

Reference

NBS  
Publi-  
cations

NAT'L INST. OF STAND & TECH R.I.C.



A11105 036983

NBSIR 80-1630

~~A11101-729021~~

# DETERMINATION OF MUTUAL COUPLING BETWEEN CO-SITED MICROWAVE ANTENNAS AND CALCULATION OF NEAR-ZONE ELECTRIC FIELD

---

C.F. Stubenrauch  
A.D. Yaghjian

Electromagnetic Fields Division  
National Engineering Laboratory  
National Bureau of Standards  
Boulder, Colorado 80303

QC  
100  
.U56  
80-1630  
1981

June 1981



NBSIR 80-1630

NATIONAL BUREAU  
OF STANDARDS  
LIBRARY

AUG 7 1981

# DETERMINATION OF MUTUAL COUPLING BETWEEN CO-SITED MICROWAVE ANTENNAS AND CALCULATION OF NEAR-ZONE ELECTRIC FIELD

---

C.F. Stubenrauch  
A.D. Yaghjian

Electromagnetic Fields Division  
National Engineering Laboratory  
National Bureau of Standards  
Boulder, Colorado 80303

June 1981



---

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director



TABLE OF CONTENTS

	Page
INTRODUCTION.....	1
1. FORMULATION OF THE MUTUAL COUPLING BETWEEN TWO ANTENNAS.....	3
1.1 The Basic Coupling Formula (Transmission Integral).....	4
1.1.1 The Plane-Wave Scattering Matrix Approach.....	4
1.1.2 The Coupling Quotient in Terms of Far Field of Each Antenna.....	6
1.1.3 Coupling Quotient When the Roles of Transmitting and Receiving Are Exchanged.....	8
1.2 Eulerian Angle Transformations Describing the Arbitrary Orientation of the Antennas.....	9
1.2.1 Rotational Transformations from $(k_x, k_y)$ to the Far-Field Direction in the Fixed Coordinate System of Each Antenna.....	9
1.2.2 Vector Component Transformations Required to Compute the Coupling Dot Product.....	12
1.3 The Sampling Theorem, Limits of Integration, and Fast Fourier Transform.....	15
1.3.1 The Point Spacing of $k_x$ and $k_y$ Required by the Sampling Theorem.....	15
1.3.2 The Limits of Integration and Number of Points Required.....	16
1.3.3 Application of the Fast Fourier Transform.....	19
1.4 Preliminary Numerical Results.....	20
2. TRANSFORMATION FROM FAR FIELD TO NEAR FIELD.....	25
2.1 Relationship of Near-Field Intensities to Power Input and Antenna Gain or Efficiency.....	26
3. PHYSICAL OPTICS MODEL FOR REFLECTOR ANTENNAS.....	28
3.1 Physical Optics Subroutines Employed by USC.....	30
3.2 Test of Near-Field Program.....	31
4. COMPARISON OF PHYSICAL OPTICS AND MEASURED FAR FIELDS.....	34
5. COMPARISON OF PREDICTED AND MEASURED NEAR-FIELD COUPLING.....	50
6. CONCLUSIONS AND RECOMMENDATIONS.....	53
ACKNOWLEDGMENT.....	57
REFERENCES.....	58
APPENDIX A. POMODL - PHYSICAL OPTICS ANTENNA MODEL.....	59
A.1 GENERAL OVERVIEW OF COMPUTER PROGRAM.....	59
A.1.1 PROGRAM POMODL.....	61
A.1.2 SUBROUTINE FAR2D(EPL,HPL,EY,NTHETA,NPHI,DATA,IR2X2,IC2TON).....	69
A.1.3 SUBROUTINE FFKXY(DATAY,NTHX2,NPHI,DATA,IR2X2,IC2TON).....	72
A.1.4 SUBROUTINE NFKXY(DATA,IR2X2,IC2TON).....	77
A.1.5 SUBROUTINE ETIOGAM(DATA(1,COL),NROW,NCOL,ICOL,ISGN,FLMDA,DELX, DELY,DIST).....	82
A.1.6 SUBROUTINE PHSCOR2(DATA,NRX2,NCOL).....	85
A.1.7 SUBROUTINE SWAP(NRX2,NCOL,DATA).....	88
A.1.8 SUBROUTINE ARAYPTR(DATA,NRX2,NCOL).....	91
A.1.9 SUBROUTINE FFOUT(DATA,NRX2,NCOL,LUOUT).....	94
A.1.10 SUBROUTINE FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK).....	97
A.1.11 SUBROUTINE PARAB (FOD,DOL,BLOCK,DFOCUS,ACOSE,ACOSH,THETA,ETHETA, EPhi).....	105
A.1.12 SUBROUTINE PLT12OR(X,Y,XMAX,XMIN,YMAX,YMIN,LAST,ISYMBOL,NO,MOST).....	113
A.2 SAMPLE PROGRAM INPUT AND OUTPUT.....	115
Appendix B. CUPLNF - CALCULATION OF COUPLING BETWEEN ANTENNAS.....	126
B.1 GENERAL OVERVIEW OF COMPUTER PROGRAM.....	126

B.1.1	PROGRAM CUPLNF(INPUT,OUTPUT,TAPE 1,TAPE 3,...,TAPE 8).....	128
B.1.2	SUBROUTINE ANGLGEN(PKXOXK,PKYOXK,PHI,THETA,PSI,PHIP,THETAP,PSIP, PHIT,THETAT,PHIR,THETAR).....	143
B.1.3	SUBROUTINE FINDFF(IDAYHR,LUIN,LUA,LUOY,LUOZ,DATA,NRX2,NCOL,FFY FFZ,STOR).....	147
B.1.4	SUBROUTINE VECTGEN(FOX,FOY,FOZ,PH,THET,PS,FX,FY,FZ).....	153
B.1.5	SUBROUTINE MINMAX(Z,ZMIN,ZMAX,LEX,LEY).....	156
B.2	SAMPLE PROGRAM INPUT AND OUTPUT.....	158

## LIST OF FIGURES

	<u>Page</u>	
Figure 1.	Coupling Schematic for two antennas (0 and 0' will be chosen at roughly the center of the radiating part of their respective antenna).....	5
Figure 2.	Definition of coordinates for the left antenna of figure 1.....	10
Figure 3.	Definition of coordinate systems for the right antenna of figure 1.....	13
Figure 4.	Physical interpretation for limits of integration. To a good approximation, only that portion of the spectrum within $\alpha$ is required to compute the coupling quotient $b'_0/a_0$ for the two antennas.....	18
Figure 5.	Hypothetical circular antennas directly facing each other in the near field.....	22
Figure 6.	Coupling of circular antennas computed first using FFT integration, and then directly from far field along direction of separation.....	23
Figure 7.	Typical coupling curve for antennas skewed in their near field.....	24
Figure 8.	Geometry of vectors for surface integral.....	29
Figure 9a.	Field strength in a uniformly illuminated aperture calculated using physical optics far fields. Dashed line indicates theoretical distribution.....	32
Figure 9b.	Phase of field in a uniformly illuminated aperture calculated using physical optics for fields.....	33
Figure 10a.	Comparison of measured and calculated far-field patterns for antenna No. 1. E-plane cut, solid line - measured pattern, dashed line - physical optics.....	35
Figure 10b.	Comparison of measured and calculated far-field patterns for antenna No. 1. H-plane cut, solid line - measured pattern, dashed line - physical optics.....	36
Figure 11a.	Comparison of measured and calculated far-field patterns for antenna No. 2. E-plane cut, solid line - measured pattern, dashed line - physical optics.....	37
Figure 11b.	Comparison of measured and calculated far-field patterns for antenna No. 2. H-plane cut, solid line - measured pattern, dashed line - physical optics.....	38

Figure 12a.	Comparison of measured and calculated far-field patterns for antenna No. 3. E-plane cut, solid line - measured pattern, dashed line - physical optics.....	39
Figure 12b.	Comparison of measured and calculated far-field patterns for antenna No. 3. H-plane cut, solid line - measured pattern, dashed line - physical optics.....	40
Figure 13.	Comparison of measured and calculated far-field patterns for antenna No. 4. H-plane cut, solid line - measured pattern, dashed line - physical optics.....	41
Figure 14.	Comparison of effective current distribution used in physical optics and geometrical theory of diffraction calculations. (Uniform distribution assumed).....	43
Figure 15.	Diagram of multiple reflections involving feed structure.....	43
Figure 16a.	Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. E-plane cut, solid curve - measured pattern, dashed curve - physical optics.....	44
Figure 16b.	Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. H-plane cut, solid curve - measured pattern, dashed curve - physical optics.....	45
Figure 17a.	Feed region of antenna with absorber collar.....	46
Figure 17b.	Feed support struts with absorber attached.....	46
Figure 18a.	Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. E-plane cut, solid curve - measured pattern, dashed curve - physical optics.....	47
Figure 18a.	Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. H-plane cut, solid curve - measured pattern, dashed curve - physical optics.....	48
Figure 19.	Photograph of experimental set up for measuring coupling between two reflector antennas.....	51
Figure 20.	Schematic showing relative orientations of antennas for the three test cases.....	52
Figure 21.	Mutual coupling between 1.2 meter reflector antennas. Case 1: $\theta_r=0^\circ$ , $\theta_t=0^\circ$ . Solid lines indicate envelope of measured mutual coupling.....	54
Figure 22.	Mutual coupling between 1.2 meter reflector antennas. Case 2: $\theta_r=15^\circ$ , $\theta_t=0^\circ$ . Solid lines indicate envelope of measured mutual coupling.....	55
Figure 23.	Mutual coupling between 1.2 meter reflector antennas. Case 3: $\theta_r=15^\circ$ , $\theta_t=20^\circ$ . Solid lines indicate envelope of measured mutual coupling.....	56





DETERMINATION OF MUTUAL COUPLING BETWEEN CO-SITED MICROWAVE  
ANTENNAS AND CALCULATION OF NEAR-ZONE ELECTRIC FIELD

By

C. F. Stubenrauch and A. D. Yaghjian

The theory and computer programs which allow the efficient computation of coupling between co-sited antennas given their far-field patterns are developed. Coupling between two paraboloidal reflector antennas is computed using both measured far-field patterns and far-field patterns which were obtained from a physical optics (PO) model. These computed results are then compared to the coupling measured directly on an outdoor antenna range. Far fields calculated using the PO model are compared to those obtained from transformed near-field measurements for several reflector antennas. Theory and algorithms are also developed for calculating near-field patterns from far fields obtained from the PO model. Documentation of the near-field and coupling computer programs is presented in the appendices. Conclusions and recommendations for future work are included.

Key words: Co-sited antennas; coupling; far fields; near fields; physical optics; plane-wave spectrum; reflector antennas.

#### INTRODUCTION

This report discusses work done at the National Bureau of Standards (NBS) concerning problems related to the prediction of mutual coupling between antennas and the prediction of antenna near fields. In addition, comparisons for several paraboloidal reflector antennas were made between far-field patterns obtained from near-field measurements and those which were predicted using a physical optics (PO) model for the antennas.

A consequence of the scattering matrix theory of antennas and antenna-antenna interactions developed at NBS over the past 20 years [1] is that mutual coupling and near fields can be calculated provided the plane-wave spectra for the antenna or antennas are known. The essential, propagating part of a spectrum is related by a simple expression to the antenna's far-field pattern which may be determined, e.g., through model computation, direct far-field measurements, or transformed near-field measurements. For engineering studies of co-sited coupling or antenna near fields, expressing the quantities of interest in terms of the far fields proves especially convenient. In many cases the measured patterns are unavailable. Because it is possible to predict these patterns by employing a suitable model, part of the work described herein discusses the capability of a particularly convenient and efficient model: the physical optics computer program obtained from the University of Southern California (USC).

Formulations of the mutual coupling problem in terms of antenna far fields are well known [7]; however, calculations using previous theories have been deficient because of the large amounts of computation time and data required. In this work, it is shown that the functions to be integrated can be made band-limited; and thus the sampling theorem can be employed to determine the required point spacing, rather than the more usual trial-and-error method of testing convergence. Further, it is shown that the evaluation of mutual coupling requires only the far fields lying within the mutually subtended angles of the antennas. As a result of these improvements in the theory, an efficient program for calculating mutual coupling was written.

Section 1 of this report details the theory which allows rapid calculation of the mutual coupling between two antennas without restrictions on the separation distances. Section 2 discusses the specific problem of obtaining the near fields of an antenna given the far-field pattern. The PO model for reflector antennas is briefly discussed in section 3 as is the particular model employed. Far-field patterns which were predicted by the PO model and far-field patterns obtained from near-field measurements of actual antennas are compared in section 4. In section 5 coupling values measured directly in the laboratory are compared to those predicted from the theory of section 1 employing both modeled and measured far-fields. Conclusions and recommendations are given in section 6.

The appendices describe the computer programs which perform the coupling and near-field calculations. Appendix A discusses and documents POMODL, a program which uses a PO model to calculate the far-field pattern for a reflector antenna and which calculates from this pattern the near-field distribution on a specified plane. The predicted far field also provides output for use as input by the program CUPLNF (described in sec. 1 and documented in Appendix B) which calculates the mutual coupling between two arbitrarily located and oriented antennas from their far-field patterns.

## 1. FORMULATION OF THE MUTUAL COUPLING BETWEEN TWO ANTENNAS

The plane-wave scattering matrix (PWSM) description of antennas, introduced by Kerns at the NBS, forms an ideal theoretical framework on which to base the determination of mutual coupling between two collocated antennas. In fact, the basic PWSM formula required for the determination of the coupling between two antennas has existed for nearly twenty years [1]. However, before the existing formulas could be translated into a convenient program which computed coupling efficiently, three important tasks needed to be accomplished:

1) The Kerns coupling formula or transmission integral, as he calls it, was originally written in terms of the appropriate plane-wave spectrum for each antenna. For our purposes, we wanted to express the near-field mutual coupling in terms of the far field of each antenna (assuming reciprocal antennas) because usually the far field most conveniently characterizes an antenna and is most efficiently computed from, e.g., a PO-GTD (physical optics and/or geometrical theory of diffraction) program or from near-field measurements. This task, although straightforward, requires careful attention to the details of definition of the far field, the plane-wave spectrum, and the reciprocity for each antenna.

2) The far fields of each antenna are usually expressed in a Cartesian coordinate system fixed in each antenna. To compute coupling for an arbitrary separation and orientation of two antennas, the coupling formula requires an integration of the dot product of the two vector far-field patterns in reoriented coordinate systems. Thus, task two consisted of expressing the reoriented coordinates of each antenna in terms of the Eulerian angles from the preferred or fixed coordinates in which the far field of the antenna was given. In addition, a similar transformation had to be applied in order to compute the dot product of the two vector far-field patterns. Again this task was fairly straightforward, yet rather tedious.

3) Finally, even though tasks (1) and (2) above recast the coupling or transmission integral in terms of the far fields of each antenna expressed in the preferred coordinate system fixed in each antenna, repeated evaluation of the double integrals (actually a double Fourier transform) would require a prohibitive amount of computer time for electrically large microwave antennas unless the sampling theorem and FFT (fast Fourier transform) algorithm could be applied effectively. However, the application of the sampling theorem to these double Fourier transforms requires a sample spacing which, in general, is so small that repeated evaluation even by means of the FFT still becomes prohibitive. Moreover, the required sample spacing becomes smaller with increasing separation distance between antennas. Thus, the third major task was to discover a way to reduce drastically the computer time needed to evaluate the final form of the double integrals expressing the mutual coupling between two antennas.

The details of these three tasks and their accomplishment are described in the following three major sections (1.1, 1.2, 1.3).

## 1.1. The Basic Coupling Formula (Transmission Integral)

This section begins with the transmission integral derived by Kerns [1] for the coupling of two antennas (when multiple reflections are neglected) in terms of the transmitting and receiving spectra of the respective antennas. The receiving antenna is assumed reciprocal, and its receiving spectrum is written in terms of its transmitting spectrum through the reciprocity relations. The transmitting spectrum of each antenna is then expressed in terms of the antenna's far electric field, which in turn yields a transmission integral or coupling formula in terms of the dot product of the vector far fields of each antenna. Finally, reciprocity is invoked for both antennas to prove that the mutual coupling is essentially the same when the roles of transmission and reception are exchanged.

### 1.1.1. The Plane-Wave Scattering Matrix Approach

Consider an arbitrary antenna transmitting with  $e^{-i\omega t}$  time dependence to the left of an arbitrary receiving antenna, as shown in figure 1. The antennas may have arbitrary separation and orientation. Assume that only one mode propagates in the waveguide feed to each antenna.<sup>1</sup> The incident waveguide mode coefficients for the left antenna are labeled  $a_0$  and  $b_0$  respectively, and for the right antenna,  $a'_0$  and  $b'_0$  respectively. The reflection coefficients of the right (receiving) antenna and its passive termination are denoted by  $\Gamma'_0$  and  $\Gamma'_L$  respectively.

The quantity  $b'_0/a_0$ , which we shall call the coupling quotient, is a measure of how much signal couples into the receiving antenna per unit input into the transmitting antenna. If the same type of waveguide feeds each antenna and the receiving waveguide is terminated in a perfectly matched load,  $|b'_0/a_0|^2$  equals the amount of power coupled to the receiving antenna per unit power incident to the transmitting antenna. (This power ratio expressed in decibels is commonly referred to as the insertion loss ratio.) Thus,  $b'_0/a_0$  is indeed the major parameter of interest in determining mutual interference between antennas.

The transmission integral which gives the coupling quotient in terms of transmitting and receiving plane-wave spectra of the respective antennas can be found directly from Kerns [1b]:

$$\frac{b'_0}{a_0} = \frac{1}{1-\Gamma'_L\Gamma'_0} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \underline{s}'_{02}(\underline{k}) \cdot \underline{s}_{10}(\underline{k}) e^{i\gamma d} d\underline{k}, \quad (1)$$

where  $\underline{s}_{10}(\underline{k})$  and  $\underline{s}'_{02}(\underline{k})$  are the "complete" transmitting and receiving spectra defined with respect to plane waves traveling in the common  $\underline{k}$  direction but with phase reference to the

<sup>1</sup>If more than one mode propagates in one or both of the feeds, this analysis can be applied for each possible transmit-receive pair of modes; and thus the analysis can be applied to "out-of-band" coupling.

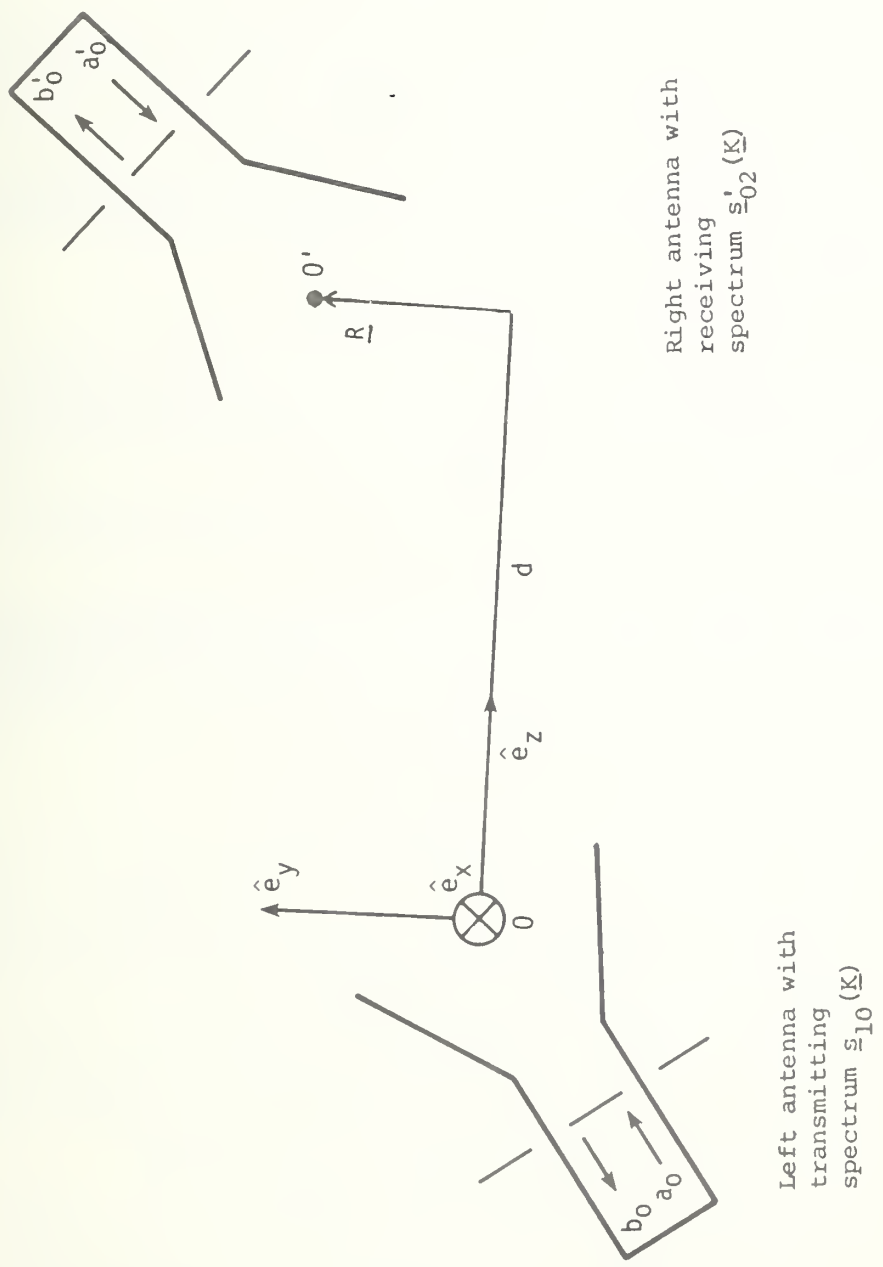


Figure 1. Coupling Schematic for two antennas ( $0$  and  $0'$  will be chosen at roughly the center of the radiating part of their respective antenna).

origins 0 and 0' of the left (transmitting) and right (receiving) antennas respectively. The z axis is chosen to run from 0 to 0', with the distance  $d = 00'$  and the x-y axes perpendicular to the z axis at 0 (see fig. 1).  $\underline{K} = k_x \hat{e}_x + k_y \hat{e}_y$  is the transverse part of the propagation vector  $\underline{k} = \underline{K} + \gamma \hat{e}_z$  ( $k = \frac{2\pi}{\lambda}$ , where  $\lambda$  is the wavelength), and  $\gamma = (k^2 - K^2)^{1/2}$  is taken positive real for  $K < k$  and positive imaginary for  $K > k$ .  $d\underline{K}$  is shorthand notation for the double differential  $dk_x dk_y$ .

Equation (1) is an exact result from Maxwell's equation for two linear antennas operating with  $e^{-i\omega t}$  time dependence in free space, when multiple reflections between the antennas are neglected. (In other words, the  $b'_0/a_0$  computed from eq (1) neglects power which enters the receiver after having been reflected from receiving antenna to transmitting antenna and back one or more times.) No other restrictive assumptions are involved. For example, the antennas may be lossy or even nonreciprocal.

Of course, eq (1) cannot be used to evaluate  $b'_0/a_0$  unless the spectra  $\underline{s}'_{02}$  and  $\underline{s}_{10}$  are determined explicitly in terms of commonly measured or computed characteristics of the antenna. Toward this end, both spectra and eq (1) are recast in the next subsection in terms of the far electric fields of the antennas.

### 1.1.2. The Coupling Quotient in Terms of Far Field of Each Antenna

As a preliminary to expressing eq (1) in terms of the far fields of the antennas, assume that the receiving antenna contains no nonreciprocal devices or material so that its receiving functions  $\underline{s}'_{02}$  are related to its transmitting functions  $\underline{s}'_{20}$  by the simple reciprocity formula [1b],

$$\eta'_0 \underline{s}'_{02}(\underline{K}) = \frac{\gamma}{kZ_0} \underline{s}'_{20}(-\underline{K}). \quad (2)$$

All quantities in eq (2) have been defined in the previous section except the impedance of free space  $Z_0$  and  $\eta'_0$ , which is the characteristic admittance of the propagated mode in the feed waveguide of the right (receiving) antenna of figure 1.

Substitution of  $\underline{s}'_{02}$  from eq (2) into eq (1) gives,

$$\frac{b'_0}{a_0} = \frac{(1 - \Gamma_L \Gamma'_0)^{-1}}{kZ_0 \eta'_0} \int_{K < k} \gamma \underline{s}'_{20}(-\underline{K}) \cdot \underline{s}_{10}(\underline{K}) e^{i\gamma d} d\underline{K}. \quad (3)$$

Note that the integration limits in eq (3) have been made finite by eliminating the integration over the evanescent part of the spectra (included in the original infinite limits of eq (1)), thereby leaving only the radiating part of the spectra. This is permissible for all nonsuper-reactive antennas which are separated by a distance greater than a wavelength or so, i.e., if the antennas are outside each other's reactive field zone [2];

and if the contribution from the integration in eq (3) near the critical point  $K = k$  is negligible, as is usually the case.

A major advantage of the PWSM techniques is that the radiating part of the spectrum of an antenna is proportional to the vector far field  $\underline{E}(\underline{r})_{r \rightarrow \infty}$  of the antenna. Specifically, if  $\underline{f}(\underline{r})$  refers to the normalized, complex far-electric-field pattern of the left (transmitting) antenna of figure 1, i.e.,

$$\underline{f}(\underline{r}) \equiv \frac{re^{-ikr}}{a_0} \underline{E}(\underline{r})_{r \rightarrow \infty}, \quad (4)$$

then the radiating spectrum,  $\underline{s}_{10}(K)$ ,  $K < k$ , is related to the complex far-field pattern by the disarmingly simple proportionality [1b],

$$\underline{s}_{10}(K) = \frac{i}{\gamma} \underline{f}(k) \quad (5)$$

Although  $\underline{f}$  is shown as a function of  $\underline{r}$  in eq (4), we know that the complex far-field pattern is a function only of the direction of  $\underline{r}$ ; and thus  $\underline{f}(k)$  in eq (5) is also only a function of the direction of  $\underline{k}$  which is determined solely by the relative size of  $k_x$  and  $k_y$ , the integration variables of eq (3).

Similarly, the radiating spectrum,  $\underline{s}'_{20}$ ,  $K < k$ , for the right (receiving) antenna in figure 1 can be written in terms of the normalized, complex, far-electric-field pattern  $\underline{f}'$  of that antenna:

$$\underline{s}'_{20}(-K) = \frac{i}{\gamma} \underline{f}'(-k), \quad (6)$$

where, as in eq (4),  $\underline{f}'$  is defined in terms of the far-electric-field  $\underline{E}'(\underline{r})_{r \rightarrow \infty}$  of the right antenna when it is radiating:

$$\underline{f}'(\underline{r}) \equiv \frac{re^{-ikr}}{a'_0} \underline{E}'(\underline{r})_{r \rightarrow \infty}. \quad (7)$$

Substitution of the spectra from eqs (6) and (7) into eq (3) produces the coupling quotient for two antennas as a double integral over the dot product of the complex far-electric-field patterns of the antennas:

$$\frac{b'_0}{a_0} = -C' \int \int_{K < k} \frac{\underline{f}'(-k) \cdot \underline{f}(k)}{\gamma} e^{i\gamma d} d\underline{k}. \quad (8)$$

In eq (8),  $C'$  is a consolidated notation for the "mismatch factor"  $(1 - \Gamma'_L \Gamma'_0)^{-1} / kZ_0 \eta'_0$ .

### 1.1.3. Coupling Quotient When the Roles of Transmitting and Receiving are Exchanged

The coupling quotient  $b'_0/a_0$  in eq (8) is a measure of the signal which is received by the passively terminated antenna on the right side of figure 1 when an input mode of unit amplitude is applied to the transmitting antenna on the left. A natural and important question is what will be the coupling to the left antenna when the right antenna transmits at the same frequency and the left antenna is terminated in a passive load. Specifically, what is the expression for  $b_0/a'_0$  and how is it related to  $b'_0/a_0$  of eq (8).

The answer to this question can be obtained immediately by retracing the steps in the derivation of eq (8) but with the left antenna in figure 1 receiving and the right antenna transmitting. So doing, yields an expression for  $b_0/a'_0$  very similar to eq (8).

$$\frac{b_0}{a'_0} = -C \iint_{K' < k} \frac{f(-k') \cdot f'(k')}{\gamma'} e^{i\gamma'd} d\underline{K}' , \quad (9)$$

where the "mismatch factor" C is defined as before,

$$C = (1 - \Gamma_L \Gamma_0)^{-1} / kZ_0 \eta_0 . \quad (10)$$

$\Gamma_0$  and  $\Gamma_L$  are now the reflection coefficients to the antenna on the left and its passive termination, respectively. And  $\eta_0$  is now the characteristic admittance of the propagated mode in the waveguide feed to the left antenna. Because  $\hat{e}_z = -\hat{e}_z$ , we can choose  $\hat{e}_y = \hat{e}_y$  and  $\hat{e}_x = -\hat{e}_x$ . Then changing the dummy integration variables in eq (9) from  $k'_x$  and  $k'_y$  to  $k_x$  and  $-k_y$  shows that the integration in eq (9) is identical to eq (8), i.e.,

$$\frac{b_0}{a'_0} = -C \iint_{K < k} \frac{f(k) \cdot f'(-k)}{\gamma} e^{i\gamma d} d\underline{K} . \quad (11)$$

Comparing eqs (8) and (11), we see that the two coupling quotients,  $b'_0/a_0$  and  $b_0/a'_0$ , are related merely through a constant factor, i.e.

$$C' \frac{b_0}{a'_0} = C \frac{b'_0}{a_0} . \quad (12)$$

This means that if the coupling between two antennas is measured or computed with one of the antennas transmitting and the other receiving, the coupling, when the roles of transmitting and receiving are reversed, is also known (through eq (12)). A separate measurement or computation need not be done. Use of eq (12), of course, requires knowledge of the reflection coefficients and input admittances of each antenna contained in the definitions of C and C'.



As a check, eq (12) was also derived directly from the "system two-port" equations describing the two antennas, by applying the Lorentz reciprocity theorem [1b] and knowing that multiple reflections between the antennas are being neglected. It can be further proven that if scattered fields are also negligibly received by the transmitting antenna, then the available power at the receiving antenna per unit input power to the transmitting antenna is the same when the rules of receiving and transmitting are reversed.

## 1.2. Eulerian Angle Transformations Describing the Arbitrary Orientation of the Antennas

From a quick look at eq (8), it might be concluded that the analysis required to compute the coupling between two antennas is essentially finished. All we need to do is compute or measure the vector far-field patterns of each antenna, take their dot product, and perform the double integration on a computer.

Unfortunately, a major problem, ignored so far, is the fact that the far-field pattern of an antenna is given with respect to a Cartesian coordinate system which is fixed in the antenna and which is not, in general, aligned with the Cartesian system shown in figure 1 to which the far-field patterns  $\underline{f}(\underline{k})$  and  $\underline{f}'(-\underline{k})$  in eq (8) are referenced. Thus, to use eq (8), it is mandatory that the far-field direction in the coordinate system fixed in each antenna corresponding to a given  $(k_x, k_y)$  in eq (8) be determined explicitly. Moreover, to evaluate the dot product  $\underline{f}' \cdot \underline{f}$ , the rectangular components of  $\underline{f}$  and  $\underline{f}'$  in the x-y-z system of figure 1 must be expressed in terms of the rectangular components of the coordinate systems fixed in the antennas.

Fortunately, all these necessary transformations can be accomplished by specifying the Eulerian angles required to align the axes fixed in each antenna with the (x, y, z) axes chosen in figure 1, as the following two subsections explain.

### 1.2.1. Rotational Transformations from $(k_x, k_y)$ to the Far-Field Direction in the Fixed Coordinate System of Each Antenna

Assume the left antenna in figure 1 has a fixed coordinate system with rectangular axes  $(x_A, y_A, z_A)$  centered at 0) in which the normalized far-electric-field pattern is given in terms of the spherical angles  $\phi_A$  and  $\theta_A$ , as shown in figure 2a. That is, we have at our disposal, obtained from either measurement or computation, the vector far-field pattern  $\underline{f}(\phi_A, \theta_A)$  as a function of  $\phi_A$  and  $\theta_A$ .

Let  $(\phi, \theta, \psi)$  be the Eulerian angles needed to rotate the  $(x_A, y_A, z_A)$  axes in line with the (x, y, z) coupling axes of figure 1. Specifically, as shown in figure 2b, rotate an angle  $\phi (0 \leq \phi < 2\pi)$  about the positive  $z_A$  axis, thereby changing the direction of  $x_A$  and  $y_A$  but not  $z_A$ . Then rotate an angle  $\theta (0 \leq \theta \leq \pi)$  about the new positive  $y_A$  axis, thereby changing the direction of  $z_A$  (to z) and again  $x_A$  but not  $y_A$ . ( $\phi$  and  $\theta$  are the usual spherical angles.) Finally, rotate an angle  $\psi (0 \leq \psi < 2\pi)$  about the positive z axis to align the new  $x_A$  and  $y_A$  axes with x and y. These are fairly common definitions of Eulerian angle rotations found in a number of textbooks such as reference [3].

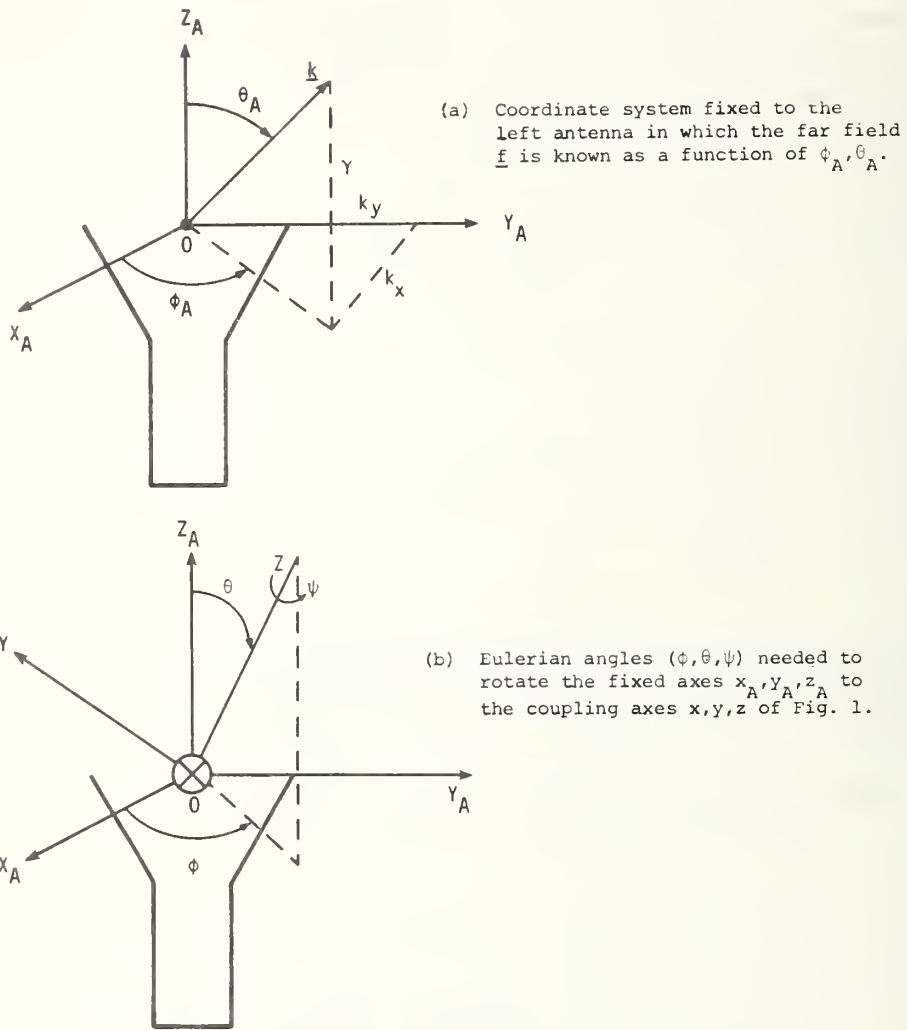


Figure 2. Definition of coordinates for the left antenna of figure 1.

To understand the transformation needed to evaluate eq (8), note in eq (8) that  $\underline{f}$  and  $\underline{f}'$  are written as functions of  $\underline{k}=k_x\hat{e}_x+k_y\hat{e}_y+\gamma\hat{e}_z$  or, in other words, as functions of  $k_x$  and  $k_y$  because  $\gamma$  is determined from  $k_x$  and  $k_y$ . However, we are given as known (measured or computed)  $\bar{f}$  as a function of  $\phi_A$  and  $\theta_A$ , not  $k_x$  and  $k_y$ . Consequently, to evaluate eq (8) numerically, a transformation is needed which will convert  $(k_x, k_y)$  to  $(\phi_A, \theta_A)$  under the given Eulerian angles  $(\phi, \theta, \psi)$  defining the  $x_A-y_A-z_A$  system with respect to the  $x-y-z$  system. This Eulerian transformation, which is a straightforward, rather lengthy, linear transformation found in a number of textbooks [3], will not be derived here but simply stated in the form useful for our purposes of evaluating eq (8).

Before actually writing the required expression for  $\phi_A$  and  $\theta_A$ , the antenna on the right side of figure 1 should also be discussed because it will require a similar transformation to convert  $k_x$  and  $k_y$  to the spherical angles of its preferred system. That is, if the far-field pattern  $\underline{f}'$  of this right antenna is known (measured or computed) in terms of spherical angles  $\phi_p$  and  $\theta_p$  with respect to  $(x_p, y_p, z_p)$  axes fixed to the antenna (and centered at  $0'$ ), then  $(\phi_p, \theta_p)$  are needed as functions of  $(k_x, k_y)$  in order to evaluate  $\underline{f}'(-\underline{k})$  in eq (8) (see fig. 3). (An important point to remember is that  $\underline{f}'(-\underline{k})$  denotes the value of the far-field pattern in the  $-\underline{k}$  direction.) Also, as shown in figure 3, let  $\phi', \theta',$  and  $\psi'$  denote the Eulerian angles which rotate the  $(x_p, y_p, z_p)$  axes fixed in the right antenna parallel to the  $(-x, y, -z)$  coupling axes of figure 1.

Both transformations, from  $(k_x, k_y)$  to  $(\phi_A, \theta_A)$  and  $(\phi_p, \theta_p)$ , are similar and can be written explicitly as:

$$\cos \begin{pmatrix} \theta_A \\ \theta_p \end{pmatrix} = -\sin \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \cos \begin{pmatrix} \psi \\ \psi' \end{pmatrix} \frac{k_x}{k} \pm \sin \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \sin \begin{pmatrix} \psi \\ \psi' \end{pmatrix} \frac{k_y}{k} + \cos \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \frac{\gamma}{k} \quad (13a)$$

$$\tan \begin{pmatrix} \phi_A \\ \phi_p \end{pmatrix} = \frac{\left[ \sin \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \cos \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \cos \begin{pmatrix} \psi \\ \psi' \end{pmatrix} + \cos \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \sin \begin{pmatrix} \psi \\ \psi' \end{pmatrix} \right] \frac{k_x}{k} \mp \left[ \sin \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \cos \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \sin \begin{pmatrix} \psi \\ \psi' \end{pmatrix} - \cos \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \cos \begin{pmatrix} \psi \\ \psi' \end{pmatrix} \right] \frac{k_y}{k} + \sin \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \sin \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \frac{\gamma}{k}}{\left[ \cos \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \cos \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \cos \begin{pmatrix} \psi \\ \psi' \end{pmatrix} - \sin \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \sin \begin{pmatrix} \psi \\ \psi' \end{pmatrix} \right] \frac{k_x}{k} \mp \left[ \cos \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \cos \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \sin \begin{pmatrix} \psi \\ \psi' \end{pmatrix} + \sin \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \cos \begin{pmatrix} \psi \\ \psi' \end{pmatrix} \right] \frac{k_y}{k} + \cos \begin{pmatrix} \phi \\ \phi' \end{pmatrix} \sin \begin{pmatrix} \theta \\ \theta' \end{pmatrix} \frac{\gamma}{k}} \quad (13b)$$

The top signs in eqs (13) go with  $(\phi_A, \theta_A)$ , the bottom with  $(\phi_p, \theta_p)$ . Equations (13) look rather cumbersome at first sight, yet computationally they are quite manageable because they involve only sines and cosines of the Eulerian angles and linear dependence upon  $k_x, k_y,$  and  $\gamma$  (which equals  $\sqrt{k^2 - (k_x^2 + k_y^2)}$ ). The computer program merely contains a subroutine which yields  $(\phi_A, \theta_A)$  and  $(\phi_p, \theta_p)$  from eqs (13) when given the Eulerian angles  $(\phi, \theta, \psi), (\phi', \theta', \psi'),$  and  $(k_x, k_y)$  as input.

With the transformations of eqs (13), eq (8) can now be expressed in terms of  $(\theta_A, \phi_A)$  and  $(\phi_p, \theta_p)$ :

$$\frac{b'_0}{a_0} = -C' \iint_{K < k} \frac{\underline{f}'(\phi_p, \theta_p) \cdot \underline{f}(\phi_A, \theta_A)}{\gamma} e^{i\gamma d} dK. \quad (14)$$

### 1.2.2. Vector Component Transformations Required to Compute the Coupling Dot Product

In the previous subsection a transformation was written that yielded  $\underline{f}$  and  $\underline{f}'$  in eq (14) as functions of the spherical angles  $(\phi_A, \theta_A)$  and  $(\phi_p, \theta_p)$  in which the far-field patterns were measured or computed. Still, a method is needed to compute the dot product  $\underline{f}' \cdot \underline{f}$ , because the components of  $\underline{f}$  and  $\underline{f}'$  are given in terms of unit vectors of the  $(x_A, y_A, z_A)$  and  $(x_p, y_p, z_p)$  coordinate systems fixed respectively in the left and right antennas of figure 1. And these two sets of unit vectors have relative directions which depend also on the Eulerian angles  $(\phi, \theta, \psi)$  and  $(\phi', \theta', \psi')$ .

A convenient way to evaluate  $\underline{f}' \cdot \underline{f}$  is to first write  $\underline{f}$  and  $\underline{f}'$  in the  $(x, y, z)$  and  $(x', y', z')$  rectangular components respectively shown in figures 2 and 3,

$$\underline{f} = f_x \hat{e}_x + f_y \hat{e}_y + f_z \hat{e}_z \quad (15a)$$

$$\underline{f}' = f'_{x'} \hat{e}_{x'} + f'_{y'} \hat{e}_{y'} + f'_{z'} \hat{e}_{z'}. \quad (15b)$$

Because by definition,

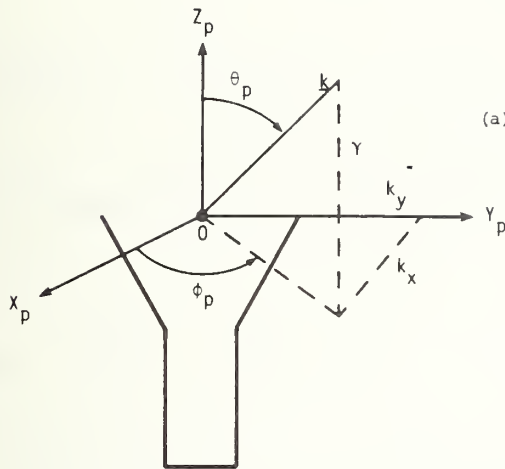
$$\hat{e}_{x'} = -\hat{e}_x, \quad \hat{e}_{y'} = \hat{e}_y, \quad \text{and} \quad \hat{e}_{z'} = -\hat{e}_z, \quad (16)$$

the dot product becomes

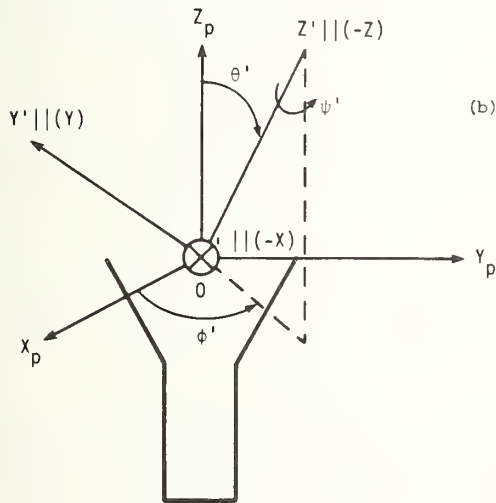
$$\underline{f}' \cdot \underline{f} = -f'_{x'} f_x + f'_{y'} f_y - f'_{z'} f_z. \quad (17)$$

Next, we express the rectangular components of eq (17) in the rectangular components with respect to the fixed axes  $(x_A, y_A, z_A)$  and  $(x_p, y_p, z_p)$ , again through the appropriate Eulerian transformation. In matrix notation

$$\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} (\cos \phi \cos \theta \cos \psi - \sin \phi \sin \psi)(\sin \phi \cos \theta \cos \psi + \cos \phi \sin \psi)(-\sin \theta \cos \psi) \\ (-\cos \phi \cos \theta \sin \psi - \sin \phi \cos \psi)(-\sin \phi \cos \theta \sin \psi + \cos \phi \cos \psi)(\sin \theta \sin \psi) \\ (\cos \phi \sin \theta) & (\sin \phi \sin \theta) & (\cos \theta) \end{pmatrix} \begin{pmatrix} f_{xA} \\ f_{yA} \\ f_{zA} \end{pmatrix} \quad (18)$$



(a) Coordinate system fixed to the right antenna in which the far field  $\underline{f}'$  is known as a function of  $\phi_p, \theta_p$



(b) Eulerian angles  $(\phi', \theta', \psi')$  needed to rotate the fixed axes  $x_p, y_p, z_p$  to the coupling axes  $x', y', z'$ , which are in the direction of the  $(-x), (y), (-z)$  axes of Fig. 1.

Figure 3. Definition of coordinate systems for the right antenna of figure 1.

The counterpart equation for  $(f'_{x'}, f'_{y'}, f'_{z'})$  is the same as eq (18) but with  $(\phi', \theta', \psi')$  and  $(f'_{xp}, f'_{yp}, f'_{zp})$  replacing  $(\phi, \theta, \psi)$  and  $(f_{xA}, f_{yA}, f_{zA})$ , respectively. It should also be noted that the x, y, and z components of the far field are not independent because there is no radial component of far field. Using  $f_A$ , for an example, the rectangular components are related by  $\cos \phi_A \sin \theta_A f_{xA} + \sin \phi_A \sin \theta_A f_{yA} + \cos \theta_A f_{zA} = 0$ .

If the far-field components  $(f_{xA}, f_{yA}, f_{zA})$  for the left antenna and  $(f'_{xp}, f'_{yp}, f'_{zp})$  for the right antenna of figure 1 are known, eq (18) and its counterpart equation yield  $(f_x, f_y, f_z)$  and  $(f'_{x'}, f'_{y'}, f'_{z'})$  in terms of the given Eulerian angles. In turn, eq (17) yields the dot product  $\underline{f}' \cdot \underline{f}$ . Again, the computer program which computes the double integral (14) need only contain a simple subroutine to evaluate eq (18), and the dot product  $\underline{f}' \cdot \underline{f}$  is immediately computable from eq (17).

One other set of transformations often proves useful, however. Usually, the far field of an antenna is given not in terms of rectangular components but in terms of spherical components. If the far-electric-field pattern of the left and right antennas of figure 1 are known in terms of  $(f_{\theta A}, f_{\phi A})$  and  $(f'_{\theta p}, f'_{\phi p})$  respectively, then the rectangular components are related to these spherical components by the spherical angles. Specifically,

$$\begin{pmatrix} f_{xA} \\ f_{yA} \\ f_{zA} \end{pmatrix} = \begin{pmatrix} -\sin \phi_A & \cos \theta_A \cos \phi_A \\ \cos \phi_A & \cos \theta_A \sin \phi_A \\ 0 & -\sin \theta_A \end{pmatrix} \begin{pmatrix} f_{\phi A} \\ f_{\theta A} \end{pmatrix} \quad (19)$$

The counterpart equation giving  $(f'_{xp}, f'_{yp}, f'_{zp})$  as functions of  $(f'_{\phi p}, f'_{\theta p})$  is formed from eq (19) merely by replacing  $(\phi_A, \theta_A)$  in the matrix with  $(\phi_p, \theta_p)$ .

In summary, if  $(f_{\phi A}, f_{\theta A})$  and  $(f'_{\phi p}, f'_{\theta p})$  are the known far-electric-field patterns in the fixed coordinate systems of the left and right antennas of figure 1, respectively, eq (19) and its counterpart transform these spherical components to rectangular components. Equation (18) and its counterpart transform these rectangular components in the fixed systems to rectangular components in the coupling  $(x, y, z)$  or  $(x', y', z')$  coordinates. Finally, eq (17) yields the required dot product from the transformed components.

These transformations must be done for each  $(k_x, k_y)$  within the limits of integration needed to evaluate eq (14). Moreover, eqs (13) must be evaluated for each  $(k_x, k_y)$ . Fortunately, the nature of the integrals in eq (14) allows the application of the sampling theorem and fast Fourier transform, as well as the limits of integration to be reduced inversely proportional to d. These topics, which enable the efficient computer evaluation of the mutual coupling quotient, are covered in the following section.

### 1.3. The Sampling Theorem, Limits of Integration, and Fast Fourier Transform

This section shows how the sampling theorem converts the double integration in eq (14) to a double summation which can be summed using the fast Fourier transform (FFT) algorithm. In addition, the effective limits of integration are shown to reduce inversely proportional to  $d$ , the separation distance  $00'$  between the two antennas.

#### 1.3.1. The Point Spacing of $k_x$ and $k_y$ Required by the Sampling Theorem

Equation (14) represents the coupling quotient for the two antennas positioned in figure 1. If the antenna on the right side of figure 1 is displaced by a vector  $\underline{R}$  perpendicular to the  $z$  axis, the integrand in eq (14) changes only by the phase factor  $\exp(i\underline{K}\cdot\underline{R}) = \exp(ik_x x + ik_y y)$ . That is, eq (14) can be written more generally as

$$\frac{b'_0(\underline{R}, d)}{a_0} = -C' \int \int_{K < k} \frac{f'(\phi_p, \theta_p) \cdot f(\phi_A, \theta_A)}{\gamma} e^{i\gamma d} e^{i\underline{K}\cdot\underline{R}} d\underline{K} . \quad (20)$$

The sampling theorem [4] could be applied to convert the double Fourier transform in eq (20) to a double Fourier series, if  $b'_0(\underline{R}, d)$  were zero outside a finite  $|\underline{R}| = R_0$ . Now  $b'_0(\underline{R}, d)$  behaves as  $1/\sqrt{R^2 + d^2}$  as  $R \rightarrow \infty$ , and thus, strictly speaking, will never vanish for finite  $R_0$ . However, if we choose  $R_0 \gg d$ ,  $b'_0$  is small and the "aliasing" error introduced by using the sampling theorem should be small, especially near  $\underline{R} = 0$ , even though  $b'_0$  is not strictly "band limited" (i.e., zero outside a finite range).

In view of the decay of  $b'_0$  with  $R$ , choose

$$R_0 = Bd , \quad (21)$$

where  $B$  is a number much greater than 1. (Computations show that in practice, a  $B$  no larger than 1 or 2 is often sufficient for the accurate calculation of  $b'_0(\underline{R}, d)$  near  $\underline{R} = 0$  from eq (23) below. For larger  $\underline{R}$ , greater  $B$  is generally required. Also,  $R_0$  should never be smaller than about the sum of the diameters of the two antennas.) The sampling theorem applied to eq (20) then requires a sample spacing no larger than

$$\frac{\Delta k_x}{k} , \frac{\Delta k_y}{k} = \frac{\lambda}{2Bd} , \quad (22)$$

in order to convert eq (20) to the double summation,

$$\frac{b'_0(\underline{R}, d)}{a_0} = -C' \Delta k_x \Delta k_y \sum_{m=-M}^M \sum_{\ell=-L}^L \frac{f'(\phi_p^{\ell m}, \theta_p^{\ell m}) \cdot f(\phi_A^{\ell m}, \theta_A^{\ell m})}{\gamma_{\ell m}} e^{i\gamma_{\ell m} d} e^{i\underline{K}_{\ell m} \cdot \underline{R}} , \quad (23)$$

where

$$\frac{\underline{K}_{\ell m}}{k} = \frac{\ell \lambda}{2Bd} \hat{e}_x + \frac{m \lambda}{2Bd} \hat{e}_y , \quad (24)$$

and  $\ell, m$  are integers which range to cover the limits of integration  $|k_{\ell m}| < k$  (i.e.,  $L, M \approx \frac{2Bd}{\lambda}$ ).

The beauty of eq (23) is not only that the integrals have been converted to summations, which can be performed on a computer, but also that the summation is ideally suited for evaluation by means of the FFT algorithm, which decreases the running time considerably when the coupling quotient over a range of  $R$  is desired.

### 1.3.2 The Limits of Integration and Number of Points Required

The number of points required to compute the double summation of eq (23) is approximately  $(2Bd/\lambda)^2$  for each separation  $(R, d)$  and orientation of the antennas. For  $d/\lambda$  of appreciable size, the number of points can become so large that the computer time required to evaluate eq (23) over a range of  $R$ , even using the FFT, can become exorbitant. For example, if  $d = 10$  meters and  $\lambda = 3$ cm, choosing a typical value of  $B = 2$  yields  $(2Bd/\lambda)^2 = 1.8 \times 10^6$  terms to be summed for each separation and orientation of the antennas. Fortunately, however, it can be shown that the effective limits of integration, i.e.,  $M$  and  $L$  in eq (23), can be reduced inversely proportional to the separation distance  $d$  to keep the total number of summation points bounded to a manageable number regardless of the value of the separation  $d$  between antennas.

Consider eq (20) and rewrite the phase factor  $e^{i\gamma d} e^{i\mathbf{k} \cdot \mathbf{R}}$  in the plane-wave form  $e^{i\mathbf{k} \cdot \mathbf{r}}$ , where  $\mathbf{r} = \mathbf{R} + d\hat{\mathbf{z}}$ . For  $r$  much larger than the dimension of either antenna, the function  $e^{i\mathbf{k} \cdot \mathbf{r}}$  oscillates more rapidly than the oscillations of the far-field pattern dot product  $\mathbf{f}' \cdot \mathbf{f}$ , except when  $\mathbf{k}$  is in the directions approximately parallel to  $\mathbf{r}$ . This means that the integration in eq (20) will essentially cancel to zero except for the contribution near  $\mathbf{k}$  equal to  $\mathbf{r}$ , provided the contribution from near the critical point  $K = k$  is negligible, as is usually the case. In particular, a more thorough analysis of the integration in eq (20) reveals that in order to compute the coupling quotient for values of  $|R|$  between 0 and  $R$ , only the part of the spectrum defined by

$$\frac{K}{k} < \frac{R}{r} + \frac{(D+D')}{r}, \quad (r > R + D + D') \quad (25)$$

contributes significantly to the integration (under the assumed provision of negligible contribution from the end critical point). The quantities  $D$  and  $D'$  in the inequality (25) refer to the overall dimension of each of the antennas except when  $D$  and/or  $D'$  is less than  $2\lambda$ , in which case  $D$  and/or  $D'$  is set equal to  $2\lambda$ .<sup>2</sup> For example, if each antenna were an electrically large, circular aperture type of radiator,  $D$  and  $D'$  would be their respective diameters; but if one or the other of the antennas were a short dipole, its effective diameter would be set equal to  $2\lambda$ . Of course, nearly all microwave antennas have dimensions much greater than  $2\lambda$ .

<sup>2</sup>Equation (25) assumes implicitly that the origins  $0$  and  $0'$  for the two antennas by which  $r$  is defined ( $r = 00'$ ) are chosen near the physical centers of their respective antennas.



For  $R \ll (D+D')$ , i.e., coupling along the z axis as shown in figure 1, the criterion (25) reduces to simply  $K/k < \frac{D+D'}{d}$ , and the limits of integration in eq (20) become

$$K < \frac{k(D+D')}{d}, \quad (d \gg D+D' >> R). \quad (26)$$

As d gets much larger than the sum of the overall dimensions of the two antennas, eq (26) shows that the effective limits of integration become much less than the original  $K < k$ . This means that the summation limits L and M of eq (23) reduce to

$$L, M \approx \frac{2B(D+D')}{\lambda}. \quad (27)$$

The result (27), which holds for all separation distances for fixed B, is significant. It implies that the number of terms in the summation which evaluates the coupling quotient depends only on the electrical size of the antennas and not on the separation distance of the antennas. We will now show as a result of this reduction in effective limits of integration that the  $\Delta k_x, \Delta k_y$  sample spacing can be increased beyond that of eq (22) to an interval independent of the separation distance d until d reaches the mutual Rayleigh distance; and thus the summation limits L and M can be decreased with increasing d below the values given by eq (27).

