NASA Technical Memorandum 78721

A COMPUTER PROGRAM TO CALCULATE RADIATION PROPERTIES OF REFLECTOR ANTENNAS

(NASA-TM-78721) A COMPUTER PROGRAM TO CALCULATE RADIATION PROPERTIES OF REFLECTOR ANTENNAS (NASA) 77 p HC A05/MF A01 CSCL 20N
N80-26574
Unclas
G3/32 23578

Pradeep K. Agrawal

May 1978

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

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.

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 ($(x^2/a^2)+(y^2+z^2)/b^2=1$ 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

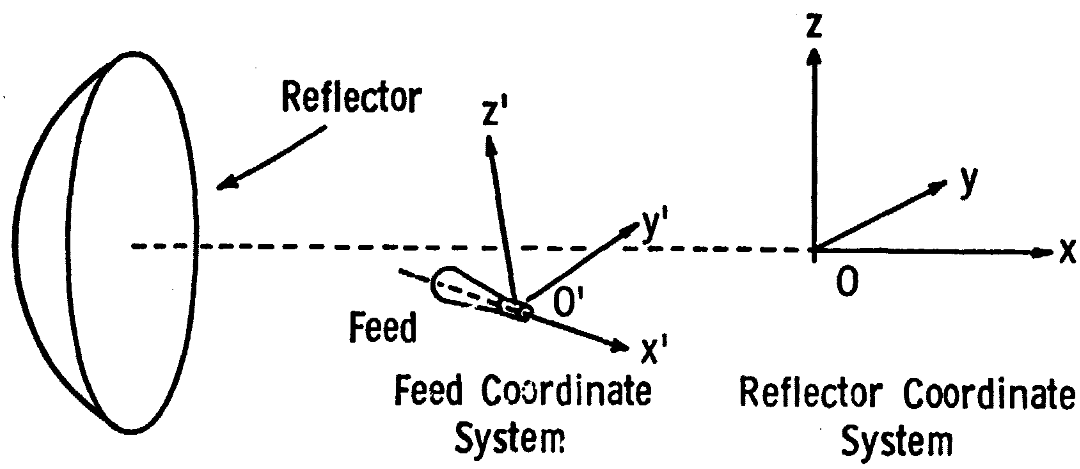


Figure 1 -- Feed and Reflector Coordinate Systems

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 $\vec{0'0}$ expressed in the primed coordinate system i.e., the coordinates of point 0 in the feed coordinate system. The three components of this vector $\vec{0'0}$ 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 x'z'-plane about the y'-axis to make the x'-axis parallel to the x-axis. GAMMA allows the feed to be pointed

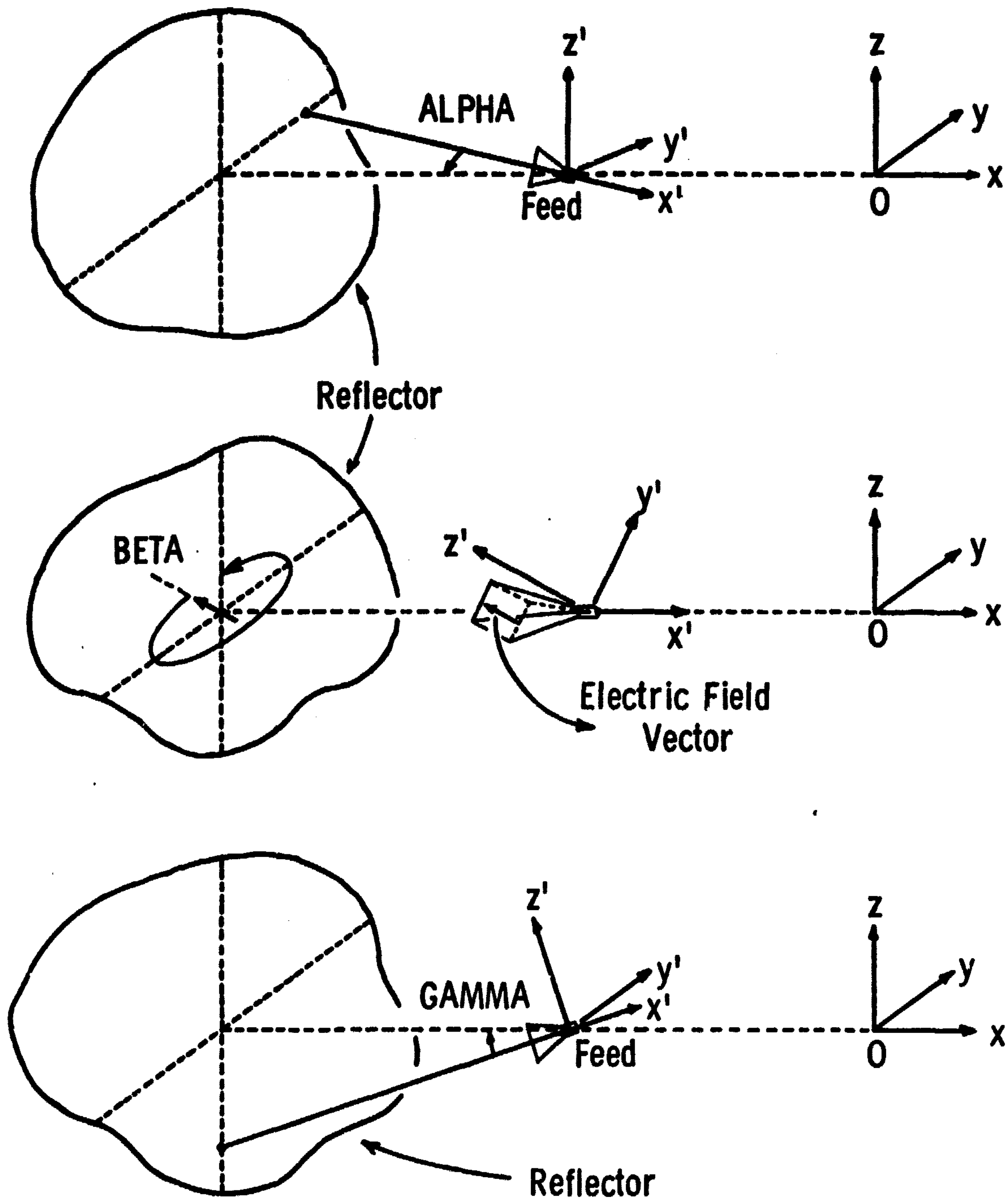


Figure 2 -- Angles ALPHA, BETA, and GAMMA

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 θ_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 θ' and ϕ' so that the rays emanating from the feed over these ranges of θ' and ϕ' more than illuminate the reflector, i.e., the rays with the upper and the lower limit values of θ' and ϕ' miss the reflector. A typical ray along (θ_i', ϕ_j') direction is shown in Figure 3. Index i is used to denote the change in the angle θ' which has NT equispaced values between θ_1' and θ_{NT}'. Similarly, index j is used to denote the change in the angle ϕ' which has NP equispaced values between ϕ_1' and ϕ_{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 θ' and ϕ' are required to be such that at least the rays with $\theta' = \theta_1'$ and θ_{NT}' and the rays with $\phi' = \phi_1'$ and ϕ_{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_θ' and E_ϕ', the two components of the electric field along the ray, the phase of the electric field, and the

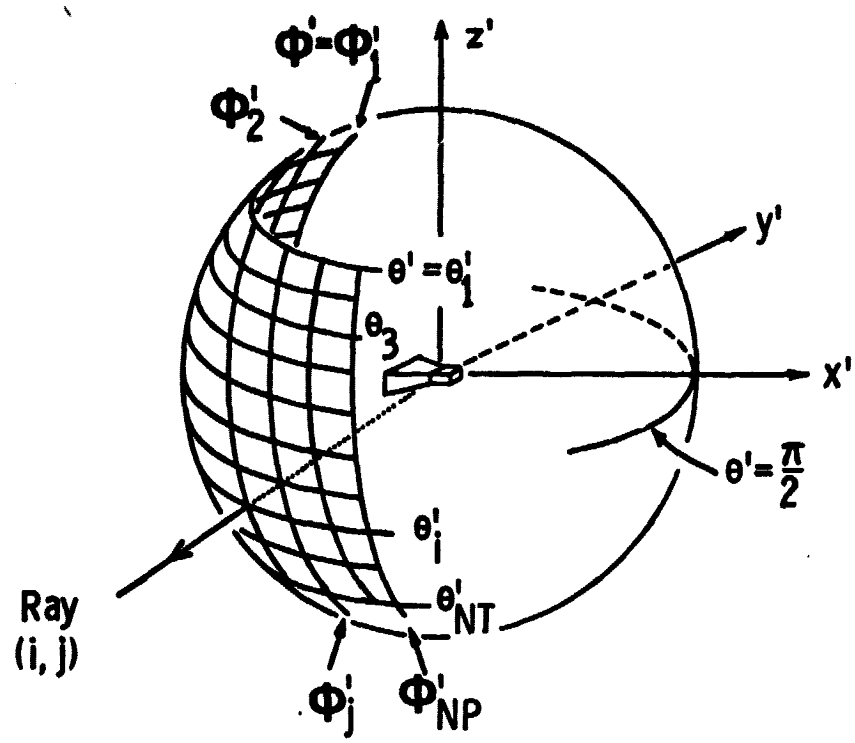


Figure 3 -- Feed Pattern Angles

angles (θ_i', ϕ_j') specifying the direction of the ray. These five quantities are stored for each ray in a three-dimensional array P(5,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, NHAT(1), NHAT(2), and 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_θ and E_ϕ 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

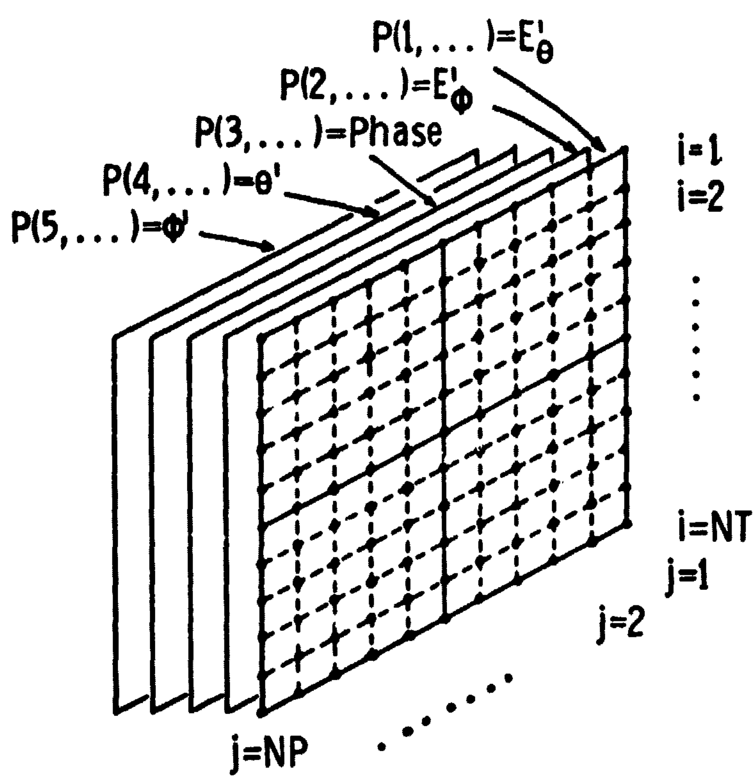


Figure 4 -- P Array and the Feed Pattern Information

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.

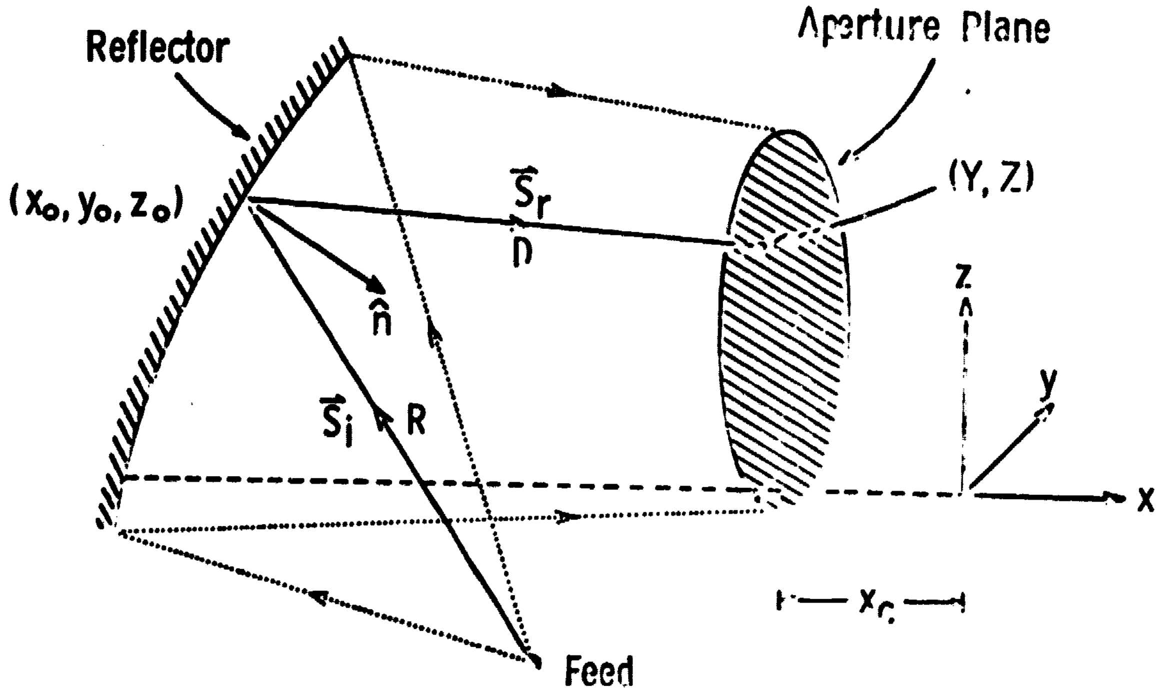


Figure 5 -- Geometry of the Reflector Antenna

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.

