Blucher Mechanical Engineering Proceedings
May 2014, vol. 1 , num. 1
www.proceedings.blucher.com.br/evento/10wccm

10th World Congress on Computational Mechanics
8-13 July 2012 • São Paulo • Brazil

COMPUTATION AND VISUALIZATION OF THREE DIMENSIONAL RADIATION PATTERNS OF ANTENNAS

M. Joaquim[1], S. Scheer[2]

[1]Department of Electronic at the Federal Technology University of Paraná (marinoel@utfpr.edu.br)

[2]Graduate Studies Program in Numerical Methods in Engineering at the Federal University of Paraná

Abstract. *This paper describes developed learning objects (or simulators) to aid in the antenna design teaching at an engineering technology level. An important motivation for the development of learning objects is to allow a quality and adapted teaching for Brazilian Universities, in a free form, without spending with an expensive commercial software. The numerical methods are often used to compute important parameters of antennas. An essential parameter of any antenna is the three-dimensional (3D) radiation pattern. With the computation of this 3D radiation pattern is possible to calculate other parameters as directivity, half-power angle, beam efficiency, levels of side lobes, and so on. Thus, an important aid in design of antennas is the visualization of 3D radiation patterns. A few learning objects was developed making use of the OpenGL graphics library to allow the visualization of three-dimensional radiation patterns of the following kinds of antennas: arrays of isotropic antennas, dipole antennas in free space and presence of perfect ground plane, and Yagi-Uda antennas. Other more adequate graphics software named Visualization Toolkit (VTK) has been used in the development of 3D simulators for antenna radiation patterns. VTK is an open-source, object-oriented software for computer graphics, visualization and image processing. Although it is large and complex, VTK is easy to be used once learned its basic object-oriented design and implementation methodology. VTK is a library of C++ classes. Nevertheless, application developers may use interpreted languages Tcl, Python and Java. In this paper it was used Python and wxPython, a cross platform toolkit, for graphical user interface creation. The wxPython is a wrapper around wxWidgets, which is a mature cross platform C++ library. One of the objectives in use Python, wxPython and VTK is to allow fast development of scientific visualization prototypes. The second objective is to work with object-oriented design and to transform easily programs in Python (with wxPython and VTK) for programs in C++ (with wxWidgets and VTK). The last objective is to build large and complex applications of scientific visualization, which will be portable, fast and without cost for the students.*

Keywords: *Radiation Patterns, Antennas, Scientific Visualization, VTK.*

1. INTRODUCTION

In the last decades, the availability of powerful and cheap personal computers has begun a revolution in teaching and instruction processes. Nowadays, there are many ways to use of computers for teaching and learning. One of these is to utilize learning objects (or simulators) to aid in the learning of technology matters, for example, antenna theory, that it is subject this paper.

Wireless technology is very present in the life of people. Some of the common applications are satellite broadcast TV, radio broadcast, radar, cordless phones, cellphones, WiFi (Wireless Fidelity), Bluetooth, and so on. The wireless technology needs the antennas to work. Thus, one important motivation is to facilitate the antenna theory learning, throughout the visualization of two-dimensional (2D) and three-dimensional (3D) radiation patterns. Another important motivation for development of the described learning objects in this paper is to allow Brazilian Universities students to have free and costless access to these learning objects.

2. NUMERICAL AND GRAPHICAL COMPUTATION OF ANTENNAS

Antennas are generally designed making use of numerical methods as the method of moment (MoM), the finite element method (FEM) and the finite-difference time-domain (FDTD) method [1]. These numerical methods are often used to compute important parameters of antennas, as 2D and 3D radiation patterns, complex impedance, and so on.

Three-dimensional radiation pattern is a fundamental parameter for analysis and design of antennas. With the computation of this 3D radiation pattern is possible to calculate other parameters such 2D radiation patterns, directivity, half-power angle, beam efficiency, as well as levels of lateral lobes. The actual radiation pattern is always three-dimensional, though generally can be divided in two 2D radiation patterns to make easy its analysis. Hence, it is very necessary the simulation and visualization of these 2D and 3D radiation patterns. The simpler antennas, for instance, dipoles and monopoles, need simple equations to generate the 2D and 3D radiation patterns. Antennas that are more complex, for instance, microstrip arrays, need numerical methods to generate 2D and 3D radiation patterns.

Another very important parameter of the antenna is its complex impedance (in ohms). It is composed for real (resistive) and imaginary (reactive) parts. The knowledge of accurate value for this impedance is necessary for a good impedance matching design between antenna and transmission line (or electronic device). In order to calculate antenna impedance is used complex equations, same for simple antennas, for instance, dipoles and monopoles, but it is easily implemented in a computer program. Antennas of more complex structures need numerical methods to calculate their impedances.

