NASA Technical Memorandum 105425

IN-32
71345
P-47

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

NASA

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 use 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):

$$E(\theta,\phi) = \frac{e^{-jkr}}{r} F(\theta,\phi) \qquad (1)$$

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).

$$F(\theta,\phi) = \theta\, UE(\theta)(a\, e^{jp} \cos\phi + b \sin\phi) + \phi\, UH(\theta)(b \cos\phi + a\, e^{jp} \sin\phi) \qquad (2)$$

where $UE(\theta)$ is the E-plane pattern, $UH(\theta)$ 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

	a	b	p
Linear x	1	0	0
Linear y	0	1	0
Right-hand circular	0.707	0.7007	+90
Left-hand circular	0.707	0.707	-90

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

$$UE(\theta) = \cos^{qe}(\theta) \tag{3a}$$

$$UH(\theta) = \cos^{qh}(\theta) \tag{3b}$$

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).

$$p_{rad} = \frac{(qe+qh+1)}{60(2qe+1)(2qh+1)} \tag{4}$$

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

$$\mathbf{J} = 2(\mathbf{n} \times \mathbf{H}^{inc}) \quad \text{illuminated region}$$

$$\mathbf{J} = 0 \quad \text{otherwise}$$

where **n** is the unit normal to the surface and **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 (xs,ys,zs) and the feed located at (xf,yf,zf) the incident fields on the sub-reflector are given

$$\mathbf{E} = \frac{e^{-jkr}}{r} \; F(xs,ys,zs) \tag{5a}$$

where F(xs,ys,zs) 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

$$\mathbf{H} = (\mathbf{r} \times \mathbf{E})/Z_0 \tag{5b}$$

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

$$\mathbf{H}(xm,ym,zm) = jk \iint (\mathbf{J} \times \mathbf{r1}) \; \frac{e^{-jkr1}}{4\pi r1} \; ds \tag{6a}$$

$$\mathbf{E}(xm,ym,zm) = -jkZ_0 \iint (\mathbf{J}-(\mathbf{J} \cdot \mathbf{r1})\mathbf{r1}) \; \frac{e^{-jkr1}}{4\pi r1} \; ds \tag{6b}$$

Where **J** is the induced current on the sub-reflector, r1 is the distance from any point in the sub-reflector to the observation point (xm,ym,zm). **r1** is a unit vector in the direction from any point in the sub-reflector to any observation point on the main reflector (xm,ym,zm).

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.

$$E(\theta,\phi) = -jkZ_0\, e^{-jkR} \iint (\mathbf{J}-(\mathbf{J}\cdot\mathbf{R})\mathbf{R}) \frac{e^{jkr}}{4\pi R}\, ds \qquad (7a)$$

$$H(\theta,\phi) = (\mathbf{R} \times \mathbf{E})/Z_0 \qquad (7b)$$

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

$$\mathbf{R} = \boldsymbol{\theta}\,(a\, e^{jp} \cos\phi + b \sin\phi) + \boldsymbol{\phi}\,(-a\, e^{jp} \sin\phi + b \cos\phi) \qquad (8a)$$

$$\mathbf{C} = \boldsymbol{\theta}\,(a\, e^{jp} \sin\phi - b \cos\phi) + \boldsymbol{\phi}\,(a\, e^{jp} \cos\phi + b \sin\phi) \qquad (8b)$$

if the secondary pattern can be expressed as

$$\mathbf{E} = \frac{e^{-jkr}}{r}\,(E_\theta(\theta,\phi) + E_\phi(\theta,\phi)) \qquad (9)$$

The reference-polarization expression is

$$E_{ref} = E \cdot (R^*)^* \tag{10a}$$

and the cross-polarization expression is

$$C_{cross} = E \cdot (C^*)^* \tag{10b}$$

The directivity for the reference polarization is defined by

$$DR(\theta,\phi) = \frac{4\pi\,(E_{ref} \cdot E_{ref}{*})\,/\,Z_0}{p_{rad}} \tag{11a}$$

similarly the directivity for the cross polarization is defined by

$$DC(\theta,\phi) = \frac{4\pi\,(E_{cross} \cdot E_{cross}{*})\,/\,Z_0}{p_{rad}} \tag{12b}$$

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 : $z = \dfrac{x^2+y^2}{4F}$

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 : $z = z_0 + a\sqrt{1+\dfrac{x^2+y^2}{b^2}}$

C: 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
DMX,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
rtempl (sub)	parameter a in wavelength
rtemp2	parameter $b=\sqrt{a^2-c^2}$
rtemp6	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
rtempl (main)	1/4F, F is focal length in wavelength