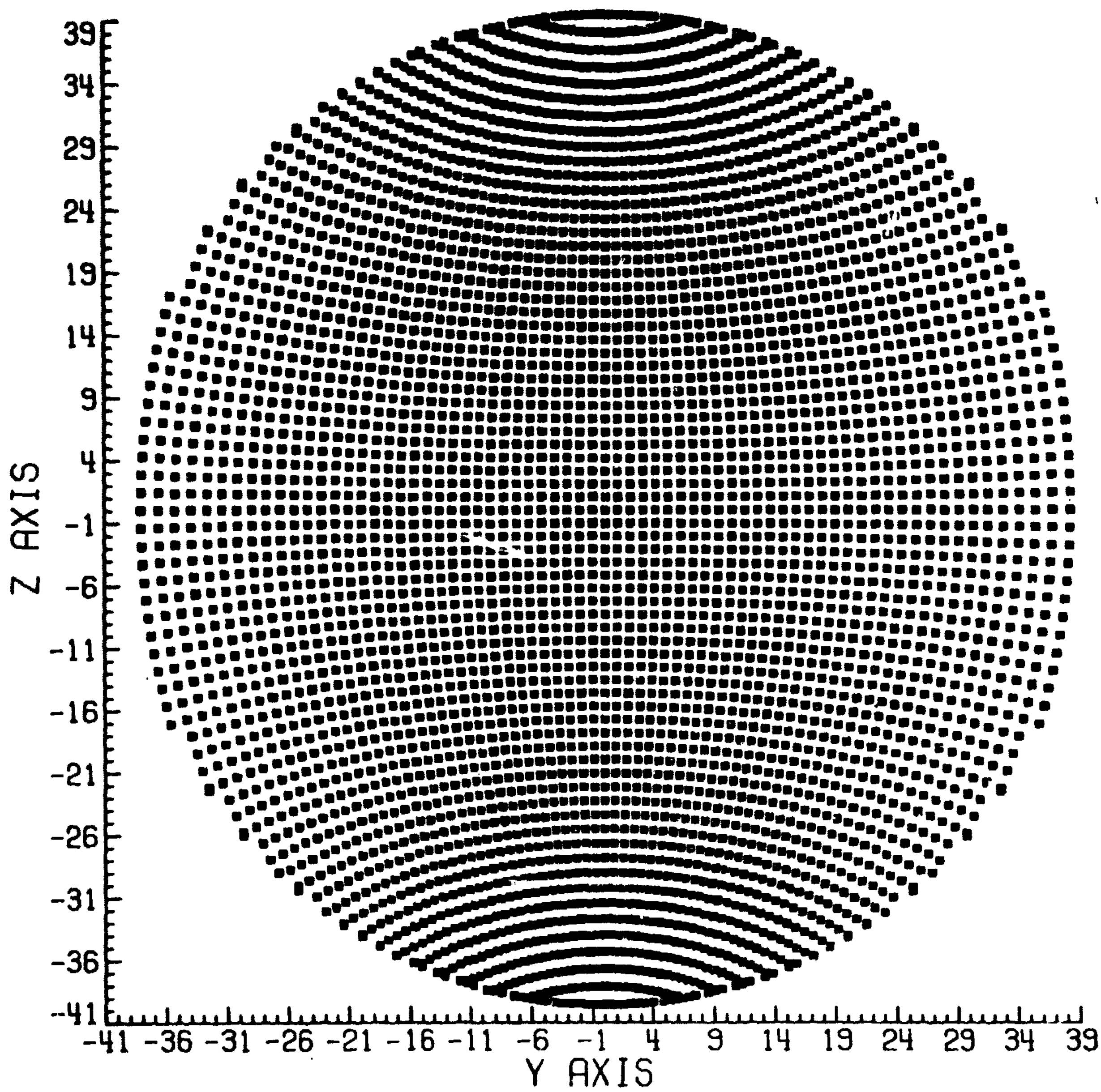


Figure 6 -- Points of Intersection of the Reflected Rays and the Aperture Plane

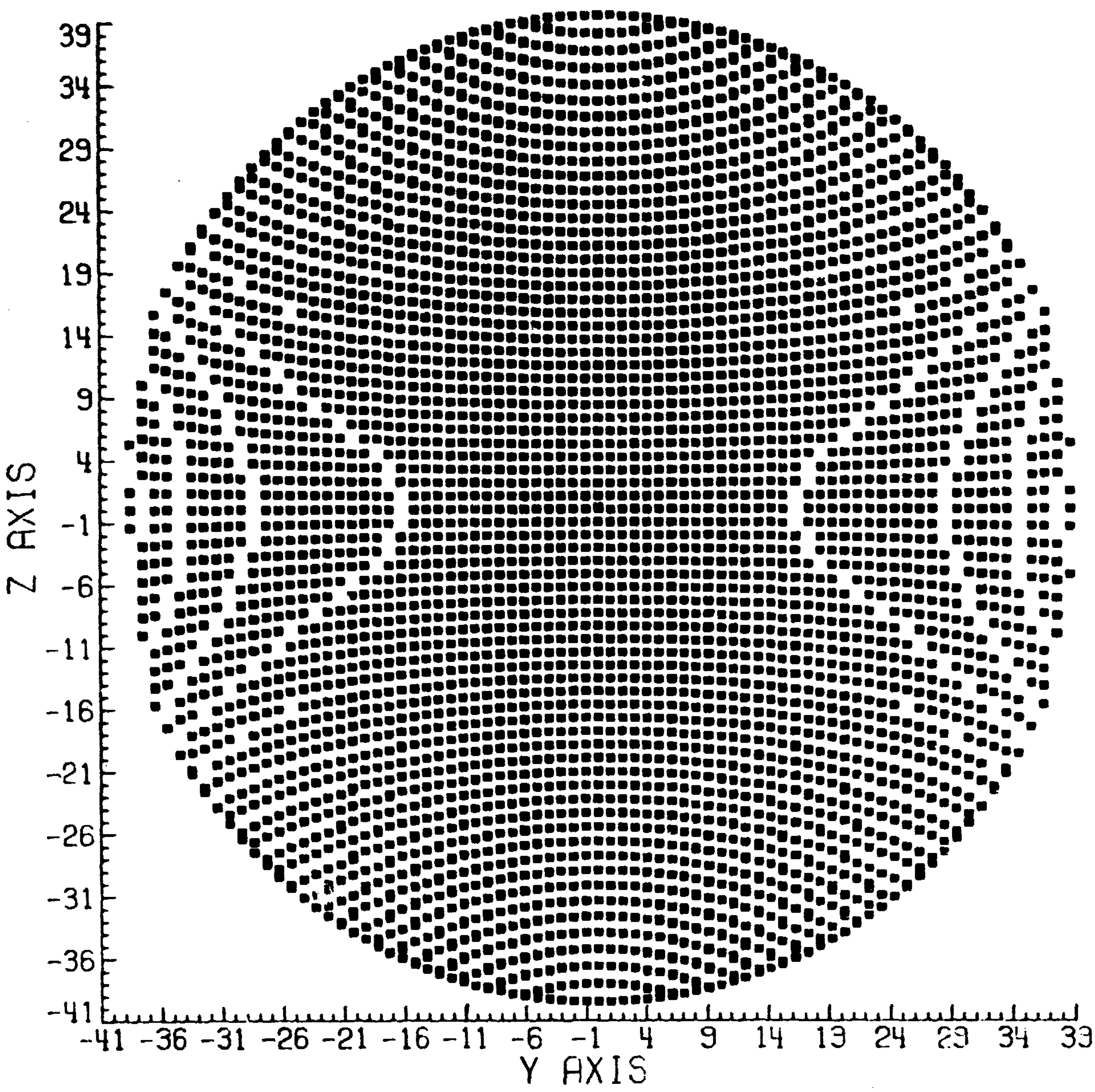


Figure 7 -- Points of Intersection of the Reflected Rays and the Aperture Plane after Aligning along Equispaced y = Constant Lines

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.

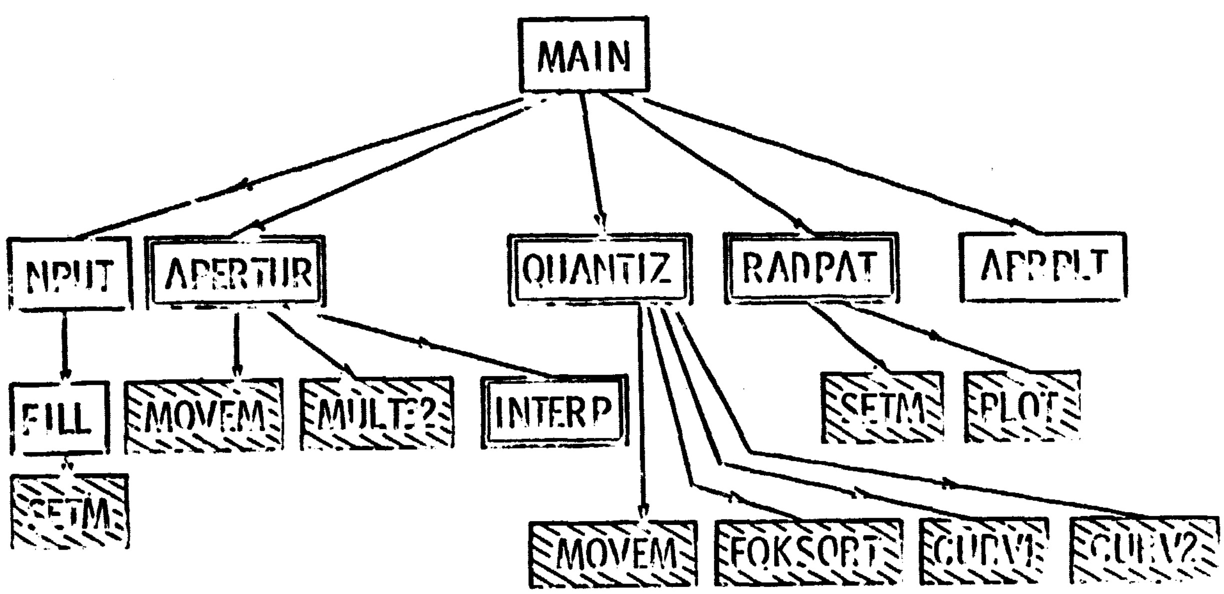


Figure 8 -- Block Diagram of the Computer Program

ORIGINAL PAGE IS
OF POOR QUALITY

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 equi-spaced 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'_θ, E'_ϕ, phase, θ', and ϕ'. 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 alphanumeric information in columns 1-80 for title or identification purposes.

```
      PROGRAM REFLCTR(INPUT,OUTPUT,TAPE8,TAPE20,
     .                TAPE7=INPUT,TAPE6=OUTPUT,PUNCH)
      DIMENSION YPLT(1),ZPLT(1),EORDB(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,HFMAEX,HFMIEX,BMTP,BMPP,
     .              NT,NP,NPOINT,MAXPTS,BELLP
      COMMON/MATH/PI,PI2,PID2,DTOR,RTOD
      EQUIVALENCE (YPLT,P(1)),(ZPLT,P(2751)),(EORDB,P(5501)),
     .           (Y,P(8251)),(Z,P(11001))
      DATA MAXPTS/2750/,NT,NP,NPOINT/0,0,0/
776   FORMAT(*      ----------------- FINISHED INPUT   ----------------*)
777   FORMAT(*      ---------------- FINISHED APERTUR ----------------*)
778   FORMAT(*      ---------------- FINISHED QUANTIZ ----------------*)
779   FORMAT(*      ----------------- FINISHED RADPAT ----------------*)
      REWIND 20                                                       $ ARRAYS
      CALL NPUT(P)
      PRINT 776
      CALL APERTUR(P,NT,NP)
      PRINT 777
      CALL QUANTIZ(P,MAXPTS)
      ENDFILE 20                                                      TAPF 20
      PRINT 778
      CALL RADPAT(P,MAXPTS)
      PRINT 779
      REWIND 20                                                       TAPF 20
      IF(APRDTA.GT.0.0) CALL APRPLT(YPLT,ZPLT,EORDB,Y,Z,MAXPTS)
      STOP
      END
```

Figure 9 -- The Main Computer Program

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE NPUT(P)
      COMMON/PARAMS/TITLE(16),AORORF,XLAM,GRID,SURFACE,APRDTA,FEED(3),
     .                ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAEX,HFMIEX,BMTP,BMPP,
     .                NT,NP,NPOINT,MAXPTS,BELLP
      COMMON/BLOCKG/YCBL,ZCBL,HFMABL,HFMIBL
      COMMON/PATTERN/PHI(3),THETA(3)
      COMMON/MATH/PI,PI2,PID2,DTOR,RTOD
      INTEGER TITLE
      READ 200, TITLE
      READ *, AORORF,XLAM,GRID,SURFACE,APRDTA,BELLP
      READ *, FEED,ALPHA,BETA,GAMMA
      READ *, XC,YC,ZC,HFMAEX,HFMIEX,YCBL,ZCBL,HFMABL,HFMIBL
      READ *, PHI,THETA
      IF(APRDTA.GT.0.0) WRITE(20,555)TITLE
      IF(APRDTA.GT.0.0) WRITE(20,556)FEED,AORORF,XLAM,GRID,ALPHA,BETA,   TAPE 20
     .                            GAMMA,XC,YC,ZC,HFMAEX,HFMIEX        TAPE 20
      IF (SURFACE) 140,150,160                                        ********
 140  PRINT 579, TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,AORORF,BELLP
      GO TO 170
 150  PRINT 580, TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,AORORF
      GO TO 170
 160  PRINT 581, TITLE,XLAM,FEED,ALPHA,BETA,GAMMA,AORORF
 170  PRINT 582, XC,YC,ZC,HFMAEX,HFMIEX,GRID,YCBL,ZCBL,HFMABL,HFMIBL,
     .          THETA,PHI
 200  FORMAT(8A10)
 555  FORMAT(1X,8A10)
 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(1H1,////,11X,*SPHERICAL 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
      .       /* RADIUS OF THE REFLECTOR SPHERE......................*F7.2)
  581  FORMAT(1H1,////,11X,*PARABOLIC 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
      .       /* 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 FEED PATTERN (L,H,I - DEGREES).....*3F7.2
      .       /* PHI RANGE FOR FEED PATTERN (L,H,I - DEGREES).......*3F7.2
      .        )
  590  FORMAT(* ----- INSUFFICIENT WORK STORAGE, NEEDED *I5* AVAILABLE IS
      . ONLY *I5* ----- *)
       PI=ACOS(-1.0)
       PI2=PI+PI
       PID2=0.5*PI
       DTOR=PI/180.
       RTOD=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

