NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE
A COMPUTER PROGRAM TO CALCULATE RADIATION PROPERTIES OF REFLECTOR ANTENNAS

May 1978

NASA Technical Memorandum 78721

Pradeep K. Agrawal

NASA
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665
A COMPUTER PROGRAM TO CALCULATE RADIATION PROPERTIES OF REFLECTOR ANTENNAS

Pradeep K. Agrawal
INDEX

1. Summary
2. Introduction
3. Theory
4. Computer Program
5. An Example
6. Subroutine APERTUR
7. Subroutine QUANTIZ
8. Subroutine RADPAT
9. Subroutine INTERP
10. Other Subroutines
11. Conclusions and Remarks
12. References
1. SUMMARY

A computer program to calculate the radiation properties of the reflector antennas is presented. In its present form the program can be used for paraboloidal, spherical, or ellipsoidal reflector surfaces. However, it can be easily modified to handle any surface that can be expressed analytically. The program is general enough to allow any arbitrary location and pointing angle for the feed antenna. The effect of blockage due to the feed horn is also included in the computations.

The computer program is based upon the technique of tracing the rays from the feed antenna to the reflector to an aperture plane. The far field radiation properties are then calculated by performing a double integration over the field points in the aperture plane. To facilitate the computation of double integral, however, the field points are first aligned along the equispaced straight lines in the aperture plane. The computation time is relatively insensitive to the absolute size of the aperture and even though no limits on the largest reflector size have been determined, the program has been used for reflector diameters of 1000\(\lambda\).
2. INTRODUCTION

The purpose of this report is to present a detailed description of a computer program called REFLCTR which has been written to calculate the radiation properties of reflector antennas whose surfaces can be analytically expressed in Cartesian coordinates. The reflector feed can have any arbitrary location (does not have to be at focus) and arbitrary pointing angle. In its present form, the program is set up to handle either of a paraboloidal \( y^2+z^2=4f(f+x) \) where \( f \) is the focal length, spherical \( x^2+y^2+z^2=r^2 \) where \( r \) is the radius, or ellipsoidal \( \left(\frac{x^2}{a^2}\right)+\left(\frac{y^2+z^2}{b^2}\right)=1 \) where \( a \) and \( b \) are semi major and semi minor axes) reflector surface whereas the reflector can be defined by the curved surface of a frustum of any one of the above mentioned shapes.

The theory associated with the computer program is briefly presented in Section 3 and then the computer program is presented in Section 4. An example is presented in Section 5 for a spherical reflector to demonstrate the methodology of using the computer program. A detailed description of some of the subroutines is given in Sections 6-10. Section 11 shows how the program can be very easily modified to handle any other reflector surface that can be expressed analytically.
3. THEORY

The underlying theory is described in detail in Reference 1. For the purpose of understanding the computer program, however, a brief description of the theory is presented.

Consider a reflector and its feed as shown in Figure 1. The reflector surface is known in the \((x,y,z)\) coordinate system, \(0\) being the origin of this reflector coordinate system. The feed antenna which is allowed to have any arbitrary location and direction is directed along the negative \(x'\)-axis of its own coordinate system whose origin is \(0'\). The radiation pattern of the feed antenna is assumed to be known in the primed or the feed coordinate system. And therefore, to be able to write the equation of a ray emanating from the feed antenna in the reflector coordinate system.

![Figure 1 -- Feed and Reflector Coordinate Systems](image_url)
system, a transformation from the feed to the reflector coordinates is needed. In general, these two coordinate systems are related to each other by a translation and a rotation, each of which will have three orthogonal components. The computer program is written such that the translation is defined as the vector $0^\prime 0$ expressed in the primed coordinate system i.e., the coordinates of point 0 in the feed coordinate system. The three components of this vector $0^\prime 0$ are called FEED(1), FEED(2) and FEED(3) in the computer program.

The rotation needed to relate the primed and the unprimed coordinate systems also has three components which are called ALPHA, BETA, and GAMMA in the computer program. These three angles are defined such that if the primed coordinate system is rotated by ALPHA, BETA, and GAMMA, the primed coordinate axes become parallel to the unprimed coordinate axes. Specifically, ALPHA is the rotation of the $x'y'$-plane about the $z'$-axis needed to make the $y'$-axis parallel to the $y$-axis, and as shown in Figure 2(a) for a simple case, ALPHA allows the feed to be pointed anywhere along a horizontal line on the reflector. BETA is the rotation of the $y'z'$-plane about the $x'$-axis to align the $z'$-axis with the $z$-axis and allows the feed antenna to have any arbitrary direction of polarization, i.e., not necessarily coincident with either the $z$- or the $y$-axis as shown in Figure 2(b). Finally, GAMMA is the rotation of the $xz'$-plane about the $y'$-axis to make the $x'$-axis parallel to the $x$-axis. GAMMA allows the feed to be pointed
Figure 2 -- Angles ALPHA, BETA, and GAMMA
anywhere along a vertical line on the reflector as shown in Figure 2(c). The rotation angles are considered positive for counter clockwise rotation when looking in the negative direction along the axis of rotation. Thus, for example, if the feed antenna was directed upward along the offset focal axis by an angle $\theta_0$, GAMMA will be $-\theta_0$. To describe the general orientation of a feed, however, all three of ALPHA, BETA, and GAMMA will be needed.

The feed pattern is needed at enough equally separated values of $\theta'$ and $\phi'$ so that the rays emanating from the feed over these ranges of $\theta'$ and $\phi'$ more than illuminate the reflector, i.e., the rays with the upper and the lower limit values of $\theta'$ and $\phi'$ miss the reflector. A typical ray along $(\theta_i, \phi_j)$ direction is shown in Figure 3. Index $i$ is used to denote the change in the angle $\theta'$ which has NT equispaced values between $\theta_1$ and $\theta_{NT}$. Similarly, index $j$ is used to denote the change in the angle $\phi'$ which has NP equispaced values between $\phi_1$ and $\phi_{NP}$ making the total number of rays considered emanating from the feed equal to the product of NT and NP. As mentioned above, the limits on $\theta'$ and $\phi'$ are required to be such that at least the rays with $\theta' = \theta_1$ and $\theta'_{NT}$ and the rays with $\phi' = \phi_1$ and $\phi'_{NP}$ miss the reflector. The reason for this is explained later.

Associated with each ray (i.e., with each $(i,j)$) are five quantities — $E_\theta$ and $E_\phi$, the two components of the electric field along the ray, the phase of the electric field, and the
angles \((\theta'_i, \phi'_j)\) specifying the direction of the ray. These five quantities are stored for each ray in a three-dimensional array \(P(S, NT, NP)\) which is shown pictorially in Figure 4. Using the three pointing angles ALPHA, BETA, and GAMMA of the feed with respect to the \((x,y,z)\) coordinate axes, the Euler Rotation Matrix (dimensioned as \(A(3,3)\)) is calculated. Use of the matrix \(A\) allows one to write a unit vector along each of the rays in the \((x,y,z)\) coordinate system, following which, the point of intersection of the ray with the reflector \((x_0', y_0', z_0')\) is found, and the \(x-, y-, z-\) components of the unit normal, \(\text{NHAT}(1), \text{NHAT}(2),\) and \(\text{NHAT}(3)\) are calculated at that point (Figure 5). Knowing the vector along the incident ray and along the normal at the point of incidence,
the x-, y-, z-components of the vector along the reflected ray, SR(1), SR(2), and SR(3) are calculated. Also, with the knowledge of the unit normal vector, and $E_\theta$ and $E_\phi$ of the incident ray, the three cartesian components of the electric field in the reflected ray, ER(1), ER(2), and ER(3) are evaluated.

A plane parallel to the yz-plane is defined as the aperture plane. $X_c$ is the distance of this plane from the origin and it is chosen such that the aperture plane lies in front of and near the edge of the reflector. The boundary of the aperture plane is a contour formed by the intersection of the aperture plane and the rays reflected from the edge of the reflector. This contour in general can be a combination

---

**Figure 4 -- P Array and the Feed Pattern Information**
of two ellipses. For the sake of simplicity in calculation, though, the boundary of the aperture plane is approximated by a single best fit ellipse. The parameters that define this ellipse are its half major axis (HFMAEX, along the y-direction), the half minor axis (HFMIEX, along the z-direction), and the coordinates of the center of the ellipse (XC, YC, ZC).

Those rays emanating from the feed which will miss the reflector are assumed to be reflected as though the reflector surface extended past its boundary. The point of intersection of each of the reflected rays with the aperture plane called (Y, Z) is calculated. Using the points which fall immediately inside and outside the aperture plane elliptical boundary, the edge points are interpolated at the boundary of the ellipse.

Figure 5 -- Geometry of the Reflector Antenna
The points outside the aperture plane elliptical boundary are then discarded. For the points that fall inside the aperture plane ellipse and the ones interpolated on the ellipse boundary, $Y, Z, ER(2), ER(3)$, and phase are stored in the $P$ array.

The points of intersection of the reflected rays with the aperture plane are in general, not uniformly distributed over the aperture plane. However, in order to facilitate the calculation of the far field radiation pattern from the aperture plane distribution, these points in the aperture plane are aligned along equispaced $y$-constant lines. Or, equivalently, the magnitude and the phase at each of the intersection points are arbitrarily assigned to the nearest point on a $y$-constant line. In this way, the double integral to be performed on the aperture plane reduces to an integral over straight line integrals along $y$-constant lines. The spacing between the $y$-constant lines must be chosen very carefully. For example, shown in Figure 6 is a computer generated plot of the points of intersection of the reflected rays and the aperture plane. Observe that, as stated earlier, these are not uniformly distributed. Figure 7 is another computer generated plot showing the locations of the same points after they have been aligned along equispaced $y$-constant lines. Notice the blanks created in the lines as a result of aligning the points. The spacing between $y$-constant lines needs to be chosen to minimize these blanks and to place them as near to the aperture edge (low field strength region).
as possible. The best value for line spacing has been found to be approximately equal to the average spacing of the points (Figure 6) along the y-direction. Since the integration time depends upon the number of points (of intersection of the reflected rays and the aperture plane) which in turn is determined by the angular increment between the rays used in the feed pattern, the integration time is relatively insensitive to the absolute size of the reflector.
Figure 7 -- Points of Intersection of the Reflected Rays and the Aperture Plane after Aligning along Equispaced \( y \) = Constant Lines
4. COMPUTER PROGRAM

The computer program has a main program that calls several subroutines, some of which call other subroutines. The order in which the subroutines are called is illustrated in Figure 8. The remainder of this section describes the overall working of the program and the subroutines. The subroutines shown in double lined boxes are also discussed individually in greater detail in the next sections. The subroutines shown in the hatched boxes are the adaptations of standard library subroutines. A brief description and listing of these is given in Section 10.

Figure 8 -- Block Diagram of the Computer Program
MAIN: In addition to the ordinary input and output files, the REFLCTR program (Figure 9) uses two extra files - tape 8 and tape 20. Tape 8 is used to temporarily store a part of the aperture plane data while the remainder of the data is being sorted and rearranged. Tape 20, dictated by the value of a variable called APRDTA is used to store the aperture plane data both before and after being aligned along the equispaced y=constant lines. This information on Tape 20 is later printed out and is also used by subroutine APRPLT to generate plots similar to those shown in Figures 6 and 7.

The P array contains the information on all the rays emanating from the feed antenna, five quantities being associated with each ray viz., $E_\theta$, $E_\phi$, phase, $\theta'$, and $\phi'$. MAXPTS, which is the maximum number of rays that can be stored in the P array, is declared in the DATA statement and is also used in dimensioning the P array.

NPUT: The purpose of this subroutine (Figure 10) is to read the input parameters and the feed pattern information. The input parameters are read from six cards which are described below. The feed pattern information, however, is read in a separate subroutine called FILL which is called by NPUT. All input parameters are in the same units of measure.

