | X, Y and Z components of the unit binormal vector | (0) |

| | | | |
|---|---|---|---|
| X | ($X_{ME}$) | Coordinates of the new rim point ME | (O) |
| XM | | Coordinates of the midpoint of edge ME | |
| ZOP | | Z-coordinate of the vertex of the parabolic reflector | (I) |

CODE LISTING

```
      SUBROUTINE GEOM(NRIM,RIML,RIM)
C
      DIMENSION RIM(67,2),VI(3),VIM(67,3),RMM(67),DEL(2)
      LOGICAL LDEBUG,LTEST
      COMMON /GEOM1/X(67,3),V(67,3),MRIM
      COMMON /GEOM2/VP(67,3),VN(67,3),BD(67,2),VMAG(67),RMC(67),
     2VIC(67,3),XM(67,3)
      COMMON /FOCAL/F,ZOP
      COMMON /DIM/MDRIM
      COMMON /SORINF/XS(3)
      COMMON /PIS/PI,TPI,DPR
      COMMON /OUT/NW
      COMMON /TEST/LDEBUG,LTEST,NTEST
      IF (LDEBUG) WRITE (NW,2)
    2 FORMAT (/T5,'DEBUGGING SUBROUTINE GEOM',//)
C
C     *** APPROXIMATE THE CURVE EDGES BY LINE SEGMENTS ***
C
      ME=0
      DO 22 NE=1,NRIM
      NEP=NE+1
      IF (NE.EQ.NRIM) NEP=1
      DSQ=0.
      DO 5 N=1,2
      XX=RIM(NEP,N)-RIM(NE,N)
    5 DSQ=DSQ+XX**2
      DL=SQRT(DSQ)
      K=DL/RIML+1
      IF (RIML.LE.0) K=1
      IF (LDEBUG) WRITE (6,8) NE,DL,RIML,K
    8 FORMAT (/T10,4HNE =,I2,5X,4HDL =,2F8.2,5X,3HK =,I2,/)
      DO 10 N=1,2
   10 DEL(N)=(RIM(NEP,N)-RIM(NE,N))/K
      DO 20 I=1,K
      L=I-1
      ME=ME+1
      IF (ME.GT.MDRIM) GO TO 50
      DO 15 N=1,2
   15 X(ME,N)=RIM(NE,N)+L*DEL(N)
      X(ME,3)=(X(ME,1)**2+X(ME,2)**2)/(4.*F)-ZOP
```

```
      IF (LDEBUG) WRITE (6,18) ME,(X(ME,N),N=1,3)
   18 FORMAT (I15,5F10.3)
   20 CONTINUE
   22 CONTINUE
      MRIM=ME
C
C!!!  DETERMINATION OF EDGE UNIT VECTORS
C
      MEX=MRIM
      DO 38 ME=1,MEX
      MME=ME+1
      IF(MME.GT.MEX) MME=1
      VM=0.
      VMM=0.
      VMC=0.
      DO 25 N=1,3
      V(ME,N)=X(MME,N)-X(ME,N)
      XM(ME,N)=(X(MME,N)+X(ME,N))/2.
      VIM(ME,N)=XM(ME,N)-XS(N)
      VIC(ME,N)=X(ME,N)-XS(N)
      VMM=VMM+VIM(ME,N)*VIM(ME,N)
      VMC=VMC+VIC(ME,N)*VIC(ME,N)
   25 VM=VM+V(ME,N)*V(ME,N)
      VMAG(ME)=SQRT(VM)
      RMM(ME)=SQRT(VMM)
      RMC(ME)=SQRT(VMC)
   28 FORMAT (T10,2F12.4)
      IF (LDEBUG) WRITE (NW,30) ME
   30 FORMAT (/T8,4HME =,I2,4X,3HVIM,7X,3HVIC,/)
      DO 32 N=1,3
      VIM(ME,N)=VIM(ME,N)/RMM(ME)
      IF (LDEBUG) WRITE (NW,28) VIM(ME,N),VIC(ME,N)
   32 V(ME,N)=V(ME,N)/VMAG(ME)
C
C     ***** CALCULATE THE NORMAL VECTORS OF THE EDGES ***
C
      PSIR=BTAN2(SQRT(VIM(ME,1)**2+VIM(ME,2)**2),-VIM(ME,3))
      PHIPR=BTAN2(VIM(ME,2),VIM(ME,1))
      PSI=PSIR*DPR
      PHIP=PHIPR*DPR
      SINPP=SIN(PHIPR)
      COSPP=COS(PHIPR)
      THNR=PSIR/2.
      SINR=SIN(THNR)
      COSR=COS(THNR)
      VN(ME,1)=-SINR*COSPP
      VN(ME,2)=-SINR*SINPP
      VN(ME,3)=COSR
      VNM=0.
      DO 34 N=1,3
   34 VNM=VNM+VN(ME,N)*VN(ME,N)
      VNM=SQRT(VNM)
```

```
      DO 35 N=1,3
   35 VN(ME,N)=VN(ME,N)/VNE
C
C!!!  DETERMINATION OF UNIT VECTOR FOR RAY FIXED COORDINATE SYSTEM
C
      VP(ME,1)=VN(ME,2)*V(ME,3)-VN(ME,3)*V(ME,2)
      VP(ME,2)=VN(ME,3)*V(ME,1)-VN(ME,1)*V(ME,3)
      VP(ME,3)=VN(ME,1)*V(ME,2)-VN(ME,2)*V(ME,1)
      IF (LDEBUG) WRITE(NW,36) (V(ME,N),VN(ME,N),VP(ME,N),
     2N=1,3)
   36 FORMAT (T10,3F12.4)
      IF (LDEBUG) WRITE (NW,37) RMM(ME),RMC(ME)
   37 FORMAT (/T10,5HRMM =,F7.3,5X,5HRMC =,F7.3,/)
   38 CONTINUE
C
C!!!  DETERMINATION OF PERMISSABLE RANGE FOR DIFFRACTION ANGLE
C
      DO 45 ME=1,MEX
      VME=0.
      DO 40 N=1,3
      VI(N)=X(ME,N)-XS(N)
   40 VME=VME+VI(N)*VI(N)
      RME=SQRT(VME)
      DO 41 J=1,2
      MJ=ME+1-J
      IF(MJ.EQ.0) MJ=MEX
      BD(MJ,J)=0.
      DO 41 N=1,3
   41 BD(MJ,J)=BD(MJ,J)+V(MJ,N)*VI(N)/RME
   45 CONTINUE
      RETURN
   50 MRIM=ME
      RETURN
      END
```

## SUBROUTINE GRID

### PURPOSE

To set up a rotated coordinate system such that the aperture integration for far field results can be carried out efficiently. This subroutine is also used to set up the principal grid which is used for aperture field calculations and aperture integration for near field results.

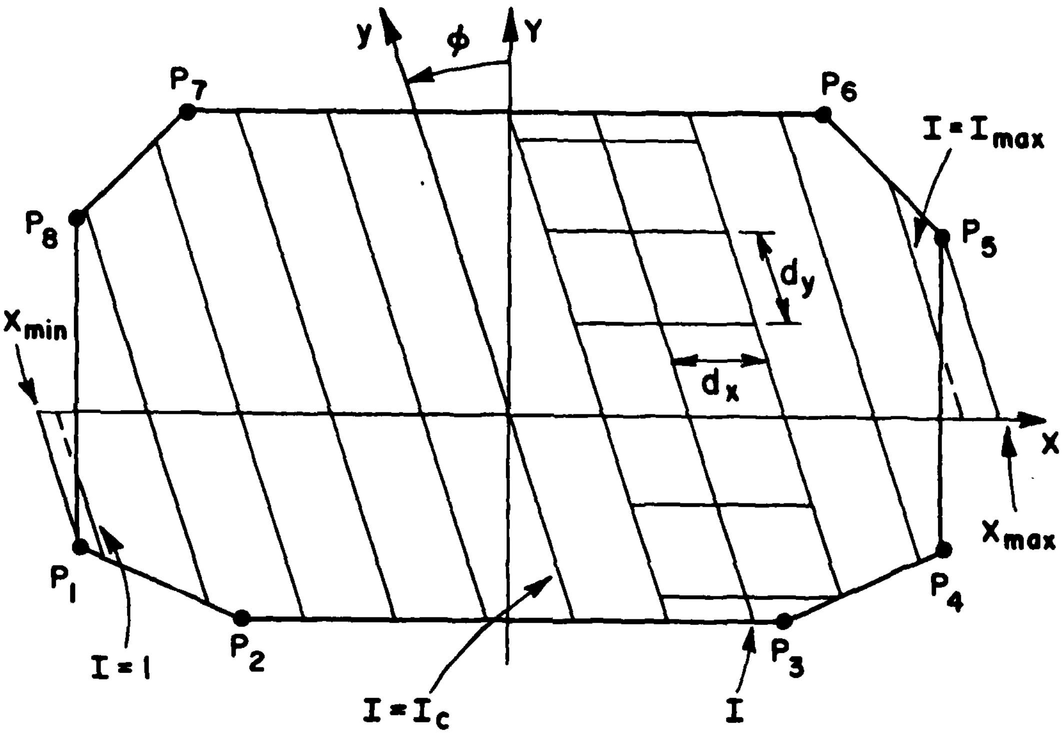


Figure 1. Rotated grid.

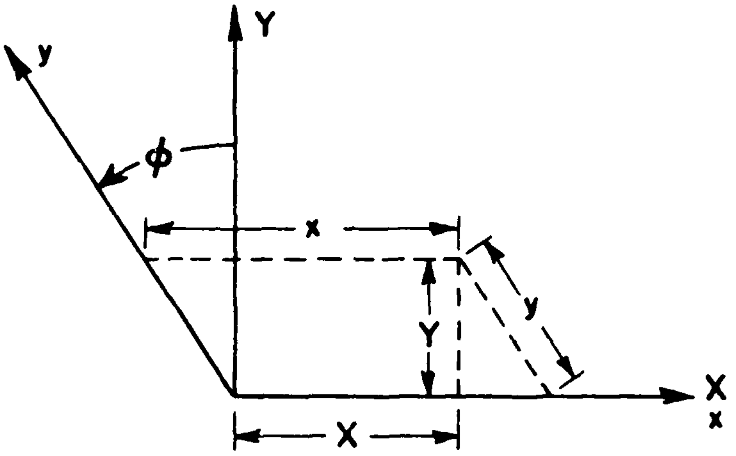


Figure 2. Coordinate transformation from principal rectangular grid to rotated grid.

METHOD

The rotating grid method sets up a nonorthogonal rotating grid as shown in Fig. 1 by rotating the principal Y-axis with $\phi$ such that the y-integrations are independent of $\theta$. Consequently, the far field pattern in the plane perpendicular to the y-axis is reduced to a one-dimensional integration.

The coordinate transformation from the principal rectangular grid (X,Y) to the rotated grid is shown in Fig. 2 and is given by

$$x = X + Y \tan\phi$$

and

$$y = Y/\cos\phi.$$

The rotated grid sizes are expressed by

$$d_X = D_X$$

$$d_y = D_Y/\cos\phi.$$

Note that the rotated grid size $d_y$ becomes quite large if the rotated angle is close to 90°. This may affect the accuracy of the result. Consequently, the rotated angle is restricted to be not greater than 45°. Thus, for PHI cuts in the interval (45°,135°), the x-axis is effectively rotated instead of the y-axis. This is done indirectly in the code by transforming rim points such that the x- and y-coordinates of the rim points are interchanged and the indices are adjusted to stay in a counterclockwise order. Then the new y-axis is rotated by an angle 90°-$\phi$ which is less than 45°. For PHI cuts in the other quadrants, a similar procedure is followed. To implement the interchange, two integer parameters related to the quadrants are used. These parameters are defined as

$$K_{QUAD} = \text{Integ. } (\phi_{+}+45°)/90°$$

and

$$L_{QUAD} = \text{Integ. } K_{QUAD}/2$$

where $\phi_{+}$ is the positive angle expression for $\phi$, i.e., $0\leq\phi_{+}<360°$. Then the interchange parameter, given by

$$CHG = (-1)^{K_{QUAD}}$$

is defined in such a way that a rim point transformation takes place when CHG<0. Note that the rotated grid sizes are also interchanged when CHG<0, i.e.,

$$d_X = D_Y$$

$$d_y = D_X/\cos\phi$$

*Values assigned to $K_{QUAD}$ and CHG are shown in Fig. 3.*

In order to maintain the correct aperture distribution over the transformed antenna aperture, the array of the aperture fields is transposed at the same time as an interchange of the x- and y-coordinates of the rim points.

Note that the parameter $L_{QUAD}$ is used to correct the sign of the phase path of the x-integration. The phase variable associated with $L_{QUAD}$ is given by

$$PG = kd_x|\cos\phi|(-1)^{L_{QUAD}} \quad .$$

In setting up the rotated grid the coordinates of each rim point $P_k$ are first transformed to rotated grid coordinates $(x_k,y_k)$. Then the reflector rim is separated into upper and lower rim sections by finding the rim points where x is minimum and maximum, respectively. Furthermore, the "vertical" grid lines of the rotated grid system are numbered from I=1 to $I_{max}$ as shown in Fig. 1. The index of the origin (IC) is also calculated for future use in the main program.

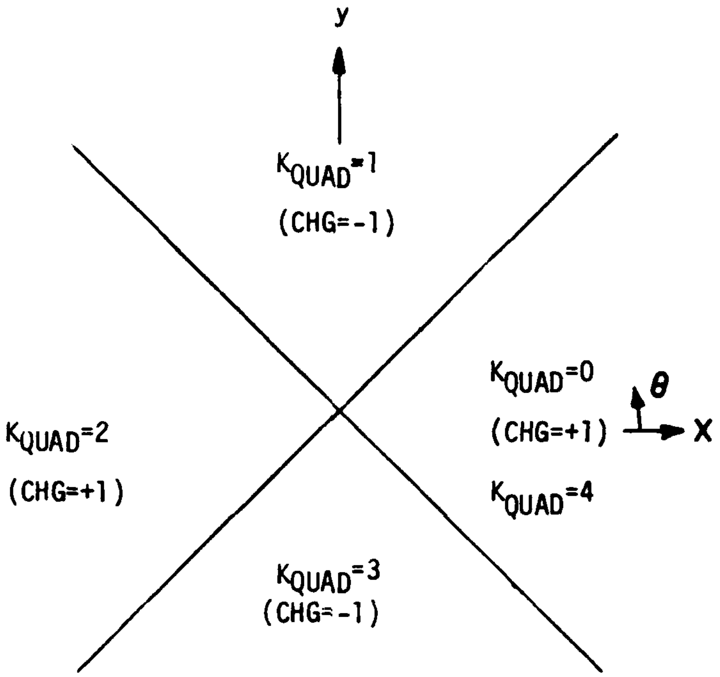


Figure 3. Quadrants for interchange parameters.

FLOW DIAGRAM

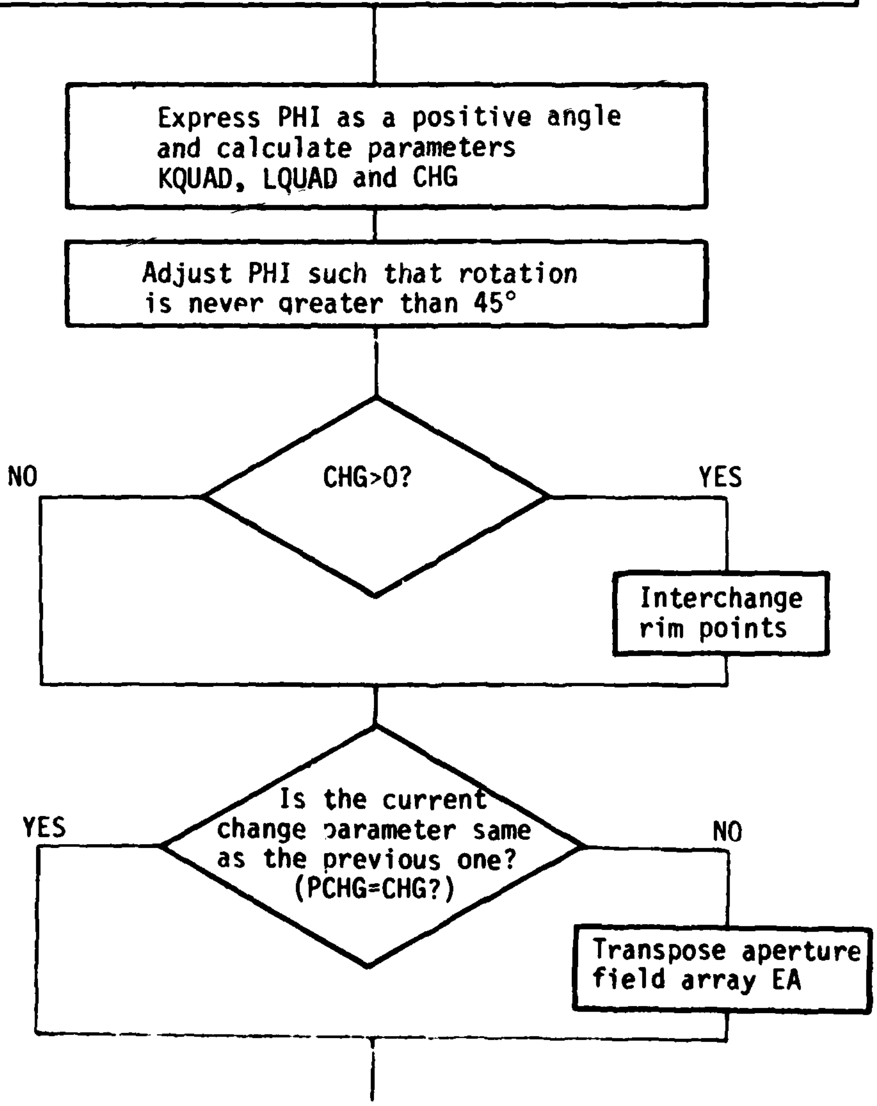
GRID(PHI,IC,IMAX)

