Solving Helmholtz Equation with Compressed Sensing Enhanced Finite Element Methods

Student Name: Pragya Sharma

RollNumber: MT13160
July 10, 2015

Indraprastha Institute of Information Technology
New Delhi

Under the Supervision of
Dr. Shobha Sundar Ram (IIITD)
Dr. Angshul Majumdar (IIITD)

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Electronics and Communication Engineering

Keywords: Finite Element Method, Compressed Sensing, l_1 minimization, Sparsity

Certificate

This is to certify that the thesis titled **"Solving Helmholtz Equation with Compressed Sensing Enhanced Finite Element Methods"** submitted by **Pragya Sharma** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Electronics and Communication & Engineering* is a record of the bonafide work carried out by her / him under my / our guidance and supervision at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Shobha Sundar Ram
Indraprastha Institute of Information Technology, New Delhi

Dr. Angshul Majumdar
Indraprastha Institute of Information Technology, New Delhi

Abstract

The generalized finite element method (FEM) approach towards solving the Helmholtz equation involves a very high computational complexity of the order of $O(n^3)$ where n is the number of nodes of the FEM formulation. Prior research involved exploiting the special properties of FEM matrices for reducing the computation time and memory involved in solving the large FEM problems. These included both direct and iterative solvers. In more recent times, graphical processing units (GPUs) are being used to accelerate the solvers.

In this work, we propose an alternative different approach for solving the Helmholtz equation with reduced memory requirements by incorporating compressed sensing (CS) techniques into the original FEM formulation. Our approach is based on the fundamental assumption that electromagnetic fields are continuous except at source locations and can be represented with sparse coefficients in alternate transform domains such as wavelets or DCT. We present different practical aspects of this approach with respect to one-dimensional FEM problems and conclude by pointing out some open-ended questions with respect to this area of research.

Acknowledgement

This work would not have been possible without the guidance and support of several individuals who boosted me for completion of this work.

I would like to first thank my advisors Dr. Shobha Sundar Ram and Dr. Angshul Majumdar, for providing excellent guidance, encouragement throughout the project work. Without his indispensable guidance, this work would never have been a successful one.

Thanks to IIIT-D for providing excellent environment and infrastructure. I would also like to thank all my classmates for their valuable suggestions and fruitful discussions. Last but not the least, I would like to thank my family for supporting me spiritually and emotionally throughout my life.

Pragya Sharma

Contents

1 Introduction **1**

1.1 Background 1

1.2 Motivation 2

1.3 Organization of Thesis 3

2 Precursors **4**

2.1 Compressed Sensing 4

2.1.1 Sparsity 4

2.1.2 Measurement Basis 6

2.2 Electromagnetic Modeling 6

2.2.1 Finite Element Method 7

3 Literature Survey **9**

3.1 Finite Element Method using LU Decomposition 10

3.2 Finite Element Method using LDLT decomposition 11

3.3 Finite Element Method using Frontal and Multifrontal Meth-od 11

3.4 Finite Element Method using Conjugate Gradient Method 12

3.5 Finite Element Method using Preconditioned Technique 13

3.6 Comparative study of Direct and Iterative solvers 14

3.7 Other Studies 15

4 Proposed Formulation **17**

4.1 Finite Element Method with Compressed Sensing 17

5 Results **21**

5.1 Evaluation Parameter 21

5.2 Results 21

6 Conclusion and Future Scope **29**

6.1 Conclusion 29

6.2 Future Scope 30

List of Figures

2.1 Finite Element One-dimensional Problem Space subdivided into linear elements 8

5.1 Effective Cell Size versus Energy:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space 25

5.2 Electric Field versus Grid Points for five different cases:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space 26

5.3 Effective Cell Size versus Energy for different frequencies:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space 27

5.4 Effective Cell Size versus Energy comparison of other methods and our approach:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space 28

List of Tables

5.1 Description of electrical characteristics (permittivity $\epsilon_r(x)$, permeability $\mu_r(x)$ and conductivity $\sigma(x)$) of the 5 media 22

5.2 Normalized mean square error between the electrical energy estimated from CS enhanced FEM and ordinary FEM 23

5.3 Computational time calculated from original FEM, FEM with direct solver, FEM with iterative solver and FEM with our compressed sensing approach 24

5.4 Computational Memory calculated from original FEM, FEM with direct solver, FEM with iterative solver and FEM with our compressed sensing approach . . . 25

Chapter 1

Introduction

1.1 Background

The problem related to electromagnetics e.g. calculating electric or magnetic field from Helmholtz equations are very complex. Finite Element Method is one of the powerful numerical technique to solve Partial Differential Equations (PDEs) and has various applications in different areas. For dealing with larger size numerical problems in finite element analysis, many algorithms have been developed on parallel computers which utilizes a vast number of CPUs to achieve high speedup [24].

The main attraction point of finite element analysis is the assembly and solution of the sparse linear system of equations resulting from FEM discretization because these steps are the most computation-intensive steps. In [13] general approaches of assembling finite element equations were summarized and discussed which introduce multiple strategies for efficient use of global, shared and local memory, methods to achieve optimal performance.

Finite element method generates a matrix which has some special properties such as sparse, banded and symmetric. Direct solvers such as LU and LDLT decomposition techniques exploit these special properties for solving Helmholtz equation. Iterative solvers, such as conjugate gradient techniques, reduce the computation time by iteratively solving for unknown. However,

both of these types of solvers have some limitations especially when they are used in systems with scarce memory resources such as personal computers. Direct solver has slow response for larger number of systems. Also, they require too much memory resources for solving large number of systems. Irregular structured finite element problems of order of $100,000$ unknowns may be solved by direct methods given a large enough storage space in computer but at tremendous cost and difficulty. Though iterative solvers require less storage capacity but their performance depends upon the problems. An adequate preconditioning is required in this case which is not an easy way to solve the system in an efficient and reliable manner [10]. Also, iterative solvers approach the solution gradually rather than in one large computational step. Therefore, a technique can be developed for solving these equations in a systematic manner and without having problems related to computational complexity and memory space.

1.2 Motivation

The idea behind this thesis work is to provide benefits in terms of computational complexity and memory. The solution of conventional finite element equations requires computational complexity of the order of $O(n^3)$, where n-1 is the number of discrete elements in the problem space. As the size of the problem grows means if n is large, this complexity becomes a limiting factor especially when finite element method is used in systems with scarce memory resources such as personal computers. The alternative of utilizing large servers, distributed or parallel computing, and expensive computational software and licensed packages often involves prohibitive expenditure. Considering this factor, a different approach for solving the Helmholtz equation with reduced memory requirements, is proposed by incorporating compressed sensing formulation into original Finite Element Method (FEM). Compressed sensing basically involves the sampling (either 50% or 25%) of the FEM solution, thus reduces the memory requirement and complexity.