C013

```
      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.

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

Card 4: Feed information.

FEED	A one-dimensional array containing coordinates of the (x,y,z) origin with respect to the feed coordinate system,
ALPHA, BETA, GAMMA	Three rotation angles that define the pointing angle of the feed. (See Figure 2 for definition).

Card 5: Aperture plane information.

XC,YC,ZC	x-,y-,z-coordinates of the center of the aperture plane ellipse,
HFMAEX,HFMIEX	Semi major and minor axes (along y- and z-directions) of the aperture plane ellipse,
YCBL,ZCBL	y- and z-coordinates of the center of the feed shadow ellipse in the aperture plane,
HFMABL,HFMIBL	Semi major and minor axes (along y- and z-directions) of the feed shadow ellipse on the aperture plane.

ORIGINAL PAGE IS
OF POOR QUALITY

Card 6: Illumination information.

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

Next, the number of θ' values NT and the number of ϕ' values NP are calculated. These are maximum values of indices i and j respectively in Figure 4. The product of NT and NP called NPOINT 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'_θ, E'_ϕ, phase, θ', and ϕ' 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'_θ, E'_ϕ, phase, θ', and ϕ'. 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 θ', and ϕ' 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 θ' and ϕ' 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_θ', 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.[2]. 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_θ' and the E_ϕ' components for $\theta_1' \leq \theta' \leq \theta_{NT}'$ and $\phi_1' \leq \phi' \leq \phi_{NP}'$. 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/PI,PI2,PID2,DTOR,RTOD
      DIMENSION P(5,NTX,NPX)
      CALL SETM(0.0,P,5*NPX*NTX)
      THR=THETA(1)*DTOR
      DO 300 I=1,NTX
      FAC2=PI*3.9*COS(THR)/2.68
      FACB=COS(FAC2)/(1.0-4.0*FAC2*FAC2/(PI*PI))
      IF (FAC2.NE.0.0)  GO TO 50
      FACC=1.0
      GO TO 51
   50 FACC=SIN(FAC2)/FAC2
   51 CONTINUE
      PHR=(PHI(1)-180.0)*DTOR
      DO 200 J=1,NPX
      FAC1=PI*3.9*SIN(THR)*SIN(PHR)/2.68
      FACD=COS(FAC1)/(1.0-4.0*FAC1*FAC1/(PI*PI))
      IF (FAC1.NE.0.0)  GO TO 60
      FACA=1.0
      GO TO 61
   60 FACA=SIN(FAC1)/FAC1
   61 CONTINUE
      XLZ=FACA*FACB
      XLY=FACC*FACD
      XMUL1=SIN(PHR)*COS(THR)
      XMUL2=SIN(THR)+COS(PHR)
      P(1,I,J)=XLZ*XMUL1-XLY*XMUL2
      P(2,I,J)=XLY*XMUL1-XLZ*XMUL2
      P(1,I,J)=ABS(P(1,I,J))/2.0
      P(2,I,J)=ABS(P(2,I,J))/2.0
      P(4,I,J)=THR
      P(5,I,J)=PHR+PI
      PHR=PHR+PHI(3)*DTOR
  200 CONTINUE
      THR=THR+THETA(3)*DTOR
  300 CONTINUE
      RETURN
      END
```

Figure 11 -- Subroutine FILL

QUANTIZ: 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 alignes the points in the aperture plane along the equispaced y=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 E_y and E_z (E_x being zero in the far field) for any θ and ϕ. The subroutine is set up such that a far field pattern is requested by first picking a plane (constant θ-, or constant ϕ-plane) and then specifying in that plane the initial value, the final value, and the increment size of the other variable angle (ϕ or θ, respectively). The output of this subroutine is as follows:

1. A table listing the variable angle and the quantities $E_z/(E_z)_{max}$, $E_y/(E_z)_{max}$, $E_z/(E_y)_{max}$, $E_y/(E_y)_{max}$, and $(E_y^2 + E_z^2)/[(E_y)^2_{max} + (E_z)^2_{max}]$ in decibels for each angle,
2. Values of $(E_z)_{max}$ and $(E_y)_{max}$, and
3. Line printer plots of normalized E_z and E_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 θ and φ 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     90.0 PHI     -10.0       0.0       0.5
```

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

Ex. 2:
```
MAX-T     -4.0 PHI      -5.0       2.0       1.0
```

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

Ex. 3:
```
MAX-P     10.0MAX-T      8.0      10.0       0.1
```

In $\phi=(\phi_{BMAX}+10^{\circ})$ plane, the far field is calculated for θ values between $(\theta_{BMAX}+8^{\circ})$ to $(\theta_{BMAX}+10^{\circ})$ in steps of 0.1°.

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,NTXNPX)
      COMMON/PARAMS/TITLE(16),AORORF,XLAM,GRID,SURFACE,APRDTA,FEED(3),
     .               ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAEX,HFMIEX,BMTP,BMPP,
     .               NT,NP,NPOINT,MAXPTS,BELLP
      DIMENSION YPLT(NTXNPX),ZPLT(NTXNPX),EORDB(NTXNPX),Y(NTXNPX),
     .          Z(NTXNPX),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)/80.0
      SF=5*ISF+5
      YORG=AINT(YC-(HFMAEX+5))
      ZORG=AINT(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
      DBLVEL=-ILEVEL*3.0
      J=1
      DO 752 I=1,IP
      IF (ILEVEL.EQ.11) DBLVEL=-59.0
      IF(EORDB(I).LE.DBLVEL) 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 LINPLT (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(0.0,0.0,999)
RETURN
END
```


Figure 12 -- Subroutine APRPLT -- Continued

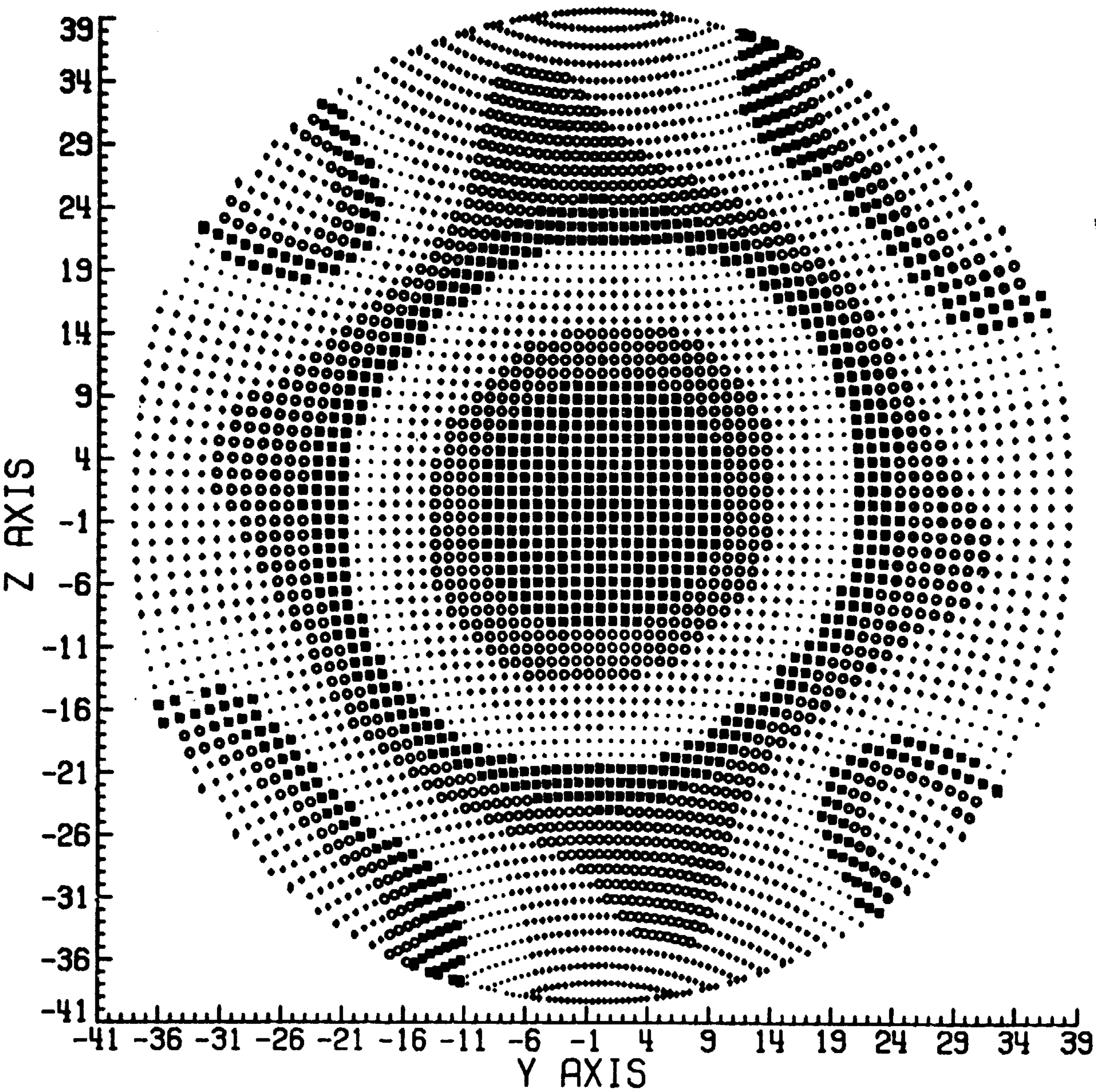


Figure 13 -- Points of Intersection of the Reflected Rays and the Aperture Plane Showing the Contours of Constant Electric Field Amplitude

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 $\pm 20.4^{\circ}$ angle at the feed. However, as pointed out in

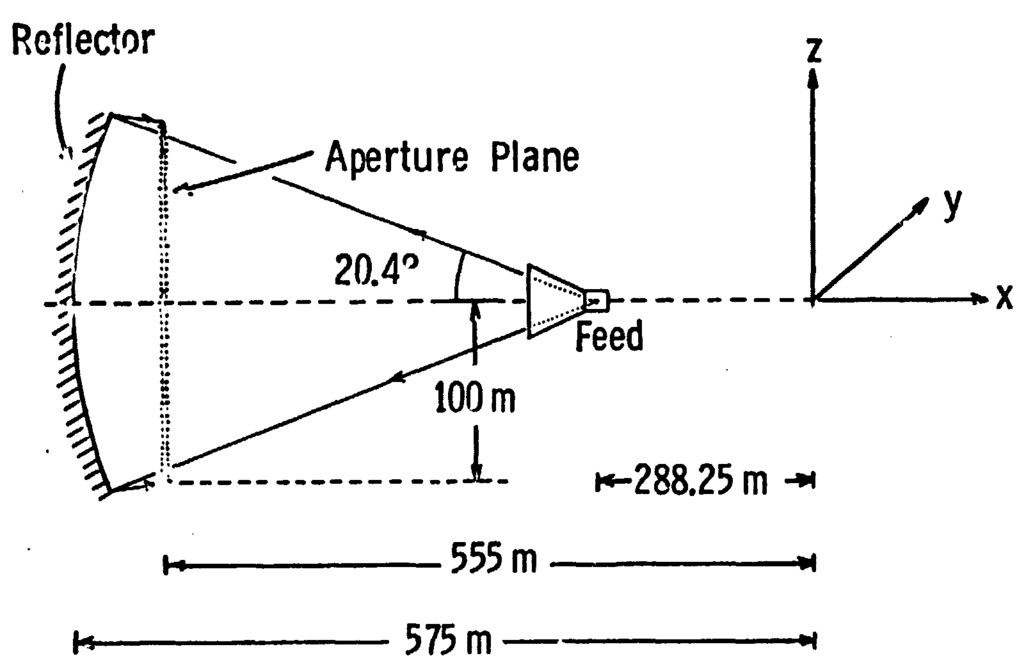


Figure 14 -- Dimensions of the Reflector Antenna

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 $\pm 23^{\circ}$ 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.

