NBSIR 81-1643

EFFICIENT COMPUTATION OF THE FAR FIELD RADIATED BY AN ARBITRARY RECTANGULAR-APERTURE DISTRIBUTION

COMPUTER PROGRAM DOCUMENTATION

Richard Lewis

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

March 1981

NBSIR 81-1643

EFFICIENT COMPUTATION OF THE FAR FIELD RADIATED BY AN ARBITRARY RECTANGULAR-APERTURE DISTRIBUTION

COMPUTER PROGRAM DOCUMENTATION

Richard Lewis

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

March 1981

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

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

TABLE OF CONTENTS

LIST OF FIGURES

Efficient Computation of the Far Field
Radiated by an Arbitrary Rectangular-Aperture Distribution

Computer Program Documentation

Richard Lewis

This report contains the computer documentation for calculating the far-zone electric field due to a user-prescribed electric-field distribution within a rectangular aperture. The far-field output is computed along two arbitrarily selected, perpendicular, spatial-frequency plane cuts. Program execution time is minimized by the use of fast Fourier transform (FFT) processing. The program was designed so that the required far-field output is obtained by processing only two, vector, one-dimensional FFTs. The far-field results are obtained in the form of elevation and azimuth vector components and electric-field-vector magnitude. A complete analytical discussion of the problem is presented, along with sample graphical output to illustrate how aliasing and output resolution limitations effect the graphical results.

Key words: Algorithm; antenna; aperture; computer program; electromagnetic; far-field; FFT; Fourier transform.

I. INTRODUCTION

This report constitutes the computer-program documentation for calculating the far-zone electric field, due to a prescribed electric-field distribution within a rectangular aperture, along two perpendicular spatial-frequency plane cuts. Program execution time is minimized by use of fast Fourier transform processing. The analytical basis for the computations is presented in Appendix A. The following text describes the computer program, listed in Appendix B, that has been developed for implementing these computations. The same nomenclature is used by the text as in the program listing for the variable parameters; consequently, reader familiarity with the input parameter list, as specified by the program comment cards, should prove beneficial.

II. PROBLEM DESCRIPTION

We divide space into two regions by an infinite plane surface (the aperture plane) containing the rectangular-aperture surface. The region in which the far-field computation is desired is assumed to be homogeneous and source free. The theoretical basis for calculating the far field due to a tangential electric-field distribution over an aperture plane is well known [1,2,3]. For physical antenna apertures that are large compared with a wavelength, the electromagnetic field amplitude will be small everywhere on the aperture plane except on the aperture surface. As a first approximation the far-zone radiation field may be found from the field on the aperture surface alone [1]. The aperture field may be determined either through direct near-field measurements or by suitable approximate analytical methods, such as ray tracing. It is presumed that the computer-program user can specify the tangential, aperture, electric-field vector (both amplitude and phase) at equidistant points on a rectangular grid over the aperture surface, while the field elsewhere on the aperture plane is assumed to be zero. The far-field output is obtained as a function of the spatial-frequency components k_x and k_y. Here, we compute the far-field output along user-chosen k_x- and k_y-plane cuts using fast Fourier transform (FFT) processing. We will refer to the k_x=0 and k_y=0 plane cuts as the principal-plane cuts. With just one k_x-plane cut and one k_y-plane cut far-field computation, the computer program herein described was designed so that the input data set would only be read in once while the far-field output would be obtained by just carrying out two vector one-dimensional FFTs. The result is an extremely efficient program, in terms of computation time and computer storage, for computing the far fields arising from a rectangular-grid aperture distribution along two perpendicular plane cuts.

The x-dimension of the rectangular aperture over which the user specifies the tangential electric-field vector is a; the y-dimension is b. The aperture plane is given by z=0, while the assumed aperture radiation travels in the positive z-direction. The number of data points within the aperture along the x-direction is NX, while the number of data points along the y-direction is NY. Let us presume that a matrix of x-component data and a matrix of y-component data is available for specifying the aperture field, and that each component value at a given data point is specified as a complex number,

$$|E|e^{i\psi} = |E|\cos\psi + i|E|\sin\psi,$$

$|E|$ being the amplitude and ψ being the phase of a particular datum. The dimensions of these two vector-component matrices would each be given by

NY x NX, so that each matrix row corresponds to a constant value of the y-coordinate, while each matrix column corresponds to a constant value of the x-coordinate. Thus, the row index r of the data matrix is proportional to the y-coordinate, while the column index c of the data matrix is proportional to the x-coordinate. For each row or column, a data point at the center of the respective x-dimension or y-dimension of the aperture shall be specified, along with an equal number of data points on each side of the midpoint. Consequently, NX and NY are both odd numbers. The spacing between adjacent data points in the x-direction is $\delta_x = a/NX$, while the spacing in the y-direction is $\delta_y = b/NY$. It is presumed, by these equalities, that the edges of the rectangular aperture are located half a data-point spacing beyond the farthest data points within the aperture.

III. USER INPUT TO THE PROGRAM

The program user provides a subroutine, GETARAY, that obtains or calculates one row of x-coordinate aperture-field data, corresponding to a particular value of the y-coordinate and to a particular vector component, each time it is called. The subroutine arguments are NX and STOR, where NX is the number of data points in the row and STOR is a complex array of dimension NX for storing the aperture-field data. The value of NX may not be changed by the user's subroutine, GETARAY. This subroutine is called NY times for the x-component data, followed by another NY times for the y-component data. Each successive call represents a lower row (greater row-index number) in the data matrix of each vector component, or a larger value of the y-coordinate. Within the row of data supplied by GETARAY, each successive datum represents a larger value of the x-coordinate. In those regions of the rectangular aperture that are beyond the boundary of the user's physical aperture, the user is expected to provide zero-fill via subroutine GETARAY.

It may be observed that spatial integration of a physical electric field over the aperture plane will give rise to a function of spatial frequency k_x and k_y that is virtually bandlimited [2]. According to the sampling theorem [4], a band-limited two-dimensional function with bandlimits K_x and K_y may be completely recovered using sampling intervals δ_x and δ_y that satisfy the relations,

$$K_x = \frac{\pi}{\delta_x} \text{ and } K_y = \frac{\pi}{\delta_y} \quad .$$

A virtual bandlimit only slightly greater than the spatial propagation constant $k = 2\pi/\lambda$ ordinarily prevails, where λ is the wavelength in the medium, so that an aperture data-point spacing of $\delta_x \simeq \delta_y \lesssim \lambda/2$ generally is adequate. However, in order to minimize aliasing errors in the far-sidelobe region, closer data-point spacings may be required (cf. Appendix A, eq. (16) and the preceding discussion, for analytical details.) Consequently, computer-program flexibility is maintained with the specification

$$\delta_x \equiv \frac{\lambda}{S_x} \text{ and } \delta_y \equiv \frac{\lambda}{S_y}.$$

Here, $S_x \simeq SX$ and $S_y \simeq SY$, where SX and SY are dimensionless numbers specified by the program user. There is only a small difference between S_x and SX and between S_y and SY, which the computer program can automatically compute. Thus, the user need only be concerned with the order of magnitude of SX and SY. With a uniformly-excited-aperture test case, highly accurate comparisons with theoretical results required values of SX, SY > 9 due to the presence of evanescent modes. If evanescent modes are not present, values of SX, SY ≃ 2 should suffice for most applications. Values of SX, SY < 2 should be used cautiously. In addition to SX and SY, the user must specify the variables ALPHA = α and BETA = β, where α and β serve to specify the aperture dimensions normalized to the wavelength. Thus,

$$\alpha \equiv \frac{a}{\lambda} \quad \text{and} \quad \beta \equiv \frac{b}{\lambda}.$$

As a consequence of these definitions, the values of the aperture-spacing normalization constants, S_x and S_y, are given by

$$S_x = \frac{NX}{\alpha} \quad \text{and} \quad S_y = \frac{NY}{\beta}.$$

In practice, the computer program makes the initial computations, NX = α*SX and NY = β*SY, and then checks to see if the values of NX and NY so computed are odd. If not, then unity is added to the result so that the final values of NX and NY will be odd. Then, the exact values of S_x and S_y are computed.

Although only the field within the aperture region will contribute to the integrand, with FFT processing the resolution of the far-field output will depend on the size of the area of integration. As an example, if a uniform aperture excitation is specified, and if the area of integration just equals the aperture area, then the program would produce far-field output at the points $k_x = 0, \pm 2\pi/a, \pm 4\pi/a, \pm 6\pi/a$, etc., corresponding to the exact (elevation-plane) far-field pattern function,

$$E \propto \frac{\sin (k_x \frac{a}{2})}{k_x \frac{a}{2}} .$$

In other words, except for one point, one would just obtain output values at the nulls of the far-field pattern function, which would provide extremely limited information. A more detailed picture would be obtained by increasing the output resolution so as to produce output values at intermediate points between the pattern-function nulls. By extending the area of integration and adding zero-fill to the FFT data input, one can increase the output resolution.

The extent of the area of integration is increased by increasing the number of data points supplied to the FFT processing routine. The data amplitude values at each additional data point are automatically set equal to zero by the computer program. The number of data points supplied to the FFT processor is designated NNX in the case of x-direction FFT processing and NNY in the case of y-direction FFT processing, where NNX > NX and NNY > NY. In the case of the FFT processor used by the computer program, the number of input data points must equal a power of two. Thus, NNX = 2**MMX and NNY = 2**MMY. The program automatically chooses NNX to be twice the largest power of two that is just greater than NX, and similarly for NNY and NY. However, these automatically generated values may not suffice for all applications; consequently, the automatically generated values of NNX and NNY are multiplied by 2**NZX and 2**NZY, respectively, to produce updated values of NNX and NNY. The two constants NZX and NZY are specified by the user, where NZX, NZY $\geq$ -1. The recommended specification is NZX=NZY=1 or greater, although specifying NZX or NZY equal to zero may be adequate if the aperture field tapers to near zero at the aperture boundaries. With a uniformly-excited-aperture test case, the values NZX=NZY=2 gave smooth plotting resolution with a microfilm plotter subroutine. It is not recommended that the -1 specification for NZX or NZY be used unless NX or NY is just slightly larger than some power of 2, as otherwise there will not be adequate zero-fill added to the input data. It should be recognized that the larger that NNX and NNY become, the longer the FFT computation time becomes and the larger the computer-storage requirement becomes.

In addition to the input variables already described, the user must specify the azimuth angle A_0 and the elevation angle E_0 corresponding to the particular plane cuts desired. These input quantities are best discussed within the context of the next section.

IV. PROGRAM OUTPUT

The program computes the azimuth (E_A) and elevation (E_E) vector-components of the far field, along with the antenna pattern or electric-vector magnitude (the program listing for azimuth and elevation vector-component output is as given in appendix B, while a listing of substitute cards for use when theta and phi vector-component output is preferred may be found in appendix C). The program computes both output vector components along a constant k_x-coordinate plane cut and along a constant k_y-coordinate plane cut as respective functions of elevation (e) and azimuth (A). That is, the spatial-frequency component that is not held constant becomes the sole output-function variable. Note that both the elevation (e) and azimuth (A) angles become equal to the polar angle theta in the event that principal-plane cuts are selected (users who only require principal-plane-cut computations may wish to utilize the simpler version of the program given in appendix D). The relationships between the various spherical-coordinate-system angles that have been mentioned are shown in figures 1 and 2, with supplementary discussion found in [5]†.

At this point it would be appropriate to note that the final computed values of the far-zone electric fields are renormalized, such that the factor

†The notation used here to designate the alternative azimuth and elevation angles is "a" and "e", whereas in [5] these alternative angles are designated α and β. The notation A and E normally is used with az/el antenna mounts, while the a and e designations apply to el/az antenna mounts. Our A and E angle designations agree with the notation used in [5] for these quantities.

$\frac{\lambda}{r} e^{ikr}$ is divided out.

Corresponding to each start-up of the program, a complete set of input x-component and y-component aperture data is requested from GETARAY by the program, and one k_x-plane cut and one k_y-plane cut far-field computation is carried out. The particular k_x-plane cut generated is selected by specifying the azimuthal angle A_o, at an elevation angle of zero degrees, that corresponds to the particular value of k_x desired. Similarly, the particular k_y-plane cut generated is selected by specifying the elevation angle E_o corresponding to the particular value of k_y desired (the "o" subscripts are used here to distinguish the user-specified azimuth and elevation angles that define the plane cuts, whereas the omission of this subscript implies the general angular-coordinate variables). The relationships between the far-field spatial-frequency variables, k_x and k_y, and the spherical-coordinate angles, depicted in figures 1 and 2, are given by,

$$k_x = k \text{ Cos } E \text{ Sin } A = k \text{ Sin } \theta \text{ Cos } \phi = k \text{ Sin } a$$

$$k_y = k \text{ Sin } E = k \text{ Sin } \theta \text{ Sin } \phi = k \text{ Cos } a \text{ Sin } e$$

From the preceding, we see that the desired k_x- and k_y-plane cuts are given by $k_x/k = \text{Sin } A_o$ and $k_y/k = \text{Sin } E_o$, respectively, where the angles A_o and E_o are specified by the user.

Upon completing the far-field computations, the azimuth and elevation vector-components and the antenna pattern are plotted as a function of elevation (e) and azimuth (A), respectively, for the k_x-plane cut and the k_y-plane cut. The plotting is limited to angles in the visible range (i.e., real angles within the spherical spatial-frequency region of radius k, such that $k_z = \sqrt{k^2 - k_x^2 - k_y^2}$ is real). This visible range not only excludes complex evanescent angles, but also limits the range on k_y and k_x to $k_y \leq k \text{ Cos } A_o$ and $k_x \leq k \text{ Cos } E_o$, respectively, in the case of the k_x- and k_y-plane cuts. Since the k_x-plane cut is plotted versus elevation (e), while the k_y-plane cut is plotted versus azimuth (A), one obtains a full ± 90° plotting range. However, it must be recognized that rapid variations, as respective functions of k_y or k_x, will appear greatly expanded when A_o or E_o do not equal zero.

V. PLOTTING SUBROUTINES

The subroutine PAMPLOT that is supplied with the program for setting up the plotting arrays permits the user to restrict the plotting range through the specification of maximum and minimum abscissa values. Both relative amplitude in dB and phase in degrees are plotted for each vector-component, with just relative amplitude in dB plotted for the antenna pattern. This routine labels each plot with the peak amplitude level in dBL† and with a self-generated caption that automatically indicates the vector-component plotted; phase, amplitude, or magnitude plot; and the value of either the azimuthal angle (A_0) or the elevation angle (E_0) corresponding to the particular k_x- or k_y-plane cut that has been generated. Also, a user-specified 40-character graph legend is placed on each plot. In addition to plotting far-field output, subroutine PAMPLOT allows the user to obtain a printout of every jth data point, j=1,2,..., etc. (Note that the reciprocal of j times 100 gives the percentage of output points to be printed.) If j=0, printout is suppressed. This printout gives the abscissa point, absolute amplitude, and phase in degrees.

