Convergence and Solver Study of FEM Systems With PML Truncation

Y. Botros
J. Volakis

Final Report

August 1997

PROJECT INFORMATION

PROJECT TITLE: Accelerated 3D Arbitrary EM Simulation Methods

REPORT TITLE: Convergence and Solver Study of FEM Systems with PML
Implementation

U-M REPORT No.: 034702-4-T

CONTRACT
START DATE: 1 August 1996
END DATE: 31 July 1997

DATE: August 1997
Final Report

U-M PRINCIPAL
INVESTIGATOR: John L. Volakis
EECS Dept.
University of Michigan
1301 Beal Ave
Ann Arbor, MI 48109-2122
Phone: (313) 764-0500   FAX: (313) 747-2106
volakis@umich.edu
http://www-personal.engin.umich.edu/~volakis/

CONTRIBUTORS
TO THIS REPORT: Y. Botros, J. Volakis

Design Guidelines for the Perfectly Matched Layer (PML) Absorbers

Youssry Y. Botros and John L. Volakis

Radiation Laboratory

Department of Electrical Engineering

and Computer Science,

The University of Michigan

Ann Arbor, MI 48109-2212

Forward

Project Goals

This report serves as the 4th quarterly and final report of the subject contract with Compact software under the MAFET program. In summary, the first year goals of this project were

1. to investigate the performance of perfectly matched absorbers for truncating finite element meshes associated with microwave circuit modeling.

2. to recommend guidelines for using PML and recommendations for the optimum PML loss and phase parameters.

3. to investigate convergence characteristics of finite element systems with PML truncation.

4. to apply the research results for different microwave circuits and structures.

5. to suggest design guidelines for the PML parameters and to propose efficient and robust scheme for splving the FEM system.

Because of the slower convergence of finite element systems with PML truncations, we also proceeded (in consultation with Sponsor) to investigate improved solvers for reliable solution with these systems. We began looking at improved solvers during the middle of the year, but results of these investigations were not generated till the fourth quarter. After a survey of solvers, we specifically concentrated on the generalized minimal residual method (GMRES) with and without preconditioning and on the sparse LU solver developed at NASA Langley. The latter became available over the past year. A brief summary of our accomplishments during the one-year period is given below. It should be noted that the subject of solver technology is evolving quite rapidly and one could not therefore expect great strides over a period of few months. This is particularly so for University efforts which are aimed at high risk endeavors with higher than average payoff. At this moment, we have just began to apply the new solvers to applications of interest and we have developed a good sense of their capabilities over previously used solvers. Our results and recommendations are therefore preliminary but very promising. It is highly recommended that the solver study be continued before finalization of our present recommendations. Moreover, new approaches and solvers are evolving. For example, it is worth examining the

- flexible GMRES

- new flexible preconditioning schemes

- optimized LU solvers like the FMS

- fast algorithms like the FMM and AIM

- frequency extrapolation schemes

We have some experience with all of these solvers, but have not investigated them for microwave circuit applications.

Summary of Accomplishments

The following is a brief summary of our accomplishments over the past year.

1- PML performance (1st quarter)

During our first report, we examined the performance of the PML developed by Sacks et.al. [17] for truncating finite element domains associated with microwave circuits. It was found that the PML can deliver better than 60 dB absorption using an anisotropic layer of 0.3 or so wavelengths for truncating the FEM domain. Of importance is that this absorption is nearly angle independent except near grazing and this is a major advantage of the PML over the absorbing boundary conditions. It was also found that the attenuation constant and discretization rate must be selected with care for optimal performance. The optimality curves generated for the one-dimensional study before the start of the contract were applied to the three-dimensional simulations considered here and were found equally effective.

2- Convergence Study (2nd quarter)

Although the PML was found to be quite superior for wave absorption, our first quarter study also indicated that the FEM system convergence deteriorates with PML truncations. The convergence deterioration may actually be severe and prompted the investigation of preconditioning schemes and new solvers. We found that the diagonal preconditioning improved the convergence rate substantially without computational effort. Most importantly, in our second quarter we found that an optimal phase parameter could be specified for improved convergence. The effect of the phase parameter was not realized till this now and an investigation was performed to determined optimal loss as well as phase parameters. It was determined that setting α and β to

3

unity was a good choice for most cases without a need to refer design curves.

During the second quarter we also began looking at the more robust GMRES solver. Our preliminary study during the second quarter demonstrated that GMRES, although requiring more resources, delivered faster convergence and without erratic residual behaviors.

3- Applications (3rd quarter)

The solver study was interrupted to concentrate on applications of the designed PML along with the preconditioned systems. We examined the extraction of the S_{11} and S_{12} parameters for coupled microstrip lines and gave coupling results. We actually concentrated on the extraction of the S parameters using the vertical fields underneath the transmission lines. These are equivalent to the currents on the microstrip lines which are typically used for the S parameter extractions. Our procedure for parameter extraction was explained in the quarterly report. For feed excitation, it was determined that 4 to 5 horizontal electric probes were sufficient to achieve convergence. We concluded that the PML can be brought very close to the feed and still get very good S parameter predictions. However, the PML distance from the feed may need to be at least 0.5 wavelengths to permit the extraction of the S parameters from the standing wave on the microstrip line.

During the 3rd quarter, we also generated results and validated our PML truncation codes for the geometries provided by Compact. Specifically, we modeled a benchmark geometry consisting of a spiral inductor with a via and compared our computations for its frequency response with reference data. Our computations (with the PML used for mesh truncation) were in excellent agreement with the reference data, verifying our PML design and solvers. As expected, most of the time was spent in developing the geometry for the spiral inductor.

4- Continuation of Solver Studies (4th quarter)

During the 4th quarter, we returned to the solver study. After consultation with the Sponsor, we limited our concentration to the GMRES iterative solver and the VSS sparse LU solver. The latter is a solver recently introduced by Dr. Olaf Storaasli at NASA Langley (o.o.storaasli@larc.nasa.gov) and Majde Buddourah (LBL) . This solver is continuously under revision and should be noted that the results in this report are based on the VSS 1.3 version. The solver is only available to U.S. organization and cannot be transmitted to third parties. We found the performance of this solver quite impressive. We were able to solve 60,000 unknown sparse FEM systems in 10 minutes of so. For a 20,000 unknown system, the VSS solver was at least 3 times faster than standard iterative solvers.

Our greatest effort during the 4th quarter was devoted to optimization, preconditioning and case studies for the GMRES solver. Various preconditioning schemes were implemented, including an approximate inverse preconditioner. These preconditioners allowed for much faster convergence and removed uncertainty with the 'restart' number associated with all GMRES algorithms. We generated results which showed that for microwave circuit structures, the preconditioned GMRES algorithms converges with the number of restarts equal to only 1% of the total unknown count. This leads to substantial memory and CPU savings. Specific account of our results are given in this report.

Abstract

The superior absorption characteristics of the developed uniaxial perfectly matched layer (PML) absorbers increases their utilization as a robust, efficient and reliable mechanism for truncating finite elment meshes. Because of their anisotropic characteristics, their implementation becomes equivalent to imposing active elements inside the main mesh. Thus, the system condition is deteriorated and the overall convergence scheme will be affected. Also, because of their implementation as part of the main, the size of FEM system increases. Therefore, in this work, we focus on the convergence characteristics for FEM systems terminated by the ficticious PML absorbers. Starting by the PML parameters, we attempted to optimize the loss and phase shift parameters to speed up convergence while keeping the desired levels of absorption. Resulting linear systems are solved using iterative solvers and because of the ill-conditioning nature of such systems, convergence rate becomes slow and in some cases convergence may not be achieved at all. Therefore, our work was basically based on two aspects, the first is to develop and test different kinds of preconditioners and propose one or two of them. The latter involves the selection of a robust iterative solver. The preconditioned iterative solver should be able to achieve several goals. Among them, it must a convergence even with ill-conditioned systems. Also, it must be independent of any outside parameter or case specific. After dealing with several types of solvers, we concluded the Generalized Minimal Residual (GMRES) family of solvers can be tailored and optimized to achieve these features. The preconditioned GMRES solver was applied and tested for several systems. Numerical examples representing actual circuits and systems are considered at the end of the report.