APPENDIX C

COMPUTER PROGRAM

```
      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)*1.000      )
      REAL       LAMBSQ
      PARAMETER (LAMBSQ = LAMBDA*LAMBDA       )
******************************************************************
******INPUT : QQ : FEED PATTERN EXPONENT**************************
******************************************************************
      REAL       QQ
      PARAMETER (QQ     = 62.00               )
*****************************************************************
      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)
      REAL       DMX,DMY
      REAL       INCMX,INCMY
      REAL       DSX,DSY
      REAL       INCSX,INCSY
      REAL       FREQ
      REAL       Q
************************************************************************
** 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
********************************************************************
** INITIAL CALCUALTIONS
********************************************************************
*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-1)))
      INCMY = DMY/(REAL((MAXMY-1)))
********************************************************************
******INPUT : LENGHT 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-1))
      INCSY =  (DSY/REAL(MAXSY-1))
********************************************************************
      FREQ  = FREQQ
      Q     = QQ
      CALL SUBMAI(MAXMY,MAXMX,MAXSY,MAXSX,MDTX,DATARY)
      END
********************************************************************
**********SUBROUTINE SUBMAI*****************************************
********************************************************************
      SUBROUTINE SUBMAI(PMXMY,PMXMX,PMXSY,PMXSX,PDTX,DTAARX)
      INTEGER    PMXMY
      INTEGER    PMXMX
      INTEGER    PMXSY
      INTEGER    PMXSX
      INTEGER    PDTX
      REAL       DTAARX(PDTX)
      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********************************************
*******************************************************************
      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:EXPONED 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
      INTEGER     IDXZX
      INTEGER     IDXZY
      INTEGER     IDXMSK
      INTEGER     MXMX,MXMY,TMXMX,TMXMY
      INTEGER     MXSX,MXSY,TMXSX,TMXSY
      PARAMETER (IDXZ   =  1                   )
      PARAMETER (IDXZX  =  2                   )
      PARAMETER (IDXZY  =  3                   )
      PARAMETER (IDXMSK =  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 (TMXSY  = MXSY + 1             )
************************************************************************
************************************************************************
      REAL        MAIARY(4,TMXMY,TMXMX)
      REAL        SUBARY(4,TMXSY,TMXSX)
      REAL        XYZARY(DXYZ,MXYZ)
      REAL        MRXYZ0(DXYZ),XM0,YM0,ZM0
      REAL        SRXYZ0(DXYZ),XS0,YS0,ZS0
      REAL        FEDXYZ(DXYZ),XF ,YF ,ZF
      REAL        REFXYZ(DXYZ),XR ,YR ,ZR
      REAL        DMX,DMY
```

```
      REAL       INCMX,INCMY
      REAL       DSX,DSY
      REAL       INCSX,INCSY
      REAL       FREQ
      REAL       Q
************************************************************************
** EQUIVALENCE
************************************************************************
      EQUIVALENCE (XYZARY(1,1),MRXYZ0(1))
      EQUIVALENCE (XYZARY(1,2),SRXYZ0(1))
      EQUIVALENCE (XYZARY(1,3),FEDXYZ(1))
      EQUIVALENCE (XYZARY(1,4),REFXYZ(1))
      EQUIVALENCE (MRXYZ0(1),XM0   )
      EQUIVALENCE (MRXYZ0(2),YM0   )
      EQUIVALENCE (MRXYZ0(3),ZM0   )
      EQUIVALENCE (SRXYZ0(1),XS0   )
      EQUIVALENCE (SRXYZ0(2),YS0   )
      EQUIVALENCE (SRXYZ0(3),ZS0   )
      EQUIVALENCE (FEDXYZ(1),XF    )
      EQUIVALENCE (FEDXYZ(2),YF    )
      EQUIVALENCE (FEDXYZ(3),ZF    )
      EQUIVALENCE (REFXYZ(1),XR    )
      EQUIVALENCE (REFXYZ(2),YR    )
      EQUIVALENCE (REFXYZ(3),ZR    )
************************************************************************
************************************************************************
      REAL       MAIXYZ(3),XM, YM, ZM
      REAL       SUBXYZ(3),XS, YS, ZS
      REAL       GENXYZ(3),XI, YJ, ZIJ
      REAL       RADSQ,XSQ
      REAL       CNTRX
      REAL       CNTRY
      REAL       RTEMP0
      REAL       RTEMP1
      REAL       RTEMP2
      REAL       RTEMP3
      REAL       RTEMP4
      REAL       RTEMP5
      REAL       RTEMP6
      INTEGER    I,J,V,W
************************************************************************
** EQUIVALENCE
************************************************************************
      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   )
************************************************************************
** Initialize Arrays
** MAIFIL <== MAIARY()
** SUBFIL <== SUBARY()
** XYZFIL <== XYZARY() <== MRXYZ0(),SRXYZ0(),FEDXYZ(),REFXYZ()
** DTAFIL <== DTAARX() <== DMX,DMY,INCMX,INCMY,DSX,DSY,INCSX,INCSY,FREQ,
************************************************************************
      DO 20200 I = 1,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 : LOWER LEFT CORNER OF MAIN REF GRID************************
************************************************************************
      XM0 = (-110.00000000    *LAMBDA)
      YM0 = (  79.65000000    *LAMBDA)
      ZM0 = (  1.000000000    *LAMBDA)
************************************************************************
******INPUT:LOWER LEFT CORNER OF SUB REFLECTOR GRID*********************
*****************************************************************
      XS0 = ( -41.25000000    *LAMBDA)
      YS0 = (  000.00000000    *LAMBDA)
      ZS0 = (    1.00000000    *LAMBDA)
*****************************************************************
******INPUT : FEED COORDINATES***********************************
*****************************************************************
******BORSIGHT BEAM FEED COORDINATES:
      XF   = (  0.000000000    *LAMBDA)
*     YF   = ( 0.000000000    *LAMBDA)
*     ZF   = ( 103.93560000    *LAMBDA)
*****CLEVELAND BEAM FEED COORDINATES:
      YF   = ( 5.629           * LAMBDA)
      ZF   = ( 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)
*     YR   = (  30.00000000    *LAMBDA)
      Y    =   YR*39.36
      ZR   = (  11.5*SQRT(1+(XR**2+Y**2)/1058)+97.5)
      ZR=ZR/39.36
*****************************************************************
      DMX        = DTAARX( 1)
      DMY        = DTAARX( 2)
      INCMX      = DTAARX( 3)
```

```
      INCMY      = DTAARX( 4)
      DSX        = DTAARX( 5)
      DSY        = DTAARX( 6)
      INCSX      = DTAARX( 7)
      INCSY      = DTAARX( 8)
      FREQ       = DTAARX( 9)
      Q          = DTAARX(10)
*****************************************************************
*******TO GENEARATE 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)
 806   FORMAT(5X,E15.8)
 12    CONTINUE
 11    CONTINUE
***************************************************************
** Calculate Z, Zx, Zy, Usage for SubReflector.
****Z = SUBARY(IDXZ,... SURFACE Z
****DX= SUBARY(IDXZX... DERIVATIVE WITH RESP. TO X
****DY= SUBARY(IDXZY... DERIVATIVE WITH RESP. TO Y
**************************************************************
      RTEMP0 = (    LAMBDA)
***************************************************************
******INPUT : A :PARAMETER OF SURFACE OF REVOLUTION***********
*********************************************************************
      RTEMP1 = (18.97*RTEMP0)
***************************************************************
******INPUT : 1/B**2 PARAMETER OF SURFACE OF REVOLUTION*******
***************************************************************
      RTEMP2 = (1./(.6829)                  )
****************************************************************
******INPUT : Z0 OFFSET OF THE CENTER OF THE SURFACE OFF REV.**
****************************************************************
      RTEMP6 = 160.85*RTEMP0
*********************************************************************
****INPUT: RADIUS OF CYLINDER OF SUB-REFLECTOR*********************
*********************************************************************
       FRADSQ=(19.0*RTEMP0)**2
*********************************************************************
****INPUT:  CENTER OF COORDINATES OF CYLINDER**********************
*********************************************************************
       FCENX=0.0
       FCENY=31.8*RTEMP0
*********************************************************************
      XS      = XS0 - INCSX
      DO 20700   I = 1,TMXSX,1
         XS  = XS + INCSX
         XSQ = XS*XS
         YS     = YS0 - INCSY
         DO 20600   J = 1,TMXSY,1
            YS = YS  + INCSY
      FTEMP=(XS-FCENX)**2+(YS-FCENY)**2
*     IF(FTEMP.GT.FRADSQ) GO TO 309
            RTEMP3 = ((XSQ) + (YS*YS))
            RTEMP4 = (SQRT(1+ RTEMP3*RTEMP2))
```

```
      RTEMP5 = ((RTEMP1*RTEMP2)/RTEMP4)

*     +=================================================================
*
*     | SubReflector: Hyperbola
*
*     |                          +----------------
*
*     |                         /      2   2
*
*     |   Z    = 18.97*LAMBDA X  / 1 + (X + Y) )     +160.85*LAMBDA
*
*     |                       /      ---------------
*
*     |                     \/      (    0.6829    )
*
*     |
*
*     |
*
*     |                                  +----------------
*
*     |                                 /      2   2
*
*     |                    X           /      X + Y
*
*     |   ZX   = 18.97 (-----------)  /  / 1+(-------------)
*
*     |               150.XLAMBDA    \/         0.6829
*
*     |
*
*     |                                  +---------------
*
*     |                                 /      2   2
*
*     |                    Y           /     X + Y
*
*     |   ZY   = 18.97 (-----------)  /  / 1+(-------------)
*
*     |               150.XLAMBDA    \/       0.6829
*
*     |
*