The actual plotting routine supplied with the program is limited to point plotting on the user's printout. The user can elect either to point-plot the amplitude and phase of each vector component separately, or else the amplitude and phase curves can both be point-plotted on the same graph. The latter plotting mode has the advantage of placing two closely related curves in proximity to each other. The disadvantage is that ordinate-axis labeling of the phase plot will be suppressed by ordinate-axis labeling of the amplitude plot. A microfilm plotting routine is highly desirable. Subroutine PAMPLOT contains an illustrative calling statement to the dummy routine CRTPLOT, which may be replaced by the user's own plotting-routine call. The dummy subroutine that is called here is SUBROUTINE CRTPLOT (XARRAY, YARRAY, XMAX, XMIN, YMAX, YMIN, N, LABEL, NO, NOMAX), where XARRAY contains the N abscissa points, YARRAY contains the N ordinate points, XMAX is the greatest abscissa value plotted, XMIN is the least abscissa value, YMAX is the greatest ordinate

†Here, dBL is taken to mean 20 $\text{Log}_{10}(E)$, where E designates the quantity whose amplitude level is desired. (cf. IEEE Standard Dictionary of Electrical & Electronics Terms under "level".)

value, YMIN is the least ordinate value, N equals the number of data points per curve, LABEL is a nine-word graph legend, NO is the number of the curve being plotted, and NOMAX is the number of curves per graph ($1 \leq$ NO $\leq$ NOMAX ≤ 2).

VI. DIMENSION STATEMENT SPECIFICATIONS

The program comment cards supplied with each routine should enable the user to specify the correct input to run the program. Particular attention is called to the array-dimension specifications in the main program. Insufficient dimensions (as specified by the user's data statement) will cause the program to abort. Note that the dimensions of arrays THETA and DATA are based on the final computed value of NNX, while the dimension of array EI equals twice the sum of the final computed values of NNX and NNY. The dimensions of arrays EXPX and STOR are equal to NX, while the dimension of array EXPY is equal to NY.

VII. PROGRAM VERIFICATION

In order to verify that the computer program was working properly, it was tested to see how accurately it could compute the far fields of uniformly excited apertures, for which exact formulas are available [7, pp. 332-336]. Good agreement was obtained between the theoretical results and the computer program output, particularly as the data-sampling interval was decreased. As mentioned earlier, very short data-point spacings were required due to the presence of evanescent modes in these hypothetical aperture distributions. Tests were carried out both for a rectangular aperture and for a circular aperture with zero fill to the boundary of the enclosing square aperture. In order to facilitate numerical comparisons, computations with the exact formulas were carried out at the same far-field coordinates that the computer program used to calculate the far-zone fields.

In figure 4 we show some principal-plane-cut results for a uniformly-illuminated slit aperture five wavelengths wide, in which we compare results from computations with a data-point spacing of $\lambda/5$ and $\lambda/3$ to the corresponding exact theoretical curve. The solid-line curve in figure 4 corresponds to the $\lambda/5$ data-point spacing curve, while the dots and crosses, respectively, correspond to the theoretical curve and to the $\lambda/3$ data-point spacing curve. Just those portions of the latter two curves which do not overlap the solid line curve are shown in the figure. Thus, the only differences shown are in the far side-lobe region where the overlapping ends of the replicated far-field function contribute the most to the aliasing error (refer to Appendix A, eq. (15) for analytical details). As may be anticipated from this figure, we have verified that the computed curve does get closer and closer to the theoretical curve as the data-point spacing gets smaller and smaller.

To illustrate the effect of adding zero fill to the FFT to increase the output resolution, in figure 5 we show a comparison between two curves having the same data-point spacing but different specified values of NZY. Here, the solid line curve corresponds to NZY = 0 while the dotted line curve corresponds to NZY = 3. Adjacent points on each curve are connected by a straight line with no curve fitting between points. Both figures 4 and 5 correspond to the same plane cut and the same aperture excitation, but the solid line curve of figure 4 was generated with NZY = 4; consequently, the nulls shown in figure 4 are uniformly deeper than the nulls shown in the dotted line curve of figure 5. However, the computed values of figure 5 are the more accurate, as they were obtained using a data point spacing of $\lambda/7$.

Finally, in figures 6 and 7 we show some antenna pattern magnitudes for a uniformly excited circular aperture three wavelengths in diameter. Here, we were interested in comparing the computer program output against the theoretical model for some arbitrarly selected plane cuts. Consequently, we show two k_y-plane-cut antenna-pattern curves plotted versus azimuth, in figures 6 and 7, corresponding respectively to a 10° elevation angle and a 30° elevation angle. The peak amplitude in figure 7, incidently, is about 15 dB below the peak amplitude in figure 6. In order to achieve highly accurate program output, a data point spacing of $\frac{\lambda}{21}$ was used, which resulted in the generation of numerical dB values for the furthest side lobes that agreed with the theoretical results within eight-tenths of one percent. This compares to just five or ten percent agreement between the furthest side lobes of the theoretical model and the far side lobes of the computed curves plotted in figure 4. A value of NZX = NZY = 2 was selected for our circular aperture computations, resulting in just moderately smooth plotting resolution. It might be remarked that the aperture excitation was assumed to be diagonal to the cartesian coordinates of the aperture, resulting in the asymetrical k_y-plane-cut patterns shown.

APPENDIX A

Computation of the Far Field due to a Tangential Electric-Field Distribution over an Aperture Plane using Fast Fourier Transform Processing; Theoretical Analysis

A.1 Far-Zone Electric-Field Components Expressed in Terms of Plane-Wave Spectrum Components

An expression for the far-zone electric-field vector transverse to the z-direction is readily obtained by integrating the near-zone tangential field over an aperture plane [1], [2]. Thus, we have the far-zone electric field transverse to the z-direction expressed as

$$\underline{E}_t(\underline{r}) \sim \frac{-ik\ \mathrm{Cos}\theta}{2\pi} \frac{e^{ikr}}{r} \underline{B}\left(\frac{k}{r}\underline{R}\right) \tag{1}$$

where the aperture-plane integral is defined as

$$\underline{B}\left(\frac{k}{r}\underline{R}\right) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \underline{E}_t(\underline{P}')\, e^{-i\frac{k}{r}\underline{R}\cdot\underline{P}'} dx'dy'. \tag{2}$$

Here, $\underline{P}' = x'\underline{a}_{x'} + y'\underline{a}_{y'}$ and $\underline{R} = x\underline{a}_x + y\underline{a}_y$, where $\underline{a}_x = \underline{a}_{x'}$ and $\underline{a}_y = \underline{a}_{y'}$ are unit vectors along the x and y coordinate-system axes, and where x and y denote far-zone position coordinates while x' and y' denote aperture-plane position coordinates. The far-zone radial coordinate is $r = \sqrt{x^2+y^2+z^2}$. These geometrical relationships are depicted in figure 3. The propagation constant is $k = 2\pi/\lambda$, where λ is the wavelength. An $e^{-i\omega t}$ time dependence is implicitly assumed by (1); thus $k = \frac{\omega}{c}$ where c is the velocity of light. The spatial-frequency components k_x and k_y are related to the far-zone position coordinates by the relations, $k_x = k\frac{x}{r}$ and $k_y = k\frac{y}{r}$. $\underline{E}_t(\underline{P}')$ is the tangential electric-field vector in the aperture plane. The z-component of the far-zone electric field is also required, but it can be obtained from the expression

$$B_z\left(\frac{k}{r}\underline{R}\right) = -\frac{1}{r\ \mathrm{Cos}\theta}\underline{R}\cdot\underline{B}\left(\frac{k}{r}\underline{R}\right), \tag{3}$$

which results from the fact that each plane-wave spectrum vector, $\underline{B}\left(\frac{k}{r}\underline{R}\right) + \underline{a}_z B_z\left(\frac{k}{r}\underline{R}\right)$, must be orthogonal to the corresponding propagation direction $\underline{k} = \frac{k}{r}(\underline{R} + z\underline{a}_z)$.

Consequently, it readily follows from (1) and (3) that the theta and phi components of the far field can be expressed as [1]

$$E_\theta(\underline{r}) \sim -\frac{ik}{2\pi}\frac{e^{ikr}}{r}[B_x(\frac{k}{r}\underline{R})\ \mathrm{Cos}\phi + B_y(\frac{k}{r}\underline{R})\ \mathrm{Sin}\phi] \tag{4a}$$

$$E_\phi(\underline{r}) \sim -\frac{ik}{2\pi}\frac{e^{ikr}}{r}[-B_x(\frac{k}{r}\underline{R})\ \mathrm{Sin}\phi + B_y(\frac{k}{r}\underline{R})\ \mathrm{Cos}\phi]\ \mathrm{Cos}\theta \tag{4b}$$

In place of the theta-phi component description of the far field, the alternative elevation and azimuth component description is frequently chosen (cf. figure 1). These far-field components are given by

$$E_E(\underline{r}) \sim \frac{-ik}{2\pi}\frac{e^{ikr}}{r} B_y(\frac{k}{r}\underline{R})\ \mathrm{Cos}A \tag{5a}$$

$$E_A(\underline{r}) \sim \frac{-ik}{2\pi}\frac{e^{ikr}}{r}[B_x(\frac{k}{r}\underline{R})\ \mathrm{Cos}E + B_y(\frac{k}{r}\underline{R})\ \mathrm{Sin}E\ \mathrm{Sin}A] \tag{5b}$$

The radial component of the far field, which is perpendicular to each of the components in equations (4) and (5), is equal to zero.

The elevation (E) and azimuth (A) angles, shown in figure 1, are obtained from

$$\mathrm{Sin}E = \mathrm{Sin}\theta\ \mathrm{Sin}\phi, \quad \mathrm{Tan}A = \mathrm{Tan}\theta\ \mathrm{Cos}\phi, \tag{6}$$

whereas the usual spherical-coordinate angles theta (θ) and phi (ϕ) may be expressed in terms of the far-field spatial-frequency components k_x and k_y using

$$\mathrm{Tan}\phi = \frac{k_y}{k_x}, \quad \mathrm{Sin}\theta = \frac{\sqrt{k_x^2 + k_y^2}}{k}. \tag{7}$$

The alternative azimuth (a) and elevation (e) angles, depicted in figure 2, are obtained from the expressions

$$\mathrm{Sin}\ a = \mathrm{Sin}\theta\ \mathrm{Cos}\phi = \mathrm{Cos}E\ \mathrm{Sin}A \tag{8a}$$

and

$$\mathrm{Tan}\ e = \mathrm{Tan}\theta\ \mathrm{Sin}\phi = \mathrm{Tan}E\ \mathrm{Sec}A. \tag{8b}$$

We can express the magnitude of the far-zone electric-field vector, $|\underline{E}|^2 = \underline{E}\ \underline{E}^*$, where * implies the complex conjugate, as

$$|\underline{E}|^2 = |E_\theta|^2 + |E_\phi|^2 = |E_E|^2 + |E_A|^2.$$

Substituting from either (4) or (5) into the above expression results in

$$|\underline{E}|^2 \sim \frac{k^2}{4\pi^2 r^2}\{|B_x|^2\ \mathrm{Cos}^2 E + |B_y|^2\ \mathrm{Cos}^2 a + (B_x B_y^* + B_x^* B_y)\ \mathrm{Sin}\ E\ \mathrm{Sin}\ a\}, \tag{9}$$

where the argument, $\frac{k}{r}\underline{R}$, of $B_x(\frac{k}{r}\underline{R})$ and $B_y(\frac{k}{r}\underline{R})$ is understood.

Finally, we give some explicit relations between the vector-components which hold on the principal-plane cuts. Thus, for the principal plane $\phi = 0$ we have:

$$E = e = 0,\quad A = a = \theta$$
$$\underline{R} = r\ \sin\theta\ \underline{a}_x$$

and

$$E_A(\underline{r}) = E_\theta(\underline{r}) \sim -\frac{ik}{2\pi}\frac{e^{ikr}}{r} B_x(\frac{k}{r}\underline{R})$$

$$E_E(\underline{r}) = E_\phi(\underline{r}) \sim -\frac{ik}{2\pi}\frac{e^{ikr}}{r} B_y(\frac{k}{r}\underline{R})\ \mathrm{Cos}\theta. \tag{10}$$

On the principal plane $\phi = \frac{\pi}{2}$ the relationships become:

$$A = a = 0,\quad E = e = \theta$$
$$\underline{R} = r\ \sin\theta\ \underline{a}_y$$

and

$$E_A(\underline{r}) = -E_\phi(\underline{r}) \sim -\frac{ik}{2\pi}\frac{e^{ikr}}{r} B_x(\frac{k}{r}\underline{R})\ \mathrm{Cos}\theta$$

$$E_E(\underline{r}) = E_\theta(\underline{r}) \sim -\frac{ik}{2\pi}\frac{e^{ikr}}{r} B_y(\frac{k}{r}\underline{R}). \qquad (11)$$

It may be noted that a jump change in sign of the θ and ϕ components, due to the jump change in ϕ by π radians upon passing through the coordinate-system origin, has been suppressed in writing expressions (10) and (11).

A.2 Finite Fourier Transform Representation of the Plane-Wave Spectrum Integral

We next require computational expressions for evaluating the aperture-plane integral defined by (2). If we presume that $\underline{B}(\frac{k}{r}\underline{R})$ is a bandlimited function of the spatial-frequency components k_x and k_y, having bandlimits $-K_x \leq k_x \leq K_x$ and $-K_y \leq k_y \leq K_y$, then the aperture-plane integral (2) can be expressed as [2]

$$\underline{B}(\frac{k}{r}\underline{R}) = \delta_x\delta_y \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \underline{E}_t(\underline{P}_{mn})\, e^{-i\frac{k}{r}\underline{R}\cdot\underline{P}_{mn}};\quad |k_x| \leq K_x,\ |k_y| \leq K_y. \qquad (12)$$

Here, $\underline{P}_{mn} = m\delta_x\underline{a}_x + n\delta_y\underline{a}_y$, where $\delta_x = \frac{\pi}{K_x}$, and $\delta_y = \frac{\pi}{K_y}$.

Furthermore, if the aperture-plane tangential electric-field vector is restricted to a rectangular aperture of dimensions $a = NX\delta_x$ and $b = NY\delta_y$, then the aperture electric field will be non-trivial only over the range $\frac{1-NX}{2} \leq m \leq \frac{NX-1}{2}$, $\frac{1-NY}{2} \leq n \leq \frac{NY-1}{2}$, so that the doubly infinite series in (12) may be replaced by two finite sums. Moreover, we can sample the transverse plane-wave spectrum vector $\underline{B}(\frac{k}{r}\underline{R})$ at the equally spaced points $k_x = j\frac{k}{r}\Delta_x$ and $k_y = \ell\frac{k}{r}\Delta_y$, where $\frac{k}{r}\Delta_x$ and $\frac{k}{r}\Delta_y$ are the spatial-frequency sampling intervals, and where j and ℓ are integers. The total number of x-coordinate spatial-frequency sampling points is N_x, and the total number of y-coordinate spatial-frequency sampling points is N_y, where $N_x > NX$ and $N_y > NY$. The range on the integers j and ℓ is

$$-\frac{N_x}{2} \leq j \leq \frac{N_x}{2} - 1 \quad \text{and} \quad -\frac{N_y}{2} \leq \ell \leq \frac{N_y}{2} - 1, \qquad (13)$$

so that the bandlimits are

$$K_x = \frac{k}{2r}\,\Delta_x N_x \quad \text{and} \quad K_y = \frac{k}{2r}\,\Delta_y N_y\ . \tag{14}$$

The numbers N_x and N_y respectively correspond to the quantities NNX and NNY described in the computer-program documentation. For the sake of computer program compatability with existing software, the number of input data points should equal the number of output data points. This may be accomplished by zero-filling the input data array corresponding to data points outside the rectangular-aperture boundaries.

