Analysis of a Generalized Dual Reflector Antenna System Using Physical Optics

Roberto J. Acosta and Alan R. Lagin

Lewis Research Center
Cleveland, Ohio

February 1992
ANALYSIS OF A GENERALIZED DUAL REFLECTOR ANTENNA SYSTEM USING PHYSICAL OPTICS

Roberto J. Acosta and Alan R. Lagin
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

ABSTRACT

Reflector antennas are widely used in communication satellite systems because they provide high gain at low cost. Offset-fed single paraboloids and dual reflector offset Cassegrain and Gregorian antennas with multiple focal region feeds provide a simple, blockage-free means of forming multiple, shaped and isolated beams with low sidelobes. Such antennas are applicable to communications satellite frequency reuse systems and earth stations requiring access to several satellites. While the single offset paraboloid has been the most extensively used configuration for the satellite multiple-beam antenna, the trend toward large apertures requiring minimum scanned beam degradation over the field of view 18 degrees for full earth coverage from geostationary orbit may lead to impractically long focal length and large feed arrays. Dual reflector antennas offer packaging advantages and more degrees of design freedom to improve beam scanning and cross-polarization properties. The Cassegrain and Gregorian antennas are the most commonly used dual reflector antennas.

A computer program for calculating the secondary pattern and directivity of a generalized dual reflector antenna system has been developed and implemented at the NASA Lewis Research Center. The theoretical foundation for this program is based on the use of physical optics methodology for describing the induced currents on the sub-reflector and main reflector. The resulting induced currents on the main reflector are integrated to obtained the antenna far-zone electric fields. The computer program is verified with other physical optics programs and with measured antenna patterns. The comparison shows good agreement in far-field sidelobe reproduction and directivity.

INTRODUCTION

The accurate prediction of radiation characteristics for a microwave antenna is essential in designing antenna systems. Antenna radiation characteristics such as beamwidth, gain, aperture efficiency, side-lobe level, and cross polarization are used in analyzing and designing advanced antenna systems. The physical optics-current integration approach (ref. 1) described in this report is one of several method that can be used for predicting antenna performance characteristics. The method assumes that the complex currents in both reflectors are known. This, currents satisfies Maxwell's equations and are used to solve the complex-vector wave equation at any arbitrary observation location. The computation of the induced currents on the main and sub-
reflector are briefly described. A dual reflector configuration (figure 1.) is analyzed, and the results compared with other dual reflector computer programs. A description of the input parameters (user guide) and a copy of the program are included in Appendixes A,B and C.

PHYSICAL OPTICS-CURRENT INTEGRATION APPROACH

DESCRIPTION OF PROBLEM

The geometry of a dual-reflector with a feed at an arbitrary position is shown in Figure 2. Three coordinate systems are shown to define the main reflector, the sub-reflector, and the feed position (or array of feeds). The position and field vectors of these coordinate system can be interrelated by using the Eulerian angles (Figure 3) construction (ref. 2). For instance, the fields of the feed can be expressed in feed coordinates (xf, yf, zf) and then transformed into sub-reflector coordinates (xs, ys, zs) to determine the scattered field from the sub-reflector and then transformed again into main reflector coordinates (xm, ym, zm) to finally obtain the radiated field of the main reflector.

INCIDENT ELECTRIC FIELD ON SUB-REFLECTOR

The radiated electric field from the feed antenna has the asymptotic form given by equation (1):

\[ E(\theta, \phi) = \frac{e^{-jkr}}{r} F(\theta, \phi) \]  

where \( F(\theta, \phi) \) is the element pattern, \( k=2 \pi / \lambda \) is the wavenumber, and \( r \) is the distance from the source (feed) to the sub-reflector point. The vector function in equation (1) can be approximated (ref. 3) by equation (2).

\[ F(\theta, \phi) = \theta U_E(\theta)(a e^{jp} \cos \phi + b \sin \phi) + \phi U_H(\theta)(b \cos \phi + a e^{jp} \sin \phi) \]  

where \( U_E(\theta) \) is the E-plane pattern, \( U_H(\theta) \) is the H-plane pattern, and \( a, b, \) and \( p \) are polarization parameters. The various feed polarization parameters are described in the following table:
TABLE I: Polarization Parameters

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear x</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Linear y</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Right-hand circular</td>
<td>0.707</td>
<td>0.707</td>
<td>+90</td>
</tr>
<tr>
<td>Left-hand circular</td>
<td>0.707</td>
<td>0.707</td>
<td>-90</td>
</tr>
</tbody>
</table>

Typically these elements patterns can be approximated by a cosine to a power function, that is,

\[ \text{UE}(\theta) = \cos^{q_e}(\theta) \]  
\[ \text{UH}(\theta) = \cos^{q_h}(\theta) \]

If equations (3a) and (3b) are used to represent the element pattern, the power radiated (ref 3.) by this source is given by equation (4).

\[ P_{\text{rad}} = \frac{(q_e+q_h+1)}{60(2q_e+1)(2q_h+1)} \]  

**SURFACE CURRENT APPROXIMATION**

The foundations of physical optics (PO), rests on the assumption that the induced current on the reflector surface is given (for a perfect conductor) by

\[ \mathbf{J} = 2 (\mathbf{n} \times \mathbf{H}^{\text{inc}}) \text{ illuminated region} \]
\[ \mathbf{J} = 0 \text{ otherwise} \]

where \( \mathbf{n} \) is the unit normal to the surface and \( \mathbf{H} \) is the incident magnetic field. This incident field may emanate directly from the source or be scattered from the sub-reflector. Although the PO current is an approximation for the true current on the reflector surface, it nevertheless gives very accurate results for predicting far-field patterns of reflectors.
SCATTERED FIELDS FROM SUB-REFLECTOR

For a given point on the sub-reflector \((x_s, y_s, z_s)\) and the feed located at \((x_f, y_f, z_f)\) the incident fields on the sub-reflector are given

\[
E = \frac{e^{-jkx}}{r} F(x_s, y_s, z_s) \tag{5a}
\]

where \(F(x_s, y_s, z_s)\) is the feed pattern, \(k\) the wavenumber and \(r\) the distance from the feed to sub-reflector point. The magnetic field incident on the sub-reflector is given by

\[
H = (r \times E)/Z_0 \tag{5b}
\]

The scattered fields from the sub-reflector are given by (ref.4)

\[
H(x_m, y_m, z_m) = jk \int \int (J \times r_1) \frac{e^{-jk r_1}}{4\pi r_1} \, ds \tag{6a}
\]

\[
E(x_m, y_m, z_m) = -jkZ_0 \int \int (J - (J \cdot r_1)r_1) \frac{e^{-jk r_1}}{4\pi r_1} \, ds \tag{6b}
\]

Where \(J\) is the induced current on the sub-reflector, \(r_1\) is the distance from any point in the sub-reflector to the observation point \((x_m, y_m, z_m)\). \(r_1\) is a unit vector in the direction from any point in the sub-reflector to any observation point on the main reflector \((x_m, y_m, z_m)\).

MAIN REFLECTOR FAR-FIELDS

The resulting induced currents produced by the sub-reflector scattering the main reflector are integrated to obtain the far-zone electric fields.
\[ E(\theta, \phi) = -jkZ_0 e^{-jkR} \int \int (J - (J \cdot R)R) \frac{e^{jkr}}{4\pi R} \, ds \quad (7a) \]

\[ H(\theta, \phi) = (R \times E)/Z_0 \quad (7b) \]

Where \( J \) is the induced current in the main reflector, \( R \) is a unit vector from any point in the main reflector to the far-field observation point. \( r \) is the distance from the origin of the main reflector coordinate system to any point in the main reflector.

This method of calculating secondary pattern is accurate in cases where the antenna diameter is of the order greater than 50 to 100 wavelength. If the antenna diameter is of the order less than 50 wavelength, the accuracy is reduced, specifically in the sidelobe region. The reflector configuration described in figure 1 was analyzed by using various methods and computer codes. The calculated \( E \)- and \( H \)-plane far-field antenna pattern and directivities are shown in figures 4a and 4b respectively. The directivity and the far-field pattern are in a very good agreement among computer programs. The computer program given in appendix C was used to analyze the configuration.

**DIRECTIVITY**

The far zone electric field is usually divided into two orthogonal polarizations. Following Ludwig's definition 3 (ref. 4) the following unitary polarization vectors are introduced

\[ R = \theta \left( a e^{jp} \cos \phi + b \sin \phi \right) + \phi (-a e^{jp} \sin \phi + b \cos \phi) \quad (8a) \]

\[ C = \theta \left( a e^{jp} \sin \phi - b \cos \phi \right) + \phi \left( a e^{jp} \cos \phi + b \sin \phi \right) \quad (8b) \]

if the secondary pattern can be expressed as

\[ E = \frac{e^{-jkR}}{r} (E_\theta(\theta, \phi) + E_\phi(\theta, \phi)) \quad (9) \]
The reference-polarization expression is

\[ E_{\text{ref}} = E \cdot (R^*)^* \]  

(10a)

and the cross-polarization expression is

\[ C_{\text{cross}} = E \cdot (C^*)^* \]  

(10b)

The directivity for the reference polarization is defined by

\[ DR(\theta,\phi) = \frac{4\pi (E_{\text{ref}}E_{\text{ref}}^*)/Z_0}{P_{\text{rad}}} \]  

(11a)

similarly the directivity for the cross polarization is defined by

\[ DC(\theta,\phi) = \frac{4\pi (E_{\text{cross}}E_{\text{cross}}^*)/Z_0}{P_{\text{rad}}} \]  

(12b)

CONCLUDING REMARKS

A computer program using physical optics-current integration method, has been developed for calculating the far-field antenna pattern of dual reflector antennas illuminated by a feed with arbitrary polarization. The program utilizes a 3th order polynomial spline or nth order polynomial interpolation algorithms for cases in which the reflectors are numerically specified. The results for the far-field sidelobes and directivity are in good agreement with those obtained by other well-known techniques.

