A Modular Approach to Numerical EM Modeling

Todd Hubing
University of Missouri-Rolla
Rolla, MO 65401

Introduction

Every numerical electromagnetic modeling code has some or all of the elements indicated in Figure 1. The *Geometry Description* element interprets the input data provided by the user and puts it in a form usable by the numerical solver. This may be a single subroutine for codes that read simple text files; or, if the code employs a sophisticated graphical user interface, the Geometry Description element may represent a significant fraction of the total code.

Some codes employ an *Automatic Mesh Generation* element, which reads the input geometry information and breaks it into simple pieces that are readily analyzed by the solver. Codes that do not employ automatic mesh generation rely on the user to define the geometry in terms of these simple pieces.

The *Input Validation* element checks to see that the input geometry makes sense and can be efficiently analyzed by the solver. In many numerical modeling codes, this element is minimal or missing entirely.

The *Numerical Solution* element numerically solves the applicable integral or differential equations subject to the boundary constraints provided. It may employ any of a number of different techniques in one, two, or three dimensions to solve for electric fields, magnetic fields, currents, or some other parameter of interest.

The *Output Validation* element evaluates the output to make sure that it is reasonable and consistent with the abilities of the solver. It may also provide an error estimate. Like Input Validation, this element is often neglected.

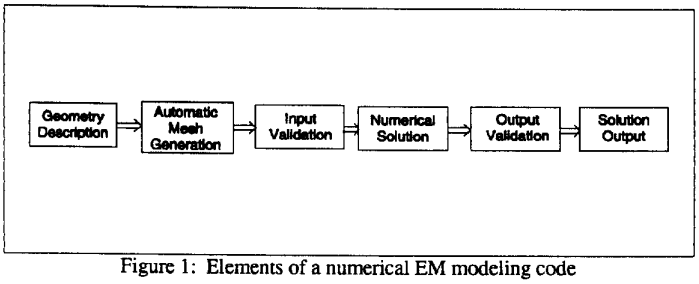

Figure 1: Elements of a numerical EM modeling code

0-7803-2009-3/94/$4.00 © 1994 IEEE.

The *Solution Output* element may simply format and list the numbers generated by the solver to an output device; or it may generate charts and graphs. Commercial modeling codes usually have very sophisticated output capabilities.

A Modular Approach

Since code developers that are experts in numerical EM modeling may not have the expertise to develop easy-to-use graphical interfaces, codes that have the most powerful Numerical Solution element do not always have the best user interface. Commercial EM software developers who have invested considerable resources developing sophisticated Geometry Description and Solution Output elements for their numerical codes cannot afford to scrap their user interface every time a new, more powerful modeling technique is introduced. As a result, it is often the case that the most powerful modeling codes have the weakest interfaces and the codes with the most usable interfaces are not the most efficient codes available.

Suppose it were possible to develop each of the elements in Figure 1 independently. A researcher developing a new numerical solution technique would not have to be concerned with the details of the user interface. A software company with a large investment in a particular graphical interface could upgrade their numerical solver without substantially changing the interface. Each element of the modeling code could be upgraded or replaced without affecting the other elements.

This modular approach to EM software development would allow developers to focus their efforts on one particular element of the code. Each element could be developed by the most qualified people. Code users could select the elements best-suited to their own needs in order to assemble customized EM modeling codes.

A modular approach relies on standard format input and output files for each element (or module) of the code. Figure 2 illustrates how data would be passed in a modular code. Note that the Input Validation and Output Validation modules have been removed from the main stream of flow. These modules do not modify the data passed to them, they merely evaluate the data and provide output directly to the user. The Input Validation module reads the same input as the Numerical Solution module and these modules may be executed in parallel. The Output Validation module reads the same input as the Solution Output module and these modules may also be executed in parallel. Although they could be developed independently, the Input and Output Validation modules should usually be designed to work only with codes that employ specific numerical modeling techniques.

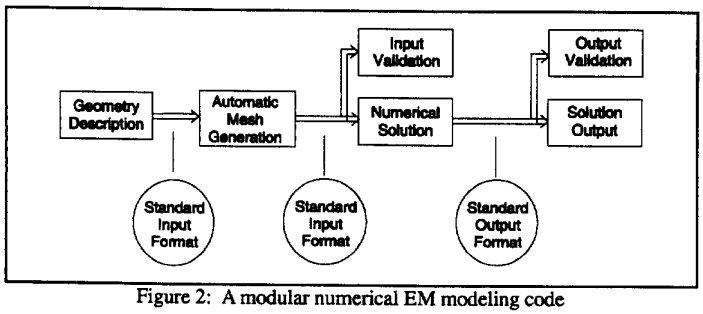

Figure 2: A modular numerical EM modeling code

The key to making a modular approach work is defining an input format and an output format that are flexible enough to work with a variety of numerical EM modeling techniques. These file formats should be concise and not overly complex. Ideally, they would be text files so that some or all of the elements could be platform independent.

A Standard Input Format

Given the variety of numerical modeling techniques available and the wide range of input information they require, it may seem that an input file with all the necessary information required for any technique would be very long and complex. However, when we set out to develop an input file format to work with our own FDTD, finite element, and hybrid FEM/MOM codes, the format we used resulted in a relatively simple and readable text file for most geometries [1]. The format employs a selection of keywords followed by parameters related to each keyword. For example, a line in an input file that reads,

```
dielectric 0 0 0 1 1 1 4 .0005 1 d
```

describes a lossy dielectric cube with unit size located at the origin. The relative permittivity of the dielectric is 4, the conductivity is .0005 mhos/m, the relative permeability is 1, and the mesh density is twice the default. A basic set of keywords has been defined to describe geometries that are readily analyzed by electromagnetic modeling codes. Additional keywords control program execution, define sources, and specify the output format.

Of course, if the user is employing a numerical technique that cannot analyze lossy dielectrics, then a lossy dielectric should not appear in the input file. The Input Validation module should recognize this problem if it were to occur and halt the program.

Since codes may or may not use automatic mesh generation, the output of the Geometry Description module must have the same format as the output of the Automatic Mesh Generation module. The mesh generator simply adds keywords and/or keyword parameters to the input file that describe how things are meshed.

Standard Output Formats

Since different numerical techniques calculate different parameters, it is necessary for the standard output format to be very flexible. By defining a finite set of possible output formats, a variety of techniques can be accommodated. This means that the Solution Output module must be able to recognize a number of different output formats. The following outputs are user-selectable by way of a keyword in the input file:

1. Electric field at a user-defined set of data points at a user-defined frequency
2. Magnetic field at a user-defined set of data points at a user-defined frequency
3. Electric field at a user-defined data point over a user-defined time interval
4. Magnetic field at a user-defined data point over a user-defined time interval
5. Radiated power
6. Input impedance at each source
7. Surface currents on wires and PEC patches
8. Custom output
9. Default output

All of the above outputs are text files. The first seven output files have a well-defined format. The Custom Output option is for codes that calculate a non-standard parameter such as radiated power through an aperture. The Default Output option allows user to let the output choice default to the solver's most natural output format. This prevents the user from being forced to select a specific output format and allows the input file to remain independent of the type of solver used.

Most codes will not be able to calculate all of the above outputs. It is up to the user to choose a reasonable output format or to select the default output. The Input Validation module should ensure that this choice is made properly.

Reference

[1] T. Hubing, H. Lim, and J. Drewniak, "A Geometry Description Language for 3D Electromagnetic Analysis Codes," Proceedings of the Tenth Annual Review of Progress in Applied Computational Electromagnetics, Monterey, CA, March 1994.