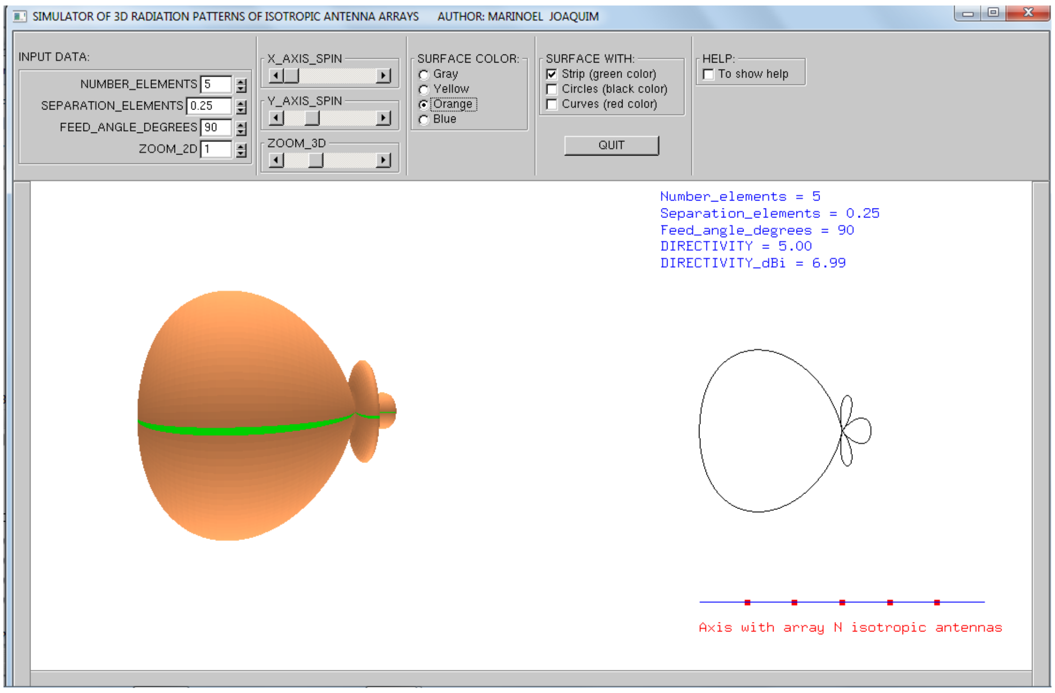
3. DESCRIPTION OF THE LEARNING OBJECTS

The first learning objects by the authors were developed making use of the Open

Graphics Library (OpenGL) [2], whereas the most recent learning objects were developed using the Visualization Toolkit (VTK) [3]. Comparison and analysis between the two approaches for the development appear along this paper.

3.1. Simulator of 3D radiation patterns for arrays of isotropic antennas

This simulator was developed making use of the OpenGL graphics library, Glut and Glu libraries, and C++ compiler. The OpenGL is an open specification and cross platform of the graphics libraries for development of sophisticated graphics applications such animations, computer games, scientific visualization and so on. The OpenGL allows utilize the inner resources and a speed of graphics processing units (GPUs) that do part of the modern computers. In the Figure 1, it is possible to notice 2D and 3D field radiation patterns for array of N elements, where one single element is an isotropic antenna.


SIMULATOR OF 3D RADIATION PATTERNS OF ISOTROPIC ANTENNA ARRAYS AUTHOR: MARINOEL JOAQUIM

INPUT DATA:
NUMBER_ELEMENTS 5
SEPARATION_ELEMENTS 0.25
FEED_ANGLE_DEGREES 90
ZOOM_2D 1

SURFACE COLOR: Gray Yellow [✓ Orange] Blue

SURFACE WITH: [✓ Strip (green color)] Circles (black color) Curves (red color)

HELP: To show help

```
Number_elements = 5
Separation_elements = 0.25
Feed_angle_degrees = 90
DIRECTIVITY = 5.00
DIRECTIVITY_dBi = 6.99
```

Figure 1. Snapshot of field radiation patterns for array of five isotropic antennas.

