

Cylindrical RF Tomography

by

Kim Lam

B.A.Sc, University of British Columbia, 2001

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Masters of Applied Science

in

The Faculty of Graduate Studies

Department of Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

November 2003

© Kim Lam, 2003

Library Authorization

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Kim Lam Dec 10, 2003
Name of Author (please print) Date

Title of Thesis: Cylindrical RF Tomography

Degree: M.A.Sc. Year: 2003

Abstract

Tomography allows the examination of an object's interior without having to destroy the object. There have been many forms of tomography including MRIs and CT scans. All forms of tomography infer the interior of an object from a set of measurements. An algorithm to infer the features of infinite dielectric cylinders using electromagnetic waves is developed in this thesis. The intended application for this algorithm is the imaging of lumber.

The algorithm recovers the dielectric permittivity distribution from an infinite cylinder. It uses Richmond's Method to model the physical behavior of the infinite cylinder. The algorithm uses an iterative non-linear inversion scheme to recover the dielectric permittivity distribution. The non-linear inversion scheme uses a regularization term that minimizes the structure of the permittivity distribution subject to a constraint involving the measured scattered field from the cylinder.

It was found that by decreasing the weighting of the regularization, and eventually turning off the regularization, the exact permittivity distribution can be recovered for a noiseless and over-determined system. In the presence of noise, an approximate permittivity distribution can be recovered; however regularization cannot be turned off. Similarly, for an under-determined system, an approximate permittivity distribution can be found, but regularization cannot be turned off.

Table of contents

Abstract	ii
Table of contents	iii
List of figures	v
List of tables	vii
Chapter 1: Introduction	
1.0 Introduction	1
1.1 Thesis overview	8
1.2 Novel content	9
Chapter 2: Examination of a 2-D homogeneous cylinder	
2.0 Introduction	11
2.1 2-D scattering solution	11
2.2 How to compare observations of different permittivity values	15
2.3 Numerical results	15
2.4 Conclusion	22
Chapter 3: Mathematical background for the tomography algorithm	
3.0 Introduction	23
3.1 Richmond's Method	24
3.2 The development of the integral equation for Richmond's Method	24
3.3 The scattered field	26
3.4 Observations for inversion	26
3.5 Gridding for the Jacobian calculation	28
3.6 Derivation of the Jacobian for locations internal to the dielectric body	30
3.7 Derivation of the Jacobian Matrix for the scattered field	31
3.8 The Jacobian for multiple frequencies	32
3.9 The flatness term	33
3.10 The N^T system of linear equations	34
3.11 Conclusion	36
Chapter 4: Implementation of the tomography algorithm	
4.0 Introduction	37
4.1 Tessellating the cylinder	38
4.2 Calculating the flatness matrix	41
4.3 Calculating the internal field using Richmond's method	45
4.4 Generation of the observation matrix	45
4.5 The Jacobian Calculation	46
4.6 Inversion function with a single beta value	51
4.7 Inversion function with multiple beta values	54
4.8 Summary	56
Chapter 5: Numerical experiments using the tomography algorithm	
5.0 Introduction	57

5.1 Setup for the experimental procedure	58
5.2 Test cases 1,2 and 3	61
5.3 Norms for assessing the quality of the solution	64
5.4 Experimental set 1	65
5.5 Experimental set 2	71
5.6 Experimental set 3	76
5.7 Experimental set 4	81
5.8 Experimental set 5	85
5.9 Conclusion	88
Chapter 6: Summary and future work	
6.0 Summary	90
6.2 Future work	92
Bibliography	93

List of figures

Figure 1.1 Abstraction of the scattering from a dielectric body	1
Figure 1.2 Abstraction of forward modeling process	2
Figure 1.3 Schematic for linear inversion	2
Figure 1.4 Schematic for non-linear inversion	3
Figure 2.1 Geometry of the scattering problem	11
Figure 2.2 Definition of electromagnetic fields inside and outside of the cylinder.	12
Figure 2.3 Geometric configuration of observation points	16
Figure 2.4 Variation surface for permittivities from 1 to 10	17
Figure 2.5 Variation slices for a homogenous cylinder	18
Figure 2.6 Zeros of the $a_n(\epsilon_1) - a_n(\epsilon_2)$ for ϵ_1 equals 5	19
Figure 2.7 Multifrequency variation slices for a homogenous cylinder	21
Figure 2.8 Multifrequency variation surface for permittivities from 1 to 10	22
Figure 3.1 Arrangement of the multi-frequency observation vector	27
Figure 3.2 Illustration of the mapping concept	28
Figure 3.3 Structure of the multi-frequency Jacobian	33
Figure 4.1 Flow chart for the complete inversion system	37
Figure 4.2 Illustrative diagram of the discretization concept	39
Figure 4.3 Flow chart for the BasicInversion function	51
Figure 5.1 Inversion cells for Experimental sets 1, 2 and 3	59
Figure 5.2 Modeling cells for Experimental sets 1, 2 and 3	60
Figure 5.3 Superposition of modeling regions and inversion regions	61
Figure 5.4 Test case 1	62
Figure 5.5 Test case 2	63
Figure 5.6 Test case 3	64
Figure 5.7 Variation plot for experimental set 1	66
Figure 5.8 Modeling error plot for experimental set 1	67
Figure 5.9 Recovered permittivity distribution for experimental set 1, test case 2 and $\beta = 0$	68
Figure 5.10 Recovered permittivity distribution for experimental set 1, test case 2 and $\beta = 0.001$.	69
Figure 5.11 Variation as a function of iteration for experimental set 1, test case 2, $\beta = 0$ and $\beta = 0.001$	70
Figure 5.12 Variation plot for experimental set 2, method 1	71
Figure 5.13 Variation plot for experimental set 2, method 2	72
Figure 5.14 Modeling for experimental set 2, method 1	73
Figure 5.15 Modeling for experimental set 2, method 2	74
Figure 5.16 Image of permittivity distribution experimental set 2, test case 3, method 1, and iteration 7.	75
Figure 5.17 Image of permittivity distribution	

experimental set 2, test case 3, method 2, and iteration 21.	76
Figure 5.18 Variation plot for experimental set 2 (20% noise), test case 3	77
Figure 5.19 Modeling error plot for experimental set 2 (20% noise), test case 3	78
Figure 5.20 Permittivity distribution for experimental set 3, test case 3, and iteration 30	79
Figure 5.21 Permittivity distribution for experimental set 3, test case 3, and iteration 31	80
Figure 5.22 Permittivity distribution for experimental set 4	81
Figure 5.23 Variation plot for experimental set 4, experiment 1	83
Figure 5.24 Modeling error plot for experimental set 4, experiment 1	84
Figure 5.25 Permittivity distribution for experimental set 5	85
Figure 5.26 Variation plot for experimental set 5, experiment 1	86
Figure 5.27 Modeling error plot for experimental set 5, experiment 1	87
Figure 5.28 Permittivity distribution for experimental set 5, experiment 1, iteration 41	88

List of tables

Table 2.1 Local minimum values at 4 discrete points	18
Table 5.1 Summary of experiments for chapter 5	57
Table 5.2 Parameters for the experimental sets 1, 2 and 3	58
Table 5.3 Parameters for the experimental set 4	82

Chapter 1 Introduction

1.0 Introduction

There has been much interest in microwave imaging (MI) otherwise known as microwave tomography. The Greek root ‘tomo’ means to slice or section, and tomography is a general term for obtaining an image of a slice of a target. Tomography techniques such as CT scans and MRIs are routinely used in medicine [Hendee and Wells, 1997].

MI is based on the measurement of a scattered field from an object illuminated by electromagnetic (EM) waves. An object is first illuminated with a microwave source. This results in secondary waves, known as the scattered field. MI is the recovery of the permittivity distribution of an object, based on measurements of the scattered field. To recover the distribution, an inversion algorithm is needed. The process of inversion recovers the parameters of an operator when the operator and the scattered field are known. Figure 1.1 shows a schematic of the physical system. Figure 1.2 illustrates the forward modeling problem. The differential operator simulates the physical system mathematically and can simulate the scattered field, when operating on the incident field. The first step in Figure 1.2 illustrates that a mathematical description of the entire system must be generated. It is then inverted to produce an operator that relates the incident field to the scattered field. The second step is to obtain the scattered field.

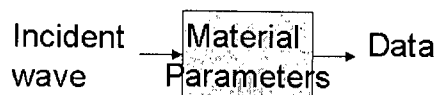
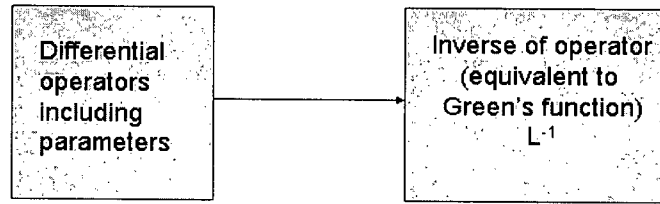


Figure 1.1: The diagram represents the abstraction of the physical system. The incident wave interacts with the Physical system, denoted by the “Material Parameters” box, and the resulting electromagnetic fields are measured.

