Finite Element Method Modeling for Computational Electromagnetics

Development of a Perfectly Matched Layer for Domain Termination

Second Semester Report
Spring Semester 2015
-Full report-

By
Aaron Smull

Team Members:
Ana Manic
Sanja Manic

Prepared to partially fulfill the requirements for ECE402

Department of Electrical and Computer Engineering
Colorado State University
Fort Collins, CO 80523

Project advisor: Dr. Branislav Notaros
Approved By: Branislav Notaros
ABSTRACT

The field of computational electromagnetics requires constant innovation and research into new numerical methods. In order for computer aided design software to be even considered useful in an extremely competitive market, it must be able to solve intense numerical problems both faster and with greater efficiency in memory usage than other contenders in the area. In order to achieve these two goals, new methods must continually be researched, applied and refined. Only through optimal implementation of new methods can a piece of software hope to achieve the level of efficiency present in industry standard electromagnetics.

In computational electromagnetics, the finite element method in particular, is an extremely effective method for modeling electromagnetics problems with complicated geometries and media. Since the finite element method involves physical modeling of computational domains, it requires a finite domain for simulation. For computational problems which require infinite domain simulation (e.g., scattering and antenna radiation problems), we must use artificial methods to effectively terminate the computational domain without introducing large amounts of numerical error. The perfectly matched layer is one of these methods. The purpose of this project is to take already existing code for finite element method simulation, to optimize and simplify the code, to prepare the code for implementation of an effective perfectly matched layer, and finally to implement and test perfectly matched layer examples.

In order to effectively tackle the problem at hand, the team has done extensive research into theory behind artificial absorptive boundaries, including the square and conformal perfectly matched layers, the first and second-order absorbing conditions as well as other less common forms of boundary termination. We approached this problem by doing literature searches among common journals (IEEE Xplore) as well as utilizing scholarly search engines (Google Scholar). In order to simplify and optimize the code, we stripped it down to a bare-bones structure and reevaluated the necessity of code sections, choosing to remove or optimize many pieces of code.

The PML was eventually implemented with great success, having generated near-field results in scattering problems that agree almost exactly with those generated by WIPL-D, an industry software. The field pattern in the results cannot be differentiated by eye alone.
Table of Contents

Title ............................................................................................................................................................... 1
Abstract ............................................................................................................................................................ 2
Table of Contents ........................................................................................................................................... 3
List of Figures ................................................................................................................................................ 4
Introduction .................................................................................................................................................. 6
  a.) Computational Electromagnetics ........................................................................................................ 6
 b.) The Finite Element Method ................................................................................................................ 6
 c.) Project Description ............................................................................................................................. 7
d.) Project Specifications ......................................................................................................................... 7
e.) Project Impact ...................................................................................................................................... 8
I. Finite Element Method .............................................................................................................................. 8
  a.) Theory .................................................................................................................................................. 8
 b.) Choice of Basis Functions ................................................................................................................... 9
 c.) Geometrical Elements ........................................................................................................................ 10
d.) Anisotropic and Inhomogeneous Media ............................................................................................ 11
e.) Comparison to Other Numerical Methods ......................................................................................... 12
II. Perfectly Matched Layer ........................................................................................................................ 12
  a.) Purpose ............................................................................................................................................... 12
 b.) Theory ................................................................................................................................................ 13
c.) Wave Equation Reformulation .......................................................................................................... 15
d.) Extension to Conformal PML ............................................................................................................ 17
III. Applications .......................................................................................................................................... 18
  a.) Electromagnetic Compliance ............................................................................................................. 18
 b.) Radar Analysis ................................................................................................................................... 19
c.) Antenna Design .................................................................................................................................. 20
d.) Signal Integrity .................................................................................................................................. 20
IV. Testing and Validation .......................................................................................................................... 21
V. Successful Results .................................................................................................................................. 22
VI. Update to Input File .............................................................................................................................. 24
VII. Output Plotting .................................................................................................................................... 25
VIII. Pre-calculation of Excitation Vector .................................................................................................. 25
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Basis Functions for FEM</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Generalized Hexahedral Element</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Perfectly Matched Layer</td>
<td>15</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Conformal PML Mesh</td>
<td>18</td>
</tr>
<tr>
<td>Figure 5a</td>
<td>Model of Human Brain</td>
<td>19</td>
</tr>
<tr>
<td>Figure 5b</td>
<td>Normalized Electric Field inside Brain</td>
<td>19</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Stealth Bomber Model</td>
<td>20</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Horn Antenna Model</td>
<td>20</td>
</tr>
<tr>
<td>Figure 8</td>
<td>CAD Microstrip Model</td>
<td>21</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Sphere Model</td>
<td>22</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Analytic Solution for Scattering from Sphere</td>
<td>22</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Result Comparison for Dielectric Cube</td>
<td>23</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Result Comparison for Dielectric Sphere</td>
<td>23</td>
</tr>
</tbody>
</table>
Introduction

a.) Computational Electromagnetics

Computational electromagnetics (CEM) is a field that involves finding numerical solutions to Maxwell’s equations in the context of any system where electric and magnetic fields have a large bearing on the system’s behavior. Its applications are numerous and frequent – Computer-aided design software is an integral part of any sort of microwave or RF design project. From validating signal integrity to simulating radar scattering, from antenna radiation to electromagnetic compliance of communications devices, CEM is a vast field. Being that it is such an important field, it makes sense to spend large amounts of time and research resources in developing continually faster and more efficient numerical methods and algorithms.

The focus of this project is on open-domain electromagnetics problems. That is, problems which take into account a system in a large amount of free space, such as simulations of radar scattering or antenna radiation behavior. Normally, to simulate antennas or scattering in practice, large anechoic chambers must be built. Any waves reflected from a scattering object during a radar simulation should not continue to bounce around after their initial reflection. This behaves the same as a radar target in the sky. For instance, when a radar EM wave is sent at a plane, it only reflects a single time. The reflected wave is measured by the target, but radar targets are usually far enough isolated in the sky to avoid interference with other possible targets.