1.3 Organization of Thesis

This thesis is organized as follows.

Chapter 2 provides a brief understanding of compressed sensing and a brief intoduction of Fimite Element Method.

Chapter 3 presents the literature survey of work done on Finite Element Method with different solvers and a comparitive study of them.

Chapter 4 proposes Finite Element method with our compresses sensing based approach to reduce computational complexity and memory requirement.

Chapter 5 discusses the results for different-different cases.

Chapter 6 concludes the thesis by summarizing the contribution of the work and research done along with the possible extension of our work that can be explored.

Chapter 2

Precursors

This chapter briefly discusses the theoritical concepts of Compressed Sensing. Also, a brief introduction of Finite Element Method is presented.

2.1 Compressed Sensing

Compressed Sensing (CS) is an emerging methodology that is used for solving underdetermined linear system of equations. Compressed sensing is based on the fact that the sparsity of a signal can be exploited to recover it from sub nyquist samples [5].

2.1.1 Sparsity

Compressed Sensing solves the underdetermined system of linear equations, when the solution is known to be sparse. The linear system of equation can be expressed as:

$$y_{m\times 1} = A_{m\times n} x_{n\times 1} \tag{2.1}$$

The underdetermined system has infinite number of solution. There exists an orthonormal basis ϕ such that x=ϕc with c being sparse. In Eq.2.1 A is called as measurement matrix or sensing matrix where m<n. Then the compressed sensing problem from Eq.2.1 can also be formulated as

$$y_{m\times 1} = A_{m\times n}\phi_{n\times n}c_{n\times 1} \tag{2.2}$$

In both the equations 2.1 and 2.2 we have prior information related to the sparsity of x. If the unknown x is $k-$sparse and the k locations of the nonzero elements in x are known, then the problem can be solved with reasonable accuracy if the number of equations (m) is larger than the number of nonzero locations (k). Since, the l_2-norm measures signal energy not signal sparsity, we consider l_0-norm that counts the number of non-zero entries in x.

The solution to the Eq.2.1 is sparse then the sparsest solution can be found and this solution would be unique [4].

$$\min_x ||x||_0 \tag{2.3}$$

$$\text{subject to } y{=}Ax$$

Minimizing the l_0-norm would yield the sparsest solution and recover a $k-$sparse signal exactly with high probability using only $m = k + 1$ iid Gaussian measurements [1]. Unfortunately minimizing the l_0-norm is both numerically unstable and an NP-hard problem [7]. There are many greedy algorithm such as Orthogonal Matching Pursuit (OMP) [22] for solving this NP hard problem but recovery of the signal is poor. So we solve it by relaxing the constraint using l_1-norm.

$$\min_x ||x||_1 \tag{2.4}$$

$$\text{subject to } y{=}Ax$$

The l_1-norm minimization problem can be solved easily by linear programming. It would exactly recover $k-$sparse signal and closely approximate compressible signals with high probability using only $m \geq cklog(n/k)$ iid Gaussian measurements [1]. This is a convex optimization problem that conveniently reduces to a linear program known as basis pursuit [6] whose computational complexity is about $O(n^3)$.

2.1.2 Measurement Basis

Compressed Sensing has two fundamental key points: sparsity and incoherence. In CS signal should be sparse in some sparsifying transform domain. The domain can be wavelet or DCT. Coherence measures the largest correlation between measurement and sparsifying basis. CS mainly concerns with low coherence pairs. The fourier basis is maximally incoherence in 1D, 2D and 3D also. The coherence between wavelet bases (Haar wavelet) and measurement bases is $\sqrt{2}$. Our interest in measurement bases comes from the fact that 1) they are incoherent with systems providing sparse representations of image data and other types of data, and 2) they come with very fast algorithms; these transform runs in $O(n)$ time. It has been found that random matrices are largely incoherent with any fixed basis [4].

Gaussian Matrices

This matrix have entries chosen independently from normal distribution with zero mean and 1/n variance. This is the most commonly used sensing or measurement basis. Being a dense matrix i.e. values present at every location, they have large storage and involves lot of computation when a signal is projected onto them.

Binary Matrices

The entries of a random binary matrix takes the value of 1 or 0 with equal probability. These matrices have lesser storage and computation as compared to Gaussian random matrices while performing equally well [23].

2.2 Electromagnetic Modeling

Computational Electromagnetics deals with the modeling of electromagnetic fields with physical

objects and the environment. It typically involves efficient approximation to Maxwell's equations and notifies the electromagnetic wave propagation in various medium. This section includes the basic understanding of Finite Element Method.

2.2.1 Finite Element Method

Electromagnetic wave propagation is described by four particular equations, Maxwell equations, two of them can be expressed as follows

$$\bigtriangledown \times \overrightarrow{\mathrm{E}} = -\frac{\partial \overrightarrow{\mathrm{B}}}{\partial t} \tag{2.5}$$

$$\bigtriangledown \times \overrightarrow{\mathrm{H}} = -\frac{\partial \overrightarrow{\mathrm{D}}}{\partial t} \tag{2.6}$$

where E and H are electric and magnetic field vectors respectively. Therefore, we use these two Maxwell equations to formulate Helmholz equations

$$\bigtriangledown^2 \overrightarrow{\mathrm{E}} + k^2 \overrightarrow{\mathrm{E}} = -\epsilon \overrightarrow{\mathrm{H}} \tag{2.7}$$

where $\bigtriangledown^2$ is laplacian and ϵ is the permittivity in the medium and k is the propagation constant.

To find the electric and magnetic field, these Helmholtz equation are solved. There are two methods for solving these Helmholtz equation in differential form: Finite Difference Time Domain(FDTD) and Finite Element Method(FEM). Here, we have used finite element method for one-dimensional problem. The finite element method is a versatile tool for solution of partial differential equations with specified boundary conditions and has been extensively used in various engineering problems. Consider a one-dimensional Helmholtz equation

$$\frac{d}{dx}\left(\alpha \frac{du}{dx}\right) + \beta u = s, a \leq x \leq b \tag{2.8}$$

This equation is called the reaction-diffusion equation. Reaction term is given by βu and the diffusion term is given by $\frac{d}{dx}\left(\alpha \frac{du}{dx}\right)$. The solution of the equation is subject to boundary conditions given by Dirichlet or Neumann boundary conditions u(a)=p or u'(a)=q. Here, u is

the unknown electric or magnetic field, s is the source excitation, $\alpha(x)$ and $\beta(x)$ are the material properties of the medium within the interval and x is the length of the problem space. For a one-dimensional problem we have problem space Ω. This problem space has infinite number of points so it is not possible to calculate field at all these infinite number of points. Therefore in FEM, we divide this interval into finite number of points as shown in Fig.2.1. and calculate the field at all those finite number of points with the Dirichlet boundary condition Γ_D at the end points of problem space as shown in Fig.2.1.