Although (12) is an exact expression if the aperture-field integration results in a bandlimited function, our restriction of the input data to a rectangular aperture is not compatable with this condition. Consequently, there will result an aliasing error with the finite-Fourier-transform relation, so that the actual finite-Fourier-transform relation becomes [6]

$$\underline{B}_{j\ell} \equiv \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} \underline{B}\left(\frac{k}{r}\,\underline{R}_{j+pN_x,\ell+qN_y}\right)$$

$$= \delta_x\delta_y \sum_{m=-\frac{N_x}{2}}^{\frac{N_x}{2}-1} \sum_{n=-\frac{N_y}{2}}^{\frac{N_y}{2}-1} \underline{E}_{mn}\, e^{-i\frac{k}{r}\,\underline{R}_{j\ell}\cdot\underline{P}_{mn}}\ , \tag{15}$$

where the range on the integers j, ℓ is given by (13), and where we have written $\underline{E}_{mn}$ for $\underline{E}_t(\underline{P}_{mn})$. Also, $\underline{R}_{s,t} = s\Delta_x\underline{a}_x + t\Delta_y\underline{a}_y$, where s,t take on the values indicated in (15). From (15), we see that $\underline{B}\ (\frac{k}{r}\,\underline{R}_{j\ell})$ is replicated as $\underline{B}\ (\frac{k}{r}\,\underline{R}_{j+pN_x,\ell+qN_y})$ for p,q = 0,±1,±2, etc.; thus $\underline{B}_{j\ell}$ is formed, within the range on j and ℓ given in (13), by summing contributions from the primitive function $\underline{B}\ (\frac{k}{r}\,\underline{R}_{j\ell})$ to those non-negligible contributions, from the overlapping ends of these replicated functions, that arise from spatial-frequency values outside the assumed bandlimits K_x and K_y. Here, of course, K_x and K_y simply correspond to the quantities π/δ_x and π/δ_y, rather than to actual bandlimits. It should be noted that an analytical transition from the right-hand side of (12) to the right-hand side of (15) also involves a replication of $\underline{E}_t(\underline{P}_{mn})$, so that both the input and output functions in the finite Fourier transform relation (15) are doubly periodic with periods N_x and N_y. However, there is no overlap of the replicated input functions within the integration interval.

Although the finite-Fourier-transform relation (15) does not give a perfect evaluation of the aperture integral (2), it can be recognized that aperture-plane integration of a physical electric field will produce a virtually bandlimited function. This implies that the approximation

$$\underline{B}\left(\frac{k}{r}\,\underline{R}_{j\ell}\right) \simeq \underline{B}_{j\ell} \quad ; \quad \frac{-N_x}{2} < j < \frac{N_x}{2} \quad , \quad \frac{-N_y}{2} < \ell < \frac{N_y}{2} \tag{16}$$

will be valid provided K_x and K_y are large enough. This in turn implies that δ_x and δ_y must be sufficiently small.

One difficulty that arises with utilizing (15) is that fast Fourier transform (FFT) subroutines require non-negative integer values for the indices j, ℓ, m, and n. Fortunately, this difficulty is readily resolved using the periodic character of $\underline{B}_{j\ell}$ and $\underline{E}_{mn}$. Thus we have

$$\frac{1}{\delta_x \delta_y}\,\underline{B}_{j\ell} = \sum_{m=-\frac{N_x}{2}}^{\frac{N_x}{2}-1} \; \sum_{n=-\frac{N_y}{2}}^{\frac{N_y}{2}-1} \underline{E}_{mn}\, e^{-i2\pi(jm/N_x + \ell n/N_y)}$$

$$= \sum_{m=0}^{N_x-1} \; \sum_{n=0}^{N_y-1} \underline{E}_{m-\frac{N_x}{2},\, n-\frac{N_y}{2}}\, e^{i\pi(j+\ell)}\, e^{-i2\pi(jm/N_x+\ell n/N_y)} .$$

Finally, we obtain

$$\frac{1}{\delta_x \delta_y}\,\underline{B}_{j+\frac{N_x}{2},\,\ell+\frac{N_y}{2}} = e^{i\pi(j+\ell)} \sum_{m=0}^{N_x-1} \; \sum_{n=0}^{N_y-1} e^{-i\pi(m+\ell)}\,\underline{E}_{m-\frac{N_x}{2},\,n-\frac{N_y}{2}}\, e^{-i2\pi(jm/N_x+\ell n/N_y)} . \tag{17}$$

This result enables us to integrate over the aperture surface, $-T_x \leq x' \leq T_x$; $-T_y \leq y' \leq T_y$, where $T_x = \frac{1}{2} N_x\delta_x$ and $T_y = \frac{1}{2} N_y\delta_y$, using standard FFT-program format. The output of the FFT routine is a function of j and ℓ, which range over the values

$$0 \leq j \leq N_x - 1, \quad 0 \leq \ell \leq N_y - 1.$$

Consequently, since $\underline{B}_{j,\ell}$ is periodic in j and ℓ with periods N_x and N_y, respectively, by using (17) the FFT output can be plotted directly as though the range on j and ℓ were as given in (16).

Finally, we write down expressions that are compatible with the problem of computing the far field on the plane cuts. Thus, for the plane k_y = const. we have

$$\ell + \frac{1}{2} N_y = \frac{N_y}{S_y} \operatorname{Sin} E , \tag{18}$$

where $S_y = \frac{\lambda}{\delta_y}$ and E assumes a fixed value. We can now collapse the summation over y (i.e., reformulate the problem so as to just carry out an FFT on data formed by summing the input-matrix columns) to obtain

$$\frac{1}{\delta_x \delta_y} \underline{B}_{j+\frac{N_x}{2}, \ell+\frac{N_y}{2}} = e^{i\pi j} \sum_{m=0}^{N_x-1} e^{-i\pi m} \sum_{n=0}^{N_y-1} \{\underline{E}_{m-\frac{N_x}{2}, n-\frac{N_y}{2}} e^{-i(\frac{2\pi}{S_y} \operatorname{Sin} E)(n-\frac{N_y}{2})}\} e^{-i2\pi jm/N_x}. \tag{19}$$

Similarly, for the plane k_x = const. we have

$$j + \frac{1}{2} N_x = \frac{N_x}{S_x} \operatorname{Sin} a , \tag{20}$$

where $S_x = \frac{\lambda}{\delta_x}$ and a assumes a fixed value. We now collapse the summation over x (i.e., reformulate the problem so as to just carry out an FFT on data formed by summing the input-matrix rows) to obtain

$$\frac{1}{\delta_x \delta_y} \underline{B}_{j+\frac{N_x}{2}, \ell+\frac{N_y}{2}} = e^{i\pi \ell} \sum_{n=0}^{N_y-1} e^{-i\pi n} \sum_{m=0}^{N_x-1} \{\underline{E}_{m-\frac{N_x}{2}, n-\frac{N_y}{2}} e^{-i(\frac{2\pi}{S_x} \operatorname{Sin} a)(m-\frac{N_x}{2})}\} e^{-i2\pi \ell n/N_y}. \tag{21}$$

REFERENCES

[1] Collin, R. E., Zucker, F. J., Antenna Theory Part I, McGraw Hill (1969).

[2] (a) Kerns, D. M. Correction of near-field antenna measurements made with an arbitrary but known measuring antenna, Electronics Letters 6, pp. 346-347 (1970).

(b) Kerns, D. M. Plane wave scattering matrix theory of antennas and antenna-antenna interactions, NBSIR 78-890 (June 1978) (forthcoming as NBS Monograph 162).

[3] Yaghjian, A. D. Upper-bound errors in far-field parameters determined from planar near-field measurement, Part I: Analysis, NBS Tech Note 667 (Oct. 1975).

[4] Landau, H. J. Sampling, data transmission, and the nyquist rate, Proc. IEEE 55, #10, 1701-1706 (Oct. 1967).

[5] Newell, A. C. Planar near-field measurement techniques, (lecture notes for the NBS short course, Antenna Parameter Measurement by Near-Field Techniques, 7-11 July 1975).

[6] Cooley, J. W., Lewis, P. A. W., Welch, P. D. The fast Fourier transform algorithm and its applications, IBM Research Paper RC-1743, (Feb. 9, 1967).

[7] Johnson, C.C., Field and Wave Electrodynamics, McGraw Hill, 1965.

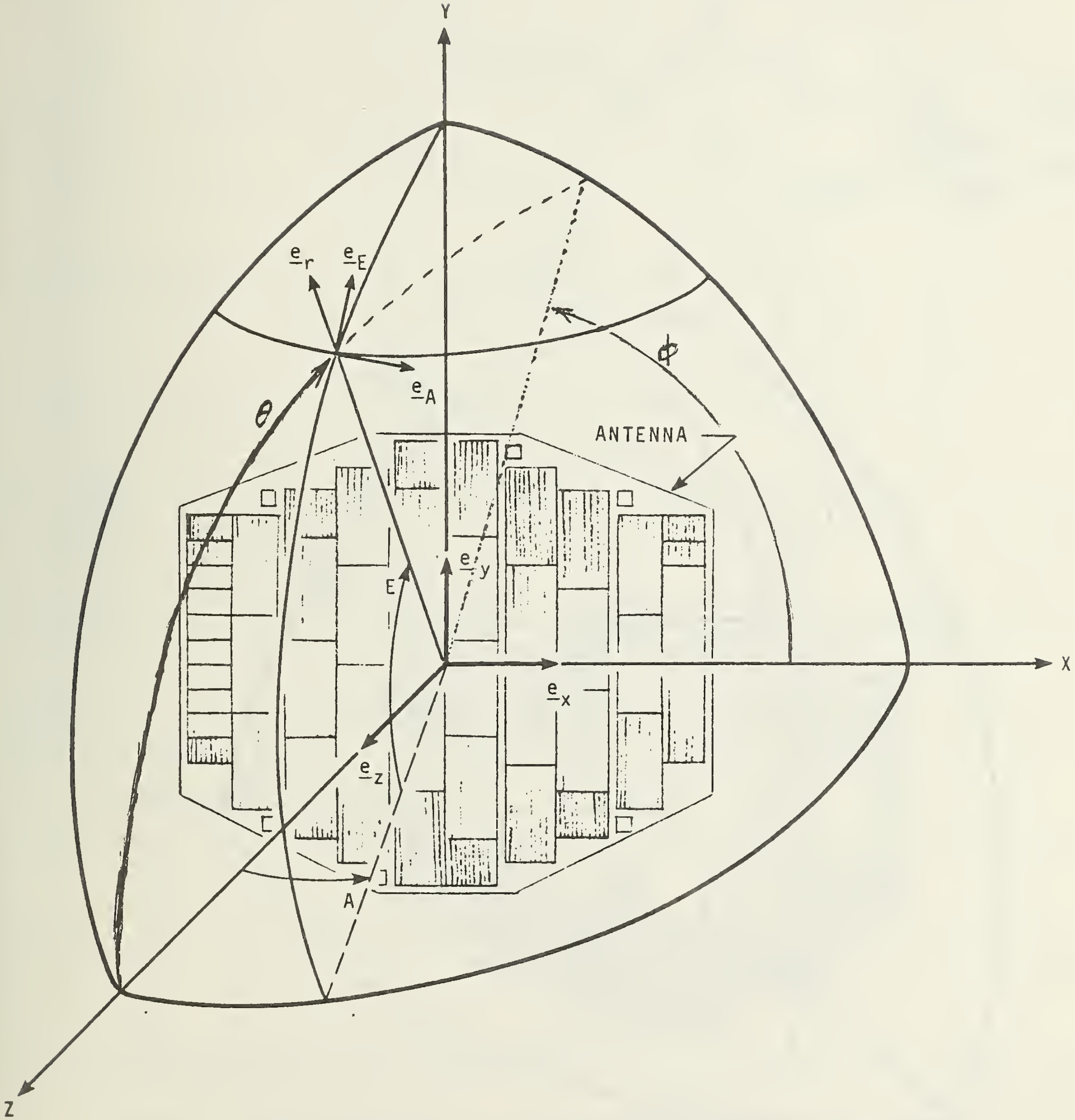

Figure 1. Antenna coordinate system using A and E spherical angles with y as the polar axis.

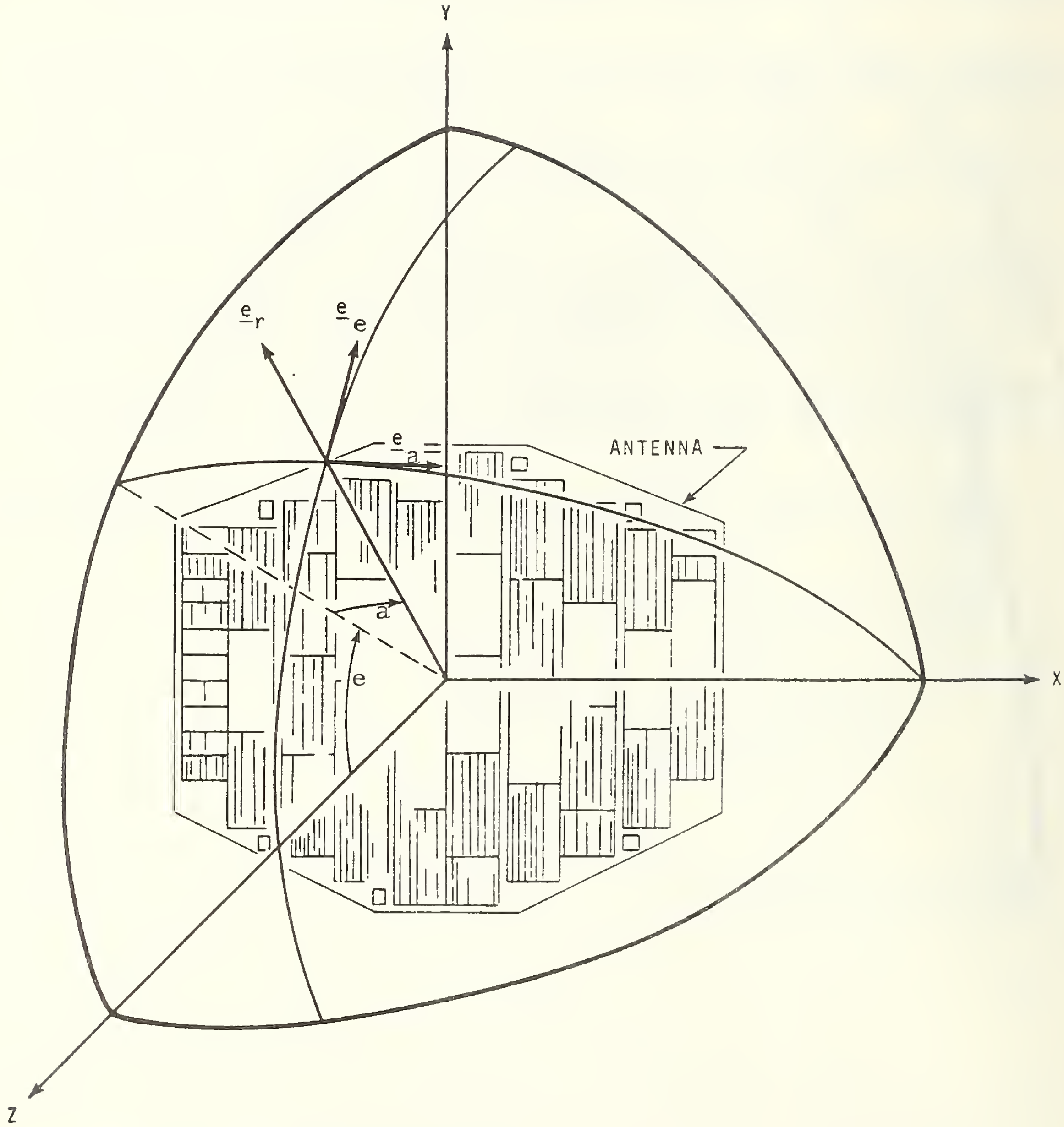

Figure 2. Antenna coordinate system using a and e spherical angles with x as the polar axis.

