A GENERAL PROGRAM FOR PLOTTING THREE-DIMENSIONAL ANTENNA PATTERNS

Waymond R. Scott, Jr.
School of Electrical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250

[Editor's note: This feature article is somewhat different, in that it describes software routines for plotting antenna patterns, explains their usage, and provides examples of results. The set of subroutines, along with the example programs for generating Figures 2b, 3, and 4a, are included in the material that is available from the author. Those interested should send their requests to above address, and enclose a check or money order for US$10.00, payable to the author, to cover duplication, postage, and media. The listings of the source code will be provided on both paper and on a MS-DOS 5.25" 360K floppy disk. Although not the first time the *Newsletter* has published an article such as this – describing a program of interest to AP-S, which is available free to the community (except possibly for the costs of postage and duplication) – this may still be viewed as an experiment. The *Newsletter* welcomes readers' comments and suggestions, as well as candidates for submissions from those having programs of general interest they may which to share. The *Newsletter* thanks Waymond Scott for sharing his program with us. WRS]

Abstract

A set of subroutines is developed in this work which can be used to generate a three-dimensional graph of an antenna pattern. The graph is a projection of the surface representing the pattern. The three-dimensional shape of the pattern is readily apparent from this type of graph. The set of subroutines provides several options and features which can be used to enhance the graph. The set of subroutines is available from the author.

Introduction

Antenna patterns are commonly represented graphically by pattern cuts, where the amplitude of the pattern is graphed as a function of one of the angular coordinates. A pattern cut is usually plotted as a rectangular graph, where the vertical coordinate is proportional to the amplitude, and the horizontal coordinate is proportional to the angle. Figure la is a rectangular graph of a pattern cut for a dipole antenna, with a length of one and one-half wavelengths. This type of graph provides useful quantitative information about the pattern: beamwidth, sidelobe levels, etc. However, the shape of the pattern is severely distorted. In order to graph the pattern cut without distorting its shape, the pattern cut is sometimes plotted as a polar graph, where the radial coordinate, which is proportional to the amplitude of the pattern, is plotted as a function of the angular coordinate. Figure lb is a polar graph of the pattern cut for the same dipole antenna. The polar graph provides information about the shape of one cut of the antenna pattern, but does not provide much information about the overall three-dimensional shape of the pattern.

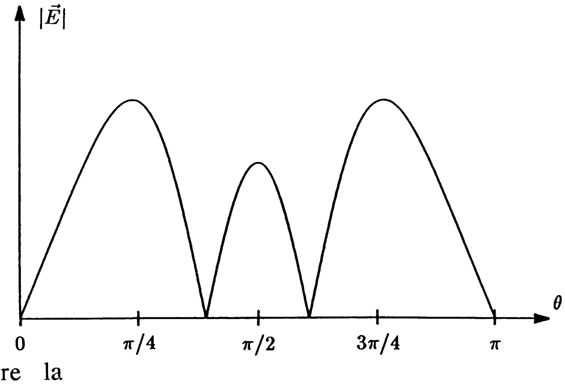


Figure la

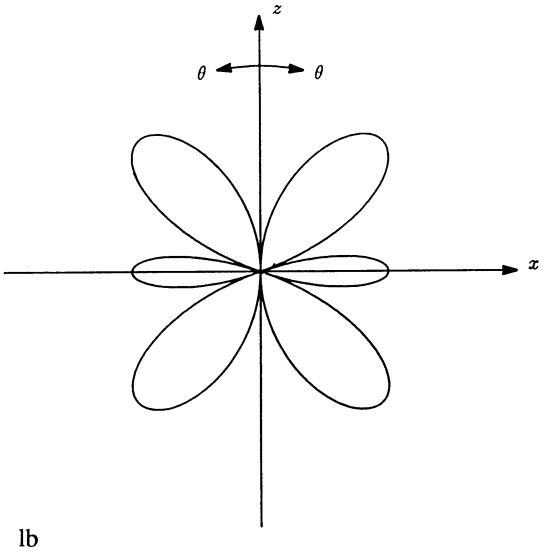


Figure lb

More information about the overall three-dimensional shape of the pattern can be obtained from a two-dimensional projection of a three-dimensional surface representing the pattern. Figure 1c is such a graph of the pattern of the dipole antenna. Here, the pattern is plotted in the spherical coordinate system, where the radial coordinate, which is proportional to the amplitude of the pattern, is plotted as a function of the two angular coordinates. One fourth of the pattern is cut away so that the shape of the pattern cuts can be seen. The three-dimensional shape of the pattern is readily apparent from the graph.

Description

The set of subroutines developed in this work can be used to generate such a three-dimensional graph of an antenna pattern. The subroutines are suitable for use with either a microcomputer or a mainframe computer. The graph consists of a mesh of connecting line segments, which form connecting quadrilaterals that represent the surface of the three- dimensional data. The subroutines provide several features and options:

1. Hidden line removal: All completely hidden line segments and hidden portions of partially hidden line segments are removed: therefore, the visible portion

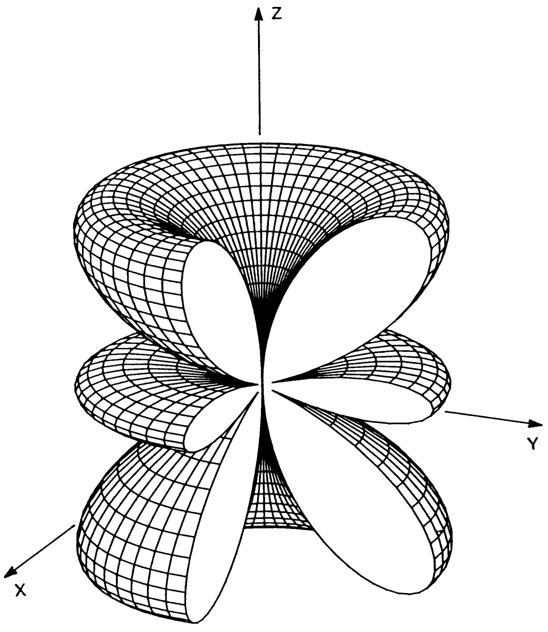


Figure 1c

of each line segment is drawn. Optionally, all completely hidden and all partially hidden line segments are removed: therefore, only the completely visible line segments are drawn. This decreases the amount of computer time required to generate the graph, at the cost of having an incomplete graph.