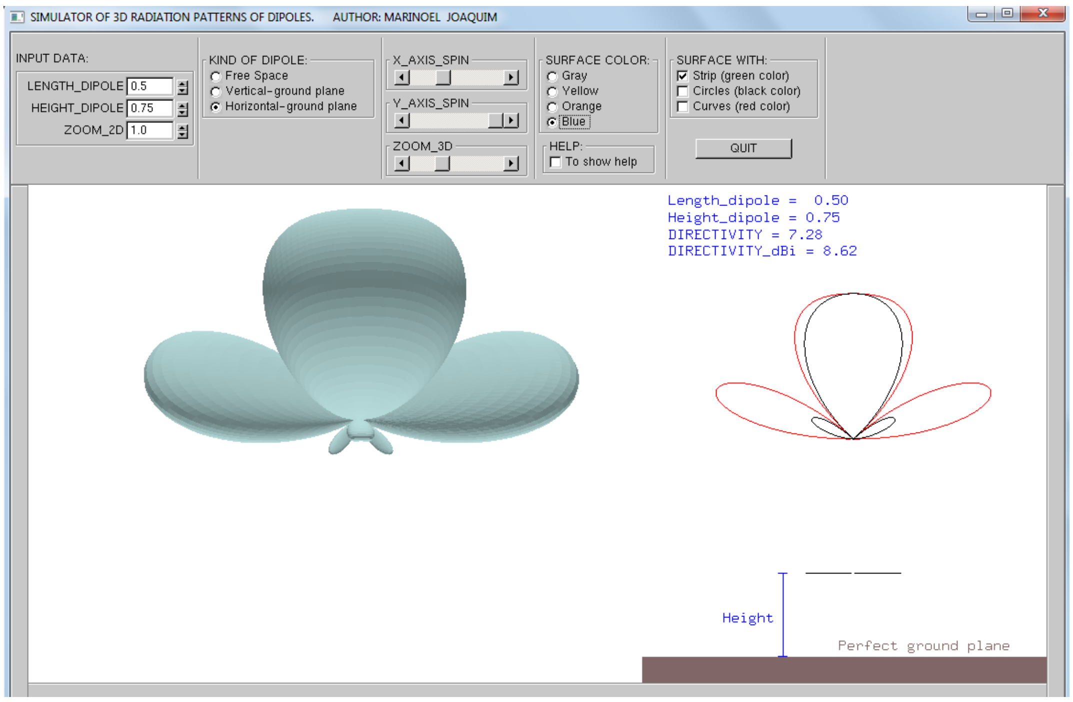
Besides the number of elements, other input parameters of the graphical user interface (GUI) are separation (in wavelengths or electric length) and current feed angle (in degrees) between elements, and zoom of 2D radiation pattern. The 3D field radiation pattern can be rotated around X and Y axis, zoomed, and changed its surface color, and so on.

3.2. Simulator of 3D radiation patterns for dipoles

This simulator also utilizes the same libraries of the former simulator. It generates 2D and 3D field radiation patterns of short or long dipoles in free space or in presence of perfect ground plane. Perfect ground plane have high conductivity as the car bodywork and is good reflector of electromagnetic waves. The dipole can to be vertical or horizontal in relation to the perfect ground plane.

The important input parameters are length of dipole and height up ground plane, both in wavelengths (or electric length). Other features are as the former simulator.

The 3D radiation pattern more complex is generated by horizontal dipole over perfect ground plane, as shown in Figure 2. The same displays details or particularities are not shown in the two 2D radiation patterns.


SIMULATOR OF 3D RADIATION PATTERNS OF DIPOLES. AUTHOR: MARINOEL JOAQUIM

INPUT DATA:
LENGTH_DIPOLE 0.5
HEIGHT_DIPOLE 0.75
ZOOM_2D 1.0

KIND OF DIPOLE: Free Space, Vertical-ground plane, [✓ Horizontal-ground plane]

SURFACE COLOR: Gray, Yellow, Orange, [✓ Blue]

SURFACE WITH: [✓ Strip (green color)], Circles (black color), Curves (red color)

HELP: To show help

```
Length_dipole =  0.50
Height_dipole = 0.75
DIRECTIVITY = 7.28
DIRECTIVITY_dBi = 8.62
```

Figure 2.Snapshot of field radiation patterns for horizontal dipole over ground plane.

As shown in Figure 2, the 2D radiation pattern (in black color) is in the plane that contains the dipole axis; the other 2D radiation pattern (in red color) is in the transversal plane at the dipole axis.

3.3. Simulator of 3D radiation patterns for Yagi-Uda antennas

This simulator has more resources and output data than the two former simulators. The Yagi-Uda antennas are made of N dipoles (or elements) as shown in Figure 3. The only active dipole (connected at transmission line) is the second. The first dipole is called reflector and third dipole until N dipole are called directors.