Step 1:



Step 2:

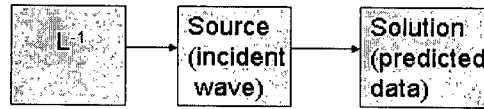


Figure 1.2: Schematic of the forward modeling process. A differential operator models the characteristics of the physical system. This operator is then inverted and applied to the incident wave to generate a scattered field.

Inversion can be separated into linear or non-linear inversion. The linear inverse for MI is obtained by discretization of the integral operator to obtain a linear vector-matrix relationship between the data and the parameters. A generalized matrix inversion is then performed to obtain the parameters. The generalized matrix inversion minimizes an objective function typically interpreted to be a best linear fit, while minimizing a square error. The minimization occurs in a single step. In the case of a square matrix the solution is exact and the objective function is zero.

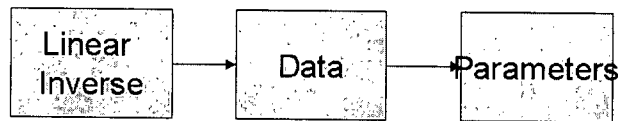


Figure 1.3: Schematic for the linear recovery of the material parameters. A linear inverse is first taken. This is applied to the measured scattered field to recover the permittivity values.

For non-linear inversion, a direct matrix mapping from data to the parameters cannot be obtained. For this type of inversion an objective function is, also, defined and is either minimized or maximized. This technique is iterative. The objective function approaches its minimum after each iteration. The minimum is reached after several iterations. During each iteration, the parameters are estimated by examining some

property of the objective function, either its value or its derivative. Based on the results of the objective function, the inversion algorithm determines the next set of estimated parameters. This set of parameters leads to a better solution. Figure 1.4 illustrates schematically the results of an iteration.



Figure 1.4: Schematic for non-linear inversion. The Non-Linear operator acts upon the data to give the parameters. The non-linear operator may be iterative in nature or may have multiple parameters as possible solutions.

MI can be applied to many problems. Much recent literature [Hagness et al., 1998][Semenov et al., 2002] points to medical imaging as a promising area of study. To image biological objects, microwaves travel through material with high dielectric permittivity and high dielectric contrast. A particular biological application of importance is the imaging of trees. Regions of high dielectric permittivity indicate areas of rot, due to the accumulation of water. Therefore, the location of these regions can be found using microwave tomography providing in-vivo and in-situ assessment of the wood quality of the target tree. The author of this thesis is unaware of any previous successful attempts to apply MI to trees.

When a microwave travels into an object, part of the wave is trapped inside the object, while part of the wave is scattered from the object. The trapped waves constantly reflect off the boundary of the object. If the object is inhomogeneous, the reflections within the object become more complicated. The trapped waves continuously leak out of the object. These leaked waves form part of the scattered field. The scattered field is, therefore, composed of the leaked and scattered waves.

Many techniques have been used to solve the inverse scattering problem, with varying degrees of success. The Born approximation is the simplest imaging method. The Born approximation approximates the internal field of the scatterer with that of the incident field. The total field, exterior to the scatterer, is then assumed to be a linear combination of the incident field and a scattered field. The inverse operator can be calculated using the assumed incident field, yielding a linear inversion algorithm which is solved by a matrix inverse, as described previously. The Born approximation fails when a moderate to high dielectric contrast exists [Slaney et al., 1984]. This failure occurs because of the strong scattering behavior leading to an internal field that is quite different than that of the incident field. This behavior is due to the scatterer behaving like a leaky waveguide structure.

More recently, imaging using high bandwidth chirp pulses, and confocal imaging, has been studied [Hagness et al., 1998]. The technique of confocal imaging is based on the idea that the microwave chirp behavior approximates that of rays. Simulations of confocal imaging indicate that structures as small as 2mm [Hagness et al., 1998] can be detected inside breast tissue. The work on confocal imaging has largely been related to breast cancer detection.

One of the first researchers in MI [Joachimowicz et al. 1991] applied non-linear inversion techniques to an EM inverse scattering problem used to detect bone and muscle tissue. Joachimowicz et al. used a Newton-Kantorovich's procedure to find the minimum of an objective function. Their process was iterative and used multiple illumination angles of the object.

Caorsi develops a number of inversion algorithms. Caorsi et al. also builds a physical system for imaging. He uses simulated annealing (SA) [Caorsi et al, 1991] for one of his inversion procedures. The SA method recovers an approximate permittivity distribution, while using multiple illumination angles. Caorsi, also, experiments using a pseudo-inverse technique as an inversion scheme [Caorsi et al., 2003]. The pseudo-inverse is the analog of an inverse for a non-square matrix [Golub and Van loan, 1996][Wilkinson and Peter, 1970].

Chew and Lin [Chew and Lin, 1995] explore the use of multiple frequencies for dielectric imaging. Chew and Lin hypothesize that the Born approximation is good for low frequencies. Chew uses the Born approximation to find an approximate distribution at low frequencies and then moving to an optimization scheme at higher frequencies, Chew uses results from a low frequency imaging technique as a starting point for a higher frequency technique. Since only one frequency is used at a time, their technique is known as “frequency hopping”. For low frequencies, Chew and Lin uses the distorted Born method [Wang and Chew, 1989] and for high frequencies the conjugate gradient method is used [Borup and Gandi, 1985]. The distorted Born method assumes that the internal field is dependent on the permittivity distribution, compared to the Born method where the internal field is assumed to be that of the incident field. In the Born method, the Green’s function for a homogenous background is used, while the distorted Born method employs an approximate Green’s function for an inhomogeneous background. Chew and Lin quantitatively recover the permittivity distribution when an object is submerged in water. Submerging the object in water reduces the contrast between the

background and the object, leading to less reflection from the surface of the object, resulting in very limited trapped wave energy.

In 1998, Pastorino et al. [Pastorino et al., 1998] experiments with using genetic algorithms for solving the inverse scattering problem. He, also, presents an overview of imaging technology for MI. A genetic algorithm is an optimization technique based on the concept of mutation and a fitness function. The fitness function is similar to the objective function previously described and is used to determine the quality of the solution. To obtain a better solution, many sets of random changes are introduced. The set of random changes with the highest fitness value is the solution selected as the starting point for the next iteration.

More recently, Semenov et al. [Semenov et al., 2002] builds a 3-D tomography system based on earlier work on 2-D imaging [Souvorov et al., 1998]. It was noted that a 2-D slice method is inappropriate for modeling 3-D structures and an intrinsically 3-D algorithm must be used. Semenov et al. are capable of recovering an approximate distribution of a slice of a dog using their 2-D and 3-D algorithms. A modified Newton iterative scheme is used for their 2-D imaging method [Souvorov et al., 1998]. A gradient method for is used for the 3-D imaging method [Semenov et al., 1999].

Lin et al [Lin and Chui., 2001] produced a 2 step algorithm, similar to that of Chew and Lin [Chew and Lin., 1995]. Lin's method differs from Chew's method because frequency hopping is not used. Lin separates the problem into a linear step and a non-linear step. The Lin algorithm uses the distorted Born approximation to recover an approximate distribution of the dielectric permittivity for the linear step. The approximate distribution is used as a starting point for the conjugate gradient algorithm to

find a detailed solution. The conjugate gradient algorithm uses the full non-linear formulation of the EM problem to solve the inverse problem, resulting in a better solution. Their experimental setup requires that the body be placed in a matching fluid. Their solution allows the recovery of inhomogeneities in the body.

Caorsi [Caorsi et al., 2003] recently introduced a multi-scaling scheme that can be applied to any iterative inversion technique. The technique starts with a coarse grid; when an anomaly is detected in a square quadrant of the grid, that quadrant is discretized more finely. Success is achieved in recovering a number of permittivity distributions by coupling this scheme with the conjugate gradient method.

The work presented here is different from that of previous work in several regards. First, this work is inspired by work done in the field of geophysics inversion, specifically the work done by the University Of British Columbia Geophysical Inversion Facility (UBC GIF). Second, this work is not focused on medical imaging; instead imaging of trees is the intended application. The imaging of trees poses a difficult problem because of the existence of a high dielectric contrast boundary at the surface of the tree. This creates two problems. The first problem is the large amount of reflection at the boundary of the tree. The second problem is the tendency for waves to be trapped inside the tree. The inclusion of the trapped waves in the inversion algorithm represents a significant departure from previous work [Joachimowicz et al., 1990].

The UBC GIF is an academic research unit focusing on the development of geophysical forward modeling and inversion methodologies. Their work is based on physical and intuitive ideas for the use of inversion. The UBC GIF view the minimization problem as a problem of minimizing the structure of the solution using the

data as constraints. This work applies the regularization term of “Minimizing Structure” [Farquharson and Oldenburg, 1998] introduced in geophysical optimization.