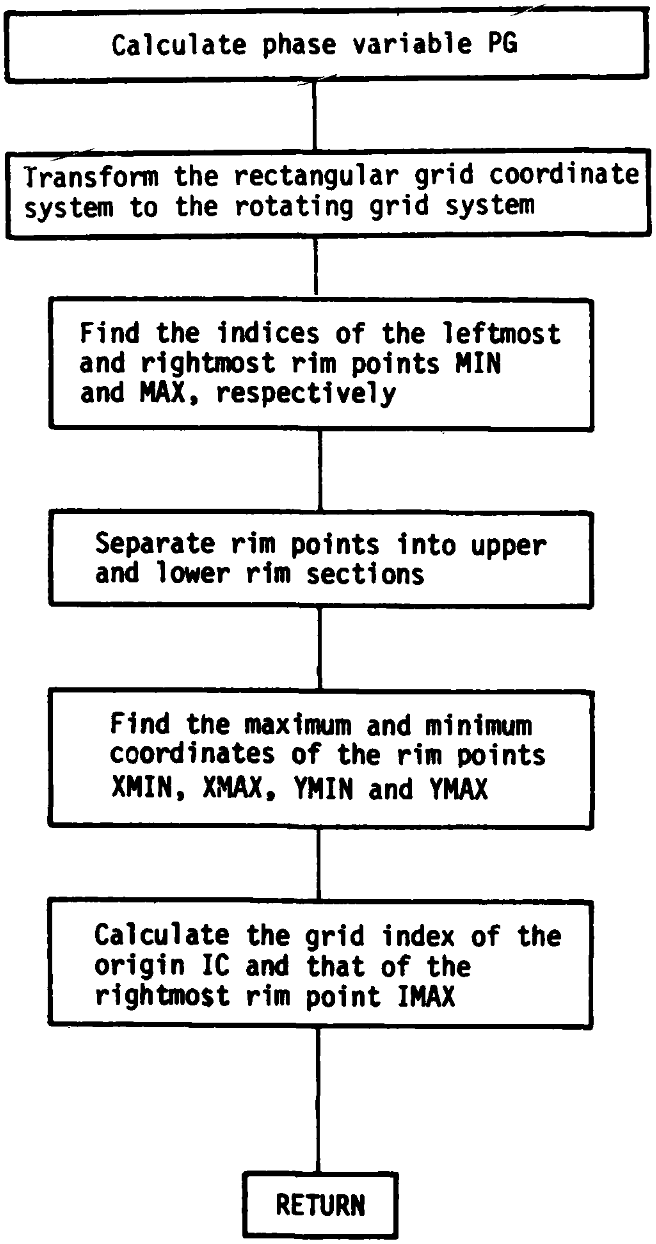
INPUT VARIABLES

| | | |
|---|---|---|
| PHI | ($\phi$) | Rotation angle |

OUTPUT VARIABLES

| | |
|---|---|
| IC | Index of the "vertical" gridline passing through the origin |
| IMAX | Index of the rightmost "vertical" gridline inside the aperture |




Calculate phase variable PG
Transform the rectangular grid coordinate system to the rotating grid system
Find the indices of the leftmost and rightmost rim points MIN and MAX, respectively
Separate rim points into upper and lower rim sections
Find the maximum and minimum coordinates of the rim points XMIN, XMAX, YMIN and YMAX
Calculate the grid index of the origin IC and that of the rightmost rim point IMAX
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| CHG | | Interchange parameter | |
| CLRIM | | Coordinates of lower rim points | |
| CURIM | | Coordinates of upper rim points | |
| EA | | Aperture Field Array | (I) |
| GRDX | ($d_x$) | Rotated horizontal grid size | (O) |
| GRDY | ($d_y$) | Rotated "vertical" grid size | (O) |
| GRIDX | ($D_X$) | Principal horizontal grid size | (I) |
| GRIDY | ($D_Y$) | Principal vertical grid size | (I) |
| IC | | Grid index of the origin | |
| IMAX | | Maximum grid indes of "vertical" grid lines after rotation | |
| KQUAD | | Integer parameter to determine if an interchange of x- and y-coordinate of rim points is required | |
| LQUAD | | Integer parameter to specify the sign of the phase argument for x-integration | |
| MAX | | Index of the rim point with maximum x-coordinate | |
| MIN | | Index of the rim point with minimum x-coordinate | |
| NLRIM | | Number of lower rim points | |
| NRIM | | Number of input rim points | |
| NURIM | | Number of upper rim points | |
| PCHG | | Previous value of CHG | |
| PG | | Variable used for phase argument | (O) |
| POS | ($\phi_+$) | Positive angle representation for PHI | |
| RIM | | Coordinates of input rim points | (I) |
| XMAX | | Maximum x-coordinate of all rotated rim points | (O) |

| | | |
|---|---|---|
| **XMIN** | **Minimum x-coordinate of all rotated rim points** | **(0)** |
| **YMAX** | **Maximum y-coordinates of all rotated rim points** | **(0)** |
| **YMIN** | **Minimum y-coordinates of all rotated rim points** | **(0)** |

**CODE LISTING**

```
      SUBROUTINE GRID(PHI,IC,IMAX)
      DIMENSION RIM(67,2),CP(67,2),CLRIM(67,2),CURIM(67,2),MIN(2),MAX(2
     )
      COMPLEX CJ,DUMMY,EA(2,50,50)
      LOGICAL LDEBUG,LTEST
      COMMON /GRID1/GRIDX,GRIDY,EA
      COMMON /GRID2/CJ,CLRIM,CURIM,RIM,PG,XMIN,XMAX,YMIN,YMAX,
     2NLRIM,NURIM,GRDX,GRDY,ACOSP,TANP,PCHG,MAXO,NRIM
      COMMON /TEST/LDEBUG,LTEST,NTEST
      COMMON /PIS/PI,TPI,DPR
      COMMON /OUT/NW
      DATA DEL/0.01/
      IF (LTEST) WRITE (NW,3)
    3 FORMAT (/T5,'TESTING SUBROUTINE GRID',//)
      DO 1 K=1,NRIM
      CP(K,1)=RIM(K,1)
      CP(K,2)=RIM(K,2)
    1 CONTINUE
      GRDX=GRIDX
      GRDY=GRIDY
      SIGN=1.
      POS=PHI
      IF (PHI.GE.0.) GO TO 2
      SIGN=-1.
      POS=PHI+360.
    2 KQUAD=(POS+45.)/90.
      LQUAD=KQUAD/2
      CHG=(-1.)**KQUAD
      IF (CHG.GT.0.) GO TO 6
      DO 4 K=1,NRIM
      KP=NRIM+1-K
      CP(K,1)=RIM(KP,2)
      CP(K,2)=RIM(KP,1)
    4 CONTINUE
      TEMG=GRDY
      GRDY=GRDX
      GRDX=TEMG
```

```
      IF (PHI.GT.180.) PHI=PHI-360.
      PHI=SIGN*(90.-ABS(PHI))
    6 IF (PCHG*CHG.GT.0.) GO TO 12
      MT=MAXO-1
      DO 11 NI=1,2
      DO 10 K=1,MT
      IT=K+1
      DO 10 JT=1,K
      DUMMY=EA(NI,JT,IT)
      EA(NI,JT,IT)=EA(NI,IT,JT)
      EA(NI,IT,JT)=DUMMY
   10 CONTINUE
   11 CONTINUE
   12 PHIR=PHI/DPR
      ACOSP=ABS(COS(PHIR))
      TANP=TAN(PHIR)
      GRDY=GRDY/ACOSP
      PG=2.*PI*ABS(ACOSP)*GRDX*(-1)**LOUAD
      PCHG=CHG
      IF (LTEST) WRITE (NW,18) GRDX,GRDY
   18 FORMAT (T10,7HGRDX = ,F5.2,5X,7HGRDY = ,F5.2,' WAVELENGTHS',
     /)
C
C                         * COORDINATE TRANSFORMATION *
C
      DO 20 K=1,NRIM
      CP(K,1)=CP(K,1)+CP(K,2)*TANP
      CP(K,2)=CP(K,2)/ACOSP
   20 CONTINUE
      CP(NRIM+1,1)=CP(1,1)
      CP(NRIM+1,2)=CP(1,2)
      CP(NRIM+2,1)=CP(2,1)
      CP(NRIM+2,2)=CP(2,2)
      MX=0
      MN=0
      IN=1
      ND=NRIM
      IF (CP(2,1).NE.CP(1,1)) GO TO 21
      IN=2
      ND=NRIM+1
   21 DO 25 I=IN,ND
      DX1=CP(I+1,1)-CP(I,1)
      DX2=CP(I+2,1)-CP(I+1,1)
      IF (ABS(DX1).LT.0.01.OR.ABS(DX2).LT.0.01) GO TO 22
      IF (DX1*DX2.GT.0.) GO TO 25
   22 IF (DX1.GT.DX2) GO TO 24
      MN=MN+1
      MIN(MN)=I+1
      GO TO 25
   24 MX=MX+1
      MAX(MX)=I+1
   25 CONTINUE
```

```
C
C                       *FIND UPPER AND LOWER RIM POINT SETS*
C
      IF (MN .EQ. 1) MIN(2)=MIN(1)
      IF (MX .EQ. 1) MAX(2)=MAX(1)
      NLRIM=MAX(1)-MIN(2)+1
      IF (NLRIM .LE. 0) NLRIM=NLRIM+NRIM
      NURIM =MIN(1)-MAX(2)+1
      IF (NURIM .LE. 0) NURIM=NURIM+NRIM
      DO 30 K=1,NLRIM
      I=MIN(2)+K-1
      IF (I .GT. NRIM) I=I-NRIM
      CLRIM(K,1)=CP(I,1)
      CLRIM(K,2)=CP(I,2)
   30 CONTINUE
      DO 32 K=1,NURIM
      I=MIN(1)-K+1
      IF (I .LE. 0) I=I+NRIM
      CURIM(K,1)=CP(I,1)
      CURIM(K,2)=CP(I,2)
   32 CONTINUE
      IF (.NOT.LTEST) GO TO 38
      WRITE (NW,35)
   35 FORMAT (//T10,'LOWER RIM POINT COORDINATES',//)
      WRITE (NW,33) (K,(CLRIM(K,I),I=1,2),K=1,NLRIM)
      WRITE (NW,37)
   37 FORMAT (//T10,'UPPER RIM POINT COORDINATES',//)
      WRITE (NW,33) (K,(CURIM(K,I),I=1,2),K=1,NURIM)
   33 FORMAT (20(T10,I5,2F10.2,/))
   38 CONTINUE
      GRSQ=GRDX*GRDY
      YMIN=CLRIM(1,2)
      YMAX=CURIM(1,2)
      N1=NLRIM-1
      DO 40 K=1,N1
      YLKP=CLRIM(K+1,2)
      YUKP=CURIM(K+1,2)
      IF (YUKP.GT.YMAX) YMAX=YUKP
      IF (YLKP.LT.YMIN) YMIN=YLKP
   40 CONTINUE
      XMIN=CLRIM(1,1)
      XMAX=CLRIM(NLRIM,1)
      FIC=-XMIN/GRDX+DEL
      IC=FIC+1
      IF (FIC.LT.-1.) IC=IC-1
      FI=XMAX/GRDX+DEL
      IMAX=FI+IC
      IF (LTEST) WRITE (NW,50) XMIN,XMAX,YMIN,YMAX
   50 FORMAT (T5,6HXMIN =,F10.3,5X,6HXMAX =,F10.3,/T5,6HYMIN =,
     2F10.3,5X,6HYMAX =,F10.3,/)
      RETURN
      END
```

## SUBROUTINE GTD

### PURPOSE

To use the Geometrical Theory of Diffraction (GTD) to calculate the edge, corner and slope diffraction fields in the wide angle sidelobe and backlobe regions for the reflector antenna patterns. For near field calculations, GTD is sometimes used for the whole region including the near axis region if the near field points are close to the aperture.

### METHOD

This subroutine calculates and sums the diffracted field contribution for each rim segment. If the contribution for a rim segment is expected to be negligible, the subroutine skips to the next rim segment without further calculation. The subroutine uses BDLOW and BDHI for this test as discussed below. To determine if the diffraction from rim segment ME is significant, the cosine of the diffracted cone angle $\beta_0$ is calculated by taking the dot product of the edge unit $\hat{V}$ and the diffracted ray unit vector $\hat{d}$, and then is compared with the upper and lower bounds BDHI and BDLOW, respectively, of the diffracted angle. The diffraction contribution from rim segment ME is added only if

$$\mathrm{BDLOW} < \mathrm{DV} < \mathrm{BDHI}$$

where

$$\mathrm{DV} = \hat{d}\cdot\hat{V} = \cos\beta_0$$

$$\mathrm{BDLOW} = \begin{cases} \mathrm{BD(ME,1)} & \text{if edge diffraction only} \\ \mathrm{BD(ME,1)-0.5} & \text{if corner diffraction included} \end{cases}$$

$$\mathrm{BDHI} = \begin{cases} \mathrm{BD(ME,2)} & \text{if edge diffraction only} \\ \mathrm{BD(ME,2)+0.5} & \text{if corner diffraction included} \end{cases}$$

and BD is defined and calculated in subroutine GEOM.

Note that for the near field, the unit vector $\hat{d}$ is approximated for this purpose by taking the midpoint $X_M$ of the edge instead of the diffraction point $X_D$ which is calculated next.

If the contribution from the rim segment is significant, the coordinates of the diffraction point $X_D$ are computed by calling subroutine DFPTWD. The diffracted ray unit vector $\hat{d}$ for near field is recalculated by using the actual diffraction point $X_D$ as

$$\hat{d} = \frac{\bar{X}_N - \bar{X}_D}{|\bar{X}_N - \bar{X}_D|}$$

where $X_N$ is the near field point.

If the diffraction point lies on the rim segment as shown in Fig. 1a (LDIF=true), both edge diffraction and corner diffraction are included and the incident vector $\hat{VI}$ is calculated to the diffraction point $X_D$. If the diffraction point does not lie on the rim segment as shown in Fig. 1b (LDIF=false), there are only contributions from corner diffraction and the incident vector $\hat{VI}$ is calculated to the nearest corner.

The incident and diffraction angles are calculated by using the orthogonal unit vectors $\hat{V}$, $\hat{VN}$ and $\hat{VP}$ of the rim segment ME. These unit vectors are computed and stored by subroutine GEOM. The incident and diffracted PHI angles* are given by

$$\phi' = \tan^{-1}\left(\frac{-\hat{VI}\cdot\hat{VN}}{-\hat{VI}\cdot\hat{VP}}\right)$$

and

$$\phi = \tan^{-1}\left(\frac{\hat{d}\cdot\hat{VN}}{\hat{d}\cdot\hat{VP}}\right)$$

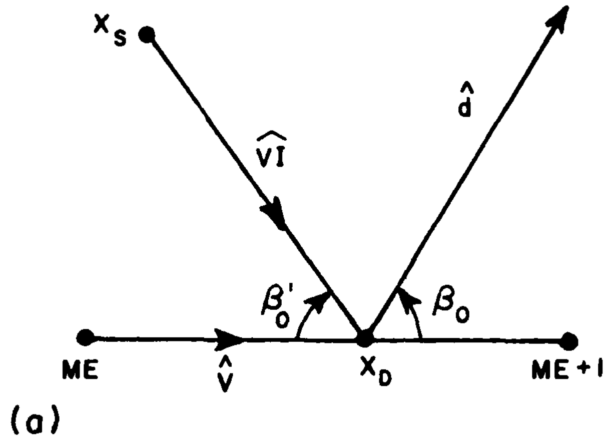
Note that the diffracted field from one rim segment is shadowed by the reflector over a certain range of $\theta$ as shown in Fig. 2. The subroutine will skip to the next rim segment if $\theta$ falls in this range, i.e., if

$$\phi \geq 0 \text{ and } \theta > \theta_B$$

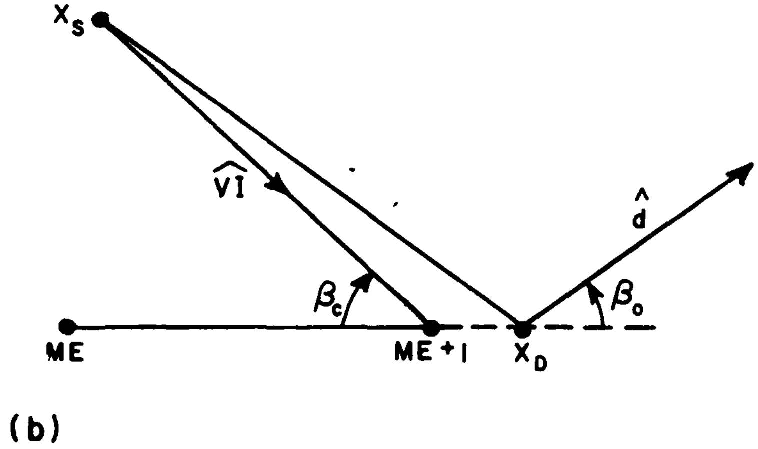
where $\theta_B$ is the diffracted shadow boundary angle calculated in subroutine SBDY.

---

*Note that $\phi$ and $\phi'$ are used in this section for the wedge diffraction angles as shown in Fig. 3a. They should not be confused with the phi coordinate angles PHI and PHIP which represent the field point and the feed observation directions, respectively.



a) Diffraction point inside the edge (edge diffraction + corner diffraction).



b) Diffraction point outside the edge: (corner diffraction only).

Figure 1. Geometry for edge and corner diffracted fields.

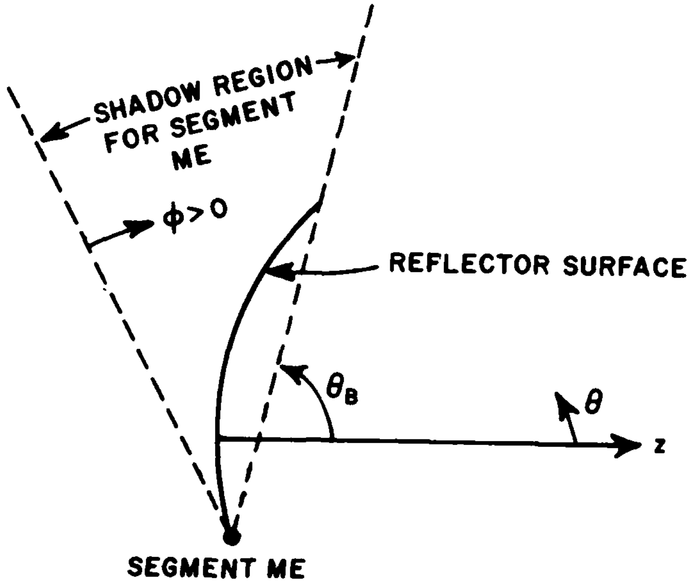


Figure 2. Geometry for diffracted shadow boundary for rim segment ME.

