David B. Davidson
Dept. E&E Engineering
University of Stellenbosch
Stellenbosch 7600, South Africa
Tel: +27 21 808 4458
Fax: +27 21 808 4981
E-mail: davidson@sun.ac.za

Foreword by the Editor

MATLAB continues to entrench itself as one of the leading platforms for rapid algorithm prototyping in computational engineering, and computational electromagnetics is no exception. In the April 2004 column, the use of *MATLAB* to visualize three-dimensional radiation patterns was addressed. In the present column, the extension of that work to address conformal arrays is described. We thank the authors for their continuing contributions.

Whilst the focus of this paper is on visualization using *MATLAB*, its core numerical-analysis capabilities probably still remain the main reason for *MATLAB*'s popularity. A good overview of this may be found in [1], which the editor recently acquired.

Reference

1. C. B. Moler, *Numerical Computing with MATLAB*, (revised reprint), Philadelphia, SIAM, 2008.

A *MATLAB* Tool for Visualizing the 3D Polar Power Patterns and Excitations of Conformal Arrays

J. C. Brégains, J. A. García-Naya, A. Dapena, and M. González-López

Electronic Technology and Communications Group, Department of Electronics and Systems
University of A Coruña
Faculty of Informatics, 15701 A Coruña, Spain
Tel: +34 981167000-1350; Fax: +34 981167160
E-mail: julio.bregains@udc.es; jagarcia@udc.es; adriana@udc.es; mgonzalezlopez@udc.es

Abstract

This paper describes the implementation of a *MATLAB*® tool to plot the three-dimensional polar radiation diagram generated by a conformal antenna array. A related tool has also been developed for visualizing the geometrical arrangement (positions and orientations) of the elements, as well as the relative values of the amplitudes and phases of their excitations. Supporting M-files are available.

Keywords: Visualization; computer graphics; conformal antennas; antenna arrays; antenna radiation patterns; *MATLAB*

1. Introduction

Currently, there is a broad set of computer tools available for any antenna designer for manipulating equations, numerical algorithms, and/or plotting two-dimensional (2D) or three-dimensional (3D) functions. In this sense, there is no doubt that *MATLAB* [1, 2] lately reigns supreme as the most widely used numerical computing environment for performing calculations applied to technical sciences and engineering. This is not surprising, because with a minimum knowledge about basic *MATLAB* functions, it is possible not only to manipulate data and carry out accurate computations in a fast and simple way, but to also create very instructive graphics in little time.

In an earlier work [3], a *MATLAB* tool to plot both the power pattern and the excitation (amplitude and phase) distribution of a planar array antenna was presented. That tool had the drawback of not being designed to represent power patterns generated by conformal arrays (namely, an array the geometry of which was adapted to a previously established, not necessarily flat, surface [4]). Such a drawback was a consequence of not taking into account elements having a three-dimensional spatial distribution with rotations associated with them. Based upon the source code described in that work, here we present a tool capable of representing the polar power patterns radiated by conformal antenna arrays, as well as their geometrical and electrical configurations.

2. The Method

Here, we first mathematically describe the problem, and then describe the proposed tool to solve it.

2.1 Mathematical Description: Radiation Emitted by a Conformal Array

Suppose we have a set of N antennas in a bounded space (see Figure 1). For simplicity, consider all of them to have the same geometrical and electrical configuration in such a way that if any of them is located at the center of a global coordinate system (x,y,z), it will radiate a complex-valued field $f_e(\theta,\varphi)$ at a very distant point, P, specified by the position vector, $\mathbf{R}$. As each element n is not only displaced from the center to a certain position $\mathbf{r}_n$, but is also correspondingly rotated so as to keep associated with it a local coordinate system (x_n,y_n,z_n), the field at point P becomes

$$f_e(\theta_n,\varphi_n)e^{j\mathbf{k}\cdot\mathbf{r}_n}, \tag{1}$$

where $\mathbf{k}=(2\pi/\lambda)\mathbf{a}_R$ is the wave vector [5], with λ representing the wavelength at the working frequency of the array, and $\mathbf{a}_R$ is the unitary vector pointing radially towards P. As can be seen from Figure 1, now the (θ,φ) angles are measured from the local coordinate system, thus becoming (θ_n,φ_n). It can also be observed that P being a point located in the far field [5], $\mathbf{R}$ and $\mathbf{R}_n$ (which are virtually pointing to the infinity) can be considered to be parallel vectors.