The computer program based on physical optics-current integration techniques is one of the main system engineering tools used at NASA Lewis Research Center for analyzing advanced antenna systems.
APPENDIX A

IDEAL REFLECTOR CONFIGURATIONS

Offset dual-reflectors are carved-out of portions of surfaces of revolutions (paraboloids, ellipsoids, hyperboloids, etc.) resulting from the intersection with cylinders or cones. The cylinders have their axes parallel to the axes of the parent reflector surfaces and the cones have their tips at one of the foci of the reflectors. In this appendix the geometrical characteristic of offset conic sections are presented.

The following are the analytical equations describing parabolic, hyperbolic and elliptical surfaces of revolution all are shown in main reflector coordinate system.

A: Parabolic reflector: The geometry associated with a parabolic reflector is shown in figure A-1

Parameters: F focal length

Equation: \[ z = \frac{x^2+y^2}{4F} \]

B: Hyperbolic Sub-reflector: The geometry associated with a hyperbolic sub-reflector is shown in figure A-2.

Parameters: \( z_0 \) offset distance
\( a \) vertex distance
\( b = \sqrt{c^2-a^2} \)
\( 2c \) foci distance

Equation: \[ z = z_0 + a \sqrt{1 + \frac{x^2+y^2}{b^2}} \]
Elliptical sub-reflector: The geometry associated with an elliptical sub-reflector is shown in figure A-3.

Parameters:
- \( z_0 \): offset distance
- \( a \): vertex distance
- \( b = \sqrt{a^2 - c^2} \)
- \( 2c \): foci distance

Equation:
\[
z = z_0 + a \sqrt{1 - \frac{x^2 + y^2}{b^2}}
\]
APPENDIX B
PROGRAM INPUT USER GUIDE

A computer program was designed to calculate the antenna performance characteristics. The method of analysis is physical optics. The program runs in an IBM370 using VM operating system. All the inputs are put into the program DRSG FORTRAN and are describe as follows:

- **FFREQ**: frequency GHz
- **QQ**: feed pattern exponent
- **OMX, DMY**: x and y length in wavelength main reflector rectangle
- **DSX, DSY**: x and y length in wavelength sub-reflector rectangle
- **MAXMX, MAXSY**: number of points in the x and y direction
- **xm0, ym0, zm0**: lower left corner of main reflector rectangle
- **xs0, ys0, zs0**: lower left corner of sub-reflector rectangle
- **xf, yf, zf**: feed location in wavelength
- **xr, yr, zr**: feed boresight point on sub-reflector
- **rtemp1 (sub)**: parameter \( b=\sqrt{a^2-c^2} \)
- **rtemp2**: offset distance in wavelength
- **fradsq**: radius square of cylinder sub-reflector
- **FCENX, FCENY**: center of sub-reflector cylinder
- **radsq**: radius of cylinder of main reflector
- **CNTRX, CNTRY**: center of cylinder of main reflector
- **rtemp1 (main)**: \( 1/4F, F \) is focal length in wavelength
PROGRAM DRSG
********************************************************************
* NASA LEWIS RESEARCH CENTER  
* AUTHOR : R. ACOSTA AND A. LAGIN IBM 370 VM VERSION  
* DATE : 7/14/91  
* PURPOSE : TO SIMULATE GENERALIZED SURFACES OF REVOLUTION  
* FOR THE GENERALIZED DUAL REFLECTOR ANALYSIS PROGRAM
********************************************************************
** COMPILE CONSTANTS
********************************************************************************
REAL C
PARAMETER (C = 2.99792458E+08 )
REAL PI
PARAMETER (PI = 3.141592653589793238)
REAL ETA
PARAMETER (ETA = 4.0*PI*C*(1.E-7) )
********************************************************************************
******INPUT : FREQUENCY (HZ)********************************************************************************
REAL FREQQ
PARAMETER (FREQQ = 19.45*(1.E+9) )
********************************************************************************
REAL LAMBDA
PARAMETER (LAMBDA = (C/FREQQ)**.000 )
REAL LAMBSQ
PARAMETER (LAMBSQ = LAMBDA*LAMBDA )
*********************************************************************************INPUT : QQ : FEED PATTERN EXPONENT********************************************************************************
REAL QQ
PARAMETER (QQ = 62.00 )
*********************************************************************************INPUT : QQ : FEED PATTERN EXPONENT********************************************************************************
REAL Q
PARAMETER (Q = 62.00 )
*********************************************************************************INPUT : QQ : FEED PATTERN EXPONENT********************************************************************************
INTEGER MXYZ
PARAMETER (MXYZ = 4 )
INTEGER DXYZ
PARAMETER (DXYZ = 3 )
INTEGER LNRY
PARAMETER (LNRY = 1 )
INTEGER MDTX
PARAMETER (MDTX = 10 )
********************************************************************************
** INTRINSIC FUNCTIONS
********************************************************************************
INTRINSIC SQRT
INTRINSIC INT
INTRINSIC NINT
********************************************************************************
** EXTERNAL SUBROUTINES
********************************************************************************
EXTERNAL SUBMAI
********************************************************************************
REAL DATARY(MDTX) DRS00060
REAL DMX,DMY DRS00060
REAL INCMX,INCMY DRS00060
REAL DSX,DSY DRS00060
REAL INCSX,INCSY DRS00060
REAL FREQ DRS00060
REAL Q DRS00060
********************************************************************************
** EQUIVALENCE
*EQUIVALENCE (DATARY(1),DMX )*
*EQUIVALENCE (DATARY(2),DMY )*
*EQUIVALENCE (DATARY(3),INCMX)*
*EQUIVALENCE (DATARY(4),INCMY)*
*EQUIVALENCE (DATARY(5),DSX )* 
*EQUIVALENCE (DATARY(6),DSY )* 
*EQUIVALENCE (DATARY(7),INCSX)*
*EQUIVALENCE (DATARY(8),INCSY)*
*EQUIVALENCE (DATARY(9),FREQ)*
*EQUIVALENCE (DATARY(10),Q )*

INTEGER MAXMX,MAXMY
INTEGER MAXSX,MAXSY

********************************************************************
******INPUT : LENGTH OF MAIN REF. RECTANGLE GRID***************
********************************************************************
DMX = (220.0 *LAMBDA)
DMY = (220.0 *LAMBDA)

********************************************************************
******INPUT : NUMBER POINTS IN X AND Y MAIN REF. RECTANGLE GRID*****
********************************************************************
MAXMX = 101
MAXMY = 101
INCMX = DMX/(REAL((MAXMX-I)))
INCMY = DMY/(REAL((MAXMY-I)))

********************************************************************
******INPUT : LENGTH OF SUB REF. RECTANGLE GRID **************
********************************************************************
DSX = ( 82.50 *LAMBDA)
DSY = ( 82.50 *LAMBDA)

********************************************************************
******INPUT : NUMBER OF POINTS IN X AND Y IN THE SUB REFLECTOR GRID********
********************************************************************
MAXSX = 61
MAXSY = 61
INCSX = (DSX/REAL(MAXSX-I))
INCSY = (DSY/REAL(MAXSY-I))
FREQ = FREQQ
Q = QQ

CALL SUBMAI(MAXMY,MAXMX,MAXSY,MAXSX,MDTX,DATARY)
END

********************************************************************
******SUBROUTINE SUBMAI************************************************
********************************************************************
SUBROUTINE SUBMAI(PMXMY,PMXMX,PMXSY,PMXSX,PDTX,DTAARX)
PARAMETER (C  = 2.99792458E+08 )
PARAMETER (PI  = 3.141592653589793238)
REAL ETA
PARAMETER (ETA = 4.0*PI*C*(1.E-7) )

** INPUT : FREQUENCY

REAL FREQQ
PARAMETER (FREQQ = 19.45*(1.E+9) )

REAL LAMBDA
PARAMETER (LAMBDA = (C/FREQQ)*1.000 )
REAL LAMBSQ
PARAMETER (LAMBSQ = LAMBDA*LAMBDA )

** INPUT : QQ: EXPONENT

REAL QQ
PARAMETER (QQ = 62.000 )

INTEGER MXYZ
PARAMETER (MXYZ = 4 )
INTEGER DXYZ
PARAMETER (DXYZ = 3 )
INTEGER LNRY
PARAMETER (LNRY = 1 )
INTEGER VDTX
PARAMETER (VDTX = 4 )
INTEGER MDTX
PARAMETER (MDTX = 10 )
INTEGER IDXZ
PARAMETER (IDXZ = 1 )
INTEGER IDXZX
PARAMETER (IDXZX = 2 )
INTEGER IDXZY
PARAMETER (IDXZY = 3 )
INTEGER IDMXSK
PARAMETER (IDMXSK = 4 )

** INPUT : NUMBER OF POINTS (-1) MAIN AND SUB REFLECTOR GRIDS

PARAMETER (MXMX = 101 )
PARAMETER (MXMY = 101 )
PARAMETER (MXSX = 61 )
PARAMETER (MXSY = 61 )
PARAMETER (TMXMX = MXMX + 1 )
PARAMETER (TMXMY = MXMY + 1 )
PARAMETER (TMXSX = MXSX + 1 )
PARAMETER (TMXSX = MXSY + 1 )

**

REAL MAIARY(4,TMXMY,TMXMX)
REAL SUBARY(4,TMXSY,TMXSX)
REAL XYZARY(DXYZ,MXYZ)
REAL MRXYZO(DXYZ),XMO,YMO,ZMO
REAL SRXYZO(DXYZ),XSO,YSO,ZSO
REAL FEDXYZ(DXYZ),XF ,YF ,ZF
REAL REFXYZ(DXYZ),XR ,YR ,ZR
REAL DMX, DMY
** Initialize Arrays

** MAIFIL <= MAIARY()
** SUBFIL <= SUBARY()
** XYZFIL <= XYZARY() <= MRXYZO(),SRXYZO(),FEDXYZ(),REFXYZ()
** DTAFIL <= DTAARX() <= DMX,DMY,INCMX,INCMY,DSX,DSY,INCSX,INCSY,FREQ

DO 20200 I = I, TMXMX, 1
DO 20100 J = 1, TMXMY, 1
    DO 20000 W = 1, 4, 1
        MAIARY(W, J, I) = 0.
    20000 CONTINUE
20100 CONTINUE
20200 CONTINUE
    DO 20500 I = 1, TMXSX, 1
        DO 20400 J = 1, TMXSY, 1
            DO 20300 W = 1, 4, 1
                SUBARY(W, J, I) = 0.
            20300 CONTINUE
        20400 CONTINUE
    20500 CONTINUE