Contents

1   Introduction

When Introduced by Sacks et al [17], the perfectly matched layer (PML) absorbers were considered a novel, efficient and reliable way for terminating finite element method (FEM) computational domains. Their excellent absorption characteristics for waves incident at different angles and frequencies resulted in employing them extensively. In addition to their superior performance, they offer the ease of implementation because of their inclusion as part of the main mesh. Also, they avoid the use of any boundary derivatives. Due to their outstanding absorption features, the computational domain can be enclosed by a perfect electric conductor (PEC) without affecting the accuracy of the calculations. Unlike the absorbing boundary conditions (ABCs), the PML termination scheme does not require any *a priori* knowledge about the propagation constants in all parts of the system. Moreover, PML termination scheme facilitates deembedding and parameter extraction. Based on all these advantages and due to the huge number of applications that use the PML for mesh truncation, we started to look at their implementation aspects to recommend or suggest an optimal design for these ficticious layers.

In the first stage of the project, we summarized the PML theory and proceed by applying it to an extensive number of the applications. These applications included microstrip lines, filters and scattering problems. Because of the anisotropy nature of the PML, the system condition is deteriorated and this slow convergence of iterative solvers significantly. This encouraged us to study the solver convergence to speed up the convergence rate. We briefly studied the convergence as a function of the PML parameters to find their range that achieve faster convergence. After this study, we examined the effect of simple preconditioners on the convergence and three different types of solvers were studied and tested. Then, we carried out and implemented

examples representing actual circuits and filters to validate our proposed approach. Finally, in this quarter, we implemented an efficient preconditioner in conjunction with the GMRES solver.

In the next following three sections, we summarize the work during the first three quarters. We will quickly review the major contributions by presenting some of the results we generated.

2 PML Theory and Implementation

In the first quarter, we utilized the PML for terminating finite element meshes. Based on a simple one-dimensional study,the PML design curves was generated as shown in figure 1. Using these design curves, PML, we proceeded by applying the PML for a given set of three dimensional applications. This set includes, radiation problem, filters and scattering applications. Referring to Figure 1, the several parameters of the PML that achieve a desired dB absorption can be directly extracted from this set of figures. These parameters include the loss value, layer thickness and the number of PML layers.

3 Convergence Study on FEM Systems with PML Implementation

In the second quarter of the project, we started to look closer to the PML implementation. The PML implementation deteriorates the solver convergence and thus the CPU time increases dramatically. For any FEM system, the solver consumes at least 90% of the execution time and as a result significant increases in the CPU time are observed. This problem affects badly problems of moderate and big sizes. Our

strategy to address this issue involved three different steps

1. For a certain number of circuits, we tried to optimally select the PML parameters required for fast convergence concurrently with high absorption characteristics.

2. After getting these values (ranges), we proceeded by preconditioning the resulting linear system. The ideal preconditioner should:

   - work for all the iterative solvers.

   - achieve significant improvements for all the systems.

   - have trivial computational effort and memory requirements.

   - not affect the parallelization.

3. Finally, we reached the stage of selecting the iterative solver suitable for a given linear system. For each kind of solvers we tried to examine its:

   - robustness.

   - speed.

   - memory requirement.

   - convergence characteristics.

3.1   The Effect of the PML Parameters

We first started to discuss the effect of the PML material parameters on the convergence. It was found that when α (phase parameter) and β (loss parameter) becomes close to unity, both convergence and absorption were optimized. This conclusion was based on microstrip and patch antenna geometries. The design curves for α and β

are indicated in Figures 2 and 3. The physical interpretation behind these ranges for α and β is as follows:

1. For the case where α is less than unity, the system will be highly ill-conditioned due to the presence of active elements inside the main mesh. Thus, we expect slower convergence. On the other hand, when α is equal or larger than unity, the condition of the system is improved and hence convergence becomes faster.

2. For β, low values of this parameter leads to lower dB absorption and higher values will lead to large field variations between the adjacent elements in the FEM mesh. Therefore, the solver ability to track the solution will be reduced.

3.2   Diagonal Preconditioning and Solvers

Regarding the preconditioning of the FEM matrix, we tested the Diagonal Preconditioner (DPC) and found that it reduces the overall CPU time by a percentage ranges from 30% to 60%. An example is indicated in figure 4 where the number of iterations (and consequently the CPU time) is reduced by a factor of two when DPC was invoked. A more complicated and rebust preconditioner will be discussed later in the solvers study.

Also, in the second quarter, we began to look at different solvers by implementing and testing three different kinds of solvers. The first is the BiConjugate gradient and it has the ease of implementation feature plus a low CPU and memory costs. However, it lacks the robustness and does not guarantee convergence. Also, it has irregular convergence characteristics with two breakdown possibilities. In addition to that, for ill-conditioned systems (as in the PML case), it may not converge at all. Therefore, with large and badly ill-conditioned systems, it is not recommended. The Quasi Minimal Residue (QMR) solver has better convergence features and lower

breakdown or divergence possibilities. For a fixed system, both BCG and QMR converge nearly in the same number of iterations but QMR has better error history. On the other hand, the Generalized Minimal Residual (GMRES) solver is the most robust solver since it guarantees convergence for all the systems and it leads to the smallest residual for a fixed number of iterations. Figure 5 shows the outstanding convergence scheme of the GMRES over the BCG and QMR for a typical standard problem of a microstrip line terminated by the PML.

4 PML Applications

In the third quarter, after obtaining some guidelines for the implementation of the PML and solvers, we started to test actual circuits representing more complicated structures. Because of the numerical uncertainty associated with the parameters extraction process, we had to address the feeding issues for different circuits and structures. We found that the minimum number of feeding (excitation) probes required for stable parameter calculations was on the order from 3 to 5. With this number of probes, we were able to achieve stable and converging S parameters. Then, we studied the coupling between two parallel transmission lines printed on a planar substrate. We obtained the coupling amount as a function of the operating (excitation) frequency. Finally, with our computational model, we matched the measured data for a spiral inductor with an air bridge. The spiral geometry was discretized accurately and applied to our FEM code that has:

- The PML as a mesh truncation scheme.

- Diagonal preconditioning.

- GMRES solver with 40 restarts.

The geometry and the dimensions are shown in Figure ?? and the results of our FEM simulator and the measured data supplied to us are illustrated in Figure 8. As displayed in this figure, excellent agreement exists between the computed and measured data.

5 GMRES Solver Studies and Preconditioners

In this quarter (4th quarter), we focused on three different issues related to the convergence study. They are:

1- The GMRES as a robust, efficient and reliable solver.

2- Simple and complicated preconditioners and their effects on the solvers especially on the GMRES.

3- The proposed Approximate Inverse Preconditioner (AIPC) combined with the GMRES solver.

5.1 GMRES Solver

A detailed careful study of the GMRES was carried out and several systems with different sizes and conditions were implemented and tested The GMRES was chosen because of the following reasons

1. GMRES solver is robust solver with regular convergence characteristics.

2. It guarantees convergence even for ill-conditioned system.

3. It leads to the smallest error among all the solvers for a fixed number of iterations.

4. It allows for optimization by changing the number of restarts, preconditioning scheme or by using different versions and the GMRES techniques.

A detailed careful study of the GMRES was carried out and several systems with different sizes and conditions were implemented and tested and the results will be discussed later after summarizing the preconditioners.