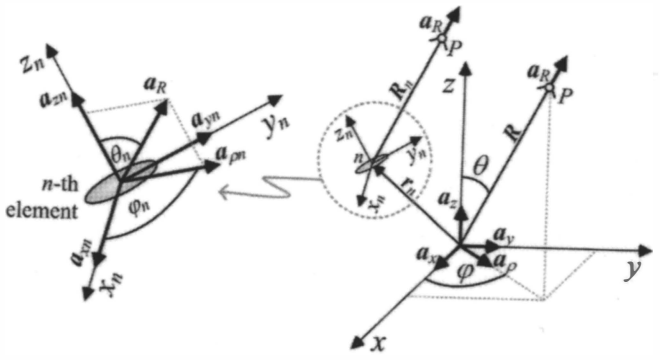

Figure 1. A set of elements of a conformal array in a bounded space, considering a global Cartesian coordinate system (x,y,z). Element n is characterized by its local coordinates (x_n,y_n,z_n) at the left-hand side, which are required to calculate the field radiated at a distant point P.

The complex field radiated by the whole array is then obtained after applying the superposition principle (and for simplicity neglecting the electromagnetic coupling between contiguous elements)

$$F(\theta,\varphi)=\sum_{n=1}^{N} f_e(\theta_n,\varphi_n)I_n e^{j\mathbf{k}\cdot\mathbf{r}_n} \tag{2}$$

The (complex-valued) phasors $I_n=|I_n|e^{j\Phi_n}$ specify the amplitude factor, $|I_n|$, and the relative phase, Φ_n, applied to the nth element's signal. As usual, F can be measured in dB: $|F(\theta,\varphi)|_{\text{dB}}=10\log_{10}|F|^2$, with $|F|^2|F|^2$ representing the (averaged-over-time) power density (W/m^2).

The greatest difficulty in numerically calculating Equation (2) lies in the proper computation of the (θ_n,φ_n) angles, which requires obtaining the local coordinate system associated with each element. To this end, the traditional approach consists of establishing three consecutive rotations for each element until the proper position is reached. Those rotations are specified by the well-known Euler angles [6], which make possible the implementation of appropriate coordinate transformations through matrix multiplications. As an intuitive alternative, we propose to avoid such transformations in the implementation, instead directly using as inputs the direction angles belonging to each axis of the local coordinate system associated with each element. For example, the x_n axis has associated with it three angles, $\alpha_{xn},\beta_{xn},\gamma_{xn}$, measured with respect to the x, y, and z axes, respectively; in the same way, the $\alpha_{yn},\beta_{yn},\gamma_{yn}$ angles are related to the y_n axis. Each set of (six) angles allows the calculation of the corresponding direction cosines, and, consequently, the linked unitary vectors. Once the unitary vectors of those axes are obtained, it is possible to further calculate the unitary vector of z_n by means of cross vector multiplication. Next, once the x_n, y_n, and z_n axes are calculated, the angles θ_n and φ_n are obtained with the following expressions (see Figure 1):

$$\theta_n = \text{acos}\left(\mathbf{a}_R \bullet \mathbf{a}_{zn}\right)$$
$$\varphi_n = \text{atan}\left(\frac{a_{\rho y_n}}{a_{\rho x_n}}\right), \tag{3}$$

with

$$\mathbf{a}_{\rho n} = \frac{\mathbf{a}_R - \mathbf{a}_{Zn}\cos\theta_n}{\sin\theta_n} = a_{\rho nx}\mathbf{a}_x + a_{\rho ny}\mathbf{a}_y + a_{\rho nz}\mathbf{a}_z$$
$$= a_{\rho nx_n}\mathbf{a}_{xn} + a_{\rho ny_n}\mathbf{a}_{yn} + a_{\rho nz_n}\mathbf{a}_{zn} \tag{4}$$

being the components

$$a_{\rho nx_n} = a_{\rho nx}x_{xn} + a_{\rho ny}y_{xn} + a_{\rho nz}z_{xn}$$
$$a_{\rho ny_n} = a_{\rho nx}x_{yn} + a_{\rho ny}y_{yn} + a_{\rho nz}z_{yn} \tag{5}$$
$$a_{\rho nz_n} = a_{\rho nx}x_{zn} + a_{\rho ny}y_{zn} + a_{\rho nz}z_{zn}$$

obtained after considering the direct and inverse transformations $\mathbf{T}$ and $\mathbf{T}^{-1}$, respectively:

$$\begin{bmatrix}\mathbf{a}_{xn}\\ \mathbf{a}_{yn}\\ \mathbf{a}_{zn}\end{bmatrix} = \begin{bmatrix}x_{nx} & x_{ny} & x_{nz}\\ y_{nx} & y_{ny} & y_{nz}\\ z_{nx} & z_{ny} & z_{nz}\end{bmatrix}\begin{bmatrix}\mathbf{a}_x\\ \mathbf{a}_y\\ \mathbf{a}_z\end{bmatrix} \Rightarrow \mathbf{a}_n = \mathbf{T}\mathbf{a}$$

$$\begin{bmatrix}\mathbf{a}_x\\ \mathbf{a}_y\\ \mathbf{a}_z\end{bmatrix} = \begin{bmatrix}x_{xn} & x_{yn} & x_{zn}\\ y_{xn} & y_{yn} & y_{zn}\\ z_{xn} & z_{yn} & z_{zn}\end{bmatrix}\begin{bmatrix}\mathbf{a}_{xn}\\ \mathbf{a}_{yn}\\ \mathbf{a}_{zn}\end{bmatrix} \Rightarrow \mathbf{a} = \mathbf{T}^{-1}\mathbf{a}_n \tag{6}$$

It is important to note that the use of the expression on the right-hand side of Equation (3) instead of the simpler[1] $\varphi_n = \arccos\left(\mathbf{a}_{pn} \bullet \mathbf{a}_{xn}\right)$ was justified by the fact that φ_n ranges from 0 to 2π, a condition that is fulfilled by using the `atan2()` function – a proper *MATLAB* adaptation of `atan()` – instead of `acos()`.

2.2 The *MATLAB* Tool

From now on, we will briefly describe the application that has been addressed.

The tool is started by invoking it in the *MATLAB* command line. During the loading process, two input files are read. The first one contains the $\mathbf{r}_n = (\xi_n, \psi_n, \zeta_n)$ vector components – measured in units of λ – of the position of each of the N elements. This is followed by the amplitude, $|I_n|$, and phase, Φ_n (measured in degrees, between 0° and 360°), of their complex excitations, I_n. Consequently, the input file will have those values arranged in five columns and N rows. The second file contains the direction angles of the local axes, x_n and y_n, of each element, thus being arranged in six columns and N rows.

After reading those files and storing the data into the computer's memory, with the aid of the direction angles the program calculates the unitary vectors $\mathbf{a}_{xn}$ and $\mathbf{a}_{yn}$ (see Figure 1). The program next proceeds to obtain the remaining unitary vectors $\mathbf{a}_{zn}$ by utilizing the cross product.

In a further step, the program asks the user whether or not the spatial distribution of the elements is to be plotted. If the plot is requested, then the conformal array is shown as a set of circles, each circle representing an element of the array. The program automatically determines the maximum size of the array (in units of λ) in order to plot a set of Cartesian axes the proportions of which fit the axes of the array. The program can also handle the one-element case (located at the center of the coordinate system), which allows the user to study the $f_e(\theta,\varphi)$ function of a single antenna.

Afterwards, the program asks for the excitation amplitude distribution to be plotted. If requested, for each element of the array a corresponding plot shows a cone with an axis of revolution that coincides with the z_n axis of such an element, and a height that is proportional to the corresponding normalized excitation amplitude (i.e., the program normalizes the amplitudes, making $0 \le |I_n| \le 1$, $\forall n \in [1, N]$). The constant factor of proportionality, A_f, applied to the height will be a value to be specified by the user in the code. In this way, $A_f = 1$ (for example) indicates that when the oriented amplitude distribution is represented, the cones are measured in wavelength units, and their heights vary between 0 (for $|I_n| = 0$) and λ (for $|I_n| = 1$). In the same way, R_f corresponds to a factor for the radius of the cones. In order to represent the conical surfaces, it is necessary to use parametric equations. It is important to note that these parametric equations must be established in global coordinates. Under this assumption, by considering the direct transformation given by the left-hand side of Equation (6), we arrive at the following expression for the parametric representation of the nth cone:

$$\mathbf{C}_n = \left\{R_f(1-t)\left[\cos(2\pi t)x_{nx} + \sin(2\pi t)y_{nx}\right] + A_f|I_n|tz_{nx} + \xi_n\right\}\mathbf{a}_x$$
$$+\left\{R_f(1-t)\left[\cos(2\pi t)x_{ny} + \sin(2\pi t)y_{ny}\right] + A_f|I_n|tz_{ny} + \psi_n\right\}\mathbf{a}_y$$
$$+\left\{R_f(1-t)\left[\cos(2\pi t)x_{nz} + \sin(2\pi t)y_{nz}\right] + A_f|I_n|tz_{nz} + \zeta_n\right\}\mathbf{a}_z \tag{7}$$

with the parameter t ranging from 0 to 1.