If θ is outside this shadow region, the unit vectors $\hat{\phi}'$, $\hat{\phi}$, $\hat{\beta}_o'$ and $\hat{\beta}_o$ of the ray fixed coordinate system are calculated. These unit vectors are defined by

$$\hat{\phi}' = -\widehat{VP}\sin\phi' + \widehat{VN}\cos\phi'$$

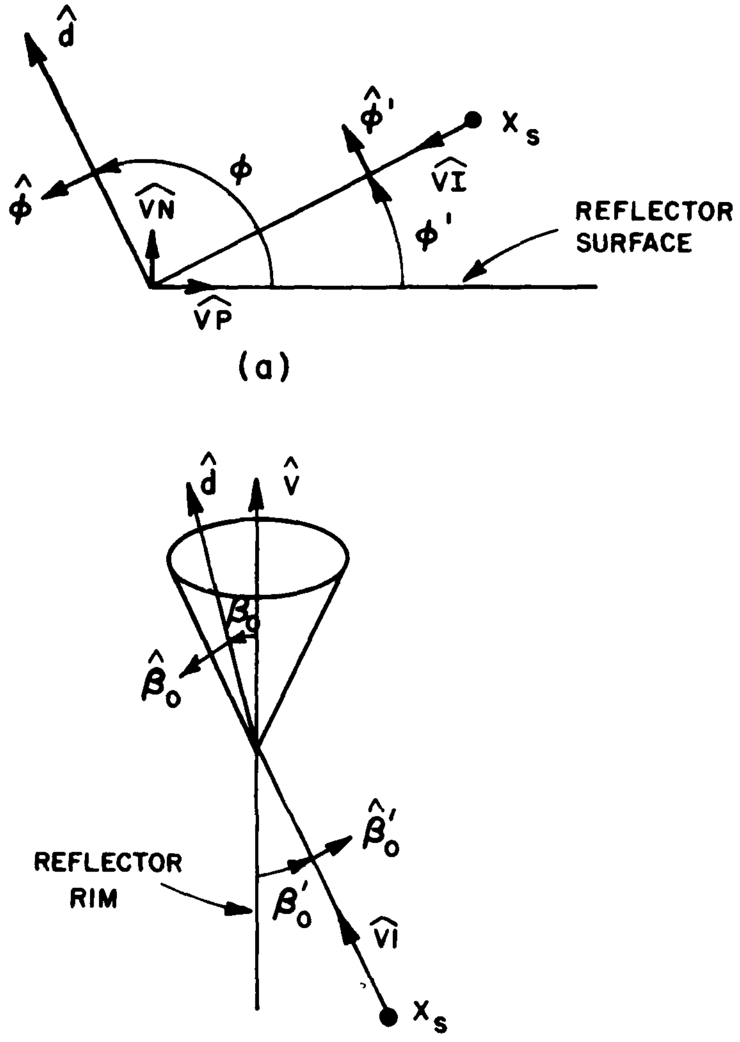
$$\hat{\phi} = -\widehat{VP}\sin\phi + \widehat{VN}\cos\phi$$

$$\hat{\beta}_o' = \hat{\phi}' \times \widehat{VI}$$

and

$$\hat{\beta}_o = \hat{\phi} \times \hat{d}$$

as illustrated in Fig. 3.



(a)

(b)

Figure 3a,b. Geometry for three dimensional diffraction of a half plane.

To determine the incident field at the diffraction point $X_D$, the spherical coordinate angles PSI and PHIP corresponding to the feed pattern direction are calculated. Then the feed pattern value incident on the diffraction point is calculated by calling the subroutine FEED. The rectangular components $E^i_x$, $E^i_y$ and $E^i_z$ of the feed pattern are then calculated in the subroutine FPOL. These are then transformed to perpendicular and parallel components $E^i_\perp$ and $E^i_\parallel$ in the ray fixed coordinate system.

For slope diffraction the slope of the incident field at the diffraction point $X_D$ is used. The slope of the incident field is calculated from two adjacent values of the feed pattern. The perpendicular and parallel components of the slope of the incident field $\partial E^i_\perp/\partial n$ and $\partial E^i_\parallel/\partial n$ are calculated in the same way as for the incident field by using the subroutine FPOL.

The distance parameters L and the spread factors A(S) of the diffracted fields are given below.

For far field

$$L = S' \sin^2\beta_0$$

and

$$A(S) = \sqrt{S'} \quad .$$

For near field

$$L = \frac{S'S}{S+S'} \sin^2\beta_0$$

and

$$A(S) = \sqrt{\frac{S'}{S(S+S')}}$$

where

$\beta_0 = \sin^{-1}|\hat{d}x\hat{V}|$ is the half diffracted cone angle (see Fig. 3b) and

S' and S are the distances from the diffraction point to the source point and the field point respectively.

For corner diffraction as shown in Fig. 4, the spread factor $A_C(S) = \frac{1}{S_S}$ has the form of a spherical wave, since the corner is treated as a point source to radiate the corner diffracted field. The distance parameter is given by

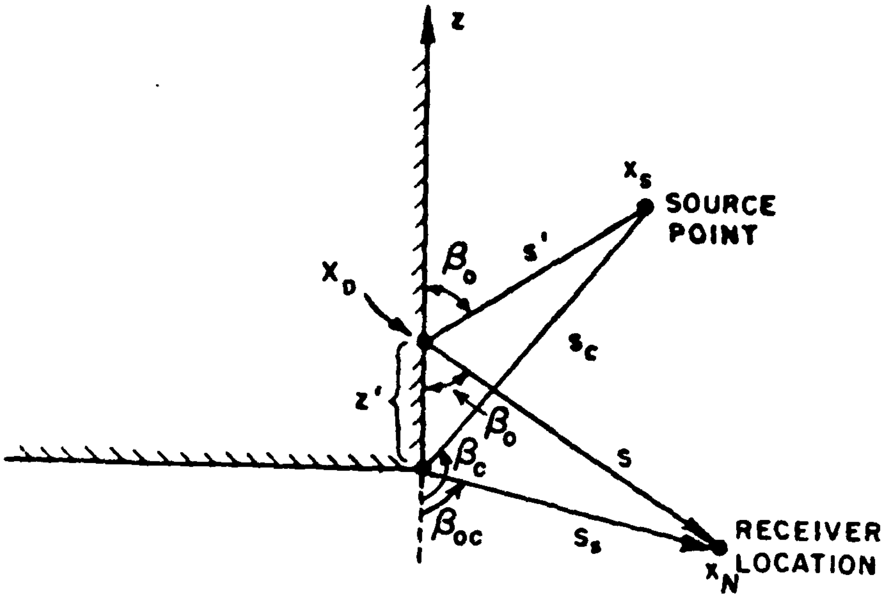


Figure 4. Geometry for corner diffraction problem for near field.

$$L_C = S_C \qquad \text{for far field}$$

$$L_C = \frac{S_C \cdot S_S}{S_C + S_S} \qquad \text{for near field}$$

where $S_C$ and $S_S$ are the distances from the corner to the source point and the field point, respectively.

The corner diffracted field also depends on the corner angles $\beta_C$ and $\beta_{OC}$ (see Fig. 4) as defined by

$$\beta_C = \cos^{-1}|\hat{V} \cdot \hat{V} I_C|$$

and

$$\beta_{OC} = \begin{cases} \cos^{-1}|\hat{d} \cdot \hat{V}| & \text{for far field} \\ \cos^{-1}\left|\dfrac{(\bar{X}_N - \bar{X}_{ME}) \cdot \hat{V}}{S_S}\right| & \text{for near field} \end{cases}$$

where $VI_c$ is the incident ray unit vector at the corner $X_{ME}$, and is calculated in subroutine GEOM.

Two variables which are used in calculating the corner diffraction coefficients are defined by

$$DEL(I) = k\ L_c\ a(\beta_{oc}+\beta_c)$$

and

$$CORN(I) = -\frac{\sin\beta_c e^{-j\frac{\pi}{4}}}{2\pi(\cos\beta_{oc}+\cos\beta_c)}\ F|kL_c a(\beta_{oc}+\beta_c)|\ \sqrt{\frac{S'}{S_c}}\ e^{-jk(S_c-S')}$$
$$\frac{e^{-jks_s}}{S_s}$$

where I=1,2 representing the first and second corners of the edge ME, respectively.

Next the subroutine DCHP is called to calculate the edge diffraction coefficients $D_s$,$D_h$; the slope diffraction coefficients $\partial D_s/\partial\phi$, $\partial D_h/\partial\phi$; the corner diffraction coefficients $B_s$, $B_h$ and the slope corner diffraction coefficients $\partial B_s/\partial\phi'$, $\partial B_h/\partial\phi'$.

Thus the diffracted field is given by

$$\begin{Bmatrix} E^d_\parallel \\ E^d_\perp \end{Bmatrix} = -\begin{Bmatrix} D_s E^i_\parallel \\ D_h E^i_\perp \end{Bmatrix}\ A(S)\ e^{jk\gamma} \qquad ,$$

the slope diffracted field by

$$\begin{Bmatrix} E^s_\parallel \\ E^s_\perp \end{Bmatrix} = \begin{Bmatrix} \frac{\partial D_s}{\partial\phi'}\ \frac{\partial E^i_\parallel}{\partial n} \\ \frac{\partial D_h}{\partial\phi'}\ \frac{\partial E^i_\perp}{\partial n} \end{Bmatrix}\ \frac{A(S)}{C_p}\ e^{jk\gamma} \qquad ,$$

the corner diffracted field by

$$\begin{Bmatrix} E_{\parallel}^{c} \\ E_{\perp}^{c} \end{Bmatrix} = - \begin{Bmatrix} B_s E_{\parallel}^{i} \\ B_h E_{\perp}^{i} \end{Bmatrix}$$

and the slope corner diffracted field by

$$\begin{Bmatrix} E_{\parallel}^{sc} \\ E_{\perp}^{sc} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial B_s}{\partial \phi'} & \dfrac{\partial E_{\parallel}^{i}}{\partial n} \\ \dfrac{\partial B_h}{\partial \phi'} & \dfrac{\partial E_{\perp}^{i}}{\partial n} \end{Bmatrix} \frac{1}{C_p}$$

where $C_p = jkS'\sin\beta_o$ and $\gamma$ is the phase factor which refers the contribution from each rim segment to the origin. The total diffracted field for segment ME is summed in terms of perpendicular and parallel components for that segment as expressed by

$$E^D = E^d + E^s + (E^c + E^{sc})_{ME} + (E^c + E^{sc})_{ME+1} \quad .$$

The diffracted field from segment ME is then transformed to rectangular components in the reflector coordinate system so that the total diffracted field from the reflector rim can be summed.

For near field calculations, the geometrical optics reflected field must also be included in the total field if the observation point is inside the projected aperture. The reflected field is calculated by using interpolation between the aperture field values at the adjacent grid points (see Fig. 5) as given by

$$E^R = \left[ E^a(M,N) \left( 1 - \frac{\Delta x}{D_x} - \frac{\Delta y}{D_y} \right) + E^a(M+1,N) \frac{\Delta x}{D_x} + E^a(M,N+1) \frac{\Delta y}{D_y} \right] e^{-jkz}$$

where z is the distance from the observation point to the aperture plane.

If the field point is in the spillover region, the feed spillover field is calculated and added to the total field.

Finally, for far field calculations or for near field calculation with constant range, the total field is converted to principal and cross polarized components as referred to the polarization of the field components from a Huygen's source. For near field calculations with constant z, the field is still expressed in rectangular components.

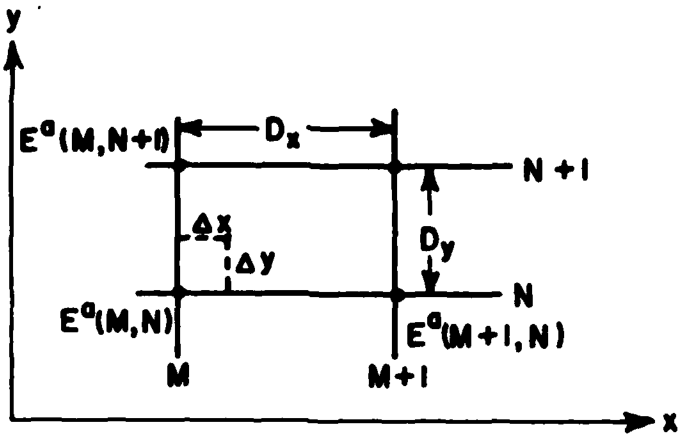
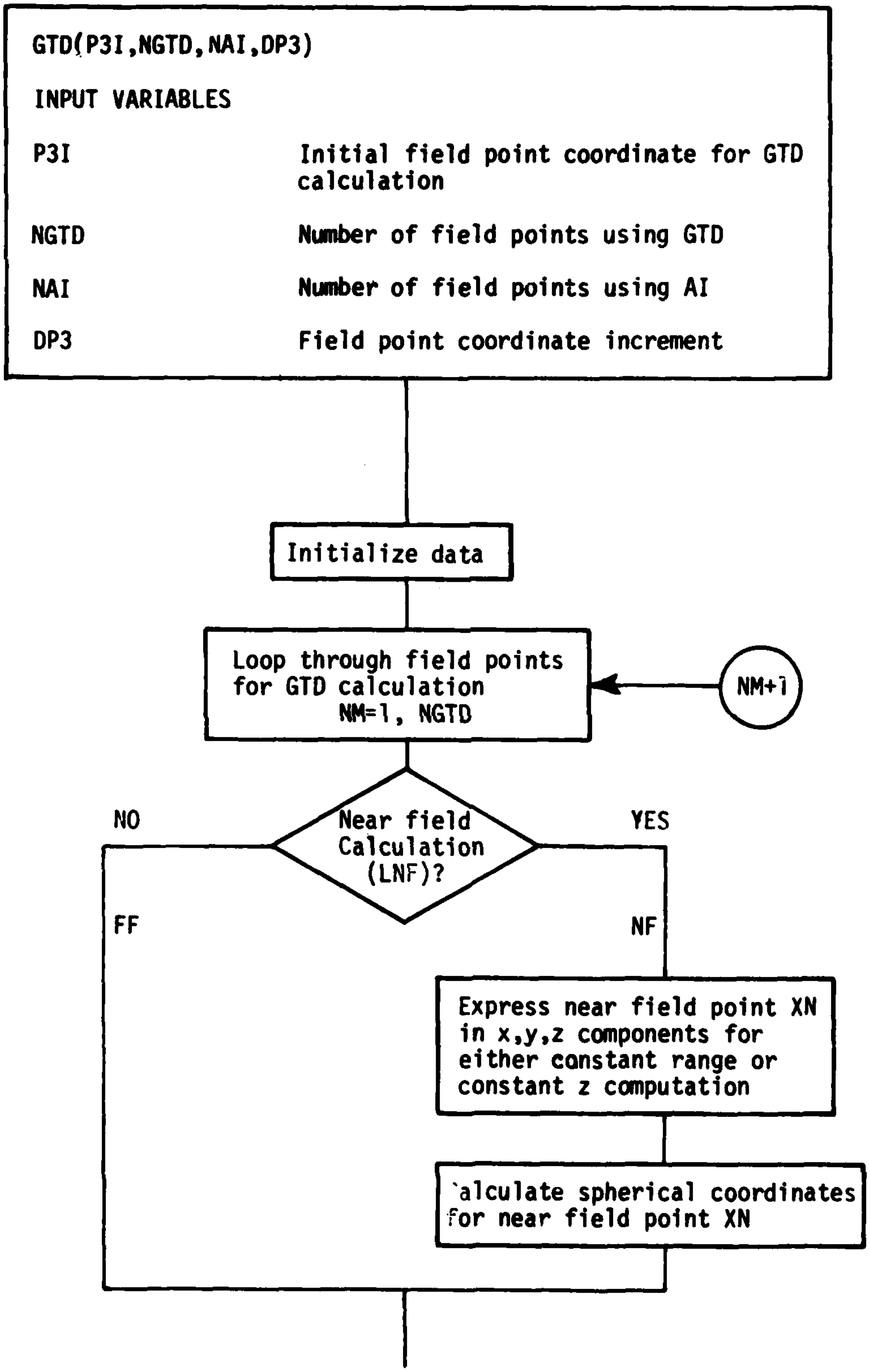


Figure 5. Interpolation of aperture field.

