CONF-9605102--1

SAND96-0646C

EIGER: A New Generation of Computational Electromagnetics Tools

D. R. Wilton

Department of Electrical Engineering, University of Houston, Houston, TX, USA

W. A. Johnson and R. E. Jorgenson

Sandia National Laboratories, Albuquerque, NM, USA

R. M. Sharpe and J. B. Grant

Lawrence Livermore National Laboratory, Livermore, CA, USA

Abstract

The EIGER project (Electromagnetic Interactions GenERalized) endeavors to bring the next generation of spectral domain electromagnetic analysis tools to maturity and to cast them in a general form which is amenable to a variety of applications. The tools are written in Fortran 90 and with an object oriented philosophy to yield a package that is easily ported to a variety of platforms, simply maintained, and above all efficiently modified to address wide ranging applications. The modular development style and the choice of Fortran 90 is also driven by the desire to run efficiently on existing high performance computer platforms and to remain flexible for new architectures that are anticipated. The electromagnetic tool box consists of extremely accurate physics models for 2D and 3D electromagnetic scattering, radiation, and penetration problems. The models include surface and volume formulations for conductors and complex materials. In addition, realistic excitations and symmetries are incorporated, as well as, complex environments through the use of Green's functions.

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

1 Introduction

Many of today's engineers or scientists instinctively turn to computer modeling as their principal tool for electromagnetic device analysis and design. Indeed, computational electromagnetics has evolved into a proven technology that is able to provide detailed information for many previously intractable problems. Its success, however, has not only spawned a large user community, but has also fueled demands for increasing its capabilities for modeling even more complex systems. In recent years the pressure of these demands has increased such as to produce a software impasse in computational electromagnetics.

Over the years, many industrial, governmental, and educational research laboratories have developed or acquired large libraries of special purpose programs for electromagnetic modeling, often supplementing them with expensive commercial software. Though recent commercial offerings come equipped with powerful, user-friendly interfaces, their computational capabilities frequently lag behind current engineering practice; more importantly, these programs cannot be updated or modified by the user—nor easily by the vendor. Consequently, much engineering analysis and design is still accomplished by programs developed for special purpose applications.

Logically, the most desirable approach to extending capabilities of current software is to modify currently available programs to meet new requirements. In practice, however, one finds that available source codes not only are extremely complicated, but have inherent limitations that make extending them difficult. Some of this difficulty is connected with limitations inherent in current versions of the FORTRAN programming language, the overwhelming language-of-choice in engineering and scientific circles. One also finds that key algorithms are buried deeply within the software and during modifications become poorly organized and documented. Thus as new engineering requirements arise, it becomes increasingly difficult to further modify the associated modeling software on a reasonable schedule. For all these reasons, practicing engineers, graduate students, and researchers alike often make a conscious decision to ignore existing software and turn to developing their own codes. The process is, of course, self-perpetuating and merely contributes to the proliferation of special purpose codes.

This paper describes an attempt to at least partially address this current software impasse. It describes the previous efforts and experiences of a development group that motivated the move to use new software engineering techniques. The goals of this current effort are to develop portable, modular, and reusable general-purpose electromagnetics software which is appropriate to current and next-generation computing platforms for resonance regime physics.

2 Foundation of EIGER

The fundamental building blocks for this effort began with the development of a general purpose code for modeling arbitrarily shaped conducting bodies [1]. That code, which used triangular surface elements to model geometry and employed a moment method solution of an integral equation, demonstrated the basic numerical procedures that would be applied and generalized by many subsequent tools. A follow on effort [2], applied a more structured approach to the software and added useful features to yield a production code, still applicable only to conducting surfaces.

Following these initial codes, a variety of research efforts were applied to extend the basic techniques to handle other geometrical descriptions and boundary conditions. One effort extended the tools to treat conducting wires and junctions of the wires with surfaces [3]. This successful effort began to demonstrate the difficulty of extending a code to treat multiple types of elements even for the same requisite physics. Each of the interactions between surface, wire and junction had to be explicitly treated to yield an interaction matrix that was 3×3 blocks. Each of the nine submatrices required independent logic and bookkeeping software, in addition to the unique software required for treating the different geometrical elements and local basis (expansion) functions used. An additional effort to extend this junction tool to treat coated conductors and thin material shells didn't require any additional software for the geometrical elements, however, the interaction matrix which represents all possible coupling in such a structure was now 5×5 blocks. Each of the 25 submatrices now required independent logic and bookkeeping software. Although it is relatively straight forward to identify the numerical techniques and fundamental physics that would be required to extend this tool to treat additional elements or boundary conditions, it becomes nearly intractable to generate the required software in a timely or cost effective manner using standard programming practices.