A more thorough discussion of CEM’s possible applications is given in chapter IV, with critical examples in engineering and science, as well as graphic illustrations of CEM models.

b.) The Finite Element Method

The Finite Element Method (FEM) is just one of many methods used in the field of computational electromagnetics. It involves discretizing the computational domain into smaller subdomains (elements), and then approximating the EM fields within each element by a linear combination of basis functions, chosen a priori to the simulation. One of FEMs greatest advantages is in its ability to handle unusual or complicated domains. By choosing an effective set of elements and basis functions, we can efficiently model complicated geometries as well as complex media, such as anisotropic and/or inhomogeneous materials. Chapter I contains a more thorough, theoretical description of the Finite Element Method, as well as details on basis function and modeling choices.

The finite element method is not the only commonly used method for computational electromagnetics. Among others are the finite difference time domain method (FDTD) and the volume integral equation (VIE) and surface integral equation (SIE) formulations of the method of moments (MoM). Each of these methods has its own place, and each of them has strengths and weaknesses. One of the unfortunate drawbacks of the finite element method is that we must
have some method to terminate the computational domain. In many problems, sufficient boundary conditions exist that make modeling quite easy, but when it comes to the previously mentioned open-domain problems, we do not have the convenience of nice boundary conditions. Foregoing proper boundaries leads to spurious (nonphysical) reflections of EM waves from the edge of the computational domain[1]. These nonphysical reflections lead to highly inaccurate electric field distributions, and make computations anything less than useful. Chapter II proposes a solution to this with the perfectly matched layer (PML), a computational technique used to accurately and effectively terminate a computer model.

c.) Project Description

The main purpose of this project is to make improvements to the already existing FEM electromagnetic developed in part by the Colorado State University computational electromagnetics lab. This code has already proven effective in applications to waveguides, cavities, scattering, and radiation problems (see publications on FEM by Dr. Branislav Notaros and Dr. Milan Ilic for further details). However, the software required some amount of modernization and optimization, including implementation of new numerical techniques, and quicker computation of older numerical techniques.

In particular, one of the biggest goals of this project is to develop a perfectly matched layer (PML) scheme for use in scattering and radiation problems. The end goal of this long-running project is to develop a full-fledged electromagnetic simulator for effective and fast simulation of all manner of computational electromagnetics problems, from full wave simulation of radar targets and antennas, to waveguide and cavity analysis. In order to have an effective, multi-use solver, it must be able to implement all manner of computational tools. The perfectly matched layer is just one among a score of these tools.

d.) Project Specifications

As this is primarily a research oriented project, the project specifications behind the development of the FEM code are not extremely strict. However, there are a few things taken into consideration when implementing new methods such as the PML, or when just doing general updates or maintenance to the code. For implementation of the PML, we expected to see improvement in accuracy from the previously used absorbing boundary condition (ABC) method.

With regards to optimization of the code, we expected a decrease in the computational time required to simulate a standard problem. Additionally, we required a decrease in the amount of code. Repetition of code is considered poor practice, and in general makes software more difficult to maintain or edit. Sections of code can be combined or removed entirely, to develop a generally more robust and flexible piece of software. As changes are made to the code, it is
essential to the project to continually undergo debugging and testing. Testing and validation are discussed in chapter IV.

One important thing to consider when dealing with any sort of software project is that of system compatibility. Fortunately, due to standards, we can write effective code that can be compiled to run on any sort of hardware system. It is one of the labs goals to eventually parallelize the code, so it can be run on a high-performance computing systems such as the CSU Cray XT6m system. Each platform has its limitations, but we have the potential to eventually port the code to an assortment of computational platforms.

e.) Project Impact

The impact of this project is enormous. Licenses for electromagnetic simulators on the market can run from the low thousands to over 100,000 dollars. As such, any program wishing to survive on this market must be extremely competitive. It must have advanced features comparable with any other electromagnetic solver. For instance, ANSYS HFSS offers a cubical PML implementation for scattering analysis, but not a spherical or conformal PML (different PML shapes can reduce the number of floating-point operations (FLOPS) in a given simulation and decrease the total analysis time). In general, a competitive piece of software should offer all manner of computational options. To keep the electromagnetics lab at the vanguard of research software, it is necessary to continually make improvements upon existing code. Currently, only absorbing boundary conditions have been implemented, but these have been known by the scientific community [2] to be generally less accurate than the PML method. The PML gives a more accurate way to terminate the computational domain, and can actually be combined with the ABC to make our results more accurate than either method alone.

I. Finite Element Method [1]

a.) Theory

In the finite element method, the computational domain is discretized into multiple smaller domains, which we call elements. The electric fields within each element are then approximated using a linear combination of chosen basis functions. These basis functions can be in general vectors or scalar functions. We opt to use vector basis functions (often referred to as edge-based basis functions).

From Maxwell’s equations in the frequency domain, we can easily derive the double curl electric field wave equation:

\[ \nabla \times \mu_r \omega \nabla \times E - k_0^2 \varepsilon_r E = 0 \]

From this equation, we apply the Galerkin testing procedure. We multiply the equation by a testing function, and integrate over all space.
\[
\int_{V} f_{ijk} \cdot (\nabla \times \mu_{r}^{-1} \nabla \times E) \, dV - k_{0}^{2} \int_{V} f_{ijk} \cdot \varepsilon_{r} E \, dV = 0
\]

By applying the Green’s first identity for vector functions, we can rewrite this equation as:

\[
\int_{V} (\nabla \times f_{ijk}) \cdot \mu_{r}^{-1} (\nabla \times E) \, dV - k_{0}^{2} \int_{V} f_{ijk} \cdot \varepsilon_{r} E \, dV = \oint_{S} f_{ijk} \times \mu_{r}^{-1} \nabla \times E \cdot dS
\]

The surface integral on the right hand side denotes the surface enclosing the entire computational domain. In general, the right hand side of this equation is used to apply various boundary conditions. The integrals on the left hand side can be evaluated for each combination of basis and testing functions to generate a linear system of equations, which we here represent in matrix form:

\[
([A] - k_{0}^{2} [B]) [\alpha] = [G]
\]