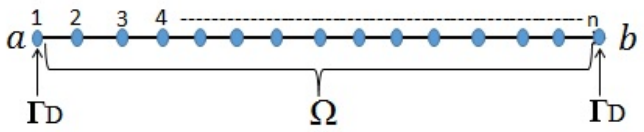


Figure 2.1: Finite Element One-dimensional Problem Space subdivided into linear elements

Hence, finite element method converts the Helmholtz equation into weak formulation and recast as a linear inverse problem:

$$H_{n\times n}u_{n\times 1} = b_{n\times 1} \tag{2.9}$$

where $H_{n\times n}$ is the Helmholtz operator, $u_{n\times 1}$ are the discrete values of u at n nodal positions and $b_{n\times 1}$ arises from manipulations of the source s. H and b matrix are generated by the weak formulation of Eq.2.8. In weak formulation, both the side of Eq.2.8 is multiplied by a basis function [21]. These basis function can be defined by interpolation which can have linear, quadratic and cubic elements. Hence, we calculate the unknown field at all those n points which has been defined in the problem space Ω. The computational resources (CPU time and memory) required for solving FEM is of $O(n^3)$.

Chapter 3

Literature Survey

The finite element formulation can be represented as a set of linear equations which can be written as

$$Hu = b \tag{3.1}$$

where H is an n $\times$ n matrix, u represents the unknown field vector that is to be determined and b represents the source vector which is known to us. Many algorithms have been developed for solving Helmholtz equation. As discussed in the previous chapters, during computation of the numerical solution of the problem two basic key points which can be considered: large memory requirement and excessively long computational time.

In Eq.3.1 if H is a non-singular matrix, a great variety of algorithms have been developed. When H is not sparse, Gaussian elimination is the most preferable approach. But this requires $O(n^2)$ word to store and $O(n^3)$ floating-point arithmatic operations. CPU time is also very high. The matrices resulting from finite element discretization are always sparse, banded and symmetric. By using these properties memory requirement and computing time both can be reduced. To utilize these properties many algorithms have been developed in past few years for handling finite element matrices properly [11]. In this chapter, a review of these algorithms is presented.

These methods used to solve linear algebraic Eq.3.1 can be categorized into two groups: di-

rect methods and iterative methods. All direct methods are based on Gaussian elimination. Sparse direct solvers save memory and computing time by considering the sparsity pattern of matrix H [11]. We will discuss these algorithms in a detailed manner.

3.1 Finite Element Method using LU Decomposition

One way of solving Eq.3.1 is LU decompositon. This method decomposes matrix H into two triangular factors L and U where L is a lower triangular matrix and U is an upper triangular matrix. For a square matrix H, LU factorization can be represented as

$$H = LU \tag{3.2}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & \cdots & \cdots & 0 \\ l_{31} & l_{32} & \ddots & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \cdots & u_{n-1n} \\ 0 & 0 & \cdots & \cdots & u_{nn} \end{bmatrix}$$

The solution to the Eq. 3.1 can be obtained by solving

$$Ly = b \tag{3.3}$$

and

$$Ux = y \tag{3.4}$$

where y can be obtained by forward substitution procedure

$$y_1 = \frac{b_1}{l_{11}} \tag{3.5}$$

$$y_i = \frac{1}{l_{ii}} \left(b_i - \sum_{k=1}^{i-1} l_{ik} y_k \right); i > 1 \tag{3.6}$$

and the unknown field x can be obtained by backward substitution procedure.

$$x_n = \frac{y_n}{u_{nn}} \tag{3.7}$$

$$x_i = \frac{1}{u_{ii}}\left(y_i - \sum_{k=i+1}^{n} u_{ik}x_k\right); i < n \tag{3.8}$$

This method has the benefit that we do not need to do Gaussian elimination each time and it is faster than any other decomposition. In this algorithm, computation increases as $O(n^2)$ for two-dimensional problem and $O(n^{7/3})$ for three-dimensional problem. Also, the memory requirement increases as $O(n^{3/2})$ in 2D and $O(n^{5/3})$ in 3D.

3.2 Finite Element Method using LDLT decomposition

In this method the matrix H can be decomposed as

$$H = LDL^T \tag{3.9}$$

where L is the lower triangular matrix and D is the diagonal matrix of L. In this process, compared to LU decomposition we don't require upper triangular matrix U. So there is no need to store the entries from both sides of the diagonal. Hence, the operation count reduces to $O(n^3/3)$.

3.3 Finite Element Method using Frontal and Multifrontal Method

Frontal and multifrontal method is a way to reduce a large amount of core memory required to

solve large finite element systems. FEM is assembled from elemental matrices:

$$H = \sum_{e=1}^{M} H^{(e)} \tag{3.10}$$

where H is the Helmholtz operator and M-1 is the number of elements.

For calculating specific unknown (field values) only the elements associated with it directly contribute to its equation and all other elements have no contributions. In this method once the equation is established, it is written to the out of core memory hence, leaving the core memory for new equations. The matrix in the core memory is called the frontal matrix [11]. The size of the frontal matrix varies in the process and maximum front width determines the core memory requirement. So, proper ordering of elements can reduce the maximum front width which further reduces the core memory requirement.

For large scale two and three dimensional problems, the front width can be quite large. A large frontal matrix increases the memory requirement as well as the cost of factorization notably. As a remedy of this problem, in [11] a multifrontal method has been described very effectively with suitable exapmle. This would reduce the cost of factorization and core memory requirements significantly.

3.4 Finite Element Method using Conjugate Gradient Method

Conjugate Gradient is an iterative method for solving a linear system of n equations where n is large. It terminates in at most n steps. The complexity reaches to $O(n)$. Memory is typically linear that is $O(n)$. It automatically generates the direction vectors. The successive approximations to the solution vector H are calculated as follows

$$u_{k+1} = u_k + \alpha_k p_k \tag{3.11}$$

where p_k are known as direction vectors and α_k is chosen to minimize the function in the direction of p_k. The iterative methods like Conjugate Gradient method best suits to sparse systems. If matrix is dense then best way to solve the system is by back substitution.

