Plotting Circularly Polarized Field Patterns Using Processed NEC 4 Output Files

J. B. Knorr
Naval Postgraduate School
Monterey, California

B. Neta
Naval Postgraduate School
Monterey, California

I. Introduction

NEC4 computes E_θ and E_φ , the linearly polarized components of the far field radiated by an antenna. Software packages such as GNEC, which are built around the NEC4 engine [Ref. 1], allow the user to plot far field patterns for E_θ and E_φ from the data in the NEC4 output file. When the far field of an antenna is elliptically polarized, the NEC4 output file indicates the sense of polarization, right or left hand. While this information is useful, antennas that radiate a field that is basically circularly polarized are better characterized by their right and left hand circularly polarized field patterns. The cavity backed spiral, for example, is an antenna that is used in applications where circular polarization is desired and one would prefer to see the far field patterns of the right and left hand components, E_{right} and E_{left}. This note describes a simple approach to plotting the patterns for E_{right} and E_{left} by processing the data in the standard NEC4 output file.

II. Approach

The standard NEC4 output file provides the user with the magnitude and angle of E_θ and E_φ . Most user interfaces simply allow the user to access this data to plot the far field patterns for E_θ and E_φ . The current version of GNEC is an example. If the NEC4 output file is processed and E_θ and E_φ are replaced with E_{left} and E_{right}, then any plotter designed to extract data from the NEC4 output file can be used to display left and right hand circularly polarized field patterns instead of the usual linearly polarized field patterns. The required processing is straightforward. A Matlab program is provided in Appendix 1.

III. Field Decomposition

The linearly polarized field components E_θ and E_φ can be decomposed quite simply into left and right hand circularly polarized components. Write

$$\mathbf{E} = E_\theta \mathbf{a}_\theta + E_\varphi \mathbf{a}_\varphi \quad \text{or}$$

$$= \underbrace{0.5E_\theta[\, \mathbf{a}_\theta + j\mathbf{a}_\varphi \,]}_{\text{(LHC)}} + \underbrace{0.5E_\theta[\, \mathbf{a}_\theta - j\mathbf{a}_\varphi \,]}_{\text{(RHC)}} \underbrace{- j0.5E_\varphi[\, \mathbf{a}_\theta + j\mathbf{a}_\varphi \,]}_{\text{(LHC)}} + \underbrace{j0.5E_\varphi[\, \mathbf{a}_\theta - j\mathbf{a}_\varphi \,]}_{\text{(RHC)}}.$$

In the above expressions, bold type denotes vector quantities, **a** is the unit vector, LHC indicates left hand circular and RHC indicates right hand circular polarization. Now write

$$\mathbf{E} = \mathbf{E}_{left} + \mathbf{E}_{right}$$

where

$$\mathbf{E}_{left} = 0.5[E_\theta - jE_\varphi][\mathbf{a}_\theta + j\mathbf{a}_\varphi] \text{ and}$$

$$\mathbf{E}_{right} = 0.5[E_\theta + jE_\varphi][\mathbf{a}_\theta - j\mathbf{a}_\varphi].$$

IV. Field Patterns of an Eight Arm Cavity Backed Spiral

The eight arm cavity backed spiral is an interesting antenna that is often used in applications requiring broad bandwidth and circular polarization. The spiral can be excited to produce a sum beam (mode 1) or a difference beam (mode 2). This antenna serves as a good example for illustrating the difference between linearly and circularly polarized field patterns.