********************************************************************************
******INPUT : FEED COORDINATES***********************************
********************************************************************************
******BORSIGHT BEAM FEED COORDINATES:
* XF = ( 0.000000000 *LAMBDA)
* YF = ( 0.000000000 *LAMBDA)
* ZF = ( 103.93560000 *LAMBDA)
******CLEVELAND BEAM FEED COORDINATES:
* XF = ( 5.629 *LAMBDA)
* YF = ( 101.067 *LAMBDA)
******MIAMI BEAM FEED COORDINATES:
* XF = ( -16.4 * LAMBDA)
* YF = ( 9.62287 * LAMBDA)
* ZF = ( 99.0325 * LAMBDA)
******LOS ANGELES BEAM FEED COORDINATES:
* XF = ( -7.996 * LAMBDA)
* YF = ( -30.52 * LAMBDA)
* ZF = ( 119.49 * LAMBDA)
******SEATLE BEAM FEED COORDINATES:
* XF = ( 3.4579 * LAMBDA)
* YF = ( -30.2 * LAMBDA)
* ZF = ( 119.32 * LAMBDA)
********************************************************************************
******INPUT : REFERENCE RAY LOCATION COORDINATES *******
********************************************************************************
** XR = ( 0.000000000 *LAMBDA)
** YR = ( 41.25000000 *LAMBDA)
** Y = YR*39.36
** ZR = ( 11.5*(3)SQRT((1+(XR+Y)**2)/1058)+97.5)
** ZR = ZR/39.36
********************************************************************************
INCMY = DTAARX(4)
DSX = DTAARX(5)
DSY = DTAARX(6)
INCSX = DTAARX(7)
INCSY = DTAARX(8)
FREQ = DTAARX(9)
Q = DTAARX(10)

**********TO GENERATE THE INPUT ARRAYS***********************

DO 10 I=1,10
  WRITE(15,805) DTAARX(I)
805 FORMAT(5X,E15.8)
10 CONTINUE

DO 11 I=1,4
  DO 12 J=1,3
    WRITE(16,806) XYZARY(J,I)
12 CONTINUE
11 CONTINUE

************************************************************
** Calculate Z, Zx, Zy, Usage for SubReflector.**
****Z = SUBARY(IDXZ,... SURFACE Z**
****DX= SUBARY(IDXXZ... DERIVATIVE WITH RESP. TO X**
****DY= SUBARY(IDXY... DERIVATIVE WITH RESP. TO Y**
************************************************************

RTEMPO = ( LAMBDA)

************************************************************

RTEMP1 = (18.97*RTEMPO)

************************************************************

RTEMP2 = (1./(.6829) )

************************************************************

RTEMP6 = 160.85*RTEMPO

************************************************************

FRADSQ=(19.0*RTEMPO)**2

************************************************************

FCENX=0.0
FCENY=31.8*RTEMPO

XS = XS0 - INCSX
DO 20700 I = 1,TMXSX,1
  XS = XS + INCSX
20700 CONTINUE

YS = YSO - INCSY
DO 20600 J = 1,TMXSY,1
  YS = YS + INCSY
20600 CONTINUE

FTEMP=(XS-FCENX)**2+(YS-FCENY)**2
* IF(FTEMP.GT.FRADSQ) GO TO 309
  RTEMP3 = ((XSQ) + (YS*YS))
  RTEMP4 = (SQRT(I+ RTEMP3*RTEMP2))
  
15
\[
RTEMP5 = \frac{((RTEMP1 \times RTEMP2))}{RTEMP4}
\]

**SubReflector: Hyperbola**

\[
\begin{align*}
Z &= 18.97 \times \text{LAMBDA} \times \frac{X}{1 + (X + Y)} \quad + 160.85 \times \text{LAMBDA} \\
\sqrt{} &= \frac{0.6829}{1+\left(\frac{X+Y}{11}\right)}
\end{align*}
\]

**Subary\((IDXZ,J,I) = ((RTEMP6)+RTEMP1\times RTEMP4)\)**

**Subary\((IDXZX,J,I) = (XS\times RTEMP5)\)**

**Subary\((IDXZY,J,I) = (YS\times RTEMP5)\)**

**Subary\((IDXMSK,J,I) = 1)\)**

```
20520 FORMAT (5X,4(EI5.8,2X))
309 WRITE(17,20520) SUBARY(IDXZ,J,I),SUBARY(IDXZX,J,I),
1 SUBARY(IDXZY,J,I),SUBARY(IDXMSK,J,I)
20600 CONTINUE
20700 CONTINUE
```
** Calculate Z, Zx, Zy, Usage for MainReflector.  

***Z = MAIARY(IDXZ,.... SURFACE FUNCTION  
***Z = MAIARY(IDXZX,... DERIVATIVE WITH RESP. X  
***Z = MAIARY(IDXZY,... DERIVATIVE WITH RESP. Y  

********************************************************************************

RTEMPO=LAMBDA

XM = XMO - INCMX

********************************************************************************

***INPUT: RADIUS OF CYLINDER MAIN REFLECTOR************************

* RADSQ = (107.2*RTEMPO)**2  
* RADSQ = (108.2*RTEMPO)**2

********************************************************************************

***INPUT: CENTER OF COORDINATES OF CYLINDER************************

CNTRX = 0.0
CNTRY = 189.7*RTEMPO

********************************************************************************

***INPUT : 1/4F PARAMETER FOCAL LENGTH************************

RTEMP1 = (1./(870.O0*RTEMPO))  
RTEMP2 = 2*RTEMP1

DO 20900  I = 1,TMXMX,1  
XM = XM + INCMX  
XSQ = XM*XM  
YM = YMO - INCMY

DO 20800  J = I,TMXMY,1  
YM=YM+INCMY  
RTEMP4 = (((XM-CNTRX)*(XM-CNTRX))+((YM-CNTRY)*(YM-CNTRY)))  
IF (RTEMP4 .GT. RADSQ) GO TO 450

* +:: ........

MaiReflector: Parabola

Z  = -------------------
(870. X LAMBDA)

Zx = -------------------
(435. X LAMBDA)

Zy = -------------------
(435. X LAMBDA)

17
MAIARY(IDXZ,J,I) = ((XM*XM)+(YM*YM))*RTEMP1
MAIARY(IDXZX,J,I) = (XM*RTEMP2)
MAIARY(IDXZY,J,I) = (YM*RTEMP2)
MAIARY(IDXMSK,J,I) = 1.

WRITE(18,20520) MAIARY(IDXZ,J,I),MAIARY(IDXZX,J,I),MAIARY(IDXZY,J,I),MAIARY(IDXMSK,J,I)
PROGRAM DUALREF

******************************************************************************
* AUTHOR : R. ACOSTA AND A. LAGIN VM VERSION
* DATE : 7/15/91
* PURPOSE : TO COMPUTE FAR FIELD CO-POL AND CROSS POL OF A
* GENERALIZED DUAL REFLECTOR SYSTEM
* GENERALIZED DUAL REFLECTOR ANALYSIS PROGRAM
******************************************************************************

** COMPILE CONSTANTS

REAL C
PARAMETER (C = 2.99792458E+08 )
REAL PI
PARAMETER (PI = 3.141592653589793238)
REAL ETA
PARAMETER (ETA = 4.0*PI*C*(1.E-7) )
REAL R2DEG
PARAMETER (R2DEG = 180./PI )
REAL D2RAD
PARAMETER (D2RAD = PI/180. )
INTEGER IDXZ
PARAMETER (IDXZ = 1 )
INTEGER IDXZX
PARAMETER (IDXZX = 2 )
INTEGER IDXZY
PARAMETER (IDXZY = 3 )
INTEGER IDXMSK
PARAMETER (IDXMSK = 4 )
INTEGER IDXVCX
PARAMETER (IDXVCX = 1 )
INTEGER IDXVCY
PARAMETER (IDXVCY = 2 )
INTEGER IDXVCZ
PARAMETER (IDXVCZ = 3 )
INTEGER IDXRMJ
PARAMETER (IDXRMJ = 1 )
INTEGER IDXIMJ
PARAMETER (IDXIMJ = 2 )
INTEGER IDXUNM
PARAMETER (IDXUNM = 3 )
INTEGER IDXJVX
PARAMETER (IDXJVX = 1 )
INTEGER MXYZ
PARAMETER (MXYZ = 1 )
INTEGER DXYZ
PARAMETER (DXYZ = 2 )
INTEGER LNRY
PARAMETER (LNRY = 3 )
INTEGER VDTX
PARAMETER (VDTX = 4 )
INTEGER NDTX
PARAMETER (NDTX = 5 )
INTEGER MDTX
PARAMETER (MDTX = 6 )
INTEGER KDTX
PARAMETER (KDTX = 7 )

******************************************************************************
PARAMETER (IDXNRM = 2)
PARAMETER (IDXDTX = 3)
PARAMETER (IDXMMM = 1)
PARAMETER (IDXMSI = 2)
PARAMETER (IDXAOI = 3)
PARAMETER (IDXPR = 1)
PARAMETER (IDXADB = 2)
PARAMETER (IDXRDB = 3)

PARAMETER (MXMX = 101)
PARAMETER (MXMY = 101)

PARAMETER (XMXX = 61)
PARAMETER (XMXY = 61)

PARAMETER (TMXMX = MXMX + 1)
PARAMETER (TMXMY = MXMY + 1)
PARAMETER (TMXMSX = MXSX + 1)
PARAMETER (TMXMSY = MXSY + 1)

PARAMETER (MXXZ = 4)
PARAMETER (DXYZ = 3)
PARAMETER (LNRY = 1)
PARAMETER (VDTX = 4)
PARAMETER (NDTX = 3)
PARAMETER (MDTX = 10)
PARAMETER (KDTX = 18)

