IMPLEMENTATION OF AN INTERFACE FOR ELECTROMAGNETIC ANALYSIS USING UTD

R. Fernández-Recio[1], **L.E. García-Castillo**[1] **and E. Escolano**[2]

[1] Departamento de Teoría de la Señal y Comunicaciones, Universidad de Alcalá
Escuela Politécnica, Campus Universitario, Ctra. Madrid-Barcelona Km. 33.600
28806 Alcalá de Henares (Madrid), Spain. Email: luise.garcia@uah.es

[2]Centro Internacional de Métodos Numéricos en Ingeniería (CIMNE)
Universidad Politécnica de Catalunya (Barcelona), Spain

Key words: Uniform Theory of Diffraction (UTD), TCL-TK, OpenGL, canvas, RamDebugger

Abstract. *A user interface oriented to the analysis of electromagnetic problems based on the Uniform Theory of Diffraction (UTD) has been implemented using GiD. The UTD is a high frequency technique based on ray-tracing where the electric and magnetic field in an observation point are obtained as the sum of the different rays that arrives to that point. The UTD kernel uses the NURBS surfaces generated by GiD. The TCL-TK and the TKWidget have been used to improve the functionality and the quality of the interface. The RamDebugger has been used to debug the TCL-TK codes. The interface allows to draw the different rays (direct, reflected, diffracted ..) using OpenGL. The user can get the different parameters of the rays (reflected and diffracted points, angles, coefficients ..) just selecting with the mouse on the desired ray or rays. The interface allows to plot, either the total field, or the different contributions from the different rays for the desired polarizations using a package based on the* widget canvas *from TCL-TK.*

1 INTRODUCTION

A user interface oriented to the analysis of electromagnetic problems based on the Uniform Theory of Diffraction (UTD) has been implemented using GiD. The UTD [1] predicts the electric and magnetic fields using a ray tracing scheme from the source point to the observation point (or observation direction). The electric field in the observation point is calculated as the sum of the contributions of the different rays that arrives to that point. Different kinds of rays can be found depending on the geometry of the problem under consideration such as direct, reflected, diffracted, and high order effects involving different objects such as double reflected, reflected-diffracted . . .. The UTD is an asymptotic technique or high frequency technique that only can be apply when the objects of the problem are electrically large and do not have small features. Because of the fact that it is an asymptotic technique the solution tends to the correct solution when the electric size of objects tends to infinity.

[1]This work has been supported by the Ministerio de Educacion y Ciencia, Spain, under projects TIC2001-1019 and TEC2004-06252

The Reseach Group, which the authors belong to, has developed a program based on the UTD written in FORTRAN. The UTD codes have been successfully hybridized with the Finite Element Method (FEM) [2]. Actually, the developments of the UTD interface are planned to be used on the interface for the hybrid FEM-UTD method. The decision of using GiD as the pre- and post- processor for the interface with the UTD kernel has been taken due to several reasons:

1. GiD supports NURBS surfaces, which are the base of the UTD geometrical modelling for arbitrary objects.

2. It is very easy to adapt to any numerical simulation code.

3. Portable in *Linux* and *Windows*.

4. GiD features such as TKWidget and its extension to TCL-TK allows the possibility of a hihg quality interface [3]-[4].

The interaction of the UTD kernel with GiD is in the pre and postprocessing stages.

2 PREPROCESS

Two aspects can be pointed out in the preprocess stage. On one hand, the general data related to the UTD (antenna position, frequency . . .) is presented using a facility of GiD called TKWidget that allows to communicate the information got in the graphical user interfaces (GUI) with the internal data of GiD and also attends some events related to the window and its data.

On the other hand, the user can draw the geometry in GiD or export from another format supported by GiD. Finally, the geometry is exported to IGES format, which is the geometrical input data to the UTD kernel. It is very important to highlight that the geometry is described using NURBS surfaces for arbitrary objects, which are directly used in the minimization/maximization process to find the critical points of the UTD kernel (e.g. reflected, diffracted points . . .). The geometry may also be given in terms of planar facets and some canonical objects. In this case, the critical points can be calculated analytically.