The electric fields in each element are expanded as a series of basis functions:

\[
E = \sum_{i=0}^{N_i-1} \sum_{j=2}^{N_j} \sum_{k=2}^{N_k} \alpha_{uijk} f_{uijk} (u, v, w) + \sum_{i=2}^{N_i} \sum_{j=0}^{N_j} \sum_{k=0}^{N_k} \alpha_{vijk} f_{vijk} (u, v, w) + \sum_{i=2}^{N_i} \sum_{j=2}^{N_j} \sum_{k=0}^{N_k-1} \alpha_{wijk} f_{wijk} (u, v, w)
\]

Where \( f_{uijk} (u, v, w) \) are the edge-based basis functions, and \( \alpha_{uijk} \) are the corresponding coefficients, which are unknowns to be solved for. By choosing a series of different testing functions, the above testing procedure gives us a set of linear equations, which can be put into a matrix and solved. The Galerkin testing procedure specifies that the testing and basis functions are chosen to be the same.

b.) Choice of Basis Functions

We use the following set of arbitrarily high-order basis functions:

\[
\begin{align*}
    f_{uijk} &= u^{i} P_{j}(v) P_{k}(w) a_{u}' = P_{uijk} a_{u}' \\
    f_{vijk} &= P_{i}(u) v^{j} P_{k}(w) a_{v}' = P_{vijk} a_{v}' \\
    f_{wijk} &= P_{i}(u) P_{j}(v) w^{k} a_{w}' = P_{wijk} a_{w}'
\end{align*}
\]

\[
P_{i}(u) = \begin{cases} 
1 - u, & i = 0 \\
u + 1, & i = 1 \\
v^i - 1, & i \geq 2, \text{ even} \\
v^i - u, & i \geq 2, \text{ odd}
\end{cases}
\]

Where \( a_{u}', a_{v}' \) and \( a_{w}' \) are unitary vectors.

These basis functions have the advantage of being hierarchical [3] (each set of basis functions is a subset of any higher order set) as well as being curl-conforming (i.e., this choice of basis functions satisfies the continuity of the electric fields’ tangential component across the boundaries between elements). It is important to note that the basis functions are only defined over a single element. Therefore, the testing integrals only need to be evaluated over the volume.
of a single element. Additionally, these integrals will be zero if the basis and testing function correspond to unknowns from two different elements.

The first couple of functions $P_i(u)$ are shown in figure 1. The first two basis functions provide continuity between the different elements of the system (we do this by equating these unknowns between elements), while the higher order basis functions provide a more accurate approximation to the electric field inside the element.

Expanding the field approximation with higher order basis functions allows us to increase the size of each element and reduces the number of overall unknowns in the linear system. We are able to model the entire computational domain with a few high order elements rather than a large amount of first order elements.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{basis_functions.png}
\caption{Basis functions for FEM (in 1-D) [4]}
\end{figure}

\textbf{c.) Geometrical Elements}

When the size of each element increases, objects must be carefully constructed from a set of these elements. In the past, tetrahedral and brick shaped elements have been used. These choices are not extremely flexible for arbitrary shaped domains, especially when dealing with large or unusually shaped models. Instead, we opt to use a set of generalized hexahedral elements, which are mapped via Lagrange-type interpolation polynomials from the physical model shape to a set of local coordinates $u, v, w$, over which the basis functions are defined. This mapping allows us to define a (theoretically) arbitrary high geometrical order to the element. This additional high order geometric approximation allows the elements to be fit to more complicated geometries, which are not necessarily easy to model with tetrahedral or brick elements.

The mapping from the element to the local coordinates is given by [5]:

\[ r(u, v, w) = \sum_{i=1}^{M} r_i \hat{L}_i^{K_w}(u, v, w), \quad -1 \leq u, v, w \leq 1 \]

Where \( r_i \) define the interpolation nodes used to define the element, and \( \hat{L}_i^{K_w}(u, v, w) \) are the polynomials defined by:

\[
\hat{L}_i^{K_u}(u, v, w) = L_m^{K_u}(u)L_n^{K_v}(v)L_k^{K_w}(w),
\]

\[
0 \leq m \leq K_u, \quad 0 \leq n \leq K_v, \quad 0 \leq l \leq K_w \quad 1 \leq i \leq M = (K_u + 1)(K_v + 1)(K_w + 1)
\]

\( K_u, K_v \) and \( K_w \) are geometrical orders of the element, and \( L_m^K \) are Lagrange-type interpolating polynomials, given by the formula

\[
L_m^K(u) = \prod_{j=0, j \neq m}^{K} \frac{u - u_j}{u_m - u_j}
\]

Where \( u_j \) are interpolating nodes. Figure 2 shows an example of one of these elements, mapped from the cubical domain over which the basis functions are defined, to the physical, hexahedral domain.

**Figure 2: Mapping from element to local coordinate system [1]**

d.) Anisotropic and Inhomogeneous Media

In order to implement the perfectly matched layer, it’s important to have code that can efficiently and accurately simulate anisotropic and inhomogeneous materials. Fortunately, due to the flexibility of the geometrical elements we use, simulating these complicated media is very
doable. For a general anisotropic and inhomogeneous element, we can perform the same Lagrange interpolation scheme used in the geometrical modeling to expand the permittivity tensor in terms of a series of Lagrange polynomials, each weighted by the tensor at one of the interpolation nodes [6].

$$\overline{\epsilon}_r(u, v, w) = \begin{bmatrix} \epsilon_{r,xx}^e(u, v, w) & \epsilon_{r,xy}^e(u, v, w) & \epsilon_{r,xz}^e(u, v, w) \\ \epsilon_{r,yy}^e(u, v, w) & \epsilon_{r,yy}^e(u, v, w) & \epsilon_{r,yz}^e(u, v, w) \\ \epsilon_{r,zz}^e(u, v, w) & \epsilon_{r,zy}^e(u, v, w) & \epsilon_{r,zz}^e(u, v, w) \end{bmatrix} = \sum_{m=0}^{M_x^e} \sum_{n=0}^{M_y^e} \sum_{p=0}^{M_z^e} \overline{e}_{r,mnp} L_{m^e}^{M_x^e}(u) L_{n^e}^{M_y^e}(v) L_{p^e}^{M_z^e}(w)$$

With this, our code can provide an accurate continuous approximation to any general material. To put the use of this feature in perspective, ANSYS HFSS, an industry standard full wave simulator, can only approximate inhomogeneous materials with piecewise constant approximations. This requires a larger number of elements and a much larger amount of unknowns. The flexibility of using higher order approximations greatly decreases the computational time required.