2. Cuts in the graph: A section may be cut out of the graph, making it easier to see the shape of a particular pattern cut. When the pattern is cut, line segments which form the interior surface of the pattern become visible through the cut: these line segments clutter up the graph and make it confusing. The subroutines provide the option of removing these line segments.

3. Flexibility: The subroutines can be used to plot other types of three-dimensional graphs. For example, a cylindrical graph, where the height is plotted as a function of the radial coordinate and the angular coordinate, can be produced. Rectangular three-dimensional graphs, where the height is plotted as a function of the coordinates x and y, can also be generated with the subroutines. However, this is not recommended, since subroutines which are much more efficient are available for plotting the rectangular graphs.

4. Transportability: The subroutines are written in standard Fortran V. The standard CALCOMP subroutine call, PLOT(X,Y,IPEN), is used to do the plotting, and is the only subroutine that the user must supply. If the subroutine PLOT is not available to the user, it must be replaced by an equivalent subroutine.

5. Ease of use: Specialized subroutines are supplied which make it easy to generate the spherical and cylindrical three-dimensional graphs.

6. Axes: The subroutines provide the option of drawing the projections of the axes. The axes are truncated so that they do not interfere with the graph.

7. Efficient plotting: The line segments are plotted in a manner suitable for efficient use with pen plotters, as well as with raster-scan displays. Unnecessary "pen ups" and "pen downs" are minimized, and each line segment is only plotted once.

8. Selectable viewing angles: The viewing angles for the projection of the graph are user selectable.

Brief descriptions, with examples of how to use the subroutines developed in this work, are presented in the following sections. An understanding of the descriptions will be aided by the example programs for generating Figures 2b, 3, and 4a. The example programs along with the subroutines developed in this work are included in the material which is available from the author.

Spherical Plots

The subroutine, PLT3DS, is supplied specifically for generating spherical three-dimensional graphs. For this type of graph, the radial coordinate, r, is plotted as a function of the angular coordinates θ and ϕ, where r, θ, and ϕ are the standard spherical coordinates in which θ is the polar angle, measured from the z-axis. The subroutine generates a graph which consists of a two-dimensional projection for the lines of constant θ and ϕ on the surface of the pattern. The subroutine is called with the Fortran statement

```
CALL PLT3DS(R,THETA,PHI,NI,NJ,ICUT,IPHIDN,
                              AXISLN,THETAV,PHIV)
```

where the inputs are as follows:

R(I,J) – Two-dimensional array containing the radius in inches at the angular coordinates THETA(I) and PHI(J).

THETA(I) and PHI(J) – One-dimensional arrays containing the angular coordinate in radians of the Ith row and Jth column of R(I, J). The elements in the THETA and PHI arrays must be in monotonically increasing order, i.e., THETA(I) < THETA(I+1) for I = 1,2,3,..., NI-1, and PHI(J) < PHI(J+1) for J = 1, 2, 3,...,NJ-1.

NI and NJ – Dimensions for the portions of the arrays R, THETA, and PHI which are filled with data. The actual dimensions for the arrays are set by the parameters IDIM and JDIM which are defined in the subroutine. These parameters must agree with those defined in the subroutine PLT3DG and the actual dimensions of the arrays in the calling routine. (N1 ≤ IDIM and NJ ≤ JDIM).

ICUT – If ICUT = O, the lines behind the cut are drawn, and if ICUT = 1, the lines behind the cut are not drawn.

IPHIDN – If IPHIDN = 1, both the completely and partially visible line segments are drawn; otherwise, only the completely visible line segments are drawn.

AXISLN – Length of the axes before projection. If AXISLN = O, the axes are not drawn.

THETAV and PHIV – The viewer angles for the projection, in radians.

The radius R(I,J) must be greater than or equal to zero and less than or equal to some value set by the physical dimensions of the plotter. Normally, the radius is either proportional to the magnitude of the antenna pattern, or to a scaled version of the log magnitude of the antenna pattern. For example, let E be the field for a certain antenna, which has a maximum magnitude of E_{max}, and let the maximum radius for the graph be 5 inches. A linear graph of the pattern is obtained by setting R equal to $5|E|/E_{max}$. A log graph of the pattern is obtained by setting R equal to the maximum of $\frac{5}{2}\log_{10}(|\mathbf{E}|/E_{max}) + 5$ or zero. Here the values from -40dB to 0dB are plotted on a scale of 8dB per inch, and values less than -40dB are set equal to -40dB.