Equation (7) can be handled more efficiently in *MATLAB* by considering matrix operations, as observed in the portion of the source code shown in Figure 2, where the meaning of each variable is easily deduced from its name. Once the amplitude-oriented cones are represented, the program asks the user to specify whether the phase distribution is to be plotted or not, this time by means of vertical (non-oriented) cones. For such a case, the relative phases are specified by heights ranging from zero ($\Phi_n = 0°$) to λ ($\Phi_n = 360°$), conveniently resized by a proportionality factor, P_f. Finally, the program gives the option of plotting the three-dimensional polar radiation diagram, whether over the upper spatial hemisphere ($0 \le \theta \le 90°$, $0 \le \varphi \le 360°$) or over the whole space ($0 \le \theta \le 180°, 0 \le \varphi \le 360°$). The normalized radiation pattern is

[1] The dot product $\mathbf{a}_{pn} \bullet \mathbf{a}_{xn}$ would not need any inverse transform $\mathbf{T}^{-1}$ previously calculated: see Equations (6).

```
hold on;                              % Allow drawing the cones in the active window
t = 0:1/14:1;                         % Parametric variable vector t
th=2*pi*t;                            % Parametric angle
costh=cos(th); sinth=sin(th);         % cos(2*pi*t) and sin(2*pi*t) components
rcone=Rf*(1-t);                       % Variable radius of cones
ident=ones(1, size(t,2));             % Identity vector (same size as vector t)
rident=Rf*ident;                      % Constant radius of the base of the cones
 for(n=1:N)                           % For each element, build up the parametric matrices
          Xpc=rcone'*(costh*Xn(n,1)+sinth*Yn(n,1))+Af*AmpIn(n,1)*t'*ident*Zn(n,1)+rn(n,1);
          Ypc=rcone'*(costh*Xn(n,2)+sinth*Yn(n,2))+Af*AmpIn(n,1)*t'*ident*Zn(n,2)+rn(n,2);
          Zpc=rcone'*(costh*Xn(n,3)+sinth*Yn(n,3))+Af*AmpIn(n,1)*t'*ident*Zn(n,3)+rn(n,3);
         %  Now plot the nth cone by using Xpc, Ypc and Zpc matrices…
       ps=surf(Xpc,Ypc,Zpc,'EdgeColor', 'red', 'EdgeAlpha', 0.40, 'LineWidth', 0.50);
    % Set the cone color to red, and apply certain degree of transparency …
       set(ps,'FaceColor',[1.000 0.502 0.502]); alpha(0.75);
end;
hold off;                             % The control over the active window is no longer necessary
```

Figure 2. The implementation of Equation (7) and plotting of the cones in *MATLAB*.