5.2   Preconditioners

Preconditioners are used in conjunction with iterative solvers to improve the condition of the system and hence achieve faster convergence. As indicated before, large and ill-conditioned systems may not converge at all without preconditioning. Moreover, with the PML implemented, there is a huge deterioration in the system condition and thus robust and efficient preconditioners are needed. There are enormous types of preconditioners with different CPU and memory costs. We will implement and test two types of these preconditioners, the diagonal and the approximate inverse ones. The first is the Diagonal Preconditioner (DPC) and it simply scales the element of the FEM matrix using its diagonal values. Thus, it leads to substantial improvements in the overall condition number. The main features of DPC are:

1. It is easy to implement (diagonal scaling of the matrix elments).

2. It is implemented with no computational effort or cost.

3. Achieves significant improvement with most of the systems (This was shown before).

4. This kind of preconditioning techniques works for all kinds of solvers and nearly if affects them in the same way.

5. Diagonal Preconditioning does not affect the parallelization of the code.

On the other hand, the approximate inverse preconditioner (AIPC) is more complicated and it involves the evaluation of an approximate inverse to the FEM matrix. As a result, ill-conditioned systems will be directly mapped to well conditioned ones with dramatic improvements in the condition. It is obvious that the AIPC will need more computational effort than the DPC but it will eventually eliminate any convergence problems even for indefinite systems. The AIPC algorithm tries to find the matrix M that minimizes the frobenius norm of the residue R given by

$$R = I - AM \tag{1}$$

where I is the identity matrix, A is the FEM matrix and M is the right approximate inverse of A.

The norm minimization can be done by one of the minimal residual methods such as the Global Steepest Descent (GSD) [23] or the column oriented algorithms. The GSD type of techniques needs the storage of an extra matrix with the same size as A plus it performs two matrix matrix products per iteration and that increases dramatically the computational costs. The column minimization algorithm depends on the minimization of the individual columns of the residual R and this will be done via any of the known methods such as the Minimal Residual (MR)or the Steepest Descent (SD). The MR algorithm will be utilized in our simulations [23]. In the following section, we will present the effects of DPC and AIPC on the convergence of the GMRES and from the results, we will be able to obtain the main features of the AIPC.

As a conclusion, the AIPC has the following features

1. It achieves dramatic improvements in the system condition and this leads to significant improvements in the convergence rate. In many cases, the total number of iterations dropped by large factors (may be 50 or more).

2. This preconditioning scheme can be used independent of the system type or condition. The worst ill-conditioned systems can be preconditioned using the AIPC at no extra cost (because the preconditioning is done once before starting the iterative solver).

3. It does not affect the parallelization.

5.3 Preconditioned GMRES

According to the previous results, combining the AIPC with the GMRES solver may have huge impact on the convergence. Thus, we had implemented and tested an ill-conditioned system representing a microstrip line terminated by the PML. The size of this system was approximately 7000 unknowns. We will apply this system to the following three solvers:

- GMRES with no preconditioning.

- GMRES with DPC.

- GMRES with AIPC.

In each case, we compute the total CPU time as a function of the number of restarts (m). The results are shown in Figure 6 and from them, we can deduce the following:

1. Without preconditioning the CPU time was high for any m. Moreover, it is very difficult to predict m that leads to minimal CPU times. This is due to the fact that the optimal selection of m depends on the geometry, parameters, sampling rate, basis functions, problem size,...etc.

2. With the DPC applied, the CPU times become lower and the dependence on m was reduced.

3. With the application of the AIPC, the CPU time values were minimized and the dependence on mwas greatly removed.

Finally, our recommendations for robust, efficient solvers are:

a- For any system Ax=b, try to find a matrix M that is a good approximation for the inverse of A. This can be achieved by several methods, one efficient way is the minimal residual algorithm.

b- Use this matrix to precondition the GMRES code from the right or the left. Then, proceed by applying the normal GMRES iterations.

6 Appendix I: Code Manual

The following notes are helpful in running the FEM codes we are enclosing. We start by giving some details about the main file **fem_compact.m** that generates the FEM system. This system will be solved by one of the attached iterative solvers. It also includes the input geometry file as well as the layers (materials) inside the cavity. In addition to that, it has the other external parameters such as the absorber used to terminate the FEM mesh (isotropic or anisotropic) and the operating frequency.

6.1 The FEM main code

In this section, we present a simple introduction to the main FEM code utilized in generating the FEM system. this code dumps the overall FEM matrix A as well as the excitation vector b in addition to some geometrical data needed for field calculations. The name of this file is **fem_compact.m**.

- The file **fem_compact.m** generates the FEM matrix based in the geometry parameters given at the top lines of the code and these parameters are:

  1. **lx,ly,lz** are the cavity dimensions in x,y and z directions respectively. It should be noted that all of these values are given in cm.

  2. **nxe, nye and nze** indicate the sampling rates in the x,y and z directions respectively. Note that the longitudinal direction is along the y-direction.

  3. **xst, yst and zst** give the starting segments of the non-conducting material filling in the corresponding direction of the cavity. The length (number of elements) in each gives the number of the non-metallic layers inside the cavity.

4. **xend, yend and zend** are the ending segments of the same layers previously described.

5. **xstc** is a vector containing the starting segments of the metallic (pec) patches parallel to the x-direction.

6. **ystc** is a vector containing the starting segments of the metallic (pec) patches parallel to the y-direction.

7. **xendc** is a vector containing the end segments of the metallic (pec) patches parallel to the x-direction.

8. **yendc** is a vector containing the end segments of the metallic (pec) patches parallel to the y-direction. It should be noted that the length of any of **xstc, ystc, xendc, and yendc** gives directly the number of metallic (conductor) patches existing inside the cavity.

9. **ep** and **mu** are the complex diagonal elements of the permittivity and permeability of the filler or absorber. The length of each should be equal three times that of **xst or yst**. This is because each layer of the filling has three diagonal parameters.

10. **Jy** is the feeding currents in y direction (complex in general). Note that the feeding is put along a certain segment in the y-direction.

11. **xf, yf** specify the node location of the feeding probes.

12. **hc** gives the node location of the height of the conducting patches. The length of this vector equals the number of patches inside the cavity.

13. **zf** is the node location of the feeding probes along the z-direction.

14. **freq** is the operating frequency in Hz.

15. α **and** β are the PML real and imaginary part respectively.

- As indicated, the input file specifies the permittivity and permeability tensors for all the layers inside the cavity and for the included example, these layers are as follows:

    1. The substrate which is defined by the first three elements of the vectors **ep and mu** represent the diagonal part of both permittivity and permeability tensors respectively. It should be noted that for isotropic substrates, these three elements must be equal.

       For example, in the attached input file, the substrate is isotropic with relative permittivity of 3.2 and unity permeability .

    2. The cavity filling (between the substrate and the walls) is represented by the next three elements in both **ep and mu** vectors. Usually, this layer of filling is considered to be air.

    3. The PML absorber termination for the substrate layer is given in elements 7 to 9 of both tensors. If the isotropic termination is desired, the two lines of the PML termination should be commented out by removing the "%".

- The FEM code **fem_compact.m** will produce the file **data_compact.mat** which contains the following:

    1. **rvec** is a vector specifying the row indices of the non-zero elements of the overall FEM matrix. That is, rvec gives the ith index of the matrix entry A(i,j).

    2. **cvec** is a vector specifying the corresponding column indices of the non-zero elements of the overall FEM matrix. That is, cvec gives the jth index of the matrix entry A(i,j).

3. **Cvec** is a vector containing the entries of the non-zero elements of the FEM matrix.

4. **b** is the complex Excitation Vector.

5. Example: If A is give by

$$
A = \begin{bmatrix}
15 & 0 & 20 & 0 \\
0 & 16 & 0 & 40 \\
0 & 0 & 12 & -4 \\
10 & 14 & 0 & -10
\end{bmatrix}
\tag{2}
$$

then

rvec={1,1,2,2,3,3,4,4,4}

cvec={1,3,2,4,3,4,1,2,4}

Cvec={15,20,16,40,12,-4,10,14,-10}

6. **SS3** is a vector containing the numbers "Global numbers" of the non-metallic edges.