*     +=================================================================
*


*-----------------------------------------------------------------------
      SUBARY(IDXZ  ,J,I) = ((RTEMP6)+RTEMP1*RTEMP4)
      SUBARY(IDXZX ,J,I) = (XS*RTEMP5                    )
      SUBARY(IDXZY ,J,I) = (YS*RTEMP5                    )
      SUBARY(IDXMSK,J,I) = 1.
*-----------------------------------------------------------------------
20520 FORMAT (5X,4(E15.8,2X))
 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
*************************************************************************
      RTEMP0=LAMBDA
      XM    = XM0 - INCMX
*******************************************************************
*****INPUT: RADIUS OF CYLINDER MAIN REFLECTOR**********************
*******************************************************************
*     RADSQ  = (107.2*RTEMP0)**2
      RADSQ  = (108.2*RTEMP0)**2
*******************************************************************
****INPUT: CENTER OF COORDINATES OF CYLINDER***********************
*******************************************************************
      CNTRX  = 0.0
      CNTRY  = 189.7*RTEMP0
*******************************************************************
*****INPUT : 1/4F PARAMETER FOCAL LENGTH***************************
*******************************************************************
      RTEMP1 = (1./(870.00*RTEMP0))
      RTEMP2 = 2*RTEMP1
      DO 20900   I = 1,TMXMX,1
         XM     = XM + INCMX
         XSQ    = XM*XM
         YM  = YM0 - INCMY
         DO 20800   J = 1,TMXMY,1
            YM=YM+INCMY
            RTEMP4 = (((XM-CNTRX)*(XM-CNTRX))+((YM-CNTRY)*(YM-CNTRY)))
            IF (RTEMP4 .GT. RADSQ) GO TO 450
*           +============================================================
*
*           | MaiReflector: Parabola
*
*           |
*
*           |               2   2
*
*           |             ( X + Y )
*
*           |   Z    = ----------------
*
*           |           (870. X LAMBDA)
*
*           |
*
*           |                 X
*
*           |   Zx   = ----------------
*
*           |           (435. X LAMBDA)
*
*           |
*
*           |                 Y

*           |   Zy   = ----------------
*
*           |           (435. X LAMBDA)
*
```

```
*         |
*
*         +=============================================================
*

              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.
 450       WRITE(18,20520) MAIARY(IDXZ ,J,I),MAIARY(IDXZX ,J,I)
     1     ,MAIARY(IDXZY ,J,I),MAIARY(IDXMSK ,J,I)
20800   CONTINUE
20900 CONTINUE
      END
```

```fortran
      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
      INTEGER    IDXZX
      INTEGER    IDXZY
      INTEGER    IDXMSK
      INTEGER    IDXVCX
      INTEGER    IDXVCY
      INTEGER    IDXVCZ
      INTEGER    IDXRMJ
      INTEGER    IDXIMJ
      INTEGER    IDXUNM
      INTEGER    IDXJVX
      INTEGER    IDXNRM
      INTEGER    IDXDTX
      INTEGER    IDXMNM
      INTEGER    IDXMSI
      INTEGER    IDXAOT
      INTEGER    IDXPWR
      INTEGER    IDXADB
      INTEGER    IDXRDB
      INTEGER    MXMX,MXMY,TMXMX,TMXMY
      INTEGER    MXSX,MXSY,TMXSX,TMXSY
      INTEGER    MXTHE,MXPHI
      INTEGER    MXYZ
      INTEGER    DXYZ
      INTEGER    LNRY
      INTEGER    VDTX
      INTEGER    NDTX
      INTEGER    MDTX
      INTEGER    KDTX
      PARAMETER (IDXZ   = 1                   )
      PARAMETER (IDXZX  = 2                   )
      PARAMETER (IDXZY  = 3                   )
      PARAMETER (IDXMSK = 4                   )
      PARAMETER (IDXVCX = 1                   )
      PARAMETER (IDXVCY = 2                   )
      PARAMETER (IDXVCZ = 3                   )
      PARAMETER (IDXRMJ = 1                   )
      PARAMETER (IDXIMJ = 2                   )
      PARAMETER (IDXUNM = 3                   )
      PARAMETER (IDXJVX = 1                   )
```

```
      PARAMETER (IDXNRM = 2                         )
      PARAMETER (IDXDTX = 3                         )
      PARAMETER (IDXMNM = 1                         )
      PARAMETER (IDXMSI = 2                         )
      PARAMETER (IDXAOT = 3                         )
      PARAMETER (IDXPWR = 1                         )
      PARAMETER (IDXADB = 2                         )
      PARAMETER (IDXRDB = 3                         )
*******************************************************************
*****INPUT : NUMBER OF POINTS IN X AND Y MAIN REF. GRID.***********
*******************************************************************
      PARAMETER (MXMX   = 101                      )
      PARAMETER (MXMY   = 101                      )
**********************************************************************
******INPUT : NUMBER OF POINTS IN X AND Y SUB REF. GRID***************
**********************************************************************
      PARAMETER (MXSX   = 61                        )
      PARAMETER (MXSY   = 61                        )
**********************************************************************
      PARAMETER (TMXMX  = MXMX + 1                  )
      PARAMETER (TMXMY  = MXMY + 1                  )
      PARAMETER (TMXSX  = MXSX + 1                  )
      PARAMETER (TMXSY  = MXSY + 1                  )
**********************************************************************
******INPUT : NUMBER OF FAR-FIELD GRID POINTS*************************
**********************************************************************
      PARAMETER (MXTHE  = 50                        )
      PARAMETER (MXPHI  = 360                       )
**********************************************************************
      PARAMETER (MXYZ   = 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)
      REAL        DMX,DMY
      REAL        INCMX,INCMY
      REAL        DSX,DSY
      REAL        INCSX,INCSY
      INTEGER     MAXMX,MAXMY
      INTEGER     MAXSX,MAXSY
      REAL        FREQ
      REAL        Q
************************************************************************
** EQUIVALENCE
************************************************************************
      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    )
************************************************************************
** RUN     TIME VARIABLES
************************************************************************
      REAL        LAMBDA
      REAL        K
************************************************************************
** EQUIVALENCE
************************************************************************
** Input Arrays
** MAIFIL  ==> MAIARY()
** SUBFIL  ==> SUBARY()
** XYZFIL  ==> XYZARY() ==> MRXYZO(),SRXYZO(),FEDXYZ(),REFXYZ()
** DTAFIL  ==> DTAARY() ==> DMX,DMY,INCMX,INCMY,DSX,DSY,INCSX,INCSY,FREQ
************************************************************************
******READ IN THE XYZARY AND DTAARY FROM FILE GENERATOR****************
************************************************************************
        DO 10 I=1,10
        READ(15,805) DTAARY(I)
805     FORMAT(5X,E15.8)
10      CONTINUE
*****************************************************************
*******************************************************************
** Initial Calculations
******************************************************************
      LAMBDA = C/FREQ
      K      = 2*PI/LAMBDA
      MAXMX = (NINT(DMX/INCMX) + 1)
      MAXMY = (NINT(DMY/INCMY) + 1)
      MAXSX = (NINT(DSX/INCSX) + 1)
      MAXSY = (NINT(DSY/INCSY) + 1)
      CALL SUBMAI(MAXMX,MAXMY,MAXSX,MAXSY,MDTX,DTAARY)
      END
*******************************************************************
***********MAIN PROGRAM********************************************
*******************************************************************
      SUBROUTINE SUBMAI(PMXMX,PMXMY,PMXSX,PMXSY,PDTX,DTXARX)
      INTEGER     PMXMX
```

```
      INTEGER    PMXMY
      INTEGER    PMXSX
      INTEGER    PMXSY
      INTEGER    PDTX
      REAL       DTXARX(PDTX)
************************************************************************
** 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
      INTEGER    IDXZX
      INTEGER    IDXZY
      INTEGER    IDXMSK
      INTEGER    IDXVCX
      INTEGER    IDXVCY
      INTEGER    IDXVCZ
      INTEGER    IDXRMJ
      INTEGER    IDXIMJ
      INTEGER    IDXUNM
      INTEGER    IDXJVX
      INTEGER    IDXNRM
      INTEGER    IDXDTX
      INTEGER    IDXMNM
      INTEGER    IDXMSI
      INTEGER    IDXAOT
      INTEGER    IDXPWR
      INTEGER    IDXADB
      INTEGER    IDXRDB
      INTEGER    MXMX,MXMY,TMXMX,TMXMY
      INTEGER    MXSX,MXSY,TMXSX,TMXSY
      INTEGER    MXTHE,MXPHI
      INTEGER    MXYZ
      INTEGER    DXYZ
      INTEGER    LNRY
      INTEGER    VDTX
      INTEGER    NDTX
      INTEGER    MDTX
      INTEGER    KDTX
      PARAMETER (IDXZ   = 1                   )
      PARAMETER (IDXZX  = 2                   )
      PARAMETER (IDXZY  = 3                   )
      PARAMETER (IDXMSK = 4                   )
      PARAMETER (IDXVCX = 1                   )
      PARAMETER (IDXVCY = 2                   )
      PARAMETER (IDXVCZ = 3                   )
      PARAMETER (IDXRMJ = 1                   )
      PARAMETER (IDXIMJ = 2                   )
      PARAMETER (IDXUNM = 3                   )
      PARAMETER (IDXJVX = 1                   )
      PARAMETER (IDXNRM = 2                   )
      PARAMETER (IDXDTX = 3                   )
```

```
      PARAMETER (IDXMNM = 1                           )
      PARAMETER (IDXMSI = 2                           )
      PARAMETER (IDXAOT = 3                           )
      PARAMETER (IDXPWR = 1                           )
      PARAMETER (IDXADB = 2                           )
      PARAMETER (IDXRDB = 3                           )
*********************************************************************
******INPUT : NUMBER OF POINTS IN X AND Y MAIN REF. GRID*************
*********************************************************************
      PARAMETER (MXMX   = 101                        )