A subsequent development effort entitled EMPACK [4] organized the key components of these codes into a modular software "tool box". This improved organization of the software contained subroutines for treating a variety of 2 and 3 dimensional elements and for computing the potentials due to source distributions on those elements. These routines were written with standardized interfaces which now allowed an end user to simply write just enough software to dimension arrays and call the appropriate subroutines to generate a specialized code for their application.

The EMPACK development was an important step in the evolution to the present development effort being described here. Several of the key advances made during this process will be briefly described. EMPACK was strongly influenced by the extensive efforts of the finite element community [5]. Particularly, in other disciplines, such as civil and mechanical engineering, the computing algorithms and data structures needed for handling a wide variety of geometries, problem types, and formulations have been systematically organized into a collection of modular subroutines constituting a general-purpose computer code. Though accomplishing the same task is more difficult in electromagnetics, particularly when dealing with integral equations, recent theoretical developments in computational electromagnetics now make this approach not only practical, but desirable. One such development was the ability to cast the various potential integrals required for an integral equation solution into a common form containing scalar integrals only [4]. Another effort developed a formulation that would allow various dyadic Green's functions to be written in a potential form for easy assimilation into existing formulations [6]. As these software interfaces for EMPACK were standardized and common components of apparently dissimilar methods were identified and generalized, it was decided that an object oriented design would yield great benefits. Specifically, by carefully abstracting the basic components of the electromagnetics computational model, general analysis modules could be developed. For example, basic procedures can be developed which encapsulate such aspects as geometry modeling and mathematical operators.

The resulting project–a code called EIGER for Electromagnetic Interactions GEneRalized–consists of a toolbox of modular software for accurately modeling two- and three-dimensional electromagnetic scattering, radiation, penetration, and guided wave problems. A key feature of the project is the use of Fortran 90 as the language of choice for code development. Its many nice features outweigh its object oriented limitations which can be overcome with programming practices discussed later. Some of the key features include: multidimensional array syntax, user-definable operator names, scientific orientation (e.g., complex numbers, good optimization) , and familiar input/output syntax. Furthermore, since Fortran 90 is a subset of High Performance FORTRAN, it provides a direct path to parallel programming. In the following sections, some of the key features of EIGER are discussed.

3 Features of EIGER

In this section we give an overview of the organization of EIGER. We briefly consider some of the most important code modules, some of the features currently implemented in each module, and possible extensions to the module. We also discuss how the module would support the overall capabilities of the code.

3.1 Elements and Basis Functions

All geometries of interest are modeled by subdividing them into subsections called elements. Key software already exists to support many of the elements commonly used and is being incorporated into EIGER as needed. Line-segment elements, for example, are often used to model wires, the cross sections of two-dimensional cylinders, and the generating curves of bodies of revolution (BOR). Planar polygonal elements—triangles and rectangles—are used to model two-dimensional inhomogeneous regions, BOR volumes, and three-dimensional surfaces. Volumetric elements—tetrahedrons, pentahedrons, and hexahedrons—are used to model three-dimensional volumes. With these simple element building blocks, one is able to model the principal geometrical features of virtually any electromagnetics problem of interest.

Recent extensions to the code permit curved elements to be modeled. In certain applications, such as the computation of radar cross section (RCS) or of magnetic fields produced by magnetic resonance imaging devices, very high accuracy is required. The ability to model geometrical curvature significantly extends the accuracy for these problems. Modeling the curvature has also been found to significantly increase the convergence rate in applications where the geometry's tangent vector changes more rapidly than the field quantities of interest.

The actual inheritance tree of these 'element' objects is quite rich. The top of the tree is naturally the data structure common to all of the element types. Through inheritance, the object 'line segment' appends data structures specific to its dimensionality. Likewise for the class 'surface'. However, the lack of inheritance in F90 forces us to use more of a C-style union. Hence, the inheritance tree becomes one dimensional with overlapping components. While C allows a true overlaying of structures, Fortran 90 does not. Thus, we must carefully select unique component names and identify common features wherever possible. For example, the nodes of a surface element outnumber those of wires, so wires use the same array as surfaces. To avoid unwieldy structures we dynamically allocate components we deem large enough, as well as those of indeterminate length. In short, the one dimensional (collapsed) tree allows us to share as much functionality as possible.

In addition to collapsing the tree, we add a flag for each branch that is collapsed to mimic polymorphism to some extent. Any generic operation on the generalized object 'element' can then quickly determine which specialized operation to execute. Thus, we gain the highly desired polymorphism. However, this inserts an extra layer into the subroutine calling chain which is deep inside the code and executed for each element. This dispersed determination of element type introduces additional effort in maintenance relative to other languages such as C++. The typical overhead of this extra layer is small relative to the computations it initiates.