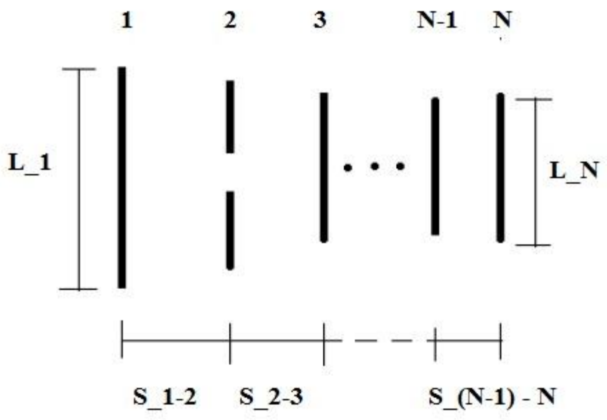
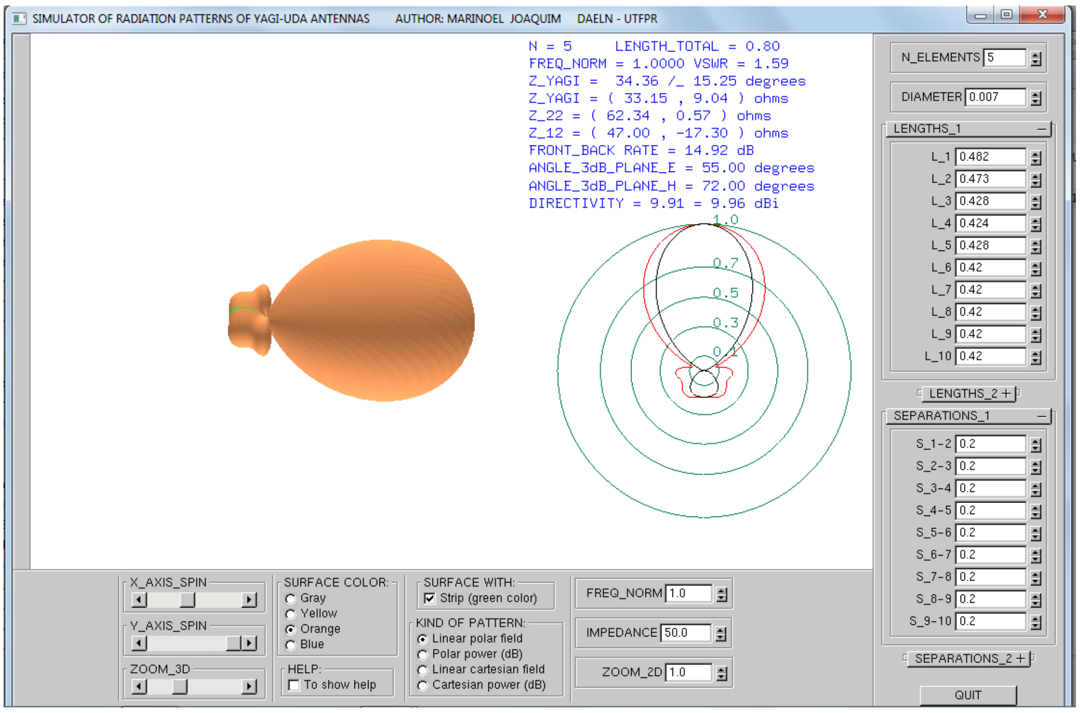


Figure 3.Schematic design of Yagi-Uda antenna.

The usual important input parameters are number of dipoles (until 17), diameter (only by simplicity), dipole lengths and separations between next dipoles; all these parameter dimensions are in wavelengths (or electric length), as shown in Figure 4.


SIMULATOR OF RADIATION PATTERNS OF YAGI-UDA ANTENNAS   AUTHOR: MARINOEL JOAQUIM   DAELN - UTFPR
1.0
0.7
0.5
0.3
0.1
LENGTHS_1
LENGTHS_2
SEPARATIONS_1
SEPARATIONS_2
QUIT
X_AXIS_SPIN
Y_AXIS_SPIN
ZOOM_3D

```
N = 5      LENGTH_TOTAL = 0.80
FREQ_NORM = 1.0000 VSWR = 1.59
Z_YAGI =  34.36 /_ 15.25 degrees
Z_YAGI = ( 33.15 , 9.04 ) ohms
Z_22 = ( 62.34 , 0.57 ) ohms
Z_12 = ( 47.00 , -17.30 ) ohms
FRONT_BACK RATE = 14.92 dB
ANGLE_3dB_PLANE_E = 55.00 degrees
ANGLE_3dB_PLANE_H = 72.00 degrees
DIRECTIVITY = 9.91 = 9.96 dBi
```

N_ELEMENTS 5
DIAMETER 0.007

L_1	0.482
L_2	0.473
L_3	0.428
L_4	0.424
L_5	0.428
L_6	0.42
L_7	0.42
L_8	0.42
L_9	0.42
L_10	0.42

S_1-2	0.2
S_2-3	0.2
S_3-4	0.2
S_4-5	0.2
S_5-6	0.2
S_6-7	0.2
S_7-8	0.2
S_8-9	0.2
S_9-10	0.2

SURFACE COLOR: Gray Yellow [✓ Orange] Blue

SURFACE WITH: [✓ Strip (green color)]

HELP: To show help

KIND OF PATTERN: [✓ Linear polar field] Polar power (dB) Linear cartesian field Cartesian power (dB)

FREQ_NORM 1.0
IMPEDANCE 50.0
ZOOM_2D 1.0

Figure 4.Snapshot of radiation patterns for a Yagi-Uda antenna.