3 POSTPROCESS

A UTD interface requires two main kinds of postprocesses:

1. To visualize the ray tracing problem.

2. To plot the radiation pattern of the problem.

The ray visualization has been implemented using OpenGL. The user can draw any of the contributions independently (see Figure 1). Moreover, the user can select with the mouse one or several rays and see the characteristics of the selected rays. A lot of information related to the UTD kernel such as the type of ray, critical points, general data of the problem, UTD coefficients, angles of the ray with the surfaces and auxiliary system of vectors in the critical

points is shown in the information window. This information is very useful for debugging purposes with the UTD kernel or being used by experienced users or students to understand the problem under consideration.

The radiation pattern is visualized in terms of two-dimensional (2D) cuts in the three-dimensional radiation pattern describing the amplitude of the electric or magnetic far-field in given directions. At the moment, GiD does not have a powerful tool to represent this kind of information in the postprocess stage. Nevertheless, the flexibility of GiD and its extension to the TCL-TK language allows to find solutions to this problem. In this context, the authors have developed a powerful tool to draw 2D curves based on a widget of TCL-TK called *canvas* (see Figure 2). The starting point of the developmet of the facility has come from the free package *plotchart* at [5] where the features of the tool has been extended to the requirements of the interface. The actual package allows to draw for the desired contribution independently or at the same time the electric and magnetic fields for different polarizations in magnitude or in phase. It allows to edit the features of the curves such as color, type of line, width, type of mark . . . using an option table generated using the widget *TableList*. Moreover, a postcript file can be generated to print the information depicted in the *canvas*. Other features are being developed at the present time. Finally, it is important to note that the debugging of the interface has been carried out with the help of the tool RamDebugger [6].

4 CONCLUSIONS

A graphical interface for UTD has been developed using GiD. The two main aspects of a postprocess of a UTD program have been successfully implemented making use of OpenGL routines and the TCL-TK language.

REFERENCES

[1] R.G. Kouyoumjian and P.H. Pathak, *A Uniform Geometrical Theory of Diffraction for an Edge in a Perfectly Conducting Planes*, Proceedings of the IEEE, pp. 1448-1461, Nov. 1974

[2] R. Fernandez-Recio, I. Gomez-Revuelto and L.E. Garcia-Castillo, *A Hybrid FEM-UTD Method for the Analysis of Radiation Problems in Complex Environments*, in International Conference on Electromagnetics in Advanced Applications (ICEAA'05), Torino, (Italy), pp. 465-462, Sept. 2005

[3] J. Suit Perez, E. Escolano and M. Pasenau, *Advanced Problem Type Development: Data Dependencies and GUI Customization (TKWIDGET)*, 2^{nd} Conference on Advances and Applications of GiD, Barcelona, 2004

[4] A. Valls, *GiD-NASTRAN Interface: Guideline for Development of a High Quality Interface*, 2^{nd} Conference on Advances and Applications of GiD, Barcelona, 2004