Cards 1 and 2: These are called title cards and contain any alphanumerical information in columns 1-80 for title or identification purposes.
PROGRAM REFLECT(NINPUT,OUTPUT,TAPER,TAPE20,  
TAPE7=INPUT,TAPE6=OUTPUT,PUNCH)  
DIMENSION YPLT(1),ZPLT(1),ERORDB(1),Y(1),Z(1)  
COMMON P(5,2750)  
COMMON/PARAMS/TITLE(16),AORORF,XLAM,GRID,SURFACE,APRDTA,FEED(3),  
   ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAFX,HFMIEX,RMTP,BMPP,  
   NT,NP,NPOINT,MAXPTS,RFELL  
COMMON/MATH/PI,P12,PID2,DTOR,RTON  
EQUIVALENCE (YPLT,P(1)),(ZPLT,P(2751)),(ERORDB,P(5501)),  
   (Y,P(8251)),(Z,P(11001))  
DATA MAXPTS/2750/,NT,NP,NPOINT/0,0,0/  
776 FORMAT(*)                  FINISHED INPUT ------------------------*)  
777 FORMAT(*)                  FINISHED APERTR ------------------------*)  
778 FORMAT(*)                  FINISHED QUANTIZ ------------------------*)  
779 FORMAT(*)                  FINISHED RADPAT ------------------------*)  
REWIND 20  
CALL NPUT(P)  
PRINT 776  
CALL APERTR(P,NT,NP)  
PRINT 777  
CALL QUANTIZ(P,MAXPTS)  
ENDFILE 20  
PRINT 778  
CALL RADPAT(P,MAXPTS)  
PRINT 779  
REWIND 20  
IF(APRDTA.GT.0.0) CALL APRPLT(YPLT,ZPLT,ERORDB,Y,Z,MAXPTS)  
STOP  
END

Figure 9 -- The Main Computer Program
SUBROUTINE NPUT(P)
COMMON/PARAMS/TITLE(16),NORDF,XLAM,GRID,SURFACE,APRDTA,FEED(3),
  ALPHA,BETA,GAMMA,XC,YC,ZC,FMFX,FMIX,BMTP,BMPP,
  NT,NP,NPOINT,NMAXPTS,BELLP
COMMON/BLOCKG/YCBL,ZCBL,FMABL,FMIBL
COMMON/PATTERN/PHI(3),THETA(3)
COMMON/MATH/PI,P12,P122,DTOR,RTOD
INTEGER TITLE
READ 200,TITLE
READ *,AORDRF,XLAM,GRID,SURFACE,APRDTA,BELLP
READ *,FEED,ALPHA,BETA,GAMMA
READ *,XC,YC,ZC,FMFX,FMIX,YCBL,ZCBL,FMABL,FMIBL
READ *,PHI,THETA
IF(APRDTA.GT.0.0) WRITE(20,555)TITLE
IF(APRDTA.GT.0.0) WRITE(20,556)FEED,AORDRF,XLAM,GRID,ALPHA,BETA,TAPE 20
  GAMMA,XC,YC,ZC,FMFX,FMIX
  TAPE 10
  TAPE 20
IF(SURFACE)140,150,160
  PRINT 579,TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,AORDRF,BELLP
  GO TO 170
140 PRINT 580,TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,AORDRF
  GO TO 170
150 PRINT 581,TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,AORDRF
160 PRINT 582,XC,YC,ZC,FMFX,FMIX,GRID,YCBL,ZCBL,FMABL,FMIBL
  THETA,PHI
200 FORMAT(8A10)
555 FORMAT(1X,RA10)
556 FORMAT(1X,F15.4)
579 FORMAT(1H1,///,11X,*ELLIPTICAL REFLECTOR FAR FIELD RADIATION */
  * PATTERN CALCULATION*/// *8A10/* *8A1
  0/* WAVELENGTH OF ELECTRIC FIELD....................................*F9.4/
  * LOCATION OF COORDINATE ORIGIN WRT FEED (X,Y,Z)..............*3F7.2
  /* FEED ROTATION ANGLES (ALPHA,BETA,GAMMA)......................*3F7.2
  /* MAJOR AXIS OF THE ELLIPTICAL REFLECTOR......................*F7.2/

Figure 10 -- Subroutine NPUT
* MINOR AXIS OF THE ELLIPTICAL REFLECTOR..............*F7.2*)

580 FORMAT(H1,///,1X,*SPHERICAL REFLECTOR FAR FIELD RADIATION */
* PATTERN CALCULATION/ */ *RA10/* *RA1
* 0/* WAVELENGTH OF ELECTRIC FIELD..............................*F9.4/
* LOCATION OF COORDINATE ORIGIN WRT FFED (X,Y,Z)...........*3F7.2
* FEED ROTATION ANGLES (ALPHA,BETA,GAMMA)...................*3F7.2
* RADIUS OF THE REFLECTOR SPHERE..............................*F7.2*)

581 FORMAT(H1,///,1X,*PARABOLIC REFLECTOR FAR FIELD RADIATION */
* PATTERN CALCULATION/ */ *RA10/* *RA1
* 0/* WAVELENGTH OF ELECTRIC FIELD..............................*F9.4/
* LOCATION OF COORDINATE ORIGIN WRT FFED (X,Y,Z)...........*3F7.2
* FEED ROTATION ANGLES (ALPHA,BETA,GAMMA)...................*3F7.2
* FOCAL LENGTH OF THE REFLECTOR.............................*F7.2*)

582 FORMAT(* APERTURE PLANE LOCATION (XC)..........................*F7.2/
* COORDINATES OF THE APERTURE PLANE CENTER................*2F7.2
* HALF MAJOR AXIS OF APERTURE PLANE (ALONG Y).............*F7.2/
* HALF MINOR AXIS OF APERTURE PLANE (ALONG Z)............*F7.2/
* GRID SIZE USED FOR NUMERICAL INTEGRATION................*F9.4/
* FEED SHADOW CENTER COORDINATES IN APERTURE PL...........*2F7.2
* HALF MAJOR AXIS OF FEED SHADOW............................*F7.2/
* HALF MINOR AXIS OF FEED SHADOW............................*F7.2/
* THETA RANGE FOR FFED PATTERN (L,H,I - DEGREES)..........*3F7.2
* PHI RANGE FOR FFED PATTERN (L,H,I - DEGREES)............*3F7.2
)

590 FORMAT(* ---- INSUFFICIENT WORK STORAGE, NEEDFD *15* AVAILABLE IS
* ONLY *15* ---- * )
PI=ACOS(-1.0)
PI2=PI+PI
PID2=0.5*PI
DTHOR=PI/180.
RTOR=180./PI
NP=(PHI(2)-PHI(1))/PHI(3)+1.5
NT=(THETA(2)-THETA(1))/THETA(3)+1.5

Figure 10 -- Subroutine NPUT -- Continued
NPOINT=NT*NP
IF(NPOINT .GE. MAXPTS) GO TO 600
CALL FILL(P,NT,NP)
RETURN
600 PRINT 590, NPOINT, MAXPTS
STOP
END

Figure 10 -- Subroutine NPUT -- Continued
### Card 3: General information about the reflector.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AORORF</td>
<td>Semi major axis of the ellipsoidal reflector surface, or the radius of the reflector sphere, or the focal length of the paraboloid.</td>
</tr>
<tr>
<td>XLAM</td>
<td>Wavelength in free space.</td>
</tr>
<tr>
<td>GRID</td>
<td>Spacing between the y=constant lines in the aperture plane.</td>
</tr>
<tr>
<td>SURFACE</td>
<td>A number that is negative, zero, or positive for ellipsoidal, spherical, or paraboloidal surface, respectively.</td>
</tr>
<tr>
<td>APRDTA</td>
<td>A variable that should be positive if aperture plane plots and a printout of the aperture plane data is desired.</td>
</tr>
<tr>
<td>RELLP</td>
<td>Semi minor axis of the ellipsoidal reflector surface. Even if the reflector is not an ellipsoid, a number should still be read.</td>
</tr>
</tbody>
</table>

### Card 4: Feed information.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEED</td>
<td>A one-dimensional array containing coordinates of the (x,y,z) origin with respect to the feed coordinate system.</td>
</tr>
<tr>
<td>ALPHA, BETA, GAMMA</td>
<td>Three rotation angles that define the pointing angle of the feed. (See Figure 2 for definition).</td>
</tr>
</tbody>
</table>

### Card 5: Aperture plane information.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC,YC,ZC</td>
<td>x-,y-,z-coordinates of the center of the aperture plane ellipse.</td>
</tr>
<tr>
<td>HFMAEX,HFMIEX</td>
<td>Semi major and minor axes (along y- and z-directions) of the aperture plane ellipse.</td>
</tr>
<tr>
<td>YCRL,ZCRL</td>
<td>y- and z-coordinates of the center of the feed shadow ellipse in the aperture plane.</td>
</tr>
<tr>
<td>HFMABL,HFMIRL</td>
<td>Semi major and minor axes (along y- and z-directions) of the feed shadow ellipse on the aperture plane.</td>
</tr>
</tbody>
</table>
Card 6: Illumination information.

**PHI**
A one-dimensional array containing the initial, the final, and the increment values of $\phi'$ in degrees over which the feed rays illuminate the reflector in the feed coordinate system.

**THETA**
A one-dimensional array containing the initial, the final, and the increment values of $\theta'$ in degrees over which the feed rays illuminate the reflector in the feed coordinate system.

Next, the number of $\theta'$ values $N_T$ and the number of $\phi'$ values $N_P$ are calculated. These are maximum values of indices $i$ and $j$ respectively in Figure 4. The product of $N_T$ and $N_P$ called $N_{POINT}$ is the total number of rays emanating from the feed and should be less than the value of $MAXPTS$ which is set in the main program.

Finally, NPUT calls subroutine FILL where $E_\theta'$, $E_\phi'$, phase, $\theta'$, and $\phi'$ values associated with each ray are read into the $P$ matrix.

**FILL:** Associated with each of the rays emanating from the feed antenna, there are five quantities viz., $E_\theta'$, $E_\phi'$, phase, $\theta'$, and $\phi'$. In this subroutine these quantities are stored in the $P$ matrix for all the rays emanating from the feed antenna. One of the requirements on filling the $P$ array is that the beam maximum of the feed pattern should be along the negative $x'$-axis of the feed coordinate system (Figure 1). If the feed pattern is known analytically as a function of $\theta'$, and $\phi'$ then filling the $P$ matrix is merely a matter of evaluating $E_\theta'(\theta',\phi')$, $E_\phi'(\theta',\phi')$, and phase for the needed values of $\theta'$ and $\phi'$ and storing them in the appropriate
locations as shown in Figure 4. For example, in the listing of the subroutine FILL presented in Figure 11, E_0 and E_φ are computed for each (θ', φ') and then stored in the matrix P for a diagonal horn with d=3.9 cm at 11.2 GHz. The phase for each ray is assumed to be zero.

However, if the feed pattern is known in only one plane, or even two orthogonal planes, a scheme has to be devised to evaluate the E_0 and the E_φ components for $0_1 < \theta' < 0_N$ and $\phi_1 < \phi' < \phi_N$. The subroutine FILL will have to be rewritten accordingly. In the next Section an example is presented where a circularly symmetric feed pattern is known in only one plane. A simple algorithm is used to evaluate the field values for the other angles needed. This algorithm is shown in the listing of the subroutine FILL in Section 5.

**APERTUR**: This subroutine computes the points of intersection of the aperture plane with the rays reflected from the reflector surface. If a point does not fall inside or on the aperture plane ellipse, its information is discarded. At the same time a separate count is kept of the number of points which lie inside and on the aperture plane ellipse. These are called NINTR and NEDGE respectively, and are also printed out. Simultaneously, the direction of the reflected ray corresponding to the incident ray with maximum field intensity is determined. In general, the direction of the beam maximum of the total secondary pattern is almost coincident with this reflected ray. The components of a unit vector along this ray are also printed out. If the value of APRDTA is positive, the data associated with the edge points and with the interior points are written on tape 20.
SUBROUTINE FILL(P,NTX,NPX)
COMMON/PATTERN/PHI(3),THETA(3)
COMMON/MATH/P1,P12,P1D,P2,NTOR,RTND
DIMENSION P(5,NTX,NPX)
CALL SETM(0.0,P,5*NPX*NTX)
THR=THETA(1)*NTOR
DN 300 I=1,NTX
FAC2=P1*3.9*COS(THR)/2.6A
FAC0=COS(FAC2)/(1.0-4.0*FAC2*FAC2/(PI*PI))
IF (FAC2.NE.0.0) GN TO 50
FACC=1.0
GN TO 51
50 FAC0=FAC2/FAC2
51 CONTINUE
PHR=(PHI(1)-180.0)*NTOR
DN 200 J=1,NPX
FAC1=P1*3.9*SIN(THR)*SIN(PHR)/2.6A
FAC0=COS(FAC1)/(1.0-4.0*FAC1*FAC1/(PI*PI))
IF (FAC1.NE.0.0) GN TO 60
FACA=1.0
GN TO 61
60 FAC0=FAC1/FAC1
61 CONTINUE
XLZ=FACA*FACR
XLY=FACC*FACD
XMUL1=SIN(PHR)*COS(THR)
XMUL2=SIN(THR)*COS(PHR)
P(1,1,J)=XLZ*XMUL1-XLY*XMUL2
P(2,1,J)=XLY*XMUL1-XLZ*XMUL2
P(1,1,J)=ABS(P(1,1,J))/2.0
P(2,1,J)=ABS(P(2,1,J))/2.0
P(4,1,J)=THR
P(5,1,J)=PHR+P1
PHR=PHR+PHI(3)*NTOR
200 CONTINUE
THR=THR+THETA(3)*NTOR
300 CONTINUE
RETURN
END

Figure 11 -- Subroutine FILL

ORIGINAL PAGE IS OF POOR QUALITY
QUANTIZE: The points of intersection of the reflected rays with the aperture plane (which from now on will be referred to simply as 'points' in the aperture plane) are generally not uniformly distributed. To simplify performing the double integration over the aperture plane, this subroutine aligns the points in the aperture plane along the equispaced $\gamma$-constant lines. This aligning of points (also called quantizing of points) is done in two steps - first to the points on the perimeter of the aperture plane ellipse or to the edge points, and then to the internal points. The information related to the equispaced lines or grid bars and the resulting number of quantized points in printed out in appropriate format.