This simulator creates 3D linear field radiation pattern and two 2D radiation patterns named electric (E) plane pattern (in black color) and magnetic (H) plane pattern (in red color). The electric plane pattern contains the dipoles' axis, while magnetic plane pattern is transversal at the electric plane. The simulator can generate four kinds of 2D radiation patterns, such polar field pattern in linear scale, polar power pattern in log scale, cartesian field pattern in linear scale and cartesian power pattern in log scale. It calculates a complex impedance of the Yagi-Uda antennas, which is joined to a paper for optimized design of Yagi-Uda antenna's dimensions [4] and techniques of impedance matching design [5]. It allows the design and practical test in laboratory of the Yagi-Uda antennas. Others features are like the former simulators. For more details it to see reference [6].

3.4. Simulators for 3D radiation patterns developed with the VTK

The learning objects described in the former sections are fast, portable, and easy utilization. Nevertheless, there are limitations such as they do not utilize the object-oriented design (OOD), though C++ is object-oriented programing (OOP) language. The OpenGL library is powerful, but also does not use OOD. The Glui library (GUI) is easy to be utilized with OpenGL, but is outdated and does not use OOD too.

A new development of learning objects have been made to overcome these limitations and produces other benefits. The Visualization Toolkit (VTK), that is built over OpenGL, is an open-source, object-oriented software for computer graphics, scientific visualization and image processing. The VTK is a large and complex library of C++ classes. Nevertheless, VTK is easy to use once one learn about is basic object-oriented design and implementation methodology. Another advantage of the VTK is that application developers may use interpreted languages as Tcl, Python and Java, beyond the C++ native language.

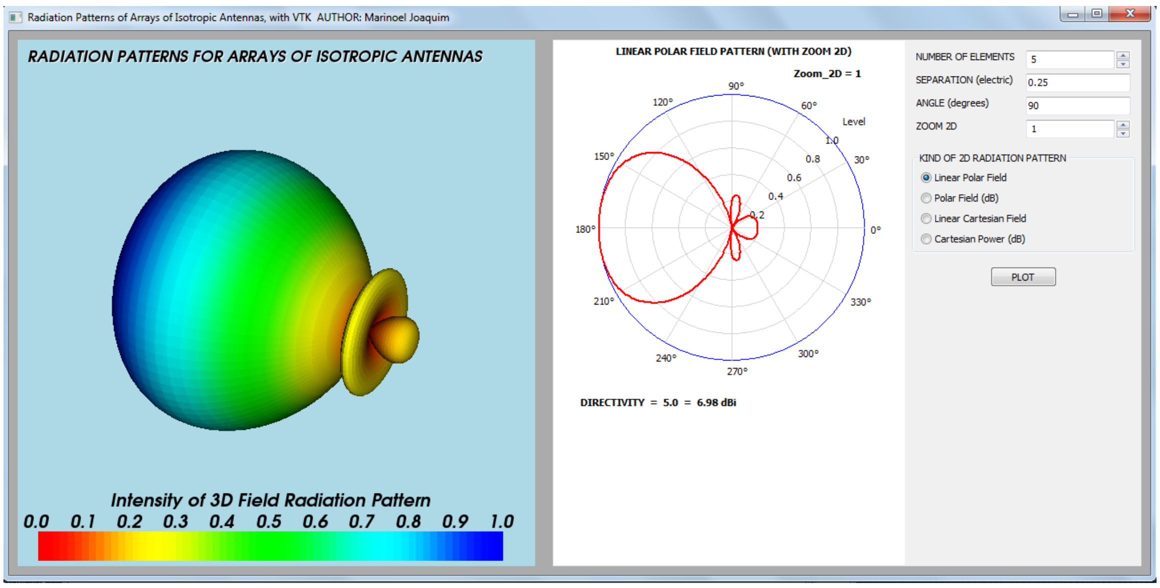
During the development stage it was decided to utilize the Python language. Python is a free use, high level, general purpose, multiplatform, interpreted language. Python has also very good mathematics libraries. Today Python is maintained by a large group of volunteers worldwide. Due to its features, Python is an excellent programing language to develop prototypes that use mathematic libraries.