[5] http://tcllib.sourceforge.net/

[6] http://www.gidhome.com/RamDebugger/RamDebugger_toc.html

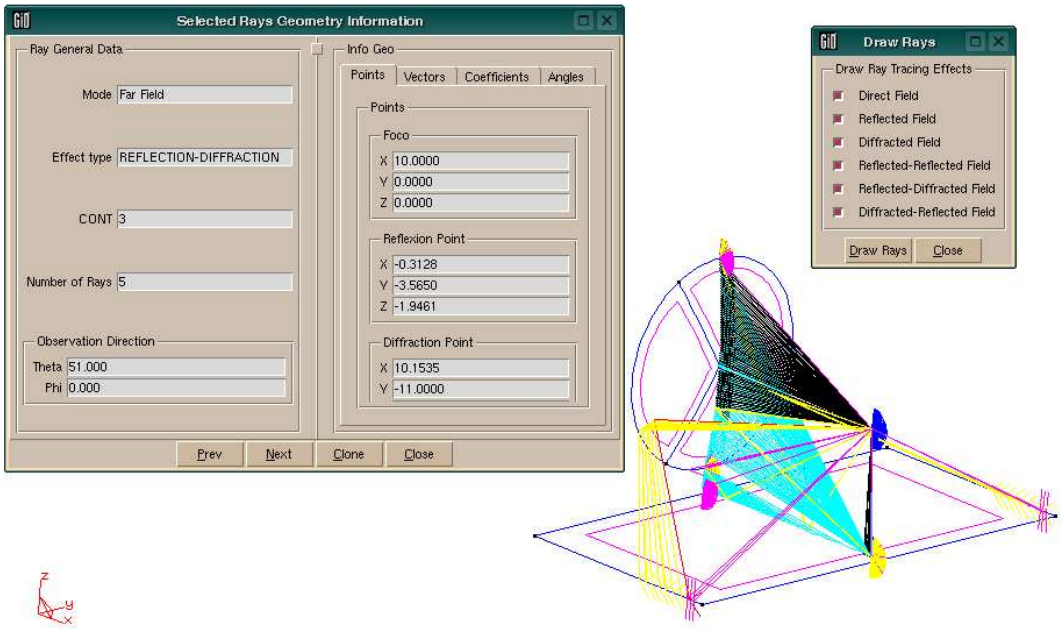

Figure 1: Postprocess related to the ray tracing visualization

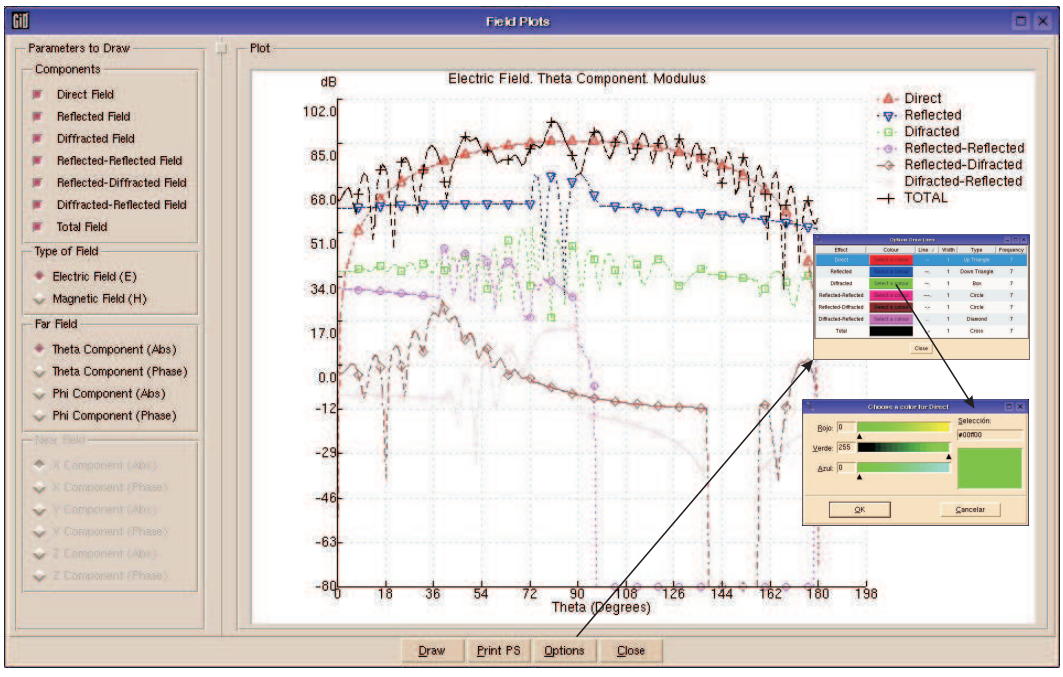

Figure 2: Postprocess related to the Representation of the radiation pattern using the widget *Canvas*