NBSIR 80-1630

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

NBSIR 80-1630

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

LIST OF FIGURES

Page

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.[1] 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 Γ_0' and Γ_L' respectively.

The quantity b_0'/a_0, which we shall call the <u>coupling quotient</u>, 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, $|b_0'/a_0|^2$ <u>equals the amount of power coupled to the receiving antenna per unit power incident to the transmitting antenna</u>. (This power ratio expressed in decibels is commonly referred to as the <u>insertion loss ratio.</u>) 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]:

$$\frac{b_0'}{a_0} = \frac{1}{1-\Gamma_L'\Gamma_0'} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \underline{s}_{02}' (\underline{K}) \cdot \underline{s}_{10} (\underline{K}) \, e^{i\gamma d} \, d\underline{K}, \tag{1}$$

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

[1] 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.

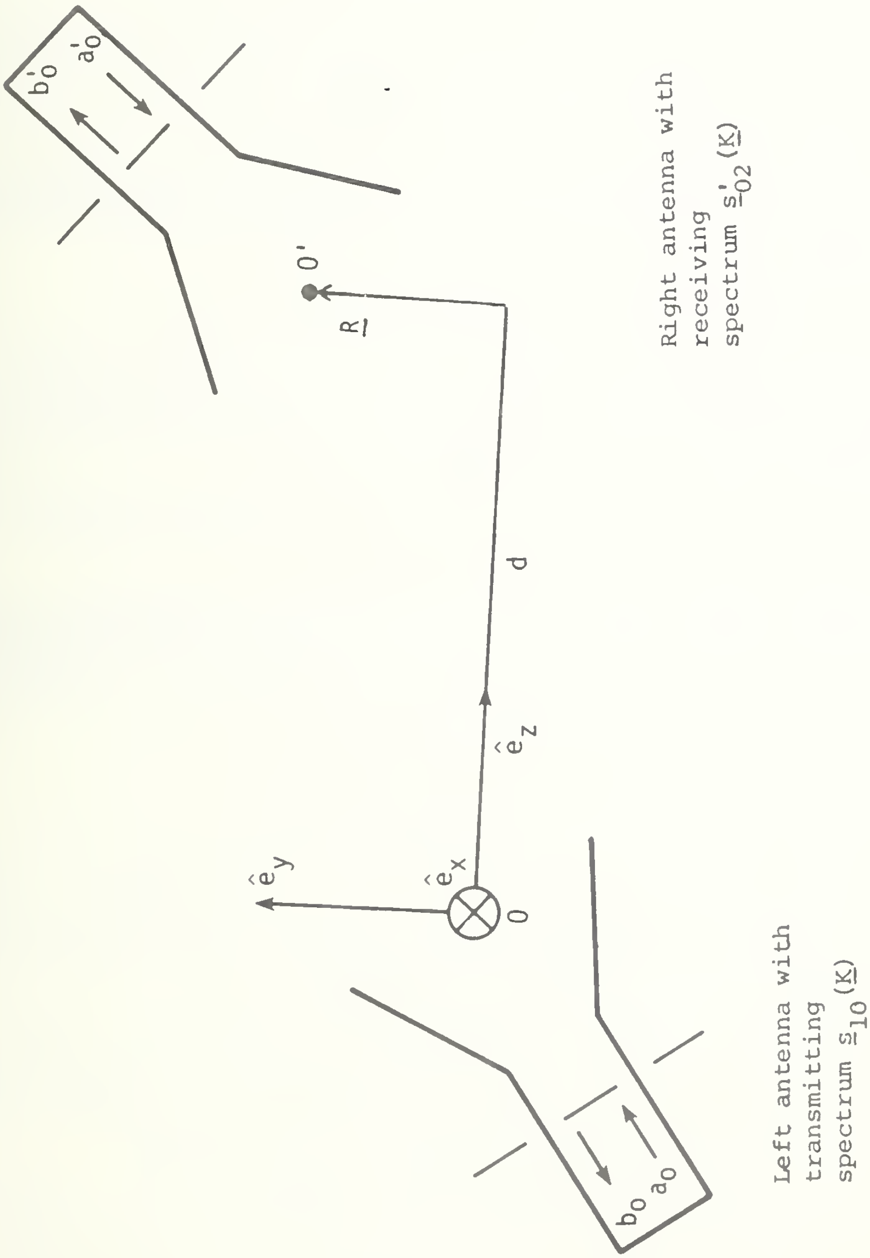


Figure 1. Coupling Schematic for two antennas (0 and 0' will be chosen at roughly the center of the radiating part of their respective antenna).

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 λ 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],

$$\eta_0' \, \underline{s}_{02}' \, (\underline{K}) = \frac{\gamma}{kZ_0} \, \underline{s}_{20}' \, (-\underline{K}). \qquad (2)$$

All quantities in eq (2) have been defined in the previous section except the impedance of free space Z_0 and η_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,

$$\frac{b_0'}{a_0} = \frac{(1-\Gamma_L'\Gamma_0')^{-1}}{kZ_0\eta_0'} \iint_{K<k} \gamma \underline{s}_{20}'(-\underline{K}) \cdot \underline{s}_{10}(\underline{K}) e^{i\gamma d} \, d\underline{K}. \qquad (3)$$

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 $\underline{E}(\underline{r})_{r\to\infty}$ 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.,

$$\underline{f}(\underline{r}) \equiv \frac{re^{-ikr}}{a_o} \underline{E}(\underline{r})_{r\to\infty} , \qquad (4)$$

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

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

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 $\underline{f}(\underline{k})$ 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:

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

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

$$\underline{f}'(\underline{r}) \equiv \frac{re^{-ikr}}{a'_o} \underline{E}'(\underline{r})_{r\to\infty} . \qquad (7)$$

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:

$$\frac{b'_o}{a_o} = -C' \iint_{K<k} \frac{\underline{f}'(-\underline{k}) \cdot \underline{f}(\underline{k})}{\gamma} e^{i\gamma d} \, d\underline{K} . \qquad (8)$$

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

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).

$$\frac{b_0}{a_0'} = -C \iint_{K'<k} \frac{\underline{f}(-\underline{k}') \cdot \underline{f}'(\underline{k}')}{\gamma'} e^{i\gamma' d} \, d\underline{K}' \,, \tag{9}$$

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

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

Γ_0 and Γ_L are now the reflection coefficients to the antenna on the left and its passive termination, respectively. And η_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.,

$$\frac{b_0}{a_0'} = -C \iint_{K<k} \frac{\underline{f}(\underline{k}) \cdot \underline{f}'(-\underline{k})}{\gamma} e^{i\gamma d} \, d\underline{K} \,. \tag{11}$$

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.

$$C' \frac{b_0}{a_0'} = C \frac{b_0'}{a_0} \quad . \tag{12}$$

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.

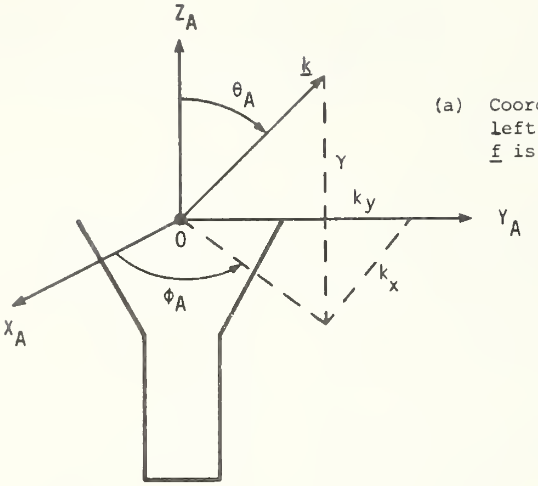
Unfortunately, a major problem, ignored so far, is the fact that the far-field pattern of an antenna is given with respect to a Cartesian coordinate system which is fixed in the antenna and which is not, in general, aligned with the Cartesian system shown in figure 1 to which the far-field patterns $\underline{f}(\underline{k})$ 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.

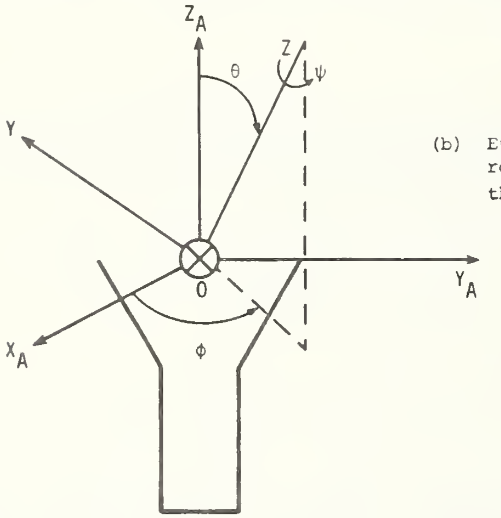
1.2.1. Rotational Transformations from (k_x, k_y) to the Far-Field Direction in the Fixed Coordinate System of Each Antenna

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 ϕ_A and θ_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 ϕ_A and θ_A.

Let (ϕ, θ, ψ) 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. (ϕ and θ 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].



(a) Coordinate system fixed to the left antenna in which the far field $\underline{f}$ is known as a function of ϕ_A, θ_A.



(b) Eulerian angles (ϕ, θ, ψ) needed to rotate the fixed axes x_A, y_A, z_A to the coupling axes x,y,z of Fig. 1.

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 γ is determined from k_x and k_y. However, we are given as known (measured or computed) $\overline{f}$ as a function of ϕ_A and θ_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 (ϕ_A,θ_A) under the given Eulerian angles (ϕ,θ,ψ) 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 ϕ_A and θ_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 ϕ_p and θ_p with respect to (x_p, y_p, z_p) axes fixed to the antenna (and centered at 0'), then (ϕ_p,θ_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 ϕ', θ', and ψ' 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 (ϕ_A, θ_A) and (ϕ_p, θ_p), are similar and can be written explicitly as:

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

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

The top signs in eqs (13) go with (ϕ_A,θ_A), the bottom with (ϕ_p,θ_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 γ (which equals $\sqrt{k^2-(k_x^2+k_y^2)}$). The computer program merely contains a subroutine which yields (ϕ_A,θ_A) and (ϕ_p,θ_p) from eqs (13) when given the Eulerian angles (ϕ,θ,ψ), (ϕ',θ',ψ'), and (k_x,k_y) as input.

With the transformations of eqs (13), eq (8) can now be expressed in terms of (θ_A,ϕ_A) and (ϕ_p,θ_p):

$$\frac{b_o'}{a_o} = -C' \iint_{K<k} \frac{\underline{f}'(\phi_p,\theta_p)\cdot\underline{f}(\phi_A,\theta_A)}{\gamma} e^{i\gamma d}\, d\underline{K}. \tag{14}$$

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 (ϕ_A,θ_A) and (ϕ_p,θ_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 (ϕ,θ,ψ) and (ϕ',θ',ψ').

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,

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

$$\underline{f}' = f_{x'}'\hat{e}_{x'} + f_{y'}'\hat{e}_{y'} + f_{z'}'\hat{e}_{z'}. \tag{15b}$$

Because by definition,

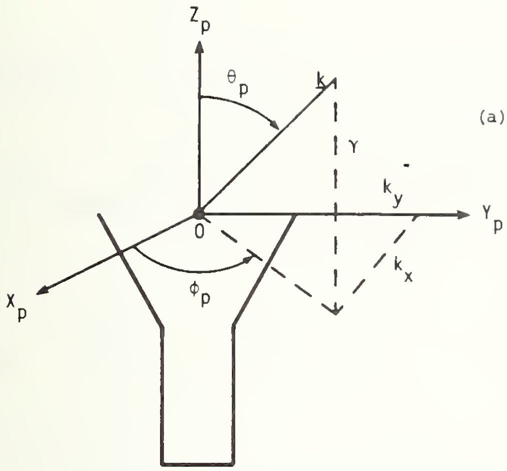
$$\hat{e}_{x'} = -\hat{e}_x,\ \hat{e}_{y'} = \hat{e}_y, \text{ and } \hat{e}_{z'} = -\hat{e}_z, \tag{16}$$

the dot product becomes

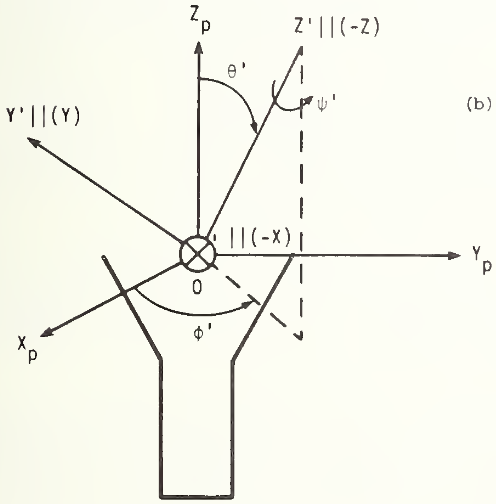
$$\underline{f}'\cdot\underline{f} = -f_{x'}'f_x + f_{y'}'f_y - f_{z'}'f_z\ . \tag{17}$$

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

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



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



(b) Eulerian angles (ϕ', θ', ψ') 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 (ϕ',θ',ψ') and $(f'_{xp},f'_{yp},f'_{zp})$ replacing (ϕ,θ,ψ) 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 $\underline{f}_A$, 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,

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

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 (ϕ_A,θ_A) in the matrix with (ϕ_p,θ_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

$$\frac{b_0'(\underline{R},d)}{a_0} = -C' \iint_{K<k} \frac{\underline{f}'(\phi_p,\theta_p)\cdot\underline{f}(\phi_A,\theta_A)}{\gamma} e^{i\gamma d} e^{i\underline{K}\cdot\underline{R}} d\underline{K} \ . \tag{20}$$

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 $R \to \infty$, 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

$$R_0 = Bd \ , \tag{21}$$

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

$$\frac{\Delta k_x}{k}, \frac{\Delta k_y}{k} = \frac{\lambda}{2Bd} \ , \tag{22}$$

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

$$\frac{b_0'(\underline{R},d)}{a_0} = -C'\Delta k_x \Delta k_y \sum_{m=-M}^{M} \sum_{\ell=-L}^{L} \frac{\underline{f}'(\phi_p^{\ell m},\theta_p^{\ell m})\cdot\underline{f}(\phi_A^{\ell m},\theta_A^{\ell m})}{\gamma_{\ell m}} e^{i\gamma_{\ell m} d} e^{i\underline{K}_{\ell m}\cdot\underline{R}} \ , \tag{23}$$

where

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

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

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 $\underline{R}$ 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 ($\underline{R}$,d) and orientation of the antennas. For d/λ of appreciable size, the number of points can become so large that the computer time required to evaluate eq (23) over a range of $\underline{R}$, even using the FFT, can become exorbitant. For example, if d = 10 meters and λ = 3cm, choosing a typical value of B = 2 yields $(2Bd/\lambda)^2$ = 1.8×10^6 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 $e^{i\gamma d}\, e^{i\underline{K}\cdot\underline{R}}$ in the plane-wave form $e^{i\underline{k}\cdot\underline{r}}$, where $\underline{r} = \underline{R}+d\hat{e}_z$. For r much larger than the dimension of either antenna, the function $e^{i\underline{k}\cdot\underline{r}}$ oscillates more rapidly than the oscillations of the far-field pattern dot product $\underline{f}'\cdot\underline{f}$, except when $\underline{k}$ is in the directions approximately parallel to $\underline{r}$. This means that the integration in eq (20) will essentially cancel to zero except for the contribution near $\underline{k}$ equal to $\underline{r}$, 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 $|\underline{R}|$ between 0 and R, only the part of the spectrum defined by

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

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 2λ, in which case D and/or D' is set equal to 2λ.[2] 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 2λ. Of course, nearly all microwave antennas have dimensions much greater than 2λ.

[2] Equation (25) assumes implicitly that the origins O and O' for the two antennas by which r is defined (r = OO') are chosen near the physical centers of their respective antennas.

For $R<<(D+D')$, 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

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

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

$$L,M \simeq \frac{2B(D+D')}{\lambda} \text{ .} \tag{27}$$

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 Δk_x, Δk_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).

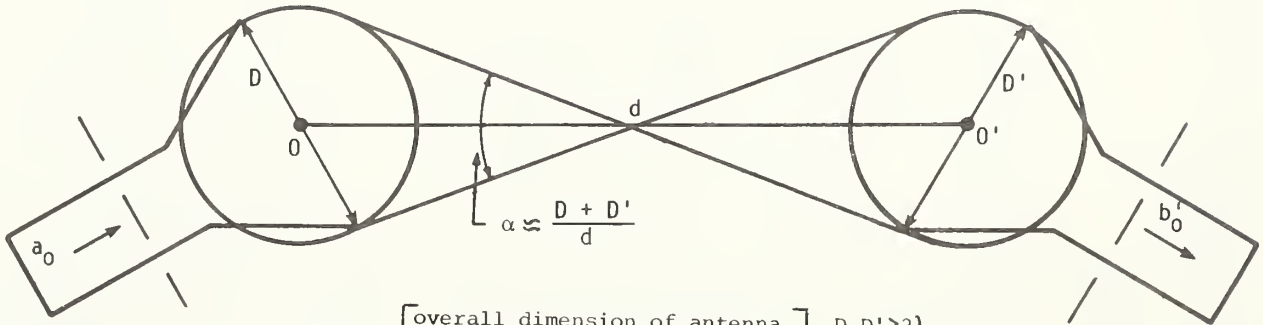
Physically, eq (26) has a very simple interpretation. Referring to figure 4, it says that to a good approximation, for ordinary antennas larger than a couple of wavelengths across, only that portion of the plane-wave spectrum within the sheaf of angles mutually subtended by the smallest spheres circumscribing 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.,

$$R<<(D + D'), \tag{28}$$

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 $b_0'(\underline{R},d)$ 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 artifically band-limits the coupling quotient to

$$R_0 = \text{larger of} \begin{Bmatrix} B(D + D') \\ \dfrac{B\lambda d}{(D + D')} \end{Bmatrix} \text{ .} \tag{29}$$

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



$$D, D' = \begin{bmatrix} \text{overall dimension of antenna} & D, D' \geq 2\lambda \\ 2\lambda & D, D' < 2\lambda \end{bmatrix}$$

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

$$\frac{\Delta k_x}{k}, \frac{\Delta k_y}{k} = \text{smaller of} \left\{ \begin{array}{c} \dfrac{\lambda}{2B(D + D')} \\ \dfrac{(D + D')}{2Bd} \end{array} \right\} , \qquad (30)$$

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

$$L,M \simeq \text{larger of} \left\{ \begin{array}{c} \dfrac{2B(D + D')^2}{\lambda d} \\ 2B \end{array} \right\} . \qquad (31)$$

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

$$\frac{b_0'(\underline{R},d)}{a_0} = -C' \, e^{-ik(a_1 x + b_1 y)} \, \frac{(a_1+a_2)(b_1+b_2)}{N_1 N_2} \sum_{j_1=1}^{N_1} \sum_{j_2=1}^{N_2} A[j_1,j_2] \, e^{2\pi i \left(\frac{(j_1-1)(m_1-1)}{N_1} + \frac{(j_2-1)(m_2-1)}{N_2} \right)} \qquad (32)$$

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,

$$-a_1 \le \frac{k_x}{k} \le a_2 \qquad (33a)$$

$$-b_1 \le \frac{k_y}{k} \le b_2 \quad . \qquad (33b)$$

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_xk_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:

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

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

where

$$m_1 = 1, 2, \ldots, N_1 \tag{35a}$$

$$m_2 = 1, 2, \ldots, N_2 \quad . \tag{35b}$$

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

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

where (ϕ_p,θ_p) and (ϕ_A,θ_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,

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

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

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 γ 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:

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

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.

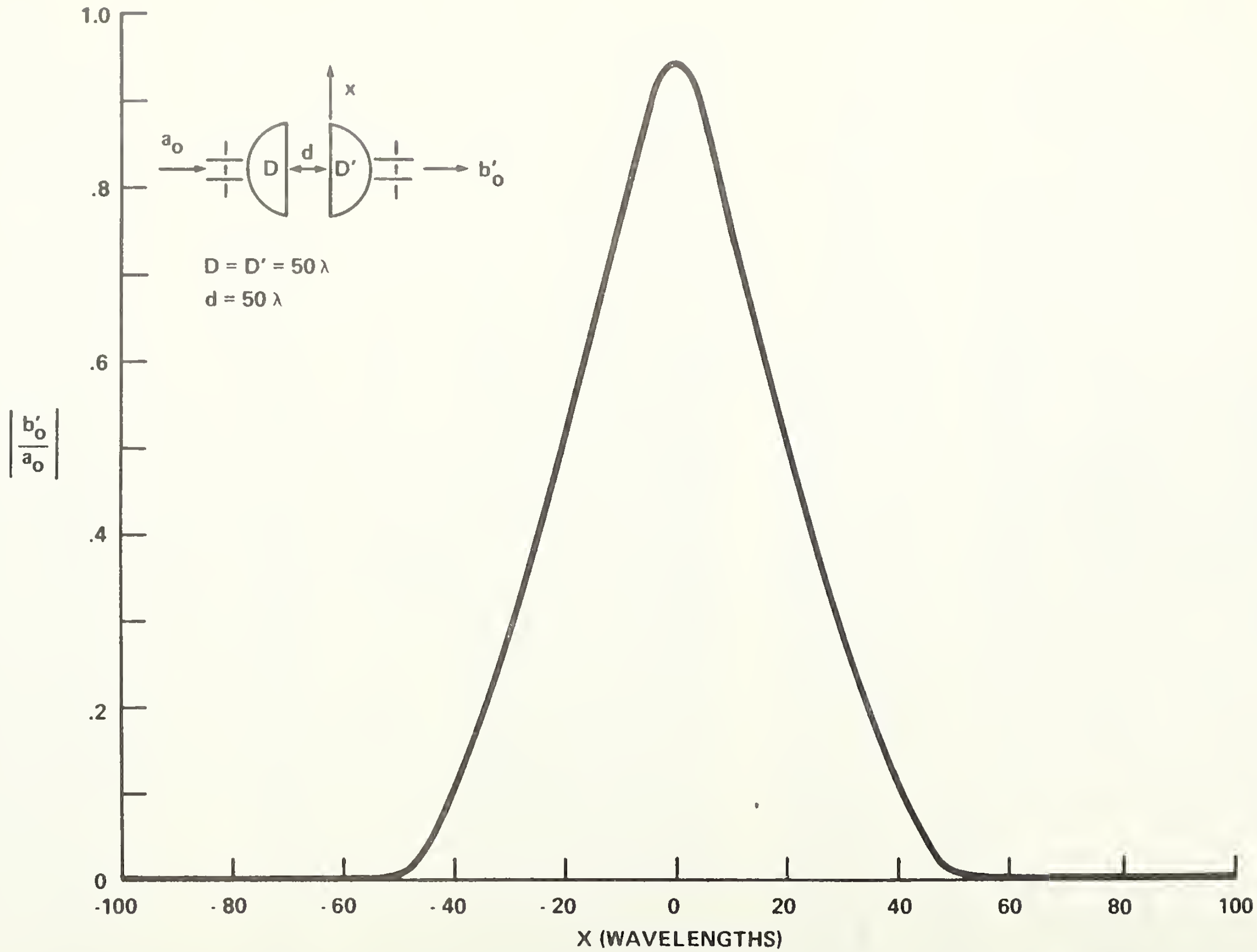


Figure 5. Hypothetical circular antennas directly facing each other in the near field.

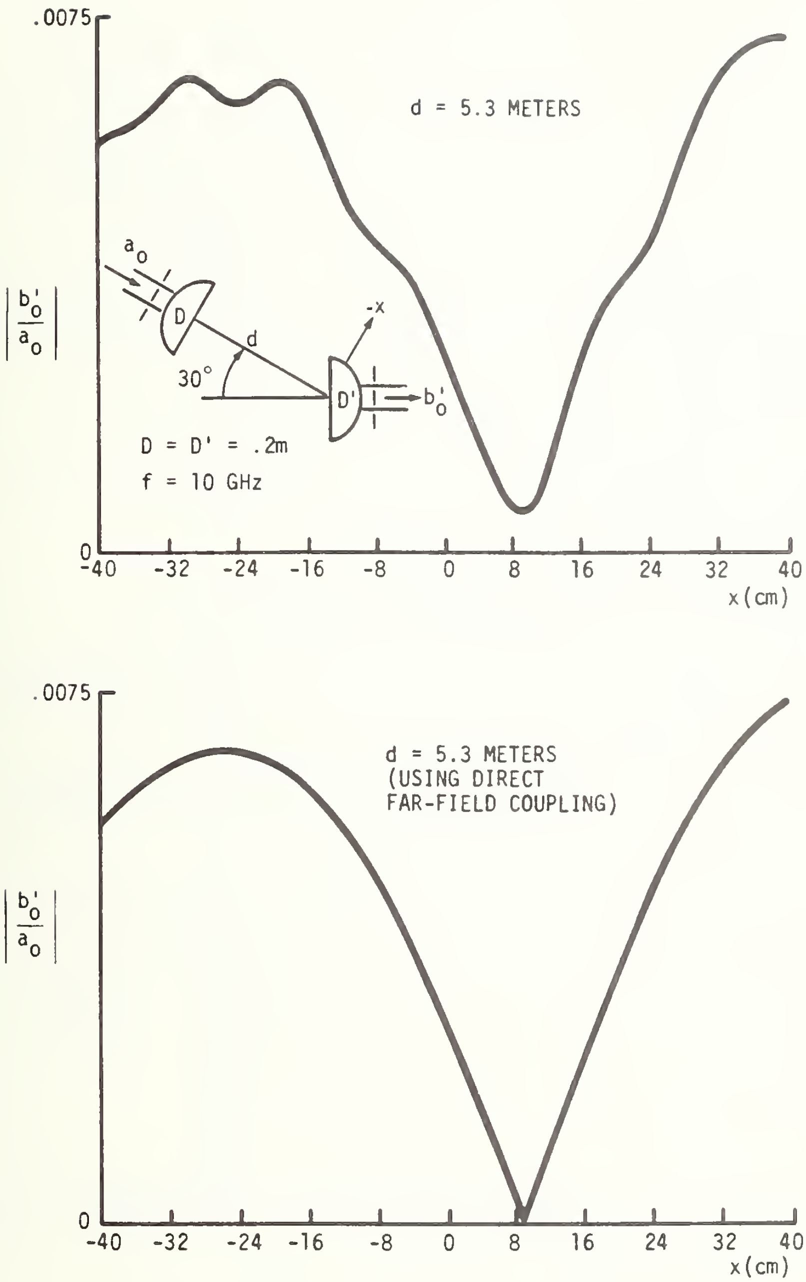


Figure 6. Coupling of circular antennas computed first using FFT integration, and then directly from far field along direction of separation.

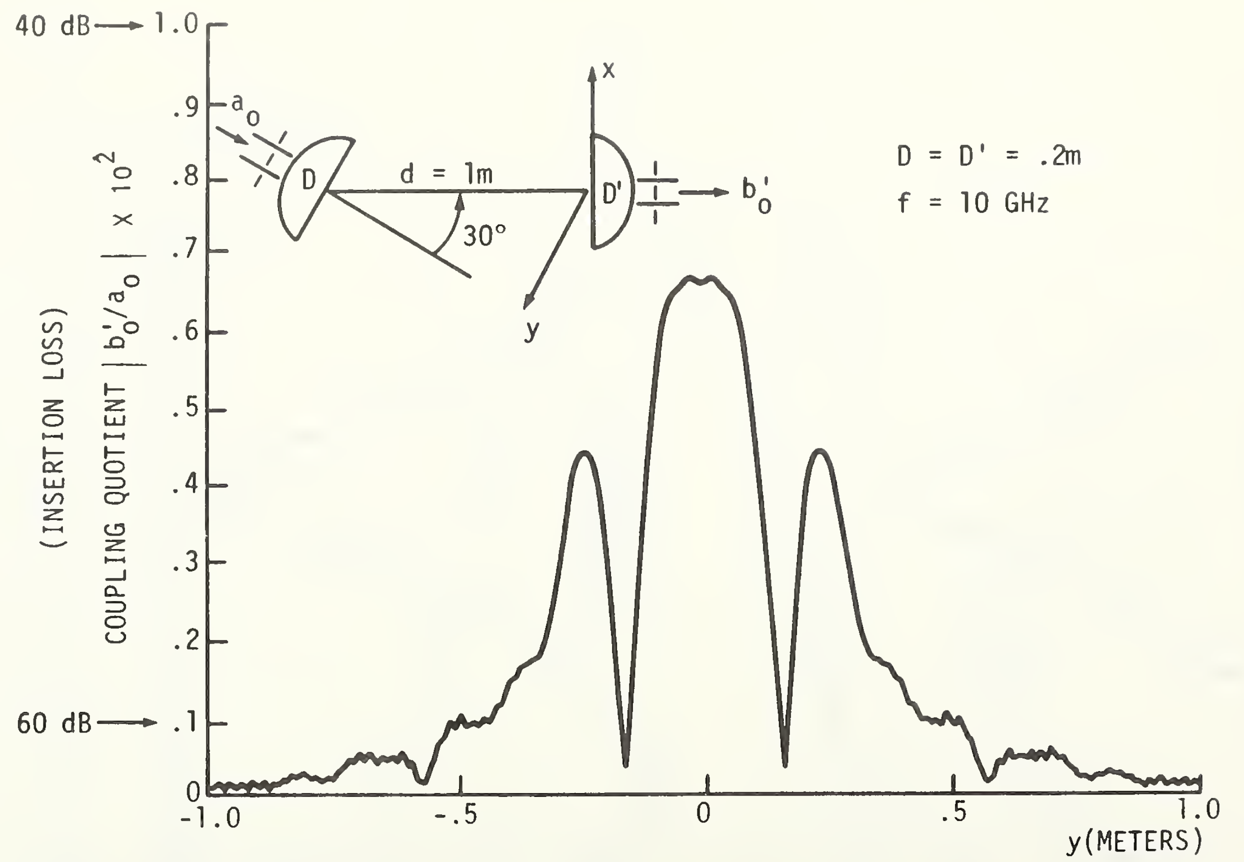


Figure 7. Typical coupling curve for antennas skewed in their near field.

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

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

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.

$k^2 = \omega^2 \mu\varepsilon$; 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

$$\underline{k}^{+} \bullet \underline{b} = 0; \qquad \underline{k}^{-} \bullet \underline{a} = 0 .$$

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

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

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

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

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

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

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

$$G(\underline{K}) = \frac{4\pi Y_o \gamma^2 |\underline{s}_{10}(\underline{K})|^2}{\eta_o (1-|\Gamma_o|^2)} , \qquad (43)$$

where, as in section 1, $Y_0 = 1/Z_0$ is the admittance of free space, η_0 is the characteristic admittance of the feed mode, and Γ_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

$$\underline{E}(\underline{r}) = \frac{a_o}{2\pi} \sqrt{\frac{\eta_o (1-|\Gamma_o|^2) \, G(0)}{4\pi Y_o k^2}} \int \hat{\underline{s}}_{10} (\underline{K}) \, e^{i|\gamma|z} \, e^{i\underline{K}\bullet\underline{R}} \, d\underline{K} , \qquad (44)$$

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

$$P_o = \frac{1}{2} \eta_o (|a_o|^2 - |b_o|^2) ,$$

but because $b_0 = \Gamma_0 a_0$

$$P_o = \frac{1}{2} \eta_o |a_o|^2 (1 - |\Gamma_o|^2) .$$

Substituting this into eq (44) gives

$$\underline{E}(\underline{r}) = \frac{1}{2\pi} \sqrt{\frac{P_o \, G(0)}{2\pi Y_o k^2}} \int \hat{\underline{s}}_{10} (\underline{K}) \, e^{i|\gamma|z} \, e^{i\underline{K}\bullet\underline{R}} \, d\underline{K} . \qquad (45)$$

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 σ, can be related to its physical area by the expression

$$\sigma = \eta A \ ,$$

where

η = 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.

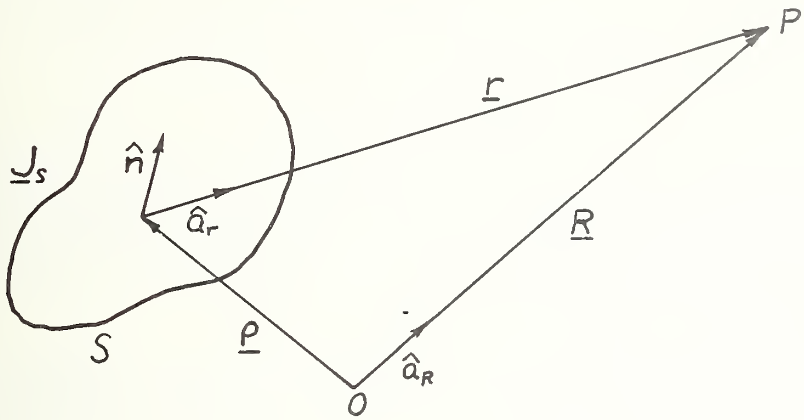


Figure 8. Geometry of vectors for surface integral.

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 $r >> \lambda$ and $|\rho|_{max} << R$ or r, we can write the electric field at P as

$$\underline{E}(\underline{R}) = \frac{i\omega\mu_o}{4\pi} \frac{e^{ikR}}{R} \int_S [\underline{J}_s - (\underline{J}_s \cdot \hat{a}_R)\, \hat{a}_R] e^{-ik\underline{\rho}\bullet\hat{a}_R} \, dS \; . \tag{46}$$

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.,

$$\underline{J}_s = 2[\hat{n} \times \underline{H}_{inc}] \; , \tag{47}$$

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 Ø) requires that the field be calculated only in the E- and H-planes, i.e., $\emptyset = \pi/2$ and 0, respectively. The field at any point (R,θ,Ø) is given by

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

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

$$\underline{E} = \frac{e^{ikR}}{R} \left[E_E(\theta) \cos\theta - E_H(\theta)\right] \cos\emptyset \sin\emptyset \hat{a}_x$$

$$+ \left[E_E(\theta) \cos\theta \sin^2\emptyset + E_H(\theta) \cos^2\emptyset\right] \hat{a}_y - E_E(\theta) \sin\theta \sin\emptyset \hat{a}_z \ . \tag{49}$$

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 θ' is the angle measured from the feed axis. For this case, the feed patterns in the E- and H-planes are given by

$$E_E = \cos^{n_E} \theta'$$

$$E_H = \cos^{n_H} \theta'.$$

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.

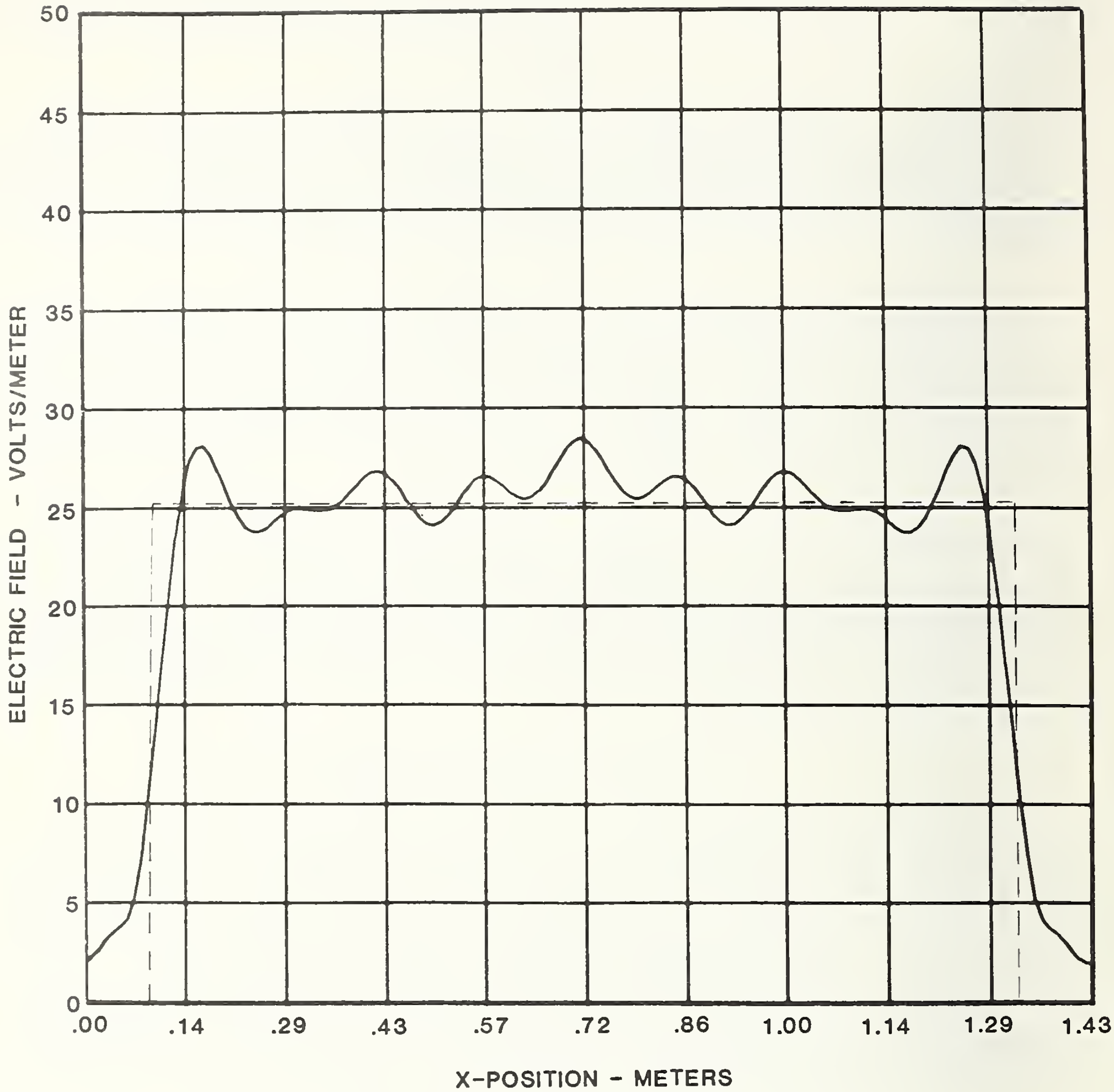


Figure 9a. Field strength in a uniformly illuminated aperture calculated using physical optics far fields. Dashed line indicates theoretical distribution.

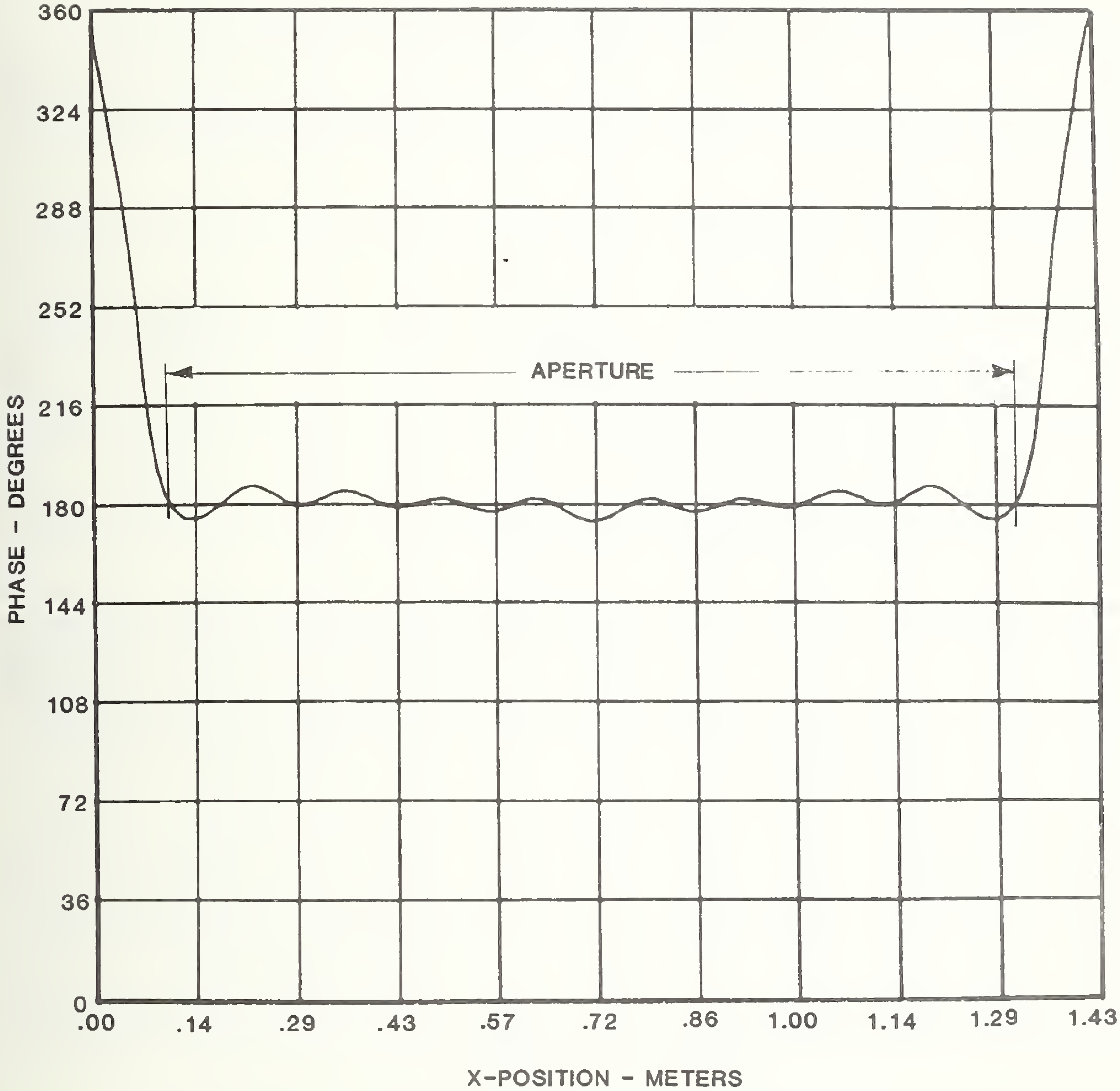


Figure 9b. Phase of field in a uniformly illuminated aperture calculated using physical optics for fields.