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

```
      SUBROUTINE FILL(P,NTX,NPX)
      COMMON/PATTERN/PHI(3),THETA(3)
      COMMON/MATH/PI,PI2,PID2,DTOR,RTOD
      DIMENSION P(5,NTX,NPX)
      DIMENSION E(181),E2(24),IP(47)
6320  FORMAT (5F15.10)
      CALL SETM(0.0,P,5*NPX*NTX)
      READ 6320, (E(I),I=1,91)
      DO 10 I=1,24
      E2(25-I)=E(I)
10    CONTINUE
      DO 50 I=1,24
      II=48-I
      DO 50 J=1,24
      JJ=48-J
      P(1,I,J)=E2(I)+E2(J)
      P(1,II,JJ)=P(1,I,J)
      P(1,I,JJ)= 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.      ] Two title cards
HORN RADIUS=3.333 WAVELENGTHS, HALF FLARE ANGLE=3.43626 DEGREES.      ]
575.0,0.3,5.26,0.0,1.0,7932.0                                           General information
288.25,0.0,0.0,0.0,0.0,0.0                                              Feed information
-555.0,0.0,0.0,100.0,100.0,0.0,0.0.1.0,1.0                              Aperture Plane information
157.0,203.0,1.0,67.0,113.0,1.0                                          Illumination information
  0.0000000000  -.0933062610  -.3744782665  -.8473717300 -1.5187488578 }
 -2.3989163877 -3.5023243262 -4.8495610232 -6.4692989190 -8.4020586898 }
-10.7058765293 -13.4639153550 -16.7860023614 -20.7386062010 -24.8060318905 }
-26.6841075952 -25.9645996039 -25.1538986040 -25.0729156808 -25.7624723335 }
-27.2127240439 -29.4989308715 -32.8387085572 -37.5604573045 -41.8187709455 }
-39.6384702782 -36.8164737944 -35.3829838711 -35.0174676928 -35.5037538892 }
-36.7839991881 -38.9419151117 -42.2633214025 -47.3633943624 -52.2702735560 }
-48.2908042132 -44.7089002959 -42.8180233576 -42.0050126938 -41.9759487908 }  Feed pattern in dBs
-42.6135831782 -43.9082194933 -45.9549311241 -49.0182091172 -53.7162102189 }  between 0 and 90°
-59.9126567715 -56.9082402291 -52.4163366242 -49.8505109856 -48.4162153049 }
-47.7167195387 -47.5645739480 -47.8686624429 -48.5925326441 -49.7411045564 }
-51.3640790133 -53.5786384645 -56.6299013220 -61.0136930406 -66.6159294262 }
-64.9974338927 -60.4873691442 -57.5871074564 -55.7257619923 -54.5064228902 }
-53.7228787013 -53.2584459711 -53.0411785183 -53.0235020974 -53.1722001353 }
-53.4630652514 -53.8777346647 -54.4018889361 -55.0239798056 -55.7343803505 }
-56.5248620044 -57.3875697938 -58.3152108538 -59.3000667376 -60.3334972462 }
-61.4051425563 -62.5024699965 -63.6093625895 -64.7056919487 -65.7668025859 }
-66.7637085843 -67.6650656507 -68.4390877790 -69.0574252981 -69.4981366095 }
-69.6483956986                                                           }
PHI  0.0       MAX-TO.0       0.825    0.025                            Pattern request
```

OUTPUT:

```
                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.........................    .3000
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...........   5.2600
FEED SHADOW CENTER COORDINATES IN APERTURE PL......   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    -.00    -.00
THE REFLECTED BEAM MAX IS.................THETA=  90.00
                     .....................PHI=   -.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 ---------------
```

PHI 0.000 THETA 90.000 90.825 .025

THETA	DBZ/Z	DBY/Z	DBZ/Y	DBY/Y	PWRDB
90.000	0.00000	-119.46398	119.46398	0.00000	0.00000
90.025	-.33085	-119.50804	119.13313	-.04406	-.33085
90.050	-1.32658	-119.55315	118.13740	-.08916	-1.32650
90.075	-3.00119	-119.59931	116.46280	-.13533	-3.00119
90.100	-5.39134	-119.64654	114.07264	-.18255	-5.39134
90.125	-8.57678	-119.69483	110.88720	-.23084	-8.57670
90.150	-12.70717	-119.74418	106.75681	-.28020	-12.70717
90.175	-18.03187	-119.79461	101.43211	-.33063	-18.03107
90.200	-24.89860	-119.84612	94.56538	-.38214	-24.89050
90.225	-32.87010	-119.89870	86.59388	-.43472	-32.87010
90.250	-35.15854	-119.95237	84.30544	-.48839	-35.15054
90.275	-33.68027	-120.00713	85.78371	-.54315	-33.60027
90.300	-34.02303	-120.06298	85.44095	-.59900	-34.02303
90.325	-36.71076	-120.11993	82.75323	-.65595	-36.71076
90.350	-39.20057	-120.17798	80.26341	-.71400	-39.20057
90.375	-38.18636	-120.23714	81.27762	-.77316	-38.18636
90.400	-39.29269	-120.29741	80.17129	-.83343	-39.29269
90.425	-47.60922	-120.35879	71.85477	-.89481	-47.60921
90.450	-47.01386	-120.42130	72.45012	-.95731	-47.01306
90.475	-39.59706	-120.48493	79.86693	-1.02094	-39.59706
90.500	-38.48771	-120.54969	80.97628	-1.08570	-38.40771
90.525	-39.43915	-120.61558	80.02484	-1.15160	-39.43915
90.550	-39.94135	-120.68262	79.52263	-1.21864	-39.94135
90.575	-40.67842	-120.75080	78.78556	-1.28682	-40.67842
90.600	-44.38868	-120.82013	75.07530	-1.35615	-44.38050
90.625	-56.31351	-120.89062	63.15047	-1.42664	-56.31351
90.650	-48.55359	-120.96227	70.91039	-1.49829	-48.55359
90.675	-44.25915	-121.03509	75.20484	-1.57111	-44.25915
90.700	-42.82569	-121.10908	76.63829	-1.64510	-42.82569
90.725	-41.96038	-121.18425	77.50360	-1.72027	-41.96030
90.750	-42.16232	-121.26060	77.30166	-1.79662	-42.16232
90.775	-44.29341	-121.33814	75.17057	-1.87416	-44.29341
90.800	-47.71295	-121.41688	71.75103	-1.95289	-47.71295
90.825	-48.07972	-121.49681	71.38426	-2.03283	-48.07972

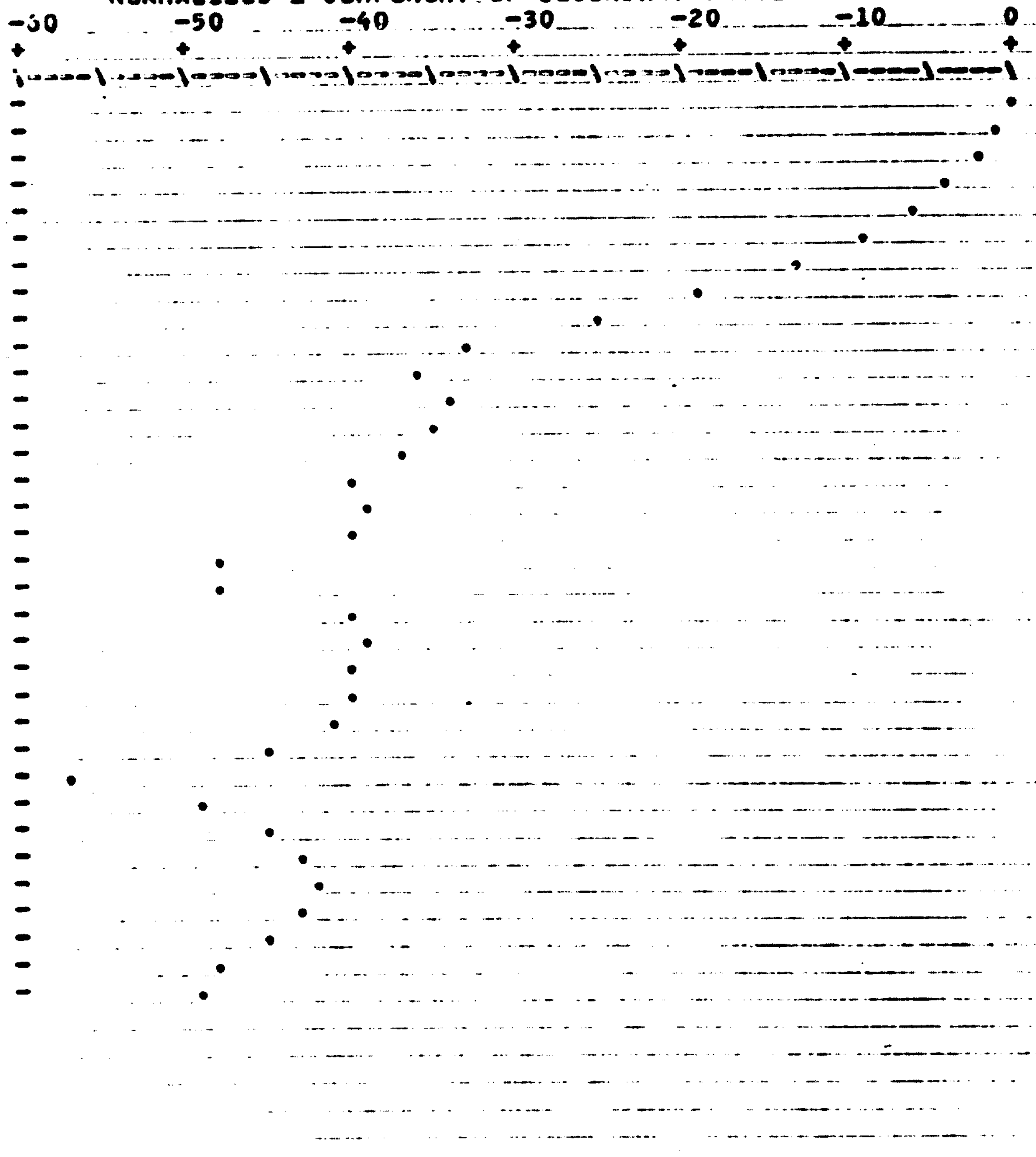
20LOG(MAX(FIELD-Z))=20LOG(22.19400)= 26.92471

20LOG(MAX(FIELD-Y))=20LOG(.2360676E-04)= -92.53927

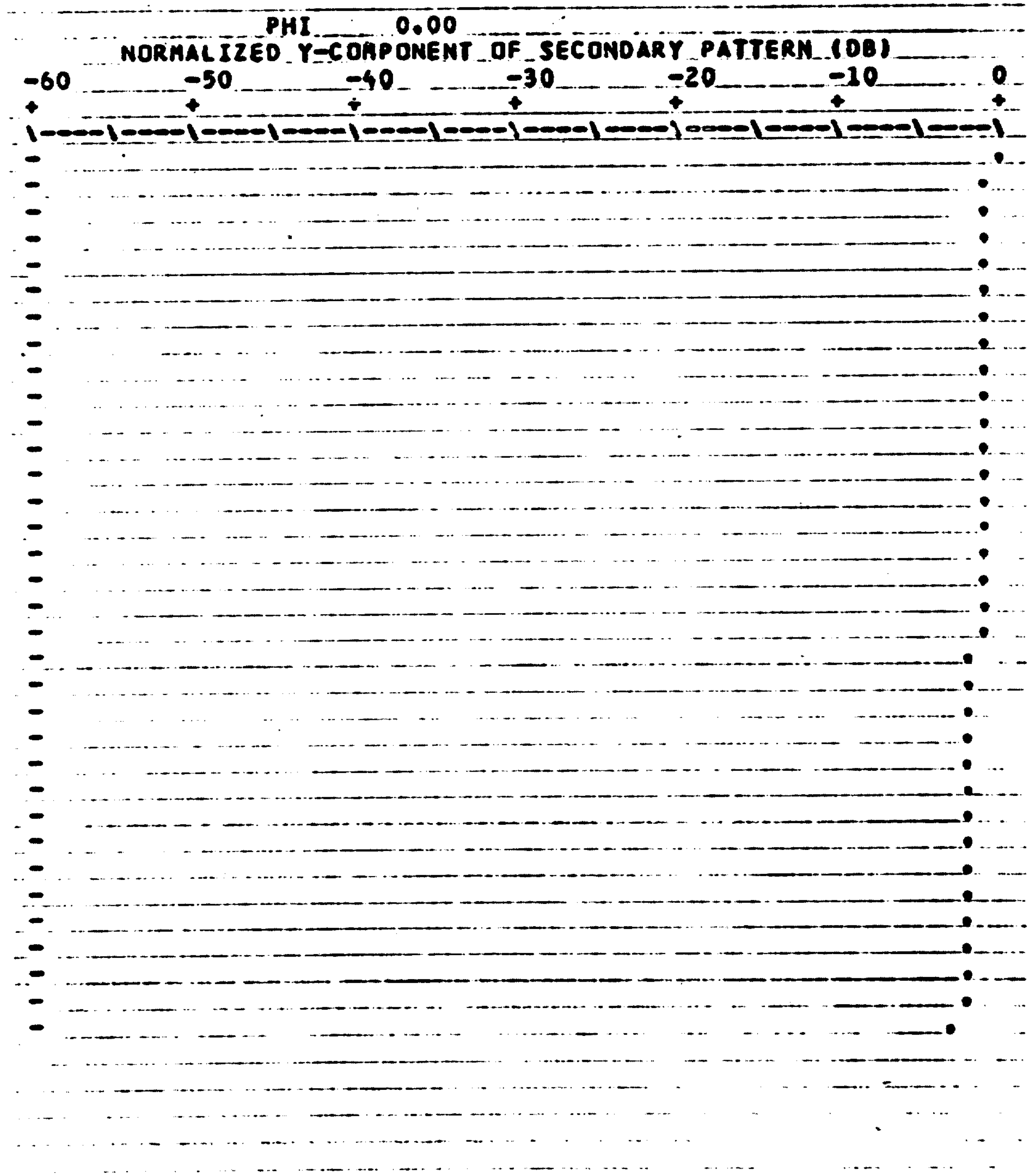
INTERPOLATION NUMBER USED FOR INTEGRATION IS.......... 7

PHI 0.00

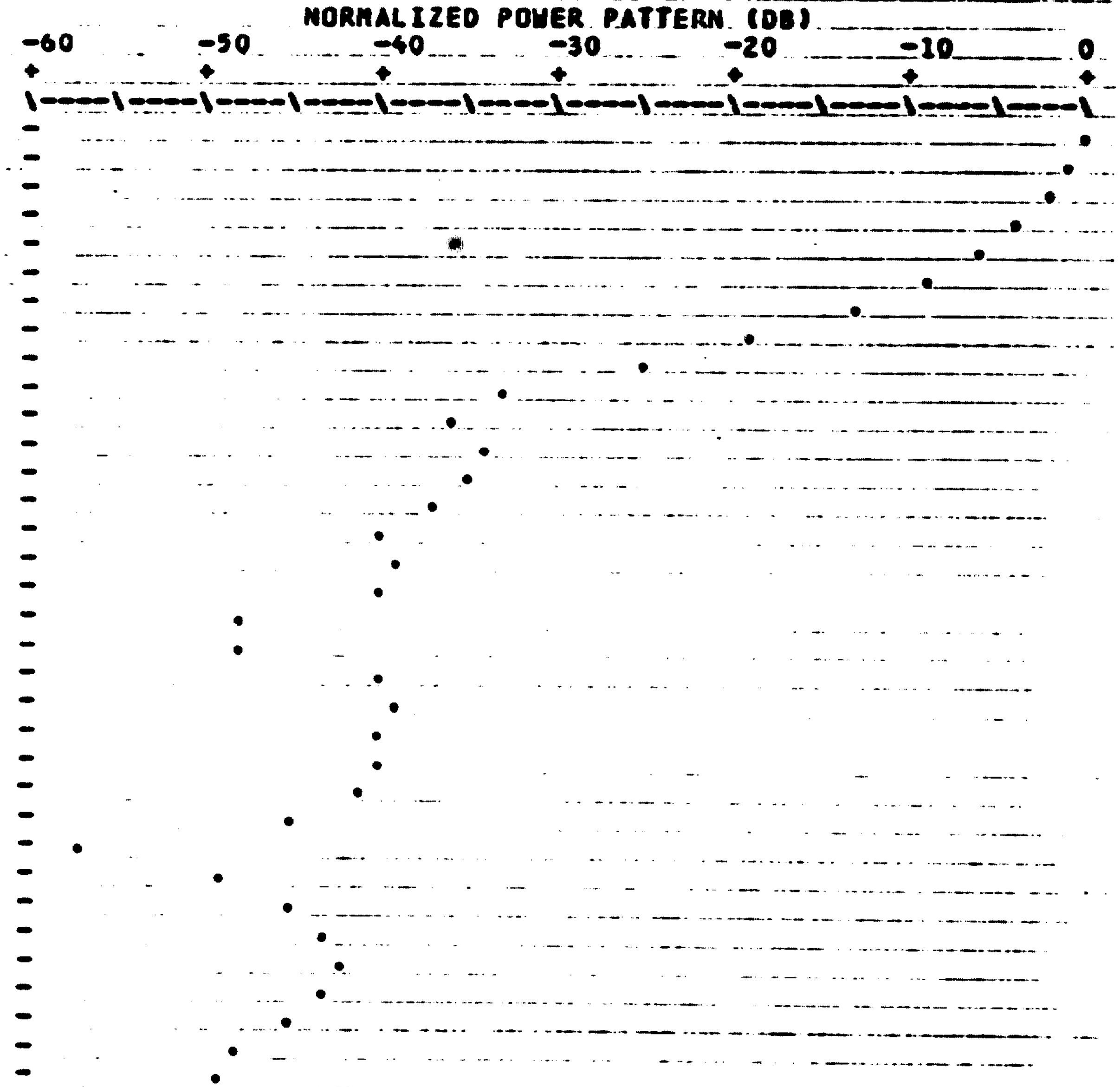
NORMALIZED Z-COMPONENT OF SECONDARY PATTERN (DB)