RADPAT: Performing a double integration over the aperture plane points, this subroutine calculates the far field values of $F_x$ and $F_z$ ($F_x$ being zero in the far field) for any $\theta$ and $\phi$. The subroutine is set up such that a far field pattern is requested by first picking a plane (constant $\theta$-, or constant $\phi$-plane) and then specifying in that plane the initial value, the final value, and the increment size of the other variable angle ($\phi$ or $\theta$, respectively). The output of this subroutine is as follows:

1. A table listing the variable angle and the quantities $F_x/(E_z)_{max}$, $F_y/(E_y)_{max}$, $F_z/(E_y)_{max}$, $F_y/(E_y)_{max}$ and $(E_y^2 + E_z^2)/(E_y)_{max}^2 + (E_z)_{max}^2$ in decibels for each angle,

2. Values of $(E_z)_{max}$ and $(E_y)_{max}$,

3. Line printer plots of normalized $F_z$ and $F_y$ in decibels.
A unique feature of this subroutine which is illustrated in examples 1, 2, and 3 below is that in addition to requesting far field patterns in planes and for angles specified by numerical values (example 1), the patterns can also be requested by using $\theta$ and $\phi$ values related to the beam maximum position of the secondary pattern as shown in examples 2 and 3. Refer to format statement number 110 in the listing of subroutine RADPAT in Section 8.

Ex. 1: $\Theta$ETA  90.0 PHI  -10.0  0.0  0.5

In $\theta = 90^\circ$ plane, the far field is calculated for $\phi$ between -10 to 0\degree in steps of 0.5\degree.

Ex. 2: $\Theta$MAX-T  -4.0 PHI  -5.0  2.0  1.0

In $\theta = (\theta_{BMAX} - 4^\circ)$ plane, the far field is calculated for $\phi$ values between $(\phi_{BMAX} - 5^\circ)$ to $(\phi_{BMAX} + 2^\circ)$ in steps of 1.0\degree.

Ex. 3: $\Theta$MAX-P  10.0$\Theta$MAX-T  8.0  10.0  0.1

In $\phi = (\phi_{BMAX} + 10^\circ)$ plane, the far field is calculated for $\theta$ values between $(\theta_{BMAX} + 8^\circ)$ to $(\theta_{BMAX} + 10^\circ)$ in steps of 0.1\degree.
APRPLT: This subroutine, a listing of which is given in Figure 12, is called by the main program if APRDTA is positive. In this subroutine the information about the points in the aperture plane is read from tape 20 for both before and after quantizing. This information was written on tape 20 by subroutines APERTUR and QUANTIZ respectively. Two plots of the locations of the aperture plane points similar to Figures 6 and 7 are then generated but with an added feature, which is that different symbols are used every time the field strength changes in steps of 3 dB. For example, the output of subroutine APRPLT for a case shown in Figure 6 will be as shown in Figure 13.

Notice that the equivalence statement used in the main program makes it possible to restore the aperture plane data from tape 20 in newly defined one-dimensional arrays without requiring additional storage.
SUBROUTINE APRPLT (YPLT, ZPLT, EORDB, Y, Z, NTXNX)
COMM/N/PRAMS/TITLE(16), ADRORF, XLAM, GRID, SURFACE, APRDTA, FEED(3),
   ALPH@, RETA, GAMMA, XC, YC, ZC, HFMAEX, HFMIEX, BMTP, BMPP,
   NT, NP, NPOINT, MAXPTS, BELLP
DIMENSION YPLT(NTXNX), ZPLT(NTXNX), EORDB(NTXNX), Y(NTXNX),
   Z(NTXNX), INFO(8)
DATA SAME/999., HGT/0.36/, IP, ICOUNT/0, 0/
10 FORMAT(8A10)
100 FORMAT(25X, 2F10, 4, 2F10, 7)
CALL PSEUDO
ISF=(HFMAEX+HFMIEX)/R0, 0
SF=5*ISF+5
YORG=AIN(T(YC-(HFMAEX+5))
ZORG=AIN(T(ZC-(HFMIEX+5))
500 READ(20,10) INFO
IF(INFO(1).NE.10H***********) GO TO 500
510 READ(20,10) INFO
IF(EOF(20)) 700, 511
511 IF(INFO(1).EQ.10H $ ) GO TO 600
IF(INFO(1).EQ.10H QUANTIZED) GO TO 700
GO TO 510
600 IP=IP+1
DECODE(65, 100, INFO) YPLT(IP), ZPLT(IP), EY, EZ
EORDB(IP)=SQRT(EY**2+EZ**2)
GO TO 510
700 ICOUNT=ICOUNT+1
EMAX=-1, 0
DO 710 I=1, IP
IF(EORDB(I).GT. EMAX) EMAX=EORDB(I)
710 CONTINUE
DO 720 I=1, IP
IF(EORDB(I).EQ.0.0) GO TO 715
EORDB(I)=20.0*ALOG10(EORDB(I)/EMAX)

Figure 12 -- Subroutine APRPLT
IF(EORDB(I).LE.-59.0) EORDB(I)=-59.0
GO TO 720
715 EORDB(I)=-60.0
720 CONTINUE
DO 753 ILEVEL=1,11
DRLVEL=-ILEVEL*3.0
J=1
DO 752 I=1,IP
IF (ILEVEL.EQ.11) DRLVEL=-59.0
IF(EORDB(I).LE.DRLVEL) GO TO 752
EORDB(I)=-60.0
Y(J)=YPLT(I)
Z(J)=ZPLT(I)
J=J+1
752 CONTINUE
IF (ILEVEL.EQ.1) ISYM=12
IF (ILEVEL.EQ.2) ISYM=1
IF (ILEVEL.EQ.3) ISYM=22
IF (ILEVEL.EQ.4) ISYM=21
IF (ILEVEL.EQ.5) ISYM=12
IF (ILEVEL.EQ.6) ISYM=1
IF (ILEVEL.EQ.7) ISYM=22
IF (ILEVEL.EQ.8) ISYM=21
IF (ILEVEL.EQ.9) ISYM=12
IF (ILEVEL.EQ.10) ISYM=1
IF (ILEVEL.EQ.11) ISYM=22
Y(J)=YORG
Z(J)=ZORG
Y(J+1)=SF
Z(J+1)=SF
CALL LNPLOT (Y,Z,J-1,1,-1,ISYM,1,0)
753 CONTINUE
CALL AXES(0.,0.,0.,16.,YORG,SF,-1.,5.0,6HY AXIS,HGT,-6)

Figure 12 -- Subroutine APRPLT -- Continued
CALL AXES(0., 0., 90., 16., ZORG, SF, -1., 5.0, 6Hz AXIS, HGT, 6)
IP = 0
IF (ICOUNT .EQ. 1) CALL NFRAME
IF (ICOUNT .EQ. 1) GO TO 510
IF (ICOUNT .EQ. 2) CALL CALPLT(0.0, 0.0, 0.999)
RETURN
END

Figure 12 -- Subroutine APRPLT -- Continued
Figure 13 -- Points of Intersection of the Reflected Rays and the Aperture Plane Showing the Contours of Constant Electric Field Amplitude
5. AN EXAMPLE

A spherical reflector shown in Figure 14 is fed by a circular corrugated horn. The horn is z-polarized and its E-plane pattern is given.

SOLUTION: It will be assumed that the horn has a circularly symmetrical far field radiation pattern with no cross polarized component. The FILL subroutine, which is listed in Figure 15, is then written such that it reads the E-plane field values in decibels (which are also the H-plane field values) and calculates the field values for all the intermediate values of θ' and φ'.

The geometry of the reflector is such that it subtends a ±20.4° angle at the feed. However, as pointed out in

![Figure 14 -- Dimensions of the Reflector Antenna](image-url)
Section 3, the feed pattern must also be used for angles greater than those needed to illuminate the reflector. In the present case it has been used over a $+23^\circ$ range. The boundary of the aperture plane for this axially fed reflector is a circle whose diameter is chosen such that it collects all the rays reflected by the reflector. Also, the location of the aperture plane is chosen such that it lies very near the edge of the reflector.

**INPUT-OUTPUT:** A listing of input data cards and the output data along with the generated aperture plane plots is presented.

```fortran
SUBROUTINE FILL(P,NTX,NPX)
COMMON/PATTERN/PHI(3),THETA(3)
COMMON/MATH,PI,P12,P1D2,DTOR,RTAN
DIMENSION P(5,NTX,NPX)
DIMENSION E(191),F2(24),IP(47)

6320 FORMAT (5F15.10)
CALL SETM(0.0,P,50,NPX,NTX)
READ 6320, (E(I),I=1,91)
DO 10 I=1,24
  E2(75-I)=F(I)
  10 CONTINUE
DO 50 I=1,24
  II=I+6
  DO 50 J=1,24
    JJ=J+6
    P(1+I,J)=E2(1)+E2(J)
    P(1,II+J)=P(1,I,J)
    P(1+I,J)=P(1+I,J)
    P(1,II,J)=P(1,I,J)
  50 CONTINUE
DO 60 I=1,NTX
  DO 60 J=1,NPX
    P(1+I,J)=10.0*(P(1+I,J)/20.0)
    P(4+I,J)=(THETA(1)+(I-1)*THETA(3))*DTOR
    P(5+I,J)=(PHI(1)+(J-1)*PHI(3))*DTOR
  60 CONTINUE
RETURN
END
```

Figure 15 -- Subroutine FILL
**INPUT:**

Spherical Reflector Example Using a Corrugated Horn Feed, 1 GHz.

<table>
<thead>
<tr>
<th>Horn Radius</th>
<th>3.333 wavelengths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half Flare Angle</td>
<td>3.43626 degrees</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>575.0, 0.0, 3.526, 0.0, 1.0, 0.7932.0</td>
<td></td>
</tr>
<tr>
<td>288.25, 0.0, 0.0, 0.0, 0.0, 0.0</td>
<td></td>
</tr>
<tr>
<td>-555.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.1, 0.0, 0.1, 0.0</td>
<td></td>
</tr>
<tr>
<td>157.0, 0.203, 0.1, 0.0, 67.0, 0.113, 0.1, 0.0</td>
<td></td>
</tr>
<tr>
<td>0.0000000000</td>
<td>-0.0933062610</td>
</tr>
<tr>
<td>-2.3989163877</td>
<td>-0.8495610232</td>
</tr>
<tr>
<td>-10.7058765923</td>
<td>-1.4639153550</td>
</tr>
<tr>
<td>-24.6841075952</td>
<td>-2.9645996039</td>
</tr>
<tr>
<td>-27.2127240439</td>
<td>-2.989308715</td>
</tr>
<tr>
<td>-39.6384707282</td>
<td>-3.8164737944</td>
</tr>
<tr>
<td>-36.743999190359</td>
<td>-3.941915117</td>
</tr>
<tr>
<td>-43.2980421326</td>
<td>-4.7810823357</td>
</tr>
<tr>
<td>-42.5188317849</td>
<td>-4.9549131124</td>
</tr>
<tr>
<td>-59.9126567715</td>
<td>-5.9082402291</td>
</tr>
<tr>
<td>-47.7161953847</td>
<td>-5.5645739480</td>
</tr>
<tr>
<td>-51.3640790135</td>
<td>-5.576394465</td>
</tr>
<tr>
<td>-54.9974338927</td>
<td>-5.6299013220</td>
</tr>
<tr>
<td>-53.7723770701</td>
<td>-5.5871074564</td>
</tr>
<tr>
<td>-53.7287870701</td>
<td>-5.5725761992</td>
</tr>
<tr>
<td>-53.6630692541</td>
<td>-5.5064228902</td>
</tr>
<tr>
<td>-56.5746920044</td>
<td>-5.5137989056</td>
</tr>
<tr>
<td>-61.4051255563</td>
<td>-5.5734380350</td>
</tr>
<tr>
<td>-66.7637058483</td>
<td>-5.7668025859</td>
</tr>
<tr>
<td>-69.6483956986</td>
<td>-6.4981366095</td>
</tr>
</tbody>
</table>