Farquharson goes into extensive detail on L2 norms, and non-L2 norms. This work is limited to L2 norms. Minimizing structure in the solution is an idea that has an associated physical interpretation. Large structural variations, measured by the norm of the first order derivatives, are penalized, resulting in more plausible models. For this work, an objective function based on a linear combination of the norm of the structure of the system and the norm of the difference between predicted data and measured data is defined for this work. This effectively incorporates the idea of minimizing structure, subject to data constraints, into the inversion algorithm. A linear approximation, based on the Taylor expansion of the objective function, will be generated. The linear approximation will be minimized. By repeating this process many times, an iterative process is created to find the minimum of the objective function.

Previous regularization techniques, in fields outside geophysics, used purely mathematical regularization techniques to eliminate the ill-posedness of the problem. Previous work is MI [Ciric and Qin, 1997][Tanaka et al., 2001] viewed the problem as one of minimizing the difference between the measured scattered field and guessed scattered field. These techniques have little relationship to the actual physical system.

1.1 Thesis overview

Chapter 2 will present the solution to scattering from a homogenous circular cylinder illuminated by a plane wave. The solution is based on the Bessel mode expansion of the problem. An odd scattering behavior is presented. Chapter 2 provided a great deal of insight into the problem of scattering.

Chapter 3 presents all the mathematics necessary to perform the inversion. This includes Richmond's Method [Richmond, 1965] for forward modeling. Richmond's Method is used to obtain the scattered field. Next, the Jacobian calculation is derived from Richmond's Method. The concept of "Minimizing Structure" is formally introduced in matrix form as the flatness matrix. The objective function is introduced that includes the flatness matrix and the scattered field. The Jacobian matrix is used to linearize the scattered field and finally, a method to linearize the objective function is shown.

Chapter 4 demonstrates the implementation of the algorithm using MatLab. The mapping between Chapter 3 and the source code is presented.

Chapter 5: presents results, using the algorithm. This chapter explores the behavior of the algorithm and presents some very promising results. First, the algorithm uses a constant scaling to the regularization. This demonstrates that regularization helps to find the correct solution. Next, decreasing the scaling of the regularization as the iterative process proceeds. By decreasing the scaling of the regularization, the system is allowed to assume more structure and is analogous to slowly cooling the system in simulated annealing [Caorsi et al., 1991]. The most novel result is that the solution slowly emerges as the amount of regularization is decreased. The algorithm recovers the permittivity distribution in noisy data and in an under constrained system.

1.2 Novel content

This thesis demonstrates the successful development of a tomography algorithm, by using electromagnetic waves. The regularization technique is taken from geophysical inversion and has not been applied the electromagnetic scattering problem. The

regularization enforces a flat starting model, reducing the choices of starting model. This represents a significant improvement over previous algorithms, where the problem of convergence has been solved, but the equally complicated starting model problem has not been solved. A novel unified modeling-inversion notation has been created through the use of indicator matrices.

This thesis, also, introduces a previously unknown scattering behavior. Scattering from homogeneous cylinders with different permittivities can have very similar scattered fields. The difference in permittivities that yields very similar scattered field follows a regular pattern. The approximate non-uniqueness of the scattered field is resolved by using multiple frequencies.

Chapter 2: Examination of a 2-D homogeneous cylinder

2.0 Introduction

This chapter examines scattering from a homogenous dielectric circular cylinder, illuminated by a TM wave. This problem is examined to give insight into scattering from an inhomogeneous cylinder. Initially, the solution to scattering from a cylinder is presented. Next, the similarity between the scattering patterns of different permittivity values is examined. This provides insight into the inverse problem.

2.1 2-D Scattering Solution

Incoming plane wave:

$$\vec{E}^{inc} = E_z^{inc} \exp\left(-j \frac{2\pi x}{\lambda} + j\omega t\right) \hat{z} = E_z^{inc} \exp\left(-j \frac{2\pi \rho}{\lambda} \cos\phi + j\omega t\right) \hat{z}$$

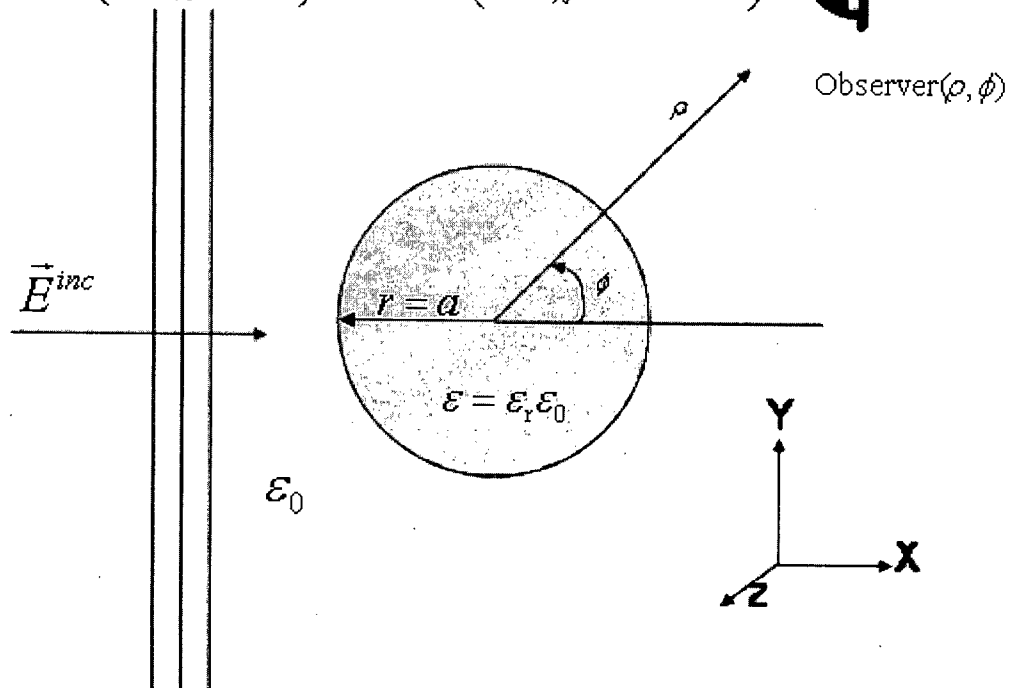


Figure 2.1: Geometry of the scattering problem.

The geometry of the system is shown in Figure 2.1. The origin, for both Cartesian coordinates and polar coordinates, is at the center of the cylinder. For derivation, the time factor $e^{j\omega t}$ is assumed and phasor quantities are used. A TM wave, where the \vec{E} field is parallel to the cylinder axis, is assumed and can be written as Equation 2.1 with k , the wave number, and E_z^i , the amplitude of the z component of the incident field.

$$\vec{E}^i(x, y) = E_z^i e^{-jkx} \hat{z} = E_z^i e^{-ik\rho \cos(\phi)} \hat{z} \quad (2.1)$$

Because the problem involves cylindrical geometry, rewriting Equation 2.1 in a Bessel function expansion makes the problem easier to solve. Rewriting Equation 2.1 as a Bessel function expansion, and using polar coordinates Equation 2.2 is obtained [Harrington, 1961].

$$E^i(\rho, \phi) = E_z^i \sum_{n=-\infty}^{\infty} j^{-n} J_n(k\rho) e^{jn\phi} \hat{z} \quad (2.2)$$

The scattering problem can be solved as a boundary value problem. The boundary is defined as the surface of the cylinder. At infinity, the scattered field is expected to be zero.

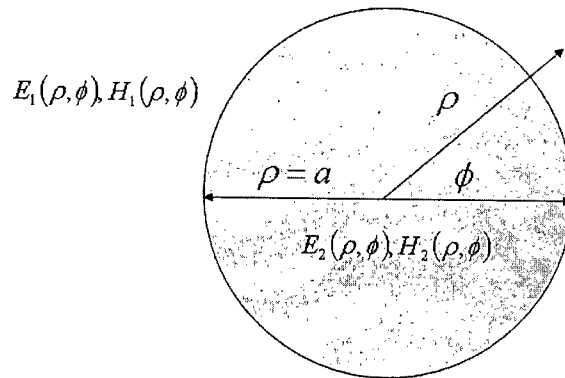


Figure 2.2: Definition of electromagnetic fields inside and outside of the cylinder.