### e.) Comparison to Other Numerical Methods

Outside of the finite element method, several other numerical methods exist for solving scattering and radiation problems. Perhaps one of the oldest and most used methods for solving PDE’s is the finite-difference (FD) method, which involves discretizing the computational domain into a grid and approximating all derivatives by difference equations. This method has the advantage of being extremely flexible (changes to the underlying PDE are often easy to implement). However, it does not provide a nice way to model curved geometries, and is notoriously inaccurate when compared with Galerkin methods such as the method of moments or finite element method.

The method of moments (MoM) is similar to the FEM, in that it involves discretizing the domain into individual elements or patches, but it differs in that the equation being solved is formulated as an integral equation rather than a PDE. There are two main implementations of the MoM. The surface integral equation (SIE) formulation requires only meshing of the surface surrounding our scattering body. This in convenient in its speed, but only allows us to model simple, homogeneous, isotropic media. The volume integral equation (VIE) formulation does have the same adaptability as the FEM when modeling complex media, but scales differently than the FEM. Due to the sparse nature of FEM matrices, the FEM scales very well with an increasing number of elements. The same cannot always be said for the VIE method, which involves discretizing the problem into a dense matrix problem.
II. Perfectly Matched Layer

a.) Purpose

The perfectly matched layer is currently the biggest focus of this project. Since in the Finite Element Method we cannot model all of free space, we are required to terminate the computational domain in such a way that does not cause unphysical results. Several different schemes have been suggested in order to do this. One extremely common method is the ABC, as presented in several scientific papers and many well-known books on electromagnetics (e.g., [7]). Many variants on the ABC have arisen, including the extension to a second-order ABC.

More recently, the PML boundary condition has surfaced, which is able to be carefully designed so as to provide better computational results than the ABC [2]. As opposed to the ABC, which involves applying a boundary condition at the edge of the domain, the PML actually involves us modeling a layer around the physical computational domain, which will not cause unphysical reflections, but will attenuate any outgoing waves.

The ABC is problematic in that its effectiveness is dependent on the size of the computational domain. Additionally, it requires terminating the domain in a spherical surface. For scattering bodies of arbitrary shapes, it often doesn’t make sense to have a large, spherical domain. The PML works regardless of size, and can be made to work regardless of shape. By shrinking the computational domain, we reduce the number of unknowns in the corresponding linear system of equations, and thus the computational time required.

This decrease in computational is very important, and also is typically paralleled with a decrease in needed RAM. In the world of numerical modeling, these two factors are absolutely the most critical considerations, and will help keep our FEM code more competitive with industry standard solvers.

b.) Theory

Implementing the PML involves a reformulation of the double curl vector wave equation to a more general equation [8]. This can be seen as an analytic continuation of Maxwell’s equations from the familiar real-valued 3-dimensional space to a more general complex-valued 3-dimensional Cartesian space:

We define new complex coordinates as:

\[ \tilde{x} = x_0 + \int_{x_0}^{x} s_x(x') dx' \]

Where \( s_x \) is a complex stretching factor to be optimized for the most desirable behavior of the PML layer. \( x_0 \) is the coordinate at the boundary between the air and PML layers. By choosing
the real part of $s_x$ to be continuous across the boundary, the interface will exhibit zero reflection.
In a Cartesian coordinate system with a cubical PML, the double-curl wave equation can be rewritten as follows, if the stretching factor is assumed to be constant in the PML:

$$\nabla \times \mu_0^{-1} \nabla \times E - k_0^2 \varepsilon_0 E = 0,$$

where $\nabla_s = \frac{1}{s_x} \frac{\partial}{\partial x} + \frac{1}{s_y} \frac{\partial}{\partial y} + \frac{1}{s_z} \frac{\partial}{\partial z}$.

In general, fields that satisfy this equation do not satisfy Maxwell’s equations, unless $s_x = s_y = s_z = 1$. If $s_x, s_y$, and $s_z$ are chosen appropriately (as complex numbers), it can be shown that the medium will attenuate any propagating waves. Additionally, an interface between a medium with these parameters and a medium with $s_x = s_y = s_z = 1$ will exhibit zero reflection, as long as both media have the same relative permittivity and permeability $\varepsilon_0$ and $\mu_0$.

In order for $s_x, s_y$, and $s_z$ to properly attenuate outgoing waves, they must be chosen differently depending on their location within the computational domain. For instance, if a cubical PML layer is modeled, each side of the cube will have different values for $s_x, s_y$, and $s_z$. For this reason, it is a more difficult problem to generate an arbitrarily-shaped PML that will attenuate any outgoing wave. The eventual goal of this project is to implement a conformal PML that will accomplish exactly this task.

It can be shown that the so-called “stretching” parameters, $s_x, s_y$, and $s_z$, can be absorbed into $\varepsilon_0$ and $\mu_0$ if they are taken to be 3x3 complex tensors instead. The problem then does not require reformulation of the curl operator, but merely modeling of the proper anisotropic medium to correspond to the desired values of $s_x, s_y$, and $s_z$. [7]

The appropriate equation is now just

$$\nabla \times \tilde{\Lambda}^{-1} \mu_0^{-1} \nabla \times E - k_0^2 \varepsilon_0 E = 0$$

Where

$$\tilde{\Lambda} = \begin{bmatrix}
S_x S_z & 0 & 0 \\
S_z & 0 & 0 \\
S_x S_z & 0 & 0 \\
0 & 0 & S_x S_y \\
\end{bmatrix}$$

By rearranging the equations, we have gone from a so-called “Non-Maxwellian” PML, in which we redefined the 3D space to be complex (change in the curl operator), to a “Maxwellian” PML, in which we again return to a 3D real space (standard curl operator), with a complex tensor for multiplication with the permeability and permittivity of the media. If this tensor is absorbed into
the permittivity and permeability tensors, the PML equation is again just the standard vector wave equation.