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(λ)	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

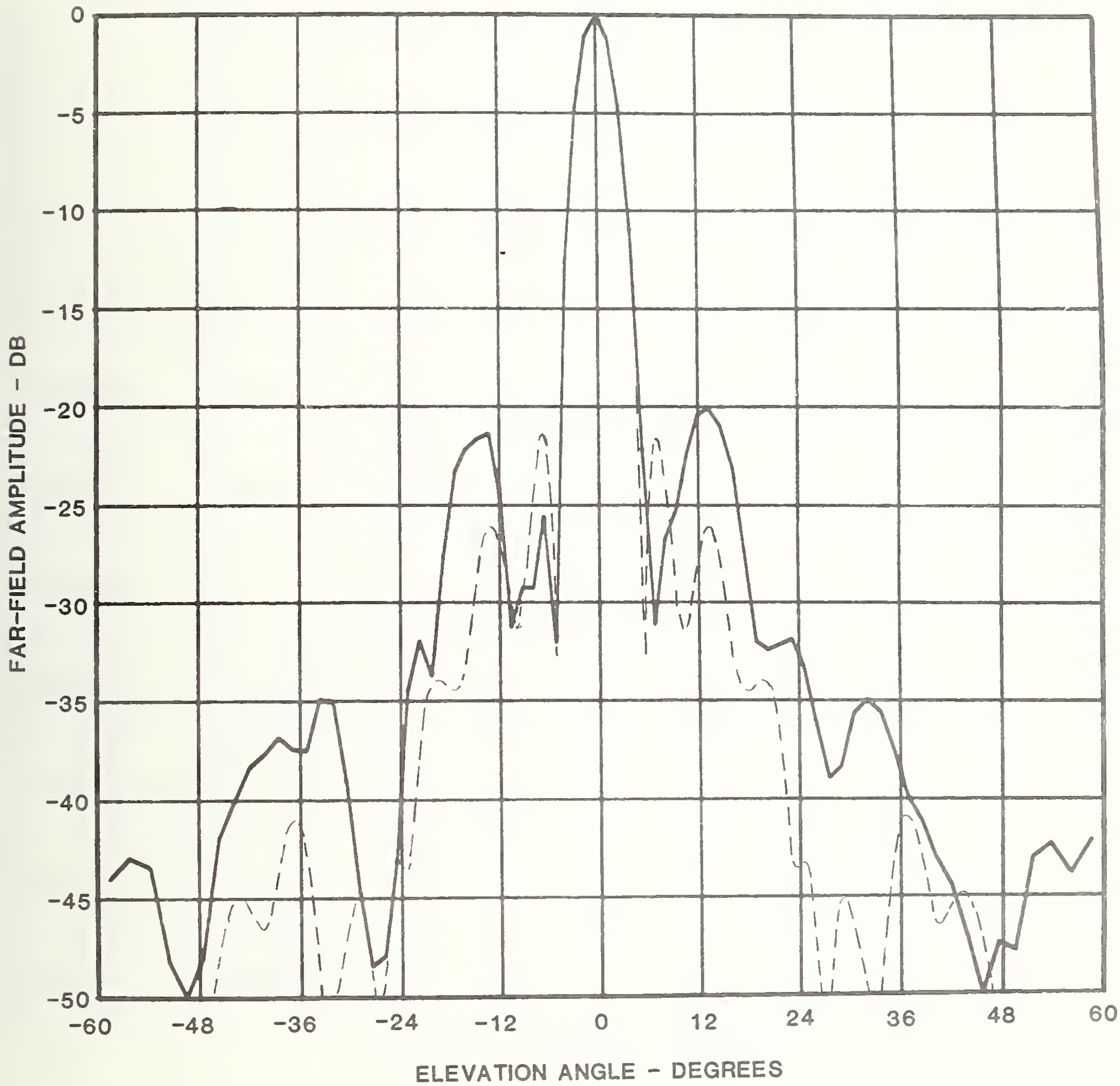


Figure 10a. Comparison of measured and calculated far-field patterns for antenna No. 1. E-plane cut, solid line - measured pattern, dashed line - physical optics.

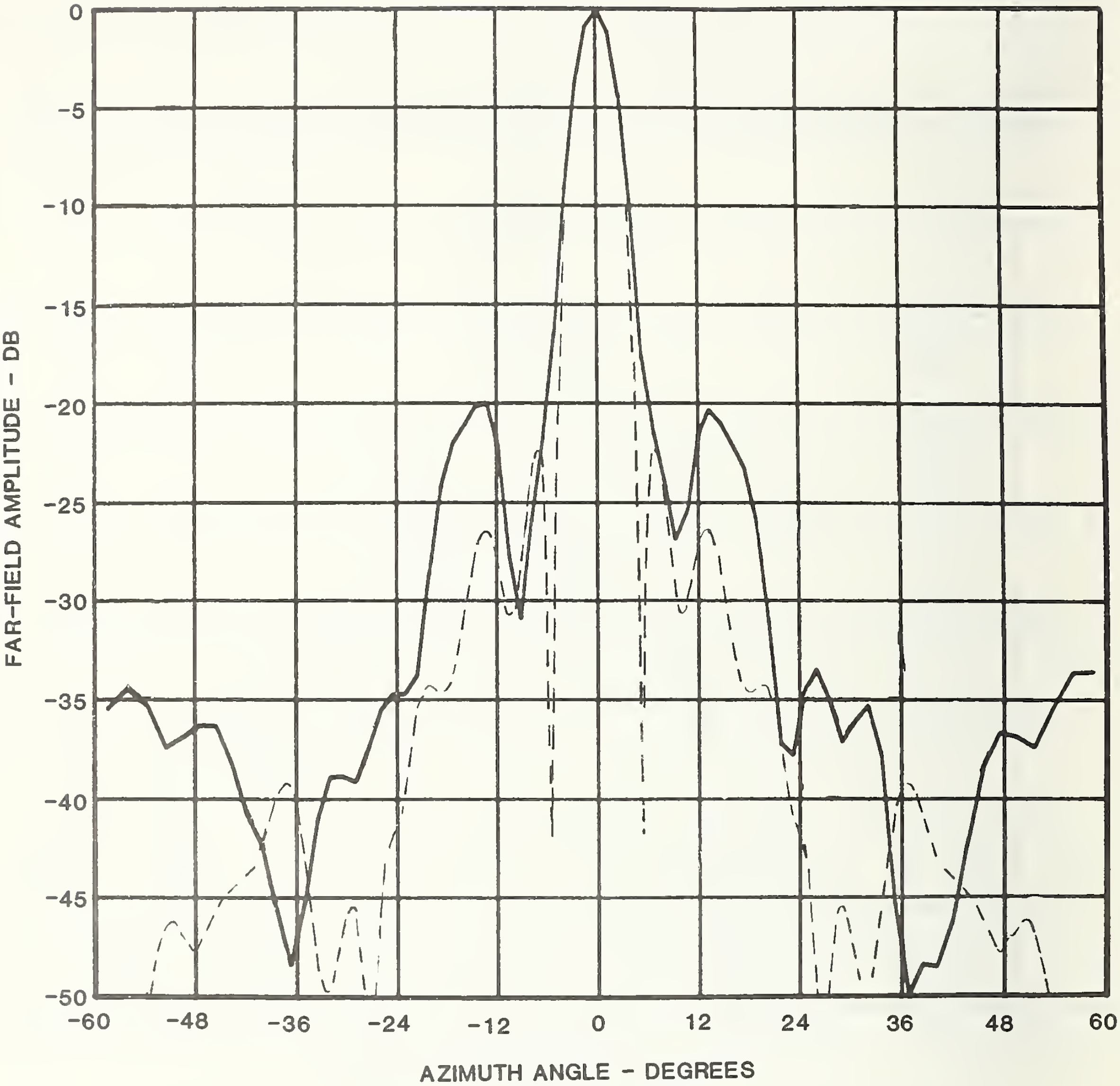


Figure 10b. Comparison of measured and calculated far-field patterns for antenna No. 1. H-plane cut, solid line - measured pattern, dashed line - physical optics.

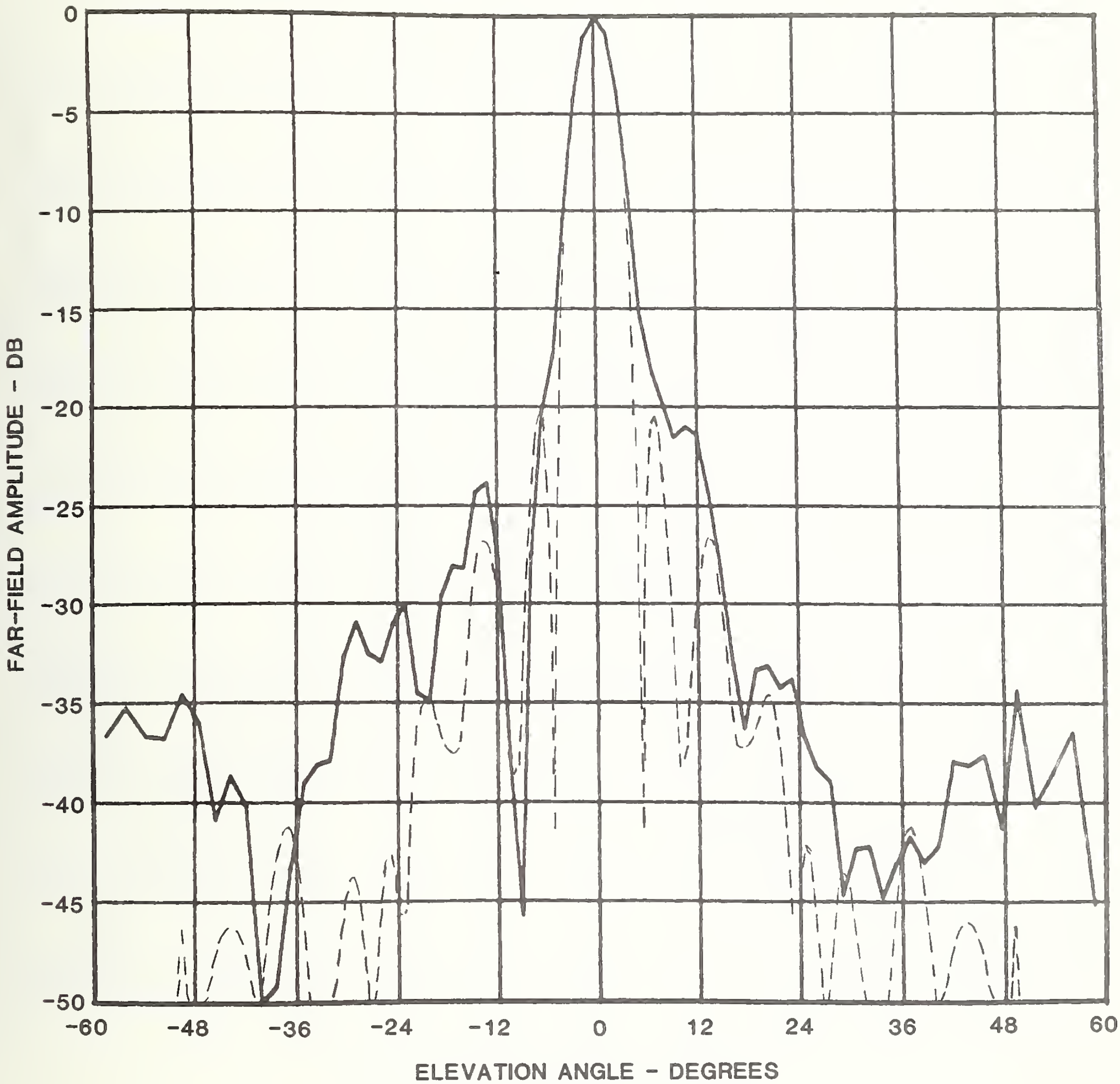


Figure 11a. Comparison of measured and calculated far-field patterns for antenna No. 2. E-plane cut, solid line - measured pattern, dashed line - physical optics.

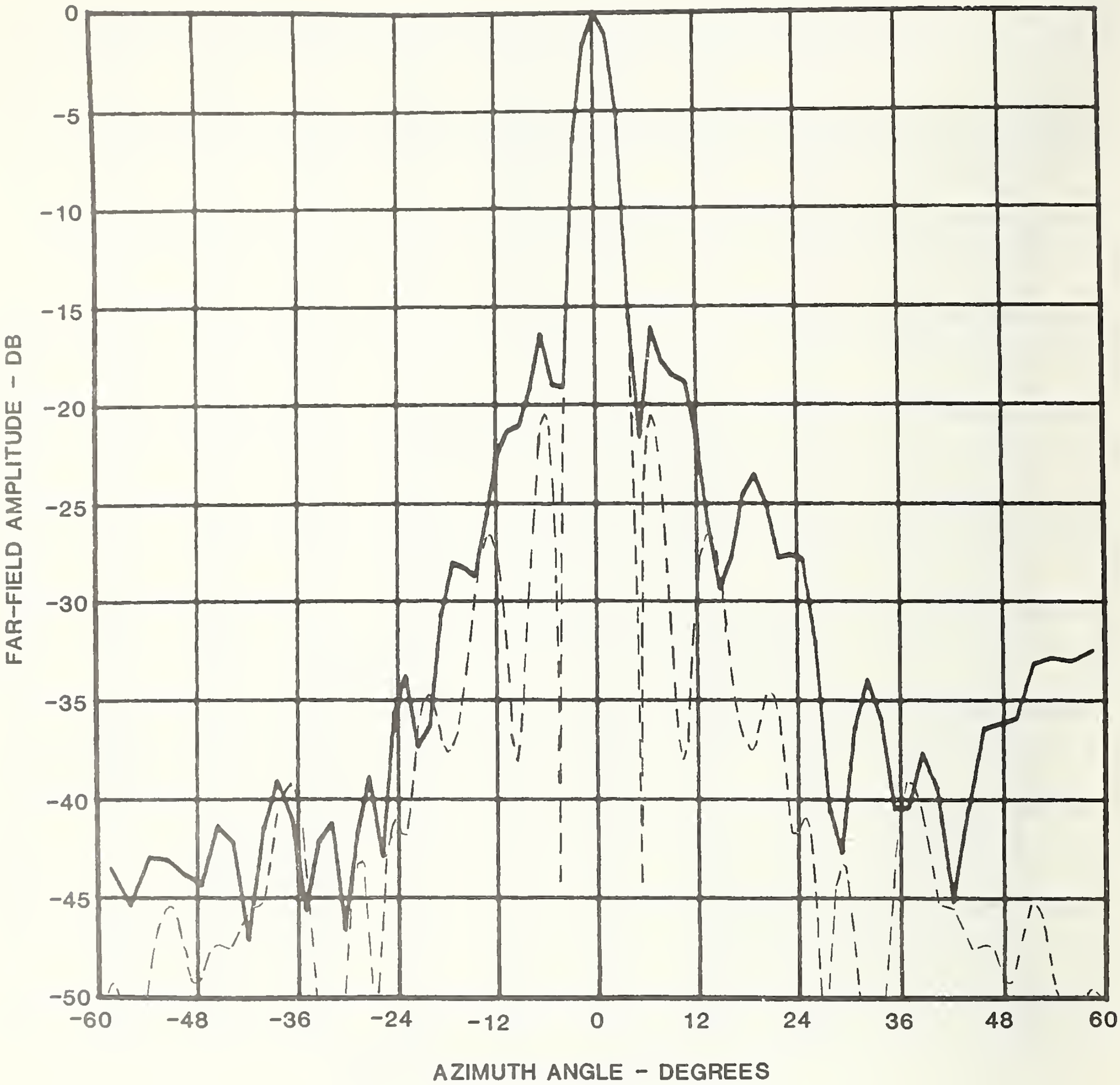


Figure 11b. Comparison of measured and calculated far-field patterns for antenna No. 2. H-plane cut, solid line - measured pattern, dashed line - physical optics.

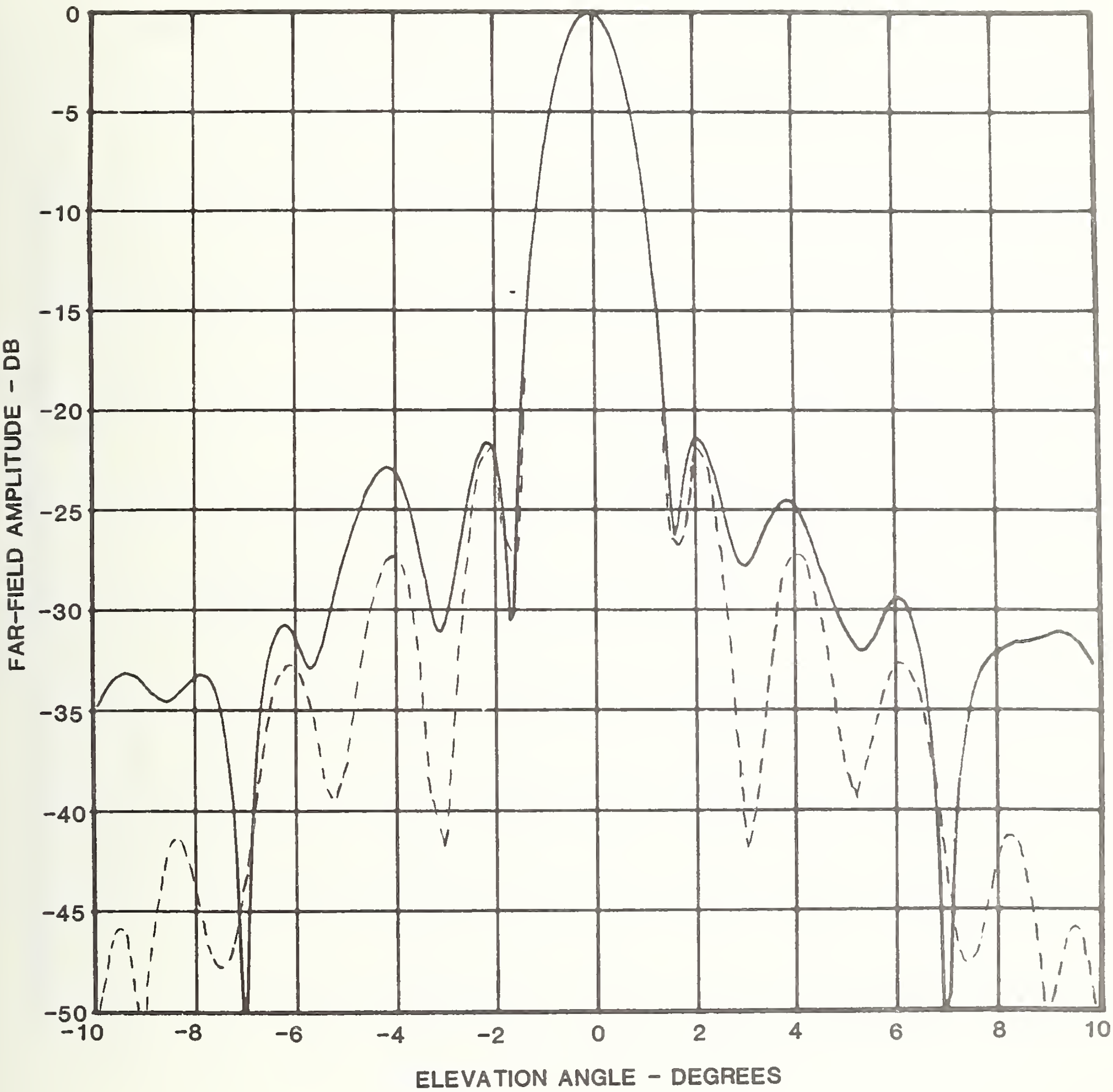


Figure 12a. Comparison of measured and calculated far-field patterns for antenna No. 3. E-plane cut, solid line - measured pattern, dashed line - physical optics.

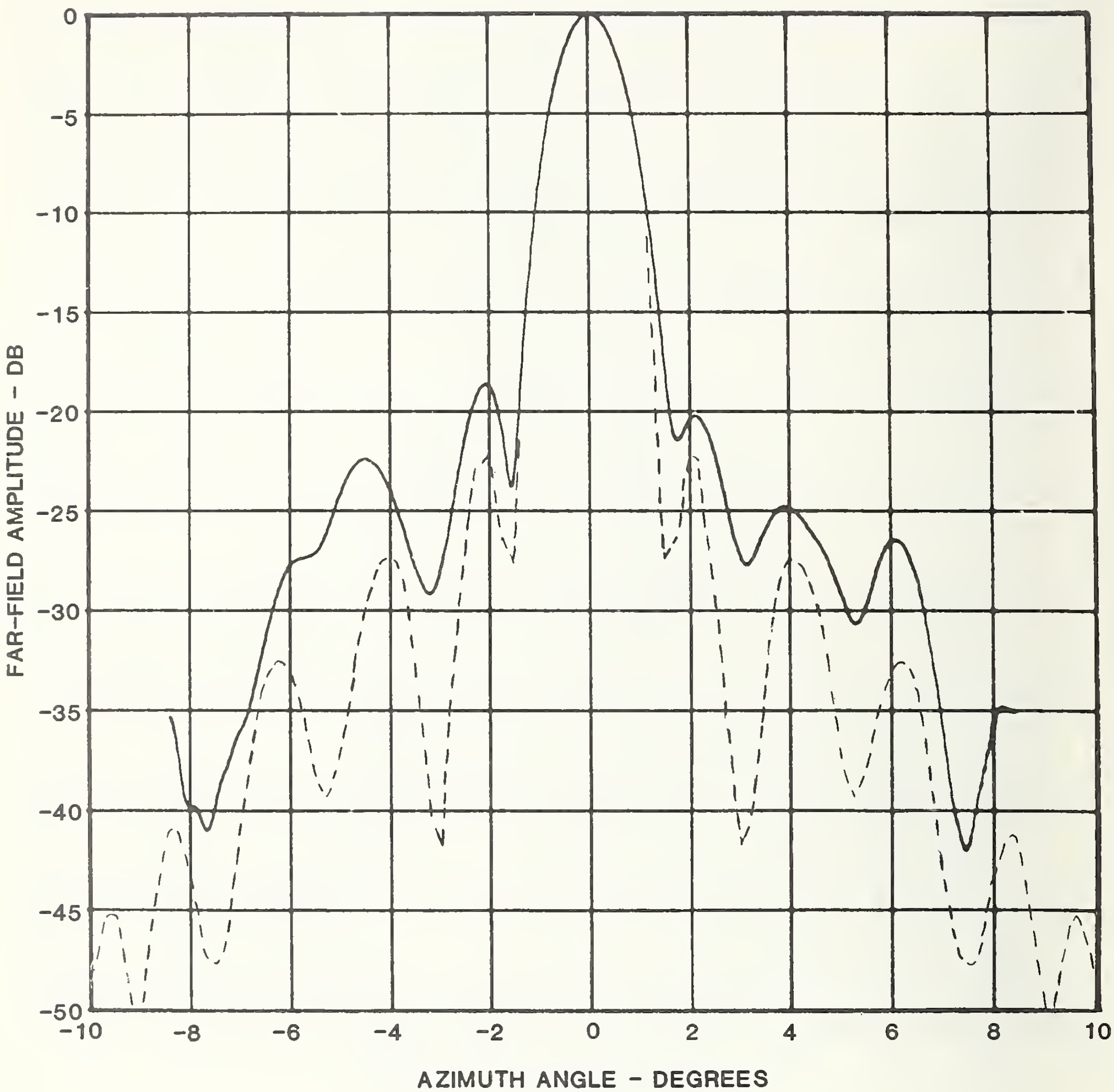


Figure 12b. Comparison of measured and calculated far-field patterns for antenna No. 3. H-plane cut, solid line - measured pattern, dashed line - physical optics.

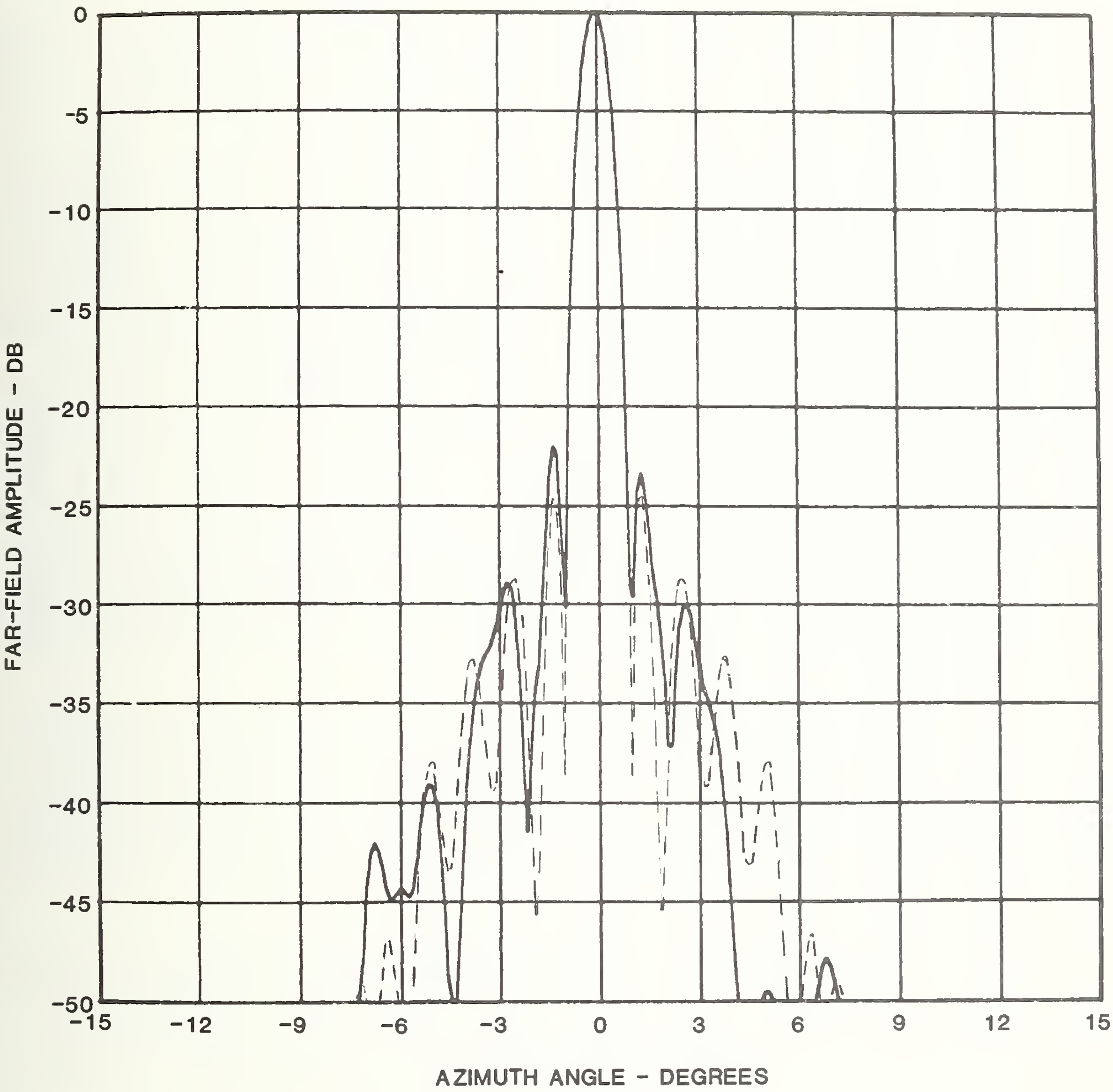


Figure 13. Comparison of measured and calculated far-field patterns for antenna No. 4. H-plane cut, solid line - measured pattern, dashed line - physical optics.

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.

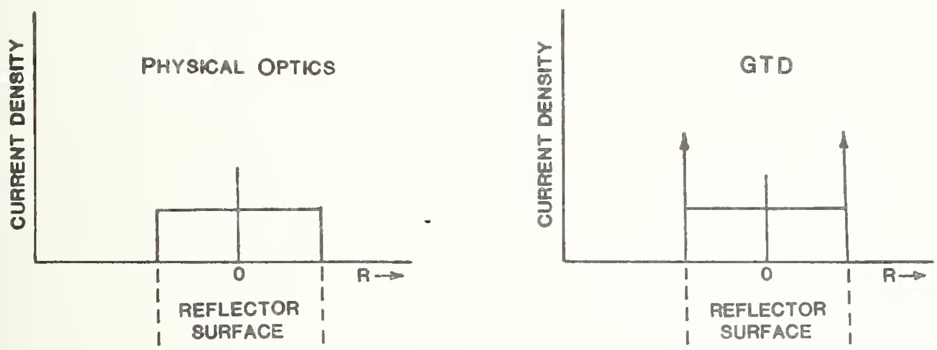


Figure 14. Comparison of effective current distribution used in physical optics and geometrical theory of diffraction calculations. (Uniform distribution assumed).

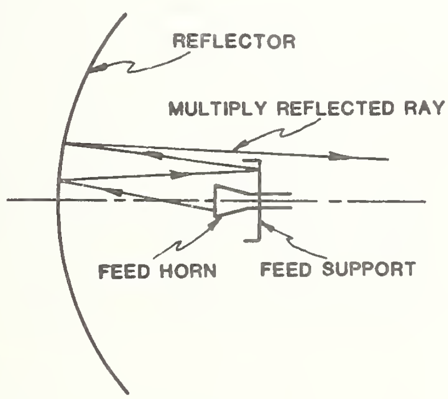


Figure 15. Diagram of multiple reflections involving feed structure.

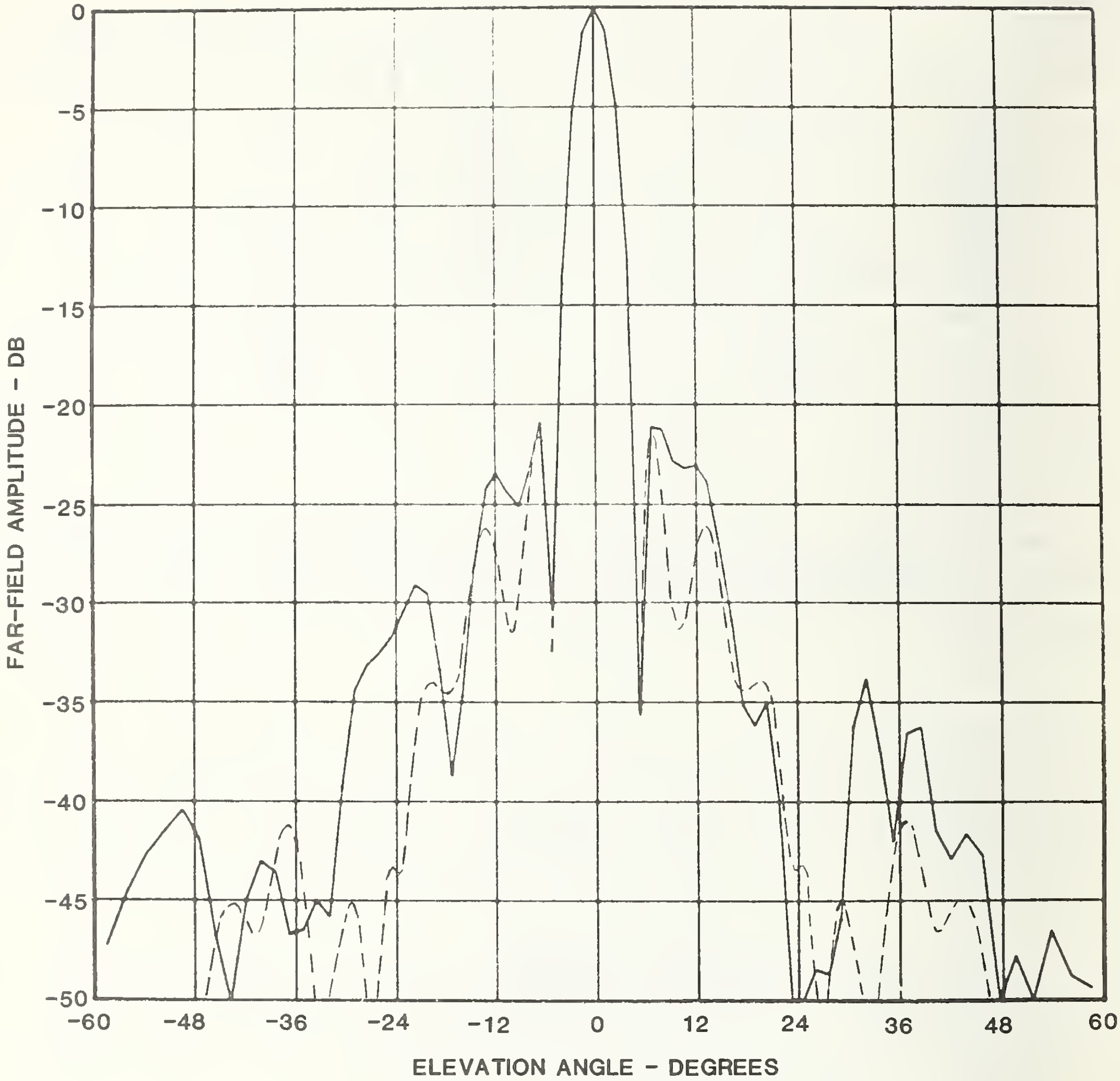


Figure 16a. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. E-plane cut, solid curve - measured pattern, dashed curve - physical optics.

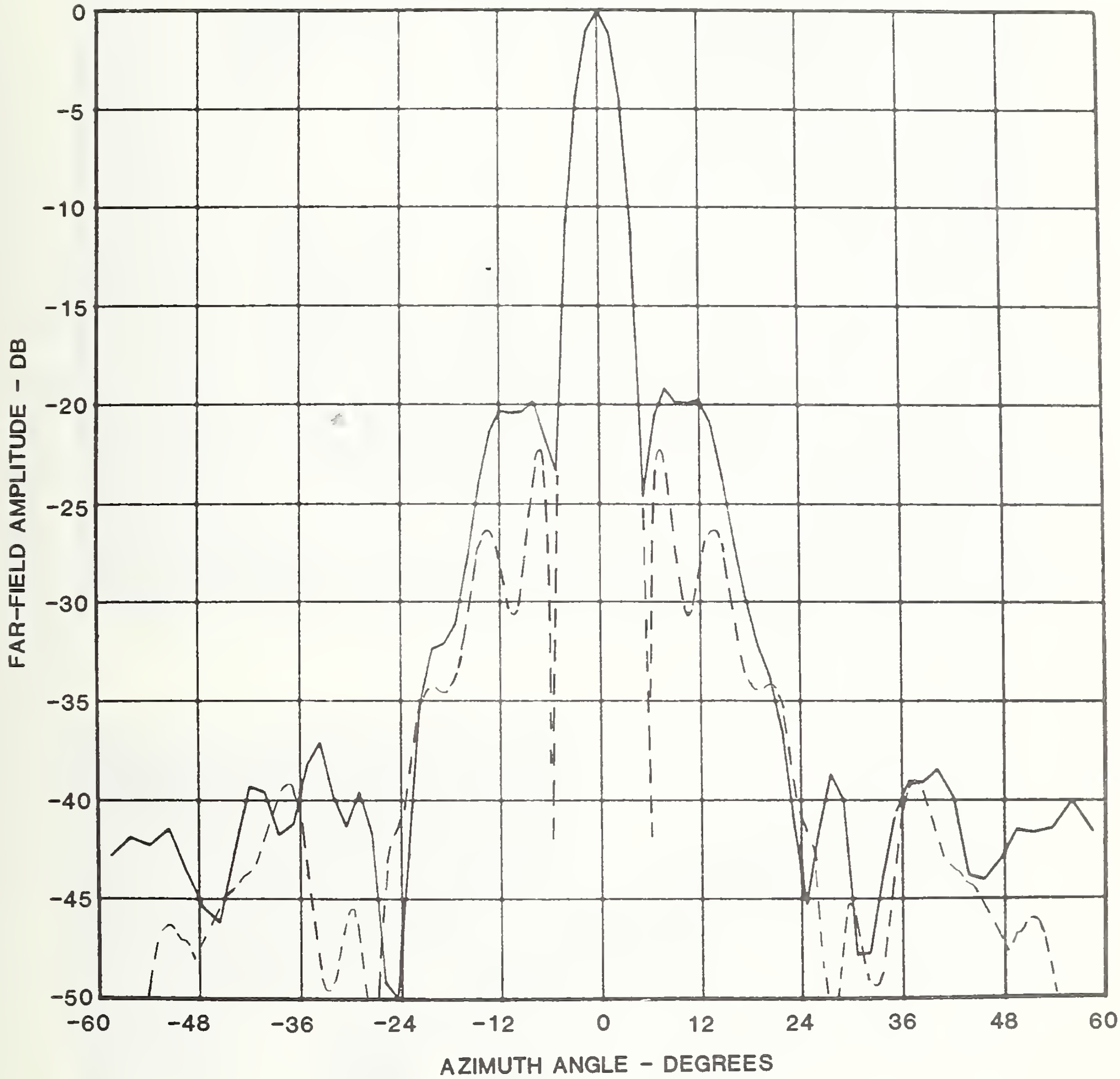


Figure 16b. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. H-plane cut, solid curve - measured pattern, dashed curve - physical optics.

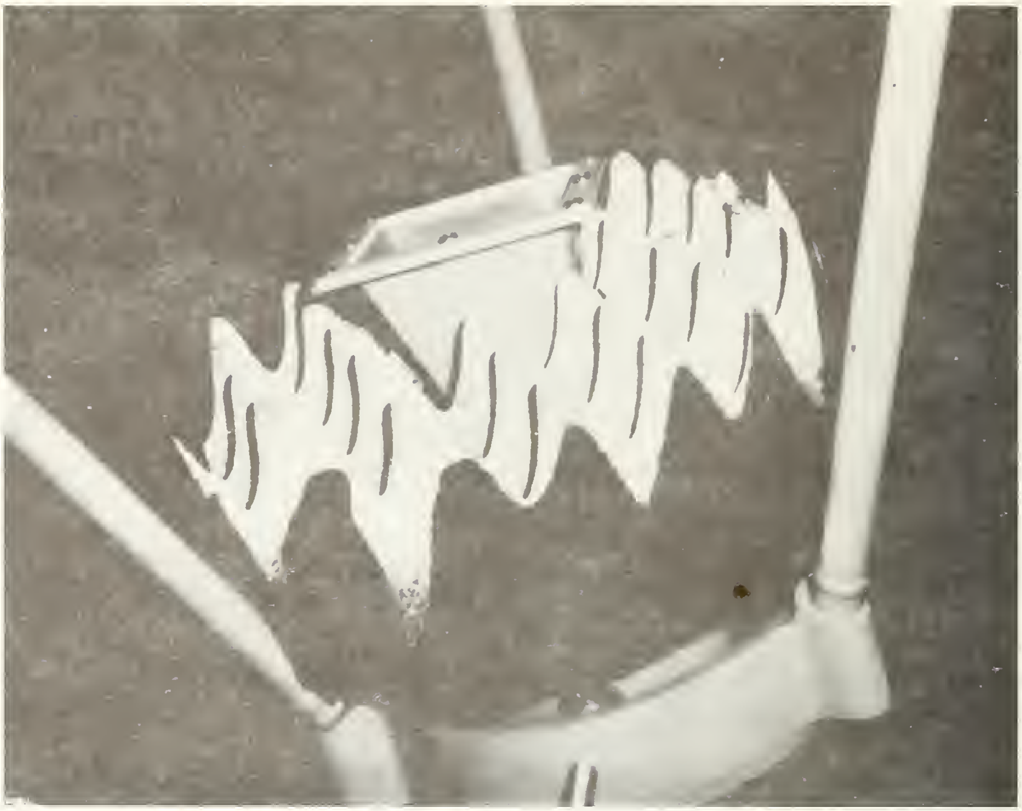


Figure 17a. Feed region of antenna with absorber collar.

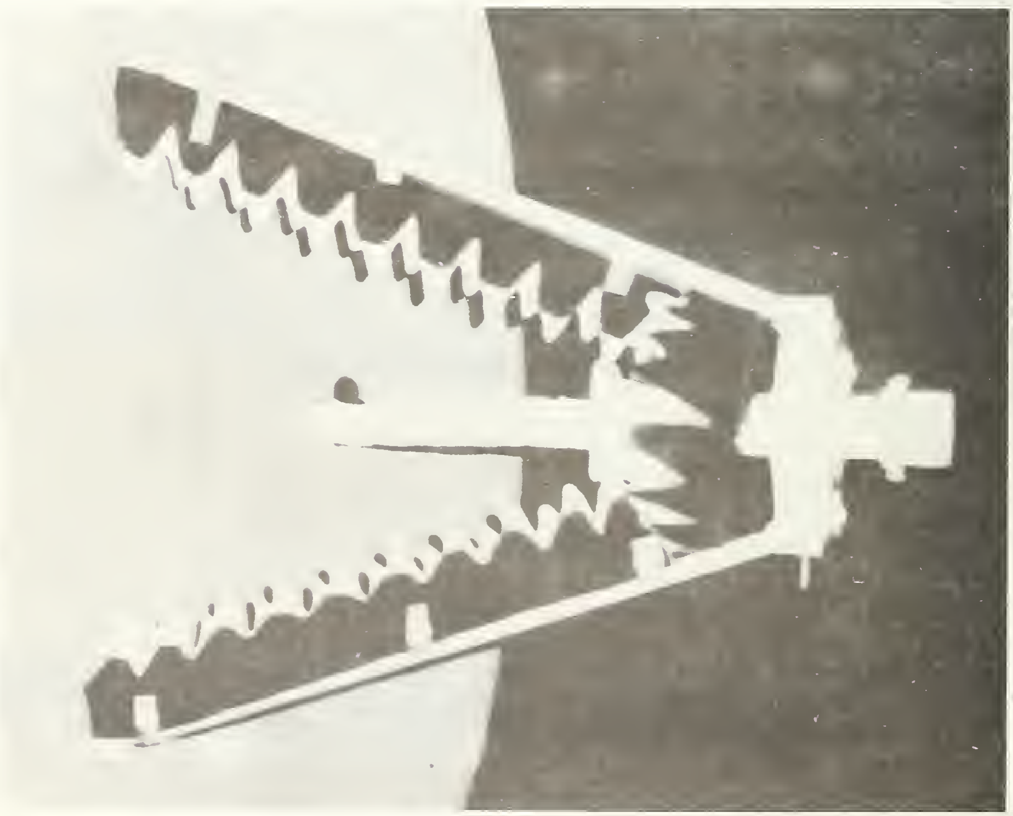


Figure 17b. Feed support struts with absorber attached.