**Two title cards**

**General information**

**Feed information**

**Aperture Plane information**

**Illumination information**

**Pattern request**
SPHERICAL REFLECTOR FAR FIELD RADIATION PATTERN CALCULATION

SPHERICAL REFLECTOR EXAMPLE USING A CORRUGATED HORN FEED, 1 GHZ.
HORN RADIUS=3.333 WAVELENGTHS, HALF_FLARE ANGLE=3.43626 DEGREES.
WAVELENGTH OF ELECTRIC FIELD.................................................. 3.000
LOCATION OF COORDINATE ORIGIN WRT FEED (X,Y,Z).......... 233.25 0.00 0.00
FEED ROTATION ANGLES (ALPHA, BETA, GAMMA)......................... 0.00 0.00 0.00
RADIUS OF THE REFLECTOR SPHERE............................................. 575.00
APERTURE PLANE LOCATION (XC).................................................. -555.00
COORDINATES OF THE APERTURE PLANE CENTER.......................... 0.00 0.00
HALF MAJOR AXIS OF APERTURE PLANE (ALONG Y)................. 100.00
HALF MINOR AXIS OF APERTURE PLANE (ALONG Z)................. 100.00
GRID SIZE USED FOR NUMERICAL INTEGRATION....................... 9.2600
FEED SHADOW CENTER COORDINATES IN APERTURE PLANE........... 0.00 0.00
HALF MAJOR AXIS OF FEED SHADOW............................................. 1.00
HALF MINOR AXIS OF FEED SHADOW............................................. 1.00
THETA RANGE FOR FEED PATTERN (L,H,I - DEGREES)............. 67.00 113.00 1.00
PHI RANGE FOR FEED PATTERN (L,H,I - DEGREES).................. 157.00 203.00 1.00

------------------------ FINISHED INPUT ------------------------
NUMBER OF INTERNAL POINTS........................................... 1237
NUMBER OF EDGE POINTS ................................................ 78
THE FEED MAX IS...................................................... THETA = 90.00
.......................................................... PHI = 180.00
THE REFLECTED RAY VECTOR (X,Y,Z)................................. 1.00 0.00 0.00
THE REFLECTED BEAM MAX IS........................................ THETA = 90.00
.......................................................... PHI = 0.00

------------------------ FINISHED APERTUR ------------------------
LOWER CURVE POINTS WITH MAX AND/OR MIN POINTS............ 41
UPPER CURVE POINTS WITH MAX AND/OR MIN POINTS............ 41
GRID RANGES FROM ...................................................... -99.9400 TO 99.9400
TOTAL NO. OF GRID BARS ............................................... 39
 THEREFORE NUMBER OF QUANTIZED EDGE POINTS .............. 78
 THEREFORE INTERNAL + QUANTIZED EDGE POINTS ............ 1315
NUMBER OF QUANTIZED POINTS IN THE APERTURE PLANE .... 1312

------------------------ FINISHED QUANTIZ ------------------------
<table>
<thead>
<tr>
<th>PHI</th>
<th>0.000</th>
<th>THETA</th>
<th>90.000</th>
<th>90.025</th>
<th>90.825</th>
<th>0.025</th>
</tr>
</thead>
<tbody>
<tr>
<td>THETA</td>
<td>DBZ/Z</td>
<td>DBY/Z</td>
<td>DBZ/Y</td>
<td>DBY/Y</td>
<td>PWRDB</td>
<td></td>
</tr>
<tr>
<td>90.000</td>
<td>0.0000</td>
<td>-119.46398</td>
<td>119.46398</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
<tr>
<td>90.025</td>
<td>-32.62658</td>
<td>-119.53315</td>
<td>119.53315</td>
<td>-0.4406</td>
<td>-330.05</td>
<td></td>
</tr>
<tr>
<td>90.050</td>
<td>-3.0119</td>
<td>-119.59931</td>
<td>119.59931</td>
<td>-1.3533</td>
<td>-3.0011</td>
<td></td>
</tr>
<tr>
<td>90.075</td>
<td>-9.42614</td>
<td>-119.64654</td>
<td>119.64654</td>
<td>-1.8255</td>
<td>-5.3913</td>
<td></td>
</tr>
<tr>
<td>90.100</td>
<td>-8.57678</td>
<td>-119.69888</td>
<td>119.69888</td>
<td>-2.3056</td>
<td>-8.5760</td>
<td></td>
</tr>
<tr>
<td>90.125</td>
<td>-12.70717</td>
<td>-119.7418</td>
<td>119.7418</td>
<td>-2.8020</td>
<td>-12.7071</td>
<td></td>
</tr>
<tr>
<td>90.150</td>
<td>-18.03187</td>
<td>-119.79461</td>
<td>119.79461</td>
<td>-3.3063</td>
<td>-18.0310</td>
<td></td>
</tr>
<tr>
<td>90.200</td>
<td>-32.07010</td>
<td>-119.89870</td>
<td>119.89870</td>
<td>-4.3472</td>
<td>-32.0701</td>
<td></td>
</tr>
<tr>
<td>90.250</td>
<td>-46.68027</td>
<td>-120.00713</td>
<td>120.00713</td>
<td>-5.4315</td>
<td>-33.6007</td>
<td></td>
</tr>
<tr>
<td>90.275</td>
<td>-55.76724</td>
<td>-120.06298</td>
<td>120.06298</td>
<td>-6.0026</td>
<td>-55.7624</td>
<td></td>
</tr>
<tr>
<td>90.300</td>
<td>-66.71076</td>
<td>-120.11993</td>
<td>120.11993</td>
<td>-6.5575</td>
<td>-66.7107</td>
<td></td>
</tr>
<tr>
<td>90.325</td>
<td>-77.02057</td>
<td>-120.17798</td>
<td>120.17798</td>
<td>-7.1400</td>
<td>-77.0205</td>
<td></td>
</tr>
<tr>
<td>90.350</td>
<td>-89.30536</td>
<td>-120.23714</td>
<td>120.23714</td>
<td>-7.7316</td>
<td>-89.3053</td>
<td></td>
</tr>
<tr>
<td>90.375</td>
<td>-103.0186</td>
<td>-120.29741</td>
<td>120.29741</td>
<td>-8.3343</td>
<td>-103.0186</td>
<td></td>
</tr>
<tr>
<td>90.400</td>
<td>-118.9269</td>
<td>-120.35879</td>
<td>120.35879</td>
<td>-8.9491</td>
<td>-118.9269</td>
<td></td>
</tr>
<tr>
<td>90.425</td>
<td>-137.1385</td>
<td>-120.41230</td>
<td>120.41230</td>
<td>-9.5731</td>
<td>-137.1385</td>
<td></td>
</tr>
<tr>
<td>90.450</td>
<td>-156.34906</td>
<td>-120.46893</td>
<td>120.46893</td>
<td>-10.2094</td>
<td>-156.3490</td>
<td></td>
</tr>
<tr>
<td>90.475</td>
<td>-177.55976</td>
<td>-120.54969</td>
<td>120.54969</td>
<td>-10.8570</td>
<td>-177.5597</td>
<td></td>
</tr>
<tr>
<td>90.500</td>
<td>-199.76711</td>
<td>-120.62155</td>
<td>120.62155</td>
<td>-11.5160</td>
<td>-199.7671</td>
<td></td>
</tr>
<tr>
<td>90.525</td>
<td>-222.97415</td>
<td>-120.70826</td>
<td>120.70826</td>
<td>-12.1864</td>
<td>-222.9741</td>
<td></td>
</tr>
<tr>
<td>90.550</td>
<td>-247.18042</td>
<td>-120.79500</td>
<td>120.79500</td>
<td>-12.8662</td>
<td>-247.1804</td>
<td></td>
</tr>
<tr>
<td>90.575</td>
<td>-272.38524</td>
<td>-120.88201</td>
<td>120.88201</td>
<td>-13.5615</td>
<td>-272.3852</td>
<td></td>
</tr>
<tr>
<td>90.600</td>
<td>-299.58868</td>
<td>-120.97030</td>
<td>120.97030</td>
<td>-14.2664</td>
<td>-299.5887</td>
<td></td>
</tr>
<tr>
<td>90.625</td>
<td>-329.79151</td>
<td>-121.05962</td>
<td>121.05962</td>
<td>-15.0000</td>
<td>-329.7915</td>
<td></td>
</tr>
<tr>
<td>90.650</td>
<td>-362.00539</td>
<td>-121.14927</td>
<td>121.14927</td>
<td>-15.7711</td>
<td>-362.0054</td>
<td></td>
</tr>
<tr>
<td>90.675</td>
<td>-414.96398</td>
<td>-121.23825</td>
<td>121.23825</td>
<td>-16.5410</td>
<td>-414.9639</td>
<td></td>
</tr>
<tr>
<td>90.700</td>
<td>-494.25915</td>
<td>-121.32606</td>
<td>121.32606</td>
<td>-17.3027</td>
<td>-494.2592</td>
<td></td>
</tr>
<tr>
<td>90.725</td>
<td>-6.35135</td>
<td>-121.41488</td>
<td>121.41488</td>
<td>-18.0746</td>
<td>-6.35136</td>
<td></td>
</tr>
<tr>
<td>90.750</td>
<td>-6.70424</td>
<td>-121.50329</td>
<td>121.50329</td>
<td>-18.8465</td>
<td>-6.70424</td>
<td></td>
</tr>
<tr>
<td>90.775</td>
<td>-8.29341</td>
<td>-121.59188</td>
<td>121.59188</td>
<td>-19.6283</td>
<td>-8.29341</td>
<td></td>
</tr>
<tr>
<td>90.800</td>
<td>-10.71295</td>
<td>-121.68061</td>
<td>121.68061</td>
<td>-20.4125</td>
<td>-10.7129</td>
<td></td>
</tr>
</tbody>
</table>

20LOG(MAX(FIELD-Z)) = 22.19400
20LOG(MAX(FIELD-Y)) = 23606766-04

INTERPOLATION NUMBER USED FOR INTEGRATION IS...........
6. SUBROUTINE APERTUR

The purpose of subroutine APERTUR is to calculate the points of intersection of the reflected rays with the aperture plane and the field values at each point.

To be able to express the components of each ray from the feed in the reflector coordinate system i.e., (x,y,z) coordinate system, the rotation matrix A is calculated. Then for each ray i.e., for each $\theta'$ and $\phi'$, using the equation of the reflector, the following are calculated:

- $R$ Distance along the ray between the feed and the reflector,
- $(X_0,Y_0,Z_0)$ Point on the reflector where the incident ray strikes,
- NHAT A one-dimensional array containing three components of the unit normal vector at $(X_0,Y_0,Z_0)$,
- SR A one-dimensional array containing three components of a vector along the reflected ray,
- ER A one-dimensional array containing three components of $E$ along the reflected ray,
- $(Y,Z)$ Point of intersection of the reflected ray with the aperture plane,
- $D$ Distance along the reflected ray between the reflector and the aperture plane, and
- PHASE $(R+D)2\pi/\lambda + \text{the initial phase of the ray}$

$Y,Z,ER(2), ER(3)$, and PHASE are then temporarily stored in array PNEW and $Y,Z 0.0, 0.0, 0.0$ in array PBLK. Next the values of $Y$ and $Z$ are tested to determine whether the point of intersection is outside, on, or inside the aperture plane.
ellipse and whether it is inside the shadow region of the feed on the aperture plane. Based on the results of these tests Y, Z, ER(2), ER(3), and PHASE are stored in the P array with the internal points at the top and the edge points at the bottom. The same information but with a different order is also written on tape 20. The flow chart in Figure 16 shows the logic of this sorting process. Figure 17 is a simplified pictorial representation of the P array which shows the order in which the data are stored.
If blocked set $E = 0/0$ and label it 'Blocked'.

1. If $\text{APRDTA} = 0$, write $\text{PNEW}$ or $\text{FBLK}$ or $\text{PINT}$ on tape 20 and put appropriate label.
2. Store $\text{PNEW}$ or $\text{FBLK}$ starting at the second position from bottom of $P$ array.

If blocked set $E = 0/0$ and label it 'Blocked'.

1. If $\text{APRDTA} = 0$, write $\text{PNEW}$ or $\text{FBLK}$ or $\text{PINT}$ on tape 20 and put appropriate label.
2. Store $\text{PNEW}$ or $\text{FBLK}$ or $\text{PINT}$ starting at the second position from bottom of $P$ array.

Previous & present points w.r.t. aperture pl. ellipse

In & Out

Interpolate a point on the edge and label it 'Interpolated'.