The boundary conditions are presented in Equation 2.3a and Equation 2.3b and are applied at the boundary of the cylinder. The tangential components of the total electric(E) and magnetic(H) field must be continuous across the boundary of the cylinder.

$$\hat{n} \times (\vec{H}_1(a, \phi) - \vec{H}_2(a, \phi)) = 0 \quad (2.3a)$$

$$\hat{n} \times (\vec{E}_1(a, \phi) - \vec{E}_2(a, \phi)) = 0 \quad (2.3b)$$

Equation 2.4 is a convenient representation of the resulting field with a known and an unknown portion. The Incident field is the known portion and the scattered field is the unknown portion.

$$\vec{E}_1(\rho, \phi) = \vec{E}^{inc}(\rho, \phi) + \vec{E}^{scat}(\rho, \phi) \quad (2.4)$$

Upon examining Equation 2.3a and Equation 2.3b, it appears that there are 4 unknowns and 2 boundary conditions. To reduce the number of unknowns to 2, Equation 2.5a is used to link E and H . The explicit relationship in polar coordinates is given in Equation 2.5b.

$$\nabla \times \vec{E} = -j\omega\mu(\omega)\vec{H} \quad (2.5a)$$

$$\vec{H} = \frac{-1}{j\omega\mu(\omega)} \left(\frac{1}{\rho} \frac{\partial E}{\partial \phi} \hat{\rho} - \frac{\partial E}{\partial \rho} \hat{\phi} \right) \quad (2.5b)$$

The scattered field is assumed to be written as Equation 2.6 [Harrington, 1961], with unknown coefficients a_n and the total field exterior to the cylinder satisfies Equation 2.4. The total field inside the cylinder can be represented as Equation 2.7a with the unknown coefficient c_n [Harrington, 1961]. In Equation 2.7b, ϵ_r is the permittivity inside the cylinder.

$$\vec{E}_1^s = E_z^i \sum_{n=-\infty}^{\infty} j^{-n} a_n H_n^{(2)}(k\rho) e^{jn\phi} \hat{z} \quad (2.6)$$

$$\vec{E}_2 = E_z^i \sum_{n=-\infty}^{\infty} j^{-n} c_n J_n(k_d \rho) e^{jn\phi} \hat{z} \quad (2.7a)$$

$$k_d = k\sqrt{\epsilon_r} \quad (2.7b)$$

By applying the TM assumption, Equation 2.3b can be rewritten as Equation 2.9a which immediately implies equation 2.9b because the unit vector $\hat{\phi}$ cannot be zero.

$$\hat{n} \times (E_1 - E_2) \hat{z} = \hat{\rho} \times (E_1 - E_2) \hat{z} = -(E_1 - E_2) \hat{\phi} = 0 \quad (2.9a)$$

$$E_1 - E_2 = 0 \quad (2.9b)$$

Applying Equation 2.5b to Equation 2.3a, gives Equation 2.10. The normal vector to a circle is $\hat{\rho}$. The term involving $\hat{\rho}$ in Equation 2.3a disappears because of the $\hat{n} \times$ operation.

$$\frac{\partial E_1}{\partial \rho} - \frac{\partial E_2}{\partial \rho} = 0 \quad (2.10)$$

By plugging Equations 2.6 and 2.7a into Equation 2.9 and Equation 2.10, a system of 2 equations with 2 unknowns is produced. The 2 equations are Equation 2.11a and Equation 2.11b.

$$a_n H_n^{(2)}(ka) + J_n(ka) - c_n J_n(k_d a) = 0 \quad (2.11a)$$

$$a_n k H_n^{(2)'}(ka) + k J_n'(ka) - c_n k_d J_n'(k_d a) = 0 \quad (2.11b)$$

The explicit solution for a_n is given by Equation 2.12. The only coefficient needed is a_n because the solution to the field inside the cylinder is not needed.

$$a_n = \frac{-J_n(ka)}{H_n^{(2)'}} \left[\frac{\varepsilon_d J_n'(k_d a) / k_d a J_n(k_d a) - J_n'(ka) / ka J_n(ka)}{\varepsilon_d J_n'(k_d a) / k_d a J_n(k_d a) - H_n^{(2)'}(ka) / ka H_n'(ka)} \right] \quad (2.12)$$

2.2 How to compare observations of different permittivity values

This thesis is concerned with the inverse scattering problem. The similarities between the scattered fields are now examined as a function of permittivity, yielding insight into the inverse problem. The scattering field is expected to be very similar for permittivity values that are close together and significantly different for permittivity values that are far apart. The variation of the scattered field at different permittivity values is quantified in Equation 2.13, where M is the number of observations points.

$$\Phi_g = \frac{\sum_m^M |E_m^s(\varepsilon_1) - E_m^s(\varepsilon_2)|^2}{M} \quad (2.13)$$

Equation 2.13 is clearly small when two scattering fields are similar.

2.3 Numerical results

Equation 2.13 is examined in this section. The geometric configuration of the observation is a circle. The configuration is shown in Figure 2.3.

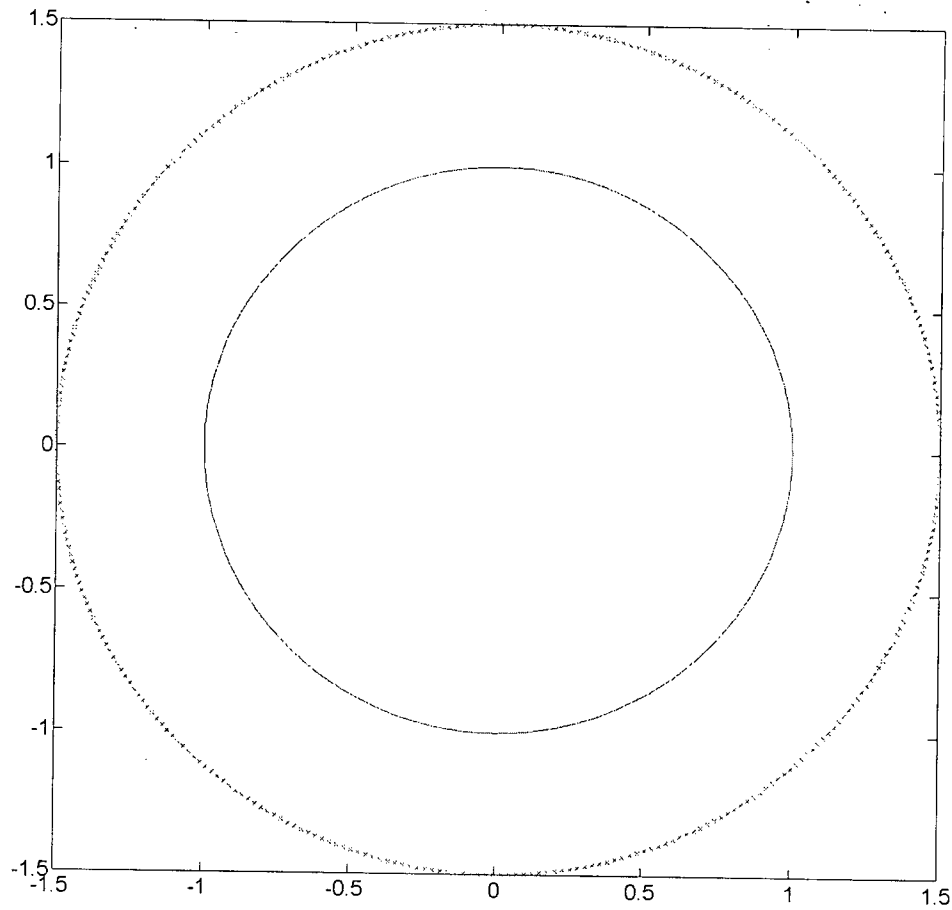


Figure 2.3: Geometric configuration of observation points. There are 360 observations points spaced 1 degree apart.

Figure 2.4 is a 2-D variation surface computed using Equation 2.13. The x and y axes represents ε_1 and ε_2 in Equation 2.13, respectively. The line, where ε_1 equals ε_2 , represents the global minimum. There are also several bands of local minima approximately parallel to the ε_1 equals ε_2 line.

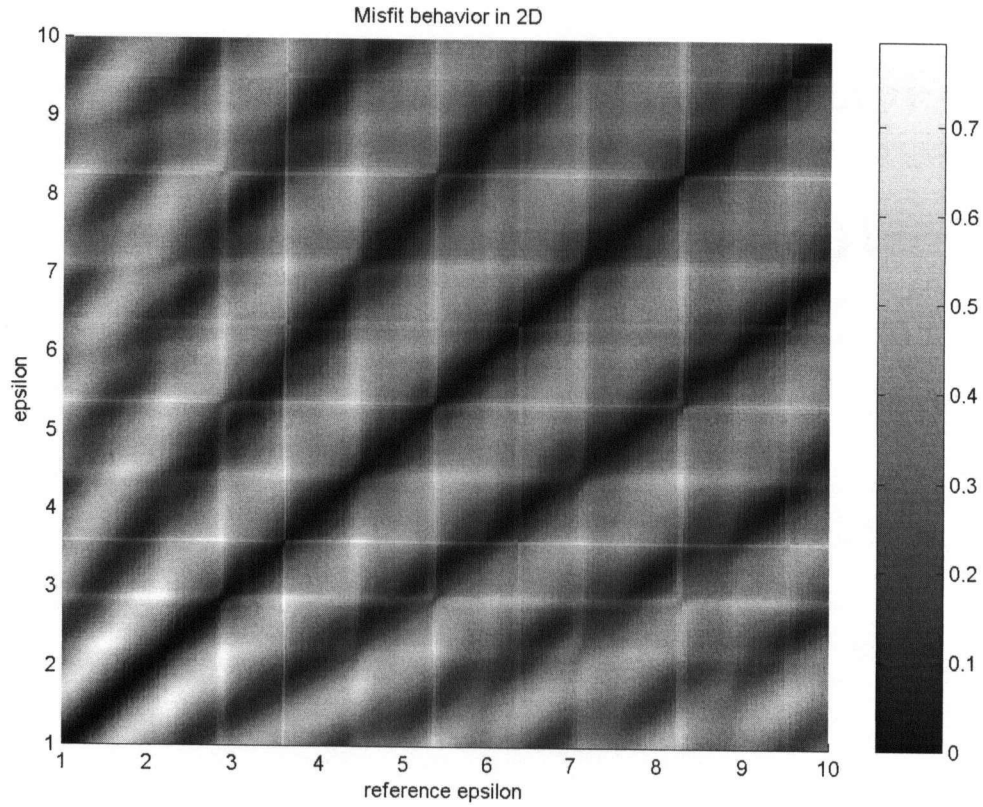


Figure 2.4: Variation surface for permittivities from 1 to 10. The shading represents the variation value. The line, where ε_1 equals ε_2 , is the global minimum.