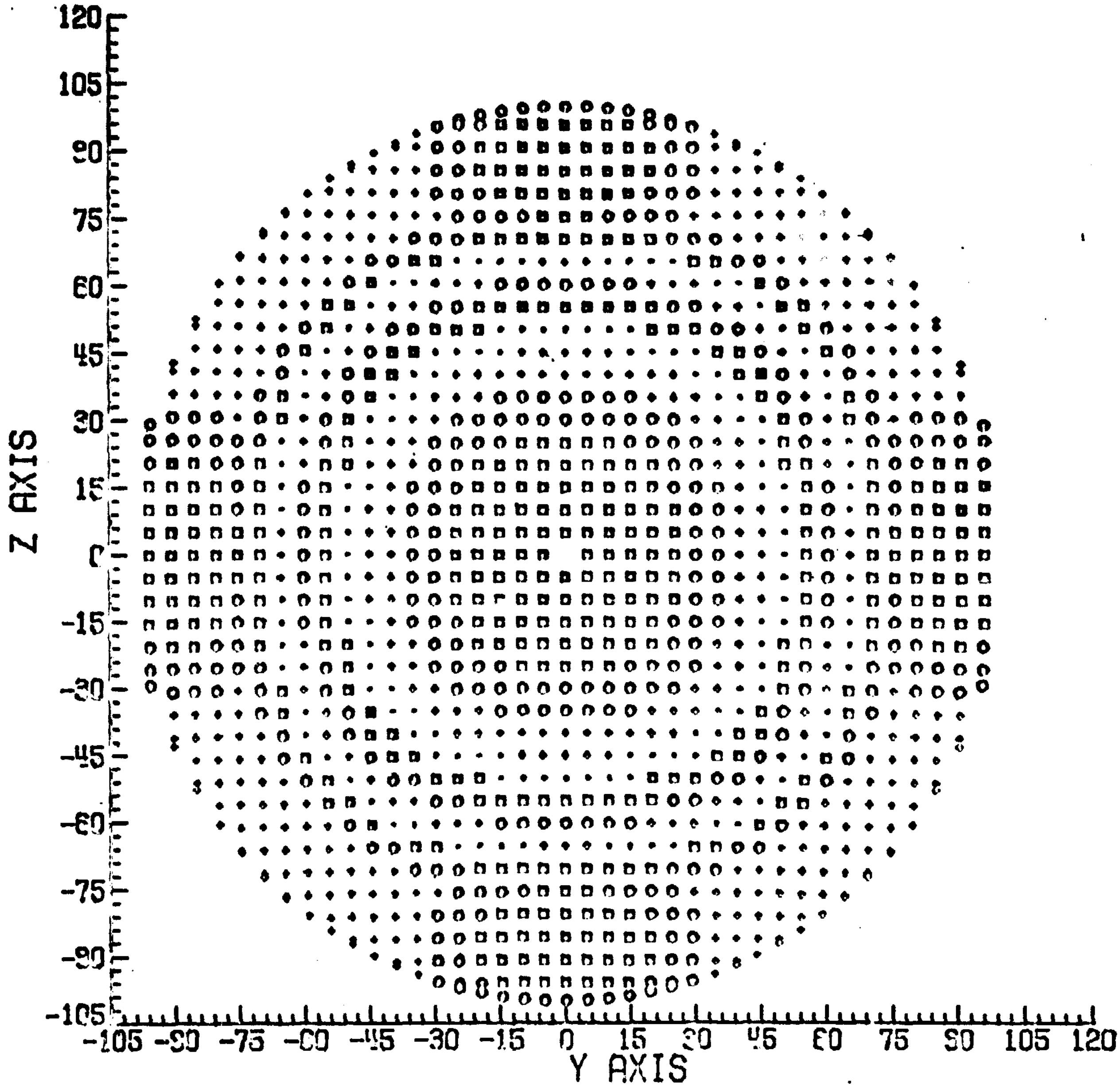
ORIGINAL PAGE IS
OF POOR QUALITY

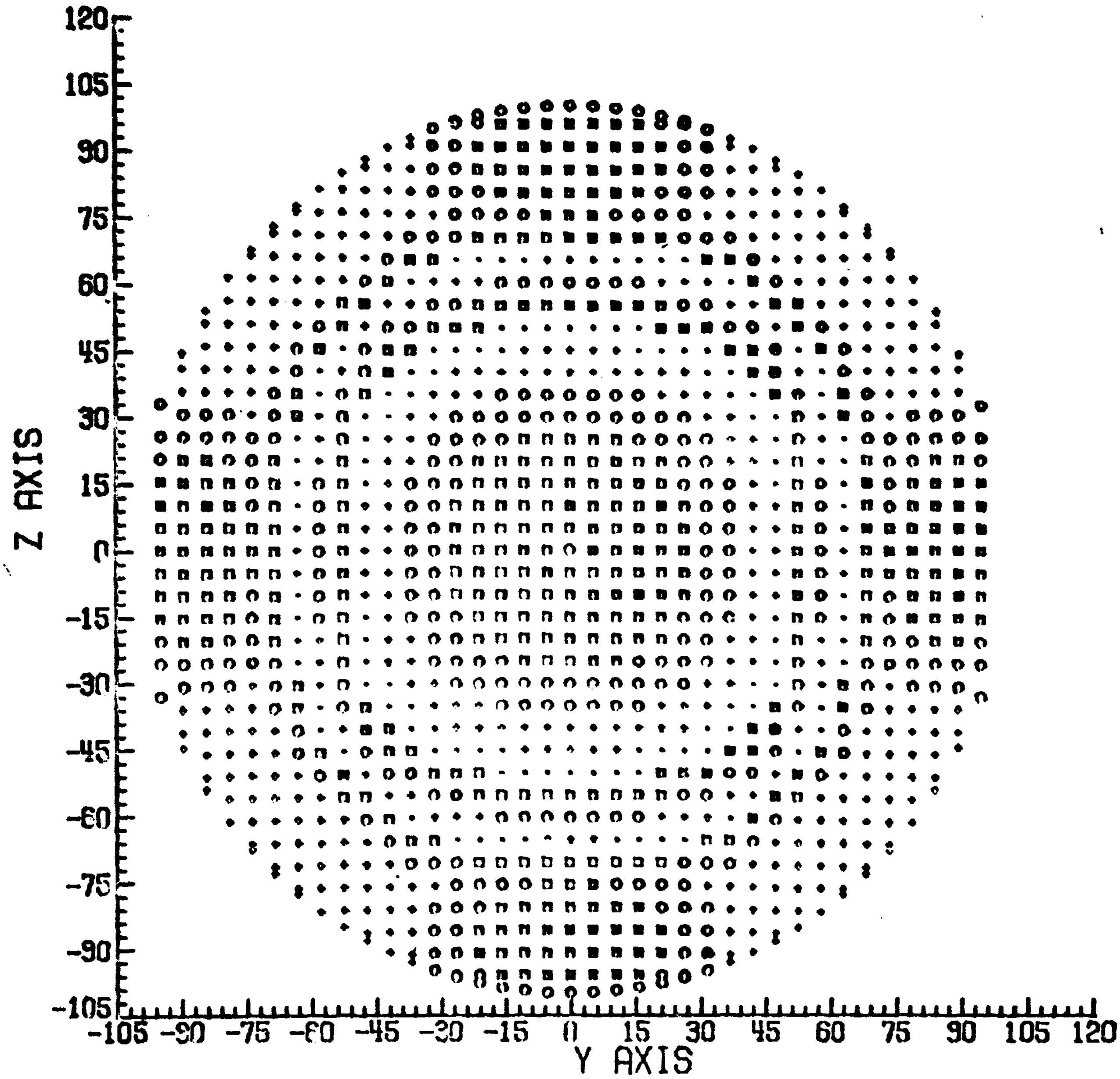

PHI 0.00
NORMALIZED Y-COMPONENT OF SECONDARY PATTERN (DB)
-60
-50
-40
-30
-20
-10
0

PHI 0.00



---------------- FINISHED RADPAT ----------------


Z AXIS
Y AXIS
120
105
90
75
60
45
30
15
0
-15
-30
-45
-60
-75
-90
-105
-105 -90 -75 -60 -45 -30 -15 0 15 30 45 60 75 90 105 120


Z AXIS
Y AXIS
120
105
90
75
60
45
30
15
0
-15
-30
-45
-60
-75
-90
-105
-105 -90 -75 -60 -45 -30 -15 0 15 30 45 60 75 90 105 120

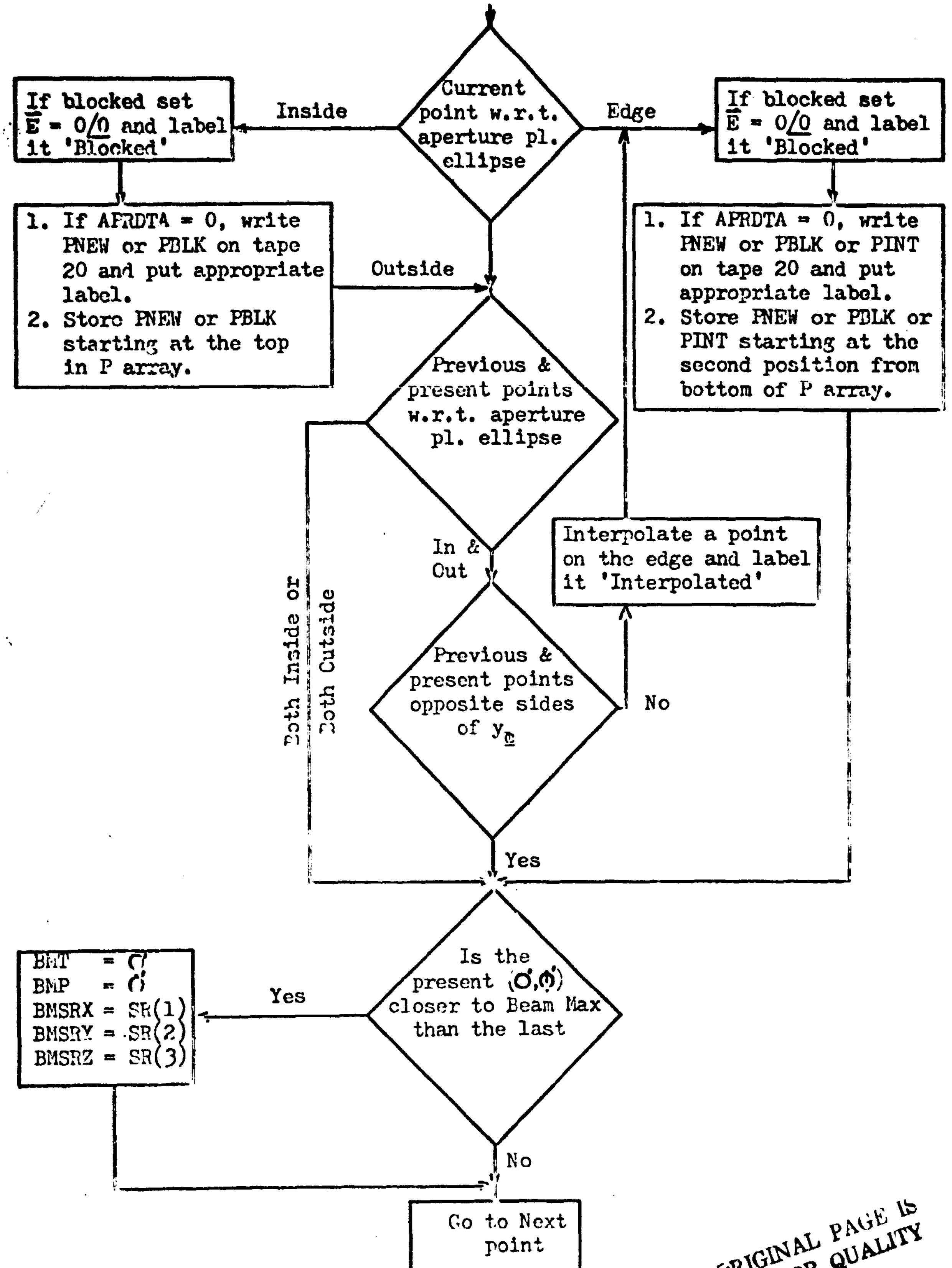
6. SUBROUTINE APERTUR

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 θ' and ϕ', using the equation of the reflector, the following are calculated:

R	Distance along the ray between the feed and the reflector,
(XO,YO,ZO)	Point on the reflector where the incident ray strikes,
NHAT	A one-dimensional array containing three components of the unit normal vector at (XO,YO,ZO),
SR	A one-dimensional array containing three components of a vector along the reflected ray,
ER	A one-dimensional array containing three components of $\bar{E}$ 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	$(R+D)2\pi/\lambda$ + 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.