Previous & present points opposite sides of $y_e$

Yes

Is the present $(0,0)$ closer to Beam Max than the last

Yes

No

Go to Next point

Figure 16 -- Partial Flow Chart of Subroutine APERTUR
Figure 17 -- P Array Showing Locations of Internal and Edge Points
SUBROUTINE APERTURE(P, NTX, NPX)
COMMON/PARAMS/TITLE(16), AORDR, XLAM, GRID, SURFACE, APROTA, FEED(3),
   ALPH, BETAR, GAMMA, XC, YC, ZC, HFMAEX, HMIEX, BMTP, BMPP,
   NT, NP, NPOIN, MAXPTS, BELLP
COMMON/STOCK/YCBL, ZCBL, HFMBL, HFMIBL
COMMON/NATH/P1, P2, PIN2, DTPR, RTOD
COMMON/POINTL/NEDGE, NINTR
REAL NHAT
INTEGER SEDE
DIMENSION P(5,NTX,NPX), POLD(5), PNEW(5), PINT(5), PBLK(5), A(3,3),
   B(3,2), BZ(3,2), NHAT(3), C(3), SR(3), EI(3), ER(3)

ALPH=ALPHA*DTOR
BETAR=BETA*DTOR
GAMMAR=GAMMA*DTOR
A(1,1)=COS(ALPHAR)*COS(GAMMAR)-SIN(ALPHAR)*SIN(BETAR)*SIN(GAMMAR)
A(1,2)=SIN(ALPHAR)*COS(GAMMAR)+COS(ALPHAR)*SIN(BETAR)*SIN(GAMMAR)
A(1,3)=-COS(BETAR)*SIN(GAMMAR)
A(2,1)=-SIN(ALPHAR)*COS(BETAR)
A(2,2)=COS(ALPHAR)*COS(BETAR)
A(2,3)=SIN(BETAR)
A(3,1)=COS(ALPHAR)*SIN(GAMMAR)+SIN(ALPHAR)*SIN(BETAR)*COS(GAMMAR)
A(3,2)=SIN(ALPHAR)*SIN(GAMMAR)-COS(ALPHAR)*SIN(BETAR)*COS(GAMMAR)
A(3,3)=COS(BETAR)*COS(GAMMAR)

IF(APROTA.GT.0.0) WRITE(20,110)

110 FORMAT(1X,1OH************7X,*THETA*,9X,*Y*,9X,*Z*,7X,*FRY*,7X,
   *ERZ*,5X,*PHASE*,9X,*R*)

BMTEST=1.0E+40
NINTR=NEDGE=0
PBLK(5)=PBLK(4)=PBLK(5)=0.0
DO 5000 IP=1,NP
   DEGPHI=P(5,1,IP)*RTOD
   IF(APROTA.GT.0.0) WRITE(20,120) DEGPHI

5000 CONTINUE

120 FORMAT(1X,*PHI=**,F10.4)
   DO 4000 IT=1,NT
      DEGTHET=P(4,IT,IP)*RTOD
      SINF=SIN(P(5,IT,IP))
      COSF=COS(P(5,IT,IP))
      SINT=SIN(P(4,IT,IP))
      COST=COS(P(4,IT,IP))
`RR(1,1)=SINT*COSP  
RR(2,1)=SINT*SINP  
RR(3,1)=COSP  
RR(1,2)=+FEED(1)  
RR(2,2)=+FEED(2)  
RR(3,2)=+FEED(3)  
CALL MULT32(R,A,RR)  
IF (SURFACE) 122, 124, 126  
122 AR=R(1,1)**2/AORDRF**2+B(2,1)**2+B(3,1)**2)/BELLP**2  
BR=-2.0*(B(1,1)*B(1,2)/AORDRF**2+(B(2,1)*B(2,2)+B(3,1)*B(3,2))/BELLP**2)  
CR=B(1,2)**2/AORDRF**2+B(3,2)**2/BELLP**2-1.0  
GO TO 128  
124 AR=R(1,1)*B(1,1)+B(2,1)*B(2,1)+B(3,1)*B(3,1)  
BR=-2.0*(R(1,1)*R(1,2)+R(2,1)*R(2,2)+R(3,1)*R(3,2))  
CR=R(1,2)*B(1,2)+B(2,2)+B(3,2)+B(1,1)-AORDRF*ANORF  
GO TO 128  
126 AR=R(2,1)*B(2,1)+B(3,1)*B(3,1)  
BR=-2.0*(R(2,1)*R(2,2)+R(3,1)*R(3,2)+2.0*AORDRF*B(1,1))  
CR=B(2,2)*R(2,2)+B(3,2)+B(3,2)+4.0*AORDRF*B(1,2)-4.0*AORDRF**2  
128 IF (AR.LT.1.0E-10) R=CR/AR  
IF (AR.LT.1.0E-10) GO TO 130  
R=(-HR+SORT(BR-BR-4.0*AR*CR))/(AR+AR)  
130 CONTINUE  
X0=R(1,1)*R=R(1,2)  
YO=R(2,1)*R=R(2,2)  
ZO=R(3,1)*R=R(3,2)  
IF (SURFACE) 132, 134, 136  
132 NHAT(1)=-X0*BELLP**2/SQRT(X0**2*BELLP**4+(Y0**2+Z0**2)*AORDRF**4)  
NHAT(2)=-Y0*AORDRF**2/SQRT(X0**2*BELLP**4+(Y0**2+Z0**2)*AORDRF**4)  
NHAT(3)=-Z0*AORDRF**2/SQRT(X0**2*BELLP**4+(Y0**2+Z0**2)*AORDRF**4)  
GO TO 138  
134 NHAT(1)=-X0/AORDRF  
NHAT(2)=-Y0/AORDRF  
NHAT(3)=-Z0/AORDRF  
GO TO 138  
136 NHAT(1)=-2.0*AORDRF/SQRT(4.0*AORDRF**2+Y0**2+Z0**2)  
NHAT(2)=-Y0/SQRT(4.0*AORDRF**2+Y0**2+Z0**2)  
NHAT(3)=-Z0/SQRT(4.0*AORDRF**2+Y0**2+Z0**2)  
`
SCALAR=2.0*(RI(1,1)*NHA(T(1)+RI(2,1)*NHA(T(2)+RI(3,1)*NHA(T(3))
DO 1500 I=1,3
1500 SR(I)=RI(1,1)-SCALAR*NHA(T(I)
EI=PI(1,IT,IP)/R
EPI=PI(2,IT,IP)/R
C(1)=COST*COSP*EI-SINP*EPI
C(2)=COST*SINP*EI+COSP*EPI
C(3)=SINT*EI
DO 2000 I=1,3
EI(I)=0.0
DO 2000 J=1,3
2000 EI(I)=EI(I)+AI(I,J)*C(J)
SCALAR=2.0*(EI(I)*NHA(T(1)+EI(2)*NHA(T(2)+EI(3)*NHA(T(3))
DO 2500 I=1,3
2500 ER(I)=SCALAR*NHA(T(I)-EI(I)
Y=YO+(XC-XO)*SR(2)/SR(I)
Z=ZO+(XC-XO)*SR(3)/SR(I)
D=SRT((XC-XO)*(XC-XO)+(Y-YO)*(Y-YO)+(Z-ZO)*(Z-ZO))
PHASE=PI(2*(R+D))/XLMAP+P(3,IT,IP
PN=PN(1)=PNK(1)=Y
PNF(2)=PNK(2)=Z
PN(3)=FR(2)
PN(4)=FR(3)
PN(5)=PHASE
TEST=HFMAEX*HFMAEX*HFMI**HFMI**HFMAE*HFMAE*(Z-ZC)*(Z-ZC)
-HFMAE*HFMI**HFMI**HFMAE*(Y-YC)*(Y-YC)
-TESTRL=HFMAE*HFMAE*HFMI**HFMI**HFMAE*(Z-ZCRL)*(Z-ZCRL)
-HFMAE*HFMAE*(Y-YC)*HMI*(Y-YCRL)
* IF (TEST) 2701,2501,2601
2501 NEDGE=NEDGE+1
SEDGE=MAXPTS-EDGE
IF (TESTBL,LT,0.0) GO TO 2510
CALL MOVEM(PBLK,P(1,SEDGE),5)
IF(APRDTA,GT,0.0) WRITE(20,2505) PBLK
2505 FORMAT(1X,**,23X,2F10.4,2F10.7,F10.4,12X,*EDGE POINT, BLOCKED*)
GO TO 2515
2510 CALL MOVEM(PNEW,P(1,SEDGE),5)
IF(APRDTA,GT,0.0) WRITE(20,2512) PNEW
2512 FORMAT(1X,**,23X,2F10.4,2F10.7,F10.4,12X,*EDGE POINT*)

2515 CONTINUE
GO TO 2800

2601 NINTR=NINTR+1
IF (TESTBL.LT.0.0) GO TO 2610
CALL MOVEM(PBLK,P(1,NINTR),5)
IF(APRDTA.GT.0.0) WRITE(20,2605) DEGTHET,PBLK,R
2605 FORMAT(*$*,13X,3F10.4,2F10.7,2F10.4,14X,*BLOCKED*)
GO TO 2615

2610 CALL MOVEM(PNEW,P(1,NINTR),5)
IF(APRDTA.GT.0.0) WRITE(20,2612) DEGTHET,PNEW,R
2612 FORMAT(*$*,13X,3F10.4,2F10.7,2F10.4)
2615 CONTINUE

2701 IF (IT.EQ.1) GO TO 2800
IF (TEST*TESTO) 2704,2800,2800

2704 ZTEST1=ZC-POLD(2)
ZTEST2=ZC-PNEW(2)
IF(ZTEST1*ZTEST2.LT.0.0) GO TO 2800
CALL INTERP(POLD,PNEW,PINT)
NEDGE=NEDGE+1
SEDGE=MAXPTS-NEDGE
CALL MOVEM(PINT,PBLK,2)
Y=PINT(1)
Z=PINT(2)
TFSTBL=HFMABL*HFMARL*HFMIBL*HFMIBL-HFMABL*HFMARL*(Z-ZCRL)*(Z-ZCRL)
     -HFMIBL*HFMIBL*(Y-YCRL)*(Y-YCRL)

2710 IF (TESTBL.LT.0.0) GO TO 2710
CALL MOVEM(PBLK,P(1,SEdge),5)
IF(APRDTA.GT.0.0) WRITE(20,2705) PBLK
2705 FORMAT(*$*,23X,2F10.4,2F10.7,F10.4,10X,*INTERPOLATED,BLOCKED*)
GO TO 2715

2710 CALL MOVEM(PINT,P(1,SEdge),5)
IF(APRDTA.GT.0.0) WRITE(20,2712) PINT
2712 FORMAT(*$*,23X,2F10.4,2F10.7,F10.4,10X,*INTERPOLATED*)
2715 CONTINUE

2300 CALL MOVEM(PNEW,POLD,5)
TESTO=TEST
TEST=(DEGTHET-90.0)**2+(DFGPHI-180.0)**2
IF (TEST-AMTEST) 2980,3000,3000
2980 AMTEST=TEST
BMT=DEGTETH
BMP=DEGPHI
BMSRX=SR(1)
BMSRY=SR(2)
BMSRZ=SR(3)
3000 CONTINUE
4000 CONTINUE
5000 CONTINUE
PRINT 5025, NINTR, NEDGE
5025 FORMAT(* NUMBER OF INTERNAL POINTS........................................*I5/
* NUMBER OF EDGE POINTS ..................................................*I5)
COSAMTP=BMSRZ/SORT(BMSRX**2+BMSRY**2+BMSRZ**2)
SINHMP=BSMRYSORT(BMSRX**2+BMSRY**2)
BMP=RTOD*ACOS(COSAMTP)
BMPP=RTOD*ASIN(SINHMP)
PRINT 5050, BHT, BMP, BMSRX, BMSRY, BMSRZ, BMTP, BMPP
5050 FORMAT(* THE FEED MAX IS..................................................THETA=*F7.2/
* PHII=F7.2/
* THE REFLECTED RAY VECTOR (X,Y,Z)......................................*F7.2
* THE REFLECTED BEAM MAX IS.............................................THETA=*F7.2/
* PHII=F7.2)
RETURN
END
7. SUBROUTINE QUANTIZE

The edge points and the internal points are quantized in this subroutine along the equispaced y=constant lines. First, the internal points are temporarily stored on tape 8 and the edge points are quantized and then later, the internal points are quantized.

EDGE POINTS: A call to subroutine FQKSORT arranges all the edge points as obtained from APERTUR in ascending order first with respect to the y-coordinate and then with respect to the z-coordinate and stores them in the P array starting at location NEDGE1 and ending at location NEDGE2 (Figure 17). If there is not an edge point at A in Figure 18, a point is added at A and stored at the beginning of the sequence and INLO is set to 1.

Figure 18 -- Edge Points
Similarly, if there is no edge point at B, an edge point is created at B and stored at the end of the existing sequence where storage locations were previously left vacant and NHI is set to 1. NEDGE1, NEDGE2, and NEDGE are appropriately modified so that NEDGE1 is still the location of the first edge point and NEDGE2 is the location of the last edge point. Next, NLO (The number of edge points on and below $Y_e$) and NHI (the number of edge points on and above $Y_e$) are determined.