      PARAMETER (MXMY   = 101                        )
*********************************************************************
******INPUT : NUMBER OF POINTS IN X AND Y SUB REF. GRID**************
*********************************************************************
      PARAMETER (MXSX   = 61                          )
      PARAMETER (MXSY   = 61                          )
*********************************************************************
      PARAMETER (TMXMX   = MXMX + 1                   )
      PARAMETER (TMXMY   = MXMY + 1                   )
      PARAMETER (TMXSX   = MXSX + 1                   )
      PARAMETER (TMXSY   = MXSY + 1                   )
********************************************************************
******* INPUT : FAR-FIELD GRID POINTS THETA AND PHI*****************
**********************************************************************
      PARAMETER (MXTHE  = 50                          )
      PARAMETER (MXPHI  = 360                         )
*******************************************************************
      PARAMETER (MXYZ   = 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       MAIARY(VDTX,TMXMY,TMXMX)
```

```
      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(DXYZ),     SRX ,SRY ,SRZ
      REAL         USR(DXYZ),    USRX,USRY,USRZ
      REAL         MAGSR
      REAL         DTAARY(MDTX)
      REAL         DMX,DMY
      REAL         INCMX,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 )                                   DUA03610
      EQUIVALENCE (DTAARY( 1),DMX  )                                   DUA03620
      EQUIVALENCE (DTAARY( 2),DMY  )                                   DUA03630
      EQUIVALENCE (DTAARY( 3),INCMX)                                   DUA03640
      EQUIVALENCE (DTAARY( 4),INCMY)                                   DUA03650
      EQUIVALENCE (DTAARY( 5),DSX  )                                   DUA03660
      EQUIVALENCE (DTAARY( 6),DSY  )                                   DUA03670
      EQUIVALENCE (DTAARY( 7),INCSX)                                   DUA03680
      EQUIVALENCE (DTAARY( 8),INCSY)                                   DUA03690
      EQUIVALENCE (DTAARY( 9),FREQ )                                   DUA03700
      EQUIVALENCE (DTAARY(10),Q    )                                   DUA03710
      EQUIVALENCE (DTXARY( 1),BEGPHX)                                  DUA03720
      EQUIVALENCE (DTXARY( 2),ENDPHX)                                  DUA03730
      EQUIVALENCE (DTXARY( 3),IDXPHX)                                  DUA03740
      EQUIVALENCE (DTXARY( 4),STPPHX)                                  DUA03750
      EQUIVALENCE (DTXARY( 5),INCPHX)                                  DUA03760
      EQUIVALENCE (DTXARY( 6),BEGTHX)                                  DUA03770
      EQUIVALENCE (DTXARY( 7),ENDTHX)                                  DUA03780
      EQUIVALENCE (DTXARY( 8),IDXTHX)                                  DUA03790
      EQUIVALENCE (DTXARY( 9),STPTHX)                                  DUA03800
      EQUIVALENCE (DTXARY(10),INCTHX)                                  DUA03810
      EQUIVALENCE (DTXARY(11),PTTMNX)                                  DUA03820
      EQUIVALENCE (DTXARY(12),PTTMXX)                                  DUA03830
      EQUIVALENCE (DTXARY(13),ADBMNX)                                  DUA03840
      EQUIVALENCE (DTXARY(14),ADBMXX)                                  DUA03850
      EQUIVALENCE (DTXARY(15),RDBMNX)                                  DUA03860
      EQUIVALENCE (DTXARY(16),PRAD  )                                  DUA03870
      EQUIVALENCE (DTXARY(17),RINTNS)                                  DUA03880
      EQUIVALENCE (DTXARY(18),DIRCTV)                                  DUA03890
***********************************************************************DUA03900
** RUN     TIME VARIABLES                                              DUA03910
***********************************************************************DUA03920
      REAL        MCDARY(DXYZ,DXYZ,TMXMY,TMXMX)                        DUA03930
      REAL        SCDARY(DXYZ,DXYZ,TMXSY,TMXSX)                        DUA03940
      REAL        MAIXYZ(DXYZ),XM, YM, ZM                              DUA03950
      REAL        SUBXYZ(DXYZ),XS, YS, ZS                              DUA03960
      REAL        GENXYZ(DXYZ),XI, YJ, ZIJ                             DUA03970
      REAL        TMPXYZ(DXYZ,2)                                       DUA03980
      REAL        TMRXYZ(DXYZ),TMRX,TMRY,TMRZ                          DUA03990
      REAL        TMIXYZ(DXYZ),TMIX,TMIY,TMIZ                          DUA04000
      REAL        HFLD(DXYZ,2)                                         DUA04010
      REAL        HVR (DXYZ),HVRX,HVRY,HVRZ                            DUA04020
      REAL        HVI (DXYZ),HVIX,HVIY,HVIZ                            DUA04030
      REAL        JFLD(DXYZ,2)                                         DUA04040
      REAL        JVR (DXYZ),JVRX,JVRY,JVRZ                            DUA04050
      REAL        JVI (DXYZ),JVIX,JVIY,JVIZ                            DUA04060
      REAL        SUM (DXYZ,2)                                         DUA04070
      REAL        SUMR(DXYZ),SUMRX,SUMRY,SUMRZ                         DUA04080
      REAL        SUMI(DXYZ),SUMIX,SUMIY,SUMIZ                         DUA04090
      REAL        NORM(DXYZ),NX ,NY ,NZ                                DUA04100
      REAL        MAGNRM                                               DUA04110
      REAL        SI  (DXYZ),SIX ,SIY ,SIZ                             DUA04120
      REAL        USI (DXYZ),USIX,USIY,USIZ                            DUA04130
      REAL        MAGSI                                                DUA04140
      REAL        PV  (DXYZ),PVX ,PVY ,PVZ                             DUA04150
      REAL        UPV (DXYZ),UPVX,UPVY,UPVZ                            DUA04160
      REAL        MAGPV                                                DUA04170
      REAL        HV  (DXYZ),HVX ,HVY ,HVZ                             DUA04180
      REAL        UHV (DXYZ),UHVX,UHVY,UHVZ                            DUA04190
      REAL        MAGHV                                                DUA04200
```

```
      REAL      JV (DXYZ),JVX ,JVY ,JVZ                                DUA04210
      REAL      UJV(DXYZ),UJVX,UJVY,UJVZ                               DUA04220
      REAL      MAGJV                                                  DUA04230
      REAL      R1 (DXYZ),R1X ,R1Y ,R1Z                                DUA04240
      REAL      UR1(DXYZ),UR1X,UR1Y,UR1Z                               DUA04250
      REAL      MAGR1                                                  DUA04260
      REAL      INTG(DXYZ),INTX,INTY,INTZ                              DUA04270
      REAL      MAGINT                                                 DUA04280
      REAL      RFF (DXYZ),RFFX,RFFY,RFFZ                              DUA04290
      REAL      PTTRN(NDTX,0:MXTHE,0:MXPHI)                            DUA04300
      REAL      PTTMIN                                                 DUA04310
      REAL      PTTMAX                                                 DUA04320
      REAL      LAMBDA                                                 DUA04330
      REAL      K                                                      DUA04340
      REAL      KR                                                     DUA04350
      REAL      RR                                                     DUA04360
      REAL      PSI(2),COSKR,SINKR                                     DUA04370
      REAL      MIN                                                    DUA04380
      REAL      MAX                                                    DUA04390
      REAL      R1TMP                                                  DUA04400
      REAL      R2TMP                                                  DUA04410
      REAL      CMPTMP(2),CMPTMR,CMPTMI                                DUA04420
      REAL      SCALE                                                  DUA04430
      REAL      ANGLE                                                  DUA04440
      INTEGER   ANGPHX                                                 DUA04450
      REAL      ANGPHI                                                 DUA04460
      REAL      BEGPHI                                                 DUA04470
      REAL      ENDPHI                                                 DUA04480
      INTEGER   IDXPHI                                                 DUA04490
      INTEGER   STPPHI                                                 DUA04500
      REAL      INCPHI                                                 DUA04510
      REAL      SINPHI                                                 DUA04520
      REAL      COSPHI                                                 DUA04530
      INTEGER   ANGTHX                                                 DUA04540
      REAL      ANGTHE                                                 DUA04550
      REAL      BEGTHE                                                 DUA04560
      REAL      ENDTHE                                                 DUA04570
      INTEGER   IDXTHE                                                 DUA04580
      INTEGER   STPTHE                                                 DUA04590
      REAL      INCTHE                                                 DUA04600
      REAL      SINTHE                                                 DUA04610
      REAL      COSTHE                                                 DUA04620
      REAL      EPHRE                                                  DUA04630
      REAL      EPHIM                                                  DUA04640
      REAL      ETHRE                                                  DUA04650