** INTRINSIC FUNCTIONS

INTRINSIC SQRT
INTRINSIC SIN
INTRINSIC COS
INTRINSIC ACOS
INTRINSIC NINT

** EXTERNAL FUNCTIONS

REAL DOT
EXTERNAL DOT
REAL FDPTRN
EXTERNAL FDPTRN

** EXTERNAL SUBROUTINES

EXTERNAL CROSS
EXTERNAL SCALER
EXTERNAL VECADD
EXTERNAL VECSUB

** RUN TIME CONSTANTS
** REAL DTAARY(MDTX) DUAO1210
** REAL DMX,DMY DUAO1220
** REAL INCX,INCMY DUAO1240
** REAL DSX,DSY DUAO1250
** REAL INCX,INCSY DUAO1260
** INTEGER MAXMX,MAXMY DUAO1270
** INTEGER MAXSX,MAXSY DUAO1280
** REAL FREQ DUAO1290
** REAL Q DUAO1300
** EQUIVALENCE (DTAARY( 1),DMX ) DUAO1340
** EQUIVALENCE (DTAARY( 2),DMY ) DUAO1350
** EQUIVALENCE (DTAARY(3),INCMX) DUAO1360
** EQUIVALENCE (DTAARY(4),INCMY) DUAO1370
** EQUIVALENCE (DTAARY( 5),DSX ) DUAO1380
** EQUIVALENCE (DTAARY( 6),DSY ) DUAO1390
** EQUIVALENCE (DTAARY( 7),INCSX) DUAO1400
** EQUIVALENCE (DTAARY( 8),INCSY) DUAO1410
** EQUIVALENCE (DTAARY( 9),FREQ ) DUAO1420
** EQUIVALENCE (DTAARY(IO),Q ) DUAO1430
** RUNTIME VARS DUAO1440
** EQUIVALENCE DUAO1450
** REAL LAMBDADUAO1460
** REAL KDUAO1470
** EQUIVALENCE DUAO1480
** Input Arrays DUAO1490
** MAINFILE => MAINARY() DUAO1500
** SUBFILE => SUBARY() DUAO1510
** XYZFILE => XYZARY() => MRSYR(),SRSYR(),FEDX(),REFXY() DUAO1520
** DTAFILE => DTAARY() => DMX,DMY,INCMX,INCMY,DSX,DSY,INCSX,INCSY,FREQ DUAO1530
** READ IN THE XYZARY AND DTAARY FROM FILE GENERATOR DUAO1540
** DO 10 I=1,10 DUAO1590
** READ(15,805) DTAARY(I) DUAO1600
** 805 FORMAT(5X,E15.8) DUAO1610
** CONTINUE DUAO1620
** Initial Calculations DUAO1630
** LAMBDADUAO1640
** K = 2*PI/LAMBDADUAO1650
** MAXMX = (NINT(DMX/INCMX) + 1) DUAO1660
** MAXMY = (NINT(DMY/INCMY) + 1) DUAO1670
** MAXSX = (NINT(DSX/INCSX) + 1) DUAO1680
** MAXSY = (NINT(DSY/INCSY) + 1) DUAO1690
** CALL SUBMAI(MAXMX,MAXMY,MAXSX,MAXSY,MDTX,DTAARY) DUAO1700
** END DUAO1710
** MAIN PROGRAM DUAO1720
** SUBROUTINE SUBMAI(PMXMX,PMXMY,PMXSX,PMXSY,MDTX,DTAARY) DUAO1730
** INTEGER PMMXM DUAO1740
**
** COMPILE CONSTANTS
********************************************************************************
REAL C
PARAMETER (C = 2.99792458E+08 )
REAL PI
PARAMETER (PI = 3.141592653589793238)
REAL ETA
PARAMETER (ETA = 4.0*PI*C*(1.E-7) )
REAL R2DEG
PARAMETER (R2DEG = 180./PI )
REAL D2RAD
PARAMETER (D2RAD = PI/180. )
PARAMETER (IDXNM = 1)
PARAMETER (IDXMSI = 2)
PARAMETER (IDXAO = 3)
PARAMETER (IDXPR = 1)
PARAMETER (IDXAB = 2)
PARAMETER (IDXRD = 3)

PARAMETER (MXM = 101)
PARAMETER (MXY = 101)

PARAMETER (TMX = MXM + 1)
PARAMETER (TMY = MXY + 1)
PARAMETER (TMX = MXM + 1)
PARAMETER (TMY = MXY + 1)

PARAMETER (MNT = 50)
PARAMETER (MNP = 360)

PARAMETER (M = 4)
PARAMETER (DYZ = 3)
PARAMETER (LRY = 1)
PARAMETER (VTX = 4)
PARAMETER (NDT = 3)
PARAMETER (MDT = 10)
PARAMETER (KDT = 18)

** INTRINSIC FUNCTIONS

** EXTERNAL FUNCTIONS

** EXTERNAL SUBROUTINES

** RUN TIME CONSTANTS

REAL MAIARY(VTX, TMX, TMX)

DUA02410
DUA02420
DUA02430
DUA02440
DUA02450
DUA02460
DUA02470
DUA02480
DUA02490
DUA02500
DUA02510
DUA02520
DUA02530
DUA02540
DUA02550
DUA02560
DUA02570
DUA02580
DUA02590
DUA02600
DUA02610
DUA02620
DUA02630
DUA02640
DUA02650
DUA02660
DUA02670
DUA02680
DUA02690
DUA02700
DUA02710
DUA02720
DUA02730
DUA02740
DUA02750
DUA02760
DUA02770
DUA02780
DUA02790
DUA02800
DUA02810
DUA02820
DUA02830
DUA02840
DUA02850
DUA02860
DUA02870
DUA02880
DUA02890
DUA02900
DUA02910
DUA02920
DUA02930
DUA02940
DUA02950
DUA02960
DUA02970
DUA02980
DUA02990
DUA03000

23
REAL SUBARY(VDTX,TMXSY,TMXSX)
REAL XYZARY(DXYZ,MXYZ)
REAL SRXYZO(DXYZ,XSO,YSO,ZSO)
REAL MRXYZO(DXYZ,XMO,YMO,ZMO)
REAL FEDXYZ(DXYZ,XF,YF,ZF)
REAL REFXYZ(DXYZ,XR,YR,ZR)
REAL SR(SXYZ),SRX,SRY,SRZ
REAL USR(DXYZ),USRX,USRY,USRZ
REAL MAGSR
REAL DTAARY(MDTX)
REAL DMX,DMY
REAL INCX,INCMY
REAL DSX,DSY
REAL INCSX,INCSY
REAL FREQ
REAL Q
REAL DTXARY(KDTX)
REAL BEGPHX
REAL ENDPHX
REAL IDXPHX
REAL STPPHX
REAL INCPHX
REAL BEGTHX
REAL ENDTHX
REAL IDXTHX
REAL STPTHX
REAL INCTHX
REAL PTTMNX
REAL PTTMXX
REAL ADBMNX
REAL ADBMXX
REAL RDBMNX
REAL PRAD
REAL RINTNS
REAL RIFCTR
REAL DIRCTV

*********************************************************************************************
** EQUIVALENCE
*********************************************************************************************
EQUIVALENCE (XYZARY(1,1),MRXYZO(1))
EQUIVALENCE (XYZARY(1,2),SRXYZO(1))
EQUIVALENCE (XYZARY(1,3),FEDXYZ(1))
EQUIVALENCE (XYZARY(1,4),REFXYZ(1))
EQUIVALENCE (MRXYZO(1),XMO)
EQUIVALENCE (MRXYZO(2),YMO)
EQUIVALENCE (MRXYZO(3),ZMO)
EQUIVALENCE (SRXYZO(1),XSO)
EQUIVALENCE (SRXYZO(2),YSO)
EQUIVALENCE (SRXYZO(3),ZSO)
EQUIVALENCE (FEDXYZ(1),XF)
EQUIVALENCE (FEDXYZ(2),YF)
EQUIVALENCE (FEDXYZ(3),ZF)
EQUIVALENCE (REFXYZ(1),XR)
EQUIVALENCE (REFXYZ(2),YR)
EQUIVALENCE (REFXYZ(3),ZR)
EQUIVALENCE (SR(1),SRX)
EQUIVALENCE (SR(2),SRY)
EQUIVALENCE (SR(3),SRZ)
EQUIVALENCE (USR(1),USRX)
EQUIVALENCE (USR(2),USRY)
EQUIVALENCE (USR (3), USRZ )
EQUIVALENCE (DTAARY (1), DMX )
EQUIVALENCE (DTAARY (2), DMY )
EQUIVALENCE (DTAARY (3), INCMX )
EQUIVALENCE (DTAARY (4), INCMY )
EQUIVALENCE (DTAARY (5), DSX )
EQUIVALENCE (DTAARY (6), DSY )
EQUIVALENCE (DTAARY (7), INCSX )
EQUIVALENCE (DTAARY (8), INCSY )
EQUIVALENCE (DTAARY (9), FREQ )
EQUIVALENCE (DTAARY(10), Q )
EQUIVALENCE (DTXARY (1), BEGPHX )
EQUIVALENCE (DTXARY (2), ENDPHX )
EQUIVALENCE (DTXARY (3), IDXPHX )
EQUIVALENCE (DTXARY (4), STPPHX )
EQUIVALENCE (DTXARY (5), INCPhX )
EQUIVALENCE (DTXARY (6), BEGTHX )
EQUIVALENCE (DTXARY (7), ENDTHX )
EQUIVALENCE (DTXARY (8), IDXTHX )
EQUIVALENCE (DTXARY (9), STPTHX )
EQUIVALENCE (DTXARY (10), INCTHX )
EQUIVALENCE (DTXARY (11), PTMNX )
EQUIVALENCE (DTXARY (12), PTMXX )
EQUIVALENCE (DTXARY (13), ADBMNX )
EQUIVALENCE (DTXARY (14), ADBMX )
EQUIVALENCE (DTXARY (15), RDBMX )
EQUIVALENCE (DTXARY (16), PRAD )
EQUIVALENCE (DTXARY (17), RINTNS )
EQUIVALENCE (DTXARY (18), DIRCTV )