7. **edge1** is a matrix containing the global edge numbering with zeros in the locations of the metallic elements. For example, if the element **edge1(i,j)=k**, this means that the global number of the jth edge of the ith element is equal to k.

8. **fel** is a vector containing the element numbers which have the feed probes and **b7** is a vector containing the global numbers of the feed edges.

6.2   Iterative Solvers

- We include three kinds of solvers, the BCG, QMR and GMRES and each of them will produce the following data files:

1. BCG: It dumps the following in the file **out_compact_bcg.mat**:

   x — the solution vector for the system Ax=b.

   ebcg — the error history.

   error — error at the last iteration.

   iter — no. of iterations before convergence.

   flag — indication if convergence was achieved.

   nfbcg — the total no. of flops.

   E2 — the field under the microstrip line.

   timebcg — total CPU time.

2. QMR: It dumps the following in the file **out_compact_qmr.mat** :

   x — the solution vector for the system Ax=b.

   eqmr — the error history.

   error — error at the last iteration.

   iter — no. of iterations before convergence.

   flag — indication if convergence was achieved.

   nfqmr —- the total no. of flops.

   E2 — the field under the microstrip line.

   timeqmr — total CPU time.

3. GMRES: It dumps the following in the file **out_compact_qmr.mat** :

   x — the solution vector for the system Ax=b.

   egmres — the error history.

   error — error at the last iteration.

   iter — no. of iterations before convergence.

   flag — indication if convergence was achieved.

nfgmres —— the total no. of flops.

E2 —— the field under the microstrip line.

timegmres — total CPU time.

6.3   List of Files

We are enclosing the following matlab files:

1. **fem_compact.m**: generates the FEM system and has the option of terminating the mesh by either **isotropic or anisotropic (PML)** schemes.

2. **fem_compact.mexhp7**: the matlab executable (binary) file which can be run directly from the matlab environment. It can be generated as follows:

   - Make all the necessary changes in the main file **fem_compact.m**. These changes may be in the frequency, geometry or termination scheme.

   - From matlab screen, type **mcc -l fem_compact.m** and then two files will be directly generated: the stand alone C code which name is **fem_compact.c** and the matlab executable (mex) file **fem_compact.mexhp7**. The latter can be run from matlab.

   - From the matlab screen, type **fem_compact** to run the code and generate the output data file **data_compact.mat**.

   - The solvers are run directly from the matlab prompt without compilation by typing the following:

     (a)- **[x, error, iter, flag,ebcg,nfbcg,E2,timebcg] =bicg_compact** to run the BCG. solver.

(b)- [x, error, egmres,nfgmres ,iter,E2,timegmres]= gmres_compact to run GMRES solver.

(c)- [x, error, eqmr,nfqmr,E2,timeqmr,iter] = qmr_compact to run the QMR solver.

3. **bicg_compact.m** : BCG solver.

4. **gmres_compact.m**: the GMRES solver.

5. **qmr_compact.m**: QMR solver.

6. **rotmat.m**: file needed when running the GMRES.

7. **data_compact.mat**, FEM data generated by the files **fem_compact.mexhp7** or **fem_compact.m**.

8. **out_compact_bcg.mat**: output data file generated by BCG solver.

9. **out_compact_qmr.mat**: output data file generated by the QMR solver.

10. **out_compact_gmres.mat**: output data file generated by the GMRES solver.

11. **readme.tex, readme.ps, printrawps.ndvips.tex and macros**: postscript and latex file documentation of this manual.

6.4 Validation Results

- For checking the codes performance, the normalized values of the electric field under the microstrip line should be as follows:

```
E2=
-0.8618 - 0.0926i
```

-0.7120 + 0.3301i

-0.4555 + 0.6304i

-0.1281 + 0.7979i

 0.2239 + 0.8244i

 0.5513 + 0.7146i

 0.8084 + 0.4890i

 0.9589 + 0.1828i

 0.9818 - 0.1590i

 0.8734 - 0.4871i

 0.6480 - 0.7545i

 0.3365 - 0.9230i

-0.0184 - 0.9686i

-0.3682 - 0.8840i

-0.6648 - 0.6802i

-0.8678 - 0.3851i

-0.9494 - 0.0387i

-0.8984 + 0.3115i

-0.7219 + 0.6173i

-0.4446 + 0.8368i

-0.1046 + 0.9400i

 0.2511 + 0.9125i

 0.5734 + 0.7582i

 0.8178 + 0.4986i

 0.9504 + 0.1691i

 0.9530 - 0.1849i

 0.8253 - 0.5143i

```
 0.5848 - 0.7740i

 0.2648 - 0.9284i

-0.0905 - 0.9561i

-0.4324 - 0.8535i

-0.7135 - 0.6350i

-0.8950 - 0.3311i

-0.9517 + 0.0163i

-0.8754 + 0.3587i

-0.6764 + 0.6487i

-0.3810 + 0.8457i

-0.0289 + 0.9210i

 0.2182 + 0.5779i

 0.2837 + 0.2980i

 0.2548 + 0.1022i

 0.1920 - 0.0182i

 0.1333 - 0.0813i
```

7   Appendix II: Answer to Questions

In this section, we answer the questions received during the forth quarter. It should
be noted that the main objective of the project during the first year was to charac-
terize the PML performance, obtain the optimal PML parameters and improve the
convergence scheme for the systems terminated by this kind of anisotropic absorbers.

Below are the questions and their answers.

Q1) We have not found where the S parameter output is.

A1) In all of the applications we studied, we used the following definition of the S

parameters

$$S_{jk} = v_{rj}/v_{ik} \tag{3}$$

where v_{rj} is the reflected voltage at the jth port (location) due to the incident voltage v_{ik} at port k. Because of the planner nature of our circuits, equation 3 can be written in terms of the electric field values. Thus,

$$S_{jk} = E_{rj}/E_{ik} \tag{4}$$

In the third report, we presented a detailed deembedding scheme for extracting the incident and reflected fields after solving the finite element system. Therefore, we can arbitrarily define the ports and put the excitation anywhere inside the mesh. Then, the linear system is solved and the total (incident plus reflected) field values are obtained everywhere inside the mesh. Then, using equations 3 and 4, the scattering parameters can be obtained. In the examples used to characterize the PML performance, we defined two main ports, the input and the output. The input port was the one where we put the excitation probes while the output was the PML matched one. According to the microstrip example, the S_{11} parameter is the reflection coefficient at the input port. Because of the quasi TEM nature of the propagating mechanism in this case, we have only one mode and to extract its S_{11}, we selected three adjacent points at the input port and then computed the incident and reflected fields (as indicated in the third report). Then, we obtained S_{11} from equation 4. Regarding the coupling example, we excite one port, terminate two by the PML and solve the whole system. Again, using the deembedding scheme given in the third report, we got the fields at all the locations and hence calculate the incident and reflected fields at the excited and parasitic ports to calculate the scattering parameters between the two ports. This parameter gives the amount of coupling between the two ports.

Q2) The physical meaning of the field under strips is difficult to understand. What is the quantity being plotted?

A2) This quantity represents the values of the total vertical electric field E_z along the microstrip line. The standing wave pattern obtained when plotting this quantity examines the performance of the absorber. If the absorber is perfectly matched, the field values along the line should be constant (flat). On the other hand, if this absorber performance is poor, high fluctuations will be observed. Cutting a section on the microstrip line, we can get the total field at a specific location (segment) and if we decompose this field to the incident and reflected values, we will be able to extract the reflection coefficient at this section of the transmission line. In addition to this, based on the field equivalence principle, these values are proportional to the current on the microstrip line which is difficult to be extracted directly.

Q3) Provision of FEM formulation of the PML involved, the treatment on surface integral and detailed feeding treatment.

i)- In the first quarter report of UofM, the formulation for the matrix element of volume type has been provided (see page 5, 22-25 of their first report). However, the matrix element of the surface type, which is related to the incident field at the incident plane (their Sin in formula (1), page 5, first report) and the treatment on Sout if it is otherwise not a PEC.