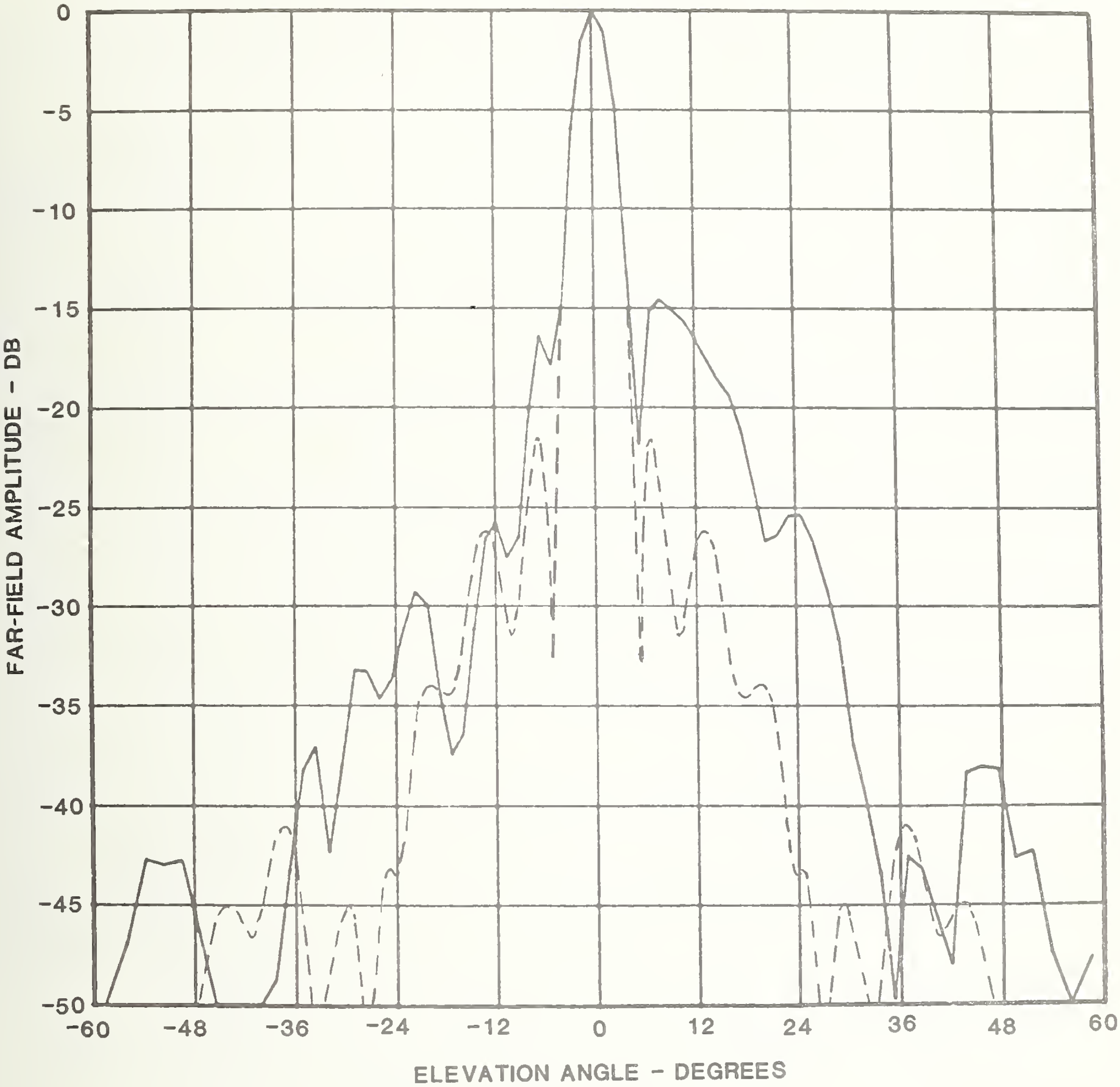


Figure 18a. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. E-plane cut, solid curve - measured pattern, dashed curve - physical optics.

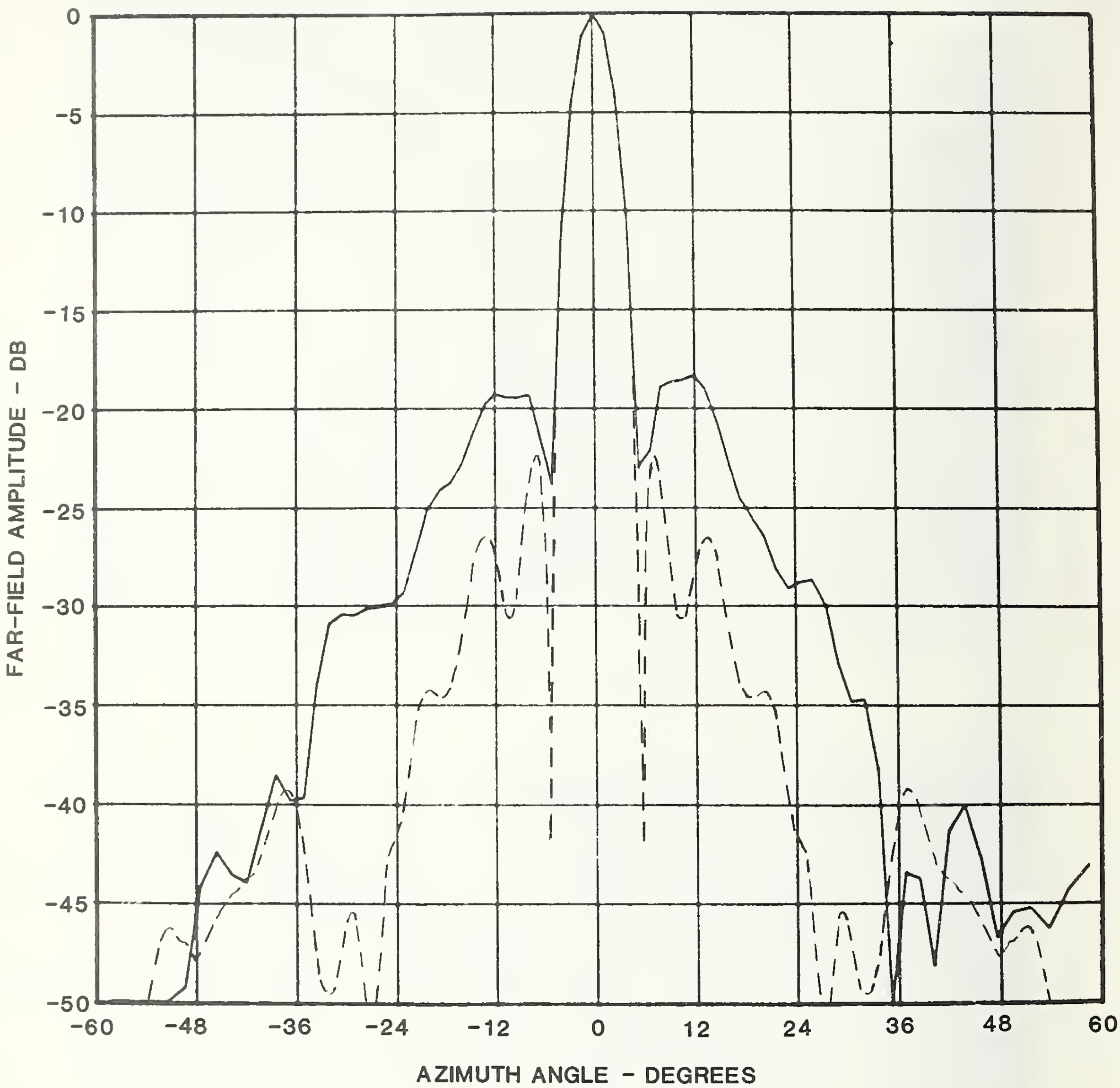


Figure 18a. Comparison of measured and calculated far-field patterns for antenna No. 1 with feed region covered with absorber. H-plane cut, solid curve - measured pattern, dashed curve - physical optics.

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/λ 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 (<50λ) 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 $\pm 4^o$ 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^o to 50^o, and, this too, limits the angles. For wider angle coverage, nonplanar scanning techniqes 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.

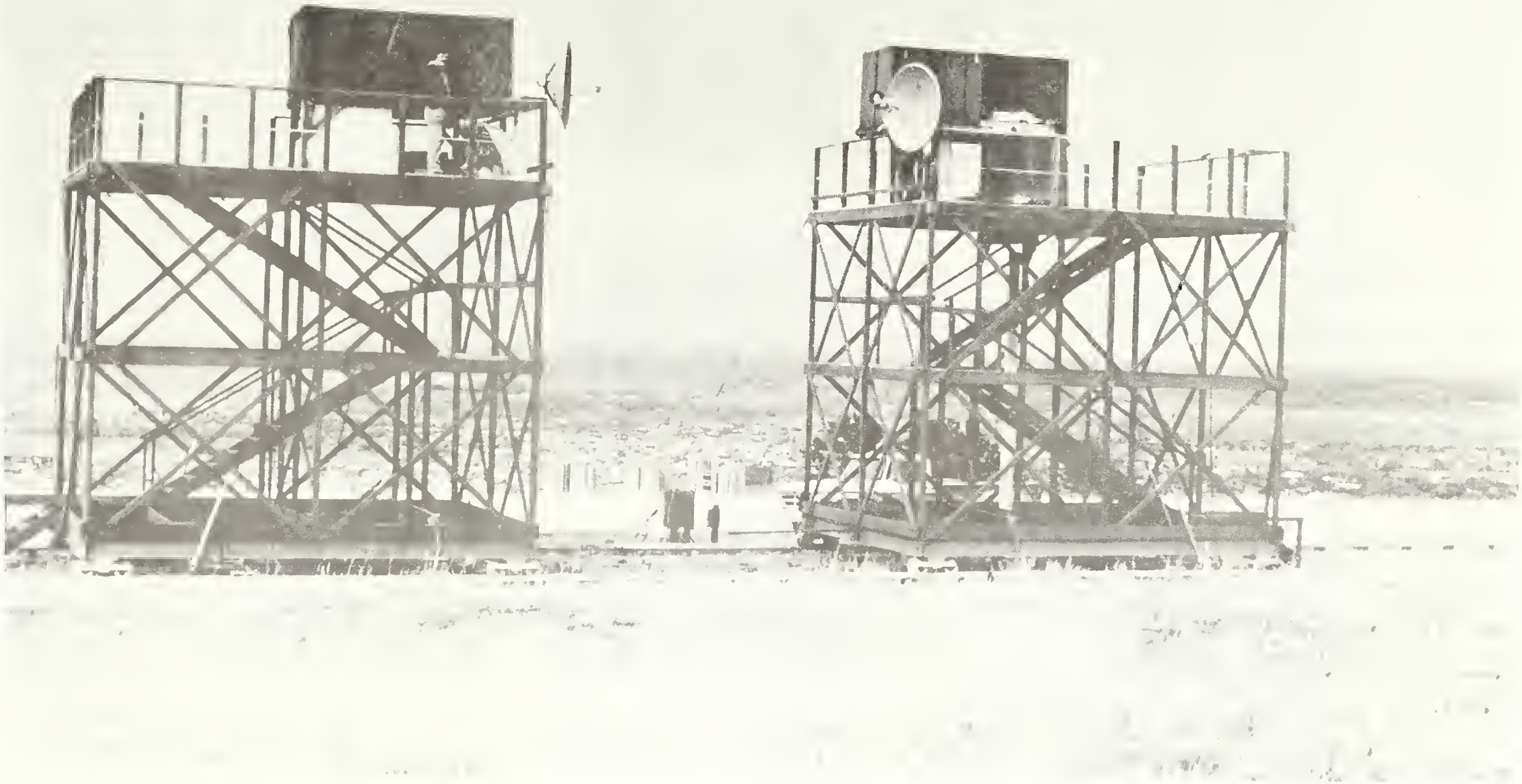


Figure 19. Photograph of experimental set up for measuring coupling between two reflector antennas.

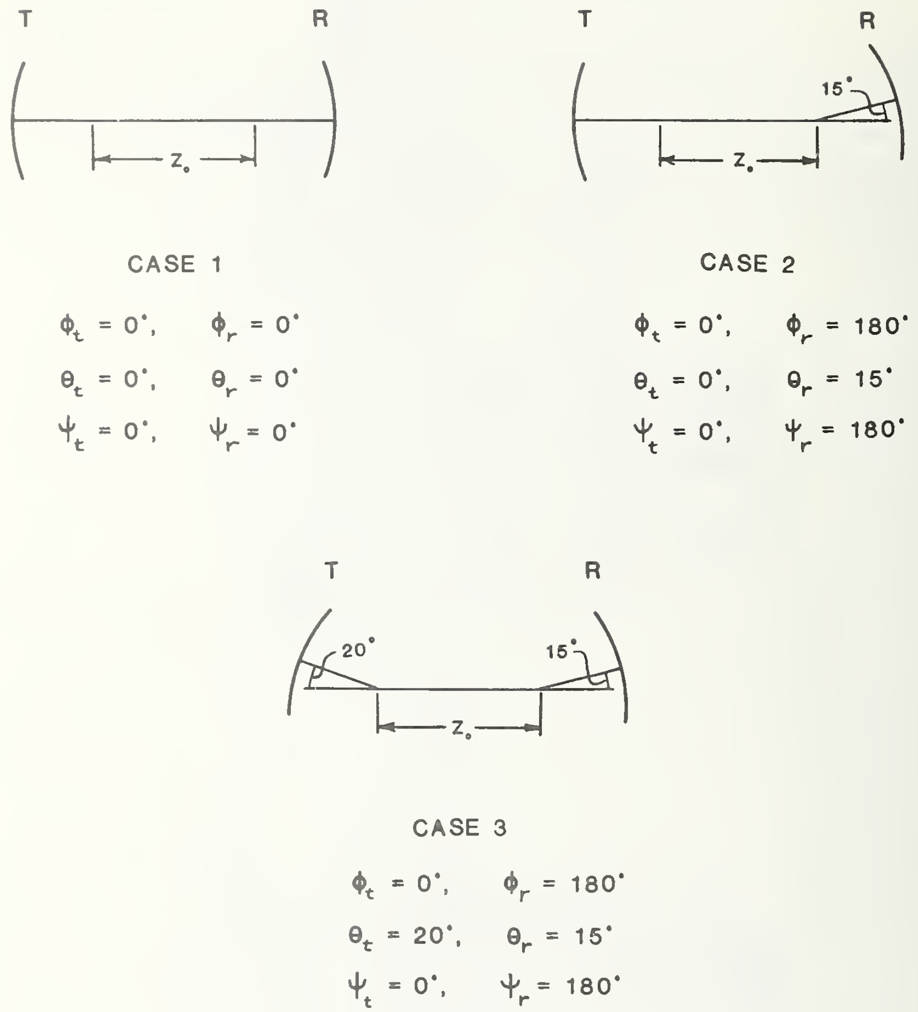


Figure 20. Schematic showing relative orientations of antennas for the three test cases.

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°, 15°) data. This disagreement will be discussed shortly. In the (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°, 15°) 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 $1/2^{\circ}$ about all three axes.

We now discuss the (0°, 15°) 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°. Calculations show that a peak in the predicted coupling occurs at an angle of 12.4° with a magnitude of -26.4 dB. Predicted and observed nulls also occur at about 20° to 22°, 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°, 15°) 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.

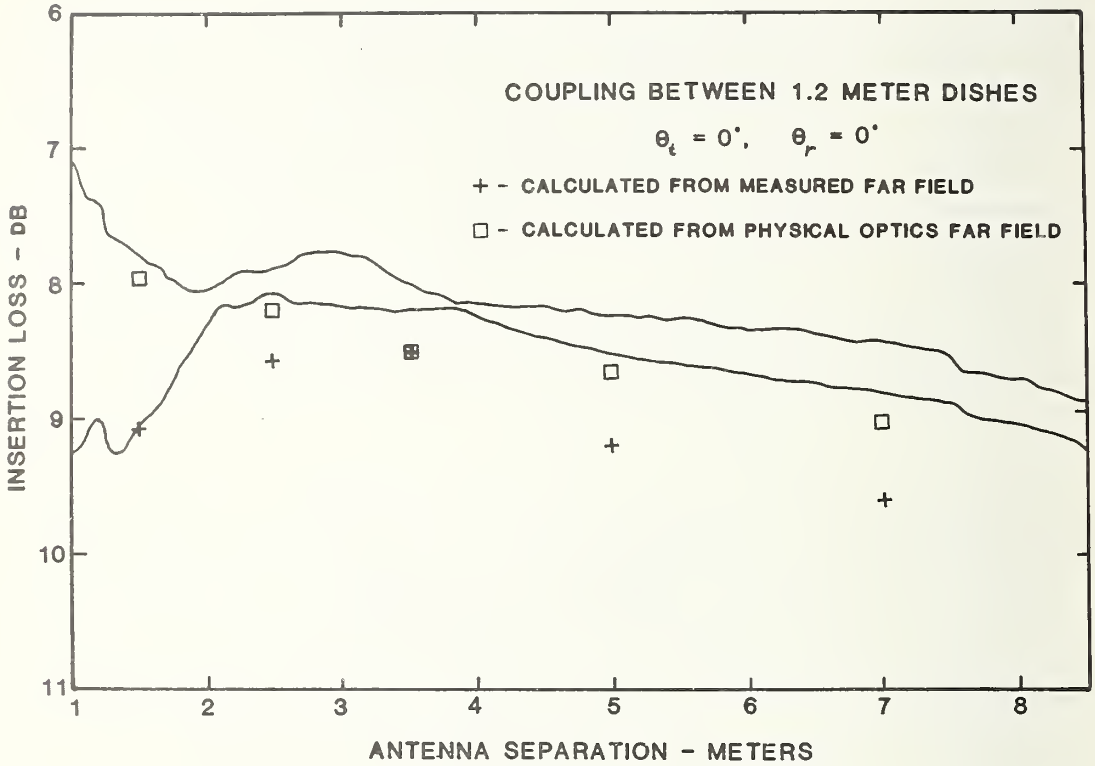


Figure 21. Mutual coupling between 1.2 meter reflector antennas. Case 1: θ_r=0°, θ_t=0°. Solid lines indicate envelope of measured mutual coupling.

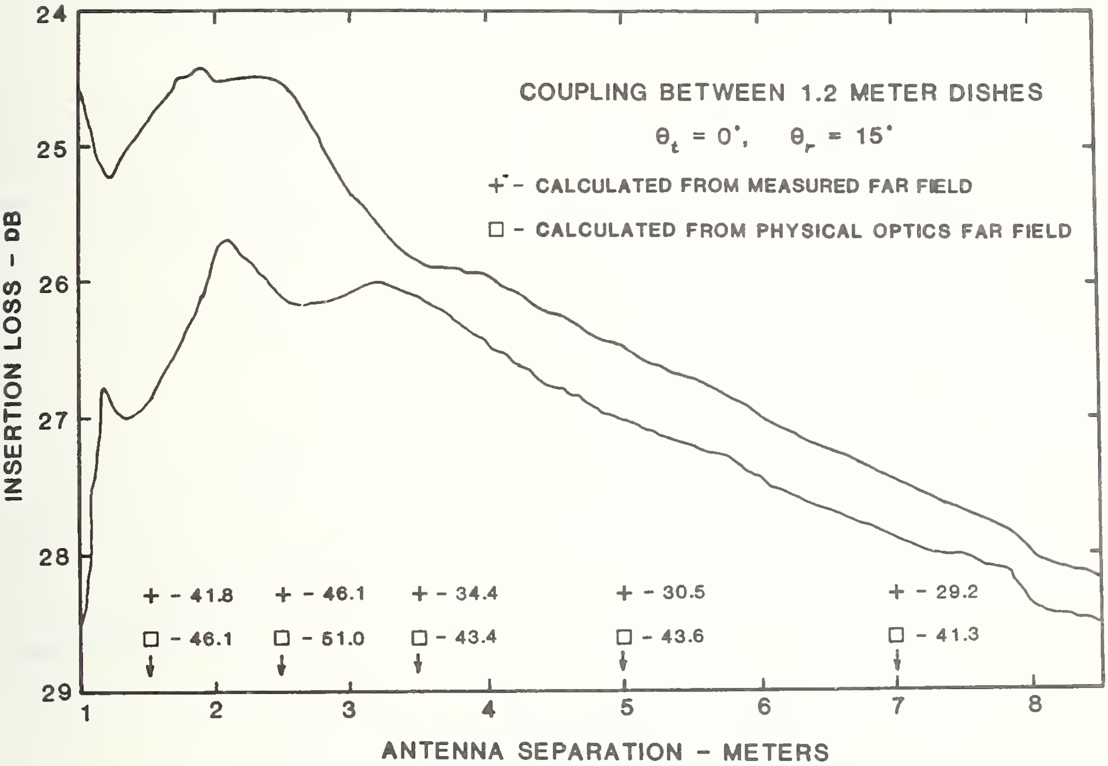


Figure 22. Mutual coupling between 1.2 meter reflector antennas. Case 2: θ_r=15°, θ_t=0°. Solid lines indicate envelope of measured mutual coupling.

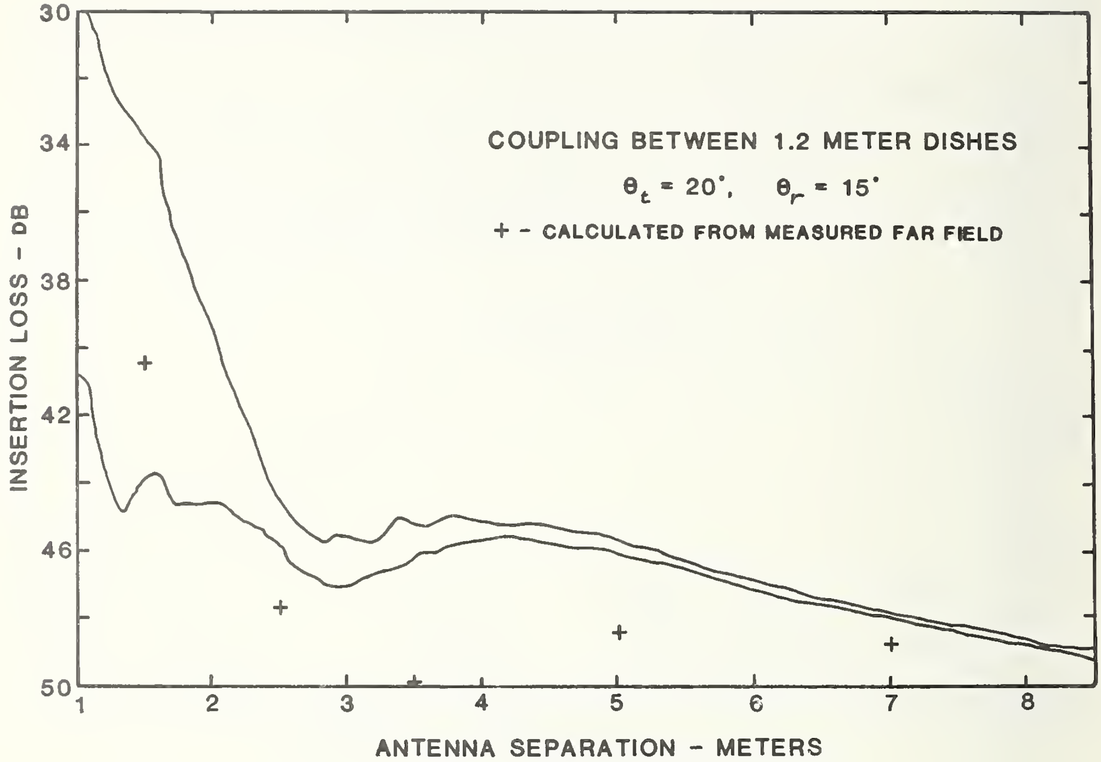


Figure 23. Mutual coupling between 1.2 meter reflector antennas. Case 3: θ_r=15°, θ_t=20°. Solid lines indicate envelope of measured mutual coupling.

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 sophisicated 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 θ 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).

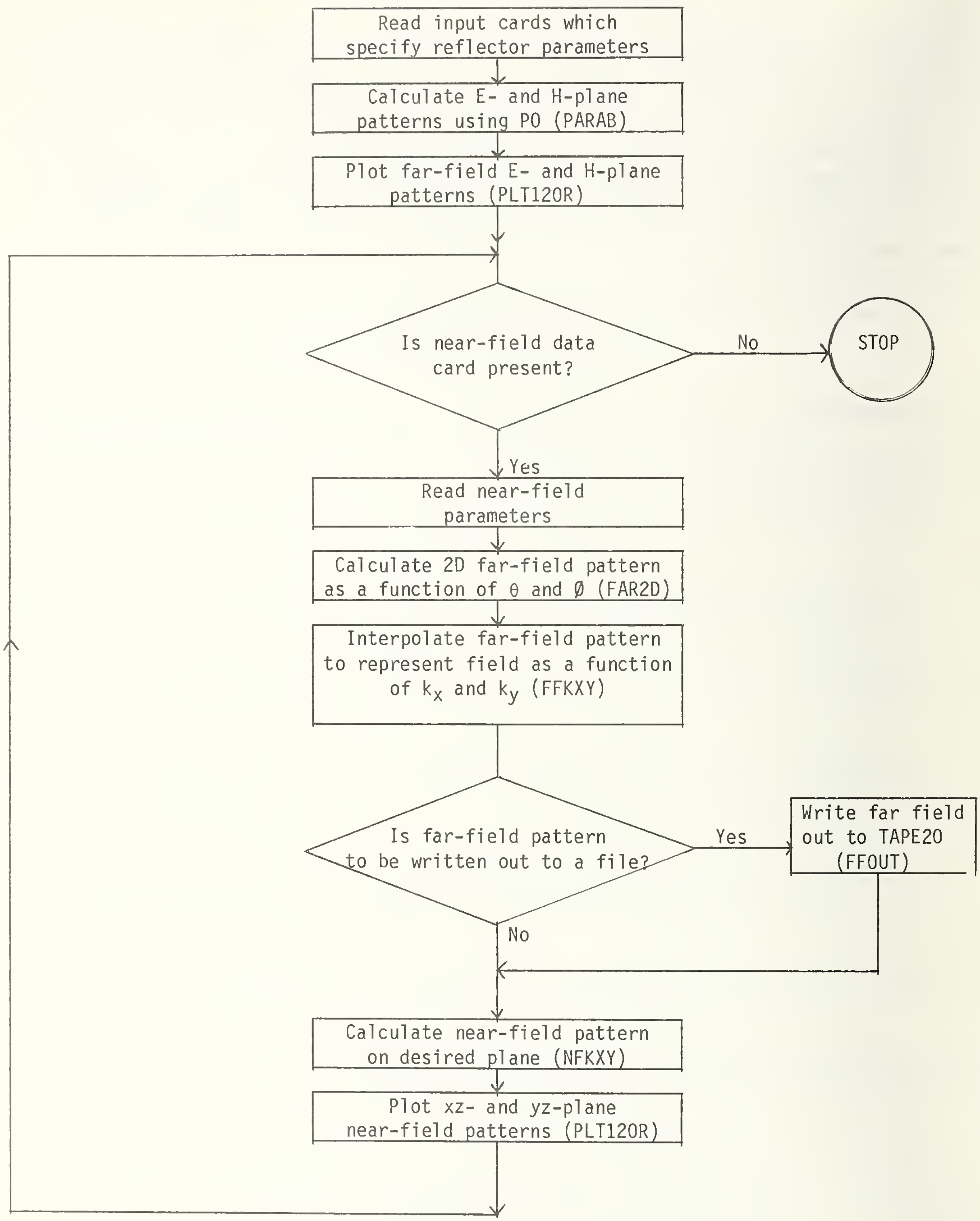
[10] Tai, C-T, Dyadic Green's Functions in Electromagnetic Theory (Intext, New York, N.Y., 1972).

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.


Read input cards which specify reflector parameters
Calculate E- and H-plane patterns using PO (PARAB)
Plot far-field E- and H-plane patterns (PLT120R)
Is near-field data card present?
No
STOP
Yes
Read near-field parameters
Calculate 2D far-field pattern as a function of θ and Ø (FAR2D)
Interpolate far-field pattern to represent field as a function of kx and ky (FFKXY)
Is far-field pattern to be written out to a file?
Yes
Write far field out to TAPE20 (FFOUT)
No
Calculate near-field pattern on desired plane (NFKXY)
Plot xz- and yz-plane near-field patterns (PLT120R)

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 x (number of points to be calculated in θ-direction) x (number of points in Ø-direction). For DATAX, the size must be at least 2 x (number of near-field points in x-direction) x (number of near-field in y-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.

<u>Group I</u>

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 PO. 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 x 10^{-9}
DATAX(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
ETREF	= Magnitude of ETHETA(1) used for normalization purposes
EY(I,J)	= Array reserved for far field versus θ and Ø
FKAY	= Propagation constant = 2π/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 x JTHETA - 1 used for array indexing
JTHETA	= Theta loop index
JTHX2	= 2 x JTHETA 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	= π = 3.14159....
PNRM	= Power normalization factor
PIN	= Input power to antenna
PNRM	= Power normalization factor
RTD	= Radians to degrees conversion factor = π/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

```
      PROGRAM POMODL (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE20) POMODL
C                                                                       POMODL
C     DRIVER PROGRAM FOR PHYSICAL OPTICS SUBROUTINE PARAB, WRITTEN BY   POMODL
C        PROFESSOR W. V. T. RUSCH OF THE UNIVERSITY OF SOUTHERN         POMODL
C        CALIFORNIA, WHICH INCLUDES CAPABILITY OF CALCULATING           POMODL
C        NEAR FIELDS ON A SPECIFIED PLANE.                              POMODL
C                                                                       POMODL
C     Z-AXIS IS AXIS OF SYMMETRY POINTING AWAY FROM PARABOLOID. X IS    POMODL
C        POLAR ANGLE THETA-PRIME MEASURED FROM THE POSITIVE-Z AXIS.     POMODL
C        X = PI IS THE DIRECTION OF THE REFLECTOR VERTEX.               POMODL
C                                                                       POMODL
C     XP IS THE POLAR ANGLE THETA-DOUBLE-PRIME MEASURED FROM THE        POMODL
C        POSITIVE Z-AXIS WITH THE DEFOCUSED FEED AS ORIGIN.             POMODL
C                                                                       POMODL
C     THE FIELDS OF THE FEED ARE THE FIELDS OF A CIRCULAR APERTURE      POMODL
C        EXCITED IN THE M=1 AZIMUTHAL MODE. THE E-PLANE IS THE          POMODL
C        YZ-PLANE AND THE H-PLANE IS THE XZ-PLANE. THE COMPLEX POLAR    POMODL
C        PATTERNS A1(TP) AND D1(TP) ARE SUCH THAT MOST OF THE POWER     POMODL
C        IS RADIATED TOWARD THE REFLECTOR AND VERY LITTLE POWER IS      POMODL
C        RADIATED INTO THE HALF-SPACE XP.LT.PI/2. FURTHERMORE, TO       POMODL
C        ASSURE CONTINUITY OF THE FIELD WHEN XP = PI, IT IS NECESSARY   POMODL
C        THAT D1(PI) = -A1(PI).                                         POMODL
C                                                                       POMODL
C                                                                       POMODL
C     FOD = REFLECTOR F/D                                               POMODL
C     DCL = REFLECTOR DIAMETER IN WAVELENGTHS                           POMODL
C     BLOCK = FRACTIONAL DIAMETER BLOCKING                              POMODL
C     DEFOCUS = AMOUNT OF AXIAL DEFOCUSING BEYOND THE FOCUS IN WAVELENGTHPOMODL
C     IF(ACOSE.LT.(-100.0)) THE APERTURE IS UNIFORMLY ILLUMINATED       POMODL
C     IF(ACOSE.GE.(-100.0).AND.LT.0.0) THE FEED IS Y-DIRECTED ELECTRIC  POMODL
C        DIPOLE                                                         POMODL
C     IF(ACOSE.GE.0.0) A1 = (COS(PI-XP))**ACOSE, XP.GE.PI/2             POMODL
C                         = 0, XP.LT.PI/2                               POMODL
C                      D1 =-(COS(PI-CP))**ACOSH, XP.GE.PI/2             POMODL
C                         = 0, XP.LT.PI/2                               POMODL
C     FREQ = FREQUENCY                                                  POMODL
C                                                                       POMODL
C     THETAF = INITIAL VALUE OF THETA, DEGREES                          POMODL
C     DTHETA = DIFFERENTIAL VALUE OF THETA                              POMODL
C     PHIF = INITIAL VALUE OF PHI                                       POMODL
C     DLPH = DIFFERENTIAL VALUE OF PHI                                  POMODL
C     NTHETA = NUMBER OF THETA VALUES                                   POMODL
C                                                                       POMODL
C     PIN = POWER INPUT TO ANTENNA FOR NEAR-ZONE FIELD STRENGTH         POMODL
C     EFF = APERTURE EFFICIENCY OF ANTENNA                              POMODL
C                                                                       POMODL
C                                                                       POMODL
      COMPLEX ETHETA,EPHI                                               POMODL
C                                                                       POMODL
      DIMENSION ETHDB(200), EPHDB(200), THETA(200)                      POMODL
      DIMENSION EPLANE(400), HPLANE(400)                                POMODL
      DIMENSION ID(4)                                                   POMODL
C                                                                       POMODL
      COMMON EY(6000), DATAX(8192)                                      POMODL
      COMMON /ID/ CASEID(8)                                             POMODL
      COMMON /CNTRL/ DTHETA, DLPH, DELX, DELY, FREQ, DIST, PNRM         POMODL
C                                                                       POMODL
C     INPUT                                                             POMODL
      READ (5, 5000) ID                                                 POMODL
      PRINT 6001, ID                                                    POMODL
      READ(5,5000)         CASEID                                       POMODL


      WRITE(6,6000)CASEID                                               POMODL
      READ(5,5020)FOD,DCL,BLOCK,DEFOCUS,ACOSE,ACOSH, FREQ               POMODL
      READ(5, 5040) THETAF, DTHETA, PHIF, DLPH, NTHETA                  POMODL
      READ(5, 5020) PIN, EFF                                            POMODL
      IF (PIN .EQ. 0.)  PIN = 1.0                                       POMODL
      IF (EFF .EQ. 0.) EFF = 1.                                         POMODL
      WRITE(6,6020)FOD,DCL,BLOCK,DEFOCUS, FREQ                          POMODL
      IF(ACOSE.LT.(-100.0))WRITE(6,6005)                                POMODL
      IF(ACOSE.GE.(-100.0).AND.ACOSE.LT.0.0)WRITE(6,6010)               POMODL
      IF(ACOSE.GE.0.0)WRITE(6,6015)ACOSE,ACOSH                          POMODL
C                                                                       POMODL
C     MISCELLANEOUS                                                     POMODL
C                                                                       POMODL
      PI = 4.0*ATAN(1.0)                                                POMODL
      RTD = 180.0/PI                                                    POMODL
      CEE = .2997925                                                    POMODL
```

```
      FKAY = 2.*PI*FREQ/CFF
      NPHI = 90./DLPH + 1.CCD001
      GAIN = EFF*PI*PI*DDL*DDL
      GDB = 10.*ALOG10(GAIN)
      PNRM = SQRT(15.*PIN*GAIN/FKAY/FKAY)/PI
      WRITE(6, 6D02) EFF*1CC., GDB, PIN
      WRITE(6,6030)
C
C     ENTER THE THETA LOOP
C
      DO 10D JTHETA = 1,NTHETA
      THETA(JTHETA) = THETAF + (JTHETA-1)*DTHETA
      CALL PARAR(FOD,DDL,BLOCK,DEFOCUS,ACOSE,ACOSH,THETA(JTHETA), ETHETA,
     1EPHI)
      ETMAG = CABS(ETHETA)
      IF(JTHETA.EQ.1) ETREF = ETMAG
      ETHDB(JTHETA) = 20.0*ALOG10(ETMAG/ETREF)
      ETFAZE = -ATAN2(AIMAG(ETHETA),REAL(ETHETA))*RTD
      EPMAG = CABS(EPHI)
      IF(JTHETA.EQ.1) EPREF = EPMAG
      EPHDB(JTHETA) = 20.D*ALOG10(EPMAG/EPREF)
      EPFAZE = -ATAN2(AIMAG(EPHI),REAL(EPHI))*RTD
      PRINT 6040, THETA(JTHETA), ETMAG, ETHDB(JTHETA), ETFAZE, EPMAG,
     1EPHDB(JTHETA), EPFAZE
      JTHX2 = JTHETA*2
      JTH2M1 = JTHX2 - 1
C
C     NORMALIZE E TO 1.C AT THETA = O. AND CALCULATE S10.
C
      EPLANE(JTH2M1) = CABS(ETHETA)/ETREF/COS(THETA(JTHETA)/RTD)
      EPLANE(JTHX2) = ETFAZE/RTD
      HPLANE(JTH2M1) = CABS(EPHI)/ETREF/COS(THETA(JTHETA)/RTD)
      HPLANE(JTHX2) = EPFAZE/RTD
  100 CONTINUE
C
C        PLOT E AND H-PLANE AMPLITUDES
C
      CALL PLT12DP(THETA, ETHDB, 60., -60., 0., -50., NTHETA, 1H*, 1, 1)
      PRINT 6080, CASEID, 1CH E-PLANE
      CALL PLT12OP(THETA, EPHDB, 60., -60., 0., -5D., NTHETA, 1H*, 1, 1)
      PRINT 6080, CASEID, 1CH H-PLANE
C
C
C     READ DATA FOR NEAR FIELD
C
C        DELX = NEAR FIELD X-SPACING
C        DELY = NEAR FIELD Y-SPACING
C        DIST = Z-POSITION OF NEAR FIELD PLANE.  (DIST = 0. IS FOCAL
C                 PLANE OF PARABOLA)
C        DUMMY - NOT CURRENTLY USED
C        IR2TON = NUMBER OF POINTS IN Y-ARRAY
C        IC2TON = NUMBER OF POINTS IN X-ARRAY
C
    1 READ 5040, DELX, DELY, DIST, DUMMY, IR2TON, IC2TON
      IF (EOF(5)) 200,2
    2 PRINT 6070, DELX*100., IC2TON, DELY*100., IR2TON, DIST*10D.
      CALL FAR2D (EPLANE, HPLANE, EY, NTHETA, NPHI, DATAX, IR2TON*2,
     1IC2TON)
      GO TO 1
  200 WRITE(6,6D60)
 5000 FORMAT(8A10)


 5020 FORMAT(8F10.0)
 5D4D FORMAT(4F10.C,2I5)
 6000 FORMAT(1H1,T7,*RADIATION PATTERN OF AN AXIALLY DEFOCUSED PARABOLOI
     *D*,//,T7,8A10)
 6001 FORMAT(1H , 8A10)
 6002 FORMAT (T16, *ASSUMED EFFICIENCY = *, F10.2, * PERCENT*,/,
     1 T16, *NOMINAL GAIN = *, F10.2, * DB*,/, T16, *POWER INPUT = *,
     2 F10.2, * WATTS*,//)
 6005 FORMAT(1H0,T7,*APERTURE IS UNIFORMLY ILLUMINATED WHEN FEED IS FOCU
     *SED*,/)
 6010 FORMAT(1H0,T7,*THE FEED IS AN ELECTRIC DIPOLE ALONG THE Y-AXIS*,/)
 6015 FORMAT(1HD,T7,≠FEED E-PLANE PATTERN = (COS(Y))**(≠,F5.2,≠)≠,/,
     *T12,≠H-PLANE PATTERN =-(COS(Y))**(≠,F5.2,≠)≠,/)
 6D20 FORMAT(1H ,T7,*REFLECTOR PARAMETERS - *,//,T16,*F/D = *,F5.3,/,
     *T16,*DIAMETER = *,F6.2,* WAVELENGTHS*,/,T16,*FRACTIONAL DIAMETER B
     *LOCKING = *,F5.3,/,T16,*AXIAL DEFOCUSING = *,F6.3,* WAVELENGTHS BE
```

```
      *YOND FOCUS*,/, T16,*FREQUENCY = *,F8.4,*  GHZ.*,/)                   POMODL
 6030  FORMAT(1H0,T30,*E-PLANE*,T62,*H-PLANE*,/,                            POMODL
      *T11,*THETA*,T22,*MAG*,T32,*MAG*,T41,*PHASE*,                         POMODL
      *T54,*MAG*,T64,*MAG*,T73,*PHASE*,/,                                   POMODL
      * T11,*(DEG)*,T20,*(VOLTS)*,T31,*(DB)*,T41,*(DEG)*,                   POMODL
      *T52,*(VOLTS)*,T63,*(DB)*,T73,*(DEG)*,/)                              POMODL
 6040  FORMAT(1H ,T9,F7.2,T20,F6.3,T29,F7.2,T39,F7.2,T52,F6.3,T61,F7.2,     POMODL
      *T71,F7.2)                                                            POMODL
 6060 FORMAT(///,*  EOF FOUND ON LU5*)                                      POMODL
 6070 FORMAT(///, T7, *NEAR-FIELD PARAMETERS-*,//,T16, *X-SPACING =*,       POMODL
      1F6.2, * CM*, T50, I6,* POINTS*,/,T16, *Y-SPACING =*, F6.2, * CM*,    POMODL
      2T50,I6, * POINTS*,/, T16, *DISTANCE FROM REFLECTOR FOCAL POINT =*,   POMODL
      3F8.2, * CM AWAY FROM REFLECTOR SURFACE*,//)                          POMODL
 6080 FORMAT(/, 5X, 8A10, 5X, A10)                                          POMODL
      END                                                                   POMODL
```

A.1.2 SUBROUTINE FAR2D(EPL,HPL,EY,NTHETA,NPHI,DATAX,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.
DATAX 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 θ and $\emptyset$ given the E- and H-plane patterns ($\emptyset$=0, $\pi/2$) as a function of θ.