Now considering P as an one-dimensional array, the data associated with the NLO edge points are stored in the first $5 \times NLO$ locations of the P array, and the data associated with the NHI edge points are stored in the next $5 \times NHI$ locations of the P array (DO loop ending in statement number 7500). Internal points stored in these locations have already been saved on tape 8. However, one must check to be sure that $(NLO+NHI)<NEDGE1$ so that the information is not written over the memory space still occupied by the edge point data. For ease in understanding the rearrangement of the edge point data in one-dimensional form, let the first $5 \times (NLO+NHI)$ locations of the P array be laid out as shown in Figure 19. The data are assigned to these locations as shown. The locations enclosed by heavy boundaries could be vacant at this point depending upon the values of INLO and INHI. These are assigned the same ER$_y$, ER$_z$, and Phase values as the point next to them (DO loop ending in statement number 7505). Dictated by the aperture plane ellipse size and the quantizing interval, the
Figure 19 -- P Array Laid Out in One Dimensional Form
total number of equispaced grid lines which fall within the aperture plane ellipse (NBARS) is calculated. The number of quantized edge points (NQEDGE) and the total number of quantized edge points (NPOINTS) are also calculated.

Inside the DO loop ending in statement number 8500, Z, ER_y, ER_z, and Phase are interpolated at the end points of each equispaced grid line by performing a spline fit through the edge points along the boundary of the aperture plane ellipse. L1 and L2 are the starting points of two temporary one-dimensional work arrays which are contained in the P array. LSTART is a number such that P(1, LSTART), beginning where the interpolated values at the end points of the grid lines are stored, is beyond the above mentioned temporary work arrays. These 5\times2\times NBARS quantized values are then moved to the beginning of the P array. Next, the information about the internal points which was temporarily stored on tape 8 is transferred to the P array starting at the 5\times2\times NBARS+1 location.

INTERNAL + EDGE POINTS: In the DO loop ending in statement number 8600, the y-coordinates of all the aperture plane points are aligned along the grid lines. All the points are then sorted and rearranged in ascending order with respect to the y-coordinate and then with respect to the z-coordinate. If the number of points along any grid line is less than three, that grid line is deleted.

Finally, if APRDTA>0, the quantized data are written on tape 20 to be used later for aperture plots by subroutine APRPLT.
SUBROUTINE QUANTIZ (P,NTXNPX)
COMMON/PARAMS/TITLE(16),AORORF,XYAM,GRID,SURFACE,APRDTA,FEED(3),
                        ALPHA,BETA,GAMMA,XC,YC,ZC,HFMGLX,HFMIEX,BMTP,BMP2,
                        NT,NN,NPPOINT,MAPP,MAPL
COMMON/POINTS/NEDGE,NINTR
DIMENSION P(5,NTXNPX)
DATA SIGMA/1.0/
REWIND 8
WRITE(8) ((P(I,J),I=1,5),J=1,NINTR)
NEDGE2=MAXPTS-1
NEDGE1=MAXPTS-NEDGE
CALL FQKSORT(P1,NEDGE1,5,NEDGE)
INLO=INHI=0
YMIN=YC-HFMGLX
YMAX=YC+HFMIEX
IF(P(1,NEDGE1).LT.YMIN) GO TO 7000
NEDGE1=NEDGE1+1
NEDGE=NEDGE+1
P(1,NEDGE1)=YMIN
P(2,NEDGE1)=ZC
INLO=1
7000 CONTINUE
IF(P(1,NEDGE2).GT.YMAX) GO TO 7100
NEDGE2=NEDGE2+1
NEDGE=NEDGE+1
P(1,NEDGE2)=YMAX
P(2,NEDGE2)=ZC
INHI=1
7100 CONTINUE
INHI=INLO=0
DO 7400 I=NEDGE1,NEDGE2
IF (P(I,J).LT.ZC) 7350,7360,7370
7350 NLO=NLO+1
GO TO 7400
7360 NLO=NLO+1
7370 NHI=INHI+1
7400 CONTINUE
PRINT 7410, NLO,NHI
7410 FORMAT(* LOWER CURVE POINTS WITH MAX AND/OR MIN POINTS............*5/
* UPPER CURVE POINTS WITH MAX AND/OR MIN POINTS

IF (NEDGE1-NL0-NHI) 7414, 7420, 7420
7414 PRINT 7415
7415 FORMAT(* ---- PROBLEM FOUND IN REARRANGING EDGE POINTS IN ONE DIM
*ENTIONAL ARRAY FORM ---- *)
STOP 7415
7420 CONTINUE
IL0=0
IHI=5*NLO
DO 7450 I=1, NEDGE1, NEDGE2
IF (P(2,I)-ZC) 7450, 7460, 7470
7450 IL0=IL0+1
DO 7451 J=1, 5
P(IL0+(J-1)*NLO)=P(J,I)
7451 CONTINUE
GO TO 7500
7460 IL0=IL0+1
DO 7461 J=1, 5
P(IL0+(J-1)*NL0(I))=P(J,I)
7461 CONTINUE
IHI=IHI+1
DO 7471 J=1, 5
P(IHI+(J-1)*NHI)=P(J,I)
7471 CONTINUE
7500 CONTINUE
DO 7505 J=2, 4
IF (NL0.EQ.0) GO TO 7501
P(J+NL0+1)=P(J+NL0+2)
P((5*NL0+J*NHI+1))=P((5*NL0+J*NHI+2))
7501 IF (NHI.EQ.0) GO TO 7505
P((J+1)*NLO(I))=P((J+1)*NLO-1)
P((5*NL0+(J+1)*NHI))=P((5*NL0+(J+1)*NHI-1))
7505 CONTINUE
J=EXMAEX/GRID
BARMIN=YC-J*GRID
BARMAX=YC+J*GRID
NHARS=2*J+1
NOEDGE=2*NHARS
NP0INT=N1TR+NOEDGE
7520  FORMAT(* GRID RANGES FROM .....................................*F10.4* TO*F
   .10.4*/
   * TOTAL NO. OF GRID BARS........................................*I5/
   * THEREFORE NUMBER OF QUANTIZED EDGE POINTS................*I5/
   * THEREFORE INTERNAL + QUANTIZED EDGE POINTS................*I5")
IF (NP0INT.LE.MAXPTS) GO TO 8000
PRINT 7550
7550  FORMAT(* SUFFICIENT CORE ----- *)
STOP 7550
8000  CONTINUE
MAXSZ=MAXO(NLO,NHI)
L1=5*(NLO+NHI)+1 $ L2=L1+MAXSZ+1 $  LSTART=(L2+MAXSZ)/5+1
L=LSTART $ NPTS=NLO $  LI=1
DO 8500 II=1,2
   LD=LI+NPTS
   DO 8400 I=2,5
      SIGMA=-ARS(SIGMA)
      CALL CURV1( NPTS,P(LI),P(LD),SLP1,SLPN,P(L1),P(L2),SIGMA)
      BAR=BARMIN $  K=L
      DO 8300 J=1,NRARS
         P(I,K)=CURV2(BAR,NPTS,P(LI),P(LD),P(L1),SIGMA,J)
      P(I,K)=BAR
      BAR=BAR+GRID $  K=K+1
   ALL LINS
8300  CONTINUE
LD=LD+NPTS
8400  CONTINUE
L=L+NHARS $ NPTS=NHI $ LI=LI+5*NLO
8500  CONTINUE
CALL MOVEM(P1,LSTART),P,5*NOEDGE)
REWIND A
READ(3)((P(I,NOEDGE+J),I=1,5),J=1,NITR)
ENDFILE A
DO 3500 I=1,NP0INT
   YQ=AIN(AS(P1(I)),GRID+0.5)*GRID
   IF(P1(I,LT.0) YQ=-YQ
   P1(I,I)=YQ
8600  CONTINUE
CALL FOKSORT(P,5,NPOINT)
N=L=0
YQ=P(1,1)
DO 9500 I=1,NPOINT
IF(P(1,I) .EQ. YQ) GO TO 9000
IF(L.LE.2) N=N-L
L=0
YQ=P(1,I)
9000  L=L+1
N=N+1
CALL MOVEM(P(1,I),P(1,N),5)
9500  CONTINUE
IF(L.LE.2) N=N-L
NPOINT=N
PRINT 9550, NPOINT
9550  FORMAT(= NUMBER OF QUANTIZED POINTS IN THE APERTURE PLANE.....*15)   TAPE 20
IF(APRDTA.GT.0.0) WRITE(20,9560)
IF(APRDTA.GT.0.0) WRITE(20,9570) ((P(I,J),I=1,5),J=1,NPOINT)   TAPE 20
9560  FORMAT(= QUANTIZED POINTS*)
9570  FORMAT(IX,*$*,23X,2F10,4,2F10,7,F10,4)
RETURN
END
8. SUBROUTINE RAdPAT

By performing a double integration over the quantized aperture plane points, this subroutine calculates the electric field in any far field direction ($\theta, \phi$). The double integration is done by first evaluating the line integrals along the $y=$ constant grid lines and then performing an integration over these line integral values along the transverse direction ($Y$ in Figure 18). On each of the lines along which the above line integrals are performed, the electric field is known only at a finite number of discrete points. To more accurately evaluate the line integrals, the distance between every two quantized points on a line is subdivided in several parts and the electric field is linearly interpolated at these intermediate points. In the subroutine, this number of subdivisions has been called NPARTS. Also, to identify the start and the end of the $y=$ constant lines, the $y$-coordinates of the first and the last quantized points are replaced by an identifying variable called SEN. Between statements number 115 and 160, the pattern request information is sorted out. As explained in section 4, a pattern is requested by first choosing a plane defined by $\theta=$constant or $\phi=$constant (MAJOR=THETA or PHI), the value of the angle defining the plane being called AMAJOR. Then in the plane selected above, the range and the increment size of the other variable angle ($\phi$ or $\theta$, respectively) is specified (MINOR=PHI or THETA, respectively), AMINOR(1), AMINOR(2), and AMINOR(3) being the initial, the final, and the incremental values of the variable angle in degrees.
Computation of the far field takes place between statements numbered 3450 and 5000. The integrals to be evaluated are

\[ E_z = \int \int_{\text{aperture}} \tilde{E}_z \cos \phi \cdot e^{j[ky\sin\theta + kz\cos\theta - \phi]} \, dy \, dz \]

and

\[ E_y = \int \int_{\text{aperture}} (\tilde{E}_y \sin \theta + \tilde{E}_z \cos \theta \sin \phi) \cdot e^{j[ky\sin\theta + kz\cos\theta - \phi]} \, dy \, dz \]

where \( \tilde{E}_y \), \( \tilde{E}_z \), and \( \phi \) are the y- and z-components and the Phase of the aperture electric field at a point \((y, z)\) in the aperture plane. As pointed out earlier, the above integrals are evaluated in two steps - first by performing integration in the z-direction along the \( y=\text{constant} \) lines and computing \( ZI \) and \( YI \) which are functions of \( y \),

\[ ZI = \int_{y=\text{const. line}} \tilde{E}_z \cos \phi \cdot e^{j[kz\cos\theta - \phi]} \, dz \]

and

\[ YI = \int_{y=\text{const. line}} (\tilde{E}_y \sin \theta + \tilde{E}_z \cos \theta \sin \phi) e^{j[kz\cos\theta - \phi]} \, dz \]

and then computing \( E_z \) and \( E_y \) (called FLDZ and FLDY in the subroutine) as

\[ E_z = \text{FLDZ} = \int ZI \cdot e^{jk\sin\theta \sin \phi} \, dy \]
and

\[ E_y = \text{FLDY} = \int YI e^{jkysin\theta \sin\phi} dy. \]

A flow chart explaining the steps in the above computations is shown in Figure 20.

After statement number 5000, the computed values of FLDE and FLDY are normalized and printed out in decibels. To avoid the underflow problem in computing decibels, whenever the field strength is less than \(10^{-10}\), -60 dB is used.