Physically, eq (26) has a very simple interpretation. Referring to figure 4, it says that to a good approximation, for ordinary antennas larger than a couple of wavelengths across, only that portion of the plane-wave spectrum within the sheaf of angles mutually subtended by the smallest spheres circumscribing the radiating part of both antennas (including feeds, struts, edges and all other parts of the antenna which radiate significantly) is required to compute the coupling quotient. Thus, if the coupling quotient is desired only near  $R = 0$ , i.e.,

$$R \ll (D + D'), \quad (28)$$

the integration limits in eq (20) need extend only over K given by criterion (26). In other words, the spectrum can be set equal to zero outside the mutually subtended angle of figure 4. This means that the coupling quotient  $b'_0(R, d)$  computed from the limited integrations will no longer be equal, even approximately, to the actual coupling quotient for R greater than about  $(D+D')$ , but will in fact become zero more rapidly beyond  $(D+D')$ . Specifically, a more detailed analysis shows that limiting the range of integration to  $K < k(D+D')/d$  also artificially band-limits the coupling quotient to

$$R_0 = \text{larger of } \left\{ \frac{B(D + D')}{\frac{B\lambda d}{(D + D')}} \right\}. \quad (29)$$

From eq (22), the sampling theorem spacing is then

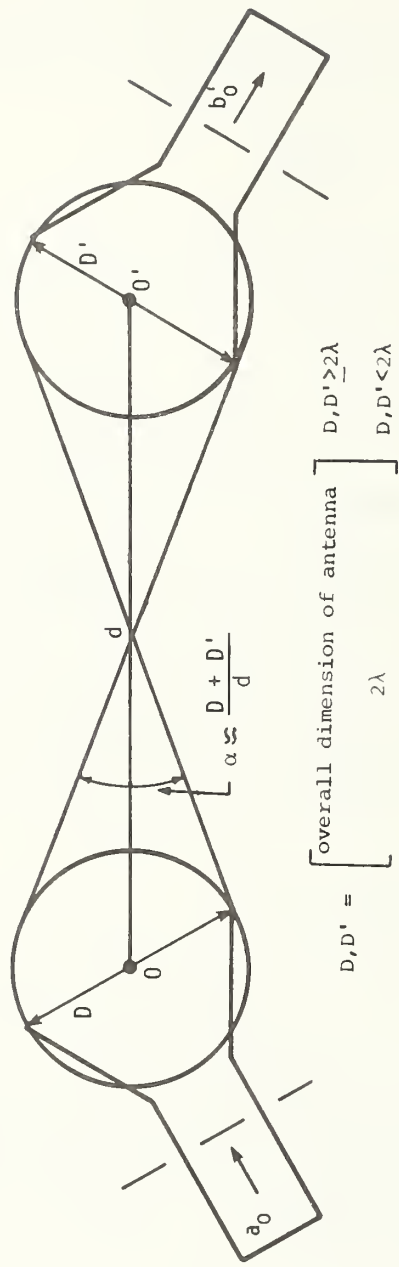


Figure 4. Physical interpretation for limits of integration. To a good approximation, only that portion of the spectrum within  $\alpha$  is required to compute the coupling quotient  $b'_0/a_0$  for the two antennas.

$$\frac{\Delta k_x}{k}, \frac{\Delta k_y}{k} = \text{smaller of } \left\{ \frac{\frac{\lambda}{2B(D+D')}}{\frac{(D+D')}{2Bd}} \right\}, \quad (30)$$

and from this equation and eq (26), the summation limits become

$$L, M \approx \text{larger of } \left\{ \frac{\frac{2B(D+D')^2}{\lambda d}}{2B} \right\}. \quad (31)$$

Note that when the separation  $d$  becomes larger than the "mutual Rayleigh distance,"  $(D+D')^2/\lambda$ , only a few  $(2B)$  points of integration are required, as one might expect from physical intuition because only the near-axis plane waves contribute to the coupling as the far field is approached.

### 1.3.3. Application of the Fast Fourier Transform

As mentioned above, eq (23) is amenable to computation by means of the efficient algorithm often referred to as the fast Fourier transform (FFT) [5]. The particular FFT algorithm we use is called FOURT and was written by Norman Brenner of MIT Lincoln Laboratories. FOURT, like all FFT algorithms, requires the summation in eq (23) to be written in a specific form, namely

$$\frac{b'_0(\underline{R}, d)}{a_0} = -C' e^{-ik(a_1 x + b_1 y)} \frac{(a_1 + a_2)(b_1 + b_2)}{N_1 N_2} \sum_{j_1=1}^{N_1} \sum_{j_2=1}^{N_2} A[j_1, j_2] e^{2\pi i \left[ \frac{(j_1-1)(m_1-1)}{N_1} + \frac{(j_2-1)(m_2-1)}{N_2} \right]} \quad (32)$$

The definition of the various parameters in eq (32) in terms of quantities defined previously can probably be best understood by referring back to eq (20). As usual,  $C'$  is the mismatch factor (defined after eq (8)), and  $(x, y)$  are the components of the transverse vector  $\underline{R}$ . The real numbers  $(a_1, a_2)$  and  $(b_1, b_2)$  define the limits of integration on  $k_x$  and  $k_y$ ; specifically,

$$-a_1 \leq \frac{k_x}{k} \leq a_2 \quad (33a)$$

$$-b_1 \leq \frac{k_y}{k} \leq b_2 \quad (33b)$$

$N_1$  and  $N_2$  are the number of terms in the  $k_x$  and  $k_y$  summations respectively, and are equal to  $(2M+1)$  and  $(2L+1)$  defined under eq (23). (In light of the discussion leading to eqs (26) and (31), for  $\underline{R}$  near zero,  $a_1, a_2, b_1$ , and  $b_2$  will all lie within a circle of radius  $k(D+D')/d$  ( $d > D+D'$ ) in the  $k_x, k_y$  plane; and  $N_1$  and  $N_2$  need be no larger than about twice the  $L, M$  given in eq (31).) The exponential immediately following  $C'$  in eq (32) arises from making the summation indices range only over positive integers.

In eq (32) the FFT will compute the double summation for the following values of  $x$  and  $y$ :

$$x = \frac{(-N_1/2+m_1-1)\lambda}{(a_1+a_2)} \quad (34a)$$

$$y = \frac{(-N_2/2+m_2-1)\lambda}{(b_1+b_2)} \quad , \quad (34b)$$

where

$$m_1 = 1, 2, \dots, N_1 \quad (35a)$$

$$m_2 = 1, 2, \dots, N_2 \quad . \quad (35b)$$

Finally, the matrix  $A(j_1, j_2)$  in eq (32) needs defining:

$$A(j_1, j_2) = \frac{k^2}{\gamma} \underline{f}'(\phi_p, \theta_p) \cdot \underline{f}(\phi_A, \theta_A) e^{i\gamma d} (-1)^{j_1+j_2} \quad , \quad (36)$$

where  $(\phi_p, \theta_p)$  and  $(\phi_A, \theta_A)$  are determined from the transformations (13) for given Eulerian angles and  $(k_x, k_y)$ , which are defined in terms of  $(j_1, j_2)$  by,

$$\frac{k_x}{k} = \frac{(a_1+a_2)}{N_1} (j_1-1) - a_1 \quad (37a)$$

$$\frac{k_y}{k} = \frac{(b_1+b_2)}{N_2} (j_2-1) - b_1 \quad . \quad (37b)$$

The  $(-1)^{j_1+j_2}$  factor in eq (36) arises from requiring the algorithm FOURT to yield the coupling quotient directly for every value of  $x$  and  $y$  without the need of "rearranging." The  $z$  component  $\gamma$  of the propagation vector is, of course, determined from  $k_x$  and  $k_y$  through a simple relation, which for completeness will be repeated here:

$$\gamma = \sqrt{k^2 - k_x^2 - k_y^2} \quad . \quad (38)$$

The dot product  $\underline{f}' \cdot \underline{f}$  is also computed as explained in section 1.2.2.

In short, eq (32) for the coupling quotient between two antennas is ready for efficient evaluation on the computer using the FFT algorithm FOURT.

#### 1.4. Preliminary Numerical Results

In order to build confidence in the computer program which was written to evaluate coupling products from eq (32), the far fields of two hypothetical antennas were inserted into the program. The hypothetical antennas were linearly polarized (in  $x$  direction), uniform, circular aperture antennas for which the complex far-field patterns are well known in terms of simple analytic expressions involving the first-order Bessel function [6]. The radius and operating frequency of the antennas could be chosen arbitrarily along with their mutual orientation and separation.

One check performed on the program is displayed graphically in figure 5, which shows the coupling quotient for two identical antennas facing each other in their very near field. Here the coupling should be very high, actually approaching unity when the antennas are directly aligned, as figure 5 confirms. (It should be mentioned that the curve in fig 5 and those in figs 6 and 7 took no more than a few seconds to compute.)

A second check of the computer program involves computing the coupling when the antennas are separated by a large enough distance for coupling to take place mainly between the far fields along the direction between the antennas. As mentioned in section 1.3.2., this critical distance which we call the "mutual Rayleigh distance" can be shown to be approximately  $(D+D')^2/\lambda$ . In figure 6 the coupling between the antennas is computed at this mutual Rayleigh distance for the antennas by two methods--first, by the FFT integration of eq (32), and then directly from the far-field coupling along the direction of separation. The close agreement between the two results again imbues confidence in the correctness of the coupling computer program.

Finally, figure 7 shows a typical coupling curve for two antennas skewed in the near field of each other. Note that a small lateral displacement appreciably less than an antenna diameter can make a 20 dB or more change in coupling.

In summary, the results of these and numerous other sample computations with hypothetical circular antennas yielded reasonable curves in every case; thus, we entered the experimental stage of the program, confident of the reliability of the computer program.

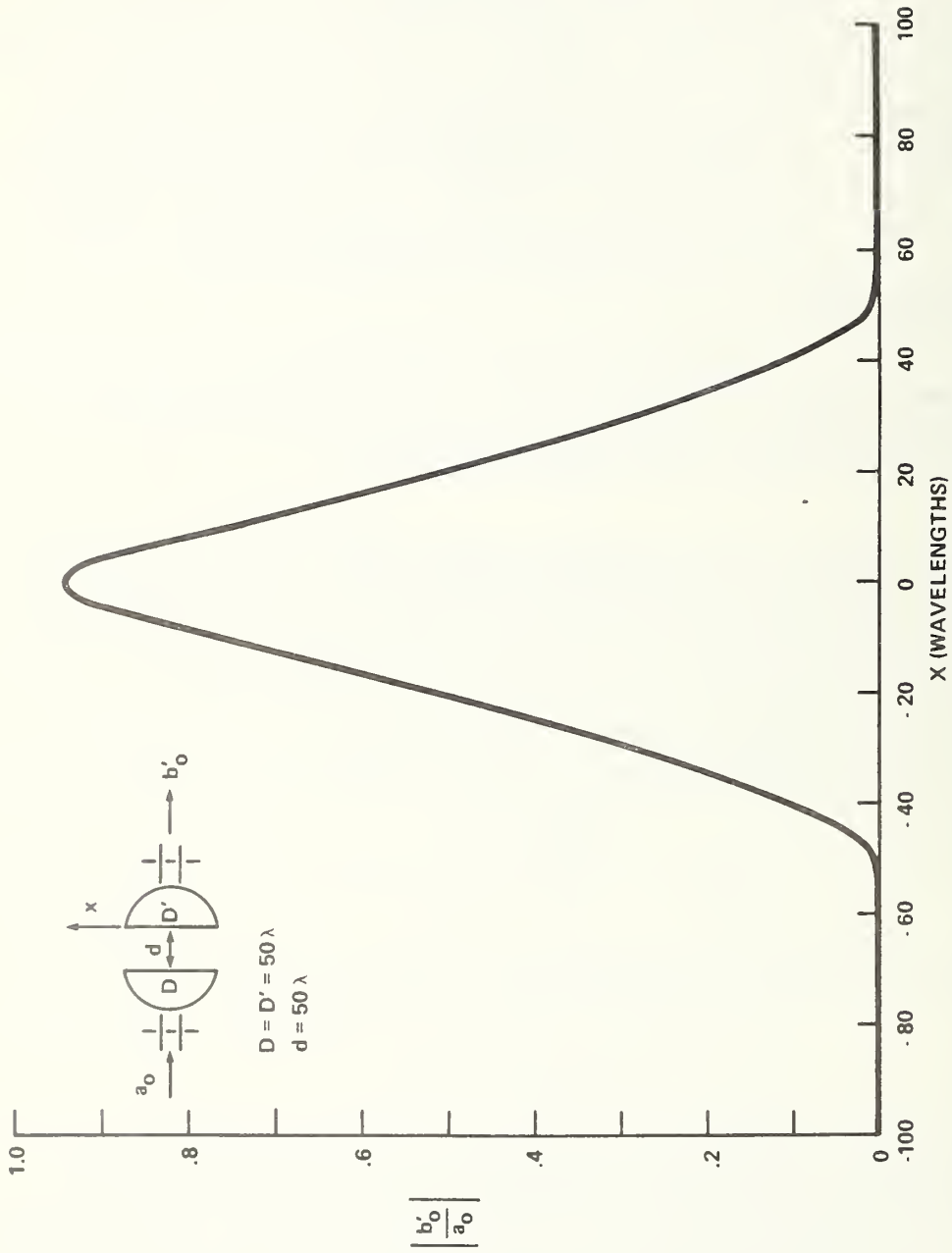


Figure 5. Hypothetical circular antennas directly facing each other in the near field.

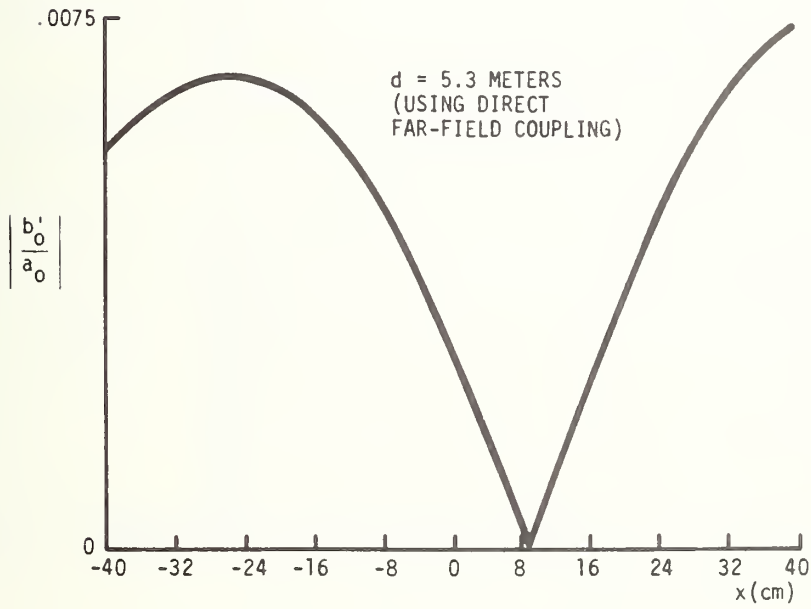
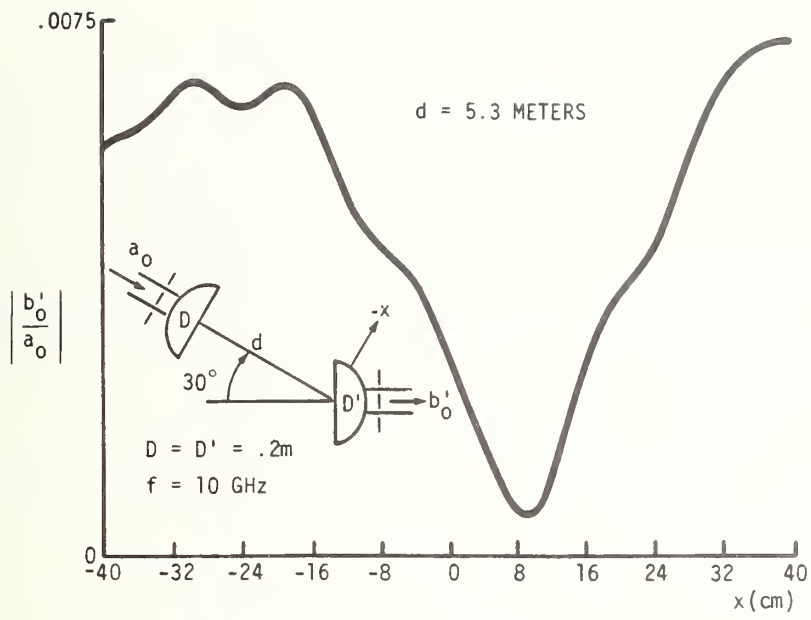


Figure 6. Coupling of circular antennas computed first using FFT integration, and then directly from far field along direction of separation.

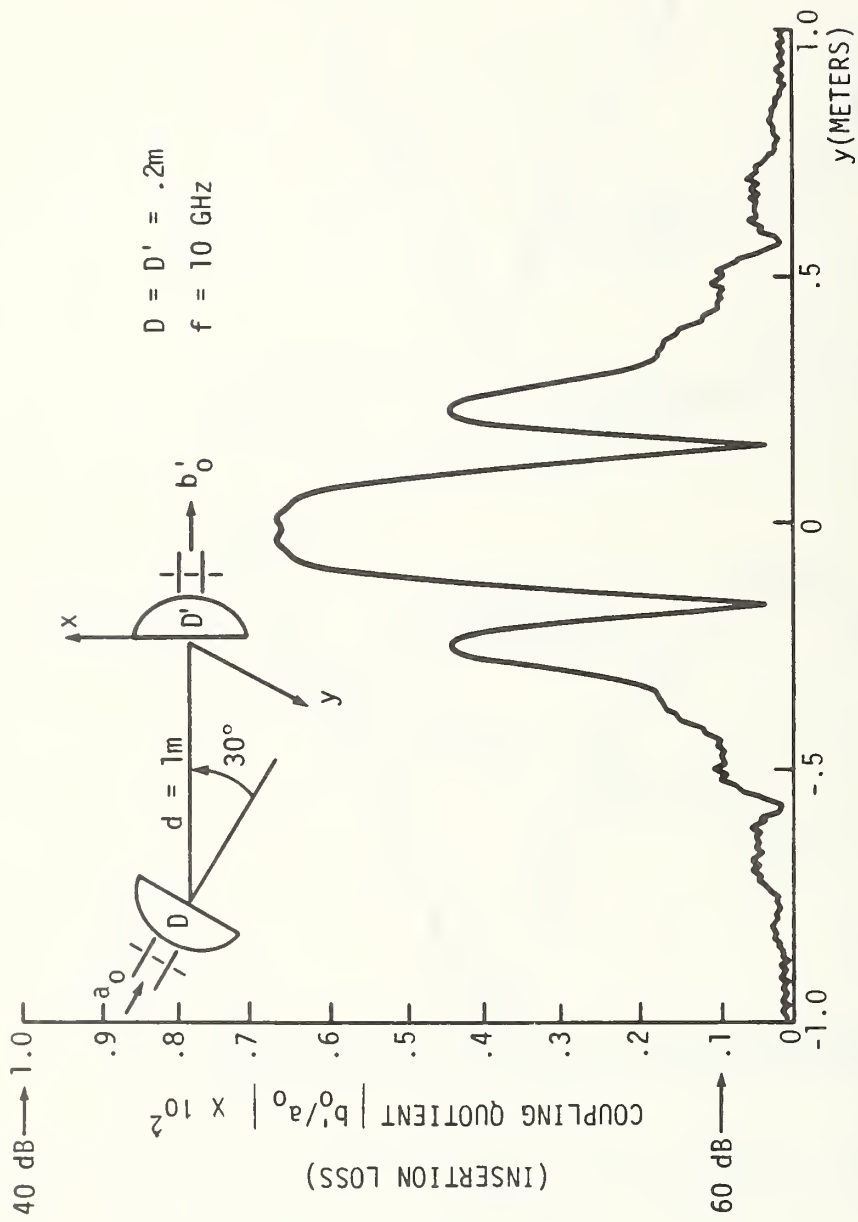


Figure 7. Typical coupling curve for antennas skewed in their near field.



## 2. TRANSFORMATION FROM FAR FIELD TO NEAR FIELD

This section details the theory which underlies the transformation from far field to near field. As in the case of coupling between antennas, the techniques are based on the scattering matrix theory of antennas developed at NBS. A brief review of the points applicable to the calculation of near fields is presented here. For a more thorough discussion, see Kerns [1b].

We consider a finite antenna system which is located between the planes  $z = z_1$  and  $z = z_2$ ;  $z_1 < z_2$ . The fields to the right of plane  $z_2$  can be expressed by a superposition of plane waves in the following form

$$\underline{E}(\underline{r}) = \frac{1}{2\pi} \iint_{-\infty}^{\infty} [\underline{b}(\underline{K})e^{i|\gamma|z} + \underline{a}(\underline{K})e^{-i|\gamma|z}] e^{i\underline{K}\cdot\underline{R}} d\underline{K}, \quad (39)$$

where

$\underline{b}(\underline{K})$  is the spectral density function for plane waves travelling to the right (outgoing);

$\underline{a}(\underline{K})$  is the spectral density function for plane waves travelling to the left (incoming);

$\underline{K} = k_x \hat{e}_x + k_y \hat{e}_y$  is the transverse propagation vector;

$\gamma = (k^2 - k_x^2 - k_y^2)^{1/2} = (k^2 - k^2)^{1/2}$  is positive real or imaginary.

$k^2 = \omega^2 \mu \epsilon$ ; and

$d\underline{K} = dk_x dk_y$ .

Each plane wave is specified by its propagation vector

$$\underline{k}^{\pm} = k_x \hat{e}_x + k_y \hat{e}_y \pm \gamma \hat{e}_z = \underline{K} \pm \gamma \hat{e}_z.$$

Further, each component satisfies the transversality relation

$$\underline{k}^+ \cdot \underline{b} = 0; \quad \underline{k}^- \cdot \underline{a} = 0.$$

We note that eq (1) indicates a Fourier transform relation exists between the electric field and the spectrum.

A surprisingly simple relationship exists between the far-field radiation from a finite antenna and its spectrum, as noted in section 1.1.2, and is given by

$$\underline{E}^r(\underline{r}) = -i\gamma \underline{b}(\underline{Rk}/r) e^{ikr}/r. \quad (40)$$

Hence, knowledge of the far-field pattern immediately permits calculation of the spectrum, from which we can calculate the near-field pattern at any point using eq (39).

For our purposes here, we consider an antenna radiating into free space; hence, there are no waves travelling left for  $z > z_2$ . Thus,  $\underline{a}(\underline{k}) \equiv 0$  and eq (39) becomes

$$\underline{E}(\underline{r}) = \frac{1}{2\pi} \iint_{-\infty}^{\infty} C_1 \frac{\underline{E}^r(\underline{r})}{\gamma} e^{i|\gamma|z} e^{i\underline{k} \cdot \underline{R}} d\underline{k}. \quad (41)$$

$C_1$  has been introduced as a constant which normalizes the magnitude of the far field. It will be evaluated in the following section.

## 2.1 Relationship of Near-Field Intensities to Power Input and Antenna Gain or Efficiency

The constant  $C_1$  will be determined by the power input to the antenna and the intrinsic properties of the antenna itself. We will let the property be the antenna gain as it is the one most often measured or specified. In the case of a reflector antenna, with  $\underline{E}(\underline{r})$  determined by a mathematical model, we use the physical size and efficiency to provide the appropriate normalization.

Recall that, for a single antenna radiating into free space

$$\begin{aligned} \underline{E}(\underline{r}) &= \frac{1}{2\pi} \iint_{-\infty}^{\infty} \underline{b}(\underline{k}) e^{i|\gamma|z} e^{i\underline{k} \cdot \underline{R}} d\underline{k} \\ &= \frac{a_0}{2\pi} \iint_{-\infty}^{\infty} \underline{s}_{10}(\underline{k}) e^{i|\gamma|z} e^{i\underline{k} \cdot \underline{R}} d\underline{k}. \end{aligned} \quad (42)$$

Further, as shown by Kerns, the gain of an antenna is given by

$$G(\underline{K}) = \frac{4\pi Y_0 \gamma^2 |\underline{s}_{10}(\underline{K})|^2}{\eta_0 (1 - |\Gamma_0|^2)}, \quad (43)$$

where, as in section 1,  $Y_0 = 1/Z_0$  is the admittance of free space,  $\eta_0$  is the characteristic admittance of the feed mode, and  $\Gamma_0$  is the antenna input reflection coefficient.

Now we are interested in normalizing our calculation to the gain in a single direction. This is usually the boresight or "on axis" direction (though in the case of a monopulse difference pattern we may need to specify the gain in a different direction.) For the antennas and models considered in this study, however, the boresight direction corresponds to the peak of the main lobe and thus makes a convenient normalization point. Solving for  $\underline{s}_{10}(\underline{K}=0)$  in terms of the boresight gain and substituting into eq (42) gives

$$\underline{E}(\underline{r}) = \frac{a_0}{2\pi} \sqrt{\frac{\eta_0 (1 - |\Gamma_0|^2) G(0)}{4\pi Y_0 k^2}} \int \hat{\underline{s}}_{10}(\underline{K}) e^{i|\underline{r}|k} e^{i\underline{K} \cdot \underline{R}} d\underline{K}, \quad (44)$$

where

$$\hat{\underline{s}}_{10}(\underline{K}) = \frac{\underline{s}_{10}(\underline{K})}{|\underline{s}_{10}(0)|}.$$

Now, for an antenna connected to a source which delivers an average power input  $P_0$ , we have

$$P_0 = \frac{1}{2} \eta_0 (|a_0|^2 - |b_0|^2),$$

but because  $b_0 = \Gamma_0 a_0$

$$P_0 = \frac{1}{2} \eta_0 |a_0|^2 (1 - |\Gamma_0|^2).$$

Substituting this into eq (44) gives

$$\underline{E}(\underline{r}) = \frac{1}{2\pi} \sqrt{\frac{P_0 G(0)}{2\pi Y_0 k^2}} \int \hat{\underline{s}}_{10}(\underline{K}) e^{i|\underline{r}|k} e^{i\underline{K} \cdot \underline{R}} d\underline{K}. \quad (45)$$

For the case of an antenna pattern determined from a model, we may estimate the gain of the antenna from its physical size and assumed efficiency. The receiving cross section  $\sigma$ , can be related to its physical area by the expression

$$\sigma = \eta A ,$$

where

$\eta$  = aperture efficiency

$A$  = physical area of the antenna.

Further, for a reciprocal antenna, gain and receiving cross section are related by

$$G = \frac{4\pi\sigma}{\lambda^2} .$$

Finally, for a circular antenna we have

$$G = \eta \pi^2 d_\lambda^2 ,$$

where  $d_\lambda = \frac{d}{\lambda}$  is the diameter expressed in wavelengths.

### 3. PHYSICAL OPTICS MODEL FOR REFLECTOR ANTENNAS

In order to calculate the radiated fields of a reflector antenna, it is necessary to employ some sort of approximate theory because an exact solution is essentially impossible to complete. Of several approximate theories, the one most appropriate for prediction of the antenna is main beam and near sidelobes is physical optics (PO). For farther out sidelobes, better results can usually be obtained from asymptotic theories such as the geometrical theory of diffraction (GTD).

The model employed in this work was physical optics and the basic theory will be discussed here. Several good references are available on the subject of physical optics. Here, we follow the development of Rusch [8,9].

As is well known, the fields in space can be calculated if all currents are known. A general expression for these fields can be written in terms of the free-space dyadic Green's function [10]. This expression is quite complicated if we want to calculate fields at any point. However, if we desire only "far-field" expressions, considerable simplification can be made.

We consider an arbitrary conducting surface  $S$  with surface current density  $\underline{J}_S$ , as illustrated in figure 8.

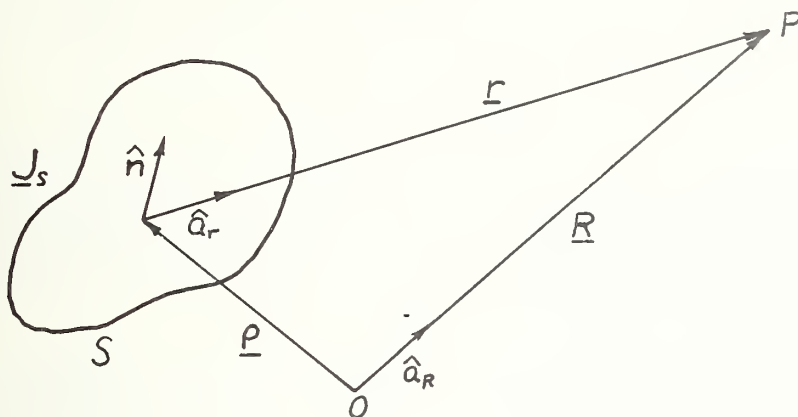


Figure 8. Geometry of vectors for surface integral.

Here,  $O$  is the origin of the reference coordinate system,  $P$  is the field point,  $\underline{R}$  is a vector which locates  $P$  in the reference system, and  $\hat{a}_R$  is a unit vector parallel to  $\underline{R}$ . The integration point is located by the vector  $\underline{\rho}$ , while the vector  $\underline{r}$  designates the location of  $P$  with respect to the integration point and  $\hat{a}_r$  is a parallel unit vector.

Now, under the usual far-field assumptions  $r \gg \lambda$  and  $|\rho|_{\max} \ll R$  or  $r$ , we can write the electric field at  $P$  as

$$\underline{E}(\underline{R}) = \frac{i\omega\mu_0}{4\pi} \frac{e^{ikR}}{R} \int_S [\underline{J}_S - (\underline{J}_S \cdot \hat{a}_R) \hat{a}_R] e^{-ik\underline{\rho} \cdot \hat{a}_R} dS. \quad (46)$$

This expression can be evaluated relatively easily using numerical techniques, provided that  $\underline{J}_S$  is known. The crux of the problem, then, is the evaluation of  $\underline{J}_S$ .

A useful approximate theory for obtaining  $\underline{J}_S$  is PO. Simply stated, PO approximates the surface currents with those that are obtained by the assumption of a local plane-wave reflection field, i.e.,

$$\underline{J}_S = 2[\hat{n} \times \underline{H}_{inc}], \quad (47)$$

where  $\hat{n}$  is the unit normal to the surface and  $\underline{H}_{inc}$  is the incident magnetic field.

Numerical evaluation of the two-dimensional integral in eq (46) can be time consuming for many cases. The size of the cell required to obtain a given accuracy with the numerical integration scheme decreases as the observation point moves off axis, and may

approach a small fraction of a wavelength. Thus, we see that calculation of the fields off axis for a large aperture antenna requires a large number of points. Further, the near-field calculations which are to be performed using the far-field patterns require a large number of individual far-field calculations.

In order to arrive at a practical model, some simplifications must be employed. The model, which is employed by the USC programs, assumes that the reflector is axially symmetric. This assumption allows the performance of the azimuthal integration in eq (46) analytically, thus reducing drastically the number of points required in the integration. Details of this simplification may be found in Rusch [8].

Another consequence of the assumption of axial symmetry is that a complete far-field pattern (i.e., specification for all values of  $\vartheta$ ) requires that the field be calculated only in the E- and H-planes, i.e.,  $\vartheta = \pi/2$  and 0, respectively. The field at any point  $(R, \theta, \vartheta)$  is given by

$$\underline{E}(R, \theta, \vartheta) = \frac{e^{-ikR}}{R} [E_E(\theta) \sin\vartheta \hat{a}_\theta + E_H(\theta) \cos\vartheta \hat{a}_\vartheta]. \quad (48)$$

For the purposes of this study, we require the rectangular components of the antenna pattern, which are given by

$$\begin{aligned} \underline{E} = & \frac{e^{-ikR}}{R} [E_E(\theta) \cos\theta - E_H(\theta)] \cos\vartheta \sin\vartheta \hat{a}_x \\ & + [E_E(\theta) \cos\theta \sin^2\vartheta + E_H(\theta) \cos^2\vartheta] \hat{a}_y - E_E(\theta) \sin\theta \sin\vartheta \hat{a}_z. \end{aligned} \quad (49)$$

### 3.1 Physical Optics Subroutines Employed by USC

The subroutines used to compute the PO fields of the paraboloidal reflector antennas were written by Prof. W. V. T Rusch, of the University of Southern California and obtained at a short course, Reflector Antenna Theory and Design, given in June 1976.

The subroutine package will calculate far-field patterns for an axially symmetric reflector antenna which has a circular blockage on axis caused by the feed. Further, it allows the feed pattern to be specified in the E- and H-planes independently to control the reflector illumination function.

Three options are available for the feed pattern. These are: uniform illumination, dipole illumination, and  $\cos^n\theta'$  illumination where  $\theta'$  is the angle measured from the feed axis. For this case, the feed patterns in the E- and H-planes are given by

$$E_E = \cos^n \theta,$$

$$E_H = \cos^n \theta'.$$

Other parameters of the antenna which are required as input include focal length to diameter ratio, fractional diameter blockage, diameter in units of wavelength, and axial position of the feed relative to the focal point of the reflector.

The subroutines use a Romberg type of algorithm to perform the necessary integrations. This is an adaptive algorithm in the sense that it selects the necessary interval size based on a required accuracy. The result is a rapidly executing program, because advantage can be taken of the fact that rather large increments can be used near the main beam, thus reducing time to compute the far fields for these points.

If the integration routine is unable to achieve the required accuracy, either because of accumulated round-off error or because the integration range cannot be sufficiently subdivided, an appropriate error flag is set. This condition is noted in the program output, so that this data may be deleted in further calculations. Further discussion of these errors occurs in the program description.

### 3.2 Test of Near-Field Program

In order to check the operation of the near-field transformation in conjunction with the far-field PO model, a test case consisting of a 52-wavelength, uniformly illuminated aperture was run. Near fields were calculated in the aperture plane from the far fields calculated using PO, and were compared with the original uniform distribution. Results are shown in figure 9. As can be seen, the calculated results agree well with the uniform distribution. Note that the scale is electric field in volts/meter, not relative field in dB. Total variation from the original distribution is +1.1 dB, -0.55 dB.

The ripple can be attributed to several causes. Since the PO program encounters round-off error problems for angles which lie too far off boresight, the far field must be truncated beyond a critical angle. For this example, the truncation occurred at an angle of 10.2 degrees, which was also chosen because it was a null position. Even so, eight sidelobes were included in the far-field pattern, the last one having an amplitude of about -40 dB relative to the main beam. The spacing of far-field points also affects the ripple to some extent. Here, there were about 10 points per sidelobe. Finally, evanescent modes were neglected because of the point spacing chosen in k-space. The results do indicate that useful near fields can be calculated from the model for this case.

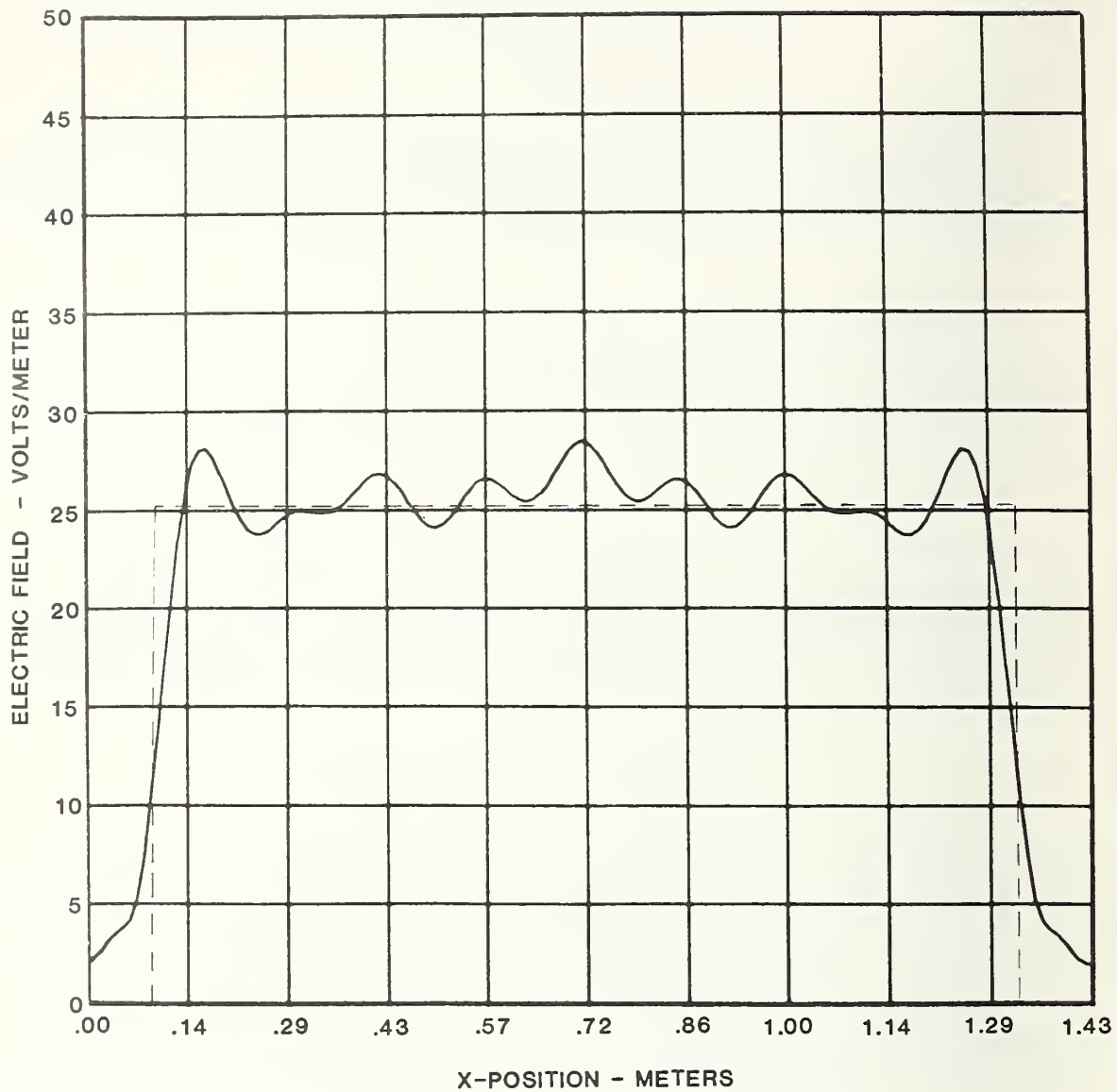


Figure 9a. Field strength in a uniformly illuminated aperture calculated using physical optics far fields. Dashed line indicates theoretical distribution.



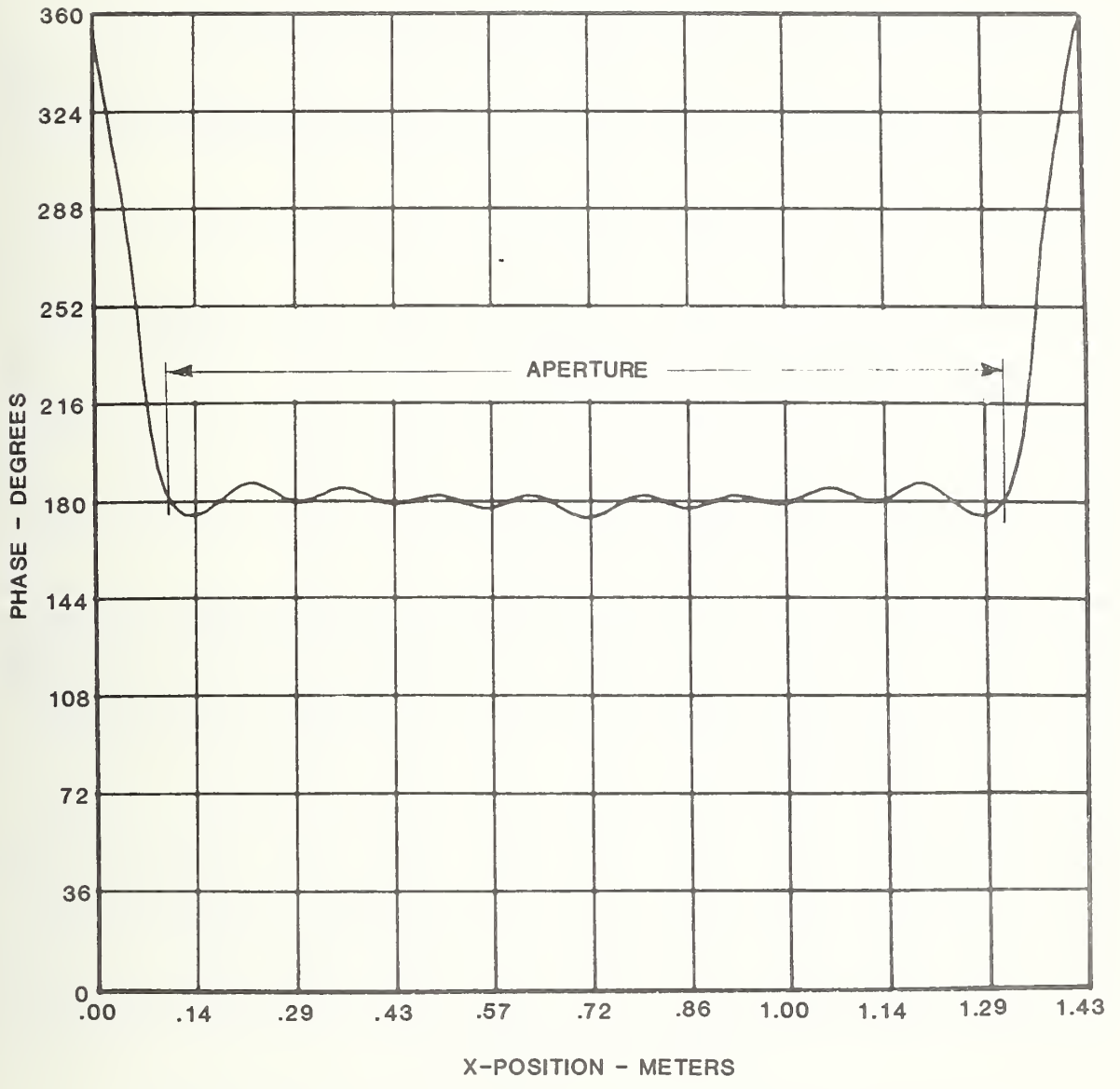


Figure 9b. Phase of field in a uniformly illuminated aperture calculated using physical optics for fields.

#### 4. COMPARISON OF PHYSICAL OPTICS AND MEASURED FAR FIELDS

As noted in section 3, the PO model represents an approximation to the true fields generated by the reflector antenna. Because of the approximations involved, it was considered desirable to compare the results obtained using a PO model to actual measured far-field patterns. Four cases were considered, and some additional experimental work was done in one case to attempt to determine the cause of observed discrepancies. The four cases are listed in table 4.1.

TABLE 4.1

Antenna	Frequency GHz	Diameter m( $\lambda$ )	Fractional Aperture Blockage	$n^E$	$n^H$	Measured Gain dB
1	4.0	1.22(16.25)	.164	1.57	1.72	29.66
2	4.0	1.22(16.25)	.164	1.02	1.07	28.34
3	12.73	1.22(51.8)	.143	1.09	1.09	40.70
4	57.5	.45(87.5)	.120		1.10	46.3

Each antenna had an essentially circular blockage at the feed, and each had three support struts. Antennas 1, 2, and 3 were essentially identical, being built by the same manufacturer, the only difference being in the feed. The feeds of antennas 1 and 2 were adjusted in the NBS near-field facility to obtain optimum focus and coincidence of electrical and mechanical axes.

The adjustment procedure consisted of moving the feed axially and laterally in order to obtain a minimum near-field phase curvature (focus adjustment) and a near-field phase with no linear component (boresight adjustment). It should be noted that for antennas 1 and 2, at least, it was not possible to obtain a flat phase front in both E- and H-planes. A compromise adjustment was made. Thus, either the E- or H-plane pattern can be somewhat improved, but only at the expense of a worse pattern in the other plane. It is not known whether the problem exists in the case of antennas 3 and 4, as these antennas had been previously measured at NBS and were not available for further experimentation.

In order to determine the parameters  $n^E$  and  $n^H$  for antenna 3, the dimensions of the feed were obtained and the patterns estimated using standard horn theory. For antenna 4, a cassegrain antenna, the near-field data obtained were used to estimate the parameters when the antenna was calibrated at NBS. For antennas 1 and 2, the feed patterns were measured on a far-field range before the feeds were installed on the reflector.

The far-field patterns for these antennas are shown in figures 10 to 13, with the PO predicted patterns superimposed. We note that, in general, the agreement between the

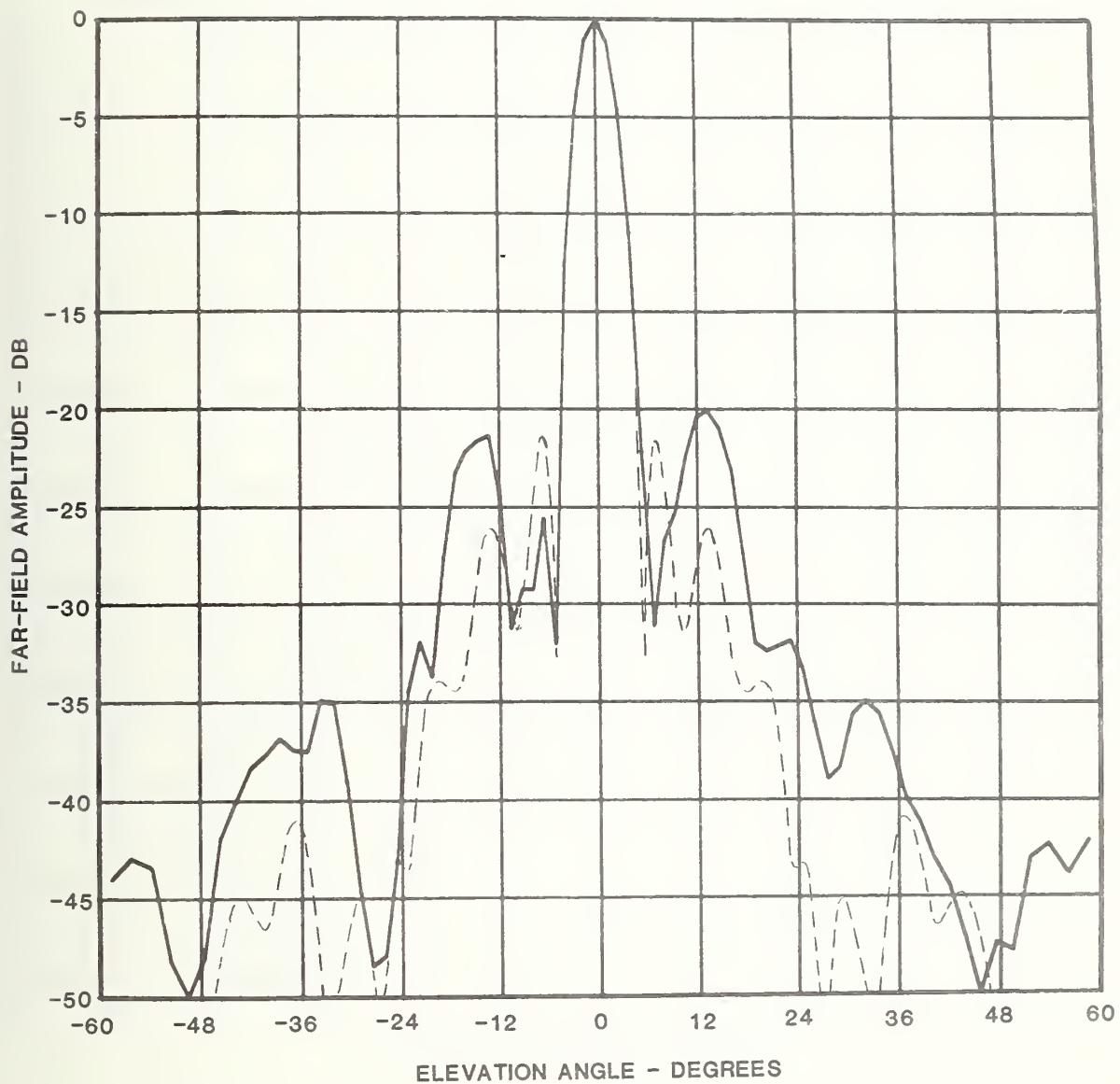


Figure 10a. Comparison of measured and calculated far-field patterns for antenna No. 1. E-plane cut, solid line - measured pattern, dashed line - physical optics.

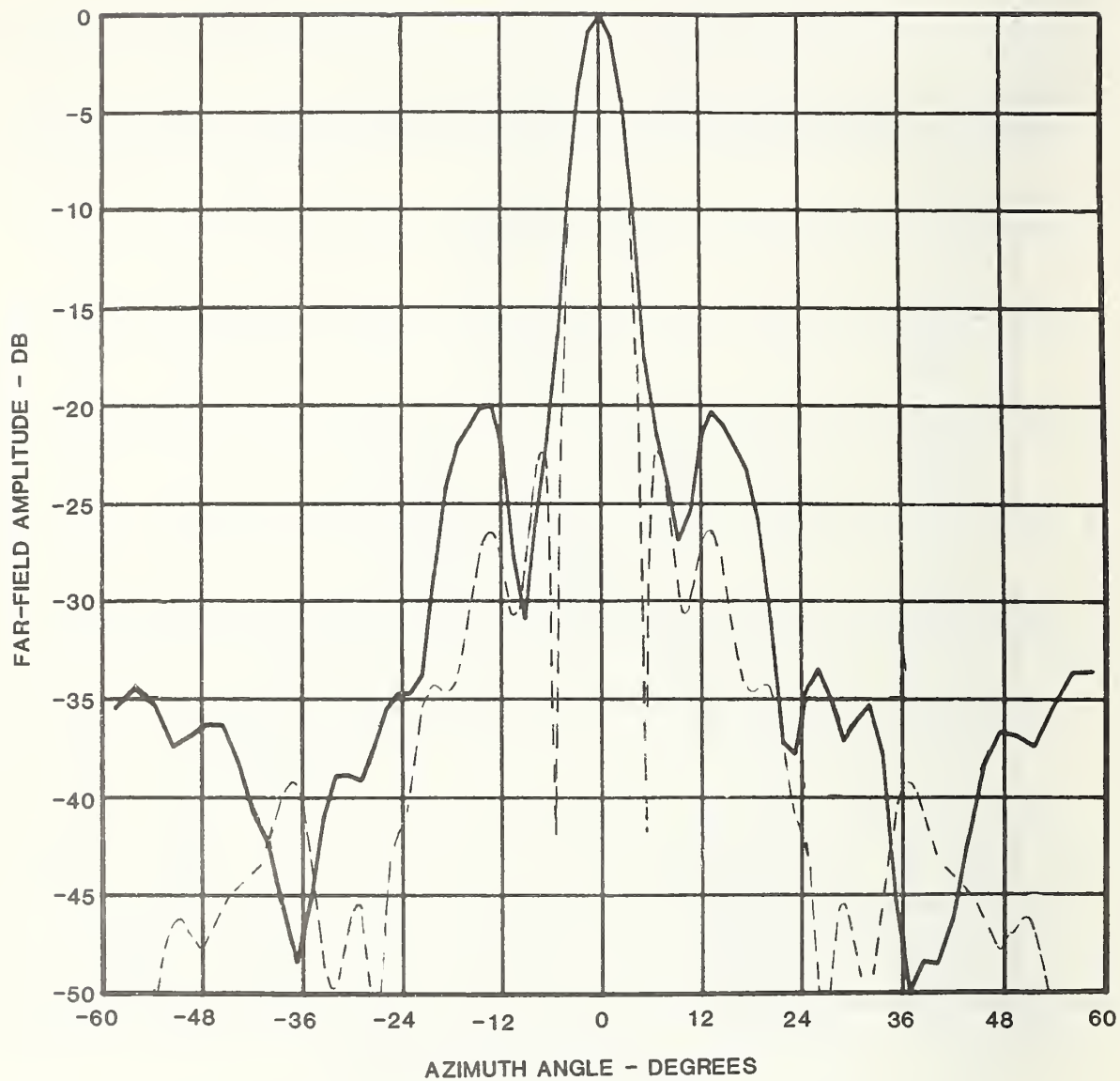


Figure 10b. Comparison of measured and calculated far-field patterns for antenna No. 1. H-plane cut, solid line - measured pattern, dashed line - physical optics.

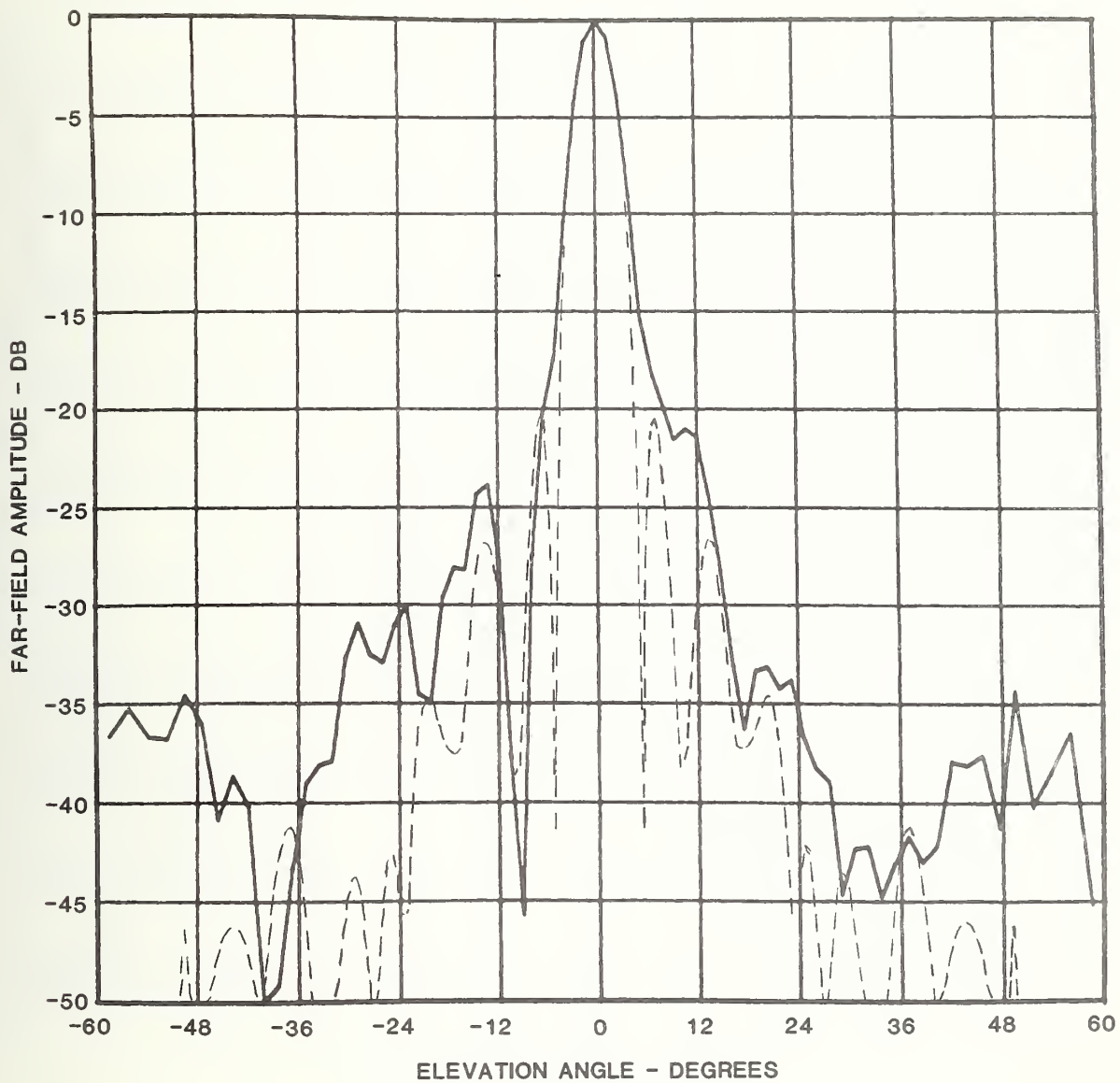


Figure 11a. Comparison of measured and calculated far-field patterns for antenna No. 2. E-plane cut, solid line - measured pattern, dashed line - physical optics.

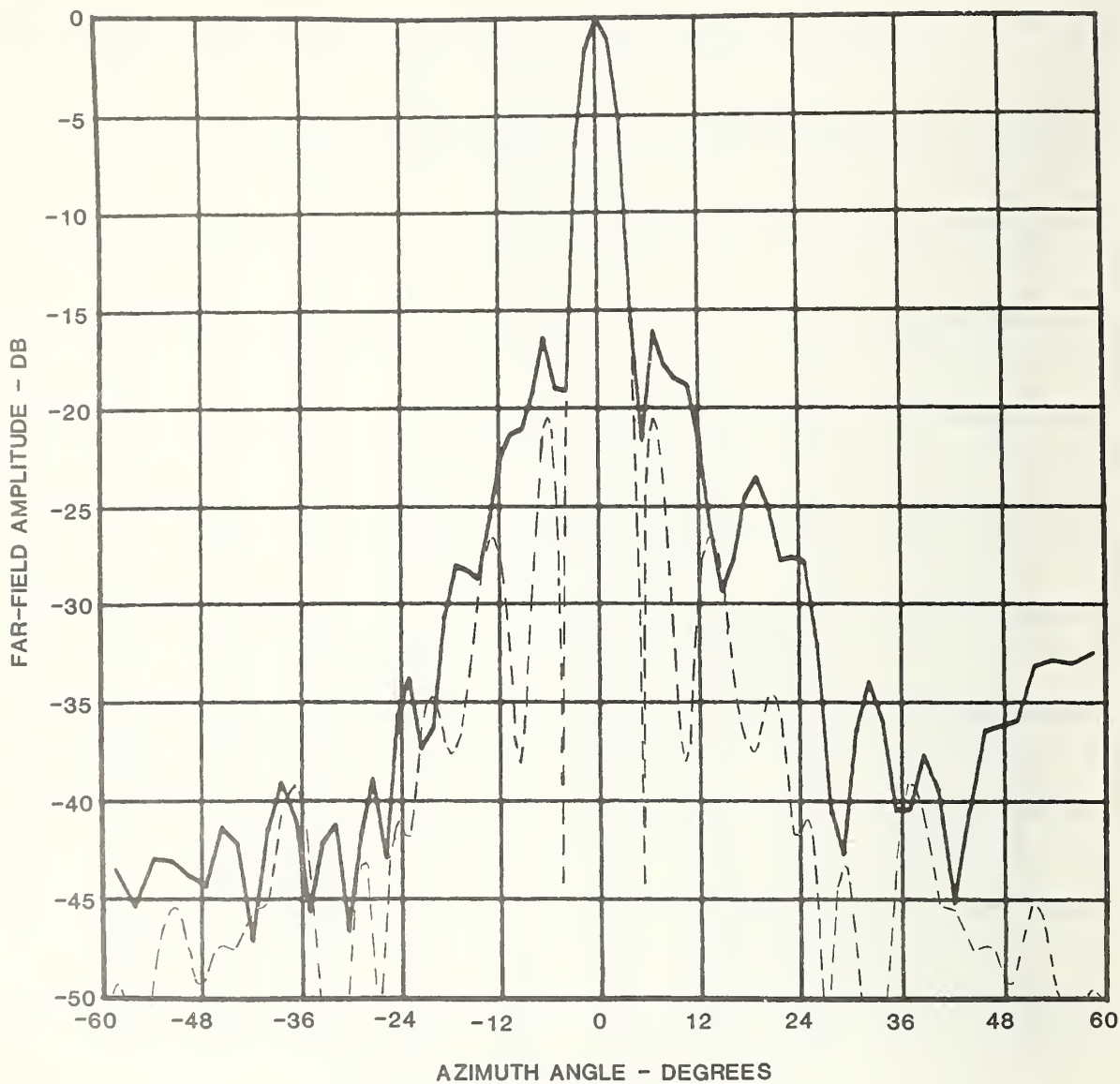


Figure 11b. Comparison of measured and calculated far-field patterns for antenna No. 2. H-plane cut, solid line - measured pattern, dashed line - physical optics.