The subroutine can be used to generate a graph of the entire pattern or only a portion of the pattern, i.e., to cut out a section or sections from the pattern. To generate a graph of the entire pattern, data must be supplied for the entire surface, e.g., THETA(1) = 0, THETA(NI) = π, and PHI(NJ) = PHI(1) + 2π. Note that this means that a column of the data must be duplicated: R(I,1) = R(I,NJ). To generate a graph with a cut, data is only supplied for the region to be plotted. For example, to cut out the section between $\phi = 0$ and $\pi/2$ in a pattern, data is supplied for the region $0 \le \theta \le \pi$ and $\pi/2 \le \phi \le 2\pi$: THETA(1) = 0, THETA(NI) = π; PHI(1) = $\pi/2$, PHI(NJ) = 2π.

The subroutine also provides the option of removing the lines behind a cut when a section in ϕ is cut out. This requires that the two curves formed by the points on the edge of the cut start and end on the z-axis:

$$\mathrm{R(I,J)\sin(THETA(I))} = 0, \begin{cases} \mathrm{I=1} & \text{with } \mathrm{J} = 1 \text{ and NJ} \\ \mathrm{I=NI} & \text{with } \mathrm{J} = 1 \text{ and NJ} \end{cases}$$

or that the two curves formed by the points on the edge of the cut start and end at the same point:

$$\mathrm{R(1,J)\sin(THETA(1))} = \mathrm{R(NI,J)\sin(THETA(NI))}, \quad \mathrm{J} = 1 \text{ and NJ},$$

$$\mathrm{R(1,J)\cos(THETA(1))} = \mathrm{R(NI,J)\cos(THETA(NI))}, \quad \mathrm{J} = 1 \text{ and NJ}.$$

If these conditions are not met, the lines behind the cut may not be removed properly.

Graphs of the pattern of a traveling-wave antenna are presented in Figure 2. Four graphs are presented to demonstrate some of the features of the subroutines. The traveling-wave antenna has a length of six wavelengths, and is directed along the positive z-axis. In all four graphs, the radius is proportional to the magnitude of the far-zone electric field, the lines of constant ϕ are 6° apart (NJ=61 for Figure 2a, and NJ=46 for the other figures), the lines of constant θ are 1° apart (NI = 181), and the viewer angles are THETAV = 70° and PHIV = 25°. The entire surface of the pattern is drawn in Figure 2a. The three-dimensional shape of the pattern is apparent from this graph. However, when the section between $\phi = 0$ and $\pi/2$ is cut out of the graph, as in Figure 2b, the three-dimensional shape becomes more apparent, and the shape of the pattern cuts in the $\theta = 0$ and $\pi/2$ planes become apparent. The section between $\phi = 0$ and $\pi/2$ is also cut out in Figures 2c and 2d. The lines behind the cut are removed in Figures 2b and 2d, but they are not removed in Figure 2c. The graphs with the lines behind the cut removed are seen to be less confusing.

In Figures 2a, 2b, and 2c, both the completely and partially visible line segments are drawn. In Figure 2d, only the completely visible line segments are drawn; this decreases the amount of computer time required to generate the graph, at the cost of an incomplete graph, as seen in Figure 2d. The ratio of the computer run time required for generating Figure 2b, to that required for generating Figure 2d, is approximately 2 to 1 (see Table I). This ratio can vary dramatically, depending on the type of graph and the number of grid points. The graphs with only the completely visible line segments can be useful as preliminary graphs, since they take less time to generate. This feature is usually only useful when using a relatively slow computer.