      REAL      ETHIM                                                  DUA04660
      REAL      COZ                                                    DUA04670
      REAL      MSKFAC                                                 DUA04680
      INTEGER   MAXMX,MAXMY                                            DUA04690
      INTEGER   MAXSX,MAXSY                                            DUA04700
      REAL      DS                                                     DUA04710
      REAL      INCMXY                                                 DUA04720
      REAL      INCSXY                                                 DUA04730
      INTEGER   I,J,IP,JP,V,W                                          DUA04740
      INTEGER   IOS                                                    DUA04750
      INTEGER   FLG                                                    DUA04760
      INTEGER   ITMP                                                   DUA04770
      CHARACTER TIME*8                                                 DUA04780
      REAL AAAKR                                                       DUA04790
      INTEGER IIIKR                                                    DUA04800
```

```
************************************************************************DUA04810
** EQUIVALENCE                                                           DUA04820
************************************************************************DUA04830
      EQUIVALENCE (SUBXYZ(1),XS    )                                      DUA04840
      EQUIVALENCE (SUBXYZ(2),YS    )                                      DUA04850
      EQUIVALENCE (SUBXYZ(3),ZS    )                                      DUA04860
      EQUIVALENCE (MAIXYZ(1),XM    )                                      DUA04870
      EQUIVALENCE (MAIXYZ(2),YM    )                                      DUA04880
      EQUIVALENCE (MAIXYZ(3),ZM    )                                      DUA04890
      EQUIVALENCE (GENXYZ(1),XI    )                                      DUA04900
      EQUIVALENCE (GENXYZ(2),YJ    )                                      DUA04910
      EQUIVALENCE (GENXYZ(3),ZIJ   )                                      DUA04920
      EQUIVALENCE (TMPXYZ(1,1),TMRXYZ(1))                                 DUA04930
      EQUIVALENCE (TMPXYZ(1,2),TMIXYZ(1))                                 DUA04940
      EQUIVALENCE (TMRXYZ(1),TMRX  )                                      DUA04950
      EQUIVALENCE (TMRXYZ(2),TMRY  )                                      DUA04960
      EQUIVALENCE (TMRXYZ(3),TMRZ  )                                      DUA04970
      EQUIVALENCE (TMIXYZ(1),TMIX  )                                      DUA04980
      EQUIVALENCE (TMIXYZ(2),TMIY  )                                      DUA04990
      EQUIVALENCE (TMIXYZ(3),TMIZ  )                                      DUA05000
      EQUIVALENCE (SUM(1,1),SUMR(1))                                      DUA05010
      EQUIVALENCE (SUM(1,2),SUMI(1))                                      DUA05020
      EQUIVALENCE (SUMR (1),SUMRX  )                                      DUA05030
      EQUIVALENCE (SUMR (2),SUMRY  )                                      DUA05040
      EQUIVALENCE (SUMR (3),SUMRZ  )                                      DUA05050
      EQUIVALENCE (SUMI (1),SUMIX  )                                      DUA05060
      EQUIVALENCE (SUMI (2),SUMIY  )                                      DUA05070
      EQUIVALENCE (SUMI (3),SUMIZ  )                                      DUA05080
      EQUIVALENCE (NORM (1),NX     )                                      DUA05090
      EQUIVALENCE (NORM (2),NY     )                                      DUA05100
      EQUIVALENCE (NORM (3),NZ     )                                      DUA05110
      EQUIVALENCE (SI   (1),SIX    )                                      DUA05120
      EQUIVALENCE (SI   (2),SIY    )                                      DUA05130
      EQUIVALENCE (SI   (3),SIZ    )                                      DUA05140
      EQUIVALENCE (USI  (1),USIX   )                                      DUA05150
      EQUIVALENCE (USI  (2),USIY   )                                      DUA05160
      EQUIVALENCE (USI  (3),USIZ   )                                      DUA05170
      EQUIVALENCE (PV   (1),PVX    )                                      DUA05180
      EQUIVALENCE (PV   (2),PVY    )                                      DUA05190
      EQUIVALENCE (PV   (3),PVZ    )                                      DUA05200
      EQUIVALENCE (UPV  (1),UPVX   )                                      DUA05210
      EQUIVALENCE (UPV  (2),UPVY   )                                      DUA05220
      EQUIVALENCE (UPV  (3),UPVZ   )                                      DUA05230
      EQUIVALENCE (HFLD(1,1),HVR(1))                                      DUA05240
      EQUIVALENCE (HFLD(1,2),HVI(1))                                      DUA05250
      EQUIVALENCE (HVR    (1),HVRX )                                      DUA05260
      EQUIVALENCE (HVR    (2),HVRY )                                      DUA05270
      EQUIVALENCE (HVR    (3),HVRZ )                                      DUA05280
      EQUIVALENCE (HVI    (1),HVIX )                                      DUA05290
      EQUIVALENCE (HVI    (2),HVIY )                                      DUA05300
      EQUIVALENCE (HVI    (3),HVIZ )                                      DUA05310
      EQUIVALENCE (HV     (1),HVX  )                                      DUA05320
      EQUIVALENCE (HV     (2),HVY  )                                      DUA05330
      EQUIVALENCE (HV     (3),HVZ  )                                      DUA05340
      EQUIVALENCE (UHV    (1),UHVX )                                      DUA05350
      EQUIVALENCE (UHV    (2),UHVY )                                      DUA05360
      EQUIVALENCE (UHV    (3),UHVZ )                                      DUA05370
      EQUIVALENCE (JFLD(1,1),JVR(1))                                      DUA05380
      EQUIVALENCE (JFLD(1,2),JVI(1))                                      DUA05390
      EQUIVALENCE (JVR    (1),JVRX )                                      DUA05400
```

```
      EQUIVALENCE (JVR   (2),JVRY )
      EQUIVALENCE (JVR   (3),JVRZ )
      EQUIVALENCE (JVI   (1),JVIX )
      EQUIVALENCE (JVI   (2),JVIY )
      EQUIVALENCE (JVI   (3),JVIZ )
      EQUIVALENCE (JV    (1),JVX  )
      EQUIVALENCE (JV    (2),JVY  )
      EQUIVALENCE (JV    (3),JVZ  )
      EQUIVALENCE (UJV   (1),UJVX )
      EQUIVALENCE (UJV   (2),UJVY )
      EQUIVALENCE (UJV   (3),UJVZ )
      EQUIVALENCE (R1    (1),R1X  )
      EQUIVALENCE (R1    (2),R1Y  )
      EQUIVALENCE (R1    (3),R1Z  )
      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 (CMPTMP(1),CMPTMR)
      EQUIVALENCE (CMPTMP(2),CMPTMI)
************************************************************************
** INITIALIZE INPUT ARRAYS
** MAIFIL ==> MAIARY()
** SUBFIL ==> SUBARY()
** XYZFIL ==> XYZARY() ==> MRXYZO(),SRXYZO(),FEDXYZ(),REFXYZ()
** DTAFIL ==> DTAARY() ==> DMX,DMY,INCMX,INCMY,DSX,DSY,INCSX,INCSY,FREQ,
************************************************************************
** GENERATE   OUTPUT ARRRAYS
** MCDFIL <== MCDARY()
** SCDFIL <== SCDARY()
** RFXFIL <== PTTRN ()
** DTXFIL <== DTXARY() <==  BEG,END,IDX,STP,INC * THE,PHI , PTT * ADR *
************************************************************************
******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)
806       FORMAT(5X,E15.8)
12        CONTINUE
11        CONTINUE
************************************************************************
*****************************************************************
          DO 13 III=1,TMXSX,1
          DO 14 JJJ=1,TMXSY,1
          READ(17,807)SUBARY(IDXZ,JJJ,III),
     1                SUBARY(IDXZX,JJJ,III),
     2                SUBARY(IDXZY,JJJ,III),
     3                SUBARY(IDXMSK,JJJ,III)
807       FORMAT(5X,4(E15.8,2X))
14        CONTINUE
13        CONTINUE
          DO 15 IIII=1,TMXMX,1
```

```
        DO 16 JJJJ=1,TMXMY,1
        READ(18,807)MAIARY(IDXZ,JJJJ,IIII),
     1              MAIARY(IDXZX,JJJJ,IIII),
     2              MAIARY(IDXZY,JJJJ,IIII),
     3              MAIARY(IDXMSK,JJJJ,IIII)
16          CONTINUE
15          CONTINUE
************************************************************************
** INITIAL CALCULATIONS
************************************************************************
      DO 00100   I = 1,PDTX,1
         DTAARY(I) = DTXARX(I)
00100 CONTINUE
      LAMBDA = C/FREQ
      K      = 2*PI/LAMBDA
      MAXMX = PMXMX
      MAXMY = PMXMY
      MAXSX = PMXSX
      MAXSY = PMXSY
      MAGSR  = 0
      DO 00200   V = 1,3,1
         R1TMP = REFXYZ(V) - FEDXYZ(V)
         SR(V) = R1TMP
         MAGSR = MAGSR + R1TMP*R1TMP
00200 CONTINUE
      MAGSR=SQRT(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
      INCPHI = (ENDPHI-BEGPHI)/IDXPHI
*     INCPHI = 0
**********************************************************************
      BEGTHE = 0.
      ENDTHE = 5.*PI/180
      IDXTHE = 50
      STPTHE = 1
      INCTHE = (ENDTHE-BEGTHE)/IDXTHE
************************************************************************
** Calculate Current Densities on the SubReflector Resulting from Source
************************************************************************
      INCSXY =   INCSX*INCSY
      XS = XS0 - INCSX
      DO 00700   I = 1,MAXSX,1
         XS = XS + INCSX
         YS = YS0 - 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
```

```
              MAGSI = 0.
              DO 00500   V = 1,3,1
                R1TMP = SUBXYZ(V) - FEDXYZ(V)
                SI(V) = R1TMP
                MAGSI = MAGSI + R1TMP*R1TMP
00500         CONTINUE
              MAGSI=SQRT(MAGSI)
      IF (MAGSI .EQ. 0.) THEN
      STOP
      ENDIF
              DO 00510   V = 1,3,1
                 USI(V) = SI(V)/MAGSI