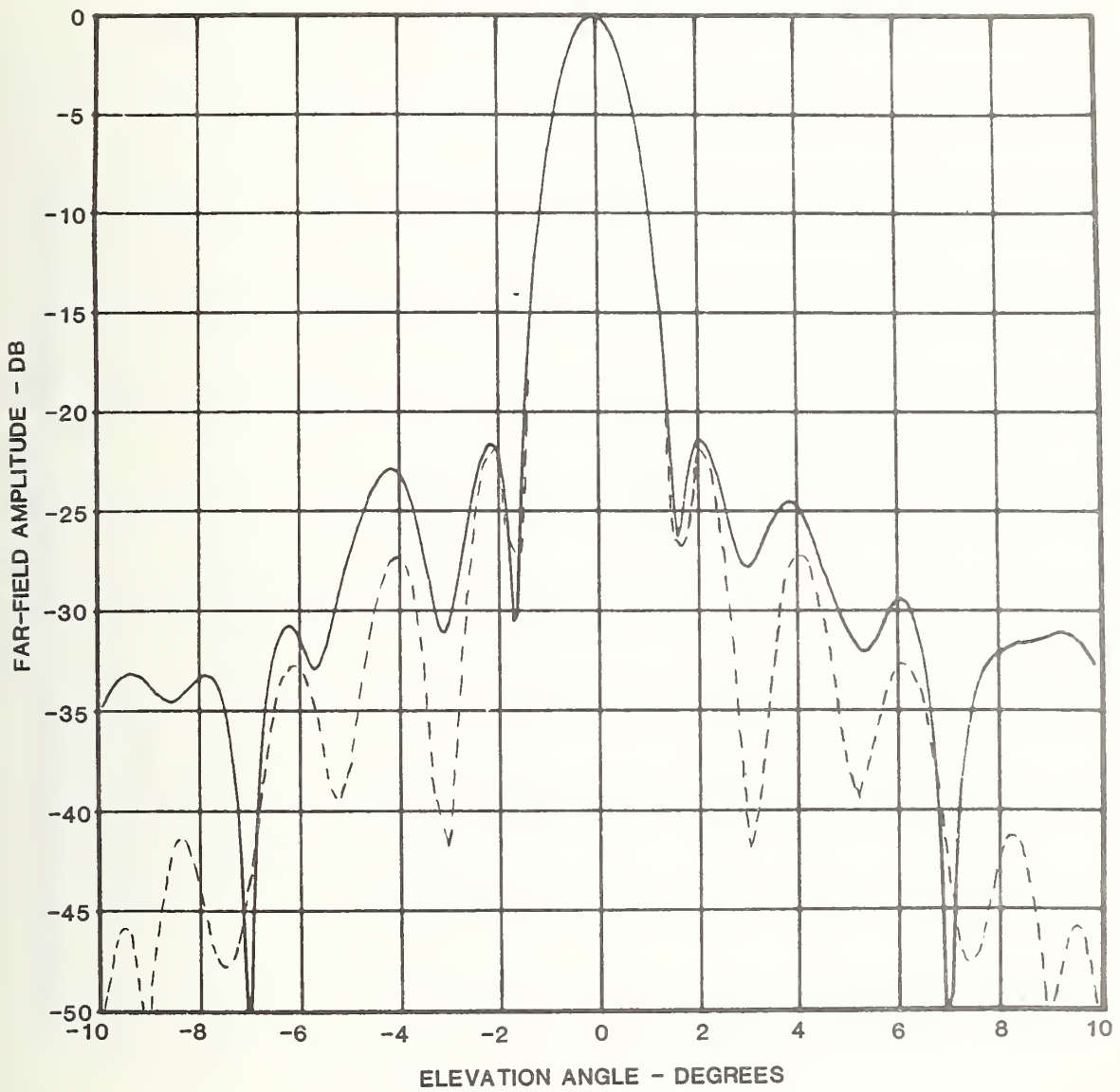


Figure 12a. Comparison of measured and calculated far-field patterns for antenna No. 3. E-plane cut, solid line - measured pattern, dashed line - physical optics.

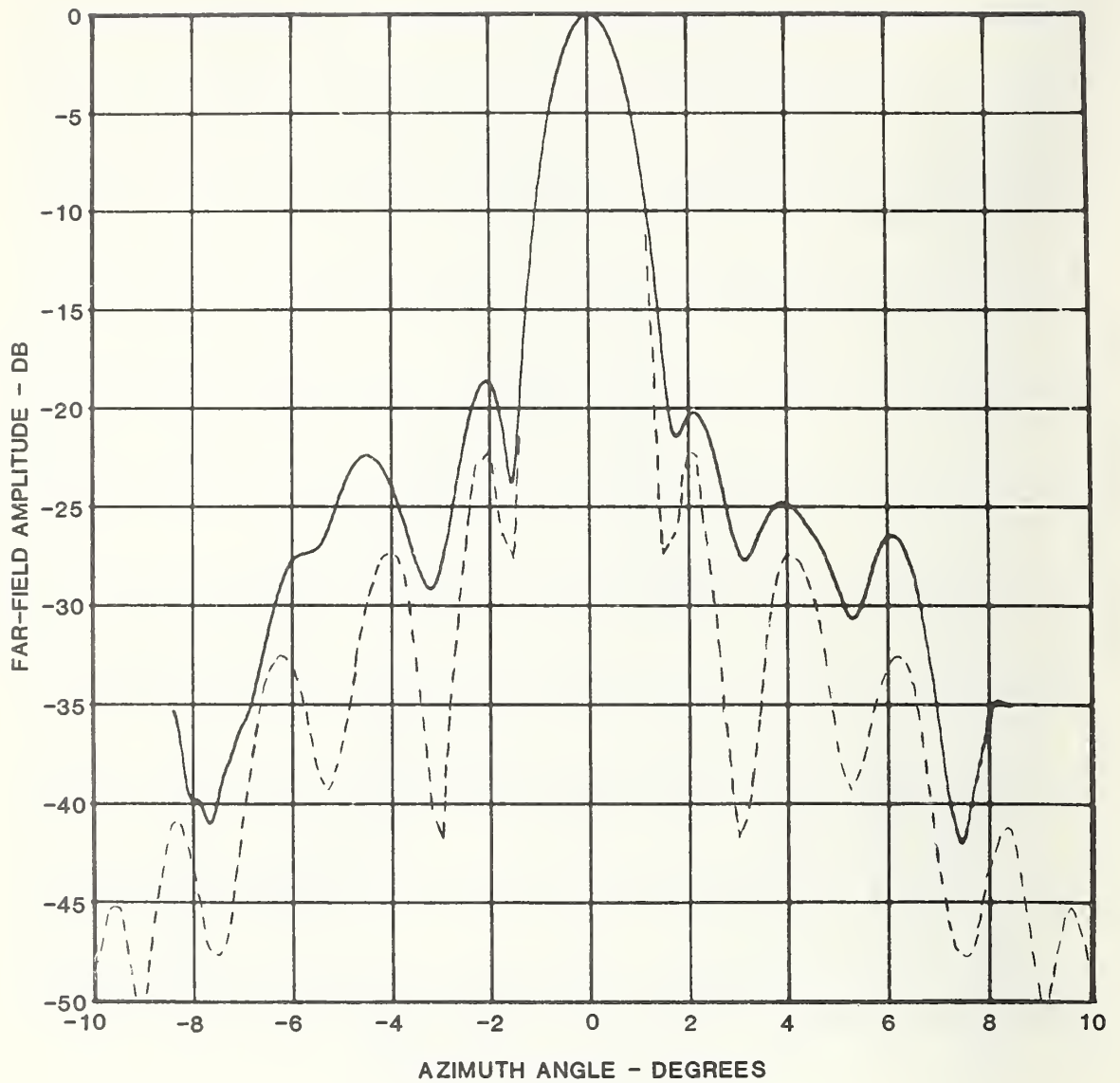


Figure 12b. Comparison of measured and calculated far-field patterns for antenna No. 3. H-plane cut, solid line - measured pattern, dashed line - physical optics.



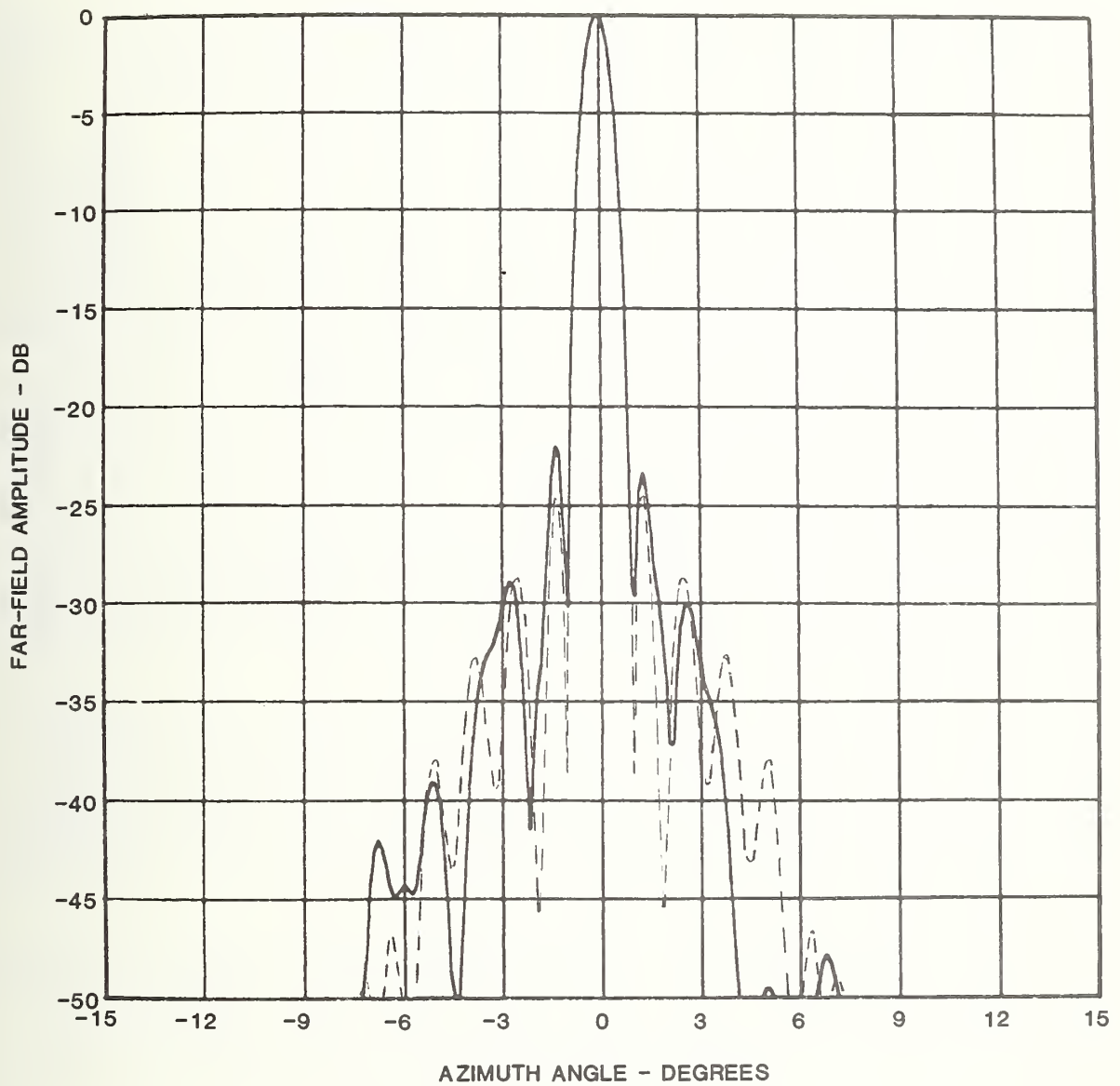


Figure 13. Comparison of measured and calculated far-field patterns for antenna No. 4. H-plane cut, solid line - measured pattern, dashed line - physical optics.

PO computations and measurements improves as the diameter to wavelength ratio increases; and further, by comparing 1 and 2, we note that a higher value of edge illumination seems to allow a better prediction.

Several possible explanations for the discrepancies exist. These can be grouped into five categories: edge effects, diffraction by struts, aperture blockage effects, back and sidelobe radiation from the feed, and violation of the assumed circular symmetry.

The first of these arises because of the sharp discontinuity in current which occurs at the edge of the reflector surface. The effect of this discontinuity is imperfectly accounted for by the PO model. In order to better describe edge effects, it is necessary to employ the geometrical theory of diffraction (GTD) or similar asymptotic theories to predict more accurately the sidelobes generated by these edge effects. To clearly see the difference between the edge as described by PO and GTD, it is useful to consider the "effective" currents which are used. These are illustrated in figure 14. We note that, in both cases, there is a sharp discontinuity in current density at the edge of the reflector surface. The GTD model includes the effect of the singularity in the current at an edge. GTD models usually assume a sharp edge. However, the antennas used in this study were made with a rolled edge as is common; and thus the normal GTD theory will not apply. The effect of the edge singularity manifests itself more as the angle off boresight increases. It is thus assumed that the use of PO rather than GTD is not significant in explaining the observed discrepancies.

The remaining processes are more likely candidates for the observed discrepancies. While blockage is taken into account, diffraction from the feed structure is not. In addition, because of the structure of the particular antennas used, multiple reflections between the feed structure and the reflector surface are likely to occur. An approximate cross section is shown in figure 15.

In order to test the multiple reflection hypothesis, the feed support plate was lined with rf-absorbing material, and near-field scans were again taken. The resulting far fields are shown in figure 16. Note that the agreement between the PO model and measured far fields is better. This suggests that at least part of the problem is in neglecting multiple reflections between the feed housing and the reflector.

The struts were now covered as shown in figure 17 to try to minimize diffraction by them. Results of this test showed an increase in the discrepancy between experiment and theory as shown in figure 18. However, this should not be taken to mean that strut reflection is negligible because, as can be noted in the photograph, there is significantly more blockage for rays travelling off axis than in the uncovered strut case. A better method for determining the strut diffraction effect experimentally would be to support the feed with dielectric material and measure patterns in this configuration. The asymmetry observed in the E-plane pattern is an indication of significant strut effects.

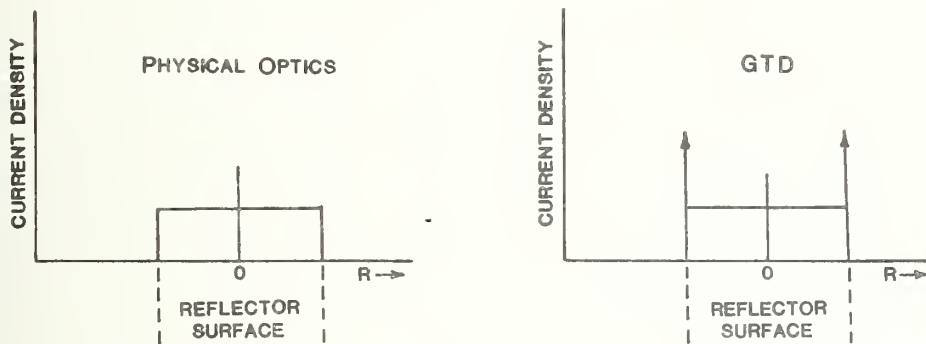


Figure 14. Comparison of effective current distribution used in physical optics and geometrical theory of diffraction calculations. (Uniform distribution assumed).

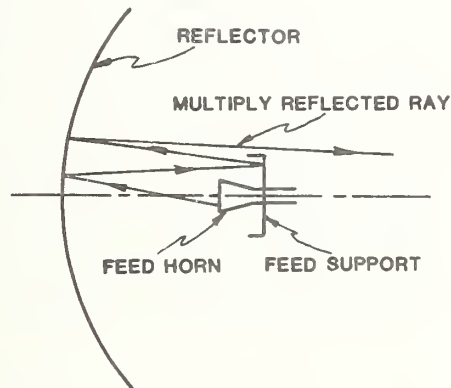


Figure 15. Diagram of multiple reflections involving feed structure.

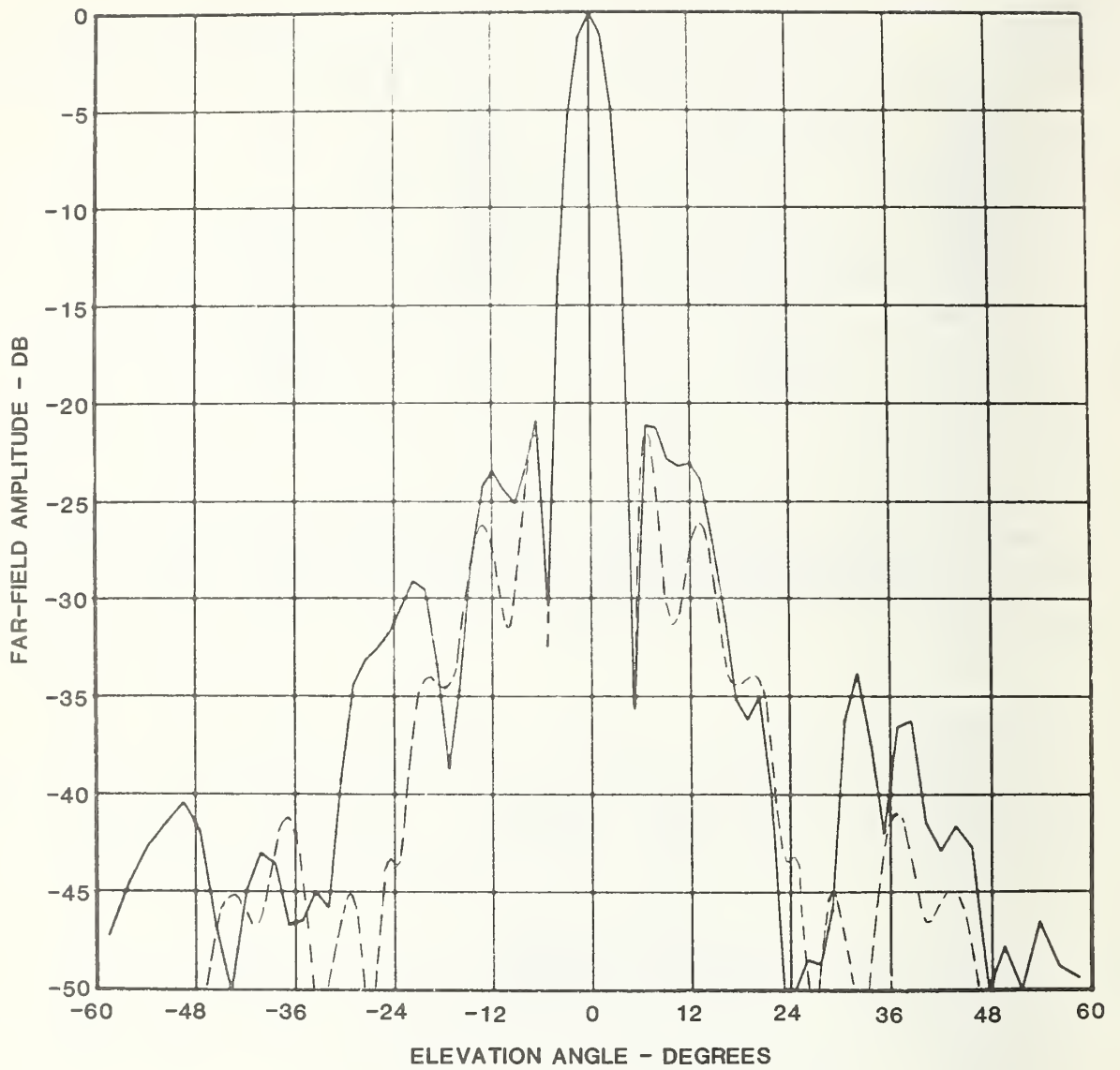


Figure 16a. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. E-plane cut, solid curve - measured pattern, dashed curve - physical optics.

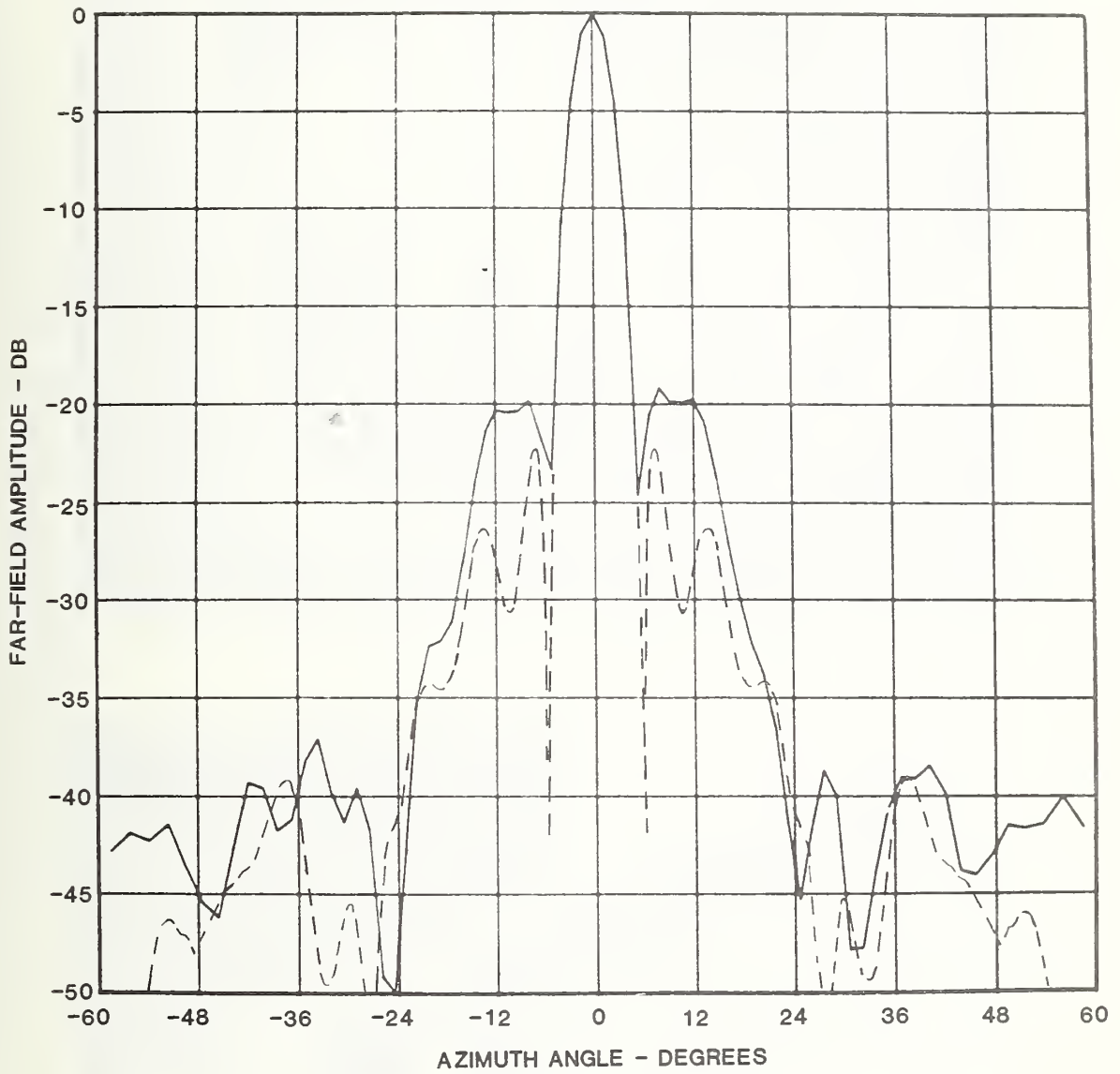


Figure 16b. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. H-plane cut, solid curve - measured pattern, dashed curve - physical optics.

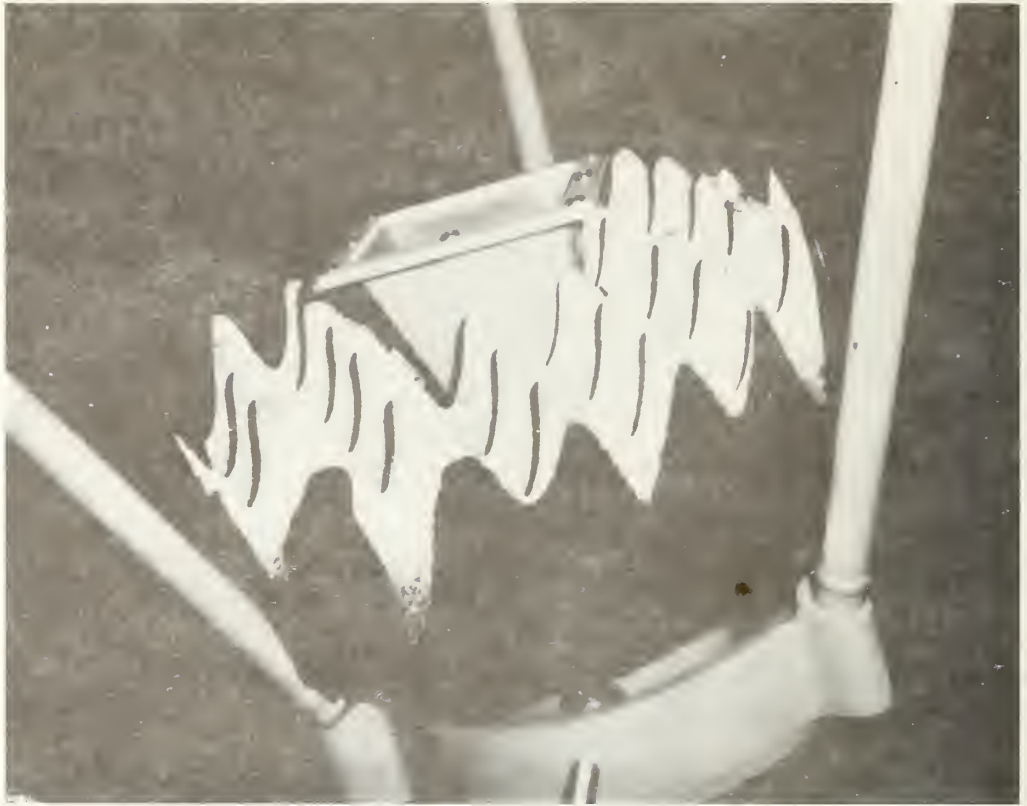


Figure 17a. Feed region of antenna with absorber collar.

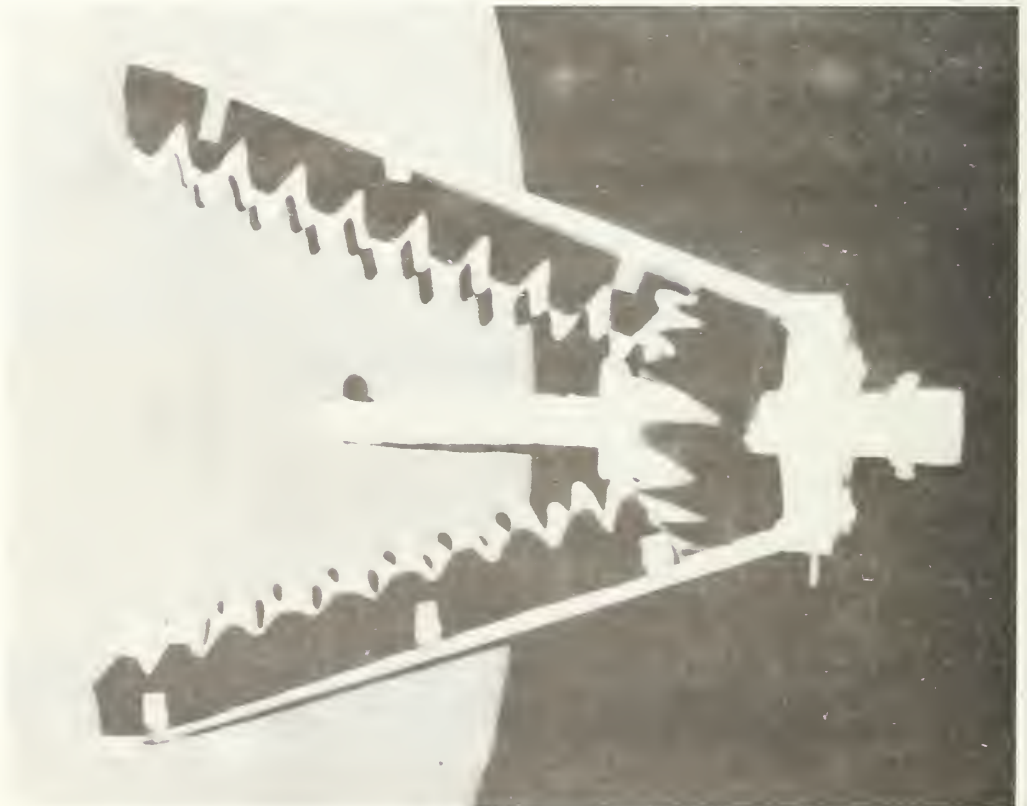


Figure 17b. Feed support struts with absorber attached.

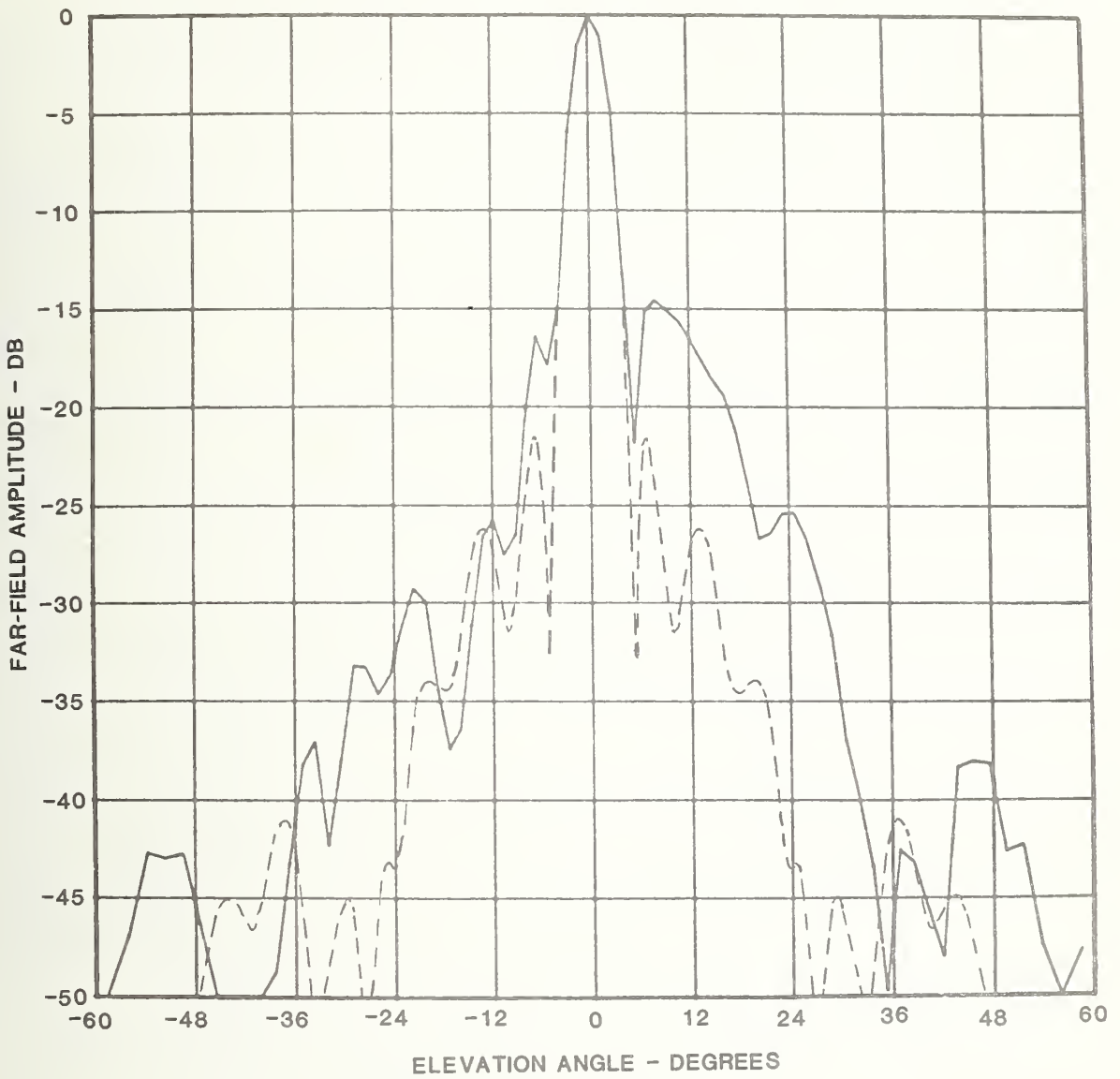


Figure 18a. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. E-plane cut, solid curve - measured pattern, dashed curve - physical optics.

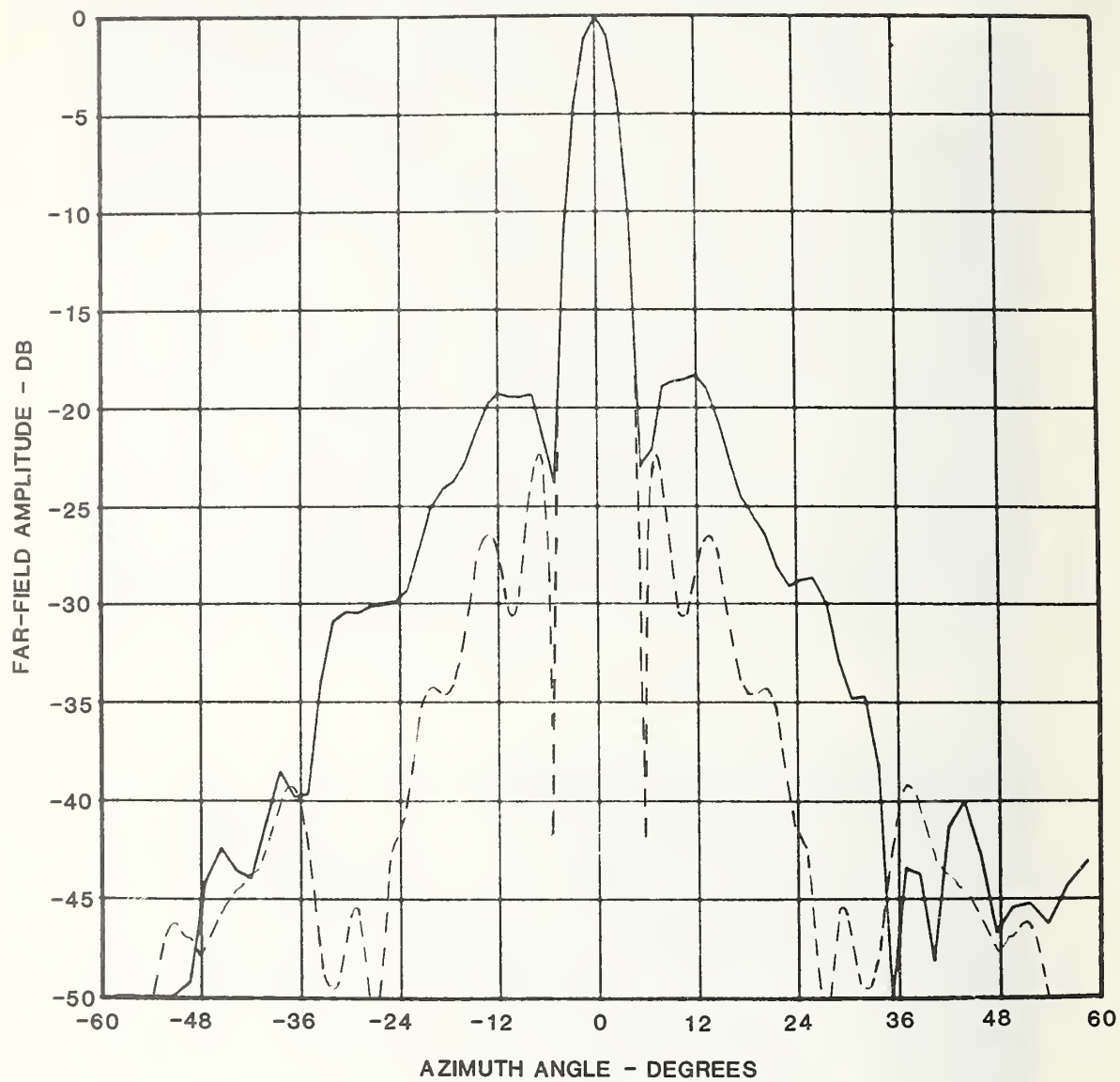


Figure 18a. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. H-plane cut, solid curve - measured pattern, dashed curve - physical optics.



Backlobe radiation from the feed antenna is not considered. It is difficult to estimate the magnitude of this effect. While patterns were taken for the feeds of antennas number 1 and 2, it is in a completely different mounting structure when in place in the antenna; and, as a result, the pattern in the rear hemisphere for the feed will not give any valid data about its back lobes.

The following general conclusions can be stated concerning the usefulness of this particular PO model.

1. The model appears to give better results for larger  $D/\lambda$  ratios.
2. Sidelobe positions are fairly accurately predicted for the first few sidelobes.
3. The magnitudes of the predicted sidelobes can be as much as several dB off for small ( $<50\lambda$ ) antennas.
4. A contributor to the observed differences in the case of antenna 1 (and also 2 because its construction was the same) is multiple reflections between the reflector surface and feed structure.
5. For far sidelobe regions (beyond 4 or 5 lobes) it appears that a better model such as a PO-GTD combination should be employed.
6. A model which takes struts into account would be useful.

Because the theoretical model is used to predict near-zone fields and coupling, it is useful to consider the effect of discrepancies between the modeled and actual fields on the prediction of near-fields and coupling.

For determination of the near-field radiation in front of the antenna, it is expected that the sidelobe discrepancies will have a negligible effect. The major source of error will occur because the true gain is not known and must be estimated. The current PO model will not give results in the region far off boresight or in the back direction.

The coupling results will be affected by the sidelobe region, however. Calculation of the coupling depends on that portion of the far field of each antenna which is subtended by the other; hence, the sidelobe structure is important. Because the locations of the sidelobes are accurately predicted, the basic structure of the coupling as a function of

relative position of the two antennas will be retained. Any errors in the magnitude of the far field predicted by PO will be carried over into the coupling ratio.

## 5. COMPARISON OF PREDICTED AND MEASURED NEAR-FIELD COUPLING

In order to utilize the near-field coupling program (CUPLNF) to predict actual near-field coupling, the two C-Band reflector antennas (numbers 1 and 2 of table 4.1) which were modeled using PO, were set up to measure the near-field coupling directly for various relative orientations and separations. The frequency of operation was 4.0 GHz which gives a diameter of 16.25 wavelengths and a combined or mutual Rayleigh distance  $(D_1+D_2)^2/\lambda$  of about 80 meters.

The antennas were mounted on movable wooden towers at a height of about 7 meters above the ground. The coupling was measured as a function of separation distance for separations ranging from 1 to 8.5 meters and for three relative orientations of the antennas. This procedure also gives a measure of the level of multiple reflections between the antennas (which are neglected in the calculations).

A photograph of the experimental setup is shown in figure 19, and figure 20 illustrates schematically the three relative orientations employed.

For cases two and three, the angle of the receiving antenna was varied over approximately a  $\pm 40^\circ$  range at a fixed separation of 3.5 meters to test the coupling as a function of angle.

Small angles were deliberately chosen for two reasons. Because the PO model used here does not perform well in the sidelobe region, the measurements must be restricted to small angles so the model can successfully predict coupling from boresight. Further, the planar scan data yields far fields which are valid only to about  $45^\circ$  to  $50^\circ$ , and, this too, limits the angles. For wider angle coverage, nonplanar scanning techniques such as cylindrical or spherical would prove useful.

It should be noted that in case 1, the primary source of coupling is the interaction of the main lobes of the two antennas. Case 2 corresponds to the main lobe of the transmitting antenna interacting with the sidelobes of the receiving antenna. In case 3, the sidelobes of each antenna interact with each other.

Calculation of the coupling between the antennas was carried out for five separations in the range 1.5 to 7.5 meters for each case measured. Far fields used as input were from two sources. The experimentally determined far fields obtained from transformation of near-field data were used in one set of calculations, and the far fields obtained from the model using the adapted USC PO subroutines were used in the other calculations.

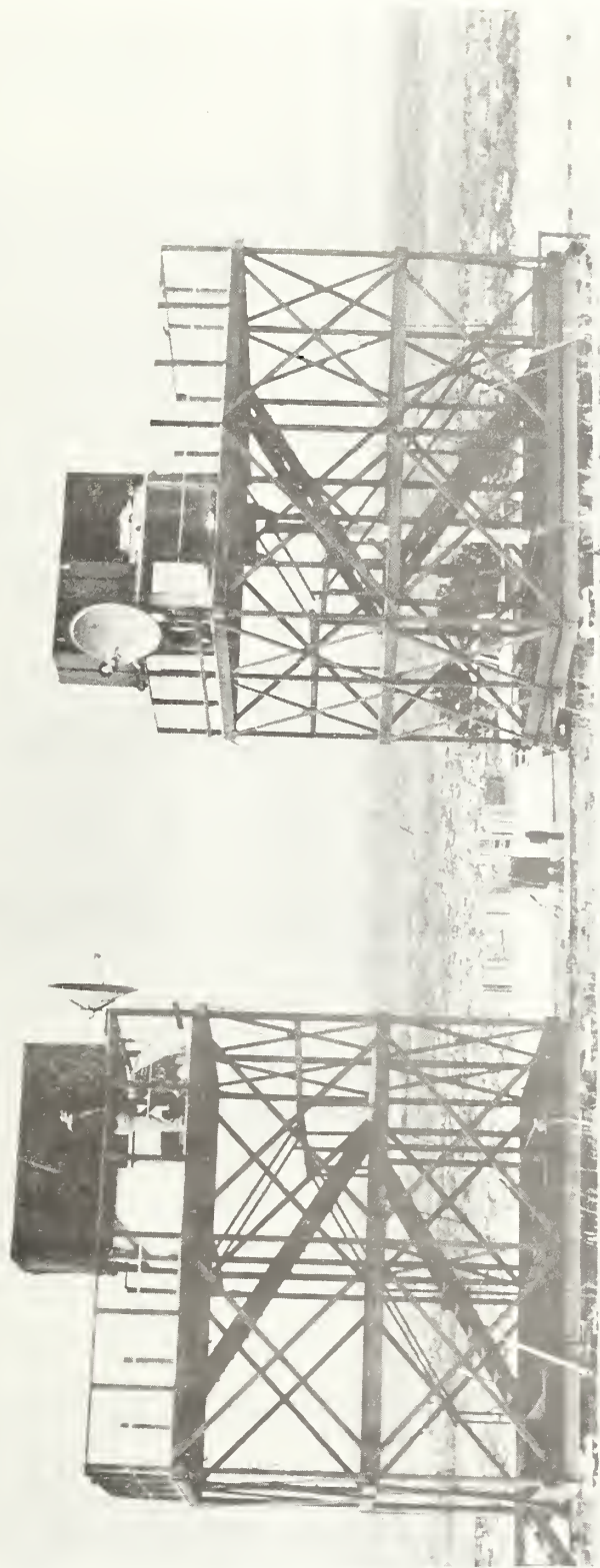
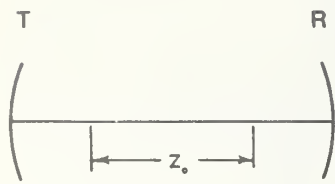
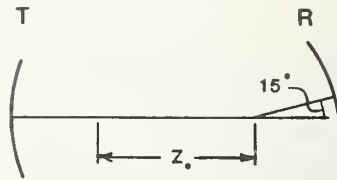


Figure 19. Photograph of experimental set up for measuring coupling between two reflector antennas.



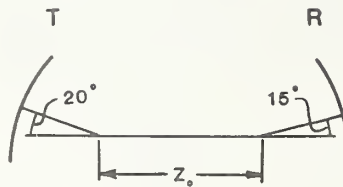
CASE 1

$$\begin{aligned} \phi_t &= 0^\circ, & \phi_r &= 0^\circ \\ \theta_t &= 0^\circ, & \theta_r &= 0^\circ \\ \psi_t &= 0^\circ, & \psi_r &= 0^\circ \end{aligned}$$



CASE 2

$$\begin{aligned} \phi_t &= 0^\circ, & \phi_r &= 180^\circ \\ \theta_t &= 0^\circ, & \theta_r &= 15^\circ \\ \psi_t &= 0^\circ, & \psi_r &= 180^\circ \end{aligned}$$



CASE 3

$$\begin{aligned} \phi_t &= 0^\circ, & \phi_r &= 180^\circ \\ \theta_t &= 20^\circ, & \theta_r &= 15^\circ \\ \psi_t &= 0^\circ, & \psi_r &= 180^\circ \end{aligned}$$

Figure 20. Schematic showing relative orientations of antennas for the three test cases.

The results of the three cases are shown in figures 21 to 23. In each case, the envelope of the measured data is shown, rather than the actual data, which consists of approximately sinusoidal oscillations of period  $\lambda/2$  superimposed on the data which arise because of multiple reflections between the two antennas.

We note fairly good agreement between the measured data and that predicted using measured far fields except in the case of the  $(0^0, 15^0)$  data. This disagreement will be discussed shortly. In the  $(0^0, 0^0)$  case, the prediction using actual far-field data is approximately 0.5 dB low, and follows the shape of the average of the measured data very well. In the  $(20^0, 15^0)$  case, we again observe fairly good agreement between the shape of the predicted and measured curves with an average error of about 2 dB. As in the case of the measurement of low sidelobes, this error is not unacceptable. It might be expected that a greater error would occur when the sidelobes are interacting because of their complicated structure and resultant sensitivity to orientation. While every effort was made in the experimental procedure to ensure accurate positioning, the accuracy was probably no better than  $1/2^0$  about all three axes.

We now discuss the  $(0^0, 15^0)$  case where agreement is not good. Here, we suggest that slight misalignment may be the primary cause. In the rotation performed at a separation of 3.5 meters, a peak of -25.2 dB occurred at about  $12.0^0$ . Calculations show that a peak in the predicted coupling occurs at an angle of  $12.4^0$  with a magnitude of -26.4 dB. Predicted and observed nulls also occur at about  $20^0$  to  $22^0$ , though the magnitude comparison of the null depth is not so good. Because of multiple reflections and multipath and because the cross-polarized component is not included in the calculations, null comparisons cannot be expected to be so good as that observed at relative maxima. It would thus appear that the discrepancy at  $(0^0, 15^0)$  can be explained by a small error in orientation.

## 6. CONCLUSIONS AND RECOMMENDATIONS

Programs and subroutines were written to calculate near fields of reflector antennas and to calculate mutual coupling between antennas whose separation and orientation are arbitrary. The basic data required for these calculations are the two-dimensional complex far-field patterns of the antennas involved.

Documentation for the programs including listings and sample input and output are given in Appendices A and B.

It was seen that the coupling program provides good results if proper far fields are used as input data. When a model such as the physical optics discussed here is employed, the coupling program fails to adequately predict the coupling for off-axis directions.

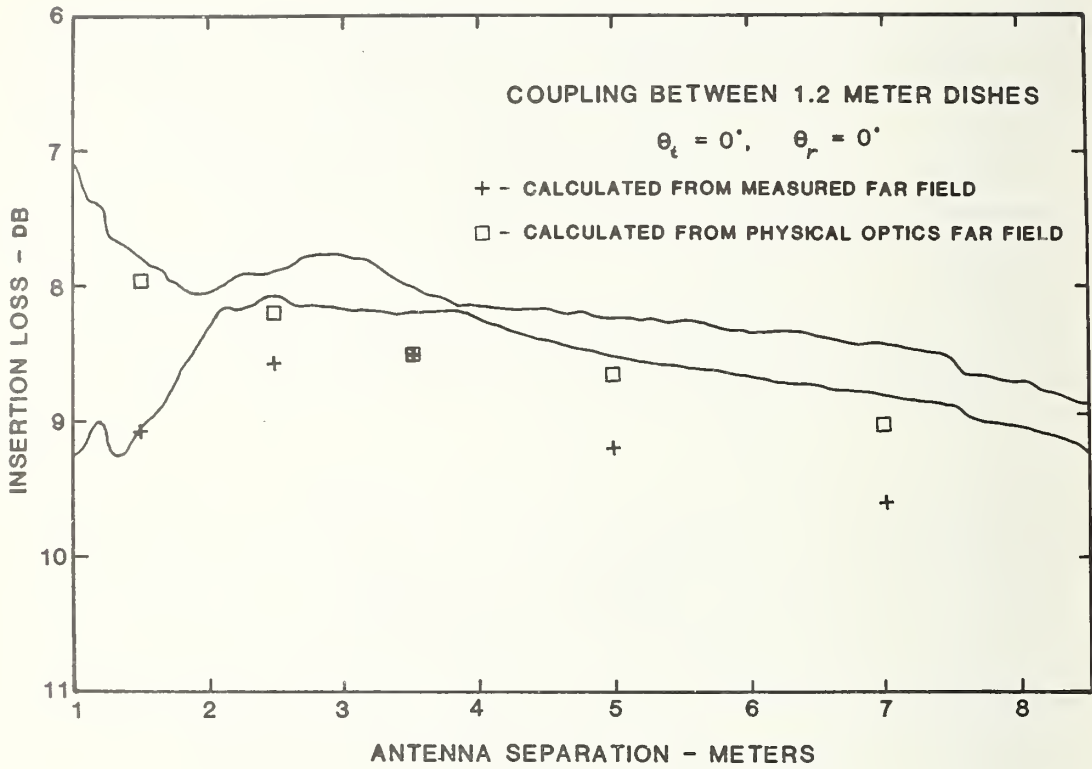


Figure 21. Mutual coupling between 1.2 meter reflector antennas. Case 1:  $\theta_r=0^\circ, \theta_t=0^\circ$ . Solid lines indicate envelope of measured mutual coupling.

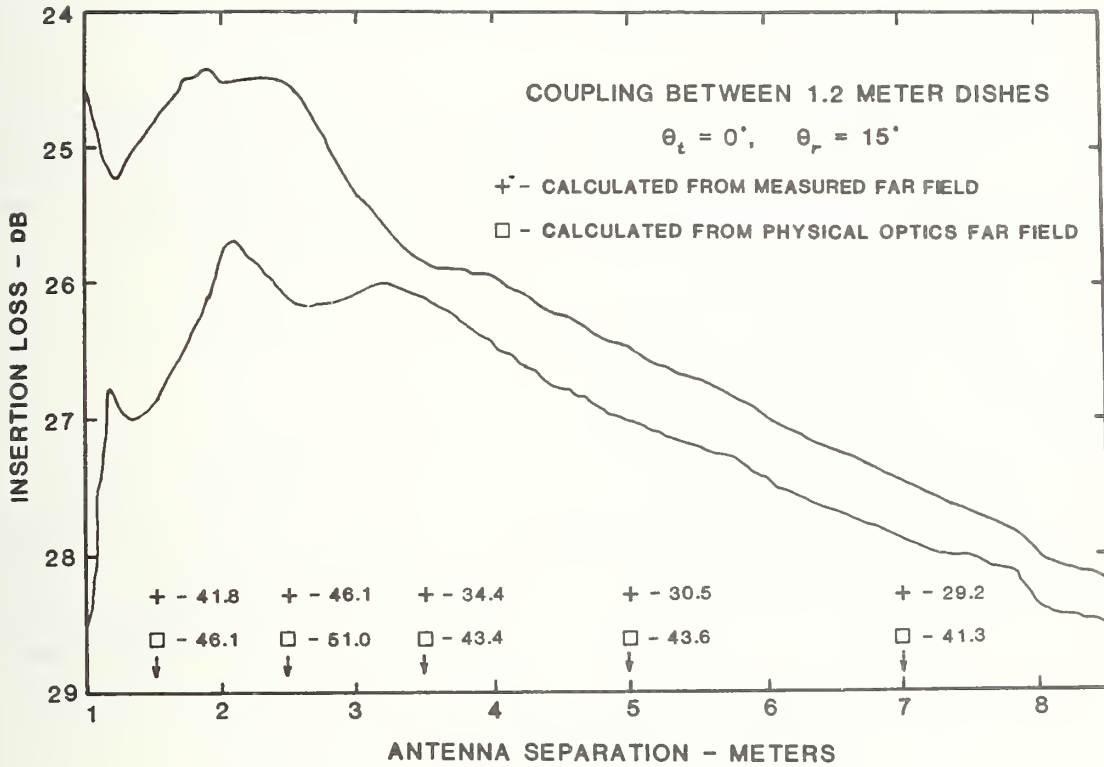


Figure 22. Mutual coupling between 1.2 meter reflector antennas. Case 2:  $\theta_r=15^\circ, \theta_t=0^\circ$ . Solid lines indicate envelope of measured mutual coupling.

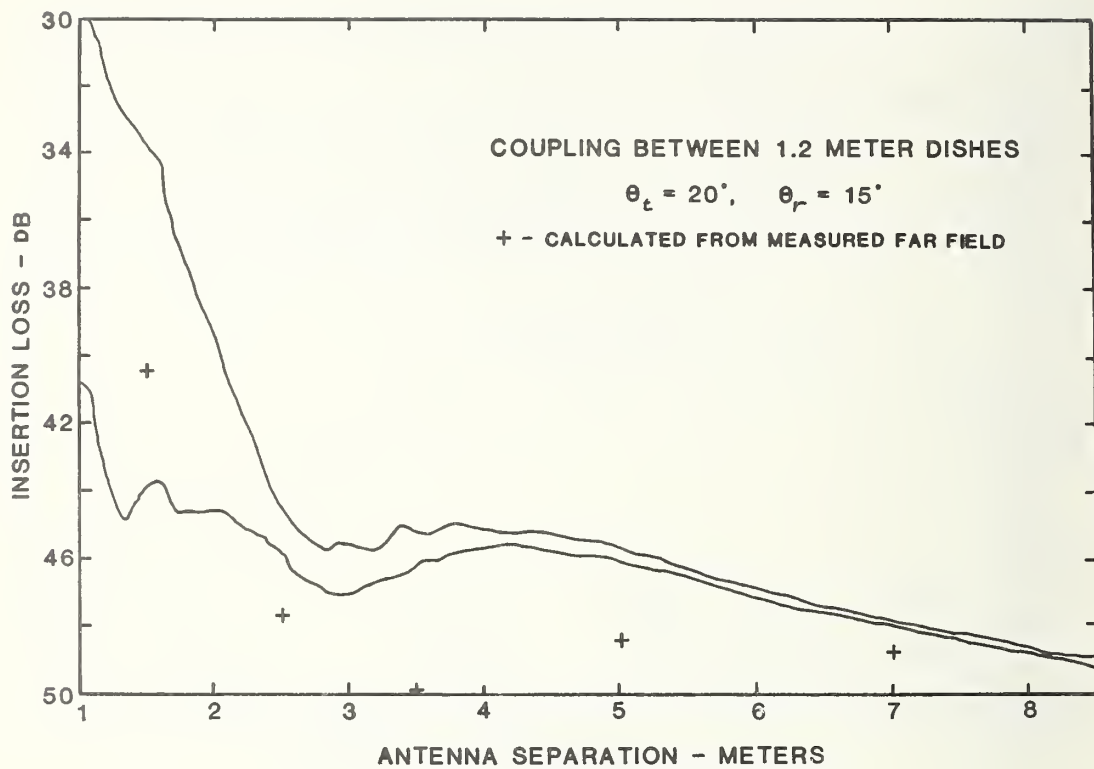


Figure 23. Mutual coupling between 1.2 meter reflector antennas. Case 3:  $\theta_r=15^\circ, \theta_t=20^\circ$ . Solid lines indicate envelope of measured mutual coupling.



Several areas would appear to be worth pursuing. Certainly better models can be obtained. For the types of data required (complete two-dimensional, far-field patterns), a two-dimensional integration PO model is probably not practical. For this type of approach each far-field point would require a two-dimensional rather than a one-dimensional numerical integration. Further, because no symmetry is assumed, all needed far-field points must be computed rather than only the E- and H-plane cuts. Because of these considerations, the computation of the complete pattern by a model which requires two-dimensional integration appears to be impractical. An alternative would be to calculate the main beam and first few sidelobes with PO, and use a GTD analysis for points farther off axis. Such a combination would use the best features of each technique.

A second alternative would be to reformulate the PO model in terms of aperture fields rather than surface currents. This approach would allow efficient computations using the FFT.

Contrasted with the above is the question of whether application might permit the use of less sophisticated models which would give upper-bound values for the desired quantities. Note that regardless of the sophistication of the model employed, certain antennas of a given type may fail to perform as predicted because of unit-to-unit variations. These variations have been observed to be as large as the discrepancies observed between measured and modeled fields for certain types of antennas.

With this in mind, we suggest two alternatives to the use of a sophisticated model. First, a catalog of measured far fields for antenna types in use could be compiled and these data used in coupling or near-field calculation. It would probably be necessary to measure several samples in order to determine expected unit-to-unit variations. An alternate approach would be to employ an envelope type of far-field pattern, such as the amplitude pattern that the CCIR recommended ( $32-25 \log \theta$  function), if a reasonable phase function is also included.

It is recommended that these approaches be studied to determine if, in fact, they can give useful results.

#### ACKNOWLEDGMENT

The physical optics computer program was supplied by Prof. W. V. T. Rusch of the University of Southern California. Near-field measurements were performed by Mr. D. P. Kremer, who also assisted in the mutual coupling measurements. Helpful discussions with Dr. R. C. Baird, Mr. A. C. Newell, and Prof. Rusch are also acknowledged.