Finally, subroutine PLOT is called to generate a line printer plot of the normalized \(E_z\), \(E_y\), and the total power patterns.
Figure 20 -- Partial Flow Chart of Subroutine RADPAT
SUBROUTINE RAPAT (P, NTPX)
COMMON/PARMS/TITLE(16), AORORF, XLAM, GRID, SURFACE, APRDTA, FEED(3),
  ALPHA, BETA, GAMMA, XC, YC, ZC, HFMAEX, HFMIEX, BMTP, BMPP,
  NT, NPOINT, MAXPTS, BELLP
COMMON/MATH/PI, PI2, PI02, DOR, RTOD
LOGICAL LOOPI
DIMENSION P(5, NTPX), FIELDZ(300), FELDY(300), PWER(300), AMOEOR(3)
COMPLEX CTENP, CZ1, CZ2, CY1, CY2, TSZ, TSX, DZI, DZI, ZIOLD, YIOLD, ZI, YI,
  FLD2, FLDY
SEH=999.0
NPTAS=7
RPTAS=1./NPTAS
XLM=PI2/XLM
CALL SETM(SEH, P(1, NPOINT+1), 5)
100 CONTINUE
READ 110, MAJOR, AMAJOR, MINOR, AMOINR
110 FORMAT (A5, F10.0, A5, 3F10.0)
IF (EOF(7)) 9000, 115
115 CONTINUE
IF (MAJOR.EQ.5HMAX-P) AMAJOR=AMAJOR+BMPP
IF (MAJOR.EQ.5HMAX-P) MAJOR=5HPHI
IF (MAJOR.EQ.5HMAX-T) AMAJOR=AMAJOR+BMTP
IF (MAJOR.EQ.5HMAX-T) MAJOR=5HTHETA
IF (MINOR.EQ.5HMAX-P) AMINOR(1)=AMINOR(1)+BMPP
IF (MINOR.EQ.5HMAX-P) MINOR=5HPHI
IF (MINOR.EQ.5HMAX-T) AMINOR(1)=AMINOR(1)+BMTP
IF (MINOR.EQ.5HMAX-T) AMINOR(2)=AMINOR(2)+BMPP
IF (MINOR.EQ.5HMAX-T) MINOR=5HTHETA
160 CONTINUE
DEG=AMAJOR
DEG=DEG*DOR
DLOR=AMINOR(1)*DOR
DHR=AMINOR(2)*DOR
DICR=AMINOR(3)*DOR
FMAX2=FMAXY=-1.E+40
NTH=0
LOOPI=.TRUE.
D=DLOR
IF(MAJOR .NE. 3*PHI) LOOPHI=.FALSE.
IF (LOOPHI) 400,3400

400  
COSP=COS(DEGR)
SINP=SIN(DEGR)
COST=COS(D)
SINT=SIN(D)
GO TO 3425

3400  
COSP=COS(D)
SINP=SIN(D)
COST=COS(DEGR)
SINT=SIN(DEGR)

3425  
NTH=NTH+1
CTSP=COST*SINP
ZK=ZLAM*COST
YK=ZLAM*SINP*SINT
IOLD=1
INEW=2
FLDZ=FLDY=(0.0,0.0)
YOLD=SEN
ZI=YA=(0.,0.)

3450  
CONTINU
IF(P(1,1OLD) .NE. P(1,INEW)) GO TO 4000
Z=P(2,1OLD)
ERY=P(3,1OLD)
ERZ=P(4,1OLD)
PH=P(5,1OLD)

DZ=(P(2,INEW)-Z)*RPART
DERY=(P(3,INEW)-ERY)*RPART
DERZ=(P(4,INEW)-ERZ)*RPART
DPh=(P(5,INEW)-PH)*RPART
CTEMP=CEXP(CMPLX(0.0,ZK*Z-PH))
CZ1=ERZ*COSP*CTEMP
CY1=(ERY*SINT+ERZ*CTSP)*CTEMP
TSZ=TSY=(0.0,0.0)

DO 3700 N=1,NPARTS
Z=Z+DZ
ERY=ERY+DERY
ERZ=ERZ+DERZ
PH=PH+DPh
CTEMP=CEXP(CMPLX(0.0,ZK-Z-PH))
CZ2=ERZ*COSP*CTEMP
CY2=(ERY*SI+ERZ*CTSP)*CTEMP
TSZ=TSZ+CZ1+CZ2
TSY=TSY+CY1+CY2
CZ1=CZ2
CY1=CY2
3700 CONTINUE
ZI=ZI+TSZ*(.5*DZ)
YI=YI+TSY*(.5*DZ)
3900 IOLD=IOLD+1
INEW=INEW+1
GO TO 3450
4000 CONTINUE
YNEW=P(1,1,1,1,1)
IF(YOLD.EQ.SEN) GO TO 4400
4200 DZI=(ZI-ZIOLD)*RPart
DYI=(YI-YIOLD)*RPart
DY=(YNEW-YOLD)*RPart
CTEMP=CEXP(CMPLX(0.0,YK*YOLD))
CZ1=ZIOLD*CTEMP
CY1=YIOLD*CTEMP
TSZ=TSY=(0.0,0.0)
DO 4300 N=1,NPARTS
YOLD=YOLD+DY
ZIOLD=ZIOLD+DZI
YIOLD=YIOLD+DYI
CTEMP=CEXP(CMPLX(0.0,YK*YOLD))
CZ2=ZIOLD*CTEMP
CY2=YIOLD*CTEMP
TSZ=TSZ+CZ1+CZ2
TSY=TSY+CY1+CY2
CZ1=CZ2
CY1=CY2
4300 CONTINUE
FLDZ=FLDZ+TSZ*(.5*DY)
FLDY=FLDY+TSY*(.5*DY)
4400 CONTINUE
YOLD=YNEW
ZIOLD=ZI
YIOLD=YL
ZI=YL=(0,0,0)
IF(P1,INEW,NE,SEN) GO TO 3900
FIELDZ(NTH)=CARSFLDZ
FIELDY(NTH)=CARSFLDY
FMAXZ=AMAX1(FMAXZ,FIELDZ(NTH))
FMAXY=AMAX1(FMAXY,FIELDY(NTH))
D=D+DIER
IF(D.GT.OMIR) GO TO 5000
IF(LOOPHI) 400,3400
5000 CONTINUE
D=MINOR(1)
FMZDB=FMYDB=PWRMDB=-60,0
IF(FMAXZ.GT.1.0E-10) FMZDB=20.0*ALOG10(FMAXZ).
IF(FMAXY.GT.1.0E-10) FMYDB=20.0*ALOG10(FMAXY)
IF(FMAXZ*FMAXZ+FMAXY*FMAXY.GT.1.0E-10) PWRMDB=10.0*ALOG10(FMAXZ*FMAXZ+FMAXY*FMAXY)
PRINT 600, MAJOR,AMAJOR,MINOR,AMINOR
600 FORMAT(1HI,1X,A5,FR.3,5X,A5,3FR.3)
PRINT 666, MINOR
666 FORMAT(4X,A10,*DBZ/Z*,5X,*DBZ/Y*,6X,*DBZ/Z*,6X,*DBZ/Y*,6X,*DBZ/Z*,6X,*PWRDB*)
DO 700 I=1,NTH
DBZ=BY=PWR(1)=-60,0
IF(FIELDZ(I).GT.1.0E-10) DBZ=20.0*ALOG10(FIELDZ(I))
IF(FIELDY(I).GT.1.0E-10) DBY=20.0*ALOG10(FIELDY(I))
IF(FIELDZ(I)**2+FIELDY(I)**2.GT.1.0E-10) PWR(I)=10.0*ALOG10(FIELDZ(I)**2+FIELDY(I)**2)
DBZZ=DBZ-FMZDB
DBYY=DBY-FMYDB
DBZY=DBZ-FMYDB
DBYZ=DBY-FMZDB
PWRDB=PWR(I)-PWRDB
PRINT 690, D,DBZZ,DBYZ,DBZY,DBYY,PWRDB
690 FORMAT(F9.3,5F1.5)
D=D+AMINOR(3)
FIELDZ(I)=DBZ
FIELDY(I)=DBY
FOR PLOT
FOR PLOT
700 CONTINUE
PRINT 750, FMAXZ, FMZDB, FMAXY, FMYDB
750 FORMAT(* 20LOG(MAX(FIELD-Z))=*20LOG(*G15.7,*)=*G15.7,*)
        * 20LOG(MAX(FIELD-Y))=*20LOG(*G15.7,*)=*G15.7)
        PRINT 755, NPARTS
755 FORMAT(* INTERPOLATION NUMBER USED FOR INTEGRATION IS...........*I5)
        PRINT 775, MAJOR, DEG
775 FORMAT(*I6//,20X,A6,F6.2)
        CALL PLOT(50H NORMALIZED Z-COMPONENT OF SECONDARY PATTERN (DB),
                FMZDB, FIELDZ, NTH)
        PRINT 775, MAJOR, DEG
        CALL PLOT(50H NORMALIZED Y-COMPONENT OF SECONDARY PATTERN (DB),
                FMYDB, FIELDY, NTH)
        PRINT 775, MAJOR, DEG
        CALL PLOT(50H NORMALIZED POWER PATTERN (DB)
                ,PWRMDB, PWER, NTH)
        GO TO 100
9000 CONTINUE
RETURN
END
9. SUBROUTINE INTERP

INTERP is called by APERTUR to interpolate a point at the edge of the aperture plane ellipse when, in the sequence of sorting out the reflected rays, the point of intersection of a reflected ray with the aperture plane falling inside (outside) the aperture plane ellipse is followed by another which falls outside (inside) the aperture plane ellipse, the two points not being on the opposite sides of \( Y_e \) as shown in Figure 21.

Of these two points, based upon ENEAR1 and ENEAR2, the point closer to the ellipse circumference is selected. The \( y \)-coordinate of this closer point, \( Y_{EDGE} \) is then taken as the \( y \)-coordinate of the interpolated edge point and a

![Figure 21 -- Interpolated Point at the Edge of Aperture Plane Ellipse](attachment:image.png)
corresponding value of z-coordinate, \( Z_{\text{EDGE}} \) is calculated using the equation of the ellipse. Next, calculating the distances \( D_1 \) and \( D_2 \), \( ER_y \), \( ER_z \), and Phase are linearly interpolated at the edge point and assigned to \( \text{PINT}(3) \), \( \text{PINT}(4) \), and \( \text{PINT}(5) \), respectively.
SUBROUTINE INTERP(P1,P2,PINT)
COMM/COMMON/PARAMS/TITLE(16),ADOREF,XLAM,GRID,SURFACE,APROTA,FEED(3),
  ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAEX,HFMIEX,BMTP,BMPP,
  NT,NP,NPOINT,MAXPTS,RELP
DIMENSION P1(5),P2(5),PINT(5)
ENEAR1=((P1(1)-YC)**2+(P1(2)-ZC)**2)*ABS(1.0-
  1.0/(((P1(1)-YC)/HFMAEX)**2+((P1(2)-ZC)/HFMIEX)**2))
ENEAR2=((P2(1)-YC)**2+(P2(2)-ZC)**2)*ABS(1.0-
  1.0/(((P2(1)-YC)/HFMAEX)**2+((P2(2)-ZC)/HFMIEX)**2))
YEDGE=P1(1)
IF(ENEAR1.GT ENEAR2) YEDGE=P2(1)
TD1=YEDGE-YC
IF(ABS(TD1).LT HFMAEX) GO TO 7
YEDGE=SIGN(HFMAEX,TD1)+YC
TD1=HFMAEX
CONTINUE
ZEDGE=HFMIEX*SQRT(1.0-TD1*TD1/(HFMAEX*HFMAEX))
ZEDGE=SIGN(ZEDGE,P2(2)-ZC)+ZC
D1=SQRT(((P1(1)-YEDGE)**2+((P1(2)-ZEDGE)**2))
D2=SQRT(((P2(1)-YEDGE)**2+((P2(2)-ZEDGE)**2))
PINT(1)=YEDGE
PINT(2)=ZEDGE
IF(D1.LT D2) GO TO 20
PINT(3)=P2(3)+(((P1(3)-P2(3))*D2)/(D1+D2))
PINT(4)=P2(4)+(((P1(4)-P2(4))*D2)/(D1+D2))
PINT(5)=P2(5)+(((P1(5)-P2(5))*D2)/(D1+D2))
GO TO 25
20 PINT(3)=P1(3)+(((P2(3)-P1(3))*D1)/(D1+D2))
PINT(4)=P1(4)+(((P2(4)-P1(4))*D1)/(D1+D2))
PINT(5)=P1(5)+(((P2(5)-P1(5))*D1)/(D1+D2))
CONTINUE
RETURN
END
10. OTHER SUBROUTINES

MULT32: This subroutine multiplies a 3x3 matrix OPA by a 3x2 matrix OPB to yield a 3x2 matrix ROP. A listing of the subroutine is shown in Figure 22.

```plaintext
SUBROUTINE MULT32(ROP, OPA, OPB)
DIMENSION ROP(3,2), OPA(3,3), OPB(3,2)
ROP(1,2) = OPA(1,1)*OPB(1,2) + OPA(1,2)*OPB(2,2) + OPA(1,3)*OPB(3,2)
ROP(2,2) = OPA(2,1)*OPB(1,2) + OPA(2,2)*OPB(2,2) + OPA(2,3)*OPB(3,2)
ROP(3,2) = OPA(3,1)*OPB(1,2) + OPA(3,2)*OPB(2,2) + OPA(3,3)*OPB(3,2)
ROP(1,1) = OPA(1,1)*OPB(1,1) + OPA(1,2)*OPB(2,1) + OPA(1,3)*OPB(3,1)
ROP(2,1) = OPA(2,1)*OPB(1,1) + OPA(2,2)*OPB(2,1) + OPA(2,3)*OPB(3,1)
ROP(3,1) = OPA(3,1)*OPB(1,1) + OPA(3,2)*OPB(2,1) + OPA(3,3)*OPB(3,1)
RETURN
END
```

Figure 22 -- Subroutine MULT32

FQKSORT: This subroutine arranges in ascending order the NN sets of numbers in the A(LTH,NN) array according to numbers stored with LTH=1,2,...etc. A listing of this subroutine is presented in Figure 23.