Ai)- Because of the high absorption nature of the PML, we can use it without affecting the accuracy of the calculations and this allows us to enclose the PML with a PEC layer. Thus, the surface integral vanishes.

ii)- In the third quarter report of UofM, on page 12, the coupling between two microstrip lines has been studied. However the formulation and how to extract S parameters for this structure for this study has not been reported so far.

Aii)- As indicated in the answer of question one, the excited port is chosen as port number k and the parasitic one is as port number j, after solving the FEM system, we get the fields (total fields) E_j and E_k, we use the deembedding technique provided in the third report to extract the incident and reflected fields in these ports and by utilizing equations 3 and 4, the S_{jk} parameter is achieved.

Q4) Provision of the derivation for formulas (3) and (4), pages 5 and 6, Q1 report, and the proof for conclusion that "By choosing a2 = b2 and c2 = 1/b2, it follows that R(te) = R(tm) = 0 for ALL INCIDENCE ANGLES, .." on page 6 of their first quarter report.

A4) Answers to this question are directly found in the following reference:

Z.S. Sacks, D.M. Kingsland, R. Lee and J.F. Lee, "A perfectly matched anisotropic absorber for use as an absorbing boundary condition," *IEEE Trans. Antennas and Propagation*, December 1994.

Q5) Information on how to input a test structure into the simulator.

A5) The code manual attached to this report includes a full description of the input files. It contains the geometry files, the absorber (isotropic or anisotropic) parameters, feed specifications, sampling requirements,...etc.

Q6) The CVSS solver code and documentation.

A6) The CVSS solver is included in this report.

Q7) Final simulator code and reports.

A7) We sent the codes to CSI last June with the manuals.

References

[1] Y.Y. Botros, S.R. Legault, J. Gong, J.L. Volakis and T.B.A. Senior, "Perfectly Matched Absorbers (PML):Theory, Analysis and Implementation Radiation Laboratory Report (No. 034702-1), The university of Michigan, Ann Arbor, October 1996.

[2] Y.Y. Botros and J.L. Volakis "Convergence Study on FEM Systems with PML Implementation" Radiation Laboratory Report (No. 034702-2), The University of Michigan, Ann Arbor, March 1997.

[3] B. Engquist and A. Majda, "Absorbing boundary conditions for the numerical simulation of waves," *Math. Comput.* Vol. 31, pp. 629–651, 1977

[4] L.Halpern and L.N. Trefethen, "Wide–angle one-way wave equations," *J. Acoust. Soc. Amer.* Vol. 84, pp. 1397–1404, 1988

[5] W. Sun and C.A. Balanis, "Vecter one–way wave absorbing boundary conditions for FEM applications," *IEEE Trans. Antennas Propagat.* Vol. AP–42, pp. 872–878, 1994

[6] Webb and Kanellopoulos, " Absorbing boundary conditions for the finite element solution of the vector wave equation," *Microwave Opt. Tech. Lett.* Vol. 2, pp. 370–372, 1989

[7] A. Chatterjee and J.L.Volakis, "Conformal absorbing boundary conditions for the vector wave equation," *Microwave Opt. Tech. Lett.* Vol. 6, pp. 886–889,1993

[8] R.L. Higdon, "Absorbing boundary conditions for acoustic and elastic waves in stratified media," *J. Comp. Phys.* Vol. 101, pp. 386–418, 1992

[9] Z.P. Liao H.L. Wong, B.P. Yang and Y.F. Yuan, "A transmitting boundary for transient wave analysis," *Sicentia Sinica* Vol. 28, pp. 1063–1076, 1984

[10] M. Moghaddam and W.C. Chew, "Stabilizing Liao's absorbing boundary conditions using single–precision arithmetic," it IEEE AP-S Int. Symp., London, Canada, pp. 430–433, 1991

[11] J.Fang, "ABCs applied to model wave propagation in Microwave integrated–circuits," *IEEE Trans. MTT*, Vol. 42, No. 8, pp. 1506–1513, Aug. 1994

[12] R. Luebbers and C. Penney, "Scattering from apertures in infinite ground planes using FDTD," *IEEE Trans. Antennas Propagat.* Vol. 42, pp. 731–735, May 1994

[13] B. Stupfel and R. Mittra, "Efficiency of numerical absorbing boundary conditions for finite element applications," *URSI Radio Science Meeting*, pp. 165, 1994

[14] K.K. Mei, R. Pous, Z. Chen, Y.W. Liu and M.D. Prouty, " Measured equation of Invariance: A new concept in field computations," *IEEE Trans. Antennas Propagat.* Vol. 42, pp. 320–328, March 1994

[15] T. Özdemir and J.L.Volakis, "A comparative study of an absorbing boundary condition and an artificial absorber for truncating finite element meshes," *Radio Science* Vol. 29, No. 5, pp. 1255–1263, Sept. 1994

[16] J.-S. Wang and R. Mittra, "Finite element analysis of MMIC structures and electronic packages using absorbing boundary conditions," *IEEE Trans. Microwave Theory and Techn.*, Vol. 42, pp. 441-449, March 1994.

[17] Z.S. Sacks, D.M. Kingsland, R. Lee and J.F. Lee, "A perfectly matched anisotropic absorber for use as an absorbing boundary condition," *IEEE Trans. Antennas and Propagation*, December 1994.

[18] Berenger, J.P. "A Perfectly Matched Layer for the Absorption of Electromagnetic Waves," *J. Comp. Physics*, 114 : 185-200, 1994.

[19] Chew, W.C. and Weedon, W.H. "A 3-D Perfectly Matched Medium from Modified Maxwell's Equations with Stretched Coordinates," *Microwave and Optical Technology Letters*, p.599-604. September, 1994.

[20] Katz, D.S, Thiele, E.T., and Taflove, A. "Validation and Extension to Three dimensions of the Berenger PML Absorbing Boundary Condition for FD-TD Meshes," *IEEE Microwave and Guided Wave Letters*, August, 1994, p.268-270.

[21] C. Rappaport and L. Bahrmasel, "An absorbing boundary condition based on anechoic absorber for EM scattering computation," *J. Electromagn. Waves Appl.*, Vol. 6, No. 12, pp. 1621-1634, Dec. 1992.

[22] R. R. Bonetti and P. Tissi, "Analysis of Planar Disk Networks", IEEE MTT-26, July 1978, pp. 471-477.