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

SYMBOL DICTIONARY:

COSTH	= cos(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 θ and Ø.
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 x IR2TON
IROW	= Row loop index
NPHI	= Number of data output points in phi direction
NPHIM1	= NPHI - 1
NTHETA	= Number of data output points in theta direction
NTHM1	= NTHETA - 1
NTHX2	= 2 x NTHETA
PHI	= Azimuth angle - far-field pattern coordinate
PI	= π = 3.14159....
RTD	= Radian to degree conversion factor = $180/\pi$
SINPH	= sin(PHI)
TEMI	= Intermediate variable
TEMR	= Intermediate variable
THETA	= Polar angle from boresight - far-field pattern coordinate

```
      SUBROUTINE FAR2D(EPL, HPL, EY, NTHETA, NPHI, DATAX, IR2X2, IC2TON)FAR2D
C-                                                                      FAR2D
C-    THIS SUBROUTINE TAKES E-PLANE AND H-PLANE DATA GENERATED AS A     FAR2D
C-    FUNCTION OF ANGLE FROM BORESIGHT, AND GENERATES A TWO-DIMENSIONAL FAR2D
C-    ARRAY OF DATA AS A FUNCTION OF THETA AND PHI WHERE THETA AND PHI  FAR2D
C-    ARE THE USUAL SPHERICAL ANGLES DEFINED IN A COORDINATE SYSTEM     FAR2D
C-    WHOSE POLAR AXIS COINCIDES WITH BORESIGHT.                        FAR2D
C-                                                                      FAR2D
C-    *NOTE*  THIS SUBROUTINE ONLY PRODUCES VALID RESULTS FOR ANTENNAS  FAR2D
C-    WHICH HAVE SEPARABLE FAR-FIELD PATTERNS.                          FAR2D
C-                                                                      FAR2D
      DIMENSION EPL(1), HPL(1), EY(NTHETA,NPHI), DATAX(IR2X2, IC2TON)   FAR2D
      COMMON /CNTRL/ DLTH, DLPH, DELX, DELY, FREQ, DIST, PNRM           FAR2D
      COMPLEX EY                                                        FAR2D
      PI = 4.*ATAN(1.)        $   RTD = 180./PI    $   DTR = PI/ 180.   FAR2D
      NTHX2 = 2*NTHETA                                                  FAR2D
      IR2TON = IR2X2/2                                                  FAR2D
      DO 10 I = 1, NTHETA                                               FAR2D
        TEMR = EPL(2*I - 1)*COS(EPL(2*I))*PNRM                          FAR2D
        TEMI = EPL(2*I - 1)*SIN(EPL(2*I))*PNRM                          FAR2D
        EY(I, 1) = CMPLX(TEMR, TEMI)                                    FAR2D
        TEMR = HPL(2*I - 1)*COS(HPL(2*I))*PNRM                          FAR2D
        TEMI = HPL(2*I - 1)*SIN(HPL(2*I))*PNRM                          FAR2D
        EY(I, NPHI) = CMPLX(TEMR,TEMI)                                  FAR2D
   10   CONTINUE                                                        FAR2D
      NTHM1 = NTHETA - 1      $   NPHM1 = NPHI - 1                      FAR2D
      DO 20 IROW = 1, NTHETA                                            FAR2D
        THETA = (IROW - 1)*DLTH*DTR                                     FAR2D
        COSTH = COS(THETA)                                              FAR2D
        DO 30 ICOL = 2, NPHM1                                           FAR2D
          PHI = (ICOL - 1)*DLPH*DTR                                     FAR2D
          SINPH = SIN(PHI)                                              FAR2D
          EY(IROW, ICOL) = (EY(IROW, 1)*COSTH - EY(IROW, NPHI))*SINPH*  FAR2D
     1    SINPH + EY(IROW, NPHI)                                        FAR2D
   30     CONTINUE                                                      FAR2D
   20   CONTINUE                                                        FAR2D
      CALL FFKXY (EY, NTHX2, NPHI, DATAX, IR2TON*2, IC2TON)             FAR2D
      RETURN                                                            FAR2D
      END                                                               FAR2D
```

A.1.3 SUBROUTINE FFKXY(DATAY,NTHX2,NPHI,DATAX,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 θ and $\emptyset$.

ARGUMENTS:

DATAY is a two-dimensional array of far-field values, expressed as a function of equally spaced θ 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 θ direction.

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

DATAX 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 DATAX array.

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

METHODS:

FFKXY is basically an interpolation routine which fills each point in the DATAX array, by calculating the corresponding values of θ and $\emptyset$ locating the four nearest points corresponding to these values in the DATAY array. The value stored in DATAX 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 δ_x, δ_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 θ and Ø corresponding to k_x and k_y are calculated. These are given by

$$\theta = \cos^{-1}\sqrt{1 - k_x^2/k^2 - k_y^2/k^2}$$

$$= \tan^{-1}(k_y/k_x)$$

The indices corresponding to the four elements in the DATAY array that lie closest to the value of θ and Ø 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°. 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°. 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 form 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 x 10^{-9}
D33J1	= Intermediate variable used in phase test
D43J	= Intermediate variable used in phase test
D43J1	= Intermediate variable used in phase test

DATAX(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 θ and Ø

DFI = Fractional part of FI

DFJ = Fractional part of FJ

DLPHI = Ø increment in radians

DLTHTA = θ increment in radians

DTEMP1 = Intermediate variable

DTEMP2 = Intermediate variable

DTEMP3 = Intermediate variable

DTR = Degree to radian conversion factor = π/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 DATAX

IC2TON = Number of points in k_x direction in DATAX array

ICN = Row counter for filling remaining three quadrants of DATAX

IR = Row interpolation loop index

IR2 = Intermediate index

IR2D2 = IR2TON/2

IR2TON = Number of rows in DATAX array

IR2X2 = 2 x IR2TON

IRN = Row counter for filling remaining three quadrants of DATAX

IRX = Index for center row of far-field array

J = Integer part of FJ

NPHI = Number of points in Ø direction in DATAY array

NTHX2 = 2 x Number of points in θ direction in DATAY

PHI = Ø = Azimuth angle in far-field

PHIO = Initial value of Ø

PI = π = 3.14159....

PIX2 = 2π

THETAO = Initial value of θ

THMAX = Maximum value of θ in radians

TST = Test variable to determine if z-component of propagation vector is real

```
      SUBROUTINE FFKXY(DATAY, NTHX2, NPHI, DATAX, IR2X2, IC2TON)
C-
C-    THIS SUBROUTINE INTERPOLATES AN ARRAY OF FAR-FIELD DATA WHICH
C-    IS EQUALLY SPACED IN THETA AND IN PHI TO PRODUCE AN ARRAY WHICH
C-    IS EQUALLY SPACED IN KX AND KY.
C-
      COMMON /CNTRL/ DLTH, DLPH, DFLX, DFLY, FREQ, DIST, PNRM
      DIMENSION DATAY(NTHX2 , NPHI), C(2), DATAX(IR2X2, IC2TON)

      LUOUT = 20
      IR2TON = IR2X2/2
      PI = 4. * ATAN(1.)      $  PIX2 = 2. * PI
      CEE = .2997925        $  FLMDA = CEE/FREQ
      DTR = PI/180.
      FKAY = PIX2/FLMDA       $   FKAYSQ = FKAY*FKAY
      THETO = 0.              $   PHIO = 0.
      DLTHTA = DLTH*DTR       $   DLPHI = DLPH*DTR
      THMAX = (NTHX2/2 - 2)*DLTHTA
      C(1) = PIX2/(DFLX*IC2TON)
      C(2) = PIX2/(DFLY*IR2TON)
      IC2D2 = IC2TON/2        $   IR2D2 = IR2TON/2
C-
C-   CHANGE DATAY ARRAY FROM RECTANGULAR TO POLAR FORM.
C-
      CALL ARAYRTP(DATAY, NTHX2, NPHI)
      ICN = 0
      DO 61 IC = IC2D2,IC2TON
      FKX = C(1)*(IC - IC2D2)      $    FKXSQ = FKX*FKX
      IRN = 0
      DO 62 IR = IR2D2, IR2TON
      IR2 = IR*2 - 1
      FKY = C(2)*(IR - IR2D2)       $    FKYSQ = FKY*FKY
      TST = FKAYSQ - FKXSQ - FKYSQ
      IF (TST .LT. 0.)  GO TO 80
      THETA = ACOS((SQRT(FKAYSQ - FKXSQ - FKYSQ))/FKAY)
      IF (THETA .GT. THMAX) GO TO 80
      IF (FKY .LT. 0.)   THETA = -THETA
      IF (FKX .EQ. 0.   .AND.  FKY .EQ. 0.)    GO TO 63
      PHI = ATAN2(FKY, FKX)
      GO TO 64
   63 PHI = 0.
   64 IF (PHI .LT. 0.)   PHI = PHI + PI
C-
C-   INTERPOLATE DATAY ARRAY TO PRODUCE DATAX ARRAY WHICH IS EQUALLY
C-   SPACED IN KX AND KY.
C-
C FIND THE INDICES FOR THE INPUT DATA WHICH IDENTIFY THE COORDINATES
C CLOSEST TO THE DESIRED THETA AND PHI VALUES. INTERPOLATE TO FIND THE
C PROBE PATTERN AT THE DESIRED POINT.
C
      FI=((THETA-THETO)/DLTHTA)+1.0
      FJ=((PHI-PHIO)/DLPHI)+.99999999
      IF(PHI .EQ. 0.)  FJ=1.
      I=FI
      J=FJ
      DFI=FI-I
      DFJ=FJ-J
      I1=2*I-1
      I2=I1+2
      I3=2*I
      I4=I3+2


      IRX = IR2D2*2 - 1
C
C DETERMINE AMP AT (THETA,PHI) BY WEIGHTED AVERAGE OF ALL 4 POINTS
C AROUND THETA, PHI.
C
      DATAX(IR2,IC)=(DFI*DATAY(I2,J)+(1.0-DFI)*DATAY(I1,J))*(1.0-DFJ)
     1+(DFI*DATAY(I2,J+1)+(1.0-DFI)*DATAY(I1,J+1))*DFJ
C
C RESET PHASES AT THREE CORNERS TO BE ON SAME CYCLE AS REFERENCE AT
C (I3,J)
C
C
C RESET PHASE AT (I4,J) IF NECESSARY SO THAT THE ABSOLUTE VALUE OF D43J
C IS LESS THAN 180.0 DEGREES.
C
      D43J=DATAY(I4,J)-DATAY(I3,J)
```

```
      IF(D43J.GT.180.0)                                                  FFKXY  78
     1DTEMP1=DATAY(I4,J)-360.0                                          FFKXY  79
      IF(D43J.LE.180.0.AND.D43J.GE.-180.0)                              FFKXY  80
     1DTEMP1=DATAY(I4,J)                                                FFKXY  81
      IF(D43J.LT.-180.0)                                                FFKXY  82
     1DTEMP1=DATAY(I4,J)+360.0                                          FFKXY  83
C                                                                       FFKXY  84
C RESET PHASE AT (I3,J+1) IF NECESSARY SO THAT THE ABSOLUTE VALUE OF    FFKXY  85
C D33J1 IS LESS THAN 180.0                                              FFKXY  86
C                                                                       FFKXY  87
      D33J1=DATAY(I3,J+1)-DATAY(I3,J)                                   FFKXY  88
      IF(D33J1.GT.180.0)                                                FFKXY  89
     1DTEMP2=DATAY(I3,J+1)-360.0                                        FFKXY  90
      IF(D33J1.LE.180.0.AND.D33J1.GE.-180.0)                            FFKXY  91
     1DTEMP2=DATAY(I3,J+1)                                              FFKXY  92
      IF(D33J1.LT.-180.0)                                               FFKXY  93
     1DTEMP2=DATAY(I3,J+1)+360.0                                        FFKXY  94
C                                                                       FFKXY  95
C RESET PHASE AT (I4,J+1) IF NECESSARY SO THAT THE ABSOLUTE VALUE OF    FFKXY  96
C D43J1 IS LESS THAN 180.0 DEGREES.                                     FFKXY  97
C                                                                       FFKXY  98
      D43J1=DATAY(I4,J+1)-DATAY(I3,J)                                   FFKXY  99
      IF(D43J1.GT.180.0)                                                FFKXY 100
     1DTEMP3=DATAY(I4,J+1)-360.0                                        FFKXY 101
      IF(D43J1.LE.180.0.AND.D43J1.GE.-180.0)                            FFKXY 102
     1DTEMP3=DATAY(I4,J+1)                                              FFKXY 103
      IF(D43J1.LT.-180.0)                                               FFKXY 104
     1DTEMP3=DATAY(I4,J+1)+360.0                                        FFKXY 105
C                                                                       FFKXY 106
C DETERMINE PHASE AT (THETA,PHI) BY WEIGHTED AVERAGE OF ALL 4 POINTS    FFKXY 107
C AROUND THETA,PHI.                                                     FFKXY 108
C                                                                       FFKXY 109
      DATAX(IR2+1,IC)=(DFI*DTEMP1+(1.0-DFI)*DATAY(I3,J))*(1.0-DFJ)      FFKXY 110
     1+(DFI*DTEMP3+(1.0-DFI)*DTEMP2)*DFJ                                FFKXY 111
      IF(FKY.LT.0.0) DATAX(IR2+1,IC)=DATAX(IR2+1,IC)-180.0              FFKXY 112
      DATAX(IR2+1,IC)=AMOD(DATAX(IR2+1,IC),360.0)                       FFKXY 113
      IF(DATAX(IR2+1,IC).LT.0.0) DATAX(IR2+1,IC)=DATAX(IR2+1,IC)+360.0  FFKXY 114
C                                                                       FFKXY 115
      GO TO 90                                                          FFKXY 116
   80 DATAX(IR2, IC) = 0.                                               FFKXY 117
      DATAX(IR2 + 1, IC) = 0.                                           FFKXY 118
   90 CONTINUE                                                          FFKXY 119
                                                                        FFKXY 120
      IF (IC2D2 - ICN .LE. 0)  GO TO 102                                FFKXY 121
      IF (IRX - IRN .LE. 0)  GO TO 101                                  FFKXY 122
      DATAX(IRX - IRN,IC) = DATAX(IR2,IC2D2 - ICN) =                    FFKXY 123
     1DATAX(IRX - IRN,IC2D2 - ICN) = DATAX(IR2, IC)                     FFKXY 124
      DATAX(IRX - IRN + 1,IC) = DATAX(IR2 + 1,IC2D2 - ICN) =            FFKXY 125
     1DATAX(IRX - IRN + 1,IC2D2 - ICN) = DATAX(IR2 + 1, IC)             FFKXY 126
      GO TO 100                                                         FFKXY 127
  101 DATAX(IR2, IC2D2 - ICN) = DATAX(IR2, IC)                          FFKXY 128
      DATAX(IR2 + 1, IC2D2 - ICN) = DATAX(IR2 + 1, IC)                  FFKXY 129
      GO TO 100                                                         FFKXY 130
  102 IF (IRX - IRN .LE. 0)  GO TO 100                                  FFKXY 131
      DATAX(IRX - IRN,IC) = DATAX(IR2,IC)                               FFKXY 132
      DATAX(IRX - IRN + 1,IC) = DATAX(IR2 + 1,IC)                       FFKXY 133
  100 CONTINUE                                                          FFKXY 134
      IRN = IRN + 2                                                     FFKXY 135
   62  CONTINUE                                                         FFKXY 136
      ICN = ICN + 1                                                     FFKXY 137
   61 CONTINUE                                                          FFKXY 138


C                                                                       FFKXY 139
C WRITE FAR-FIELD OUT ON UNIT LUOUT                                     FFKXY 140
C                                                                       FFKXY 141
      CALL FFOUT(DATAX, IR2X2, IC2TON, LUOUT)                           FFKXY 142
C                                                                       FFKXY 143
C CALCULATE NEAR-FIELD                                                  FFKXY 144
C                                                                       FFKXY 145
      CALL NFKXY(DATAX, IR2X2, IC2TON)                                  FFKXY 146
C                                                                       FFKXY 147
 2000  FORMAT (I5, (T6,10F12.4))                                        FFKXY 148
 2001 FORMAT (1H0)                                                      FFKXY 149
      RETURN                                                            FFKXY 150
      END                                                               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.

$$E_y(\underline{r}) = \frac{1}{2\pi}\sqrt{\frac{P_o G(0)}{2\pi Y_o k^2}} \int s_{10_y}(\underline{k})\, e^{i|\gamma|z}\, e^{i\underline{K}\bullet\underline{R}}\, d\underline{K} \ . \tag{45}$$

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

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

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,

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

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 2π rather than $-\pi$ to π. 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	= π = 3.14159....
PIX2	= 2π
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

```
      SUBROUTINE NFKXY(DATA, IR2X2, IC2TON)                             NFKXY
C-                                                                      NFKXY
C-    CALCULATES THE NEAR-FIELD DISTRIBUTION IN A PLANE GIVEN THE       NFKXY
C-    FAR-FIELD ANGULAR SPECTRUM.                                       NFKXY
C-                                                                      NFKXY
      DIMENSION E(128), P(128), X(128), Y(128)                          NFKXY
      DIMENSION DATA(IR2X2, IC2TON), NN(2)                              NFKXY
C                                                                       NFKXY
      COMMON /ID/ CASEID(8)                                             NFKXY
      COMMON /CNTRL/ DLTH, DLPH, DELX, DELY, FREQ, DIST, PNRM           NFKXY
      PI = 4.*ATAN(1.)     $    PIX2 = 2.*PI                            NFKXY
      RTD = 180./PI                                                     NFKXY
      NN(1) = IR2TON = IR2X2/2   $    NN(2) = IC2TON                    NFKXY
      CEE = .2997925     $    ELMDA = CEE/FREQ                          NFKXY
      ISIGN = 1                                                         NFKXY
C-                                                                      NFKXY
C-    APPLY E TO I*GAMMA*D PHASE CORRECTION COLUMN BY COLUMN            NFKXY
C-                                                                      NFKXY
      DO 50 IC = 1, IC2TON                                              NFKXY
        CALL FTOIGAM(DATA(1,IC), IR2TON, IC2TON, IC, +1, ELMDA, DELX,   NFKXY
     1  DELY, DIST)                                                     NFKXY
   50   CONTINUE                                                        NFKXY
C-                                                                      NFKXY
C-    CHANGE DATA ARRAY FROM POLAR TO RECTANGULAR FORM.                 NFKXY
C-                                                                      NFKXY
      CALL ARAYPTR(DATA, IR2X2, IC2TON)                                 NFKXY
C-                                                                      NFKXY
C-    PERFORM FOURIER TRANSFORM OF DATA ARRAY TO PRODUCE NEAR-FIELD.    NFKXY
C-                                                                      NFKXY
      CALL FOURT(DATA, NN, 2, ISIGN, +1, 0)                             NFKXY
C-                                                                      NFKXY
C-    CHANGE NEAR-FIELD DATA FROM RECTANGULAR TO POLAR FORM.            NFKXY
C-                                                                      NFKXY
      CALL ARAYRTP(DATA, IR2X2, IC2TON)                                 NFKXY
      FACTOR = PIX2/(FLOAT(IR2TON)*FLOAT(IC2TON)*DELX*DELY)             NFKXY
      DO 55 ICOL = 1, IC2TON                                            NFKXY
        DO 55 IROW = 1, IR2X2, 2                                        NFKXY
          DATA(IROW, ICOL) = DATA(IROW, ICOL)*FACTOR                    NFKXY
   55   CONTINUE                                                        NFKXY
      CALL SWAP(IR2X2,IC2TON,DATA)                                      NFKXY
      CALL PHSCOR2(DATA,IR2X2,IC2TON)                                   NFKXY
C                                                                       NFKXY
      DO 100 IX = 1, IR2TON                                             NFKXY
  100 X(IX) = DELX*(IX - IR2TON/2)                                      NFKXY
C                                                                       NFKXY
      DO 110 IY = 1, IC2TON                                             NFKXY
  110 Y(IY) = DELY*(IY - IC2TON/2)                                      NFKXY
C                                                                       NFKXY
      PRINT 2001                                                        NFKXY
      PRINT 2002, (X((IK+1)/2),DATA(IK, IC2TON/2), DATA(IK+1, IC2TON/2),NFKXY
     1 Y((IK+1)/2), DATA(IR2TON-1, (IK+1)/2), DATA(IR2TON, (IK+1)/2),   NFKXY
     2 IK = 1, IR2X2, 2)                                                NFKXY
      XMIN = X(1)    $    XMAX = X(IR2TON)                              NFKXY
      DO 60 IK = 1, IR2X2, 2                                            NFKXY
        I = (IK + 1)/2                                                  NFKXY
        E(I) = DATA (IK, IC2TON/2)                                      NFKXY
        P(I) = DATA (IK + 1, IC2TON/2)                                  NFKXY
   60   CONTINUE                                                        NFKXY
C-                                                                      NFKXY
C-    PLOT E-PLANE AMPLITUDE AND PHASE.                                 NFKXY
C-                                                                      NFKXY
      CALL PLT120R(X, E, XMAX, XMIN, 10., 0., IC2TON, 1H*, 1, 1)        NFKXY
      PRINT 2003, CASEID, 10HY-Z PLANE    , 10HAMPLITUDE                NFKXY
      CALL PLT120R(X, P, XMAX, XMIN, 360., C., IC2TON, 1H*, 1, 1)       NFKXY
      PRINT 2003, CASEID, 10HY-Z PLANE    , 10HPHASE                    NFKXY
      YMIN = Y(1)       $    YMAX = Y(IC2TON)                           NFKXY
      DO 61 I = 1, IC2TON                                               NFKXY
        E(I) = DATA(IR2TON - 1, I)                                      NFKXY
        P(I) = DATA(IR2TON, I)                                          NFKXY
   61   CONTINUE                                                        NFKXY
C-                                                                      NFKXY
C-    PLOT H-PLANE AMPLITUDE AND PHASE.                                 NFKXY
C-                                                                      NFKXY
      CALL PLT120R(Y, E, YMAX, YMIN, 10., 0., IC2TON, 1H*, 1, 1)        NFKXY
      PRINT 2003, CASEID, 10HX-Z PLANE    , 10HAMPLITUDE                NFKXY
      CALL PLT120R(Y, P, YMAX, YMIN, 360., 0., IC2TON, 1H*, 1, 1)       NFKXY
      PRINT 2003, CASEID, 10HX-Z PLANE    , 10HPHASE                    NFKXY
```

```
      RETURN                                                            NFKXY
 2001 FORMAT(///,T64,*CENTERLINE DATA*,//,T37,*X-Z PLANE*, T97, *Y-Z PLANFKXY
     1NE*,/,T22,*X*,T40,*AMP*,T59,*PHASE*,T82,*Y*,T100,*AMP*,T119,*PHASENFKXY
     2*)                                                                NFKXY
 2002 FORMAT(T6, 6F20.4)                                                NFKXY
 2003 FORMAT(/,5X, 8A10, 5X, 2A10)                                      NFKXY
      END                                                               NFKXY
```

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 = ± 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

$$\mathrm{DATA} = \mathrm{DATA} \quad e^{i\gamma d}$$

$$\gamma = \sqrt{k^2 - k_x^2 - k_y^2}$$

$$k = \omega\sqrt{\mu\varepsilon}$$

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

$$k_y = k \sin\theta \sin\emptyset .$$

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

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

for γ real, and

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

for γ imaginary

γ is computed from the row and column positions of the data elements.

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

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

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 γ
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	= ± 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	= π = 3.14159....
PIX2	= 2π
RTOD	= $180/\pi$ = radian to degree conversion
SUMSQ	= $k_x^2 + k_y^2$
TEMP	= Intermediate variable

```
      SUBROUTINE ETOIGAM (DATA, NROW, NCOL, ICOL, ISGN, FLMDA, DELX, DEL
     1Y, DIST)
C
      DIMENSION DATA (1)
C
      PI = 3.1415926536
      PIX2 = 2. * PI
      FKAY = PIX2 / FLMDA
      FKAYSQ = FKAY * * 2
      RTOD = 180. / PI
      DTOR = 1. / RTOD
      FKX = PIX2 * (ICOL - (NCOL / 2)) / DELX / NCOL
      FKXSQ = FKX * * 2
C
      IF (NROW .LT. 1) GO TO 130
      DO 120 IROW = 1, NROW
      FKY = PIX2 * (IROW - (NROW / 2)) / DELY / NROW
      FKYSQ = FKY * * 2
      SUMSQ = FKXSQ + FKYSQ
      PHACORR = 0.0
      FACTOR = 1.0
      IF (SUMSQ .GT. FKAYSQ) GO TO 100
C
      PHACORR = ISGN * SQRT (FKAYSQ - SUMSQ) * DIST * RTOD
      GO TO 110
C
  100 FACTOR = (SQRT (SUMSQ - FKAYSQ)) * DIST
      IF (FACTOR .GT. 100.)FACTOR = 100.
      FACTOR = EXP (ISGN * FACTOR)
C
  110 CONTINUE
C
      DATA (2 * IROW - 1) = DATA (2 * IROW - 1) * FACTOR
      TPHASE = DATA (2 * IROW)
      TEMP = DATA (2 * IROW) + PHACORR
      TEMP = TEMP - INT (TEMP / 360.) * 360.
      IF (TEMP .LT. 0.0)TEMP = TEMP + 360.0
      DATA (2 * IROW) = TEMP
C
  120 CONTINUE
  130 CONTINUE
 1500 FORMAT (1X, 4I10, 5F12.3,//)
 1510 FORMAT (1X, 2I5, 8F12.3)
      RETURN
      END
```

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

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

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° is added to each element in order to allow the near-field phase to be conveniently plotted in the range 0°-360°.

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
IO2	= 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 (IO2,ICOL)

```
      SUBROUTINE PHSCOR2(DATA, NRX2, NCOL)
C
      DIMENSION DATA(NRX2, NCOL)
C
      NROW = NRX2 / 2
C
C
      CONST1 = -180.*FLOAT(NCOL - 2)/FLOAT(NCOL)
      CONST2 = -180.*FLOAT(NROW - 2)/FLOAT(NROW)
      IF (NCOL .LT. 1) GO TO 130
      DO 100 ICOL = 1, NCOL
      C1 = CONST1 * (ICOL - 1)
      IF (NRX2 .LT. 2) GO TO 110
      DO 100 IROW = 2, NRX2, 2
      IO2 = IROW / 2
      C2 = CONST2 * (IO2 - 1) + C1
      TEMP = DATA(IROW, ICOL) + C2 + 90.
      TEMP = TEMP - INT(TEMP / 360.) * 360.
      IF (TEMP .LT. 0.) TEMP = TEMP + 360.
      DATA(IROW, ICOL) = TEMP
  100 CONTINUE
  110 CONTINUE
  130 CONTINUE
C
      RETURN
      END
```

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 0-2π and transforms them into a domain of 0-2π. 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 π to 2π. 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

```
      SUBROUTINE SWAP(NRX2, NCOL, DATA)
      DIMENSION DATA(NRX2, NCOL)

C
      NROW = NRX2 / 2
      NCO2 = NCOL / 2
C
C-MOVING EDGES OF ARRAY TO CENTER AND VICE VERSA BY COLUMNS
C
      IF (NRX2 .LT. 1) GO TO 220
      DO 200 IROW = 1, NRX2
      IF (NCO2 .LT. 1) GO TO 210
      DO 200 ICOL = 1, NCO2
      ICPNC = ICOL + NCO2
      TEMP = DATA(IROW, ICPNC)
      DATA(IROW, ICPNC) = DATA(IROW, ICOL)
  200 DATA(IROW, ICOL) = TEMP
  210 CONTINUE
  220 CONTINUE
C
      NCM1 = NCOL - 1
      IF (NRX2 .LT. 1) GO TO 310
      DO 300 IROW = 1, NRX2
      TEMP1 = DATA(IROW, 1)
      IF (NCM1 .LT. 1) GO TO 280
      DO 230 ICOL = 1, NCM1
  230 DATA(IROW, ICOL) = DATA(IROW, ICOL + 1)
  280 CONTINUE
  300 DATA(IROW, NCOL) = TEMP1
  310 CONTINUE
C
C-MOVING EDGES OF ARRAY TO CENTER AND VICE VERSA BY ROWS
C
      IF (NCOL .LT. 1) GO TO 340
      DO 320 ICOL = 1, NCOL
      IF (NROW .LT. 1) GO TO 330
      DO 320 IROW = 1, NROW
      IRPNR = IROW + NROW
      TEMP = DATA(IRPNR, ICOL)
      DATA(IRPNR, ICOL) = DATA(IROW, ICOL)
  320 DATA(IROW, ICOL) = TEMP
  330 CONTINUE
  340 CONTINUE
C
      NR2M2 = NRX2 - 2
      IF (NCOL .LT. 1) GO TO 380
      DO 370 ICOL = 1, NCOL
      TEMP1 = DATA(1, ICOL)
      TEMP2 = DATA(2, ICOL)
      IF (NR2M2 .LT. 1) GO TO 360
      DO 350 IROW = 1, NR2M2
  350 DATA(IROW, ICOL) = DATA(IROW + 2, ICOL)
  360 CONTINUE
      DATA(NR2M2 + 1, ICOL) = TEMP1
  370 DATA(NR2M2 + 2, ICOL) = TEMP2
  380 CONTINUE
C
      RETURN
      END
```

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 arithemtic. 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.
   $\mathrm{Re(DATA)} = |\mathrm{DATA}| \cos(\mathrm{ANGLE(DATA)})$.
   $\mathrm{Im(DATA)} = |\mathrm{DATA}| \sin(\mathrm{ANGLE(DATA)})$.

2. ARAYRTP

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

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

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

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 • NROW
PI	= π = 3.14159....
RTOD	= $180/\pi$ = radian to degree conversion factor
TAMP	= Intermediate variable, amplitude of complex number

```
      SUBROUTINE ARAYPTR(DATA, NRX2, NCOL)
C
      DIMENSION DATA(NRX2, NCOL)
C
      PI = 3.1415926536
      RTOD = 180. / PI
      DTOR = 1. / RTOD
C
      IF (NCOL .LT. 1) GO TO 130
      DO 120 ICOL = 1, NCOL
      IF (NRX2 .LT. 1) GO TO 110
      DO 100 IROW = 1, NRX2, 2
      IRP1 = IROW + 1
      TAMP = DATA(IROW, ICOL)
      ANGLE = DATA(IRP1, ICOL) * DTOR
      DATA(IROW, ICOL) = TAMP * COS(ANGLE)
      DATA(IRP1, ICOL) = TAMP * SIN(ANGLE)
  100 CONTINUE
  110 CONTINUE
  120 CONTINUE
  130 CONTINUE
      RETURN
C
      ENTRY ARAYRTR
C
      PI = 3.1415926536
      RTOD = 180. / PI
      DTOR = 1. / RTOD
C
      IF (NCOL .LT. 1) GO TO 190
      DO 180 ICOL = 1, NCOL
      IF (NRX2 .LT. 1) GO TO 170
      DO 160 IROW = 1, NRX2, 2
      IRP1 = IROW + 1
      FREAL = DATA(IROW, ICOL)
      FIMAG = DATA(IRP1, ICOL)
      DATA(IROW, ICOL) = SQRT(FREAL * FREAL + FIMAG * FIMAG)
      IF (FREAL .EQ. 0.0 .AND. FIMAG .EQ. 0.0) GO TO 160
      DATA(IRP1, ICOL) = ATAN2(FIMAG, FREAL) * RTOD
  160 CONTINUE
  170 CONTINUE
  180 CONTINUE
  190 CONTINUE
C
      RETURN
      END
```

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 O
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

```
      SUBROUTINE FFOUT(DATA, NRX2, NCOL, LUOUT)
C
C  THIS SUBROUTINE WRITES ARRAY DATA TO FILE LUOUT
C
      DIMENSION DATA(NRX2, NCOL), ID(10)
      COMMON / ID/ CASEID(8)
      INTEGER CASEID
C
      ID(1) = 10HPHYSICAL O
      ID(2) = 10HPTICS SIML
      DO 10 I = 3, 7
   10   ID(I) = CASEID(I - 2)
      CALL DATE(DT)
      CALL TIME(TM)
      DECODE(10, 1500, DT) YR, MON, DY
      DECODE(10, 1500, TM) HR, MN, SC
      ENCODE(10, 1510, ID(8))  MON, DY, YR, HR, MN
      ID(9) = NCOL
      ID(10) = NRX2
      WRITE (LUOUT) (ID(I), I = 1,10)
      PRINT 1520, ID
      DO 20 IC = 1, NCOL
        WRITE (LUOUT) (DATA(IR, IC), IR = 1, NRX2)
   20   CONTINUE
      ENDFILE LUOUT
      RETURN
 1500 FORMAT (1X, 3(A2, 1X))
 1510 FORMAT (5A2)
 1520 FORMAT (*0OUTPUT FILE ID -- *, 10X, 8A10, 2I10)
      END
```

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

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

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 MAXO.

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

```
      SUBROUTINE FOURT (DATA, NN, NDIM, ISIGN, IFORM, WORK)              FOURT
C                                                                       FOURT
C     THE COOLEY-TUKEY FAST FOURIER TRANSFORM IN USASI BASIC FORTRAN    FOURT
C                                                                       FOURT
C     TRANSFORM(K1,K2,...) = SUM(DATA(J1,J2,...)*EXP(ISIGN*2*PI*SQRT(-1)FOURT
C     *((J1-1)*(K1-1)/NN(1)+(J2-1)*(K2-1)/NN(2)+...))), SUMMED FOR ALL  FOURT
C     J1, K1 BETWEEN 1 AND NN(1), J2, K2 BETWEEN 1 AND NN(2), ETC.      FOURT
C     THERE IS NO LIMIT TO THE NUMBER OF SUBSCRIPTS.  DATA IS A         FOURT
C     MULTIDIMENSIONAL COMPLEX ARRAY WHOSE REAL AND IMAGINARY           FOURT
C     PARTS ARE ADJACENT IN STORAGE, SUCH AS FORTRAN IV PLACES THEM.    FOURT
C     IF ALL IMAGINARY PARTS ARE ZERO (DATA ARE DISGUISED REAL), SET    FOURT
C     IFORM TO ZERO TO CUT THE RUNNING TIME BY UP TO FORTY PERCENT.     FOURT
C     OTHERWISE, IFORM = +1.  THE LENGTHS OF ALL DIMENSIONS ARE         FOURT
C     STORED IN ARRAY NN, OF LENGTH NDIM.  THEY MAY BE ANY POSITIVE     FOURT
C     INTEGERS, THO THE PROGRAM RUNS FASTER ON COMPOSITE INTEGERS, AND  FOURT
C     ESPECIALLY FAST ON NUMBERS RICH IN FACTORS OF TWO.  ISIGN IS +1   FOURT
C     OR -1.  IF A -1 TRANSFORM IS FOLLOWED BY A +1 ONE (OR A +1        FOURT
C     BY A -1) THE ORIGINAL DATA REAPPEAR, MULTIPLIED BY NTOT (=NN(1)*  FOURT
C     NN(2)*...).  TRANSFORM VALUES ARE ALWAYS COMPLEX, AND ARE RETURNEDFOURT
C     IN ARRAY DATA, REPLACING THE INPUT.  IN ADDITION, IF ALL          FOURT
C     DIMENSIONS ARE NOT POWERS OF TWO, ARRAY WORK MUST BE SUPPLIED,    FOURT
C     COMPLEX OF LENGTH EQUAL TO THE LARGEST NON 2**K DIMENSION.        FOURT
C     OTHERWISE, REPLACE WORK BY ZERO IN THE CALLING SEQUENCE.          FOURT
C     NORMAL FORTRAN DATA ORDERING IS EXPECTED, FIRST SUBSCRIPT VARYING FOURT
C     FASTEST.  ALL SUBSCRIPTS BEGIN AT ONE.                            FOURT
C                                                                       FOURT
C     RUNNING TIME IS MUCH SHORTER THAN THE NAIVE NTOT**2, BEING        FOURT
C     GIVEN BY THE FOLLOWING FORMULA.  DECOMPOSE NTOT INTO              FOURT
C     2**K2 * 3**K3 * 5**K5 * ....  LET SUM2 = 2*K2, SUMF = 3*K3 + 5*K5 FOURT
C     + ... AND NF = K3 + K5 + ....  THE TIME TAKEN BY A MULTI-         FOURT
C     DIMENSIONAL TRANSFORM ON THESE NTOT DATA IS T = T0 + NTOT*(T1+    FOURT
C     T2*SUM2+T3*SUMF+T4*NF).  ON THE CDC 3300 (FLOATING POINT ADD TIME FOURT
C     OF SIX MICROSECONDS), T = 3000 + NTOT*(500+43*SUM2+68*SUMF+       FOURT
C     320*NF) MICROSECONDS ON COMPLEX DATA.  IN ADDITION, THE           FOURT
C     ACCURACY IS GREATLY IMPROVED, AS THE RMS RELATIVE ERROR IS        FOURT
C     BOUNDED BY 3*2**(-B)*SUM(FACTOR(J)**1.5), WHERE B IS THE NUMBER   FOURT
C     OF BITS IN THE FLOATING POINT FRACTION AND FACTOR(J) ARE THE      FOURT
C     PRIME FACTORS OF NTOT.                                            FOURT
C                                                                       FOURT
C     PROGRAM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES       FOURT
C     RADER.  RALPH ALTER SUGGESTED THE IDEA FOR THE DIGIT REVERSAL.    FOURT
C     MIT LINCOLN LABORATORY, AUGUST 1967.  THIS IS THE FASTEST AND MOSTFOURT
C     VERSATILE VERSION OF THE FFT KNOWN TO THE AUTHOR.  SHORTER PRO-   FOURT
C     GRAMS FOUR1 AND FOUR2 RESTRICT DIMENSION LENGTHS TO POWERS OF TWO.FOURT
C     SEE-- IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON FFT.  FOURT
C                                                                       FOURT
C     THE DISCRETE FOURIER TRANSFORM PLACES THREE RESTRICTIONS UPON THE FOURT
C     DATA.                                                             FOURT
C     1.  THE NUMBER OF INPUT DATA AND THE NUMBER OF TRANSFORM VALUES   FOURT
C     MUST BE THE SAME.                                                 FOURT
C     2.  BOTH THE INPUT DATA AND THE TRANSFORM VALUES MUST REPRESENT   FOURT
C     EQUISPACED POINTS IN THEIR RESPECTIVE DOMAINS OF TIME AND         FOURT
C     FREQUENCY.  CALLING THESE SPACINGS DELTAT AND DELTAF, IT MUST BE  FOURT
C     TRUE THAT DELTAF=2*PI/(NN(I)*DELTAT).  OF COURSE, DELTAT NEED NOT FOURT
C     BE THE SAME FOR EVERY DIMENSION.                                  FOURT
C     3.  CONCEPTUALLY AT LEAST, THE INPUT DATA AND THE TRANSFORM OUTPUTFOURT
C     REPRESENT SINGLE CYCLES OF PERIODIC FUNCTIONS.                    FOURT
C                                                                       FOURT
C     EXAMPLE 1.  THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A      FOURT
C     COMPLEX ARRAY DIMENSIONED 32 BY 25 BY 13 IN FORTRAN IV.           FOURT
C     DIMENSION DATA(32,25,13),WORK(50),NN(3)                           FOURT


C     COMPLEX DATA                                                      FOURT
C     DATA NN/32,25,13/                                                 FOURT
C     DO 1 I=1,32                                                       FOURT
C     DO 1 J=1,25                                                       FOURT
C     DO 1 K=1,13                                                       FOURT
C   1 DATA(I,J,K)=COMPLEX VALUE                                         FOURT
C     CALL FOURT(DATA,NN,3,-1,1,WORK)                                   FOURT
C                                                                       FOURT
C     EXAMPLE 2.  ONE-DIMENSIONAL FORWARD TRANSFORM OF A REAL ARRAY OF  FOURT
C     LENGTH 64 IN FORTRAN II.                                          FOURT
C     DIMENSION DATA(2,64)                                              FOURT
C     DO 2 I=1,64                                                       FOURT
C     DATA(1,I)=REAL PART                                               FOURT
C   2 DATA(2,I)=0.                                                      FOURT
C     CALL FOURT(DATA,64,1,-1,0,0)                                      FOURT
C                                                                       FOURT
```

```
      DIMENSION DATA (1), NN (1), IFACT (32), WORK (1)                 FOURT   78
      WR = 0.                                                          FOURT   79
      WI = 0.                                                          FOURT   80
      WSTPR = 0.                                                       FOURT   81
      WSTPI = 0.                                                       FOURT   82
      TWOPI = 6.283185307                                              FOURT   83
      IF (NDIM - 1)1280, 100, 100                                      FOURT   84
  100 NTOT = 2                                                         FOURT   85
      DO 110 IDIM = 1, NDIM                                            FOURT   86
      IF (NN (IDIM))1280, 1280, 110                                    FOURT   87
  110 NTOT = NTOT * NN (IDIM)                                          FOURT   88
C                                                                      FOURT   89
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
      NP2 = NP1 * N                                                    FOURT   95
      IF (N - 1)1280, 1260, 120                                        FOURT   96
C                                                                      FOURT   97
C     FACTOR N                                                         FOURT   98
C                                                                      FOURT   99
  120 M = N                                                            FOURT  100
      NTWO = NP1                                                       FOURT  101
      IF = 1                                                           FOURT  102
      IDIV = 2                                                         FOURT  103
  130 IQUOT = M / IDIV                                                 FOURT  104
      IREM = M - IDIV * IQUOT                                          FOURT  105
      IF (IQUOT - IDIV)210, 140, 140                                   FOURT  106
  140 IF (IREM)160, 150, 160                                           FOURT  107
  150 NTWO = NTWO + NTWO                                               FOURT  108
      M = IQUOT                                                        FOURT  109
      GO TO 130                                                        FOURT  110
  160 IDIV = 3                                                         FOURT  111
  170 IQUOT = M / IDIV                                                 FOURT  112
      IREM = M - IDIV * IQUOT                                          FOURT  113
      IF (IQUOT - IDIV)230, 180, 180                                   FOURT  114
  180 IF (IREM)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
  200 IDIV = IDIV + 2                                                  FOURT  120
      GO TO 170                                                        FOURT  121
  210 IF (IREM)230, 220, 230                                           FOURT  122
  220 NTWO = NTWO + NTWO                                               FOURT  123
      GO TO 240                                                        FOURT  124
  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
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
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           ARE THE ODD NUMBERED REAL VALUES.  SEPARATE AND SUPPLY     FOURT  139
C           THE SECOND HALF BY CONJUGATE SYMMETRY.                     FOURT  140
C                                                                      FOURT  141
  240 NON2 = NP1 * (NP2 / NTWO)                                        FOURT  142
      ICASE = 1                                                        FOURT  143
      IF (IDIM - 4)250, 300, 300                                       FOURT  144
  250 IF (IFORM)260, 260, 300                                          FOURT  145
  260 ICASE = 2                                                        FOURT  146
      IF (IDIM - 1)270, 270, 300                                       FOURT  147
  270 ICASE = 3                                                        FOURT  148
      IF (NTWO - NP1)300, 300, 280                                     FOURT  149
  280 ICASE = 4                                                        FOURT  150
      NTWO = NTWO / 2                                                  FOURT  151
      N = N / 2                                                        FOURT  152
      NP2 = NP2 / 2                                                    FOURT  153
      NTOT = NTOT / 2                                                  FOURT  154
```