In the first semester of the project, the PML layer was derived fully for the case of a cubical PML. However, this is not the optimal method for attenuation of outgoing waves. For an arbitrarily shaped scatterer, it may be more efficient to model other shaped PML layers to surround the domain.

In spherical coordinates, a similar complex tensor can be derived, as derived in [11]:

\[
\tilde{\Lambda} = \begin{bmatrix}
\tilde{r} & 1 & 0 & 0 \\
r & S_r & 0 & 0 \\
0 & S_r & 0 & 0 \\
0 & 0 & S_r & 0
\end{bmatrix}
\]

Note that in the spherical case, the analytical continuation is only performed in the radial direction. In general, the analytical continuation will be done in the direction normal to the domain boundary.

**c.) Wave Equation Reformulation**

In the traditional implementation for scattering problems, the excitation of the scatterer is done on the very outside of the computational domain. However, as the outer layer of the computational domain (the PML) now experiences attenuation, we cannot excite the domain
from the very outside. We require a reformulation of the original computational problem. We recognize that in a scattering problem, the total electric field $E^{\text{Total}}$ can be decomposed into two separate fields $E^{\text{Incident}}$ and $E^{\text{Scattered}}$ [2]. The incident electric field is generally given by a plane wave, which we know analytically. We can rewrite the double-curl wave equation in terms of these two fields separately.

$$\nabla \times \mu^{-1}_r \nabla \times E^{\text{Scattered}} - k_0^2 \varepsilon_r E^{\text{Scattered}} \neq \nabla \times \mu^{-1}_r \nabla \times E^{\text{Incident}} - k_0^2 \varepsilon_r E^{\text{Incident}}$$

With this new formulation, we now have two separate domains, each with their own form of this equation. Rather than expanding the entire electric field in terms of the basis functions, we just expand the scattered field, since the incident field is already known. Inside the scattering object (which has either $\varepsilon_r \neq 1$ or $\mu_r \neq 1$), the right hand side becomes a vector after applying the testing procedure. The left hand side remains the same as before, only the unknowns are now correlated only to the scattered field.

After applying the testing procedure we arrive at:

$$\int \left( \nabla \times \mathbf{f}_{ijk} \right) \cdot \mu_r^{-1} \left( \nabla \times E^{\text{Scattered}} \right) \, dV - k_0^2 \int \mathbf{f}_{ijk} \cdot \varepsilon_r E^{\text{Scattered}} \, dV - \oint_{S} \mathbf{f}_{ijk} \times \mu_r^{-1} \nabla \times E^{\text{Scattered}} \cdot dS =$$

$$\int \left( \nabla \times \mathbf{f}_{ijk} \right) \cdot \left( \mu_r^{-1} - I \right) \left( \nabla \times E^{\text{Incident}} \right) \, dV - k_0^2 \int \mathbf{f}_{ijk} \cdot \left( \varepsilon_r - I \right) E^{\text{Incident}} \, dV$$

Here, $I$ is the 3x3 unit or identity matrix. This is somewhat different than the formulation derived during the first semester of this project:

$$\int \left( \nabla \times \mathbf{f}_{ijk} \right) \cdot \mu_r^{-1} \left( \nabla \times E^{\text{Scattered}} \right) \, dV - k_0^2 \int \mathbf{f}_{ijk} \cdot \varepsilon_r E^{\text{Scattered}} \, dV - \oint_{S^D} \mathbf{f}_{ijk} \times \mu_r^{-1} \nabla \times E^{\text{Scattered}} \cdot dS =$$

$$\int \left( \nabla \times \mathbf{f}_{ijk} \right) \cdot \mu_r^{-1} \left( \nabla \times E^{\text{Incident}} \right) \, dV - k_0^2 \int \mathbf{f}_{ijk} \cdot \varepsilon_r E^{\text{Incident}} \, dV - \oint_{S^{Sc}} \mathbf{f}_{ijk} \times \mu_r^{-1} \nabla \times E^{\text{Incident}} \cdot dS$$

Here, $S^D$ denotes the outer border of the computational domain, and $S^{Sc}$ denotes the surface of the scattering object.

After extensive literature reviews and testing, the wave equation we are solving needed to be continuously reformulated until finding a model that worked. The equation we chose to use in the first semester was found to be unnecessarily complicated and identical to the new formulation. The surface integral term is no longer necessary, which is one of the most difficult integrals to implement.

In free space and inside the PML, the right hand side of this equation reduces to 0 and we are only left with the left hand side, the scattered field formulation. After reformulating the problem
at hand, the rest of the task is mainly an issue of generating models with PML shapes appropriate to the particular scattering problem.

During the first semester of this project, the wave equation was reformulated, but not verified. During the second semester, the reformulated wave equation was finally implemented successfully, and compared to the original code with excellent agreement in all results.

As part of the reformulation, extensive modularization and pre-computation of key equations was done. It is easiest to optimize code while it is being completely redone, rather than afterwards. Via smarter computation, the code was sped up substantially.

**d.) Extension to conformal PML**

The analytic continuation of variables above is applicable to any arbitrary shape. We can, in general, always do the same analytic continuation. In a general coordinate system, the variable we choose to do the analytic continuation of is simply the direction normal to the air cushion at the air/PML interface. By proper coordinate transforms, the chosen fields (generally non-maxwellian), can be mapped back to real space. So in general, we now have a scheme that can be applied to scatterers of any shape, simply by careful modeling of an anisotropic, inhomogeneous PML later.