[23] Y. Saad *Iterative Methods for Sparse Linear System,* PWS publishing co., 1996.
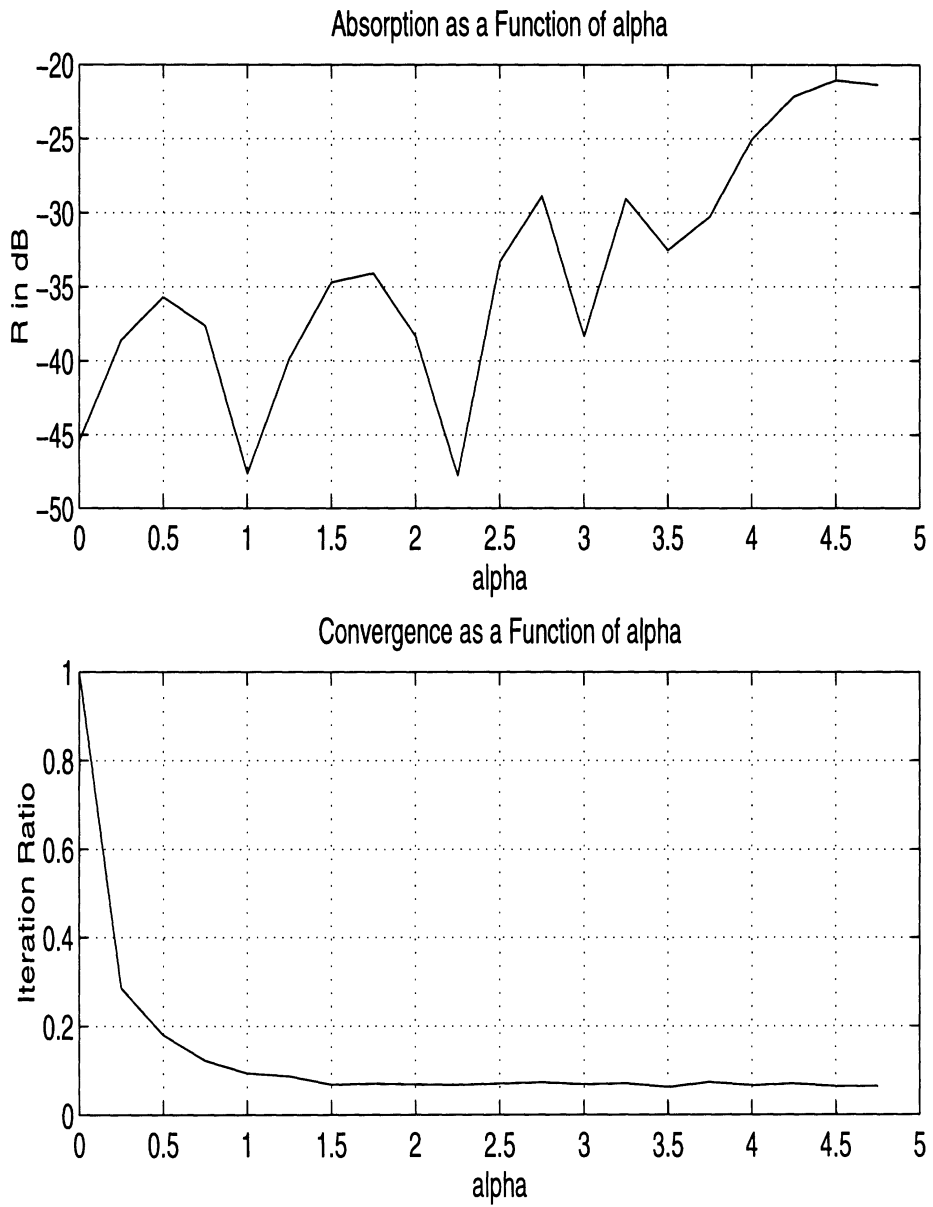
List of Figures

Figure 1: PML Design Curves

Figure 2: Effects of α on the Convergence and Absorption
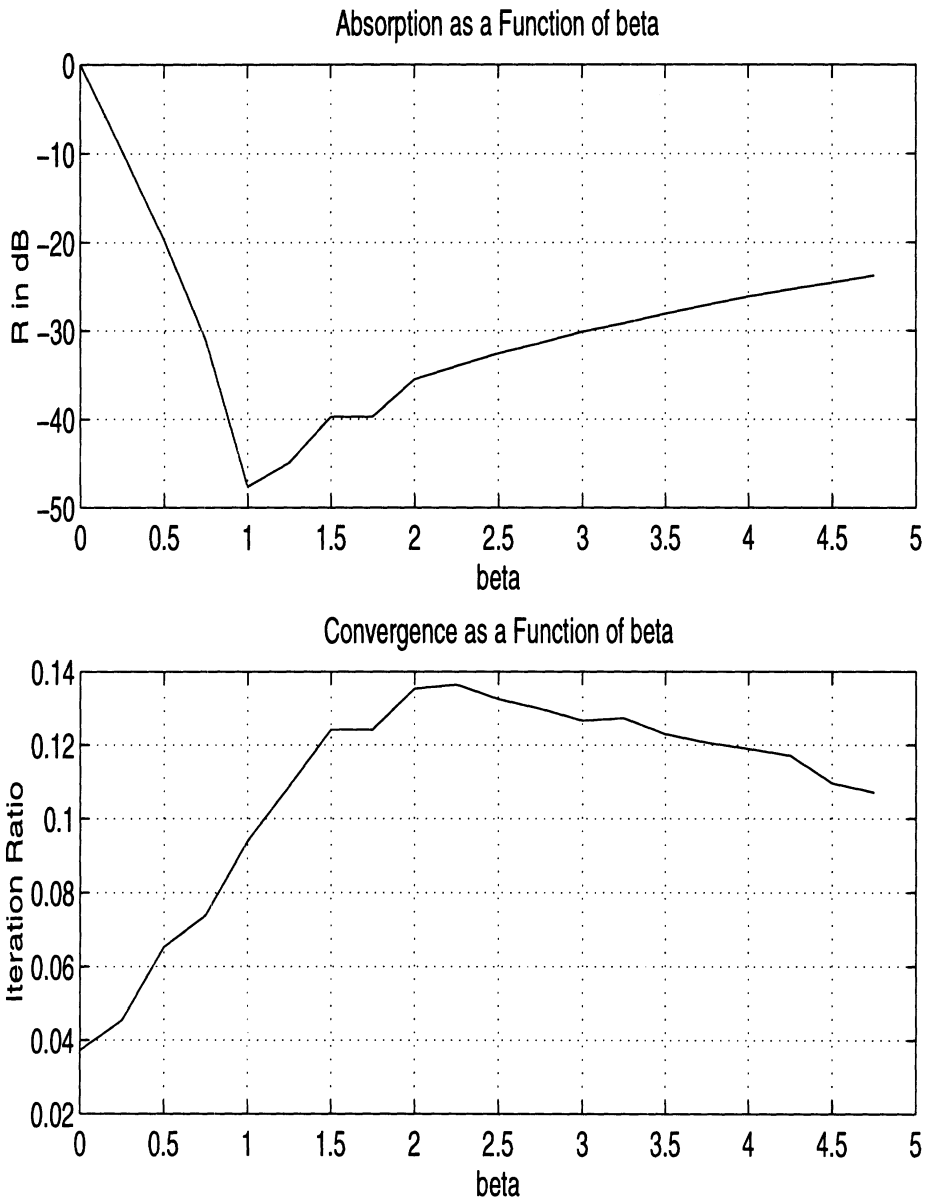
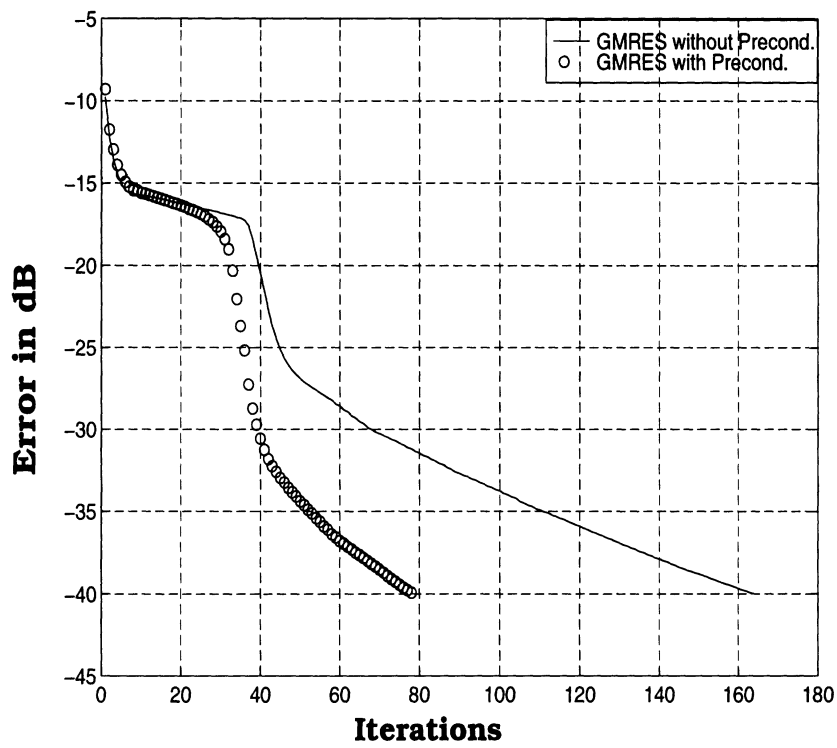Figure 3: Effects of β on the Convergence and Absorption
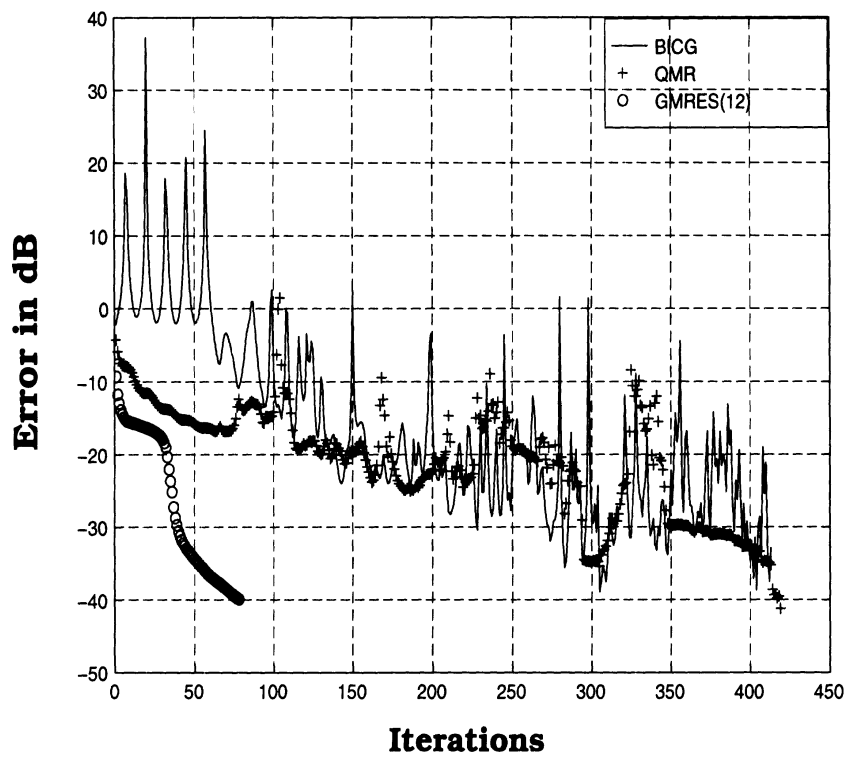
Figure 4: Effect of Diagonal Preconditioning on the GMRES