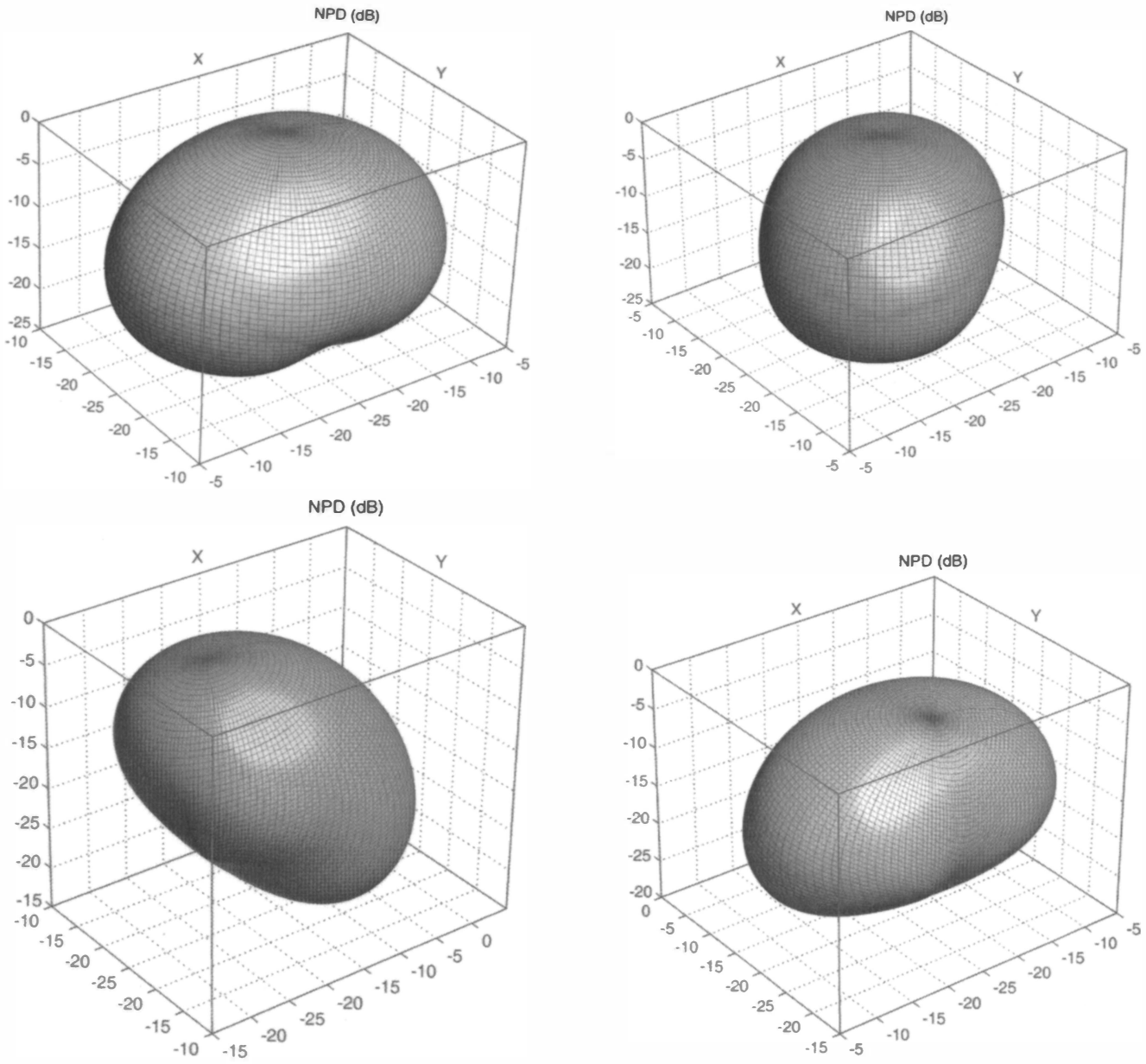

Figure 3. The power patterns radiated by a $\lambda/2$ dipole, backed by a ground plane located a distance $-\lambda/4$ away. Top left: The pattern for a dipole aligned along the y axis. Top right: The pattern for a dipole rotated 45° with respect to the z axis. Bottom left: The pattern when the ground plane rotated –45° with respect to the y axis. Bottom right: The pattern for a dipole rotated –30° with respect to the x axis.

now represented by means of a parametrized polar surface. The radial coordinate of such a surface, being a function of the θ and φ parameters, indicates the power-density value expressed in dB. The procedure for plotting the power pattern is derived from the source code described in [3], but by considering a convenient adaptation to Equation (2), and using the transformations given in Equations (3) to (6). The interested reader is referred to [3] for more information about the abovementioned source code.

3. Examples

This section is devoted to presenting several examples, in order to show the scope as well as the limitations of the implementation.

3.1. Half-Wavelength Dipole Subject to Several Rotations

As a first example, we considered the radiation diagram of a single $\lambda/2$ dipole, aligned along the y axis, centered on the origin of the coordinate system, and backed by an infinite ground plane parallel to the $z=0$ plane (and located at $z=-0.25\lambda$). This first example just served to verify the appropriate rotation of the radiation diagram, according to the rotation applied to the dipole itself. The far-field element factor, $f_e(\theta,\varphi)$, used in this case was the factor specified in [7], conveniently adapted (null radiation below the ground plane).

Figure 3 shows the patterns obtained. The first pattern (top left) corresponded to a radiation diagram obtained when the dipole was along the y axis. In this case, the local axes of the dipole matched the global axes: $x_1 = x$, $y_1 = y$, $z_1 = z$, or (which is the same) $\alpha_{x1}=0°$, $\beta_{x1}=90°$, $\gamma_{x1}=90°$, $\alpha_{y1}=90°$, $\beta_{y1}=0°$, $\gamma_{y1}=90°$. The top right plot represents the pattern after rotating the dipole (and the ground plane) 45° degrees about the z axis, i.e., $\alpha_{x1}=45°$, $\beta_{x1}=45°$, $\gamma_{x1}=90°$, $\alpha_{y1}=135°$, $\beta_{y1}=45°$, $\gamma_{y1}=90°$. The bottom left plot was obtained by rotating the ground plane 45° about the y axis (it was not necessary to rotate the dipole): $\alpha_{x1}=45°$, $\beta_{x1}=90°$, $\gamma_{x1}=135°$, $\alpha_{y1}=90°$, $\beta_{y1}=0°$, $\gamma_{y1}=90°$. Finally, the bottom right plot shows the pattern after rotating the dipole 30° about the x axis. In all of the plots, the acronym NPD [dB] stands for normalized power density (expressed in decibels). For the sake of brevity, the single oriented cones were not represented in this example.