In order to uniquely define the conformal PML, it is usually desirable to design a layer that conforms to a convex shape. So, typically in literature, the convex hull (smallest convex surrounding area) of a scattering object is considered. The total domain would then consist of the scatterer, a convex air cushion surrounding it, and finally the PML layer itself. Figure 4 is an example of an arbitrary mesh from literature, with the PML chosen to be conformal to the scatterer. There is a necessarily always a thin layer of air between the PML and the scatterer, to ensure proper reflection from the scatterer surface.
III. Applications of Computational Electromagnetics

One of the most important questions arising during a discussion of this project is what the purpose of computational electromagnetics really is. This field has a vast amount of applications, ranging from military applications, to medical information, and communications systems. Electromagnetic interactions are a fundamental part of any electronics system, and cannot be ignored, especially as electronics systems move to higher and higher frequencies.

a.) Electromagnetic Compliance

Often a question of great importance is how radiating devices interact with the human body and with other electromagnetic devices. Devices are expected not to interfere with each other. Additionally, sometimes we have to worry about the amount of radiated power. A cell phone, for instance, is often placed right up against the human head. It might be desirable during the design process to figure out what the relevant field strengths in the brain are.
Figures 5a and 5b show an example of a model made for simulating scattering from a human brain. By choosing appropriate material values, we have a model that behaves identically to a human brain at radio frequencies. This model was meshed for use with the electromagnetics lab’s VIE code.

![Model of a Human Brain for Scattering Simulation in MoM](image)

**Figure 5a: Model of a Human Brain for Scattering Simulation in MoM [9]**

![Computed Electric Field (normalized) inside the Near-Field Plane](image)

**Figure 5b: Computed Electric Field (normalized) inside the Near-Field Plane at 900MHz**

### b.) Radar Design and Simulation

Modern radar systems have vast applications, from weather prediction, to target detection, and even analysis of cosmic background radiation. In the case of target detection we may, for instance, want to design a stealth plane to be as undetectable as possible. By building a CAD model, we can carefully minimize the backscattered wave, thus making the plane nearly invisible to a standard monostatic radar system.
c.) Antenna Design

CAD is used commonly in the field of antenna design, in which we analyze the input impedances of an antenna, at a range of frequencies. We may also wish to know the strength of the antennas radiation as a function of direction.

Figure 7 shows a metallic corrugated horn antenna meshed for simulation with SIE code. This is just an example of the multitude of antenna devices that can be simulated with CAD.

d.) Signal Integrity

When dealing with microwave and RF circuit design, it is also important to ensure signal integrity. For instance, in designing a stripline circuit, we want to make sure power signals aren’t bouncing around too much on the lines after being reflected from a load. Additionally, coupling between different traces on a PCB can lead to noisy circuits and bad signals.
Arguably the most important phase of software development is the testing and validation phase. Any software that is shipped to a customer is expected to be 100% robust and accurate. Similarly for our software, which is used in a research setting; if the software is not working completely, we cannot expect to apply it in solving major research problems. For this reason, the software must be continually tested as it is refined.

Since this software was already in operation prior to the beginning of this project, a large portion of the needed testing is simply assuring that any changes we’ve made to the code do not affect its operation in a negative way. This is particularly important when optimizing older sections of the code. For this reason, we have a series of benchmark tests, which we can continually rerun to ensure accuracy and expected operation of the code are maintained. These benchmarks have known correct results, whether from analytic solutions, literature comparisons, or comparisons with commercial electromagnetic solvers. Figure 9 shows a sphere meshed in WIPL-D v11. WIPL-D is a commercial electromagnetic solver used to simulate scattering problems as well as antenna radiation problems. We typically use software such as this as a good comparison point.

It is important to have benchmarks for every possible section of code we change. For the largest part of this project, we are dealing with scattering problems. We must simulate multiple shapes and sizes of scatterers to ensure robustness of the code.
V. Successful Results

In developing the PML for the FEM code, several models had to be and tested. For the purposes of this project, we simulated fairly simple and straightforward models, for ease of meshing. All comparisons were done with WIPL-D electromagnetic solver, which computes scattering from surface meshes via the method of moments.

The first model consisted of a cubical domain, divided into a 7x7x7 grid of cubical elements. The center element was chosen to have a dielectric constant of 2.25. This is a form of small-domain modeling, in which we model a domain with a larger amount of smaller elements. This model had excellent results when compared with WIPL-D, which are shown in figure 11.

After getting the small domain cubical scatterer to work, we moved on to a larger domain model. This consists of the same center dielectric element, except now just surrounded with 26 PML elements, shaped as to better approximate the same cubical domain with smaller elements. The results again were eventually made to be just as good.

Finally, we replaced the cubical scatterer with a spherical scatterer in order to ensure the method worked for geometrically higher-order domains. The results of this are shown in figure 12, and again match almost exactly with WIPL-D.

The stretching factor $s$ was chosen to be a constant of $s = 1 - 2i$ for all examples shown here (after a little bit of trial and error.)
Figure 11: Scattered Z-field component of electric field: A.) Lab FEM code, linear scale. B.) WIPL-D solver, linear scale. C.) FEM code, log scale. D.) WIPL-D, Log Scale

Figure 12: Scattered Z-field component of electric field from spherical scatterer. Comparison of our code vs. WIPL-D
Overall, these results are very good. We have shown that the PML works with the higher order FEM method presented here, and that it can be used instead of the previous inefficient ABC method.

The proper operation of the PML proved to be more difficult than expected in the first semester. Numerous setbacks due to small errors in the code continued to delay the PML’s proper operation, but successful results were finally demonstrated.

VI. Update to Input File

With the update of the FEM code and the way in which anisotropic/inhomogeneous materials are handled, the FEM input file needed to be modernized.