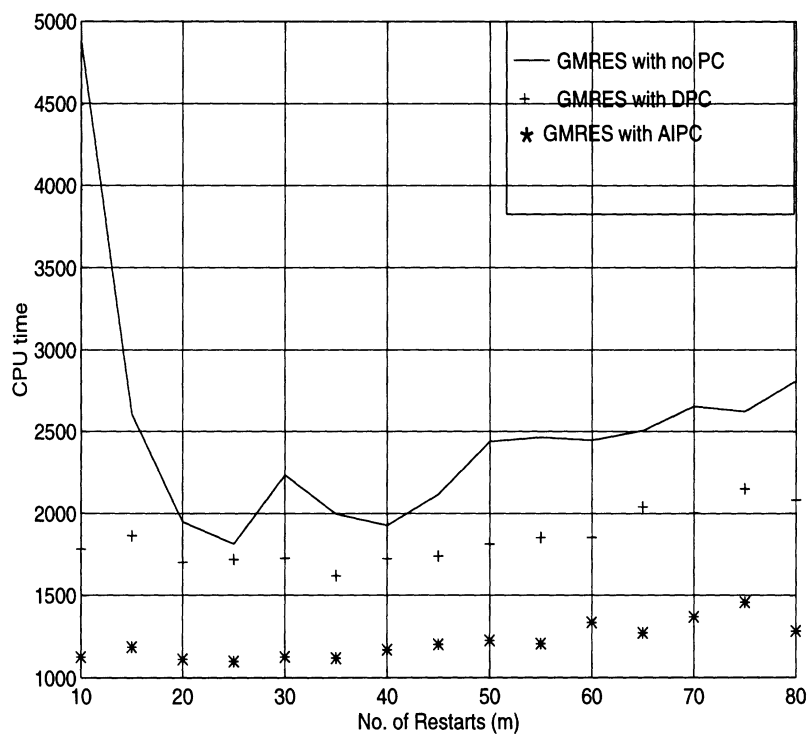Figure 5: Convergence Characteristics of Different Solvers

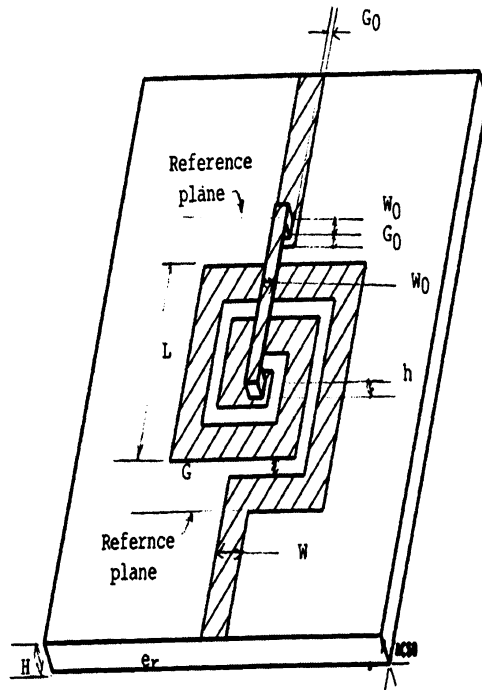Figure 6: Effect of the Preconditioning on the GMRES Solver

DIMENSION:

L = 3.4375 mm
G0 = 0.15625 mm
W_0 = 0.3125 mm
W = 0.625 mm
G = 0.3125 mm
H = 0.635 mm
h = 0.3125 mm
e_r = 9.8

Figure 7: Geometry of the Spiral with an Air Bridge
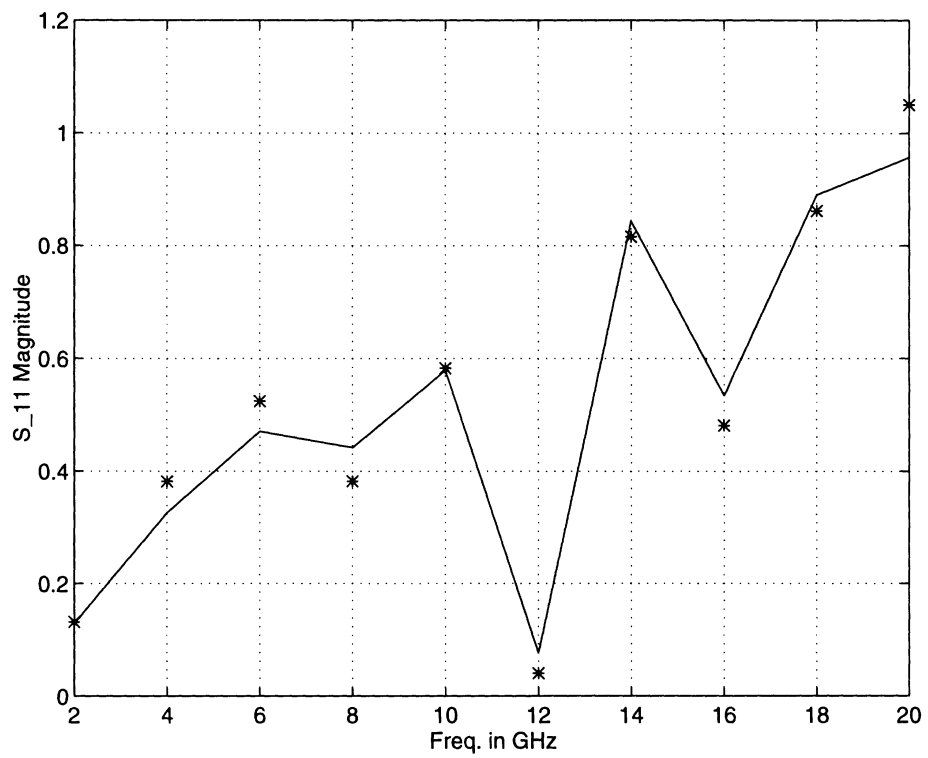
Figure 8: Comparison Between the Measured and Computed data for the Spiral Antenna

