1 of 1

GRAPHICAL USER INTERFACE FOR AMOS AND POISSON

RECEIVED
AUG 09 1993
OSTI

T. L. Swatloski

*Lawrence Livermore National Laboratory, University of California
P.O. Box 5508, L-440, Livermore California 94550, USA*

This paper was prepared for submittal to the
*Proceedings of the Computational Accelerator Physics Conference
Pleasanton, California
February 22-26, 1993*

March 2, 1993

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

MASTER

de
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

GRAPHICAL USER INTERFACE FOR AMOS AND POISSON *

T.L. Swatloski
Lawrence Livermore National Laboratory
Livermore, CA 94550

ABSTRACT

A graphical user interface (GUI) exists for building model geometry for the time-domain field code, AMOS. This GUI has recently been modified to build models and display the results of the Poisson electrostatic solver maintained by the Los Alamos Accelerator Code Group called POISSON. Included in the GUI is a 2-D graphic editor allowing interactive construction of the model geometry. Polygons may be created by entering points with the mouse, with text input, or by reading coordinates from a file. Circular arcs have recently been added. Once polygons are entered, points may be inserted, moved, or deleted. Materials can be assigned to polygons, and are represented by different colors. The unit scale may be adjusted as well as the viewport. A rectangular mesh may be generated for AMOS or a triangular mesh for POISSON. Potentials from POISSON are represented with a contour plot and the designer is able to mouse click anywhere on the model to display the potential value at that location. This was developed under the X windowing system using the Motif look and feel.

INTRODUCTION

Part of the value of an experimental physics computer code is related to the number of runs the designer can perform, or the length of time it takes for a run to complete. Traditionally this has been addressed by using more and more powerful hardware, and better software algorithms. Graphical user interfaces can also increase the number of runs a designer can perform by minimizing the turn-around time of a run. The concept of turn-around time as used here is the total time it takes the designer to define the problem, to run the modeling code, and to see the results of the run. The purpose of this GUI, named Dragon, is to minimize the time it takes the designer to define a problem and in the case of POISSON, minimize the time to display the results to the designer, both of which happen to be where the designer's time is spent; generally a more valuable resource than computer time in today's world.

The interface was designed to be simple to use, promoting experimentation in building the model, and giving as much feedback as possible in the process. The GUI uses intuitive techniques such as pull-down menus, multiple choice wherever possible, and use of the mouse to select or

* Work performed under the auspices of the U.S. Department of Energy by LLNL under contract W-7405-ENG-48.

inquire about an object or location. There is a help menu that gives short descriptions of all fields in the GUI including parameters for the AMOS input file, and parameters to POISSON so the designer need not go to the user's manual for simple questions.

GRAPHIC EDITOR

The File menu (Fig. 1) allows the designer to open of a set of previously saved geometry information, open a file containing simple x-y coordinates of points to be added, save a set of geometry information, save an input file for AMOS, print the geometry model, or quit the Dragon. Filters and display of directories and existing files make retrieving and saving of files more convenient.

The Edit menu allows the designer to manipulate points and lines and a newly added feature is the ability to add an arc. Most edit functions are performed with the mouse.

The designer can change attributes of the model such as scale factor, materials, and grid size through the Attributes menu as shown in Figs. 2 and 3.

The View menu allows the designer to manipulate the display of the model, such as changing what is visible in the viewport, hiding or showing the material fill, and hiding or showing the grid.

AMOS

AMOS (Azimuthal Mode Simulator) is a physics simulation code solving Maxwell's time dependent curl equations for axisymmetric bodies (2-1/2 D). The azimuthal mode number, m , entered by the designer allows solution of the fields of the form $e^{im\phi}$. The code models lossy dielectrics, ferrites, and has up to five pole fit for dispersive media modeling. The code is used to identify and suppress modes leading to beam instabilities in accelerator cells, to study the rf in microwave structures and to model scattering problems.¹

The AMOS menu allows the designer to easily build additional information for the AMOS input file by using mouse clicks on the model to fill in fields that require model coordinates, and giving lists of values from which to choose. An example of this is shown in Fig 4.

POISSON

POISSON solves Maxwell's magnetostatic (electrostatic) equations for the vector (scalar) potential with nonlinear, isotropic iron (dielectric) and electric current (charge) distribution for two-dimensional Cartesian or three-dimensional cylindrical symmetry. It calculates the derivatives of the potential, namely, the fields and their gradients, calculates the stored energy, and performs harmonic (multipole) analysis of the potential. The code uses successive over-relaxation algorithm and an iterative scheme that steps successively throughout the mesh points.²

Fig. 5 shows the drawn model of an example in the POISSON reference manual for an H-shaped dipole magnet.² Fig. 6 shows the window for entering parameters to the mesh generation routines, AUTOMESH and LATTICE. Specific region information is in a separate window which can be scrolled to view each region's parameters. Fig. 7 shows the visual output from AUTOMESH and LATTICE, and the input window for POISSON. Fig. 8 shows the results of applying the contour plot and a window that allows the designer to change the number of intervals for the plot or the interval width.

FUTURE WORK

Dragon's graphic editor is somewhat limited and does not behave in accordance with currently accepted standards for graphic editors. A more modern graphic editor has been recently written and implemented in another application. (See Fig. 9.) It was written with the intention of being a generic 2D graphic editor, so that it may be used with other applications as well. This graphic editor will be used in any future 2D modeling development efforts in our area. Plans are to integrate this new graphic editor with Dragon in the near future, possibly by the end of this year (1993).

This new graphic editor allows more types of objects to be drawn such as rectangles, circles, free-form polygons, and Bezier splines. Objects are drawn only with the mouse, no text input. It also allows better and easier manipulation of the objects with such functions as cut, copy and paste, grouping, rotation, and dynamic update display of dimensions and angles as the object is being drawn or manipulated.

The new editor uses ANSI C, Motif, and X-window graphic primitives, making it fully portable to most UNIX-based systems, with the executable easily distributed because there are no licensing issues.

FIGURES

Fig. 1. Opening a Dragon file from the File Menu.

Fig. 2. Assigning materials to regions from the Attributes menu.

Fig. 3. Specifying the mesh and scale factor from the Attributes menu.

Fig. 4. Entering sample information in building the input file for AMOS.

Fig. 5. H-shaped dipole magnet example.

Fig. 6. Entering mesh information for AUTOMESH and LATTICE.

Fig. 7. Display of mesh and entering parameters for POISSON.

Fig. 8. Viewing default parameters for the contour plot.

Fig. 9. Future graphic editor for Dragon.

ACKNOWLEDGMENTS

We wish to thank C.C. Shang and J.F. DeFord (LLNL) for developing the AMOS modeling code, giving reason for the existence of the GUI. Also, thanks to R.K. Cooper (LANL) for his help in implementing the GUI for the POISSON group of codes. And, thanks to M.E. King and R.R. McLeod for developing the original interface for AMOS.

REFERENCES

1. C.C. Shang (Livermore, CA, 1992)
2. POISSON/SUPERFISH Reference Manual, LA-UR-87-126 (Los Alamos National Laboratory, NM, Jan. 1987) part A.1.1 & part B.2

**DATE
FILMED**

10/04/93

END