For creating graphical user interfaces python programmers can choose among three main options: PyGTK, wxPython and PyQt. The wxPython is chosen because is of free use and has other adequate features for development purposes, as explained later on. With wxPython, developers can create applications on Windows, Mac and various Unix systems. The wxPython is a wrapper around wxWidgets, which is a nature cross platform C++ library.

It was developed two learning objects making use of Python language, wxPython and VTK libraries. The first learning object is a simulator for isotropic antenna arrays, like the one in Figure 1. It is noticeable the improvements that were introduced, as shown in Figures 5 and 6.

The VTK has many classes, as for instance, the class vtkScalarBar, which is used to create a color coded key that relates color values to relative intensity values of the 3D field radiation pattern, as shown in Figure 5. It permits more intuitive and significant visualization of 3D radiation pattern. Moreover, the OOD of VTK makes more simple, intuitive and robust the task of writing the necessary program code.

The GUI code of this simulator was written using wxPython classes and OOP Python. The structure of the 2D graphics uses the Graphics Device Interface (GDI) module of the wxPython. The quality of created 2D graphics is very good. The OOD Python allows efficient reuse of the developed 2D graphics class for four kinds of 2D radiation patterns. This class is used with few modifications in the next developed simulator, as shown in Figure 7 and 8. This learning object is the simulator of 3D radiation patterns of dipoles, like the Figure 2. The 3D field radiation pattern, for both simulators, can be rotated around X and Y axis, and receive zoom through the mouse use.


Radiation Patterns of Arrays of Isotropic Antennas, with VTK AUTHOR: Marinoel Joaquim

RADIATION PATTERNS FOR ARRAYS OF ISOTROPIC ANTENNAS

LINEAR POLAR FIELD PATTERN (WITH ZOOM 2D)

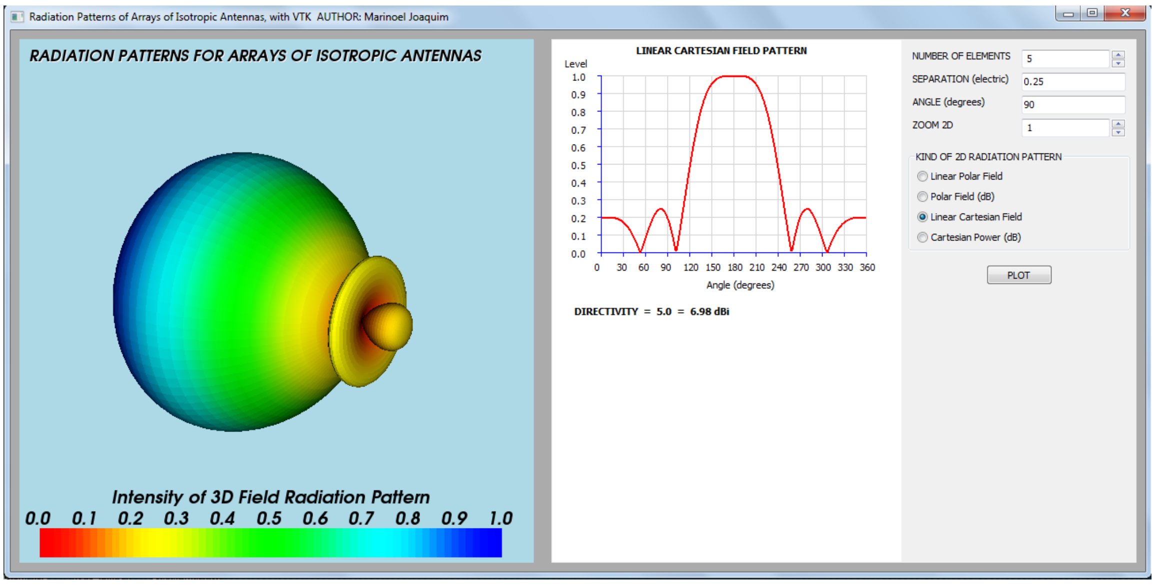
Zoom_2D = 1

NUMBER OF ELEMENTS: 5
SEPARATION (electric): 0.25
ANGLE (degrees): 90
ZOOM 2D: 1

KIND OF 2D RADIATION PATTERN
[✓ Linear Polar Field]
Polar Field (dB)
Linear Cartesian Field
Cartesian Power (dB)

DIRECTIVITY = 5.0 = 6.98 dBi

Figure 5.Snapshot of radiation patterns of isotropic antenna array.