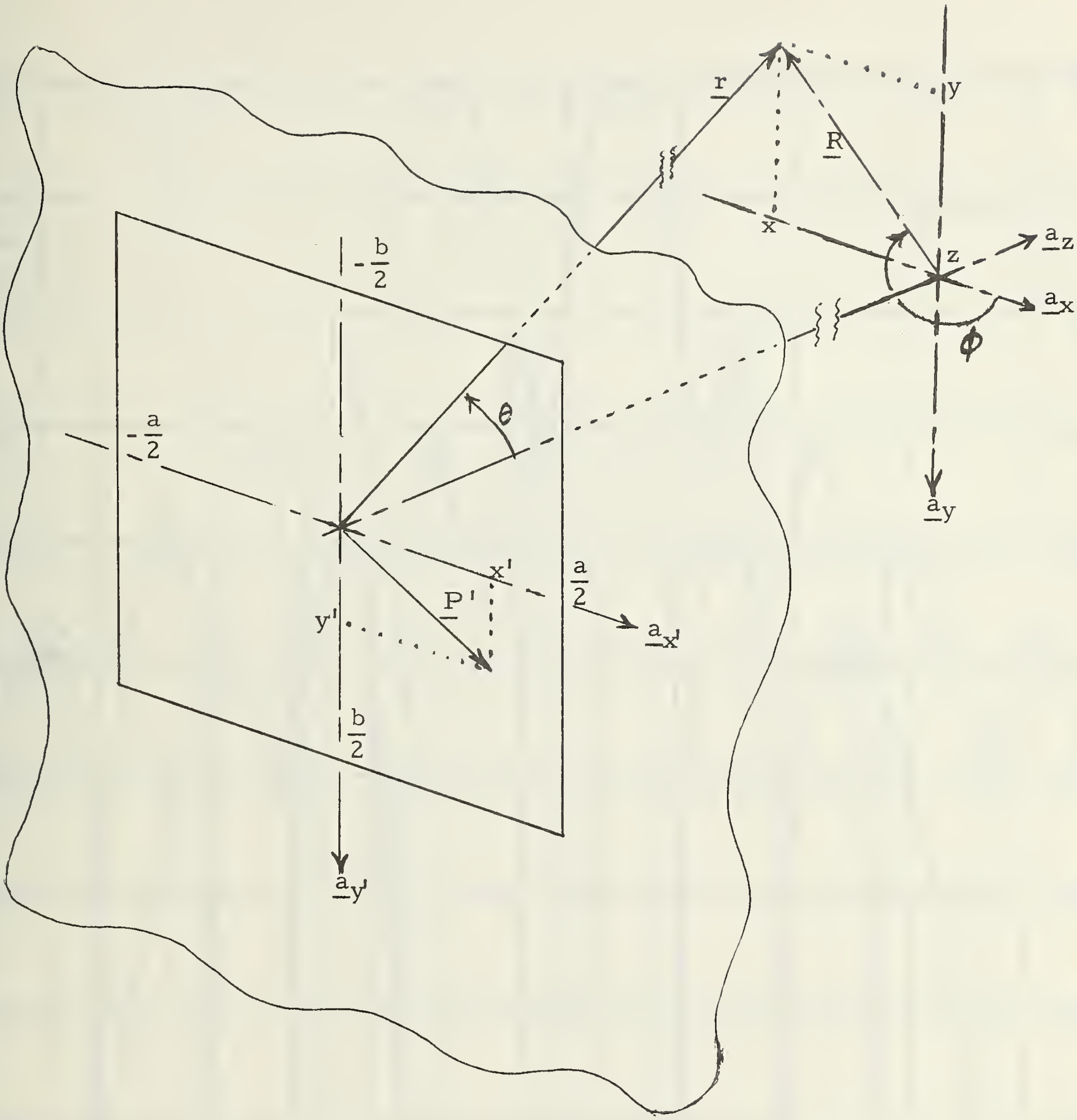

Figure 3. Problem geometry, showing the rectangular aperture within an aperture plane and the far-zone position coordinates.

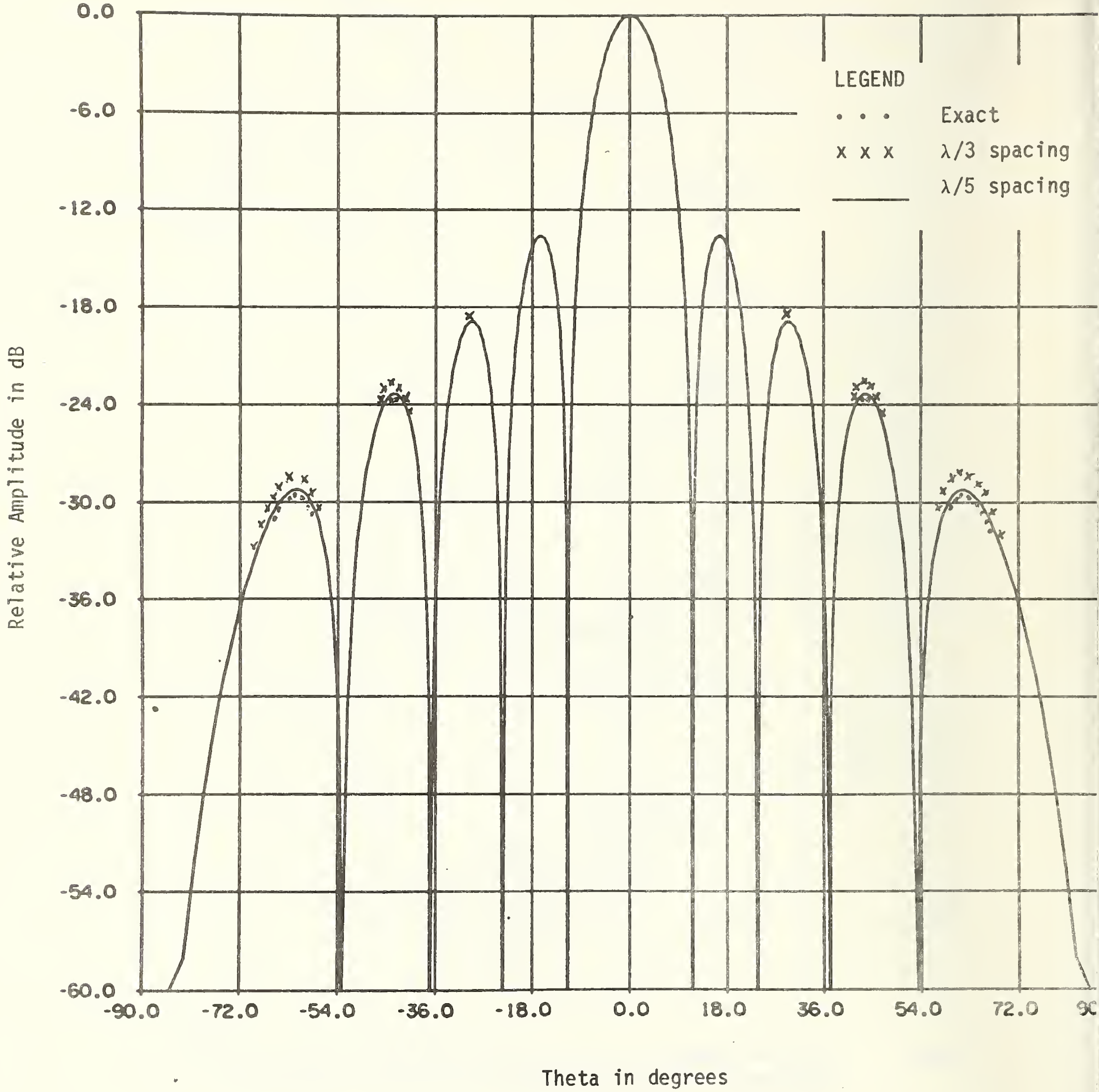

Figure 4. The principal plane cut $\phi = \frac{\pi}{2}$. PHI-component amplitude plot for a 5-wavelength-wide slit aperture (varied data-point-spacing increments).

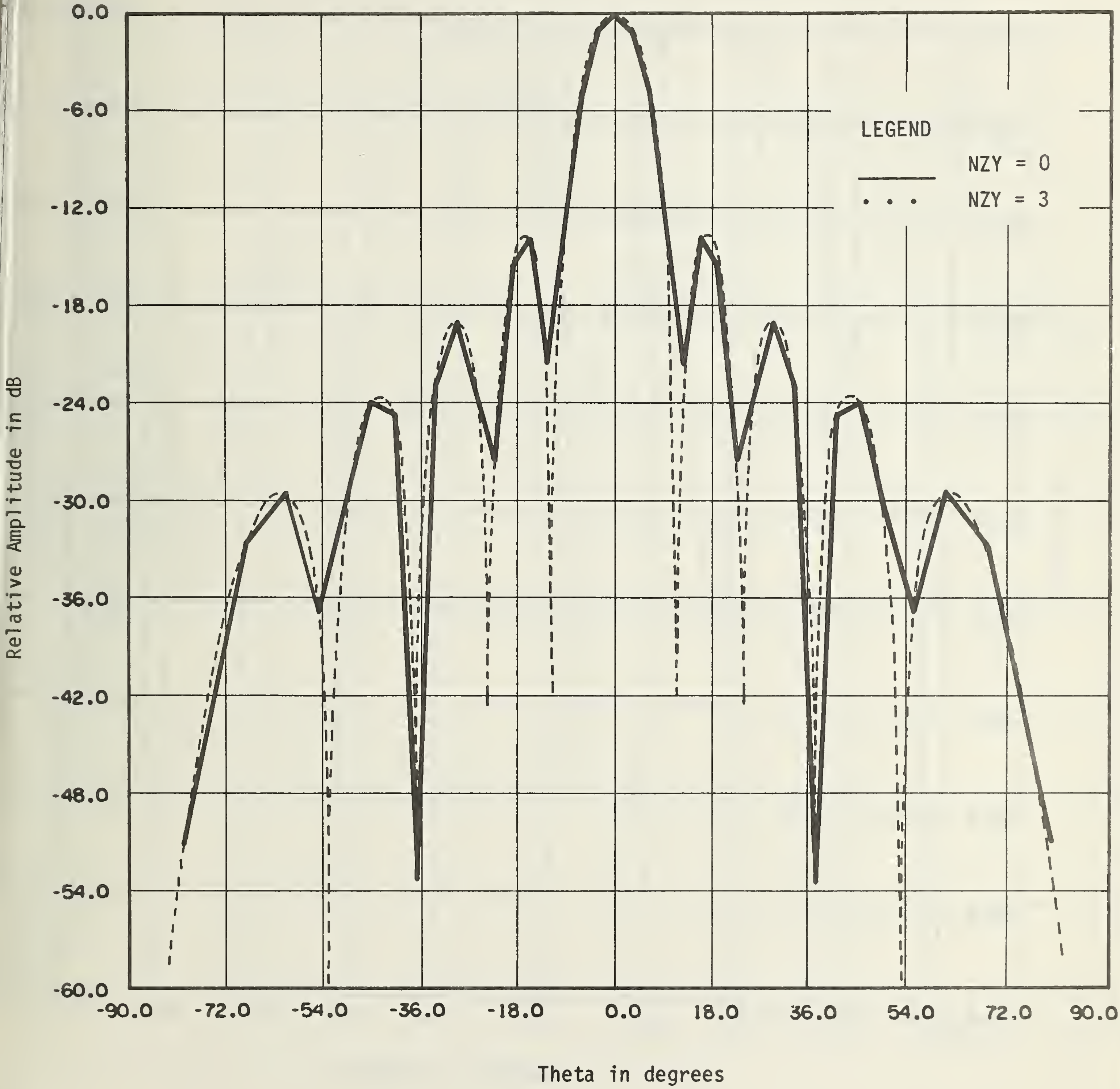

Figure 5. The principal plane cut $\phi = \frac{\pi}{2}$. PHI-component amplitude plot for a 5-wavelength-wide slit aperture (varied Fourier integration ranges).

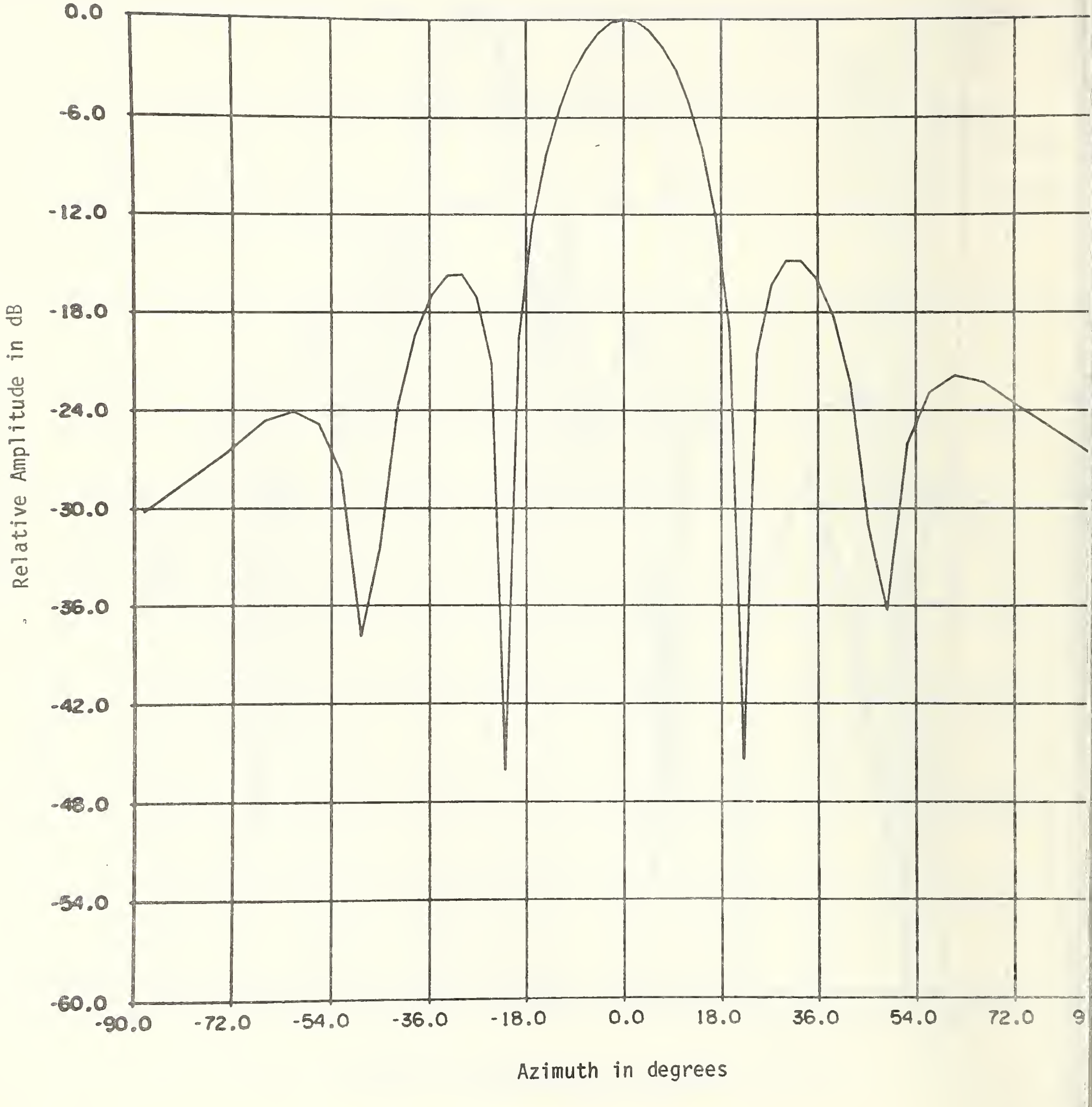

Figure 6. Antenna pattern magnitude for a uniformly excited circular aperture. Y-plane cut with constant elevation angle $E = 10^{0}$.