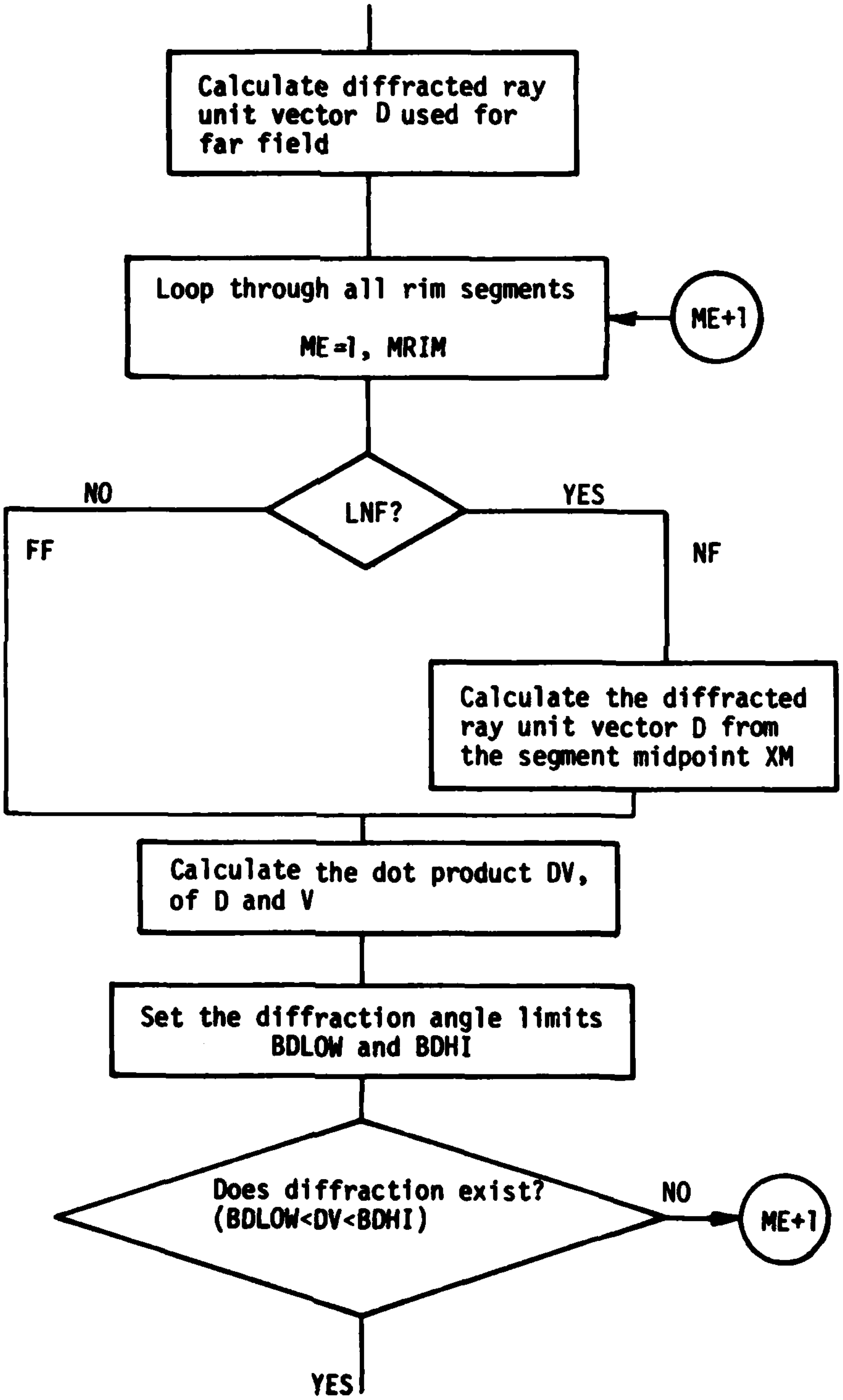
FLOW DIAGRAM

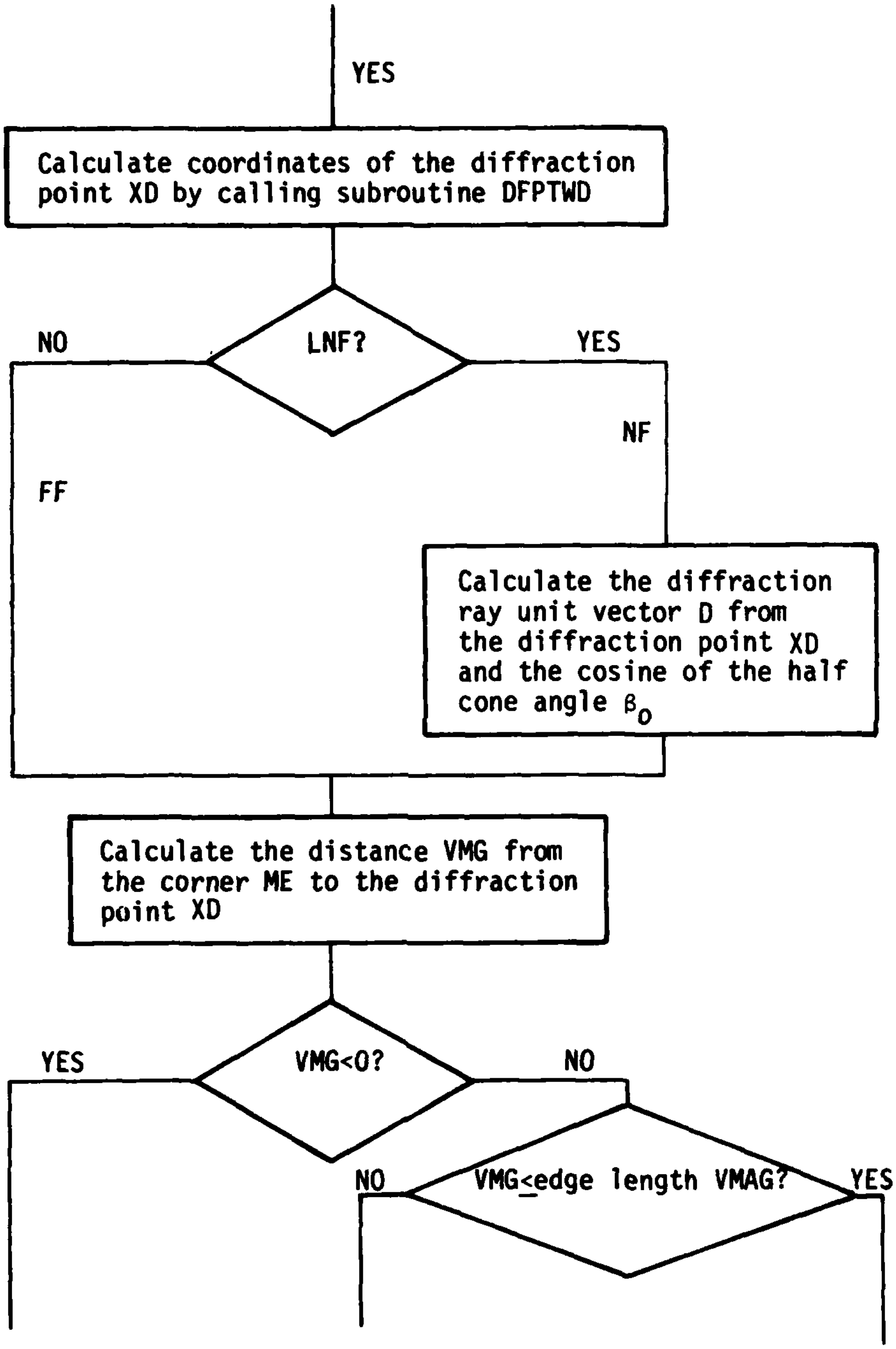


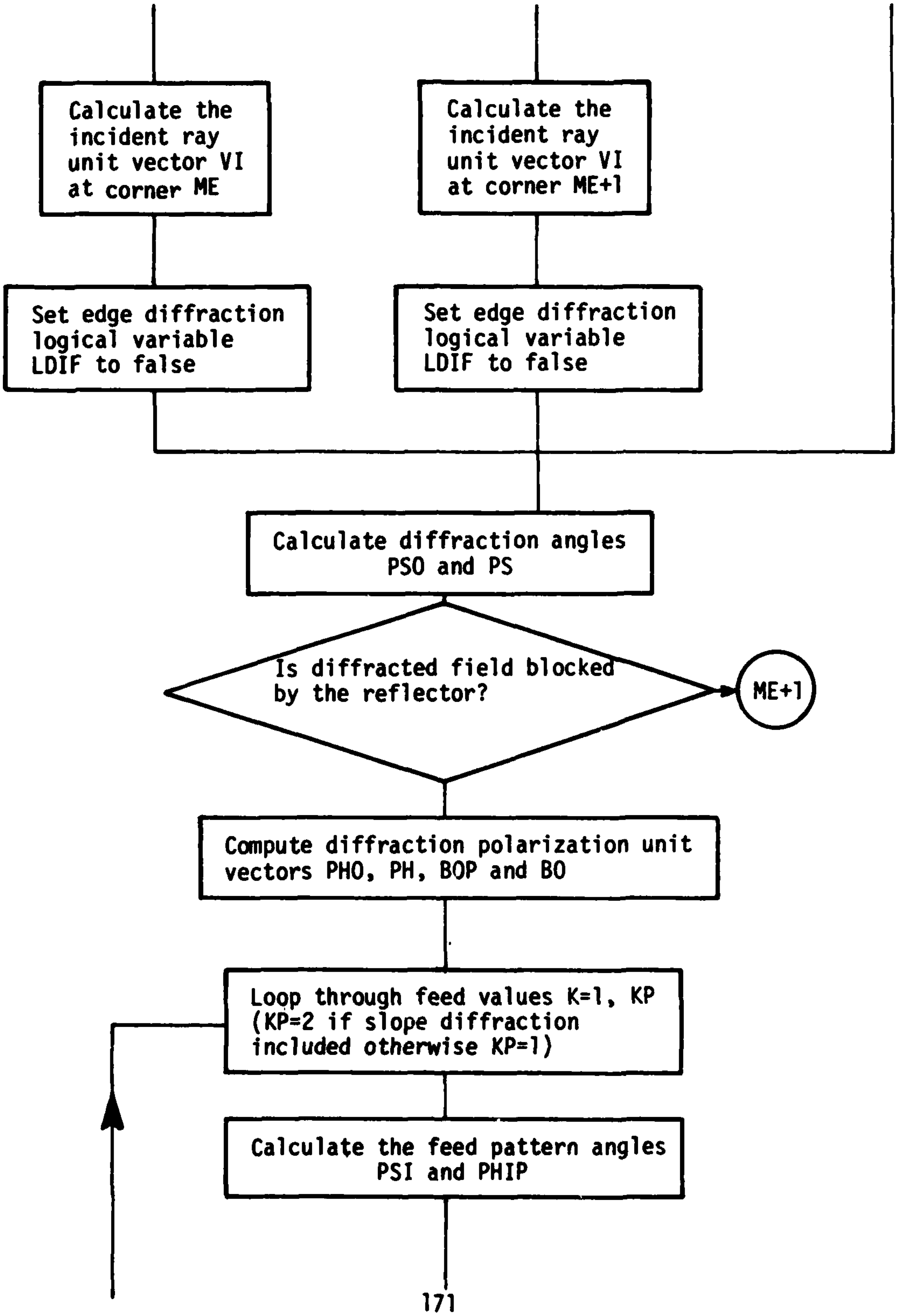
GTD(P3I,NGTD,NAI,DP3)

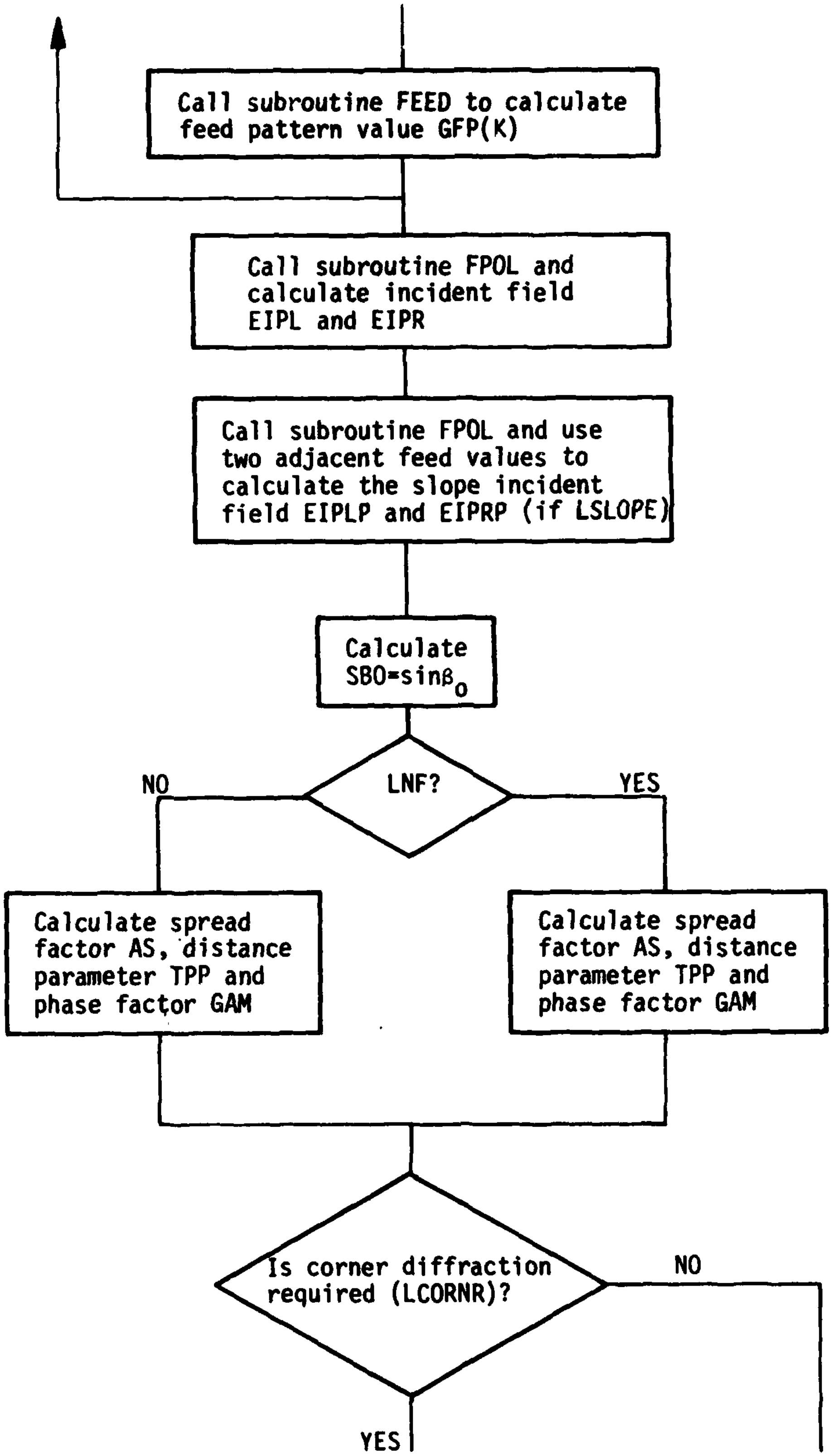
INPUT VARIABLES

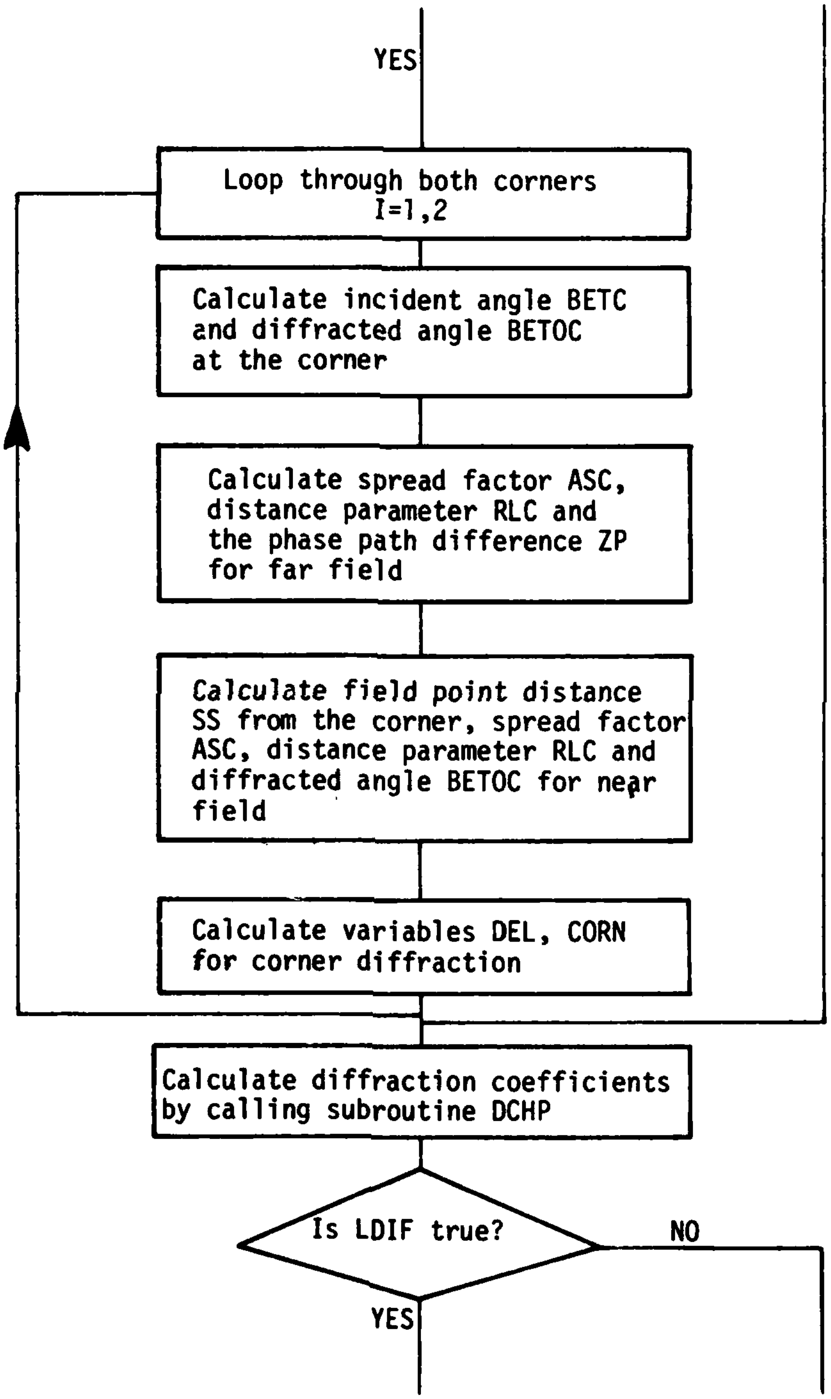
| | |
|---|---|
| P3I | Initial field point coordinate for GTD calculation |
| NGTD | Number of field points using GTD |
| NAI | Number of field points using AI |
| DP3 | Field point coordinate increment |