3.5 Finite Element Method using Preconditioned Technique

Over the decades, several works have been proposed such as classical preconditioning techniques [8] [17] to solve these linear equations because for achieving 100% accuracy Conjugate Gradient method demands for many iterations. Therefore, the finite element matrices become ill-conditioned with larger dimensions. In proceeding of this work, to accelerate the convergence of the conjugate method preconditioning is required. One of the studies in this area is [12] [19] where a 2D and 3D solutions of Helmholtz equation is discussed by means of a standard Galerkin or Galerkin least-squares (GLS) scheme. The solution is obtained by a preconditioned Krylov subspace technique, specifically a preconditioned GMRES iteration [20] [18]. Three types of preconditioners have been proposed: ILUT, ILUTC and ILUT0. A preconditioned Krylov subspace method for solving the linear system of Eq.3.1 consists of an accelerator and a preconditioner. Therefore, a right preconditioned system can be written as

$$HM^{-1}y = b \tag{3.12}$$

and

$$x = M^{-1}y \tag{3.13}$$

where M is the preconditioning matrix.

Preconditoning matrix can be defined by incomplete LU factorizations which can be obtained from Gaussian elimination process. A large number of nonzero elements may appear in locations originally occupied by zero elements. These fill-in elements are often small and may be dropped to obtain incomplete LU factorizations. Among these procedures, one is ILU(0) which is obtained by performing the standard LU factorization of H and dropping all fill-in elements. Here L and U factors have the same pattern as the lower and upper triangular parts of H respectively. Another one is ILU(k) for accurate factorization in which fill-ins are dropped according

to their levels in the elimination process. One more class of preconditioner is ILUT (ILU with Threshold) in which dropping of fill-ins is based on their numerical values. ILUT is based on two parameters: ς and lfil. Small values are dropped during the elimination using a parameter ς and large values are kept by lfil.

All the incomplete LU factorization techniques have been tested for different cases when matrix H has some diagonal dominance properties and found out that ILU techniques are the best across a broad spectrum of applications even when matrix H is not diagonally dominant. Also, the effect for different frequency regimes and for different preconditioners have been presented. In case of ILUT, as the lfil is increased the convergence rate increases.

As our previous discussion that computing time should be less so the number of iteration sholud be less, this can be achieved by highest values of lfil and lowest value of drop tolerance (ς). Also, the GLS method gives better accuracy than classical Galerkin method in case of higher frequency. For Dirichlet problem, it has been seen that ILUT-GMRES solver requires fewer step than ILU(0)-GMRES to converge. However, the CPU time is slightly greater for ILUT-GMRES. For Neumann problem, ILUT-GMRES is faster than the ILU(0)-GMRES in terms of iteration count and CPU time.

3.6 Comparative study of Direct and Iterative solvers

In [10] a detailed comparison of direct and iterative solvers has been discussed.

1. Direct solvers always work for invertible matrx and they are fast for less than 100k unknowns. But they require too much memory and CPU time for larger systems. On the other side, iterative solvers can solve large problems and require $O(n)$ memory.

2. There is a wide variety of iterative solvers, hence the selection of these solvers depends upon the properties of the matrix. Iterative methods are very effective concerning computer storage and time requirements.

3. One of the ease of using iterative methods is that they require fewer multiplications for large systems. Iterative methods automatically adjust to errors during study. They can be implemented in smaller programmes than direct methods. They are fast and simple to use when coefficient matrix is sparse. In terms of benefit, they have fewer rounds off errors as compared to other direct methods.

4. On the opposite side, the goal of direct methods is to calculate an exact solution in a finite number of operations whereas iterative methods includes initial approximation and reproduction of usually improved approximations in an infinite sequence whose limit is the exact solution.

5. Direct methods are suitable for such kind of systems in which most of the entries are non-zero whereas iterative methods are appropriate for large sparse systems which contains most zeros. Even when direct methods exists we should give priority to iterative methods because they are fast and efficient.

3.7 Other Studies

In [9] another method for solving partial differential equation has been introduced. The main idea is to construct finite element base functions which records the small scale information within each element and then the small scale information is brought to the large scales through the coupling of global stiffness matrix. This method is called as Multiscale FEM. Also, they have shown that the operation count of this method is about twice of that conventional FEM for a 2D case. Another advantage of this method is its ability to reduce the size of a large scale computation. This offered a big saving in memory. This is suitable for those problems which are too large to handle by direct methods. For N number of elements and M number of subcell elements, conventional FEM requires $O(M^n N^n)$ CPU memory for solving the problem on fine grid whereas MFEM requires $O(M^n + N^n)$ CPU memory, where n is the dimension.

Some other methods have also been proposed such as Algebric recursive Multilevel Solver (ARMS) [16], Multilevel Fast Multipole Algorithm (MLFMA) [15]. But the MLFMA technique is very time consuming because of a large constant associated with its computational complexity.

Chapter 4

Proposed Formulation

In chapter 2 we have formalized the Finite Element method which includes discretization of the Helmholtz equation into finite element with n nodes and convert it into linear inverse problem. We solve this linear inverse problem by incorporating compressed sensing techniques into the original FEM formulation. Instead of solving the problem on a fine n-point grid we solve it on a coarse m-point (m < n) grid. So, the proposed technique reduces the computational complexity as well as the memory requirement. Our approach is based on the assumption that fields can be represented with sparse coefficients in alternate transform domains such as wavelets or DCT. For recovery of the original field from compressed field several recovery algorithms are proposed. This chapter is dedicated to provide an understanding of the propsed formulation. In this chapter, we discuss the compressed sensing which has been introduced into the original FEM problem.

4.1 Finite Element Method with Compressed Sensing

In this section we will discuss about the proposed formulation. Finite element solves the field for n finite points. But as the size of the problem is increased that means if the value of n is large then it becomes computationally complex to calculate the field. Hence, we take a different approach to solve the Helmholtz equation efficiently. Therefore, we solve the linear inverse problem of Eq.2.9 only for m grid points instead of n grid points. The new problem can be

expressed as:

$$H_{n\times n}R_{n\times m}u'_{m\times 1} = b_{n\times 1} \tag{4.1}$$

Here R is a random sampling operator from the n-point grid to the sub-sampled m-point grid. Eq. 4.1 is an over-determined problem so it is easy to solve. Assuming that u' is the solution on the coarse m-point grid,

$$u' = (HR)^{\dagger}b \tag{4.2}$$

$$\Rightarrow u' = R^{\dagger}H^{-1}b \tag{4.3}$$

The formula regarding the pseudo-inverse of a product holds under certain conditions [2]. From Eq. 2.9 and 4.3 we can write:

$$u = H^{-1}b \Rightarrow u' = R^{\dagger}u \tag{4.4}$$

The pseudo-inverse of R is formed from the Canonical basis (Dirac) which means it is defined by the kronecker delta which is a function of two variables. This function has a value 1 if the variables are equal and 0 otherwise.