3.2. Semi-Cylindrical Array of Dipoles with Ground Planes

We departed from the expression of the element factor $f_e(\theta,\varphi)$ of a $\lambda/2$ dipole aligned along the z axis, with a ground plane parallel to the $x=0$ plane and located a distance 0.25λ from the dipole [7]. In this way, 50 dipoles were conformed onto a semi-cylindrical surface with a base radius equal to 2λ and a total height of 3.3λ (measured from the extremes of the top and bottom dipoles), so that the main axes of the dipoles, z_n, were tangential to the cylinder. The dipoles were arranged in five rows. The vertical separation between contiguous dipoles was 0.7λ, with each row containing 10 dipoles uniformly distributed along their corresponding semi-circles. The ground plane of each dipole was also tangential to the semi-cylindrical surface. The excitations of the dipoles were the result of multiplying (i.e., by using separable distributions [7]) those of a two linear arrays with triangular excitations ($I_{nHoriz}=\{1,2,3,4,5,5,4,3,2,1\}$ and $I_{mVert}=\{1,2,3,2,1\}$). Figure 4 shows the spatial distribution of the dipoles, represented by circles, as mentioned above. Figure 5 shows the amplitude-excitation distribution, with the cones oriented along the axis of each dipole. As $\Phi_n = 0°$ for all the elements, it was not necessary to represent the phase distribution. The power-density pattern generated by this conformal array is shown in Figure 6.

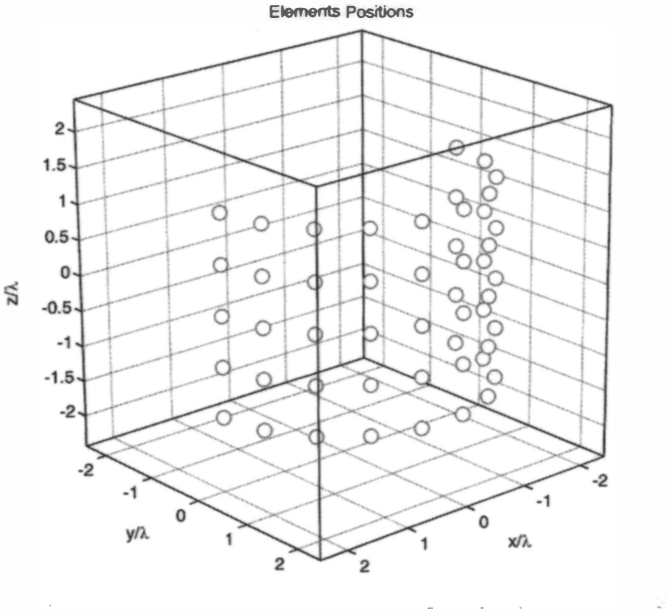

Figure 4. An arrangement of 50 dipoles on a semi-cylinder with a base radius equal to 2λ and a height equal to 3.3λ (see text). The vertical separation between contiguous dipoles was 0.7λ. Each circle represents a dipole.

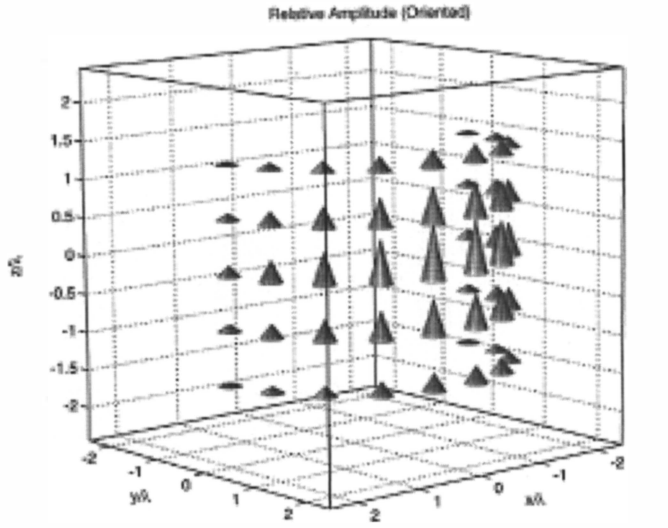

Figure 5. The amplitude distribution of the dipoles conformed on the semi-cylinder given in Figure 3. The lengths of the cones indicate the relative excitation amplitudes (see text), and their main axes coincide with the axes of the corresponding dipoles.