A. Mode 1

To produce a sum beam, the eight arms of the spiral are excited with a 45 degree

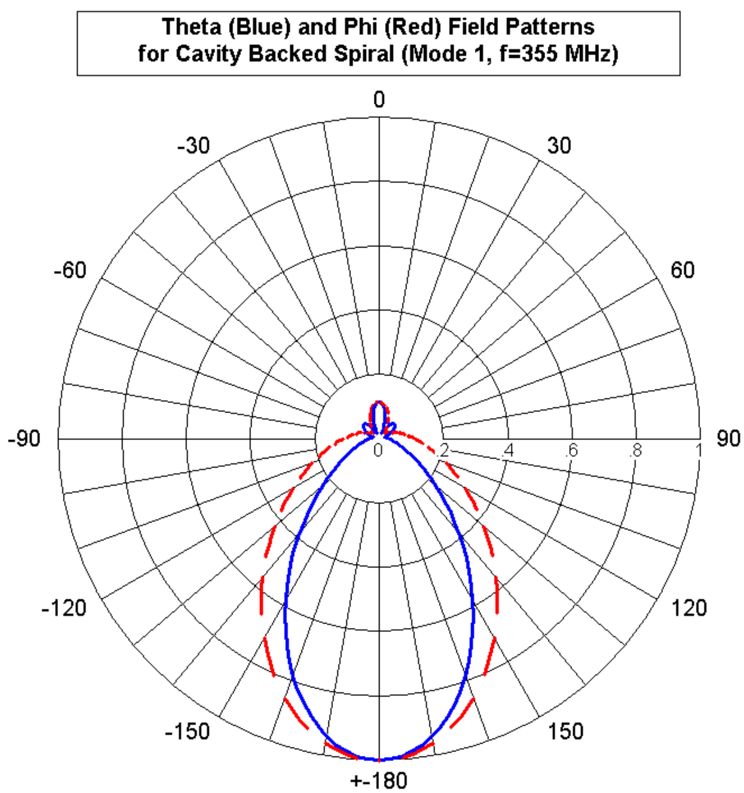

Figure 1. Linearly polarized field patterns for mode 1 excitation of an eight arm cavity backed spiral antenna. Patterns are in a plane containing the axis of the spiral (normal to plane of spiral). E_θ solid line, E_φ dashed line.

phase shift from one arm to the next. This results in opposing arms being excited in phase. Figure 1 shows the linearly polarized field pattern and Figure 2 shows the circularly polarized field patterns.

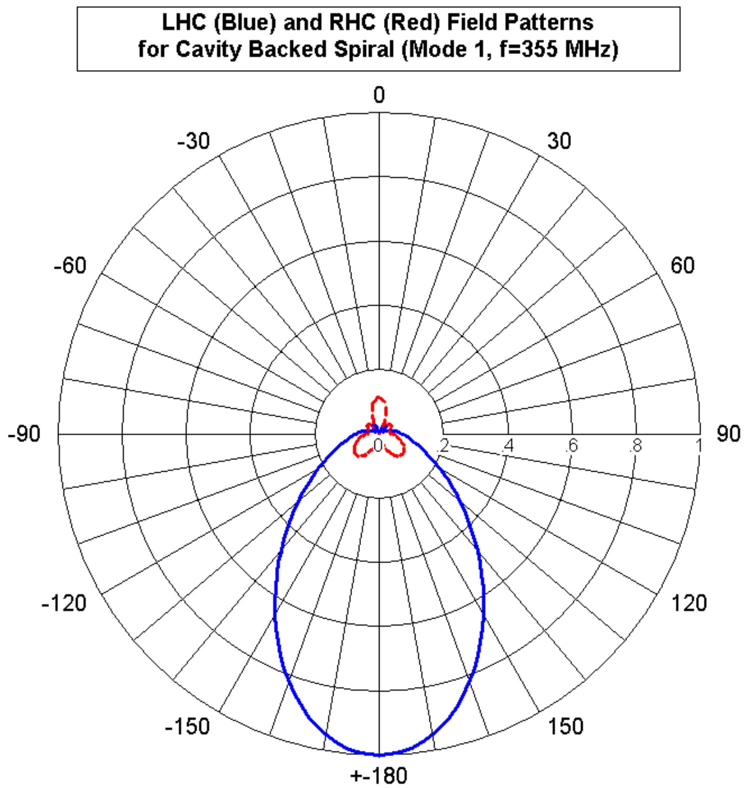

Figure 2. Circularly polarized field patterns for mode 1 excitation of an eight arm cavity backed spiral antenna. Patterns are in a plane containing the axis of the spiral (normal to plane of spiral). E_{left} solid line, E_{right} dashed line.

An inspection of Figures 1 and 2 reveals the patterns for the linearly and circularly polarized fields are very different. One can determine the relative magnitudes of the linearly polarized field components from the patterns displayed in Figure 1 but without phase information, any other field pattern is impossible to deduce.

B. Mode 2

To produce a difference beam, the eight arms of the spiral are excited with a 90 degree phase shift from one arm to the next. This results in opposing arms being excited 180 degrees out of phase so that in a plane normal to the plane of the spiral and containing the axis of the spiral, the far field contributions of the opposing arms will cancel, producing a null in the pattern. Figure 3 shows the linearly polarized field patterns and Figure 4 shows the circularly polarized field patterns. Again, the difference between the patterns for the linearly polarized field components displayed in Figure 3 and the circularly polarized field