## REFERENCES

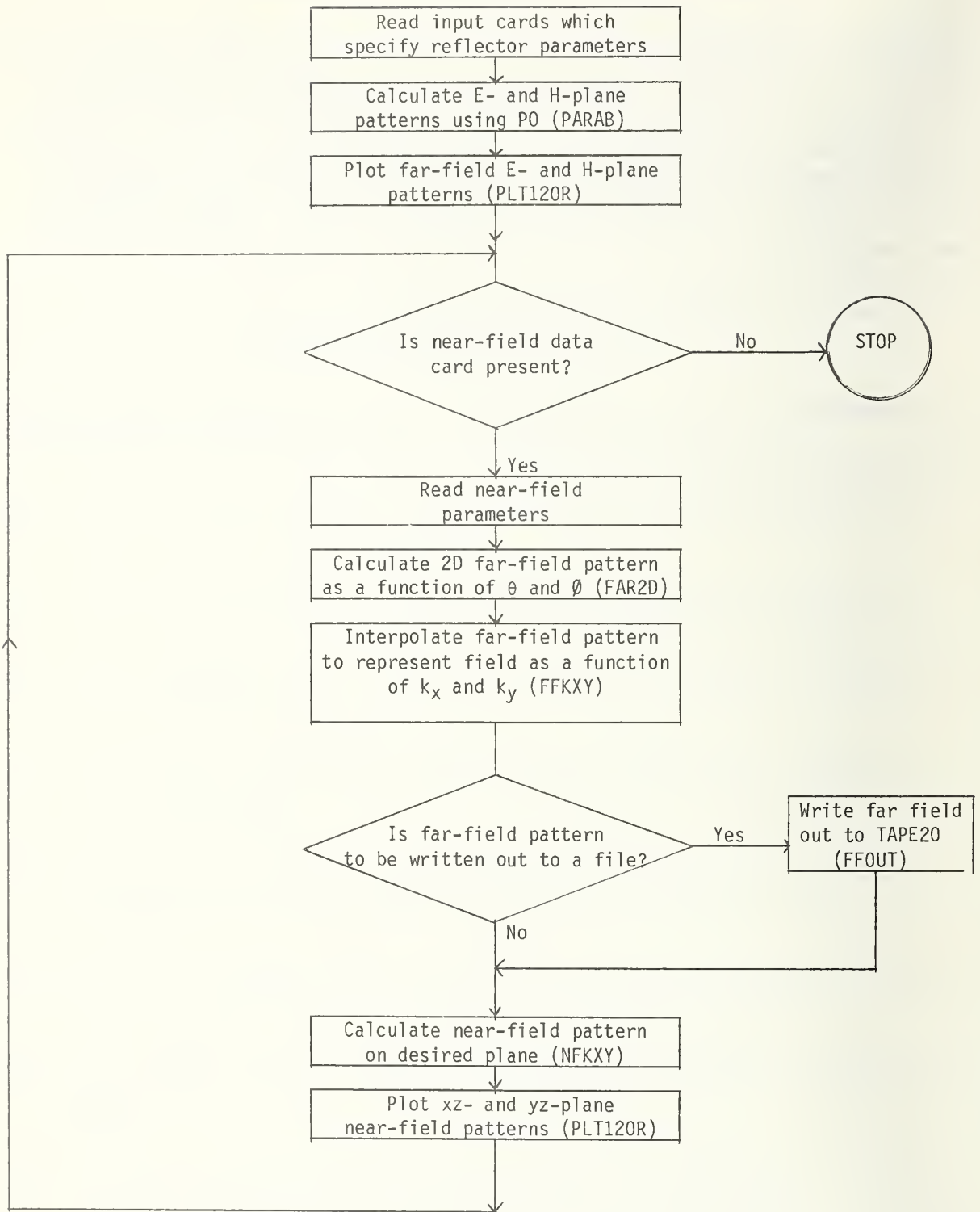
- [1a] Kerns, D. M., and Dayhoff, E. S., Theory of diffraction in microwave interferometry, J. Res. Nat. Bur. Stand., 64B, 1-13 (Jan.-March 1960).
- [1b] Kerns, D. M., Plane-wave scattering-matrix theory of antennas and antenna-antenna interactions: Formulation and applications, J. Res. Nat. Bur. Stand., 80B, 5-51 (Jan.-March 1976).
- [2] Yaghjian, A. D., The reactive and far-field boundaries for arbitrary antennas derived from their quality factor, National Radio Science Meeting (USNC/URSI), University of Colorado at Boulder (Jan. 9-13, 1978).
- [3] Ames, J. S., and Murnaghan, F. D., Theoretical Mechanics, Ch. II (Ginn, Boston, Massachusetts, 1929).
- [4] See, e.g., Cathey, W.T., Optical Information Processing and Holography, Ch. 2 (John Wiley & Sons, Inc., New York, N.Y., 1974).
- [5] See, e.g., the June 1967 issue of the IEEE Trans. on Audio and Electroacoustics.
- [6] See, e.g., Johnson, C. C., Field and Wave Electrodynamics, section 10.5 (McGraw-Hill, New York, N.Y., 1965).
- [7] Hu, Ming-Kuei, Near zone power transmission formulas, IRE Convention Record, 6, Pt. 8, 128-138 (1958).
- [8] Rusch, W. V. T., Reflector antennas, in Numerical and Asymptotic Techniques in Electromagnetics, R. Mittra, Ed. (Springer-Verlag, New York, N.Y., 1975).
- [9] Rusch, W. V. T., Course notes for short course, Reflector Antenna Theory and Design; University of Southern California, Los Angeles, California (July 1976).
- [10] Tai, C-T, Dyadic Green's Functions in Electromagnetic Theory (Intext, New York, N.Y., 1972).

## APPENDIX A. POMODL - PHYSICAL OPTICS ANTENNA MODEL

This appendix includes detailed documentation of the program which models reflector antennas using physical optics and, at the user's option, calculates a two-dimensional far-field pattern for use by CUPLNF and also calculates near-field patterns on a specified plane. Each subroutine is documented individually, except for those which were obtained from other institutions and used unaltered, in which case only a brief description and listing is included. The final section of the appendix includes a sample input deck and a sample program output.

### A.1 GENERAL OVERVIEW OF COMPUTER PROGRAM

The program POMODL and its associated subroutines are described in detail in the following subsections. The flow chart below is presented in order to give the reader an overview of the operation of the program package.



## A.1.1 PROGRAM POMODL

### PURPOSE:

To control input, output and flow of far-field calculation and transformation to near field.

### GENERAL DISCUSSION:

This subroutine is a modified and extended version of SUBROUTINE PDRIVE written by Professor W. V. T. Rusch of the University of Southern California (USC). This subroutine reads data cards which specify the physical parameters of a paraboloidal reflector antenna and the parameters of the desired near-field patterns. It is basically a driver program for the USC PO subroutine PARAB and the subroutines which perform the far- to near-zone transformation.

The program produces plots and tables for far field in the E- and H-planes and near-field cuts on a plane or planes perpendicular to the axis of symmetry of the reflector. In addition, the program calculates the near field on the complete plane and stores it in an array. This data may be obtained by a minor program modification. The far field presented in a table of values at equally spaced increments in  $(k_x, k_y)$  space is also available at the user's option.

Because the techniques used require a substantial amount of computer core, it is recommended that the DIMENSION and COMMON statements specifying the size of arrays EY and DATAX be changed to suit the problem considered. Minimum size for EY is  $2 \times (\text{number of points to be calculated in } \theta\text{-direction}) \times (\text{number of points in } \phi\text{-direction})$ . For DATAX, the size must be at least  $2 \times (\text{number of near-field points in } x\text{-direction}) \times (\text{number of near-field in } y\text{-direction})$ . Because arrays EY and DATAX are not directly used by the main program but are dimensioned only for storage allocation purposes, they may be dimensioned as single dimensioned arrays whose sizes are greater than or equal to the values specified above.

### INPUT CARDS

The input card deck consists of two groups of cards. The first five cards must be included in every run and specify the parameters of the antenna being modeled and the ranges and increments for the far field.

The second group of cards specifies the desired parameters of the near field to be calculated. If no cards of this group are present (i.e. only five input cards), only the E- and H-plane far-field patterns will be calculated and plotted. The near-field

parameters are specified by a single card. Near fields for planes lying at different z-distances can be calculated by including multiple cards.

In addition, it is possible to specify that the far-field array which is calculated at evenly spaced points in  $(k_x, k_y)$ , space may be written out to logical unit 20 for use as input data for the mutual coupling program CUPLNF.

The following is a list and description of the data cards.

### Group I

Card 1	Col. 1-40	This card contains alphanumeric information, usually the name and telephone extension of the person submitting the job.
Card 2	Col. 1-80	An alphanumeric identifier which is used to identify the case being studied. It appears as headings of tables and plots and on identification records of output files.
Card 3		This card specifies antenna parameters. All numbers on this card must have the decimal point explicitly specified.
	Col. 1-10	FOD - the F/D ratio for the reflector.
	Col. 11-20	FOL - the diameter in wavelengths of the reflector.
	Col. 21-30	BLOCK - the feed blockage as a fraction of the reflector diameter.
	Col. 31-40	DFOCUS - amount of axial defocussing in wavelengths, positive defocussing if the feed is beyond the focal point.
	Col. 41-50	ACOSE-E-plane illumination factor.
		If $ACOSE < -100$ . aperture is uniformly illuminated.
		$-100. \leq ACOSE < 0$ . feed is a y-directed electric dipole.
		$ACOSE \geq 0$ . E-plane feed pattern is $\cos^{ACOSE}(\pi-\theta)$

Col. 51-60 ACOSH - H-plane illumination factor.

If  $ACOSH \geq 0$ . H-plane feed pattern is  $\cos^{ACOSH}(\pi-\theta)$ .

Col. 61-70 FREQ - frequency in GHz.

Card 4

This card specifies parameters related to the far-field pattern calculated from P0. Except as noted, decimal points must be explicitly specified.

Col. 1-10 THETHF - initial value of theta - degrees.

Col. 11-20 DTHETA - theta increment - degrees.

Col. 21-30 PHIF - initial value of phi - degrees.

Col. 31-40 DLPH - phi increment - degrees.

Col. 41-45 NTHETA - number of theta points desired, no decimal point, right justified in field.

Card 5

This card gives data which allow calculation of magnitude of near electric field.

Col. 1-10 PIN - power input to antenna, a blank in field gives default value of 1.0 watt.

Col. 11-20 EFF - assumed aperture efficiency of antenna in percent, a blank in field gives default value of 100 percent.

## Group II

Card 6

This card specifies the parameters of the near field which is to be calculated. This card may be repeated to calculate near fields on different planes. If card 6 is omitted, only a far field will be computed and plotted.

Col. 1-10 DELX - near field x-increment in meters.

Col. 11-20 DELY - near-field y-increment in meters.

Col. 21-30	DIST - distance from focal point of antenna reflector to near-field plane in meters.
Col. 31-40	Blank - field not used.
Col. 41-45	IR2TON - number of y points desired in near field, no decimal point specified, right justified in field.
Col. 46-50	IC2TON - number of x points desired in near field, no decimal point specified, right justified in field.

## OUTPUT

A copy of typical output for the program is included in section A.2. A table of input parameters is given first followed by the E- and H-plane far-field patterns for the antenna. Page printer plots for the E- and H-plane are then included.

The near-field parameters are then printed in a table giving the x- and y- near-field centerline cuts. Finally, the amplitude and phase of the near-field centerline cuts are plotted.

## SYMBOL DICTIONARY:

ACOSE	= E-plane aperture illumination factor
ACOSH	= H-plane aperture illumination factor
BLOCK	= Fractional diameter blocking
CASEID	= Alphanumeric identifier
CEE	= Speed of light $\times 10^{-9}$
DATA(I,J)	= Array reserved for far field versus $k_x$ and $k_y$
DELX	= Near-field x-increment
DELY	= Near-field y-increment
DFOCUS	= Amount of axial defocussing beyond focus in wavelengths
DIST	= Distance between near-field plane and focal plane in meters
DLPH	= Far-field phi increment in degrees
DOL	= Reflector diameter in wavelengths
DTHETA	= Far-field increment in degrees
EFF	= Assumed antenna efficiency
EPFAZE	= Phase of EPHI in degrees
EPHDB(I)	= Normalized phi component magnitude expressed in dB
EPHI	= Phi component of far field
EPLANE(I)	= y-component of $s_{10}$
EPMAG	= Intermediate variable - magnitude of EPHI
EPREF	= Magnitude of EPHI(1) used for normalization purposes



ETFAZE = Phase of ETHETA in degrees  
 ETHDB(I) = Normalized theta component magnitude expressed in dB  
 ETHETA = Theta component of far field  
 ETMAG = Intermediate variable - magnitude of ETHETA  
 ETRF = Magnitude of ETHETA(1) used for normalization purposes  
 EY(I,J) = Array reserved for far field versus  $\theta$  and  $\phi$   
 FKAY = Propagation constant =  $2\pi/\text{wavelength}$   
 FOD = Reflector focal length/diameter  
 FREQ = Frequency in GHz.  
 GAIN = Theoretical gain of antenna  
 GDB = Gain of antenna expressed in dB  
 HPLANE(I) = x-component of  $s_{10}$   
 IC2TON = Number of near-field points in x-direction  
 ID = Alphanumeric identifier, usually programmer's name  
 IR2TON = Number of near-field points in y-direction  
 JTH2M1 =  $2 \times J\text{THETA} - 1$  used for array indexing  
 JTHETA = Theta loop index  
 JTHX2 =  $2 \times J\text{THETA}$  used for array indexing  
 NPHI = Number of phi points to be calculated  
 NTHETA = Number of theta points to be calculated PARAB  
 PARAB = Main subroutine to calculate far field of paraboloidal reflector antenna  
 PHIF = Initial value of phi in degrees  
 PI =  $\pi = 3.14159\dots$   
 PNRM = Power normalization factor  
 PIN = Input power to antenna  
 PNRM = Power normalization factor  
 RTD = Radians to degrees conversion factor =  $\pi/180$   
 THETA(I) = Polar angle measured from boresight axis  
 THETAF = Initial value of theta in degrees

COMMON BLOCKS:

The labeled common used in POMODL is described below with a list of routines in which it is used. The variables are defined in the symbol dictionary.

COMMON /CNTRL/ DTHETA, DLPH, DELX, DELY, FREQ, DIST, PNRM

Routines using /CNTRL/: POMODL, FAR2D, FFKXY, NFKXY

```

1      PROGRAM POMODL (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE20) POMODL      1
      C                                                                    POMODL      2
      C DRIVER PROGRAM FOR PHYSICAL OPTICS SUBROUTINE PAPAB, WRITTEN BY POMODL      3
      C PROFESSOR W. V. T. RUSCH OF THE UNIVERSITY OF SOUTHERN POMODL      4
      C CALIFORNIA, WHICH INCLUDES CAPABILITY OF CALCULATING POMODL      5
      C NEAP FIELDS ON A SPECIFIED PLANE. POMODL      6
      C                                                                    POMODL      7
      C 7-AXIS IS AXIS OF SYMMETRY POINTING AWAY FROM PARABOLOID. X IS POMODL      8
      C POLAR ANGLE THETA-PRIME MEASURED FROM THE POSITIVE-Z AXIS. POMODL      9
10     C X = PI IS THE DIRECTION OF THE REFLECTOR VERTEX. POMODL     10
      C                                                                    POMODL     11
      C XP IS THE POLAR ANGLE THETA-DOUBLE-PRIME MEASURED FROM THE POMODL     12
      C POSITIVE Z-AXIS WITH THE DEFOCUSSED FEED AS ORIGIN. POMODL     13
      C                                                                    POMODL     14
15     C THE FIELDS OF THE FEED ARE THE FIELDS OF A CIRCULAR APERATURE POMODL     15
      C EXCITED IN THE M=1 AZIMUTHAL MODE. THE F-PLANE IS THE POMODL     16
      C YZ-PLANE AND THE H-PLANE IS THE XZ-PLANE. THE COMPLEX POLAR POMODL     17
      C PATTERNS A1(TP) AND D1(TP) ARE SUCH THAT MOST OF THE POWER POMODL     18
      C IS RADIATED TOWARD THE REFLECTOR AND VERY LITTLE POWER IS POMODL     19
20     C RADIATED INTO THE HALF-SPACE XP.LT.PI/2. FURTHERMORE, TO POMODL     20
      C ASSUME CONTINUITY OF THE FIELD WHEN XP = PI, IT IS NECESSARY POMODL     21
      C THAT D1(PI) = -A1(PI). POMODL     22
      C                                                                    POMODL     23
      C                                                                    POMODL     24
25     C FFD = REFLECTOR F/D POMODL     25
      C DCL = REFLECTOR DIAMETER IN WAVELENGTHS POMODL     26
      C BLOCK = FRACTIONAL DIAMETER BLOCKING POMODL     27
      C DEFOCUS = AMOUNT OF AXIAL DEFOCUSING BEYOND THE FOCUS IN WAVELENGTH POMODL     28
      C IF(ACDSE.LT.(-100.0)) THE APERTURE IS UNIFORMLY ILLUMINATED POMODL     29
20     C IF(ACDSE.GE.(-100.0).AND.LT.D.D) THE FEED IS Y-DIRECTED ELECTRIC POMODL     30
      C DIRECTION POMODL     31
      C IF(ACDSE.GE.0.0) A1 = (COS(PI-XP))**ACDSE, XP.GE.PI/2 POMODL     32
      C = 0, XP.LT.PI/2 POMODL     33
      C D1 = -(COS(PI-CP))**ACDSE, XP.GE.PI/2 POMODL     34
35     C = 0, XP.LT.PI/2 POMODL     35
      C FREQ = FREQUENCY POMODL     36
      C                                                                    POMODL     37
      C THETA# = INITIAL VALUE OF THETA, DEGREES POMODL     38
      C DTHETA = DIFFERENTIAL VALUE OF THETA POMODL     39
40     C PHIF = INITIAL VALUE OF PHI POMODL     40
      C DLPH = DIFFERENTIAL VALUE OF PHI POMODL     41
      C NTHETA = NUMBER OF THETA VALUES POMODL     42
      C                                                                    POMODL     43
      C PIN = POWER INPUT TO ANTENNA FOR NEAP-ZONE FIELD STRENGTH POMODL     44
45     C EFF = APERTURE EFFICIENCY OF ANTENNA POMODL     45
      C                                                                    POMODL     46
      C                                                                    POMODL     47
      C COMPLEX FTHETA,EPHI POMODL     48
      C                                                                    POMODL     49
50     C DIMENSION FTHDB(200), EPHDB(200), THETA(200) POMODL     50
      C DIMENSION EPLANE(400), HPLANE(400) POMODL     51
      C DIMENSION ID(4) POMODL     52
      C                                                                    POMODL     53
      C COMMON EY(6000), DATAX(8192) POMODL     54
55     C COMMON /ID/ CASEID(P) POMODL     55
      C COMMON /CNTPL/ DTHETA, DLPH, DFLX, DELY, FREQ, DIST, PNRM POMODL     56
      C                                                                    POMODL     57
      C INPUT POMODL     58
      C READ (5, 5000) ID POMODL     59
60     C PRINT 6001, ID POMODL     60
      C READ(5,5000) CASEID POMODL     61
      C                                                                    POMODL     62
      C WRITE(6,6000)CASEID POMODL     62
      C READ(5,5020)FFD,DCL,BLOCK,DEFOCUS,ACDSE,ACDSEH, FREQ POMODL     63
      C READ(5, 5040) THETA#, DTHETA, PHIF, DLPH, NTHETA POMODL     64
65     C READ(5, 5020) PIN, EFF POMODL     65
      C IF (PIN .EQ. 0.) PIN = 1.0 POMODL     66
      C IF (EFF .EQ. 0.) EFF = 1. POMODL     67
      C WRITE(6,6020)FFD,DCL,BLOCK,DEFOCUS, FREQ POMODL     68
      C IF(ACDSE.LT.(-100.0))WRITE(6,6005) POMODL     69
70     C IF(ACDSE.GE.(-100.0).AND.ACDSE.LT.D.D)WRITE(6,6010) POMODL     70
      C IF(ACDSE.GE.0.0)WRITE(6,6015)ACDSE,ACDSEH POMODL     71
      C                                                                    POMODL     72
      C MISCELLANEOUS POMODL     73
      C                                                                    POMODL     74
75     C PI = 4.0*ATAN(1.0) POMODL     75
      C PTD = 180.0/PI POMODL     76
      C CFF = .2997925 POMODL     77

```

```

      FKAY = 2.*PI*FPFC/CFE                                POMODL    78
      NPHI = 90./DLRH + 1.CC001                            POMODL    79
      GAIN = EFF*PI*PI*DDL*DCL                             POMODL    80
      GDR = 10.*ALOG10(GAIN)                               POMODL    81
      PNRM = SQRT(15.*PIN*GAIN/FKAY/FKAY)/PI              POMODL    82
      WRITE(6,6D02) EFF*1CC., GDR, PIN                   POMODL    83
      WRITE(6,6030)                                        POMODL    84
85      C
      C      ENTER THE THETA LCCP                           POMODL    85
      C                                                    POMODL    86
      C                                                    POMODL    87
      DO 100 JTHETA = 1,NTHETA                             POMODL    88
      THETA(JTHETA) = THETAE + (JTHETA-1)*DTHETA          POMODL    89
90      CALL PARAP(FOD,DDL,BLOCK,DFOCUS,ACOSE,ACOSH,THETA(JTHETA),ETHETA,
      1FRHI)                                               POMODL    90
      ETMAG = CABS(ETHETA)                                 POMODL    91
      IF(JTHETA.EQ.1) FTREF = ETMAG                       POMODL    92
      ETHDR(JTHETA) = 20.0*ALOG10(ETMAG/ETREF)           POMODL    93
95      EFAZE = -ATAN2(AIMAG(ETHETA),REAL(ETHETA))*RTD    POMODL    94
      FPMAG = CABS(FRHI)                                   POMODL    95
      IF(JTHETA.EQ.1) FRREF = FPMAG                       POMODL    96
      EPHDR(JTHETA) = 20.0*ALOG10(FPMAG/FRREF)           POMODL    97
      EFAZF = -ATAN2(AIMAG(FRHI),REAL(FRHI))*RTD         POMODL    98
100     PRINT 6040, THETA(JTHETA), ETMAG, ETHDR(JTHETA), EFAZE, FPMAG,
      1EPHDR(JTHETA), EFAZE                               POMODL    99
      JTHX2 = JTHETA*2                                    POMODL   100
      JTH2M1 = JTHX2 - 1                                  POMODL   101
      C                                                    POMODL   102
      C                                                    POMODL   103
105     C      NORMALIZE E TO 1.0 AT THETA = 0. AND CALCULATE S10.
      C                                                    POMODL   104
      C                                                    POMODL   105
      C                                                    POMODL   106
      EPLANE(JTH2M1) = CABS(ETHETA)/ETREF/COS(THETA(JTHETA))/RTD
      EPLANE(JTHX2) = EFAZE/RTD                           POMODL   107
      HPLANE(JTH2M1) = CABS(FRHI)/FRREF/COS(THETA(JTHETA))/RTD
      HPLANE(JTHX2) = EFAZF/RTD                           POMODL   108
110     CONTINUE                                          POMODL   109
      C                                                    POMODL   110
      C      PLOI E AND H-PLANE AMPLITUDES                 POMODL   111
      C                                                    POMODL   112
115     CALL PIT12DP(THETA, ETHDR, FC., -60., 0., -50., NTHETA, 1H*, 1, 1)
      PRINT 6080, CASEID, 1CH E-PLANE                     POMODL   113
      CALL PLT12DP(THETA, EPHDR, 60., -60., 0., -50., NTHETA, 1H*, 1, 1)
      PRINT 6080, CASEID, 1CH H-PLANE                     POMODL   114
      C                                                    POMODL   115
      C                                                    POMODL   116
120     C      READ DATA FOR NEAR FIELD                   POMODL   117
      C                                                    POMODL   118
      C                                                    POMODL   119
      C      DELX = NEAR FIELD X-SPACING                   POMODL   120
      C      DELY = NEAR FIELD Y-SPACING                   POMODL   121
125     C      DIST = Z-POSITION OF NEAR FIELD PLANE. (DIST = 0. IS FOCAL
      C      PLANE OF PARABOLA)                             POMODL   122
      C      DUMMY - NOT CURRENTLY USED                     POMODL   123
      C      IP2TON = NUMBER OF POINTS IN Y-ARRAY         POMODL   124
      C      IC2TON = NUMBER OF POINTS IN X-ARRAY         POMODL   125
130     C
      1 READ 5040, DELX, DELY, DIST, DUMMY, IP2TON, IC2TON
      IF (FOD(5)) 200,2
      2 PRINT 6070, DELX*100., IC2TON, DELY*100., IP2TON, DIST*100.
      CALL PAR2D (EPLANE, HPLANE, EY, NTHETA, NPHI, DATA, IP2TON*2,
      1IC2TON)
      GO TO 1
200  WRITE(6,6D60)
5000  FORMAT(A10)

```

155	*YOND FOCUS*,/, T16,*FREQUENCY = *,F6.4,* GHZ.*,/)	POMDDL	155
6030	FORMAT(1H0,T30,*F-PLANE*,T62,*H-PLANE*,/,	POMDDL	156
	*T11,*THETA*,T22,*MAG*,T32,*MAG*,T41,*PHASE*,	POMDDL	157
	*T54,*MAG*,T64,*MAG*,T73,*PHASE*,/,	POMDDL	158
160	* T11,* (DFG)*, T20,* (VOLTS)*, T31,* (DB)*, T41,* (DFG)*,	POMDDL	159
	*T52,* (VOLTS)*, T63,* (DB)*, T73,* (DFG)*, /)	POMDDL	160
6040	FORMAT(1H ,T9,F7.2,T20,F6.3,T29,F7.2,T39,F7.2,T52,F6.3,T61,F7.2,	POMDDL	161
	*T71,F7.2)	POMDDL	162
	6060 FORMAT(///,* EDF FOUND ON LU5*)	POMDDL	163
	6070 FORMAT(///, T7, *NEAR-FIELD PARAMETERS-*,//,T16, *Y-SPACING =*,	POMDDL	164
165	1F6.2, * CM*, T50, I6,* POINTS*,/,T16, *Y-SPACING =*, F6.2, * CM*,	POMDDL	165
	2T50,I6, * POINTS*,/, T16, *DISTANCE FROM REFLECTOR FOCAL POINT =*,	POMDDL	166
	3F6.2, * CM AWAY FROM REFLECTOR SURFACE*,//)	POMDDL	167
	6080 FORMAT(/, 5X, B A10, 5X, A10)	POMDDL	168
	END	POMDDL	169

### A.1.2 SUBROUTINE FAR2D(EPL,HPL,EY,NTHETA,NPHI,DATA,IR2X2,IC2TON)

#### PURPOSE:

To produce a two dimensional array of far-field data given the E-and H-plane cuts for an axially symmetric antenna.

#### ARGUMENTS:

EPL is a complex vector containing the far-field, E-plane pattern of the antenna stored in amplitude phase form.

HPL is a complex vector containing the far-field, H-plane pattern of the antenna stored in amplitude-phase form.

EY is a complex array which contains a two-dimensional, far-field array, arranged as a function of theta and phi.

NTHETA is the number of points in theta direction.

NPHI is the number of points in phi direction.

DATA is not used in this subroutine, see FFKXY.

IR2X2 is twice the number of rows of data produced as a function of  $k_x$  and  $k_y$ .

IC2TON is the number of columns of data in far-field array produced as a function of  $k_x$  and  $k_y$ .

#### METHODS:

Circular symmetry is assumed in the antenna, hence, it is necessary to calculate the far-field pattern over one quadrant only. The subroutine calculates the main rectangular component of the far field which is given by

$$E_y(\theta, \emptyset) = E_{yE}(\theta) \cos\theta \sin^2\emptyset + E_{yH}(\theta) \cos^2\emptyset$$

where

$E_{yE}(\theta)$  = Electric field in E-plane as a function of theta.

$E_{yH}(\theta)$  = Electric field in H-plane as a function of theta.

Under the assumption of circular symmetry, this subroutine calculates the far-field y-component as a function of  $\theta$  and  $\emptyset$  given the E- and H-plane patterns ( $\emptyset=0, \pi/2$ ) as a function of  $\theta$ .

This subroutine uses library functions: ATAN, COS, and SIN.

SYMBOL DICTIONARY:

COSTH	= $\cos(\text{THETA})$
DATAX(I,J)	= Array of angular spectrum data as a function of $k_x$ and $k_y$
DTR	= Degree to radian conversion factor = $\pi/180$
EPL(I)	= y-component of $\underline{s}_{10}$
EY(I,J)	= Array of angular spectrum data as a function of $\theta$ and $\phi$ .
HPL(I)	= x-component of $\underline{s}_{10}$
I	= DO loop index
IC2TON	= Number of columns of data in DATAX
ICOL	= Column loop index
IR2TON	= Number of rows of data in DATAX array
IR2X2	= $2 \times \text{IR2TON}$
IROW	= Row loop index
NPHI	= Number of data output points in phi direction
NPHIM1	= $\text{NPHI} - 1$
NTHETA	= Number of data output points in theta direction
NTHM1	= $\text{NTHETA} - 1$
NTHX2	= $2 \times \text{NTHETA}$
PHI	= Azimuth angle - far-field pattern coordinate
PI	= $\pi = 3.14159\dots$
RTD	= Radian to degree conversion factor = $180/\pi$
SINPH	= $\sin(\text{PHI})$
TEMI	= Intermediate variable
TEMR	= Intermediate variable
THETA	= Polar angle from boresight - far-field pattern coordinate

1	SUBROUTINE FAR2D(EPL, HPL, EY, NTHETA, NPHI, DATA, IR2X2, IC2TON)	FAR2D	1
	C-	FAR2D	2
	C- THIS SUBROUTINE TAKES E-PLANE AND H-PLANE DATA GENERATED AS A	FAR2D	3
	C- FUNCTION OF ANGLE FROM BORESIGHT, AND GENERATES A TWO-DIMENSIONAL	FAR2D	4
5	C- ARRAY OF DATA AS A FUNCTION OF THETA AND PHI WHERE THETA AND PHI	FAR2D	5
	C- ARE THE USUAL SPHERICAL ANGLES DEFINED IN A COORDINATE SYSTEM	FAR2D	6
	C- WHOSE POLAR AXIS COINCIDES WITH BORESIGHT.	FAR2D	7
	C-	FAR2D	8
	C- *NOTE* THIS SUBROUTINE ONLY PRODUCES VALID RESULTS FOR ANTENNAS	FAR2D	9
10	C- WHICH HAVE SEPARABLE FAR-FIELD PATTERNS.	FAR2D	10
	C-	FAR2D	11
	DIMENSION EPL(1), HPL(1), EY(NTHETA,NPHI), DATA(IR2X2, IC2TON)	FAR2D	12
	COMMON /CNTRL/ DLTH, DLPH, DELX, DELY, FRFQ, DIST, PNRM	FAR2D	13
	COMPLEX FY	FAR2D	14
15	PI = 4.*ATAN(1.) \$ RTD = 180./PI \$ DTR = PI/ 180.	FAR2D	15
	NTHX2 = 2*NTHETA	FAR2D	16
	IR2TON = IR2X2/2	FAR2D	17
	DO 10 I = 1, NTHETA	FAR2D	18
	TEMP = EPL(2*I - 1)*COS(EPL(2*I))*PNRM	FAR2D	19
20	TEMI = EPL(2*I - 1)*SIN(EPL(2*I))*PNRM	FAR2D	20
	EY(I, 1) = CMPLX(TEMP, TEMI)	FAR2D	21
	TEMP = HPL(2*I - 1)*COS(HPL(2*I))*PNRM	FAR2D	22
	TEMI = HPL(2*I - 1)*SIN(HPL(2*I))*PNRM	FAR2D	23
	EY(I, NPHI) = CMPLX(TEMP, TEMI)	FAR2D	24
25	10 CONTINUE	FAR2D	25
	NTHM1 = NTHETA - 1 \$ NPHM1 = NPHI - 1	FAR2D	26
	DO 20 IROW = 1, NTHM1	FAR2D	27
	THETA = (IROW - 1)*DLTH*DTR	FAR2D	28
	COSTH = COS(THETA)	FAR2D	29
30	DO 30 ICOL = 2, NPHM1	FAR2D	30
	PHI = (ICOL - 1)*DLPH*DTR	FAR2D	31
	SINPH = SIN(PHI)	FAR2D	32
	EY(IROW, ICOL) = (EY(IROW, 1))*COSTH - EY(IROW, NPHI))*SINPH*	FAR2D	33
35	1 SINPH + EY(IROW, NPHI)	FAR2D	34
	30 CONTINUE	FAR2D	35
	20 CONTINUE	FAR2D	36
	CALL FFKXY (EY, NTHX2, NPHI, DATA, IR2TON*2, IC2TON)	FAR2D	37
	RETURN	FAR2D	38
	END	FAR2D	39

### A.1.3 SUBROUTINE FFKXY(DATAY,NTHX2,NPHI,DATA,IR2X2,IC2TON)

#### PURPOSE:

To produce an array of two-dimensional, far-field data which is equally spaced in the coordinates  $k_x$  and  $k_y$ , given an array which is equally spaced in the coordinates  $\theta$  and  $\emptyset$ .

#### ARGUMENTS:

DATAY is a two-dimensional array of far-field values, expressed as a function of equally spaced  $\theta$  and  $\emptyset$  coordinates in the quadrant  $0 \leq \emptyset \leq \pi/2$ . Complex far-field values are expressed with real and imaginary parts adjacent in storage, such as FORTRAN IV stores them. Note, after execution, DATAY is expressed in polar form because of a call to ARAYRTP.

NTHX2 is twice the number of points in  $\theta$  direction.

NPHI is the number of  $\emptyset$  points in one quadrant.

DATA is the output array of far-field points which are equally spaced in  $k_x$  and  $k_y$ . Complex far-field values are expressed in polar form with amplitudes and phases stored in adjacent locations. This array contains far-field values of an entire hemisphere rather than a single quadrant as is the case for DATAY.

IR2X2 is twice the number of rows ( $k_y$  values) in the DATA array.

IC2TON is the number of columns ( $k_x$  values) in the DATA array.

#### METHODS:

FFKXY is basically an interpolation routine which fills each point in the DATA array, by calculating the corresponding values of  $\theta$  and  $\emptyset$  locating the four nearest points corresponding to these values in the DATAY array. The value stored in DATA is then a weighted average of these four points. The program assumes that the far-field input array is from a single quadrant such as produced by FAR2D, and produces a far-field output array over the entire hemisphere by reflecting about the lines  $k_x = 0$  and  $k_y = 0$ .

Because the FFT is used to calculate the near-field distribution, it is necessary to have a far field which is sampled on equally spaced points in  $k_x$  and  $k_y$ . Further, we chose the spacing so that the near-field spacing will satisfy the sampling theorem criteria. Thus, the far-field increments  $k_x$  and  $k_y$  are fundamentally related to



the near-field spacing which is specified and transmitted into the subroutine via common CNTRL. Relationship between  $k_x$ ,  $k_y$ , the far-field increment, and  $\delta_x$ ,  $\delta_y$ , the near-field spacings, are,

$$\Delta k_x = \frac{2\pi}{\delta_x N_x}, \quad \Delta k_y = \frac{2\pi}{\delta_y N_y}$$

Beginning at the center ( $k_x=k_y=0$ ) of the DATAX array, the value of  $\theta$  and  $\emptyset$  corresponding to  $k_x$  and  $k_y$  are calculated. These are given by

$$\begin{aligned} \theta &= \cos^{-1} \sqrt{1 - k_x^2/k^2 - k_y^2/k^2} \\ &= \tan^{-1}(k_y/k_x) \end{aligned}$$

The indices corresponding to the four elements in the DATAY array that lie closest to the value of  $\theta$  and  $\emptyset$  are computed. A linear two-dimensional interpolation is then performed using these four points in order to compute the value desired. The interpolation is performed on the amplitude and phase, not on the real and imaginary parts of the DATAY array.

Care must be exercised in interpolating the phase, because the phase is only given modulo  $360^\circ$ . This causes errors in performing the interpolation when the phase function makes a jump between two points in question unless a correction is applied to one of the phases. In this subroutine, three of the four phases are reset to lie on the same cycle as the reference phase by testing to see that the absolute value of the phase difference between the point in question and the reference is less than  $180^\circ$ . This procedure is valid provided that the far-field data points are spaced closely enough. A reasonable requirement would be to have at least 4 or 5 far-field points in an angular range of a sidelobe, a requirement which is met anyway if a sufficiently smooth pattern is produced.

The interpolation is performed by taking a weighted average of the amplitude or adjusted phases of the four surrounding points, the weighting of an individual point being inversely proportional to its distance from the point in question.

#### SYMBOL DICTIONARY:

C(I)	= Coefficients used to calculate $k_x$ and $k_y$ from near-field spacing
CEE	= Speed of light $\times 10^{-9}$
D33J1	= Intermediate variable used in phase test
D43J	= Intermediate variable used in phase test
D43J1	= Intermediate variable used in phase test

DATA(I,J) = Far-field data array as a function of  $k_x$  and  $k_y$   
 DATAY(I,J) = Far-field data array as a function of  $\theta$  and  $\emptyset$   
 DFI = Fractional part of FI  
 DFJ = Fractional part of FJ  
 DLPHI =  $\emptyset$  increment in radians  
 DLTHTA =  $\theta$  increment in radians  
 DTEMP1 = Intermediate variable  
 DTEMP2 = Intermediate variable  
 DTEMP3 = Intermediate variable  
 DTR = Degree to radian conversion factor =  $\pi/180.$   
 FI = Reference theta position for interpolation  
 FJ = Reference phi position for interpolation  
 FKAY =  $k$  = Propagation constant  
 FKAYSQ =  $k^2$   
 FKX =  $k_x$  = x-component of propagation vector  
 FKXSQ =  $k_x^2$   
 FKY =  $k_y$  = y-component of propagation vector  
 FKYSQ =  $k_y^2$   
 FLMDA = Wavelength  
 I = Integer part of FI  
 I1 = Interpolation point index  
 I2 = Interpolation point index  
 I3 = Interpolation point index  
 I4 = Interpolation point index  
 IC = Column interpolation loop index  
 IC2D2 = IC2TON/2 = Center column of far-field array DATA  
 IC2TON = Number of points in  $k_x$  direction in DATA array  
 ICN = Row counter for filling remaining three quadrants of DATA  
 IR = Row interpolation loop index  
 IR2 = Intermediate index  
 IR2D2 = IR2TON/2  
 IR2TON = Number of rows in DATA array  
 IR2X2 =  $2 \times IR2TON$   
 IRN = Row counter for filling remaining three quadrants of DATA  
 IRX = Index for center row of far-field array  
 J = Integer part of FJ  
 NPHI = Number of points in  $\emptyset$  direction in DATAY array  
 NTHX2 =  $2 \times$  Number of points in  $\theta$  direction in DATAY  
 PHI =  $\emptyset$  = Azimuth angle in far-field  
 PHI0 = Initial value of  $\emptyset$   
 PI =  $\pi = 3.14159....$   
 PIX2 =  $2\pi$   
 THETA0 = Initial value of  $\theta$   
 THMAX = Maximum value of  $\theta$  in radians  
 TST = Test variable to determine if z-component of propagation vector is real

```

1      SUBROUTINE FFKXY(DATAY, NTHX2, NPHI, DATA, IR2X2, IC2TON)      FFKXY      1
C-                                     FFKXY      2
C- THIS SUBROUTINE INTERPOLATES AN ARRAY OF FAR-FIELD DATA WHICH  FFKXY      3
C- IS EQUALLY SPACED IN THETA AND IN PHI TO PRODUCE AN ARRAY WHICH FFKXY      4
5 C- IS EQUALLY SPACED IN KX AND KY. FFKXY      5
C-                                     FFKXY      6
      COMMON /CNTRL/ DLTH, DLPH, DFLY, DFLY, FREQ, DIST, PNRM      FFKXY      7
      DIMENSION DATAY(NTHX2, NPHI), C(2), DATA(IR2X2, IC2TON)      FFKXY      8
C-                                     FFKXY      9
10      LHOOT = 20 FFKXY      10
      IP2TON = IP2X2/2 FFKXY      11
      PI = 4. * ATAN(1.) $ PIX2 = 2. * PI FFKXY      12
      CFF = .2997925 $ FLMDA = CFF/FREQ FFKXY      13
      DTP = PI/180. FFKXY      14
15      FKAY = PIX2/FLMDA $ FKAYSO = FKAY*FKAY FFKXY      15
      THETO = 0. $ PHIO = 0. FFKXY      16
      DLHTA = DLTH*DTP $ DLPHI = DLPH*DTP FFKXY      17
      THMAX = (NTHX2/2 - 2)*DLHTA FFKXY      18
20      C(1) = PIX2/(DFLY*IC2TON) FFKXY      19
      C(2) = PIX2/(DFLY*IP2TON) FFKXY      20
      IC2D2 = IC2TON/2 $ IP2D2 = IR2TON/2 FFKXY      21
C-                                     FFKXY      22
C- CHANGE DATAY ARRAY FROM RECTANGULAR TO POLAR FORM. FFKXY      23
C-                                     FFKXY      24
25      CALL APAYRTP(DATAY, NTHX2, NPHI) FFKXY      25
      ICN = 0 FFKXY      26
      DO 61 IC = IC2D2, IC2TON FFKXY      27
      FKX = C(1)*(IC - IC2D2) $ FKXSO = FKX*FKX FFKXY      28
      IPN = 0 FFKXY      29
30      DO 62 IR = IP2D2, IR2TON FFKXY      30
      IP2 = IP*2 - 1 FFKXY      31
      FKY = C(2)*(IP - IP2D2) $ FKYSO = FKY*FKY FFKXY      32
      TST = FKAYSO - FKXSO - FKYSO FFKXY      33
35      IF (TST .LT. 0.) GO TO 60 FFKXY      34
      THETA = ACOS((SQRT(FKAYSO - FKXSO - FKYSO))/FKAY) FFKXY      35
      IF (THETA .GT. THMAX) GO TO 60 FFKXY      36
      IF (FKY .LT. 0.) THETA = -THETA FFKXY      37
      IF (FKY .EQ. 0. .AND. FKY .EQ. 0.) GO TO 63 FFKXY      38
      PHI = ATAN2(FKY, FKX) FFKXY      39
40      GO TO 64 FFKXY      40
      63 PHI = 0. FFKXY      41
      64 IF (PHI .LT. 0.) PHI = PHI + PI FFKXY      42
C-                                     FFKXY      43
C- INTERPOLATE DATAY ARRAY TO PRODUCE DATAY ARRAY WHICH IS EQUALLY FFKXY      44
45 C- SPACED IN KX AND KY. FFKXY      45
C-                                     FFKXY      46
C FIND THE INDICES FOR THE INPUT DATA WHICH IDENTIFY THE COORDINATES FFKXY      47
C CLOSEST TO THE DESIRED THETA AND PHI VALUES. INTERPOLATE TO FIND THE FFKXY      48
C ORDER PATTERN AT THE DESIRED POINT. FFKXY      49
50      C FFKXY      50
      FI=((THETA-THETO)/DLHTA)+1.0 FFKXY      51
      FJ=((PHI-PHIO)/DLPHI)+.99999999 FFKXY      52
      IF (PHI .EQ. 0.) FJ=1. FFKXY      53
      I=FI FFKXY      54
      J=FJ FFKXY      55
      DF1=FI-I FFKXY      56
      DFJ=FJ-J FFKXY      57
      I1=2*I-1 FFKXY      58
      I2=I1+2 FFKXY      59
      I3=2*I FFKXY      60
      I4=I3+2 FFKXY      61
C-                                     FFKXY      62
      IPX = IR2D2*2 - 1 FFKXY      62
C-                                     FFKXY      63
C DETERMINE AMP AT (THETA,PHI) BY WEIGHTED AVERAGE OF ALL 4 POINTS FFKXY      64
65 C AROUND THETA, PHI. FFKXY      65
C-                                     FFKXY      66
      DATA(IP2,IC)=(DFI*DATAY(I2,J)+(1.0-DFI)*DATAY(I1,J))*(1.0-DFJ) FFKXY      67
      1+(DFJ*DATAY(I2,J+1)+(1.0-DFJ)*DATAY(I1,J+1))*DFJ FFKXY      68
C-                                     FFKXY      69
70 C RESET PHASES AT THREE CORNERS TO BE ON SAME CYCLE AS REFERENCE AT FFKXY      70
C (I3,J) FFKXY      71
C-                                     FFKXY      72
C-                                     FFKXY      73
C RESET PHASE AT (I4,J) IF NECESSARY SO THAT THE ABSOLUTE VALUE OF D43J FFKXY      74
75 C IS LESS THAN 180.0 DEGREES. FFKXY      75
C-                                     FFKXY      76
C-                                     FFKXY      77
      D43J=DATAY(I4,J)-DATAY(I3,J) FFKXY      77

```

```

      IF (D43J.GT.180.0)
      1DTEMP1=DATAY(I4,J)-360.0
90      IF (D43J.LE.180.0.AND.D43J.GF.-180.0)
      1DTEMP1=DATAY(I4,J)
      IF (D43J.LT.-180.0)
      1DTEMP1=DATAY(I4,J)+360.0
C
C RESET PHASE AT (I3,J+1) IF NECESSARY SO THAT THE ABSOLUTE VALUE OF
C D33J1 IS LESS THAN 180.0
C
      D33J1=DATAY(I3,J+1)-DATAY(I3,J)
      IF (D33J1.GT.180.0)
90      1DTEMP2=DATAY(I3,J+1)-360.0
      IF (D33J1.LE.180.0.AND.D33J1.GF.-180.0)
      1DTEMP2=DATAY(I3,J+1)
      IF (D33J1.LT.-180.0)
      1DTEMP2=DATAY(I3,J+1)+360.0
95      C
C RESET PHASE AT (I4,J+1) IF NECESSARY SO THAT THE ABSOLUTE VALUE OF
C D43J1 IS LESS THAN 180.0 DEGREE.
C
      D43J1=DATAY(I4,J+1)-DATAY(I3,J)
      IF (D43J1.GT.180.0)
100      1DTEMP3=DATAY(I4,J+1)-360.0
      IF (D43J1.LE.180.0.AND.D43J1.GF.-180.0)
      1DTEMP3=DATAY(I4,J+1)
      IF (D43J1.LT.-180.0)
105      1DTEMP3=DATAY(I4,J+1)+360.0
C
C DETERMINE PHASE AT (THETA,PHI) BY WEIGHTED AVERAGE OF ALL 4 POINTS
C AROUND THETA,PHI.
C
      DATAX(IP2+1,IC)=(DFI*DTEMP1+(1.0-DFI)*DATAY(I3,J))*(1.0-DFJ)
      1+(DFI*DTEMP3+(1.0-DFI)*DTEMP2)*DFJ
      IF (FKY.LT.0.0) DATAX(IP2+1,IC)=DATAX(IR2+1,IC)-180.0
      DATAX(IP2+1,IC)=AMOD(DATAX(IR2+1,IC),360.0)
      IF (DATAX(IP2+1,IC).LT.0.0) DATAX(IP2+1,IC)=DATAX(IR2+1,IC)+360.0
115      C
      GO TO 90
      80 DATAX(IP2, IC) = 0.
      DATAX(IR2 + 1, IC) = C.
      90 CONTINUE
120
      IF (IC2D2 - ICN .LE. 0) GO TO 102
      IF (IPX - IPN .LE. 0) GO TO 101
      DATAX(IRX - IPN,IC) = DATAX(IP2,IC2D2 - ICN) =
125      1DATAX(IRX - IPN,IC2D2 - ICN) = DATAX(IR2, IC)
      DATAX(IRX - IPN + 1,IC) = DATAX(IR2 + 1,IC2D2 - ICN) =
      1DATAX(IPY - IPN + 1,IC2D2 - ICN) = DATAX(IR2 + 1, IC)
      GO TO 100
101 DATAX(IP2, IC2D2 - ICN) = DATAX(IR2, IC)
      DATAX(IP2 + 1, IC2D2 - ICN) = DATAX(IR2 + 1, IC)
130      GO TO 100
102 IF (IPX - IPN .LE. 0) GO TO 100
      DATAX(IPX - IPN,IC) = DATAX(IR2,IC)
      DATAX(IPX - IPN + 1,IC) = DATAX(IR2 + 1,IC)
100 CONTINUE
135      IPN = IPN + 2
      62 CONTINUE
      ICN = ICN + 1
      61 CONTINUE

C
C WRITE FAR-FIELD OUTPUT ON UNIT LUOUT
C
      CALL FFOUT(DATAX, IR2X2, IC2TON, LUOUT)
C
C CALCULATE NEAR-FIELD
145      C
      CALL NFKXY(DATAX, IR2X2, IC2TON)
C
      2000 FORMAT (I5, (T6,10F12.4))
      2001 FORMAT (1H0)
150      PRTUON
      END

```

```

FFKXY 78
FFKXY 79
FFKXY 80
FFKXY 81
FFKXY 82
FFKXY 83
FFKXY 84
FFKXY 85
FFKXY 86
FFKXY 87
FFKXY 88
FFKXY 89
FFKXY 90
FFKXY 91
FFKXY 92
FFKXY 93
FFKXY 94
FFKXY 95
FFKXY 96
FFKXY 97
FFKXY 98
FFKXY 99
FFKXY 100
FFKXY 101
FFKXY 102
FFKXY 103
FFKXY 104
FFKXY 105
FFKXY 106
FFKXY 107
FFKXY 108
FFKXY 109
FFKXY 110
FFKXY 111
FFKXY 112
FFKXY 113
FFKXY 114
FFKXY 115
FFKXY 116
FFKXY 117
FFKXY 118
FFKXY 119
FFKXY 120
FFKXY 121
FFKXY 122
FFKXY 123
FFKXY 124
FFKXY 125
FFKXY 126
FFKXY 127
FFKXY 128
FFKXY 129
FFKXY 130
FFKXY 131
FFKXY 132
FFKXY 133
FFKXY 134
FFKXY 135
FFKXY 136
FFKXY 137
FFKXY 138

FFKXY 139
FFKXY 140
FFKXY 141
FFKXY 142
FFKXY 143
FFKXY 144
FFKXY 145
FFKXY 146
FFKXY 147
FFKXY 148
FFKXY 149
FFKXY 150
FFKXY 151

```

#### A.1.4 SUBROUTINE NFKXY(DATA,IR2X2,IC2TON)

##### PURPOSE:

To calculate an array of near-field electric field values for an antenna given an array expressing the far-field radiation pattern of the antenna.

##### ARGUMENTS:

DATA - A two-dimensional array which on entry contains one component of the far-field radiation pattern of an antenna expressed in polar form as a function of equally spaced  $k_x$  and  $k_y$  coordinates. The amplitudes and phases are stored in adjacent locations in memory. On exit, this array contains the near-field pattern in polar form as a function  $x$  and  $y$ .

IR2X2 is twice the number of rows ( $k_y$  or  $y$  values) in the array DATA.

IC2TON is the number of columns in the array DATA.

##### METHODS:

The expression evaluated in this subroutine is basically eq (45) repeated below.

$$E_y(\underline{r}) = \frac{1}{2\pi} \sqrt{\frac{P_o G(0)}{2\pi \gamma_o k_o^2}} \int s_{10}(\underline{k}) e^{i|\underline{\gamma}|z} e^{i\underline{k} \cdot \underline{R}} d\underline{k} . \quad (45)$$

The quantity  $s_{10}(\underline{k})$  is the normalized transmitting coefficient and is given in terms of the far field by

$$s_{10}(\underline{k}) = \frac{E(\theta, \phi)}{\gamma E_y(0)}$$

This quantity is multiplied by the power normalization factor in front of the integral and stored in the input array on entry to the program. The integral is converted into a discrete Fourier transform (DFT) and evaluated using the FFT algorithm. The resulting summation is,

$$E_y(\underline{r}) = \frac{1}{2\pi} \sqrt{\frac{P_o G(0)}{2 \gamma_o k_o^2}} \frac{\Delta k_x \Delta k_y}{E_y(0)} \sum_{i=-N}^N \sum_{j=-M}^M \frac{E(i,j)}{\gamma_{ij}} e^{i\gamma_{ij}z} e^{iK_{ij} R_{ij}}$$

The subroutine ETOIGAM is called for each column in order to multiply the input data by

$$e^{i\gamma_{iy}z}.$$

This array DATA is then converted from polar to rectangular form using subroutine ARAYPTR. The Fourier transform is then performed using FFT routine FOURT and the results converted to polar form using subroutine ARAYRTP.

The results of the FFT must be corrected in two ways because of the nature of the FFT algorithm and the indexing system used in FORTRAN. First, the summation indices must be changed to 1 to  $2N(M)+1$ , rather than  $-N$  to  $N$ , as in eq (1). Second, the output is in a range 0 to  $2\pi$  rather than  $-\pi$  to  $\pi$ . The first is equivalent to a shift in origin, and, by the shifting theorem of Fourier analysis, produces a linear phase shift after transformation to the near field. The second effect causes the center of the near field to be located at the point (1,1) in the output array and the negative and x- and y-positions are in the outer portions of the array. The output data are rearranged in order to place the center of the near field at the center of the array. This is accomplished using subroutine SWAP. The phase shift is corrected in PHSCOR2.

The data in corrected form now reside in array DATA. Printer plots are produced using subroutine PLT12OR. This subroutine uses library functions, ATAN and subroutines ARAYPTR, ETOIGAM, FOURT, SWAP, and PHSCOR2.

#### SYMBOL DICTIONARY:

CEE	= Speed of light x $10^{-9}$
DATA(I,J)	= Angular spectrum which is transformed to near-electric field
E(I)	= Near-field magnitude array for plot (single cut)
FACTOR	= Scale factor to give near-field units of volts/meter
FLMDA	= Wavelength
I	= Index for plotting array
IC	= Column loop index
IC2TON	= Number of columns in array DATA
ICOL	= Column loop index
IK	= Row loop index
IR2TON	= Number of rows in array DATA
IR2X2	= 2 x IR2TON
IROW	= Row loop index
ISIGN	= +1 for forward Fourier transform: -1 for inverse Fourier transform
NN(I)	= Array specifying the dimensions of the FFT to be processed in each direction
P(I)	= Near-field phase array for plot (single cut)

PI =  $\pi$  = 3.14159....  
PIX2 =  $2\pi$   
RTD = Radian to degree conversion factor =  $180/\pi$   
X(I) = x-coordinate array used in near-field plots  
XMAX = Maximum value of x for plots  
XMIN = Minimum value of x for plots

```

1          SUBROUTINE NEKXY(DATA, IR2X2, IC2TON)                                NEKXY      1
C-                                               NEKXY      2
C- CALCULATES THE NEAR-FIELD DISTRIBUTION IN A PLANE GIVEN THE          NEKXY      3
C- FAR-FIELD ANGULAR SPECTRUM.                                         NEKXY      4
5          C-                                                             NEKXY      5
          DIMENSION F(128), P(128), X(128), Y(128)                          NEKXY      6
          DIMENSION DATA(IR2X2, IC2TON), NN(2)                            NEKXY      7
C-                                                                 NEKXY      8
          COMMON /I0/ CASEID(8)                                           NEKXY      9
10         COMMON /CNTRL/ DLTH, DLPH, DELX, DELY, FREQ, DIST, PNRM         NEKXY     10
          PI = 4.*ATAN(1.)          &    PIX2 = 2.*PI                    NEKXY     11
          RTD = 180./PI                                                    NEKXY     12
          NN(1) = IR2TON = IR2X2/2          &    NN(2) = IC2TON          NEKXY     13
15         CFF = .2997925          &    FLMDA = CFF/FREQ                  NEKXY     14
          ISIGN = 1                                                         NEKXY     15
C-                                                                 NEKXY     16
C- APPLY F TO I*GAMMA*0 PHASE CORRECTION COLUMN BY COLUMN              NEKXY     17
C-                                                                 NEKXY     18
          DO 50 IC = 1, IC2TON                                             NEKXY     19
20         CALL FT0IGAM(DATA(1,IC), IP2TON, IC2TON, IC, +1, FLMDA, DELX, NEKXY     20
          1 DELY, DIST)                                                    NEKXY     21
          50 CONTINUE                                                     NEKXY     22
C-                                                                 NEKXY     23
C- CHANGE DATA ARRAY FROM POLAR TO RECTANGULAR FORM.                  NEKXY     24
25         C-                                                             NEKXY     25
          CALL ARAYPTR(DATA, IR2X2, IC2TON)                                NEKXY     26
C-                                                                 NEKXY     27
C- REPERFORM FOURIER TRANSFORM OF DATA ARRAY TO PRODUCE NEAR-FIELD.   NEKXY     28
C-                                                                 NEKXY     29
30         CALL FOURT(DATA, NN, 2, ISIGN, +1, 0)                            NEKXY     30
C-                                                                 NEKXY     31
C- CHANGE NEAR-FIELD DATA FROM RECTANGULAR TO POLAR FORM.            NEKXY     32
C-                                                                 NEKXY     33
          CALL ARAYRTP(DATA, IR2X2, IC2TON)                                NEKXY     34
35         FACTOR = PIX2/(FLMDA*(IP2TON)*FLMDA*(IC2TON)*DELX*DELY)       NEKXY     35
          DO 55 ICOL = 1, IC2TON                                           NEKXY     36
          DO 55 IROW = 1, IR2X2, 2                                          NEKXY     37
          DATA(IROW, ICOL) = DATA(IROW, ICOL)*FACTOR                    NEKXY     38
          55 CONTINUE                                                     NEKXY     39
40         CALL SWAP(IR2X2, IC2TON, DATA)                                  NEKXY     40
          CALL PHSCOR2(DATA, IR2X2, IC2TON)                                NEKXY     41
C-                                                                 NEKXY     42
          DO 100 IX = 1, IP2TON                                             NEKXY     43
45         100 X(IX) = DELX*(IX - IP2TON/2)                                NEKXY     44
C-                                                                 NEKXY     45
          DO 110 IY = 1, IC2TON                                             NEKXY     46
          110 Y(IY) = DELY*(IY - IC2TON/2)                                NEKXY     47
C-                                                                 NEKXY     48
          PRINT 2001                                                       NEKXY     49
50         PRINT 2002, (X((IK+1)/2), DATA(IK, IC2TON/2), DATA(IK+1, IC2TON/2), NEKXY     50
          1 Y((IK+1)/2), DATA(IP2TON-1, (IK+1)/2), DATA(IP2TON, (IK+1)/2), NEKXY     51
          2 IK = 1, IR2X2, 2)                                              NEKXY     52
          XMIN = Y(1)          &    XMAX = X(IP2TON)                       NEKXY     53
          DO 60 IK = 1, IR2X2, 2                                          NEKXY     54
          I = (IK + 1)/2                                                  NEKXY     55
          E(I) = DATA (IK, IC2TON/2)                                     NEKXY     56
          R(I) = DATA (IK + 1, IC2TON/2)                                 NEKXY     57
          60 CONTINUE                                                     NEKXY     58
C-                                                                 NEKXY     59
60         C- PLOT E-PLANE AMPLITUDE AND PHASE.                          NEKXY     60
C-                                                                 NEKXY     61
          CALL PLOT120R(X, E, XMAX, XMIN, 10., 0., IC2TON, 14*, 1, 1)     NEKXY     62
          PRINT 2003, CASEID, 10HY-Z PLANE , 10HAMPLITUDE                NEKXY     63
          CALL PLOT120R(X, P, XMAX, XMIN, 360., 0., IC2TON, 14*, 1, 1)   NEKXY     64
65         PRINT 2003, CASEID, 10HY-Z PLANE , 10HPPHASE                  NEKXY     65
          YMIN = Y(1)          &    YMAX = Y(IC2TON)                       NEKXY     66
          DO 61 I = 1, IC2TON                                             NEKXY     67
          E(I) = DATA(IP2TON - 1, I)                                     NEKXY     68
          P(I) = DATA(IP2TON, I)                                       NEKXY     69
70         61 CONTINUE                                                     NEKXY     70
C-                                                                 NEKXY     71
C- PLOT H-PLANE AMPLITUDE AND PHASE.                                     NEKXY     72
C-                                                                 NEKXY     73
          CALL PLOT120R(Y, E, YMAX, YMIN, 10., 0., IC2TON, 14*, 1, 1)   NEKXY     74
75         PRINT 2003, CASEID, 10HX-Z PLANE , 10HAMPLITUDE                NEKXY     75
          CALL PLOT120R(Y, P, YMAX, YMIN, 360., 0., IC2TON, 14*, 1, 1)   NEKXY     76
          PRINT 2003, CASEID, 10HX-Z PLANE , 10HPPHASE                  NEKXY     77

```



RD

```
      RETURN                                NFKXY      78
2001  FORMAT(///,T64,*CENTERLINE DATA*,//,T37,*X-Z PLANE*, T97, *Y-Z PLANENFKXY 79
      1NF*,/,T22,*X*,T40,*AMP*,T59,*PHASE*,T82,*Y*,T100,*AMP*,T119,*PHASENFKXY 80
      2*)                                    NFKXY      81
2002  FORMAT(T6, 6F20.4)                    NFKXY      82
2003  FORMAT(/,5X, 8A10, 5X, 2A10)          NFKXY      83
      END                                    NFKXY      84
```

### A.1.5 SUBROUTINE

ETOIGAM(DATA(1,ICOL),NROW,NCOL,ICOL,ISGN,FLMDA,DELX,DELY,DIST)

PURPOSE:

To multiply each element of complex array DATA by the factor  $\exp(\pm i\gamma d)$ .