Figure 2.5 shows slices of Figure 2.4 by fixing ε_1 and varying ε_2 from 1 to 10. Examination of a slice of the variation surface at ε_1 equals 9.9 shows the existence of local minima; there are two very deep local minima. The variation equals 0.0359 at the closest local minimum. Because 0.0359 is small, when compared to the maximum variation of approximately 0.8, the two scattered fields are very similar. The foregoing case of ε_1 equals 9.9 is shown in the bottom right corner of Figure 2.5. The other variation plots exhibit a similar behavior which is tabulated in Table 2.1.

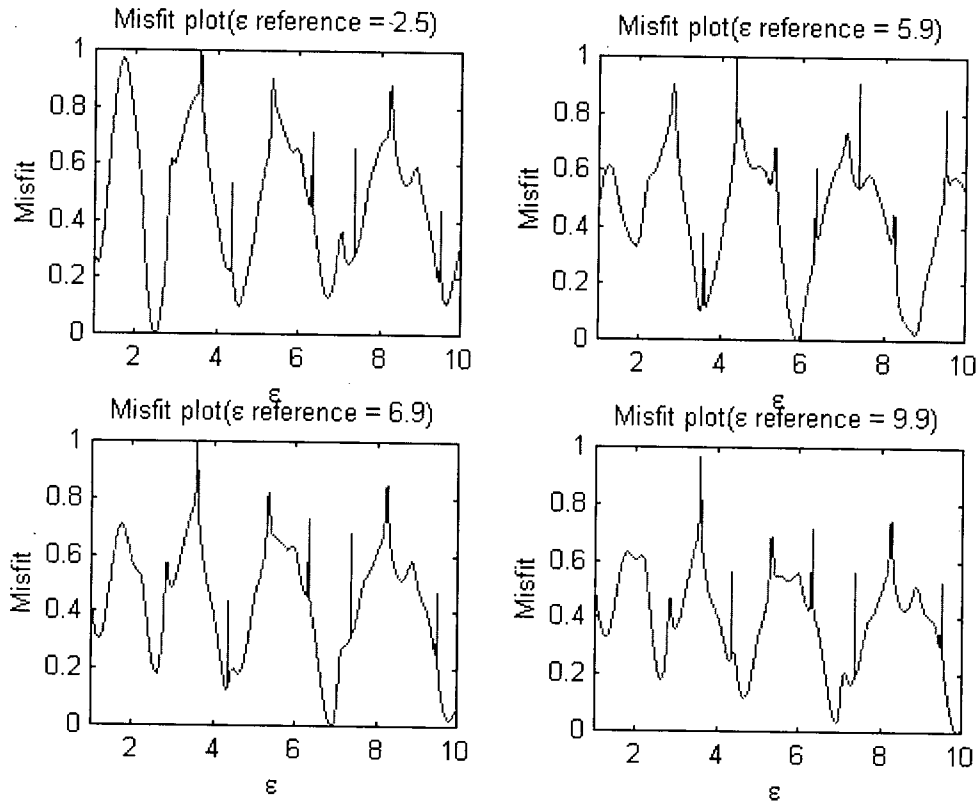


Figure 2.5: Variation slices for a homogeneous cylinder of radius 1λ and observation distance of 1.5λ . Top left, cylinder with permittivity of 2.5, top right, cylinder with permittivity of 5.9, bottom left, cylinder with permittivity 6.9, bottom right, cylinder with permittivity 9.9.

The local minima are problematic for inversion using gradient descent search techniques. Choosing a starting permittivity value far from the true permittivity value, the search algorithm may get trapped inside a local minimum.

Table 2.1 Local minimum values at 4 discrete points.

Permittivity	Deepest local minimum
2.5	0.0997
5.9	0.0218
6.9	0.0225
9.9	0.0359

Equation 2.7 is identical to that of a Discrete Time Fourier Transform(DTFT) if ρ is fixed, ϕ replaces the frequency variable and a_n replaces the coefficient in the DTFT.

The DTFT has a one-to-one mapping between the infinite set of coefficients and the

continuous signal. For Equation 2.13, if the two minima are exactly zero then Equation 2.14a must be true for all n values. By combining Equation 2.12 and Equation 2.14a, and simplifying, Equation 2.14b is produced.

$$a_n(\varepsilon_1) - a_n(\varepsilon_2) = 0 \quad (2.14a)$$

$$\frac{1}{\delta_1} \frac{J_n(\alpha\delta_1)}{J'_n(\alpha\delta_1)} = \frac{1}{\delta_2} \frac{J_n(\alpha\delta_2)}{J'_n(\alpha\delta_2)} \quad (2.14b)$$

$$\alpha = ka \quad (2.14c)$$

$$\delta_1 = \sqrt{\varepsilon_1} \quad (2.14d)$$

$$\delta_2 = \sqrt{\varepsilon_2} \quad (2.14e)$$

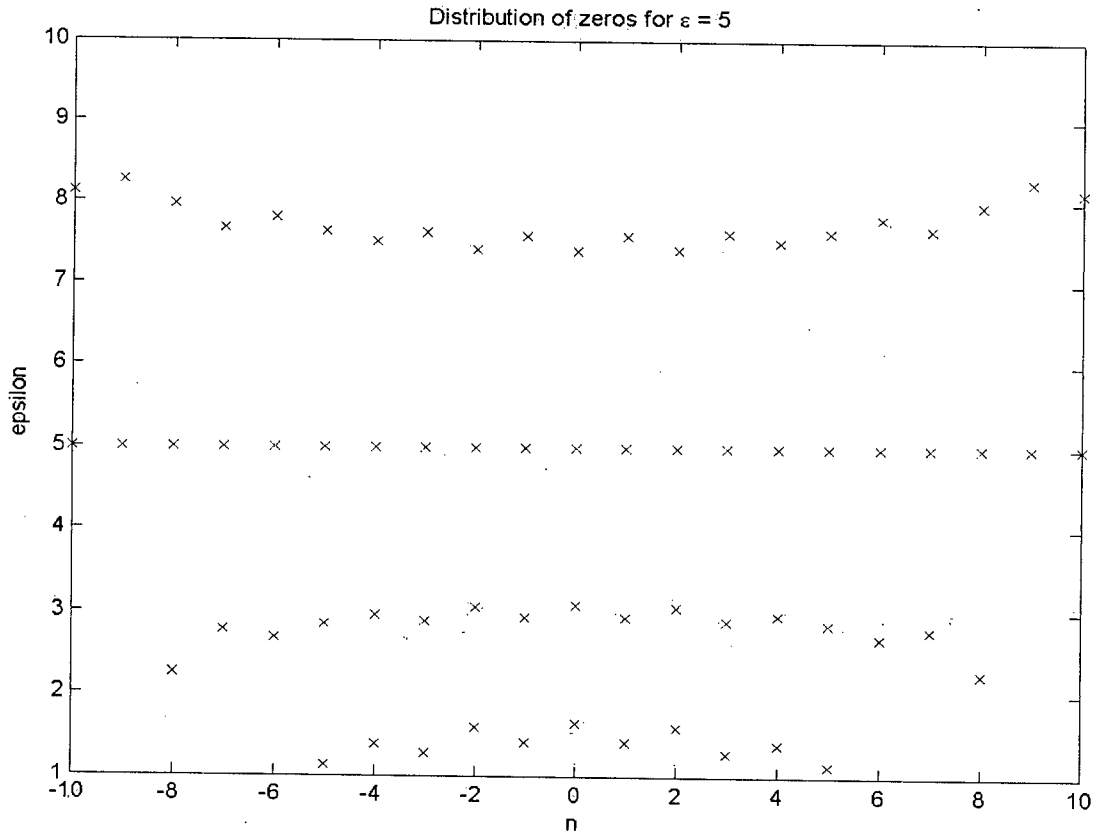


Figure 2.6: Zeros of $a_n(\varepsilon_1) - a_n(\varepsilon_2)$ for ε_1 equals 5. The x's mark the location where the equation is satisfied.