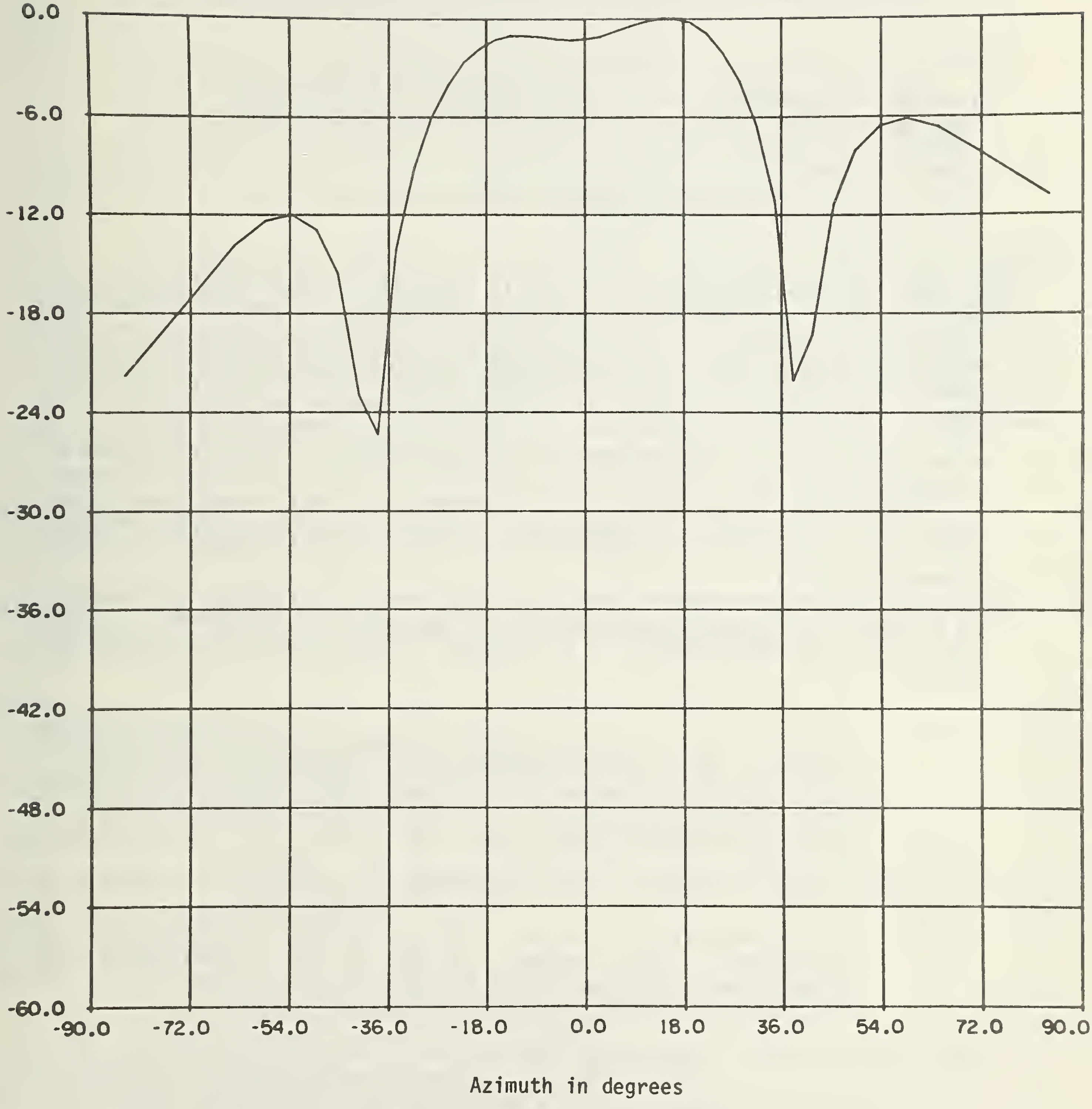

Figure 7. Antenna pattern magnitude for a uniformly excited circular aperture. Y-plane cut with constant elevation angle E = 30°.