ORIGINAL PAGE IS OF POOR QUALITY

Figure 16 -- Partial Flow Chart of Subroutine APERTUR

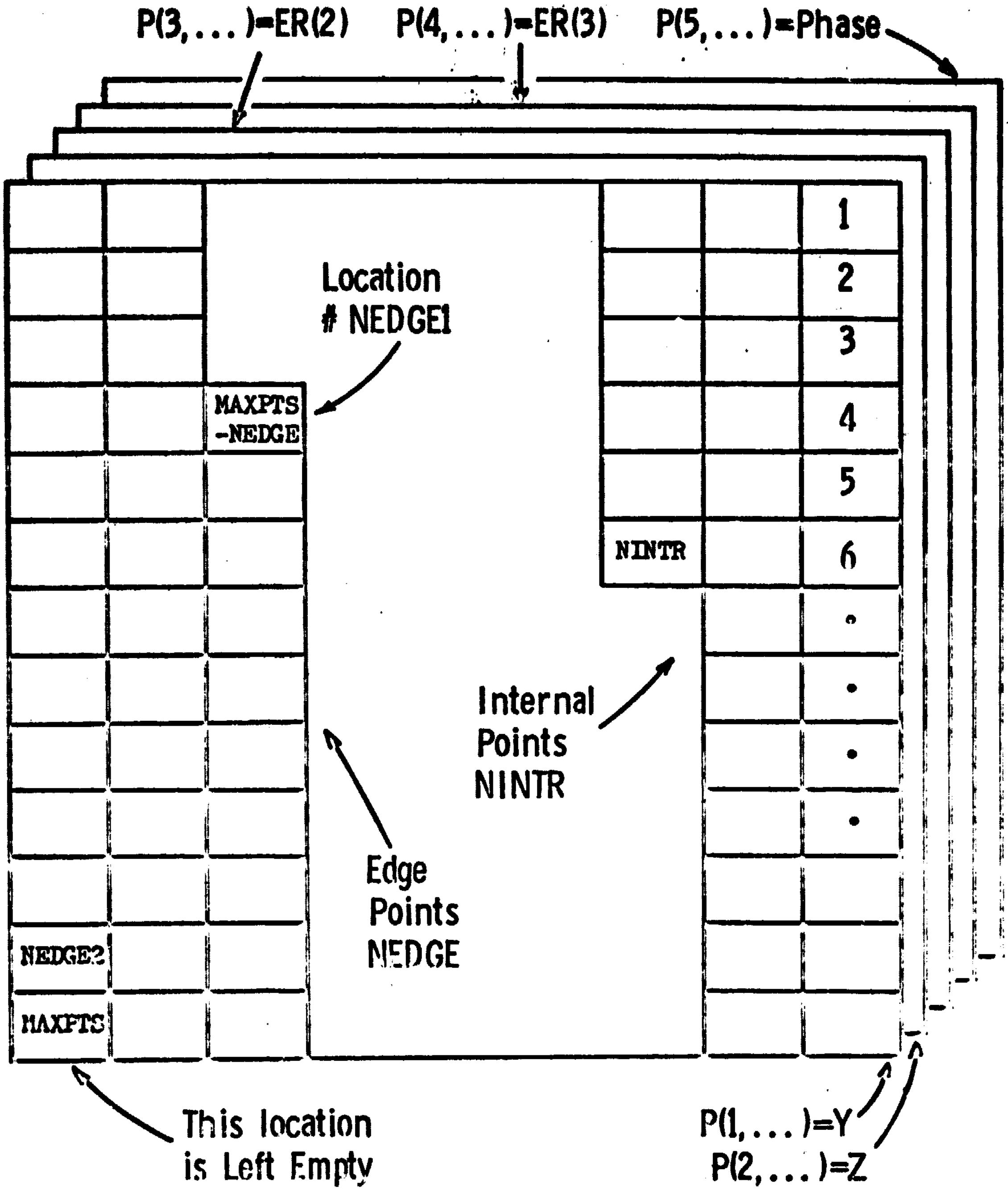


Figure 17 -- P Array Showing Locations of Internal and Edge Points

```
      SUBROUTINE APERTUR(P,NTX,NPX)
      COMMON/PARAMS/TITLE(16),AORORF,XLAM,GRID,SURFACE,APRDTA,FEED(3),
     .               ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAEX,HFMIEX,BMTP,BMPP,
     .               NT,NP,NPOINT,MAXPTS,BELLP
      COMMON/BLOCKG/YCBL,ZCBL,HFMABL,HFMIBL
      COMMON/MATH/PI,PI2,PID2,DTOR,RTOD
      COMMON/POINTS/NEDGE,NINTR
      REAL NHAT
      INTEGER SEDGE
      DIMENSION P(5,NTX,NPX),POLD(5),PNEW(5),PINT(5),PBLK(5),A(3,3),
     .          B(3,2),BB(3,2),NHAT(3),C(3),SR(3),EI(3),ER(3)
      ALPHAR=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(APRDTA.GT.0.0) WRITE(20,110)                                  TAPE 20
110   FORMAT(1H1/,10H**********,7X,*THETA*,9X,*Y*,9X,*Z*,7X,*ERY*,7X,
     .       *ERZ*,5X,*PHASE*,9X,*R*)
      BMTEST=1.0E+40
      NINTR=NEDGE=0
      PBLK(3)=PBLK(4)=PBLK(5)=0.0
      DO 5000 IP=1,NP
      DEGPHI=P(5,1,IP)*RTOD
      IF(APRDTA.GT.0.0) WRITE(20,120) DEGPHI                           TAPE 20
120   FORMAT(1X,*PHI=*,F10.4)
      DO 4000 IT=1,NT
      DEGTHET=P(4,IT,IP)*RTOD
      SINP=SIN(P(5,IT,IP))
      COSP=COS(P(5,IT,IP))
      SINT=SIN(P(4,IT,IP))
      COST=COS(P(4,IT,IP))
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      BB(1,1)=SINT*COSP
      BB(2,1)=SINT*SINP
      BB(3,1)=COST
      BB(1,2)=+FEED(1)
      BB(2,2)=+FEED(2)
      BB(3,2)=+FEED(3)
      CALL MULT32(B,A,BB)
      IF (SURFACE) 122,124,126                                   ********
122   AR=B(1,1)**2/AORORF**2+(B(2,1)**2+B(3,1)**2)/BELLP**2
      BR=-2.0*(B(1,1)*B(1,2)/AORORF**2+(B(2,1)*B(2,2)+B(3,1)*B(3,2))/
                              BELLP**2)
      CR=B(1,2)**2/AORORF**2+(B(2,2)**2+B(3,2)**2)/BELLP**2-1.0
      GO TO 128
124   AR=B(1,1)*B(1,1)+B(2,1)*B(2,1)+B(3,1)*B(3,1)
      BR=-2.*(B(1,1)*B(1,2)+B(2,1)*B(2,2)+B(3,1)*B(3,2))
      CR=B(1,2)*B(1,2)+B(2,2)*B(2,2)+B(3,2)*B(3,2)-AORORF*AORORF
      GO TO 128
126   AR=B(2,1)*B(2,1)+B(3,1)*B(3,1)
      BR=-2.0*(B(2,1)*B(2,2)+B(3,1)*B(3,2)+2.0*AORORF*B(1,1))
      CR=B(2,2)*B(2,2)+B(3,2)*B(3,2)+4.0*AORORF*B(1,2)-4.0*AORORF**2
128   IF (AR.LT.1.0E-10) R=-CR/BR
      IF (AR.LT.1.0E-10) GO TO 130
      R=(-BR+SQRT(BR*BR-4.*AR*CR))/(AR+AR)
130   CONTINUE
      XO=B(1,1)*R-B(1,2)
      YO=B(2,1)*R-B(2,2)
      ZO=B(3,1)*R-B(3,2)
      IF (SURFACE) 132,134,136                                   ********
132   NHAT(1)=-XO*BELLP **2/SQRT(XO**2*BELLP**4+(YO**2+ZO**2)*AORORF**4)
      NHAT(2)=-YO*AORORF**2/SQRT(XO**2*BELLP**4+(YO**2+ZO**2)*AORORF**4)
      NHAT(3)=-ZO*AORORF**2/SQRT(XO**2*BELLP**4+(YO**2+ZO**2)*AORORF**4)
      GO TO 138
134   NHAT(1)=-XO/AORORF
      NHAT(2)=-YO/AORORF
      NHAT(3)=-ZO/AORORF
      GO TO 138
136   NHAT(1)=2.0*AORORF/SQRT(4.0*AORORF**2+YO**2+ZO**2)
      NHAT(2)=        -YO/SQRT(4.0*AORORF**2+YO**2+ZO**2)
      NHAT(3)=        -ZO/SQRT(4.0*AORORF**2+YO**2+ZO**2)
```

```
138    SCALAR=2.0*(B(1,1)*NHAT(1)+B(2,1)*NHAT(2)+B(3,1)*NHAT(3))
       DO 1500 I=1,3
1500   SR(I)=B(I,1)-SCALAR*NHAT(I)
       ETI=P(1,IT,IP)/R
       EPI=P(2,IT,IP)/R
       C(1)=COST*COSP*ETI-SINP*EPI
       C(2)=COST*SINP*ETI+COSP*EPI
       C(3)=-SINT*ETI
       DO 2000 I=1,3
       EI(I)=0.0
       DO 2000 J=1,3
2000   EI(I)=EI(I)+A(I,J)*C(J)
       SCALAR=2.0*(EI(1)*NHAT(1)+EI(2)*NHAT(2)+EI(3)*NHAT(3))
       DO 2500 I=1,3
2500   ER(I)=SCALAR*NHAT(I)-EI(I)
       Y=YO+(XC-XO)*SR(2)/SR(1)
       Z=ZO+(XC-XO)*SR(3)/SR(1)
       D=SQRT((XC-XO)*(XC-XO)+(Y-YO)*(Y-YO)+(Z-ZO)*(Z-ZO))
       PHASE=PI2*(R+D)/XLAM+P(3,IT,IP)
       PNEW(1)=PBLK(1)=Y
       PNEW(2)=PBLK(2)=Z
       PNEW(3)=ER(2)
       PNEW(4)=ER(3)
       PNEW(5)=PHASE
       TEST=  HFMAEX*HFMAEX*HFMIEX*HFMIEX-HFMAEX*HFMAEX*(Z-ZC)*(Z-ZC)
      .                                 -HFMIEX*HFMIEX*(Y-YC)*(Y-YC)
       TESTBL=HFMABL*HFMABL*HFMIBL*HFMIBL-HFMABL*HFMABL*(Z-ZCBL)*(Z-ZCBL)
      .                                 -HFMIBL*HFMIBL*(Y-YCBL)*(Y-YCBL)
       IF (TEST) 2701,2501,2601
2501   NEDGE=NEDGE+1                                                   EDGE
       SEDGE=MAXPTS-NEDGE
       IF (TESTBL.LT.0.0) GO TO 2510
       CALL MOVEM(PBLK,P(1,SEDGE),5)
       IF(APRDTA.GT.0.0) WRITE(20,2505) PBLK                           TAPE 20
2505   FORMAT(1X,*$*,23X,2F10.4,2F10.7,F10.4,12X,*EDGE POINT, BLOCKED*)
       GO TO 2515
2510   CALL MOVEM(PNEW,P(1,SEDGE),5)
       IF(APRDTA.GT.0.0) WRITE(20,2512) PNEW                           TAPE 20
2512   FORMAT(1X,*$*,23X,2F10.4,2F10.7,F10.4,12X,*EDGE POINT*)
```

```
2515  CONTINUE
      GO TO 2800
2601  NINTR=NINTR+1                                                       INSIDE
      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                     TAPE 20
2605  FORMAT(1X,*$*,13X,3F10.4,2F10.7,2F10.4,14X,*BLOCKED*)
      GO TO 2615
2610  CALL MOVEM(PNEW,P(1,NINTR),5)
      IF(APRDTA.GT.0.0) WRITE(20,2612) DEGTHET,PNEW,R                     TAPE 20
2612  FORMAT(1X,*$*,13X,3F10.4,2F10.7,2F10.4)
2615  CONTINUE
2701  IF (IT.EQ.1) GO TO 2800                                             OUTSIDE
      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)
      TESTBL=HFMABL*HFMABL*HFMIBL*HFMIBL-HFMABL*HFMABL*(Z-ZCBL)*(Z-ZCBL)
     .                                   -HFMIBL*HFMIBL*(Y-YCBL)*(Y-YCBL)
      IF (TESTBL.LT.0.0) GO TO 2710
      CALL MOVEM(PBLK,P(1,SEDGE),5)
      IF(APRDTA.GT.0.0) WRITE(20,2705) PBLK                               TAPE 20
2705  FORMAT(1X,*$*,23X,2F10.4,2F10.7,F10.4,10X,*INTERPOLATED, BLOCKED*)
      GO TO 2715
2710  CALL MOVEM(PINT,P(1,SEDGE),5)
      IF(APRDTA.GT.0.0) WRITE(20,2712) PINT                               TAPE 20
2712  FORMAT(1X,*$*,23X,2F10.4,2F10.7,F10.4,10X,*INTERPOLATED*)
2715  CONTINUE
2800  CALL MOVEM(PNEW,POLD,5)
      TESTO=TEST
      TEST=(DEGTHET-90.0)**2+(DEGPHI-180.0)**2
      IF (TEST-BMTEST) 2980,3000,3000
2980  BMTEST=TEST
```

```
      BMT=DEGTHET
      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)
      COSBMTP=BMSRZ/SQRT(BMSRX**2+BMSRY**2+BMSRZ**2)
      SINBMPP=BMSRY/SQRT(BMSRX**2+BMSRY**2)
      BMTP=RTOD*ACOS(COSBMTP)
      BMPP=RTOD*ASIN(SINBMPP)
      PRINT 5050, BMT,BMP,BMSRX,BMSRY,BMSRZ,BMTP,BMPP
5050  FORMAT(* THE FEED MAX IS...............................THETA=*F7.2/
     .        *                ...............................PHI=*F7.2/
     .        * THE REFLECTED RAY VECTOR (X,Y,Z).....................*3F7.2
     .      /* THE REFLECTED BEAM MAX IS.....................THETA=*F7.2/
     .        *                           .....................PHI=*F7.2)
      RETURN
      END
```

7. SUBROUTINE QUANTIZ

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.