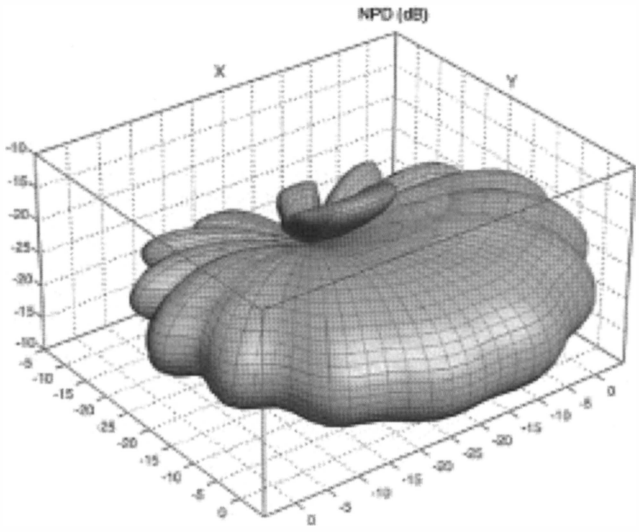

Figure 6. A three-dimensional polar plot of the power pattern of a conformal array of 50 dipoles with ground planes, arranged over a semi-cylinder (see Figure 4).

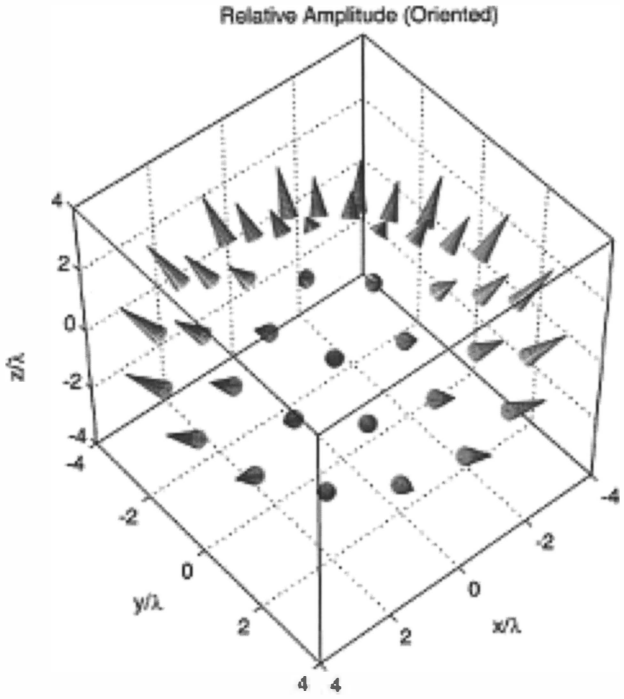

Figure 7. The amplitude distribution of 40 patches conformed over the surface given in Figure 3. The length of each cone indicates the relative excitation amplitude of each patch (see text), and the main axis of each cone coincides with the normal to the corresponding patch.

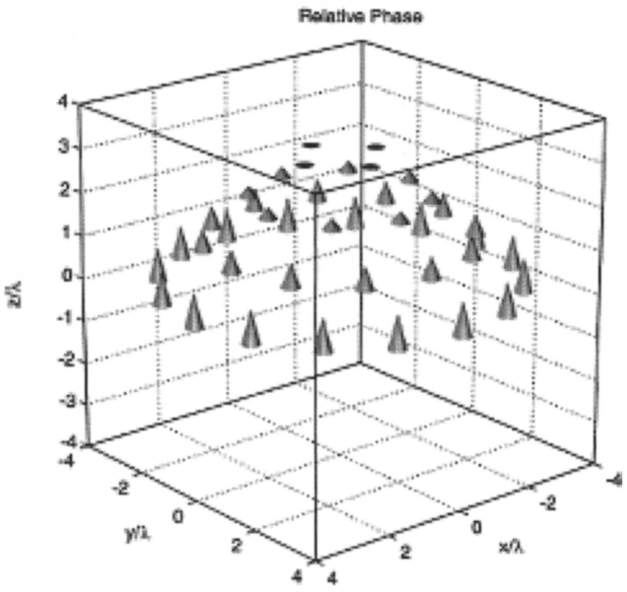

Figure 8. The relative phase distributions (given by the lengths of the little cones) of the patches conformed over a conical surface (see text).