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

Hence, it can be efficiently implemented as a selection operation having linear complexity. Our solution is based on the key feature of the wave field - the field is a piecewise smooth function with finite number of discontinuities at the source and at material boundaries. Such piecewise smooth signals may have a sparse representation in various transform domains like wavelets and discrete cosine transform. CS has exploited the sparsity of a solution to recover it from its sub-sampled values in several problems across different disciplines and is specially suited for recovering the fully sampled field at all n points from the partially estimated field at m points.

Compressed sensing assumes that the signal is measured in a basis - in this case the measurement basis is the Dirac. We have mentioned before that the field is not sparse but has a sparse representation in a transform doamin like wavelet reference or DCT reference. Unfortunately, if the signal is measured in the dirac basis, sparsifying transforms like DCT and wavelet

are not incoherent with it. CS requires that the measurement basis and the sparsifying basis should be maximally incoherent from each other. If the coherence is high (low incoherence), the recovery error is more for a given number of measurements [3].

Owing to such an incoherence requirement, the Fourier basis is the only viable alternative in this situation since it is maximally incoherent with the Dirac basis. In fact the incoherence between the Fourier and the Dirac basis is theoretically the maximum possible; between any other measurement and sparsifying basis (like Gaussian / Binary measurement basis and DCT / wavelet sparsifying basis), the incoherence is always less than the Fourier-Dirac pair reference. Therefore, Fourier basis will yield the best recovery results owing to maximal incoherence with the Dirac sparsifying basis.

Hence, the field u will be sparse in some transform domain D where it is expressed as Dχ. This is a realistic assumption since the field will always be continuous except at material boundaries and at the sources. Several mathematical transforms exist that can sparsely represent such piecewise smooth signals, e.g. wavelet, DCT or higher order total variation (TV). The sparsity of the piecewise smooth field can be exploited to solve Eq.4.4 via $l1$-norm minimization:

$$\chi = \min_{\chi} ||\mathbf{u}' - \mathbf{R}^{\dagger}\mathbf{D}\chi||_2 + \lambda||\chi||_1 : u = D\chi \tag{4.5}$$

Here, λ denotes the regularization parameter. We have proposed two steps. In the first step Eq.2.9 is solved on a sub-sampled grid (u) using any standard solver since the problem will be over-determined. In the second step, we recover the field (u) for the fully sampled grid from the solution obtained from the first step. Solving the second step is not trivial since the problem in the second step is under-determined.

The computational complexity of the first step of our proposed algorithm is $O(m^3)$ where $m < n$. The computational complexity in each iteration of the l_1-minimization in the second step is the cost of selection $O(n)$ and the cost of applying the transform D is $O(n \log n)$ for Fourier / DCT and $O(n)$ for wavelets / first order total variation. Therefore, the overall complexity of solving our problem is significantly less than solving the full FEM directly.

For recovery of the field (u) for fully sampled grid from the solution obtained after subsampling, many algorithms have been suggested. [14] shows the comparison of different algorithms. The best algorithm is Spectral Projected Gradient for $l1$-minimization (SPGL1). SPGL1 is a solver for large-scale sparse reconstruction problems. At each iteration, a spectral gradient-projection method approximately minimizes a least-squares problem with an explicit one-norm constraint. Only matrix-vector operations are required. SPGL1 does not work for TV minimization.

From Eq.4.5 after getting the values of χ we can easily reconstruct the field u by multiplying it with the coefficients of some sparsifying transform domain. Eq.4.5 is a regularized least sqaure problem and the regularized term λ is added to the equation to solve an ill-posed problem or to prevent from overfitting. λ parameter controls the trade-off between the sparsity and reconstruction fidelity.

Chapter 5

Results

5.1 Evaluation Parameter

We have considered various cases and generated the results accordingly. To estimate the performance an evaluation parameter is required. In our case, we compare our algorithm with the original existing technique and selected parameter is Normalized Mean Square Error (NMSE).

$$NMSE= \frac{||original-reconstructed||_2}{||original||_2}$$

5.2 Results

We have considered a one-dimensional space $0 \leq x \leq 10$m for the case when source is $\delta(x - 0.2) + \delta(x - 0.8)$. The boundary condition which has been applied is Neumann boundary condition (u'(0) = 0, u'(10) = 0) at both the ends of the problem space. We solve for field u using full FEM and using the CS enhanced FEM. In order to study the effectiveness of the CS based solution, we consider two reconstruction scenarios - First, the reconstruction is carried out with a 50% partially sampled grid space (m=n/2) and in the second case, with 25% sampled grid space (m=n/4). This is done for five different cases

1. Free Space
2. Dielecrtic within free space
3. Metal within free sapce (Hign Conductivity medium)

S.No	Medium	$0 \leq x < 3.5$	$3.5 \leq x < 6.5$	$6.5 \leq x < 10$
1	Free space	$\epsilon_r(x) = 1, \sigma(x) = 0$	$\epsilon_r(x) = 1, \sigma(x) = 0$	$\epsilon_r(x) = 1, \sigma = 0$
2	Dielectric within free space	$\epsilon_r(x) = 1, \sigma(x) = 0$	$\epsilon_r(x) = 7, \sigma(x) = 0$	$\epsilon_r(x) = 1, \sigma(x) = 0$
3	Conductor within free space	$\epsilon_r(x) = 1, \sigma(x) = 0$	$\epsilon_r(x) = 18, \sigma(x) = 3 \times 10^{18}$	$\epsilon_r(x) = 1, \sigma(x) = 0$
4	Conductor within free space	$\epsilon_r(x) = 1, \sigma(x) = 0$	$\epsilon_r(x) = 7, \sigma(x) = 0.01$	$\epsilon_r(x) = 1, \sigma(x) = 0$
5	Inhomogeneity within free space	$\epsilon_r(x) = 1, \sigma(x) = 0$	$\epsilon_r(x) = \frac{R_2}{R_2 - R_1} \frac{(x-R_1)^2}{x^2}, \sigma(x) = 0$ $R_1 = 3.5m, R_2 = 6.5m$	$\epsilon_r(x) = 1, \sigma(x) = 0$

Table 5.1: Description of electrical characteristics (permittivity $\epsilon_r(x)$, permeability $\mu_r(x)$ and conductivity $\sigma(x)$) of the 5 media

4. Conductor having low conductivity within free space

5. Inhomogeneity within free sapce

These five different mediums have been applied for 3.5m $\leq x \leq$ 6.5m. Description of their electrical characteristics (conductivity, permittivity, permeability) for these five mediums are shown in the Table 5.1. Then, the energy is calculated and energy versus effective cell size (x(m)/Number of points) is plotted in Fig.5.1. This figure conatins the result of convergence analysis for full FEM and our approach with 50% and 25% sampling. It is obvious from the figure that convergence is achieved even with less number of points than the conventional FEM method. Here, we have used DCT as a sparsifying basis. Therefore, we get the reconstructed field with half of the number of nodes or even with the one-fourth nodes rather than calculating field for all number of nodes.