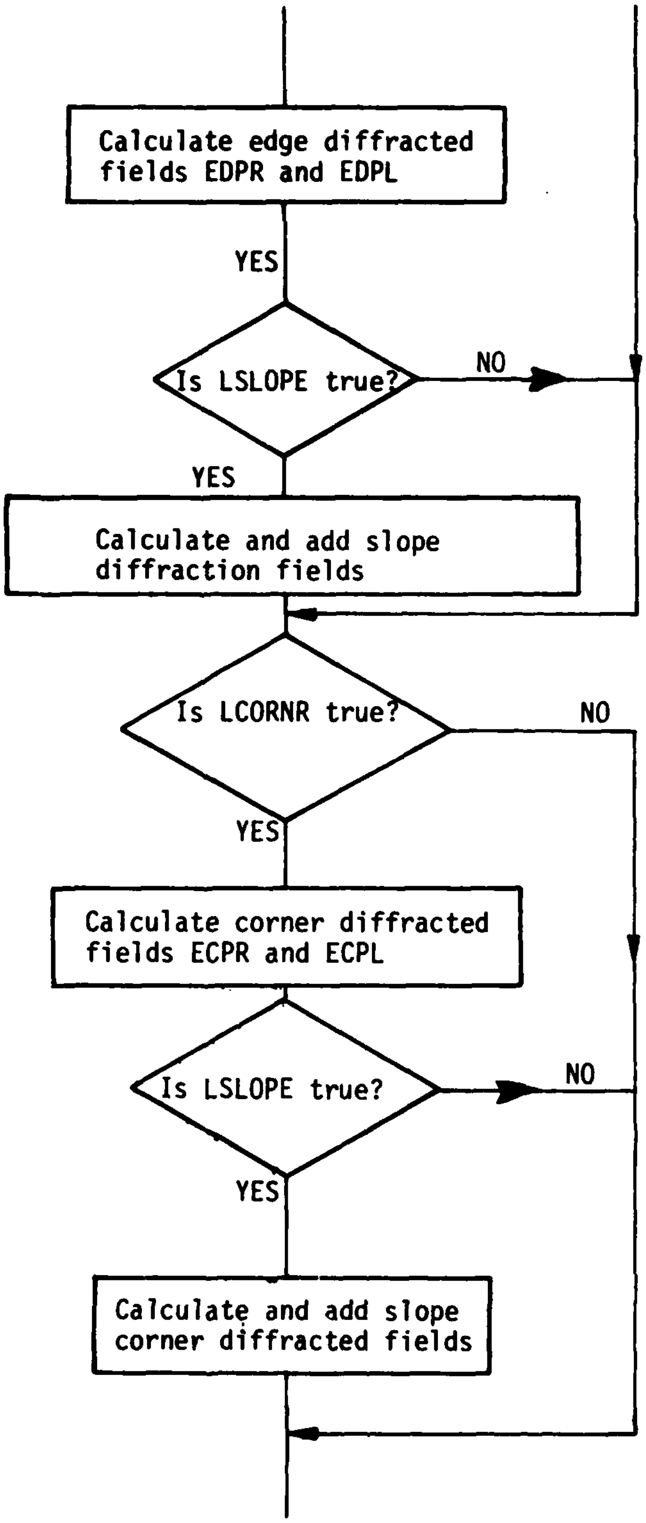

Calculate diffracted ray unit vector D used for far field
Loop through all rim segments
ME=1, MRIM
ME+1
LNF?
NO
YES
FF
NF
Calculate the diffracted ray unit vector D from the segment midpoint XM
Calculate the dot product DV, of D and V
Set the diffraction angle limits BDLOW and BDHI
Does diffraction exist? (BDLOW<DV<BDHI)
NO
ME+1
YES

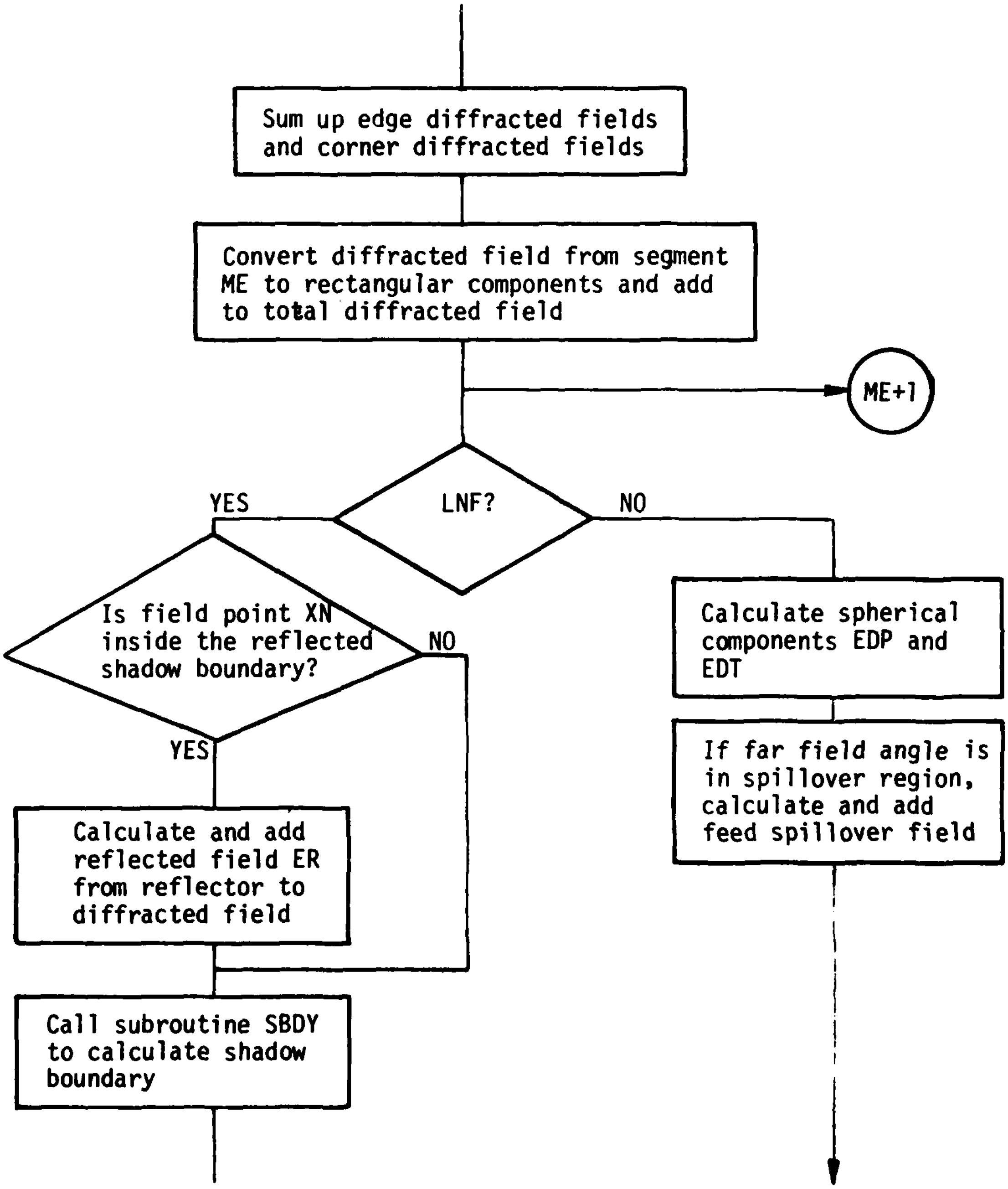

YES
Calculate coordinates of the diffraction point XD by calling subroutine DFPTWD
LNF?
NO
YES
NF
FF
Calculate the diffraction ray unit vector D from the diffraction point XD and the cosine of the half cone angle $\beta_0$
Calculate the distance VMG from the corner ME to the diffraction point XD
VMG<0?
YES
NO
VMG≤edge length VMAG?
NO
YES

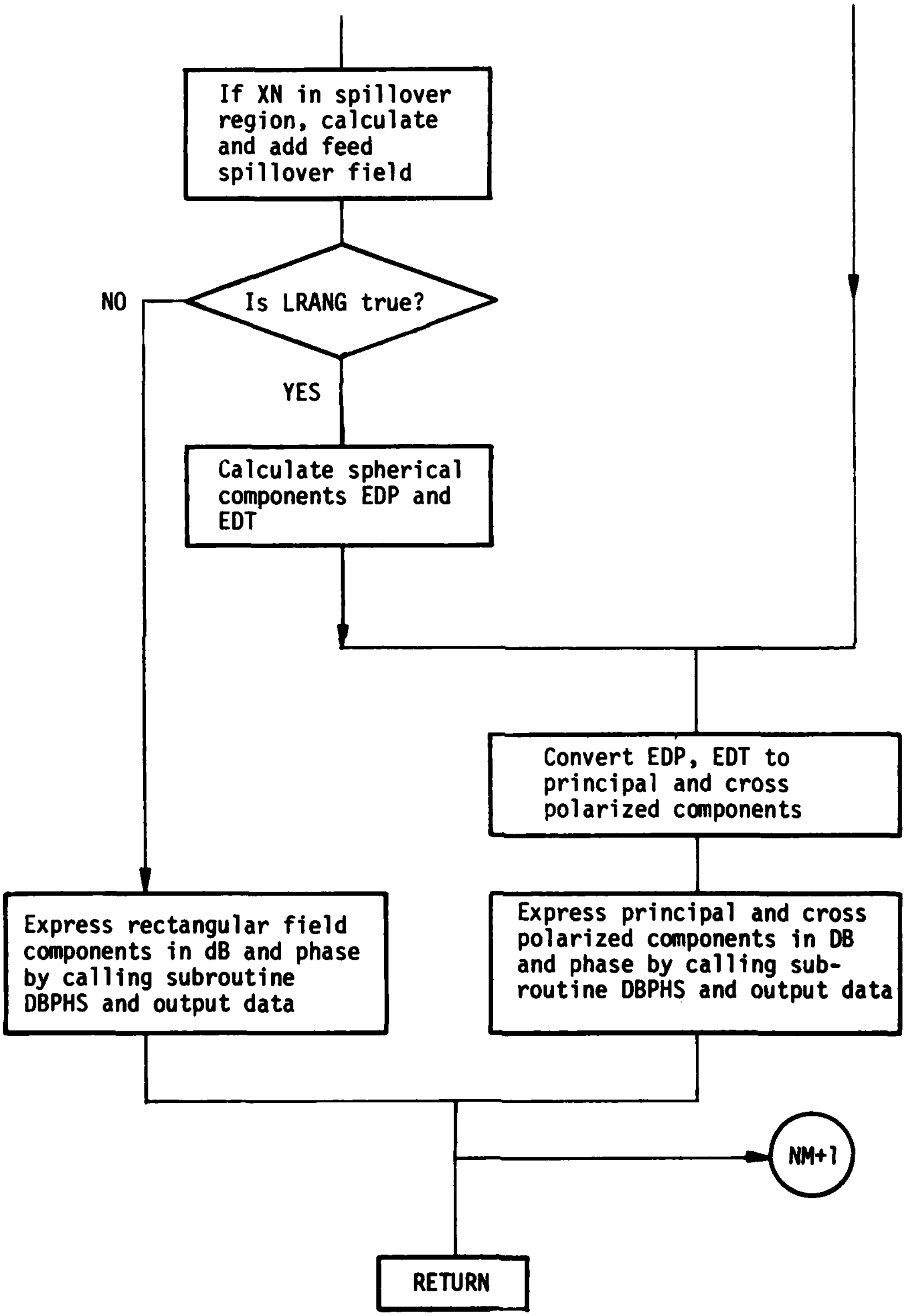

Calculate the incident ray unit vector VI at corner ME
Calculate the incident ray unit vector VI at corner ME+1
Set edge diffraction logical variable LDIF to false
Set edge diffraction logical variable LDIF to false
Calculate diffraction angles PSO and PS
Is diffracted field blocked by the reflector?
ME+1
Compute diffraction polarization unit vectors PHO, PH, BOP and BO
Loop through feed values K=1, KP (KP=2 if slope diffraction included otherwise KP=1)
Calculate the feed pattern angles PSI and PHIP


Call subroutine FEED to calculate feed pattern value GFP(K)
Call subroutine FPOL and calculate incident field EIPL and EIPR
Call subroutine FPOL and use two adjacent feed values to calculate the slope incident field EIPLP and EIPRP (if LSLOPE)
Calculate SBO=sinβ$_o$
LNF?
NO
YES
Calculate spread factor AS, distance parameter TPP and phase factor GAM
Calculate spread factor AS, distance parameter TPP and phase factor GAM
Is corner diffraction required (LCORNR)?
NO
YES


YES
Loop through both corners I=1,2
Calculate incident angle BETC and diffracted angle BETOC at the corner
Calculate spread factor ASC, distance parameter RLC and the phase path difference ZP for far field
Calculate field point distance SS from the corner, spread factor ASC, distance parameter RLC and diffracted angle BETOC for near field
Calculate variables DEL, CORN for corner diffraction
Calculate diffraction coefficients by calling subroutine DCHP
Is LDIF true?
NO
YES


Calculate edge diffracted fields EDPR and EDPL
YES
Is LSLOPE true?
NO
YES
Calculate and add slope diffraction fields
Is LCORNR true?
NO
YES
Calculate corner diffracted fields ECPR and ECPL
Is LSLOPE true?
NO
YES
Calculate and add slope corner diffracted fields


Sum up edge diffracted fields and corner diffracted fields
Convert diffracted field from segment ME to rectangular components and add to total diffracted field
ME+1
LNF?
YES
NO
Is field point XN inside the reflected shadow boundary?
NO
YES
Calculate and add reflected field ER from reflector to diffracted field
Call subroutine SBDY to calculate shadow boundary
Calculate spherical components EDP and EDT
If far field angle is in spillover region, calculate and add feed spillover field


If XN in spillover region, calculate and add feed spillover field
Is LRANG true?
NO
YES
Calculate spherical components EDP and EDT
Convert EDP, EDT to principal and cross polarized components
Express rectangular field components in dB and phase by calling subroutine DBPHS and output data
Express principal and cross polarized components in DB and phase by calling sub-routine DBPHS and output data
NM+1
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| AS | $(A(S))$ | Spread factor for diffracted field | |
| ASC | $(A_c(S))$ | Spread factor for corner diffracted field | |
| BD | | Bounds of the permissible range for diffraction angle | (I) |
| BDEL | | Adjustment to the bounds for corner diffraction | |
| BDHI | | Upper bound for diffraction angle after adjusted | |
| BDLOW | | Lower bound for diffraction angle after adjusted | |
| BETC | $(\beta_c)$ | Incident angle at the corner | |
| BETOC | $(\beta_{oc})$ | Diffraction angle at the corner | |
| BO | $(\hat{\beta}_o)$ | Rectangular components of the unit vector in the direction of increasing diffraction cone angle $\beta_o$ | |
| BOP | $(\hat{\beta}_o')$ | Rectangular components of the unit vector in the direction of increasing incident angle $\beta_o'$ | |
| CBOC | $(\cos\beta_c)$ | Cosine of angle BETC | |
| CORN | | Variable used for calculating the corner diffraction coefficients | |
| CP | $(C_p)$ | Variable used for slope diffraction | |
| CPH | $(\cos\phi)$ | Cosine of diffraction angle PS | |
| CPHO | $(\cos\phi')$ | Cosine of incident angle PSO | |
| D | $(\hat{d})$ | Rectangular components of the unit diffracted ray vector | |
| DEL | | Variables used for corner diffraction | |
| DLVI | | Increment of the incident ray vector VI along the normal vector VN | |
| DMAG | | Diffracted ray path length for near field | |

| | | | |
|---|---|---|---|
| DV | | Dot product of the unit vectors D and V | |
| EA | ($E^a$) | x and y components of the aperture fields | (I) |
| ECPL | ($E^c_{\parallel}$) | Parallel component of the corner diffracted field | |
| ECPR | ($E^c_{\perp}$) | Perpendicular component of the corner diffracted field | |
| ED | | Rectangular components of the total diffracted field from the segment ME. | |
| EDP | | PHI component of the total diffracted field from all the segments. Also used for cross polarization component | (O) |
| EDPL | ($E^d_{\parallel}$) | Parallel component of the diffracted field | |
| EDPR | ($E^d_{\perp}$) | Perpendicular component of the diffracted field | |
| EDT | | Theta component of the total diffracted field from all the segments. Also used for principal polarization component | (O) |
| EDX | | x component of the total diffracted field | (O) |
| EDY | | y component of the total diffracted field | (O) |
| EDZ | | z component of the total diffracted field | (O) |
| ERX | | x component of the reflected field | |
| ERY | | y component of the reflected field | |
| EXPH | | Phase term associated with the diffracted field | |
| GAM | ($\gamma$) | Phase factor for diffracted field | |
| GF | ($g_f$) | Feed pattern value calculated in subroutine FEED | (I) |
| GFP | | Feed pattern values used for incident field and its slope | |
| KP | | Loop index for calculating the incident feed value and its slope | |

| | | | |
|---|---|---|---|
| LCORNR | | Logical variable for corner diffraction | (I) |
| LDIF | | Logical variable for edge diffraction | (I) |
| LNF | | Logical variable for near field calculation | (I) |
| LRANG | | Logical variable for constant range field calculation | (I) |
| LSLOPE | | Logical variable for slope diffraction | (I) |
| P2,P3 | | Field point coordinates (see User's Manual) | (I) |
| PH | ($\hat{\phi}$) | Rectangular components of the unit vector of the direction of increasing diffraction angle PS | |
| PHEI | | Phase term associated with the feed spillover field | |
| PHGAM | ($\phi_\gamma$) | PHI coordinate of the field point referred to the tilted feed system | |
| PHI | | PHI coordinate of the field point | |
| PHIP | | PHI coordinate of the feed observation direction as referred to the source XS | |
| PHO | ($\hat{\phi}'$) | Rectangular components of the unit vector in the direction of increasing incident PHI angle PSO | |
| PS | ($\phi$) | Wedge diffraction angle (see Fig. 3a) | |
| PSA | ($\psi_\alpha$) | Theta coordinate of the observation direction measured from the feed axis | |
| PSI | ($\psi$) | Theta coordinate of the feed observation direction measured from the negative z-axis of the reflector | |
| PSO | ($\phi'$) | Incidence angle for wedge diffraction (see Fig. 3a) | |
| RHON | ($\rho$) | Radial coordinate of the near field point XN | |
| RHOS | | Radii to the reflected shadow boundaries calculated in subroutine SBDY | (I) |
| RLC | ($L_c$) | Distance parameter for corner diffraction | |

| | | | |
|---|---|---|---|
| RR | $(R)$ | Range to the field point from the origin | |
| S | $(s)$ | Distance from the diffraction point XD to the near field point XN | |
| SBO | $(\sin\beta_o)$ | Sine of the diffracted cone half angle | |
| SC | $(s_c)$ | Incident ray path length to the corner | |
| SP | $(s')$ | Incident ray path length to the diffraction point XD | |
| SPH | $(\sin\phi)$ | Sine of diffraction angle PS | |
| SPHO | $(\sin\phi')$ | Sine of incident angle PSO | |
| SS | $(s_s)$ | Diffracted ray path length from the corner to the near field point | |
| TERM | | Temporary variable for corner diffraction | |
| THEB | $(\theta_B)$ | Theta coordinate of diffraction shadow boundary to opposite side of reflector rim | (I) |
| THETA | $(\theta)$ | Theta coordinate of the field point | |
| TPP | $(L)$ | Distance parameter for edge diffraction | |
| V | $(\hat{V})$ | Rectangular components of the edge unit vector of segment ME | (I) |
| VI | $(\widehat{VI})$ | Rectangular components of the incident ray unit vector to the diffraction point | |
| VIC | $(\widehat{VI}_c)$ | Rectangular components of the incident ray unit vector to the corner | (I) |
| VMAG | | Segment length | |
| VMG | | Distance from the first corner to the diffraction point XD | |
| VN | $(\widehat{VN})$ | Rectangular components of the unit normal vector of the segment ME | (I) |
| VP | $(\widehat{VP})$ | Rectangular components of the unit binormal vector of the segment ME | (I) |
| X1 | | X component of the direct incident ray path for near field | |

| | | |
|---|---|---|
| X2 | Y component of the direct incident ray path for near field | |
| X3 | Z component of the direct incident ray path for near field | |
| XD | Rectangular coordinates of the diffraction point | |
| XN | Rectangular components of the near field point coordinates | |
| XOO | Origin of the near field plane cut | (I) |