ARGUMENTS:

DATA is a two-dimensional complex array in polar form whose magnitude and phase are adjacent in storage.

NROW is the number of rows in array DATA.

NCOL is the number of columns in array DATA.

ICOL is column number of the data to be operated on.

ISGN =  $\pm 1$  depending on whether DATA is to be multiplied by  $\exp(\pm i\gamma d)$ .

FLMDA operating wavelength.

DELX x-increment of desired near-field data.

DELY y-increment of desired near-field data.

DIST spacing between antenna reference point and desired near-field plane.

METHODS:

The subroutine does not employ complex arithmetic. It is assumed that the numbers in array DATA are the magnitude and phase stored in adjacent locations. If DATA contains complex data in real and imaginary form, a call to ARAYRTP must be made prior to the call to ETOIGAM. The pertinent relationships are

$$\begin{aligned} \text{DATA} &= \text{DATA} \frac{e^{i\gamma d}}{\gamma} \\ \gamma &= \sqrt{k^2 - k_x^2 - k_y^2} \\ k &= \omega \sqrt{\mu\epsilon} \\ k_x &= k \sin\theta \cos\phi \\ k_y &= k \sin\theta \sin\phi. \end{aligned}$$

Because DATA is assumed to be in magnitude, argument form, we calculate

$$\text{ARG}(\text{DATA}) = \text{ARG}(\text{DATA}) + \gamma d$$

for  $\gamma$  real, and

$$\text{MAG}(\text{DATA}) = \text{MAG}(\text{DATA}) \text{Exp}(-\gamma d)$$

for  $\gamma$  imaginary

$\gamma$  is computed from the row and column positions of the data elements.

$$k_y = \frac{2\pi (IROW-NROW/2)}{NROW \cdot \Delta_y}$$

$$k_x = \frac{2\pi (ICOL-NCOL/2)}{NCOL \cdot \Delta_x}$$

Array DATA is assumed to correspond to points equally spaced in  $k_x$ ,  $k_y$  with  $k_x=k_y=0$  being the center point of the array.

#### SYMBOL DICTIONARY:

DATA	= Input data array
DELX	= Near-field x-increment
DELY	= Near-field y-increment
DIST	= Distance from antenna reference point to desired near-field plane
DTOR	= $\pi/180$ = degree to radian conversion
FACTOR	= Amplitude correction factor for imaginary $\gamma$
FKAY	= $k = 2\pi/\lambda$ = free space wave number
FKAYSQ	= $k^2$
FKX	= $k_x$ = x-component of propagation vector
FKXSQ	= $k_x^2$
FKY	= $k_y$ = y-component of propagation vector
FKYSQ	= $k_y^2$
FLMDA	= Operating wavelength
ICOL	= Running index for column number
IROW	= Running index for row number
ISGN	= $\pm 1$ = desired sign for exponential phase factor
NCOL	= Number of columns in far-field array
NROW	= Number of rows in far-field array
PHACORR	= Phase correction factor added to data array phases for real
PI	= $\pi = 3.14159\dots$
PIX2	= $2\pi$
RTOD	= $180/\pi$ = radian to degree conversion
SUMSQ	= $k_x^2 + k_y^2$
TEMP	= Intermediate variable

1		SUBROUTINE ETOIGAM (DATA, NROW, NCOL, ICOL, ISGN, FLMDA, DELX, DELETOIGAM	1
	C	1Y, DISI)	2
			3
5	C	DIMENSION DATA (1)	4
			5
		PI = 3.1415926536	6
		PIX2 = 2. * PI	7
		EKAY = PIX2 / FLMDA	8
		EKAYSQ = EKAY * * 2	9
10		RTOD = 180. / PI	10
		DTOR = 1. / RTOD	11
		EKX = PIX2 * (ICOL - (NCOL / 2)) / DELX / NCOL	12
		EKXSQ = EKX * * 2	13
	C		14
15		IF (NROW .LT. 1) GO TO 130	15
		OR 120 IROW = 1, NROW	16
		EKY = PIX2 * (IROW - (NROW / 2)) / DELY / NROW	17
		EKYSQ = EKX * * 2	18
		SUMSQ = EKXSQ + EKYSQ	19
20		PHACORR = 0.0	20
		FACTOR = 1.0	21
		IF (SUMSQ .GT. EKAYSQ) GO TO 100	22
	C		23
25		PHACORR = ISGN * SQRT (EKAYSQ - SUMSQ) * DIST * RTOD	24
		GO TO 110	25
	C		26
	100	FACTOR = (SQRT (SUMSQ - EKAYSQ)) * DIST	27
		IF (FACTOR .GT. 100.)FACTOR = 100.	28
		FACTOR = EXP (ISGN * FACTOR)	29
30			30
	C		31
	110	CONTINUE	32
			33
		DATA (2 * IROW - 1) = DATA (2 * IROW - 1) * FACTOR	34
35		TPHASE = DATA (2 * IROW)	35
		TEMP = DATA (2 * IROW) + PHACORR	36
		TEMP = TEMP - INT (TEMP / 360.) * 360.	37
		IF (TEMP .LT. 0.0)TEMP = TEMP + 360.0	38
		DATA (2 * IROW) = TEMP	39
	C		40
40		120 CONTINUE	41
		130 CONTINUE	42
		1500 FORMAT (1X, 4I10, 5F12.3, //)	43
		1510 FORMAT (1X, 2I5, 3F12.3)	44
		RETURN	45
45		END	45

#### A.1.6 SUBROUTINE PHSCOR2(DATA,NRX2,NCOL)

##### PURPOSE:

To correct the phase of the near-field data which arises because the reference point of the FFT algorithm is the point (1,1) rather than the center of the far-field array.

##### ARGUMENTS:

DATA is a two-dimensional array containing the complex near-field data in polar form. Amplitude and phase in degrees are located adjacent in storage.

NRX2 is twice the number of rows in the array DATA.

NCOL is the number of columns.

##### METHODS:

As shown by the shifting theorem, a shift in coordinates in one domain introduces a linear phase shift in the transformed domain. This subroutine corrects for the phase shift which occurs as a result of the different reference points of far-field pattern and the FFT algorithm. The shift added because of this change of origin is

$$e^{i(ax+by)}$$

where a and b are the shifts in far-field origin in the  $k_x$  and  $k_y$  directions respectively and x and y are the coordinates of the specific near-field point.

The subroutine adds a phase shift equal to

$$-180^\circ \left[ \left( \frac{NCOL-2}{NCOL} \right) (ICOL-1) + \left( \frac{NROW-2}{NROW} \right) (IROW-1) \right]$$

to the phase of each complex number in the array in order to compensate for the above shift. It is assumed in this factor that the center of the far-field pattern lies at (NROW/2,NCOL/2).

An additional phase of  $90^\circ$  is added to each element in order to allow the near-field phase to be conveniently plotted in the range  $0^\circ$ - $360^\circ$ .

This subroutine uses inline functions FLOAT and INT.

SYMBOL DICTIONARY:

C1	= Phase correction for column ICOL
C2	= Phase correction for row IROW
CONST1	= Column phase increment
CONST2	= Row phase increment
DATA	= Input data array
ICOL	= Column loop index
I02	= IROW/2
IROW	= Row loop index
NCOL	= Number of columns in array DATA
NRX2	= Twice the number of rows in array DATA
TEMP	= Intermediate variable, the corrected phase at point (I02,ICOL)

1		SUBROUTINE PHSCOR2(DATA, NRX2, NCOL)	PHSCOR2	1
	C		PHSCOR2	2
		DIMENSION DATA(NRX2, NCOL)	PHSCOR2	3
	C		PHSCOR2	4
5		NROW = NRX2 / 2	PHSCOR2	5
	C		PHSCOR2	6
	C		PHSCOR2	7
		CONST1 = -180.*FLOAT(NCOL - 2)/FLOAT(NCOL)	PHSCOR2	8
		CONST2 = -180.*FLOAT(NROW - 2)/FLOAT(NROW)	PHSCOR2	9
10		IF (NCOL .LT. 1) GO TO 130	PHSCOR2	10
		DO 100 ICOL = 1, NCOL	PHSCOR2	11
		C1 = CONST1 * (ICOL - 1)	PHSCOR2	12
		IF (NRX2 .LT. 2) GO TO 110	PHSCOR2	13
		DO 100 IROW = 2, NRX2, 2	PHSCOR2	14
15		IC2 = IROW / 2	PHSCOR2	15
		C2 = CONST2 * (IC2 - 1) + C1	PHSCOR2	16
		TEMP = DATA(IROW, ICOL) + C2 + 90.	PHSCOR2	17
		TEMP = TEMP - INT(TEMP / 360.) * 360.	PHSCOR2	18
20		IF (TEMP .LT. 0.) TEMP = TEMP + 360.	PHSCOR2	19
		DATA(IROW, ICOL) = TEMP	PHSCOR2	20
	100	CONTINUE	PHSCOR2	21
	110	CONTINUE	PHSCOR2	22
	130	CONTINUE	PHSCOR2	23
	C		PHSCOR2	24
25		RETURN	PHSCOR2	25
		END	PHSCOR2	26

### A.1.7 SUBROUTINE SWAP(NRX2,NCOL,DATA)

#### PURPOSE:

To perform the rearrangement of data necessary to place center of near field at center of near-field data array.

#### ARGUMENTS:

NRX2 is twice the number of rows in the array DATA.

NCOL is the number of columns in the array DATA.

DATA is an array containing the near-field pattern of an antenna which is to be rearranged.

#### METHODS:

The FFT algorithm fundamentally takes data over a range of  $0-2\pi$  and transforms them into a domain of  $0-2\pi$ . Suitable scaling is employed to fit the far-field (angular spectrum) and near field (x-y position) into these ranges. The negative portion of the x-y range occurs from  $\pi$  to  $2\pi$ . Thus, to have a continuous near field at  $x,y=0$ , the data are rearranged.

The rearrangement is done in place, the rearranged array replacing the original one in core, requiring only three temporary storage locations. The rearrangement takes place in two steps. First, the edges of the array are moved to the center and the center to the edges by columns. The process is then repeated by rows.

The array DATA contains complex numbers which may be in either polar or rectangular form. This routine does not use complex arithmetic.

#### SYMBOL DICTIONARY:

DATA	= Complex array to be rearranged
ICOL	= Column loop index
ICPNC	= Intermediate subscript
IROW	= Row loop index
IRPNR	= Intermediate subscript
NCM1	= NCOL -1
NCOL	= Number of columns in DATA
NCO2	= NCOL/2
NROW	= Number of rows of complex data
NRX2	= 2 * NROW = dimension of DATA in row directio.
NR2M2	= NRX2-2



TEMP = Intermediate variable  
TEMP1 = Intermediate variable  
TEMP2 = Intermediate variable

1	SUBROUTINE SWAP(NRX2, NCOL, DATA)	SWAP	1
	DIMENSION DATA(NRX2, NCOL)	SWAP	2
	C	SWAP	3
5	NRPW = NRX2 / 2	SWAP	4
	NCOL2 = NCOL / 2	SWAP	5
	C	SWAP	6
	C-MOVING EDGES OF ARRAY TO CENTER AND VICE VERSA BY COLUMNS	SWAP	7
	C	SWAP	8
10	IF (NRX2 .LT. 1) GO TO 220	SWAP	9
	DO 200 IROW = 1, NRX2	SWAP	10
	IF (NCOL2 .LT. 1) GO TO 210	SWAP	11
	DO 200 ICOL = 1, NCOL2	SWAP	12
	ICPNC = ICOL + NCOL2	SWAP	13
15	TEMP = DATA(IROW, ICPNC)	SWAP	14
	DATA(IROW, ICPNC) = DATA(IROW, ICOL)	SWAP	15
	200 DATA(IROW, ICOL) = TEMP	SWAP	16
	210 CONTINUE	SWAP	17
	220 CONTINUE	SWAP	18
20	C	SWAP	19
	NCM1 = NCOL - 1	SWAP	20
	IF (NRX2 .LT. 1) GO TO 310	SWAP	21
	DO 300 IROW = 1, NRX2	SWAP	22
	TEMP1 = DATA(IROW, 1)	SWAP	23
25	IF (NCM1 .LT. 1) GO TO 280	SWAP	24
	DO 300 ICOL = 1, NCM1	SWAP	25
	230 DATA(IROW, ICOL) = DATA(IROW, ICOL + 1)	SWAP	26
	280 CONTINUE	SWAP	27
	300 DATA(IROW, NCOL) = TEMP1	SWAP	28
30	310 CONTINUE	SWAP	29
	C	SWAP	30
	C-MOVING EDGES OF ARRAY TO CENTER AND VICE VERSA BY ROWS	SWAP	31
	C	SWAP	32
35	IF (NCOL .LT. 1) GO TO 340	SWAP	33
	DO 320 ICOL = 1, NCOL	SWAP	34
	IF (NRPW .LT. 1) GO TO 330	SWAP	35
	DO 320 IROW = 1, NRPW	SWAP	36
	IRPNR = IROW + NRPW	SWAP	37
	TEMP = DATA(IRPNR, ICOL)	SWAP	38
40	DATA(IRPNR, ICOL) = DATA(IROW, ICOL)	SWAP	39
	320 DATA(IROW, ICOL) = TEMP	SWAP	40
	330 CONTINUE	SWAP	41
	340 CONTINUE	SWAP	42
45	C	SWAP	43
	NR2M2 = NRX2 - 2	SWAP	44
	IF (NCOL .LT. 1) GO TO 380	SWAP	45
	DO 370 ICOL = 1, NCOL	SWAP	46
	TEMP1 = DATA(1, ICOL)	SWAP	47
	TEMP2 = DATA(2, ICOL)	SWAP	48
50	IF (NR2M2 .LT. 1) GO TO 360	SWAP	49
	DO 350 IROW = 1, NR2M2	SWAP	50
	350 DATA(IROW, ICOL) = DATA(IROW + 2, ICOL)	SWAP	51
	360 CONTINUE	SWAP	52
	DATA(NR2M2 + 1, ICOL) = TEMP1	SWAP	53
55	370 DATA(NR2M2 + 2, ICOL) = TEMP2	SWAP	54
	380 CONTINUE	SWAP	55
	C	SWAP	56
	RETURN	SWAP	57
	END	SWAP	58
		SWAP	59

## A.1.8 SUBROUTINE ARAYPTR(DATA,NRX2,NCOL)

### PURPOSE:

To convert a two-dimensional complex array from polar form to rectangular form or from rectangular form to polar form (ENTRY ARAYRTP).

### ARGUMENTS:

DATA is a two-dimensional complex array whose real and imaginary parts are adjacent in storage, such as FORTRAN IV places them. On exit, DATA contains adjacent amplitudes and phases.

NRX2 is twice the number of rows in DATA.

NCOL is the number of columns in DATA.

### ENTRY POINT:

ARAYRTP performs rectangular to polar conversion.

### METHODS:

This subroutine does not use complex arithmetic. However, array DATA is stored in the same fashion as is required by FORTRAN IV for complex numbers. Thus, while the subroutine operates on a complex array, complex FORTRAN functions are not used.

#### 1. ARAYPTR

DATA(IROW,ICOL) contains magnitude of complex number.

DATA(IROW+1,ICOL) contains phase of complex number, expressed in degrees.

$\text{Re}(\text{DATA}) = |\text{DATA}| \cos(\text{ANGLE}(\text{DATA}))$ .

$\text{Im}(\text{DATA}) = |\text{DATA}| \sin(\text{ANGLE}(\text{DATA}))$ .

#### 2. ARAYRTP

DATA(IROW,ICOL) contains real part of the complex number.

DATA(IROW+1,ICOL) contains imaginary part of the complex number.

$$|\text{DATA}| = \sqrt{[\text{Re}(\text{DATA})]^2 + [\text{Im}(\text{DATA})]^2}$$

$$\text{ARG}(\text{DATA}) = \tan^{-1} \left[ \frac{\text{Im}(\text{DATA})}{\text{Re}(\text{DATA})} \right] \times 180^\circ/\pi$$

This subroutine uses library functions, SIN, COS, ATAN2, and SQRT.

## SYMBOL DICTIONARY

ANGLE	= Intermediate variable, phase angle of complex number
DATA	= Input data array
DTOR	= $\pi/180$ = degree to radian conversion factor
FIMAG	= Imaginary part of complex number
FREAL	= Real part of complex number
ICOL	= Column loop running index
IROW	= Row loop running index
IRP1	= IROW + 1
NCOL	= Number of columns in input array DATA
NROW	= Number of rows in input array DATA
NRX2	= $2 \cdot \text{NROW}$
PI	= $\pi = 3.14159\dots$
RTOD	= $180/\pi$ = radian to degree conversion factor
TAMP	= Intermediate variable, amplitude of complex number

1	C	SUBROUTINE ARAYPTR(DATA, NRX2, NCOL)	ARAYPTR	1
	C	DIMENSION DATA(NRX2, NCOL)	ARAYPTR	2
5	C	PI = 3.1415926536	ARAYPTR	3
		RTOD = 180. / PI	ARAYPTR	4
		DTOR = 1. / RTOD	ARAYPTR	5
	C	IF (NCOL .LT. 1) GO TO 130	ARAYPTR	6
10		DO 120 ICOL = 1, NCOL	ARAYPTR	7
		IF (NRX2 .LT. 1) GO TO 110	ARAYPTR	8
		DO 100 IPRW = 1, NRX2, 2	ARAYPTR	9
		IRP1 = IPRW + 1	ARAYPTR	10
		TAMP = DATA(IPRW, ICOL)	ARAYPTR	11
15		ANGLE = DATA(IRP1, ICOL) * DTOR	ARAYPTR	12
		DATA(IPRW, ICOL) = TAMP * COS(ANGLE)	ARAYPTR	13
		DATA(IRP1, ICOL) = TAMP * SIN(ANGLE)	ARAYPTR	14
	100	CONTINUE	ARAYPTR	15
	110	CONTINUE	ARAYPTR	16
20	120	CONTINUE	ARAYPTR	17
	130	CONTINUE	ARAYPTR	18
		RETURN	ARAYPTR	19
	C	ENTRY ARAYRTR	ARAYPTR	20
25	C	PI = 3.1415926536	ARAYPTR	21
		RTOD = 180. / PI	ARAYPTR	22
		DTOR = 1. / RTOD	ARAYPTR	23
	C	IF (NCOL .LT. 1) GO TO 190	ARAYPTR	24
30		DO 180 ICOL = 1, NCOL	ARAYPTR	25
		IF (NRX2 .LT. 1) GO TO 170	ARAYPTR	26
		DO 160 IPRW = 1, NRX2, 2	ARAYPTR	27
		IRP1 = IPRW + 1	ARAYPTR	28
35		FFREAL = DATA(IPRW, ICOL)	ARAYPTR	29
		FIMAG = DATA(IRP1, ICOL)	ARAYPTR	30
		DATA(IPRW, ICOL) = SQRT(FFREAL * FFREAL + FIMAG * FIMAG)	ARAYPTR	31
		IF (FFREAL .EQ. 0.0 .AND. FIMAG .EQ. 0.0) GO TO 160	ARAYPTR	32
		DATA(IRP1, ICOL) = ATAN2(FIMAG, FFREAL) * RTOD	ARAYPTR	33
40	160	CONTINUE	ARAYPTR	34
	170	CONTINUE	ARAYPTR	35
	180	CONTINUE	ARAYPTR	36
	190	CONTINUE	ARAYPTR	37
	C	RETURN	ARAYPTR	38
45		END	ARAYPTR	39
			ARAYPTR	40
			ARAYPTR	41
			ARAYPTR	42
			ARAYPTR	43
			ARAYPTR	44
			ARAYPTR	45
			ARAYPTR	46

## A.1.9 SUBROUTINE FFOUT(DATA,NRX2,NCOL, LUOUT)

### PURPOSE:

This subroutine writes the array DATA out to logical unit LUOUT. A header record is written as the first record of the file.

### ARGUMENTS:

DATA is the array to be written out.  
NRX2 is the number of floating point numbers in a row.  
NCOL is the number of columns.  
LUOUT is the logical unit on which file is to be written.

### METHODS:

A file consisting of NCOL + 1 records is written on unit LUOUT. The first record is an identification (ID) record, and each of the NCOL rows in the array DATA is a record. The records are written using unformatted WRITE statements.

The ID record consists of ten ten-character words, these are listed below

WORD 1	PHYSICAL 0
WORD 2	PTICS SIML
WORD 3-7	Alphanumeric information from the first 5 words in common block CASEID
WORD 8	MMDDYYHHNN Month, Day, Year, Hour, Minute
WORD 9	Number of columns in DATA = NCOL
WORD 10	Twice the number of rows in DATA = NRX2

Words 1 through 8 are written in Hollerith format, and words 9 and 10 are written in integer (I) format. The ID record may be read with a (8A10,2I10) format. The date and time for word 8 are generated by calls to DATE and TIME and are thus the date and time when the output file was created.

The subroutine uses library functions DATE and TIME.

### SYMBOL DICTIONARY:

CASEID	= Hollerith identification supplied from calling program
DATA	= Array to be written to unit LUOUT
DT	= Date information obtained from function DATE
DY	= Day of month

HR	= Hour
I	= DO loop index
IC	= Column loop index
ID	= Identification array
IR	= Row loop index
LUOUT	= Output logical unit number
MN	= Minute
MON	= Month
NCOL	= Number of columns in DATA
NRX2	= Number of rows in DATA
SC	= Second
TM	= Time information obtained from function TIME
YR	= Year

1	SUBROUTINE FFOUT(DATA, NRX2, NCOL, LUOUT)	FFOUT	1
	C	FFOUT	2
	C THIS SUBROUTINE WRITES ARRAY DATA TO FILE LUOUT	FFOUT	3
	C	FFOUT	4
5	DIMENSION DATA(NRX2, NCOL), ID(10)	FFOUT	5
	COMMON / ID/ CASEID(8)	FFOUT	6
	INTEGER CASEID	FFOUT	7
	C	FFOUT	8
	ID(1) = 10HPHYSICAL C	FFOUT	9
10	ID(2) = 10HPTICS SIML	FFOUT	10
	DO 10 I = 3, 7	FFOUT	11
	10 ID(I) = CASEID(I - 2)	FFOUT	12
	CALL DATE(DT)	FFOUT	13
	CALL TIME(TM)	FFOUT	14
15	DECODE(10, 1500, DT) YR, MON, DY	FFOUT	15
	DECODE(10, 1500, TM) HR, MN, SC	FFOUT	16
	ENCODE(10, 1510, ID(8)) MON, DY, YR, HR, MN	FFOUT	17
	ID(9) = NCOL	FFOUT	18
	ID(10) = NRX2	FFOUT	19
20	WRITE (LUOUT) (ID(I), I = 1,10)	FFOUT	20
	PRINT 1520, ID	FFOUT	21
	DO 20 IC = 1, NCOL	FFOUT	22
	WRITE (LUOUT) (DATA(IR, IC), IR = 1, NRX2)	FFOUT	23
	20 CONTINUE	FFOUT	24
25	ENDFILE LUOUT	FFOUT	25
	RETURN	FFOUT	26
	1500 FORMAT (1X, 3(A2, 1X))	FFOUT	27
	1510 FORMAT (5A2)	FFOUT	28
	1520 FORMAT (*00OUTPUT FILE ID -- *, 10X, 8A10, 2I10)	FFOUT	29
30	END	FFOUT	30



## A.1.10 SUBROUTINE FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)

### PURPOSE:

To compute the discrete Fourier transform of the array DATA using the fast Fourier transform algorithm.

### ARGUMENTS:

DATA is a multidimensional complex array whose real and imaginary parts are adjacent in storage, such as FORTRAN IV places them.

NN is an array giving the lengths of the array in each dimension.

NDIM is the number of dimensions of the array DATA, hence the number of elements in array NN.

ISIGN is +1 for a forward transform -1 for a reverse transform.

IFORM If all imaginary parts of the input array are zero (input array is real), set IFORM = 0 to reduce running time by approximately 40 percent, otherwise set IFORM = +1.

WORK if all dimensions of DATA are not integral powers of 2, specify array WORK in calling routine with dimension greater than largest non  $2^k$  dimension, otherwise set WORK = 0.

### METHODS:

Using the Fast Fourier transform algorithm, FOURT replaces the array DATA with its discrete Fourier transform given by

$$\text{TRANSFORM}(K1,K2,\dots) = \sum_{J1=1}^{NN(1)} \sum_{J2=1}^{NN(2)} \text{DATA}(J1,J2) e^{i 2\pi \text{ISIGN} \left\{ \frac{(J1-1)(K1-1)}{NN(1)} + \frac{(J2-1)(K2-1)}{NN(2)} + \dots \right\}}$$

For a more complete description of the subroutine and its usage, see the comments included at the beginning of its listing or the supplementary comments by the programmer, Norman Brenner of MIT.

Uses external library functions COS, SIN, FLOAT, and MAX0.

Note: Brenner, Norman, "FOUR2 and FOURT program description," private communication, 1968.

```

1      SUPPLEMENT FOURT (DATA, NN, NDIM, ISIGN, IFORM, WORK)          FORT 1
C
C      THE COLLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN FORT 2
C      FORT 3
C      FORT 4
5      TRANSFORM(K1,K2,...) = SUM(DATA(J1,J2,...)*FXP(ISIGN*2*PI*SQRT(-1) FORT 5
C      *((J1-1)*(K1-1)/NN(1)+(J2-1)*(K2-1)/NN(2)+...))), SUMMED FOR ALL FORT 6
C      J1, K1 BETWEEN 1 AND NN(1), J2, K2 BETWEEN 1 AND NN(2), ETC. FORT 7
C      THERE IS NO LIMIT TO THE NUMBER OF SUBSCRIPTS. DATA IS A FORT 8
C      MULTIDIMENSIONAL COMPLEX ARRAY WHOSE REAL AND IMAGINARY FORT 9
10     C PARTS ARE ADJACENT IN STORAGE, SUCH AS FORTRAN IV PLACES THEM. FORT 10
C      IF ALL IMAGINARY PARTS ARE ZERO (DATA ARE DISGUISED REAL), SET FORT 11
C      IFORM TO ZERO TO CUT THE RUNNING TIME BY UP TO FORTY PERCENT. FORT 12
C      OTHERWISE, IFORM = +1. THE LENGTHS OF ALL DIMENSIONS ARE FORT 13
C      STORED IN ARRAY NN, OF LENGTH NDIM. THEY MAY BE ANY POSITIVE FORT 14
15     C INTEGERS, THO THE PROGRAM RUNS FASTER ON COMPOSITE INTEGERS. AND FORT 15
C      ESPECIALLY FAST ON NUMBERS RICH IN FACTORS OF TWO. ISIGN IS +1 FORT 16
C      OR -1. IF A -1 TRANSFORM IS FOLLOWED BY A +1 ONE (OR A +1 FORT 17
C      BY A -1) THE ORIGINAL DATA REAPPEAR, MULTIPLIED BY NTOT (=NN(1)* FORT 18
C      NN(2)*...). TRANSFORM VALUES ARE ALWAYS COMPLEX, AND ARE RETURNED FORT 19
20     C IN ARRAY DATA, REPLACING THE INPUT. IN ADDITION, IF ALL FORT 20
C      DIMENSIONS ARE NOT POWERS OF TWO, ARRAY WORK MUST BE SUPPLIED, FORT 21
C      COMPLEX OF LENGTH EQUAL TO THE LARGEST NON 2**K DIMENSION. FORT 22
C      OTHERWISE, REPLACE WORK BY ZERO IN THE CALLING SEQUENCE. FORT 23
C      NORMAL FORTRAN DATA ORDERING IS EXPECTED, FIRST SUBSCRIPT VARYING FORT 24
25     C FASTEST. ALL SUBSCRIPTS BEGIN AT ONE. FORT 25
C      FORT 26
C      RUNNING TIME IS MUCH SHORTER THAN THE NAIVE NTOT**2, BEING FORT 27
C      GIVEN BY THE FOLLOWING FORMULA. DECOMPOSE NTOT INTO FORT 28
C      2**K2 * 3**K3 * 5**K5 * .... LET SUM2 = 2**K2, SUM5 = 3**K3 + 5**K5 FORT 29
30     C + ... AND NF = K3 + K5 + .... THE TIME TAKEN BY A MULTI- FORT 30
C      DIMENSIONAL TRANSFORM ON THESE NTOT DATA IS T = T0 + NTOT*(T1+ FORT 31
C      T2*SUM2+T3*SUM5+T4*NF). ON THE CDC 3300 (FLOATING POINT ADD TIME FORT 32
C      OF SIX MICROSECONDS), T = 3000 + NTOT*(500+43*SUM2+68*SUM5+ FORT 33
C      320*NF) MICROSECONDS ON COMPLEX DATA. IN ADDITION, THE FORT 34
35     C ACCURACY IS GREATLY IMPROVED, AS THE RMS RELATIVE ERROR IS FORT 35
C      BOUNDED BY 3**(-B)*SUM(FACTOR(J)**1.5), WHERE B IS THE NUMBER FORT 36
C      OF BITS IN THE FLOATING POINT FRACTION AND FACTOR(J) ARE THE FORT 37
C      PRIME FACTORS OF NTOT. FORT 38
C      FORT 39
40     C PROGRAM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES FORT 40
C      RABER. RABER ALTER SUGGESTED THE IDEA FOR THE DIGIT REVERSAL. FORT 41
C      MIT LINCOLN LABORATORY, AUGUST 1967. THIS IS THE FASTEST AND MOST FORT 42
C      VERSATILE VERSION OF THE FFT KNOWN TO THE AUTHOR. SHORTER PRO- FORT 43
C      GRAMS FOUR1 AND FOUR2 RESTRICT DIMENSION LENGTHS TO POWERS OF TWO. FORT 44
45     C SEE-- IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON FFT. FORT 45
C      FORT 46
C      THE DISCRETE FOURIER TRANSFORM PLACES THREE RESTRICTIONS UPON THE FORT 47
C      DATA. FORT 48
50     C 1. THE NUMBER OF INPUT DATA AND THE NUMBER OF TRANSFORM VALUES FORT 49
C      MUST BE THE SAME. FORT 50
C      2. BOTH THE INPUT DATA AND THE TRANSFORM VALUES MUST REPRESENT FORT 51
C      EQUISPACED POINTS IN THEIR RESPECTIVE DOMAINS OF TIME AND FORT 52
C      FREQUENCY. CALLING THESE SPACINGS DELTAT AND DELTAF, IT MUST BE FORT 53
C      TRUE THAT DELTAF=2*PI/(NN(I)*DELAT). OF COURSE, DELTAT NEED NOT FORT 54
55     C BE THE SAME FOR EVERY DIMENSION. FORT 55
C      3. CONCEPTUALLY AT LEAST, THE INPUT DATA AND THE TRANSFORM OUTPUT FORT 56
C      REPRESENT SINGLE CYCLES OF PERIODIC FUNCTIONS. FORT 57
C      FORT 58
C      EXAMPLE 1. THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A FORT 59
60     C COMPLEX ARRAY DIMENSIONED 32 BY 25 BY 13 IN FORTRAN IV. FORT 60
C      DIMENSION DATA(32,25,13),WORK(50),NN(3) FORT 61
C
C      COMPLEX DATA FORT 62
C      DATA NN/32,25,13/ FORT 63
C      DO 1 I=1,32 FORT 64
65     C DO 1 J=1,25 FORT 65
C      DO 1 K=1,13 FORT 66
C      1 DATA(I,J,K)=COMPLEX VALUE FORT 67
C      CALL FOURT(DATA,NN,3,-1,1,WORK) FORT 68
C      FORT 69
70     C EXAMPLE 2. ONE-DIMENSIONAL FORWARD TRANSFORM OF A REAL ARRAY OF FORT 70
C      LENGTH 64 IN FORTRAN II. FORT 71
C      DIMENSION DATA(2,64) FORT 72
C      DO 2 I=1,64 FORT 73
C      DATA(1,I)=REAL PART FORT 74
75     C 2 DATA(2,I)=0. FORT 75
C      CALL FOURT(DATA,64,1,-1,0,0) FORT 76
C      FORT 77

```

		DIMENSION DATA (1), NN (1), IFACT (32), WORK (1)	FOURT	78
		WP = 0.	FOURT	79
		WI = 0.	FOURT	80
		WSTPR = 0.	FOURT	81
		WSTPI = 0.	FOURT	82
		TWOP1 = 6.293185307	FOURT	83
		IF (NDIM - 1)1290, 100, 100	FOURT	84
85	100	NTOT = 2	FOURT	85
		DO 110 IDIM = 1, NDIM	FOURT	86
		IF (NN (IDIM))1280, 1290, 110	FOURT	87
	110	NTOT = NTOT * NN (IDIM)	FOURT	88
	C		FOURT	89
90	C	MAIN LOOP FOR EACH DIMENSION	FOURT	90
	C		FOURT	91
		NP1 = 2	FOURT	92
		DO 1270 IDIM = 1, NDIM	FOURT	93
		N = NN (IDIM)	FOURT	94
95		NP2 = NP1 * N	FOURT	95
		IF (N - 1)1290, 1260, 120 -	FOURT	96
	C		FOURT	97
	C	FACTORY N	FOURT	98
	C		FOURT	99
100	120	M = N	FOURT	100
		NTW0 = NP1	FOURT	101
		IF = 1	FOURT	102
		IDIV = 2	FOURT	103
	130	IQUOT = M / IDIV	FOURT	104
105		IPEM = M - IDIV * IQUOT	FOURT	105
		IF (IQUOT - IDIV)210, 140, 140	FOURT	106
	140	IF (IPEM)160, 150, 160	FOURT	107
	150	NTW0 = NTW0 + NTW0	FOURT	108
		M = IQUOT	FOURT	109
110		GO TO 130	FOURT	110
	160	IDIV = 3	FOURT	111
	170	IQUOT = M / IDIV	FOURT	112
		IPEM = M - IDIV * IQUOT	FOURT	113
		IF (IQUOT - IDIV)230, 180, 180	FOURT	114
115	180	IF (IPEM)200, 190, 200	FOURT	115
	190	IFACT (IF) = IDIV	FOURT	116
		IF = IF + 1	FOURT	117
		M = IQUOT	FOURT	118
		GO TO 170	FOURT	119
120	200	IDIV = IDIV + 2	FOURT	120
		GO TO 170	FOURT	121
	210	IF (IPEM)230, 220, 230	FOURT	122
	220	NTW0 = NTW0 + NTW0	FOURT	123
		GO TO 240	FOURT	124
125	230	IFACT (IF) = M	FOURT	125
	C		FOURT	126
	C	SEPARATE FOUR CASES--	FOURT	127
	C	1. COMPLEX TRANSFORM OR REAL TRANSFORM FOR THE 4TH, 5TH, ETC.	FOURT	128
	C	DIMENSIONS.	FOURT	129
130	C	2. REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION. METHOD--	FOURT	130
	C	TRANSFORM HALF THE DATA, SUPPLYING THE OTHER HALF BY CON-	FOURT	131
	C	JUGATE SYMMETRY.	FOURT	132
	C	3. REAL TRANSFORM FOR THE 1ST DIMENSION, N ODD. METHOD--	FOURT	133
	C	TRANSFORM HALF THE DATA AT EACH STAGE, SUPPLYING THE OTHER	FOURT	134
135	C	HALF BY CONJUGATE SYMMETRY.	FOURT	135
	C	4. REAL TRANSFORM FOR THE 1ST DIMENSION, N EVEN. METHOD--	FOURT	136
	C	TRANSFORM A COMPLEX ARRAY OF LENGTH N/2 WHOSE REAL PARTS	FOURT	137
	C	ARE THE EVEN NUMBERED REAL VALUES AND WHOSE IMAGINARY PARTS	FOURT	138
	C			
	C	ARE THE ODD NUMBERED REAL VALUES. SEPARATE AND SUPPLY	FOURT	139
140	C	THE SECOND HALF BY CONJUGATE SYMMETRY.	FOURT	140
	C		FOURT	141
	240	NON2 = NP1 * (NP2 / NTW0)	FOURT	142
		ICASE = 1	FOURT	143
		IF (IDIM - 4)250, 300, 300	FOURT	144
145	250	IF (IFOPM)260, 260, 300	FOURT	145
	260	ICASE = 2	FOURT	146
		IF (IDIM - 1)270, 270, 300	FOURT	147
	270	ICASE = 3	FOURT	148
		IF (NTW0 - NP1)300, 300, 280	FOURT	149
150	280	ICASE = 4	FOURT	150
		NTW0 = NTW0 / 2	FOURT	151
		N = N / 2	FOURT	152
		NP2 = NP2 / 2	FOURT	153
		NTOT = NTOT / 2	FOURT	154

155	I = 3	F0URT	155
	DO 290 J = 2, NTOT	F0URT	156
	DATA (J) = DATA (I)	F0URT	157
290	I = I + 2	F0URT	158
300	I1PNG = NP1	F0URT	159
160	IF (ICASE - 2)320, 310, 220	F0URT	160
310	I1PNG = NP0 * (1 + NPPEV / 2)	F0URT	161
		F0URT	162
C	SHUFFLE ON THE FACTORS OF TWO IN N. AS THE SHUFFLING	F0URT	163
C	CAN BE DONE BY SIMPLE INTERCHANGE, NO WORKING AROUND IS NEEDED	F0URT	164
165		F0URT	165
	320 IF (NTWO - NP1)700, 700, 330	F0URT	166
	330 NP2HF = NP2 / 2	F0URT	167
	J = 1	F0URT	168
	DO 390 I2 = 1, NP2, NON2	F0URT	169
170	IF (J - I2)340, 360, 360	F0URT	170
340	I1MAX = I2 + NON2 - 2	F0URT	171
	DO 350 I1 = I2, I1MAX, 2	F0URT	172
	DO 350 I3 = I1, NTOT, NP2	F0URT	173
	J3 = J + I3 - I2	F0URT	174
175	TEMPR = DATA (I3)	F0URT	175
	TEMPI = DATA (I3 + 1)	F0URT	176
	DATA (I3) = DATA (J3)	F0URT	177
	DATA (I3 + 1) = DATA (J3 + 1)	F0URT	178
	DATA (J3) = TEMPR	F0URT	179
180	350 DATA (J3 + 1) = TEMPI	F0URT	180
	360 M = NP2HF	F0URT	181
	370 IF (J - M)390, 390, 380	F0URT	182
	380 J = J - M	F0URT	183
	M = M / 2	F0URT	184
185	IF (M - NON2)390, 370, 370	F0URT	185
	390 J = J + M	F0URT	186
C		F0URT	187
C	MAIN LOOP FOR FACTORS OF TWO. PERFORM FOURIER TRANSFORMS OF	F0URT	188
C	LENGTH FOUR, WITH ONE OF LENGTH TWO IF NEEDED. THE TWIDDLE FACTOR	F0URT	189
190	W=EXP(ISIGN*2*PI*SQRT(-1)*M/(4*MMAX)). CHECK FOR W=ISIGN*SQRT(-1)	F0URT	190
C	AND REPEAT FOR W=ISIGN*SQRT(-1)*CONJUGATE(W).	F0URT	191
C		F0URT	192
	NON2I = NON2 + NON2	F0URT	193
	IPAR = NTWO / NP1	F0URT	194
195	400 IF (IPAR - 2)440, 420, 410	F0URT	195
	410 IPAR = IPAR / 4	F0URT	196
	GO TO 400	F0URT	197
	420 DO 430 I1 = 1, I1PNG, 2	F0URT	198
	DO 430 J3 = I1, NON2, NP1	F0URT	199
200	DO 430 K1 = J3, NTOT, NON2I	F0URT	200
	K2 = K1 + NON2	F0URT	201
	TEMPR = DATA (K2)	F0URT	202
	TEMPI = DATA (K2 + 1)	F0URT	203
	DATA (K2) = DATA (K1) - TEMPR	F0URT	204
205	DATA (K2 + 1) = DATA (K1 + 1) - TEMPI	F0URT	205
	DATA (K1) = DATA (K1) + TEMPR	F0URT	206
	430 DATA (K1 + 1) = DATA (K1 + 1) + TEMPI	F0URT	207
	440 MMAX = NON2	F0URT	208
	450 IF (MMAX - NP2HF)460, 700, 700	F0URT	209
210	460 LMAX = MAX0 (NON2I, MMAX / 2)	F0URT	210
	IF (MMAX - NON2)500, 500, 470	F0URT	211
	470 THETA = - TWOPI * FLOAT (NON2) / FLOAT (4 * MMAX)	F0URT	212
	IF (ISIGN)490, 480, 480	F0URT	213
	480 THETA = - THETA	F0URT	214
215	490 WR = COS (THETA)	F0URT	215
	WI = SIN (THETA)	F0URT	216
	WSTPR = - 2. * WI * WI	F0URT	217
	WSTPI = 2. * WP * WI	F0URT	218
500	DO 690 L = NON2, LMAX, NON2I	F0URT	219
220	M = L	F0URT	220
	IF (MMAX - NON2)520, 520, 510	F0URT	221
510	W2P = WP * WR - WI * WI	F0URT	222
	W2I = 2. * WR * WI	F0URT	223
	W3P = W2P * WP - W2I * WI	F0URT	224
225	W3I = W2P * WI + W2I * WR	F0URT	225
	520 DO 640 I1 = 1, I1PNG, 2	F0URT	226
	DO 640 J3 = I1, NON2, NP1	F0URT	227
	KMIN = J3 + IPAR * M	F0URT	228
	IF (MMAX - NON2)530, 530, 540	F0URT	229
230	530 KMIN = J3	F0URT	230
	540 KDIFF = IPAR * MMAX	F0URT	231

	550	KSTEP = 4 * KDIF		
		DO 620 K1 = KMIN, NNTOT, KSTEP	FOURT	232
235		K2 = K1 + KDIF	FOURT	233
		K3 = K2 + KDIF	FOURT	234
		K4 = K3 + KDIF	FOURT	235
		IF (MMAX - NON2) 560, 560, 590	FOURT	236
	560	U1R = DATA (K1) + DATA (K2)	FOURT	237
		U1I = DATA (K1 + 1) + DATA (K2 + 1)	FOURT	238
240		U2R = DATA (K3) + DATA (K4)	FOURT	239
		U2I = DATA (K3 + 1) + DATA (K4 + 1)	FOURT	240
		U3R = DATA (K1) - DATA (K2)	FOURT	241
		U3I = DATA (K1 + 1) - DATA (K2 + 1)	FOURT	242
245		IF (TSIGN) 570, 580, 590	FOURT	243
	570	U4R = DATA (K3 + 1) - DATA (K4 + 1)	FOURT	244
		U4I = DATA (K4) - DATA (K3)	FOURT	245
		GO TO 620	FOURT	246
	580	U4R = DATA (K4 + 1) - DATA (K3 + 1)	FOURT	247
		U4I = DATA (K3) - DATA (K4)	FOURT	248
250		GO TO 620	FOURT	249
	590	T2R = W2R * DATA (K2) - W2I * DATA (K2 + 1)	FOURT	250
		T2I = W2R * DATA (K2 + 1) + W2I * DATA (K2)	FOURT	251
		T3R = W3R * DATA (K3) - W3I * DATA (K3 + 1)	FOURT	252
		T3I = W3R * DATA (K3 + 1) + W3I * DATA (K3)	FOURT	253
255		T4R = W3R * DATA (K4) - W3I * DATA (K4 + 1)	FOURT	254
		T4I = W3R * DATA (K4 + 1) + W3I * DATA (K4)	FOURT	255
		U1R = DATA (K1) + T2R	FOURT	256
		U1I = DATA (K1 + 1) + T2I	FOURT	257
		U2R = T3R + T4R	FOURT	258
260		U2I = T3I + T4I	FOURT	259
		U3R = DATA (K1) - T2R	FOURT	260
		U3I = DATA (K1 + 1) - T2I	FOURT	261
		IF (TSIGN) 600, 610, 610	FOURT	262
265	600	U4R = T3I - T4I	FOURT	263
		U4I = T4R - T3R	FOURT	264
		GO TO 620	FOURT	265
	610	U4R = T4I - T3I	FOURT	266
		U4I = T3R - T4R	FOURT	267
270	620	DATA (K1) = U1R + U2R	FOURT	268
		DATA (K1 + 1) = U1I + U2I	FOURT	269
		DATA (K2) = U3R + U4R	FOURT	270
		DATA (K2 + 1) = U3I + U4I	FOURT	271
		DATA (K3) = U1R - U2R	FOURT	272
275		DATA (K3 + 1) = U1I - U2I	FOURT	273
		DATA (K4) = U3R - U4R	FOURT	274
	630	DATA (K4 + 1) = U3I - U4I	FOURT	275
		KMIN = 4 * (KMIN - J2) + J2	FOURT	276
		KDIF = KSTEP	FOURT	277
280		IF (KDIF - NP2) 550, 640, 640	FOURT	278
	640	CONTINUE	FOURT	279
		M = MMAX - M	FOURT	280
		IF (TSIGN) 650, 660, 660	FOURT	281
285	650	TEMPR = WR	FOURT	282
		WR = - WI	FOURT	283
		WI = - TEMPR	FOURT	284
		GO TO 670	FOURT	285
	660	TEMPR = WR	FOURT	286
		WR = WI	FOURT	287
		WI = TEMPR	FOURT	288
290	670	IF (M - LMAX) 680, 680, 510	FOURT	289
	680	TEMPR = WR	FOURT	290
		WR = WR * WSTPR - WI * WSTPI + WR	FOURT	291
			FOURT	292
	690	WI = WI * WSTPR + TEMPR * WSTPI + WI	FOURT	293
295		IPAR = 3 - IPAR	FOURT	294
		MMAX = MMAX + MMAX	FOURT	295
		GO TO 450	FOURT	296
	C		FOURT	297
	C	MAIN LOOP FOR FACTORS NOT EQUAL TO TWO. APPLY THE TWIDDLE FACTOR	FOURT	298
	C	W=EXP(TSIGN*2*PI*SQRT(-1)*(J2-1)*(J1-J2)/(NP2*IFP1)), THEN	FOURT	299
300	C	PERFORM A FOURIER TRANSFORM OF LENGTH IFACT(IF), MAKING USE OF	FOURT	300
	C	CONJUGATE SYMMETRIES.	FOURT	301
	C		FOURT	302
	700	IF (NTWO - NP2) 710, 990, 990	FOURT	303
305	710	IFP1 = NON2	FOURT	304
		IF = 1	FOURT	305
		NR1HE = NP1 / 2	FOURT	306
	720	IFP2 = IFP1 / IFACT (IF)	FOURT	307
		J1RNG = NP2	FOURT	308

		IF (ICASE = 3)740, 730, 740	F0URT	309
310	730	J1RNG = (NP2 + IFP1) / 2	F0URT	310
		J2STP = NP2 / IFACT (IF)	F0URT	311
		J1PC2 = (J2STP + IFP2) / 2	F0URT	312
	740	J2MIN = 1 + IFP2	F0URT	313
		IF (IFP1 - NP2)750, 800, 800	F0URT	314
315	750	DO 790 J2 = J2MIN, IFP1, IFP2	F0URT	315
		THETA = - TWOPI * FLOAT (J2 - 1) / FLOAT (NP2)	F0URT	316
		IF (ISIGN)770, 760, 760	F0URT	317
	760	THETA = - THETA	F0URT	318
	770	SINTH = SIN (THETA / 2.)	F0URT	319
320		WSTPR = - 2. * SINTH * SINTH	F0URT	320
		WSTPI = SIN (THETA)	F0URT	321
		WR = WSTPR + 1.	F0URT	322
		WI = WSTPI	F0URT	323
		J1MIN = J2 + IFP1	F0URT	324
325		DO 790 J1 = J1MIN, J1RNG, IFP1	F0URT	325
		I1MAX = J1 + I1RNG - 2	F0URT	326
		DO 780 I1 = J1, I1MAX, 2	F0URT	327
		DO 780 I3 = I1, NTOT, NP2	F0URT	328
		J3MAX = I3 + IFP2 - NP1	F0URT	329
330		DO 790 J3 = I3, J3MAX, NP1	F0URT	330
		TEMPR = DATA (J3)	F0URT	331
		DATA (J3) = DATA (J3) * WR - DATA (J3 + 1) * WI	F0URT	332
	780	DATA (J3 + 1) = TEMPR * WI + DATA (J3 + 1) * WR	F0URT	333
		TEMPR = WR	F0URT	334
335		WR = WR * WSTPR - WI * WSTPI + WR	F0URT	335
	790	WI = TEMPR * WSTPI + WI * WSTPR + WI	F0URT	336
	800	THETA = - TWOPI / FLOAT (IFACT (IF))	F0URT	337
		IF (ISIGN)820, 810, 810	F0URT	338
	810	THETA = - THETA	F0URT	339
340	820	SINTH = SIN (THETA / 2.)	F0URT	340
		WSTPR = - 2. * SINTH * SINTH	F0URT	341
		WSTPI = SIN (THETA)	F0URT	342
		KSTEP = 2 * N / IFACT (IF)	F0URT	343
		KRANG = KSTEP * (IFACT (IF) / 2) + 1	F0URT	344
345		DO 980 I1 = 1, I1RNG, 2	F0URT	345
		DO 980 I3 = I1, NTOT, NP2	F0URT	346
		DO 910 KMIN = 1, KRANG, KSTEP	F0URT	347
		J1MAX = I3 + J1RNG - IFP1	F0URT	348
		DO 880 J1 = I3, J1MAX, IFP1	F0URT	349
350		J3MAX = J1 + IFP2 - NP1	F0URT	350
		DO 880 J2 = J1, J3MAX, NP1	F0URT	351
		J2MAX = J3 + IFP1 - IFP2	F0URT	352
		K = KMIN + (J3 - J1 + (J1 - I3) / IFACT (IF)) / NP1HF	F0URT	353
		IF (KMIN - 1)830, 830, 850	F0URT	354
355	830	SUMP = 0.	F0URT	355
		SUMI = 0.	F0URT	356
		DO 840 J2 = J3, J2MAX, IFP2	F0URT	357
		SUMR = SUMR + DATA (J2)	F0URT	358
	840	SUMI = SUMI + DATA (J2 + 1)	F0URT	359
360		WOPK (K) = SUMR	F0URT	360
		WOPK (K + 1) = SUMI	F0URT	361
		GO TO 980	F0URT	362
	850	KCONJ = K + 2 * (N - KMIN + 1)	F0URT	363
		J2 = J2MAX	F0URT	364
365		SUMP = DATA (J2)	F0URT	365
		SUMI = DATA (J2 + 1)	F0URT	366
		OLDSR = 0.	F0URT	367
		OLDSI = 0.	F0URT	368
		J2 = J2 - IFP2	F0URT	369
370	860	TEMPR = SUMP	F0URT	370
		TEMPI = SUMI	F0URT	371
		SUMR = TWOPI * SUMR - OLDSR + DATA (J2)	F0URT	372
		SUMI = TWOPI * SUMI - OLDSI + DATA (J2 + 1)	F0URT	373
		OLDSR = TEMPR	F0URT	374
375		OLDSI = TEMPI	F0URT	375
		J2 = J2 - IFP2	F0URT	376
		IF (J2 - J3)870, 870, 860	F0URT	377
	870	TEMPR = WR * SUMP - OLDSR + DATA (J2)	F0URT	378
		TEMPI = WI * SUMI	F0URT	379
380		WOPK (K) = TEMPR - TEMPI	F0URT	380
		WOPK (KCONJ) = TEMPR + TEMPI	F0URT	381
		TEMPR = WR * SUMI - OLDSI + DATA (J2 + 1)	F0URT	382
		TEMPI = WI * SUMR	F0URT	383
		WOPK (K + 1) = TEMPR + TEMPI	F0URT	384
385		WOPK (KCONJ + 1) = TEMPR - TEMPI	F0URT	385

	890	CONTINUE	FOURT	386
		IF (KMIN - 1)890, 890, 900	FOURT	387
	890	WP = WSTPR + 1.	FOURT	388
		WI = WSTPI	FOURT	389
390		GO TO 910	FOURT	390
	900	TEMPR = WP	FOURT	391
		WP = WP * WSTPR - WI * WSTPI + WP	FOURT	392
		WI = TEMPR * WSTPI + WI * WSTPR + WI	FOURT	393
	910	TWOPR = WR + WR	FOURT	394
395		IF (ICASE - 3)930, 920, 930	FOURT	395
	920	IF (IFR1 - NR2)950, 930, 930	FOURT	396
	930	K = 1	FOURT	397
		I2MAX = I3 + NP2 - NP1	FOURT	398
		DO 940 I2 = I3, I2MAX, NP1	FOURT	399
400		DATA (I2) = WPRK (K)	FOURT	400
		DATA (I2 + 1) = WPRK (K + 1)	FOURT	401
	940	K = K + 2	FOURT	402
		GO TO 980	FOURT	403
	C		FOURT	404
405	C	COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N ODD, BY CON-	FOURT	405
	C	JUGATE SYMMETRIES AT EACH STAGE.	FOURT	406
	C		FOURT	407
	950	J3MAX = I3 + IFR2 - NP1	FOURT	408
		DO 970 J3 = I3, J3MAX, NP1	FOURT	409
410		J2MAX = J3 + NP2 - J2STR	FOURT	410
		DO 970 J2 = J3, J2MAX, J2STR	FOURT	411
		J1MAX = J2 + J1PG2 - IFR2	FOURT	412
		J1CNJ = J3 + J2MAX + J2STR - J2	FOURT	413
		DO 970 J1 = J2, J1MAX, IFR2	FOURT	414
415		K = 1 + J1 - I3	FOURT	415
		DATA (J1) = WPRK (K)	FOURT	416
		DATA (J1 + 1) = WPRK (K + 1)	FOURT	417
		IF (J1 - J2)970, 970, 960	FOURT	418
	960	DATA (J1CNJ) = WPRK (K)	FOURT	419
420		DATA (J1CNJ + 1) = - WPRK (K + 1)	FOURT	420
	970	J1CNJ = J1CNJ - IFR2	FOURT	421
	980	CONTINUE	FOURT	422
		IF = IF + 1	FOURT	423
		IFR1 = IFR2	FOURT	424
425		IF (IFR1 - NP1)990, 990, 720	FOURT	425
			FOURT	426
	C		FOURT	427
	C	COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N EVEN, BY CON-	FOURT	428
	C	JUGATE SYMMETRIES.	FOURT	429
	C		FOURT	430
430	990	CO TO (1260, 1180, 1260, 1000), ICASE	FOURT	430
	1000	NHALF = N	FOURT	431
		N = N + N	FOURT	432
		THETA = - TWOPR / FLUAT (N)	FOURT	433
		IF (ISIGN)1020, 1010, 1010	FOURT	434
435	1010	THETA = - THETA	FOURT	435
	1020	SINTH = SIN (THETA / 2.)	FOURT	436
		WSTPR = - 2. * SINTH * SINTH	FOURT	437
		WSTPI = SIN (THETA)	FOURT	438
		WR = WSTPR + 1.	FOURT	439
440		WI = WSTPI	FOURT	440
		IMIN = 3	FOURT	441
		JMIN = 2 * NHALF - 1	FOURT	442
		GO TO 1050	FOURT	443
	1030	J = JMIN	FOURT	444
445		DO 1040 I = IMIN, NTOT, NP2	FOURT	445
		SUMP = (DATA (I) + DATA (J)) / 2.	FOURT	446
			FOURT	447
		SUMI = (DATA (I + 1) + DATA (J + 1)) / 2.	FOURT	448
		DIFP = (DATA (I) - DATA (J)) / 2.	FOURT	449
		DIFI = (DATA (I + 1) - DATA (J + 1)) / 2.	FOURT	450
450		TEMPR = WP * SUMI + WI * DIFP	FOURT	451
		TEMPI = WI * SUMI - WP * DIFP	FOURT	452
		DATA (I) = SUMP + TEMPR	FOURT	453
		DATA (I + 1) = DIFI + TEMPI	FOURT	454
		DATA (J) = SUMP - TEMPR	FOURT	455
455		DATA (J + 1) = - DIFI + TEMPI	FOURT	456
	1040	J = J + NP2	FOURT	457
		IMIN = IMIN + 2	FOURT	458
		JMIN = JMIN - 2	FOURT	459
		TEMPR = WR	FOURT	460
460		WP = WR * WSTPR - WI * WSTPI + WP	FOURT	461
		WI = TEMPR * WSTPI + WI * WSTPR + WI	FOURT	462
	1050	IF (IMIN - JMIN)1030, 1060, 1090	FOURT	462

	1060	IF (ISIGN)1070, 1090, 1090	FOUR T	463
	1070	DO 1080 I = IMIN, NTOT, NP2	FOUR T	464
465	1080	DATA (I + 1) = - DATA (I + 1)	FOUR T	465
	1090	NP2 = NP2 + NP2	FOUR T	466
		NTOT = NTOT + NTOT	FOUR T	467
		J = NTOT + 1	FOUR T	468
		IMAX = NTOT / 2 + 1	FOUR T	469
470	1100	IMIN = IMAX - 2 * NHALF	FOUR T	470
		I = IMIN	FOUR T	471
		GO TO 1120	FOUR T	472
	1110	DATA (J) = DATA (I)	FOUR T	473
		DATA (J + 1) = - DATA (I + 1)	FOUR T	474
475	1120	I = I + 2	FOUR T	475
		J = J - 2	FOUR T	476
		IF (I - IMAX)1110, 1120, 1130	FOUR T	477
	1130	DATA (J) = DATA (IMIN) - DATA (IMIN + 1)	FOUR T	478
		DATA (J + 1) = 0.	FOUR T	479
480		IF (I - J)1150, 1170, 1170	FOUR T	480
	1140	DATA (J) = DATA (I)	FOUR T	481
		DATA (J + 1) = DATA (I + 1)	FOUR T	482
	1150	I = I - 2	FOUR T	483
		J = J - 2	FOUR T	484
485		IF (I - IMIN)1160, 1160, 1140	FOUR T	485
	1160	DATA (J) = DATA (IMIN) + DATA (IMIN + 1)	FOUR T	486
		DATA (J + 1) = 0.	FOUR T	487
		IMAX = IMIN	FOUR T	488
		GO TO 1100	FOUR T	489
490	1170	DATA (1) = DATA (1) + DATA (2)	FOUR T	490
		DATA (2) = 0.	FOUR T	491
		GO TO 1260	FOUR T	492
	C		FOUR T	493
	C	COMPLETE A REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION BY	FOUR T	494
495	C	CONJUGATE SYMMETRIES.	FOUR T	495
	C		FOUR T	496
	1180	IF (I1RNG - NP1)1190, 1260, 1260	FOUR T	497
	1190	DO 1250 J3 = 1, NTOT, NP2	FOUR T	498
		I2MAX = I3 + NP2 - NP1	FOUR T	499
500		DO 1250 I2 = I3, I2MAX, NP1	FOUR T	500
		IMIN = I2 + I1RNG	FOUR T	501
		IMAX = I2 + NP1 - 2	FOUR T	502
		IMAX = 2 * I3 + NP1 - IMIN	FOUR T	503
505		IF (I2 - I3)1210, 1210, 1200	FOUR T	504
	1200	JMAX = JMAX + NP2	FOUR T	505
	1210	IF (IMIN - 2)1240, 1240, 1220	FOUR T	506
	1220	J = JMAX + NP0	FOUR T	507
		DO 1230 I = IMIN, IMAX, 2	FOUR T	508
		DATA (I) = DATA (J)	FOUR T	509
510		DATA (I + 1) = - DATA (J + 1)	FOUR T	510
	1230	J = J - 2	FOUR T	511
	1240	J = JMAX	FOUR T	512
		DO 1250 I = IMIN, IMAX, NP0	FOUR T	513
		DATA (I) = DATA (J)	FOUR T	514
515		DATA (I + 1) = - DATA (J + 1)	FOUR T	515
	1250	J = J - NP0	FOUR T	516
	C		FOUR T	517
	C	END OF LOOP ON EACH DIMENSION	FOUR T	518
	C		FOUR T	519
520	1260	NP0 = NP1	FOUR T	520
		NP1 = NP2	FOUR T	521
	1270	NPREF = N	FOUR T	522
	1280	RETURN	FOUR T	523
		FND	FOUR T	524



A.1.11 SUBROUTINE  
PARAB(FOD,DOL,BLOCK,DFOCUS,ACOSE,ACOSH,THETA,ETHETA,EPHI)

PURPOSE:

This subroutine calculates the E- and H-plane far electric field for an axially defocused, circularly symmetric, paraboloidal reflector antenna at a specified angle from the axis.