CURV1: This subroutine determines the parameters necessary to compute an interpolating spline under tension through a sequence of functional values. The slopes at the two ends of the curve may be specified or omitted. For actual computation of points on the curve, it is necessary to call the function CURV2. At the time of input:

- N: Number of values to be interpolated,
- X: An array of the N increasing abscissae of the functional values,
SUBROUTINE FQKSORT(A,LTH,NN)
INTEGER STACK(15)
DIMENSION A(LTH,NN)
DATA KONS/1000000/
IF(NN * LE. *32768) GO TO 200
PRINT 100, NN
100 FORMAT(* FATAL ERROR* FQKSORT WILL NOT SORT*I6* RECORDS*)
STOP 001
200 K=1
STACK(1)=KONS+NN
1000 IS=I=STACK(1)/KONS
NS=N=STACK(1)-I*KONS
IINC=0
NINC=1
2000 IF(I .GE. N) GO TO 6000
3000 DO 3500 J=1,LTH
   IF(A(J,I) -A(J,N)) 5000,3500,3600
3500 CONTINUE
   GO TO 5000
3600 CONTINUE
   DO 4000 J=1,LTH
      SWAP=A(J,I)
      A(J,I)=A(J,N)
4000 A(J,N)=SWAP
   IINC=1-IINC
   NINC=1-NINC
5000 I=I+IINC
   N=N-NINC
   GO TO 2000
6000 I=I+1
   N=I+1
   I=I-1
   K1L=I-IS
   K2L=NS-N

Figure 23 -- Subroutine FQKSORT
IF(K1L .LE. 0) GO TO 7000
STACK(1) = IS*KONS + 1
IF(K2L .LE. 0) GO TO 1000
L=1
IF(K2L .GT. K1L) L=2
J=K
6500 STACK(J+1) = STACK(J)
J=J-1
IF(J .GE. L) GO TO 6500
STACK(L) = N*KONS + NS
K=K+1
GO TO 1000
7000 IF(K2L .LE. 0) GO TO 8000
STACK(1) = N*KONS + NS
GO TO 1000
8000 K=K-1
DO 8500 J=1,K
8500 STACK(J) = STACK(J+1)
IF(K .GT. 0) GO TO 1000
RETURN
END
Y

An array of the N ordinates, i.e., \( Y(K) \) is the functional value corresponding to \( X(K) \).

SLP1 and SLPN

The desired values for the first derivative of the curve at \( X(1) \) and \( X(N) \), respectively. If the quantity SIGMA is negative, these values will be determined internally and the user need only furnish place holding parameters for SLP1 and SLP2.

YP

An array of length N,

TEMP

An array of length N used for scratch,

SIGMA

Contains the tension factor. This is non zero and indicates the desired curviness of the spline fit. If the absolute value of SIGMA is nearly zero, (e.g. 0.001), the resulting curve is approximately a cubic spline. If the absolute value of SIGMA is large (e.g. 50.0), the resulting curve is nearly a polygonal line.

Upon return from this subroutine YP contains values proportional to the second derivative of the curve at the given nodes. N, X, Y, SLP1, SLPN, and SIGMA are unaltered.

A listing of this subroutine is presented in Figure 24.

CURV2: This function subroutine interpolates a curve at a given point using a spline under tension. The subroutine CURV1 should be called earlier to determine certain necessary parameters. At the time of input T contains a real value to be mapped onto the interpolating curve, N, X, Y, YP, and SIGMA are defined the same as in CURV1, and IT is an integer switch. If IT is not 1, this indicates that the function has been called previously with the same N, X, Y, YP, and SIGMA but the current value of T exceeds the previous value. Upon return, CURV2 contains the interpolated value. A listing of this function subroutine is shown in Figure 25.
SUBROUTINE CURV1 (N,X,Y,SLP1,SLPN,YP,TEMP,SIGMA)
INTEGER N
REAL X(N),Y(N),SLP1,SLPN,YP(N),TEMP(N),SIGMA
NM1 = N-1
NP1 = N+1
DELFX1 = X(2)-X(1)
DX1 = (Y(2)-Y(1))/DELFX1
IF (SIGMA.LT.0.) GO TO 5
SLPPI = SLP1
SLPNN = SLPN
1 SIGMA = ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
DELS = SIGMA*DELFX1
EXPS = EXP(DELS)
SINH = .5*(EXPS-1./EXPS)
SINHN = 1./(DELFX1*SINH)
DIAG1 = SINHN*(DELS*.5*(EXPS+1./EXPS)-SINH)
DIAGN = 1./(DIAG1
YP(1) = DIAGN*(DX1-SLPPI)
SPDIAG = SINHN*(SINH-SDEL)
TEMP(1) = DIAGN*SPDIAG
IF (N.EQ.2) GO TO 3
DO 2 I = 2,NM1
DELFX2 = X(I+1)-X(I)
DX2 = (Y(I+1)-Y(I))/DELFX2
DELS = SIGMA*DELFX2
EXPS = EXP(DELS)
SINH = .5*(EXPS-1./EXPS)
SINHN = 1./(DELFX2*SINH)
DIAG2 = SINHN*(DELS*(.5*(EXPS+1./EXPS))-SINH)
DIAGN = 1.)/(DIAG1+DIAG2-SPDIAG*TEMP(I-1))
YP(I) = DIAGN*(DX2-DX1-SPDIAG*YP(I-1))
SPDIAG = SINHN*(SINH-DELS)
TEMP(I) = DIAGN*SPDIAG

Figure 24 -- Subroutine CURV1
DX1 = DX2
2 DIAG1 = DIAG2
3 DIAGIN = 1./(DIAG1-SPDIAG*TEMP(NM1))
   YP(N) = DIAGIN*(SLPPN-DX2-SPDIAG*YP(NM1))
00 4 I = 2,N
   IBAK = NP1-I
4 YP(IBAK) = YP(IBAK)-TEMP(IBAK)*YP(IBAK+1)
RETURN
5 IF (N.EQ.2) GO TO 6
   DELX2 = X(3)-X(2)
   DELX12 = X(3)-X(1)
   C1 = -(DELX12+DELX1)/DELX12/DELX1
   C2 = DELX12/DELX1/DELX2
   C3 = -DELX1/DELX12/DELX2
   SLPP1 = C1*Y(1)+C2*Y(2)+C3*Y(3)
   DELN = X(N)-X(NM1)
   DELNM1 = X(NM1)-X(N-2)
   DELNN = X(N)-X(N-2)
   C1 = (DELNN+DELN)/DELNN/DELN
   C2 = -DELMN/DELMN/DELMN
   C3 = DELN/DELMN/DELMN
   SLPPN = C3*Y(N-2)+C2*Y(NM1)+C1*Y(N)
GO TO 1
6 YP(1) = 0.
   YP(2) = 0.
RETURN
END

Figure 24 -- Subroutine CURVI -- Continued
FUNCTION CURV2 (T,N,X,Y,YP,SIGMA,IT)  
INTEGER N,IT  
REAL T,X(N),Y(N),YP(N),SIGMA  
S = X(N) - X(1)  
SIGMAP = ABS(SIGMA) * FLOAT(N-1)/S  
IF (IT.EQ.1) I1 = 2  
1 DO 2 I = I1,N  
IF (X(I) - T) .LT. 2,2,3  
2 CONTINUE  
3 IF (X(I-1) .LE. T .OR. T .LE. X(I)) GO TO 4  
   I1 = 2  
   GO TO 1  
4 DEL1 = T - X(I-1)  
   DEL2 = X(I) - T  
   DELS = X(I) - X(I-1)  
   EXPS1 = EXP(SIGMAP*DEL1)  
   SINHD1 = .5*(EXPS1-1./EXPS1)  
   EXPS = EXP(SIGMAP*DEL2)  
   SINHD2 = .5*(EXPS-1./EXPS)  
   EXPS = EXPS1*EXPS  
   SINHS = .5*(EXPS-1./EXPS)  
   CURV2 = (YP(I)*SINHD1+YP(I-1)*SINHD2)/SINHS+(Y(I)-YP(I-1))*DEL1+(Y(I-1)-YP(I-1))*DEL2)/DFLS  
   I1 = I  
2 CONTINUE  
RETURN  
END

Figure 25 -- Function CURV2

PLOT: This subroutine generates a line printer plot using NT values stored in the F array and labels the plot as NAME. FMAX is the maximum value of F. The points less than -60 dB are indicated by '<' sign on the horizontal axis. The listing of this subroutine is presented in Figure 26.

SETM and MOVEM: These two small subroutines are used throughout the program. The subroutine SETM sets the first N consecutive
SUBROUTINE PLOT(NAME,FMAX,F,NT)
DIMENSION NUM(12),IFRM(4),F(1),NAME(5)
DATA IFRM/0,0,0,0,6HX,*,
DATA NUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,2H10,2H11/
PRINT 100,'NAME
100 FORMAT(10X,5A10,'/5X,*=-60*,7X,*=-50*,7X,*=-40*
+7X,*=-30*,7X,*=-20*,7X,*=-10*,7X,*=0*/5X,***,9X,***
+9X,***,9X,***,9X,***
PRINT 1100
1100 FORMAT(5X,* *,12(*---- *))
DO 2000 I=1,NT
  IFRM(2)=IFRM(3)=IFRM(4)=10H
  FX=F(I)-FMAX
  IF (FX,GT,0.0) GOTO 1200
  IP=(F(I)-FMAX+60.0)
  IF(IP) 1300,1350,1400
1200 IFRM(1)=10H(5X,**=*),
  IFRM(2)=10H
  IFRM(3)=10H120
  GOTO 1490
1300 IFRM(1)=10H(5X,<=*) $ GO TO 1500
1350 IFRM(1)=10H(5X,=*), $ GO TO 1500
1400 IFRM(1)=10H(5X,=*), $ IP=IP-1
  IF(IP .NE. 0) GOTO 1450
  IFRM(2)=10H(*), $ GO TO 1500
1450 IP=IP+0.1 $ IP2=IP-IP1*10.0
  IFRM(2)=NUM(IP1+1) $ IFRM(3)=NUM(IP2+1)
1490 IFRM(4)=10HX,*,*)
1500 PRINT IFRM
2000 CONTINUE
RETURN
END

Figure 26 -- Subroutine PLOT
locations of the array ANEW to X. The subroutine MOVEM is a little different. It transfers the first N consecutive numbers from the array AOLD to the N consecutive locations in the array ANEW. A listing of these two is presented in Figure 27.
SUBROUTINE MOVEM(AOLD,ANEW,N)
DIMENSION AOLD(1),ANEW(1)
IF(N.LE.0) RETURN
DO 100 I=1,N
ANEW(I)=AOLD(I)
100 CONTINUE
RETURN
END

SUBROUTINE SETM(X,ANEW,N)
DIMENSION ANEW(1)
IF(N.LE.0) RETURN
DO 100 I=1,N
ANEW(I)=X
100 CONTINUE
RETURN
END

Figure 27 --- Subroutines MOVEM and SETM
11. CONCLUSIONS AND REMARKS

The computer program presented here has been successfully used to compute the radiation properties of large spherical reflectors [3] and to study the optimization of parameters for wide band radiometric applications [4]. The computation time is very reasonable, e.g., on a CDC 6600 computer, the computation time per far field point has been found to be 0.63 sec for a 700 λ diameter spherical reflector. Since the aperture integration time required for computation of the far field depends upon the number of aperture data points, which in turn is determined by the angular increment between the rays used in the feed pattern, the computation time is somewhat insensitive to the absolute size of the reflector.

There are only three places in the entire program where a decision is made based upon the reflector surface i.e., whether it is paraboloidal, spherical, or ellipsoidal - once in the subroutine NPUT and twice in the subroutine APERTUR. These places have been labeled with ******* in columns 73-80. In the subroutine NPUT the surface information is used to output the information in appropriate format and in the subroutine APERTUR it is first used to compute R and then to write the three components of the unit normal to the reflector surface, NHAT(1), NHAT(2) and NHAT(3). Modifying the NPUT and APERTUR subroutines at these three places, the program can be used for any other reflector surface that can be expressed analytically.
12. REFERENCES


A computer program to calculate the radiation properties of the reflector antennas is presented. In its present form, the program can be used for paraboloidal, spherical, or ellipsoidal reflector surfaces. However, it can be easily modified to handle any surface that can be expressed analytically. The program is general enough to allow any arbitrary location and pointing angle for the feed antenna. The effect of blockage due to the feed horn is also included in the computations.

The computer program is based upon the technique of tracing the rays from the feed antenna to the reflector to an aperture plane. The far field radiation properties are then calculated by performing a double integration over the field points in the aperture plane. To facilitate the computation of double integral, however, the field points are first aligned along the equispaced straight lines in the aperture plane. The computation time is relatively insensitive to the absolute size of the aperture and even though no limits on the largest reflector size have been determined, the program has been used for reflector diameters of 1000 λ.

Key Words: Reflector Antennas, Offset Reflector Antennas, Spherical Reflector Antennas, Ellipsoidal Reflector Antennas, Large Reflectors.