Mediums	Resolution	NMSE (CS 25% sampling)	NMSE (CS 50% sampling)	NMSE (CS 10% sampling)
Medium 1	$\frac{\lambda}{82.5}$	0.052	0.0085	0.33
Medium 2	$\frac{\lambda}{175}$	0.85	0.19	5
Medium 3	$\frac{\lambda}{125}$	1.6	0.47	5
Medium 4	$\frac{\lambda}{82.5}$	1.6	0.38	4.14
Medium 5	$\frac{\lambda}{82.5}$	0.9	0.09	2.4

Table 5.2: Normalized mean square error between the electrical energy estimated from CS enhanced FEM and ordinary FEM

Based on the convergence point, we get the actual numbering of nodes for which the error between the field generated by conventional FEM and the field generated by our approach. At those nmuber of nodes we get the electric field versus grid points (x) graph which is shown in Fig.5.2. where NBC means Neumann Boundary Condition. This figure depicts that the reconstructed field is overlapping the original FEM field it means the field is fully reconstructable and the NMSE between the field, which we are getting from the conventional FEM approach, and the reconstructed field which we are getting from our compressed sensing approach is very less which is shown in Table 5.2.

Effect of frequency on the electric field

Further, we have done the analysis to see the effect of frequency on the energy for five cases given in Table 5.1. We also calculate the energy for λ=2.5m with 25% sampling. Fig.5.3 shows the comparison between the results for λ=0.25m and 2.5m. From this analysis, we come to know that larger wavelength means resolution is higher and better is the resolution we reach at the convergence point fast and the error would be less in this case.

Comparison of our approach with different algorithm

We compare our approach with direct solver (such as LU decomposition method) and iterative solver (such as GMRES preconditioning method) for all the four different cases. The results have been shown in Fig.5.4. The results dictate that convergence is achieved even with our

Mediums	Resolution	FEM	FEM_LU	FEM_GMRES	FEM_CS
Medium 1	$\frac{\lambda}{82.5}$	1.89sec	3.25sec	0.867sec	5.369sec
Medium 2	$\frac{\lambda}{175}$	9.122sec	11.527sec	2.07sec	21.456sec
Medium 3	$\frac{\lambda}{125}$	109.45sec	118sec	101.8sec	130.235sec
Medium 4	$\frac{\lambda}{82.5}$	110sec	116.654sec	103.335sec	128.345sec
Medium 5	$\frac{\lambda}{82.5}$	8.31sec	13.185sec	3.145sec	25sec

Table 5.3: Computational time calculated from original FEM, FEM with direct solver, FEM with iterative solver and FEM with our compressed sensing approach

approach having less number of points.

Computational Memory and Computational time

As we have previously discussed that the direct solvers and iterative solvers has a limitation that they cannot be used in a system having scarce memory resources. A personal computer with 6GB RAM is unable to solve Helmholtz equation which involves $100,000$ unknowns. The alternative of utilizing large servers, distributed or parallel computing, and expensive computational software and licensed packages often involves prohibitive expenditure. Hence our approach solves relatively large finite element problems involving $100,000$ unknowns using the limited memory resources available on personal computers.

From Table5.2 we can conclude that our proposed approach yields solutions within an accuracy of 5% while using a computational memory of $O(n/10)$ when compared to the results generated with conventional direct and iterative solvers.

Table5.3 shows the computational time and Table5.4 shows the computational memory results for the same resolution that was used in Table5.2.

Mediums	Resolution	FEM	FEM_LU	FEM_GMRES	FEM_CS
Medium 1	$\frac{\lambda}{82.5}$	98.5MB	96.7MB	46.8MB	20.25MB
Medium 2	$\frac{\lambda}{175}$	393.23MB	385.51MB	158.76MB	40.55MB
Medium 3	$\frac{\lambda}{125}$	401.12MB	351.48MB	223.31MB	21.17MB
Medium 4	$\frac{\lambda}{82.5}$	401.12MB	351.48MB	223.31MB	21.17MB
Medium 5	$\frac{\lambda}{82.5}$	393.23MB	385.51MB	158.76MB	40.55MB

Table 5.4: Computational Memory calculated from original FEM, FEM with direct solver, FEM with iterative solver and FEM with our compressed sensing approach

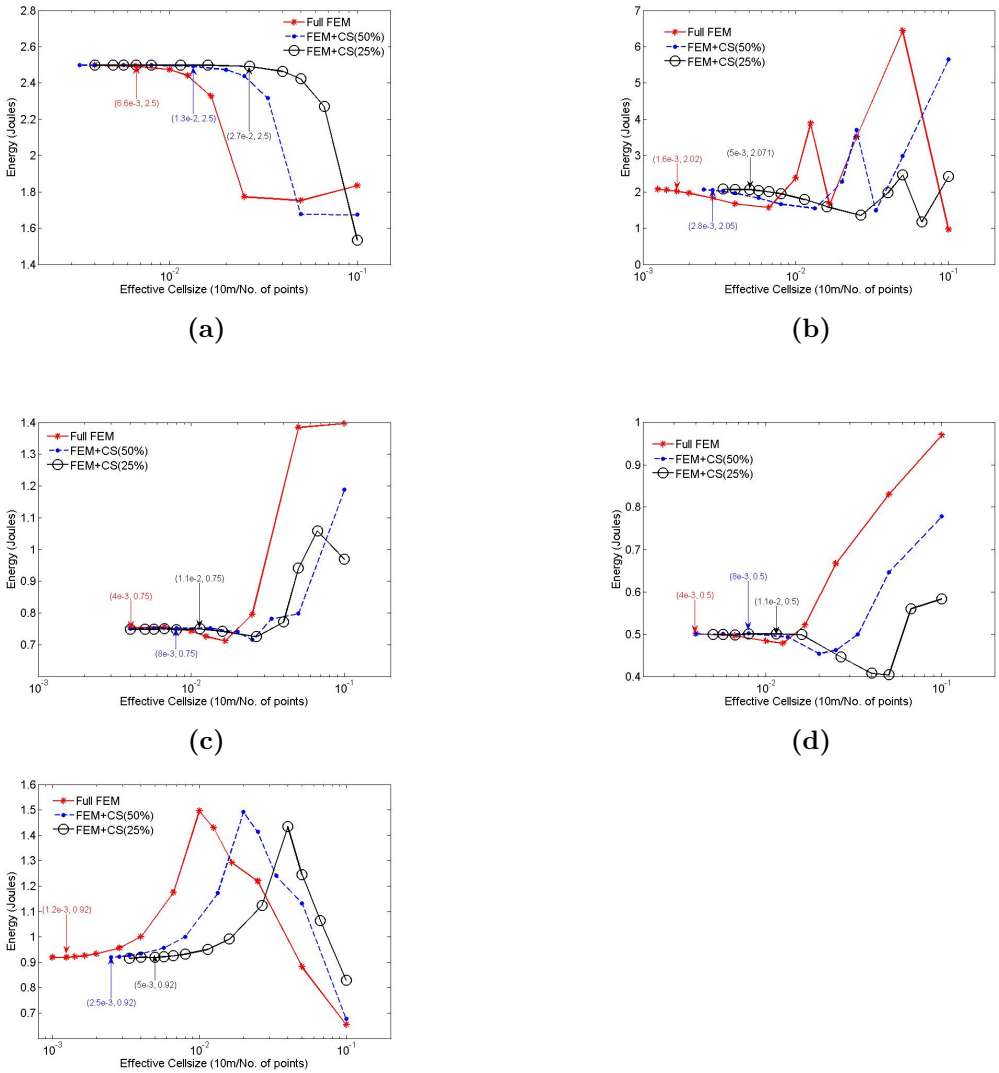
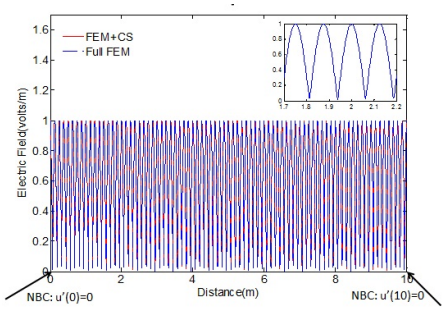
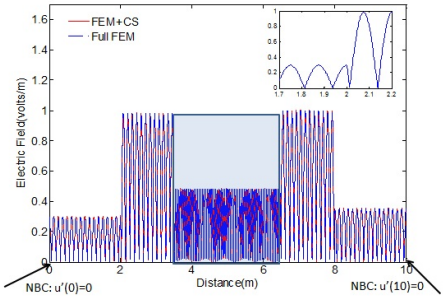