REAL MCDARY(DXYZ, DXYZ, TMXMY, TMXMX)
REAL SCDARY(DXYZ, DXYZ, TMXSY, TMXSX)
REAL MAIXYZ(DXYZ), XM, YM, ZM
REAL SUBXYZ(DXYZ), XS, YS, ZS
REAL GENXYZ(DXYZ), XI, YJ, ZIJ
REAL TMPXYZ(DXYZ, 2)
REAL TMRXYZ(DXYZ), TMRX, TMRY, TMRZ
REAL TMIXYZ(DXYZ), TMIX, TMIY, TMIZ
REAL HFLD(DXYZ, 2)
REAL TMXIZY(DXYZ), TMX, TMY, TMZ
REAL HVR (DXYZ), HVRX, HVRY, HVRZ
REAL HVI (DXYZ), HVIX, HVII, HVIJ
REAL JFLD(DXYZ, 2)
REAL JVR (DXYZ), JVRX, JVRY, JVRZ
REAL JVI (DXYZ), JVIX, JVIJ, JVIZ
REAL SUM (DXYZ, 2)
REAL SUMR(DXYZ), SUMRX, SUMRY, SUMRZ
REAL SUMI(DXYZ), SUMIX, SUMIY, SUMIZ
REAL NORM(DXYZ), NX, NY, NZ
REAL MAGNRM
REAL SI (DXYZ), SIX, SIY, SIZ
REAL USI (DXYZ), USIX, USIY, USIZ
REAL MAGSI
REAL PV (DXYZ), PVX, PVY, PVZ
REAL UPV (DXYZ), UPVX, UPVY, UPVZ
REAL MAGPV
REAL HV (DXYZ), HVX, HVY, HVZ
REAL UHV (DXYZ), UHVX, UHVY, UHVZ
REAL MAGHV
REAL JV (DXYZ), J VX, J VY, J VZ
REAL UJV (DXYZ), UJ VX, UJ VY, UJ VZ
REAL MAGJV
REAL R1 (DXYZ), R1 X, R1 Y, R1 Z
REAL URI (DXYZ), URI X, URI Y, URI Z
REAL MAGRI
REAL INTG (DXYZ), INTX, INTY, INTZ
REAL MAGINT
REAL RFF (DXYZ), RFF X, RFF Y, RFF Z
REAL PTTRN (NDTX, O: MXTHE, O: MXPHI)
REAL PTTMN
REAL PTTMX
REAL LAMBD A
REAL K
REAL KR
REAL RR
REAL PSI (2), COSKR, SINKR
REAL MIN
REAL MAX
REAL R1 TMP
REAL R2 TMP
REAL CMPTMP (2), CMPTMR, CMPTMI
REAL SCALE
REAL ANGLE
INTEGER ANGPHX
REAL ANGPHI
REAL BEGPHI
REAL ENDPHI
INTEGER IDXP HI
INTEGER STP HI
REAL INC PHI
REAL SINPHI
REAL COSPHI
INTEGER ANGTHX
REAL ANGTHE
REAL BEGTHE
REAL ENDTHE
INTEGER IDXTHE
INTEGER STPTHE
REAL INCTHE
REAL SINTHE
REAL COSTHE
REAL EPHRE
REAL EPHIM
REAL ETHRE
REAL ETHIM
REAL COZ
REAL MSKFAC
INTEGER MAXMX, MAXMY
INTEGER MAXSX, MAXSY
REAL DS
REAL INCMXY
REAL INCSXY
INTEGER I, J, IP, JP, V, W
INTEGER IOS
INTEGER FLG
INTEGER ITMP
CHARACTER TIME*8
REAL AA AKR
INTEGER II IKR
EQUIVALENCE (SUBXYZ(1), XS)
EQUIVALENCE (SUBXYZ(2), YS)
EQUIVALENCE (SUBXYZ(3), ZS)
EQUIVALENCE (MAIXYZ(1), XM)
EQUIVALENCE (MAIXYZ(2), YM)
EQUIVALENCE (MAIXYZ(3), ZM)
EQUIVALENCE (GENXYZ(1), XI)
EQUIVALENCE (GENXYZ(2), YJ)
EQUIVALENCE (GENXYZ(3), ZIJ)
EQUIVALENCE (TMPXYZ(I,1), TMRXYZ(1))
EQUIVALENCE (TMPXYZ(I,2), TMIXYZ(1))
EQUIVALENCE (TMRXYZ(1), TMRX)
EQUIVALENCE (TMRXYZ(2), TMRY)
EQUIVALENCE (TMRXYZ(3), TMRZ)
EQUIVALENCE (TMIXYZ(1), TMIX)
EQUIVALENCE (TMIXYZ(2), TMIY)
EQUIVALENCE (TMIXYZ(3), TMIZ)
EQUIVALENCE (SUM(1,1), SUMR(1))
EQUIVALENCE (SUM(1,2), SUMI(1))
EQUIVALENCE (SUMR(1), SUMRX)
EQUIVALENCE (SUMR(2), SUMRY)
EQUIVALENCE (SUMR(3), SUMRZ)
EQUIVALENCE (SUMI(1), SUMIX)
EQUIVALENCE (SUMI(2), SUMIY)
EQUIVALENCE (SUMI(3), SUMIZ)
EQUIVALENCE (NORM(1), NX)
EQUIVALENCE (NORM(2), NY)
EQUIVALENCE (NORM(3), NZ)
EQUIVALENCE (SI(1), SIX)
EQUIVALENCE (SI(2), SIY)
EQUIVALENCE (SI(3), SIZ)
EQUIVALENCE (USI(1), USIX)
EQUIVALENCE (USI(2), USIY)
EQUIVALENCE (USI(3), USIZ)
EQUIVALENCE (PV(1), PVX)
EQUIVALENCE (PV(2), PVY)
EQUIVALENCE (PV(3), PVZ)
EQUIVALENCE (UPV(1), UPVX)
EQUIVALENCE (UPV(2), UPVY)
EQUIVALENCE (UPV(3), UPVZ)
EQUIVALENCE (HFLD(1,1), HVR(1))
EQUIVALENCE (HFLD(1,2), HVI(1))
EQUIVALENCE (HVR(1), HVRX)
EQUIVALENCE (HVR(2), HVRY)
EQUIVALENCE (HVR(3), HVRZ)
EQUIVALENCE (HVI(1), HVIX)
EQUIVALENCE (HVI(2), HVIY)
EQUIVALENCE (HVI(3), HVIZ)
EQUIVALENCE (HV(1), HVX)
EQUIVALENCE (HV(2), HVY)
EQUIVALENCE (HV(3), HVZ)
EQUIVALENCE (UHV(1), UHVX)
EQUIVALENCE (UHV(2), UHYY)
EQUIVALENCE (UHV(3), UHVZ)
EQUIVALENCE (JFLD(1,1), JVR(1))
EQUIVALENCE (JFLD(1,2), JVI(1))
EQUIVALENCE (JVR(1), JVRX)
EQUIVALENCE (JVJR (2), JVRY )
EQUIVALENCE (JVJR (3), JVRYJ)
EQUIVALENCE (JVI (1), JVIX )
EQUIVALENCE (JVI (2), JVIY )
EQUIVALENCE (JVI (3), JVIZ )
EQUIVALENCE (JV (1), JVX )
EQUIVALENCE (JV (2), JYV )
EQUIVALENCE (JV (3), JVZ )
EQUIVALENCE (UJVR (1), UJVRX )
EQUIVALENCE (UJVR (2), UJVRY )
EQUIVALENCE (UJVR (3), UJVRZ )
EQUIVALENCE (RI (1), RIX )
EQUIVALENCE (RI (2), RIY )
EQUIVALENCE (RI (3), RIZ )
EQUIVALENCE (UR1 (1), UR1X )
EQUIVALENCE (UR1 (2), UR1Y )
EQUIVALENCE (UR1 (3), UR1Z )
EQUIVALENCE (INTG (1), INTX )
EQUIVALENCE (INTG (2), INTY )
EQUIVALENCE (INTG (3), INTZ )
EQUIVALENCE (RFF (1), RFFX )
EQUIVALENCE (RFF (2), RFFY )
EQUIVALENCE (RFF (3), RFFZ )
EQUIVALENCE (PSI (1), COSKR )
EQUIVALENCE (PSI (2), SINKR )
EQUIVALENCE (CMTMP (1), CMPTMR )
EQUIVALENCE (CMTMP (2), CMPTM )

********************************************************************************
** INITIALIZE INPUT ARRAYS
MAIFIL ==> MAIARY() DUA05700
SUBFIL ==> SUBARY() DUA05710
XYZFIL ==> XYZARY() ==> MRXYZO(),SRXYZO(),FEDXYZ(),REFXYZ() DUA05720
DTAFIL ==> DTAARY() ==> DMX,DMY,INCMX,INCMY,DSX,DSY,INCSX,INCSY,FREQ,DUA05730
********************************************************************************
** GENERATE OUTPUT ARRAYS
MCDFIL == MCDARY() DUA05750
SCDFIL == SCDARY() DUA05770
RFXFILE == PTTTRN () DUA05780
DXFIL == DTXARY() <= BEG,END,IDX,STP,INC * THE,PHI , PTT * ADR * DUA05790
********************************************************************************
** READ IN THE XYZARY, SUBARY, MAIARY ARRAYS FROM FILE GEN.************
DO 11 II=1,4
DO 12 JJ=1,3
READ(16,806)XYZARY(JJ,II)
FORMAT(5X,E15.8)
CONTINUE
CONTINUE
806
12 CONTINUE
11 CONTINUE

DO 13 III=1,TMXSMX,1
DO 14 JJJ=1,TMXSMY,1
READ(17,807)SUBARY(IDXZ,JJJ,III),
1 SUBARY(IDXZX,JJJ,III),
2 SUBARY(IDXZY,JJJ,III),
3 SUBARY(IDXMSK,JJJ,III)
807
14 CONTINUE
13 CONTINUE
15 CONTINUE
DO 15 IIII=1,TMXMX,1
DUA05810
DUA05820
DUA05830
DUA05840
DUA05850
DUA05860
DUA05870
DUA05880
DUA05890
DUA05900
DUA05910
DUA05920
DUA05930
DUA05940
DUA05950
DUA05960
DUA05970
DUA05980
DUA05990
DUA06000
DO 16 JJJJ=I,TMXMY,I
  READ(18,807)MAIARY(IDXZ,JJJJ,IIII), DUA06010
  1  MAIARY(IDXZX,JJJJ,IIII), DUA06020
  2  MAIARY(IDXZY,JJJJ,IIII), DUA06030
  3  MAIARY(IDXMSK,JJJJ,IIII) DUA06040
16  CONTINUE DUA06050
15  CONTINUE DUA06060