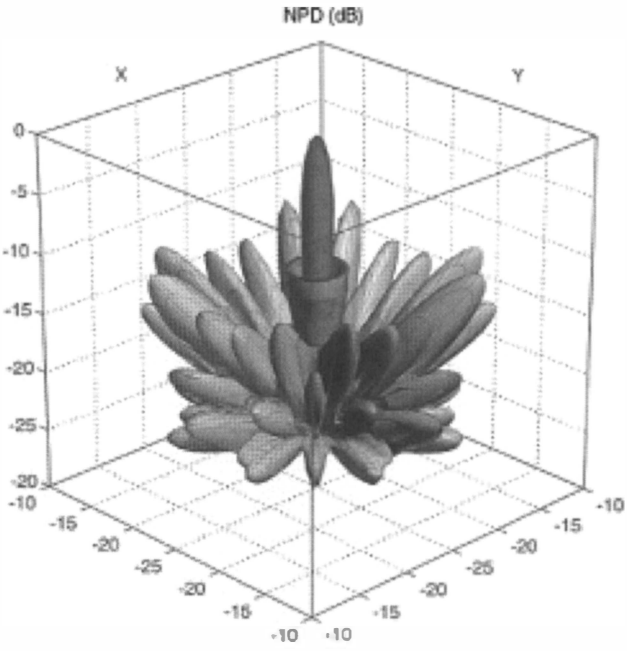

Figure 9. A three-dimensional polar plot of the power pattern of a conformal array of 40 patches arranged on a cone (see Figures 6 and 7).

3.3. Conical Array of Circular Patches

In this example, we considered an element factor $f_e(\theta,\varphi) = \cos^2\theta$, which corresponded to the theoretical simple model of a circular patch (radiating only in the upper hemisphere). Forty patches were conformed on the same conical surface used in the previous example, with the same distribution given for the dipoles, but oriented in such a way that the normal to each patch was also normal to the conical surface. The excitation distribution was as follows: the sixteen patches located at the edge of the base (first row) were fed with $I_n = e^{j135°}$, the twelve patches of the second row were uniformly excited with $I_n = 0.75e^{j90°}$, the eight patches of the following row had $I_n = 0.75e^{j45°}$, and for the four patches of the last row, $I_n = 0.25e^{j0°}$. Figures 7 and 8 show the oriented amplitude and non-oriented phase distributions, respectively. Figure 7 was drawn by considering that each cone indicated a normal to the corresponding patch. Finally, the pattern radiated by this configuration is given in Figure 9.

4. Closing Comments

The tool presented is useful for analyzing not only the geometrical configuration but the electrical behavior of conformal arrays, as well as the power patterns radiated by them. This is not only recommended as a diagnosis tool for antenna designers, but as an excellent complement for any educator whose lectures include the essentials of conformal arrays.

It is important to remark that the program was codified by bearing in mind the fundamental pedagogical use of the tool; therefore, it has not been optimized in terms of either code length or computing time.

The source code is freely available at http://gtec.des.udc.es/web/images/files/conformalpow3d.zip.

5. Acknowledgements

This work was partially supported by Xunta de Galicia, Ministry of Science and Innovation, and FEDER funds from the European Union (projects 09TIC008105PR, TEC2007-68020-C04-01, CSD2008-00010).

6. References

1. The MATLAB Group, Inc., "MATLAB Function Reference: Volumes 1-2-3," (printable version taken from Help menu in *MATLAB*), 2008.

2. The MATLAB Group, Inc., "MATLAB: 3-D Visualization," (printable version taken from Help menu in *MATLAB*), 2008.

3. J. C. Brégains, F. Ares, and E. Moreno, "Visualizing the 3D Polar Power Patterns and Excitations of Planar Arrays with *MATLAB*," *IEEE Antennas and Propagation Magazine*, **46**, 2, 2004, pp. 108-112.

4. L. Josefsson and P. Persson, *Conformal Array Antenna Theory and Design*, New York, IEEE Press/John Wiley, 2006.

5. C. A. Balanis, *Antenna Theory, Analysis and Design, Third Edition*, New York, Wiley Interscience, 2005.

6. T. Milligan, "More Applications of Euler Rotation Angles," *IEEE Antennas and Propagation Magazine*, **41**, 4, 1999, pp. 78-83.

7. R. S. Elliott, *Antenna Theory and Design, Revised Edition*, New York, Wiley Interscience, 2003.