Figure 2.6 is an examination of Equation 2.14a with fixed ε_1 . Note that, ε_2 does not line up for different n indexes. However, they are close. This implies the local minima behavior seen in Figure 2.5.

The deep local minima are problematic for inversion. The multiple minima imply that, in the presence of noise, uniquely determining the permittivity value may not be possible.

By adding the variation from multiple frequencies the contrast between the local minima and global minimum can be increased.

$$\Phi = \sum_n^N \Phi_g(f_n) \quad (2.15)$$

Figure 2.7 illustrates the use of multiple frequencies. Three values of k are used. The lowest values is 20% below the highest value. The local minimums are significantly higher than those shown in Figure 2.5. For the total variation, the global minimum stays fixed.

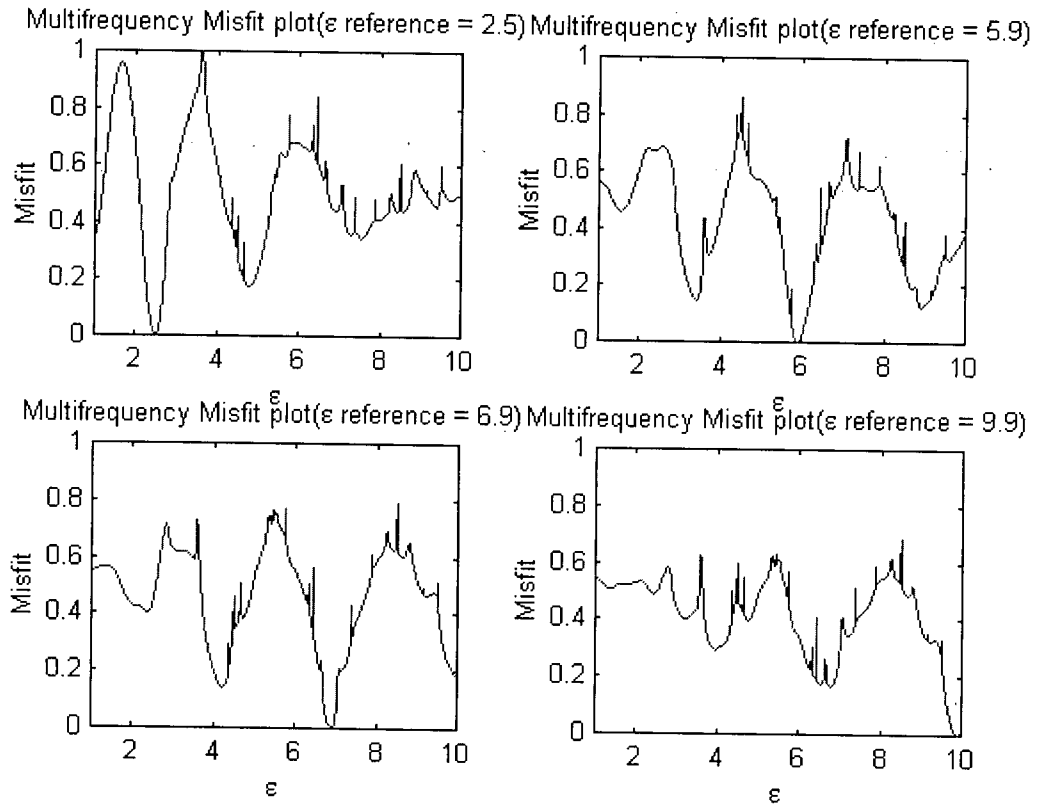


Figure 2.7: Multifrequency($k=1.6\pi, 1.8\pi, 2\pi$) variation slices for a homogenous cylinder of radius 1λ and observation distance of 1.5λ . Top left, cylinder with permittivity of 2.5, top right, cylinder with permittivity of 5.9, bottom left, cylinder with permittivity 6.9, bottom right, cylinder with permittivity 9.9

The advantage of using multiple frequencies is illustrated more dramatically in Figure 2.8.

Figure 2.8 is a variation surface, similar to that of Figure 2.4, but uses multiple frequencies. Figure 2.4 uses size frequencies, and all the deep local minimums have been eliminated.

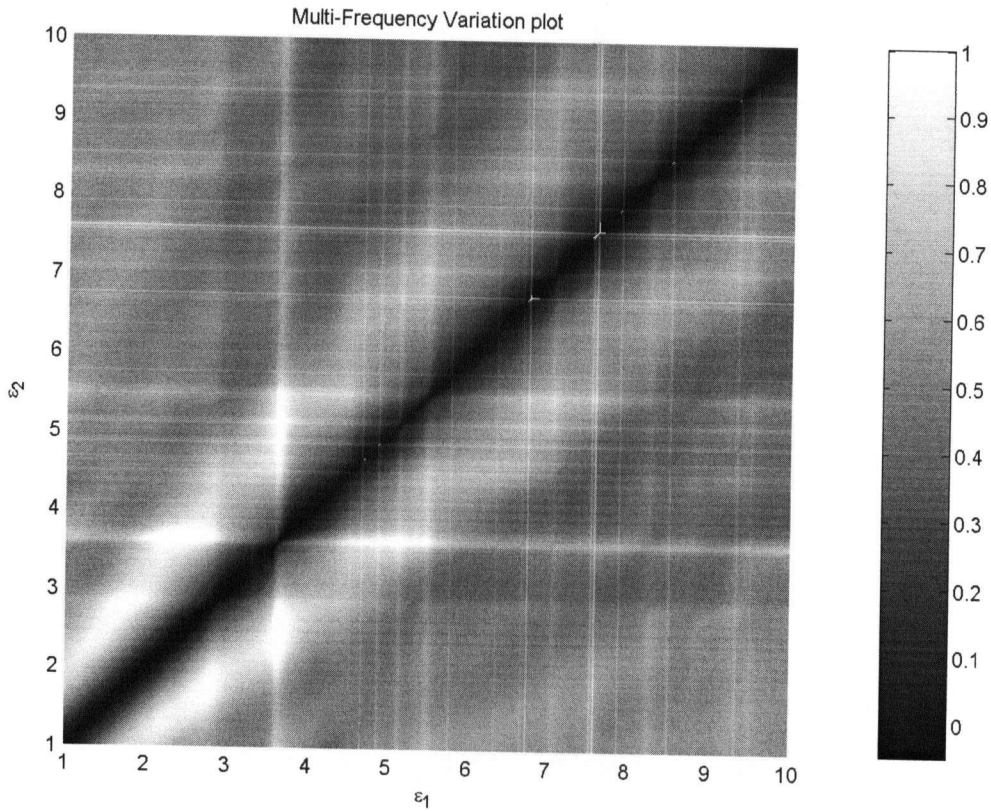


Figure 2.8: Multifrequency ($k = \pi, 1.2\pi, 1.4\pi, 1.6\pi, 1.8\pi, 2\pi$) variation surface for permittivities from 1 to 10.

2.4 Conclusion

This chapter shows that the use of multiple frequencies is necessary for the recovery of the dielectric constant of a homogenous cylinder. Without the use of multiple frequencies, scattering patterns from different permittivities may look very similar. In the presence of noise, these scattering patterns will be indistinguishable. Using more than one frequency helps make the scattering pattern from different permittivities more distinguishable. Usage of multiple frequencies changes the variation in such a way that the gradient search technique is more likely to work.

Chapter 3: Mathematical background for the tomography algorithm

3.0 Introduction

This chapter is separated into 2 main parts. The first part develops the mathematical background needed for substitution into Equation 3.1. The second part develops a constrained set of linear equations from Equation 3.1. The solution to $\Delta\vec{\epsilon}'$ can be found, once the linear system is developed. The solution for $\Delta\vec{\epsilon}'$, which forms the basis for the inversion algorithm, is given by

$$\min_{\Delta\vec{\epsilon}'} \left(\Phi = \frac{\|\vec{d}^{ref} - \vec{d}(\vec{\epsilon}') - \underline{\underline{J}}\Delta\vec{\epsilon}'\|^2}{N^f N^o} + \frac{\beta \|\underline{\underline{F}}(\vec{\epsilon}' + \Delta\vec{\epsilon}')\|^2}{N^i} \right) \quad (3.1)$$

In Equation 3.1, $\underline{\underline{J}}$, $\underline{\underline{F}}$ and \vec{d} need to be mathematically derived. \vec{d}^{ref} refers to a measured scattered field. $\underline{\underline{J}}$ and \vec{d} are derived from Richmond's Method for calculating scattering for an infinite dielectric cylinder. \vec{d} represents the scattered field, and $\underline{\underline{J}}$ represents the Jacobian matrix for the scattered field. $\underline{\underline{F}}$ represents a measure of flatness, and is known as a flatness matrix. Equation 3.1 represents the minimization of the difference between a measured scattered field and a guessed scattered field, as represented by the first term, and penalizes large variations of the dielectric permittivity.