```
      I = 3
      DO 290 J = 2, NTOT
      DATA (J) = DATA (I)
  290 I = I + 2
  300 I1RNG = NP1
      IF (ICASE - 2)320, 310, 320
  310 I1RNG = NP0 * (1 + NPREV / 2)
C
C     SHUFFLE ON THE FACTORS OF TWO IN N.  AS THE SHUFFLING
C     CAN BE DONE BY SIMPLE INTERCHANGE, NO WORKING ARRAY IS NEEDED
C
  320 IF (NTWO - NP1)700, 700, 330
  330 NP2HF = NP2 / 2
      J = 1
      DO 390 I2 = 1, NP2, NON2
      IF (J - I2)340, 360, 360
  340 I1MAX = I2 + NON2 - 2
      DO 350 I1 = I2, I1MAX, 2
      DO 350 I3 = I1, NTOT, NP2
      J3 = J + I3 - I2
      TEMPR = DATA (I3)
      TEMPI = DATA (I3 + 1)
      DATA (I3) = DATA (J3)
      DATA (I3 + 1) = DATA (J3 + 1)
      DATA (J3) = TEMPR
  350 DATA (J3 + 1) = TEMPI
  360 M = NP2HF
  370 IF (J - M)390, 390, 380
  380 J = J - M
      M = M / 2
      IF (M - NON2)390, 370, 370
  390 J = J + M
C
C     MAIN LOOP FOR FACTORS OF TWO.  PERFORM FOURIER TRANSFORMS OF
C     LENGTH FOUR, WITH ONE OF LENGTH TWO IF NEEDED.  THE TWIDDLE FACTOR
C     W=EXP(ISIGN*2*PI*SQRT(-1)*M/(4*MMAX)).  CHECK FOR W=ISIGN*SQRT(-1)
C     AND REPEAT FOR W=ISIGN*SQRT(-1)*CONJUGATE(W).
C
      NON2T = NON2 + NON2
      IPAR = NTWO / NP1
  400 IF (IPAR - 2)440, 420, 410
  410 IPAR = IPAR / 4
      GO TO 400
  420 DO 430 I1 = 1, I1RNG, 2
      DO 430 J3 = I1, NON2, NP1
      DO 430 K1 = J3, NTOT, NON2T
      K2 = K1 + NON2
      TEMPR = DATA (K2)
      TEMPI = DATA (K2 + 1)
      DATA (K2) = DATA (K1) - TEMPR
      DATA (K2 + 1) = DATA (K1 + 1) - TEMPI
      DATA (K1) = DATA (K1) + TEMPR
  430 DATA (K1 + 1) = DATA (K1 + 1) + TEMPI
  440 MMAX = NON2
  450 IF (MMAX - NP2HF)460, 700, 700
  460 LMAX = MAX0 (NON2T, MMAX / 2)
      IF (MMAX - NON2)500, 500, 470
  470 THETA = - TWOPI * FLOAT (NON2) / FLOAT (4 * MMAX)
      IF (ISIGN)490, 480, 480
  480 THETA = - THETA
  490 WR = COS (THETA)
      WI = SIN (THETA)
      WSTPR = - 2. * WI * WI
      WSTPI = 2. * WR * WI
  500 DO 690 L = NON2, LMAX, NON2T
      M = L
      IF (MMAX - NON2)520, 520, 510
  510 W2R = WR * WR - WI * WI
      W2I = 2. * WR * WI
      W3R = W2R * WR - W2I * WI
      W3I = W2R * WI + W2I * WR
  520 DO 640 I1 = 1, I1RNG, 2
      DO 540 J3 = I1, NON2, NP1
      KMIN = J3 + IPAR * M
      IF (MMAX - NON2)530, 530, 540
  530 KMIN = J3
  540 KDIF = IPAR * MMAX
```

```
  550 KSTEP = 4 * KDIF
      DO 630 K1 = KMIN, NTOT, KSTEP
      K2 = K1 + KDIF
      K3 = K2 + KDIF
      K4 = K3 + KDIF
      IF (MMAX - NON2)560, 560, 590
  560 U1R = DATA (K1) + DATA (K2)
      U1I = DATA (K1 + 1) + DATA (K2 + 1)
      U2R = DATA (K3) + DATA (K4)
      U2I = DATA (K3 + 1) + DATA (K4 + 1)
      U3R = DATA (K1) - DATA (K2)
      U3I = DATA (K1 + 1) - DATA (K2 + 1)
      IF (ISIGN)570, 580, 580
  570 U4R = DATA (K3 + 1) - DATA (K4 + 1)
      U4I = DATA (K4) - DATA (K3)
      GO TO 620
  580 U4R = DATA (K4 + 1) - DATA (K3 + 1)
      U4I = DATA (K3) - DATA (K4)
      GO TO 620
  590 T2R = W2R * DATA (K2) - W2I * DATA (K2 + 1)
      T2I = W2R * DATA (K2 + 1) + W2I * DATA (K2)
      T3R = WR * DATA (K3) - WI * DATA (K3 + 1)
      T3I = WR * DATA (K3 + 1) + WI * DATA (K3)
      T4R = W3R * DATA (K4) - W3I * DATA (K4 + 1)
      T4I = W3R * DATA (K4 + 1) + W3I * DATA (K4)
      U1R = DATA (K1) + T2R
      U1I = DATA (K1 + 1) + T2I
      U2R = T3R + T4R
      U2I = T3I + T4I
      U3R = DATA (K1) - T2R
      U3I = DATA (K1 + 1) - T2I
      IF (ISIGN)600, 610, 610
  600 U4R = T3I - T4I
      U4I = T4R - T3R
      GO TO 620
  610 U4R = T4I - T3I
      U4I = T3R - T4R
  620 DATA (K1) = U1R + U2R
      DATA (K1 + 1) = U1I + U2I
      DATA (K2) = U3R + U4R
      DATA (K2 + 1) = U3I + U4I
      DATA (K3) = U1R - U2R
      DATA (K3 + 1) = U1I - U2I
      DATA (K4) = U3R - U4R
  630 DATA (K4 + 1) = U3I - U4I
      KMIN = 4 * (KMIN - J3) + J3
      KDIF = KSTEP
      IF (KDIF - NP2)550, 640, 640
  640 CONTINUE
      M = MMAX - M
      IF (ISIGN)650, 660, 660
  650 TEMPR = WR
      WR = - WI
      WI = - TEMPR
      GO TO 670
  660 TEMPR = WR
      WR = WI
      WI = TEMPR
  670 IF (M - LMAX)680, 680, 510
  680 TEMPR = WR
      WR = WR * WSTPR - WI * WSTPI + WR


  690 WI = WI * WSTPR + TEMPR * WSTPI + WI
      IPAR = 3 - IPAR
      MMAX = MMAX + MMAX
      GO TO 450
C
C     MAIN LOOP FOR FACTORS NOT EQUAL TO TWO.  APPLY THE TWIDDLE FACTOR
C     W=EXP(ISIGN*2*PI*SQRT(-1)*(J2-1)*(J1-J2)/(NP2*IFP1)), THEN
C     PERFORM A FOURIER TRANSFORM OF LENGTH IFACT(IF), MAKING USE OF
C     CONJUGATE SYMMETRIES.
C
  700 IF (NTWO - NP2)710, 990, 990
  710 IFP1 = NON2
      IF = 1
      NP1HF = NP1 / 2
  720 IFP2 = IFP1 / IFACT (IF)
      J1RNG = NP2
```

```
      IF (ICASE - 3)740, 730, 740                                       FOURT    309
  730 J1RNG = (NP2 + IFP1) / 2                                          FOURT    310
      J2STP = NP2 / IFACT (IF)                                          FOURT    311
      J1RG2 = (J2STP + IFP2) / 2                                        FOURT    312
  740 J2MIN = 1 + IFP2                                                  FOURT    313
      IF (IFP1 - NP2)750, 800, 800                                      FOURT    314
  750 DO 790 J2 = J2MIN, IFP1, IFP2                                     FOURT    315
      THETA = - TWOPI * FLOAT (J2 - 1) / FLOAT (NP2)                    FOURT    316
      IF (ISIGN)770, 760, 760                                           FOURT    317
  760 THETA = - THETA                                                   FOURT    318
  770 SINTH = SIN (THETA / 2.)                                          FOURT    319
      WSTPR = - 2. * SINTH * SINTH                                      FOURT    320
      WSTPI = SIN (THETA)                                               FOURT    321
      WR = WSTPR + 1.                                                   FOURT    322
      WI = WSTPI                                                        FOURT    323
      J1MIN = J2 + IFP1                                                 FOURT    324
      DO 790 J1 = J1MIN, J1RNG, IFP1                                    FOURT    325
      I1MAX = J1 + I1RNG - 2                                            FOURT    326
      DO 780 I1 = J1, I1MAX, 2                                          FOURT    327
      DO 780 I3 = I1, NTOT, NP2                                         FOURT    328
      J3MAX = I3 + IFP2 - NP1                                           FOURT    329
      DO 780 J3 = I3, J3MAX, NP1                                        FOURT    330
      TEMPR = DATA (J3)                                                 FOURT    331
      DATA (J3) = DATA (J3) * WR - DATA (J3 + 1) * WI                   FOURT    332
  780 DATA (J3 + 1) = TEMPR * WI + DATA (J3 + 1) * WR                   FOURT    333
      TEMPR = WR                                                        FOURT    334
      WR = WR * WSTPR - WI * WSTPI + WR                                 FOURT    335
  790 WI = TEMPR * WSTPI + WI * WSTPR + WI                              FOURT    336
  800 THETA = - TWOPI / FLOAT (IFACT (IF))                              FOURT    337
      IF (ISIGN)820, 810, 810                                           FOURT    338
  810 THETA = - THETA                                                   FOURT    339
  820 SINTH = SIN (THETA / 2.)                                          FOURT    340
      WSTPR = - 2. * SINTH * SINTH                                      FOURT    341
      WSTPI = SIN (THETA)                                               FOURT    342
      KSTEP = 2 * N / IFACT (IF)                                        FOURT    343
      KRANG = KSTEP * (IFACT (IF) / 2) + 1                              FOURT    344
      DO 980 I1 = 1, I1RNG, 2                                           FOURT    345
      DO 980 I3 = I1, NTOT, NP2                                         FOURT    346
      DO 910 KMIN = 1, KRANG, KSTEP                                     FOURT    347
      J1MAX = I3 + J1RNG - IFP1                                         FOURT    348
      DO 880 J1 = I3, J1MAX, IFP1                                       FOURT    349
      J3MAX = J1 + IFP2 - NP1                                           FOURT    350
      DO 880 J3 = J1, J3MAX, NP1                                        FOURT    351
      J2MAX = J3 + IFP1 - IFP2                                          FOURT    352
      K = KMIN + (J3 - J1 + (J1 - I3) / IFACT (IF)) / NP1HF             FOURT    353
      IF (KMIN - 1)830, 830, 850                                        FOURT    354
  830 SUMR = 0.                                                         FOURT    355
      SUMI = 0.                                                         FOURT    356
      DO 840 J2 = J3, J2MAX, IFP2                                       FOURT    357
      SUMR = SUMR + DATA (J2)                                           FOURT    358
  840 SUMI = SUMI + DATA (J2 + 1)                                       FOURT    359
      WORK (K) = SUMR                                                   FOURT    360
      WORK (K + 1) = SUMI                                               FOURT    361
      GO TO 880                                                         FOURT    362
  850 KCONJ = K + 2 * (N - KMIN + 1)                                    FOURT    363
      J2 = J2MAX                                                        FOURT    364
      SUMR = DATA (J2)                                                  FOURT    365
      SUMI = DATA (J2 + 1)                                              FOURT    366
      OLDSR = 0.                                                        FOURT    367
      OLDSI = 0.                                                        FOURT    368
      J2 = J2 - IFP2                                                    FOURT    369

  860 TEMPR = SUMR                                                      FOURT    370
      TEMPI = SUMI                                                      FOURT    371
      SUMR = TWOWR * SUMR - OLDSR + DATA (J2)                           FOURT    372
      SUMI = TWOWR * SUMI - OLDSI + DATA (J2 + 1)                       FOURT    373
      OLDSR = TEMPR                                                     FOURT    374
      OLDSI = TEMPI                                                     FOURT    375
      J2 = J2 - IFP2                                                    FOURT    376
      IF (J2 - J3)870, 870, 860                                         FOURT    377
  870 TEMPR = WR * SUMR - OLDSR + DATA (J2)                             FOURT    378
      TEMPI = WI * SUMI                                                 FOURT    379
      WORK (K) = TEMPR - TEMPI                                          FOURT    380
      WORK (KCONJ) = TEMPR + TEMPI                                      FOURT    381
      TEMPR = WR * SUMI - OLDSI + DATA (J2 + 1)                         FOURT    382
      TEMPI = WI * SUMR                                                 FOURT    383
      WORK (K + 1) = TEMPR + TEMPI                                      FOURT    384
      WORK (KCONJ + 1) = TEMPR - TEMPI                                  FOURT    385
```

```
  880 CONTINUE
      IF (KMIN - 1)890, 890, 900
  890 WR = WSTPR + 1.
      WI = WSTPI
      GO TO 910
  900 TEMPR = WR
      WR = WR * WSTPR - WI * WSTPI + WR
      WI = TEMPR * WSTPI + WI * WSTPR + WI
  910 TWOWR = WR + WR
      IF (ICASE - 3)930, 920, 930
  920 IF (IFP1 - NP2)950, 930, 930
  930 K = 1
      I2MAX = I3 + NP2 - NP1
      DO 940 I2 = I3, I2MAX, NP1
      DATA (I2) = WORK (K)
      DATA (I2 + 1) = WORK (K + 1)
  940 K = K + 2
      GO TO 980
C
C     COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N ODD, BY CON-
C     JUGATE SYMMETRIES AT EACH STAGE.
C
  950 J3MAX = I3 + IFP2 - NP1
      DO 970 J3 = I3, J3MAX, NP1
      J2MAX = J3 + NP2 - J2STP
      DO 970 J2 = J3, J2MAX, J2STP
      J1MAX = J2 + J1RG2 - IFP2
      J1CNJ = J3 + J2MAX + J2STP - J2
      DO 970 J1 = J2, J1MAX, IFP2
      K = 1 + J1 - I3
      DATA (J1) = WORK (K)
      DATA (J1 + 1) = WORK (K + 1)
      IF (J1 - J2)970, 970, 960
  960 DATA (J1CNJ) = WORK (K)
      DATA (J1CNJ + 1) = - WORK (K + 1)
  970 J1CNJ = J1CNJ - IFP2
  980 CONTINUE
      IF = IF + 1
      IFP1 = IFP2
      IF (IFP1 - NP1)990, 990, 720
C
C     COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N EVEN, BY CON-
C     JUGATE SYMMETRIES.
C
  990 GO TO (1260, 1180, 1260, 1000), ICASE
 1000 NHALF = N
      N = N + N
      THETA = - TWOPI / FLOAT (N)
      IF (ISIGN)1020, 1010, 1010
 1010 THETA = - THETA
 1020 SINTH = SIN (THETA / 2.)
      WSTPR = - 2. * SINTH * SINTH
      WSTPI = SIN (THETA)
      WR = WSTPR + 1.
      WI = WSTPI
      IMIN = 3
      JMIN = 2 * NHALF - 1
      GO TO 1050
 1030 J = JMIN
      DO 1040 I = IMIN, NTOT, NP2
      SUMR = (DATA (I) + DATA (J)) / 2.
      SUMI = (DATA (I + 1) + DATA (J + 1)) / 2.
      DIFR = (DATA (I) - DATA (J)) / 2.
      DIFI = (DATA (I + 1) - DATA (J + 1)) / 2.
      TEMPR = WR * SUMI + WI * DIFR
      TEMPI = WI * SUMI - WR * DIFR
      DATA (I) = SUMR + TEMPR
      DATA (I + 1) = DIFI + TEMPI
      DATA (J) = SUMR - TEMPR
      DATA (J + 1) = - DIFI + TEMPI
 1040 J = J + NP2
      IMIN = IMIN + 2
      JMIN = JMIN - 2
      TEMPR = WR
      WR = WR * WSTPR - WI * WSTPI + WR
      WI = TEMPR * WSTPI + WI * WSTPR + WI
 1050 IF (IMIN - JMIN)1030, 1060, 1090
```

```
1060  IF (ISIGN)1070, 1090, 1090
1070  DO 1080 I = IMIN, NTOT, NP2
1080  DATA (I + 1) = - DATA (I + 1)
1090  NP2 = NP2 + NP2
      NTOT = NTOT + NTOT
      J = NTOT + 1
      IMAX = NTOT / 2 + 1
1100  IMIN = IMAX - 2 * NHALF
      I = IMIN
      GO TO 1120
1110  DATA (J) = DATA (I)
      DATA (J + 1) = - DATA (I + 1)
1120  I = I + 2
      J = J - 2
      IF (I - IMAX)1110, 1130, 1130
1130  DATA (J) = DATA (IMIN) - DATA (IMIN + 1)
      DATA (J + 1) = 0.
      IF (I - J)1150, 1170, 1170
1140  DATA (J) = DATA (I)
      DATA (J + 1) = DATA (I + 1)
1150  I = I - 2
      J = J - 2
      IF (I - IMIN)1160, 1160, 1140
1160  DATA (J) = DATA (IMIN) + DATA (IMIN + 1)
      DATA (J + 1) = 0.
      IMAX = IMIN
      GO TO 1100
1170  DATA (1) = DATA (1) + DATA (2)
      DATA (2) = 0.
      GO TO 1260
C
C     COMPLETE A REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION BY
C     CONJUGATE SYMMETRIES.
C
1180  IF (I1RNG - NP1)1190, 1260, 1260
1190  DO 1250 I3 = 1, NTOT, NP2
      I2MAX = I3 + NP2 - NP1
      DO 1250 I2 = I3, I2MAX, NP1
      IMIN = I2 + I1RNG
      IMAX = I2 + NP1 - 2
      JMAX = 2 * I3 + NP1 - IMIN
      IF (I2 - I3)1210, 1210, 1200
1200  JMAX = JMAX + NP2
1210  IF (IDIM - 2)1240, 1240, 1220
1220  J = JMAX + NP0
      DO 1230 I = IMIN, IMAX, 2
      DATA (I) = DATA (J)
      DATA (I + 1) = - DATA (J + 1)
1230  J = J - 2
1240  J = JMAX
      DO 1250 I = IMIN, IMAX, NP0
      DATA (I) = DATA (J)
      DATA (I + 1) = - DATA (J + 1)
1250  J = J - NP0
C
C     END OF LOOP ON EACH DIMENSION
C
1260  NP0 = NP1
      NP1 = NP2
1270  NPREV = N
1280  RETURN

      END
```

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 PO 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 θ' 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.

```
      SUBROUTINE PARAB(FOD,DOL,BLOCK,DEFOCUS,ACOSE,ACOSH,THETA,ETHETA,EPHPARAB
     *I)                                                                PARAB
C     RADIATION PATTERNS FROM A DEFOCUSED PARABOLOID                    PARAB
C     PROGRAMMER - W.V.T. RUSCH                                         PARAB
C     16 MAJ 1974                                                       PARAB
C     MODIFIED 12 MAY 1976                                              PARAB
      COMPLEX AUX(11),ROMB,CMPLX,A1,D1,ETHETA,EPHI                      PARAB
      COMMON/DATA/FOL,PI,SINT,COST,DEFOCSS,ACOSEE,ACOSHH                PARAB
      EXTERNAL ETINT,EPINT                                              PARAB
      DEFOCSS=DEFOCUS                                                   PARAB
      ACOSEE=ACOSE                                                      PARAB
      ACOSHH=ACOSH                                                      PARAB
      PI=4.0*ATAN(1.0)                                                  PARAB
      DTR=PI/180.0                                                      PARAB
      RTD=180.0/PI                                                      PARAB
      FOL = FOD*DOL                                                     PARAB
      A = 2.0*ATAN(4.0*FOD)                                             PARAB
      IF(BLOCK.LT.0.0001) B = PI                                        PARAB
      IF(BLOCK.GE.0.0001) B = 2.0*ATAN(4.0*FOD/BLOCK)                   PARAB
      COST = COS(THETA*DTR)                                             PARAB
      SINT = SIN(THETA*DTR)                                             PARAB
      CALL QATRC(A,B,1.0E-03,11,ETINT,ROMB,IER,AUX)                     PARAB
      IF (IER .EQ. 1)   PRINT 1000                                      PARAB
      IF (IER .EQ. 2)   PRINT 1010                                      PARAB
      ETHETA = CMPLX(0.0,1.0)*2.0*PI*FOL*ROMB                           PARAB
      CALL QATRC(A,B,1.0E-03,11,EPINT,ROMB,IER,AUX)                     PARAB
      IF (IER .EQ. 2)   PRINT 2010                                      PARAB
      IF (IER .EQ. 1)   PRINT 2000                                      PARAB
      EPHI   = CMPLX(0.0,1.0)*2.0*PI*FOL*ROMB                           PARAB
      RETURN                                                            PARAB
C                                                                       PARAB
 1000 FORMAT(*  REQUIRED ACCURACY NOT ACHIEVED IN E-PLANE DUE TO ROUNDINPARAB
     1G ERRORS.*)                                                       PARAB
 1010 FORMAT(*  REQUIRED ACCURACY NOT ACHIEVED IN E-PLANE DUE TO INSUFFIPARAB
     1CIENT NUMBER OF INTEGRATION STEPS.*)                              PARAB
 2000 FORMAT(*  REQUIRED ACCURACY NOT ACHIEVED IN H-PLANE DUE TO ROUNDINPARAB
     1G ERRORS.*)                                                       PARAB
 2010 FORMAT(*  REQUIRED ACCURACY NOT ACHIEVED IN H-PLANE DUE TO INSUFFIPARAB
     1CIENT NUMBER OF INTEGRATION STEPS.*)                              PARAB
      END                                                               PARAB
```

```
      SUBROUTINE QATRC(XL,XU,EPS,NDIM,FCT,Y,IER,AUX)                    QATRC
C     ..................................................................QATRC
C                                                                       QATRC
C        SUBROUTINE QATRC                                               QATRC
C           COMPLEX VERSION OF SSP-ROUTINE QATR,  SEPT.72, HS-J         QATRC
C        PURPOSE                                                        QATRC
C           TO COMPUTE AN APPROXIMATION FOR INTEGRAL OF COMPLEX         QATRC
C           FUNCTION FCT(X) WITH REAL BOUNDARIES XL AND XU.             QATRC
C                                                                       QATRC
C        USAGE                                                          QATRC
C           CALL QATRC(XL,XU,EPS,NDIM,FCT,Y,IER,AUX)                    QATRC
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT.               QATRC
C                                                                       QATRC
C        DESCRIPTION OF PARAMETERS                                      QATRC
C           XL     - THE LOWER BOUND OF THE INTERVAL.                   QATRC
C           XU     - THE UPPER BOUND OF THE INTERVAL.                   QATRC
C           EPS    - THE UPPER BOUND OF THE ABSOLUTE ERROR.             QATRC
C           NDIM   - THE DIMENSION OF THE AUXILIARY STORAGE ARRAY AUX.  QATRC
C                    NDIM-1 IS THE MAXIMAL NUMBER OF BISECTIONS OF      QATRC
C                    THE INTERVAL (XL,XU).                              QATRC
C           FCT    - THE NAME OF THE EXTERNAL FUNCTION SUBPROGRAM USED. QATRC
C           Y      - THE RESULTING APPROXIMATION FOR THE INTEGRAL VALUE.QATRC
C           IER    - A RESULTING ERROR PARAMETER.                       QATRC
C           AUX    - AN AUXILIARY STORAGE ARRAY WITH DIMENSION NDIM.    QATRC
C                                                                       QATRC
C        REMARKS                                                        QATRC
C           ERROR PARAMETER IER IS CODED IN THE FOLLOWING FORM          QATRC
C           IER=0  - IT WAS POSSIBLE TO REACH THE REQUIRED ACCURACY.    QATRC
C                    NO ERROR.                                          QATRC
C           IER=1  - IT IS IMPOSSIBLE TO REACH THE REQUIRED ACCURACY    QATRC
C                    BECAUSE OF ROUNDING ERRORS.                        QATRC
C           IER=2  - IT WAS IMPOSSIBLE TO CHECK ACCURACY BECAUSE NDIM   QATRC
C                    IS LESS THAN 5, OR THE REQUIRED ACCURACY COULD NOT QATRC
C                    BE REACHED WITHIN NDIM-1 STEPS. NDIM SHOULD BE     QATRC
C                    INCREASED.                                         QATRC
C                                                                       QATRC
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  QATRC
C           THE EXTERNAL FUNCTION SUBPROGRAM FCT(X) MUST BE CODED BY    QATRC
C           THE USER. ITS ARGUMENT X SHOULD NOT BE DESTROYED.           QATRC
C                                                                       QATRC
C        METHOD                                                         QATRC
C           EVALUATION OF Y IS DONE BY MEANS OF TRAPEZOIDAL RULE IN     QATRC
C           CONNECTION WITH ROMBERGS PRINCIPLE. ON RETURN Y CONTAINS    QATRC
C           THE BEST POSSIBLE APPROXIMATION OF THE INTEGRAL VALUE AND   QATRC
C           VECTOR AUX THE UPWARD DIAGONAL OF ROMBERG SCHEME.           QATRC
C           COMPONENTS AUX(I) (I=1,2,...,IEND, WITH IEND LESS THAN OR   QATRC
C           EQUAL TO NDIM) BECOME APPROXIMATIONS TO INTEGRAL VALUE WITH QATRC
C           DECREASING ACCURACY BY MULTIPLICATION WITH (XU-XL).         QATRC
C           FOR REFERENCE, SEE                                          QATRC
C           (1) FILIPPI, DAS VERFAHREN VON ROMBERG-STIEFEL-BAUER ALS    QATRC
C               SPEZIALFALL DES ALLGEMEINEN PRINZIPS VON RICHARDSON,    QATRC
C               MATHEMATIK-TECHNIK-WIRTSCHAFT, VOL.11, ISS.2 (1964),    QATRC
C               PP.49-54.                                               QATRC
C           (2) BAUER, ALGORITHM 60, CACM, VOL.4, ISS.6 (1961), PP.255. QATRC
C                                                                       QATRC
C     ..................................................................QATRC
C                                                                       QATRC
C                                                                       QATRC
C                                                                       QATRC
      COMPLEX FCT,Y,SM,AUX(NDIM)                                        QATRC
C                                                                       QATRC


C     PREPARATIONS OF ROMBERG-LOOP                                      QATRC
      AUX(1)=.5*(FCT(XL)+FCT(XU))                                       QATRC
      H=XU-XL                                                           QATRC
      IF(NDIM-1)8,8,1                                                   QATRC
    1 IF(H)2,10,2                                                       QATRC
C                                                                       QATRC
C     NDIM IS GREATER THAN 1 AND H IS NOT EQUAL TO 0.                   QATRC
    2 HH=H                                                              QATRC
      E=EPS/ABS(H)                                                      QATRC
      DELT2=0.                                                          QATRC
      P=1.                                                              QATRC
      JJ=1                                                              QATRC
      DO 7 I=2,NDIM                                                     QATRC
      Y=AUX(1)                                                          QATRC
      DELT1=DELT2                                                       QATRC
      HD=HH                                                             QATRC
```

```
      HH=.5*HH                                                          QATRC  78
      P=.5*P                                                            QATRC  79
      X=XL+HH                                                           QATRC  80
      SM=(0.,0.)                                                        QATRC  81
      DO 3 J=1,JJ                                                       QATRC  82
      SM=SM+FCT(X)                                                      QATRC  83
    3 X=X+HD                                                            QATRC  84
      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 ROMBERGS EXTRAPOLATION METHOD.                           QATRC  89
      Q=1.                                                              QATRC  90
      JI=I-1                                                            QATRC  91
      DO 4 J=1,JI                                                       QATRC  92
      II=I-J                                                            QATRC  93
      Q=Q+Q                                                             QATRC  94
      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
      DELT2=CABS(Y-AUX(1))                                              QATRC  99
      IF(I-5)7,5,5                                                      QATRC 100
    5 IF(DELT2-E)10,10,6                                                QATRC 101
    6 IF(DELT2-DELT1)7,11,11                                            QATRC 102
    7 JJ=JJ+JJ                                                          QATRC 103
    8 IER=2                                                             QATRC 104
    9 Y=H*AUX(1)                                                        QATRC 105
      RETURN                                                            QATRC 106
   10 IER=0                                                             QATRC 107
      GO TO 9                                                           QATRC 108
   11 IER=1                                                             QATRC 109
      Y=H*Y                                                             QATRC 110
      RETURN                                                            QATRC 111
      END                                                               QATRC 112
```

```
      COMPLEX FUNCTION ETINT(X)                                         ETINT
      COMMON/DATA/FCL,PI,SINT,COST,DFOCUS,ACOSF,ACOSH                   ETINT
      DIMENSION BJ(1000)                                                ETINT
C     NOTE THAT COS(PI-XP) = (DFOCUS-RHO*COSX)/RHOPRIME = R2/RHPOVL     ETINT
      COMPLEX CEXP,CMPLX,A1,D1,HX                                       ETINT
      SINX = SIN(X)                                                     ETINT
      COSX = COS(X)                                                     ETINT
      RHOOVL = 2.0*FCL/(1.0-COSX)                                       ETINT
      RHPOVL = SQRT(RHOOVL*RHOOVL+DFOCUS*DFOCUS-2.*DFOCUS*RHOOVL*COSX)  ETINT
      R1 = RHOOVL*SINX                                                  ETINT
      R2 = DFOCUS - RHOOVL*COSX                                         ETINT
      XP = PI - ATAN2(R1,R2)                                            ETINT
      CSPMXP = R2/RHPOVL                                                ETINT
      FAZE = 2.0*PI*(RHOOVL*COSX*COST-RHPOVL)                           ETINT
      BETA = 2.0*PI*RHOOVL*SINX*SINT                                    ETINT
      IF(BETA.GT.0.0) GO TO 2                                           ETINT
      BESS0 = 1.0                                                       ETINT
      BESS1 = 0.0                                                       ETINT
      BESS2 = 0.0                                                       ETINT
      GO TO 3                                                           ETINT
2     CALL BESFUN(BETA,BJ,4)                                            ETINT
      BESS0 = BJ(1)                                                     ETINT
      BESS1 = BJ(2)                                                     ETINT
      BESS2 = BJ(3)                                                     ETINT
3     CONTINUE                                                          ETINT
      IF(ACOSF.GE.(-100.0))GO TO 20                                     ETINT
      A1 = 2.0/(1.0+CSPMXP)                                             ETINT
      D1 = -A1                                                          ETINT
      GO TO 50                                                          ETINT
20    IF(ACOSE.GE.0.0)GO TO 40                                          ETINT
         A1 = CSPMXP                                                    ETINT
      D1 = -1.0                                                         ETINT
      GO TO 50                                                          ETINT
40    A1 = CSPMXP**ACOSF                                                ETINT
      D1 = -CSPMXP**ACOSH                                               ETINT
50    CONTINUE                                                          ETINT
      HX = (RHOOVL/RHPOVL)*(A1*COST*(BESS0-BESS2)-D1*COST*(BESS0+BESS2)*ETINT
     *SIN(XP-(X/2.0))/SIN(X/2.0)-2.0*CMPLX(0.0,1.0)*A1*BESS1*COS(X/2.0)/ETINT
     *SIN(X/2.0)*SINT)                                                  ETINT
      HX = HX/(1.0-COSX)                                                ETINT
      ETINT = HX*SINX*CEXP(CMPLX(0.0,FAZE))                             ETINT
      RETURN                                                            ETINT
      END                                                               ETINT
```

```
      COMPLEX FUNCTION EPINT(X)
      COMMON/DATA/FOL,PI,SINT,COST,DFOCUS,ACOSE,ACOSH
      DIMENSION BJ(1000)
C     NOTE THAT COS(PI-XP) = (DFOCUS-RHO*COSX)/RHOPPIME = R2/RHPOVL
      COMPLEX CEXP,CMPLX,A1,D1,HX
      SINX = SIN(X)
      COSX = COS(X)
      RHOOVL = 2.0*FOL/(1.0-COSX)
      RHPOVL = SQRT(RHOOVL*RHOOVL+DFOCUS*DFOCUS-2.*DFOCUS*RHOOVL*COSX)
      R1 = RHOOVL*SINX
      R2 = DFOCUS - RHOOVL*COSX
      XP = PI - ATAN2(R1,R2)
      CSPMXP = R2/RHPOVL
      FAZE = 2.0*PI*(RHOOVL*COSX*COST-RHPOVL)
      BETA = 2.0*PI*RHOOVL*SINX*SINT
      IF(BETA.GT.0.0) GO TO 2
      BESS0 = 1.0
      BESS1 = 0.0
      BESS2 = 0.0
      GO TO 3
    2 CALL BESFUN(BETA,BJ,4)
      BESS0 = BJ(1)
      BESS1 = BJ(2)
      BESS2 = BJ(3)
    3 CONTINUE
      IF(ACOSE.GE.(-100.0))GO TO 20
      A1 = 2.0/(1.0+CSPMXP)
      D1 = -A1
      GO TO 50
   20 IF(ACOSE.GE.0.0)GO TO 40
      A1 = CSPMXP
      D1 = -1.0
      GO TO 50
   40 A1 = CSPMXP**ACOSE
      D1 = -CSPMXP**ACOSH
   50 CONTINUE
      HX = (RHOOVL/RHPOVL)*(A1*(BESS0+BESS2)-D1*(BESS0-BESS2)*SIN(XP-(X/
     *2.0))/SIN(X/2.))
      HX = HX/(1.0-COSX)
      EPINT = HX*SINX*CEXP(CMPLX(0.0,FAZE))
      RETURN
      END
```

```
      SUBROUTINE BESFUN(X,BJ,NMAX)
      DIMENSION BJ(1)
C     NOTE BJ(1)=J0 BJ(2)=J1 ... BJ(200)=J199
10    IF(X.GT.1.0E-03) GO TO 18
      BJ(1)=1.0
      DO 15 JBJ=2,NMAX
15    BJ(JBJ)=0.0
      RETURN
18    IF(X.GT.NMAX) M=2*X+7
      IF(X.LT.NMAX) M=2*NMAX+7
      IF(M.LT.990) GO TO 19
      WRITE(6,2000)
2000  FORMAT(10X,33H M EXCEEDS 190. EXECUTION ABORTED)
      STOP
19    FM1=10.E-28
      FM=0.0
      ALPHA=0.
      IF(M-(M/2)*2) 20,30,20
20    JT=1
      GO TO 40
30    JT=-1
40    M2=M-2
      DO 160 K=1,M2
      MK=M-K
      XMK=MK
      BMK=2.*XMK*FM1/X-FM
      FM=FM1
      FM1=BMK
      BJ(MK)=BMK
      JT=-JT
      S=1+JT
      ALPHA=ALPHA+BMK*S
160   CONTINUE
      BMK=2.0*FM1/X-FM
      BJ(1)=BMK
      ALPHA=ALPHA+BMK
      DO 200 IN=1,NMAX
      BJ(IN)=BJ(IN)/ALPHA
200   CONTINUE
      RETURN
      END
```

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.

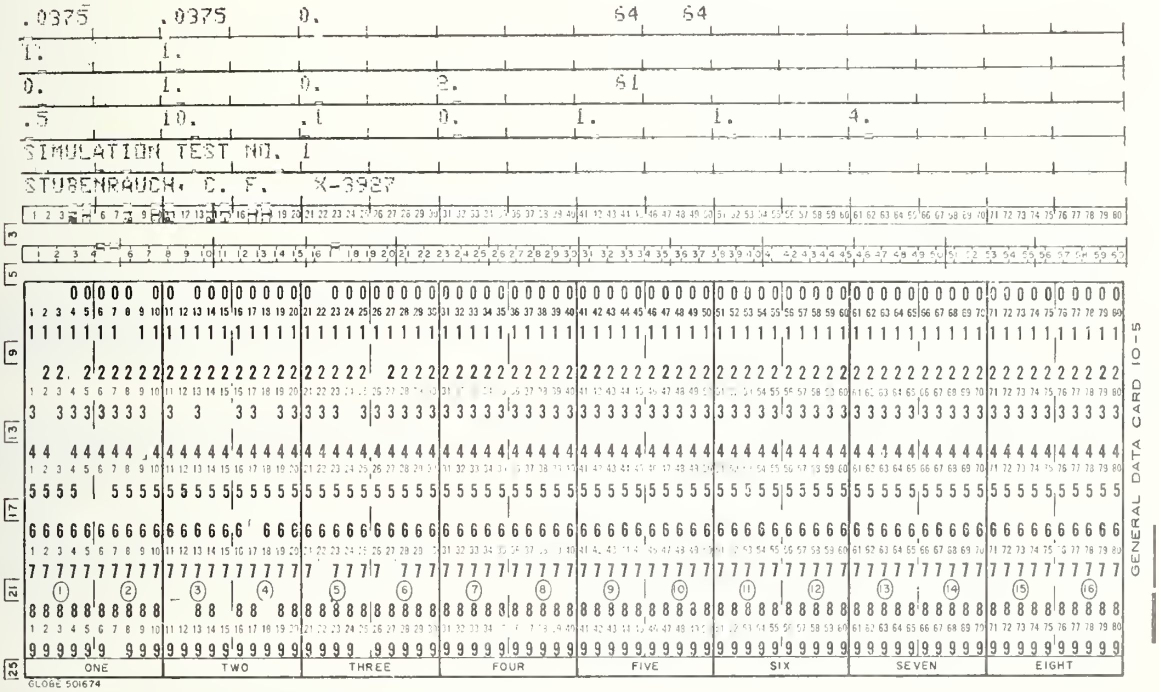
DISCUSSION:

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.