## CODE LISTING

```
      SUBROUTINE GTD(P3I,NGTD,NTHE,DP3)
C!!!
C!!!  DETERMINES THE DIFFRACTED FIELD,WITH PHASE REFERRED TO ORIGI
     N.
C!!!  FIELD DIFF. FROM EDGE #ME
C!!!  CORNER DIFF. IS OPTIONAL FROM INPUT DATA.
C!!!
      COMPLEX EA(2,50,50),ERX,ERY,PHER,RFCT
      COMPLEX CJ,DS,DH,DPS,DPH,BS,BH,BPS,BPH
      COMPLEX EF,EG,EDPR,EDPL,ED(3),IMT,EDX,EDY,EDZ,EDT,EDP
      COMPLEX EIPRP,EIPLP,EIX,EIY,EIZ,EIT,EIP,CORN(2),FFCT
      COMPLEX EIPL,EIPR,ECPL,ECPR,EXPH,CP,CX,CY,PHEI
      DIMENSION RHOS(2),GFP(2),DEL(2)
      DIMENSION VI(3),XN(3),XD(3),PHO(3),PH(3),ROP(3),RO(3),VIP(3)
      LOGICAL LSLOPE,LCORNR,LDIF,LDEBUG,LTEST,LNF,LRANG
      LOGICAL LFEED,LOUT,LCP,LWRITE
      COMMON /GEOM1/X(67,3),V(67,3),MRIM
      COMMON /GEOM2/VP(67,3),VN(67,3),BD(67,2),VMAG(67),RMC(67),
     2VIC(67,3),XM(67,3)
      COMMON /FBDY/RHOS
      COMMON /FOCAL/F,ZOP
      COMMON /SORINF/XS(3)
      COMMON /BDY2/TH1,TH2,THEB
      COMMON /DIR/D(3),EIX,EIY,EIZ
      COMMON /NF/RFCT,XOO(3),PHIF,P2,RR
      COMMON /GTDD/LFEED,LOUT,LCP,LWRITE,COSPT,SINPT,REFDE,TEMP
      COMMON /DSC/DS,DH,DPS,DPH,BS,BH,BPS,BPH
      COMMON /COMP/CX,CY,GF,PHP,PHO,KX,KY,ISYM,SINTL,COSTL
      COMMON /PIS/PI,TPI,DPR
      COMMON /LOGDIF/LSLOPE,LCORNR,LNF,LRANG
      COMMON /TEST/LDEBUG,LTEST,NTEST
      COMMON /REFL/DD,RO,ICO,JCO
      COMMON /GRID1/GRIDX,GRIDY,EA
```

```
      COMMON /OUT/NW
      DATA DLVI/0.02/
      DATA DELT/0.01/
C
      ZO=F-ZOP
      CJ=(0.,1.)
      BDEL=0.
      IF (LCORNR) BDEL=0.5
      FN=2.
      KP=1
      PHP=0.
      PHQ=90.
      IF (LSLOPE) KP=2
      IF (LDEBUG) WRITE (NW,106)
  106 FORMAT (/,' DEBUGGING GTD SUBROUTINE')
      IF (LNF) GO TO 1
      PHI=P2
      PHIR=P2/DPR+1.E-4
      SINP=SIN(PHIR)
      COSP=COS(PHIR)
      S=RR
      GO TO 255
    1 SINPE=SIN(PHIE/DPR)
      COSPE=COS(PHIE/DPR)
      IF (.NOT.LRANG) ZE=P2
      IF (LRANG) RE=P2
C
      WRITE (NW,240) RHOS(1),RHOS(2)
  240 FORMAT (/T12,'THE REFLECTED SHADOW BOUNDARIES IN THE PHIE',
     2' PLANE ARE AT',//T20,'RHOS1 =',F9.3,5X,'AND RHOS2 =',F9.3,/
 /)
      PREVP=361.
  255 P3=P3I
      DO 100 NM=1,NGTD
      NN=NM+NTHE
      IF (.NOT.LNF) GO TO 5
C
C     ***** NEAR FIELD COORDINATE CONVERSION *****
C
      IF (.NOT.LRANG) GO TO 3
      THE=P3/DPR
      SINTE=SIN(THE)
      COSTE=COS(THE)
  242 XN(1)=XCO(1)+RE*SINTE*COSPE
      XN(2)=XCO(2)+RE*SINTE*SINPE
      XN(3)=XCO(3)+RE*COSTE
      IF (LDEBUG) WRITE (NW,205) RE,P3
  205 FORMAT (/T10,'RE =',F10.3,5X,'THE =',F7.2,/)
      IF (XN(1).NE.0..OR.XN(2).NE.0.) GO TO 4
      SINTE=SINTE+0.001
      GO TO 242
```

```
3     ZL=P3
      DRHO=ZL-RHOS(1)
      IF (DRHO.LE.0.21.AND.DRHO.GE.0.) ZL=RHOS(1)-0.05
      DRHO=ZL-RHOS(2)
      IF (DRHO.LE.0.21.AND.DRHO.GE.0.) ZL=RHOS(2)+0.05
      XN(1)=XOO(1)+ZL*COSPE
      XN(2)=XOO(2)+ZL*SINPE
      XN(3)=ZE
      IF (LDEBUG) WRITE (NW,302) ZE,P3
302   FORMAT (/T10,'ZE =',F10.3,5X,'ZL =',F10.4,/)
4     PHIR=BTAN2(XN(2),XN(1))
      SINP=SIN(PHIR)
      COSP=COS(PHIR)
      RR=SQRT(XN(1)*XN(1)+XN(2)*XN(2)+XN(3)*XN(3))
      IF (LDEBUG) WRITE (NW,108) XN(1),XN(2),XN(3)
      COST=XN(3)/RR
      THER=ACOS(COST)
      THETA=THER*DPR
      GO TO 6
5     THETA=P3
      THER=THETA/DPR
C
6     IF ((LTEST).OR.(LDEBUG)) WRITE (NW,2) THETA
2     FORMAT (/T2,7HTHETA =,F7.2/)
      SINT=SIN(THER)
      COST=COS(THER)
      EDX=(0.,0.)
      EDY=(0.,0.)
      EDZ=(0.,0.)
      D(1)=SINT*COSP
      D(2)=SINT*SINP
      D(3)=COST
      DO 60 ME=1,MRIM
      EDPR=(0.,0.)
      EDPL=(0.,0.)
      ECPR=(0.,0.)
      ECPL=(0.,0.)
      MC=ME+1
      IF(MC.GT.MRIM) MC=1
      IF (.NOT.LNF) GO TO 9
      DMAG=0.
      DO 7 N=1,3
      D(N)=XN(N)-XM(ME,N)
7     DMAG=DMAG+D(N)*D(N)
      DMAG=SQRT(DMAG)
      S=DMAG
      IF (LDEBUG) WRITE (NW,199) DMAG
      DO 8 N=1,3
8     D(N)=D(N)/DMAG
9     DV=0.
```

```
      DO 10 N=1,3
      IF (LDEBUG) WRITE(NW,108) D(N)
 10   DV=DV+D(N)*V(ME,N)
      BDLOW=BD(ME,1)-BDEL
      BDHI=BD(ME,2)+BDEL
      IF (LDEBUG) WRITE (NW,12) ME,DV,BDLOW,BDHI
  12  FORMAT (/T10,I2,' DV=',F8.4,5X,'BDLOW =',F8.4,5X,'BDHI =',F8.4
 ,/)
C!!!  DETERMINE IF DIFFRACTION EXISTS
      IF(DV.LT.BDLOW.OR.DV.GT.BDHI)GO TO 60
C
C!!!  COMPUTE EDGE DIFFRACTION POINT
C
      CALL DFPTWD(XS,XN,DV,VI,SP,XD,ME)
      IF (LDEBUG) WRITE (NW,112) ME,SP,(XS(N),XM(ME,N),XD(N),VI(N),
     2N=1,3)
 112  FORMAT (I5,5X,4HSP =,F10.4,11X,2HXS,8X,2HXM,8X,2HXD,9X,2HVI,
     23(/T30,4F10.4),/)
      IF (.NOT.LNF) GO TO 14
      DMAG=0.
      DO 11 N=1,3
      D(N)=XN(N)-XD(N)
  11  DMAG=DMAG+D(N)*D(N)
      DMAG=SQRT(DMAG)
      S=DMAG
      IF (LDEBUG) WRITE (NW,199) DMAG
 199  FORMAT (/T10,'DMAG =',F10.3,/)
      DV=0.
      DO 13 N=1,3
      D(N)=D(N)/DMAG
      DV=DV+D(N)*V(ME,N)
  13  IF (LDEBUG) WRITE (NW,-) D(N)
 14   ADN=0.
      VMG=0.
C!!!  COMPUTE VMG,WHICH IS DISTANCE FROM FIRST CORNER OF
C!!!  EDGE TO DIFFRACTION POINT.
      DO 15 N=1,3
      VMG=VMG+(XD(N)-X(ME,N))*V(ME,N)
 15   ADN=ADN+(XS(N)-X(1,N))*VN(ME,N)
      LDIF=.TRUE.
      IF (LDEBUG) WRITE (NW,200) VMG,VMAG(ME),DV
 200  FORMAT (/T10,'VMG =',E10.3,5X,'EDGE LENGTH =',E10.3,/T10,
     2'DV =',F10.4,/)
      IF(VMG.LT.0.)GO TO 101
      IF(VMG.LE.VMAG(ME))GO TO 102
      SP=RMC(MC)
      DO 103 N=1,3
 103  VI(N)=VIC(MC,N)/SP
      LDIF=.FALSE.
      GO TO 102
 101  SP=RMC(ME)
```

```
      DO 104 N=1,3
  104 VI(N)=VIC(ME,N)/SP
      LDIF=.FALSE.
  102 QI=0.
      PP=0.
      QD=0.
      PD=0.
      DO 20 N=1,3
      QI=QI-VN(ME,N)*VI(N)
      PP=PP-VP(ME,N)*VI(N)
      QD=QD+VN(ME,N)*D(N)
   20 PD=PD+VP(ME,N)*D(N)
C!!!  PS,PSO ARE THE DIFFRACTION PHI ANGLES,WHERE PSO IS
C!!!  INCIDENT PHI AND PS IS DIFFRACTED PHI.
      PSOR=BTAN2(QI,PP)
      PSO=DPR*PSOR
      IF(PSO.LT.0.) PSO=360.+PSO
      PSR=BTAN2(QD,PD)
      PS=DPR*PSR
      IF (LDEBUG) WRITE (NW,107) ME,PSO,PS
  107 FORMAT (/T10,I5,5X,'PSO =',F7.2,5X,'PS =',F7.2,/)
C
C     * CHECK IF DIFFRACTED FIELD IS BLOCKED BY THE REFLECTOR *
C
      IF (PS.GE.0..AND.THER.GT.THEB) GO TO 60
      IF(PS.LT.0.) PS=360.+PS
      FNP=FN*180.
      SPHO=SIN(PSOR)
      CPHO=COS(PSOR)
      SPH=SIN(PSR)
      CPH=COS(PSR)
C!!!  COMPUTE DIFFRACTION POLARIZATION UNIT VECTORS(PHO,PH,BOP,BO)
      DO 30 N=1,3
      PHO(N)=-VP(ME,N)*SPHO+VN(ME,N)*CPHO
   30 PH(N)=-VP(ME,N)*SPH+VN(ME,N)*CPH
      BOP(1)=PHO(2)*VI(3)-PHO(3)*VI(2)
      BOP(2)=PHO(3)*VI(1)-PHO(1)*VI(3)
      BOP(3)=PHO(1)*VI(2)-PHO(2)*VI(1)
      BO(1)=PH(2)*D(3)-PH(3)*D(2)
      BO(2)=PH(3)*D(1)-PH(1)*D(3)
      BO(3)=PH(1)*D(2)-PH(2)*D(1)
      IF (LDEBUG) WRITE (NW,108) (PHO(N),PH(N),BOP(N),BO(N),N=1,3)
  108 FORMAT (T20,4F12.5)
C
C!!!  COMPUTE SOURCE PATTERN FACTORS
C
      DO 29 K=1,KP
      PSIR=BTAN2(SQRT(VI(1)*VI(1)+VI(2)*VI(2)),-VI(3))
      PHIPR=BTAN2(VI(2),VI(1))
      PSI=PSIR*DPR
```

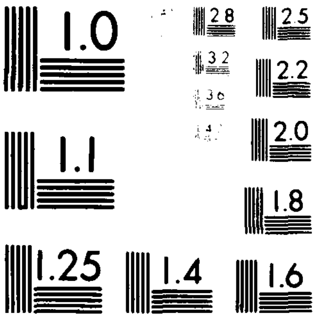
```
      PHIP=PHIPR*DPR
      CALL FEED(PSI,PHIP,PSA,PHGAM)
      IF (K.EQ.2) GO TO 24
      PSA1=PSA
      PHGAM1=PHGAM
   24 GFP(K)=GF
      DO 27 N=1,3
      VI(N)=VI(N)+PHO(N)*DLVI
   27 CONTINUE
   29 CONTINUE
      GF=GFP(1)
      CALL FPOL(EIX,EIY,EIZ,PSA1,PHGAM1)
      IF ((LDIF).AND.(LDEBUG)) WRITE (NW,25) EIX,EIY,EIZ
   25 FORMAT (/T10,5HEIX =,2F10.4,5X,5HEIY =,2F10.4,5X,5HEIZ =,
     12F10.4,/)
      EIPR=(EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3))*F/SP
      EIPL=(EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3))*F/SP
      IF (LDEBUG) WRITE (NW,31) SP,EIPR,EIPL
   31 FORMAT (T5,4HSP =,F10.4,5X,6HEIPR =,2E10.3,5X,6HEIPL =,2E10.3)
      IF (.NOT.LSLOPE) GO TO 36
      GF=(GFP(2)-GFP(1))/DLVI
      CALL FPOL(EIX,EIY,EIZ,PSA1,PHGAM1)
      EIPRP=(EIX*PHO(1)+EIY*PHO(2)+EIZ*PHO(3))*F/SP
      EIPLP=(EIX*BOP(1)+EIY*BOP(2)+EIZ*BOP(3))*F/SP
      IF (LDEBUG) WRITE (NW,37) EIPRP,EIPLP
   37 FORMAT (T24,7HEIPRP =,2E10.3,4X,7HEIPLP =,2E10.3)
C
C!!!  COMPUTE SBO=SINE(BO)
C
   36 CONTINUE
      SBO=SQRT((V(ME,3)*D(2)-V(ME,2)*D(3))**2+(V(ME,1)
      &*D(3)-V(ME,3)*D(1))**2+(V(ME,2)*D(1)-V(ME,1)*D(2))
      &**2)
      TPP=SP*SBO*SBO
      IF(LNF) GO TO 592
      GAM=0.
      AS=SQRT(SP)
      DO 590 N=1,3
  590 GAM=GAM+XD(N)*D(N)
      GO TO 595
  592 GAM=-S
      AS=SQRT(SP/(S*(S+SP)))
      TPP=TPP*S/(S+SP)
  595 IF (LDEBUG) WRITE (NW,599) ME,TPP,PS,PSO,SBO
  599 FORMAT (I10,5X,3HR =,F9.4,5X,4HPS =,F9.4,5X,5HPSO =,F9.4,
     25X,5HSBO =,F6.3)
      EXPH=CEXP(CMPLX(0.,TPI*(GAM-SP0)))
      EIPR=EIPR*EXPH
      EIPL=EIPL*EXPH
      EIPRP=EIPRP*EXPH
```

```
      EIPLP=EIPLP*EXPH
      CP=CJ*TPI*SP*SBO
      IF (LDEBUG) WRITE (NW,601) GAM,AS,EXPH,EIPR,EIPL,EIPRP,EIPLP
  601 FORMAT (T10,5HGAM =,F15.6,5X,'AS =',F15.6,/10E12.6)
      DO 361 J=1,2
      DEL(J)=0.
  361 CORN(J)=(0.,0.)
C
C     ***** SKIP LOOP 22 IF LCORNR FALSE *****
C
      IF (.NOT.LCORNR) GO TO 26
      MC=ME
      ISN=1
C!!!  LOOP THRU BOTH CORNERS ON EDGE #ME.
      DO 22 I=1,2
      IF(MC.GT.MRIM) MC=1
      ISN=-ISN
      SC=RMC(MC)
      IF (LDEBUG) WRITE (NW,301) (V(ME,N),VIC(MC,N),N=1,3)
  301 FORMAT (/3(T10,2F10.5,/))
      COSBC=V(ME,1)*VIC(MC,1)+V(ME,2)*VIC(MC,2)+V(ME,3)*VIC(MC,3)
      COSBC=-ISN*COSBC/SC
      CBOC=ISN*DV
      IF (LDEBUG) WRITE (NW,-) ISN,SC,COSBC,DV,CBOC
      BETC=ACOS(COSBC)
      SINBC=SIN(BETC)
      RLC=SC
      ASC=1.
      ZP=(X(MC,1)-XD(1))*D(1)+(X(MC,2)-XD(2))*D(2)
     &+(X(MC,3)-XD(3))*D(3)
      IF (.NOT.LNF.AND..NOT.LRANG) GO TO 305
      SV=0.
      SSM=0.
      DO 304 N=1,3
      SX=XN(N)-X(MC,N)
      SV=SV+SX*V(ME,N)
  304 SSM=SSM+SX*SX
      SS=SQRT(SSM)
      CBOC=ISN*SV/SS
      RLC=SC*SS/(SC+SS)
      ASC=1/SS
      ZP=S-SS
  305 BETOC=ACOS(CBOC)
      DEL(I)=2.*TPI*RLC*(COS(.5*(BETC+BETOC))**2)
      TERM=-SINBC*SQRT(SP/SC)*ASC/(TPI*(COSBC+CBOC))
C!!!  COMPUTE CORNER DIFFRACTION COEFFICIENT(CORN).
      CORN(I)=-TERM*FFCT(DEL(I))*CEXP(CMPLX(0.,-TPI*(SC-SP-ZP)-.25*
      IF (LDEBUG) WRITE (NW,331) BETC,BETOC,SC,SP,SS,RLC,ZP,DEL(I),
     2TERM,CORN(I)
   22 MC=MC+1
   26 CONTINUE
```