Radiation Patterns of Arrays of Isotropic Antennas, with VTK AUTHOR: Marinoel Joaquim

RADIATION PATTERNS FOR ARRAYS OF ISOTROPIC ANTENNAS

LINEAR CARTESIAN FIELD PATTERN

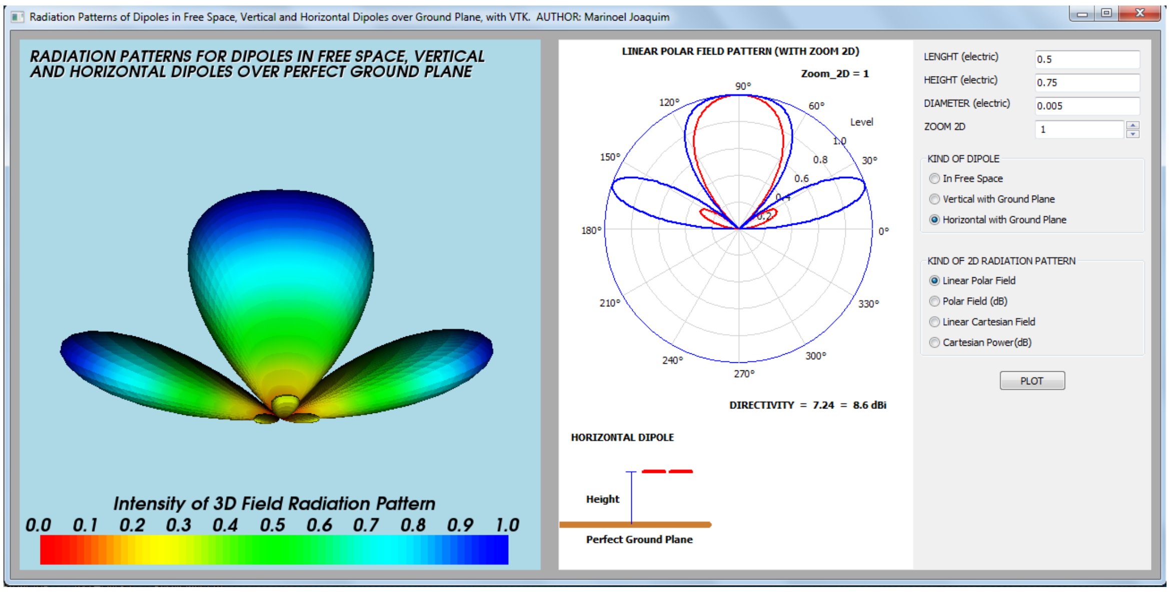
NUMBER OF ELEMENTS: 5
SEPARATION (electric): 0.25
ANGLE (degrees): 90
ZOOM 2D: 1

KIND OF 2D RADIATION PATTERN
Linear Polar Field
Polar Field (dB)
[✓ Linear Cartesian Field]
Cartesian Power (dB)

DIRECTIVITY = 5.0 = 6.98 dBi

Figure 6.Snapshot of radiation patterns of isotropic antenna array.

The disadvantage in using Python is that for a good resolution of 3D graphics, each new change in input data, needs a larger time (tens of seconds, in the utilized computer) for update the 3D graphics. In former simulators that use C++ language, the update the 3D graphics is instantaneous in the utilized computer. Nevertheless, this disadvantage is overcomed when Python code is transformed in C++ code. Thus, the developed C++ programs use wxWidgets and VTK libraries that are implemented also with C++ classes. It results in fast, portable and efficient OOD programs for development of learning objects.


Radiation Patterns of Dipoles in Free Space, Vertical and Horizontal Dipoles over Ground Plane, with VTK. AUTHOR: Marinoel Joaquim

RADIATION PATTERNS FOR DIPOLES IN FREE SPACE, VERTICAL AND HORIZONTAL DIPOLES OVER PERFECT GROUND PLANE

LINEAR POLAR FIELD PATTERN (WITH ZOOM 2D)