```
      SUBROUTINE PLT120R(X, Y, XMAX, XMIN, YMAX, YMIN, LAST, ISYMBOL, NOPLT120R
     1, MOST)                                                           PLT120R
C     MODIFIED 11/4/68                                                  PLT120R
      DIMENSION X(1), Y(1), ZX(13), GRAPH(121, 51)                      PLT120R
      INTEGER GRAPH, COLUMNS, BLANK, BORDER                             PLT120R
      DATA (LINES = 51), (COLUMNS = 121)                                PLT120R
      KMAX = COLUMNS / 10 + 1                                           PLT120R
      IF (NO .NE. 1) GO TO 190                                          PLT120R
      YLAR = YMAX                                                       PLT120R
      YSMA = YMIN                                                       PLT120R
      XLAR = XMAX                                                       PLT120R
      XSMA = XMIN                                                       PLT120R
      BORDER = 1HI                                                      PLT120R
      BLANK = 1H                                                        PLT120R
      MATRIX = COLUMNS * LINES                                          PLT120R
      IF (MATRIX .LT. 1) GO TO 120                                      PLT120R
      DO 100 I = 1, MATRIX                                              PLT120R
  100 GRAPH(I) = BLANK                                                  PLT120R
  120 CONTINUE                                                          PLT120R
      IF (LINES .LT. 1) GO TO 140                                       PLT120R
      DO 130 I = 1, LINES                                               PLT120R
  130 GRAPH(1, I) = GRAPH(COLUMNS, I) = BORDER                          PLT120R
  140 CONTINUE                                                          PLT120R
      IF (COLUMNS .LT. 1) GO TO 160                                     PLT120R
      DO 150 I = 1, COLUMNS                                             PLT120R
  150 GRAPH(I, 26) = 1H.                                                PLT120R
  160 CONTINUE                                                          PLT120R
      XSCALE = (XLAR - XSMA) / (COLUMNS - 1.)                           PLT120R
      YSCALE = (YLAR - YSMA) / (LINES - 1.)                             PLT120R
      IF (KMAX .LT. 1) GO TO 180                                        PLT120R
      DO 170 K = 1, KMAX                                                PLT120R
  170 ZX(K) = 10. * FLOAT(K - 1) * XSCALE + XSMA                        PLT120R
  180 CONTINUE                                                          PLT120R
  190 IF (LAST .LT. 1) GO TO 250                                        PLT120R
      DO 240 I = 1, LAST                                                PLT120R
      IF (X(I) .GT. XLAR .OR. X(I) .LT. XSMA) GO TO 240                 PLT120R
      IF (Y(I) .GT. YLAR .OR. Y(I) .LT. YSMA) GO TO 240                 PLT120R
      IX = (X(I) - XSMA) / XSCALE + 1.5                                 PLT120R
      IY = (Y(I) - YSMA) / YSCALE + .5                                  PLT120R
      IY = LINES - IY                                                   PLT120R
      GRAPH(IX, IY) = ISYMBOL                                           PLT120R
  240 CONTINUE                                                          PLT120R
  250 CONTINUE                                                          PLT120R
      IF (NO .NE. MOST) RETURN                                          PLT120R
      PRINT 1500                                                        PLT120R
      YES = YLAR + YSCALE                                               PLT120R
      IF (LINES .LT. 1) GO TO 270                                       PLT120R
      DO 260 I = 1, LINES                                               PLT120R
      YES = YES - YSCALE                                                PLT120R
      PRINT 1510, YES, (GRAPH(J, I), J = 1, COLUMNS)                    PLT120R
  260 CONTINUE                                                          PLT120R
  270 CONTINUE                                                          PLT120R
      PRINT 1520                                                        PLT120R
      PRINT 1530, ZX                                                    PLT120R
      RETURN                                                            PLT120R
 1500 FORMAT  (1H1,9X,24(5HI....)1HI)                                   PLT120R
 1510 FORMAT  (1H ,F8.2,1X,121A1)                                       PLT120R
 1520 FORMAT  (1H ,9X,24(5HI....)1HI)                                   PLT120R
 1530 FORMAT  (1H ,2X,13(1X,F9.3))                                      PLT120R
      END                                                               PLT120R
```

APPENDIX A.2 SAMPLE PROGRAM INPUT AND OUTPUT

Illustrated below is a typical input card deck for program POMODL. The output produced using this deck is reproduced on the following pages.



RADIATION PATTERN OF AN AXIALLY DEFOCUSED PARABOLOID

SIMULATION TEST NO. 1
REFLECTOR PARAMETERS -

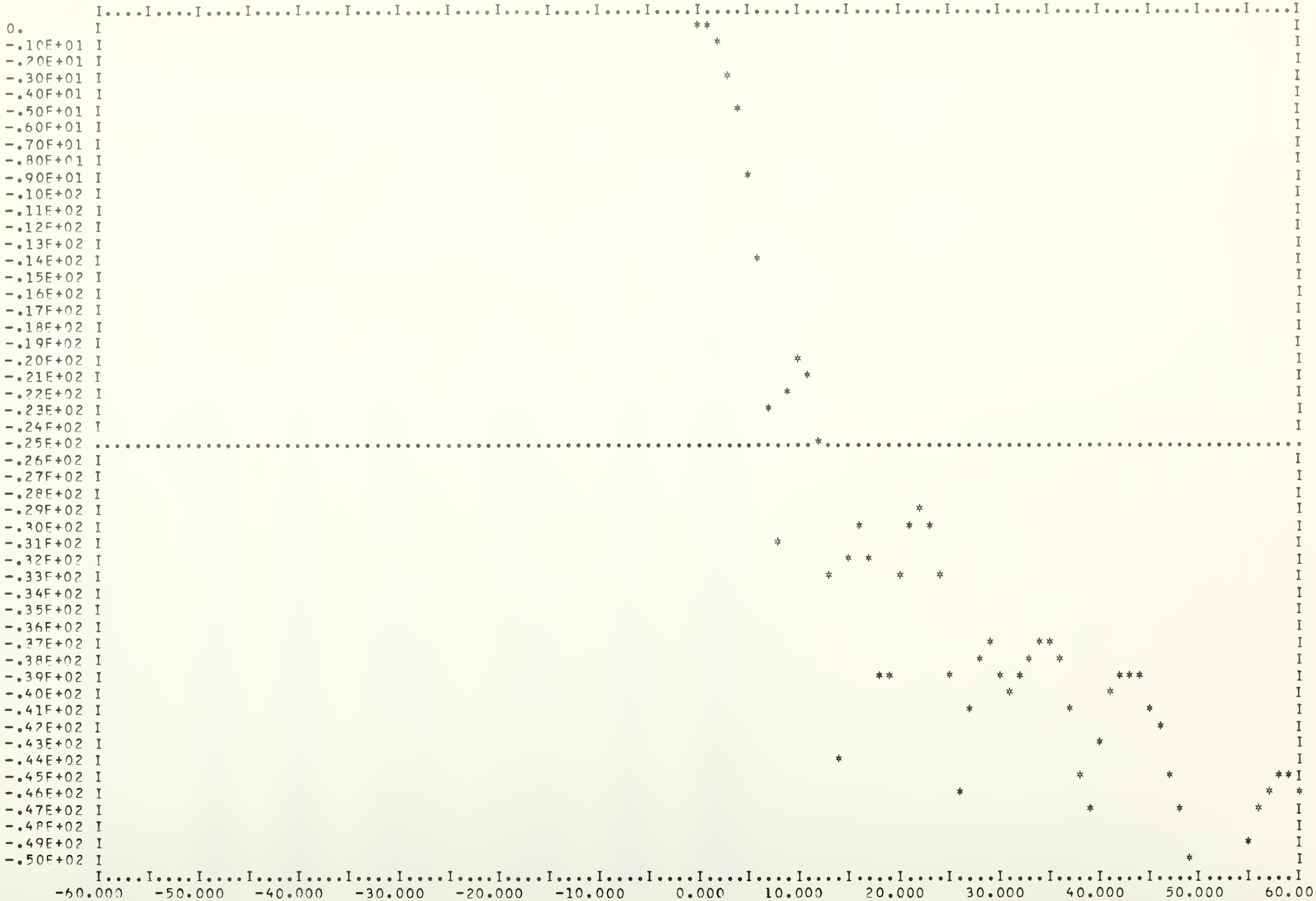
F/D = .500
DIAMETER = 10.00 WAVELENGTHS
FRACTIONAL DIAMETER BLOCKING = .100
AXIAL DEFOCUSING = 0.000 WAVELENGTHS BEYOND FOCUS
FREQUENCY = 4.0000 GHZ.

FEED E-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

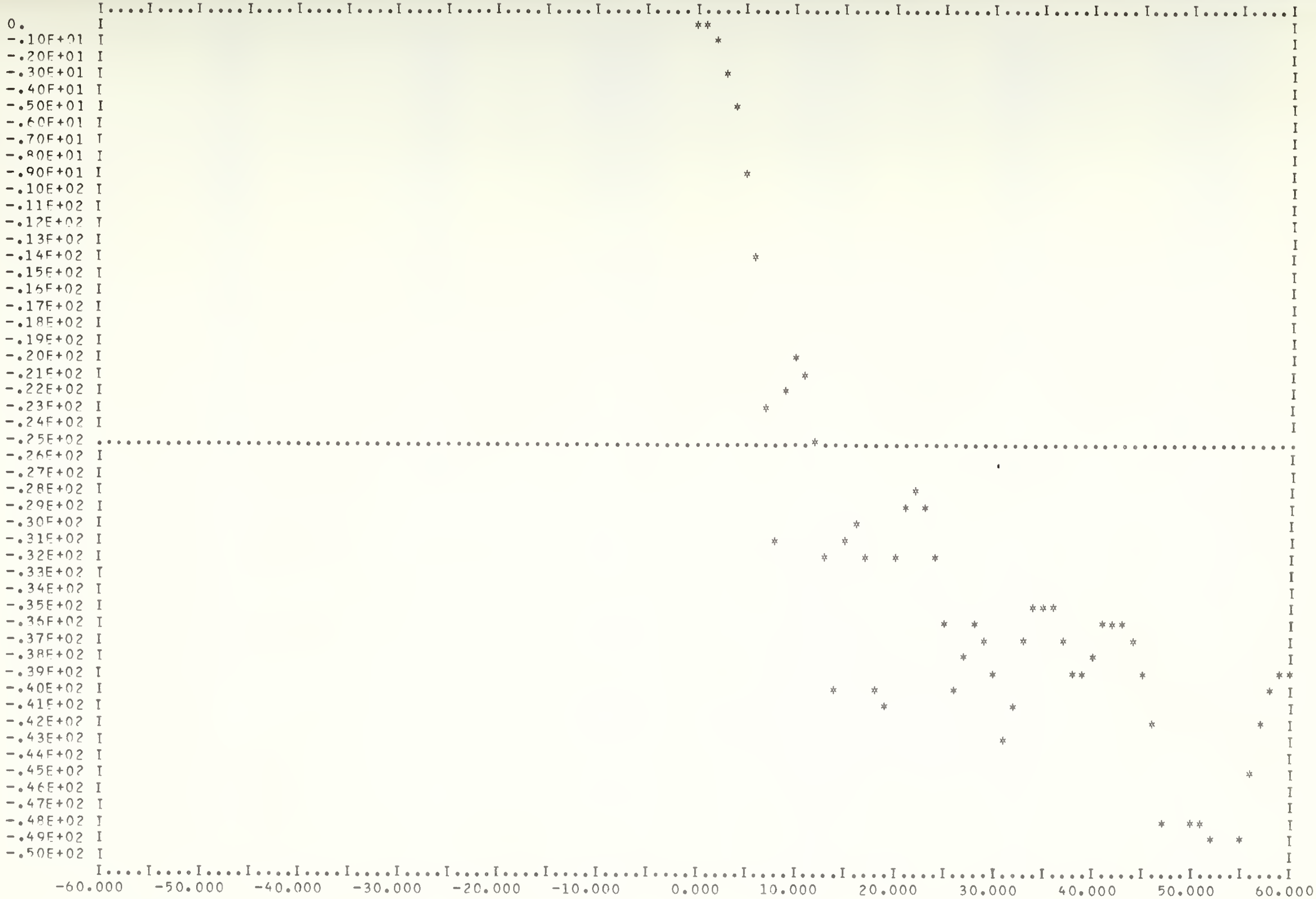
		E-PLANE			H-PLANE	
THETA (DEG)	MAG (VOLTS)	MAG (DB)	PHASE (DEG)	MAG (VOLTS)	MAG (DB)	PHASE (DEG)
0.00	10.956	0.00	-90.00	10.956	0.00	-90.00
1.00	10.591	-.29	-90.18	10.593	-.29	-90.24
2.00	9.550	-1.19	-90.73	9.556	-1.19	-90.98
3.00	7.975	-2.76	-91.62	7.986	-2.75	-92.23
4.00	6.078	-5.12	-92.85	6.092	-5.10	-94.04
5.00	4.100	-8.54	-94.35	4.114	-8.51	-96.51
6.00	2.269	-13.68	-95.92	2.280	-13.64	-100.03
7.00	.768	-23.09	-96.25	.773	-23.03	-107.16
8.00	.300	-31.26	69.95	.315	-30.83	90.05
9.00	.896	-21.75	72.27	.914	-21.57	75.78
10.00	1.075	-20.17	69.52	1.097	-19.99	69.54
11.00	.939	-21.34	65.99	.962	-21.13	63.24
12.00	.619	-24.95	62.02	.638	-24.70	55.32
13.00	.247	-32.95	57.70	.262	-32.44	39.86
14.00	.073	-43.54	-125.80	.105	-40.34	-88.29
15.00	.273	-32.08	-130.44	.292	-31.47	-124.01
16.00	.330	-30.42	-133.83	.348	-29.95	-134.64
17.00	.264	-32.37	-134.14	.273	-32.07	-142.14
18.00	.128	-38.67	-117.27	.109	-40.01	-143.42
19.00	.119	-39.25	-28.37	.093	-41.39	1.80
20.00	.260	-32.51	-9.55	.268	-32.23	1.69
21.00	.357	-29.74	-11.29	.383	-29.13	-6.86
22.00	.382	-29.15	-17.47	.418	-28.36	-17.62
23.00	.338	-30.23	-25.97	.377	-29.26	-30.63
24.00	.243	-33.08	-37.50	.283	-31.76	-47.98
25.00	.129	-38.55	-57.48	.173	-36.02	-76.70
26.00	.054	-46.10	-127.21	.114	-39.66	-133.87
27.00	.099	-40.88	164.41	.142	-37.77	173.25
28.00	.140	-37.84	146.81	.171	-36.12	147.69
29.00	.147	-37.44	140.58	.164	-36.51	133.76
30.00	.129	-38.57	143.57	.123	-39.02	129.84
31.00	.112	-39.83	158.93	.078	-42.99	150.84
32.00	.120	-39.22	178.26	.093	-41.40	-167.66
33.00	.145	-37.59	-173.63	.147	-37.45	-160.09
34.00	.162	-36.59	-175.84	.188	-35.31	-168.34
35.00	.161	-36.68	175.84	.202	-34.68	177.07
36.00	.138	-38.01	162.89	.189	-35.29	157.50
37.00	.100	-40.82	143.59	.157	-36.90	131.08
38.00	.059	-45.35	107.74	.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	-49.94
43.00	.127	-38.72	-64.50	.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	-56.54	-102.43
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	-39.23	-32.77
60.00	.056	-45.87	-67.52	.127	-38.73	-57.29



SIMULATION TEST NO. 1

E-PLANE



SIMULATION TEST NO. 1

H-PLANE

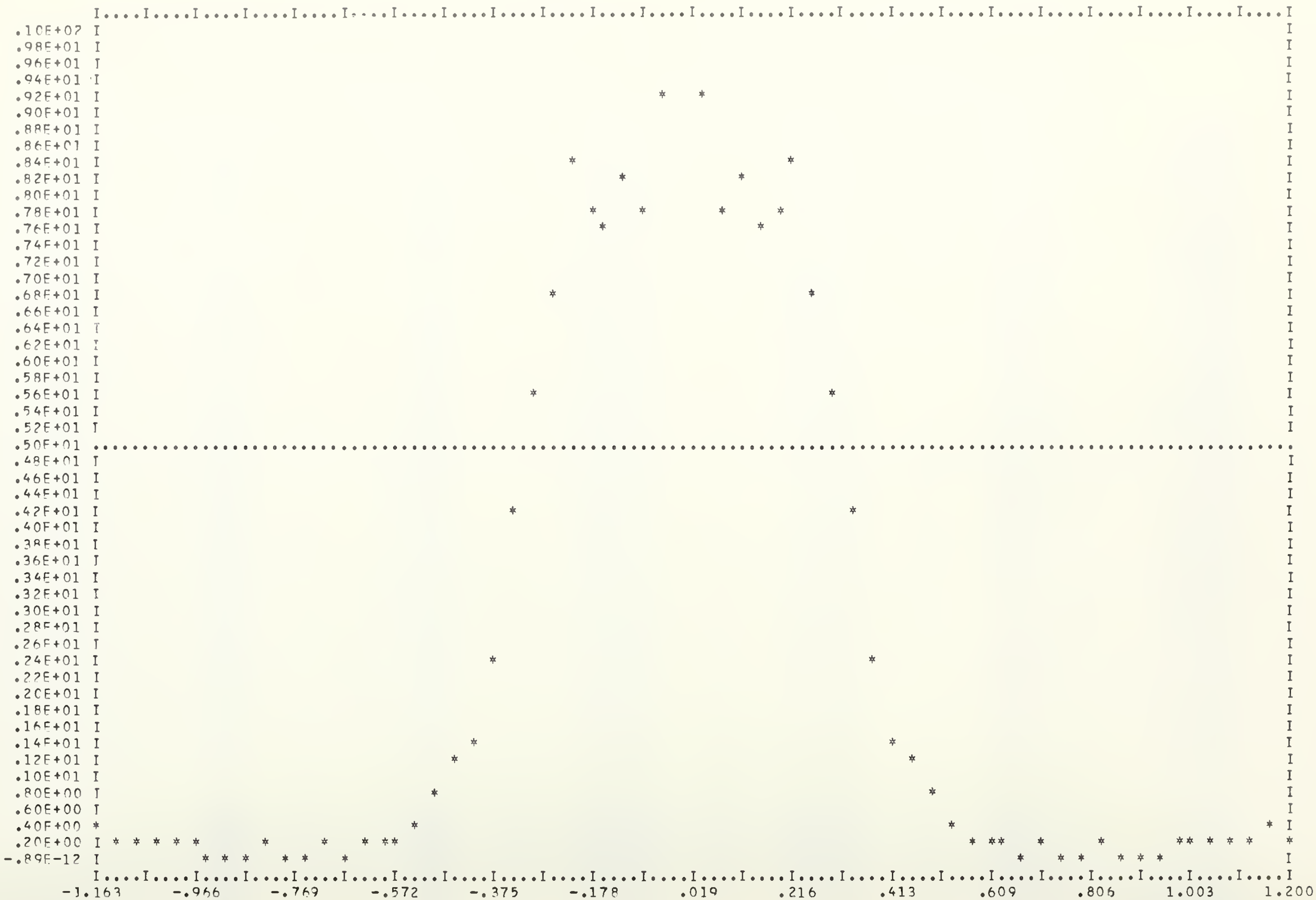
NEAR-FIELD PARAMETERS-

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

CENTERLINE DATA

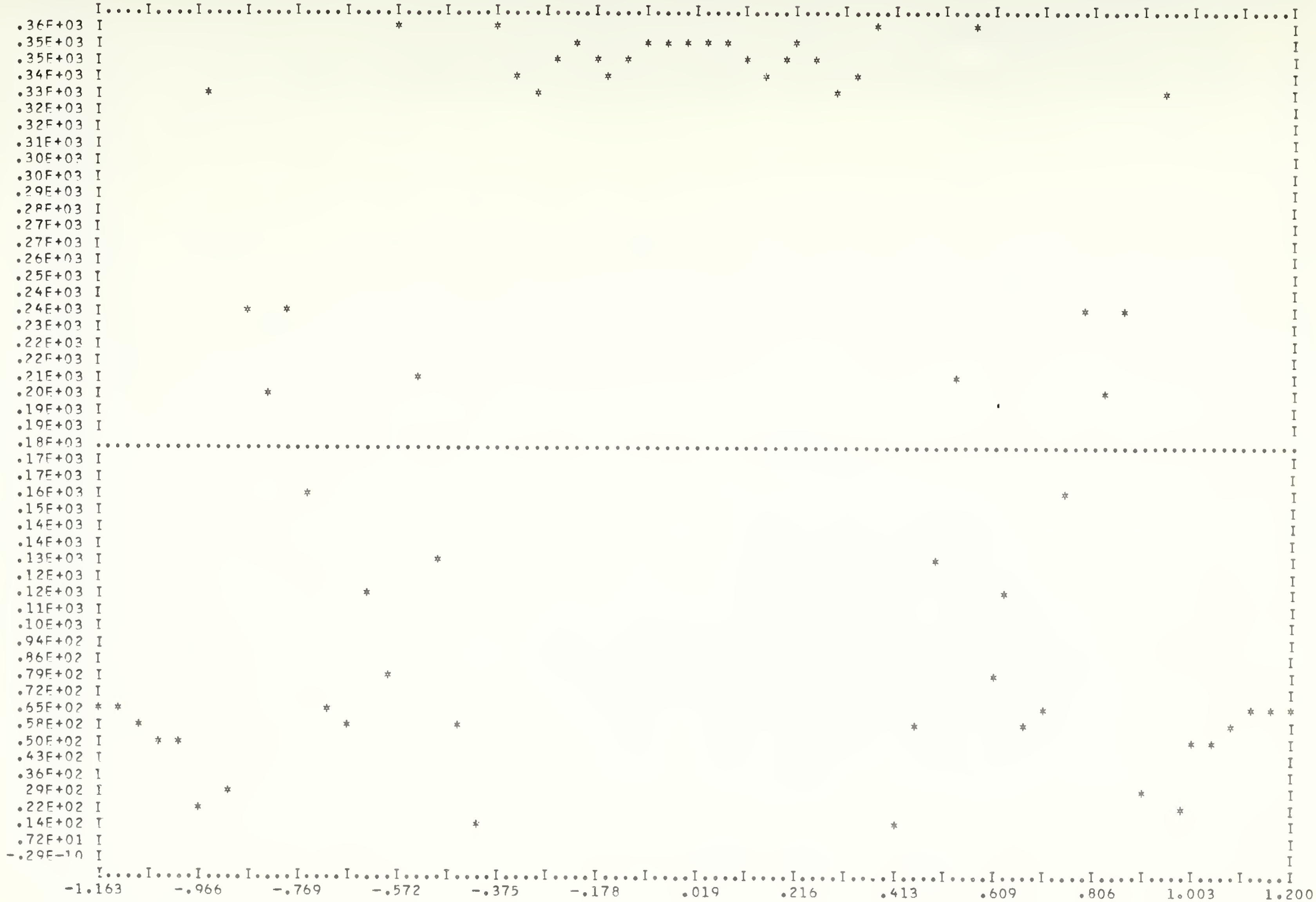
X	X-Z PLANE AMP	PHASE	Y	Y-Z PLANE AMP	PHASE
-1.1625	.3119	61.2176	-1.1625	.1165	311.0897
-1.1250	.2586	62.6402	-1.1250	.1691	251.0452
-1.0875	.2700	56.3574	-1.0875	.0678	290.6290
-1.0500	.1860	51.0251	-1.0500	.1434	288.1975
-1.0125	.1017	49.7359	-1.0125	.1306	246.4036
-.9750	.1005	22.5616	-.9750	.1648	333.6881
-.9375	.0323	334.2910	-.9375	.2296	251.3149
-.9000	.0915	25.5466	-.9000	.1018	355.7389
-.8625	.0541	234.5326	-.8625	.2632	258.5681
-.8250	.1004	202.4330	-.8250	.2007	203.5065
-.7875	.0474	235.7145	-.7875	.1884	299.5921
-.7500	.0574	158.4096	-.7500	.3507	229.5261
-.7125	.1690	67.3301	-.7125	.1132	64.1319
-.6750	.0839	55.2947	-.6750	.3866	279.2087
-.6375	.1915	117.6333	-.6375	.4112	190.2763
-.6000	.2915	78.3650	-.6000	.2265	70.3454
-.5625	.2269	358.5818	-.5625	.5183	310.5921
-.5250	.4015	211.5453	-.5250	.8110	223.4786
-.4875	.7639	129.7658	-.4875	1.0235	137.3943
-.4500	1.1069	60.6092	-.4500	1.4227	61.1873
-.4125	1.3686	16.3742	-.4125	1.4552	13.3092
-.3750	2.3373	358.8580	-.3750	2.3869	8.5232
-.3375	4.1240	338.0297	-.3375	4.3982	342.1280
-.3000	5.5214	332.0037	-.3000	5.4712	331.9989
-.2625	6.8479	344.7964	-.2625	6.6036	348.2612
-.2250	8.4890	352.4989	-.2250	8.8401	353.8764
-.1875	7.8894	345.2840	-.1875	7.9203	346.1406
-.1500	7.5194	335.4119	-.1500	7.4792	333.8087
-.1125	8.1906	347.3378	-.1125	8.2470	348.1900
-.0750	7.8913	354.3680	-.0750	7.7389	355.5398
-.0375	9.1533	353.5547	-.0375	9.0119	352.0960
0.0000	11.1600	355.6275	0.0000	11.1600	355.6275
.0375	9.1533	353.5547	.0375	9.0119	352.0960
.0750	7.8913	354.3680	.0750	7.7389	355.5398
.1125	8.1906	347.3378	.1125	8.2470	348.1900
.1500	7.5194	335.4119	.1500	7.4792	333.8087
.1875	7.8894	345.2840	.1875	7.9203	346.1406
.2250	8.4890	352.4989	.2250	8.8401	353.8764
.2625	6.8479	344.7964	.2625	6.6036	348.2612
.3000	5.5214	332.0037	.3000	5.4712	331.9989
.3375	4.1240	338.0297	.3375	4.3982	342.1280
.3750	2.3373	358.8580	.3750	2.3869	8.5232
.4125	1.3686	16.3742	.4125	1.4552	13.3092
.4500	1.1069	60.6092	.4500	1.4227	61.1873
.4875	.7639	129.7658	.4875	1.0235	137.3943
.5250	.4015	211.5453	.5250	.8110	223.4786
.5625	.2269	358.5818	.5625	.5183	310.5921
.6000	.2915	78.3650	.6000	.2265	70.3454
.6375	.1915	117.6333	.6375	.4112	190.2763
.6750	.0839	55.2947	.6750	.3866	279.2087
.7125	.1690	67.3301	.7125	.1132	64.1319
.7500	.0574	158.4096	.7500	.3507	229.5261
.7875	.0474	235.7145	.7875	.1884	299.5921

.8250	.1004	202.4330	.8250	.2007	203.5065
.8625	.0541	234.5326	.8625	.2632	258.5681
.9000	.0915	25.5466	.9000	.1018	355.7389
.9375	.0323	334.2910	.9375	.2296	251.3149
.9750	.1005	22.5616	.9750	.1648	333.6881
1.0125	.1017	49.7359	1.0125	.1306	246.4036
1.0500	.1860	51.0251	1.0500	.1434	288.1975
1.0875	.2700	56.3574	1.0875	.0678	290.6290
1.1250	.2536	62.6402	1.1250	.1691	251.0452
1.1625	.3119	61.2176	1.1625	.1165	311.0897
1.2000	.2981	56.8146	1.2000	.1658	245.8153



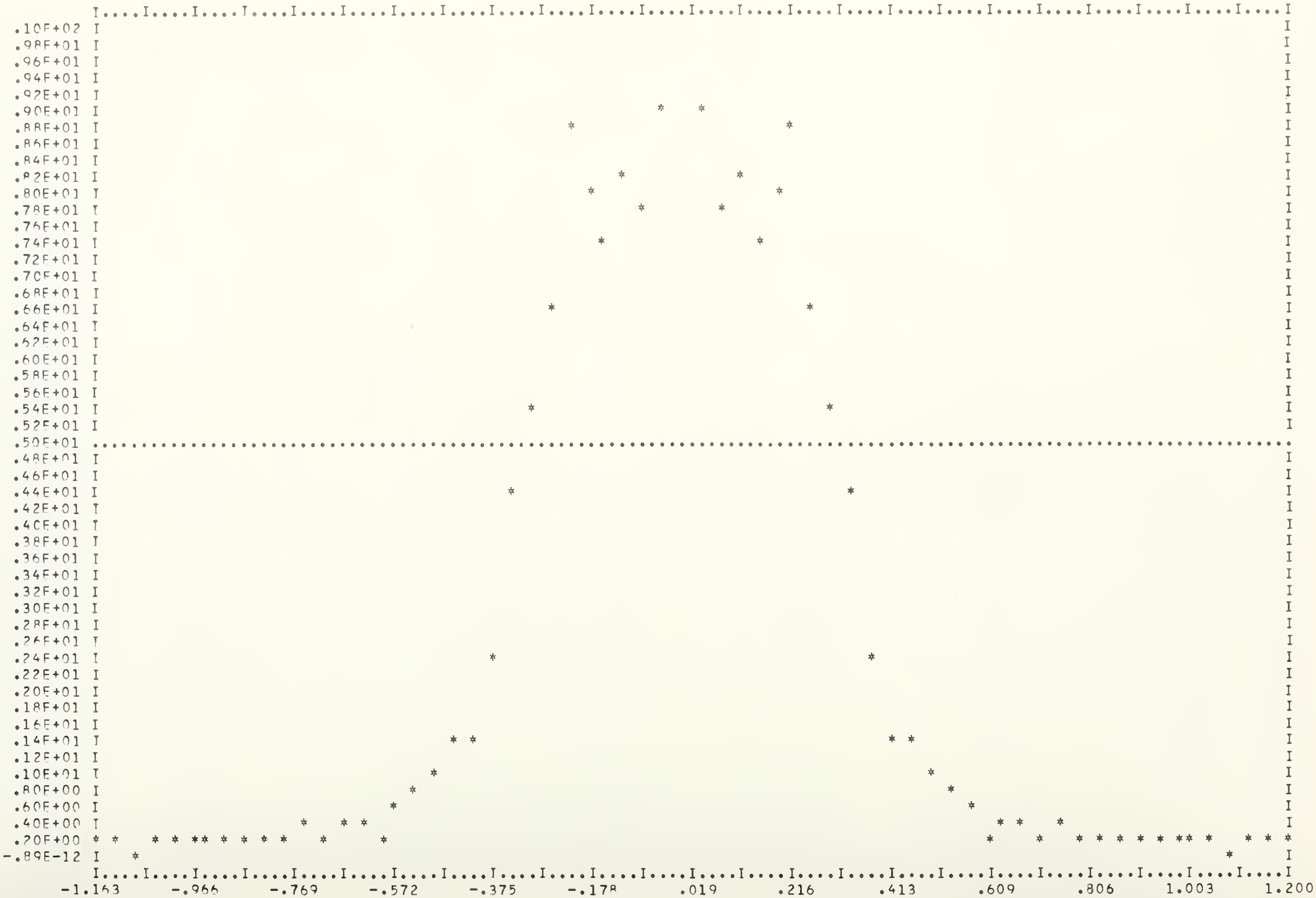
SIMULATION TEST NO. 1

Y-Z PLANE AMPLITUDE



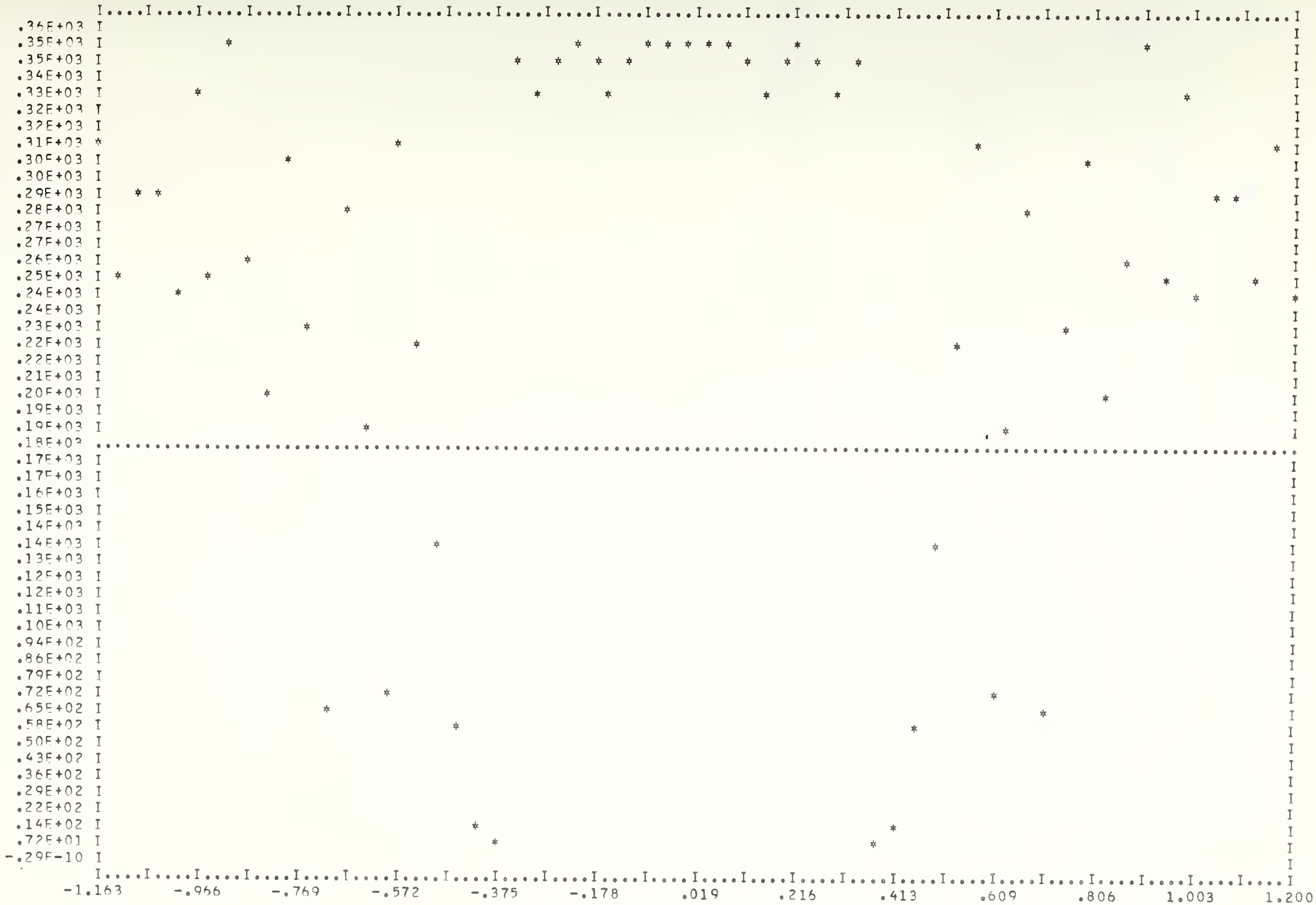
SIMULATION TEST NO. 1

Y-Z PLANE PHASE



SIMULATION TEST NO. 1

X-Z PLANE AMPLITUDE



SIMULATION TEST NO. 1

X-Z PLANE PHASE

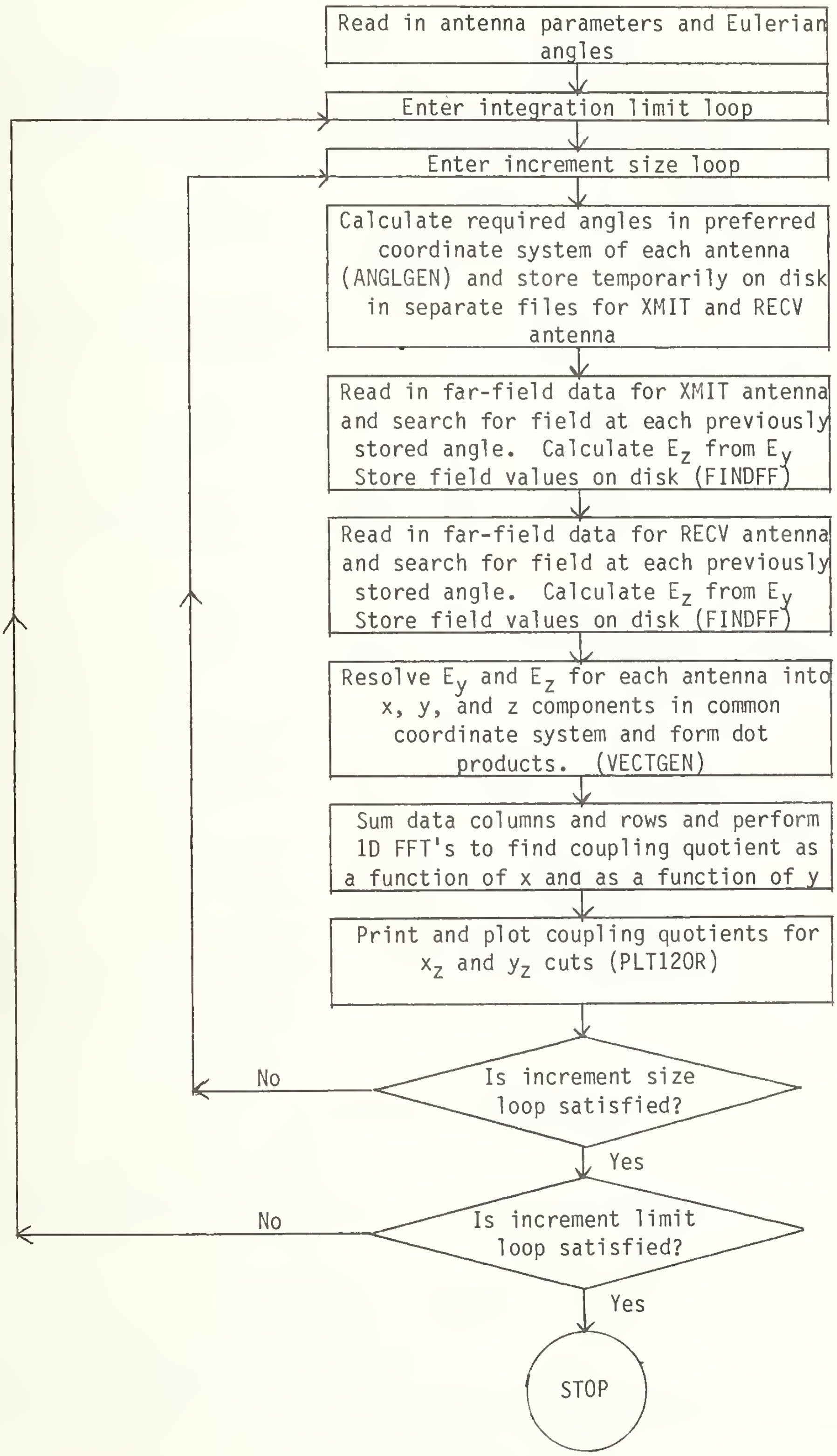
EOF FOUND ON LU5

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.


Read in antenna parameters and Eulerian angles
Enter integration limit loop
Enter increment size loop
Calculate required angles in preferred coordinate system of each antenna (ANGLGEN) and store temporarily on disk in separate files for XMIT and RECV antenna
Read in far-field data for XMIT antenna and search for field at each previously stored angle. Calculate E_z from E_y Store field values on disk (FINDFF)
Read in far-field data for RECV antenna and search for field at each previously stored angle. Calculate E_z from E_y Store field values on disk (FINDFF)
Resolve E_y and E_z for each antenna into x, y, and z components in common coordinate system and form dot products. (VECTGEN)
Sum data columns and rows and perform 1D FFT's to find coupling quotient as a function of x and as a function of y
Print and plot coupling quotients for x_z and y_z cuts (PLT12OR)
No
Is increment size loop satisfied?
Yes
No
Is increment limit loop satisfied?
Yes
STOP

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 Col. 41-50 Col. 51-60	PHI THETA - Eulerian angles of reoriented transmit antenna. 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 Col. 41-50 Col. 51-60	PHIP THETHP - Eulerian angles of reoriented receive antenna. 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 artifically 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 (PLT120R). 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., N1 ≈ (A1+A2)/DKOK and N2 ≈ (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 XO and YO, 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•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 recangular 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 preceeding 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 ($\emptyset_p$,θ_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 ($\emptyset_A$, θ_A of eqs (13) and shown in figure 2).

PI = π = 3.14159...

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 XO = 0, XO = 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 XKLIM; except for very close antennas XKLIM 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; XKLIM = XKMIN times XLIM when this product is less than XKMAX.

XKXOK,XKYOK = k_x/k and k_y/k, respectively.

XLIM = Intermediate variable used for adjusting XKLIM; 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 γ/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)
PLT12OR (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.

```
      PROGRAM CUPLNF (INPUT, OUTPUT, TAPE1, TAPE3, TAPE4, TAPE5,          CUPLNF
     1 TAPE6, TAPE7, TAPE8, TAPE60=INPUT)                                 CUPLNF
C                                                                         CUPLNF
C                                                                         CUPLNF
C-    GENERAL INFORMATION ABOUT PROGRAM                                   CUPLNF
C                                                                         CUPLNF
C     THIS PROGRAM COMPUTES THE COUPLING QUOTIENT BETWEEN A               CUPLNF
C     TRANSMITTING ANTENNA ON THE LEFT AND A RECEIVING ANTENNA ON THE     CUPLNF
C     RIGHT OF ARBITRARY RELATIVE ORIENTATION AND SEPARATION,             CUPLNF
C     FROM THE GIVEN COMPLEX FAR-FIELD PATTERN OF EACH ANTENNA.           CUPLNF
C                                                                         CUPLNF
C     THE COUPLING QUOTIENT IS COMPUTED ALONG XO AND YO PERPENDICULAR     CUPLNF
C     LINES OR CUTS.                                                      CUPLNF
C                                                                         CUPLNF
C     AX,AY,AND WORK SHOULD BE DIMENSIONED .GE. THE LARGER OF (N1,N2).    CUPLNF
C     ACLCUT AND X SHOULD BE DIMENSIONED AT LEAST 2 GREATER THAN THE      CUPLNF
C     LARGER OF (N1, N2).                                                 CUPLNF
C     EYT, EYR, EXT, EZR, P, AND T SHOULD BE DIMENSIONED .GE. N2.         CUPLNF
C     DATA SHOULD BE LARGE ENOUGH TO CONTAIN ALL OF THE INPUT FAR-FIELD   CUPLNF
C     DATA FOR EITHER ANTENNA IE. .GE. 2*NROWT*NCOLT OR 2*NROWR*NCOLR.    CUPLNF
C                                                                         CUPLNF
C     RHI,THETA,PSI ARE THE EULERIAN ANGLES OF THE REORIENTED             CUPLNF
C     TRANSMITTING AXES WITH RESPECT TO THE AXES FIXED IN THE             CUPLNF
C     TRANSMITTING ANTENNA.                                               CUPLNF
C     PHIR,THETAR,PSIR ARE THE EULERIAN ANGLES OF THE REORIENTED          CUPLNF
C     RECEIVING AXES WITH RESPECT TO THE AXES FIXED IN THE                CUPLNF
C     RECEIVING ANTENNA.                                                  CUPLNF
C     (XO,YO,ZO) ARE THE COORDINATES OF THE ORIGIN OF THE RECEIVING       CUPLNF
C     ANTENNA IN THE REORIENTED RECTANGULAR SYSTEM OF THE TRANSMITTING    CUPLNF
C     ANTENNA.                                                            CUPLNF
C     THE REORIENTED COORDINATE SYSTEMS OF EACH ANTENNA ARE THE COMMON    CUPLNF
C     MUTUAL COUPLING COORDINATE SYSTEMS OF THE ANTENNAS.                 CUPLNF
C     THE COORDINATE SYSTEM FIXED IN EACH ANTENNA IS THE ≠PREFERRED≠      CUPLNF
C     SYSTEM IN WHICH THE FAR-FIELDS OF EACH ANTENNA ARE GIVEN.           CUPLNF
C     ZO MUST BE SPECIFIED,BUT THE RANGE OF XO AND YO ARE DETERMINED      CUPLNF
C     IMPLICITLY BY THE REQUIREMENTS OF THE ALGORITHM FOURT.              CUPLNF
C     RADT=RADIUS OF SMALLEST SPHERE WHICH CIRCUMSCRIBES THE              CUPLNF
C     TRANSMITTING ANTENNA FROM ITS ORIGIN.                               CUPLNF
C     RADR=RADIUS OF SMALLEST SPHERE WHICH CIRCUMSCRIBES THE              CUPLNF
C     RECEIVING ANTENNA FROM ITS ORIGIN.                                  CUPLNF
C     DIAMT = TWICE THE LARGER OF RADT OR WAVLGTH                         CUPLNF
C     DIAMR = TWICE THE LARGER OF RADR OR WAVLGTH                         CUPLNF
C     BFAC ADJUSTS THE INTEGRATION INCREMENTS, AND SHOULD BE              CUPLNF
C     APPROXIMATELY 1 OR 2.  MAKING BFAC LARGER TESTS WHETHER             CUPLNF
C     CONVERGENCE HAS BEEN REACHED.                                       CUPLNF
C     XLIM ADJUSTS THE NONZERO-FILL PORTION OF THE INTEGRATION RANGE,     CUPLNF
C     AND SHOULD BE APPROX. 1 OR 2.  MAKING XLIM LARGER TESTS WHETHER     CUPLNF
C     A WIDE ENOUGH INTEGRATION RANGE HAS BEEN INCLUDED.  INCREASING      CUPLNF
C     XLIM ALSO DECREASES THE INTEGRATION INCREMENTS PROPORTIONATELY      CUPLNF
C     TO PREVENT ALIASING.                                                CUPLNF
C                                                                         CUPLNF
C     A1,A2,B1,B2 DEFINE THE TOTAL(WITH ZERO-FILL)INTEGRATION RANGES      CUPLNF
C     (KX/K FROM -A1 TO APPROX.A2) AND (KY/K FROM -B1 TO APPROX.B2),      CUPLNF
C     IN INCREMENTS OF (A1+A2)/N1 OR (B1+B2)/N2 ARROX. EQUAL TO DKOK.     CUPLNF
C     DKOK = WAVLGTH/(2*(DIAMT + DIAMR)*BFAC*XLIM).                       CUPLNF
C                                                                         CUPLNF
C     IF SQRT((KX/K)**2+(KY/K)**2) IS .GE. XKLIM THE SPECTRUM             CUPLNF
C     IS SET EQUAL TO ZERO. (APRRECIALBLE ZERO FILLING IS AN OPTION       CUPLNF
C     DESIGNED TO ALLOW FINER INCREMENTS DX AND DY AT WHICH THE           CUPLNF
C     COUPLING QUOTIENT IS COMPUTED BY THE FFT.)                          CUPLNF
C     XKLIM MUST BE EQUAL TO OR LESS THAN 1 BECAUSE                       CUPLNF
C     THE PROGRAM NEGLECTS THE EVANESCENT MODES. IN ORDER NOT TO GET      CUPLNF
C     TOO CLOSE TO THE 1/GAMMA SINGULARITY, IT IS SAFER TO CHOOSE XKLIM   CUPLNF
C     NO LARGER THAN XKMAX= ABOUT .9.                                     CUPLNF
C     XLIM ADJUSTS XKLIM.  IF AN ACCURATE COUPLING QUOTIENT IS            CUPLNF
C     REQUIRED ONLY FOR SMALL VALUES OF XO/((DIAMT + DIAMR)*BFAC*XLIM)    CUPLNF
C     AND YO/((DIAMT + DIAMR)*BFAC*XLIM), XLIM NEED NOT BE MORE THAN      CUPLNF
C     1 OR 2.  IF ACCURACY IS DESIRED FOR LARGER XO AND YO AS WELL,       CUPLNF
C     XLIM SHOULD BE MADE CORRESPONDINGLY LARGER.  AS MENTIONED ABOVE,    CUPLNF
C     MAKING XLIM LARGER TESTS WHETHER A WIDE ENOUGH SPECTRUM             CUPLNF
C     HAS BEEN INCLUDED.                                                  CUPLNF
C                                                                         CUPLNF
C     THE XO AND YO INCREMENTS ARE DX=WAVLGTH/(A1+A2) AND                 CUPLNF
C     DY=WAVLGTH/(B1+B2).                                                 CUPLNF
C     THE RANGE OF BOTH XO AND YO IS GIVEN APPROXIMATELY BY               CUPLNF
C     -(DIAMT + DIAMR)*BFAC*XLIM TO +(DIAMT + DIAMR)*BFAC*XLIM, BUT ONLYCUPLNF
C     -(DIAMT + DIAMR) TO +(DIAMT + DIAMR) APPPROXIMATELY IS PRINTED AND CUPLNF
```