ARGUMENTS:

FOD is the focal length to diameter ratio for the reflector.

DOL is the diameter of reflector in wavelengths.

BLOCK is the fractional diameter blockage.

DFOCUS is the amount of axial defocusing in wavelengths (positive direction corresponds to feed beyond focal point).

ACOSE is the E-plane aperture illumination factor.

ACOSH is the H-plane aperture illumination factor. (NOTE: See discussion of POMODL for a more complete discussion of ACOSE and ACOSH.)

THETA is the angle from axis at which field values are desired in degrees.

ETHETA is the electric field in E-plane.

EPHI is the electric field in H-plane.

DISCUSSION:

This and associated subroutines EPINT, ETINT, QATRC, and BESFUN were written by Professor W. V. T. Rusch of the University of Southern California. This discussion is intended to indicate the computations performed and is not a detailed description of the operation of the subroutines.

The subroutine uses P0 as discussed in section 3 of the report. It is assumed that the antenna is rotationally symmetric, thus allowing very rapid execution.

Aperture illumination may be of three types: uniform, dipole, or  $\cos^p\theta'$ , where  $\theta'$  is the angle from the axis of the feed. These are selected with parameters ACOSE and ACOSH, and the E- and H-plane tapers are independently specified.

The integration is performed by subroutine QATRC. This subroutine has error flags which are set when the desired accuracy is not achieved either because of accumulated round-off errors or because the integration range could not be sufficiently subdivided. PARAB prints an error message indicating the type of error. These errors occur at larger values of THETA. Care should be taken to delete any far-field points known to be in error.

This subroutine requires that functions ETINT, EPINT, and subroutines QATRC and BESFUN be supplied. In addition, library functions ATAN, COS, SIN, ATAN2, CEXP, SQRT, CABS, and inline functions CMPLX and ABS are employed.

1	SUBROUTINE PAPAB(FOD,DOL,BLOCK,DFOCUS,ACDSE,ACCSH,THETA,ETHETA,EPHPARAB	1
	*I)	PARAB 2
C	RADIATION PATTERNS FROM A DEFOCUSED PARABOLOID	PARAB 3
C	PROGRAMMER - W.V.T. RUSCH	PARAB 4
5	16 MAJ 1974	PARAB 5
C	MODIFIED 12 MAY 1976	PARAB 6
	COMPLEX AUX(11),ROMB,CMLX,A1,D1,ETHETA,EPHI	PARAB 7
	COMMON/DATA/FOL,PI,SINT,COST,DFOCSS,ACDSEE,ACCSHH	PARAB 8
	EXTERNAL FTINT,EPINT	PARAB 9
10	DFOCSS=DFOCUS	PARAB 10
	ACDSEE=ACDSE	PARAB 11
	ACCSHH=ACCSH	PARAB 12
	PI=4.0*ATAN(1.0)	PARAB 13
	DTR=PI/180.0	PARAB 14
15	PTD=180.0/PI	PARAB 15
	FOL = FOD*DOL	PARAB 16
	A = 2.0*ATAN(4.0*FOD)	PARAB 17
	IF(BLOCK.LT.0.0001) B = PI-	PARAB 18
20	IF(BLOCK.GE.C.CC01) B = 2.0*ATAN(4.0*FOD/BLOCK)	PARAB 19
	COST = COS(THETA*DTR)	PARAB 20
	SINT = SIN(THETA*DTR)	PARAB 21
	CALL QATPC(A,B,1.0E-02,11,FTINT,ROMB,IER,AUX)	PARAB 22
	IF (IER .EQ. 1) PPINT 1000	PARAB 23
	IF (IFR .EQ. 2) PPINT 1010	PARAB 24
25	ETHETA = CMLX(0.0,1.0)*2.0*PI*FOL*ROMB	PARAB 25
	CALL QATRC(A,B,1.0E-03,11,EPINT,ROMB,IER,AUX)	PARAB 26
	IF (IFR .EQ. 2) PRINT 2010	PARAB 27
	IF (IER .EQ. 1) PPINT 2000	PARAB 28
30	EPHI = CMLX(0.0,1.0)*2.0*PI*FOL*ROMB	PARAB 29
	RETURN	PARAB 30
		PARAB 31
C	1000 FORMAT(* REQUIRED ACCURACY NOT ACHIEVED IN E-PLANE DUE TO ROUNDIN	PARAB 32
	16 FPRDPS.*)	PARAB 33
35	1010 FORMAT(* REQUIRED ACCURACY NOT ACHIEVED IN E-PLANE DUE TO INSUFFI	PARAB 34
	1CIENT NUMBER OF INTEGRATION STEPS.*)	PARAB 35
	2000 FORMAT(* REQUIRED ACCURACY NOT ACHIEVED IN H-PLANE DUE TO ROUNDIN	PARAB 36
	16 FPRDPS.*)	PARAB 37
	2010 FORMAT(* REQUIRED ACCURACY NOT ACHIEVED IN H-PLANE DUE TO INSUFFI	PARAB 38
40	1CIENT NUMBER OF INTEGRATION STEPS.*)	PARAB 39
	END	PARAB 40

```

1      SUBROUTINE QATRC(XL,XU,FPS,NDIM,FCT,Y,IFR,AUX)          QATRC      1
C      ..... QATRC      2
C      SUBROUTINE QATRC          QATRC      3
C      COMPLEX VERSION OF SSP-ROUTINE QATP, SEPT.72, HS-J    QATRC      4
5      C      PURPOSE          QATRC      5
C      TO COMPUTE AN APPROXIMATION FOR INTEGRAL OF COMPLEX    QATRC      6
C      FUNCTION FCT(X) WITH REAL BOUNDARIES XL AND XU.        QATRC      7
C      QATRC      8
C      QATRC      9
10     C      USAGE          QATRC     10
C      CALL QATRC(XL,XU,FPS,NDIM,FCT,Y,IFR,AUX)             QATRC     11
C      PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.        QATRC     12
C      QATRC     13
C      DESCRIPTION OF PARAMETERS          QATRC     14
C      XL - THE LOWER BOUND OF THE INTERVAL.                 QATRC     15
C      XU - THE UPPER BOUND OF THE INTERVAL.                 QATRC     16
C      FPS - THE UPPER BOUND OF THE ABSOLUTE ERROR.          QATRC     17
C      NDIM - THE DIMENSION OF THE AUXILIARY STORAGE ARRAY  QATRC     18
C              AUX. NDIM-1 IS THE MAXIMAL NUMBER OF BISECTIONS OF
C              THE INTERVAL (XL,XU).                          QATRC     19
C      FCT - THE NAME OF THE EXTERNAL FUNCTION SUBPROGRAM    QATRC     20
C              USED.                                          QATRC     21
C      Y - THE RESULTING APPROXIMATION FOR THE INTEGRAL VALUE. QATRC     22
C      IFR - A RESULTING ERROR PARAMETER.                    QATRC     23
C      AUX - AN AUXILIARY STORAGE ARRAY WITH DIMENSION NDIM. QATRC     24
25     C      QATRC     25
C      REMARKS          QATRC     26
C      ERROR PARAMETER IFR IS CODED IN THE FOLLOWING FORM    QATRC     27
C      IFR=0 - IT WAS POSSIBLE TO REACH THE REQUIRED ACCURACY. QATRC     28
C              NO ERROR.                                     QATRC     29
30     C      IFR=1 - IT IS IMPOSSIBLE TO REACH THE REQUIRED ACCURACY
C              BECAUSE OF ROUNDING ERRORS.                   QATRC     30
C      IFR=2 - IT WAS IMPOSSIBLE TO CHECK ACCURACY BECAUSE NDIM
C              IS LESS THAN 5, OR THE REQUIRED ACCURACY COULD NOT
C              BE REACHED WITHIN NDIM-1 STEPS. NDIM SHOULD BE
35     C              INCREASED.                               QATRC     31
C      QATRC     32
C      QATRC     33
C      QATRC     34
C      QATRC     35
C      QATRC     36
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED          QATRC     37
C      THE EXTERNAL FUNCTION SUBPROGRAM FCT(X) MUST BE CODED BY
C      THE USER. ITS ARGUMENT X SHOULD NOT BE DESTROYED.    QATRC     38
40     C      QATRC     39
C      QATRC     40
C      METHOD          QATRC     41
C      EVALUATION OF Y IS DONE BY MEANS OF TRAPEZOIDAL RULE IN
C      CONNECTION WITH ROMBERG'S PRINCIPLE. ON RETURN Y CONTAINS
C      THE BEST POSSIBLE APPROXIMATION OF THE INTEGRAL VALUE AND
45     C      VECTOR AUX THE UPWARD DIAGONAL OF ROMBERG SCHEME. QATRC     42
C      COMPONENTS AUX(I) (I=1,2,...,IFND, WITH IEND LESS THAN OR
C      EQUAL TO NDIM) BECOME APPROXIMATIONS TO INTEGRAL VALUE WITH
C      DECREASING ACCURACY BY MULTIPLICATION WITH (XU-XL).   QATRC     43
C      FOR REFERENCE, SEE                                     QATRC     44
C      (1) FILIPPI, DAS VERFAHREN VON ROMBERG-STIEEEL-BAUER ALS
C          SPEZIALFALL DES ALLGEMEINEN PRINZIPI VON RICHARDSON,
C          MATHEMATIK-TECHNIK-WIRTSCHAFT, VOL.11, ISS.2 (1964),
C          PP.49-54.                                          QATRC     45
C      (2) BAUFER, ALGORITHM 60, CACM, VOL.4, ISS.6 (1961), PP.255.
50     C          QATRC     46
C      QATRC     47
C      QATRC     48
C      QATRC     49
C      QATRC     50
C      QATRC     51
C      QATRC     52
C      QATRC     53
55     C      QATRC     54
C      QATRC     55
C      ..... QATRC     56
C      QATRC     57
C      QATRC     58
C      QATRC     59
60     C      COMPLEX FCT,Y,SM,AUX(NDIM)          QATRC     60
C      QATRC     61

C      PREPARATIONS OF ROMBERG-LOOP          QATRC     62
3UY(1)=.5*(FCT(XL)+FCT(XU))          QATRC     63
H=XU-XL          QATRC     64
65     C      IF(NDIM-1)8,P,1          QATRC     65
1 IF(4)?,10,2          QATRC     66
C      QATRC     67
C      NDIM IS GREATER THAN 1 AND H IS NOT EQUAL TO 0.      QATRC     68
2 HH=H          QATRC     69
70     C      E=FPS/ABS(H)          QATRC     70
C      DFLT2=0.          QATRC     71
C      P=1.          QATRC     72
C      JJ=1          QATRC     73
C      DO 7 I=2,NDIM          QATRC     74
75     C      Y=AUX(1)          QATRC     75
C      DFLT1=DFLT2          QATRC     76
C      GO=HH          QATRC     77

```

	HH=.5*HH	QATRC	78
	P=.5*P	QATRC	79
80	X=XL+HH	QATRC	80
	SM=(0.,0.)	QATRC	81
	DO 3 J=1, JJ	QATRC	82
	SM=SM+F(T(X))	QATRC	83
	3 X=X+HD	QATRC	84
85	AUX(I)=.5*AUX(I-1)+P*SM	QATRC	85
	C A NEW APPROXIMATION OF INTEGRAL VALUE IS COMPUTED BY MEANS OF	QATRC	86
	C TRAPEZOIDAL RULE.	QATRC	87
	C	QATRC	88
	C START OF ROMBERG'S EXTRAPOLATION METHOD.	QATRC	89
90	Q=1.	QATRC	90
	JJ=I-1	QATRC	91
	DO 4 J=1, JJ	QATRC	92
	II=I-J	QATRC	93
	Q=Q+Q	QATRC	94
95	Q=Q+Q	QATRC	95
	4 AUX(II)=AUX(II+1)+(AUX(II+1)-AUX(II))/(Q-1.)	QATRC	96
	C END OF ROMBERG-STEP	QATRC	97
	C	QATRC	98
	DELTA2=CABS(Y-AUX(1))	QATRC	99
100	IF(T-5)7,5,5	QATRC	100
	5 IF(DELTA2-E)10,10,6	QATRC	101
	6 IF(DELTA2-DELTA1)7,11,11	QATRC	102
	7 JJ=JJ+JJ	QATRC	103
	P TFR=2	QATRC	104
105	9 Y=H*AUX(1)	QATRC	105
	RETURN	QATRC	106
	10 IFR=0	QATRC	107
	GO TO 9	QATRC	108
	11 IFR=1	QATRC	109
110	Y=H*Y	QATRC	110
	RETURN	QATRC	111
	END	QATRC	112

1		COMPLEX FUNCTION ETINT(X)	ETINT	1
		COMMON/DATA/FCL,PI,SINT,COST,DFOCUS,ACOSF,ACOSH	ETINT	2
		DIMENSION RJ(1000)	ETINT	3
C		NOTE THAT COS(PI-XP) = (DFOCUS-RHO*COSX)/RHOPRIME = R2/RHPOVL	ETINT	4
5		COMPLEX CEXP,CPLX,A1,D1,HX	ETINT	5
		SINX = SIN(X)	ETINT	6
		COSX = COS(X)	ETINT	7
		RHPOVL = 2.0*FOL/(1.0-COSX)	ETINT	8
		RHPOVL = SQRT(RHPOVL*RHPOVL+DFOCUS*DFOCUS-2.*DFOCUS*RHPOVL*COSX)	ETINT	9
10		R1 = RHPOVL*SINX	ETINT	10
		R2 = DFOCUS - RHPOVL*COSX	ETINT	11
		XP = PI - ATAN2(R1,R2)	ETINT	12
		CSPMXP = R2/RHPOVL	ETINT	13
		FAZF = 2.0*PI*(RHPOVL*COSX*COST-RHPOVL)	ETINT	14
15		BETA = 2.0*PI*RHPOVL*SINX*SINT	ETINT	15
		IF(BETA.GT.0.0) GO TO 2	ETINT	16
		BESS0 = 1.0	ETINT	17
		BESS1 = 0.0	ETINT	18
		BESS2 = 0.0	ETINT	19
20		GO TO 3	ETINT	20
	2	CALL RESFUN(BETA,BJ,4)	ETINT	21
		BESS0 = BJ(1)	ETINT	22
		BESS1 = BJ(2)	ETINT	23
		BESS2 = BJ(3)	ETINT	24
25	3	CONTINUE	ETINT	25
		IF(ACOSF.GF.(-100.0))GO TO 20	ETINT	26
		A1 = 2.0/(1.0+CSPMXP)	ETINT	27
		D1 = -A1	ETINT	28
		GO TO 50	ETINT	29
30	20	IF(ACOSF.GF.0.0)GO TO 40	ETINT	30
		A1 = CSPMXP	ETINT	31
		D1 = -1.0	ETINT	32
		GO TO 50	ETINT	33
	40	A1 = CSPMXP**ACOSF	ETINT	34
35		D1 = -CSPMXP**ACOSH	ETINT	35
	50	CONTINUE	ETINT	36
		HX = (RHPOVL/RHPOVL)*(A1*COST*(BESS0-BESS2)-D1*COST*(BESS0+BESS2)*	ETINT	37
		*SIN(XP-(X/2.0))/SIN(X/2.0)-2.0*CPLX(0.0,1.0)*A1*BESS1*COS(X/2.0)/	ETINT	38
		*SIN(X/2.0)*SINT)	ETINT	39
40		HX = HX/(1.0-COSX)	ETINT	40
		ETINT = HX*SINX*CEXP(CPLX(0.0,FAZE))	ETINT	41
		RETURN	ETINT	42
		END	ETINT	43

1		COMPLX FUNCTION EPINT(X)	EPINT	1
		COMMON/DATA/FCL,PI,SINT,COST,DFOCUS,ACOSF,ACOSH	EPINT	2
		DIMENSION BJ(1000)	EPINT	3
	C	NOTE THAT COS(PI-XP) = (DFOCUS-RHO*CO SX)/RHOPPIF = R2/RHPOVL	EPINT	4
5		COMPLEX CEXP,CMPLX,A1,D1,HX	EPINT	5
		SINX = SIN(X)	EPINT	6
		COSX = COS(X)	EPINT	7
		RHOQVL = 2.0*FOL/(1.0-COSX)	EPINT	8
		RHPOVL = SQRT(RHOQVL*RHOQVL+DFOCUS*DFOCUS-2.*DFOCUS*RHOQVL*CO SX)	EPINT	9
10		R1 = RHOQVL*SINX	EPINT	10
		R2 = DFOCUS - RHQOVL*CO SX	EPINT	11
		XP = PI - ATAN2(R1,R2)	EPINT	12
		CSPMXP = R2/RHPOVL	EPINT	13
		FAZE = 2.0*PI*(RHQOVL*CO SX*CO ST-RHPOVL)	EPINT	14
15		BETA = 2.0*PI*RHOQVL*SINX*SINT	EPINT	15
		IF(BETA.GT.0.0) GO TO 2	EPINT	16
		BESS0 = 1.0	EPINT	17
		BESS1 = 0.0	EPINT	18
		BESS2 = 0.0	EPINT	19
20		GO TO 3	EPINT	20
	2	CALL BESFUN(BETA,RJ,4)	EPINT	21
		BESS0 = RJ(1)	EPINT	22
		BESS1 = RJ(2)	EPINT	23
		BESS2 = RJ(3)	EPINT	24
25	3	CONTINUE	EPINT	25
		IF(ACOSF.GE.(-100.0))GO TO 20	EPINT	26
		A1 = 2.0/(1.0+CSPMXP)	EPINT	27
		D1 = -A1	EPINT	28
		GO TO 50	EPINT	29
30	20	IE(ACOSF.GE.0.0)GO TO 40	EPINT	30
		A1 = CSPMXP	EPINT	31
		D1 = -1.0	EPINT	32
		GO TO 50	EPINT	33
	40	A1 = CSPMXP**ACOSF	EPINT	34
35		D1 = -CSPMXP**ACOSH	EPINT	35
	50	CONTINUE	EPINT	36
		HX = (RHOQVL/RHPOVL)*(A1*(BESS0+BESS2)-D1*(BESS0-BESS2)*SIN(XP-(X/	EPINT	37
		*2.0))/SIN(X/2.))	EPINT	38
		HX = HX/(1.0-COSX)	EPINT	39
40		EPINT = HX*SINX*CEXP(CMPLX(0.0,FAZE))	EPINT	40
		RETURN	EPINT	41
		END	EPINT	42

1	SUBROUTINE RESFUN(X,RJ,NMAX)	BESFUN	1
	DIMENSION RJ(1)	BESFUN	2
C	NOTE RJ(1)=J0 RJ(?)=J1 ... RJ(200)=J199	BESFUN	3
10	IF(X.GT.1.0F-03) GO TO 18	BESFUN	4
5	RJ(1)=1.0	BESFUN	5
	DO 15 JRJ=2,NMAX	BESFUN	6
15	RJ(JRJ)=0.0	BESFUN	7
	RETURN	BESFUN	8
	IF(X.GT.NMAX) M=2*X+7	BESFUN	9
10	IF(X.LT.NMAX) M=2*NMAX+7	BESFUN	10
	IF(M.LT.990) GO TO 19	BESFUN	11
	WRITE(6,2000)	BESFUN	12
2000	FORMAT(10X,33H M EXCEEDS 190. EXECUTION ABORTED)	BESFUN	13
	STOP	BESFUN	14
15	FM1=10.F-2R	BESFUN	15
	FM=0.0	BESFUN	16
	ALPHA=0.	BESFUN	17
	IF(M-(M/2)*2) 20,30,2C	BESFUN	18
20	JT=1	BESFUN	19
	GO TO 40	BESFUN	20
20	JT=-1	BESFUN	21
40	M2=M-2	BESFUN	22
	DO 160 K=1,M2	BESFUN	23
	MK=M-K	BESFUN	24
25	XMK=MK	BESFUN	25
	BMK=2.*XMK*FM1/X-FM	BESFUN	26
	FM=FM1	BESFUN	27
	FM1=BMK	BESFUN	28
	BJ(MK)=BMK	BESFUN	29
30	JT=-JT	BESFUN	30
	S=1+JT	BESFUN	31
	ALPHA=ALPHA+BMK*S	BESFUN	32
160	CONTINUE	BESFUN	33
	BMK=2.0*FM1/X-FM	BESFUN	34
35	BJ(1)=BMK	BESFUN	35
	ALPHA=ALPHA+BMK	BESFUN	36
	DO 200 IN=1,NMAX	BESFUN	37
	RJ(IN)=BJ(IN)/ALPHA	BESFUN	38
200	CONTINUE	BESFUN	39
40	RETURN	BESFUN	40
	END	BESFUN	41



A.1.12 SUBROUTINE  
PLT120R(X,Y,XMAX,XMIN,YMAX,YMIN,LAST,ISYMBOL,NO,MOST)

PURPOSE:

To make a page plot of array Y versus array X.

ARGUMENTS:

X = Array containing abscissa values of the function to be plotted.  
Y = Array containing ordinate values of the function to be plotted.  
XMIN = Minimum abscissa value.  
XMAX = Maximum abscissa value.  
YMIN = Minimum ordinate value.  
YMAX = Maximum ordinate value.  
LAST = Number of points to be plotted.  
ISYMBOL = A Hollerith variable containing the plotting symbol, e.g., to plot with the symbol "X" ISYMBOL = 1HX.  
NO = Number of plot on page.  
MOST = Total number of plots to be made on one page.

DISCUSSION:

This subroutine produces a "quick and dirty" plot of Y versus X on the page printer. The size of the plotting area is 50 x 120 units. Multiple plots may be made on a single page. A page eject is performed before the first plot of a series is begun, but no eject is performed after completion of a series. This allows a title to be printed at the bottom of the plot. The subroutine uses inline function FLOAT.

1	SUBROUTINE PLT120R(X, Y, XMAX, XMIN, YMAX, YMIN, LAST, ISYMBOL, NOPLT120R	1
	1. MOST)	2
	MODIFIED 11/4/68	3
	10 DIMENSION Y(1), Y(1), ZX(13), GRAPH(121, 51)	4
5	INTEGER GRAPH, COLUMNS, BLANK, BORDER	5
	DATA (LINES = 51), (COLUMNS = 121)	6
	KMAX = COLUMNS / 10 + 1	7
	IF (NO .NE. 1) GO TO 190	8
	YLAP = YMAX	9
10	YSMA = YMIN	10
	XLAP = YMAX	11
	XSMA = YMIN	12
	BORDER = 1H1	13
	BLANK = 1H	14
15	MATRIX = COLUMNS * LINES	15
	IF (MATRIX .LT. 1) GO TO 120	16
	DO 100 I = 1, MATRIX	17
100	GRAPH(I) = BLANK	18
120	CONTINUE	19
20	IF (LINES .LT. 1) GO TO 140	20
	DO 130 I = 1, LINES	21
130	GRAPH(I, I) = GRAPH(COLUMNS, I) = BORDER	22
140	CONTINUE	23
	IF (COLUMNS .LT. 1) GO TO 160	24
25	DO 150 I = 1, COLUMNS	25
150	GRAPH(I, 26) = 1H.	26
160	CONTINUE	27
	XSCALE = (XLAP - XSMA) / (COLUMNS - 1.)	28
	YSCALE = (YLAP - YSMA) / (LINES - 1.)	29
30	IF (KMAX .LT. 1) GO TO 180	30
	DO 170 K = 1, KMAX	31
170	ZX(K) = 10. * FLOAT(K - 1) * YSCALE + XSMA	32
180	CONTINUE	33
190	IF (LAST .LT. 1) GO TO 250	34
25	DO 240 I = 1, LAST	35
	IF (Y(I) .GT. XLAP .OR. X(I) .LT. XSMA) GO TO 240	36
	IF (Y(I) .GT. YLAP .OR. Y(I) .LT. YSMA) GO TO 240	37
	IX = (X(I) - XSMA) / XSCALE + 1.5	38
	IY = (Y(I) - YSMA) / YSCALE + .5	39
40	IX = LINES - IY	40
	GRAPH(IY, IX) = ISYMBOL	41
240	CONTINUE	42
250	CONTINUE	43
	IF (NO .NE. MOST) RETURN	44
45	PRINT 1500	45
	YES = YLAP + YSCALE	46
	IF (LINES .LT. 1) GO TO 270	47
	DO 260 I = 1, LINES	48
	YES = YES - YSCALE	49
50	PRINT 1510, YES, (GRAPH(J, I), J = 1, COLUMNS)	50
	260 CONTINUE	51
	270 CONTINUE	52
	PRINT 1520	53
	PRINT 1530, ZX	54
55	RETURN	55
	1500 FORMAT (1H,9X,24(5H1....)1H)	56
	1510 FORMAT (1H,9X,2,1X,121A1)	57
	1520 FORMAT (1H,9X,24(5H1....)1H)	58
	1530 FORMAT (1H,2X,13(1X,F9.3))	59
60	END	60



RADIATION PATTERN OF AN AXIALLY DEFOCUS PARABOLOID

SIMULATION TEST NO. 1  
REFLECTOR PARAMETERS -

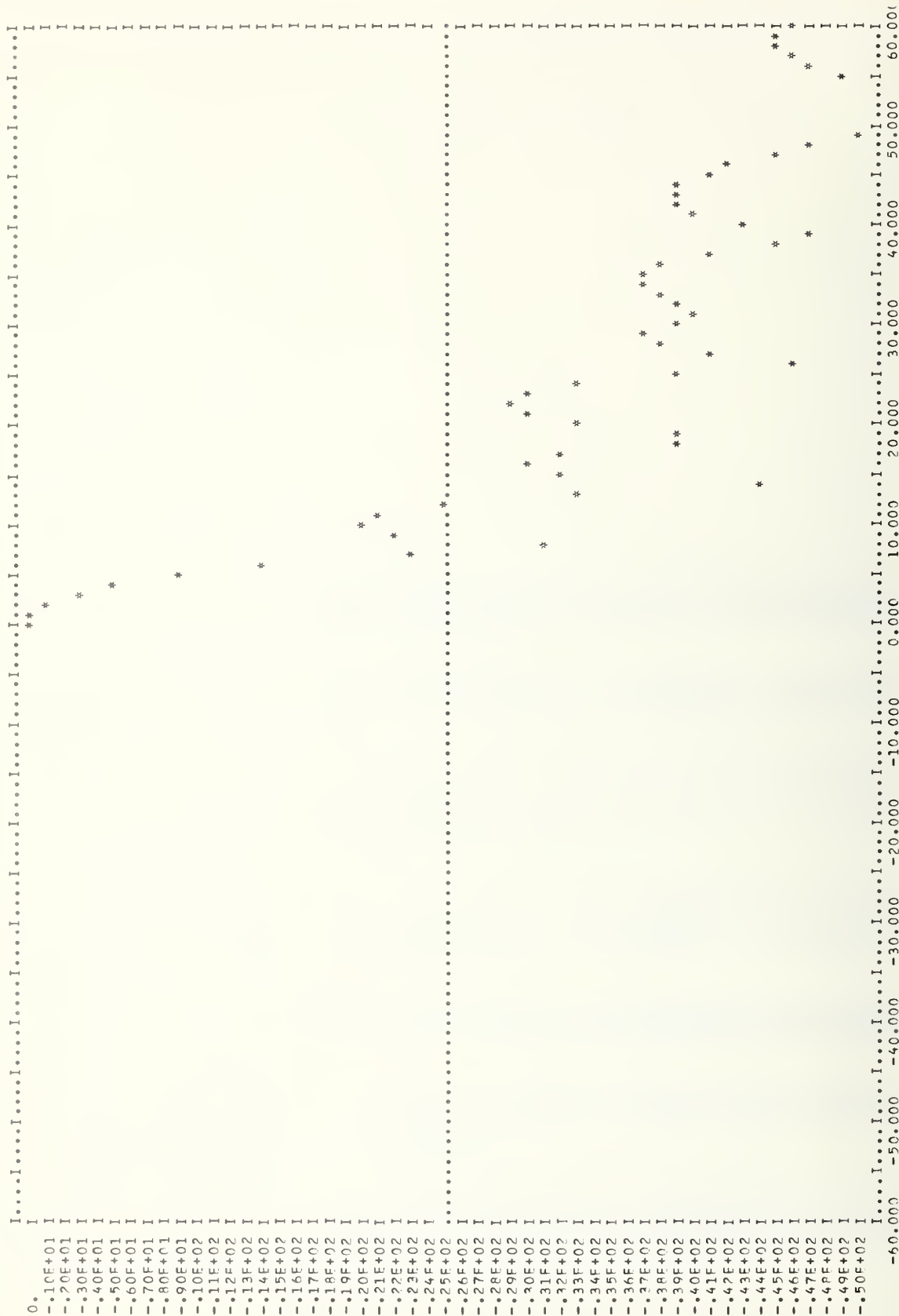
F/D = .500  
DIAMETER = 10.00 WAVELENGTHS  
FRACTIONAL DIAMETER BLOCKING = .100  
AXIAL DEFOCUSING = 0.000 WAVELENGTHS BEYOND FOCUS  
FREQUENCY = 4.0000 GHZ.

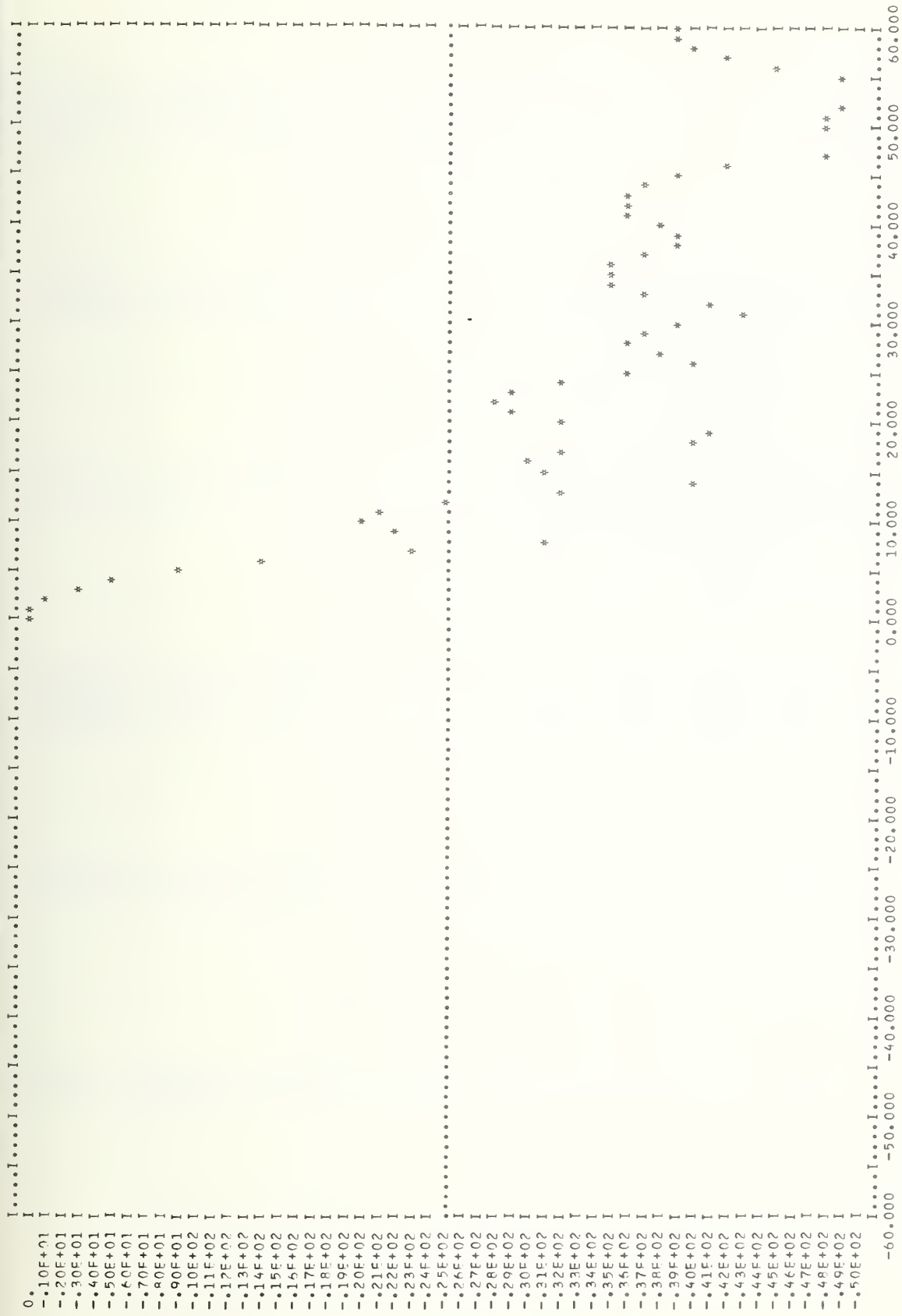
FEED F-PLANE PATTERN = (COS(Y))\*\*( 1.00)  
H-PLANE PATTERN = -(COS(Y))\*\*( 1.00)

ASSUMED EFFICIENCY = 100.00 PERCENT  
NOMINAL GAIN = 29.94 DB  
POWER INPUT = 1.00 WATTS

THETA (DEG)	F-PLANE		H-PLANE		PHASE (DEG)
	MAG (VOLTS)	MAG (DB)	MAG (VOLTS)	MAG (DB)	
0.00	10.956	0.00	10.956	0.00	-90.00
1.00	10.591	-.29	10.593	-.29	-90.24
2.00	9.550	-1.19	9.556	-1.19	-90.98
3.00	7.975	-2.76	7.986	-2.75	-92.23
4.00	6.078	-5.12	6.092	-5.10	-94.04
5.00	4.100	-8.54	4.114	-8.51	-96.51
6.00	2.269	-13.68	2.280	-13.64	-100.03
7.00	.768	-23.09	.773	-23.03	-107.16
8.00	.300	-69.95	.315	-30.83	90.05
9.00	.896	-21.75	.914	-21.57	75.78
10.00	1.075	-20.17	1.097	-19.99	69.54
11.00	.939	-21.34	.962	-21.13	63.24
12.00	.619	-24.95	.638	-24.70	55.32
13.00	.247	-32.95	.262	-32.44	39.85
14.00	.073	-43.54	.105	-40.34	-88.29
15.00	.273	-32.08	.292	-31.47	-124.01
16.00	.330	-30.42	.348	-29.95	-134.64
17.00	.264	-32.37	.273	-32.07	-142.14
18.00	.128	-38.67	.109	-40.01	-143.42
19.00	.119	-39.25	.093	-41.39	1.80
20.00	.260	-32.51	.268	-32.23	1.69
21.00	.357	-29.74	.383	-29.13	-6.86
22.00	.382	-29.15	.418	-28.36	-17.62
23.00	.338	-30.23	.377	-29.26	-30.63
24.00	.243	-33.08	.283	-31.76	-47.98
25.00	.129	-39.55	.173	-36.02	-76.70
26.00	.054	-46.10	.114	-39.66	-133.87
27.00	.099	-40.88	.142	-37.77	173.25
28.00	.140	-37.84	.171	-36.12	147.69
29.00	.147	-37.44	.164	-36.51	133.76
30.00	.129	-38.57	.123	-39.02	128.84
31.00	.112	-39.81	.078	-42.99	150.84
32.00	.120	-39.22	.093	-41.40	-167.66
33.00	.145	-37.59	.147	-37.45	-160.09
34.00	.162	-36.59	.188	-35.31	-168.34
35.00	.161	-36.68	.202	-34.68	177.07
36.00	.138	-38.01	.189	-35.29	157.50
37.00	.100	-40.82	.157	-36.90	131.08
38.00	.059	-45.35	.125	-38.86	93.40

39.00	.049	-46.98	37.98	.119	-39.30	45.86
40.00	.077	-43.02	-9.10	.141	-37.83	.4.54
41.00	.106	-40.26	-33.23	.167	-36.32	-25.82
42.00	.123	-38.96	-50.38	.182	-35.58	-48.94
43.00	.127	-38.72	-44.50	.179	-35.71	-70.97
44.00	.119	-39.30	-76.54	.160	-36.74	-90.25
45.00	.103	-40.58	-86.42	.126	-38.76	-108.15
46.00	.083	-42.44	-93.55	.086	-42.09	-124.03
47.00	.063	-44.75	-97.06	.046	-47.59	-133.89
48.00	.048	-47.23	-95.78	.016	-54.54	-107.43
49.00	.036	-49.61	-91.67	.027	-52.11	-46.16
50.00	.027	-52.03	-86.04	.042	-48.39	-53.14
51.00	.019	-55.07	-76.78	.045	-47.65	-71.52
52.00	.013	-58.51	-49.93	.038	-49.09	-96.33
53.00	.016	-56.93	-8.03	.024	-53.02	-135.54
54.00	.026	-52.57	4.36	.019	-55.33	138.78
55.00	.037	-49.40	.32	.038	-49.23	75.88
56.00	.047	-47.34	-9.79	.063	-44.79	42.85
57.00	.054	-46.09	-22.69	.087	-42.02	16.31
58.00	.058	-45.47	-37.08	.106	-40.27	-8.44
59.00	.059	-45.37	-52.29	.120	-38.23	-32.77
60.00	.056	-45.87	-67.52	.127	-38.73	-57.29





H-PLANE

NEAR-FIELD PARAMETERS-

X-SPACING = 3.75 CM 64 POINTS  
 Y-SPACING = 3.75 CM 64 POINTS  
 DISTANCE FROM REFLECTOR FOCAL POINT = 0.00 CM AWAY FROM REFLECTOR SURFACE

CENTEPLINE DATA

X-Z PLANE		Y-Z PLANE		PHASE
X	AMP	Y	AMP	PHASE
-1.1625	3119	-1.1625	1165	311.0897
-1.1250	.2586	-1.1250	.1691	251.0452
-1.0875	.2700	-1.0875	.0678	290.6290
-1.0500	.1860	-1.0500	.1434	288.1975
-1.0125	.1017	-1.0125	.1306	246.4036
-.9750	.1005	-.9750	.1648	333.6881
-.9375	.0323	-.9375	.2296	251.3149
-.9000	.0915	-.9000	.1018	355.7389
-.8625	.0541	-.8625	.2632	258.5681
-.8250	.1004	-.8250	.2007	203.5065
-.7875	.0474	-.7875	.1884	299.5921
-.7500	.0574	-.7500	.3507	229.5261
-.7125	.1690	-.7125	.1132	64.1319
-.6750	.0839	-.6750	.3866	279.2087
-.6375	.1915	-.6375	.4112	190.2763
-.6000	.2915	-.6000	.2265	70.3454
-.5625	.2259	-.5625	.5183	310.5921
-.5250	.4015	-.5250	.8110	223.4786
-.4875	.7639	-.4875	1.0235	137.3943
-.4500	1.1069	-.4500	1.4227	61.1873
-.4125	1.3686	-.4125	1.4552	13.3092
-.3750	2.3373	-.3750	2.3869	8.5232
-.3375	4.1240	-.3375	4.3982	342.1280
-.3000	5.5214	-.3000	5.4712	331.9989
-.2625	6.8479	-.2625	6.6036	348.2612
-.2250	8.4890	-.2250	8.8401	353.8764
-.1875	7.8894	-.1875	7.9203	346.1406
-.1500	7.5194	-.1500	7.4792	333.8087
-.1125	8.1906	-.1125	8.2470	348.1900
-.0750	7.8913	-.0750	7.7389	355.5398
-.0375	9.1533	-.0375	9.0119	352.0960
0.0000	11.1600	0.0000	11.1600	355.6275
.0375	9.1533	.0375	9.0119	352.0960
.0750	7.8913	.0750	7.7389	355.5398
.1125	8.1906	.1125	8.2470	348.1900
.1500	7.5194	.1500	7.4792	333.8087
.1875	7.8894	.1875	7.9203	346.1406
.2250	8.4890	.2250	8.8401	353.8764
.2625	6.8479	.2625	6.6036	348.2612
.3000	5.5214	.3000	5.4712	331.9989
.3375	4.1240	.3375	4.3982	342.1280
.3750	2.3373	.3750	2.3869	355.6275
.4125	1.3686	.4125	1.4552	352.0960
.4500	1.1069	.4500	1.4227	355.6275
.4875	.7639	.4875	1.0235	352.0960
.5250	.4015	.5250	.8110	355.5398
.5625	.2269	.5625	.5183	348.1900
.6000	.2915	.6000	.2265	342.1280
.6375	.1915	.6375	.1018	342.1280
.6750	.0839	.6750	.0678	342.1280
.7125	.0474	.7125	.1306	342.1280
.7500	.0574	.7500	.1648	342.1280
.7875	.0474	.7875	.1306	342.1280
.8250	.0574	.8250	.1648	342.1280
.8625	.0541	.8625	.1648	342.1280
.9000	.0915	.9000	.1018	342.1280
.9375	.0323	.9375	.2296	342.1280
.9750	.1005	.9750	.1648	342.1280
1.0000	.1005	1.0000	.1648	342.1280
1.0125	.1017	1.0125	.1306	342.1280
1.0250	.1017	1.0250	.1306	342.1280
1.0375	.1017	1.0375	.1306	342.1280
1.0500	.1860	1.0500	.1434	342.1280
1.0625	.2700	1.0625	.0678	342.1280
1.0750	.2586	1.0750	.1691	342.1280
1.0875	.3119	1.0875	.3119	342.1280
1.1000	.6119	1.1000	.6119	342.1280
1.1125	1.2176	1.1125	1.2176	342.1280
1.1250	2.586	1.1250	2.586	342.1280
1.1375	5.625	1.1375	5.625	342.1280
1.1500	11.160	1.1500	11.160	342.1280
1.1625	11.160	1.1625	11.160	342.1280
1.1750	11.160	1.1750	11.160	342.1280
1.1875	11.160	1.1875	11.160	342.1280
1.2000	11.160	1.2000	11.160	342.1280
1.2125	11.160	1.2125	11.160	342.1280
1.2250	11.160	1.2250	11.160	342.1280
1.2375	11.160	1.2375	11.160	342.1280
1.2500	11.160	1.2500	11.160	342.1280
1.2625	11.160	1.2625	11.160	342.1280
1.2750	11.160	1.2750	11.160	342.1280
1.2875	11.160	1.2875	11.160	342.1280
1.3000	11.160	1.3000	11.160	342.1280
1.3125	11.160	1.3125	11.160	342.1280
1.3250	11.160	1.3250	11.160	342.1280
1.3375	11.160	1.3375	11.160	342.1280
1.3500	11.160	1.3500	11.160	342.1280
1.3625	11.160	1.3625	11.160	342.1280
1.3750	11.160	1.3750	11.160	342.1280
1.3875	11.160	1.3875	11.160	342.1280
1.4000	11.160	1.4000	11.160	342.1280
1.4125	11.160	1.4125	11.160	342.1280
1.4250	11.160	1.4250	11.160	342.1280
1.4375	11.160	1.4375	11.160	342.1280
1.4500	11.160	1.4500	11.160	342.1280
1.4625	11.160	1.4625	11.160	342.1280
1.4750	11.160	1.4750	11.160	342.1280
1.4875	11.160	1.4875	11.160	342.1280
1.5000	11.160	1.5000	11.160	342.1280



203.5065  
258.5681  
355.7389  
251.3149  
333.6881  
246.4036  
288.1975  
290.6290  
251.0452  
311.0897  
245.8153

.2007  
.2632  
.1018  
.2296  
.1648  
.1306  
.1434  
.0678  
.1691  
.1165  
.1658

.8250  
.8625  
.9000  
.9375  
.9750  
1.0125  
1.0500  
1.0875  
1.1250  
1.1625  
1.2000

207.4330  
234.5326  
25.5466  
334.2910  
22.5616  
49.7359  
51.0251  
56.3574  
62.6402  
61.2176  
56.8146

.1004  
.0541  
.0915  
.0323  
.1005  
.1017  
.1860  
.2700  
.2586  
.3119  
.2981

.8250  
.8625  
.9000  
.9375  
.9750  
1.0125  
1.0500  
1.0875  
1.1250  
1.1625  
1.2000







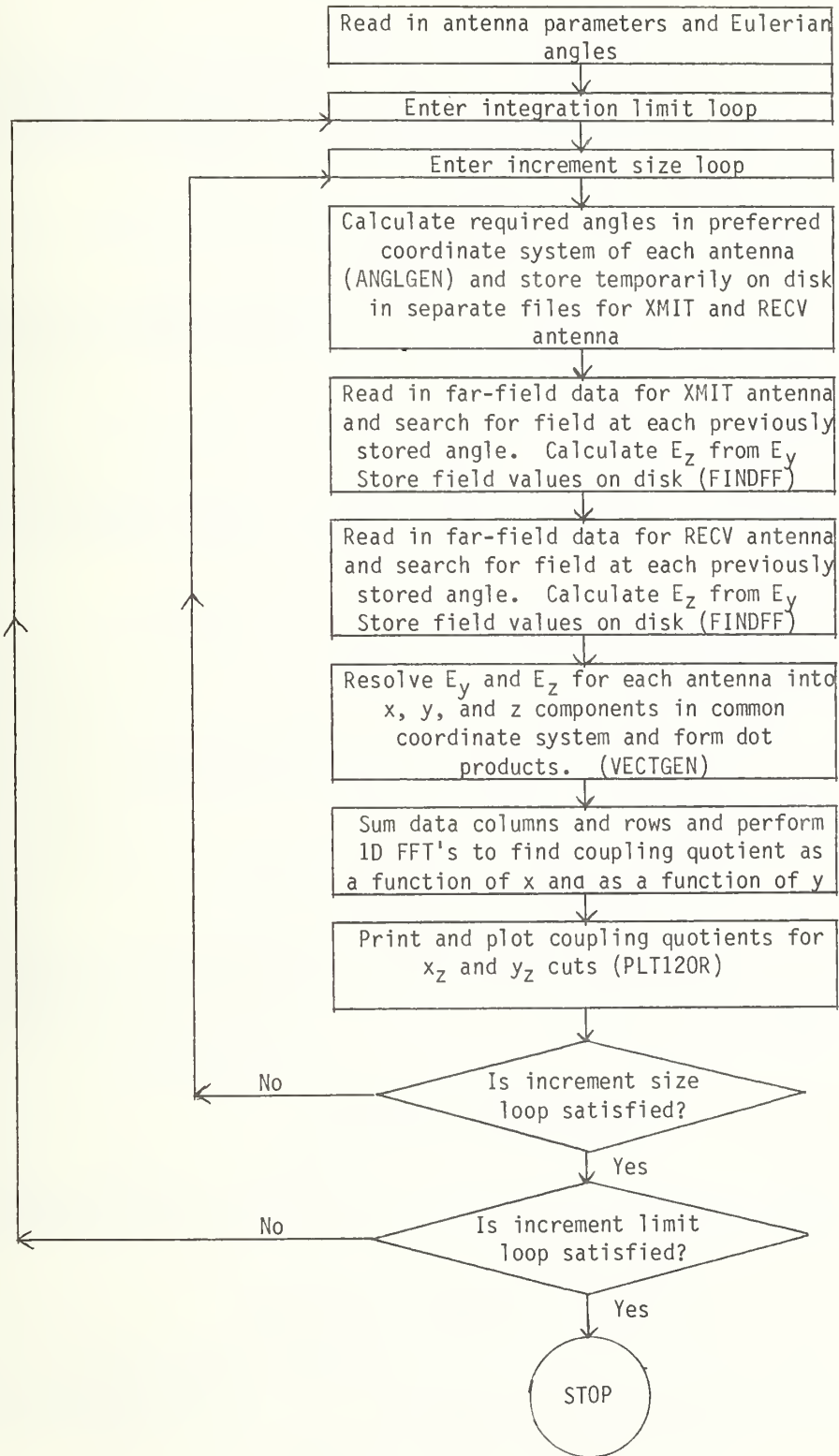


## APPENDIX B. CUPLNF - CALCULATION OF COUPLING BETWEEN ANTENNAS

This appendix includes detailed documentation of the program which calculates coupling between two antennas given their far-field patterns. This program, as presented here, uses only a single component of the far field for each antenna, and is thus applicable only for linearly polarized antennas oriented with the major components of their polarization vectors lying in a common plane. The inclusion of the cross component in the calculation is not a difficult extension to the program. Each subroutine is individually documented except for those which are also used in PROGRAM POMODL and are discussed in Appendix A. The final section of the appendix includes a sample input deck and a sample program output.

### B.1 GENERAL OVERVIEW OF COMPUTER PROGRAM

The program CUPLNF and its associated subroutines are described in detail in the following subsections. The flow chart below is presented in order to give the reader a general overview of the program package.



## B.1.1 PROGRAM CUPLNF (INPUT, OUTPUT, TAPE 1, TAPE 3, ..., TAPE 8)

### PURPOSE:

To compute and plot the mutual coupling between a transmitting and receiving antenna of arbitrary orientations and separation from the given complex far-electric-field pattern of each antenna.

### METHOD:

Evaluate eq (32) of the main text along x and y perpendicular lines or cuts, using the fast Fourier transformation.

### GENERAL DISCUSSION:

The main program divides conveniently into six subsections which list sequentially as follows:

- 1) General information about program,
- 2) Specification statements,
- 3) Definition and reading of input data,
- 4) Limits of integration and number of integration points,
- 5) Filling of the input matrices (AX and AY) to the FFT FOURT, and
- 6) Printout and plotting.

#### General Information about the Program

This subsection is a self-explanatory aid providing the program user with specific definitions of the main input parameters required by the program, as well as with a general feel for what the program does.

#### Specification Statements

This subsection merely dimensions, equivalences, and comments the appropriate arrays, and declares the necessary complex and integer variables.

#### Definition and Reading of Input Data

This subsection defines and reads from data cards the input variable parameters to the program. A list of the required data cards follows:

Card 1	Col. 1-40	An alphanumeric identifier, usually the name and telephone extension of the person submitting the job.
--------	-----------	--



Card 2 Col. 1-80 An alphanumeric identifier specifying the particular case being studied.

Except where specifically noted all data on the following cards must have the decimal point explicitly specified.

Card 3 Col. 1-10 Frequency of operation in GHz.

Col. 11-20 Distance between origins of the reference coordinates of the two antennas in meters.

Col. 21-30 x-spacing corresponding to the near-field spacing for the transmitting antenna.

Col. 31-40 y-spacing corresponding to the near-field spacing for the transmitting antenna.

Col. 41-50 x-spacing corresponding to the near-field spacing for the receiving antenna.

Col. 51-60 y-spacing corresponding to the near-field spacing for the receiving antenna.

Col. 61-70 Ratio of transmitting to receiving antenna feed mode admittances.

Col. 71 Set equal to 1 if spectrum rather than far-field pattern is given

Card 4 Col. 1-10 Maximum value for plot. If this field is left blank, the scale is chosen to fill the plot page.

The remaining data on this card are integer data, and must be right justified in the field provided.

Col. 11-15 Lower index in the increment loop.  
(Set equal to 1 if field is blank).

Col. 16-20 Upper index in the increment loop.  
(Set equal to 1 if field is blank).

Col. 21-25 Lower index in the integration limits loop.  
(Set equal to 1 if field is blank).

Col. 26-30 Upper index in the integration limits loop.  
(Set equal to 1 if field is blank).

Card 5

Col. 1-10      Transmitting antenna gain in dB.

Col. 11-20     Magnitude of transmitting antenna far-field pattern on boresight. (Allows for normalization of far-field pattern).

Col. 21-30     Radius of transmitting antenna in meters.

Col. 31-40     PHI        |  
 Col. 41-50     THETA     | - Eulerian angles of reoriented transmit antenna.  
 Col. 51-60     PSI        |

Col. 61-65     NROWT - Number of rows of data in transmit antenna pattern. (Integer data right justified in field).

Col. 66-70     NCOLT - Number of columns of data in transmit antenna pattern. (Integer data right justified in field).

Col. 71-80     Transmit antenna pattern file identifier.

Card 6

Col. 1-10      Receive antenna gain in dB.

Col. 11-20     Magnitude of receive antenna far-field pattern on boresight. (Allows for normalization of far-field pattern).

Col. 21-30     Radius of receive antenna in meters.

Col. 31-40     PHIP       |  
 Col. 41-50     THETHP   | - Eulerian angles of reoriented receive antenna.  
 Col. 51-60     PSIP       |

Col. 61-65     NROWR - Number of rows of data in receive antenna pattern. (Integer data right justified in field).

Col. 66-70     NCOLT - Number of columns of data in receive antenna pattern. (Integer data right justified in field.)

Col. 71-80     Receive antenna pattern file identifier.

Card 7

Col. 1-20      GAMT - Transmit antenna reflection coefficient. (Real part 1-10, imaginary part 11-20).

Col. 21-40     GAMR - Receive antenna reflection coefficient. (Real part 21-30, imaginary part 31-40).