In Equation 3.1, N^i represents the number of inversion cells, N^o represent the number of observation locations, and N^f represents the number of frequencies measured at each observation location. Each observation is defined by its frequency, resulting in multiple frequency observations at a particular location.

All the values in Equation 3.1 are real. Because time-harmonic electric fields are complex, the electric fields are separated into real and imaginary parts for the inversion, giving rise to a completely real Equation 3.1. This avoids the need for the complex conjugate operation in Equation 3.1.

3.1 Richmond's Method

Richmond's Method [Richmond, 1965] is used to find the scattered field for a single frequency. Richmond's Method is based on the discretization of an integral equation and enforcing the integral equation at a fixed set of points to produce a set of linear equations, leading to a constrained linear system.

3.2 The development of the integral equation for Richmond's Method

Richmond's Method constructs the integral equation by first stating the total electric field as a sum of a field due to the dielectric body, the scattered field, \vec{E}^s , and the incident field, \vec{E}^i . The dielectric body is replaced by a current distribution using the volume equivalence theorem [Harrington, 1961]. The replacement procedure is accomplished with Equation 3.2a, which represents the equivalent current of a point in the dielectric filament. The volume equivalence theorem states that a dielectric volume with electric field \vec{E} gives the same radiated field as a current written in Equation 3.2a.

$$\vec{J} = j\omega(\epsilon - \epsilon_0)\vec{E} = j\omega(\epsilon_r\epsilon_0 - \epsilon_0)\vec{E} = j\omega\epsilon_0(\epsilon_r - 1)\vec{E} \quad (3.2a)$$

$$dE^s(\rho) = -j\omega dA = -j\omega\mu \left[\frac{1}{4j} H_0^{(2)}(k\rho) \right] J dS' \quad (3.3a)$$

$$\rho(x, y) = \sqrt{(x - x')^2 + (y - y')^2} \quad (3.3b)$$

Associated with each current filament is a radiated field, shown in Equation 3.3a. Integrating over the current distribution gives an integral equation for the scattered field shown in Equation 3.4.

$$E^s(x, y) = -\left(jk^2 / 4\right) \iint (\epsilon_r - 1) E(x', y') H_0^{(2)}(k\rho) dx' dy' \quad (3.4)$$

To obtain Equation 3.5 the electric field is written as a sum of the incident and scattered (Equation 2.4) field. Equation 3.5 represents the total field electric field at any point in space.

$$E(x, y) = E^i(x, y) - \left(jk^2 / 4\right) \iint (\epsilon_r - 1) E(x', y') H_0^{(2)}(k\rho) dx' dy' \quad (3.5)$$

Next, the integral is broken up into N^m square regions. Equation 3.5 can be separated into a summation of integrals over each square region. The permittivity and the electric field are assumed to be constant within each square region. The electric field in each region is denoted by E_n and the permittivity in each region is denoted by ϵ_n . The electric field and the permittivity can be taken out of the integral, resulting in Equation 3.6a.

$$E_m + (jk^2 / 4) \sum_{n=1}^{N^m} (\epsilon_n - 1) E_n \iint_{th \text{ cell}} H_0^{(2)}(k\rho_m) dx' dy' = E_m^i \text{ with} \quad (3.6a)$$

$$\rho_m = \sqrt{(x_m - x')^2 + (y_m - y')^2} \quad (3.6b)$$

The integral can be written in closed form if the integral over the square region is approximated by an integral over a circular region with the same area. The closed form of the integral for the circular region is written as Equation 3.7.

$$(jk^2 / 4) \int_0^{2\pi} \int_0^a H_0^{(2)}(kr(\rho', \phi')) \rho' d\rho' d\phi' = \begin{cases} (j/2) [\pi ka H_1^{(2)}(ka) - 2j] & \text{if } n = m \\ (j\pi ka / 2) J_1(ka) H_0^{(2)}(k\rho_{mn}) & \text{if } n \neq m \end{cases} \quad (3.7)$$

Enforcing Equation 3.6 at the center of each region gives a constrained linear system. Substituting Equation 3.7 into Equation 3.6 gives the system of linear equations presented in Equation 3.8a.

$$\sum_{n=1}^{N^m} C_{mn} E_n = E_m^i \quad (3.8a)$$

$$C_{mn} = 1 + (\varepsilon_n - 1) \left(j / 2 \right) \left[\pi k a_n H_1^{(2)}(k a_n) - 2 j \right] \text{ if } m=n \quad (3.8b)$$

$$C_{mn} = (j \pi k a_n / 2) (\varepsilon_n - 1) J_1(k a_n) H_0^{(2)}(k \rho_{mn}) \text{ if } m \neq n \quad (3.8c)$$

$$\rho_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \quad (3.8d)$$

3.3 The scattered field

For the scattered field external to the cylinder Equation 3.9a [Richmond, 1965] is used once the electric field, E , has been computed.

$$E^s(x, y) = j(\pi k / 2) \sum_{n=1}^{N^m} (\varepsilon_n - 1) E_n a_n J_1(k a_n) H_0^{(2)}(k \rho_n(x, y)) \quad (3.9a)$$

$$\rho(x, y) = \sqrt{(x - x_n)^2 + (y - y_n)^2} \quad (3.9b)$$

3.4 Observations for inversion

There are N^O observation locations and N^f frequencies are measured at each location. This gives N complex observations ($N = N^f N^O$). When the scattered field, E^s , is separated into the real and imaginary parts, $2N$ real observations are produced. For storage purposes the real parts are stored at the top of the vector and imaginary parts are stored on the bottom of the vector. The observations are also grouped by frequency.

Figure 3.1 shows the structure of the \vec{d}^{prd} vector.

$$\bar{d}^{prd} = \begin{bmatrix} \text{Re } \bar{E}_m^s \\ \text{Im } \bar{E}_m^s \end{bmatrix} = \begin{bmatrix} k_1 \begin{bmatrix} E_1^s \\ \vdots \\ E_{N^o}^s \\ \vdots \\ E_{(N^f-1)N^o+1}^s \\ \vdots \\ E_{N^o N^f}^s \end{bmatrix} \\ k_{N^f} \begin{bmatrix} E_1^s \\ \vdots \\ E_{N^o}^s \\ \vdots \\ E_{(N^f-1)N^o+1}^s \\ \vdots \\ E_{N^o N^f}^s \end{bmatrix} \end{bmatrix}$$

Figure 3.1: Arrangement of the multi-frequency observation vector.

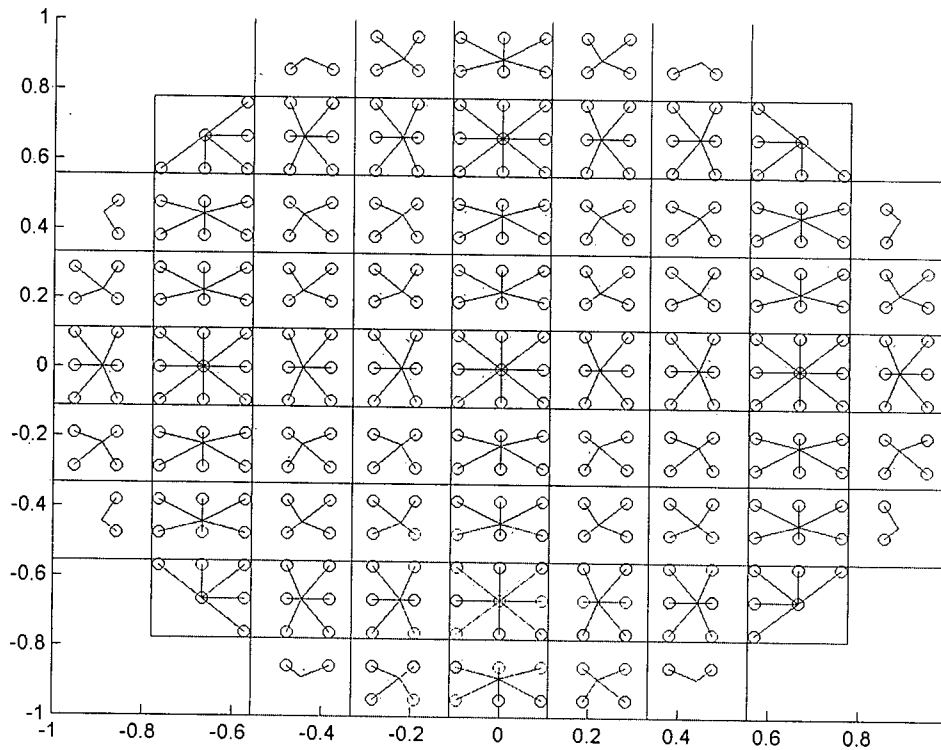


Figure 3.2: Illustration of the mapping concept. The small circles represent the center of the modeling cells and the large square regions represent the inversion cells. The lines connect the modeling region to the inversion region.