```
C        PLOTTED (WHEN XLIM*BFAC IS GREATER THAN OR EQUAL TO 1).
C        CEE IS THE SPEED OF LIGHT IN GIGAMETERS PER SECOND.
C        EMM IS THE MISMATCH FACTOR FOR THE RECEIVING ANTENNA.
C
C
C-       SPECIFICATION STATEMENTS
C
         COMPLEX AX(1000), AY(1000), T(1000), R(1000), WORK(1000)
         COMPLEX FYT(1000), FYR(1000), FZT(1000), FZR(1000)
         COMPLEX EMM, COEF, GAMT, GAMR, GAML
         COMPLEX FXT, FXR
         COMPLEX FX,FY,FZ
         COMPLEX FXP,FYP,FZP
         COMPLEX EDOTEP
         COMPLEX TSUM21,SUM2
C
         DIMENSION NN1(1), NN2(1), HEAD(8), ID(4)
         DIMENSION ACLCUT(1010), X(1010), DATA(8192)
C
         INTEGER HEAD
C
         EQUIVALENCE (T,FZT), (R,FZR), (ACLCUT,FYT), (WORK,FYR)
         EQUIVALENCE (AX, DATA(1)), (AY, DATA(2500)), (X, T)
C
         COMMON /FAR/ N1, N2, NX, NY, DLX, DLY, XK, ISPECT
C
C
C-       DEFINITION AND READING OF INPUT DATA
C
C-      ID AND HEAD ARE ALPHANUMERIC IDENTIFIERS
C-      ID IS PROGRAMMERS NAME AND PHONE EXTENSION
C-      HEAD IS THE IDENTIFIER FOR THE CASE UNDER STUDY
C
C-      FREQ    = FREQUENCY OF OPERATION IN GHZ.
C-      Z0      = SEPARATION BETWEEN ANTENNA REFERENCE POINTS (SEE COMMENTS
C                 ABOVE)
C-      DLXT    = X-INCREMENT WHICH CORRESPONDS TO KX INCREMENT  XMIT
C-      DLYT    = Y-INCREMENT WHICH CORRESPONDS TO KY INCREMENT  XMIT
C-      DLXR    = X-INCREMENT WHICH CORRESPONDS TO KX INCREMENT  RECV
C-      DLYR    = Y-INCREMENT WHICH CORRESPONDS TO KY INCREMENT  RECV
C-      ETOER   = RATIO OF CHARACTERISTIC ADMITTANCE OF TRANSMITTING
C                 ANTENNA FEED MODE TO CHARACTERISTIC ADMITTANCE OF THE
C                 RECEIVING ANTENNA FEED MODE
C-      ISPECT = SPECTRUM FLAG - SET EQUAL TO 1 IF INPUT DATA IS SPECTRUM
C-               RATHER THAN FAR FIELD
C
C-      RCUT    = MAXIMUM ORDINATE VALUE FOR PLOTS.  IF RCUT .EQ. 0, PLOT
C-                IS SELF-SCALED
C-      NBF1,NBF2 = BEGIN AND END INDEX FOR RANGE OF BFAC
C-      NXL1,NXL2 = BEGIN AND END INDEX FOR RANGE OF XLIM
C
C-      GAINT   = GAIN OF XMIT ANTENNA IN DB.
C-      FFTMX   = MAGNITUDE OF FAR-FIELD PATTERN (UNNORMALIZED) AT
C-                THETA = 0,   XMIT
C-      RADT    = RADIUS OF TRANSMIT ANTENNA (SEE COMMENTS ABOVE)
C-      PHI, THETA, PSI = EULER ANGLES IN DEGREES FOR XMIT ANTENNA (SEE
C-                COMMENTS ABOVE)
C-      NROWT   = NUMBER OF ROWS OF DATA IN TRANSMIT PATTERN
C-      NCOLT   = NUMBER OF COLUMNS OF DATA IN TRANSMIT PATTERN
C-      IDAYHRT = FILE IDENTIFIER FOR XMIT DATA
C

C-      GAINR   = GAIN OF RECV ANTENNA IN DB.
C-      FFRMX   = MAGNITUDE OF FAR-FIELD PATTERN (UNNORMALIZED) AT
C-                THETA = 0,   RECV
C-      RADR    = RADIUS OF RECEIVE ANTENNA (SEE COMMENTS ABOVE)
C-      PHIP, THETAP, PSIP = EULER ANGLES IN DEGREES FOR RECV ANTENNA (SEE
C-                COMMENTS ABOVE)
C-      NROWR   = NUMBER OF ROWS OF DATA IN RECEIVE PATTERN
C-      NCOLR   = NUMBER OF COLUMNS OF DATA IN RECEIVE PATTERN
C-      IDAYHRR = FILE IDENTIFIER FOR XMIT DATA
C
C-      GAMT    = REFLECTION COEFFICIENT OF TRANSMITTING ANTENNA
C-      GAMR    = REFLECTION COEFFICIENT OF RECEIVING ANTENNA
C-      GAML    = REFLECTION COEFFICIENT OF RECEIVING LOAD
C
         PRINT 5005
         READ 5000, (ID(I), I = 1, 4)
```

```
      PRINT 5001, (ID(I), I = 1, 4)
   20 READ 5000, HEAD
      IF (EOF(60))  1001,21
   21 PRINT 5001, HEAD
C
      READ 5022, FREQ, ZO, DLXT, DLYT, DLXR, DLYR, ETOFR, ISPECT
      PRINT 5023, FREQ, ZO, DLXT, DLYT, DLXR, DLYR, ETOFR, ISPECT
      READ 5030, RCUT, NBF1, NBF2, NXL1, NXL2
      PRINT 5031, RCUT, NBF1, NBF2, NXL1, NXL2
      READ 5010, GAINT, FFTMX, RADT, PHI, THETA, PSI, NROWT, NCOLT,
     1 IDAYHRT
      PRINT 5011, GAINT, FFTMX, RADT, PHI, THETA, PSI, NROWT, NCOLT,
     1 IDAYHRT
      READ 5010, GAINR, FFRMX, RADR, PHIR, THETAR, PSIR, NROWR, NCOLR,
     1 IDAYHRR
      PRINT 5011, GAINR, FFRMX, RADR, PHIR, THETAR, PSIR, NROWR, NCOLR,
     1 IDAYHRR
      READ 5020, GAMT, GAMR, GAML
      PRINT 5021, GAMT, GAMR, GAML
C
C
      PI=4.*ATAN(1.0)
      CEE = .2997925
      DTR = PI/180.
      WAVLGTH=CEE/FREQ
      XK=2.*PI/WAVLGTH
      EMM = 1./(1. - GAMR*GAML)
      PHIR = PHIR*DTR
      PHI = PHI*DTR
      THETAR = THETAR*DTR
      THETA = THETA*DTR
      PSIR = PSIR*DTR
      PSI = PSI*DTR
      GAINT = 10.**(GAINT/10.)
      GAINR = 10.**(GAINR/10.)
      DIAMT = 2.*AMAX1(WAVLGTH, RADT)
      DIAMR = 2.*AMAX1(WAVLGTH, RADR)
      DIAMSUM = DIAMT + DIAMR
      PRINT 7, DIAMSUM*DIAMSUM/WAVLGTH
C
      ISCL = 5
      IF(RCUT .EQ. 0.)  ISCL = 1
      IF(NBF1 .EQ. 0)  NBF1 = 1
      IF(NBF2 .EQ. 0)  NBF2 = 1
      IF(NXL1 .EQ. 0)  NXL1 = 1
      IF(NXL2 .EQ. 0)  NXL2 = 1
      IF(FFTMX .EQ. 0.)  FFTMX = 1.
      IF(FFRMX .EQ. 0.)  FFRMX = 1.
C
C
C-    LIMITS OF INTEGRATION AND NUMBER OF INTEGRATION POINTS
C
      DO 1000 IXLIM = NXL1, NXL2
      XLIM = IXLIM
      DO 1000 IBFAC = NBF1, NBF2
      BFAC=IBFAC
      XKMAX=.9
      XKMIN = DIAMSUM/ZO
      XKLIM=XLIM*XKMIN
      XKLIM=AMIN1(XKLIM,XKMAX)
C     XKLIM DEFINES THE NONZERO-FILL LIMITS OF INTEGRATION.


      ABL = XKLIM
      A1=ABL  $A2=ABL  $B1=ABL  $B2=ABL
C     FROM XKLIM TO ABL THERE IS ZERO FILLING. NEXT WE COMPUTE
C-    NUMBER OF INTEGRATION POINTS.
      DKOK = WAVLGTH/2./DIAMSUM
      DKOK = AMIN1(DKOK, DIAMSUM/ZO/2.)
      DKOK = DKOK/(BFAC*XLIM)
      NN1(1)=(A1+A2)/DKOK
      NN1(1)=2*((NN1(1)+1)/2)
      NN2(1)=(B1+B2)/DKOK
      NN2(1)=2*((NN2(1)+1)/2)
      N1=NN1(1)
      N2=NN2(1)
C
      C1=(A1+A2)/N1
      C2=(B1+B2)/N2
```

```
      COEF = -C1*C2*EMM*SQRT(GAINT*GAINR)/(4.*PI*EFTMX*EFRMX)
      RG = SQRT(1. - CABS(GAMP)**2)
      TG = SQRT(1. - CABS(GAMT)**2)
      COEF = COEF*RG*TG*SQRT(ETOER)
C
C
C-    FILLING OF THE INPUT MATRICES (AX AND AY) TO THE FFT FOURT.
C
      TSUM21=(0.,0.)
      NY = 4*N2
      NX = 4*N1
      DO 10 J2 = 1, NY
      AX(J2) = (0.,0.)
      AY(J2)=(0.,0.)
   10 CONTINUE
      PRINT 15, N1, N2
      DO 200 J1 = 1, N1
        XNX = C1*(J1 - 1.)
        XKXOK = XNX - A1
        DO 100 J2 = 1, N2
          XNY = C2*(J2 - 1.)
          XKYOK = XNY - B1
          TKOKSQ = XKXOK*XKXOK + XKYOK*XKYOK
          IF (TKOKSQ .GE. XKLIM*XKLIM)  GO TO 110
          CALL ANGLGEN(XKXOK, XKYOK, PHI, THETA, PSI, PHIP, THETAP,
     1     PSIP, PHIT, THETAT, PHIR, THETAR)
          T(J2) = CMPLX(PHIT, THETAT)
          R(J2) = CMPLX(PHIR, THETAR)
          GO TO 100
  110     T(J2) = R(J2) = (0., -1.)
  100     CONTINUE
        WRITE (3) (T(L), L = 1, N2)
        WRITE (4) (R(L), L = 1, N2)
  200   CONTINUE
      NRX2T = NROWT*2
      NRX2R = NROWR*2
      DLX = DLXT       $    DLY = DLYT
      CALL FINDEF (IDAYHRT, 1, 3, 5, 7, DATA, NRX2T, NCOLT, EYT, EZT,
     1 WORK)
      DLX = DLXR       $    DLY = DLYR
      CALL FINDEF (IDAYHRR, 1, 4, 6, 8, DATA, NRX2R, NCOLR, EYR, EZR,
     1 WORK)
C
      DO 220 J1 = 1, N1
        XNX = C1*(J1 - 1.)
        XKXOK = XNX - A1
        SUM2 = (0., 0.)
        READ (5) (EYT(L), L= 1, N2)
        READ (7) (EZT(L), L= 1, N2)
        READ (6) (EYR(L), L= 1, N2)
        READ (8) (EZR(L), L= 1, N2)
        EXT = (0., 0.)
        EXR = (0., 0.)
        DO 240 J2 = 1, N2
          XNY = C2*(J2 - 1.)
          XKYOK = XNY - B1
          TKOKSQ = XKXOK*XKXOK + XKYOK*XKYOK
          IF (TKOKSQ .GE. XKLIM*XKLIM)  GO TO 240
          XNZ = SQRT(1. - TKOKSQ)
          CALL VECTGEN(EXT, EYT(J2), EZT(J2), PHI, THETA, PSI, EX, EY,
     1     EZ)


          CALL VECTGEN(EXR, EYR(J2), EZR(J2), PHIP, THETAR, PSIP, EXP,
     1     EYP, EZP)
          EDOTEP = -EX*EXP + EY*EYP - EZ*EZP
          EDOTEP = EDOTEP*CEXP(CMPLX(0., XK*XNZ*Z0))/XNZ
          SUM2 = EDOTEP + SUM2
          AY(J2 + 3*N2/2) = EDOTEP + AY(J2 + 3*N2/2)
  240     CONTINUE
        AX(J1 + 3*N1/2) = -SUM2*(-1)**(J1 + 3*N1/2)
        TSUM21 = SUM2 + TSUM21
  220   CONTINUE
C
      DO 12 J2 = 1, NY
      AY(J2)=-AY(J2)*(-1)**J2
   12 CONTINUE
C
      NN1(1) = NX
```

```
      NN2(1) = NY                                                       CUPLNF
C                                                                       CUPLNF
      CALL FOURT(AX,NN1,1,+1,+1,WORK)                                   CUPLNF
      CALL FOURT(AY,NN2,1,+1,+1,WORK)                                   CUPLNF
      DO 400 M1 = 1, NX                                                 CUPLNF
      X0 = (-NX/2. + M1 - 1.)*WAVLGTH/(A1 + A2)                         CUPLNF
      FX=CEXP(CMPLX(0.,-XK*A1*X0))                                      CUPLNF
      AX(M1)=FX*COFF*AX(M1)                                             CUPLNF
  400 CONTINUE                                                          CUPLNF
      DO 300 M2 = 1, NY                                                 CUPLNF
      Y0 = (-NY/2. + M2 - 1.)*WAVLGTH/(B1 + B2)                         CUPLNF
      FY=CEXP(CMPLX(0.,-YK*B1*Y0))                                      CUPLNF
      AY(M2)=FY*COFF*AY(M2)                                             CUPLNF
  300 CONTINUE                                                          CUPLNF
C                                                                       CUPLNF
C                                                                       CUPLNF
C-    PRINTOUT AND PLOTTING.                                            CUPLNF
C                                                                       CUPLNF
      PRINT 5, XLIM, BFAC                                               CUPLNF
      PRINT 15, NX, NY                                                  CUPLNF
      PRINT 35,WAVLGTH,RADT,RADR,Z0                                     CUPLNF
      DPHI=PHI*180./PI  $DTHETA=THETA*180./PI  $DPSI=PSI*180./PI        CUPLNF
      DPHIP=PHIP*180./PI  $DTHETAP=THETAP*180./PI  $DPSIP=PSIP*180./PI  CUPLNF
      PRINT 45,DPHI,DTHETA,DPSI,    DPHIP,DTHETAP,DPSIP                 CUPLNF
      DX = WAVLGTH/(A1 + A2)/4.    $  DY = WAVLGTH/(B1 + B2)/4.         CUPLNF
      PRINT 55, -DX*NX/2., DX*(NX/2. - 1.), DX                          CUPLNF
      PRINT 65, -DY*NY/2., DY*(NY/2. - 1.), DY                          CUPLNF
      PRINT 75, -A1*4, A2*4 - C1, C1                                    CUPLNF
      PRINT 85, -B1*4, B2*4 - C2, C2                                    CUPLNF
      PRINT 87,XKLIM                                                    CUPLNF
      CUPLDB = 20.*ALOG10(CABS(TSUM21*COFF))                            CUPLNF
      PRINT 95, TSUM21*COFF, CUPLDB                                     CUPLNF
C                                                                       CUPLNF
C     PRINTOUT OF X0 AND Y0 CENTERLINE CUTS RESPECTIVELY                CUPLNF
C                                                                       CUPLNF
      PRINT 27                                                          CUPLNF
      PRINT 25, (AX(J1), J1 = 1, NX)                                    CUPLNF
      PRINT 29                                                          CUPLNF
      PRINT 25, (AY(J2), J2 = 1, NY)                                    CUPLNF
C                                                                       CUPLNF
C     PLOT OF MAGNITUDE OF X0 AND Y0 CENTERLINE CUTS                    CUPLNF
C                                                                       CUPLNF
      PRINT 510                                                         CUPLNF
C                                                                       CUPLNF
      N10 = NX/(XLIM*BFAC) + .000001                                    CUPLNF
      N1MIN = NX/2 + 1 - N10/2.                                         CUPLNF
      N1MAX = NX/2 + 1 + N10/2.                                         CUPLNF
      DO 501 J1 = N1MIN, N1MAX                                          CUPLNF
        ACLCUT(J1 - N1MIN + 1) = CABS(AX(J1))                           CUPLNF
        X(J1 - N1MIN + 1) = (-NX/2. + J1 - 1.)*WAVLGTH/(A1 + A2)/4.     CUPLNF
        PRINT 515, ACLCUT(J1 - N1MIN + 1.), X(J1 - N1MIN + 1)           CUPLNF
  501   CONTINUE                                                        CUPLNF
      N11 = N1MAX - N1MIN + 1                                           CUPLNF
C                                                                       CUPLNF
      XMIN = X(1)      $     XMAX = X(N11)                              CUPLNF
      IF (ISCL .NE. 1)  GO TO 511                                       CUPLNF
      CALL MINMAX(ACLCUT, XDUM, RCUT, N11, 1)                           CUPLNF
  511 CALL PLT120R(X, ACLCUT, XMAX, XMIN, RCUT, 0., N11, 1H*, 1, 1)     CUPLNF
      PRINT 5041, HEAD, 10HMAGNITUDE  , 10H VS   X0                     CUPLNF
      PRINT 610                                                         CUPLNF
      N20 = NY/(XLIM*BFAC) + .000001                                    CUPLNF
      N2MIN = NY/2 + 1 - N20/2.                                         CUPLNF
      N2MAX = NY/2 + 1 + N20/2.                                         CUPLNF
      DO 601 J2 = N2MIN, N2MAX                                          CUPLNF
        ACLCUT(J2 - N2MIN + 1) = CABS(AY(J2))                           CUPLNF
        X(J2 - N2MIN + 1) = (-NY/2. + J2 - 1.)*WAVLGTH/(B1 + B2)/4.     CUPLNF
        PRINT 615, ACLCUT(J2 - N2MIN + 1.), X(J2 - N2MIN + 1)           CUPLNF
  601   CONTINUE                                                        CUPLNF
      N22 = N2MAX - N2MIN + 1                                           CUPLNF
C                                                                       CUPLNF
      XMIN = X(1)      $     XMAX = X(N22)                              CUPLNF
      IF (ISCL .NE. 1) GO TO 611                                        CUPLNF
      CALL MINMAX(ACLCUT, XDUM, RCUT, N22, 1)                           CUPLNF
  611 CALL PLT120R(X, ACLCUT, XMAX, XMIN, RCUT, 0., N22, 1H*, 1, 1)     CUPLNF
      PRINT 5041, HEAD, 10HMAGNITUDE  , 10H VS   Y0                     CUPLNF
C                                                                       CUPLNF
C                                                                       CUPLNF
```

```
C-     REWIND FILES                                                    CUPLNF
C                                                                      CUPLNF
       REWIND 1                                                        CUPLNF
       REWIND 3                                                        CUPLNF
       REWIND 4                                                        CUPLNF
       REWIND 5                                                        CUPLNF
       REWIND 6                                                        CUPLNF
       REWIND 7                                                        CUPLNF
       REWIND 8                                                        CUPLNF
 1000 CONTINUE                                                         CUPLNF
       GO TO 20                                                        CUPLNF
 1001 CALL EXIT                                                        CUPLNF
C                                                                      CUPLNF
C                                                                      CUPLNF
C-     FORMATS                                                         CUPLNF
C                                                                      CUPLNF
    5 FORMAT (* XLIM = *, F12.5, 5X, *REAC = *, F12.5//)               CUPLNF
    7 FORMAT (*1MUTUAL RAYLEIGH DISTANCE = (DIAMSUM)SQUARED/WAVLGTH=  *,CUPLNF
     1 F12.5, * METERS*//)                                             CUPLNF
   15 FORMAT(1X,*N1=* I6,5X,*N2=*  I6,5X,*THEY BOTH SHOULD BE EVEN*//)  CUPLNF
   25 FORMAT(1X,(10E12.5))                                             CUPLNF
   27 FORMAT(1X,*XO-CUT*/)                                             CUPLNF
   29 FORMAT(1X,//* YO-CUT*/)                                          CUPLNF
   35 FORMAT(1X,*WAVLGTH,RADT,RADR,AND ZO =*4F12.5*  METERS RESPECTIVELYCUPLNF
     1*//)                                                             CUPLNF
   45  FORMAT(1X,*EULER ANGLES(PH,TH,RS) OF TR. AND RE. ANTS. RESP. ARE*CUPLNF
     1   3F10.4*  AND*3F10.4*  DEGREES*//)                              CUPLNF
   55 FORMAT(1X,*XO RANGES FROM*F12.5*  TO*F12.5*  IN INCREMENTS OF*F12.CUPLNF
     15*  METERS*//)                                                   CUPLNF
   65 FORMAT(1X,*YO RANGES FROM*F12.5*  TO*F12.5*  IN INCREMENTS OF*F12.CUPLNF
     15*  METERS*//)                                                   CUPLNF
   75 FORMAT(1X,*THE INTEGRATION VARIABLE KX/K RANGES FROM*F12.5*  TO*F CUPLNF
     1 12.5*  IN INCREMENTS OF*F12.5//)                                CUPLNF
   85 FORMAT(1X,*THE INTEGRATION VARIABLE KY/K RANGES FROM*F12.5*  TO*F CUPLNF
     1 12.5*  IN INCREMENTS OF*F12.5//)                                CUPLNF
   87 FORMAT(1X,*THE SPECTRUM IS ZERO FILLED BEYOND SQRT(KX2+KY2)=K TIMECUPLNF
     1S*F12.5//)                                                       CUPLNF
   95 FORMAT(1X,//* THE COUPLING QUOTIENT AT XO=0 AND YO=0,SUMMED DIRECTCUPLNF
     1LY WITHOUT THE FFT, EQUALS*, 2E12.5, * OR *, F10.2, * DB*//)      CUPLNF
  510 FORMAT(1X,//* MAGNITUDE (XO-CUT)*/)                              CUPLNF
  515 FORMAT(1X,F12.5*  XO=*F12.5)                                     CUPLNF
  610 FORMAT(1X,//* MAGNITUDE (YO-CUT)*/)                              CUPLNF
  615 FORMAT(1X,F12.5*  YO=*F12.5)                                     CUPLNF
 5000 FORMAT (8A10)                                                    CUPLNF
 5001 FORMAT (1H0, 8A10)                                               CUPLNF
 5005 FORMAT (1H1)                                                     CUPLNF
 5010 FORMAT (6F10.4, 2I5, A10)                                        CUPLNF
 5011 FORMAT (1X, 6F10.4, 2I5, A10)                                    CUPLNF
 5020 FORMAT (8F10.4)                                                  CUPLNF
 5021 FORMAT (1X, 8F10.4)                                              CUPLNF
 5022 FORMAT (7F10.4, I1)                                              CUPLNF
 5023 FORMAT (1X, 7F10.4, I5)                                          CUPLNF
 5030 FORMAT (F10.4, 4I5)                                              CUPLNF
 5031 FORMAT (1X, F10.4, 4I5)                                          CUPLNF
 5041 FORMAT (/,5X,8A10,5X, 2A10)                                      CUPLNF
       END                                                             CUPLNF
```

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 (γ/k).

PI = π = 3.14159....

RD,RN	= Denominator and numerator respectively of eq (13b) when computing ϕ_p.
R1,R2,R3,R4, R41,R42,R5, R51,R52,R6, R7,R71,R72, R8,R81,R82, R9	= Intermediate variables used in computing ϕ_p and θ_p from eqs (13).
SNPH,SNPHP, SNPS,SNPSP, SNTH,SNTHP	= Sine of PHI, PHIP, PSI, PSIP, THETA, and THETAP, respectively.
TD,TN	= Denominator and numerator respectively of eq (13b) when computing ϕ_A.
T1,T2,T3,T4, T41,T42,T5, T51,T52,T6, T7,T71,T72, T8,T81,T82, T9	= Intermediate variables used in computing ϕ_A and θ_A from eqs (13).
XCOSR,XCOST	= Right side of eq (13a) for θ_p and θ_A, respectively.

Functions Inline or within FORTRAN Library (in alphabetical order)

ABS(X)	= Absolute value of X.
ACOS(X)	= Angle between 0 and π whose cosine is X.
ATAN(X)	= Angle between $-\pi/2$ and $\pi/2$ whose tangent is X.
ATAN2(X,Y)	= Angle between $-\pi$ and π 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)

```
      SUBROUTINE ANGLGEN(PKXOXK,PKYOXK,PHI,THETA,PSI,PHIP,THETAP,PSIP,  ANGLGEN
     1    PHIT,THETAT,PHIR,THETAR)                                      ANGLGEN
C                                                                       ANGLGEN
C     PKXOXK AND PKYOXK ARE THE X AND Y COMPONENTS OF THE NORMALIZED    ANGLGEN
C     PROPAGATION VECTOR.                                               ANGLGEN
C     PHI,THETA,AND PSI ARE THE EULERIAN ANGLES OF THE ROTATED SYSTEM   ANGLGEN
C     OF THE LEFT,TRANSMITTING ANTENNA  T  WITH RESPECT TO THE AXES     ANGLGEN
C     FIXED IN THE TRANSMITTING ANTENNA.                                ANGLGEN
C     PHIP,THETAP,AND PSIP ARE THE EULERIAN ANGLES OF THE ROTATED       ANGLGEN
C     SYSTEM OF THE RIGHT,RECEIVING ANTENNA R WITH RESPECT TO THE       ANGLGEN
C     AXES FIXED IN THE RECEIVING ANTENNA.                              ANGLGEN
C     THETAT AND PHIT ARE THE ANGLES IN THE FIXED COORDINATE SYSTEM OF  ANGLGEN
C     T CORRESPONDING TO THE DIRECTION PKXOXK,PKYOXK.                   ANGLGEN
C     THETAR AND PHIR ARE THE ANGLES IN THE FIXED COORDINATE SYSTEM OF  ANGLGEN
C     R CORRESPONDING TO THE DIRECTION PKXOXK,PKYOXK.                   ANGLGEN
C     THETAT AND THETAR RANGE FROM FROM 0 TO PI.                        ANGLGEN
C     PHIT AND PHIR RANGE FROM 0 TO 2PI.                                ANGLGEN
C                                                                       ANGLGEN
C                                                                       ANGLGEN
      PI=4.*ATAN(1.0)                                                   ANGLGEN
      GAMOXK=SQRT(1.-PKXOXK**2-PKYOXK**2)                               ANGLGEN
C                                                                       ANGLGEN
      CSTH = COS(THETA)                                                 ANGLGEN
      SNTH = SIN(THETA)                                                 ANGLGEN
      CSTHP = COS(THETAP)                                               ANGLGEN
      SNTHP = SIN(THETAP)                                               ANGLGEN
      CSPS = COS(PSI)                                                   ANGLGEN
      SNPS = SIN(PSI)                                                   ANGLGEN
      CSPSP = COS(PSIP)                                                 ANGLGEN
      SNPSP = SIN(PSIP)                                                 ANGLGEN
      CSPH = COS(PHI)                                                   ANGLGEN
      SNPH = SIN(PHI)                                                   ANGLGEN
      CSPHP = COS(PHIP)                                                 ANGLGEN
      SNPHP = SIN(PHIP)                                                 ANGLGEN
C                                                                       ANGLGEN
C     COMPUTATION OF THETAT AND THETAP.                                 ANGLGEN
C                                                                       ANGLGEN
      T1 = SNTH*CSPS*PKXOXK                                             ANGLGEN
      R1 = SNTHP*CSPSP*PKXOXK                                           ANGLGEN
      T2 = SNTH*SNPS*PKYOXK                                             ANGLGEN
      R2 = SNTHP*SNPSP*PKYOXK                                           ANGLGEN
      T3 = CSTH*GAMOXK                                                  ANGLGEN
      R3 = CSTHP*GAMOXK                                                 ANGLGEN
      XCOST=-T1+T2+T3                                                   ANGLGEN
      XCOSR=-R1-R2+R3                                                   ANGLGEN
      THETAT=ACOS(XCOST)                                                ANGLGEN
      THETAR=ACOS(XCOSR)                                                ANGLGEN
C                                                                       ANGLGEN
C     COMPUTATION OF PHIT AND PHIR.                                     ANGLGEN
C                                                                       ANGLGEN
      T41 = CSPH*CSTH*CSPS                                              ANGLGEN
      T42 = SNPH*SNPS                                                   ANGLGEN
      T4=(T41-T42)*PKXOXK                                               ANGLGEN
C                                                                       ANGLGEN
      R41 = CSPHP*CSTHP*CSPSP                                           ANGLGEN
      R42 = SNPHP*SNPSP                                                 ANGLGEN
      R4=(R41-R42)*PKXOXK                                               ANGLGEN
C                                                                       ANGLGEN
      T51 = CSPH*CSTH*SNPS                                              ANGLGEN
      T52 = SNPH*CSPS                                                   ANGLGEN
      T5=(T51+T52)*PKYOXK                                               ANGLGEN


C                                                                       ANGLGEN
      R51 = CSPHP*CSTHP*SNPSP                                           ANGLGEN
      R52 = SNPHP*CSPSP                                                 ANGLGEN
      R5=(R51+R52)*PKYOXK                                               ANGLGEN
C                                                                       ANGLGEN
      T6 = CSPH*SNTH*GAMOXK                                             ANGLGEN
      R6 = CSPHP*SNTHP*GAMOXK                                           ANGLGEN
      TD=T4-T5+T6                                                       ANGLGEN
      RD=R4+R5+R6                                                       ANGLGEN
C                                                                       ANGLGEN
C                                                                       ANGLGEN
      T71 = SNPH*CSTH*CSPS                                              ANGLGEN
      T72 = CSPH*SNPS                                                   ANGLGEN
      T7=(T71+T72)*PKXOXK                                               ANGLGEN
C                                                                       ANGLGEN
      R71 = SNPHP*CSTHP*CSPSP                                           ANGLGEN
```

```
      R72 = CSPHP*SNPSP
      R7=(R71+R72)*PKXDXK
C
      T81 = CSPH*CSPS
      T82 = SNPH*CSTH*SNPS
      T8=(T81-T82)*PKYDXK
C
      R81 = CSPHP*CSPSP
      R82 = SNPHP*CSTHP*SNPSP
      R8=(R81-R82)*PKYDXK
C
      T9 = SNPH*SNTH*GAMDXK
      R9 = SNPHP*SNTHP*GAMDXK
      TN=T7+T8+T9
      RN=R7-R8+R9
C
C     CHANGE OF RANGE OF PHIT FROM(-PI,PI) TO (0,2PI).
      IF((ABS(TN)+ABS(TD)).EQ.0.) GO TO 10
      PHIT=ATAN2(TN,TD)
      IF(PHIT.LT.0.) PHIT=2.*PI+PHIT
      GO TO 20
   10 CONTINUE
      PHIT=0.
   20 CONTINUE
C     CHANGE OF RANGE OF PHIR FROM(-PI,PI) TO (0,2PI).
      IF((ABS(RN)+ABS(RD)).EQ. 0.) GO TO 30
      PHIR=ATAN2(RN,RD)
      IF(PHIR.LT.0.) PHIR=2.*PI+PHIR
      GO TO 40
   30 CONTINUE
      PHIR=0.
   40 CONTINUE
      RETURN
      END
```

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 Ø-coordinate and the imaginary part, the θ-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

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

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

where

$k_x = k \sin\theta \cos\emptyset$
$k_y = k \sin\theta \sin\emptyset$

and ΔK_x and Δk_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 δ_x, δ_y are the x or y <u>near-field</u> 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 θ and Ø. The fractional part of

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

is used to determine which corner of the square lies closest to θ and Ø.

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 θ and Ø. 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	= π/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	= Ø = angle in far field.
PI	= π = 3.14159.....
PIX2	= 2π.

TAMP = Intermediate variable, amplitude of input far-field data.

THETA = θ = angle in far field.

XNZ = $\text{Cos}\theta$.

```
      SUBROUTINE FINDFF (IDAYHR, LUIN, LUA, LUDY, LUDZ, DATA, NRX2,
     1 NCOL, FFY, FFZ, STOR)
C
C-  THIS SUBROUTINE READS FAR-FIELD OR SPECTRUM DATA FROM LUIN AND
C-  STORES IT IN ARRAY DATA.  ANGLES CORRESOPNDING TO FAR-FIELD
C-  DIRECTIONS IN THE ANTENNAS COORDINATE SYSTEM ARE READ IN FROM LUA.
C-  DATA IS SEARCHED FOR THE CLOSEST POINT AND THE Y-COMPONENT OF THE
C-  FIELD AT THE GIVEN ANGLE IS USED TO COMPUTE THE Z-COMPONENT.
C-  THESE FIELD COMPONENTS ARE WRITTEN ON LUDY AND LUDZ.
C
      COMPLEX FFY(1), FFZ(1)
      COMMON /FAR/ N1, N2, NX, NY, DLX, DLY, XK, ISPECT
      DIMENSION DATA(NRX2, NCOL), STOR(1), ID(10)
C-
C-  MISCELLANEOUS
C-
      PRINT 1020, LUIN, LUA, LUDY, LUDZ, NRX2, NCOL
 1020 FORMAT (6I20)
      ISE = 0
      PI = 4.*ATAN(1.)
      PIX2 = 2.*PI
      DTR = PI/180.
      NROW = NRX2/2
      DLKY = PIX2/NROW/DLY
      DLKX = PIX2/NCOL/DLX
      FKXMAX = DLKX*(NCOL - 1)/2
      FKYMAX = DLKY*(NROW - 1)/2
      PRINT 1000, DLKX, DLKY, FKXMAX, FKYMAX, XK
 1000 FORMAT(1X, 5G20.5)
C-
C-  FIND CORRECT FAR-FIELD FILE ON LOGICAL UNIT LUIN.
C-
  120 READ(LUIN) (ID(I), I = 1, 10)
      PRINT 1510, ID
      IF (ID(8) .EQ. IDAYHR) GO TO 130
      IF (EOF(LUIN)) 125, 130
  125 PRINT 1530
      CALL EXIT
  130 CONTINUE
C-
C-  READ FAR-FIELD INTO ARRAY DATA.
C-
      DO 140 ICOL = 1, NCOL
        READ(LUIN) (STOR(I), I = 1, NRX2)
        DO 150 IROW = 1, NROW
          IRX2 = IROW*2
          TAMP = STOR(IRX2 - 1)
          ANGLE = STOR(IRX2)*DTR
          DATA(IRX2 - 1, ICOL) = TAMP*COS(ANGLE)
          DATA(IRX2, ICOL) = TAMP*SIN(ANGLE)
  150     CONTINUE
  140   CONTINUE
C-
C-  REPLACE VALUES OF ANGLES WITH CORRESPONDING FAR-FIELD.
C-
      REWIND LUA
      READ (LUIN)
      IF (EOF(LUIN))   500, 600
  600 CONTINUE
      BACKSPACE LUIN
  500 CONTINUE


      DO 200 I = 1, N1
        READ(LUA) (FFY(L), L = 1, N2)
        DO 300 J = 1, N2
          PHI = REAL(FFY(J))
          THETA = AIMAG(FFY(J))
          IF (THETA .LT. 0.) GO TO 310
C-
C-  FIND THE INDICES FOR THE ARRAY DATA WHICH CORRESPOND TO THE
C-  COORDINATES CLOSEST TO THE DESIRED THETA AND PHI VALUES.
C-
          FKX = XK*SIN(THETA)*COS(PHI)
          FKY = XK*SIN(THETA)*SIN(PHI)
          FKSQ = FKX*FKX + FKY*FKY
          IF (FKSQ .GT. XK*XK)  GO TO 310
          IF (ABS(FKX) .GE. FKXMAX)  GO TO 310
          IF (ABS(FKY) .GE. FKYMAX)  GO TO 310
```

```
          FROW = FKY/DLKY + NROW/2
          IFR = FROW
          DROW = FROW - IFR
          IROW = IFR + 1
          IF (DROW .LT. .5)  IROW = IFR
          FCOL = FKX/DLKX + NCOL/2
          IFC = FCOL
          DCOL = FCOL - IFC
          ICOL = IFC + 1
          IF (DCOL .LT. .5)  ICOL = IFC
          IR2X2 = IROW*2
          FFY(J) = CMPLX(DATA(IR2X2 - 1, ICOL), DATA(IR2X2, ICOL))
          FFZ(J) = -TAN(THETA)*SIN(PHI)*FFY(J)
          IF (ISPECT .NE. 1)  GO TO 300
          XNZ = SQRT(XK*XK - FKSQ)/XK
          FFY(J) = FFY(J)*XNZ
          FFZ(J) = FFZ(J)*XNZ
          GO TO 300
  310     FFY(J) = (0., 0.)
          FFZ(J) = (0., 0.)
  300     CONTINUE
        WRITE(LUDY) (FFY(L), L = 1, N2)
        WRITE(LUDZ) (FFZ(L), L = 1, N2)
  200   CONTINUE
      REWIND LUDY
      REWIND LUDZ
      RETURN
 1510 FORMAT(1X, 8A10, 2I5)
 1520 FORMAT(*  FILE *, I5, * SKIPPED ON LU *, I5)
 1530 FORMAT(* FILE NOT FOUND, EXECUTION ABORTED*)
      END
```

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

```
      SUBROUTINE VECTGEN(FOX,FOY,FOZ,PH,THET,PS,  FX,FY,FZ)
C     IF THE COMPONENTS (FOX,FOY,FOZ) OF A COMPLEX VECTOR ARE GIVEN IN
C     A RIGHT-HANDED RECTANGULAR COORDINATE SYSTEM, AND A SECOND
C     COORDINATE SYSTEM IS FORMED BY ROTATION THROUGH EULERIAN ANGLES
C     (PH,THET,PS), THEN (FX,FY,FZ) ARE THE COMPONENTS OF THAT VECTOR
C     IN THIS SECOND ROTATED SYSTEM.
C
      COMPLEX FOX,FOY,FOZ
      COMPLEX FX,FY,FZ
C
C
C     COMPUTATION OF THE NINE ELEMENTS OF THE ROTATIONAL
C     TRANSFORMATION MATRIX.
C
      CSPH = COS(PH)
      SNPH = SIN(PH)
      CSPS = COS(PS)
      SNPS = SIN(PS)
      CSTH = COS(THET)
      SNTH = SIN(THET)
C
      A11 = CSPH*CSTH*CSPS - SNPH*SNPS
      A12 = SNPH*CSTH*CSPS + CSPH*SNPS
      A13 = - SNTH*CSPS
      A21 = -(CSPH*CSTH*SNPS + SNPH*CSPS)
      A22 = -SNPH*CSTH*SNPS + CSPH*CSPS
      A23 = SNTH*SNPS
      A31 = CSPH*SNTH
      A32 = SNPH*SNTH
      A33 = CSTH
C
      FX=A11*FOX+A12*FOY+A13*FOZ
      FY=A21*FOX+A22*FOY+A23*FOZ
      FZ=A31*FOX+A32*FOY+A33*FOZ
      RETURN
      END
```

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.

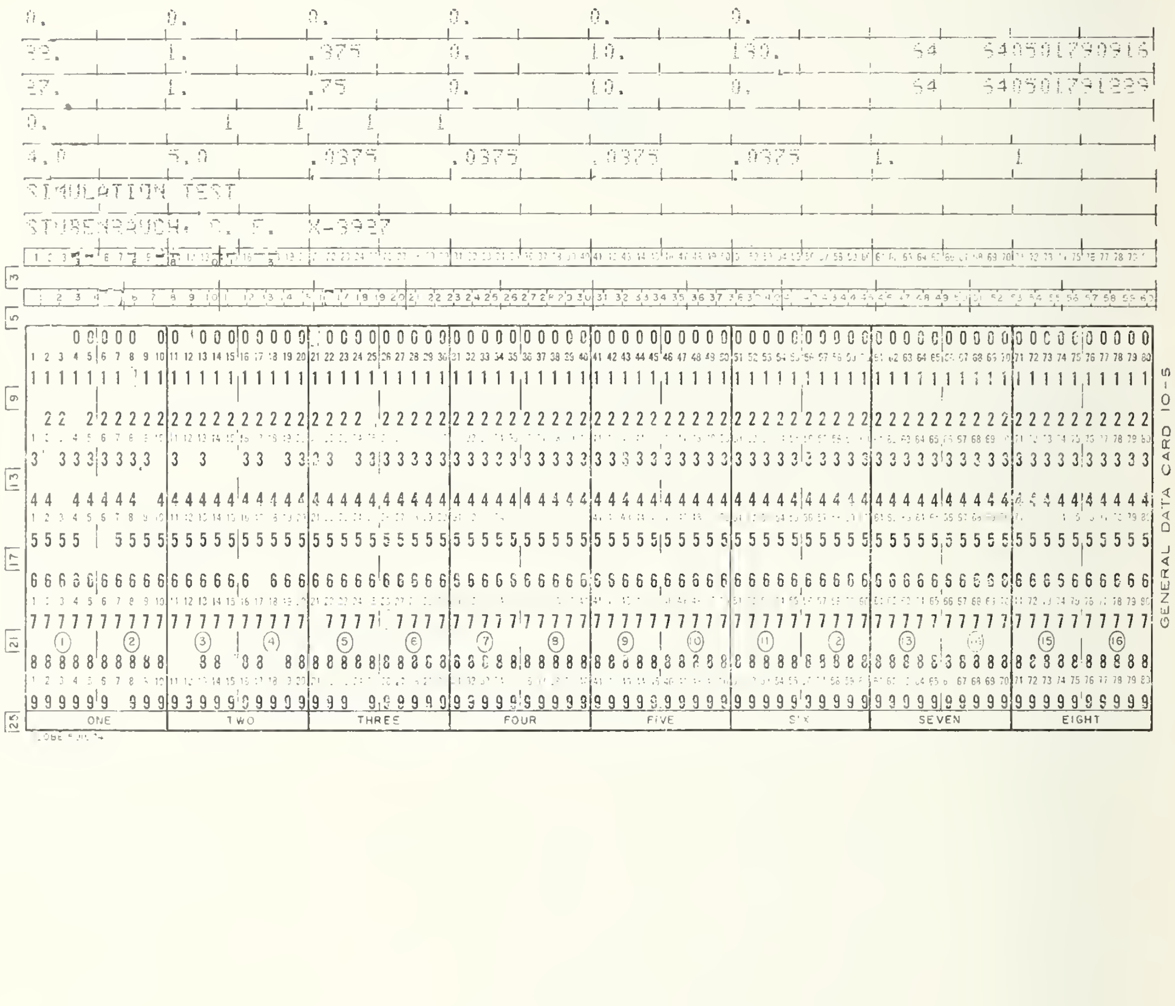
SYMBOL DICTIONARY:

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