Once the elements are defined, basis or expansion functions are needed to model fields within or on an element. Expansion functions that are employed in EIGER are interpolatory polynomials, and special basis functions are used for modeling vector quantities. For example, the vector basis functions used have continuous tangential components across element boundaries if they are operated on by curl operators, and have continuous normal components if they are operated on by divergence operators. The modeling of field singularities at edges and vertices also requires specially constructed bases. A junction basis function, which explicitly incorporates a $1/R$ local variation, is used at the junction of conducting wires and surfaces. In addition, a bases currently under construction incorporates the correct root singularity at known edge angles. These specialized basis functions significantly improve accuracy while reducing the number of unknowns that need to be used.

3.2 Operators

The formulation of problems in computational electromagnetics generally involves the use of integral, differential, or integro-differential operators supplemented by boundary conditions. These operators operate on basis functions which are defined on elements. The subsequent projection of the resulting operator equation onto a set of testing functions results in replacement of the original operator equation by its matrix approximant; the system matrix may be said to represent a discrete approximation to the operator.

The initial development effort in EIGER focused on the solution of integral equations and implemented operator objects for both electrodynamic and static formulations. In the dynamic case, operators relating equivalent electric or magnetic current sources to the electric or magnetic fields which they produce were employed. These operators are conveniently computed from various potentials and their gradients. Subprograms for these were developed during the EMPACK effort and include all the linear order elements and some quadratic order elements with constant and linear order bases listed in the previous section. Current efforts are underway to implement these potential calculations for quadratic order bases in conjunction with the singular bases listed previously. The static formulations employ analogous operators that relate charge and potential sources to the potentials and fields that they produce. The success of the overall code design was dramatically demonstrated by the ease of incorporating these operator objects for statics into the code.

Differential equation formulations typically require either a generalized Laplacian operator or a vector Helmholtz operator which are used extensively in finite element modeling. Prototype code has been developed and validated for the static (Laplacian) operator and will require very little effort to incorporate directly into EIGER.

The specific form an electromagnetic operator takes also depends on the Green's functions or symmetries used and the boundary conditions which are enforced. These aspects of operators are discussed below.

3.2.1 Green's Functions and Symmetries

One of the advantages of integral equation formulations is that a wide variety of Green's functions can be easily incorporated into the formulation. Since Green's functions automatically satisfy all but a few of the boundary conditions in a problem, their use can substantially reduce the number of unknowns required, yielding a very efficient solution procedure. They can usually be employed whenever some of the boundaries in a problem coincide with constant coordinate surfaces of standard rectangular, cylindrical, or spherical coordinate systems. Although the computation of most Green's functions require the summation or integration of a slowly converging spectral representation, the computations can be sped up by acceleration techniques.

Several recent research efforts have cast a variety of Green's functions into a form that enables them to be easily incorporated into a numerical solution [7, 8]. Currently both free space and multi-layered Green's function objects are available in EIGER. In addition, the existing spectral Green's functions for cavity geometries and periodic structures can easily be incorporated into the inheritance tree.

Use of geometrical symmetry in a problem to reduce the complexity of a calculation is closely related to the use of a Green's function. As with Green's functions, when symmetry can be used it can often substantially reduce the computer resources required to solve the problem. It should also be noted that even if the excitation of a symmetric structure is not symmetric, a decomposition of the problem into symmetrical components can still be used. Operators for automatically handling reflection about x, y, or z-axes, as well as N-fold rotational symmetry about an axis are currently being incorporated into the code.

3.2.2 Boundary Conditions

Integral equations themselves are either boundary condition statements or statements of relationships between fields and their sources. In any case, integral equations always involve the computation of some linear combination of the fields produced by equivalent sources. A logical approach in a general-purpose code, and the approach used in EIGER, is to develop boundary condition objects which have as their only task the assembly of specified linear combinations of fields. The most elemental of these combinations involves the combining of fields produced by a set of equivalent basis currents defined on a single source element, observed at and with the boundary conditions appropriate to a single observation element. Using this basic approach, as new boundary conditions appear—say due to the appearance of new exotic materials or improved boundary approximations—they can be added to the module by simply specifying the appropriate coefficients of the linear combination required.

Some of the boundary condition types which are included in EIGER, are identified below. In addition, some sample formulation types or applications in which that type of boundary condition occurs are given. One boundary condition requires the electric or magnetic fields or a linear combination to vanish on a surface. This condition is used in the Electric Field Integral Equations (EFIE), Magnetic Field Integral Equations (MFIE), and Combined field (CFIE) or source (CSIE) integral equations. Another boundary condition sets the electric and magnetic fields proportional to one another or to the equivalent sources. This condition is useful for surface impedance boundary conditions and for lumped and distributed loading., Finally, by setting the electric or magnetic fields to be continuous at an interface one can treat dielectric boundaries and apertures.