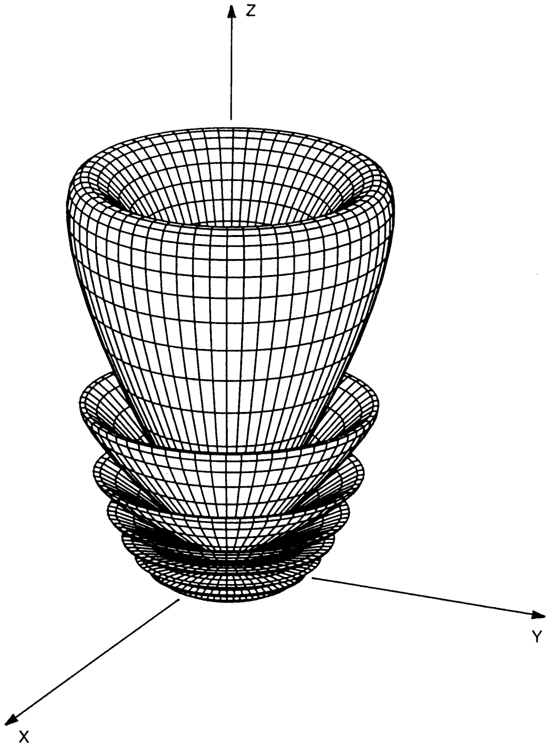


Figure 2a

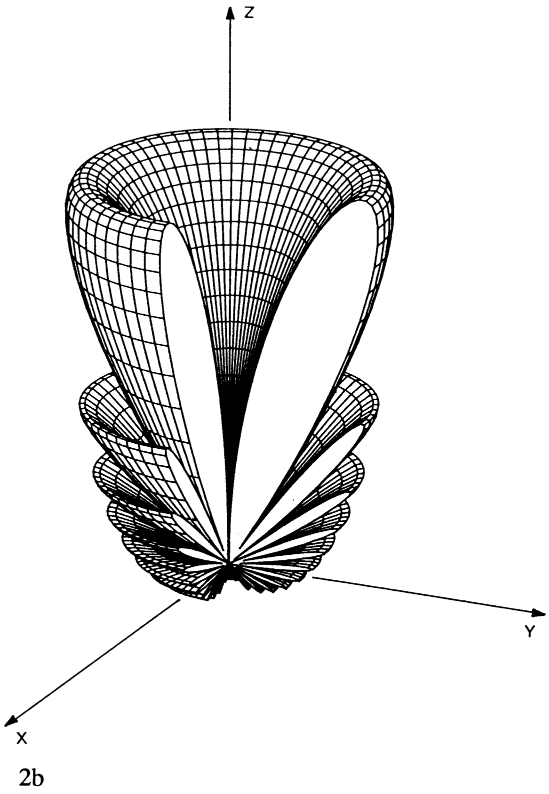


Figure 2b

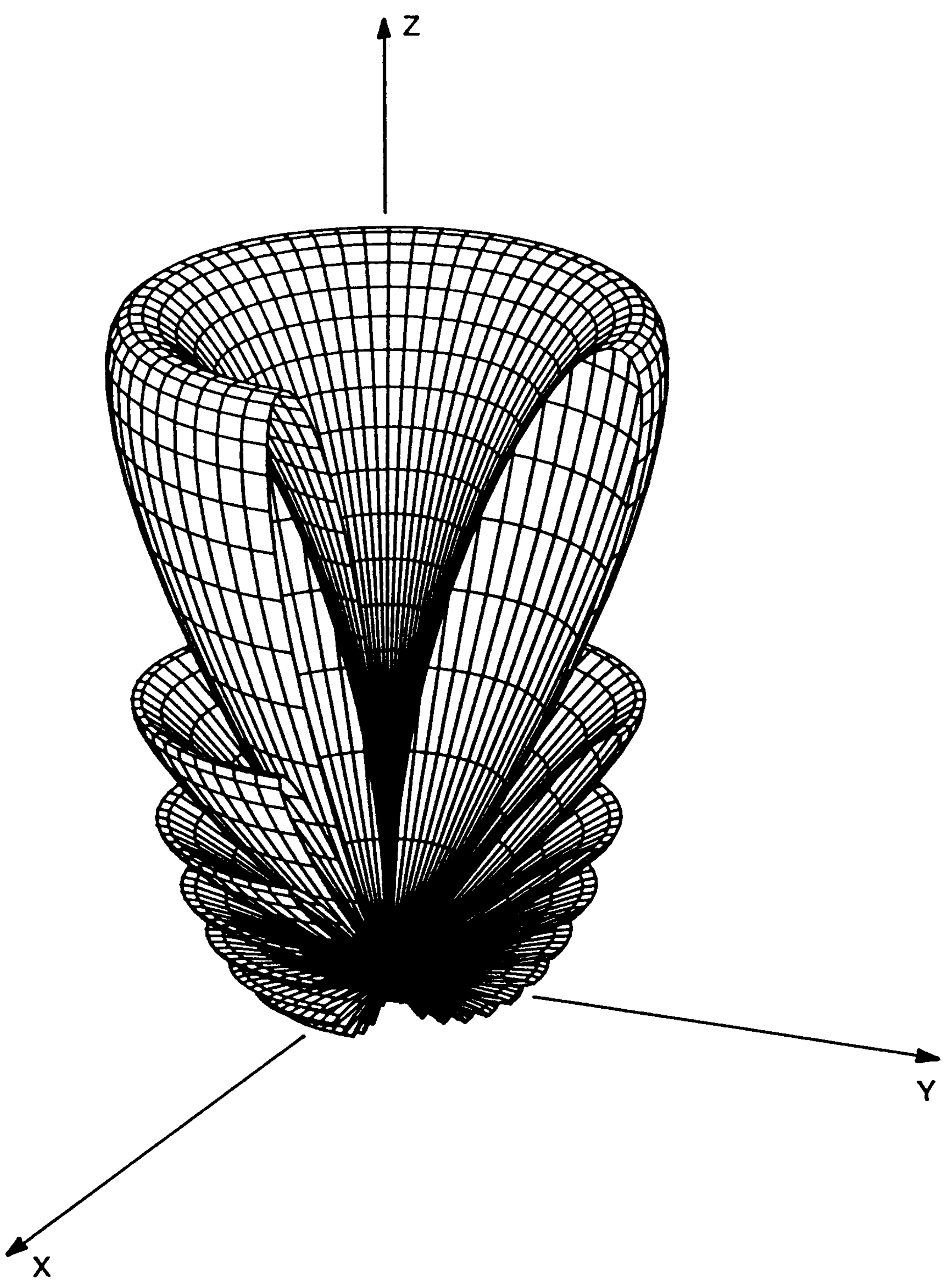


Figure 2c

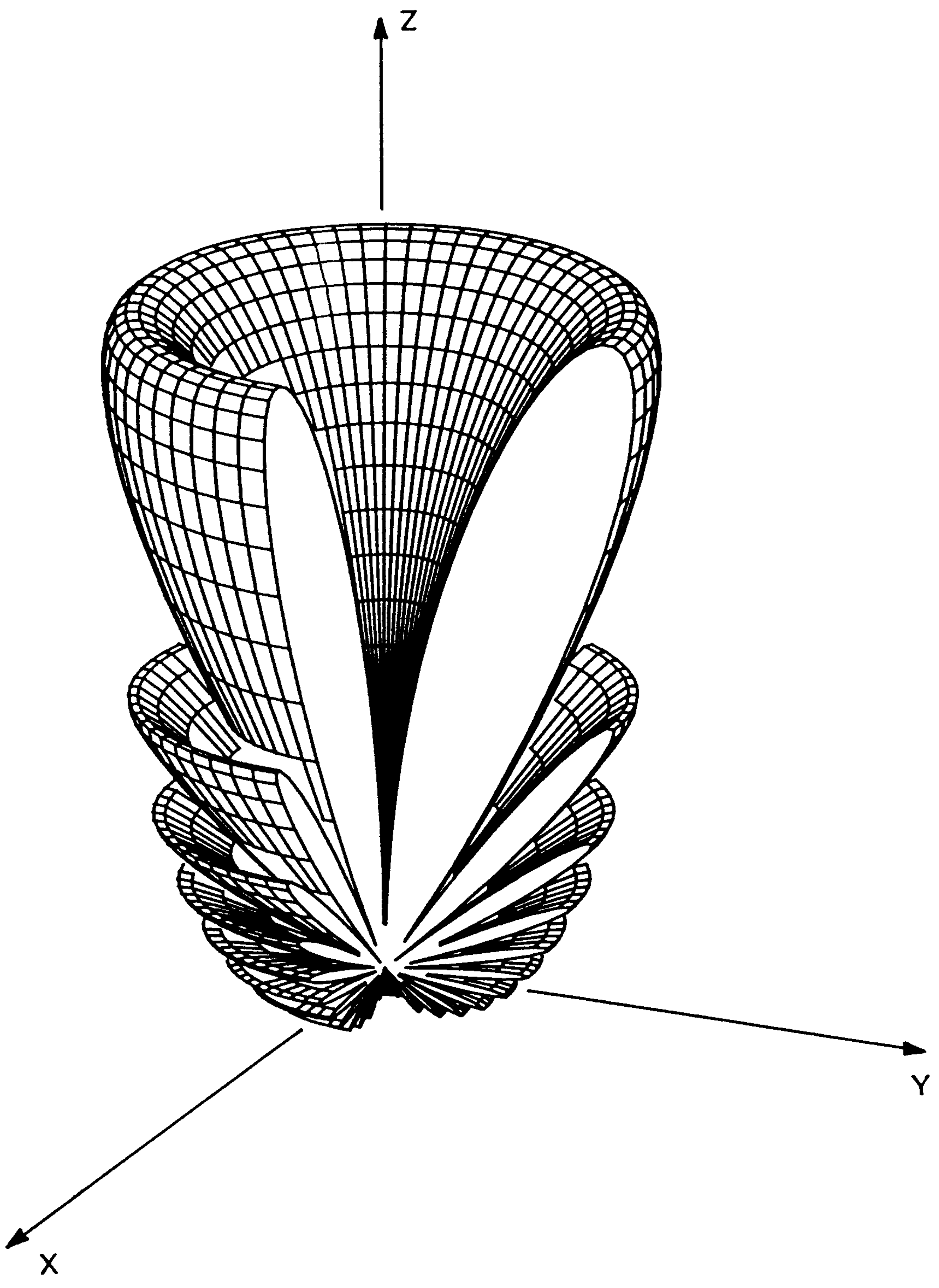


Figure 2d

The computer run time required to generate Figures 1, 2b, 2d, 3, and 4a are presented for both a mainframe computer (CDC Cyber 180/990) and a personal computer (16MHz IBM PS/2 Model 80, with an 80387 coprocessor, using Microsoft FORTRAN 5.0) in Table I. The run time for a particular graph is approximately proportional to the number grid points raised to the 1.5 power. Thus, when the number of grid points for Figure 1c is reduced from 3321 (NI=81 and NJ=41) to 861 (NI=41 and NJ=21), the run time is reduced approximately from 111 sec to $(861/3321)^{1.5}\bullet 111$ sec = 15 sec on the personal computer; the actual time is 17 sec.

Table I
Computer Run Time in Seconds

Figure	Gridpoints = NI×NJ	Time	
		CDC Cyber 180/990	IBM PS/2 Model 80
1c	3,321 = 81×41	7.8	111
2b	8,326 = 181×46	41.6	563
2d	8,326 = 181×46	16.0	324
3	1,681 = 41×41	3.5	46
4a	1,881 = 11×171	2.8	48

Cylindrical Graphs

The subroutine PLT3DC is supplied specifically for generating cylindrical three-dimensional graphs. For this type of graph, the height (the z-coordinate) is plotted as a function of the radial coordinate, r, and the angular coordinate, ϕ, where r, ϕ, and z are the standard cylindrical coordinates. The subroutine generates a graph which consists of a two-dimensional projection of the lines of constant r and ϕ onto the surface of the pattern. The subroutine is called with the following Fortran statement:

```
CALL PLT3DC(R,PHI,Z,NI,NJ,ICUT,ZCUT,IPHIDN,
                              AXISLN,THETAV,PHIV)
```

where the parameters are as follows:

R(I) – One-dimensional array containing tne radial coordinate in inches of the Ith row of Z(I,J). The elements in the R array must be in monotonically increasing order, i.e., R(I) ≤ R(I+1) for I = 1,2,3, ...,NI-1.

PHI(J) – One-dimensional array containing the angular coordinate, in radians, of the Jth column of Z(I,J). The elements in the PHI array must be in monotonically increasing order.

Z(I,J) – Two-dimensional array, containing the height in inches at the radial coordinate, R(I), and the angular coordinate, PHI(J).

ICUT – If ICUT = 0, the lines behind the cut are drawn, and if ICUT = 2, the lines behind the cut are not drawn.

ZCUT – Auxiliary point on the z-axis, used in the procedure for removing the lines behind the cut (only used when ICUT = 2).

The other inputs are the same as those for the subroutine PLT3DS.

The subroutine can be used to generate a graph of the entire pattern, or of only a portion of the pattern. To generate a graph of the entire pattern, data must be supplied for the entire surface, i.e., PHI(NJ) = PHI(1) + 2π, and R(1) = O. The subroutine can be used to draw only a portion of the pattern (i.e., to cut out a section or sections from this pattern), by only supplying data for that portion of the pattern. For example, to remove the section between ϕ = 0 and $\pi/2$, data is supplied for the region $\pi/2 \leq \phi \leq 2\pi$, i.e., PHI(1) = $\pi/2$, PHI(NJ) = 2π, and R(1) = 0.