```
      SUBROUTINE MINMAX(Z, ZMIN, ZMAX, LEX, LEY)
      DIMENSION Z(LEX,LEY)
      ZMIN=Z(1,1)    $  ZMAX=Z(1,1)
      DO 120 I = 1, LEX
      DO 120 J = 1, LEY
      TZ = Z(I,J)
      IF (TZ .LT. ZMIN) ZMIN = TZ
      IF (TZ .GT. ZMAX) ZMAX = TZ
  120 CONTINUE
      RETURN
      END
```

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.



MUTUAL RAYLEIGH DISTANCE = (DIAMSUM)SQUARED/WAVLGTH=   67.54672 METERS

N1=   64   N2=   64   THEY BOTH SHOULD BE EVEN

1	3	5	7	128	64
2.6180	2.6180	82.467	82.467	83.834	
PHYSICAL OPTICS SIMLSIMULATION TEST NO. 1			0501791229 64 128		
1	4	6	8	128	64
2.6180	2.6180	82.467	82.467	83.834	
PHYSICAL OPTICS SIMLSIMULATION TEST NO. 2			0501790916 64 128		

XLIM =   1.00000   REAC =   1.00000

N1=   216   N2=   216   THEY BOTH SHOULD BE EVEN

WAVLGTH,RADT,RADR,AND ZO =   .07495   .75000   .37500   5.00000 METERS RESPECTIVELY

EULER ANGLES(PH,TH,PS) OF TR. AND RE. ANTS. RESP. ARE   0.0000 10.0000   0.0000 AND   0.0000 10.0000 180.0000 DEGREES

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

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

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

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

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

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

XO-CUT

.11267E-02	-.35681E-04	.15799E-03	-.55538E-04	.22520E-03	-.19027E-05	.30552E-03	.54126E-04	.38612E-03	.89992E-04
.45237E-03	.89579E-04	.49055E-03	.49353E-04	.49091E-03	-.19572E-04	.45034E-03	-.94504E-04	.37405E-03	-.14856E-03
.27540E-03	-.15980E-03	.17333E-03	-.11883E-03	.87734E-04	-.31861E-04	.33983E-04	.81711E-04	.18404E-04	.19626E-03
.36611E-04	.28850E-03	.75539E-04	.34414E-03	.11860E-03	.36029E-03	.15194E-03	.34302E-03	.16915E-03	.30178E-03
.17220E-03	.24370E-03	.16823E-03	.17078E-03	.16351E-03	.81126E-04	.15771E-03	-.26211E-04	.14156E-03	-.14553E-03
.99803E-04	-.26093E-03	.18676E-04	-.34734E-03	-.10484E-03	-.37628E-03	-.25640E-03	-.32490E-03	-.40374E-03	-.18488E-03
-.50379E-03	.31936E-04	-.51538E-03	.29340E-03	-.41362E-03	.55265E-03	-.20045E-03	.75788E-03	.92512E-04	.86352E-03
.41012E-03	.83996E-03	.68695E-03	.67979E-03	.86312E-03	.39942E-03	.89782E-03	.36370E-04	.77680E-03	-.35689E-03
.51255E-03	-.72114E-03	.13881E-03	-.99933E-03	-.29766E-03	-.11449E-02	-.74493E-03	-.11283E-02	-.11513E-02	-.94140E-03
-.14685E-02	-.59314E-03	-.16534E-02	-.13324E-03	-.16710E-02	.40209E-03	-.14987E-02	.94568E-03	-.11318E-02	.14308E-02
-.58851E-03	.17927E-02	.87487E-04	.19750E-02	.82941E-03	.19369E-02	.15544E-02	.16588E-02	.21745E-02	.11484E-02
.26022E-02	.44306E-03	.27967E-02	-.39062E-03	.27031E-02	-.12634E-02	.23257E-02	-.20740E-02	.16938E-02	-.27239E-02
.86427E-03	-.31318E-02	-.84590E-04	-.32459E-02	-.10618E-02	-.30506E-02	-.19729E-02	-.25675E-02	-.27302E-02	-.18494E-02
-.32615E-02	-.97158E-03	-.35178E-02	-.21270E-04	-.34779E-02	.91238E-03	-.31503E-02	.17468E-02	-.25716E-02	.24129E-02
-.18015E-02	.28606E-02	-.91612E-03	.30631E-02	.32635E-06	.30193E-02	.86414E-03	.27527E-02	.16006E-02	.23078E-02
.21515E-02	.17424E-02	.24805E-02	.11178E-02	.25764E-02	.49045E-03	.24529E-02	-.95079E-04	.21450E-02	-.60794E-03
.17029E-02	-.10296E-02	.11831E-02	-.13500E-02	.64070E-03	-.15636E-02	.12246E-03	-.16676E-02	-.33717E-03	-.16630E-02
-.71693E-03	-.15561E-02	-.10073E-02	-.13613E-02	-.12081E-02	-.11010E-02	-.13255E-02	-.80258E-03	-.13700E-02	-.49407E-03

-.13555E-02	-.13897E-03	-.12977E-02	.66950E-04	-.12126E-02	.29595E-03	-.11148E-02	.48599E-03	-.10153E-02	.63727E-03
-.91981E-03	.74941E-03	-.82905E-03	.82083E-03	-.74005E-03	.85007E-03	-.64917E-03	.83826E-03	-.55514E-03	.79069E-03
-.46112E-03	.71647E-03	-.37479E-03	.62599E-03	-.30641E-03	.52750E-03	-.26553E-03	.42447E-03	-.25768E-03	.31539E-03
-.28223E-03	.13607E-03	-.33238E-03	.63631E-04	-.39694E-03	-.79909E-04	-.46336E-03	-.22632E-03	-.52077E-03	-.36278E-03
-.56197E-03	-.47592E-03	-.58390E-03	-.55632E-03	-.58681E-03	-.60148E-03	-.57264E-03	-.61596E-03	-.54352E-03	-.60879E-03
-.50086E-03	-.58947E-03	-.44527E-03	-.56451E-03	-.37730E-03	-.53586E-03	-.29835E-03	-.50177E-03	-.21154E-03	-.45906E-03
-.12221E-03	-.40550E-03	-.37767E-04	-.34092E-03	.32913E-04	-.26673E-03	.80639E-04	-.18448E-03	.97300E-04	-.94664E-04
.77326E-04	.32350E-05	.18894E-04	.10923E-03	-.75443E-04	.22056E-03	-.19894E-03	.33012E-03	-.34164E-03	.42647E-03
-.49195E-03	.43607E-03	-.63824E-03	.52664E-03	-.77012E-03	.51044E-03	-.87885E-03	.44604E-03	-.95715E-03	.33791E-03
-.99863E-03	.19443E-03	-.99776E-03	.25440E-04	-.94992E-03	-.15928E-03	-.85327E-03	-.34990E-03	-.71041E-03	-.53515E-03
-.53000E-03	-.70094E-03	-.32720E-03	-.83023E-03	-.12240E-03	-.90497E-03	.61759E-04	-.90988E-03	.20441E-03	-.83683E-03
.29047E-03	-.68820E-03	.31359E-03	-.47790E-03	.27714E-03	-.22985E-03	.19324E-03	.25646E-04	.80101E-04	.25630E-03
-.41387E-04	.43215E-03	-.15095E-03	.52905E-03	-.23146E-03	.53145E-03	-.27032E-03	.43481E-03	-.26011E-03	.24746E-03
-.19853E-03	-.84688E-05	-.98500E-04	-.29819E-03	.61711E-04	-.57821E-03	.23823E-03	-.80326E-03	.42167E-03	-.93481E-03
.58815E-03	-.94929E-03	.71172E-03	-.84325E-03	.76822E-03	-.63399E-03	.74011E-03	-.35518E-03	.62127E-03	-.49200E-04
.42013E-03	.24164E-03	.16007E-03	.48220E-03	-.12386E-03	.64856E-03	-.39142E-03	.72878E-03	-.60545E-03	.72155E-03
-.73968E-03	.63413E-03	-.78366E-03	.48093E-03	-.74338E-03	.28264E-03	-.63732E-03	.65261E-04	-.48963E-03	-.14190E-03
-.32283E-03	-.39986E-03	-.15295E-03	-.41467E-03	.11851E-04	-.44187E-03	.16733E-03	-.38915E-03	.30911E-03	-.26629E-03
.42799E-03	-.92568E-04	.50923E-03	.10729E-03	.53625E-03	.30683E-03	.49759E-03	.48043E-03	.39439E-03	.60458E-03
.24457E-03	.65886E-03	.81317E-04	.62800E-03	-.54679E-04	.50550E-03	-.12690E-03	.29810E-03	-.11467E-03	.28987E-04
-.20479E-04	-.26274E-03	.13001E-03	-.52788E-03	.29528E-03	-.71775E-03	.43024E-03	-.79698E-03	.49915E-03	-.75412E-03
.48495E-03	-.60619E-03	.39220E-03	-.39474E-03	.24337E-03	-.17446E-03	.70615E-04	.24885E-05	-.93735E-04	.10093E-03
-.22469E-03	.11147E-03	-.30835E-03	.50772E-04	-.34166E-03	-.46015E-04	-.32968E-03	-.13792E-03	-.28206E-03	-.19117E-03
-.21021E-03	-.18857E-03	-.12590E-03	-.13266E-03	-.40505E-04	-.42576E-04	.35393E-04	.53647E-04	.93100E-04	.12841E-03
.12730E-03	.16262E-03	.13750E-03	.15042E-03	.12864E-03	.99865E-04	.11025E-03	.29576E-04	.94332E-04	-.37200E-04
.92407E-04	-.79660E-04								

YO-CUT

-.55417E-05	-.91347E-04	-.99645E-05	.91689E-05	-.65039E-04	-.31226E-04	-.21575E-05	.14608E-03	-.18425E-03	-.36074E-04
.64589E-05	.10786E-03	-.16933E-03	.38430E-04	.28731E-04	.11604E-03	-.32405E-04	.43324E-04	.12897E-03	.30019E-04
.72498E-05	-.71986E-04	.11450E-03	-.26902E-04	-.67694E-04	-.10592E-03	.11281E-03	.23010E-04	-.13987E-03	-.28887E-04
.25991E-04	-.16995E-05	-.12683E-03	.43992E-04	.76938E-05	.92657E-04	-.18536E-04	-.36743E-04	.85875E-04	.72925E-04
-.30596E-04	-.15256E-03	.29576E-03	-.26478E-04	.17083E-04	-.11359E-03	.21338E-03	-.14977E-03	-.36668E-04	-.83343E-04
.68680E-04	.40433E-05	-.20555E-04	-.40696E-04	.84612E-04	.23276E-04	-.87832E-04	-.35491E-04	.10784E-03	.65280E-04
-.78777E-04	.12563E-04	.13590E-03	.17730E-04	-.42391E-04	.23221E-04	.68445E-04	.22950E-04	-.11237E-03	-.50569E-05
-.11511E-04	.36925E-04	-.14452E-03	.19246E-04	.14238E-03	.19149E-03	-.13121E-03	.17237E-03	.17189E-03	.74755E-04
-.39676E-05	.21890E-03	.15757E-03	.29546E-04	.46292E-04	-.85397E-05	.15496E-03	-.17156E-04	-.25954E-04	-.14243E-03
.20251E-03	-.13694E-03	-.13681E-03	-.17223E-03	.72525E-04	-.14224E-03	-.18295E-03	.13280E-04	.21309E-04	.13664E-04
-.21110E-03	.33009E-04	-.25594E-04	.36622E-04	-.23746E-03	.11953E-03	.79469E-04	.15885E-03	-.22046E-03	.12208E-03
.14813E-03	.38592E-04	-.87635E-04	.14864E-03	.12795E-03	-.41967E-04	-.25533E-05	.18897E-04	.15336E-03	.34623E-04
-.42458E-05	-.26114E-04	.31690E-03	.77134E-04	-.51603E-04	.47518E-04	.28414E-03	.11000E-04	-.34267E-04	.16279E-03
.16011E-03	-.76542E-05	-.74518E-04	.43017E-04	.15938E-03	-.61126E-04	-.10316E-03	-.46355E-04	.12971E-03	-.14929E-03
-.21035E-03	-.63392E-04	.12710E-03	-.78972E-04	-.13223E-03	.57073E-04	.10343E-03	-.14139E-03	-.11670E-03	.59725E-05
.93289E-04	-.60484E-04	-.13860E-03	.66902E-04	.19813E-03	.38997E-05	-.36082E-03	.17071E-04	.34811E-03	-.62855E-04
-.13499E-03	.15465E-03	.37697E-03	.23081E-03	.13766E-04	.94200E-04	.35629E-03	-.90594E-04	.24669E-03	-.15374E-03
.82094E-03	-.64132E-03	-.27723E-03	.47084E-04	.72256E-03	-.17838E-03	-.12903E-02	.18413E-02	.53122E-03	-.98987E-03
-.66055E-03	.63972E-03	.17515E-02	-.22653E-02	.46422E-03	-.81491E-03	-.20102E-03	-.39538E-03	-.15070E-03	.29012E-03
-.21177E-04	.24559E-02	-.10222E-02	-.73331E-03	.87670E-03	-.19695E-03	.63099E-03	-.31874E-02	-.40104E-03	-.65749E-03
-.10673E-02	-.17176E-03	.54742E-02	.10462E-02	.25336E-02	.14024E-02	-.13796E-02	-.17785E-02	.47048E-03	.36981E-02
-.30629E-02	.57991E-02	-.48311E-05	-.14398E-02	-.10660E-02	.76036E-03	-.61062E-02	-.25952E-02	-.60525E-03	-.21291E-02
-.85011E-03	.15126E-02	.17793E-02	-.65607E-02	.33684E-02	-.29863E-02	-.19133E-02	.22275E-04	.19121E-02	.12467E-02
.41906E-02	.63733E-02	-.57996E-03	-.11970E-02	-.29307E-03	.14310E-03	-.44031E-02	-.20638E-03	-.91679E-03	.20642E-02
-.12217E-03	-.16375E-02	.31634E-03	.10021E-02	-.24836E-02	.53145E-03	.15680E-03	-.32208E-03	-.26384E-03	-.99569E-04
-.34679E-02	-.53328E-03	-.44547E-04	-.10657E-02	.19453E-03	.48494E-03	-.50307E-03	-.18058E-02	-.45793E-03	-.56559E-03
.45423E-03	-.23572E-03	-.21352E-03	.37318E-04	-.21275E-03	-.67245E-03	.21594E-03	.18423E-03	.21438E-03	-.21942E-03
.80638E-04	.91182E-04	.43918E-05	-.15429E-03	.15776E-03	.22101E-03	-.17331E-03	-.24231E-03	-.18553E-03	.19343E-03
.59812E-04	-.17508E-03	-.30792E-04	.23123E-03	-.67252E-04	-.99733E-04	-.20768E-04	.86085E-04	-.36636E-04	-.19461E-03
.23142E-04	.12435E-03	.19956E-04	-.16724E-03	.79931E-05	.52166E-04	-.56075E-04	-.18208E-03	.35161E-04	.18102E-03
.80988E-04	-.13484E-03	.97902E-04	.15457E-03	.12321E-03	-.51845E-04	.10259E-03	.12126E-03	.11689E-03	-.63392E-04
.20176E-03	.87859E-04	.12320E-03	-.14822E-03	.92154E-04	.12288E-03	.12405E-04	-.71509E-04	.87974E-04	.15518E-03
.37456E-07	-.34843E-04	-.13372E-04	.19636E-03	-.88632E-04	.64901E-05	-.86928E-04	.24665E-03	-.93349E-04	.10684E-04
-.64612E-04	.13347E-03	-.18490E-03	-.86269E-04	-.96767E-04	.10153E-03	-.12631E-03	-.11247E-03	-.51363E-04	-.46130E-05

-.36040E-04	-.22847E-03	.27368E-04	-.14972E-04	.25220E-04	-.13831E-03	.18073E-03	.75231E-04	.12240E-03	-.92773E-04
.15560E-03	.11029E-03	.72594E-04	.29312E-04	.68376E-04	.22752E-03	-.61162E-05	.30006E-04	.24876E-04	.20403E-03
-.93732E-04	-.23535E-04	-.21005E-04	.13492E-03	-.28169E-04	-.39296E-06	.82171E-04	.79824E-04	.21244E-05	-.56212E-04
.58589E-04	.73585E-04	.13537E-05	-.40975E-04	.80506E-04	.10811E-03	-.36230E-04	-.95755E-04	.28121E-04	.22806E-04
-.86103E-04	-.10141E-03	.89385E-04	.21535E-04	.55319E-04	-.14983E-03	.88058E-04	-.28324E-04	.51470E-04	-.15523E-03
.13254E-03	.90830E-04	.27989E-05	-.79177E-04	.14744E-03	.56136E-04	-.78423E-04	-.26494E-04	-.29046E-04	.67557E-04
-.11981E-03	-.29736E-05	-.24035E-04	.61880E-04	-.58896E-04	-.11737E-03	.58553E-04	.19126E-04	-.36339E-04	-.14587E-03
.13796E-03	.43024E-04	.62135E-04	-.61913E-04	.15510E-03	.57726E-04	-.44485E-04	.24097E-05	.40143E-05	.15301E-03
-.85592E-04	.21655E-04	-.25677E-04	.12733E-03	-.15557E-03	-.87549E-04	.45179E-04	.45133E-04	-.78945E-04	-.21741E-04
.11644E-03	-.56983E-04								

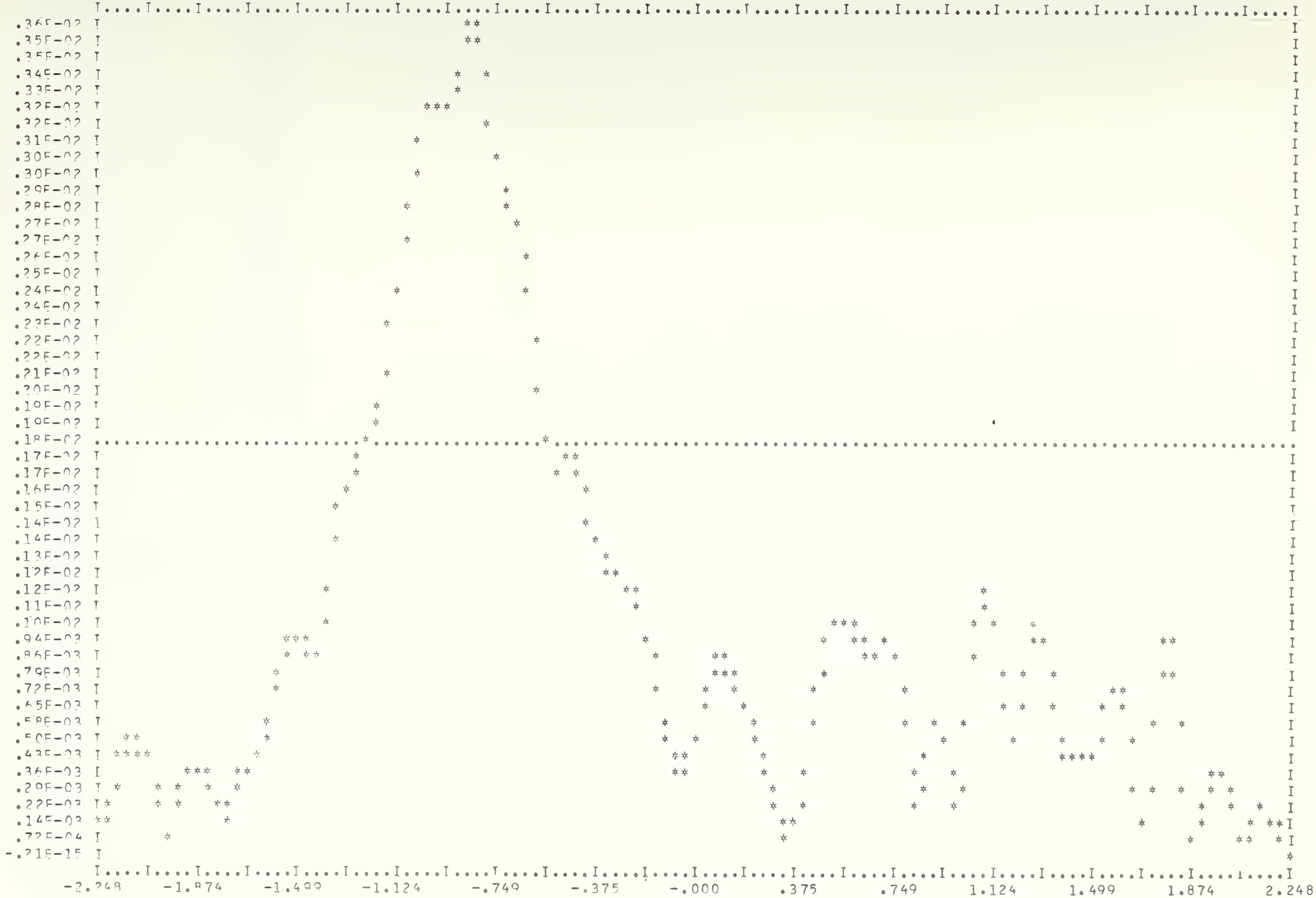
MAGNITUDE (XO-CUT)

.14155E-03	XO=	-2.24844
.16747E-03	XO=	-2.22762
.22520E-03	XO=	-2.20681
.31027E-03	XO=	-2.18599
.39647E-03	XO=	-2.16517
.46115E-03	XO=	-2.14435
.49303E-03	XO=	-2.12353
.49130E-03	XO=	-2.10271
.46015E-03	XO=	-2.08189
.40247E-03	XO=	-2.06107
.31841E-03	XO=	-2.04025
.21015E-03	XO=	-2.01944
.93340E-04	XO=	-1.99862
.88496E-04	XO=	-1.97780
.19712E-03	XO=	-1.95698
.29091E-03	XO=	-1.93616
.35233E-03	XO=	-1.91534
.37931E-03	XO=	-1.89452
.37517E-03	XO=	-1.87370
.34595E-03	XO=	-1.85288
.29841E-03	XO=	-1.83207
.23973E-03	XO=	-1.81125
.18253E-03	XO=	-1.79043
.15987E-03	XO=	-1.76961
.20392E-03	XO=	-1.74879
.27937E-03	XO=	-1.72797
.34784E-03	XO=	-1.70715
.39061E-03	XO=	-1.68633
.41388E-03	XO=	-1.66551
.44406E-03	XO=	-1.64469
.50490E-03	XO=	-1.62388
.59305E-03	XO=	-1.60306
.69029E-03	XO=	-1.58224
.78394E-03	XO=	-1.56142
.86846E-03	XO=	-1.54060
.93473E-03	XO=	-1.51978
.96644E-03	XO=	-1.49896
.95106E-03	XO=	-1.47814
.89856E-03	XO=	-1.45732
.85486E-03	XO=	-1.43651
.88473E-03	XO=	-1.41569
.10089E-02	XO=	-1.39487
.11830E-02	XO=	-1.37405
.13521E-02	XO=	-1.35323
.14872E-02	XO=	-1.33241
.15856E-02	XO=	-1.31159
.16587E-02	XO=	-1.29077
.17187E-02	XO=	-1.26995
.17721E-02	XO=	-1.24914
.18243E-02	XO=	-1.22832

.18868E-02	X0=	-1.20750
.19770E-02	X0=	-1.18668
.21070E-02	X0=	-1.16586
.22733E-02	X0=	-1.14504
.24591E-02	X0=	-1.12422
.26465E-02	X0=	-1.10340
.28239E-02	X0=	-1.08258
.29538E-02	X0=	-1.06177
.31162E-02	X0=	-1.04095
.32076E-02	X0=	-1.02013
.32489E-02	X0=	-.99931
.32470E-02	X0=	-.97849
.32301E-02	X0=	-.95767
.32280E-02	X0=	-.93685
.32976E-02	X0=	-.91603
.34032E-02	X0=	-.89521
.35179E-02	X0=	-.87439
.35956E-02	X0=	-.85358
.36022E-02	X0=	-.83276
.35264E-02	X0=	-.81194
.33896E-02	X0=	-.79112
.31972E-02	X0=	-.77030
.30193E-02	X0=	-.74948
.28852E-02	X0=	-.72866
.28096E-02	X0=	-.70784
.27686E-02	X0=	-.68702
.27208E-02	X0=	-.66621
.26227E-02	X0=	-.64539
.24547E-02	X0=	-.62457
.22295E-02	X0=	-.60375
.19999E-02	X0=	-.58293
.17959E-02	X0=	-.56211
.16897E-02	X0=	-.54129
.16721E-02	X0=	-.52047
.16968E-02	X0=	-.49965
.17133E-02	X0=	-.47884
.16935E-02	X0=	-.45802
.16345E-02	X0=	-.43720
.15405E-02	X0=	-.41638
.14564E-02	X0=	-.39556
.13791E-02	X0=	-.37474
.12994E-02	X0=	-.35392
.12482E-02	X0=	-.33310
.12161E-02	X0=	-.31228
.11987E-02	X0=	-.29146
.11865E-02	X0=	-.27065
.11667E-02	X0=	-.24983
.11271E-02	X0=	-.22901
.10602E-02	X0=	-.20819
.96612E-03	X0=	-.18737
.85204E-03	X0=	-.16655
.72961E-03	X0=	-.14573
.61003E-03	X0=	-.12491
.50068E-03	X0=	-.10409
.40727E-03	X0=	-.08328
.34365E-03	X0=	-.06246
.33842E-03	X0=	-.04164
.40490E-03	X0=	-.02082
.51567E-03	X0=	0.00000
.63468E-03	X0=	.02082
.73642E-03	X0=	.04164
.80659E-03	X0=	.06246
.84031E-03	X0=	.08328
.84102E-03	X0=	.10409

.81611E-03	X0=	.12491
.77352E-03	X0=	.14573
.71898E-03	X0=	.16655
.65536E-03	X0=	.18737
.58376E-03	X0=	.20819
.50545E-03	X0=	.22901
.42352E-03	X0=	.24983
.34301E-03	X0=	.27065
.26875E-03	X0=	.29146
.20133E-03	X0=	.31228
.13575E-03	X0=	.33310
.77393E-04	X0=	.35392
.11085E-03	X0=	.37474
.23311E-03	X0=	.39556
.38543E-03	X0=	.41638
.54644E-03	X0=	.43720
.69864E-03	X0=	.45802
.82747E-03	X0=	.47884
.92293E-03	X0=	.49965
.98556E-03	X0=	.52047
.10150E-02	X0=	.54129
.10174E-02	X0=	.56211
.99809E-03	X0=	.58293
.96318E-03	X0=	.60375
.92222E-03	X0=	.62457
.88941E-03	X0=	.64539
.87876E-03	X0=	.66621
.89238E-03	X0=	.68702
.91321E-03	X0=	.70784
.91197E-03	X0=	.72866
.86144E-03	X0=	.74948
.74699E-03	X0=	.77030
.57160E-03	X0=	.79112
.36095E-03	X0=	.81194
.19494E-03	X0=	.83276
.26853E-03	X0=	.85358
.43413E-03	X0=	.87439
.55016E-03	X0=	.89521
.57066E-03	X0=	.91603
.51199E-03	X0=	.93685
.35991E-03	X0=	.95767
.19871E-03	X0=	.97849
.31104E-03	X0=	.99931
.58150E-03	X0=	1.02013
.83784E-03	X0=	1.04095
.10255E-02	X0=	1.06177
.11167E-02	X0=	1.08258
.11035E-02	X0=	1.10340
99604E-03	X0=	1.12422
.82093E-03	X0=	1.14504
.62321E-03	X0=	1.16586
.48446E-03	X0=	1.18668
.50807E-03	X0=	1.20750
.66028E-03	X0=	1.22832
.82724E-03	X0=	1.24914
.94191E-03	X0=	1.26995
.97429E-03	X0=	1.29077
.91947E-03	X0=	1.31159
.79530E-03	X0=	1.33241
.64065E-03	X0=	1.35323
.50978E-03	X0=	1.37405
.44747E-03	X0=	1.39487
.44125E-03	X0=	1.41569
.44203E-03	X0=	1.43651

.42360E-03	X0=	1.45732
.40800E-03	X0=	1.47814
.43789E-03	X0=	1.49896
.52941E-03	X0=	1.51978
.61782E-03	X0=	1.54060
.69167E-03	X0=	1.56142
.72185E-03	X0=	1.58224
.70279E-03	X0=	1.60306
.63324E-03	X0=	1.62388
.50845E-03	X0=	1.64469
.32399E-03	X0=	1.66551
.11828E-03	X0=	1.68633
.26354E-03	X0=	1.70715
.54366E-03	X0=	1.72797
.77612E-03	X0=	1.74879
.90570E-03	X0=	1.76961
.90435E-03	X0=	1.79043
.77630E-03	X0=	1.81125
.55646E-03	X0=	1.83207
.29944E-03	X0=	1.85289
.70659E-04	X0=	1.87370
.13775E-03	X0=	1.89452
.25082E-03	X0=	1.91534
.31250E-03	X0=	1.93616
.34474E-03	X0=	1.95698
.35737E-03	X0=	1.97780
.34073E-03	X0=	1.99862
.28240E-03	X0=	2.01944
.18289E-03	X0=	2.04025
.58766E-04	X0=	2.06107
.64270E-04	X0=	2.08189
.15861E-03	X0=	2.10271
.20652E-03	X0=	2.12353
.20380E-03	X0=	2.14435
.16285E-03	X0=	2.16517
.11415E-03	X0=	2.18599
.10140E-03	X0=	2.20681
.12290E-03	X0=	2.22762
0.	X0=	2.24844



SIMULATION TEST

MAGNITUDE VS XD

MAGNITUDE (YO-CUT)

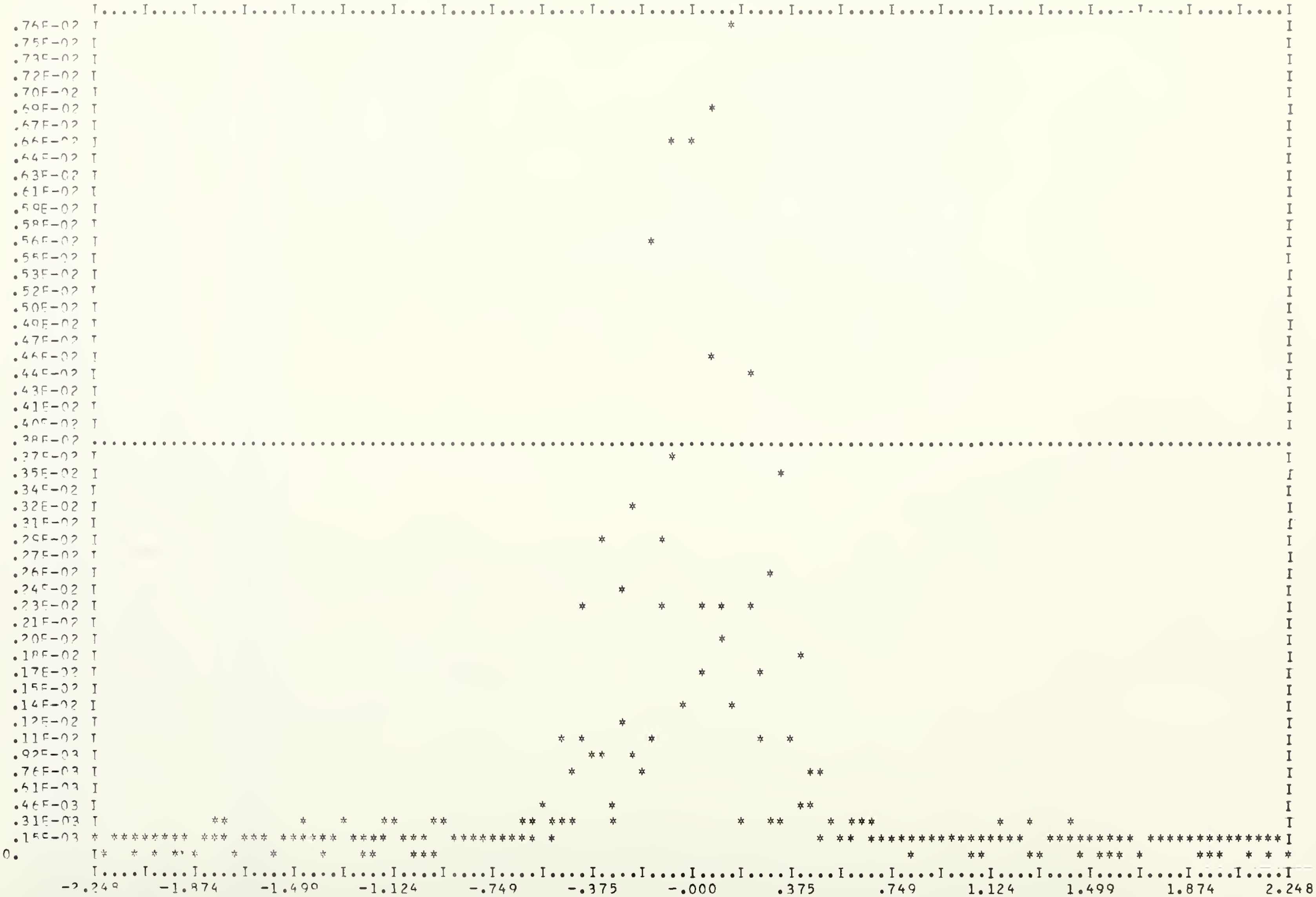
.91515E-04	YO=	-2.24844
.13541E-04	YO=	-2.22762
.72146E-04	YO=	-2.20681
.14610E-03	YO=	-2.18599

.18775E-03	Y0=	-2.16517
.10805E-03	Y0=	-2.14435
.17364E-03	Y0=	-2.12353
.11054E-03	Y0=	-2.10271
.54103E-04	Y0=	-2.08189
.13241E-03	Y0=	-2.06107
.72350E-04	Y0=	-2.04025
.11762E-03	Y0=	-2.01944
.12571E-03	Y0=	-1.99862
.11514E-03	Y0=	-1.97780
.14282E-03	Y0=	-1.95698
.26046E-04	Y0=	-1.93616
.14373E-03	Y0=	-1.91534
.92976E-04	Y0=	-1.89452
.41153E-04	Y0=	-1.87370
.11266E-03	Y0=	-1.85288
.15560E-03	Y0=	-1.83207
.29694E-03	Y0=	-1.81125
.11486E-03	Y0=	-1.79043
.26070E-03	Y0=	-1.76961
.91052E-04	Y0=	-1.74879
.68738E-04	Y0=	-1.72797
.45593E-04	Y0=	-1.70715
.87755E-04	Y0=	-1.68633
.94732E-04	Y0=	-1.66551
.12606E-03	Y0=	-1.64469
.79773E-04	Y0=	-1.62388
.13705E-03	Y0=	-1.60306
.48334E-04	Y0=	-1.58224
.72190E-04	Y0=	-1.56142
.11248E-03	Y0=	-1.54060
.97606E-04	Y0=	-1.51978
.14579E-03	Y0=	-1.49896
.23862E-03	Y0=	-1.47814
.21663E-03	Y0=	-1.45732
.18744E-03	Y0=	-1.43651
.21894E-03	Y0=	-1.41569
.16032E-03	Y0=	-1.39487
.47073E-04	Y0=	-1.37405
.15591E-03	Y0=	-1.35323
.14477E-03	Y0=	-1.33241
.24447E-03	Y0=	-1.31159
.21996E-03	Y0=	-1.29077
.15967E-03	Y0=	-1.26995
.18343E-03	Y0=	-1.24914
.25213E-04	Y0=	-1.22832
.21367E-03	Y0=	-1.20750
.44679E-04	Y0=	-1.18668
.26585E-03	Y0=	-1.16586
.17762E-03	Y0=	-1.14504
.25200E-03	Y0=	-1.12422
.15306E-03	Y0=	-1.10340
.17255E-03	Y0=	-1.08258
.13465E-03	Y0=	-1.06177
.19069E-04	Y0=	-1.04095
.15722E-03	Y0=	-1.02013
.26456E-04	Y0=	-.99931
.32615E-03	Y0=	-.97849
.70149E-04	Y0=	-.95767
.28435E-03	Y0=	-.93685
.16635E-03	Y0=	-.91603
.16022E-03	Y0=	-.89521
.86043E-04	Y0=	-.87439
.17072E-03	Y0=	-.85358

```
.11310E-03   Y0=   -.83276
.13777E-03   Y0=   -.81194
.21970E-03   Y0=   -.79112
.14964E-03   Y0=   -.77030
.14402E-03   Y0=   -.74948
.17519E-03   Y0=   -.72866
.11685E-03   Y0=   -.70784
.11118E-03   Y0=   -.68702
.15390E-03   Y0=   -.66621
.19817E-03   Y0=   -.64539
.36122E-03   Y0=   -.62457
.35374E-03   Y0=   -.60375
.29528E-03   Y0=   -.58293
.44202E-03   Y0=   -.56211
.95290E-04   Y0=   -.54129
.36763E-03   Y0=   -.52047
.29067E-03   Y0=   -.49965
.10417E-02   Y0=   -.47884
.28120E-03   Y0=   -.45802
.74425E-03   Y0=   -.43720
.22484E-02   Y0=   -.41638
.11234E-02   Y0=   -.39556
.61955E-03   Y0=   -.37474
.28635E-02   Y0=   -.35392
.93785E-03   Y0=   -.33310
.44355E-03   Y0=   -.31228
.32603E-03   Y0=   -.29146
.24560E-02   Y0=   -.27065
.12581E-02   Y0=   -.24983
.89855E-03   Y0=   -.22901
.32492E-02   Y0=   -.20819
.77015E-03   Y0=   -.18737
.10810E-02   Y0=   -.16655
.55732E-02   Y0=   -.14573
.28959E-02   Y0=   -.12491
.22508E-02   Y0=   -.10409
.37279E-02   Y0=   -.08328
.65583E-02   Y0=   -.06246
.14388E-02   Y0=   -.04164
.13094E-02   Y0=   -.02082
.66346E-02   Y0=   0.00000
.22135E-02   Y0=    .02082
.17351E-02   Y0=    .04164
.67977E-02   Y0=    .06246
.45015E-02   Y0=    .08328
.19135E-02   Y0=    .10409
.22826E-02   Y0=    .12491
.76276E-02   Y0=    .14573
.13301E-02   Y0=    .16655
.32614E-03   Y0=    .18737
.44079E-02   Y0=    .20819
.22586E-02   Y0=    .22901
.16421E-02   Y0=    .24983
.10509E-02   Y0=    .27065
.25398E-02   Y0=    .29146
.35823E-03   Y0=    .31228
.28201E-03   Y0=    .33310
.35087E-02   Y0=    .35392
.10666E-02   Y0=    .37474
.52251E-03   Y0=    .39556
.18746E-02   Y0=    .41638
.72773E-03   Y0=    .43720
.51175E-03   Y0=    .45802
.21676E-03   Y0=    .47884
```

.70530E-03	Y0=	.49965
.28398E-03	Y0=	.52047
.30676E-03	Y0=	.54129
.12172E-03	Y0=	.56211
.15435E-03	Y0=	.58293
.27154E-03	Y0:	.60375
.29791E-03	Y0=	.62457
.26802E-03	Y0=	.64539
.18502E-03	Y0=	.66621
.23327E-02	Y0=	.68702
.12029E-03	Y0=	.70784
.88555E-04	Y0=	.72866
.13803E-03	Y0=	.74948
.12648E-03	Y0=	.77030
.16843E-03	Y0=	.79112
.52775E-04	Y0=	.81194
.19052E-03	Y0=	.83276
.18441E-03	Y0=	.85358
.15729E-03	Y0=	.87439
.18296E-03	Y0=	.89521
.13367E-03	Y0=	.91603
.15884E-03	Y0=	.93685
.13297E-03	Y0=	.95767
.22006E-03	Y0=	.97849
.19274E-03	Y0=	.99931
.15359E-03	Y0=	1.02013
.72577E-04	Y0=	1.04095
.17839E-03	Y0=	1.06177
.34843E-04	Y0=	1.08258
.19741E-03	Y0=	1.10340
.88869E-04	Y0=	1.12422
.26152E-03	Y0=	1.14504
.93958E-04	Y0=	1.16586
.14829E-03	Y0=	1.13668
.20403E-03	Y0=	1.20750
.14026E-03	Y0=	1.22832
.16913E-03	Y0=	1.24914
.51570E-04	Y0=	1.26995
.23129E-03	Y0=	1.29077
.31196E-04	Y0=	1.31159
.14059E-03	Y0=	1.33241
.19576E-03	Y0=	1.35323
.15358E-03	Y0=	1.37405
.19072E-03	Y0=	1.39487
.79567E-04	Y0=	1.41569
.23757E-03	Y0=	1.43651
.30623E-04	Y0=	1.45732
.20554E-03	Y0=	1.47814
.98275E-04	Y0=	1.49896
.13655E-03	Y0=	1.51978
.28172E-04	Y0=	1.54060
.11456E-03	Y0=	1.56142
.56253E-04	Y0=	1.58224
.94061E-04	Y0=	1.60306
.40997E-04	Y0=	1.62388
.13479E-03	Y0=	1.64469
.10238E-03	Y0=	1.66551
.36207E-04	Y0=	1.68633
.13303E-03	Y0=	1.70715
.91943E-04	Y0=	1.72797
.15972E-03	Y0=	1.74879
.92501E-04	Y0=	1.76961
.16354E-03	Y0=	1.79043
.15524E-03	Y0=	1.81125

.79226E-04	Y0=	1.83207
.15776E-03	Y0=	1.85288
.82778E-04	Y0=	1.87370
.73537E-04	Y0=	1.89452
.11984E-03	Y0=	1.91534
.66383E-04	Y0=	1.93616
.13131E-03	Y0=	1.95698
.61597E-04	Y0=	1.97780
.15033E-03	Y0=	1.99862
.14451E-03	Y0=	2.01944
.87715E-04	Y0=	2.04025
.16549E-03	Y0=	2.06107
.44551E-04	Y0=	2.08189
.15307E-03	Y0=	2.10271
.88289E-04	Y0=	2.12353
.12989E-03	Y0=	2.14435
.17851E-03	Y0=	2.16517
.64572E-04	Y0=	2.18599
.81883E-04	Y0=	2.20681
.12964E-03	Y0=	2.22762
0.	Y0=	2.24844



SIMULATION TEST

MAGNITUDE VS YD

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET *(See instructions)*	1. PUBLICATION OR REPORT NO.	2. Performing Organ. Report No.	3. Publication Date
	NBSIR 80-1630		June 1981

4. TITLE AND SUBTITLE

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

5. AUTHOR(S)

C. F. Stubenrauch and A. D. Yaghjian

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No. DAEA-76-F-D760

8. Type of Report & Period Covered NBSIR July 1976 - June 1978

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

US Army Communications Electronics Engineering
Installation Agency
Fort Huachuca, Arizona 85613

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

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.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

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

13. AVAILABILITY

[X] Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

[X] Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES 176

15. Price $12.00

☆ U.S. GOVERNMENT PRINTING OFFICE: 1981 — 780-508/419 Region No. 8