00510         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
00520         CONTINUE
*             COSTHE = DOT(USR,UPV,3)
              SCALE  = FDPTRN(USR,USI,Q,COZ,FLG)/ETA
              IF (     (FLG   .EQ. 1)
     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
00530         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
                 R1TMP          = NORM(V)/MAGNRM
                 NORM(V)        = R1TMP
                 SCDARY(V,IDXNRM,J,I) = R1TMP
00540         CONTINUE
              CALL CROSS(JV,NORM,UHV)
*             MAGJV = SQRT(JVX*JVX + JVY*JVY + JVZ*JVZ)
              MAGJV = 1.
              DO 00550   V = 1,3,1
                 R1TMP               = JV(V)/MAGJV
                 UJV(V)            = R1TMP
                 SCDARY(V,IDXJVX,J,I) = R1TMP
00550         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(IDXVCZ,IDXIMJ) = -(SCALE*UJV(IDXVCZ)*SINKR)
            ELSE
          ENDIF
00599     CONTINUE
00600     CONTINUE
00700 CONTINUE
***********************************************************************
** Calculate Induced Magnetic Field on Main_Reflector by Sub_Reflector.
***********************************************************************
      INCMXY = INCMX*INCMY
      XM     = XM0 - INCMX
      DO 01900   IP = 1,MAXMX,1
        XM = XM + INCMX
        YM     = YM0 - INCMY
        DO 01800   JP = 1,MAXMY,1
          YM  = YM + INCMY
          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) = 0.
01000         CONTINUE
01100       CONTINUE
            XS     = XS0 - INCSX
            DO 01700   I = 1,MAXSX,1
              XS = XS + INCSX
              YS     = YS0 - 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(IDXMNM,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
01200             CONTINUE
                  MAGR1 = SQRT(MAGR1)
                  DO 01300   V = 1,3,1
                    UR1(V)  = R1(V)/MAGR1
                    UHV(V)  = SCDARY(V,IDXJVX,J,I)
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),UR1)
                  CALL CROSS(INTG,UHV,UR1)
```

```
                      MAGINT = 1.
*     0               MAGINT = SQRT(( (INTX*INTX)
*     1                              +(INTY*INTY)
*     2                              +(INTZ*INTZ)))
                      INTX = INTX/MAGINT
                      INTY = INTY/MAGINT
                      INTZ = INTZ/MAGINT
                      DO 01500   V = 1,3,1
                         DO 01400   W = 1,2,1
      0                     HFLD(V,W) = HFLD(V,W)
      1                       +SCALE*PSI(W)*INTG(V)*DS
01400                    CONTINUE
01500                 CONTINUE
                   ENDIF
01599           CONTINUE
01600           CONTINUE
01700        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)
01710           CONTINUE
01720        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
************************************************************************
01799    CONTINUE
01800    CONTINUE
01900 CONTINUE
************************************************************************
*** 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   ANGTHX = 0,IDXTHE,STPTHE
            ANGTHE = 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.
0         DS = INCMXY*SQRT(NORM(IDXVCX)*NORM(IDXVCX)
2                        +NORM(IDXVCY)*NORM(IDXVCY)
3                        +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
                     TMIX = CMPTMP(IDXIMJ)*RFFX
                     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
                    TMIX = MCDARY(IDXVCX,IDXIMJ,JP,IP) - TMIX
                    TMIY = MCDARY(IDXVCY,IDXIMJ,JP,IP) - TMIY
                    TMIZ = MCDARY(IDXVCZ,IDXIMJ,JP,IP) - TMIZ
                     TTMRX = ((TMRX*COSKR)-(TMIX*SINKR))
                     TTMRY = ((TMRY*COSKR)-(TMIY*SINKR))
                     TTMRZ = ((TMRZ*COSKR)-(TMIZ*SINKR))
                     TTMIX = ((TMRX*SINKR)+(TMIX*COSKR))
                     TTMIY = ((TMRY*SINKR)+(TMIY*COSKR))
                     TTMIZ = ((TMRZ*SINKR)+(TMIZ*COSKR))
                      JVRX = TTMRX*DS
                      JVRY = TTMRY*DS
                      JVRZ = TTMRZ*DS
                      JVIX = TTMIX*DS
                      JVIY = TTMIY*DS
                      JVIZ = TTMIZ*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
    1                     + SUMRY*COSTHE*SINPHI
    2                     - SUMRZ*SINTHE
    0             ETHIM =   SUMIX*COSTHE*COSPHI
    1                     + SUMIY*COSTHE*SINPHI
    2                     - SUMIZ*SINTHE
    0             EPHRE = - SUMRX*SINPHI
    1                     + SUMRY*COSPHI
    0             EPHIM = - SUMIX*SINPHI
    1                     + 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            R1TMP =   ETHRE*ETHRE + ETHIM*ETHIM
*    1                    + EPHRE*EPHRE + EPHIM*EPHIM
***************PLOT THE CO POL FIELDS********************************
                  R1TMP = AMGREF
***************PLOT THE CROSS POL FIELDS*****************************
*                 R1TMP = AMGCRP
***********************************************************************
                  PTTRN(IDXPWR,ANGTHX,ANGPHX) = R1TMP
                  IF (R1TMP .LT. PTTMIN) THEN
                     PTTMIN = R1TMP
                  ENDIF
                  IF (R1TMP .GT. PTTMAX) THEN
                     PTTMAX = R1TMP
                  ENDIF
02500    CONTINUE
02600 CONTINUE
      PRAD   = ((2.*PI)/(ETA*((2.*Q)+1)))
      RIFCTR = (((K*K*K*K)*ETA)/(256.*(PI*PI*PI*PI)))
      RINTNS = (RIFCTR*PTTMAX)
      DIRCTV = ((4.*PI*RINTNS)/(PRAD))
      DIRCTV = (10*ALOG10(DIRCTV))
***********************************************************************
      WRITE(19,897)DIRCTV
897   FORMAT(5X,F15.5)
*************************************************************************
      ADBMNX = 10.*ALOG10(PTTMIN)
      ADBMXX = 10.*ALOG10(PTTMAX)
      RDBMNX = (ADBMNX - ADBMXX)
      ANGPHI = BEGPHI - INCPHI
      DO 02800   ANGPHX = 0,IDXPHI,STPPHI
         ANGPHI = ANGPHI + INCPHI
```

```
        ANGTHE = BEGTHE - INCTHE
        DO 02700   ANGTHX = 0,IDXTHE,STPTHE
           ANGTHE = ANGTHE + INCTHE
           R1TMP = 10.*ALOG10((PTTRN(IDXPWR,ANGTHX,ANGPHX)))
           PTTRN(IDXADB,ANGTHX,ANGPHX) = (R1TMP          )
           PTTRN(IDXRDB,ANGTHX,ANGPHX) = (R1TMP - ADBMXX)
*======================================================================
*******GENERATE THE OUTPUT ARRAYS***************************************
************************************************************************
       ZETA=ANGTHE*180/PI
       APHI=ANGPHI*180/PI
        WRITE(19,895)PTTRN(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) = INCPHI
      DTXARY( 6) = BEGTHE
      DTXARY( 7) = ENDTHE
      DTXARY( 8) = REAL(IDXTHE)
      DTXARY( 9) = REAL(STPTHE)
      DTXARY(10) = INCTHE
      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()    ! Returns Real Value of DOT PRODUCT A and B
************************************************************************
      REAL FUNCTION DOT(A,B,N)
      INTEGER   N
      REAL      A(N)
      REAL      B(N)
         INTEGER   I
```

```
         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(10000),Y(10000),VARS(20)
      DIMENSION XPL(1000),YPL(1000)
      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 GPLOT3
      INTEGER * 4 IAXIS /0/
      INTEGER * 4 IVARS(20)
      INTEGER * 4 NUM/10000/
      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)
******2(NFF+1)
       NP=102
       ZM=2.

      DO 15 J=1,NP
C*****X : ANGLE POSITIONS(DEG.) ; Y: RELATIVE FAR FLD. AMPLITUDES(DB)
      READ(19,300)YPL(J),XPL(J),DUM1
300   FORMAT(5X,F15.8,3X,2(F15.8,3X))
15    CONTINUE
      DO 98 J=1,51
      Y(J)=(YPL(103-J))
      X(J)=-XPL(103-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 ; 0 :CARTESIAN ; (-ZM,ZM) :(XMIN,XMAX)
C*****(-80.,0) :(YMIN,YMAX) ; (0.,0.) :(ZMIN,ZMAX)
      CALL UXTRM(8,0,-ZM,ZM,-54.,0.,0.0,0.)
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
C*****INTERVALS ; 10=GRID OPTION ; 11=DRAW PARALLEL TO X-AXIS ; 12=
C*****VARS(9)+1 ; 13=SIZE OF LABEL ; 14=DIR. OF X-AXIS ; 15=AXIS SETTING
C*****COMPLETE.
      VARS(2)=-ZM
      VARS(3)=-54.
      VARS(4)=0.
      VARS(5)=-ZM
      VARS(6)=0.
      VARS(7)=0.
      VARS(8)=1.
      VARS(9)=9.
      VARS(10)=1.
      VARS(11)=3.
      VARS(12)=10.
      VARS(13)=20.
      VARS(14)=212.
      VARS(15)=1.
      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,0.,1.,0.)
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,-1.,0.,0.)
C*****GPLOT3 : GRAPH3D ROUTINE ; TO PLOT A CURVE WITH POINT OR VECTOR
C*****IVARS : 1=DIMENSION OF IVARS ; 2=NO. OF POINTS,EXACT ; 3=NO Z-AXIS
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)
      CALL CHARS3(13,CH,3.,10.5,0.,25,1.)
      CALL NUMBER(4,DDIR,5,2,DIR)
      CALL CHARS3(5,DIR,7.,10.5,0.,25,1.)
      CALL CHARS3(2,ADB,9.,10.5,0.,25,1.)
C*****GVIEW : GRAPH3D ROUTINE ; IDENTIFIES VIEWING ENVIRONMENT
C***** 1=DEFAULT VALUES FOR THREE REMAINING VARIABLES
      CALL GVIEW(1)
C*****WINDW : GRAPH3D ROUTINE ; SPECIFY DIMENSION OF VIEW WINDOW
C***** 6=TOTAL NO OF VARIABLES ; 0=LOWEST OF THE PARAMETER RANGE
C*****UMIN=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)                              FFP01200
C*****DISPLA : GRAPH3D ROUTINE ; DISPLAYS INTERNAL BUFFER            FFP01210
C***** 1=OPTION TO CLEAR BUFFER                                      FFP01220
      CALL DISPLA(1)                                                 FFP01230
C*****TERM : GRAPH3D ROUTINE ; REQUIRED TO CLOSE THE GRAPHICS        FFP01240
      CALL TERM                                                      FFP01250
      STOP                                                           FFP01260
      END                                                            FFP01270
```

```
/*  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 FFPLOT(CLEAR START"
```