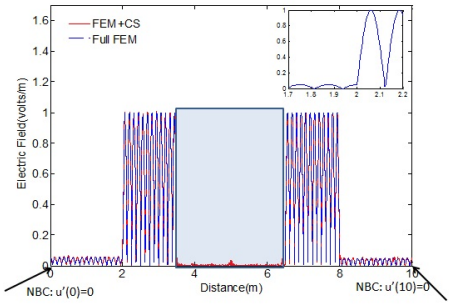
Figure 5.1: Effective Cell Size versus Energy:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space



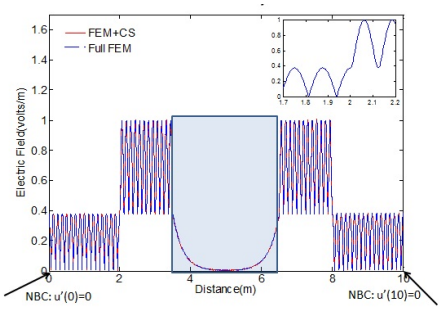
(a)



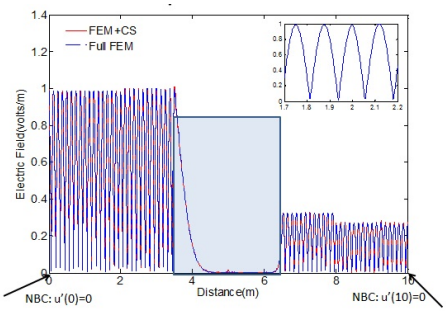
(b)



(c)

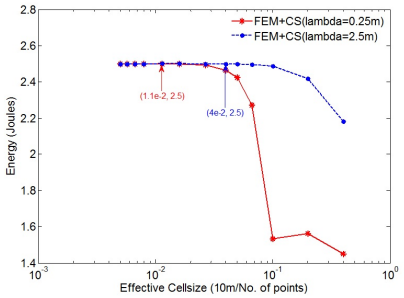


(d)

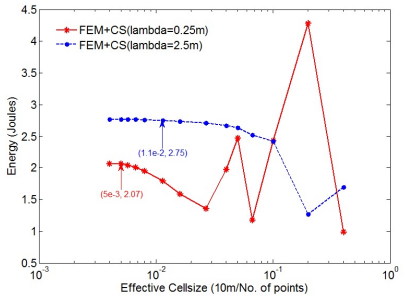


(e)

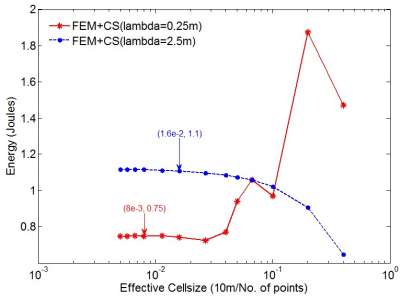
Figure 5.2: Electric Field versus Grid Points for five different cases:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space



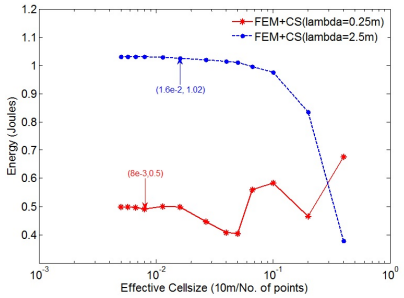
(a)



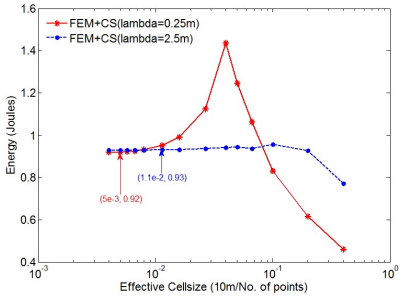
(b)



(c)

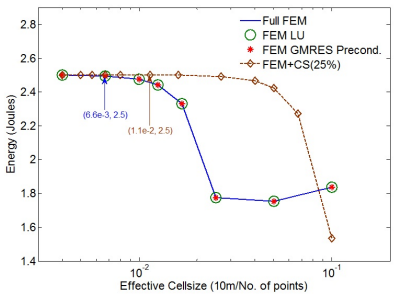


(d)

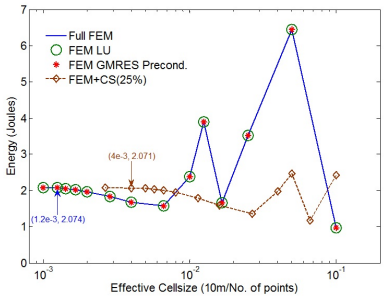


(e)