***********************************************************************
******INITIAL CALCULATIONS*********************************************
***********************************************************************
DO 00100   I = 1,PDTX,1
  DTAARY(I) = DTXARX(I)
00100  CONTINUE DUA06100

LAMBDA = C/FREQ
K = 2*PI/LAMBDA
MAXMX = PMXMX
MAXMY = PMXMY
MAXSX = PMXSX
MAXSY = PMXSX
MAGSR = O
DO 00200   V = 1,3,1
  R1TMP = REFXYZ(V) - FEDXYZ(V)
  SR(V) = R1TMP
  MAGSR = MAGSR + R1TMP*R1TMP
00200  CONTINUE

MAGSR=SQRTR(T(MAGSR))
DO 00300   V = 1,3,1
  USR(V) = SR(V)/MAGSR
00300  CONTINUE

***********************************************************************
******INPUT : FAR-FIELD LIMIT POINTS***********************************
****** E-PLANE : 90 - 270 PHI CUTS
****** H-PLANE : 0 - 180 PHI CUTS
****** 45-PLANE: 45 - 225 PHI CUTS
***********************************************************************

BEGPHI = 74.99858*pi/180.
ENDPHI = 254.99858*pi/180.
IDXPHI = 1
STPPHI = 1
INCTPHI = (ENDPHI-BEGPHI)/IDXPHI
* INC=PHI = 0

***********************************************************************
******INPUT : FAR-FIELD LIMIT POINTS***********************************
****** E-PLANE : 90 - 270 PHI CUTS
****** H-PLANE : 0 - 180 PHI CUTS
****** 45-PLANE: 45 - 225 PHI CUTS
***********************************************************************

BEGIN = 0.
ENDD = 5.*PI/180
IDXTH = 50
STPTH = 1
INCTH = (ENDD-BEGIN)/IDXTH

***********************************************************************
******INPUT : FAR-FIELD LIMIT POINTS***********************************
****** E-PLANE : 90 - 270 PHI CUTS
****** H-PLANE : 0 - 180 PHI CUTS
****** 45-PLANE: 45 - 225 PHI CUTS
***********************************************************************

** Calculate Current Densities on the SubReflector Resulting from Source**

INCSXY = INCXS*INCSY
XS = XSO - INCSX
DO 00700   I = 1,MAXSX,1
  XS = XS + INCSX
  YS = YSO - INCSY
  DO 00600   J = 1,MAXSY,1
    YS = YS + INCSY
    ZS = SUBARY(IDXZ,J,I)
    MSKFAC = SUBARY(IDXMSK,J,I)
    IF (MSKFAC .EQ. 1.) THEN
      DUA06510
      DUA06520
      DUA06530
      DUA06540
      DUA06550
      DUA06560
      DUA06570
      DUA06580
      DUA06590
      DUA06600
MAGSI = 0.
DO 00500  V = 1,3,1
   RITMP = SUBXYZ(V) - FEDXYZ(V)
   SI(V) = RITMP
   MAGSI = MAGSI + RITMP*RITMP
CONTINUE
MAGSI = SQRT(MAGSI)
IF (MAGSI .EQ. 0.) THEN
   STOP
ENDIF
DO 00510  V = 1,3,1
   USI(V) = SI(V)/MAGSI
CONTINUE
PVX = (-USIX*USIY)
PVY = (USIX*USIX + USIZ*USIZ)
PVZ = (-USIY*USIZ)
MAGPV = SQRT(PVX*PVX + PVY*PVY + PVZ*PVZ)
DO 00520  V = 1,3,1
   UPV(V) = PV(V)/MAGPV
CONTINUE
COSTHE = DOT(USR,UPV,3)
SCALE = FDPTRN(USR,USI,Q,COZ,FLG)/ETA
IF ( (FLG .EQ. 1) .OR. (SCALE .EQ. 0.)) THEN
   STOP
ENDIF
CALL CROSS(HV,USI,UPV)
MAGHV = SQRT(HVX*HVX + HVY*HVY + HVZ*HVZ)
DO 00530  V = 1,3,1
   UHV(V) = HV(V)/MAGHV
CONTINUE
NX = SUBARY(IDXZX,J,I)
NY = SUBARY(IDXZY,J,I)
NZ = -1
MAGNRM = SQRT(NX*NX + NY*NY - 1)
DO 00540  V = 1,3,1
   RITMP = NORM(V)/MAGNRM
   NORM(V) = RITMP
   SCDARY(V,IDXNRM,J,I) = RITMP
CONTINUE
CALL CROSS(JV,NORM,UHV)
MAGJV = SQRT(JVX*JVX + JVY*JVY + JVZ*JVZ)
MAGJV = 1.
DO 00550  V = 1,3,1
   RITMP = JV(V)/MAGJV
   UJV(V) = RITMP
   SCDARY(V,IDXJVX,J,I) = RITMP
CONTINUE
IF (MAGSI .EQ. 0.) THEN
   STOP
ENDIF
SCDARY(IDXMNM,IDXDTX,J,I) = MAGNRM
SCDARY(IDXMSI,IDXDTX,J,I) = MAGSI
SCDARY(IDXAOT,IDXDTX,J,I) = (2.*SCALE)
SCALE = 2.*SCALE/MAGSI
COSKR = COS(K*MAGSI)
SINKR = SIN(K*MAGSI)
JFLD(IDXVCX,IDXRMJ) = (SCALE*UJV(IDXVCX)*COSKR)
JFLD(IDXVCY,IDXRMJ) = (SCALE*UJV(IDXVCY)*COSKR)
JFLD(IDXVCZ,IDXRMJ) = (SCALE*UJV(IDXVCZ)*COSKR)
JFLD(IDXVCX,IDXIMJ) = -(SCALE*UJV(IDXVCX)*SINKR)
JFLD(IDXVCY,IDXIMJ) = -(SCALE*UJV(IDXVCY)*SINKR)
JFLD(IDXV CZ,IDXIMJ) = -(SCALE*UJV(IDXV CZ)*SINKR)
ELSE
ENDIF
00599 CONTINUE
00600 CONTINUE
00700 CONTINUE

******************************************************************************
** Calculate Induced Magnetic Field on Main Reflector by Sub Reflector. ******
******************************************************************************