The subroutine also provides the option of removing the lines behind the cut when a section in ϕ is cut out. This requires that R(1) = 0, and an auxiliary point, ZCUT, on the z-axis. The point ZCUT is chosen such that the two curves, formed by connecting the successive points

$$\begin{Bmatrix} R(1)\\ PHI(J)\\ Z(1,J)\end{Bmatrix}, \begin{Bmatrix} R(2)\\ PHI(J)\\ Z(2,J)\end{Bmatrix}, \begin{Bmatrix} R(3)\\ PHI(J)\\ Z(3,J)\end{Bmatrix},\ldots,$$

$$\begin{Bmatrix} R(NI)\\ PHI(J)\\ Z(NI,J)\end{Bmatrix}, \begin{Bmatrix} 0\\ 0\\ ZCUT\end{Bmatrix}, \begin{Bmatrix} R(1)\\ PHI(J)\\ Z(1,J)\end{Bmatrix}$$

for J = 1 and NJ, do not cross themselves. For example, the auxiliary point, ZCUT, is chosen to be -10 for the graph in Figure 3. The dashed lines and the "edges" of the graph form the aforementioned curves: these curves are seen not to cross themselves. However, if ZCUT was chosen to be zero, the curves would cross themselves, and then the lines behind the cut would not be removed properly.

Figure 3 is a cylindrical, three-dimensional graph of the field pattern for a circular aperture. The aperture lies in the x-y plane, and is centered on the z-axis. It is 2.5 wavelengths in diameter, and is uniformly illuminated with the E field in the y direction. The height, z, is proportional to the far-zone electric field of the aperture, the radius R is proportional to the spherical coordinate θ for the pattern, and the angle PHI for the graph is equal to the angular coordinate ϕ for the pattern. The cylindrical graph does not accurately represent the three-dimensional shape of the pattern, but it does provide useful information about the shape of the pattern that would not be apparent in a spherical graph, such as the slight asymmetry seen in Figure 3.

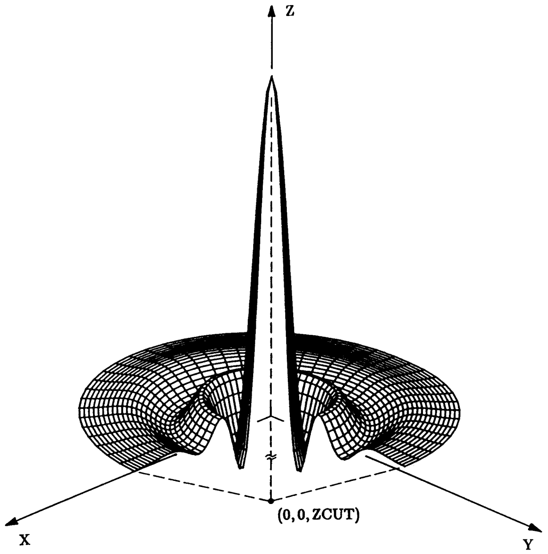


Figure 3

General Graphs

The subroutine PLT3DG is supplied for generating a general, three-dimensional graph, where any coordinate can be graphed as a function of any other two coordinates. Due to the flexibility of this subroutine, it is more complicated to use than the subroutines PLT3DS and PLT3DC. An understanding of how to use the subroutine PLT3DG will be aided by studying the example program for generating Figure 4a and the subroutines PLT3DS and PLT3DC, which call the subroutine PLT3DG. The subroutine is called with the following Fortran statement:

```
CALL PLT3DG(X,Y,Z,NI,NJ,ICUT,ZCUT,IHPIDN,
                        AXISLN,THETAV,PHIV)
```

where

X(I,J), Y(I,J), and Z(I,J) – Two-dimensional arrays containing the x, y, and z coordinates of the (I,J)th grid point.

ICUT – If ICUT = 0, the lines behind the cut are drawn, and if ICUT = 1 or 2, the lines behind the cut are not drawn. The auxiliary point, ZCUT, is included in the cut only when ICUT = 2.

The other inputs are the same as for those for subroutines PLT3DS and PLT3DC.

The surface of the pattern to be graphed is divided up into a mesh of connecting quadrilaterals. The quadrilaterals are formed by the intersection of two sets of curves. For example, curves of constant θ and ϕ are used for the spherical graphs. The x, y, and z coordinates, for each grid point of the mesh, are supplied to the subroutine in the three two-dimensional arrays X(I,J), Y(I,J), and Z(I,J). The Ith curve in one set of curves is formed by connecting the successive points

$$\begin{Bmatrix} X(I,1)\\ Y(I,1)\\ Z(I,1)\end{Bmatrix}, \begin{Bmatrix} X(I,2)\\ Y(I,2)\\ Y(I,2)\end{Bmatrix}, \begin{Bmatrix} X(I,3)\\ Y(I,3)\\ Z(I,3)\end{Bmatrix},\ldots, \begin{Bmatrix} X(I,\ NJ)\\ Y(I,\ NJ)\\ Z(I,\ NJ)\end{Bmatrix};$$

and the Jth curve in the other set of curves is formed by connecting the successive points

$$\begin{Bmatrix} X(1,J)\\ Y(1,J)\\ Z(1,J)\end{Bmatrix}, \begin{Bmatrix} X(2,J)\\ Y(2,J)\\ Z(2,J)\end{Bmatrix}, \begin{Bmatrix} X(3,J)\\ Y(3,J)\\ Z(3,J)\end{Bmatrix},\ldots, \begin{Bmatrix} X(NI,J)\\ Y(NI,J)\\ Z(NI,J)\end{Bmatrix}.$$

For a curve to start and end at the same point, the starting and ending points of the curve must be the same. For example, (X(I,1), Y(I,1), Z(I,1)) must be the same as (X(I,NJ), Y(I,NJ), Z(I,NJ)), for the Ith curve to start and end at the same point.