Computer Program Listing for Computing Azimuth and Elevation Vector Components Along User-Specified X- and Y-Plane Cuts in the Far Field Arising from a Prescribed Rectangular-Aperture Field Distribution

```
      PROGRAM FAREA(INPUT,OUTPUT,TAPE1=INPUT)
      DIMENSION STOR(1024),EXPX(1024),EXPY(1024),EI(4096)
      DIMENSION THETA(1024),DATA(2048)
      COMPLEX STOR,EXPX,EXPY,EI,ZZ
      EQUIVALENCE (THETA,STOR),(EXPX,DATA)
      DATA (NMAX=1024),(NDIMEN=4096)
C
   94 FORMAT(4F7.3,1X2I3,2F5.1,5XA10)
   95 FORMAT(1H1,2X52HALPHA=X-COORDINATE APERTURE DIMENSION IN WAVELENGT
     OHS,9X51HBETA=Y-COORDINATE APERTURE DIMENSION IN WAVELENGTHS/3X57HS
     PX=RATIO OF WAVELENGTH TO X-COORDINATE DATA-POINT SPACING,4X57HSY=R
     QATIO OF WAVELENGTH TO Y-COORDINATE DATA-POINT SPACING/3X49HNZX=X-C
     ROORDINATE ZERO-FILL-REGION DOUBLING FACTOR,12X49HNZY=Y-COORDINATE
     SZERO-FILL-REGION DOUBLING FACTOR/3X64HA=AZIMUTH ANGLE (AT ZERO DEG
     TREES ELEVATION) OF AZIMUTH-PLANE CUT,14X40HE=ELEVATION ANGLE OF EL
     UEVATION-PLANE CUT/3X39HBANNER=10-CHARACTER PLOT IDENTIFICATION,4X
     V45HNNX,NNY=NUMBER OF TERMS IN X OR IN Y FFT SUMS,6X25HNNX=2**MMX
     W,  NNY=2**MMY////8H ALPHA =,F8.3,4X6HBETA =,F8.3,4X4HSX =,F7.3,4X
     X4HSY =,F7.3,4X5HNZX =,I3,4X5HNZY =,I3,4X3HA =,F5.1,4X3HE =,F5.1,4X
     Y7HBANNER.,A10,1H.//24X5HNNX =,I5,10X5HMMX =,I3,30X5HNNY =,I5,10X
     Z5HMMY =,I3///)
   96 FORMAT(1H0,25X70HARRAY DIMENSIONS TOO SMALL OR ILLEGAL OR ELSE E O
     1R A EQUALS 90 DEGREES//1X5HNNX =,I5,5X5HMMX =,I4,10X5HNNY =,I5,5X5
     2HMMY =,I4,10X23HCOMPUTED EI DIMENSION =,I6,10X3HE =,F5.1,5X3HA =,
     3F5.1)
C
C
C              COMPUTATION OF THE FAR FIELD ALONG SPECIFIED PERPENDICULAR
C              AZIMUTH- AND ELEVATION-PLANE CUTS, WHERE THE FAR-FIELD
C              EXCITATION CORRESPONDS TO A GIVEN NEAR-FIELD DISTRIBUTION
C              IN A SPECIFIED APERTURE.  IN ADDITION, THE FAR-FIELD
C              ANTENNA PATTERN ALONG THESE SPECIFIED CUTS IS ALSO COMPUTED.
C
C     THE DATA SPACING BETWEEN X- OR Y-COORDINATE ELEMENTS IS ASSUMED TO BE FIX
C
C              NMAX=DIMENSION OF ARRAYS STOR AND THETA.  DIMENSION OF ARRAY
C              DATA SHOULD EQUAL 2*NMAX.  NDIMEN IS THE DIMENSION OF ARRAY EI.
C              NDIMEN SHOULD EQUAL TWICE THE SUM OF THE X- AND Y-FFT DIMENSIONS
C              THE MAXIMUM USEFUL VALUE OF NDIMEN IS 4*NMAX.
C
C    INPUT PARAMETERS..  ALPHA=APERTURE WIDTH IN WAVELENGTHS (X-COORDINATE)
C                        BETA=APERTURE LENGTH IN WAVELENGTHS (Y-COORDINATE)
C                        SX=RATIO OF WAVELENGTH TO X-COORDINATE DATA-POINT SPACI
C                        SY=RATIO OF WAVELENGTH TO Y-COORDINATE DATA-POINT SPACI
C                        NZX=NUMBER OF EXTRA DOUBLINGS OF THE ZERO-FILL REGION
C                            FOR THE X-FAR-FIELD-COORDINATE FFT. (NORMALLY=1)
C                        NZY=NUMBER OF EXTRA DOUBLINGS OF THE ZERO-FILL REGION
C                            FOR THE Y-FAR-FIELD-COORDINATE FFT. (NORMALLY=1)
C                        A=AZIMUTH ANGLE (AT ZERO DEGREES ELEVATION),
C                            IN DEGREES, OF AZIMUTH-PLANE CUT.
C                        E=ELEVATION ANGLE, IN DEGREES, OF ELEVATION-PLANE CUT.
C                        BANNER = 10-CHARACTER GRAPH IDENTIFICATION
```

```
C
C           THE INPUT DATA IS ASSUMED TO BE SUPPLIED AS ROWS OF X-COORDINATE
C           DATA, EACH ROW CORRESPONDING TO A FIXED VALUE OF THE Y-COORDINATE.
C           ALL OF THE E-SUB-X DATA IS TO BE SUPPLIED FIRST, AND THEN THIS
C           INPUT DATA IS TO BE FOLLOWED BY ALL OF THE E-SUB-Y DATA.
C           THE PROGRAM OBTAINS THE INPUT DATA THROUGH REPEATED CALLS TO
C           THE USER-SUPPLIED SUBROUTINE GETARAY.
C
C          EACH CALL TO SUBROUTINE GETARAY RESULTS IN A ROW OF NX COMPLEX
C          DATA POINTS, EACH SUCCESSIVE CALL CORRESPONDING TO A ROW
C          SUCCESSIVELY FURTHER BELOW THE TOP OF THE APERTURE.  HERE, NX
C          IS EQUAL TO THE PRODUCT SX * ALPHA.  THE NUMBER OF ROWS, OR
C          THE NUMBER OF CALLS TO GETARAY, IS GIVEN BY NY = SY * BETA.
C
C           NUMBER OF DATA CARDS USED BY PROGRAM+SUBROUTINES = 3.
C
C           FOLLOWING THE COMPLETION OF A GIVEN FAR-FIELD COMPUTATION ALONG
C           SPECIFIED PERPENDICULAR AZIMUTH AND ELEVATION PLANES, THE PROGRAM
C           RETURNS TO THE BEGINING TO READ A NEW DATA CARD AND START ALL
C           OVER AGAIN ON A NEW FAR-FIELD COMPUTATION.  THIS TIME, HOWEVER,
C           THE PLOTTING ROUTINE PARAMETERS AND LEGEND THAT WERE READ IN, ON
C           THE FIRST TIME THROUGH THAT SUBROUTINE, ARE SIMPLY RECALLED AND
C           USED OVER AGAIN.  THE USER MUST MAKE SURE SUBROUTINE GETARAY CAN
C           SUPPLY A NEW SET OF NEAR-FIELD DATA UPON RESTARTING THE PROGRAM.
C
C
C
C     PARAMETER READ IN
    1 READ 94,ALPHA,BETA,SX,SY,NZX,NZY,A,E,BANNER
      IF(EOF(1).NE.0) GO TO 36
C
C     CALCULATION OF NX AND NY
      NX=0.5*SX*ALPHA $ NX=2*NX+1 $ NY=0.5*SY*BETA $ NY=2*NY+1
      SX=NX/ALPHA $ SY=NY/BETA
C
C     CALCULATION OF NNX AND NNY, ALONG WITH MMX AND MMY
      MMX=MMY=NNX=NNY=1
    2 MMX=MMX+1 $ NNX=2*NNX $ IF(NNX.GT.NX) 3,2
    3 NNX=2*NNX $ IF(NZX) 44,6,4
   44 NNX=NNX/2 $ MMX=MMX-1 $ GO TO 6
    4 DO 5 M=1,NZX $ MMX=MMX+1
    5 NNX=2*NNX
    6 MMY=MMY+1 $ NNY=2*NNY $ IF(NNY.GT.NY) 7,6
    7 NNY=2*NNY $ IF(NZY) 45,10,8
   45 NNY=NNY/2 $ MMY=MMY-1 $ GO TO 10
    8 DO 9 M=1,NZY $ MMY=MMY+1
    9 NNY=2*NNY
   10 MM=MAX0(MMX,MMY) $ NN=MAX0(NNX,NNY) $ ND=NNX+NNY $ J=2*ND
      IF(J.GT.NDIMEN.OR.NN.GT.NMAX.OR.MM.GT.14) 11,12
   11 PRINT 96,NNX,MMX,NNY,MMY,J,E,A $ CALL EXIT
C
C     INPUT PARAMETER PRINT-OUT
   12 PRINT 95,ALPHA,BETA,SX,SY,NZX,NZY,A,E,BANNER,NNX,MMX,NNY,MMY
C
C     COMPUTATION OF APERTURE-DATA POSITION WITHIN THE INTEGRATION INTERVAL
      FACTOR=1.0/(SX*SY) $ IOX=(NNX-NX+1)/2 $ IOY=(NNY-NY+1)/2+NNX
      SINE=SIN(0.017453292519943*E) $ SINA=SIN(0.017453292519943*A)
C
C     COMPUTATION OF APERTURE DATA-POINT SPACINGS
      DELX=SX/NNX $ LLX=SQRT(1.0-SINE**2)/DELX $ IF(LLX.EQ.0) GO TO 11
      DELY=SY/NNY $ LLY=SQRT(1.0-SINA**2)/DELY $ IF(LLY.EQ.0) GO TO 11
```

```
C   PRE-ZEROING OF INTEGRAL FIELD
       DO 13 M=1,J
   13 EI(M)=(0.,0.)

C   COMPUTATION OF CONSTANT EXPONENTIAL MULTIPLICATION FACTOR - Y-PLANE CUT
       ZZ=CEXP(CMPLX(0.,6.2831853071796*SINE/SY))
       J=(NY-1)/2 $ K=J+1 $ EXPY(K)=(1.0,0.)
       DO 14 M=1,J $ L=K-M $ LL=K+M
       EXPY(L)=ZZ*EXPY(L+1)
   14 EXPY(LL)=CONJG(EXPY(L))

C   COMPUTATION OF CONSTANT EXPONENTIAL MULTIPLICATION FACTOR - X-PLANE CUT
       ZZ=CEXP(CMPLX(0.,6.2831853071796*SINA/SX))
       J=(NX-1)/2 $ K=J+1 $ EXPX(K)=(1.0,0.)
       DO 15 M=1,J $ L=K-M $ LL=K+M
       EXPX(L)=ZZ*EXPX(L+1)
   15 EXPX(LL)=CONJG(EXPX(L))

C   DO LOOP TO COMPUTE TWO PLANE-CUT INTEGRALS OVER APERTURE-FIELD VECTOR
       DO 21 I=1,2 $ MX=ND*(I-1) $ IX=MX+IOX $ IY=MX+IOY

C   SET-UP OF INTEGRAND - COLLAPSE ON X- AND Y-PLANE CUTS
       DO 16 K=1,NY $ J=K+IY
       CALL GETARAY(STOR,NX)
       DO 16 L=1,NX $ M=L+IX
       EI(M)=EI(M)+EXPY(K)*STOR(L)
   16 EI(J)=EI(J)+EXPX(L)*STOR(L)

C   INITIALIZE X-COMPONENT PARAMETERS FOR FFT INTEGRATION
       NXY=NX $ MM=MMX $ NN=NNX $ J=0
   17 I2=IX+NXY $ I1=IX+1 $ IF(MOD(NXY/2,2).EQ.0) I1=I1+1

C   PRE-FFT MULTIPLICATION BY (-1)**M
       DO 18 M=I1,I2,2
   18 EI(M)=-EI(M)

C   FFT COMPUTATION
       CALL CCFFT2(EI(MX+1),MM,-FACTOR,-1)
       I1=MX+2 $ I2=MX+NN

C   POST-FFT MULTIPLICATION BY (-1)**J
       DO 19 M=I1,I2,2
   19 EI(M)=-EI(M)
       IF(J.EQ.0) 20,21

C   INITIALIZE Y-COMPONENT PARAMETERS FOR FFT INTEGRATION
   20 NXY=NY $ MM=MMY $ NN=NNY $ MX=MX+NNX $ IX=IY $ J=1 $ GO TO 17
   21 CONTINUE

C   INITIALIZE Y-PLANE-CUT PARAMETERS FOR VECTOR-COMPONENT AND PATTERN COMP.
       NN=NNX $ LL=LLX $ DEL=DELX $ MX=J=0 $ ANGLE=E
       V=SINE $ VV=V**2 $ COPE=SQRT(1.0-VV)

C   RESTRICT COMPUTATIONS TO POLAR SPATIAL-FREQUENCY ANGLES IN VISIBLE RANGE
   22 NXY=NN/2+1 $ IF(NXY.LT.LL+2) 23,24
   23 I1=MX+1 $ I2=MX+NN $ GO TO 25
   24 I1=NXY-LL+MX $ I2=I1+2*LL
   25 K=I1-1 $ NXY=NXY+MX
```

```
C  VECTOR-COMPONENT COMPUTATION
      DO 33 M=I1,I2 $ L=M-K $ LL=M+ND $ U=(M-NXY)*DEL $ COST=U**2+VV
      IF(J.EQ.0) 26,27
   26 COSE=COPE $ COST=SQRT(1.0-COST)/COSE $ SINT=U*V/COSE $ GO TO 3
   27 IF(U.EQ.1.0) 28,29
   28 COST=1.0 $ SINT=COSE=0. $ IF(V.EQ.0.) 30,11
   29 COSE=SQRT(1.0-U**2) $ COST=SQRT(1.0-COST)/COSE $ SINT=U*V/COSE
   30 THETA(L)=57.2957795130 82*ASIN(U/COPE)
      EI(M)=(0.,1.0)*(EI(M)*COSE+EI(LL)*SINT)
   33 EI(LL)=CMPLX(0.,COST)*EI(LL) $ LL=I1+ND

C  VECTOR-COMPONENT PLOTTING
      CALL PAMPLOT(L,EI(I1),THETA,DATA,2*NN,BANNER,2,ANGLE)
      CALL PAMPLOT(L,EI(LL),THETA,DATA,2*NN,BANNER,2,ANGLE)

C  ANTENNA-PATTERN COMPUTATION
      DO 34 M=I1,I2 $ L=M-K $ LL=M+ND
   34 DATA(L)=SQRT(REAL(EI(M)*CONJG(EI(M))+EI(LL)*CONJG(EI(LL))))

C  ANTENNA-PATTERN PLOTTING
      CALL PAMPLOT(L,0.,THETA,DATA,L,BANNER,1,ANGLE)
      IF(J.EQ.0) 35,1

C  INITIALIZE X-PLANE-CUT PARAMETERS FOR VECTOR-COMPONENT AND PATTERN COMP.
   35 NN=NNY $ LL=LLY $ DEL=DELY $ MX=NNX $ J=1 $ ANGLE=A
      V=SINA $ VV=V**2 $ COPE=SQRT(1.0-VV) $ GO TO 22
   36 CONTINUE
      END




      SUBROUTINE PAMPLOT(N,ER,XVALUE,DATA,N2,BANNER,MODE,ANGLE)
      DIMENSION ER(N),XVALUE(N),DATA(N2)
      COMPLEX ER
      DIMENSION HEAD(9),YMAX(2),YMIN(2),AMP(4),CAM(2)
      DIMENSION BP(4),ELAZ(2),COMP(6)
      DATA(COMP(M),M=1,6)/10H  AZIMUTH ,5HCOMP.,10HELEVATION ,5HCOMP.,10
     AHANTENNA PA,5HTTERN/,(AMP(M),M=1,4)/10H AMPLITUDE,6H PHASE,10H AMP
     B/PHASE,10H MAGNITUDE/,(ELAZ(M),M=1,2)/10HELEVATION=,10H  AZIMUTH=/
      DATA(CAM(1)=10H   REAL DA),(CAM(2)=10HCOMPLEX DA)
      DATA (K=0),(IA=1),(IB=1)
   99 FORMAT(2XI1,3F7.1,I3)
  100 FORMAT(4HAMP=,G8.2,A10,F3.0,A10,A5)
  101 FORMAT(4A10)
  102 FORMAT(1H0,10X1H*,9A10,1H*)
  103 FORMAT(1H1,15X25HPLOT ROUTINE JOB COMPLETE,10X32HNUMBER OF NON-TRI
     ZVIAL DATA PTS =,I5//26X31HNUMBER OF DATA POINTS PRINTED =,I5//)
  104 FORMAT(1H1,10X20HERROR EXIT - PAMPLOT//)
  105 FORMAT(1H0,3X3HNO=,I2,3X5HXMIN=,F7.3,3X5HXMAX=,F7.3,3X8HYMIN(1)=
     1,F7.3,3X8HYMAX(1)=,F7.3,3X8HYMIN(2)=,F7.3,3X8HYMAX(2)=,F7.3,3X,14H
     2PRINT SPACING=,I3//)
  106 FORMAT(1H0//25X44HALL INPUT DATA VALUES TO PAMPLOT ARE TRIVIAL//)
  107 FORMAT(5(F10.4,G10.4,F7.3))
  108 FORMAT(1H0,25X24HMAXIMUM DATA AMPLITUDE =,G13.6//)
  109 FORMAT(50X,A10,16HTA PLOTTING MODE//)
  110 FORMAT(8(F8.2,G9.3))
C
```

```
C          ROUTINE TO SET UP ARRAYS FOR PLOTTING AMPLITUDE AND PHASE
C          OF THE COMPLEX ARRAY ER.  DIMENSION OF ARRAY ER EQUALS N.
C          DATA PLOTTED VERSUS ABSCISSA ARRAY XVALUE.
C
C          COMPLEX INPUT DATA ASSUMED IF MODE=2.  IF MODE=1, HOWEVER,
C          IT IS ASSUMED THAT ONLY REAL-AMPLITUDE DATA IS TO BE PLOTTED.
C          THIS REAL-AMPLITUDE DATA WILL BE ASSUMED TO BE IN ARRAY DATA.
C          ARRAY DATA IS ALWAYS CHANGED BY THE PROGRAM, WHILE ARRAY ER
C          IS LEFT AS IT WAS ON ENTRY TO THE PROGRAM.  IF MODE=1, ARRAY
C          ER IS IGNORED.
C
C          ROUTINE ALSO PRINTS OUT THE AMPLITUDE, PHASE, AND ABSCISSA
C          VALUE AT A SELECTED NUMBER OF DATA POINTS.  USER SPECIFIES
C          THE POINT SPACING BETWEEN THE DATA POINTS TO BE PRINTED AND
C          THEN DATA POINTS WILL BE SELECTED FOR PRINTING UNIFORMLY
C          DISTRIBUTED OVER THE ABSCISSA RANGE XMIN TO XMAX CONSISTENT
C          WITH THE DATA-POINT SPACING SELECTED.  NOTE THAT THE
C          RECIPROCAL OF THE DATA-POINT SPACING PARAMETER EQUALS THE
C          PERCENTAGE OF DATA POINTS TO BE PLOTTED.
C
C          AMPLITUDE OF ARRAY ER (DATA) IS PLOTTED IN DB BELOW MAXIMUM ARRAY
C          VALUE.  THE DB RANGE IS SPECIFIED BY YMIN(1)    (MUST BE NEGATIVE).
C
C      PHASE OF ARRAY ER IS PLOTTED IN DEGREES BETWEEN YMAX(2) AND YMIN(2)
C
C          BOTH PHASE AND AMP PLOTTED ON SAME PLOT IF NO=2.
C          PHASE AND AMP PLOTTED SEPERATELY IF NO=1.
C
C          A USER-SUPPLIED 30-CHARACTER GRAPH LABEL IS PRINTED ON EACH PLOT.
C          THIS CAPTION IS OBTAINED BY READING ONE CAPTION CARD THE FIRST
C          TIME THE SUBROUTINE IS CALLED, WHILE THE SAME 30-CHARACTER LEGEND
C          WILL SIMPLY BE USED OVER AGAIN ON SUBSEQUENT CALLS TO THE ROUTINE
C          ALSO, A TEN-CHARACTER USER MESSAGE -BANNER- IS PRINTED ON EACH
C          GRAPH, ALONG WITH THE MAXIMUM DB-LEVEL OF THE DATA AND A 40-
C          CHARACTER DESCRIPTION OF WHAT TYPE OF GRAPH IS BEING PLOTTED.
C          THIS LATTER DISCRIPTION IS OBTAINED FROM THE PRESUMED CALLING
C          SEQUENCE OF THE MAIN PROGRAM, FAREA.  THE VALUE OF THE CONSTANT
C          AZIMUTH OR ELEVATION ANGLE -ANGLE- FOR THE GRAPH IS ALSO PRINTED.
C          A TOTAL LEGEND OF 90 CHARACTERS IS PRINTED ON EACH GRAPH.
C
C          USER SUPPLIES DUMMY ARRAY DATA FOR INTERMEDIATE STORAGE WHEN
C          MODE=2.  IN THIS CASE,THE DIMENSION OF ARRAY DATA EQUALS N2=2*N.
C          WHEN MODE=1, THE INPUT IS ASSUMED TO BE IN ARRAY DATA AND ITS
C          DIMENSION IS N2=N.
C
C
C         DATA CARDS READ ON FIRST CALL TO SUBROUTINE ONLY.
C  FIRST INPUT CARD..  NOO=NUMBER OF CURVES PER GRAPH (1 OR 2)
C                      XMIN,XMAX = SMALLEST AND GREATEST ABSCISSA POINT
C                      YMIN(1)=AMPLITUDE PLOT LOWEST DB VALUE
C                      J=POINT SPACING BETWEEN ADJACENT PRINTED DATA VALUES.
C                           (IF J=0 NONE OF THE DATA VALUES ARE PRINTED)
C         SUBROUTINE ALSO READS A 30-CHARACTER LEGEND CARD.
C
      IF(K.EQ.0) 13,14
   13 READ 99,NOO,XMIN,XMAX,YMIN(1),J
   14 YMIN(2)=0. $ YMAX(2)=360.
C                      YMIN(2)=PHASE PLOT SMALLEST NUMBER OF DEGREES (0.)
C                      YMAX(2)=PHASE PLOT LARGEST NUMBER OF DEGREES (360.)
      IF(N2.LT.2*N.AND.MODE.EQ.2.OR.N2.LT.N.AND.MODE.EQ.1) 1,2
    1 PRINT 104 $ CALL EXIT
```

```
 2 IF(XMAX.LE.XMIN.OR.YMIN(1).GE.0..OR.YMAX(2).LE.YMIN(2)) 1,3
 3 HEAD(9)=BANNER $ YMAX(1)=0. $ I=0 $ KL=KI=1 $ ANORM=0. $ NP=0
   PRINT 105,NOO,XMIN,XMAX,YMIN(1),YMAX(1),YMIN(2),YMAX(2),J
   PRINT 109,CAM(MODE)
   IF(MODE.EQ.1) GO TO 22 $ IF(MODE.NE.2) GO TO 1 $ NO=NOO $ LL=0
   DO 6 M=1,N
   IF(XVALUE(M).LE.XMIN) KI=M $ IF(XVALUE(M).LE.XMAX) KL=M
   DATA(M)=CABS(ER(M))
   IF(DATA(M).EQ.0.) 4,5
 4 DATA(M+N)=YMIN(2) $ GO TO 6
 5 I=I+1
   IF(DATA(M).GT.ANORM) ANORM=DATA(M)
   B=ATAN2(AIMAG(ER(M)),REAL(ER(M)))
   IF(B.LT.0.) B=B+6.2831853071796
   DATA(M+N)=57.295779513082*B
 6 CONTINUE
   IF(I.EQ.0) 7,8
 7 PRINT 106 $ GO TO 21
 8 PRINT 108,ANORM
   IF(J.LE.0) 29,25
25 IF(J.EQ.1) 26,27
26 I1=KI $ I2=KL $ NP=KL-KI+1 $ GO TO 28
27 M=KL-KI+1 $ L=(M-1)/2
   NP=L/J $ L=J*NP $ I2=M/2+KI $ I1=I2-L $ I2=I2+L $ NP=2*NP+1
28 IF(MODE.EQ.1) 30,31
30 PRINT 110,(XVALUE(M),DATA(M),M=I1,I2,J) $ GO TO 29
31 PRINT 107,(XVALUE(M),DATA(M),DATA(M+N),M=I1,I2,J)
29 ANORM=20.0*ALOG10(ANORM)
   ENCODE(40,100,BB) ANORM,ELAZ(IB),ANGLE,COMP(IA),COMP(IA+1)
   DECODE(40,101,BB) (HEAD(M),M=1,4)
   DO 11 M=KI,KL
   IF(DATA(M).EQ.0.) 9,10
 9 DATA(M)=YMIN(1) $ GO TO 11
10 DATA(M)=20.0*ALOG10(DATA(M))-ANORM
11 CONTINUE
   IF(K.EQ.0) 15,16
15 K=1 $ READ 101,(HEAD(M),M=6,8)
16 HEAD(5)=AMP(3) $ L=1 $ B=1H. $ IF(NO-2) 18,19,1
17 PRINT 102,(HEAD(M),M=1,9)
18 HEAD(5)=AMP(L+LL) $ B=1H+
19 J0=(L-1)*N+KI $ M=L $ IF(NO.EQ.2) M=MOD(L,2)+1 $ I2=(M-1)*N+KI
   CALL CRTPLOT(XVALUE(KI),DATA(J0),XMAX,XMIN,YMAX,YMIN,KL-KI+1,HEAD,
  X L,NO)
   CALL PLT120P(XVALUE(KI),DATA(I2),XMAX,XMIN,YMAX(M),YMIN(M),KL-KI+1
  Y ,B,L,NO) $ IF(L.EQ.2.OR.MODE.EQ.1) 21,20
20 L=2 $ B=1H+ $ IF(NO-1) 1,17,19
21 PRINT 102,(HEAD(M),M=1,9)
   PRINT 103,I,NP
   GO TO 32
22 DO 24 M=1,N
   IF(XVALUE(M).LE.XMIN) KI=M $ IF(XVALUE(M).LE.XMAX) KL=M
   IF(DATA(M).EQ.0.) 24,23
23 I=I+1
   IF(DATA(M).GT.ANORM) ANORM=DATA(M)
24 CONTINUE
   NO=1 $ LL=3 $ IF(I.EQ.0) 7,8
32 IF(MODE.EQ.1) 33,34
33 IB=MOD(IB,2)+1 $ IA=1 $ GO TO 35
34 IA=IA+2
35 RETURN
   END
```

```
      SUBROUTINE PLT120R(X, Y, XMAX, XMIN, YMAX, YMIN, LAST, ISYMBOL, NOPLT1
     1, MOST)                                                           PLT1
C     MODIFIED 11/4/68                                                  PLT1
      DIMENSION X(1), Y(1), ZX(13), GRAPH(121, 51)                      PLT1
      INTEGER GRAPH, COLUMNS, BLANK, BORDER                             PLT1
      DATA (LINES = 51), (COLUMNS = 121)                                PLT1
      YLAR = YMAX                                                       PLT1
      YSMA = YMIN                                                       PLT1
      YSCALE = (YLAR - YSMA) / (LINES - 1.)                             PLT1
      IF (NO .NE. 1) GO TO 190                                          PLT1
      KMAX = COLUMNS / 10 + 1                                           PLT1
      XLAR = XMAX                                                       PLT1
      XSMA = XMIN                                                       PLT1
      BORDER = 1HI                                                      PLT1
      BLANK = 1H                                                        PLT1
      MATRIX = COLUMNS * LINES                                          PLT1
      IF (MATRIX .LT. 1) GO TO 120                                      PLT1
      DO 110 I = 1, MATRIX                                              PLT1
  110 GRAPH(I) = BLANK                                                  PLT1
  120 CONTINUE                                                          PLT1
      IF (LINES .LT. 1) GO TO 140                                       PLT1
      DO 130 I = 1, LINES                                               PLT1
  130 GRAPH(1, I) = GRAPH(COLUMNS, I) = BORDER                          PLT1
  140 CONTINUE                                                          PLT1
      IF (COLUMNS .LT. 1) GO TO 160                                     PLT1
      DO 150 I = 1, COLUMNS                                             PLT1
  150 GRAPH(I, 26) = 1H.                                                PLT1
  160 CONTINUE                                                          PLT1
      XSCALE = (XLAR - XSMA) / (COLUMNS - 1.)                           PLT1
      IF (KMAX .LT. 1) GO TO 180                                        PLT1
      DO 170 K = 1, KMAX                                                PLT1
  170 ZX(K) = 10. * FLOAT(K - 1) * XSCALE + XSMA                        PLT1
  180 CONTINUE                                                          PLT1
  190 IF (LAST .LT. 1) GO TO 250                                        PLT1
      DO 240 I = 1, LAST                                                PLT1
      IF (X(I) .GT. XLAR .OR. X(I) .LT. XSMA) GO TO 240                 PLT1
      IF (Y(I) .GT. YLAR .OR. Y(I) .LT. YSMA) GO TO 240                 PLT1
      IX = (X(I) - XSMA) / XSCALE + 1.5                                 PLT1
      IY = (Y(I) - YSMA) / YSCALE + .5                                  PLT1
      IY = LINES - IY                                                   PLT1
      GRAPH(IX, IY) = ISYMBOL                                           PLT1
  240 CONTINUE                                                          PLT1
  250 CONTINUE                                                          PLT1
      IF (NO .NE. MOST) RETURN                                          PLT1
      PRINT 1500                                                        PLT1
      YES = YLAR + YSCALE                                               PLT1
      IF (LINES .LT. 1) GO TO 270                                       PLT1
      DO 260 I = 1, LINES                                               PLT1
      YES = YES - YSCALE                                                PLT1
      PRINT 1510, YES, (GRAPH(J, I), J = 1, COLUMNS)                    PLT1
  260 CONTINUE                                                          PLT1
  270 CONTINUE                                                          PLT1
      PRINT 1520                                                        PLT1
      PRINT 1530, ZX                                                    PLT1
      RETURN                                                            PLT1
 1500 FORMAT  (1H1,9X,24(5HI....)1HI)                                   PLT1
 1510 FORMAT  (1H ,E8.2,1X,121A1)                                       PLT1
 1520 FORMAT  (1H ,9X,24(5HI....)1HI)                                   PLT1
 1530 FORMAT  (1H ,2X,13(1X,F9.3))                                      PLT1
      END                                                               PLT1
```

```
      SUBROUTINE CCFFT2(C,M,SC,NX)
C        DISCRETE COMPLEX FAST FOURIER TRANSFORM.
C        L. DAVID LEWIS,  NOAA-SEL,  730315
C        CALL CCFFT2(C,M,SC,NX)
C        COMPLEX INPUT C(J) IN NORMAL SEQUENCE.
C        COMPLEX OUTPUT C(K) IN NORMAL SEQUENCE.
C        SEQUENCE LENGTH IS N = 2**M
C        SC IS REAL SCALING MULTIPLIER.
C        NX IS THE SIGN OF THE EXPONENT IN THE TRANSFORM DEFINITION.
C        USES CFFTRCD AND REVBIND, Q. V.
      DIMENSION C(2)
      CALL REVBIND(C(1),C(2),M,2)
      CALL CFFTRCD(C(1),C(2),M,SC,NX,2)
      RETURN
      END
      SUBROUTINE REVBIND (A,B,MM,NDEL)
C        CALL REVBIND(A,B,M,ND)
C        REVERSIBLE PERMUTATION OF ARRAYS A AND B
C        FROM NORMAL SEQUENCE TO REVERSE BINARY SEQUENCE,
C        OR VICE VERSA.
C        ND IS SUBSCRIPT INCREMENT FOR A, B.
C        SEQUENCE LENGTH IS N = 2**M
C        WRITTEN BY L. DAVID LEWIS AND MARIE WEST, ESSA.
C        MODIFIED FROM, OR INSPIRED BY THE ALGOL PROCEDURE
C           REVERSEBINARY,              BY R. C. SINGLETON, SRI.
      DIMENSION A(16384),B(16384)
      COMMON /FFTCC/ M,JD(15),ST(15)
      DIMENSION JC(15)
C     CALL ROLLCALL(48HE/4/71                                         )
      M=MM $ CALL FFTC $ IF(M.LE.1) RETURN
    1 ND=NDEL
      DO 10 LC=1,15
   10 JC(LC)=JD(LC)*ND
      N=JC(M+1)-ND+1 $ NP=N+1
      K=1 $ I=ND+1 $ J=N-ND
    2 LC=M
    3 K=K+JC(LC) $ JC(LC)=-JC(LC)
      IF(JC(LC).LT.0) GO TO 4
      IF(LC.EQ.2) RETURN
      LC=LC-1 $ GO TO 3
    4 IF(K.LE.I.OR.J.LT.K) GO TO 5
      T=A(I) $ A(I)=A(K) $ A(K)=T
      T=B(I) $ B(I)=B(K) $ B(K)=T
      IF(J.EQ.K) GO TO 5
      KK=NP-K
      T=A(KK) $ A(KK)=A(J) $ A(J)=T
      T=B(KK) $ B(KK)=B(J) $ B(J)=T
    5 I=I+ND $ J=J-ND $ GO TO 2
      END
      SUBROUTINE CFFTRCD(A,B,MM,SCALE,NEXP,NDEL)
C        DISCRETE COMPLEX FAST FOURIER TRANSFORM.
C        CALL CFFTRCD(A,B,M,SC,NX,ND)
C        OUTPUT A(K) + I*B(K) IN NORMAL SEQUENCE.
C        ND IS SUBSCRIPT INCREMENT FOR A, B.
C        SEQUENCE LENGTH IS N = 2**M
C        SC IS REAL SCALING MULTIPLIER.
C        NX IS THE SIGN OF THE EXPONENT IN THE TRANSFORM DEFINITION.
C        INNER LOOP SINES AND COSINES COMPUTED
C           RECURSIVELY BY SINGLETON≠S 2ND-DIFFERENCE ALGORITHM,
C           INITIALIZED FROM A DATA TABLE.
```

```
C         WRITTEN BY L. DAVID LEWIS AND MARIE WEST, ESSA.
C         MODIFIED FROM, OR INSPIRED BY THE ALGOL PROCEDURE
C            REVERSEFOURIERC,                BY R. C. SINGLETON, SRI.
      DIMENSION A(16384),B(16384)
      COMMON /FFTCC/ M,JD(15),S(15)
C     CALL ROLLCALL(48H6/4/71                                       )
      M=MM $ CALL FFTC
      NE=NDEL
      N=JD(M+1)
      K = N/4
      ND = K
      JSPAN=ND
      ND2=2*ND
      NLIM=1+(N-1)*ND $ NN=NLIM-1
       SC = SCALE
      IF(ABS(SC-1.).LT.1.E-10) GO TO     7
6     DO 5 JC=1,NLIM,ND
       A(JC) = SC    * A(JC)
       B(JC) = SC    * B(JC)
 5     CONTINUE
7     IF(M.EQ.0) RETURN
      DO 10 KK=1,NLIM,ND2
      KS=KK+ND
            RE = A(KK) - A(KS)
            A(KK) =           A(KK) + A(KS)
            A(KS) =           RE
            FIM   = B(KK) - B(KS)
            B(KK) =           B(KK) + B(KS)
            B(KS) = FIM
   10 CONTINUE
      IF(M.EQ.1) RETURN
       EXPS = ISIGN(1,NEXP)
      DO 90 JB = 2,M
       SD = -S(JB-1)
       CD = 2.* S(JB) * S(JB)
       R = -2. * CD
            CN = 1.
            CM = 0.
            SN = 0.
            JJ = 0
            KK = 1
             SM = +EXPS
   12       JSPANH = JSPAN

            JSPAN = JSPAN + JSPAN
   20        KS = KK + JSPAN
            RE = CN * A(KS) - SN * B(KS)
            FIM = SN * A(KS) + CN * B(KS)
            A(KS) = A(KK) - RE
            A(KK) = A(KK) + RE
            B(KS) = B(KK) - FIM
            B(KK) = B(KK) + FIM
            KK = KK + JSPANH
            KS = KS + JSPANH
            FIM = SM * A(KS) + CM * B(KS)
            RE = CM * A(KS) - SM * B(KS)
            A(KS) = A(KK) - RE
            A(KK) = A(KK) + RE
            B(KS) = B(KK) - FIM
            B(KK) = B(KK) + FIM
```

```
      KK = KS + JSPANH
      IF(KK.LT.NLIM) GO TO 20
30    KK = KK - NN
      JJ = JJ + K
      IF(JJ.GE.NQ) GO TO 80
40    CD = P * CN + CD
      CN = CD + CN
      SM=CN*EXPS
      SD = R * CM + SD
      CM = SD + CM
      SN=-CM*EXPS
      GO TO 20
80    K = K/2
90    CONTINUE
      RETURN
      END
      SUBROUTINE FFTC

        COMMON SUBROUTINE FOR FFT SUBROUTINES.
        JC IS POWERS-OF-TWO ARRAY..  JC(M)=2**(M-1)
        ST IS SINE ARRAY.. ST(M)=SIN(PI/(2**M))
        M IS TESTED FOR PROPER INPUT RANGE, 0.LE.M.LE.14.

      COMMON /FFTCC/ M,JC(15),ST(15)
      DATA (JC=1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,16384)
      DATA (ST   =           1.00000000000E+000,    7.07106781187E-001,
     1                       3.82683432365E-001,    1.95090322016E-001,
     2                       9.80171403296E-002,    4.90676743274E-002,
     3                       2.45412285229E-002.    1.22715382857E-002,
     4                       6.13588464915E-003,    3.06795676297E-003,
     5                       1.53398018629E-003,    7.66990318743E-004,
     6                       3.83495187571E-004,    1.91747597311E-004,
     7                       9.58737990960E-005)
      IF(M.LT.0.OR.M.GT.14) CALL SYSTEM(52,10HM ILLEGAL.)
      IF(M.LT.0.OR.M.GT.14) CALL Q8QERROR(0,10HM ILLEGAL.)
      RETURN
      END
```