Col. 41-60     GAML - Receiving load reflection coefficient.  
(Real part 41-50, imaginary part 51-60).

#### Limits of Integration and Number of Integration Points

In the analysis of the main text, it is shown that only the far-field pattern within the sheaf of angles mutually subtended by the two antennas is necessary to accurately compute the coupling between the antennas. These reduced limits of integration artificially bandlimit the coupling and thus increase the integration increments required by the sampling theorem. In all, the number of integration points is drastically reduced. This subsection of CUPLNF computes a maximum solid angle mutually subtended by the antenna and translates this information into specific limits of integration for  $k_x$  and  $k_y$ . In addition, the integration increments and subsequently the number of integration points in the x and y directions are also obtained in this subsection.

#### Filling of the Input Matrices (AX and AY) to the FFT FOURT

Now that the previous subsection has computed the points and limits of integration, the far-field patterns of each antenna must be retrieved from input files at the specified points of integration. These far-field arrays are inserted as input into the FFT FOURT in order to compute the coupling quotient from eq (32) of the main text. The subroutine FINDFF, documented separately, takes the required array of far-field integration points (directions) searches the input files containing the far field for the value of far field in the required directions, and outputs the array of far-field values to be used eventually by FOURT.

Before calling FINDFF, the program must calculate the far-field directions corresponding to the integration variables  $k_x$ ,  $k_y$  in the integral (summation) of eq (32). This is accomplished through the subroutine ANGLGEN, which is documented separately.

After the far-fields are obtained from FINDFF, their dot product must be determined as eq (36) demonstrates explicitly. This scalar product is accomplished through the short subroutine VECTGEN, which has been documented separately from CUPLNF.

The dot products of the far-fields found from VECTGEN are appropriately placed in two arrays, AX and AY, from which the FFT subroutine FOURT computes the near-field coupling quotient along two mutually perpendicular x and y cuts.

## Printout and Plotting

This subsection simply prints and plots the magnitude of the coupling quotient (i.e., coupling loss ratio) along the mutually orthogonal x and y cuts which lie normal to the line of separation. Because of the possible lack of a common plotting system, curves are made using a general purpose page printer subroutine (PLT12OR). The coupling is plotted in the x and y directions over a distance approximately equal to twice the sum of the diameters of the two antennas.

### SYMBOL DICTIONARY:

#### Variables (in alphabetical order)

- ABL = Intermediate variable for defining the range of  $k_x/k$  and  $k_y/k$ . The range of ABL beyond XKLIM is zero filled.
- ACLCUT = A real array used to store the magnitude of the coupling quotient along X0 and Y0 perpendicular axes or cutter.
- AX,AY = Complex arrays used to store first the coupling far-field product then the coupling quotient along X0 and Y0 cuts, respectively.
- (A1,A2), (B1,B2) = The limits of integration of  $k_x/k$  and  $k_y/k$ , respectively.
- BFAC = Variable which adjusts the integration increments, and should be approximately 1 or 2; making BFAC larger tests whether convergence has been reached.
- CEE = Speed of light in gigameters per second = .2997925.
- COEF = The coefficient of the summation in eq (32) of the main text with the exponential factor omitted .
- CUPLDB = Coupling quotient for two antennas expressed in dB.
- C1,C2 = The  $k_x/k$  and  $k_y/k$  increments, respectively, i.e.,  $(a_1+a_2)/N_1$  and  $(b_1+b_2)/N_2$  in eq (32).
- DATA = Array containing far-field pattern of transmitting or receiving antenna, used in SUBROUTINE FINDFF and included here for storage allocation purposes only.
- DIAMR,(DIAMT) = Twice the larger of RADR (RADT) or WAVELGTH of the receiving (transmitting) antenna.
- DIAMSUM = DIAMR plus DIAMT.
- DKOK = The approximate  $k_x/k$  and  $k_y/k$  integration increments, i.e.,  $N_1 \approx (A1+A2)/DKOK$  and  $N_2 \approx (B1+B2)/DKOK$ .
- DLX,DLY = Subsequent labels for (DLXR and DLXT), (DLYR and DLYT).
- DLXR,DLXT = x - increment which corresponds to the  $k_x$  increment of the receiving and transmitting antenna, respectively.
- DLYR,DLYT = y - increment which corresponds to the  $k_y$  increment of the receiving and transmitting antenna, respectively.

DPHI, DPHIP, DPSI, DPSIP,  
DTHETA, DTHETAP = The Eulerian angles PHI, PHIP, ....DTHETAP expressed in degrees rather than radians.

DTR = Degree to radian conversion factor.

DX,DY = The increments in X0 and Y0, respectively, over which the coupling quotient is computed by the FFT.

ETOER = Ratio of characteristic admittance of the transmitting antenna feed mode to the characteristic admittance of the receiving antenna feed mode.

FDOTFP = The dot product of the complex far-electric-field pattern of the two antennas.

FFRMX,FFTMX = Magnitude of unnormalized far-field pattern at THETA = 0 for the receiving and transmitting antennas, respectively.

FMM = The mismatch factor,  $1/(1-GAMR \cdot GAML)$  in the right, receiving antenna.

FREQ = Frequency in Hz.

FX,FY,FZ = The complex rectangular components of far electric field in the preferred coordinate system fixed in the left, transmitting antenna. (FX and FY are also used later in the program as intermediate complex variables.)

FXP,FYP,FZP = The complex rectangular components of far electric field in the preferred coordinate system fixed in the right, receiving antenna.

FXR,FYR,FZR = The complex rectangular components of the far electric field of the right receiving antenna in its mutual coupling coordinate system.

FXT,FYT,FZT = The complex rectangular components of the far electric field of the left, transmitting antenna in its mutual coupling coordinate system.

GAINR,GAINT = Gain in dB of receiving and transmitting antennas, respectively.

GAML,GAMR,GAMT = Reflection coefficient of receiving load, receiving antenna, and transmitting antenna, respectively.

HEAD = Integer array identifier for case under study.

IBFAC = Loop index for varying BFAC.

ID(I) = Integer array (with index I) identifier for programmer's name and one extension.

IDAYHRR,IDAYHRT = File identifier for receiving and transmitting antenna data, respectively.

ISCL = Integer indexer for conditional statements.

ISPECT = Spectrum flag. Set equal to 1 if spectrum rather than far-field patterns specified.

IXLIM = Loop index for varying XLIM.

J1,J2 = Dummy loop indices used in the filling of the AX, AY coupling product arrays, and later in the printout statements.

L = Dummy index for write and read statements.

M1,M2 = Dummy loop indices used in the multiplication of the sum in eq (32) by the preceding factors.

NBF1,NBF2 = Begin and end index for range of BFAC.

NCOLR,NCOLT = Number of columns of data in receive and transmit patterns, respectively.

NN1,NN2 = Integer arrays of dimension (1) used in call to FFT subroutine FOURT and equal to N1 and N2, respectively.

NROWR,NROWT = Number of rows of data in receive and transmit pattern, respectively.

NRX2R,NRX2T = NROWR and NROWT x 2.

NX,NY = Four times N1 and N2, respectively.

NXL1,NXL2 = Begin and end index for range of XLIM.

N1,N2 = Integers equal to the number of  $k_x$  and  $k_y$  integration points, respectively.

N1MAX,N2MAX = Integers determining maximum of the x and y range, respectively, over which the coupling quotient is plotted.

N1MIN,N2MIN = Integers determining the minimum of the x and y range over which the coupling quotient is plotted.

N10,N20 = Intermediate integers used to define (N1MIN,N1MAX) and (N2MIN,N2MAX), respectively.

N11,N22 = Number of points in the x and y range, respectively, over which the coupling quotient is plotted.

PHI,THETA,PSI = Eulerian angles of the left transmitting antenna as shown in figure 2.

PHIP,THETAP,  
 PSIP = Eulerian angles of the right, receiving antenna as shown in figure 3.

PHIR,THETAR = Spherical angles in the preferred coordinate system fixed in the right, receiving antenna, corresponding to the direction  $k_x/k$ ,  $k_y/k$  ( $\theta_p, \theta_p$  of eqs (13) and shown in figure 3).

PHIT,THETAT = Spherical angles in the preferred coordinate system fixed in the left, transmitting antenna, corresponding to the direction  $k_x/k$ ,  $k_y/k$  ( $\theta_A, \theta_A$  of eqs (13) and shown in figure 2).

PI =  $\pi = 3.14159\dots$

R,T = Complex array containing the spherical angle coordinates for the coordinate system fixed in the receiving and transmitting antenna, respectively.

RADR,RADT = Radii of the smallest sphere circumscribing the right receiving and left transmitting antenna, from their respective origins.

RCUT = Maximum ordinate value for plots. If RCUT equals 0, plot is self-scaled.

RG,TG = Input reflection mismatch factor for receiving and transmitting antenna, respectively.

SUM2 = Dummy summation variables used in the filling of the AX, AY matrices.

TKOKSQ = Magnitude squared of the transverse part of the propagation vector.

TSUM21 = Summation variable used to compute the coupling quotient at  $X_0 = 0$ ,  $X_0 = 0$  by summing directly without the use of the FFT (as a check).

WAVLGTH = Wavelength in meters.

WORK = Complex array required only by FFT subroutine FOURT.

X = Array containing the abscissa values for plots.

XDUM = Dummy variable used in MINMAX when this subroutine is used with a one dimensional array.

XK =  $2\pi/\lambda$ .

XKLIM = Real variable which limits the range of  $k_x/k$ , and  $k_y/k$  integration when its value is less than XKMAX.

XKMAX = An upper bound (less than 1.0) on XLIM; except for very close antennas XLIM will usually be less than XKMAX anyway.

XKMIN = Sum of the diameters of the two antennas divided by their separation distance; this variable is approximately proportional to the mutual angle subtended by the two antennas; XLIM = XKMIN times XLIM when this product is less than XKMAX.

XXOK,XYOK =  $k_x/k$  and  $k_y/k$ , respectively.

XLIM = Intermediate variable used for adjusting XLIM; making XLIM larger tests whether a wide enough spectrum has been included.

XMAX,XMIN = Maximum and minimum abscissa values for plots.

XNX,XNY,XNZ = Variable used for incrementing  $k_x/k$ ,  $k_y/k$ , and  $\gamma/k$ , respectively.

XO,YO,ZO = X, Y, Z coordinates of the origin of the right receiving antenna in the mutual coupling coordinate system of the left transmitting antenna; specifically ZO is the separation distance (d in eq (32)) between antennas.

### File Names

INPUT, OUTPUT, TAPE 1, TAPE 3, ..., TAPE 8

### Subroutines Not Within FORTRAN Library (in alphabetical order)

ANGLGEN (Documented below)

FINDFF (Documented below)

FOURT (Standrd FFT subroutine with documentation within its own comment cards)

MINMAX (Documented below)

PLT120R (Page printer subroutine)

VECTGEN (Documented below)

Functions Inline or within Computer Library (in alphabetical order)

AMAX1(X,Y) = Maximum of X and Y.  
AMIN1(X,Y) = Minimum of X and Y.  
ATAN(X) = Angle between  $-\pi/2$  and  $\pi/2$  whose tangent is X.  
CABS(C) = Absolute value of complex number C.  
CEXP(C) = Complex exponential of complex number C.  
CMPLX(X,Y) = Complex number  $X + iY$ .  
EOF = EOF(End of File).  
EXIT = (Terminates execution and returns control to operating system.)  
SQRT(X) = Square root of X.

List of Complex Quantities

AX, AY, COEF, ETA, FDOTFP, FMM, FX, FXP, FXR, FXT, FY, FYP, FYR, FYT, FZ, FZP, FZR, FZT, R, SUM2, T, TSUM21, WORK, CEXP, CMPLX.

COMMON BLOCKS:

The labeled common in CUPLNF is described below with a list of routines in which it is used. The variables are defined in the symbol dictionary

COMMON /FAR/ N1, N2, NX, NY, DLX, DLY, XK, ISPECT

Routines using /FAR/: CUPLNF, FINDFF.



```

1      PROGRAM CURLNF (INPUT, OUTPUT, TAPE1, TAPE3, TAPE4, TAPE5,      CURLNF      1
      1 TAPE6, TAPE7, TAPE8, TAPE60=INPUT)                          CURLNF      2
C      CURLNF      3
C      CURLNF      4
5      C-  GENERAL INFORMATION ABOUT PROGRAM                          CURLNF      5
C      CURLNF      6
C      THIS PROGRAM COMPUTES THE COUPLING QUOTIENT BETWEEN A      CURLNF      7
C      TRANSMITTING ANTENNA ON THE LEFT AND A RECEIVING ANTENNA ON THE      CURLNF      8
C      RIGHT OF ARBITRARY RELATIVE ORIENTATION AND SEPARATION,      CURLNF      9
10     C      FROM THE GIVEN COMPLEX FAR-FIELD PATTERN OF EACH ANTENNA.      CURLNF     10
C      CURLNF     11
C      THE COUPLING QUOTIENT IS COMPUTED ALONG XO AND YO PERPENDICULAR      CURLNF     12
C      LINES OR CUTS.                                              CURLNF     13
C      CURLNF     14
15     C      AX,AY,AND WOPK SHOULD BE DIMENSIONED .GE. THE LARGER OF (N1,N2).      CURLNF     15
C      ACLCUT AND X SHOULD BE DIMENSIONED AT LEAST 2 GREATER THAN THE      CURLNF     16
C      LARGER OF (N1, N2).                                         CURLNF     17
C      FYT, FYP, FXT, FZP, P, AND T SHOULD BE DIMENSIONED .GE. N2.      CURLNF     18
C      DATA SHOULD BE LARGE ENOUGH TO CONTAIN ALL OF THE INPUT FAR-FIELD      CURLNF     19
20     C      DATA FOR EITHER ANTENNA IE. .GF. 2*NRQWT*NCOLT OR 2*NRDWR*NCOLR.      CURLNF     20
C      CURLNF     21
C      RHI,THETA,PSI ARE THE EULERIAN ANGLES OF THE REORIENTED      CURLNF     22
C      TRANSMITTING AXES WITH RESPECT TO THE AXES FIXED IN THE      CURLNF     23
C      TRANSMITTING ANTENNA.                                       CURLNF     24
25     C      PHIP,THETAP,PSIP ARE THE EULEPIAN ANGLES OF THE REORIENTED      CURLNF     25
C      RECEIVING AXES WITH RESPECT TO THE AXES FIXED IN THE      CURLNF     26
C      RECEIVING ANTENNA.                                          CURLNF     27
C      (XO,YO,ZO) ARE THE COORDINATES OF THE ORIGIN OF THE RECEIVING      CURLNF     28
C      ANTENNA IN THE REORIENTED RECTANGULAR SYSTEM OF THE TRANSMITTING      CURLNF     29
30     C      ANTENNA.                                              CURLNF     30
C      THE REORIENTED COORDINATE SYSTEMS OF EACH ANTENNA ARE THE COMMON      CURLNF     31
C      MUTUAL COUPLING COORDINATE SYSTEMS OF THE ANTENNAS.        CURLNF     32
C      THE COORDINATE SYSTEM FIXED IN EACH ANTENNA IS THE #PPEFFRRED#      CURLNF     33
C      SYSTEM IN WHICH THE FAR-FIELDS OF EACH ANTENNA ARE GIVEN.   CURLNF     34
35     C      ZO MUST BE SPECIFIED,BUT THE RANGE OF XO AND YO ARE DETERMINED      CURLNF     35
C      IMPLICITLY BY THE REQUIREMENTS OF THE ALGORITHM FDUPT.     CURLNF     36
C      RADT=RADIUS OF SMALLEST SPHERE WHICH CIRCUMSCRIBES THE      CURLNF     37
C      TRANSMITTING ANTENNA FROM ITS ORIGIN.                       CURLNF     38
C      RADR=RADIUS OF SMALLEST SPHERE WHICH CIRCUMSCRIBES THE      CURLNF     39
C      RECEIVING ANTENNA FROM ITS ORIGIN.                          CURLNF     40
C      DIAMT = TWICE THE LARGER OF RADT OR WAVLGTH                 CURLNF     41
C      DIAMP = TWICE THE LARGER OF RADP OR WAVLGTH                 CURLNF     42
C      BFAC ADJUSTS THE INTEGRATION INCREMENTS, AND SHOULD BE     CURLNF     43
C      APPROXIMATELY 1 OR .2. MAKING BFAC LARGER TESTS WHETHER     CURLNF     44
45     C      CONVERGENCE HAS BEEN REACHED.                         CURLNF     45
C      XLIM ADJUSTS THE NONZERO-FILL PORTION OF THE INTEGRATION      CURLNF     46
C      RANGE, AND SHOULD BE APPROX. 1 OR 2. MAKING XLIM LARGER     CURLNF     47
C      TESTS WHETHER A WIDE ENOUGH INTEGRATION RANGE HAS BEEN     CURLNF     48
C      INCLUDED. INCREASING XLIM ALSO DECREASES THE INTEGRATION     CURLNF     49
50     C      INCREMENTS PROPORTIONATELY TO PREVENT ALIASING.      CURLNF     50
C      CURLNF     51
C      A1,A2,R1,B2 DEFINE THE TOTAL (WITH ZERO-FILL) INTEGRATION      CURLNF     52
C      RANGES (KX/K FROM -A1 TO APPROX.A2) AND (KY/K FROM -R1 TO     CURLNF     53
C      APPROX.B2), IN INCREMENTS OF (A1+A2)/N1 OR (R1+B2)/N2 ARROX. EQUAL TO      CURLNF     54
55     C      DKOK = WAVLGTH/(2*(DIAMT + DIAMP)*BFAC*XLIM).        CURLNF     55
C      CURLNF     56
C      IF SQRT((KX/K)**2+(KY/K)**2) IS .GE. XKLIM THE SPECTRUM     CURLNF     57
C      IS SET EQUAL TO ZERO. (APPROXIALBLE ZERO FILLING IS AN      CURLNF     58
C      OPTION DESIGNED TO ALLOW FINER INCREMENTS DX AND DY AT      CURLNF     59
60     C      WHICH THE COUPLING QUOTIENT IS COMPUTED BY THE FFT.)   CURLNF     60
C      CURLNF     61
C      XKLIM MUST BE EQUAL TO OR LESS THAN 1 BECAUSE              CURLNF     61
C      CURLNF     62
C      THE PROGRAM NEGLECTS THE EVANESCENT MODES. IN ORDER NOT TO      CURLNF     62
C      GET TOO CLOSE TO THE 1/GAMMA SINGULARITY, IT IS SAFER TO      CURLNF     63
C      CHOOSE XLIM NO LARGER THAN XKMAX= .9.                        CURLNF     64
65     C      XLIM ADJUSTS XLKIM. IF AN ACCUPATE COUPLING QUOTIENT IS      CURLNF     65
C      REQUIRED ONLY FOR SMALL VALUES OF XO/((DIAMT + DIAMR)*BFAC*XLIM)      CURLNF     66
C      AND YO/((DIAMT + DIAMP)*BFAC*XLIM), XLIM NEED NOT BE MORE      CURLNF     67
C      THAN 1 OR 2. IF ACCURACY IS DESIRED FOR LARGER XO AND YO      CURLNF     68
C      AS WELL, XLIM SHOULD BE MADE CORRESPONDINGLY LARGER. AS      CURLNF     69
70     C      MENTIONED ABOVE, MAKING XLIM LARGER TESTS WHETHER A      CURLNF     70
C      WIDE ENOUGH SPECTRUM HAS BEEN INCLUDED.                    CURLNF     71
C      CURLNF     72
C      THE XO AND YO INCREMENTS ARE DX=WAVLGTH/(A1+A2) AND        CURLNF     73
C      DY=WAVLGTH/(B1+B2).                                         CURLNF     74
75     C      THE RANGE OF BOTH XO AND YO IS GIVEN APPROXIMATELY      CURLNF     75
C      BY -(DIAMT + DIAMR)*BFAC*XLIM TO +(DIAMT + DIAMR)*BFAC*XLIM,      CURLNF     76
C      BUT ONLY -(DIAMT + DIAMP) TO +(DIAMT + DIAMR) APPROXIMATELY      CURLNF     77
C      IS PRINTED AND

```

```

C      PLOTTED (WHEN XLIM*BFAC IS GREATER THAN OR EQUAL TO 1).          CUPLNF    78
C      CFF IS THE SPEED OF LIGHT IN GIGAMETERS PER SECOND.           CUPLNF    79
80    C      FMM IS THE MISMATCH FACTOR FOR THE RECEIVING ANTENNA.     CUPLNF    80
C      CUPLNF    81
C      CUPLNF    82
C-     SPECIFICATION STATEMENTS                                       CUPLNF    83
C      CUPLNF    84
85    C      COMPLEX AX(1000), AY(1000), T(1000), R(1000), WORK(1000)  CUPLNF    85
C      COMPLEX FYT(1000), FYR(1000), FZT(1000), FZR(1000)           CUPLNF    86
C      COMPLEX FMM, CDEF, GAMT, GAMR, GAML                             CUPLNF    87
C      COMPLEX FXT, FXR                                               CUPLNF    88
C      COMPLEX FX,FY,FZ                                              CUPLNF    89
90    C      COMPLEX FXR,FYP,FZP                                       CUPLNF    90
C      COMPLEX FDOTEP                                               CUPLNF    91
C      COMPLEX TSUM21,SUM2                                           CUPLNF    92
C      CUPLNF    93
95    C      DIMENSION NN1(1), NN2(1), HEAD(R), ID(4)                 CUPLNF    94
C      DIMENSION ACLCUT(1010), X(1010), DATA(8192)                 CUPLNF    95
C      CUPLNF    96
C      INTEGER HEAD                                                  CUPLNF    97
C      CUPLNF    98
100   C      EQUIVALENC (T,FZT), (R,FZR), (ACLCUT,FYT), (WORK,FYR)   CUPLNF    99
C      EQUIVALENC (AX, DATA(1)), (AY, DATA(2500)), (X, T)         CUPLNF   100
C      CUPLNF   101
C      COMMON /FAR/ N1, N2, NX, NY, DLX, DLY, XK, ISPFCT            CUPLNF   102
C      CUPLNF   103
105   C      DEFINITION AND READING OF INPUT DATA                    CUPLNF   104
C      CUPLNF   105
C      CUPLNF   106
C-     ID AND HEAD ARE ALPHANUMERIC IDENTIFIERS                     CUPLNF   107
C-     ID IS PROGRAMMERS NAME AND PHONE EXTENSION                   CUPLNF   108
C-     HEAD IS THE IDENTIFIER FOR THE CASE UNDER STUDY              CUPLNF   109
110   C      CUPLNF   110
C-     FRFQ = FREQUENCY OF OPERATION IN GHZ.                         CUPLNF   111
C-     ZD   = SEPARATION BETWEEN ANTENNA REFERENCE POINTS (SEE COMMENTS CUPLNF   112
C           ABOVE)                                                    CUPLNF   113
115   C-     DLXT = X-INCREMENT WHICH CORRESPONDS TO KX INCREMENT XMIT  CUPLNF   114
C-     DLYT = Y-INCREMENT WHICH CORRESPONDS TO KY INCREMENT XMIT  CUPLNF   115
C-     DLXR = X-INCREMENT WHICH CORRESPONDS TO KX INCREMENT RECV   CUPLNF   116
C-     DLYR = Y-INCREMENT WHICH CORRESPONDS TO KY INCREMENT RECV   CUPLNF   117
C-     FTOER = RATIO OF CHARACTERISTIC ADMITTANCE OF TRANSMITTING   CUPLNF   118
C           ANTENNA FEED MODE TO CHARACTERISTIC ADMITTANCE OF THE   CUPLNF   119
120   C           RECEIVING ANTENNA FEED MODE                          CUPLNF   120
C-     ISPFCT = SPECTRUM FLAG - SET EQUAL TO 1 IF INPUT DATA IS SPECTRUM CUPLNF   121
C           RATHER THAN FAR FIELD                                     CUPLNF   122
C      CUPLNF   123
C-     PCUT = MAXIMUM ORDINATE VALUE FOR PLOTS. IF PCUT .EQ. 0, PLOT CUPLNF   124
C           IS SELF-SCALED                                           CUPLNF   125
125   C-     NBEF1,NRF2 = BEGIN AND END INDEX FOR RANGE OF BFAC      CUPLNF   126
C-     NXL1,NYL2 = BEGIN AND END INDEX FOR RANGE OF XLIM            CUPLNF   127
C      CUPLNF   128
C-     GAINR = GAIN OF XMIT ANTENNA IN DB.                            CUPLNF   129
130   C-     FETMX = MAGNITUDE OF FAR-FIELD PATTERN (UNNORMALIZED) AT  CUPLNF   130
C           THETA = 0, XMIT                                           CUPLNF   131
C-     RADT = RADIUS OF TRANSMIT ANTENNA (SEE COMMENTS ABOVE)       CUPLNF   132
C-     PHI, THETA, PSI = EULER ANGLES IN DEGREES FOR XMIT ANTENNA (SEE CUPLNF   133
C           COMMENTS ABOVE)                                           CUPLNF   134
135   C-     NROWT = NUMBER OF ROWS OF DATA IN TRANSMIT PATTERN      CUPLNF   135
C-     NCOLT = NUMBER OF COLUMNS OF DATA IN TRANSMIT PATTERN      CUPLNF   136
C-     IDAYHRT = FILE IDENTIFIER FOR XMIT DATA                      CUPLNF   137
C      CUPLNF   138
C      CUPLNF   139
140   C-     GAINR = GAIN OF RECV ANTENNA IN DB.                       CUPLNF   140
C-     FEPMY = MAGNITUDE OF FAR-FIELD PATTERN (UNNORMALIZED) AT     CUPLNF   141
C           THETA = 0, RECV                                           CUPLNF   142
C-     RADR = RADIUS OF RECEIVE ANTENNA (SEE COMMENTS ABOVE)       CUPLNF   143
C-     PHIP, THETAP, PSIP = EULER ANGLES IN DEGREES FOR RECV ANTENNA (SEE CUPLNF   144
C           COMMENTS ABOVE)                                           CUPLNF   145
145   C-     NROWR = NUMBER OF ROWS OF DATA IN RECEIVE PATTERN      CUPLNF   146
C-     NCOLR = NUMBER OF COLUMNS OF DATA IN RECEIVE PATTERN      CUPLNF   147
C-     IDAYHRP = FILE IDENTIFIER FOR XMIT DATA                      CUPLNF   148
C      CUPLNF   149
150   C-     GAMT = REFLECTION COEFFICIENT OF TRANSMITTING ANTENNA   CUPLNF   150
C-     GAMR = REFLECTION COEFFICIENT OF RECEIVING ANTENNA           CUPLNF   151
C-     GAML = REFLECTION COEFFICIENT OF RECEIVING LOAD              CUPLNF   152
C      CUPLNF   153
C      PRINT 5005                                                    CUPLNF   154
C      READ 5000, (ID(I), I = 1, 4)                                  CUPLNF   154

```

155		PPINT 5001, (ID(I), I = 1, 4)	CUPLNF	155
	20	PFAD 5000, HEAD	CUPLNF	156
		IF (ENDF(60)) 1001,21	CUPLNF	157
	21	PPINT 5001, HEAD	CUPLNF	158
	C		CUPLNF	159
160		PFAD 5022, FREQ, 70, DLXT, DLYT, DLXP, DLYR, ETOFP, ISPECT	CUPLNF	160
		PPINT 5023, FREQ, Z0, DLXT, DLYT, DLXP, DLYR, ETOFR, ISPECT	CUPLNF	161
		PFAD 5030, RCUT, NBF1, NBF2, NXL1, NXL2	CUPLNF	162
		PPINT 5031, PCUT, NBF1, NBF2, NXL1, NXL2	CUPLNF	163
		RFAD 5010, GAINP, FETMX, PAOT, PHI, THETA, PSI, NROWT, NCOLT,	CUPLNF	164
165		1 IDAYHPT	CUPLNF	165
		PRINT 5011, GAINP, FETMX, PAOT, PHI, THETA, PSI, NROWT, NCOLT,	CUPLNF	166
		1 IDAYHPT	CUPLNF	167
		RFAD 5010, GAINP, FETMX, RADP, PHIP, THETA, PSIP, NROWR, NCOLP,	CUPLNF	168
		1 IDAYHRP	CUPLNF	169
170		PPINT 5011, GAINP, FETMX, RADP, PHIP, THETA, PSIP, NROWR, NCOLP,	CUPLNF	170
		1 IDAYHRP	CUPLNF	171
		PFAD 5020, GAMT, GAMR, GAML	CUPLNF	172
		PPINT 5021, GAMT, GAMR, GAML	CUPLNF	173
	C		CUPLNF	174
175		PI=4.*ATAN(1.0)	CUPLNF	175
		CFE = .2997925	CUPLNF	176
		DTP = PI/180.	CUPLNF	177
		WAVLGTH=CFE/FREQ	CUPLNF	178
180		XK=2.*PI/WAVLGTH	CUPLNF	179
		FMM = 1./(1. - GAMR*GAML)	CUPLNF	180
		PHIP = PHIP*DTP	CUPLNF	181
		PHI = PHT*DTP	CUPLNF	182
		THETA = THETA*DTP	CUPLNF	183
185		THETA = THETA*DTP	CUPLNF	184
		PSIP = PSIP*DTP	CUPLNF	185
		PSI = PSI*DTP	CUPLNF	186
		GAINP = 10.*(GAINP/10.)	CUPLNF	187
		GAINR = 10.*(GAINR/10.)	CUPLNF	188
190		DIAMP = 2.*AMAX1(WAVLGTH, PADT)	CUPLNF	189
		DIAMP = 2.*AMAX1(WAVLGTH, PADP)	CUPLNF	190
		DIAMSUM = DIAMP + DIAMP	CUPLNF	191
		PRINT 7, DIAMSUM*DIAMSUM/WAVLGTH	CUPLNF	192
			CUPLNF	193
195		ISCL = 5	CUPLNF	194
		IF(PCUT .EQ. 0.) ISCL = 1	CUPLNF	195
		IF(NBF1 .EQ. 0) NBF1 = 1	CUPLNF	196
		IF(NBF2 .EQ. 0) NBF2 = 1	CUPLNF	197
		IF(NXL1 .EQ. 0) NXL1 = 1	CUPLNF	198
200		IF(NXL2 .EQ. 0) NXL2 = 1	CUPLNF	199
		IF(FETMX .EQ. 0.) FETMX = 1.	CUPLNF	200
		IF(FETMX .EQ. 0.) FETMX = 1.	CUPLNF	201
		IF(FETMX .EQ. 0.) FETMX = 1.	CUPLNF	202
			CUPLNF	203
			CUPLNF	204
205		LIMITS OF INTEGRATION AND NUMBER OF INTEGRATION POINTS	CUPLNF	205
			CUPLNF	206
		DO 1000 IXLIM = NXL1, NXL2	CUPLNF	207
		XLIM = IXLIM	CUPLNF	208
		DO 1000 IRFAC = NBF1, NBF2	CUPLNF	209
210		RFAC=IRFAC	CUPLNF	210
		XKMAX=.9	CUPLNF	211
		XKMIN = DIAMSUM/Z0	CUPLNF	212
		XK LIM=XLIM*XKMIN	CUPLNF	213
		XK LIM=AMIN1(XK LIM,XKMAX)	CUPLNF	214
215		XK LIM DEFINES THE NONZERO-FILL LIMITS OF INTEGRATION.	CUPLNF	215
			CUPLNF	216
		ABL = XK LIM	CUPLNF	217
		A1=ARL \$A2=ARL \$R1=ARL \$R2=ARL	CUPLNF	218
	C	FROM XK LIM TO ABL THERE IS ZERO FILLING. NEXT WE COMPUTE	CUPLNF	219
	C-	NUMBER OF INTEGRATION POINTS.	CUPLNF	220
220		DKDK = WAVLGTH/2./DIAMSUM	CUPLNF	220
		DKDK = AMIN1(DKDK, DIAMSUM/Z0/2.)	CUPLNF	221
		DKDK = DKDK/(RFAC*XK LIM)	CUPLNF	222
		NN1(1)=(A1+A2)/DKDK	CUPLNF	223
		NN1(1)=2*((NN1(1)+1)/2)	CUPLNF	224
225		NN2(1)=(R1+R2)/DKDK	CUPLNF	225
		NN2(1)=2*((NN2(1)+1)/2)	CUPLNF	226
		N1=NN1(1)	CUPLNF	227
		N2=NN2(1)	CUPLNF	228
			CUPLNF	229
230		C1=(A1+A2)/N1	CUPLNF	230
		C2=(R1+R2)/N2	CUPLNF	231

```

COFF = -C1*C2*FMM*SQPT(GAINT*GAINP)/(4.*PI*FFTMX*FFRMX)
RG = SQPT(1. - CABS(GAMP)**2)
TG = SQPT(1. - CABS(GAMT)**2)
COFF = COFF*RG*TG*SQPT(FTQER)
CUPLNF 232
CUPLNF 233
CUPLNF 234
CUPLNF 235
CUPLNF 236
CUPLNF 237
CUPLNF 238
CUPLNF 239
CUPLNF 240
CUPLNF 241
CUPLNF 242
CUPLNF 243
CUPLNF 244
CUPLNF 245
CUPLNF 246
CUPLNF 247
CUPLNF 248
CUPLNF 249
CUPLNF 250
CUPLNF 251
CUPLNF 252
CUPLNF 253
CUPLNF 254
CUPLNF 255
CUPLNF 256
CUPLNF 257
CUPLNF 258
CUPLNF 259
CUPLNF 260
CUPLNF 261
CUPLNF 262
CUPLNF 263
CUPLNF 264
CUPLNF 265
CUPLNF 266
CUPLNF 267
CUPLNF 268
CUPLNF 269
CUPLNF 270
CUPLNF 271
CUPLNF 272
CUPLNF 273
CUPLNF 274
CUPLNF 275
CUPLNF 276
CUPLNF 277
CUPLNF 278
CUPLNF 279
CUPLNF 280
CUPLNF 281
CUPLNF 282
CUPLNF 283
CUPLNF 284
CUPLNF 285
CUPLNF 286
CUPLNF 287
CUPLNF 288
CUPLNF 289
CUPLNF 290
CUPLNF 291
CUPLNF 292
CUPLNF 293
CUPLNF 294
CUPLNF 295
CUPLNF 296
CUPLNF 297
CUPLNF 298
CUPLNF 299
CUPLNF 300
CUPLNF 301
CUPLNF 302
CUPLNF 303
CUPLNF 304
CUPLNF 305
CUPLNF 306
CUPLNF 307
CUPLNF 308

```

```

310      NN2(1) = NY                                CUPLNF      309
      C                                           CUPLNF      310
      CALL FOURT(AX,NN1,1,+1,+1,WDRK)             CUPLNF      311
      CALL FOUPT(AY,NN2,1,+1,+1,WDRK)             CUPLNF      312
      DO 400 M1 = 1, NX                            CUPLNF      313
      Y0 = (-NY/2. + M1 - 1.)*WAVLGTH/(A1 + A2)   CUPLNF      314
315      FX=CFXP(CMPLX(0.,-YK*A1*Y0))              CUPLNF      315
      AX(M1)=FX*COFF*AX(M1)                       CUPLNF      316
      400 CONTINUE                                 CUPLNF      317
      DO 300 M2 = 1, NY                            CUPLNF      318
      Y0 = (-NY/2. + M2 - 1.)*WAVLGTH/(B1 + B2)   CUPLNF      319
320      FY=CFXP(CMPLX(0.,-YK*B1*Y0))              CUPLNF      320
      AY(M2)=FY*COFF*AY(M2)                       CUPLNF      321
      300 CONTINUE                                 CUPLNF      322
      C                                           CUPLNF      323
      C                                           CUPLNF      324
325      C- PRINTOUT AND PLOTTING.                 CUPLNF      325
      C                                           CUPLNF      326
      PRINT 5, XLIM, RFAC                          CUPLNF      327
      PRINT 15, NX, NY                             CUPLNF      328
      PRINT 35,WAVLGTH,RADT,PAOR,ZO                CUPLNF      329
330      DPHI=PHI*180./PI $DTHETA=THETA*180./PI $DPSI=PSI*180./PI
      DPHTP=PHIP*180./PI $DTHETAP=THETAP*180./PI $DPSIP=PSIP*180./PI
      PRINT 45,DPHI,DTHEA,DPSI, DPHTP,DTHEAP,DPSIP
      DY = WAVLGTH/(A1 + A2)/4. $ DY = WAVLGTH/(B1 + B2)/4.
      PRINT 55, -DX*NX/2., DX*(NX/2. - 1.), DX     CUPLNF      333
335      PRINT 65, -DY*NY/2., DY*(NY/2. - 1.), DY   CUPLNF      334
      PRINT 75, -A1*4, A2*4 - C1, C1              CUPLNF      335
      PRINT 85, -B1*4, B2*4 - C2, C2              CUPLNF      336
      PRINT 97,XCLIM                                CUPLNF      337
      CUPLDB = 20.*ALOG10(CABS(TSUM21*COFF))        CUPLNF      338
340      PRINT 95, TSUM21*COFF, CUPLDB             CUPLNF      339
      C                                           CUPLNF      340
      C                                           CUPLNF      341
      C                                           CUPLNF      342
      C                                           CUPLNF      343
      PRINT 27                                       CUPLNF      344
345      PRINT 25, (AX(J1), J1 = 1, NX)           CUPLNF      345
      PRINT 29                                       CUPLNF      346
      PRINT 25, (AY(J2), J2 = 1, NY)             CUPLNF      347
      C                                           CUPLNF      348
      C                                           CUPLNF      349
350      C                                           CUPLNF      350
      C                                           CUPLNF      351
      C                                           CUPLNF      352
      N10 = NX/(XLIM*BFAC) + .000001              CUPLNF      353
      N1MIN = NX/2 + 1 - N10/2.                   CUPLNF      354
355      N1MAX = NX/2 + 1 + N10/2.                 CUPLNF      355
      DO 501 J1 = N1MIN, N1MAX                     CUPLNF      356
      ACLOCUT(J1 - N1MIN + 1) = CABS(AX(J1))       CUPLNF      357
      X(J1 - N1MIN + 1) = (-NX/2. + J1 - 1.)*WAVLGTH/(A1 + A2)/4.
      PRINT 515, ACLOCUT(J1 - N1MIN + 1.), X(J1 - N1MIN + 1)
360      501 CONTINUE                               CUPLNF      358
      N11 = N1MAX - N1MIN + 1                      CUPLNF      359
      C                                           CUPLNF      360
      C                                           CUPLNF      361
      YMIN = X(1) $ XMAX = X(N11)                 CUPLNF      362
      IF (ISCL.NE. 1) GO TO 511                    CUPLNF      363
365      CALL MINMAX(ACLOCUT, XDUM, PCUT, N11, 1)    CUPLNF      364
      511 CALL PLOT12OR(X, ACLOCUT, XMAX, XMIN, RCUT, 0., N11, 1H*, 1, 1)
      PRINT 5041, HEAD, 10HMAGNITUDE , 10H VS X0  CUPLNF      365
      PRINT 510                                       CUPLNF      366
      N20 = NY/(XLIM*PFAC) + .000001              CUPLNF      367
      C                                           CUPLNF      368
      C                                           CUPLNF      369
370      N2MIN = NY/2 + 1 - N20/2.                 CUPLNF      370
      N2MAX = NY/2 + 1 + N20/2.                   CUPLNF      371
      DO 601 J2 = N2MIN, N2MAX                     CUPLNF      372
      ACLOCUT(J2 - N2MIN + 1) = CABS(AY(J2))       CUPLNF      373
      X(J2 - N2MIN + 1) = (-NY/2. + J2 - 1.)*WAVLGTH/(B1 + B2)/4.
      PRINT 615, ACLOCUT(J2 - N2MIN + 1.), X(J2 - N2MIN + 1)
375      601 CONTINUE                               CUPLNF      374
      N22 = N2MAX - N2MIN + 1                      CUPLNF      375
      C                                           CUPLNF      376
      C                                           CUPLNF      377
      C                                           CUPLNF      378
      XMIN = X(1) $ XMAX = X(N22)                 CUPLNF      379
      IF (ISCL.NE. 1) GO TO 611                    CUPLNF      380
380      CALL MINMAX(ACLOCUT, XDUM, PCUT, N22, 1)    CUPLNF      381
      611 CALL PLOT12OR(X, ACLOCUT, XMAX, XMIN, RCUT, 0., N22, 1H*, 1, 1)
      PRINT 5041, HEAD, 10HMAGNITUDE , 10H VS Y0  CUPLNF      382
      C                                           CUPLNF      383
      C                                           CUPLNF      384
385      C                                           CUPLNF      385
      C                                           CUPLNF      385

```

	C-	REWIND FILES	CUPLNF	386
	C		CUPLNF	387
		REWIND 1	CUPLNF	388
		REWIND 3	CUPLNF	389
390		REWIND 4	CUPLNF	390
		REWIND 5	CUPLNF	391
		REWIND 6	CUPLNF	392
		REWIND 7	CUPLNF	393
		REWIND 8	CUPLNF	394
395		1000 CONTINUE	CUPLNF	395
		GO TO 20	CUPLNF	396
		1001 CALL EXIT	CUPLNF	397
	C		CUPLNF	398
	C		CUPLNF	399
400	C-	FORMATS	CUPLNF	400
	C		CUPLNF	401
		5 FORMAT (* XLIM = *, F12.5, 5X, *RFAC = *, F12.5//)	CUPLNF	402
		7 FORMAT (*MUTUAL RAYLEIGH DISTANCE = (DIAMSUM)SQUARE0/WAVLGTH= *, F12.5, * METERS**//)	CUPLNF	403
405		15 FORMAT(1X, *N1=* I6, 5X, *N2=* I6, 5X, *THEY BOTH SHOULD BE EVEN**//)	CUPLNF	405
		25 FORMAT(1Y, (10E12.5))	CUPLNF	406
		27 FORMAT(1X, *X0-CUT**//)	CUPLNF	407
		29 FORMAT(1X, ** Y0-CUT**//)	CUPLNF	408
410		35 FORMAT(1X, *WAVLGTH, PADT, PADR, AND 70 **4F12.5* METERS RESPECTIVELY**//)	CUPLNF	409
		45 FORMAT(1X, *FULCR ANGLES(PH, TH, RS) OF TR. AND RF. ANTS. RESP. ARE *2F10.4* AND *3F10.4* DEGREES**//)	CUPLNF	411
		55 FORMAT(1X, *X0 RANGES FROM *F12.5* TO *F12.5* IN INCREMENTS OF *F12.5* IN INCREMENTS OF *F12.5* METERS**//)	CUPLNF	413
415		65 FORMAT(1X, *Y0 RANGES FROM *F12.5* TO *F12.5* IN INCREMENTS OF *F12.5* METERS**//)	CUPLNF	415
		75 FORMAT(1X, *THE INTEGRATION VARIABLE KX/K RANGES FROM *F12.5* TO *F12.5* IN INCREMENTS OF *F12.5**//)	CUPLNF	417
		85 FORMAT(1X, *THE INTEGRATION VARIABLE KY/K RANGES FROM *F12.5* TO *F12.5* IN INCREMENTS OF *F12.5**//)	CUPLNF	419
420		87 FORMAT(1X, *THE SPECTRUM IS ZERO FILLED BEYOND SQRT(KX <sup>2</sup> +KY <sup>2</sup> )=K TIME *F12.5**//)	CUPLNF	421
		95 FORMAT(1X, ** THE COUPLING QUOTIENT AT X0=0 AND Y0=0, SUMMED OVER *F12.5* WITHOUT THE FFT, EQUALS *, 2E12.5, * OR *, F10.2, * DB**//)	CUPLNF	423
425		510 FORMAT(1X, ** MAGNITUDE (X0-CUT)**//)	CUPLNF	425
		515 FORMAT(1Y, F12.5* X0=*F12.5)	CUPLNF	426
		610 FORMAT(1X, ** MAGNITUDE (Y0-CUT)**//)	CUPLNF	427
		615 FORMAT(1Y, F12.5* Y0=*F12.5)	CUPLNF	428
430		5000 FORMAT (RA10)	CUPLNF	429
		5001 FORMAT (1H0, RA10)	CUPLNF	430
		5005 FORMAT (1H1)	CUPLNF	431
		5010 FORMAT (6F10.4, 2I5, A10)	CUPLNF	432
		5011 FORMAT (1X, 6F10.4, 2I5, A10)	CUPLNF	433
		5020 FORMAT (8F10.4)	CUPLNF	434
435		5021 FORMAT (1X, 8F10.4)	CUPLNF	435
		5022 FORMAT (7F10.4, I1)	CUPLNF	436
		5023 FORMAT (1X, 7F10.4, I5)	CUPLNF	437
		5030 FORMAT (F10.4, 4I5)	CUPLNF	438
		5031 FORMAT (1X, F10.4, 4I5)	CUPLNF	439
440		5041 FORMAT (/, 5X, RA10, 5X, 2A10)	CUPLNF	440
		END	CUPLNF	441

## B.1.2 SUBROUTINE ANGLGEN

(PKXOXK,PKYOXK,PHI,THETA,PSI,PHIP,THETAP,PSIP,PHIT,THETAT,PHIR,THETAR)

### PURPOSE:

To compute in the coordinate system fixed in each antenna, the far-field angles corresponding to a given direction of the propagation vector in the mutually common xyz coordinate system (see fig. 1).

### METHOD:

Evaluate eqs (13a) and (13b) of the main text.

### ARGUMENTS:

#### Input Parameters (in order of appearance)

- PKXOXK,PKYOXK = x and y components of the normalized propagation vector ( $k_x/k, k_y/k$  of eqs (13)).
- PHI,THETA,PSI = Eulerian angles of the antenna on the left as drawn in figure 2 ( $\emptyset, \theta$ , of eqs (13)).
- PHIP,THETAP,  
PSIP = Eulerian Angles of the antenna on the right as drawn in figure 3 ( $\emptyset^1, \theta^1, \psi^1$  of eq (13)).

#### Output Parameters (in order of appearance)

- PHIT,THETAT = Spherical angles in the coordinate system fixed in the left transmitting antenna, corresponding to the direction  $k_x/k, k_y/k$  ( $\emptyset_A, \theta_A$  of eqs (13) and shown in figure 2).
- PHIR,THETAR = Spherical angles in the coordinate system fixed in the right, receiving antenna, corresponding to the direction  $k_x/k, k_y/k$  ( $\emptyset_p, \theta_p$  of eqs (13) and shown in figure 3).

### SYMBOL DICTIONARY:

#### Variables (in alphabetical order)

- CSPH,CSPHP,  
CSPS,CSPSP,  
CSTH,CSTHP = Cosine of PHI, PHIP, PSI, PSIP, THETA, and THETAP, respectively.
- GAMOXK = Normalized z-component of propagation vector ( $\gamma/k$ ).
- PI =  $\pi = 3.14159\dots$

RD,RN = Denominator and numerator respectively of eq (13b) when computing  $\varnothing_p$ .

R1,R2,R3,R4,  
R41,R42,R5,  
R51,R52,R6,  
R7,R71,R72,  
R8,R81,R82,  
R9 = Intermediate variables used in computing  $\varnothing_p$  and  $\theta_p$  from eqs (13).

SNPH,SNPHP,  
SNPS,SNPSP,  
SINTH,SINTHP = Sine of PHI, PHIP, PSI, PSIP, THETA, and THETAP, respectively.

TD,TN = Denominator and numerator respectively of eq (13b) when computing  $\varnothing_A$ .

T1,T2,T3,T4,  
T41,T42,T5,  
T51,T52,T6,  
T7,T71,T72,  
T8,T81,T82,  
T9 = Intermediate variables used in computing  $\varnothing_A$  and  $\theta_A$  from eqs (13).

XCOSR,XCOST = Right side of eq (13a) for  $\theta_p$  and  $\theta_A$ , respectively.

Functions Inline or within FORTRAN Library (in alphabetical order)

ABS(X) = Absolute value of X.

ACOS(X) = Angle between 0 and  $\pi$  whose cosine is X.

ATAN(X) = Angle between  $-\pi/2$  and  $\pi/2$  whose tangent is X.

ATAN2(X,Y) = Angle between  $-\pi$  and  $\pi$  whose tangent is X/Y.

COS(X) = Cosine of X.

SIN(X) = Sine of X.

SQRT(X) = Square root of X.

List of Complex Quantities

(None)



```

1      SURROITINE ANGLEFN(PKYQXK,PKYQXK,PHI,THETA,PSI,PHIP,THETAP,PSIP, ANGLGEN 1
1      PHIT,THETAT,PHIR,THETAR) ANGLGEN 2
C ANGLGEN 3
C      PKYQXK AND PKYQXK ARE THE X AND Y COMPONENTS OF THE NORMALIZED ANGLGEN 4
C PROPAGATION VECTOR. ANGLGEN 5
5      C      PHIT,THETA,AND PSI ARE THE EULERIAN ANGLES OF THE ROTATED SYSTEM ANGLGEN 6
C OF THE LEFT,TRANSMITTING ANTENNA T WITH RESPECT TO THE AXES ANGLGEN 7
C FIXED IN THE TRANSMITTING ANTENNA. ANGLGEN 8
C      PHIP,THETAP,AND PSIP ARE THE EULERIAN ANGLES OF THE ROTATED ANGLGEN 9
10     C SYSTEM OF THE RIGHT,RECEIVING ANTENNA R WITH RESPECT TO THE ANGLGEN 10
C AXES FIXED IN THE RECEIVING ANTENNA. ANGLGEN 11
C      THETAT AND PHIT ARE THE ANGLES IN THE FIXED COORDINATE SYSTEM OF ANGLGEN 12
C T CORRESPONDING TO THE DIRECTION PKYQXK,PKYQXK. ANGLGEN 13
C      THETAR AND PHIR ARE THE ANGLES IN THE FIXED COORDINATE SYSTEM OF ANGLGEN 14
15     C P CORRESPONDING TO THE DIRECTION PKYQXK,PKYQXK. ANGLGEN 15
C      THETAT AND THETAR RANGE FROM FROM 0 TO PI. ANGLGEN 16
C      PHIT AND PHIR RANGE FROM 0 TO 2PI. ANGLGEN 17
C ANGLGEN 18
C ANGLGEN 19
20     C      PI=4.*ATAN(1.0; ANGLGEN 20
      GAMQXK=SQRT(1.-PKYQXK**2-PKYQXK**2) ANGLGEN 21
C ANGLGEN 22
C      CSTH = COS(THETA) ANGLGEN 23
      SINTH = SIN(THETA) ANGLGEN 24
25     C      CSTHP = COS(THETAP) ANGLGEN 25
      SINTHP = SIN(THETAP) ANGLGEN 26
      CSPTS = COS(PSI) ANGLGEN 27
      SNPTS = SIN(PSI) ANGLGEN 28
      CSRSP = COS(PSIP) ANGLGEN 29
      SNRSP = SIN(PSIP) ANGLGEN 30
30     C      CSPH = COS(PHI) ANGLGEN 31
      SNPH = SIN(PHI) ANGLGEN 32
      CSPHP = COS(PHIP) ANGLGEN 33
      SNPHP = SIN(PHIP) ANGLGEN 34
35     C ANGLGEN 35
C      COMPUTATION OF THETAT AND THETAP. ANGLGEN 36
C ANGLGEN 37
      T1 = SINTH*CSPTS*PKYQXK ANGLGEN 38
      R1 = SINTHP*CSRSP*PKYQXK ANGLGEN 39
40     C      T2 = SINTH*SNPTS*PKYQXK ANGLGEN 40
      R2 = SINTHP*SNRSP*PKYQXK ANGLGEN 41
      T3 = CSTH*GAMQXK ANGLGEN 42
      R3 = CSTHP*GAMQXK ANGLGEN 43
45     C      XCOST=-T1+T2+T3 ANGLGEN 44
      XCONS=-R1-R2+R3 ANGLGEN 45
      THETAT=ACOS(XCOST) ANGLGEN 46
      THETAP=ACOS(XCONS) ANGLGEN 47
C ANGLGEN 48
C      COMPUTATION OF PHIT AND PHIR. ANGLGEN 49
50     C ANGLGEN 50
C ANGLGEN 51
      T41 = CSPH*CSTH*CSPTS ANGLGEN 52
      T42 = SNPH*SNPTS ANGLGEN 53
      T4=(T41-T42)*PKYQXK ANGLGEN 54
55     C ANGLGEN 55
      R41 = CSPHP*CSTHP*CSRSP ANGLGEN 56
      R42 = SNPHP*SNRSP ANGLGEN 57
      R4=(R41-R42)*PKYQXK ANGLGEN 58
C ANGLGEN 59
60     C      T51 = CSPH*CSTH*SNPTS ANGLGEN 60
      T52 = SNPH*CSRSP ANGLGEN 61
      T5=(T51+T52)*PKYQXK ANGLGEN 61
C ANGLGEN 62
C ANGLGEN 63
65     C      R51 = CSPHP*CSTHP*SNRSP ANGLGEN 64
      R52 = SNPHP*CSRSP ANGLGEN 65
      R5=(R51+R52)*PKYQXK ANGLGEN 66
C ANGLGEN 67
C ANGLGEN 68
70     C      T6 = CSPH*SINTH*GAMQXK ANGLGEN 67
      R6 = CSPHP*SINTHP*GAMQXK ANGLGEN 68
      T6=T4-T5+T6 ANGLGEN 69
      R6=R4+R5+R6 ANGLGEN 70
C ANGLGEN 71
C ANGLGEN 72
75     C      T71 = SNPH*CSTH*CSPTS ANGLGEN 73
      T72 = CSPH*SNPTS ANGLGEN 74
      T7=(T71+T72)*PKYQXK ANGLGEN 75
C ANGLGEN 76
C ANGLGEN 77
      R71 = SNPHP*CSTHP*CSRSP ANGLGEN 77

```

		R7? = CSPHP*SNPSP	ANGLGFN	78
		R7=(P7)+R7?)*PKYQXK	ANGLGFN	79
80	C		ANGLGFN	80
		TR1 = CSPH*CSPS	ANGLGFN	81
		TR2 = SNPH*CSH*SNPS	ANGLGFN	82
		TR=(TR1-TR2)*PKYQXK	ANGLGFN	83
85	C		ANGLGFN	84
		RR1 = CSPHP*CSPSP	ANGLGFN	85
		RR2 = SNHP*CSH*SNPSP	ANGLGFN	86
		RR=(RR1-RR2)*PKYQXK	ANGLGFN	87
	C		ANGLGFN	88
		TC = SNPH*SNTH*GAMQXK	ANGLGFN	89
90		R9 = SNHP*SNTH*GAMQXK	ANGLGFN	90
		TN=T7+TR+T9	ANGLGFN	91
		RN=R7-RR+R9	ANGLGFN	92
	C		ANGLGFN	93
	C	CHANGE OF RANGE OF PHIT FROM(-PI,PI) TO (0,2PI).	ANGLGFN	94
95		IF((ABS(TN)+ABS(TD)).EQ.0.) GO TO 10	ANGLGFN	95
		PHIT=ATAN2(TN,TD)	ANGLGFN	96
		IF(PHIT.LT.0.) PHIT=2.*PI+PHIT	ANGLGFN	97
		GO TO 20	ANGLGFN	98
100	10	CONTINUE	ANGLGFN	99
		PHIT=0.	ANGLGFN	100
	20	CONTINUE	ANGLGFN	101
	C	CHANGE OF RANGE OF PHIP FROM(-PI,PI) TO (0,2PI).	ANGLGFN	102
		IF((ABS(RN)+ABS(RD)).EQ.0.) GO TO 30	ANGLGFN	103
		PHIP=ATAN2(RN,RD)	ANGLGFN	104
105		IF(PHIP.LT.0.) PHIP=2.*PI+PHIP	ANGLGFN	105
		GO TO 40	ANGLGFN	106
	30	CONTINUE	ANGLGFN	107
		PHIP=0.	ANGLGFN	108
110	40	CONTINUE	ANGLGFN	109
		RETURN	ANGLGFN	110
		END	ANGLGFN	111

### B.1.3 SUBROUTINE FINDFF(IDAYHR,LUIN,LUA,LUOZ,LUOE,DATA,NRX2,NCOL,FFY,FFZ,STOR)

#### PURPOSE:

To read from an input file, spectrum or far-field data whose coordinates are  $k_x$  and  $k_y$  referred to the antenna's preferred coordinate system and from this produces a file containing far fields whose coordinates are specified by the angles specified on a second input file.

#### ARGUMENTS:

IDAYHR = File identifier for file on which far field resides.  
LUIN = Logical unit on which far field or spectrum resides.  
LUA = Logical unit on which angle information resides.  
LUOY = Logical unit on which y-component of far field is to be written.  
LUOZ = Logical unit on which z-component of far field is to be written.  
DATA = Two-dimensional array in which this input far field is stored, included in argument list for dimensioning purposes only.  
NRX2 = Twice the number of rows in DATA.  
NCOL = Number of columns in DATA.  
FFY = y-component of far field, included in argument list for storage allocation purposes only.  
FFZ = z-component of far field, included in argument list for storage allocation purposes only.  
STOR = Intermediate array, included for storage allocation purposes only.

#### METHODS:

The subroutine reads the first record of the file containing the far-field or spectrum data from unit LUIN and compares the eighth word of the record with IDAYHR in order to make sure the proper data file is used. If LUIN contains the incorrect file, execution terminates. After correct file verification, the entire file is read in and stored in array DATA. Because the input data exist in polar form, a conversion to rectangular form is also performed in the operation of filling DATA.

All data transfers use FORTRAN unformatted READ and WRITE operations.

The desired far-field angles are assumed to be stored on unit LUA. These are read one record at a time into complex vector FFY with the real part containing the  $\phi$ -coordinate and the imaginary part, the  $\theta$ -coordinate. For each element of the vector FFY, the angles stored are used to locate the nearest far-field point in the array DATA. The z-component is then calculated and stored in FFZ. When all angles in FFY have been changed to the corresponding far-field values, the vectors FFY and FFZ are written out as a record on units LUOY and LUOZ, respectively.