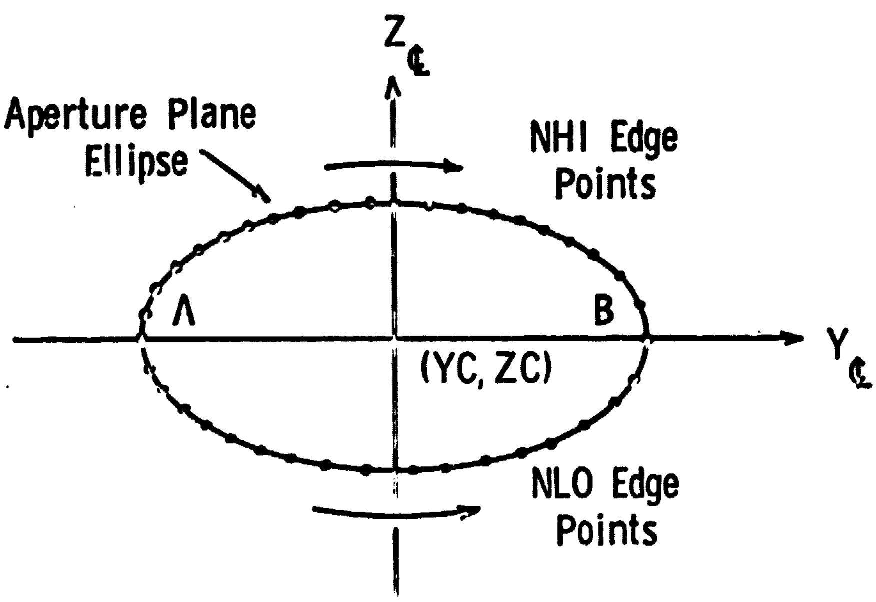


Figure 18 -- Edge Points

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 $Y_{℄}$) and NHI (the number of edge points on and above $Y_{℄}$) are determined.

Now considering P as an one-dimensional array, the data associated with the NLO edge points are stored in the first 5*NLO locations of the P array, and the data associated with the NHI edge points are stored in the next 5*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*(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

ORIGINAL PAGE IS
OF POOR QUALITY

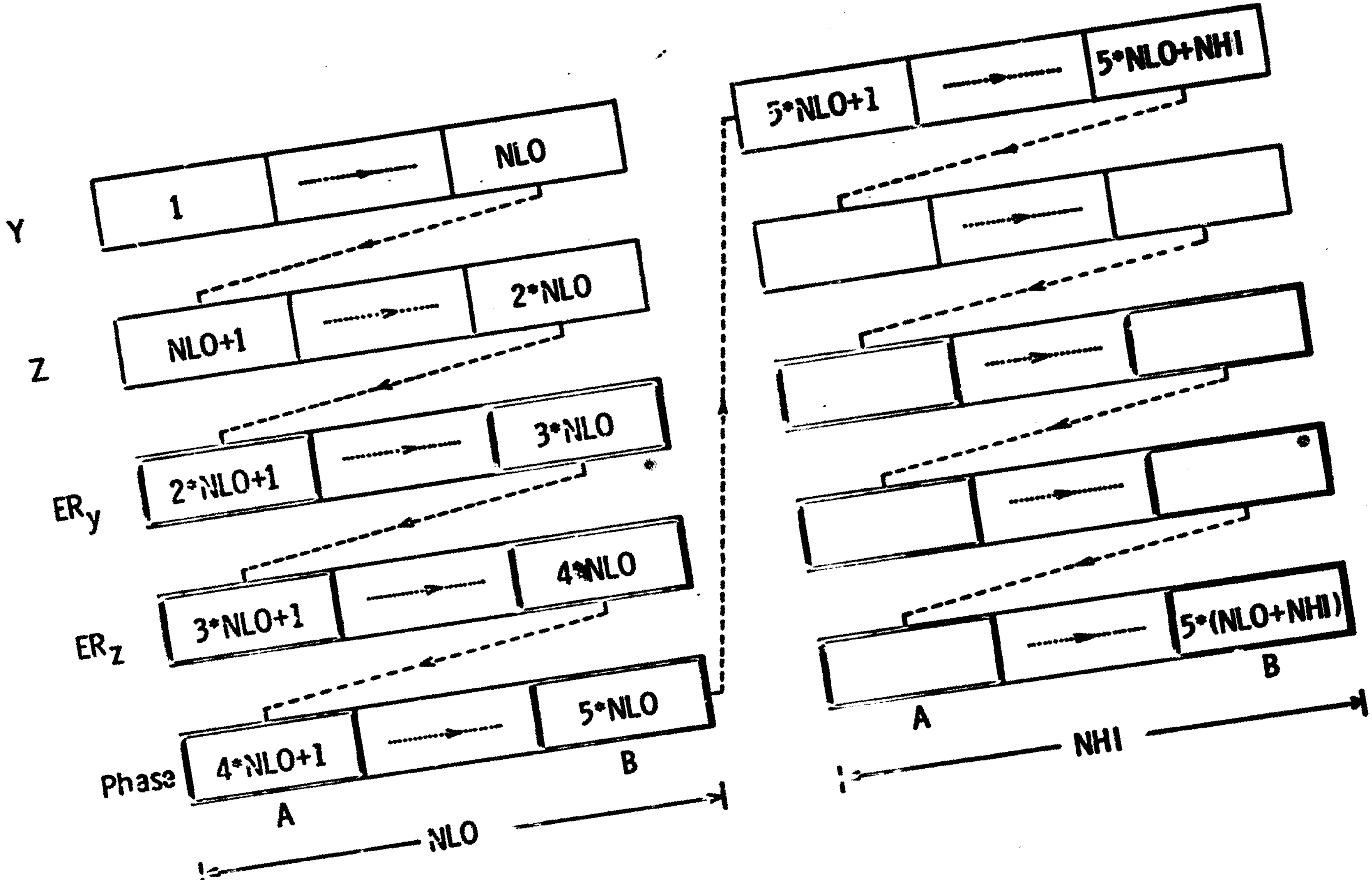


Figure 19 -- P Array Laid Out in One Dimensional Form

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*2*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*2*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,XLAM,GRID,SURFACE,APRDTA,FEED(3),
     .               ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAEX,HFMIEX,BMTP,BMPP,
     .               NT,NP,NPOINT,MAXPTS,BELLP
      COMMON/POINTS/NEDGE,NINTR
      DIMENSION P(5,NTXNPX)
      DATA SIGMA/1.0/
      REWIND 8                                                          TAPE 8
      WRITE(8) ((P(I,J),I=1,5),J=1,NINTR)                               TAPE 8
      NEDGE2=MAXPTS-1
      NEDGE1=MAXPTS-NEDGE
      CALL FQKSORT(P(1,NEDGE1),5,NEDGE)
      INLO=INHI=0
      YMIN=YC-HFMAEX
      YMAX=YC+HFMAEX
      IF(P(1,NEDGE1).LT.YMIN) GO TO 7000                                MIN PNT.
      NEDGE1=NEDGE1-1
      NEDGE=NEDGE+1
      P(1,NEDGE1)=YMIN
      P(2,NEDGE1)=ZC
      INLO=1
 7000 CONTINUE
      IF(P(1,NEDGE2).GE.YMAX) GO TO 7100                                MAX PNT.
      NEDGE2=NEDGE2+1
      NEDGE=NEDGE+1
      P(1,NEDGE2)=YMAX
      P(2,NEDGE2)=ZC
      INHI=1
 7100 CONTINUE
      NHI=NLO=0
      DO 7400 I=NEDGE1,NEDGE2
      IF (P(2,I)-ZC) 7350,7360,7370
 7350 NLO=NLO+1
      GO TO 7400
 7360 NLO=NLO+1
 7370 NHI=NHI+1
 7400 CONTINUE
      PRINT 7410, NLO,NHI
 7410 FORMAT(* LOWER CURVE POINTS WITH MAX AND/OR MIN POINTS........*I5/
```

```
.        * UPPER CURVE POINTS WITH MAX AND/OR MIN POINTS........*I5)
      IF (NEDGE1-NLO-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
      ILO=0
      IHI=5*NLO
      DO 7500 I=NEDGE1,NEDGE2
      IF (P(2,I)-ZC) 7450,7460,7470
 7450 ILO=ILO+1
      DO 7451 J=1,5
      P(ILO+(J-1)*NLO)=P(J,I)
 7451 CONTINUE
      GO TO 7500
 7460 ILO=ILO+1
      DO 7461 J=1,5
      P(ILO+(J-1)*NLO)=P(J,I)
 7461 CONTINUE
 7470 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 (INLO.EQ.0) GO TO 7501
      P(J*NLO+1)=P(J*NLO+2)
      P(5*NLO+J*NHI+1)=P(5*NLO+J*NHI+2)
 7501 IF (INHI.EQ.0) GO TO 7505
      P((J+1)*NLO)=P((J+1)*NLO-1)
      P(5*NLO+(J+1)*NHI)=P(5*NLO+(J+1)*NHI-1)
 7505 CONTINUE
      J=HFMAEX/GRID
      BARMIN=YC-J*GRID
      BARMAX=YC+J*GRID
      NBARS=2*J+1
      NOEDGE=2*NBARS
      NPOINT=NINTR+NOEDGE
```

```
      PRINT 7520, BARMIN,BARMAX,NBARS,NOEDGE,NPOINT
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 (NPOINT.LE.MAXPTS) GO TO 8000
      PRINT 7550
7550  FORMAT(* ----- INTERNAL + NEW EDGE POINTS ARE MORE THAN MAXPTS, IN
     .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                                                2 CURVES
      LD=LI+NPTS
      DO 8400 I=2,5                                                 4 THINGS
      SIGMA=-ABS(SIGMA)
      CALL   CURV1(    NPTS,P(LI),P(LD),SLP1,SLPN,P(L1),P(L2),SIGMA)
      BAR=BARMIN        $    K=L
      DO 8300 J=1,NBARS                                             ALL LINS
      P(I,K)=CURV2(BAR,NPTS,P(LI),P(LD),          P(L1),      SIGMA,J)
      P(1,K)=BAR
      BAR=BAR+GRID      $    K=K+1
8300  CONTINUE
      LD=LD+NPTS
8400  CONTINUE
      L=L+NBARS         $    NPTS=NHI       $    LI=LI+5*NLO
8500  CONTINUE
      CALL MOVEM(P(1,LSTART),P,5*NOEDGE)
      REWIND 8                                                      TAPE 8
      READ(8) ((P(I,NOEDGE+J),I=1,5),J=1,NINTR)                     TAPE 8
      ENDFILE 8                                                     TAPE 8
      DO 8600 I=1,NPOINT
      YQ=AINT(ABS(P(1,I))/GRID+0.5)*GRID
      IF(P(1,I).LT.0) YQ=-YQ
      P(1,I)=YQ
8600  CONTINUE
```

```
      CALL FQKSORT(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.....*I5)
      IF(APRDTA.GT.0.0) WRITE(20,9560)                                  TAPE 20
      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(1X,*$*,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 (θ,ϕ). 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_{ϕ} 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 informat n is sorted out. As explained in section 4, a pattern is requested by first choosing a plane defined by θ=constant or ϕ=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 (ϕ or θ, 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.

ORIGINAL PAGE IS
OF POOR QUALITY

Computation of the far field takes place between statements numbered 3450 and 5000. The integrals to be evaluated are

$$E_z = \iint_{\text{aperture}} ER_z \cos\phi \cdot e^{j[kysin\theta sin\phi + kzcos\theta - \Phi]} dydz$$

and

$$E_y = \iint_{\text{aperture}} (ER_y \sin\theta + ER_z \cos\theta \sin\phi)$$

$$\cdot e^{j[kysin\theta sin\phi + kzcos\theta - \Phi]} dydz$$

where ER_y, ER_z, and Φ 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=constant lines and computing ZI and YI which are functions of y,

$$ZI = \int_{\substack{y=\text{const.}\\ \text{line}}} ER_z \cos\phi \; e^{j(kzcos\theta - \Phi)} dz$$

and

$$YI = \int_{\substack{y=\text{const.}\\ \text{line}}} (ER_y \sin\theta + ER_z \cos\theta \sin\phi) e^{j(kzcos\theta - \Phi)} dz$$

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

$$E_z = FLDZ = \int ZI \; e^{jkysin\theta sin\phi} dy$$

and

$$E_y = FLDY = \int YI\ e^{jky\sin\theta\sin\phi} dy.$$

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

After statement number 5000, the computed values of FLDZ 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.