APPENDIX C

Computer Program Modifications to obtain Theta and Phi Vector-Component Output Instead of Azimuth and Elevation Vector-Component Output

I. Required changes to VECTOR-COMPONENT-COMPUTATION section of program FAREA; replace DO loop, DO 33 M-I1, I2 thru statement 33, with the following:

```
C   VECTOR-COMPONENT COMPUTATION
      DO 33 M=I1,I2 $L=M-K $LL=M+ND $U=(M-NXY)*DEL $IF(VV.EQ.0.) 46,47
   46 COST=U**2 $ SINTH=U $ IF(SINTH.EQ.0.) 26,29
   26 IF(J.EQ.0) 27,28
   27 COSP=1.0 $ SINP=0. $ GO TO 32
   28 SINP=1.0 $ COSP=0. $ GO TO 32
   47 COST=U**2+VV $ SINTH=SQRT(COST)
   29 IF(J.EQ.0) 30,31
   30 COSP=U/SINTH $ SINP=V/SINTH $ GO TO 32
   31 SINP=U/SINTH $ COSP=V/SINTH
   32 THETA(L)=57.295779513082*ASIN(U/COPE)
      ZZ=(0.,1.0)*(EI(M)*COSP+EI(LL)*SINP)
      EI(LL)=CMPLX(0.,SQRT(1.0-COST))*(-EI(M)*SINP+EI(LL)*COSP)
   33 EI(M)=ZZ $ LL=I1+ND
```

II. Subroutine PAMPLOT DATA-statement change; replace the first DATA statement card with the following:

```
      DATA(COMP(M),M=1,6)/10HTHETA COMP,5HONENT,10H  PHI COMP,5HONENT,10
```