3.3 Excitation

A general excitation module has been developed for EIGER. Presently these excitations include specifying incident modes, or the direction and polarization of incident plane waves for scattering and penetration problems. In addition, the excitation of antennas involves the specification of terminal voltages and aperture field distributions.

Other classes of excitations are planned for incorporation into the code. In the static and quasi-static frequency regimes, excitations are usually defined by specifying potentials or excitation fields due to assumed distributions of charge or current. In many guided wave or cavity applications, on the other hand, one seeks source free solutions; in these cases frequencies for which the system determinant vanishes are sought and no sources are involved at all.

3.4 Solution

The optimum approach for solving the system matrix arising from a given problem usually depends on both the matrix type generated by the formulation and the geometrical connectivity of the problem. These factors combine to yield matrices with various structures, symmetries and densities. EIGER presently employs a general solution scheme for either complex or real dense matrices.

Ideally, the solution module would employ the matrix solution method that is most efficient for a given problem. It appears that significant research remains before the selection of the optimum solution path can be automated, however. This is at least partially due to the fact that the optimal solution procedure depends on the connectivity of the problem geometry, and few solvers are currently equipped to take this feature into account. It appears that the best strategy for the moment is to develop a number of alternative solution strategies with emphasis on handling block matrices of various types. As means for automatically controlling the solution process are developed, these algorithms will form the basic tools needed by a more advanced subprogram which would control the calling of these routines at each stage of the solution process.

Subprograms are needed to realize at least the following algorithms which frequently occur in computational electromagnetics, and indeed, which form the core of computational linear algebra: eigenvalue determination, determinant evaluation, direct solution algorithms, and iterative solution algorithms. Fortunately, many of these basic algorithms are already available in public domain code.

3.5 Output Quantities

Post-processing capabilities have been implemented as objects operating on solutions. They generate data summaries, figures-of-merit for a problem, and other functionals desired by the user, with the results being output to appropriate devices. There is a wide variety of information of interest to users of electromagnetics codes. Provisions for computing the following and the other figures-of-merit used in particular engineering applications are included in EIGER: equivalent surface currents along a cut or in a plane, near fields at an interface or aperture, far field radiation patterns, radar cross sections, antenna gain, and input and mutual impedances. In addition, extensions that are presently envisioned include total power radiated or transmitted through an aperture, frequency dependence of a certain quantity, and Fourier transform of frequency domain data to synthesize time domain response.

4 Summary

The complexity of electromagnetic modeling codes has, historically, limited the development of general-purpose codes. Though a few do exist, they cannot be easily extended or generalized to solve many of the problems of current interest. We feel that a need now exists for a truly general-purpose electromagnetics code and that EIGER is a step in the right direction. Recent developments in electromagnetic theory, together with EIGER's object oriented abstractions and systematic notation, overcome many of the technical problems that limited previous developments.

References

[1] S. M. Rao, D. R. Wilton, and A. W. Glisson,"Electromagnetic Scattering by Surfaces of Arbitrary Shape," *IEEE Transactions on Antennas and Propagation,* Vol. AP-30, No. 3, pp. 409–418, May 1982.

[2] W. A. Johnson, D. R. Wilton, and R. M. Sharpe, "Modeling Scattering from and Radiation by Arbitrarily Shaped Objects with the Electric Field Integral Equation Triangular Surface Patch Code," *Electromagnetics,* Vol. 10, pp. 41–63, 1990.

[3] S.-U. Hwu, "Electromagnetic Modeling of Conducting and Dielectric Coated Wire, Surface, and Junction Configurations" Ph.D. dissertation, Univ. Houston, Houston, TX, May 1990.

[4] S. V. Yesantharao, "EMPACK—A Software Toolbox of Potential Integrals for Computational Electromagnetics," master's thesis, Univ. Houston, Houston, TX, August 1989.

[5] J. E. Akin, *Application and Implementation of Finite Element Methods.* Orlando: Academic Press, 1984.

[6] K. A. Michalski, "The mixed-potential Electric Field Integral Equation for objects in Layered Media," *Arch. Elektron. Ubertragungstech.,* Vol. 39, pp. 317–322, Sept./Oct. 1985.

[7] N. W. Montgomery, "Periodic Structures in Stratified Media—A Mixed Potential Formulation," Ph.D. dissertation, University of Houston, Houston, TX, 1991.

[8] R. M. Sharpe, "Moment Method Analysis of Complex Structures Containing Cavity-Backed Apertures," Ph.D. dissertation, University of Houston, Houston, TX, 1992.