The subroutine can be used to generate a graph of the entire pattern, or of only a portion of the pattern. When only a portion of the pattern is drawn, the subroutine provides the option of removing the lines behind the curves on the edge of the pattern with J = 1 or NJ. The curves are formed by connecting the successive points

$$\begin{Bmatrix} X(1,J)\\ Y(1,J)\\ Z(1,J)\end{Bmatrix}, \begin{Bmatrix} X(2,J)\\ Y(2,J)\\ Z(2,J)\end{Bmatrix}, \begin{Bmatrix} X(3,J)\\ Y(3,J)\\ Z(3,J)\end{Bmatrix},$$

$$\ldots,\begin{Bmatrix} X(NI,J)\\ Y(NI,J)\\ Z(NI,J)\end{Bmatrix}, \begin{Bmatrix} 0\\ 0\\ ZCUT\end{Bmatrix}, \begin{Bmatrix} X(1,J)\\ Y(1,J)\\ Z(1,J)\end{Bmatrix}$$

for J = 1 and NJ. The point (O,O,ZCUT) is an auxiliary point which is included in the curves only when ICUT = 2. The auxiliary point is useful for keeping the

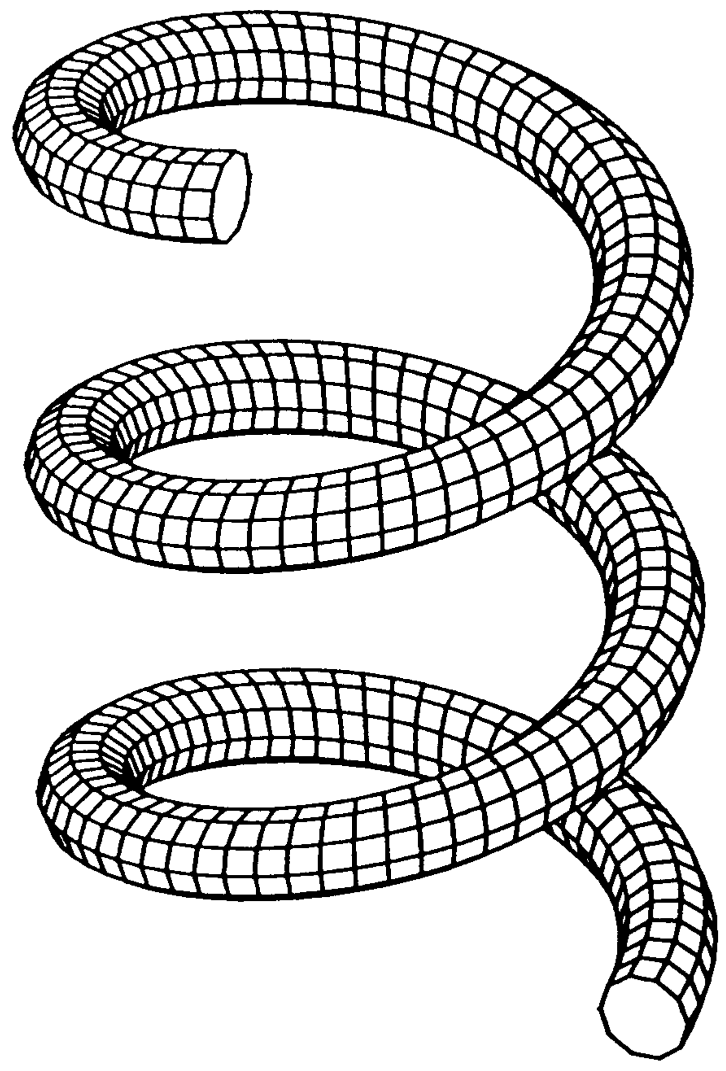


Figure 4a

curves from crossing themselves in certain types of graphs, such as the cylindrical graphs. If the curves cross themselves, the lines behind the cut will not be removed properly.

Figure 4a is a three-dimensional graph of a helix generated with PLT3DG, and Figure 4b is a sketch defining the parameters used to generate the graph. The surface of the graph is constructed of lines of constant α and ϕ. Note that the lines behind the ends (cuts) are removed. The steps in the α-coordinate are indexed with I, and those in the ϕ-coordinate are indexed with J. The roles of the indexes I and J are chosen so that the lines behind the cut can be removed. If the roles of the indexes were reversed, the subroutine would not be able to remove the lines behind the cut, since the subroutine only provides the option of removing the lines behind a cut with a constant J.