INCMXY = INC MX*INCMY
XM = XMO - INC MX
DO 01900 IP = 1,MAXMX,1
XM = XM + INC MX
YM = YMO - INC MY
DO 01800 JP = 1,MAXMY,1
YM = YM + INC MY
ZM = MAIARY(IDXZ,JP,IP)
MSKFAC = MAIARY(IDXMSK,JP,IP)
IF (MSKFAC .EQ. 1.) THEN
NX = -(MAIARY(IDXZX,JP,IP))
NY = -(MAIARY(IDXZY,JP,IP))
NZ = +1.
MAGNRM = SQRT(NX*NX + NY*NY +1.)
DO 01100 V = 1,3,1
NORM(V) = NORM(V)/MAGNRM
DO 01000 W = 1,2,1
HFLD(V,W) = O.
CONTINUE
01000 CONTINUE
01100 CONTINUE
XS = XS0 - INCSX
DO 01700 I = 1,MAXSX,1
XS = XS + INCSX
YS = YSO - INCSY
DO 01600 J = 1,MAXSY,1
YS = YS + INCSY
ZS = SUBARY(IDXZ,J,I)
MSKFAC = SUBARY(IDXMSK,J,I)
IF (MSKFAC .EQ. 1.) THEN
DS = SCDARY(IDXMM,IDXDTX,J,I)*INCSXY
MAGSI = SCDARY(IDXMSI,IDXDTX,J,I)
SCALE = SCDARY(IDXAOT,IDXDTX,J,I)
MAGR1 = 0.
DO 01200 V = 1,3,1
R1TMP = (MAIXYZ(V) - SUBXYZ(V))
R1(V) = R1TMP
MAGR1 = MAGR1 + R1TMP*R1TMP
CONTINUE
01200 CONTINUE
MAGR1 = SQRT(MAGR1)
DO 01300 V = 1,3,1
URI(V) = R1(V)/MAGR1
UHV(V) = SCDARY(V,IDXJVX,J,I)
CONTINUE
01300 CONTINUE
KR = (K*(MAGR1 + MAGSI))
RR = MAGR1*MAGSI
COSKR = +COS(KR)/RR
SINKR = +SIN(KR)/RR
*
CALL CROSS(INTG,SCDARY(IDXVCX,IDXJVX,J,I),URI)
CALL CROSS(INTG,UHV,URI)
MAGINT = 1.
MAGINT = SQRT((INTX*INTX) + (INTY*INTY) + (INTZ*INTZ))
INTX = INTX/MAGINT
INTY = INTY/MAGINT
INTZ = INTZ/MAGINT
DO 01500 V = 1,3,1
DO 01400 W = 1,2,1
HFLD(V,W) = HFLD(V,W) + SCALE*PSI(W)*INTG(V)*DS
CONTINUE
CONTINUE
CONTINUE
CONTINUE
MCDARY(IDXVCX,IDXUNM,JP,IP) = NX
MCDARY(IDXVCY,IDXUNM,JP,IP) = NY
MCDARY(IDXVCZ,IDXUNM,JP,IP) = NZ
CALL SCALER(NORM,NORM,2.)
CALL CROSS (JFLD(IDXVCX,IDXRMJ),NORM,HFLD(IDXVCX,IDXRMJ))
CALL CROSS (JFLD(IDXVCX,IDXIMJ),NORM,HFLD(IDXVCX,IDXIMJ))
DO 01720 V = 1,3,1
DO 01710 W = 1,2,1
MCDARY(V,W,JP,IP) = JFLD(V,W)
CONTINUE
CONTINUE
********************************************************************************
ELSE
JFLD(IDXVCX,IDXRMJ) = 0.
JFLD(IDXVCY,IDXRMJ) = 0.
JFLD(IDXVCZ,IDXRMJ) = 0.
JFLD(IDXVCX,IDXIMJ) = 0.
JFLD(IDXVCY,IDXIMJ) = 0.
JFLD(IDXVCZ,IDXIMJ) = 0.
MAGJV = 0.
ENDIF
********************************************************************************
*** FAR FIELD ANTENNA PATTERN COMPUTATION
PTTMIN = +1.E+38
PTTMAX = -1.E+38
ANGPHI = BEGPHI - INCPHI
DO 02600 ANGPHX = 0,IDXPHI,STPPHI
ANGPHI = ANGPHI + INCPHI
SINPHI = SIN(ANGPHI)
COSPHI = COS(ANGPHI)
ANGTHE = BEGTHE - INCTHE
DO 02500 ANGTXH = 0,IDXTHE,STPTHE
ANGTH = ANGTHE + INCTHE
SINTHE = SIN(ANGTHE)
COSTHE = COS(ANGTHE)
RFFX = SINTHE*COSPHI
RFFY = SINTHE*SINPHI
RFFZ = COSTHE
SUM(1,1) = 0.
SUM(1,2) = 0.
SUM(2,1) = 0.
SUM(2,2) = 0.
SUM(3,1) = 0.
SUM(3,2) = 0.
XM = XMO - INCMX
DO 02400 IP = 1,MAXMX,1
XM = XM + INCMX
YM = YMO - INCMY
DO 02300 JP = 1,MAXMY,1
YM = YM + INCMY
ZM = MAIARY(IDXZ,JP,IP)
KR = (K*(((RFFX*XM)+(RFFY*YM)+(RFFZ*ZM)))
COSKR = COS(KR)
SINKR = SIN(KR)
MSKFAC = MAIARY(IDXMSK,JP,IP)
IF(MSKFAC.EQ.0)GO TO 2300
NORM(IDXVCX) = -MAIARY(IDXZX,JP,IP)
NORM(IDXVCY) = -MAIARY(IDXZY,JP,IP)
NORM(IDXVCZ) = 1.
DS = INCMXY*SQRT(NORM(IDXVCX)*NORM(IDXVCX)
+ NORM(IDXVCY)*NORM(IDXVCY)
+ NORM(IDXVCZ)*NORM(IDXVCZ))
CMPTMP (IDXRMJ) =
1
RFFX*MCDARY(IDXVCX,IDXRMJ,JP,IP)
2
+ RFFY*MCDARY(IDXVCY,IDXRMJ,JP,IP)
3
+ RFFZ*MCDARY(IDXVCZ,IDXRMJ,JP,IP)
CMPTMP(IDXIMJ) =
1
RFFX*MCDARY(IDXVCX,IDXIMJ,JP,IP)
2
+ RFFY*MCDARY(IDXVCY,IDXIMJ,JP,IP)
3
+ RFFZ*MCDARY(IDXVCZ,IDXIMJ,JP,IP)
TMRX = CMPTMP(IDXRMJ)*RFFX
TMRY = CMPTMP(IDXRMJ)*RFFY
TMRZ = CMPTMP(IDXRMJ)*RFFZ
TMIY = CMPTMP(IDXIMJ)*RFFY
TMIZ = CMPTMP(IDXIMJ)*RFFZ
TMRX = MCDARY(IDXVCX,IDXRMJ,JP,IP) - TMRX
TMRY = MCDARY(IDXVCY,IDXRMJ,JP,IP) - TMRY
TMRZ = MCDARY(IDXVCZ,IDXRMJ,JP,IP) - TMRZ
TMIY = MCDARY(IDXVCX,IDXIMJ,JP,IP) - TMIY
TMIZ = MCDARY(IDXVCX,IDXIMJ,JP,IP) - TMIZ
TTMRX = ((TMRX*COSKR)-(TMIY*SINKR))
TTMRY = ((TMRY*COSKR)-(TMIY*SINKR))
TTMRZ = ((TMRZ*COSKR)-(TMIY*SINKR))
TTMIY = ((TMRY*SINKR)+(TMIY*COSKR))
TTMIY = ((TMRZ*SINKR)+(TMIY*COSKR))
TTMIY = ((TMRZ*SINKR)+(TMIY*COSKR))
TTMIY = ((TMRZ*SINKR)+(TMIY*COSKR))
JVRX = TTMRX*DS
JVRY = TTMRY*DS
JVRZ = TTMRZ*DS
JVIX = TTMIX*DS
JVIY = TTMIX*DS
JVIZ = TTMIX*DS
SUMRX = SUMRX + JVRX
SUMRY = SUMRY + JVRY
SUMRZ = SUMRZ + JVRZ
SUMIX = SUMIX + JVIX
SUMIY = SUMIY + JVIY
SUMIZ = SUMIZ + JVIZ
02299  CONTINUE
02300  CONTINUE
02400  CONTINUE
0  ETHRE = SUMRX*COSTHE*COSPHI
          + SUMRY*COSTHE*SINPHI
          - SUMRZ*SINTHE
1  ETHIM = SUMIX*COSTHE*COSPHI
          + SUMIY*COSTHE*SINPHI
          - SUMIZ*SINTHE
2  EPHRE = - SUMRX*SINPHI
          + SUMRY*COSPHI
2  EPHIM = - SUMIX*SINPHI
          + SUMIY*COSPHI
***********************************************************************
**********ETHETA AND EPHI COMPUTED ************************************
**********TRANSFORM TO CO AND CO-POL LUDWIG'S DEFINITION***************
***********************************************************************
AREFR=SINPHI*ETHRE+COSPHI*EPHRE
AREFI=SINPHI*ETHIM+COSPHI*EPHIM
ACRPR=-COSPHI*ETHRE+SINPHI*EPHRE
ACRPI=-COSPHI*ETHIM+SINPHI*EPHIM
**********************************************************************
AMGREF=AREFR**2+AREFI**2
AMGCRP=ACRPR**2+ACRPI**2
**********CO AND CROSS POL FIELDS ARE COMPLETED************************
**********TOTAL FIELD**************************************************
* 0  RITMP = ETHRE*ETHRE + ETHIM*ETHIM
* 1  + EPHRE*EPHRE + EPHIM*EPHIM
**********PLOT THE CO POL FIELDS***************************************
RITMP = AMGREF
**********PLOT THE CROSS POL FIELDS************************************
*  RITMP = AMGCRP
**********PLOT THE CROSS POL FIELDS*************************************
PTTRN(IDXPWR,ANGTHX,ANGPHX) = RITMP
IF (RITMP .LT. PTTMIN) THEN
  PTTMIN = RITMP
ENDIF
IF (RITMP .GT. PTTMAX) THEN
  PTTMAX = RITMP
ENDIF
02500  CONTINUE
02600  CONTINUE
PRAD = ((2.*PI)/(ETA*((2.*Q)+I)))
RIFCTR = (((K*K*K*K)*ETA)/(256.*(PI*PI*PI*PI)))
RINTNS = (RIFCTR*PTTMAX)
DIRCTV = ((4.*PI*RINTNS)/(PRAD))
DIRCTV = (IO*ALOGIO(DIRCTV))
***********************************************************************
WRITE(19,897)DIRCTV
897  FORMAT(SX,F15.5)
***********************************************************************
ADBMNX = IO.*ALOGIO(PTTMIN)
ADBMXX = IO.*ALOGIO(PTTMAX)
RDBMNX = (ADBMNX - ADBMXX)
ANGPHI = BEGPHI - INCPHI
DO 02800  ANGPHX = 0, IDXPHI,STPPHI
  ANGPHI = ANGPHI + INCPHI
ENDDO
02800  CONTINUE
ANGTH = BEGTHE - INC THE
DO 02700 ANGTHX = 0, IDXTHE, STPTHE
  ANGTHE = ANGTHE + INC THE
  RITMP = 10. * ALOG10((PTTRN(IDXPWR, ANGTHX, ANGPHX)))
  PTTTRN(IDXRDB, ANGTHX, ANGPHX) = (RITMP )
  PTTTRN(IDXRDB, ANGTHX, ANGPHX) = (RITMP - ADMMXX)
END

ZETA = ANGTHE * 180 / PI
APHI = ANGPHI * 180 / PI
WRITE (19, 895) PTTTRN(IDXRDB, ANGTHX, ANGPHX), ZETA, APHI
895 FORMAT (5X, F15.8, 3X, 2(F15.8, 3X))
02700 CONTINUE
02800 CONTINUE

DTXARY(2) = ENDPHI
DTXARY(3) = REAL(IDXPHI)
DTXARY(4) = REAL(STPPHI)
DTXARY(5) = INC PHI
DTXARY(6) = BEGTHE
DTXARY(7) = ENDTHE
DTXARY(8) = REAL(IDXTHE)
DTXARY(9) = REAL(STPTHE)
DTXARY(10) = INC THE
DTXARY(11) = PTTMIN
DTXARY(12) = PTTMAX
END

REAL FUNCTION FDPTRN RETURN FEED PATTERN

REAL FUNCTION FDPTRN (THETA, PHI, RHO, COZ, ERR)
REAL THETA(3)
REAL PHI(3)
REAL RHO
REAL COZ
INTEGER ERR
REAL DOT
EXTERNAL DOT
REAL DOTVAL
DOTVAL = DOT(THETA, PHI, 3)
IF (DOTVAL .LT. 0.) THEN
  ERR = 1
  ERR = 1
  COZ = 0.
  FDPTRN = 0.
ELSE
  ERR = 0
  COZ = DOTVAL
  FDPTRN = (DOTVAL)**RHO
ENDIF
RETURN
END

REAL FUNCTION DOT(A, B, N)
INTEGER N
REAL A(N)
REAL B(N)
INTEGER I

** REAL FUNCTION DOT() ! Returns Real Value of DOT PRODUCT A and B
REAL SUM

SUM = 0.