Zoom_2D = 1

DIRECTIVITY = 7.24 = 8.6 dBi

HORIZONTAL DIPOLE

LENGHT (electric) 0.5

HEIGHT (electric) 0.75

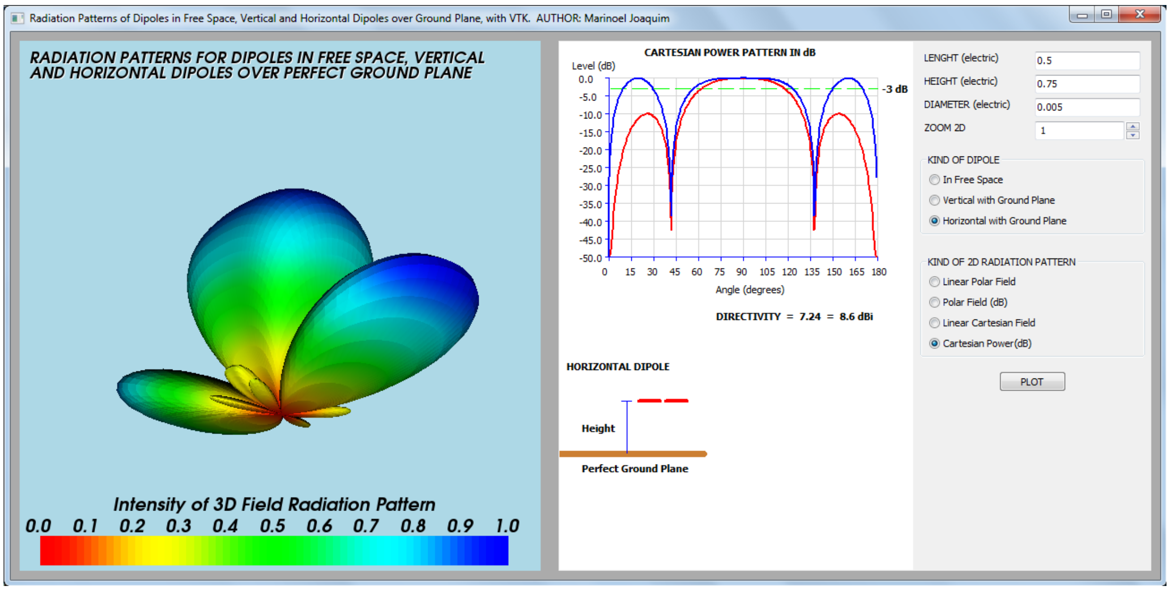
DIAMETER (electric) 0.005

ZOOM 2D 1

KIND OF DIPOLE
- In Free Space
- Vertical with Ground Plane
- [✓ Horizontal with Ground Plane]

KIND OF 2D RADIATION PATTERN
- [✓ Linear Polar Field]
- Polar Field (dB)
- Linear Cartesian Field
- Cartesian Power(dB)

Figure 7.Snapshot of radiation patterns of horizontal dipole over ground plane.


Radiation Patterns of Dipoles in Free Space, Vertical and Horizontal Dipoles over Ground Plane, with VTK. AUTHOR: Marinoel Joaquim

RADIATION PATTERNS FOR DIPOLES IN FREE SPACE, VERTICAL AND HORIZONTAL DIPOLES OVER PERFECT GROUND PLANE

CARTESIAN POWER PATTERN IN dB

DIRECTIVITY = 7.24 = 8.6 dBi

HORIZONTAL DIPOLE

LENGHT (electric) 0.5

HEIGHT (electric) 0.75

DIAMETER (electric) 0.005

ZOOM 2D 1

KIND OF DIPOLE
- In Free Space
- Vertical with Ground Plane
- [✓ Horizontal with Ground Plane]

KIND OF 2D RADIATION PATTERN
- Linear Polar Field
- Polar Field (dB)
- Linear Cartesian Field
- [✓ Cartesian Power(dB)]

Figure 8.Snapshot of radiation patterns of horizontal dipole over ground plane.

4. CONCLUSION

One of the objectives in using Python, wxPython and VTK was to allow fast development of scientific visualization prototypes. The second objective was to work with object-oriented design and to transform easily programs in Python (with wxPython and VTK) for programs in C++ (with wxWidgets and VTK). The last objective was to build large and complex applications of scientific visualization to help antenna design teaching and learning activities, which would be portable, fast and without cost for the students.

5. REFERENCES

[1] Jian-Jing Jin. *Theory and Computation of Electromagnetic Fields*. Hoboken, New Jersey: John Wiley & Sons, 2010.

[2] Cohen M., Manssour I. H. *OpenGL: Uma Abordagem Prática e Objetiva***.** São Paulo: Novatec, 2006.

[3] Schroeder W., Martin K., Lorensen B. *The Visualization Toolkit.* 3rd Edition. Clifton Park, NY: Kitware Inc., 2004.

[4] Viezbicke P. P. *NBS Technical Note 688: Yagi Antenna Design***,** 1976**.** Available in: < http://tf.nist.gov/timefreq/general/pdf/451.pdf > Access in: march, 25, 2011.

[5] Balanis C. A. *Teoria de Antenas: Análise e Síntese.* Vol. 1 e 2. 3rd Ed. Rio de Janeiro: LTC, 2009.

[6] Joaquim M., Scheer S. "Objetos de Aprendizagem para Visualização Gráfica Tridimensional de Diagramas de Radiação de Antenas". In: *XXXIX Congresso Brasileiro de Educação em Engenharia,* 3 a 6 de outubro de 2011, Blumenau, SC.