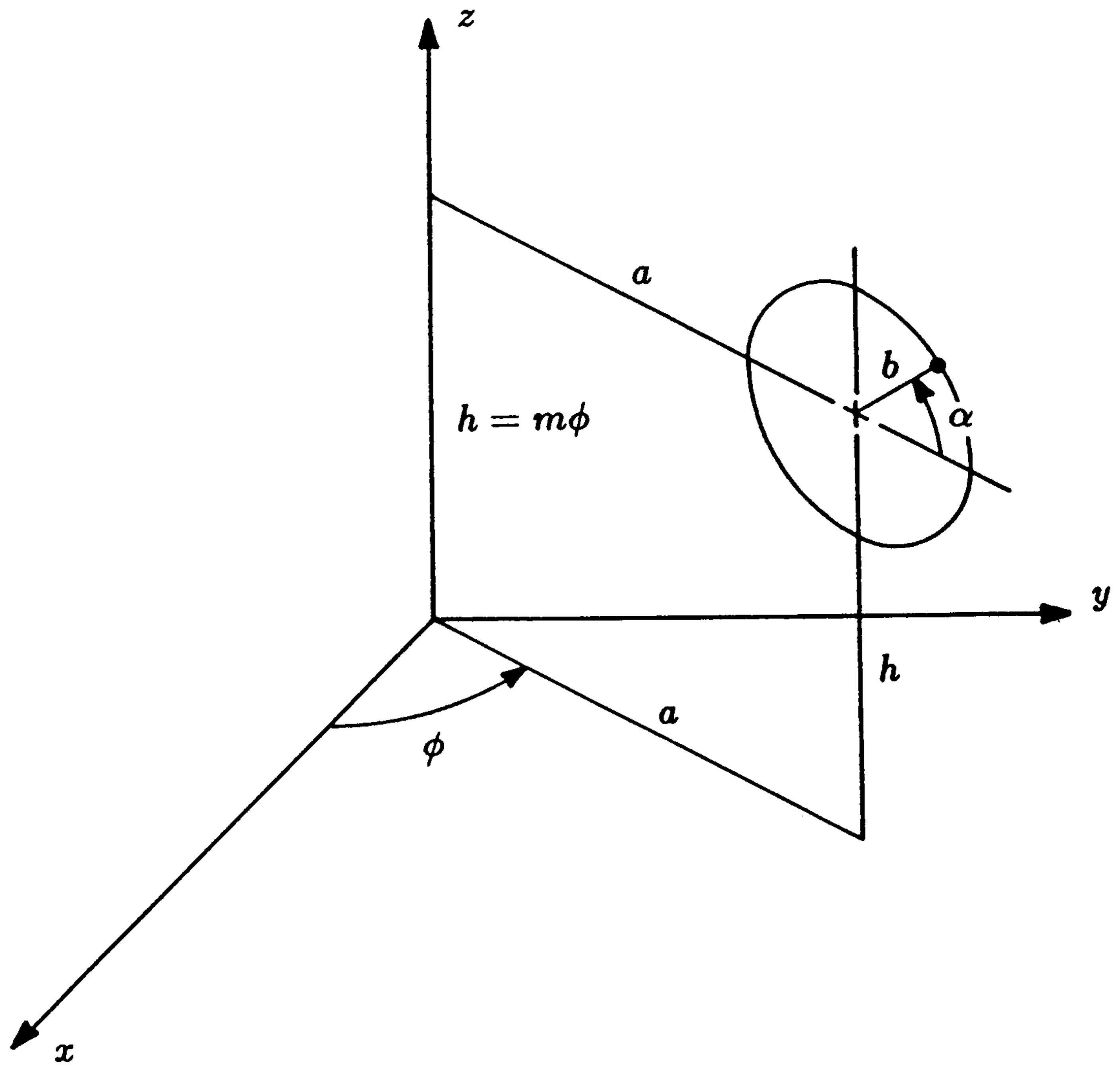
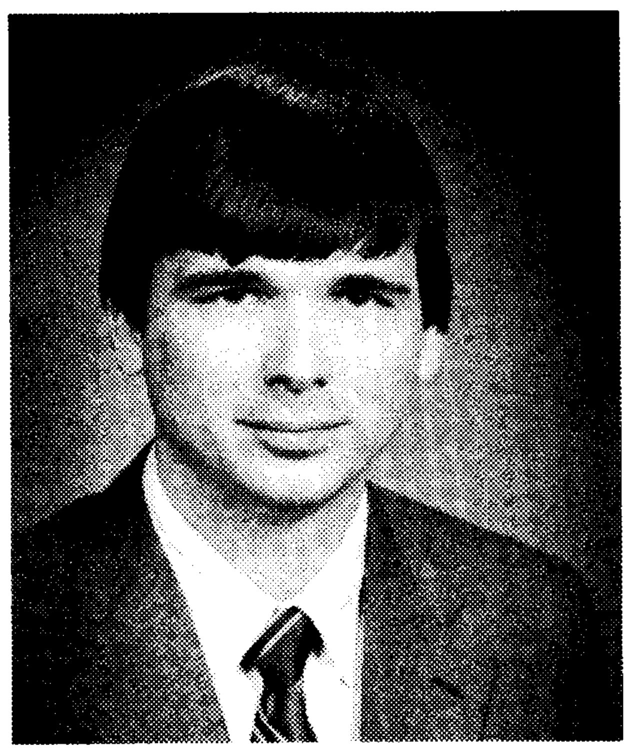


Figure 4b

This work was supported in part by the Joint Services Electronics Program, under contracts DAAG29-84-K-0024 and DAAL03-87-K-0059.

INTRODUCING WAYMOND R. SCOTT, FEATURE ARTICLE AUTHOR



Waymond R. Scott, Jr., was born in Calhoun, Georgia on April 6,1958. He received the BEE, MSEE, and PhD degrees from the Georgia Institute of Technology, Atlanta, in 1980, 1982 and 1985, respectively.

From 1979 to 1980, he was a Student Assistant and Graduate Research Assistant at the Georgia Tech Research Institute. From 1980 to 1985, he was a Graduate Research Assistant in the School of Electrical Engineering at the Georgia Institute of Technology, where he is currently an Assistant Professor of Electrical Engineering.