DO 00100 I = 1,N,1
    SUM = SUM + A(I)*B(I)
00100 CONTINUE

DOT = SUM

RETURN

END

******************************************************************************

** SUBROUTINE CROSS ! Performs C = AxB

********** SUBROUTINE CROSS(C,A,B) **********

REAL C(3)
REAL A(3)
REAL B(3)

C(1) = (A(2)*B(3))-(A(3)*B(2))
C(2) = -(A(1)*B(3))-(A(3)*B(1))
C(3) = +(A(1)*B(2))-(A(2)*B(1))

END

******************************************************************************

** SUBROUTINE SCALER ! Performs C = A<SCALER>

********** SUBROUTINE SCALER(C,A,SCALEX) **********

REAL C(3)
REAL A(3)
REAL SCALEX

C(1) = SCALEX*(A(1))
C(2) = SCALEX*(A(2))
C(3) = SCALEX*(A(3))

RETURN

END

******************************************************************************

** SUBROUTINE VECADD ! Performs C = A+B

********** SUBROUTINE VECADD(C,A,B) **********

REAL C(3)
REAL A(3)
REAL B(3)

C(1) = (A(1)+B(1))
C(2) = (A(2)+B(2))
C(3) = (A(3)+B(3))

RETURN

END

******************************************************************************

** SUBROUTINE VECSUB ! Performs C = A-B

********** SUBROUTINE VECSUB(C,A,B) **********

REAL C(3)
REAL A(3)
REAL B(3)

C(1) = (A(1)-B(1))
C(2) = (A(2)-B(2))
C(3) = (A(3)-B(3))

RETURN

END
PROGRAM FFPLOT
DIMENSION X(IO000),Y(IO000),VARS(20)
DIMENSION XPL(IO000),YPL(IO000)
CHARACTER*13 CH/'DIRECTIVITY ='/
CHARACTER*2 ADB/'DB'/
CHARACTER*5 DIR(1)

C*****THIS PROGRAM CAN BE USED TO PLOT THE ANTENNA FAR-FIELD PATTERN
C******(E-PLANE OR H-PLANE CUTS)
C*****IAXIS,NUM,Y,RTNARR:PARAMETERS IN SCLBK2
C*****IVAR:PARAMETERS IN GLOT3
INTEGER * 4 IAXIS
INTEGER * 4 IVARS(20)
INTEGER * 4 NUM/IO000/
CHARACTER*4 XTITLE(5)/'ELEV','ATIO','N AN','GLE ','DEG.'/
CHARACTER*4 YTITLE(6)/'RELA','TIVE',' AMP','LITU','DE ','(DB)'/

C*****NP : TOTAL NO. OF POINTS ; ZM: MAXM. VIEWING ANGLE(DEG.)
READ(19,756)DDIR
756 FORMAT(5X,F15.5)

NP=I02
ZM=2.

DO 15 J=I,NP
     C*****X : ANGLE POSITIONS(DEG.) ; Y: RELATIVE FAR FLD. AMPLITUDES(DB)
     READ(19,300)YPL(J) ,XPL(J),DUMI
15 CONTINUE

DO 98 J=1,51
     Y(J)=(YPL(IO3-J))
     X(J)=-XPL(IO3-J)
     Y(J+51)=(YPL(J))
     X(J+51) =XPL(J)
98 CONTINUE

C*****SCLKK2:GRAPH3D ROUTINE TO FIND MIN,MAX IN DATA
C*****0 : Y-COORDINATE ; NUM : DIMENSION OF Y-ARRAY ; Y : Y-ARRAY
C*****RTNARR(2) : DIMENSION TO STORE Y(MIN),Y(MAX) VALUES
C*****REARRANGE THE FAR FIELD VALUES***********************************
C*****UXTRM :GRAPH3D ROUTINE; DEFINES EXTREME POSITIONS OF A 3D PLOT
C*****8 :TOTAL NO. OF VARIABLES ;! :CARTESIAN ; (-ZM,ZM) :(XMIN,XMAX)
C***** (-80.,0) :(YMIN,YMAX) ; (0.,0.) :(ZMIN,ZMAX)
CALL UXTRM(8,0,-ZM,ZM,-54.,O.,O.O,O.)
C*****UMAPF :GRAPH3D ROUTINE , DEFINE MAPPING TO TRANSFORM FROM USER.
C*****TO RELATIVE UNITS.
C***** 0 :CARTESIAN ; 1. :ONE VARS,DEFAULT ; 0 :NO LOG SCALE
CALL UMAPF(0,1.,0)
C*****XAXIS3 : GRAPH3D ROUTINE , DEFINES X-AXIX COORDINATE
C***** VARS: 1=TOTAL NO OF VARS ; 2 3 4=X1 Y1 Z1 ; 5 6 7=X2 Y2 Z2
C***** 8=USER UNIT(1.) ; 9=NO. OF INTERVALS ; 10=GRID OPTION(1.)
C***** 11=DRAW PARALLEL TO Y-AXIS ; 12=VARS(9)+1 ; 13=SIZE OF LABEL
C***** 14=(DIR.OF X AXIS)(CENTERED AT GRID)(CLOCKWISE TO AXIS)
C***** 15=AXIS SETTING IS NOT COMPLETE
VARS(1)=15
VARS(2)=ZM
VARS(3)=54.
VARS(4)=0.0
VARS(5)=ZM
VARS(6)=54.
VARS(7)=0.
VARS(8)=1.
VARS(9)=10.
VARS(10)=1.
VARS(11)=4.
VARS(12)=11.
VARS(13)=20.
VARS(14)=211.
VARS(15)=0.
CALL XAXIS3(VARS)

C*****VARS : 2 3 4 =X1 Y1 Z1 ; 5 6 7=X2 Y2 Z2 ; 8=USER UNIT ; 9=NO. OF INTERVALS ; 10=GRID OPTION ; 11=DRAW PARALLEL TO X-AXIS ; 12=USER UNIT ; 13=SIZE OF LABEL ; 14=DIR. OF X-AXIS ; 15=AXIS SETTING

CALL YAXIS3(VARS)

C*****TITLE3 : GRAPH3D ROUTINE ; PRINTS TITLE OF X-AXIS
C***** 4=X-AXIS ; 20=X-ALPHANUMERIC DIMENSION ; 15=CHARACTER SIZE
CALL TITLE3(4,20,15,XTITLE,O.,1.,O.)

C*****TITLE3 : GRAPH3D ROUTINE ; PRINTS Y-AXIS TITLE
C***** 3=Y-AXIS ; 18=Y-ALPHANUMERIC DIMENSION ; 15=CHARACTER SIZE
CALL TITLE3(3,24,15,YTITLE,-I.,O.,O.)

C*****GPLOT3 : GRAPH3D ROUTINE ; TO PLOT A CURVE WITH POINT OR VECTOR
C***** 4:DO NOT CALL AXIS ROUTINES ; 5=POINT PLOT ; 6:SYMBOL FREQUENCY
C***** 7:SIZE OF SYMBOL ; 8=EXACT MIN-MAX INTERVAL
IVARS(1)=8
IVARS(2)=NP
IVARS(3)=0
IVARS(4)=0
IVARS(5)=0
IVARS(6)=1
IVARS(7)=15
IVARS(8)=1
CALL GPLOT3(IVARS,X,Y)

C*****VIEW : GRAPH3D ROUTINE ; IDENTIFIES VIEWING ENVIRONMENT
C***** 1=DEFAULT VALUES FOR THREE REMAINING VARIABLES
CALL VIEW(1)

C*****WINDOW : GRAPH3D ROUTINE ; SPECIFY DIMENSION OF VIEW WINDOW
C***** 6=TOTAL NO OF VARIABLES ; 0=LOWEST OF THE PARAMETER RANGE
C*****UIN-MIN. VALUE OF NO. OF RELATIVE UNITS FROM VIEW REFERENCE
C*****UMAX-MAX. VALUE OF NO.OF RELATIVE UNITS FROM VIEW REFERENCE
C*****VMIN= " " " " " " " " " " " " " " "
C*****VMAX= " " " " " " " " " " " " " " "
CALL WINDW(6,0,-6.5,6.5,-6.5,6.5)
C*****DISPLA : GRAPH3D ROUTINE ; DISPLAYS INTERNAL BUFFER
C***** 1=OPTION TO CLEAR BUFFER
    CALL DISPLA(1)
C*****TERM : GRAPH3D ROUTINE ; REQUIRED TO CLOSE THE GRAPHICS
    CALL TERM
    STOP
    END
/* EXEC DUAL REFLECTOR */

"GRAPH3D"

SETUP FTN

"FI 19 DISK DUALREF OUT19 A1"
"FI 15 DISK DUALREF OUT15 A1"
"FI 16 DISK DUALREF OUT16 A1"
"FI 17 DISK DUALREF OUT17 A1"
"FI 18 DISK DUALREF OUT18 A1"

"LOAD DRSG(CLEAR START"
"LOAD DUALREF(CLEAR START"
"LOAD FFPLT(CLEAR START"
REFERENCES


Figure 1, Dual reflector configuration
Figure 2, Generalized dual reflector geometry
Figure 3, Eulerian angles
Figure 4a, H-plane far-field antenna pattern
Figure 4b, E-plane far-field antenna pattern
4. TITLE AND SUBTITLE

Analysis of a Generalized Dual Reflector Antenna System Using Physical Optics

6. AUTHOR(S)

Roberto J. Acosta and Alan R. Lagin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 32

13. ABSTRACT (Maximum 200 words)

Reflector antennas are widely used in communication satellite systems because they provide high gain at low cost. Offset-fed single paraboloids and dual reflector offset Cassegrain and Gregorian antennas with multiple focal region feeds provide a simple, blockage-free means of forming multiple, shaped and isolated beams with low sidelobes. Such antennas are applicable to communications satellite frequency reuse systems and earth stations requiring access to several satellites. While the single offset paraboloid has been the most extensively used configuration for the satellite multiple-beam antenna, the trend toward large apertures requiring minimum scanned beam degradation over the field of view 18 degrees for full earth coverage from geostationary orbit may lead to impractically long focal length and large feed arrays. Dual reflector antennas offer packaging advantages and more degrees of design freedom to improve beam scanning and cross-polarization properties. The Cassegrain and Gregorian antennas are the most commonly used dual reflector antennas. A computer program for calculating the secondary pattern and directivity of a generalized dual reflector antenna system has been developed and implemented at the NASA Lewis Research Center. The theoretical foundation for this program is based on the use of physical optics methodology for describing the induced currents on the sub-reflector and main reflector. The resulting induced currents on the main reflector are integrated to obtain the antenna far-zone electric fields. The computer program is verified with other physical optics programs and with measured antenna patterns. The comparison shows good agreement in far-field sidelobe reproduction and directivity.