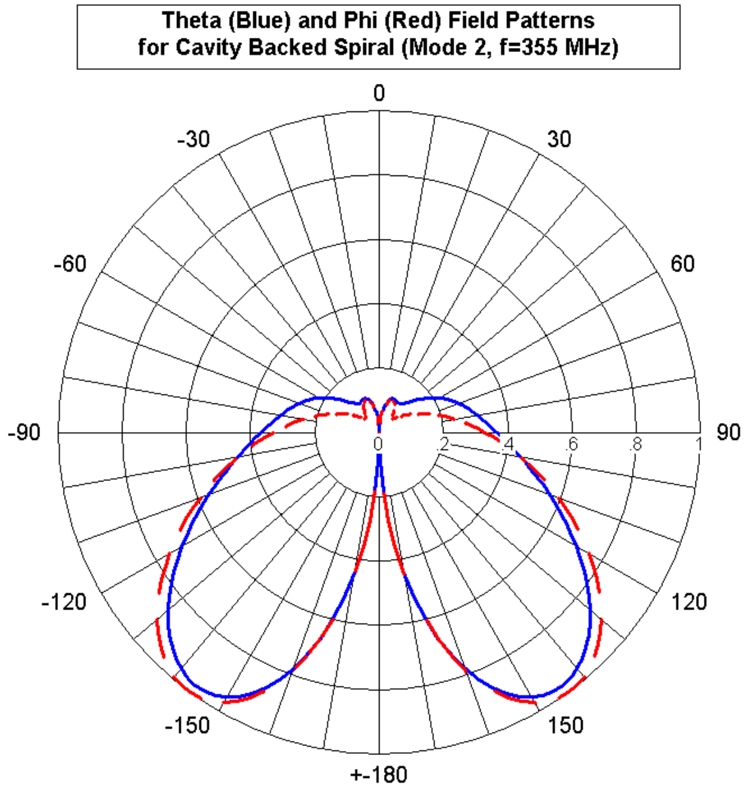

Figure 3. Linearly polarized field patterns for mode 2 excitation of an eight arm cavity backed spiral antenna. Patterns are in a plane containing the axis of the spiral (normal to plane of spiral). E_θ solid line, E_φ dashed line.

components displayed in Figure 4 are evident. One could not deduce the patterns in Figure 4 from the patterns in Figure 3.

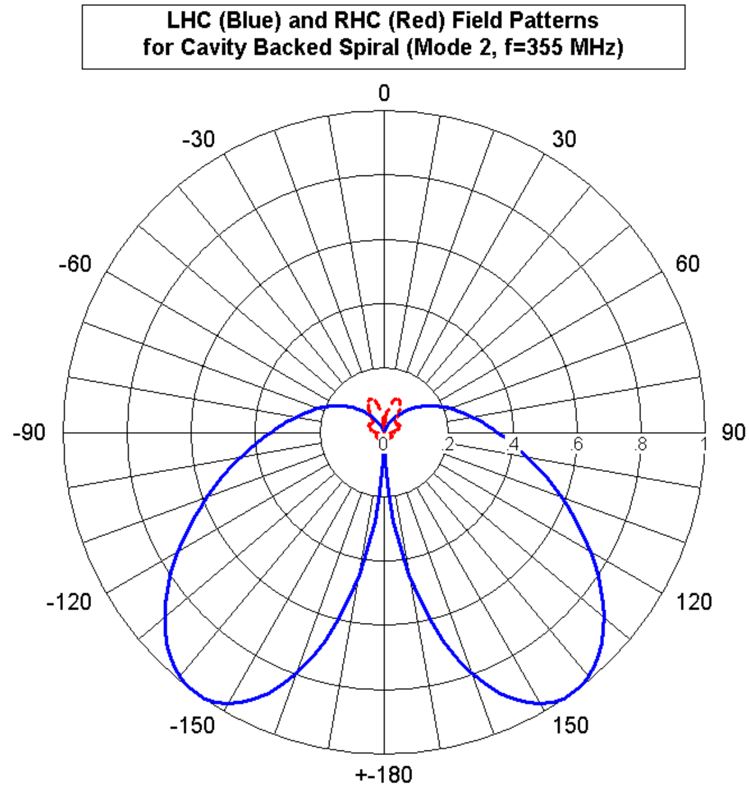

Figure 4. Circularly polarized field patterns for mode 2 excitation of an eight arm cavity backed spiral antenna. Patterns are in a plane containing the axis of the spiral (normal to plane of spiral).

V. Conclusions

A simple approach to displaying the circularly polarized field patterns of an antenna analyzed using NEC4 has been presented. Equations have been provided for decomposition of the linearly polarized field into left and right hand circularly polarized components. If the normal NEC4 output file is processed and E_{left} and E_{right} are substituted for E_θ and E_φ in the output data file, then any plotter capable of extracting data from the normal NEC4 output file can be used to display the left and right hand circularly polarized field patterns of the antenna. Work is under way to implement this display capability in a future release of GNEC.