Computer Program for Computing Principal-Plane-Cut Far Fields

(This program ignores the jump change in sign of the electric-field vector when passing through the coordinate-system origin. Refer to equations (10) and (11) of Appendix A for details).

```
      PROGRAM FARE(INPUT,OUTPUT,TAPE1=INPUT)
      DIMENSION STOR(1024),E(4096),THETA(1024),DATA(2048)
      EQUIVALENCE (THETA,STOR)
      COMPLEX STOR,E
      DATA (NMAX=1024),(NDIMEN=4096)
   94 FORMAT(4F7.0,2XA10,1X2I2)
   95 FORMAT(1H1,2X52HALPHA=X-COORDINATE APERTURE DIMENSION IN WAVELENGT
     THS,9X51HBETA=Y-COORDINATE APERTURE DIMENSION IN WAVELENGTHS/3X57HS
     UX=RATIO OF WAVELENGTH TO X-COORDINATE DATA-POINT SPACING,4X57HSY=R
     VATIO OF WAVELENGTH TO Y-COORDINATE DATA-POINT SPACING/3X39HBANNER=
     W10-CHARACTER PLOT IDENTIFICATION,4X39HNZX,NZY=ZERO-FILL AREA DOUBL
     XING FACTORS,4X44HNNX,NNY=NUMBER OF TERMS IN X- AND Y-FFT SUMS///8H
     Y ALPHA =,F8.3,4X6HBETA =,F8.3,4X4HSX =,F7.3,4X4HSY =,F7.3,4X7HBANN
     ZER.,A10,1H.,4X4HNZX=,I2,3X4HNZY=,I2,7X5HNNX =,I5,4X5HNNY =,I5//)
   96 FORMAT(1H0,25X37HARRAY DIMENSIONS TOO SMALL OR ILLEGAL//1X5HNNX =,
     AI5,3X20HNNX=2**MMX   ,  MMX =,I3,15X5HNNY =,I5,3X20HNNY=2**MMY   ,
     BMMY =,I3,15X22HCOMPUTED E-DIMENSION =,I6)
   97 FORMAT(25X5HMMX =,I3,25X5HMMY =,I3//)


            COMPUTATION OF THE FAR-FIELD PRINCIPAL-PLANE CUTS OF THE
            THETA- AND PHI-COMPONENTS OF THE ELECTRIC-FIELD VECTOR DUE
            TO A GIVEN NEAR-ZONE ELECTRIC-FIELD DISTRIBUTION IN A
            SPECIFIED APERTURE, ALONG WITH COMPUTATION OF THE FAR-FIELD
            PATTERNS FOR THESE PRINCIPAL-PLANE CUTS.

  THE DATA SPACING BETWEEN X- OR Y-COORDINATE ELEMENTS IS ASSUMED TO BE FIXED

            NMAX=DIMENSION OF ARRAYS STOR AND THETA.  DIMENSION OF ARRAY
            DATA SHOULD EQUAL 2*NMAX.  NDIMEN IS THE DIMENSION OF ARRAY E.
            NDIMEN SHOULD EQUAL TWICE THE SUM OF THE X- AND Y-FFT DIMENSIONS.
            THE MAXIMUM USEFUL VALUE OF NDIMEN IS 4*NMAX.

 INPUT PARAMETERS..  ALPHA=APERTURE WIDTH IN WAVELENGTHS (X-COORDINATE)
                     BETA=APERTURE LENGTH IN WAVELENGTHS (Y-COORDINATE)
                    SX=RATIO OF WAVELENGTH TO X-COORDINATE DATA-POINT SPACING
                    SY=RATIO OF WAVELENGTH TO Y-COORDINATE DATA-POINT SPACING
                     BANNER = 10-CHARACTER GRAPH IDENTIFICATION
                     NZX=NUMBER OF EXTRA DOUBLINGS OF THE ZERO-FILL REGION
                          FOR THE X-FAR-FIELD-COORDINATE FFT. (NORMALLY=1)
                     NZY=NUMBER OF EXTRA DOUBLINGS OF THE ZERO-FILL REGION
                          FOR THE Y-FAR-FIELD-COORDINATE FFT. (NORMALLY=1)

            THE INPUT DATA IS ASSUMED TO BE SUPPLIED AS ROWS OF X-COORDINATE
            DATA, EACH ROW CORRESPONDING TO A FIXED VALUE OF THE Y-COORDINATE.
            ALL OF THE E-SUB-X DATA IS TO BE SUPPLIED FIRST, AND THEN THIS
            INPUT DATA IS TO BE FOLLOWED BY ALL OF THE E-SUB-Y DATA.
            THE PROGRAM OBTAINS THE INPUT DATA THROUGH REPEATED CALLS TO
            THE USER-SUPPLIED SUBROUTINE GETARAY.
```

```
C              EACH CALL TO SUBROUTINE GETARAY RESULTS IN A ROW OF NX COMPLEX
C              DATA POINTS, EACH SUCCESSIVE CALL CORRESPONDING TO A ROW
C              SUCCESSIVELY FURTHER BELOW THE TOP OF THE APERTURE.  HERE, NX
C              IS EQUAL TO THE PRODUCT SX * ALPHA.  THE NUMBER OF ROWS, OR
C              THE NUMBER OF CALLS TO GETARAY, IS GIVEN BY NY = SY * BETA.
C
C               NUMBER OF DATA CARDS USED BY PROGRAM+SUBROUTINES = 3.
C
C               UPON COMPLETING A GIVEN SET OF FAR-FIELD PRINCIPAL-PLANE
C               COMPUTATIONS, THE COMPUTER PROGRAM WILL THEN RETURN TO THE
C               BEGINING OF THE PROGRAM TO READ A NEW DATA CARD AND START ALL
C               OVER AGAIN ON A NEW FAR-FIELD COMPUTATION.  THIS TIME, HOWEVER,
C               THE PLOTTING ROUTINE PARAMETERS AND LEGEND THAT WERE READ IN, ON
C               THE FIRST TIME THROUGH THAT SUBROUTINE, ARE SIMPLY RECALLED AND
C               USED OVER AGAIN.  THE USER MUST MAKE SURE SUBROUTINE GETARAY CAN
C               SUPPLY A NEW SET OF NEAR-FIELD DATA UPON RESTARTING THE PROGRAM.
C

C   PARAMETER READ-IN
     1 READ 94,ALPHA,BETA,SX,SY,BANNER,NZX,NZY
       IF(EOF(1).NE.0) GO TO 28

C   CALCULATION OF NX AND NY
       NX=0.5*SX*ALPHA $ NX=2*NX+1 $ NY=0.5*SY*BETA $ NY=2*NY+1
       SX=NX/ALPHA $ SY=NY/BETA

C   CALCULATION OF NNX AND NNY, ALONG WITH MMX AND MMY
       MMX=MMY=NNX=NNY=1
     2 MMX=MMX+1 $ NNX=2*NNX $ IF(NNX.GT.NX) 3,2
     3 NNX=2*NNX $ IF(NZX) 44,6,4
    44 NNX=NNX/2 $ MMX=MMX-1 $ GO TO 6
     4 DO 5 M=1,NZX $ MMX=MMX+1
     5 NNX=2*NNX
     6 MMY=MMY+1 $ NNY=2*NNY $ IF(NNY.GT.NY) 7,6
     7 NNY=2*NNY $ IF(NZY) 45,10,8
    45 NNY=NNY/2 $ MMY=MMY-1 $ GO TO 10
     8 DO 9 M=1,NZY $ MMY=MMY+1
     9 NNY=2*NNY
    10 MM=MAX0(MMX,MMY) $ NN=MAX0(NNX,NNY) $ ND=NNX+NNY $ J=2*ND
       IF(J.GT.NDIMEN.OR.NN.GT.NMAX.OR.MM.GT.14) 11,12
    11 PRINT 96,NNX,MMX,NNY,MMY,J $ CALL EXIT

C   INPUT PARAMETER PRINT-OUT
    12 PRINT 95,ALPHA,BETA,SX,SY,BANNER,NZX,NZY,NNX,NNY
       PRINT 97,MMX,MMY

C   COMPUTATION OF APERTURE DATA-POINT SPACINGS
       DELX=SX/NNX $ LLX=1.0/DELX $ DELY=SY/NNY $ LLY=1.0/DELY

C   COMPUTATION OF APERTURE-DATA POSITION WITHIN THE INTEGRATION INTERVAL
       FACTOR=1.0/(SX*SY) $ I0X=(NNX-NX+1)/2 $ I0Y=(NNY-NY+1)/2+NNX

C   PRE-ZEROING OF INTEGRAL FIELD
       DO 13 M=1,J
    13 E(M)=(0.,0.)

C   DO LOOP TO COMPUTE TWO PLANE-CUT INTEGRALS OVER APERTURE-FIELD VECTOR
       DO 25 I=1,2 $ MX=ND*(I-1) $ IX=MX+I0X $ IY=MX+I0Y

C   SET-UP OF INTEGRAND - COLLAPSE ON X- AND Y-PLANE CUTS
       DO 14 K=1,NY $ J=K+IY
```

```
      CALL GETARAY(STOR,NX)
      DO 14 L=1,NX $ M=L+IX
      E(M)=E(M)+STOR(L)
   14 E(J)=E(J)+STOR(L)
C     INITIALIZE Y-PLANE-CUT PARAMETERS
      NXY=NX $ MM=MNX $ NN=NNX $ DEL=DELX $ LL=LLX
      FACTOR=-FACTOR $ J=0
   15 NMID=NN/2+1 $ I2=IX+NXY $ I1=IX+1 $ IF(MOD(NXY/2,2).EQ.0) I1=I1+1
C     PRE-FFT MULTIPLICATION BY (-1)**M
      DO 16 M=I1,I2,2
   16 E(M)=-E(M)
C     FFT COMPUTATION
      CALL CCFFT2(E(MX+1),MM,FACTOR,-1)
      I1=MX+2 $ I2=MX+NN $ JJ=MOD(J+I-1,2)
C     POST-FFT MULTIPLICATION BY (-1)**J
      DO 17 M=I1,I2,2
   17 E(M)=-E(M)
C     RESTRICT COMPUTATIONS TO REAL POLAR SPATIAL-FREQUENCY ANGLES
      IF(LL.GT.NMID-2) 18,19
   18 I1=MX+1 $ I2=MX+NN $ KK=0 $ GO TO 20
   19 KK=NMID-LL $ I1=KK+MX $ I2=I1+2*LL $ KK=KK-1
   20 NMID=NMID+MX $ K=KK+MX
C     VECTOR-COMPONENT COMPUTATION
      DO 23 M=I1,I2 $ L=M-K $ LL=M-KK
      SINTHTA=(M-NMID)*DEL $ THETA(L)=57.2957795131 82*ASIN(SINTHTA)
      IF(JJ.EQ.0) 21,22
   21 E(LL)=(0.,1.0)*E(M) $ GO TO 23
   22 E(LL)=CMPLX(0.,SQRT(1.0-SINTHTA**2))*E(M)
   23 CONTINUE
C     VECTOR-COMPONENT PLOTTING
      CALL PAPLOT(L,E(MX+1),THETA,DATA,2*NN,BANNER,2)
      IF(J.EQ.0) 24,25
C     INITIALIZE X-PLANE-CUT PARAMETERS
   24 IF(I.EQ.1) FACTOR=-FACTOR $ MX=MX+NNX $ J=1 $ LX=L $ KX=I1-NMID-1
      NXY=NY $ MM=MNY $ NN=NNY $ DEL=DELY $ LL=LLY $ IX=IY $ GO TO 15
   25 LY=L
      I1=NNX+1 $ I2=NNX+LY
C     ANTENNA-PATTERN COMPUTATION
      DO 26 M=I1,I2 $ J=M+ND
   26 DATA(M)=SQRT(REAL(E(M)*CONJG(E(M))+E(J)*CONJG(E(J))))
C     ANTENNA-PATTERN PLOTTING (X-PLANE CUT)
      CALL PAPLOT(LY,0.,THETA,DATA(I1),LY,BANNER,1)
      DO 27 M=1,LX $ J=M+ND
      SINTHTA=(M+KX)*DELX $ THETA(M)=57.2957795131 82*ASIN(SINTHTA)
   27 DATA(M)=SQRT(REAL(E(M)*CONJG(E(M))+E(J)*CONJG(E(J))))
C     ANTENNA-PATTERN PLOTTING (Y-PLANE CUT)
      CALL PAPLOT(LX,0.,THETA,DATA,LX,BANNER,1)
      GO TO 1
   28 CONTINUE
      END
```

Appendix D: Modifications to subroutine PAMPLOT for compatibility with principal-plane-cut program.

I. SUBROUTINE name change and SUBROUTINE argument-list change:

```
SUBROUTINE PAPLOT(N,EP,XVALUE,DATA,N2,BANNER,MODE)
```

II. Subroutine DATA-statement change:

```
 DATA(COMP(M),M=1,6)/10HTHETA COMP,5HONENT,10H   PHI COMP,5HONENT,10
AHANTENNA PA,5HTTERN/,(AMP(M),M=1,4)/10HAMPLITUDE ,8HPHASE   ,10HAMP/
BPHASE   ,10HMAGNITUDE /,(ELAZ(M),M=1,2)/8H PHI = 0,8HPHI=PI/2/
```

III. FORMAT-statement change and change in ENCODE statement (the latter statement is located just following statement 29):

```
100 FORMAT(4HAMP=,G9.3,2XA8,1XA10,A5,1X)
```

```
ENCODE(40,100,BB) ANORM,ELAZ(IB),COMP(IA),COMP(IA+1)
```

IV. Replace statements 32 thru 35 (located at end of subroutine) with the following six statements:

```
32 IF(IA.EQ.1) 33,36
33 IF(IB.EQ.1) 34,35
34 IA=3 $ IB=2 $ GO TO 37
35 IA=5 $ GO TO 37
36 IB=MOD(IB,2)+1 $ IF(IB.EQ.2) IA=1 $ IF(MODE.EQ.1) IB=1
37 RETURN
```

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 81-1643	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE

Efficient Computation of the Far Field Radiated by an Arbitrary Rectangular-Aperture Distribution

5. Publication Date

March 1981

6. Performing Organization Code

7. AUTHOR(S)

Richard Lewis

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, DC 20234

10. Project/Task/Work Unit No.

11. Contract/Grant No.

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

Electromagnetic Compatability Analysis Center
Annapolis, Maryland 21402

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

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

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

This report contains the computer documentation for calculating the far-zone electric field due to a user-prescribed electric-field distribution within a rectangular aperture. The far-field output is computed along two arbitrarily selected, perpendicular, spatial-frequency plane cuts. Program execution time is minimized by the use of fast Fourier transform (FFT) processing. The program was designed so that the required far-field output is obtained by processing only two, vector, one-dimensional FFTs. The far-field results are obtained in the form of elevation and azimuth vector components and electric-field-vector magnitude. A complete analytical discussion of the problem is presented, along with sample graphical output to illustrate how aliasing and output resolution limitations effect the graphical results.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Algorithm; antenna; aperture; computer program; electromagnetic; far-field; FFT; Fourier transform.

18. AVAILABILITY [X] Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402,

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

19. SECURITY CLASS (THIS REPORT)	21. NO. OF PRINTED PAGES
UNCLASSIFIED	45
20. SECURITY CLASS (THIS PAGE)	**22. Price**
UNCLASSIFIED	$6.50

USCOMM-DC