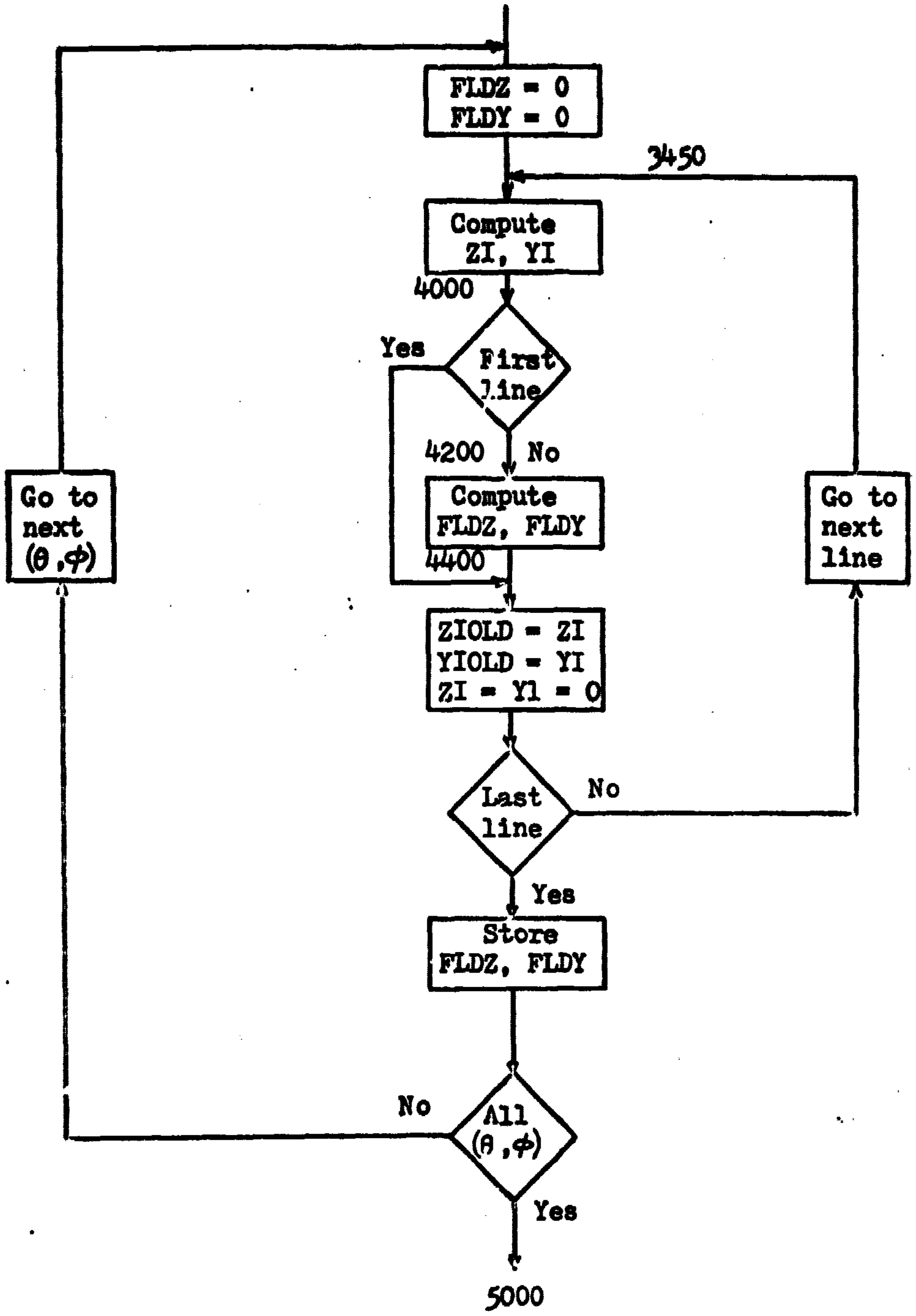


Figure 20 -- Partial Flow Chart of Subroutine RADPAT

```
      SUBROUTINE RADPAT (P,NTXNPX)
      COMMON/PARAMS/TITLE(16),AORORF,XLAM,GRID,SURFACE,APRDTA,FEED(3),
     .              ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAEX,HFMIEX,BMTP,BMPP,
     .              NT,NP,NPOINT,MAXPTS,BELLP
      COMMON/MATH/PI,PI2,PID2,DTOR,RTOD
      LOGICAL LOOPHI
      DIMENSION P(5,NTXNPX),FIELDZ(300),FIELDY(300),PWER(300),AMINOR(3)
      COMPLEX CTEMP,CZ1,CZ2,CY1,CY2,TSZ,TSY,DZI,DYI,ZIOLD,YIOLD,ZI,YI,
     .        FLDZ,FLDY
      SEN=999.0                                                         STARTEND
      NPARTS=7
      RPART=1./NPARTS
      ZLAM=PI2/XLAM
      CALL SETM(SEN,P(1,NPOINT+1),5)
100   CONTINUE
      READ 110, MAJOR,AMAJOR,MINOR,AMINOR                               CUT INFO
110   FORMAT(A5,F10.0,A5,3F10.0)
      IF(EOF(7)) 9000,115                                               ALL CUTS
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) AMINOR(2)=AMINOR(2)+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)+BMTP
      IF (MINOR.EQ.5HMAX-T) MINOR=5HTHETA
160   CONTINUE
      DEG=AMAJOR
      DEGR=DEG*DTOR
      DLOR=AMINOR(1)*DTOR
      DHIR=AMINOR(2)*DTOR
      DICR=AMINOR(3)*DTOR
      FMAXZ=FMAXY=-1.E+40
      NTH=0
      LOOPHI=.TRUE.
      D=DLOR
```

```
      IF(MAJOR .NE. 3HPHI) LOOPHI=.FALSE.
      IF (LOOPHI) 400,3400
400   COSP=COS(DEGR)                                        PHI  CUT
      SINP=SIN(DEGR)
      COST=COS(D)
      SINT=SIN(D)
      GO TO 3425
3400  COSP=COS(D)                                           THET CUT
      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=YI=(0.,0.)
3450  CONTINUE
      IF(P(1,IOLD) .NE. P(1,INEW)) GO TO 4000
      Z=P(2,IOLD)                                           BGN A LN
      ERY=P(3,IOLD)
      ERZ=P(4,IOLD)
      PH=P(5,IOLD)
      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*SINT+ERZ*CTSP)*CTEMP
      TSZ=TSZ+CZ1+CZ2
      TSY=TSY+CYI+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                                                  NEXT LIN
      YNEW=P(1,IOLD)
      IF(YOLD.EQ.SEN) GO TO 4400                                STRT LIN
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=YI
      ZI=YI=(0.0,0.0)
      IF(P(1,INEW) .NE. SEN) GO TO 3900                              END LINE
      FIELDZ(NTH)=CABS(FLDZ)
      FIELDY(NTH)=CABS(FLDY)
      FMAXZ=AMAX1(FMAXZ,FIELDZ(NTH))
      FMAXY=AMAX1(FMAXY,FIELDY(NTH))
      D=D+DICR
      IF(D .GT. DHIR) GO TO 5000                                     ALL ANGS
      IF (LOOPHI) 400,3400
5000  CONTINUE
      D=AMINOR(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(1H1,1X,A5,F8.3,5X,A5,3F8.3)
      PRINT 666, MINOR
666   FORMAT(4X,A10,*DBZ/Z*,5X,*DBY/Z*,6X,*DBZ/Y*,6X,*DBY/Y*,6X,*PWRDB*)
      DO 700 I=1,NTH
      DBZ=DBY=PWER(I)=-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) PWER(I)=10.0*ALOG10(
     .   FIELDZ(I)**2+FIELDY(I)**2)
      DBZZ=DBZ-FMZDB
      DBYY=DBY-FMYDB
      DBZY=DBZ-FMYDB
      DBYZ=DBY-FMZDB
      PWRDB=PWER(I)-PWRMDB
      PRINT 690, D,DBZZ,DBYZ,DBZY,DBYY,PWRDB
690   FORMAT(F9.3,5F11.5)
      D=D+AMINOR(3)
      FIELDZ(I)=DBZ                                                  FOR PLOT
      FIELDY(I)=DBY                                                  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(*1*///,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 $Y_{ȼ}$ 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, YEDGE is then taken as the y-coordinate of the interpolated edge point and a

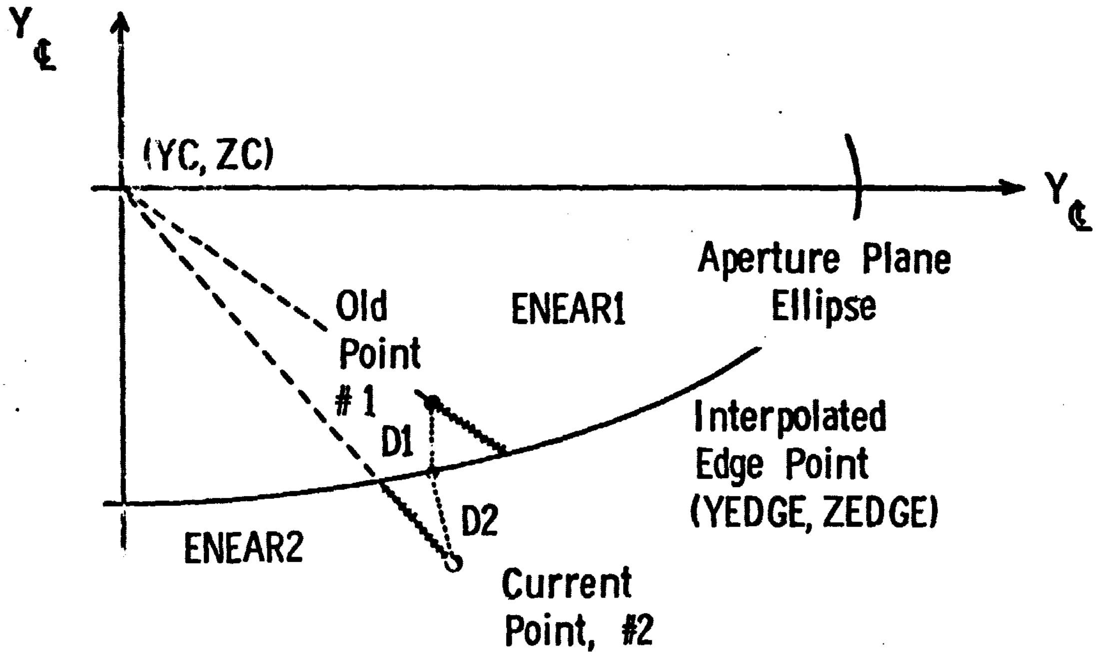


Figure 21 -- Interpolated Point at the Edge of Aperture Plane Ellipse

corresponding value of z-coordinate, ZEDGE is calculated using the equation of the ellipse. Next, calculating the distances D1 and D2, ER_y, ER_z, and Phase are linearly interpolated at the edge point and assigned to PINT(3), PINT(4), and PINT(5), respectively.

```
      SUBROUTINE INTERP(P1,P2,PINT)
      COMMON/PARAMS/TITLE(16),ADRORF,XLAM,GRID,SURFACE,APRDTA,FEED(3),
     .               ALPHA,BETA,GAMMA,XC,YC,ZC,HFMAEX,HFMIEX,BMTP,BMPP,
     .               NT,NP,NPOINT,MAXPTS,RELLP
      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
7     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))
25    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.

```
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/100000B/
      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  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+I
      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
```


Figure 23 -- Subroutine FQKSORT -- Continued

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
      DELX1 = X(2)-X(1)
      DX1 = (Y(2)-Y(1))/DELX1
      IF (SIGMA.LT.0.) GO TO 5
      SLPP1 = SLP1
      SLPPN = SLPN
    1 SIGMAP = ABS(SIGMA)*FLOAT(N-1)/(X(N)-X(1))
      DELS = SIGMAP*DELX1
      EXPS = EXP(DELS)
      SINHS = .5*(EXPS-1./EXPS)
      SINHIN = 1./(DELX1*SINHS)
      DIAG1 = SINHIN*(DELS*.5*(EXPS+1./EXPS)-SINHS)
      DIAGIN = 1./DIAG1
      YP(1) = DIAGIN*(DX1-SLPP1)
      SPDIAG = SINHIN*(SINHS-DELS)
      TEMP(1) = DIAGIN*SPDIAG
      IF (N.EQ.2) GO TO 3
      DO 2 I = 2,NM1
      DELX2 = X(I+1)-X(I)
      DX2 = (Y(I+1)-Y(I))/DELX2
      DELS = SIGMAP*DELX2
      EXPS = EXP(DELS)
      SINHS = .5*(EXPS-1./EXPS)
      SINHIN = 1./(DELX2*SINHS)
      DIAG2 = SINHIN*(DELS*(.5*(EXPS+1./EXPS))-SINHS)
      DIAGIN = 1./(DIAG1+DIAG2-SPDIAG*TEMP(I-1))
      YP(I) = DIAGIN*(DX2-DX1-SPDIAG*YP(I-1))
      SPDIAG = SINHIN*(SINHS-DELS)
      TEMP(I) = DIAGIN*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))
      DO 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 = -DELNN/DELN/DELNM1
      C3 = DELN/DELNN/DELNM1
      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 CURV1 -- 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) 2,2,3
    2 CONTINUE
      I = N
    3 IF (X(I-1) .LE. T .OR. T .LE. X(1)) 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(
     1          I))*DEL1+(Y(I-1)-YP(I-1))*DEL2)/DELS
      I1 = I
      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/0,0,0,6HX,*.*)  /
      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,*+*,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) GO TO 1450
      IFRM(2)=10H*.*)                  $     GO TO 1500
1450  IP1=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

1. Kauffman, J. F.: "Calculation of the Radiation Patterns of Reflector Antennas," N. C. State University Report, May, 1976.
2. Wolff, E. A.: Antenna Analysis, John Wiley and Sons, Inc., New York, 1966, pp. 229-232.
3. Agrawal, P. K.; Kauffman, J. F.; and Croswell, W. F.: "A Technique to Calculate Radiation Patterns of Large Spherical Reflector Antennas," presented at the 1977 IEEE AP-S International Symposium, Stanford University, Stanford, California, June 1977.
4. Agrawal, P. K.: "Optimization of the Design Parameters for a Wide Band Radiometer System," NASA Technical Memorandum No. 78662, January, 1978.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM 78721		

4. Title and Subtitle	5. Report Date
A Computer Program to Calculate Radiation Properties of Reflector Antennas	May 1978
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Pradeep K. Agrawal	
	10. Work Unit No.
9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Langley Research Center Hampton, VA 23665	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics & Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

The author is affiliated with the Joint Institute for Advancement of Flight Science, NASA-Langley Research Center, Hampton, VA 23665

16. Abstract

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.

17. Key Words (Suggested by Author(s))

Reflector Antennas,
Offset Reflector Antennas,
Spherical Reflector Antennas,
Ellipsoidal Reflector Antennas,
Large Reflectors.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price *
Unclassified	Unclassified	83	$5.00

[redacted]