The correct point in the far-field array is found by the following procedure. Calculate the reference index for the x and y directions by

$$I_c = \text{integer part of } \frac{k_y}{\Delta k_x}$$

$$I_r = \text{integer part of } \frac{k_x}{\Delta k_y}$$

where

$$k_x = k \sin\theta \cos\phi$$

$$k_y = k \sin\theta \sin\phi$$

and  $\Delta k_x$  and  $\Delta k_y$  are the far-field  $k_x$  and  $k_y$  increments. These increments are given by

$$\Delta k_x = \frac{2\pi}{N_x \delta_x}$$

$$\Delta k_y = \frac{2\pi}{N_y \delta_y}$$

where  $N_x$ ,  $N_y$  are the number of x or y points and  $\delta_x$ ,  $\delta_y$  are the x or y near-field spacing.

The far-field increments are given in terms of near-field spacings because it is assumed that the far field is obtained either by a near-field scan or the PO model program POMODL (see appendix A), which calculates its far-field array based on a desired near-field spacing.

The row and column indices  $I_r$  and  $I_c$  specify "lower left-hand corner" of the square in  $(k_x, k_y)$  space which contains the point specified by the angles  $\theta$  and  $\phi$ . The fractional part of

$$\frac{k_x}{\Delta k_x} \quad \text{or} \quad \frac{k_y}{\Delta k_y}$$

is used to determine which corner of the square lies closest to  $\theta$  and  $\phi$ .

The z-component is found from the relationship

$$E_z = E_y \tan\theta \sin\emptyset$$

Because the far-field array DATA may not contain values for all angles, a test is performed to determine if DATA does, in fact, contain a far-field value at the requested  $\theta$  and  $\emptyset$ . If it does not, the y- and z-components are set to zero.

SYMBOL DICTIONARY:

ANGLE	= Intermediate variable, phase angle of far-field input data.
DATA	= Far-field array as a function of antenna's $k_x$ , $k_y$ system.
DCOL	= Fractional part of FCOL.
DLKX	= Input far-field point spacing in $k_x$ direction.
DLKY	= Input far-field point spacing in $k_y$ direction.
DROW	= Fractional part of FROW.
DTR	= $\pi/180$ = degree to radian conversion factor.
FCOL	= $k_x/\Delta k_x$ .
FFY	= y-component of far field, also used as temporary storage for the far-field angles.
FKSQ	= $k_x^2 + k_y^2$ .
FKX	= $k_x$ = x-component of propagation vector.
FKXMAX	= Maximum value of $k_x$ for which there are far-field data.
FKY	= $k_y$ = y-component of propagation vector.
FKYMAX	= Maximum value of $k_y$ for which there are far-field data.
FROW	= $k_y/\Delta k_y$ .
I	= DO loop index.
ICOL	= Input DO loop column index, also column index for far field.
ID	= Identification array for far-field data.
IDAYHR	= Far-field file identification = ID(8) for correct file.
IFC	= Integer part of FCOL.
IFR	= Integer part of FROW.
IROW	= Input DO loop row index, also intermediate variable.
IR2X2	= 2 x IROW.
ISPECT	= 1 if DATA contains spectrum rather than far field.
J	= Search loop index.
L	= Input or output DO loop index.
NROW	= Number of rows of input far-field data.
PHI	= $\emptyset$ = angle in far field.
PI	= $\pi$ = 3.14159.....
PIX2	= $2\pi$ .

TAMP = Intermediate variable, amplitude of input far-field data.  
THETA =  $\theta$  = angle in far field.  
XNZ =  $\text{Cos}\theta$ .

```

1      SUBROUTINE FINDEF (IDAYHR, LUIN, LUA, LUDY, LUOZ, DATA, NRX2, FINDEF 1
      1 NCCL, FFY, FFZ, STOR) FINDEF 2
C- FINDEF 3
C- THIS SUBROUTINE READS FAR-FIELD OR SPECTRUM DATA FROM LUIN AND FINDEF 4
5 C- STORES IT IN ARRAY DATA. ANGLES CORRESPONDING TO FAR-FIELD FINDEF 5
C- DIRECTIONS IN THE ANTENNAS COORDINATE SYSTEM ARE READ IN FROM LUA. FINDEF 6
C- DATA IS SEARCHED FOR THE CLOSEST POINT AND THE Y-COMPONENT OF THE FINDEF 7
C- FIELD AT THE GIVEN ANGLE IS USED TO COMPUTE THE Z-COMPONENT. FINDEF 8
C- THESE FIELD COMPONENTS ARE WRITTEN ON LUDY AND LUOZ. FINDEF 9
10 C- FINDEF 10
      COMPLEX FFY(1), FFZ(1) FINDEF 11
      COMMON /FAR/ N1, N2, NY, NLY, DLY, XK, ISPECT FINDEF 12
      DIMENSION DATA(NRX2, NCCL), STOR(1), ID(10) FINDEF 13
C- FINDEF 14
15 C- MISCELLANEOUS FINDEF 15
C- FINDEF 16
      PRINT 1020, LUIN, LUA, LUDY, LUOZ, NRX2, NCCL FINDEF 17
1020 FORMAT (6I20) FINDEF 18
      TSE = 0 FINDEF 19
20 PI = 4.*ATAN(1.) FINDEF 20
      PIY2 = 2.*PI FINDEF 21
      DTP = PI/180. FINDEF 22
      NRQW = NRX2/2 FINDEF 23
      DLKY = RIY2/NRQW/DLY FINDEF 24
25 DLKX = PIY2/NCCL/DLY FINDEF 25
      FKXMAX = DLKX*(NCCL - 1)/2 FINDEF 26
      FKYMAX = DLKY*(NRQW - 1)/2 FINDEF 27
      PRINT 1000, DLKX, DLKY, FKXMAX, FKYMAX, XK FINDEF 28
1000 FORMAT(1X, 5G20.5) FINDEF 29
30 C- FINDEF 30
C- FIND CORRECT FAR-FIELD FILE ON LOGICAL UNIT LUIN. FINDEF 31
C- FINDEF 32
120 READ(LUIN) (ID(I), I = 1, 10) FINDEF 33
      PRINT 1510, ID FINDEF 34
35 IF (ID(8) .EQ. IDAYHR) GO TO 130 FINDEF 35
      IF (FDF(LUIN)) 125, 130 FINDEF 36
125 PRINT 1530 FINDEF 37
      CALL FXIT FINDEF 38
130 CONTINUE FINDEF 39
40 C- FINDEF 40
C- READ FAR-FIELD INTO ARRAY DATA. FINDEF 41
C- FINDEF 42
      DO 140 ICCL = 1, NCCL FINDEF 43
        READ(LUIN) (STOR(I), I = 1, NRX2) FINDEF 44
45 DO 150 IRQW = 1, NPCW FINDEF 45
          IPX2 = IRQW*2 FINDEF 46
          TAMP = STOR(IPX2 - 1) FINDEF 47
          ANGLE = STOR(IPX2)*DTP FINDEF 48
          DATA(IPX2 - 1, ICCL) = TAMP*COS(ANGLE) FINDEF 49
50 DATA(IPX2, ICCL) = TAMP*SIN(ANGLE) FINDEF 50
150 CONTINUE FINDEF 51
140 CONTINUE FINDEF 52
C- FINDEF 53
C- REPLACE VALUES OF ANGLES WITH CORRESPONDING FAR-FIELD. FINDEF 54
55 C- FINDEF 55
C- FINDEF 56
      REWIND LUA FINDEF 57
      READ (LUIN) FINDEF 58
      IF (FDF(LUIN)) 500, 600 FINDEF 59
60 500 CONTINUE FINDEF 60
      BACKSPACE LUIN FINDEF 61
      600 CONTINUE FINDEF 61

      DO 200 I = 1, N1 FINDEF 62
        READ(LUA) (FFY(L), L = 1, N2) FINDEF 63
65 DO 300 J = 1, N2 FINDEF 64
          PHI = REAL(FFY(J)) FINDEF 65
          THETA = AIMAG(FFY(J)) FINDEF 66
          IF (THETA .LT. 0.) GO TO 310 FINDEF 67
C- FINDEF 68
C- FIND THE INDICES FOR THE ARRAY DATA WHICH CORRESPOND TO THE FINDEF 69
70 C- COORDINATES CLOSEST TO THE DESIRED THETA AND PHI VALUES. FINDEF 70
C- FINDEF 71
          FKX = XK*SIN(THETA)*COS(PHI) FINDEF 72
          FKY = XK*SIN(THETA)*SIN(PHI) FINDEF 73
          FKSQ = FKX*FKX + FKY*FKY FINDEF 74
          IF (FKSQ .GT. XK*XK) GO TO 310 FINDEF 75
          IF (ABS(FKX) .GE. FKXMAX) GO TO 310 FINDEF 76
          IF (ABS(FKY) .GE. FKYMAX) GO TO 310 FINDEF 77

```

	FRQW = FKX/DLKY + NRQW/2	FINDFF	78
	IFR = FRQW	FINDFF	79
80	DRQW = FRQW - IFR	FINDFF	80
	IRQW = IFR + 1	FINDFF	81
	IF (DRQW .LT. .5) IRQW = IFR	FINDFF	82
	FCQL = FKX/DLKX + NCQL/2	FINDFF	83
	IFC = FCQL	FINDFF	84
85	DCQL = FCQL - IFC	FINDFF	85
	ICQL = IFC + 1	FINDFF	86
	IF (DCQL .LT. .5) ICQL = IFC	FINDFF	87
	IP2X2 = IRQW*2	FINDFF	88
	FFY(J) = CMPLX(DATA(IP2X2 - 1, ICQL), DATA(IP2X2, ICQL))	FINDFF	89
90	FFZ(J) = -TAN(THETA)*SIN(PHI)*FFY(J)	FINDFF	90
	IF (ISPECT .NE. 1) GO TO 300	FINDFF	91
	YNZ = SQRT(XK*XK - FKSQ)/XK	FINDFF	92
	FFY(J) = FFY(J)*XNZ	FINDFF	93
	FFZ(J) = FFZ(J)*XNZ	FINDFF	94
95	GO TO 300	FINDFF	95
	FFY(J) = (0., 0.)	FINDFF	96
	FFZ(J) = (0., 0.)	FINDFF	97
	300 CONTINUE	FINDFF	98
	WRITE(LUDY) (FFY(L), L = 1, N2)	FINDFF	99
100	WRITE(LUDZ) (FFZ(L), L = 1, N2)	FINDFF	100
	200 CONTINUE	FINDFF	101
	REWIND LUDY	FINDFF	102
	REWIND LUDZ	FINDFF	103
	RETURN	FINDFF	104
105	1510 FORMAT(1X, F410, 2I5)	FINDFF	105
	1520 FORMAT(* FILE #, I5, # SKIPPED ON LU #, I5)	FINDFF	106
	1530 FORMAT(* FILE NOT FOUND, EXECUTION ABORTED*)	FINDFF	107
	END	FINDFF	108



#### B.1.4 SUBROUTINE VECTGEN (FOX,FOY,FOZ,PH,THET,PS,FX,FY, FZ)

##### PURPOSE

Given the components (FOX,FOY,FOZ) of a complex vector in a right-hand rectangular coordinate system, find the transformed components (FX,FY,FZ) of that vector in a second coordinate system formed by rotation of the first through the Eulerian angles (PH,THET,PS).

##### METHOD:

Use the transformation given by eq (18) of the main text.

##### ARGUMENTS:

###### Input Parameters (in order of appearance)

FOX,FOY,

FOZ = x, y, z rectangular components of the given complex vector in the unrotated coordinate system.

PH,THET,PS = Eulerian angles of the rotated coordinate system.

###### Output Parameters (in order of appearance)

FX,FY,FZ = Transformed x, y, z rectangular components of the given complex vector in the rotated coordinate system.

##### SYMBOL DICTIONARY:

###### Variables

A11,A12,

A13,A21,

A22,A23,

A31,A32,

A33 = The nine elements of the 3 x 3 matrix on the right side of eq (18).

CSPH,CSPS,

CSTH = Cosine of PH, PS, and THET, respectively.

SNPH,SNPS,

SNTH = Sine of PH, PS, and THET, respectively.

Functions Inline within FORTRAN Library

COS(X) = Cosine of X.

SIN(X) = Sine of X.

List of Complex Quantities

FX, FY, FZ, FOX, FOY FOZ

1		SUBROUTINE VECTGEN(FOX,FOY,FOZ,PH,THET,PS, FX,FY,FZ)	VECTGEN	1
	C	IF THE COMPONENTS (FOX,FOY,FOZ) OF A COMPLEX VECTOR ARE GIVEN IN	VECTGEN	2
	C	A RIGHT-HANDED RECTANGULAR COORDINATE SYSTEM, AND A SECOND	VECTGEN	3
	C	COORDINATE SYSTEM IS FORMED BY ROTATION THROUGH EULERIAN ANGLES	VECTGEN	4
5	C	(PH,THET,PS), THEN (FX,FY,FZ) ARE THE COMPONENTS OF THAT VECTOR	VECTGEN	5
	C	IN THIS SECOND ROTATED SYSTEM.	VECTGEN	6
	C		VECTGEN	7
		COMPLEX FOY,FOY,FOZ	VECTGEN	8
		COMPLEX FX,FY,FZ	VECTGEN	9
10	C		VECTGEN	10
	C		VECTGEN	11
	C	COMPUTATION OF THE NINE ELEMENTS OF THE ROTATIONAL	VECTGEN	12
	C	TRANSFORMATION MATRIX.	VECTGEN	13
	C		VECTGEN	14
15		CSPH = COS(PH)	VECTGEN	15
		SNPH = SIN(PH)	VECTGEN	16
		CSPS = COS(PS)	VECTGEN	17
		SNPS = SIN(PS)	VECTGEN	18
		CSTH = COS(THET)	VECTGEN	19
20		SNTH = SIN(THET)	VECTGEN	20
	C		VECTGEN	21
		A11 = CSPH*CSTH*CSPS - SNPH*SNPS	VECTGEN	22
		A12 = SNPH*CSTH*CSPS + CSPH*SNPS	VECTGEN	23
		A13 = - SNTH*CSPS	VECTGEN	24
25		A21 = -(CSPH*CSTH*SNPS + SNPH*CSPS)	VECTGEN	25
		A22 = -SNPH*CSTH*SNPS + CSPH*CSPS	VECTGEN	26
		A23 = SNTH*SNPS	VECTGEN	27
		A31 = CSPH*SNTH	VECTGEN	28
		A32 = SNPH*SNTH	VECTGEN	29
30		A33 = CSTH	VECTGEN	30
	C		VECTGEN	31
		FX=A11*FOX+A12*FOY+A13*FOZ	VECTGEN	32
		FY=A21*FOX+A22*FOY+A23*FOZ	VECTGEN	33
		FZ=A31*FOX+A32*FOY+A33*FOZ	VECTGEN	34
35		RETURN	VECTGEN	35
		END	VECTGEN	36

### B.1.5 SUBROUTINE MINMAX(Z,ZMIN,ZMAX,LEX,LEY)

#### PURPOSE:

To determine the maximum and minimum values stored in the array Z.

#### ARGUMENTS:

Z is a two-dimensional array which is to be searched for its maximum and minimum values.

ZMIN contains the minimum value in the array Z on exit.

ZMAX contains the maximum value in the array Z on exit.

LEX is the number of rows in Z.

LEY is the number of columns in Z.

#### METHODS:

Array Z has dimensions (LEX,LEY). Initially ZMIN and ZMAX are set equal to Z(1,1). Each value of Z is tested to determine if it is less than ZMIN or greater than ZMAX. If either condition is satisfied, ZMIN or ZMAX is appropriately changed.

#### SYMBOL DICTIONARY:

I	= Row DO loop index.
J	= Column DO loop index.
TZ	= Temporary variable, Z(I,J).

1	SUBROUTINE MINMAX(Z, ZMIN, ZMAX, LEX, LEY)	MINMAX	1
	DIMENSION Z(LEX,LEY)	MINMAX	2
	ZMIN=Z(1,1)      *    ZMAX=Z(1,1)	MINMAX	3
5	DO 120 I = 1, LEX	MINMAX	4
	DO 120 J = 1, LEY	MINMAX	5
	TZ = Z(I,J)	MINMAX	6
	IF (TZ .LT. ZMIN) ZMIN = TZ	MINMAX	7
	IF (TZ .GT. ZMAX) ZMAX = TZ	MINMAX	8
10	CONTINUE	MINMAX	9
	RETURN	MINMAX	10
	END	MINMAX	11

## APPENDIX B.2 SAMPLE PROGRAM INPUT AND OUTPUT

Illustrated below is a typical input card deck for program CUPLNF. Far-field data for the two antennas were generated using POMODL. The output obtained for one of these runs is illustrated in Appendix A.2. The output produced by CUPLNF is reproduced on the following pages.

0.          0.          0.          0.          0.          0.											
28.	1.	.375	0.	10.	150.	54	540501790915				
27.	1.	.75	0.	10.	0.	54	540501791829				
0.	1	1	1	1							
4.0	5.0	.9375	.9375	.9375	.9375	1.		1			
SIMULATION TEST											
STUBENBRUCH, G. F. X-3927											
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80											
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80											
00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000											
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80											
11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111											
22 22222 22222 22222 2222 22222 22222 22222 22222 22222 22222 22222 22222 22222 22222 22222 22222 22222 22222											
3 33333 3333 3 33 3333 33333 33323 33333 33333 33323 33333 33333 33333 33333 33333 33333 33333 33333 33333 33333											
44 44444 4444444 444 4444444444 44444444444 44444444444 44444444444 44444444444 44444444444 44444444444 44444444444 44444444444 44444444444 44444444444 44444444444											
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80											
5555 55555 55555 5555 5555 55555 55555 55555 55555 55555 55555 55555 55555 55555 55555 55555 55555 55555 55555											
66666 66666 666666 666 6666666666 66666666666 66666666666 66666666666 66666666666 66666666666 66666666666 66666666666 66666666666 66666666666 66666666666											
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80											
77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777 77777											
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80											
88888 88888 888 88 88 88888 88888 88888 88888 88888 88888 88888 88888 88888 88888 88888 88888 88888 88888 88888											
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80											
99999 9999 99999 9999 9999 99999 9999 9999 9999 99999 99999 99999 99999 99999 99999 99999 99999 99999 99999											
ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT											

GENERAL DATA CARD IO-5  
IBM CORPORATION, ARMONK, N.Y.

MUTUAL POLARIZATION DISTANCE = (COS(SUM))SQUARED/WAVELENGTH = 67.54672 METERS

N1= 54 N2= 54 THEY BOTH SHOULD BE EVEN

PHYSICAL OPTICS SIMULATION TEST NO. 1 3 82.447 5 7 82.447 128 64 83.834 128 64  
1 2.4180 2.4180 0501791229 64 128

PHYSICAL OPTICS SIMULATION TEST NO. 2 4 82.467 6 8 82.467 128 64 83.834 128 64  
1 2.4180 2.4180 0501790916 64 128  
XLEN = 1.00000 REAC = 1.00000

N1= 214 N2= 214 THEY BOTH SHOULD BE EVEN

WAVELENGTH,PART,SECTOR,AND Z0 = .07495 .37500 5.00000 METERS RESPECTIVELY

FILLED ANGLES(DU,TH,PS) OF TP, AND OF, ANTS, RESP. ARE 0.0000 10.0000 0.0000 AND 0.0000 10.0000 180.0000 DEGREES

X0 RANGES FROM -2.24844 TO 2.22762 IN INCREMENTS OF .02082 METERS

Y0 RANGES FROM -2.24844 TO 2.22762 IN INCREMENTS OF .02082 METERS

THE INTERPOLATION VARIABLE KX/K RANGES FROM -1.80000 TO 1.78333 IN INCREMENTS OF .01667

THE INTERPOLATION VARIABLE KY/K RANGES FROM -1.80000 TO 1.78333 IN INCREMENTS OF .01667

THE SPECTRUM IS ZERO FILLED BEYOND SORT(KY2+KX2)=K TIMES .45000

THE COUPLING QUOTIENT AT XC=0 AND YC=0, SUMMED DIRECTLY WITHOUT THE FFT, EQUALS -.46336E-03 -.22632E-03 OR -65.75 DB

XO=UNIT

.11267E-02 -.35891E-04 .15799E-03 -.55533E-04 .22520E-03 -.19027E-05 .30552E-03 .54126E-04 .38612E-03 .39992E-04  
.45237E-03 .06570E-04 .46055E-03 .49333E-04 .49091E-03 -.19572E-04 .45034E-03 -.94504E-04 .37403E-03 -.14856E-03  
.27540E-03 -.15980E-03 .17333E-03 -.11893E-03 .87734E-04 -.31861E-04 .31933E-04 .81711E-04 .18404E-04 .13626E-03  
.24611E-04 .28880E-03 .75533E-04 .34414E-03 .11860E-03 .36029E-03 .15194E-03 .34302E-03 .16915E-03 .30178E-03  
.17220E-03 .24370E-03 .17074E-03 .16823E-03 .16351E-03 .81126E-04 .15771E-03 .26211E-04 .14156E-03 .14553E-03  
.09803E-03 -.26023E-03 .19674E-04 .34734E-03 -.10484E-03 .37628E-03 -.25640E-03 .32490E-03 -.40374E-03 .18488E-03  
- .50370E-03 .31934E-04 .51538E-03 .29340E-03 .41362E-03 .55265E-03 .20045E-03 .75788E-03 .92512E-04 .86352E-03  
.41012E-03 .83084E-03 .64495E-03 .67979E-03 .86312E-03 .35942E-03 .89782E-03 .36370E-04 .77680E-03 .35689E-03  
.51955E-03 -.72114E-03 .13881E-03 .99933E-03 .29766E-03 .11449E-02 .74493E-03 .11283E-02 .11513E-02 .94140E-03  
- .14685E-02 .50314E-03 .14534E-02 .13324E-03 .16710E-02 .40209E-03 .14987E-02 .94568E-03 .11318E-02 .14308E-02  
- .58851E-03 .17937E-02 .87475E-04 .19750E-02 .82941E-02 .19369E-02 .15544E-02 .16548E-02 .21745E-02 .11484E-02  
.26070E-03 .44304E-02 .27267E-02 .35062E-03 .27031E-02 .12634E-02 .23257E-02 .20740E-02 .16938E-02 .27239E-02  
.86427E-03 .31319E-02 .84590E-04 .22459E-02 .10618E-02 .30504E-02 .19729E-02 .25675E-02 .27302E-02 .18494E-02  
- .32415E-02 .97158E-03 .28178E-02 .21270E-04 .34779E-02 .91238E-03 .31503E-02 .17468E-02 .25716E-02 .24129E-02  
- .18015E-02 .28604E-02 .91512E-03 .30631E-02 .32635E-06 .30393E-02 .86414E-03 .27527E-02 .16006E-02 .23078E-02  
.21515E-02 .17424E-02 .24805E-02 .11178E-02 .25764E-02 .49045E-02 .24529E-02 .95079E-04 .21450E-02 .60794E-03  
.17020E-02 .10294E-02 .11821E-02 .13500E-02 .84070E-03 .15836E-02 .12248E-03 .16676E-02 .33717E-03 .16630E-02  
- .71602E-03 .15541E-02 .16073E-02 .13613E-02 .12091E-02 .11010E-02 .13255E-02 .80258E-03 .113700E-02 .49407E-03

-113555E-02 --12977E-03 --12977E-02 --66950E-04 --12126E-02 --29595E-03 --11148E-02 --10153E-02 --63727E-03  
-91981E-03 --76041E-03 --82083E-03 --74005E-03 --89007E-03 --64917E-03 --83926E-03 --5514E-03 --79069E-03  
-46119E-03 --37479E-03 --62599E-03 --30641E-03 --52750E-03 --26933E-03 --42447E-03 --25768E-03 --31539E-03  
-28223E-03 --13607E-03 --63631E-04 --39694E-03 --75909E-04 --46336E-03 --22532E-03 --52077E-03 --36278E-03  
-56107E-03 --47592E-03 --56332E-03 --58681E-03 --61148E-03 --57264E-03 --61594E-03 --54352E-03 --60879E-03  
-45008E-03 --55947E-03 --44527E-03 --37730E-03 --53586E-03 --29835E-03 --50177E-03 --21154E-03 --45906E-03  
-12221E-03 --49550E-03 --37747E-04 --32913E-04 --26673E-03 --80439E-04 --18448E-03 --97300E-04 --94664E-04  
77324E-04 --32350E-04 --18945E-04 --10923E-03 --75443E-04 --22056E-03 --1994E-03 --34164E-03 --42647E-03  
-46018E-03 --63824E-03 --62824E-03 --52664E-03 --77012E-03 --51044E-03 --84604E-03 --95715E-03 --33791E-03  
-90848E-03 --13443E-03 --99776E-03 --25440E-04 --94902E-03 --15228E-03 --85277E-03 --71041E-03 --53515E-03  
-53000E-03 --70904E-03 --32720E-03 --83023E-03 --12240E-03 --90497E-02 --61759E-04 --20441E-03 --83683E-03  
29047E-03 --68820E-03 --31356E-03 --47701E-03 --23244E-03 --25646E-04 --80101E-04 --80101E-04 --25630E-03  
-41387E-04 --42115E-03 --92905E-03 --59055E-03 --27146E-03 --53145E-03 --27032E-03 --43481E-03 --24746E-03  
-10859E-03 --8448E-05 --8448E-05 --28819E-04 --61711E-04 --57821E-03 --23233E-03 --80326E-03 --93481E-03  
58819E-03 --96920E-03 --71127E-03 --84325E-03 --76822E-03 --63390E-03 --74011E-03 --35518E-03 --62127E-03 --49200E-04  
42013E-03 --26143E-03 --16007E-03 --48222E-03 --12386E-03 --68856E-03 --39142E-03 --72877E-03 --48663E-03 --72155E-03  
-7394E-03 --53413E-03 --78244E-03 --48093E-03 --74349E-03 --28264E-03 --63737E-03 --58261E-04 --48663E-03 --72155E-03  
-22498E-03 --37885E-02 --15245E-02 --41467E-03 --11851E-04 --34187E-03 --15733E-03 --38915E-03 --30911E-03 --26629E-03  
42793E-03 --92568E-04 --50235E-03 --10729E-03 --53655E-03 --30683E-03 --49759E-03 --48043E-03 --39439E-03 --60458E-03  
24457E-03 --65865E-03 --91317E-04 --62800E-03 --54679E-04 --50550E-03 --12490E-03 --29810E-03 --11467E-03 --28987E-04  
-20070E-04 --26274E-03 --32001E-03 --52788E-03 --29528E-03 --71775E-03 --43024E-03 --79698E-03 --49915E-03 --75412E-03  
44949E-03 --64619E-03 --29220E-03 --39474E-03 --24337E-03 --17446E-03 --70615E-04 --24885E-05 --93735E-04 --10093E-03  
-22498E-03 --11147E-03 --30845E-03 --59772E-04 --34144E-03 --32468E-03 --13792E-03 --43792E-03 --28206E-03 --19117E-03  
-21021E-03 --18857E-03 --12590E-03 --13246E-03 --40505E-04 --42574E-04 --35393E-04 --53647E-04 --93100E-04 --12841E-03  
12720E-03 --16242E-03 --13759E-03 --15042E-03 --12864E-03 --99865E-04 --11025E-03 --29576E-04 --94332E-04 --37200E-04  
62240E-04 --29650E-04

Y0-CUT

-55417E-05 --21347E-04 --99645E-05 --91689E-05 --65039E-04 --31224E-04 --21575E-05 --14608E-03 --18425E-03 --36074E-04  
64589E-05 --10784E-03 --16924E-03 --39430E-04 --28731E-04 --11604E-03 --32405E-04 --43324E-04 --12897E-03 --30019E-04  
72489E-05 --11986E-04 --11450E-03 --26922E-04 --67694E-04 --10592E-03 --11211E-03 --23010E-04 --13987E-03 --28887E-04  
25081E-04 --14005E-05 --12682E-03 --43922E-04 --76938E-05 --92577E-04 --18936E-04 --36743E-04 --85875E-04 --72925E-04  
-30594E-04 --12545E-03 --29574E-03 --26478E-04 --11359E-03 --21339E-03 --14977E-03 --36668E-04 --83343E-04  
68689E-04 --40433E-05 --20595E-04 --40696E-04 --84512E-04 --23276E-04 --87832E-04 --35491E-04 --10784E-03 --65260E-04  
-7877E-04 --12563E-04 --13500E-03 --1730E-04 --42321E-04 --23221E-04 --68445E-04 --22950E-04 --11237E-03 --50569E-05  
-11511E-04 --36925E-04 --14452E-03 --19246E-03 --12428E-04 --13121E-03 --17237E-03 --17189E-03 --74755E-04  
-20674E-05 --21890E-03 --15757E-03 --29544E-04 --46292E-04 --85397E-05 --15496E-03 --17156E-04 --25954E-04 --14243E-03  
20281E-03 --13604E-03 --25594E-04 --36622E-04 --23746E-03 --11953E-03 --79469E-04 --13280E-04 --21309E-04 --13664E-04  
-42459E-05 --26114E-04 --87635E-04 --18644E-03 --17295E-03 --41967E-04 --25533E-05 --18807E-04 --15336E-03 --34623E-04  
14813E-03 --39592E-04 --27594E-04 --33009E-04 --33009E-04 --33009E-04 --33009E-04 --33009E-04 --33009E-04 --33009E-04  
16011E-03 --76542E-05 --74519E-04 --43017E-04 --15938E-03 --61126E-04 --10316E-03 --46355E-04 --12971E-03 --14929E-03  
-21395E-03 --63929E-04 --12710E-03 --78922E-04 --13238E-03 --57073E-04 --10343E-03 --14139E-03 --11670E-03 --59725E-05  
92329E-04 --60484E-04 --13840E-03 --66902E-03 --38997E-05 --36022E-03 --17071E-04 --34811E-03 --62855E-04  
-113409E-03 --15456E-03 --37697E-03 --23081E-03 --13766E-04 --94200E-04 --35292E-04 --24669E-03 --15374E-03  
82024E-03 --64122E-03 --27723E-03 --47084E-04 --72286E-03 --17838E-03 --12903E-02 --18413E-02 --53122E-03 --98987E-03  
-66085E-03 --63775E-02 --10225E-02 --22653E-02 --46422E-03 --81491E-03 --23102E-03 --39538E-03 --15070E-03 --29012E-03  
-21177E-04 --26592E-02 --17331E-03 --73331E-03 --24836E-02 --19454E-03 --50307E-03 --31874E-02 --40104E-03 --65749E-03  
-10672E-02 --17176E-03 --54742E-02 --10462E-02 --25336E-02 --14024E-02 --13796E-02 --17785E-02 --47048E-03 --36981E-02  
-30620E-02 --57921E-02 --48911E-05 --14388E-02 --10668E-02 --76036E-03 --61022E-02 --25952E-02 --60525E-03 --21291E-02  
-85011E-03 --15126E-02 --17793E-02 --65607E-02 --33694E-02 --29863E-02 --19133E-02 --22275E-02 --19121E-02 --12467E-02  
41904E-02 --63732E-02 --57906E-03 --11970E-02 --29307E-03 --14310E-03 --44031E-02 --20638E-03 --91679E-03 --20642E-02  
-12217E-03 --14375E-02 --31634E-03 --10021E-02 --24836E-02 --84949E-03 --18450E-03 --32208E-03 --26384E-03 --99569E-04  
-34673E-02 --53230E-03 --44547E-04 --10647E-04 --19454E-03 --50307E-03 --18058E-02 --45793E-03 --55659E-03  
45423E-03 --23572E-03 --21352E-03 --37218E-04 --21275E-03 --87245E-03 --21504E-03 --18423E-03 --21438E-03 --21942E-03  
80638E-04 --91182E-04 --43918E-05 --15429E-04 --15776E-03 --22101E-03 --17331E-03 --24231E-03 --18953E-03 --19343E-03  
55819E-03 --17508E-03 --20792E-03 --23123E-03 --99733E-04 --20768E-04 --86085E-04 --36636E-04 --19461E-03  
23142E-04 --12435E-03 --10856E-04 --16724E-03 --70931E-05 --57166E-04 --56075E-04 --18208E-03 --35161E-04 --18102E-03  
80989E-04 --13484E-03 --97020E-04 --15457E-03 --12321E-03 --61845E-04 --10259E-03 --12126E-03 --11689E-03 --63392E-04  
20174E-03 --87959E-04 --12288E-03 --12459E-04 --12288E-03 --12459E-04 --12459E-04 --71509E-04 --67974E-04 --15518E-03  
27456E-07 --34849E-04 --13272E-04 --10396E-03 --88632E-04 --64901E-05 --85928E-04 --24665E-03 --93349E-04 --10684E-04  
-64419E-04 --13347E-03 --18450E-03 --86269E-04 --96767E-04 --10153E-03 --12341E-03 --11247E-03 --51363E-04 --46130E-05



-16040E-04 -.22847E-03 .27348E-04 -.14972E-04 .25220E-04 -.13831E-03 .18073E-03 .75231E-04 .12240E-03 -.92773E-04  
 .15640E-03 .11029E-03 .72684E-04 .29312E-04 .68376E-04 .22752E-03 -.61162E-05 .30066E-04 .24876E-04 .20403E-03  
 -9.2732E-04 -.22847E-04 .27348E-04 -.14972E-04 .25220E-04 -.13831E-03 .18073E-03 .75231E-04 .12240E-03 -.92773E-04  
 .59589E-04 .73595E-04 .13537E-05 .40575E-04 .80506E-04 .10811E-03 -.36233E-04 -.95755E-04 .28121E-04 .22806E-04  
 -.86107E-04 -.10141E-02 .89388E-04 .21535E-04 .55319E-04 .14983E-03 .48058E-04 -.28324E-04 .51470E-04 .15923E-03  
 .13254E-03 .20820E-04 .27999E-05 -.79177E-04 .14744E-03 .56136E-04 -.78423E-04 -.26494E-04 -.29046E-04 .67557E-04  
 -11.081E-03 -.29736E-05 .61880E-04 .61880E-04 .58896E-04 -.11727E-03 .58553E-04 .19126E-04 -.36338E-04 .14587E-03  
 .13794E-03 .43024E-04 .62135E-04 -.61013E-04 .15510E-03 .57726E-04 -.44489E-04 .24097E-05 .40143E-05 .15301E-03  
 -.85592E-04 .21655E-04 -.25677E-04 .12733E-03 -.15557E-03 .87549E-04 .45179E-04 .78945E-04 -.21741E-04  
 .11644E-03 -.56383E-04

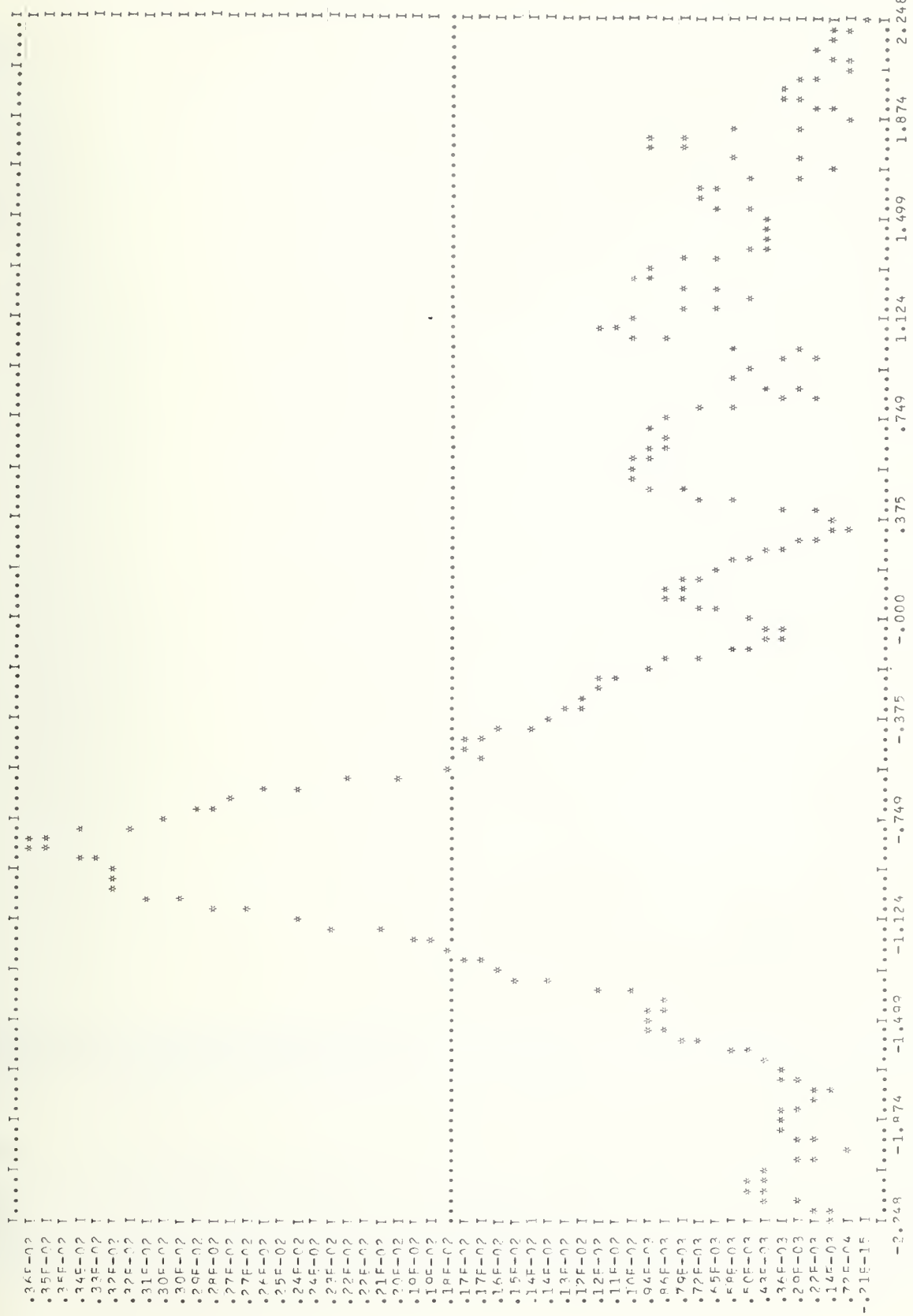
MAGNITUDE (X0-CUT)

.14155E-03 X0= -2.24544  
 .16747E-03 X0= -2.27772  
 .22520E-03 X0= -2.27481  
 .31027E-03 X0= -2.18500  
 .32647E-03 X0= -2.16517  
 .46115E-03 X0= -2.14432  
 .49222E-03 X0= -2.12153  
 .49130E-03 X0= -2.10271  
 .46015E-03 X0= -2.08199  
 .40247E-03 X0= -2.06107  
 .31841E-03 X0= -2.04025  
 .21015E-03 X0= -2.01944  
 .93240E-04 X0= -1.99862  
 .88626E-04 X0= -1.87780  
 .19712E-03 X0= -1.95698  
 .20091E-03 X0= -1.93415  
 .35223E-03 X0= -1.91534  
 .37921E-03 X0= -1.89262  
 .37517E-03 X0= -1.87370  
 .34506E-03 X0= -1.85288  
 .20841E-03 X0= -1.83207  
 .38073E-03 X0= -1.81125  
 .18253E-03 X0= -1.79043  
 .15087E-03 X0= -1.76961  
 .20202E-03 X0= -1.74870  
 .27937E-03 X0= -1.72797  
 .34784E-03 X0= -1.70715  
 .39061E-03 X0= -1.68633  
 .41388E-03 X0= -1.66551  
 .44464E-03 X0= -1.64469  
 .50480E-03 X0= -1.62388  
 .59305E-03 X0= -1.60306  
 .60029E-03 X0= -1.58224  
 .78346E-03 X0= -1.56142  
 .86946E-03 X0= -1.54060  
 .93479E-03 X0= -1.51978  
 .96644E-03 X0= -1.49896  
 .95106E-03 X0= -1.47814  
 .89856E-03 X0= -1.45732  
 .85484E-03 X0= -1.43651  
 .88473E-03 X0= -1.41569  
 .10086E-02 X0= -1.39487  
 .11308E-02 X0= -1.37405  
 .13521E-02 X0= -1.35323  
 .14872E-02 X0= -1.33241  
 .15848E-02 X0= -1.31150  
 .16587E-02 X0= -1.29077  
 .17187E-02 X0= -1.26995  
 .17721E-02 X0= -1.24914  
 .18243E-02 X0= -1.22832

.19868E-02	Y0=	-1.20750
.19770E-02	Y0=	-1.18668
.21070E-02	Y0=	-1.16494
.22722E-02	Y0=	-1.14504
.24501E-02	Y0=	-1.12422
.26465E-02	Y0=	-1.10340
.28230E-02	Y0=	-1.08258
.29638E-02	Y0=	-1.06177
.31162E-02	Y0=	-1.04095
.32076E-02	Y0=	-1.02013
.32489E-02	Y0=	-.09921
.32470E-02	Y0=	-.07840
.32201E-02	Y0=	-.05757
.32290E-02	Y0=	-.03675
.32076E-02	Y0=	-.01602
.34022E-02	Y0=	-.09821
.35176E-02	Y0=	-.07439
.35056E-02	Y0=	-.05358
.34022E-02	Y0=	-.03276
.35266E-02	Y0=	-.01194
.33876E-02	Y0=	-.79112
.31072E-02	Y0=	-.77030
.30109E-02	Y0=	-.74948
.28852E-02	Y0=	-.72866
.28094E-02	Y0=	-.70784
.27686E-02	Y0=	-.68702
.27208E-02	Y0=	-.66621
.26227E-02	Y0=	-.64539
.24547E-02	Y0=	-.62457
.22208E-02	Y0=	-.60375
.19890E-02	Y0=	-.58293
.17050E-02	Y0=	-.56211
.14607E-02	Y0=	-.54129
.12721E-02	Y0=	-.52047
.10968E-02	Y0=	-.49965
.17132E-02	Y0=	-.47884
.15035E-02	Y0=	-.45802
.14934E-02	Y0=	-.43720
.15408E-02	Y0=	-.41638
.14566E-02	Y0=	-.39556
.13701E-02	Y0=	-.37474
.12004E-02	Y0=	-.35392
.12422E-02	Y0=	-.33310
.12161E-02	Y0=	-.31228
.11987E-02	Y0=	-.29146
.11865E-02	Y0=	-.27065
.11672E-02	Y0=	-.24983
.11271E-02	Y0=	-.22901
.10602E-02	Y0=	-.20819
.96612E-03	Y0=	-.18737
.85204E-03	Y0=	-.16655
.72061E-03	Y0=	-.14573
.61007E-03	Y0=	-.12491
.50068E-03	Y0=	-.10409
.40727E-03	Y0=	-.08328
.34665E-03	Y0=	-.06246
.31842E-03	Y0=	-.04164
.40400E-03	Y0=	-.02082
.51567E-03	Y0=	0.00000
.63468E-03	Y0=	.02092
.73642E-03	Y0=	.04164
.80650E-03	Y0=	.06246
.84031E-03	Y0=	.08328
.84100E-03	Y0=	.10409

.91411E-03 Y0= .12401  
 .77352E-03 X0= .14572  
 .71898E-03 X0= .16655  
 .65537E-03 Y0= .19737  
 .60276E-03 Y0= .20819  
 .50545E-03 X0= .22001  
 .42852E-03 Y0= .24082  
 .34301E-03 X0= .27065  
 .26875E-03 X0= .29144  
 .20133E-03 Y0= .31228  
 .13575E-03 X0= .33310  
 .77303E-04 Y0= .35302  
 .11085E-03 Y0= .37474  
 .23311E-03 X0= .39554  
 .38543E-03 X0= .41639  
 .56644E-03 X0= .43720  
 .69844E-03 X0= .45802  
 .82202E-03 X0= .47884  
 .98557E-03 X0= .49965  
 .10115E-03 X0= .52047  
 .10176E-03 X0= .54129  
 .99809E-03 X0= .56211  
 .96318E-03 X0= .58293  
 .92222E-03 X0= .60375  
 .88415E-03 X0= .62457  
 .84974E-03 X0= .64539  
 .81872E-03 X0= .66621  
 .78784 X0= .68702  
 .74084 X0= .70784  
 .72866 X0= .72866  
 .74048 X0= .74048  
 .77030 X0= .77030  
 .79112 X0= .79112  
 .81194 X0= .81194  
 .83276 X0= .83276  
 .85358 X0= .85358  
 .87439 X0= .87439  
 .89521 X0= .89521  
 .91603 X0= .91603  
 .93685 X0= .93685  
 .95767 X0= .95767  
 .97849 X0= .97849  
 .99931 X0= .99931  
 1.62013 X0= 1.62013  
 1.04005 X0= 1.04005  
 1.06177 X0= 1.06177  
 1.68258 X0= 1.68258  
 1.10340 X0= 1.10340  
 1.12422 X0= 1.12422  
 1.14504 X0= 1.14504  
 1.16586 X0= 1.16586  
 1.18668 X0= 1.18668  
 1.20750 X0= 1.20750  
 1.22832 X0= 1.22832  
 1.24914 X0= 1.24914  
 1.26996 X0= 1.26996  
 1.29077 X0= 1.29077  
 1.31159 X0= 1.31159  
 1.33241 X0= 1.33241  
 1.35323 X0= 1.35323  
 1.37405 X0= 1.37405  
 1.39487 X0= 1.39487  
 1.41569 X0= 1.41569  
 1.43651 X0= 1.43651

•423ACE-03	X0=	1.45732
•40PA0E-03	Y0=	1.47814
•4378CE-03	X0=	1.40886
•5204JE-03	X0=	1.51978
•61782E-03	Y0=	1.54060
•691A7E-03	Y0=	1.56142
•72186E-03	X0=	1.58224
•70278E-03	Y0=	1.60306
•43324E-03	Y0=	1.62388
•50845E-03	X0=	1.64469
•32308E-03	Y0=	1.66551
•11828E-03	X0=	1.68633
•26354E-03	X0=	1.70715
•54366E-03	Y0=	1.72797
•77412E-03	X0=	1.74879
•90578E-03	Y0=	1.76961
•90435E-03	Y0=	1.79043
•77630E-03	X0=	1.81125
•55646E-03	X0=	1.83207
•20846E-03	Y0=	1.85289
•70658E-04	Y0=	1.87370
•13775E-03	X0=	1.89452
•25092E-03	X0=	1.91534
•31250E-03	Y0=	1.93616
•34474E-03	Y0=	1.95698
•35737E-03	Y0=	1.97780
•34073E-03	X0=	1.99862
•28240E-03	Y0=	2.01944
•18280E-03	Y0=	2.04025
•58766E-04	X0=	2.06107
•64270E-04	Y0=	2.08189
•15841E-03	Y0=	2.10271
•20388E-03	X0=	2.12353
•20388E-03	Y0=	2.14435
•16285E-03	X0=	2.16517
•11415E-03	X0=	2.18599
•10140E-03	Y0=	2.20681
•12208E-03	X0=	2.22762
•0.	Y0=	2.24844



SIMULATION TEST  
 MAGNITUDE VS XD

Simulation Test	Magnitude
-2.248	1.874
-1.409	-1.124
-0.749	-0.000
-0.375	0.375
0.749	1.124
1.499	1.499
1.874	1.874
2.248	2.248

MAGNITUDE (Y0-CUT)  
 • Q1515E-04 Y0= -2.24844  
 • I3E415E-04 Y0= -2.22762  
 • 7214E-04 Y0= -2.20681  
 • 14610E-03 Y0= -2.10600

• 19775E-03	Y0=	-2.16517
• 10005E-03	Y0=	-2.14435
• 17364E-03	Y0=	-2.12343
• 11064E-03	Y0=	-2.10271
• 54103E-04	Y0=	-2.08189
• 13241E-03	Y0=	-2.06107
• 72250E-04	Y0=	-2.04025
• 11742E-03	Y0=	-2.01944
• 12571E-03	Y0=	-1.99862
• 11514E-03	Y0=	-1.97780
• 14282E-03	Y0=	-1.95698
• 24046E-04	Y0=	-1.93616
• 14272E-03	Y0=	-1.91534
• 82974E-04	Y0=	-1.89452
• 41153E-04	Y0=	-1.87370
• 11264E-03	Y0=	-1.85288
• 15560E-03	Y0=	-1.83207
• 29604E-03	Y0=	-1.81125
• 11484E-03	Y0=	-1.79042
• 26070E-03	Y0=	-1.76961
• 91052E-04	Y0=	-1.74879
• 69728E-04	Y0=	-1.72797
• 45509E-04	Y0=	-1.70715
• 87755E-04	Y0=	-1.68633
• 94722E-04	Y0=	-1.66551
• 12606E-03	Y0=	-1.64469
• 79772E-04	Y0=	-1.62388
• 12705E-03	Y0=	-1.60306
• 48334E-04	Y0=	-1.58224
• 72100E-04	Y0=	-1.56142
• 11248E-03	Y0=	-1.54060
• 67604E-04	Y0=	-1.51978
• 14579E-03	Y0=	-1.49896
• 23262E-03	Y0=	-1.47814
• 21462E-03	Y0=	-1.45732
• 18744E-03	Y0=	-1.43651
• 21894E-03	Y0=	-1.41569
• 16032E-03	Y0=	-1.39487
• 47072E-04	Y0=	-1.37405
• 15591E-03	Y0=	-1.35323
• 14477E-03	Y0=	-1.33241
• 26647E-03	Y0=	-1.31159
• 21994E-03	Y0=	-1.29077
• 15667E-03	Y0=	-1.26995
• 19243E-03	Y0=	-1.24914
• 95212E-04	Y0=	-1.22832
• 21367E-03	Y0=	-1.20750
• 46676E-04	Y0=	-1.18668
• 24595E-03	Y0=	-1.16586
• 17762E-03	Y0=	-1.14504
• 25200E-03	Y0=	-1.12422
• 15308E-03	Y0=	-1.10340
• 17255E-03	Y0=	-1.08258
• 12665E-03	Y0=	-1.06177
• 19069E-04	Y0=	-1.04095
• 15722E-03	Y0=	-1.02013
• 26454E-04	Y0=	-1.00031
• 22615E-03	Y0=	-0.97949
• 70149E-04	Y0=	-0.95867
• 28435E-03	Y0=	-0.93785
• 16635E-03	Y0=	-0.91703
• 16002E-03	Y0=	-0.89621
• 86043E-04	Y0=	-0.87539
• 17070E-03	Y0=	-0.85458

• 113105-03	Y0=	- .83276
• 107775-03	Y0=	- .81164
• 107005-03	Y0=	- .79112
• 106445-03	Y0=	- .77030
• 144025-03	Y0=	- .76048
• 175105-03	Y0=	- .72866
• 116855-03	Y0=	- .70784
• 111105-03	Y0=	- .68702
• 153005-03	Y0=	- .66621
• 108175-03	Y0=	- .64539
• 361225-03	Y0=	- .62457
• 353745-03	Y0=	- .60375
• 205285-03	Y0=	- .58293
• 442025-03	Y0=	- .56211
• 652065-04	Y0=	- .54129
• 367425-03	Y0=	- .52047
• 290475-03	Y0=	- .49965
• 104175-02	Y0=	- .47884
• 281205-03	Y0=	- .45802
• 744225-03	Y0=	- .43720
• 224845-02	Y0=	- .41638
• 112365-02	Y0=	- .39556
• 010555-03	Y0=	- .37474
• 286355-02	Y0=	- .35392
• 937855-03	Y0=	- .33310
• 443545-03	Y0=	- .31228
• 326035-03	Y0=	- .29146
• 245605-02	Y0=	- .27065
• 125815-02	Y0=	- .24983
• 898555-03	Y0=	- .22901
• 324025-02	Y0=	- .20819
• 770155-03	Y0=	- .18737
• 108105-02	Y0=	- .16655
• 557325-02	Y0=	- .14573
• 289595-02	Y0=	- .12491
• 225085-02	Y0=	- .10409
• 372705-02	Y0=	- .08328
• 655835-02	Y0=	- .06246
• 143885-02	Y0=	- .04164
• 130845-02	Y0=	- .02082
• 632485-02	Y0=	0.00000
• 221355-02	Y0=	• 02082
• 173515-02	Y0=	• 04164
• 170775-02	Y0=	• 06246
• 450155-02	Y0=	• 08328
• 191355-02	Y0=	• 10409
• 228265-02	Y0=	• 12491
• 742765-02	Y0=	• 14573
• 133015-02	Y0=	• 16655
• 326145-03	Y0=	• 18737
• 440705-03	Y0=	• 20819
• 225845-02	Y0=	• 22901
• 144215-02	Y0=	• 24983
• 105005-02	Y0=	• 27065
• 253095-02	Y0=	• 29146
• 358235-03	Y0=	• 31228
• 282015-03	Y0=	• 33310
• 350875-02	Y0=	• 35392
• 106665-02	Y0=	• 37474
• 522515-03	Y0=	• 39556
• 187465-02	Y0=	• 41638
• 727235-03	Y0=	• 43720
• 511755-03	Y0=	• 45802
• 216745-03	Y0=	• 47884

.70530E-03	Y0=	.49965
.78385E-03	Y0=	.52047
.80674E-03	Y0=	.54129
.12172E-03	Y0=	.56211
.15435E-03	Y0=	.58293
.27154E-03	Y0=	.60375
.20791E-03	Y0=	.62457
.26802E-03	Y0=	.64539
.18503E-03	Y0=	.66621
.23327E-02	Y0=	.68702
.12029E-03	Y0=	.70784
.88585E-04	Y0=	.72866
.12603E-03	Y0=	.74948
.12448E-03	Y0=	.77030
.16843E-03	Y0=	.79112
.52775E-04	Y0=	.81194
.19052E-03	Y0=	.83276
.18441E-03	Y0=	.85358
.15220E-02	Y0=	.87439
.18204E-03	Y0=	.89521
.13367E-03	Y0=	.91603
.15884E-03	Y0=	.93685
.13297E-03	Y0=	.95767
.22004E-03	Y0=	.97849
.10274E-02	Y0=	.99931
.15350E-03	Y0=	1.02013
.72577E-04	Y0=	1.04095
.17810E-03	Y0=	1.06177
.34843E-04	Y0=	1.08258
.19741E-02	Y0=	1.10340
.88809E-04	Y0=	1.12422
.26152E-03	Y0=	1.14504
.63958E-04	Y0=	1.16586
.14820E-03	Y0=	1.18668
.20403E-03	Y0=	1.20750
.14024E-03	Y0=	1.22832
.16013E-03	Y0=	1.24914
.51570E-04	Y0=	1.26995
.23120E-03	Y0=	1.29077
.21104E-04	Y0=	1.31159
.14050E-03	Y0=	1.33241
.19574E-03	Y0=	1.35323
.15348E-03	Y0=	1.37405
.19072E-03	Y0=	1.39487
.79567E-04	Y0=	1.41569
.22757E-03	Y0=	1.43651
.20623E-04	Y0=	1.45732
.20554E-03	Y0=	1.47814
.68274E-04	Y0=	1.49896
.12655E-03	Y0=	1.51978
.28172E-04	Y0=	1.54060
.11454E-03	Y0=	1.56142
.56253E-04	Y0=	1.58224
.94041E-04	Y0=	1.60306
.40007E-04	Y0=	1.62388
.12470E-03	Y0=	1.64469
.10218E-03	Y0=	1.66551
.36297E-04	Y0=	1.68633
.13303E-03	Y0=	1.70715
.91843E-04	Y0=	1.72797
.15072E-03	Y0=	1.74879
.92501E-04	Y0=	1.76961
.16364E-03	Y0=	1.79043
.15924E-03	Y0=	1.81125



.79224E-04	Y0=	1.83207
.15774E-03	Y0=	1.85228
.82779E-04	Y0=	1.87370
.73437E-04	Y0=	1.89452
.11684E-03	Y0=	1.91534
.66389E-04	Y0=	1.93616
.13131E-03	Y0=	1.95698
.61587E-04	Y0=	1.97780
.15033E-03	Y0=	1.99862
.14451E-03	Y0=	2.01944
.87715E-04	Y0=	2.04026
.16546E-03	Y0=	2.06107
.44551E-04	Y0=	2.08189
.15307E-03	Y0=	2.10271
.88286E-04	Y0=	2.12352
.12686E-03	Y0=	2.14434
.17851E-03	Y0=	2.16517
.64572E-04	Y0=	2.18600
.81835E-04	Y0=	2.20681
.12964E-03	Y0=	2.22762
	Y0=	2.24844

0.



U.S. DEPT. OF COMM. <b>BIBLIOGRAPHIC DATA SHEET</b> <i>(See instructions)</i>	<b>1. PUBLICATION OR REPORT NO.</b> NBSIR 80-1630	<b>2. Performing Organ. Report No.</b>	<b>3. Publication Date</b> June 1981
<b>4. TITLE AND SUBTITLE</b> DETERMINATION OF MUTUAL COUPLING BETWEEN CO-SITED MICROWAVE ANTENNAS AND CALCULATION OF NEAR-ZONE ELECTRIC FIELD			
<b>5. AUTHOR(S)</b> C. F. Stubenrauch and A. D. Yaghjian			
<b>6. PERFORMING ORGANIZATION</b> <i>(If joint or other than NBS, see instructions)</i>  NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		<b>7. Contract/Grant No.</b> DAEA-76-F-D760	<b>8. Type of Report &amp; Period Covered</b> NBSIR July 1976 - June 1978
<b>9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS</b> <i>(Street, City, State, ZIP)</i> US Army Communications Electronics Engineering Installation Agency Fort Huachuca, Arizona 85613			
<b>10. SUPPLEMENTARY NOTES</b>  <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
<b>11. ABSTRACT</b> <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i>  The theory and computer programs which allow the efficient computation of coupling between co-sited antennas given their far-field patterns are developed. Coupling between two paraboloidal reflector antennas is computed using both measured far-field patterns and far-field patterns which were obtained from a physical optics (PO) model. These computed results are then compared to the coupling measured directly on an outdoor antenna range. Far fields calculated using the PO model are compared to those obtained from transformed near-field measurements for several reflector antennas. Theory and algorithms are also developed for calculating near-field patterns from far fields obtained from the PO model. Documentation of the near-field and coupling computer programs is presented in the appendices. Conclusions and recommendations for future work are included.			
<b>12. KEY WORDS</b> <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> Co-sited antennas; coupling; far fields; near fields; physical optics; plane-wave spectrum; reflector antennas.			
<b>13. AVAILABILITY</b>  <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.  <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		<b>14. NO. OF PRINTED PAGES</b>  176	<b>15. Price</b>  \$12.00