REFERENCES

1. Silver, Samuel: Microwave Antenna Theory and Design. McGraw-hill Book Co., Inc., 1949, pp. 158-162

2. Rahmat-Samii, Yahya: Useful Coordinate Transformation for Antenna Applications. IEEE A.P., vol. AP-27, no. 4, July 1979, pp. 571-574.

3. Ludwig, Arthur C.: The Definition of Cross Polarization. IEEE A.P., vol. AP-21, Jan. 1973, pp. 116-119.

4. Lam, Peter T.;Lee, Shung-Wu and Acosta, Roberto: Secondary Pattern Computation of Arbitrarily Shaped Main Reflector. Electromagnetic Laboratory, University of Illinois Scientific Report 84-7; April 1984.

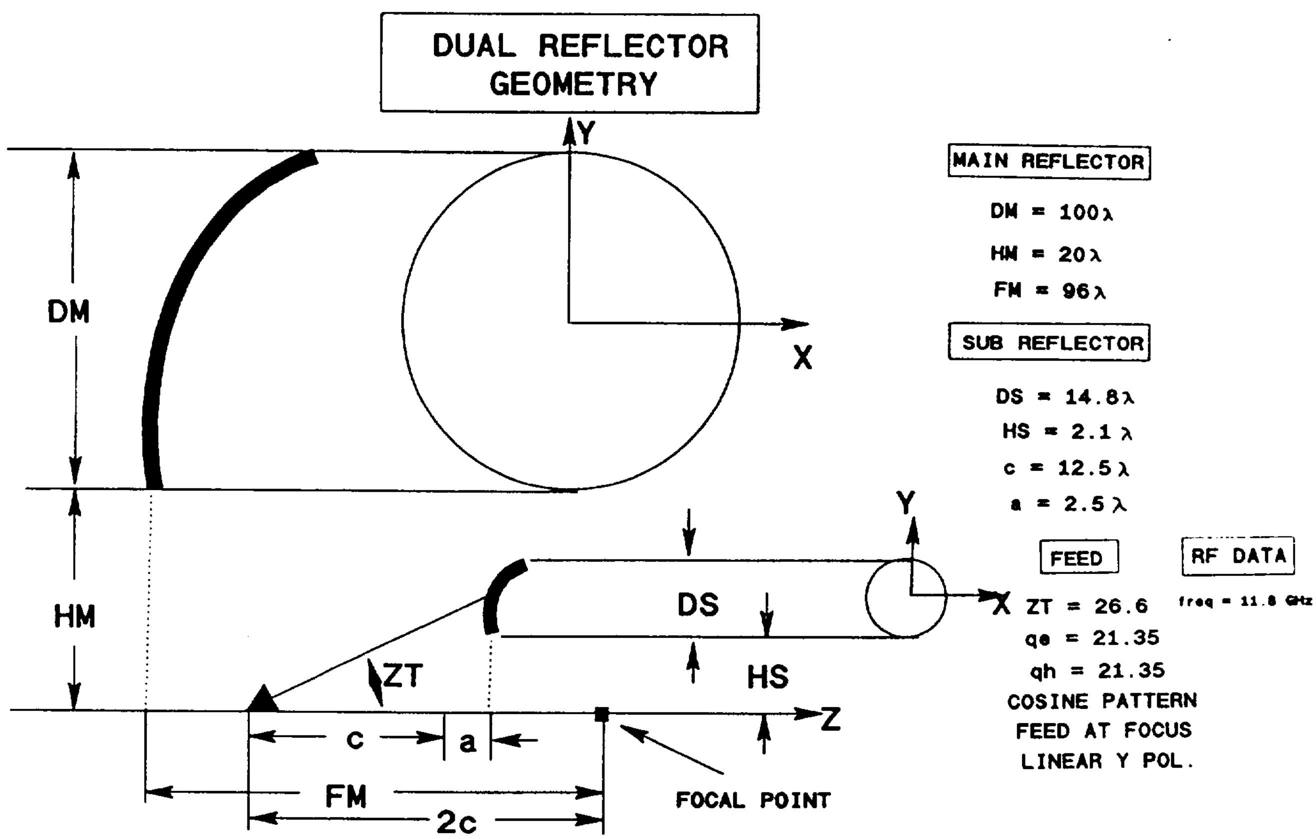

Figure 1, Dual reflector configuration

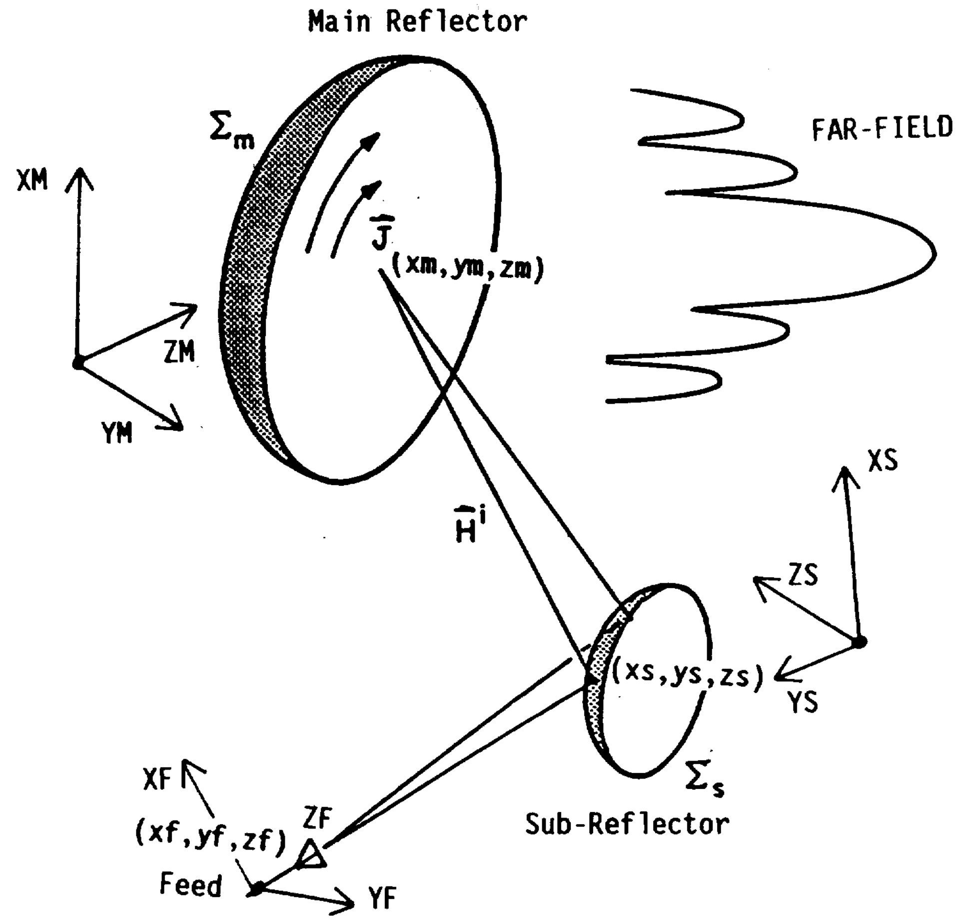

Figure 2, Generalized dual reflector geometry

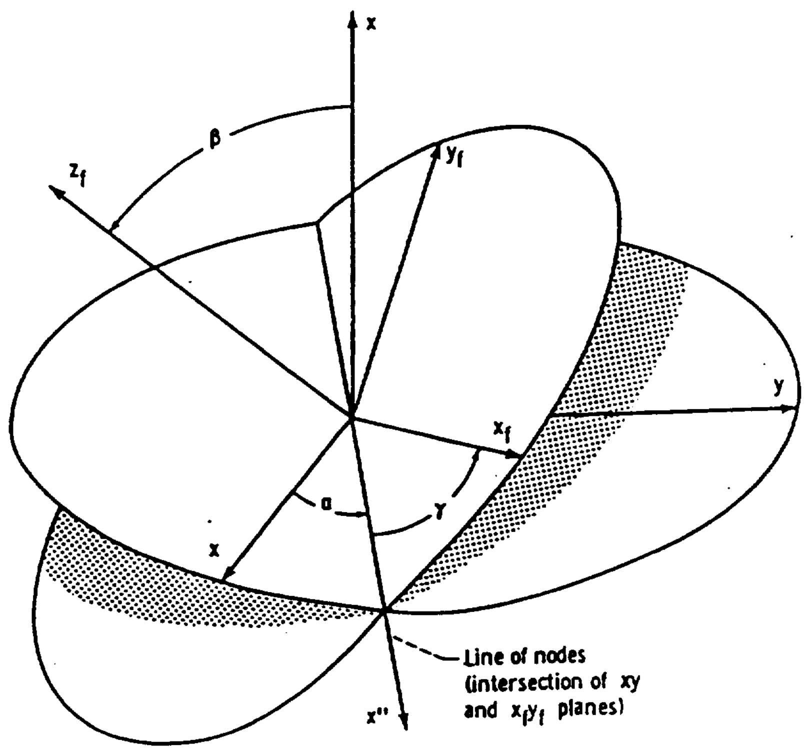

Figure 3, Eulerian angles

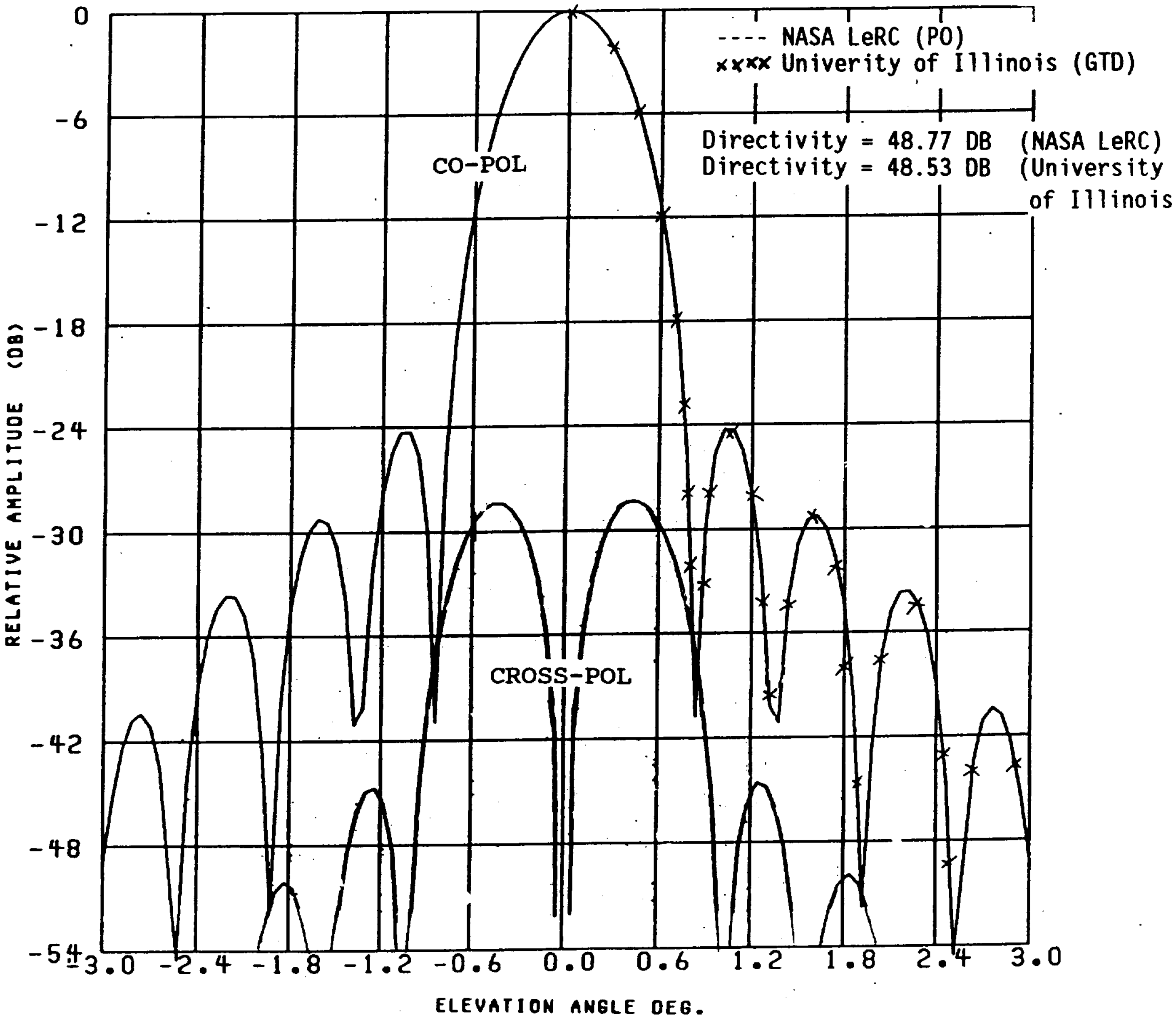

Figure 4a, H-plane far-field antenna pattern

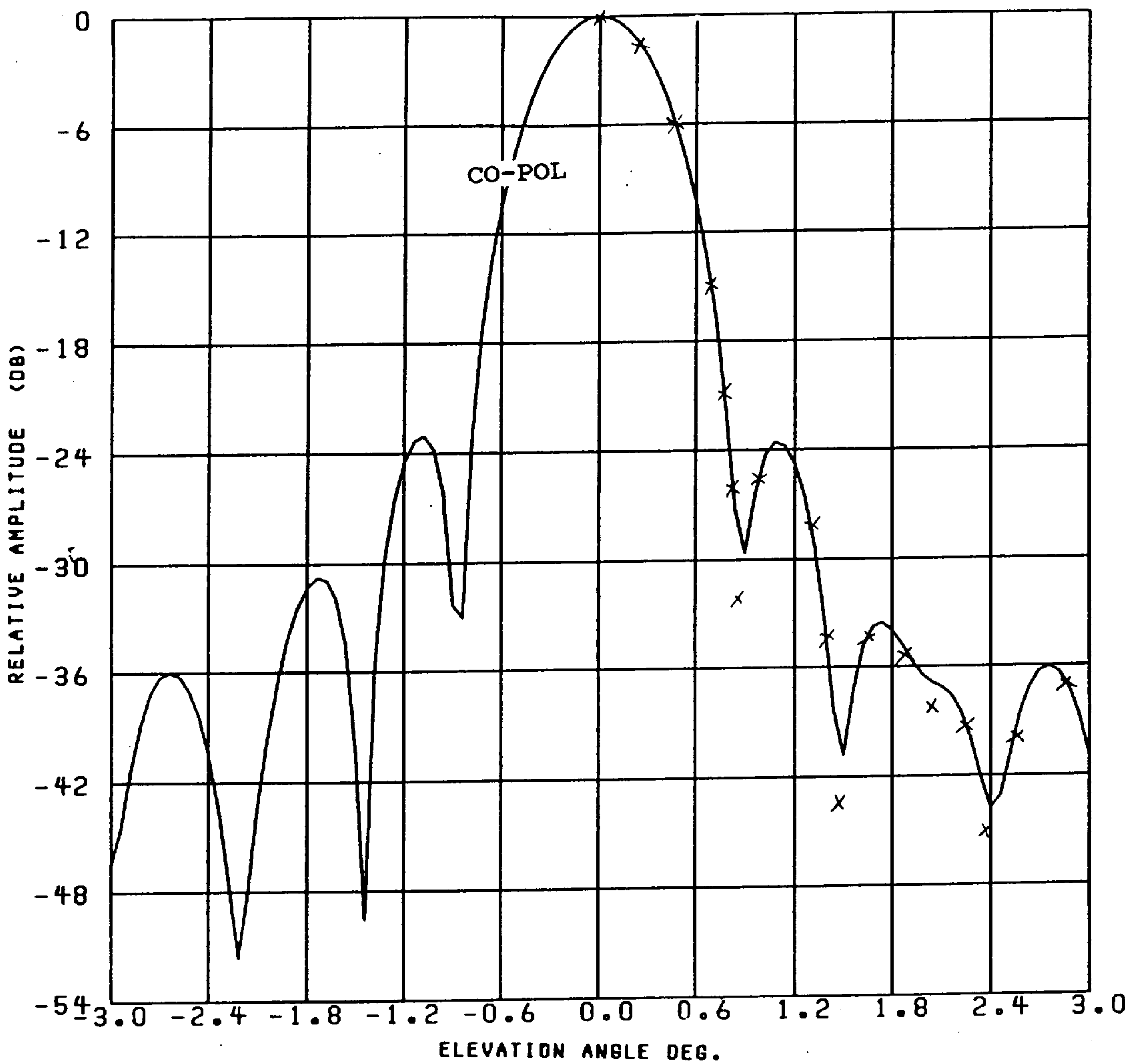

Figure 4b, E-plane far-field antenna pattern

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1992	Technical Memorandum

4. TITLE AND SUBTITLE

Analysis of a Generalized Dual Reflector Antenna System Using Physical Optics

5. FUNDING NUMBERS

WU-679-40-00

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

8. PERFORMING ORGANIZATION REPORT NUMBER

E-6842

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

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

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-105425

11. SUPPLEMENTARY NOTES

Responsible person, Roberto J. Acosta, (216) 433-6640.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 32

12b. DISTRIBUTION CODE

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.

14. SUBJECT TERMS

Reflector antennas; Multibeam antenna; Reflector analysis; Physical optics

15. NUMBER OF PAGES

46

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102