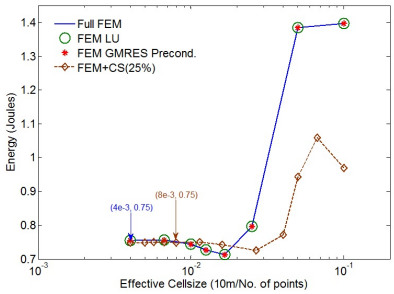
Figure 5.3: Effective Cell Size versus Energy for different frequencies:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space



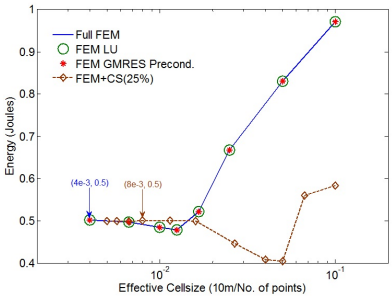
(a)



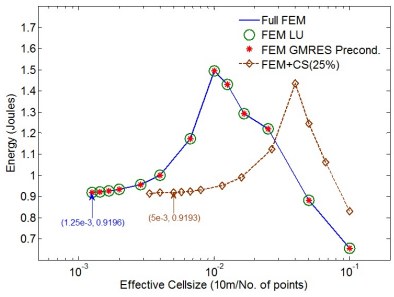
(b)



(c)



(d)



(e)

Figure 5.4: Effective Cell Size versus Energy comparison of other methods and our approach:(a)FreeSpace, (b)Dielectric within free space, (c)Metal within free space, (d)Low Conductivity Medium within free space, (e)Inhomogeneity within free space

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

For the very first time, this work of accelerating Finite Element Method using Compressed Sensing has been formulated which is very beneficial in terms of memory saving. Here, we have corroborated our approach with initial experimental results for the 1D Helmholtz equation. Finally, we conclude various results

1. Effect of frequency on the electirc field- It shows that if we decrease the frequency (wavelength increases), we reach the correct solution faster. It means we get the fully reconstructed field with less error.

2. Effect of different sampling ratio on the field- We have also generated the results for two sampling ratio (25% and 50%). The results depict that our algorithm also works even for 25% sampling ratio. We are able to reconstruct the field with very less NMSE value.

3. Effect of different sparsifying transform- The reconstructed approach comtains a measurement sparsifying basis. We have shown the results only for Discrete Fourier transform (DCT). The computational complexity is $O(nlog(n))$ for DCT and $O(n)$ for wavelet basis.

4. Comparison of our approach having 25% sampling ratio with direct method such as LU decomposition and iterative method such as preconditioned GMRES approach is done. It describes that our approach converges faster than both the direct and iterative methods.

6.2 Future Scope

In this thesis, we have focused only on solving FEM for Maxwells equation. The framework can be extended towards FEM formulations across multiple disciples such as acoustics, thermal conduction etc as well as other PDEs. We have proposed this method only in one-dimensional problem space. Therefore, it can be suitably extended across two-dimensional and three-dimensional problem spaces.

Also, we can use this compressed sensing approach in fluid dynamics, molecular dynamics and biomechanics.

Bibliography

[1] Baraniuk, R. Compressive sensing.

[2] Bouldin, R. The pseudo-inverse of a product. *SIAM Journal on Applied Mathematics 24*, 4 (1973), 489–495.

[3] Candes, E., and Romberg, J. Sparsity and incoherence in compressive sampling. *Inverse problems 23*, 3 (2007), 969.

[4] Candes, E., and Wakin, M. An introduction to compressive sampling. *Signal Processing Magazine, IEEE 25*, 2 (March 2008), 21–30.

[5] Candès, E. J. Compressive sampling. In *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures* (2006), pp. 1433–1452.

[6] Cheman, K. M. *Optimization techniques for solving basis pursuit problems.* PhD thesis, North Carolina State University, 2006.

[7] Donoho, D. L. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics 59*, 6 (2006), 797–829.

[8] Erlangga, Y. A., Vuik, C., and Oosterlee, C. W. On a class of preconditioners for solving the helmholtz equation. *Applied Numerical Mathematics 50*, 3 (2004), 409–425.

[9] Hou, T. Y., and Wu, X.-H. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics 134*, 1 (1997), 169–189.

[10] Jamil, N. A comparison of direct and indirect solvers for linear systems of equations. *Int. J. Emerg. Sci 2*, 2 (2012), 310–321.

[11] Jin, J. *The finite element method in electromagnetics*. John Wiley & Sons, 2014.

[12] Kechroud, R., Soulaimani, A., Saad, Y., and Gowda, S. Preconditioning techniques for the solution of the helmholtz equation by the finite element method. *Mathematics and Computers in Simulation 65*, 4 (2004), 303–321.

[13] Knibbe, H., Oosterlee, C. W., and Vuik, C. Gpu implementation of a helmholtz krylov solver preconditioned by a shifted laplace multigrid method. *Journal of Computational and Applied Mathematics 236*, 3 (2011), 281–293.

[14] Le Montagner, Y., Angelini, E., and Olivo-Marin, J.-C. Comparison of reconstruction algorithms in compressed sensing applied to biological imaging. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on* (2011), IEEE, pp. 105–108.

[15] Liu, J., and Jin, J.-M. A highly effective preconditioner for solving the finite element-boundary integral matrix equation of 3-d scattering. *Antennas and Propagation, IEEE Transactions on 50*, 9 (2002), 1212–1221.

[16] Osei-Kuffuor, D., and Saad, Y. Preconditioning helmholtz linear systems. *Applied numerical mathematics 60*, 4 (2010), 420–431.

[17] Rui, P., Li, S., and Chen, R. Application of ssor preconditioned gmresr algorithm for fem analysis of helmholtz equations. In *Antennas and Propagation Society International Symposium 2006, IEEE* (July 2006), pp. 1773–1776.

[18] Saad, Y. Ilut: A dual threshold incomplete lu factorization. *Numerical linear algebra with applications 1*, 4 (1994), 387–402.

[19] Saad, Y. Iterative methods for sparse linear systems. *IEEE Computational Science and Engineering 3*, 4 (1996), 87.

[20] Saad, Y., and Schultz, M. H. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing 7*, 3 (1986), 856–869.

[21] Sayas, F.-J. A gentle introduction to the finite element method.

[22] Tropp, J., and Gilbert, A. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on 53*, 12 (Dec 2007), 4655–4666.

[23] Zhang, G., Jiao, S., Xu, X., and Wang, L. Compressed sensing and reconstruction with bernoulli matrices. In *Information and Automation (ICIA), 2010 IEEE International Conference on* (2010), IEEE, pp. 455–460.

[24] Zhang, J., and Shen, D. "gpu-based implementation of finite element method for elasticity using cuda. In *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on* (Nov 2013), pp. 1003–1008.