The 3D element parameter description (input file section 6) now should be used as follows:

```
6. 3D ELEMENT PARAMETERS DESCRIPTION (Element lines):
   El. no. - hcode - icode - pmlcode - epsre - epsim - mure - muim
```

<table>
<thead>
<tr>
<th>El. no:</th>
<th>Element number</th>
</tr>
</thead>
<tbody>
<tr>
<td>hcode:</td>
<td>Homogeneity code(1 or 0) [1 - homogeneous, 0 - inhomogeneous]</td>
</tr>
<tr>
<td>icode:</td>
<td>Isotropy code (1 or 0) [1 - isotropic, 0 - anistropic]</td>
</tr>
<tr>
<td>pmlcode:</td>
<td>perfectly matched layer code (1 or 0) [1 - PML, 0 - Non-PML]</td>
</tr>
<tr>
<td>epsre:</td>
<td>Real(Epsilon Relative)</td>
</tr>
<tr>
<td>epsim:</td>
<td>Imaginary(Epsilon Relative)</td>
</tr>
<tr>
<td>mure:</td>
<td>Real(Mu Relative)</td>
</tr>
<tr>
<td>muim:</td>
<td>Imaginary(Mu Relative)</td>
</tr>
</tbody>
</table>

For every inhomogeneous or anistropic element, add the following lines:

```
El. no. - Kuvw - Symmetry
EpsRxx(1) - EpsRxx(2) - ... - EpsRxx(8) - ... - EpsRxx(27)
EpsRxy(1) - EpsRxy(2) - ... - EpsRxy(8) - ... - EpsRxy(27)
.
.
.
EpsRzz(1) - EpsRzz(2) - ... - EpsRzz(8) - ... - EpsRzz(27)
MuRxx(1) - MuRxx(2) - ... - MuRxx(8) - ... - MuRxx(27)
.
.
.
MuRzz(1) - MuRzz(2) - ... - MuRzz(8) - ... - MuRzz(27)
```

Kuvw: inhomogeneity approximation order (0 for anistropic, homogeneous)

In each EpsR and MuR line, there should be (Kuvw + 1)^3 entries

Epsr and MuR must be complex numbers specified in cartesian form(e.g. EpsRxx = 2.25 0.0)
If Symmetry = 1, there should be 6 lines for EpsR and 6 for MuR (xx,xy,xz, yy,yz,zz components of tensor)
If symmetry = 0, full tensor must be specified (9 lines each)

Following the data line should be the material parameter descriptions for every element, followed by the material parameter descriptions for anisotropic/inhomogeneous elements. Input data format may be confusing to those not familiar with the FEM code operation, so an example input file would be provided to help improve clarity.

Other sections of the input file were changed as well, but to a far less drastic degree.

**VII. Output Plotting**

We decided to use near-field results as the benchmark for comparisons with WIPL-D. In order to properly view results, we had to make several changes to the code, as well as write MATLAB scripts to properly parse and process the output data, both from WIPL-D and the FEM code.

In FEM, it is more difficult to process near-field results than results in the far-field. The basis functions are defined locally over a unit-cube, which is then mapped to the physical domain. When the electric fields are solved for, however, they are given in the local coordinate system of the cube rather than in an x,y,z cross section. It is much easier to process results outside of the FEM code than within it.

If we wish to always get an even cross section in the x,y,z domain, we calculate fields inside a uniform grid in the local u,v,w coordinate system, and then calculate their corresponding x,y, and z coordinate values. We can then perform a linear interpolation on these results to get the desired uniform cross section in x,y,z coordinates.

The above was done for all previously shown results in MATLAB. This is considered post-processing, and so is not currently done in the FEM solver itself.

**VIII. Pre-calculation of Excitation Vector**

As an example of one of the more important optimizations done in this project, and some of the considerations taken into account, we will take into question the so-called “excitation vector”. This is the series of integrals given by the right hand side of the new discrete wave equation (detailed in section II-c).

Often in computation EM, we are interested in the frequency response of an object. In general, none of the volume integrals on the left side of the wave equation are frequency dependent. They need only be calculated a single time. However, the aforementioned excitation vectors need to be recalculated for every new frequency, before the corresponding matrix equation is solved.
Previously these vectors were calculated as the frequency sweep was performed. A vector would be calculated for a given frequency before the matrix equation would be solved for the corresponding frequency point. However, it is much more efficient to pre-calculate these vectors while the main FEM matrices are being filled. This leads to an increase in the required memory (all vectors must be stored at all times), but a decrease in the overall integration time. This is just an example of one of the problems that must be taken into account when designing efficient numerical software. It is assumed here that any end-user will have access to high-performance computing systems (i.e., virtually unlimited RAM), so we here opt for computational speed over efficiency of memory.

**IX. Conclusion and Future Plans**

This project has been a success. We were able to successfully implement the PML scheme, and see decent acceleration of matrix filling. Although the true conformal PML has not yet been achieved, the code is considered completely ready for conformal PML models (once appropriate meshes have been generated). The only thing that needs to happen is careful meshing and further testing of the code.

Among the improvements that need to be made are improvement and development of benchmarks. The number of benchmarks for PML testing is still rather small, limited to the aforementioned spherical and cubical shapes.

This project will not be continued as a senior design project, but it will be worked on by members of the CSU computational electromagnetics lab as seen necessary.

**Continued development of the PML**

Now that the PML has been confirmed to work, it is important to continue revising the code and to generate a series of more realistic benchmark tests. Having benchmarks accelerates the future optimization of the FEM code. Additionally, there are a few more optimizations that can be made.

**Parallelization**

In order to realistically use the newly developed code for large domain computational problems, it must be effectively parallelized. Parallelization has been done in the past, and we expect to eventually make sure all additional parts of the program are parallelized to maximize computational efficiency. For the purposes of the electromagnetics lab’s research, parallelization will allow codes to be run on the Cray XT6m high-performance computing system, which greatly increases the size and scope of problems we can simulate.
Second Order PML

There exists a formulation for the perfectly matched layer which is considered second order, in that it exhibits much better convergence to proper results. Beyond implementing the simplest formulation of the PML talked about here, we can choose more advanced material parameters such that our PML will indeed produce even more accurate results. This should be simply a matter of more complicated modeling, however, and probably won’t prove difficult after ensuring proper operation of the conformal PML.

Applications

The PML has other uses outside of basic scattering and radiation problems. For instance, it can be used to analyze waveguides of varying dielectric composition. The PML will allow for the general advancement of the electromagnetics lab’s research by increasing the scope of simulations we can do. After implementation and exhaustive testing, we may begin applying the PML to novel research problems.