AD-A097 415 OHIO STATE UNIV COLUMBUS ELECTROSCIENCE LAB F/G 20/4
NUMERICAL ELECTROMAGNETIC CODE (NEC)-REFLECTOR ANTENNA CODE: PA--ETC(U)
SEP 79 S H LEE, R C RUDDUCK N00123-76-C-1371
UNCLASSIFIED ESL-784508-16 NL

3 of 3

END
DATE
FILMED
5-81
DTIC


1.0
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.1
1.8
1.25
1.4
1.6

```
      CALL DCHP(DEL,CORN,TPP,PS,PSO,SBO)
      IF (LDEBUG) WRITE (NW,32) DS,DH,DPS,DPH
   32 FORMAT (/T10,4HDS =,2F10.5,5X,4HDH =,2F10.5,/T10,5HDPS =,
     22F10.5,5X,5HDPH =,2F10.5,/)
      IF (.NOT.LDIF) GO TO 202
      EDPR=-EIPR*AS*DH
      EDPL=-EIPL*AS*DS
      IF (LDEBUG) WRITE (NW,34) ME,EDPR,EDPL
  201 IF (.NOT.LSLOPE) GO TO 202
      EDPR=EDPR+EIPRP*AS*DPH/CP
      EDPL=EDPL+EIPLP*AS*DPS/CP
  202 CONTINUE
      IF ((LDIF).AND.(LDEBUG)) WRITE (NW,34) ME,EDPR,EDPL
   34 FORMAT (I15,5X,6HEDPR =,2F10.4,5X,6HEDPL =,2F10.4)
C
C!!!  IS CORNER DIFFRACTED FIELD DESIRED?
C
      IF (.NOT.LCORNR) GO TO 45
      IF (LDEBUG) WRITE (NW,35) BS,BH,BPS,BPH
   35 FORMAT (T10,4HBS =,2F10.5,4X,4HBH =,2F10.5,/T10,5HBPS =,
     22F10.5,4X,5HBPH =,2F10.5,/)
      ECPR=-EIPR*BH
      ECPL=-EIPL*BS
      IF (LDEBUG) WRITE (NW,42) ME,ECPR,ECPL
      IF(.NOT.LSLOPE)GO TO 203
      ECPR=ECPR+EIPRP*BPH/CP
      ECPL=ECPL+EIPLP*BPS/CP
  203 CONTINUE
      IF (LDEBUG) WRITE (NW,42) ME,ECPR,ECPL
   42 FORMAT (I15,5X,6HECPR =,2E10.3,5X,6HECPL =,2E10.3)
      EDPR=EDPR+ECPR
      EDPL=EDPL+ECPL
      IF (LDEBUG) WRITE (NW,38) ME,EDPR,EDPL
   38 FORMAT (I15,5X,6HEDPR =,2E10.3,5X,6HEDPL =,2E10.3)
C!!!  COMPUTE THETA AND PHI COMPONENTS OF TOTAL DIFF. FIELD
C
   45 CONTINUE
      DO 48 N=1,3
   48 ED(N)=EDPL*BO(N)+EDPR*PH(N)
      IF (LDEBUG) WRITE (NW,55) ME,ED(1),ED(2),ED(3)
   55 FORMAT (I15,5X,5HED1 =,2E10.3,5X,5HED2 =,2E10.3,5X,5HED3 =,
     22E10.3)
      EDX=EDX+ED(1)
      EDY=EDY+ED(2)
      EDZ=EDZ+ED(3)
   60 CONTINUE
      IF (.NOT.LNF) GO TO 80
      IF (LOUT) WRITE (NW,62) EDX,EDY,EDZ
   62 FORMAT (/T10,T10,5HEDX =,2E10.3,5X,5HEDY =,2E10.3,5X,
     25HEDZ =,2E10.3,/)
C
C     ***** NEAR FIELD SECTION *****
C
```

```
      X1=XN(1)-XS(1)
      X2=XN(2)-XS(2)
      X3=XN(3)-XS(3)
      RHO=SQRT(X1*X1+X2*X2)
      IF (XN(3).LT.0.) GO TO 65
      RHON=SQRT((XN(1)-X00(1))**2+(XN(2)-X00(2))**2)
      IF (RHON.LT.RHOS(2).OR.RHON.GT.RHOS(1)) GO TO 65
C
C          *** REFLECTED FIELDS ***
C
      I=ICO+XN(1)/GRIDX+DELT
      J=JCO+XN(2)/GRIDY+DELT
      M=I+1
      N=J+1
      DX=XN(1)/GRIDX+ICO-I
      DY=XN(2)/GRIDY+JCO-J
      PHER=CEXP(-CJ*TPI*XN(3))
      ERX=(EA(1,M,N)*(1.-DX-DY)+EA(1,M+1,N)*DX+EA(1,M,N+1)*DY)*PHER
      ERY=(EA(2,M,N)*(1.-DX-DY)+EA(2,M+1,N)*DX+EA(2,M,N+1)*DY)*PHER
      IF (LOUT) WRITE (NW,64) ERX,ERY
   64 FORMAT (/T10,'ERX =',2E12.4,5X,'ERY =',2E12.4,/)
      EDX=EDX+ERX
      EDY=EDY+ERY
C
C          *** SPILLOVER FIELDS ***
C
   65 IF (.NOT.LFEED) GO TO 74
      PHIPR=BTAN2(X2,X1)
      PHIP=PHIPR*DPR
      PSI=BTAN2(RHO,-X3)*DPR
      THETA=180.-PSI
      RS=SQRT(RHO*RHO+X3*X3)
      PHEI=CEXP(-CJ*TPI*RS)*F/RS
      IF (XN(3).GE.0.) GO TO 70
      IF (ABS(PHIP-PREVP).GT.0.001) CALL SBDY(MRIM,X,XS,PHIE,
     2TH1,TH2,THEB)
      PREVP=PHIP
      IF (ABS(THETA-TH1).LT.0.05) THETA=THETA+0.05
      IF (ABS(THETA-TH2).LT.0.05) THETA=THETA-0.05
      IF (THETA.LE.TH2.AND.THETA.GE.TH1) GO TO 74
   70 CALL FEED(PSI,PHIP,PSA,PHGAM)
      CALL FPOL(EIX,EIY,EIZ,PSA,PHGAM)
      EIX=EIX*PHEI
      EIY=EIY*PHEI
      EIZ=EIZ*PHEI
      IF (LOUT) WRITE (NW,72) EIX,EIY,EIZ,EDX,EDY,EDZ
   72 FORMAT(2H 0,T15,5HEIX =,2E10.4,5X,5HEIY =,2E10.4,5X,5HEIZ =,
     22E10.4,T79,1H0,/2H 0,T15,5HEDX =,2E10.4,5X,5HEDY =,2E10.4,
     35X,5HEDZ =,2E10.4,T79,1H0)
      EDX=EDX+EIX
```

```
      EDY=EDY+EIY
      EDZ=EDZ+EIZ
   74 IF (.NOT.LRANG) GO TO 75
      EDT=COST*(COSP*EDX+SINP*EDY)-SINT*EDZ
      EDP=-SINP*EDX+COSP*EDY
      GO TO 84
   75 CALL DBPHS(AEDX,EDX,0.)
      CALL DBPHS(AEDY,EDY,0.)
      CALL DBPHS(AEDZ,EDZ,0.)
      IF(LWRITE)WRITE (NW,76) P3,AEDX,EDX,AEDY,EDY,AEDZ,EDZ
   76 FORMAT(2H W,T5,F6.2,4X,,3(E10.3,2F10.2))
      PLT=REAL(EDY)
      GO TO 90
C
C     ***** FAR FIELD SECTION *****
C
   80 EDT=(COST*COSP*EDX+COST*SINP*EDY-SINT*EDZ)*RFCT
      EDP=(-SINP*EDX+COSP*EDY)*RFCT
      IF (LOUT) WRITE (NW,82) EDT,EDP
   82 FORMAT(2H T,T10,'EDT =',2E11.3,5X,'EDP =',2E11.3,T79,1HT)
      IF (.NOT.LFEED) GO TO 84
      IF (THETA.LT.TH2.AND.THETA.GT.TH1) GO TO 84
C
C          *** SPILLOVER FIELDS ***
C
      PSI=180.-THETA
      PSIR=PSI/DPR
      SINS=SIN(PSIR)
      COSS=COS(PSIR)
      CALL FEED(PSI,PHI,PSA,PHGAM)
      CALL FPOL(EIX,EIY,EIZ,PSA,PHGAM)
      EIT=-COSS*COSP*EIX-COSS*SINP*EIY-SINS*EIZ
      EIP=-SINP*EIX+COSP*EIY
      PHEI=CEXP(CJ*TPI*ZO*COST)*F*RFCT
      EIT=EIT*PHEI
      EIP=EIP*PHEI
      IF (LOUT) WRITE (NW,-) EIT,EIP
      EDT=EDT+EIT
      EDP=EDP+EIP
      IF (LOUT) WRITE (NW,82) EDT,EDP
C
C          *** PRINC AND CROSS POLARIZED COMPONENTS ***
C
   84 TMT=EDT
      EDT=COSPT*EDT-SINPT*EDP
      EDP=SINPT*TMT+COSPT*EDP
      IF (.NOT.LCP) GO TO 85
      TMT=EDT
      EDT=TEM2*(EDT-CJ*EDP)
      EDP=TEM2*(TMT+CJ*EDP)
```

```
   85 CALL DBPHS(AEDT,EDT,REFDB)
      CALL DBPHS(AEDP,EDP,REFDB)
      IF(LWRITE)WRITE (NW,86) P3,AEDT,EDT,AEDP,EDP
   86 FORMAT(2H W,T5,F6.2,4X,,2(E10.3,2F10.2),T79,1HW)
      PLT=REAL(EDT)
   90 CONTINUE
      WRITE (2) PLT
      P3=P3+DP3
  100 CONTINUE
      RETURN
      END
```

## SUBROUTINE LNFD

### PURPOSE

To calculate the feed pattern value by linearly interpolating the input feed data in a given PHI cut.

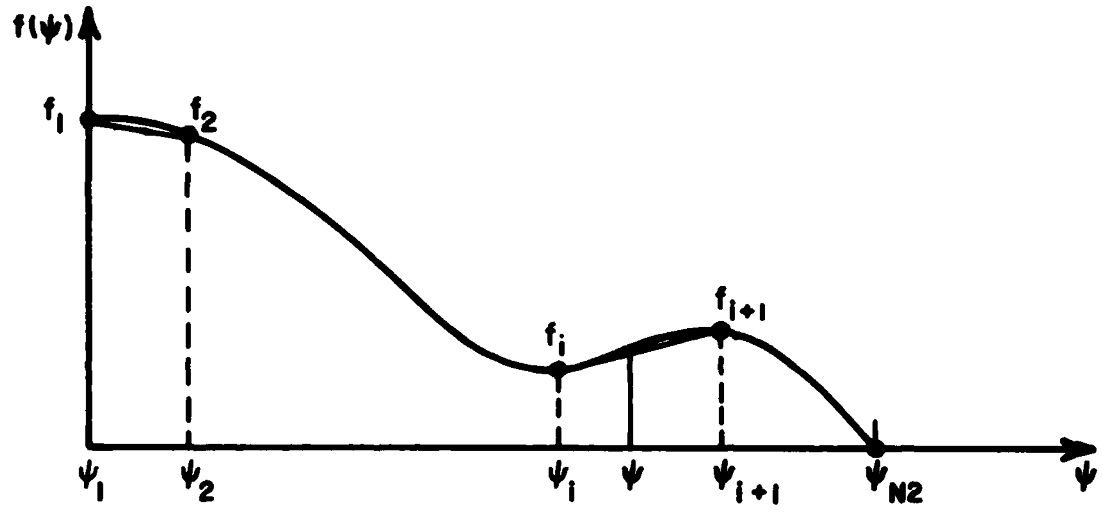


Figure 1. Piece-wise linear approximation for feed patterns.

### METHOD

The feed pattern value $f_e$ at an angle $\psi$ is calculated by

$$f_e = f_i \cdot (1-dp) + f_{i+1} \cdot dp$$

where

$$dp = \frac{\psi-\psi_i}{\psi_{i+1}-\psi_i}$$

and $f_i$'s are the input feed pattern data.

FLOW DIAGRAM

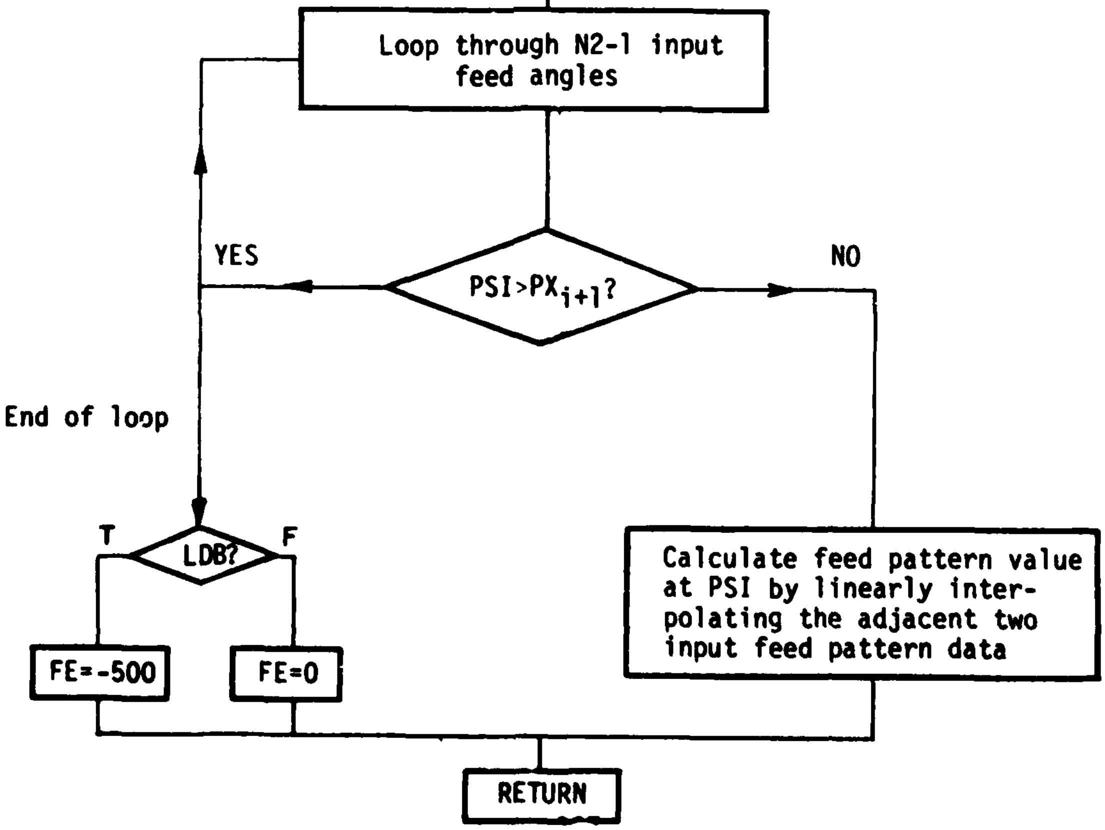
LNFD(PX,F,PSI,N2,FE,LDB)

INPUT VARIABLES

| | | |
|---|---|---|
| PX | ($\psi_i$) | Input feed pattern angles |
| F | ($f_i$) | Input feed pattern data at PX |
| PSI | ($\psi$) | Feed pattern angle at which the feed pattern value is to be calculated |
| N2 | | Total number of input data points |
| LDB | | Logical variable which specifies the feed pattern in dB values, if true, or as linear field values, if false |

OUTPUT VARIABLES

| | | |
|---|---|---|
| FE | ($f_e$) | Calculated feed pattern value |



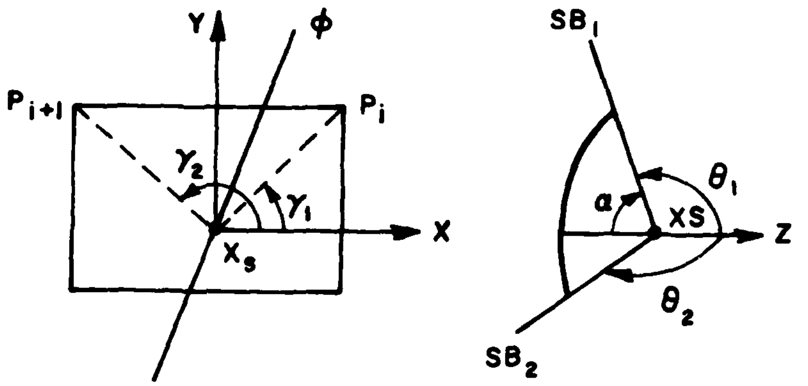
## CODE LISTING

```
      SUBROUTINE LNFD(PX,F,PSI,N2,FE,LDB)
      DIMENSION PX(15),F(15)
      LOGICAL LDB
      N3=N2-1
      DO 10 I=1,N3
      IF (PSI.GT.PX(I+1)) GO TO 10
      DP=(PSI-PX(I))/(PX(I+1)-PX(I))
      FE=F(I)*(1.-DP)+F(I+1)*DP
      RETURN
   10 CONTINUE
      FE=0.
      IF (LDB) FE=-500.
      RETURN
      END
```

## SUBROUTINE SBDY

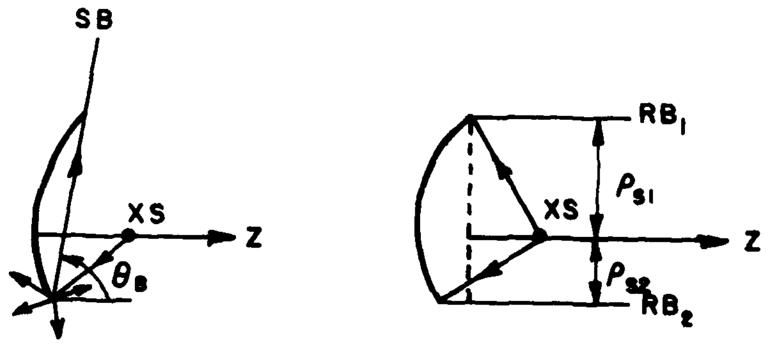
### PURPOSE

To calculate the shadow boundary angles for the spillover field, the edge diffracted field and the reflected field.