========== CVSS ============

—— Binary Release ——

```
---For UNIX Workstations
---For Electromagnetics FEM
```

```
JIAN GONG, Ph.D.

Radiation Laboratory
EECS Dept.
University of Michigan
Ann Arbor, MI 48109-2122
```

---

1 About CVSS

CVSS is a sparse system solver for numerical linear system of equations like

$$AX = b$$

and it was developed at NASA by Dr. OLAF O. STORAASLI and his group. This solver belongs to a direct approach as opposed to iterative methods, such as CG-type techniques. Direct solver has been very attractive since

limited user interface is required. Moreover it is usually considered problem independent.

However, the direct solver in general requires factorization (or LU decomposition) which consumes a significant computation effort. As compared to iterative techniques, the LU decomposition needs the floating point operations (FPO's) of order $\mathcal{O}(N^3)$, where N is the dimension of the linear system of equations. For a large N, it apparently becomes impractical to employ direct solution. This is why the iterative solver has been extensively explored especially in academic research. (Note that the FPO's for iterative techniques is proportional to $\mathcal{O}(N^2)$, instead. One order of magnitude reduction in computation with respect to the LU decomposition!)

For special numerical systems such as finite difference and finite element analysis, the sparsity nature can however be used to significantly alleviate the computational burden. This is exactly what CVSS does. The major attraibutes of the current CVSS development include

(1) Low memory requirement

(2) Fast solution time

(3) Readily parallelized/vectorized

(4) Capable to handle structural (real) and EM (complex) systems

(5) Tested on multi-platforms (supercomputers, workstations and PC's)

2 CVSS package

The CVSS package is written in Fortran 77 with multiple routines/functions arranged in over 50 files. Since sophisticated dynamic memory allocation (and out-of-core: not ready yet at the moment of this document preparation) technique has to be implemented to limit the redundant memory need, certain functions of system calls have been used. This makes the compilation and testing difficult on multi-platforms. (Note that the system functions are platform-dependent.) It is therefore non-trivial to generalize the routine calls available for all platforms.

The CVSS has been successfully compiled so far at the University of Michigan on SUN, HP, SGI using both vendor's compilers and the GNU compiler. More importantly, in certain cases, it is preferred to use C to

handle dynamic memory allocation, rather than using Fortran 77. This is because the latter is too system dependent and sometimes the performance may be deteriorated to certain degree. (No extensive testing on this has been carried out. Hope to see more evidence.)

3   Binary Executable and Library Release

The binary executable and library release available for UNIX workstations SUN, HP and SGI is intended for quick group usage. It is by no means optimized (the CVSS package itself is not optimizaed anyway!) in terms of function calls, I/O format, and convenience of interface with user's own packages. The installation will be soon ready on JLV's group machines at the University of Michigan. The location of those binaries accessible to the group will be determined soon. Any possible changes and updates will be timely posted to those having the access to the CVSS library.

The notation of the binary release is self-interpretable. Here are the file names for the executables:

```
cvss_sun
cvss_hp
cvss_sgi
```

The library file names are given by

```
cvss_sun_lib
cvss_hp_lib
cvss_sgi_lib
```

The size of the binary files ranges from 135 k to 191 k because of the difference of the compilers. The numerical system to be handled can be as large as 80,000 in dimension with 1.6 million non-zero entities.

4   How To Use CVSS Executables

The input/output has been made general and easy for interface. Here is how you should use the executables.

(1) The output of your FEM code should provide two files: amat.dat and bmat.dat, where amat.dat contains matrix A and bmat.dat stores the right hand side excitation vector.

(2) They should be all in text mode. (See To-do List later.)

(3) bmat.dat is one-column vector with ascending order of vector entries.

(4) amat.dat is of the type:

```
field1     field2     field3
```

in each row, where field1 and field2 are both integer numbers indicating the row and column indices (i,j), respectively, for the complex entry field3.

(5) It should be noted that the entries stored contain only the upper triangle AND diagonal terms of the matrix A.

(6) It is suggested to store (non-zero) matrix entries row by row. In specific, all entries in the first row should be put into amat.dat one by one and then the second row is scanned. The process is continued until all the non-zero entities are scanned.

Two additional numbers are needed for the user to key-in for interface. The first number is the numerical system dimension; the second one is the number of non-zero entries stored. Note that these two numbers coincides with the number of lines of the file bmat.dat and amat.dat, respectively. If your FEM code does not provide these numbers (which it should), do the following on UNIX:

```
cat bmat.dat | wc -l  ! ==> record the standard output as N1
cat amat.dat | wc -l  ! ==> record the standard output as N2
```

To run the code on (e.g.) HP in the directory where the files amat.dat and bmat.dat are located, type

```
cvss_hp
```

When the code prompts for key-in numbers, type

```
N1
N2
```

And wait... You should see a file called sol_cvss.dat storing your expected solution X. It is then your own responsibility to extract the needed data from sol_cvss.dat for post process or graphic display.

Enclosed in the binary version, you should see a sample of the following files:

```
amat.dat
bmat.dat
in
```

To appreciate the performance of CVSS for the problem (of size about 500), simply type

```
cvss_xxx  <  in
```

on your system, where 'xxx' should be replaced with associated platform. For instance, you should say

```
cvss_hp  (for HP machines)
cvss_sun (for SUN machines)
cvss_sgi (for SGI machines)
```
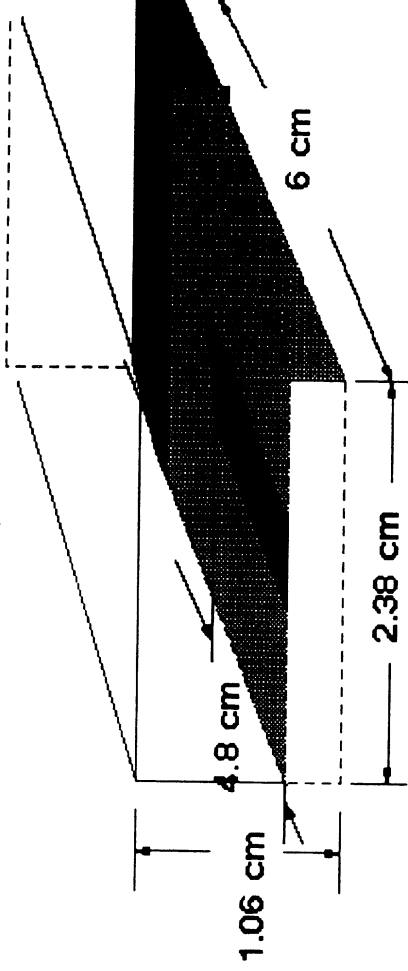
VSS Performance/Results

- Sparse LU Solver
- Developed by NASA Langley
- Specialized to Sparse Systems
- Available on all Platforms
- Relatively Low Memory
- Compiled and tested by UM for EM hybrid systems

CVSS Performance

Size	Non-zeroes	MBytes	CPU(sec)
6,448	89,365	<1	22.1
21,200	305,845	4	119.8
62,769	650.355	17	345

On an HP workstation rated @47Mflops

CPU for Iterative Solver

3-10 min or greater

6 cm

4.8 cm

2.38 cm

1.06 cm