Further Changes

As part of this project, the FEM code was carefully evaluated and documented. There are several other (minor and major) improvements that can be made to the FEM code as a whole to make it more robust, reliable, and efficient. These improvements have been noted, and will be made as the FEM project progresses.

Recommendation for Future Work

As was previously stated, this project will not be continued as a senior design project in the future. The computational electromagnetics lab will continue to work on the FEM code for the time being, with the intent of developing a fully tested, robust, reliable EM solver for full wave simulation of any arbitrarily shaped scattering body. Although this is not the main computational method we use to currently solve scattering problems, it has does have the advantage of scaling over the VIE method, which is also often used in our lab.

With the modernizations and optimizations implemented by this project, it is hopeful that the FEM can again become a viable solution in any arbitrary scattering problem. Additionally, the new numerical method of the PML increases its usefulness even more. There are more even more optimizations that can be done, and it is the recommendation of the team to continue working on the FEM code.
References


Bibliography


Appendix A: Abbreviations

ABC: Absorbing Boundary Condition
CAD: Computer Aided Design
CEM: Computational Electromagnetics
IEEE: Institute of Electrical and Electronics Engineers
EM: Electromagnetics
FD: Finite Difference
FDTD: Finite Difference Time Domain
FEM: Finite Element Method
MoM: Method of Moments
PEC: Perfect Electric Conductor
PML: Perfectly Matched Layer
RAM: Random Access Memory
RF: Radio Frequency
VIE: Volume Integral Equation

Appendix B: Budget

Since this project consists solely of literature research and software development, very little was needed with regards to financing. Around $10 in expenses were needed in the spring semester, to pay for presentation materials. The only costs in development of this software are in the licensing of development and testing software, which is taken care of by the electromagnetics lab and/or the university.
## Appendix C: Project Plan Evolution

### Original Project Plan

<table>
<thead>
<tr>
<th>Begin Date</th>
<th>End Date</th>
<th>Tasks</th>
<th>Deliverables</th>
</tr>
</thead>
</table>
| 9/1/14     | 9/15/14  | • Preliminary Research  
• Reading – PML  
• Reading FEM  
• Reading – Curvilinear Coordinate Systems | |
| 9/16/14    | 9/30/14  | • Understanding Code  
• Code Organization  
• Reading – FORTRAN Language | Document of basic need-to-know parts of code for new students, as well as edited and organized code (Aaron, Ana). |
| 10/1/14    | 11/31/14 | • Implementation of cubical PML  
• Testing of cubical PML  
• Additional code optimization  
• Reading – Conformal PML | Report of cubical PML, as well as finished code with working cubical PML, and testing results with comparisons to outside sources (Aaron, Ana, Sanja). |
| 12/1/14    | 12/31/14 | • Implementation of spherical PML  
• Testing of spherical PML | Report on spherical PML, as well as code, documentation, derivations and formulations (Aaron, Ana). |
| 1/1/15     | 3/31/15  | • Implementation of conformal PML | Working code with conformal PML and equations/derivations for conformal PML, with list of good literature sources (Aaron, Ana, Sanja). |
| 3/31/15    | End      | • Testing conformal PML | Document with computational results, compared to external sources (Aaron, Ana, Sanja). List of further changes to code, items to improve (Aaron, Ana). |
## Revised Project Plan – 1st semester

<table>
<thead>
<tr>
<th>Begin Date</th>
<th>End Date</th>
<th>Tasks</th>
<th>Deliverables</th>
</tr>
</thead>
</table>
| 9/1/14     | 9/15/14  | • Preliminary Research  
• Reading – PML  
• Reading FEM  
• Reading Curvilinear Coordinate Systems |  |
| 9/16/14    | 9/30/14  | • Understanding Code  
• Code Organization  
• Reading FORTRAN Language | Document of basic need-to-know parts of code for new students, as well as edited and organized code (Aaron, Ana). |
| 10/1/14    | 12/31/14 | • Implementation of cubical PML  
• Testing of cubical PML  
• Additional code optimization  
• Reading – Conformal PML | Report of cubical PML, as well as finished code with working cubical PML, and testing results with comparisons to outside sources (Aaron, Ana, Sanja). |
| 1/1/15     | 3/31/15  | • Implementation of conformal PML | Working code with conformal PML and equations/derivations for conformal PML, with list of good literature sources (Aaron, Ana, Sanja). |
| 3/31/15    | End      | • Testing conformal PML | Document with computational results, compared to external sources (Aaron, Ana, Sanja).  
List of further changes to code, items to improve (Aaron, Ana). |
## Revised Project Plan – 2nd Semester

<table>
<thead>
<tr>
<th>Begin Date</th>
<th>End Date</th>
<th>Tasks</th>
<th>Deliverables</th>
</tr>
</thead>
</table>
| 9/1/14     | 9/15/14  | • Preliminary Research  
             • Reading – PML  
             • Reading – FEM  
             • Reading – Curvilinear Coordinate Systems | |
| 9/16/14    | 9/30/14  | • Understanding Code  
             • Code Organization  
             • Reading – FORTRAN Language | Document of basic need-to-know parts of code for new students, as well as edited and organized code (Aaron, Ana). |
| 10/1/14    | 2/31/15  | • Implementation of cubical PML  
             • Testing of cubical PML  
             • Additional code optimization  
             • Reading – Conformal PML | Report of cubical PML, as well as finished code with working cubical PML, and testing results with comparisons to outside sources (Aaron, Ana, Sanja). |
| 2/31/15    | 3/31/15  | • Implementation of higher geometrical order cubical PML | |
| 3/31/15    | End      | • Literature research on conformal PML  
             • Spherical PML implementation. | Document with computational results, compared to external sources (Aaron, Ana, Sanja).  
List of further changes to code, items to improve (Aaron, Ana). |
ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Branislav Notaros, for his invaluable advice and assistance in my academic development, and in getting me started on this project. Additionally, I would like to thank my team members, Ana Manic and Sanja Manic, as well as the other graduate students of the electromagnetics lab, Elene Chobanyan, and Nada Sekeljic, for their advice and guidance in my learnings.