(a) FRONT VIEW

(b) SIDE VIEW FOR SPILLOVER FIELD SHADOW BOUNDARY



(c) SIDE VIEW FOR DIFFRACTED FIELD SHADOW BOUNDARY

(d) SIDE VIEW FOR REFLECTED FIELD SHADOW BOUNDARY

Figure 1. Geometry for shadow boundary angles.

## METHOD

In order to find the appropriate shadow boundary angles and distances, the intersecting points on the reflector rim cut by the PHI plane must be located. By comparing $\phi$ with $\gamma_1$ and $\gamma_2$, which are the projected angles defined by the source XS and the rim points $P_i$ and $P_{i+1}$, respectively, measured from the x-axis, on the aperture plane, (Fig. 1a) the index of the rim section cut by the $\phi$ plane is found and the intersecting point coordinates x and y can be obtained by solving the linear equation of the corresponding edge and $y=x \tan\phi$; then z by solving

$$z = \frac{x^2+y^2}{4F} \quad .$$

Once x,y and z are determined, the parameters for the different shadow boundaries are readily found as follows:

a) Incident shadow boundary angle $\theta$

$$\theta_{1,2} = \pi-\alpha_{1,2}$$

where $\alpha$ is defined by XS and the intersecting points on the rim (see Fig. 1b) and is given by

$$\alpha = \tan^{-1}\left(\sqrt{\frac{(x-XS(1))^2+(y-XS(2))^2}{z-XS(3)}}\right) \quad .$$

b) Diffracted shadow boundary angle $\theta_B$

This angle is defined by the two intersecting points on the upper and lower rim respectively, measured from the z-axis (Fig. 1c) and is given by

$$\theta_B = \tan^{-1}\left(\frac{\Delta\rho}{\Delta z}\right)$$

where

$$\Delta\rho = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

and

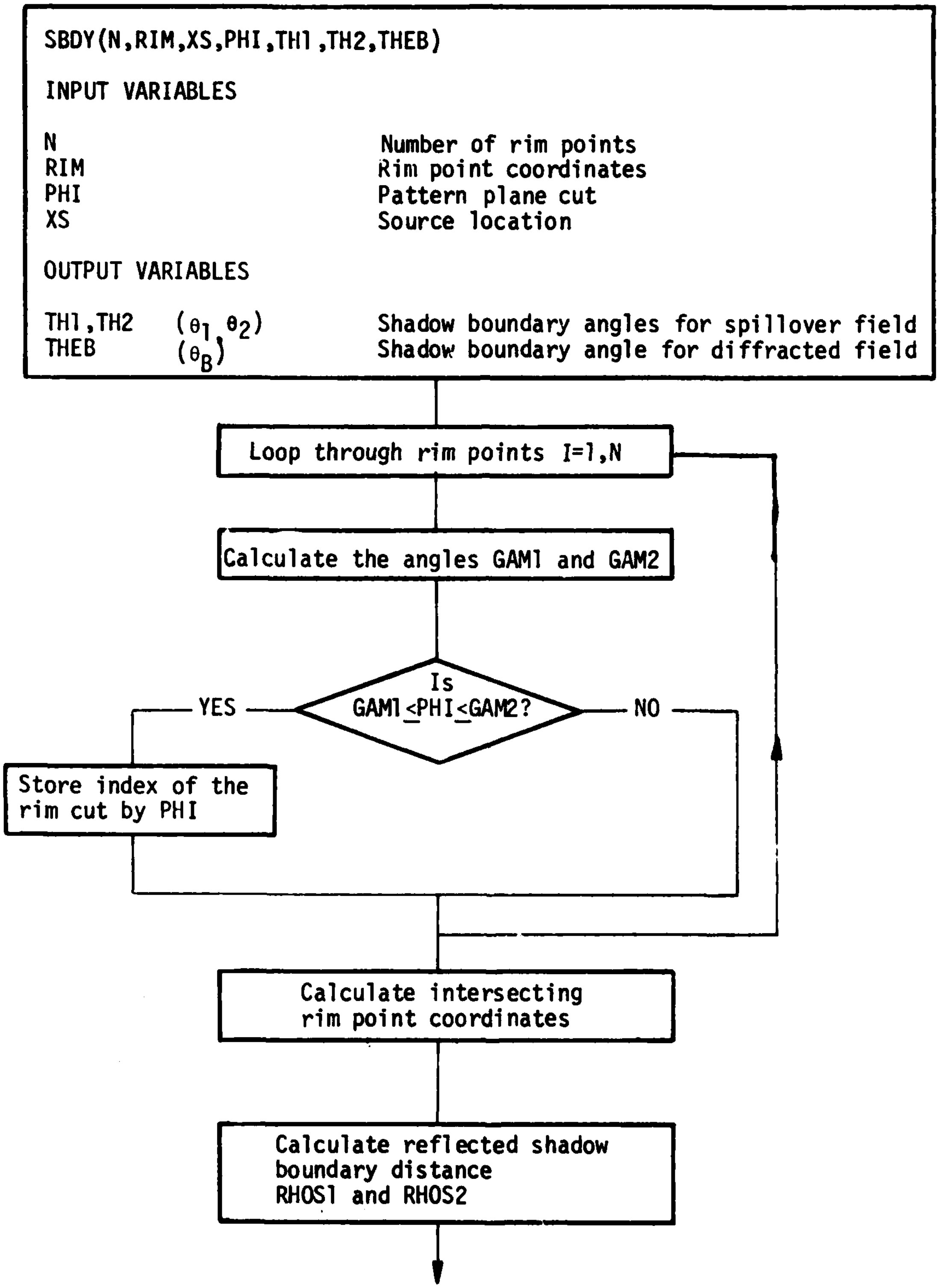
$$\Delta z = z_1-z_2$$

c) Reflected shadow boundary distance $\rho_s$

$$\rho_{s1,2} = \sqrt{(x-XS(1))^2+(y-XS(2))^2}$$

which is illustrated in Fig. 1d.

FLOW DIAGRAM



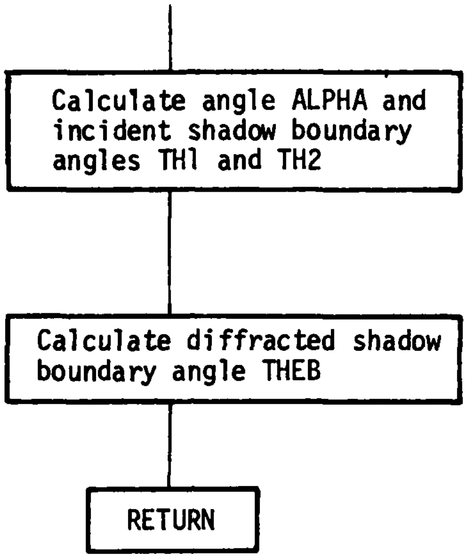
SBDY(N,RIM,XS,PHI,TH1,TH2,THEB)

INPUT VARIABLES

| | |
|---|---|
| N | Number of rim points |
| RIM | Rim point coordinates |
| PHI | Pattern plane cut |
| XS | Source location |

OUTPUT VARIABLES

| | | |
|---|---|---|
| TH1,TH2 | $(\theta_1, \theta_2)$ | Shadow boundary angles for spillover field |
| THEB | $(\theta_B)$ | Shadow boundary angle for diffracted field |


Calculate angle ALPHA and incident shadow boundary angles TH1 and TH2
Calculate diffracted shadow boundary angle THEB
RETURN

| KEY VARIABLES | | | INPUT/ OUTPUT |
|---|---|---|---|
| ALPHA | ($\alpha$) | Incident shadow boundary angles measured from negative z-axis | |
| GAM1 | ($\gamma_1$) | Projected angle defined by XS and RIM POINT P: measured from the x-axis | |
| GAM2 | ($\gamma_2$) | Projected angle defined by XS and RIM POINT $P_{i+1}$ measured from the x-axis | |
| ML | | Index of the lower rim cut by PHI | |
| MU | | Index of the upper rim cut by PHI | |
| RHOS | ($\rho_s$) | Reflected shadow boundary distances | (O) |
| X(1) | | X-coordinate of the upper intersecting rim point | |
| X(2) | | X-coordinate of the lower intersecting rim point | |
| X1 | | X-component of the distance from the source to rim point $P_i$ | |
| X2 | | X-component of the distance from the source to rim point $P_{i+1}$ | |
| Y(1) | | Y-coordinate of the upper intersecting rim point | |
| Y(2) | | Y-coordinate of the lower intersecting rim point | |
| Y1 | | Y component of the distance from the source to rim point $P_i$ | |
| Y2 | | Y component of the distance from the source to rim point $P_{i+1}$ | |

## CODE LISTING

```
      SUBROUTINE SBDY(N,RIM,XS,PHI,TH1,TH2,THEB)
C
C     *** THIS SUBROUTINE CALCULATES THE SHADOW BOUNDARY ANGLES
C         FOR SPILLOVER FIELDS AS WELL AS EDGE DIFFRACTED FIELDS
C         OF A PARABOLIC REFLECTOR ANTENNA.
C
C     *** THE RANGE OF INPUT PHI ANGLE IS IN (-180.,180.)
C
      DIMENSION RIM(67,2),ALPHA(2),X(2),Y(2),Z(2),XS(3),RHOS(2)
      DIMENSION MU(2),ML(2),SIGN(2)
      LOGICAL LTEST,LDEBUG
      COMMON /RFBDY/RHOS
      COMMON /FOCAL/F,ZOP
      COMMON /PIS/PI,TPI,DPR
      COMMON /OUT/NW
      COMMON /TEST/LDEBUG,LTEST,NTEST
      IF (LTEST) WRITE (6,*) N,PHIR
      HPI=PI/2.
      THPI=3.*HPI
      PHIR=PHI/DPR
      TANP=TAN(PHIR)
      IF (PHIR.GT.0.) PHIPR=PHIR-PI
      IF (PHIR.LE.0.) PHIPR=PHIR+PI
      L1=0
      L2=0
C
C     *** L : # OF INTERSECTING POINTS ON THE APERTURE RIM CUT BY PHI
C               AND PHIP.
C         MU,ML: INDEX OF THE RIM POINT CORRESPONDING TO THE RIM SECTI
 ON
C               CUT BY PHI AND/OR PHIP RESPECTIVELY.
C
      DO 10 I=1,N
      X1=RIM(I,1)-XS(1)
      Y1=RIM(I,2)-XS(2)
      GAM1=ATAN2(Y1,X1)
      J=I+1
      IF (J.GT.N) J=1
      X2=RIM(J,1)-XS(1)
      Y2=RIM(J,2)-XS(2)
      GAM2=ATAN2(Y2,X2)
      IF (GAM1.GE.0..AND.GAM2.LT.0.) GAM2=GAM2+TPI
      IF (ABS(GAM1-GAM2).LT.PI) GO TO 6
      IF (TAN(GAM1)*TAN(GAM2).GT.0) GO TO 100
      GO TO 8
    6 IF (GAM2.GT.GAM1) GO TO 8
      TEMP=GAM1
```

```
      GAM1=GAM2
      GAM2=TEMP
    8 CONTINUE
      IF (PHIR.LT.GAM1.OR.PHIR.GE.GAM2) GO TO 9
      L1=L1+1
      MU(L1)=I
    9 IF (PHIPR.LT.GAM1.OR.PHIPR.GE.GAM2) GO TO 10
      L2=L2+1
      ML(L2)=I
   10 CONTINUE
      L=L1+L2
      IF (L.GT.2) WRITE (6,12) L
   12 FORMAT (/T10,'L=',I2,5X,'ABNORMAL RIM SHAPE',/)
      IF (LDEBUG) WRITE (6,14) L,L1,L2,MU(1),MU(2),ML(1),ML(2)
   14 FORMAT (/T10,'L=',I2,5X,6I5,/)
      IF (L.EQ.2) GO TO 15
C
C     *** L<2, EITHER PHI CUT TANGENT TO THE APERTURE RIM (L=1),
C                  OR MISSING THE APERTURE PLANE (L=0).
C
      TH1=180.
      TH2=180.
      RETURN
   15 IF (L1-1) 16,17,18
   16 M1=ML(1)
      M2=ML(2)
      SIGN(1)=-1.
      SIGN(2)=-1.
      GO TO 20
   17 M1=MU(1)
      M2=ML(1)
      SIGN(1)=1.
      SIGN(2)=-1.
      GO TO 20
   18 M1=MU(1)
      M2=MU(2)
      SIGN(1)=1.
      SIGN(2)=1.
   20 IF (LDEBUG) WRITE (6,-) M1,M2
C
C     *** CALCULATE THE COORDINATES OF THE INTERSECTING POINTS
C         (X(I),Y(I),Z(I)) AND THEIR CORRESPONDING ANGLES ALPHA(1) AND
C         ALPHA(2) MEASURED FROM THE NEGATIVE Z-AXIS.
C         ALL REFERRING TO THE SOURCE POINT XS.
C
      I1=M1
      DO 30 I=1,2
      X1=RIM(I1,1)-XS(1)
      Y1=RIM(I1,2)-XS(2)
      IF (LDEBUG) WRITE (6,32) X1,Y1
```

```
      I2=I1+1
      IF (I2.GT.N) I2=1
      X2=RIM(I2,1)-XS(1)
      Y2=RIM(I2,2)-XS(2)
      IF (LDEBUG) WRITE (6,32) X2,Y2
      IF (ABS(X1-X2).LT.1.E-3) GO TO 25
      SLP=(Y2-Y1)/(X2-X1)
      IF (ABS(PHI-90.).LT.1.E-3) GO TO 22
      X(I)=(X2*SLP-Y2)/(SLP-TANP)
      Y(I)=X(I)*TANP
      GO TO 28
   22 Y(I)=Y2-X2*SLP
      X(I)=0.
      GO TO 28
   25 X(I)=X1
      Y(I)=X(I)*TANP
   28 RHOS(I)=SIGN(I)*SQRT(X(I)*X(I)+Y(I)*Y(I))
      IF (LDEBUG) WRITE (6,-) I,I1,X(I),Y(I),RHOS(I)
      I1=M2
   30 CONTINUE
C
      IF (L1.EQ.1) GO TO 31
      IF (RHOS(1).GE.RHOS(2)) GO TO 31
      TEMP=X(1)
      X(1)=X(2)
      X(2)=TEMP
      TEMP=Y(1)
      Y(1)=Y(2)
      Y(2)=TEMP
      TEMP=RHOS(1)
      RHOS(1)=RHOS(2)
      RHOS(2)=TEMP
C
C     *** X(K),Y(K),Z(K) REFER TO THE REFLECTOR COORDINATE
C
   31 DO 40 K=1,2
      XP=X(K)
      YP=Y(K)
      X(K)=XP+XS(1)
      Y(K)=YP+XS(2)
      RHO2=X(K)**2+Y(K)**2
      Z(K)=RHO2/(4.*F)-ZOP
      IF (LTEST) WRITE (6,32) X(K),Y(K),Z(K)
   32 FORMAT (T20,3E12.4)
      D=SQRT(XP*XP+YP*YP)
      ZZ=XS(3)-Z(K)
      ALPHA(K)=BTAN2(D,ZZ)*DPR
      IF (LTEST) WRITE (6,35) K,ALPHA(K)
   35 FORMAT (T10,'ALPHA(',I1,')',F7.2)
   40 CONTINUE
```

```
C
C     *** SHADOW BOUNDARY ANGLES FOR SPILLOVER FIELD ***
C
      TH1=180.-SIGN(1)*ALPHA(1)
      TH2=180.-SIGN(2)*ALPHA(2)
C
C     *** SHADOW BOUNDARY ANGLE FOR EDGE DIFFRACTED FIELD ***
C
      THEB=HPI
      IF (ABS(Z(1)-Z(2)).LT.1.D-4) GO TO 70
      DRHO=SQRT((Y(2)-Y(1))**2+(X(2)-X(1))**2)
      DZ=Z(1)-Z(2)
      THEB=BTAN2(DRHO,DZ)
   70 IF (LTEST) WRITE (6,80) THEB
   80 FORMAT (/T15,'THEB =',F10.4,' RADIANS',/)
      RETURN
  100 WRITE (6,120)
  120 FORMAT (/T10,'*** ERROR : TWO CONSECUTIVE RIM POINTS MUST '
     2'LOCATE IN THE SAME QUADRANT OR ADJACENT QUADRANTS',/)
      CALL EXIT
      END
```

## REFERENCES

1. R.C. Rudduck, and S.H. Lee, "User's Manual for Numerical Electromagnetic Code (NEC)," Report 784508-19, September 1979, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract N00123-76-C-1371 for Naval Regional Procurement Office, Long Beach, California 90822

2. A.C. Ludwig, "The Definition of Cross Polarization," IEEE Trans. on Antennas and Propagation, AP-21, January 1973, pp. 116-119.

3. R.J. Marhefka, "Analysis of Aircraft Wing-Mounted Antenna Patterns," Report 2902-25, June 1976, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Grant No. NGL-36-008-138 for National Aeronauctics and Space Administration.

4. Numerical Electromagnetic Code (NEC) - Basic Scattering Code, Part II: Code Manual," F.W. Schmidt and R.J. Marhefka, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract N00123-76-C-1371 for Naval Regional Procurement Office, Long Beach, California 90822.

5. R.G. Kouyoumjian and P.H. Pathak, "A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly-Conduction Surface, Proc. IEEE, Vol. 62, November 1974, pp. 1448-1461.

6. R.G. Kouyoumjian, "The Geometrical Theory of Diffraction and Its Applications," Numerical and Asymptotic Techniques in Electromagnetics, edited by R. Mittra, Spring-Verlag, New York, 1975.

7. W.D. Burnside, N. Wang and E.L. Pelton, "Near Field Pattern Computations for Airborne Antennas," Report 4685-4, June 1978, The Ohio State University ElectroScience Laboratory, Department of Electrical Engineering; prepared under Contract N00019-77-C-0299 for Department of the Navy, Naval Air Systems Command. (AD/A 057 244)

8. J. Boersma, "Computation of Fresnel Integrals," Math. Comp., Vol. 14, 1960, p. 380.

DATE
ILMED
-8