References

[1] GNEC, Nittany Scientific, http://www.nittany-scientific.com/.

Appendix 1. Matlab Code

```
% This matlab program
% (1) reads a NEC4 output file,
% (2) computes left and right hand circularly polarized fields from the linearly polarized
% theta and phi components and
```

```
% (3) replaces E theta with E left and E phi with E right.
% The program prompts the user for an input file name (NEC4 output file).
% The program also asks the user for the output file name (it overwrites the file).
% For each of these there is a default which is taken if the question is not answerd.
% The number of Thetas will be given on the screen at the end of the run.
clear
help lrnec;
%  Input the default values in the variables below:
default_infile=('PSAT_PAT.nou');
default_outfile=('psatpat.nou');
%  Prompt user for input values.  The default value is used
%  If carriage return is selected without inputting data.
disp(' '); % Line Space on screen
infile=input('Type in the input file name:  ','s');
if isempty(infile)
   infile = default_infile;
end
disp(' '); % Line Space on screen
outfile=input('Type in the output file name:  ','s');
if isempty(outfile)
   outfile = default_outfile;
end
disp(' '); % Line Space on screen
%  Display selected values.  Comment out after debugging:
infile
outfile
% open file to write
%
fid_out=fopen(outfile,'w')
%
% open the file to read
%
fid_v=fopen(infile,'r')
if fid_v > 0
  sprintf('v file opened:')
end
%     Read the file
%          ... read each text header line
while 1
%          ... This WHILE LOOP to READ EACH DATA LINE is an
%              infinite loop, since "while 1" is always true.
%              The way to exit from the loop is with an if, break.
%              IF length(line) == 0   Break out of loop!!
%              The "isstr" function will test if line is a character
%              string.  The if ~isstr(line) logic means if line is
%              NOT (~) a string.  So if ~isstr(line) is TRUE when
```

```
%               line is NOT a string, and is the end-of-file.
%               When end of file is reached, "break" exits the WHILE Loop.
  line = fgetl(fid_v);
  fprintf(fid_out,'%s\n',line);
%    length(line)
  if length(line) > 72
    if line(55:72) == 'RADIATION PATTERNS'
      line(55:72)
      last =1
      break
    end
  end
end
% end of "while 1" loop, which reads the header lines
%           ... initialize a line counter
iline=0;
% read the empty line after RADIATION PATTERNS header
%
line=fgetl(fid_v);
fprintf(fid_out,'%s\n',line);
% read 3 lines of headers after that
line=fgetl(fid_v);
fprintf(fid_out,'%s\n',line);
line=fgetl(fid_v);
fprintf(fid_out,'%s\n',line);
line=fgetl(fid_v);
fprintf(fid_out,'%s\n',line);
j=0;
%  line
while 1
   line=fgetl(fid_v);  %  line of data
%           ... Check for the effective "end-of-file"
   if feof(fid_v), break, end
%           ... Increment line counter.  This is the index for each
%               data array
   if length(line) > 0
         if  line(2:9) == 'RUN TIME'
           fclose(fid_out);
           break
         end
     if  line(4:21) == 'AVERAGE POWER GAIN'
      fprintf(fid_out,'%s\n',line);
      while 1
        line=fgetl(fid_v);  %  line of data
        fprintf(fid_out,'%s\n',line);
         if length(line) > 2
```

```
          if  line(2:9) == 'RUN TIME'
%             fclose(fid_v); % --- end of v file
            fclose(fid_out);
            break
          end
        end
      end     % end of while
    else

      j=j+1;
      etm=str2num(line(77:87));
      etp=str2num(line(90:96));
      epm=str2num(line(101:111));
      epp=str2num(line(114:120));
      vet=etm*cos(etp*pi/180)+i*etm*sin(etp*pi/180);
      vep=epm*cos(epp*pi/180)+i*epm*sin(epp*pi/180);
      ver=.5*(vet-i*vep);
      vel=.5*(vet+i*vep);
      erm=sqrt(real(ver)^2+imag(ver)^2);
      erp=atan2(imag(ver),real(ver))*180/pi;
      elm=sqrt(real(vel)^2+imag(vel)^2);
      elp=atan2(imag(vel),real(vel))*180/pi;
          ab=[erm erp elm elp];
      fprintf(fid_out,line(1:72));
      fprintf(fid_out,'%15.5e%9.2f%15.5e%9.2f\n',ab);
    end
  end
end
```