3.5 Gridding for the Jacobian calculation

Before the Jacobian calculation can be derived, the concept of using different inversion grids and modeling grids is introduced. The use of two grids is necessary owing to the competing demands of inversion and modeling. Modeling requires a minimum of 10 squares per wavelength. The finer the grid spacing the smaller the error attributable to modeling when compared to the physical system. This is problematic for inversion. Numerical experiments have shown that the inversion system becomes more unstable as more degrees of freedom are added to the system. A coarse grid works best for inversion. The modeling and inversion use different grids and a mapping is created

between the two grids to accommodate the competing demands of modeling and inversion. This mapping directly effects the Jacobian calculation.

Let there be N^I inversion cells and N^m modeling cells. For each inversion cell there are M^n modeling cells mapped to it. M^n can be different for different values of the n superscripts. Equation 3.10 must be true.

$$N^m = \sum_{n=1}^{N^I} M^n . \quad (3.10)$$

It is convenient to first introduce the concept of the indicator matrix. The indicator matrix maps the inversion cells to the modeling cells. The indicator matrix satisfies Equation 3.11a. Where $\underline{\underline{G}}$ is an $N^m \times N^I$ matrix. \vec{G}_n is a vector of length N^I that makes up a row of $\underline{\underline{G}}$ and G_{nk} is a scalar element of $\underline{\underline{G}}$. The vector $\vec{\varepsilon}^I$ denotes the inversion cells and the vector $\vec{\varepsilon}$ denotes the modeling cells. Equation 3.11c is the representation for the n 'th modeling cell as a function of the inversion cells written as a summation.

$$\vec{\varepsilon} = \underline{\underline{G}} \vec{\varepsilon}^I \quad (3.11a)$$

$$\underline{\underline{G}} = \begin{bmatrix} \vec{G}_1^T \\ \vdots \\ \vec{G}_{N^m}^T \end{bmatrix} \quad (3.11b)$$

$$\varepsilon_n = \sum_{b=1}^{N^I} \varepsilon_b^I G_{nb} \quad (\text{where } n=1..N^m) \quad (3.11c)$$

For this thesis the, \vec{G}_n vector will contain only one non-zero element and that value will be 1. The matrix element G_{nk} is equal to 1 if and only if the n 'th modeling

cell maps to the k 'th inversion cell. \underline{G} is sparse because most of the modeling cell do not map into the k 'th inversion cell.

Equations 3.8a, 3.8b, 3.8c, and 3.8d can be rewritten with the indicator notation.

$$\sum_{n=1}^{N^m} C_{mn} E_n = E_m^i \quad (3.12a)$$

$$C_{mn} = 1 + \left(\sum_{b=1}^{N^l} \varepsilon_b^l G_{nb} - 1 \right) (j/2) [\pi k a_n H_1^{(2)}(k a_n) - 2j] \text{ if } m=n \quad (3.12b)$$

$$C_{mn} = (j\pi k a_n / 2) \left(\sum_{b=1}^{N^l} \varepsilon_b^l G_{nb} - 1 \right) J_1(k a_n) H_0^{(2)}(k \rho_{mn}) \text{ if } m \neq n \quad (3.12c)$$

$$\rho_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \quad (3.12d)$$

The permittivity variable is the same for all the modeling cells mapped to the same inversion cell. Equation 3.5 is only for a single frequency. For multiple frequencies the forward modeling needs to be repeated for each frequency.

3.6 Derivation of the Jacobian for locations internal to the dielectric body

The Jacobian matrix, \underline{J} , calculates $\frac{\partial E_n}{\partial \varepsilon_k^l}$ for all combinations of n and k . Where k ranges from 1 to N^l and n ranges from 1 to N^m . To find $\frac{\partial E_n}{\partial \varepsilon_k^l}$, Equation 3.13a needs to be solved. Solving equation 3.13a will find $\frac{\partial E_n}{\partial \varepsilon_k^l}$ for all n simultaneously. The procedure needs to be repeated once for each k . There are k sets of linear equations that need to be solved to find all k derivatives for each inversion cell.

$$\sum_{n=1}^{N^m} \frac{\partial C_{mn}}{\partial \varepsilon_k^I} E_n + \sum_{n=1}^{N^m} C_{mn} \frac{\partial E_n}{\partial \varepsilon_k^I} = 0 \quad (3.13a)$$

$$\sum_{n=1}^{N^m} C_{mn} \frac{\partial E_n}{\partial \varepsilon_k^I} = - \sum_{n=1}^{N^m} \frac{\partial C_{mn}}{\partial \varepsilon_k^I} E_n \quad (3.13b)$$

$$\frac{\partial C_{mn}}{\partial \varepsilon_k^I} = G_{nk} (j/2) [\pi k a_m H_1^{(2)}(k a_m) - 2j] \quad \text{if } m = n, G_{nk} \neq 0 \quad (3.13c)$$

$$\frac{\partial C_{mn}}{\partial \varepsilon_k^I} = G_{nk} (j \pi k a_n / 2) J_1(k a_n) H_0^{(2)}(k \rho_{mn}) \quad \text{if } m \neq n, G_{nk} \neq 0 \quad (3.13d)$$

$$\frac{\partial C_{mn}}{\partial \varepsilon_k^I} = 0 \quad \text{otherwise} \quad (3.13e)$$

Solving the linear system stated in 3.8a gives the solution to E_n , leaving the $\frac{\partial E_n}{\partial \varepsilon_k^I}$ terms as unknowns in Equation 3.13a. The current system looks similar to the previous system, Equation 3.2, with the right hand side being different and a change of the unknown variables from E_n to $\frac{\partial E_n}{\partial \varepsilon_k^I}$.

Since \underline{C} is the same in every case, the LU decomposition technique will speed up computation of $\frac{\partial E_n}{\partial \varepsilon_k^I}$, as the same decomposition can be used $N^I + 1$ times.

3.7 Derivation of the Jacobian Matrix for the scattered field

Next, the scattering equation is examined. Rewriting Equation 3.9a, using indicator notation, results in Equation 3.14.

$$E^s(x, y) = j(\pi k / 2) \sum_{n=1}^{N^m} \left(\sum_{b=1}^{N^I} \varepsilon_b^I G_{nb} - 1 \right) E_n a_n J_1(k a_n) H_0^2(k \rho_n(x, y)) \quad (3.14)$$

Assuming that there are N^O observation points results in a set of N^O x_m and y_m coordinates. The subscript m distinguishes between each of the observation points.

$$E^s(x_m, y_m) = E_m^s = j(\pi k / 2) \sum_{n=1}^{N^m} \left(\sum_{b=1}^{N^l} \varepsilon_b^l G_{nb} - 1 \right) E_n a_n J_1(ka_n) H_o^{(2)}(k\rho_n(x_m, y_m)) \quad (3.15)$$

The derivative of Equation 3.15 is taken with respect to the k 'th inversion cell giving rise to Equation 3.16. The terms on the right hand side of the equation are all known quantities. The term $\frac{\partial E_m^s}{\partial \varepsilon_k^l}$ can be calculated directly from Equation 3.16.

$$\frac{\partial E^s(x_m, y_m)}{\partial \varepsilon_k^l} = -j(\pi k / 2) \left(\sum_{n=1}^{N^m} G_{nk} E_n a_n J_1(ka_n) H_o^{(2)}(k\rho_n(x_m, y_m)) + \sum_{n=1}^{N^m} \left(\sum_{b=1}^{N^l} \varepsilon_b^l G_{nb} - 1 \right) \frac{\partial E_n}{\partial \varepsilon_k^l} a_n J_1(ka_n) H_o^{(2)}(k\rho_n(x_m, y_m)) \right) \quad (3.16)$$

3.8 The Jacobian for multiple frequencies

The Jacobian matrix, $\underline{\underline{J}}$, requires the calculation of $\frac{\partial E_m^s}{\partial \varepsilon_k^l}$ for every k value, and for every frequency. Figure 3.3 shows the graphical arrangement of the Jacobian matrix.

The top half of the Jacobian matrix contains the real part of $\frac{\partial E_m^s}{\partial \varepsilon_k^l}$, the bottom half of the

Jacobian matrix contains the imaginary part of $\frac{\partial E_m^s}{\partial \varepsilon_k^l}$. Each half is further divided into

groups of frequencies. All the physical observations from a single frequency are stored within consecutive rows. The columns represent the index k . J_{mn} is a specific element inside $\underline{\underline{J}}$.

$$\Delta \bar{d} = \underline{J} \Delta \bar{\epsilon}^I = \begin{bmatrix} \text{Re} \frac{\partial E_n^s}{\partial \epsilon_k^I} \\ \vdots \\ \text{Im} \frac{\partial E_n^s}{\partial \epsilon_k^I} \end{bmatrix} \begin{bmatrix} k_1 \begin{bmatrix} \frac{\partial E_1^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_1^s}{\partial \epsilon_{N'}^I} \\ \vdots & \dots & \vdots \\ \frac{\partial E_{N^O}^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_{N^O}^s}{\partial \epsilon_{N'}^I} \end{bmatrix} \\ \vdots \\ k_{N^J} \begin{bmatrix} \frac{\partial E_{(N^J-1)N^O+1}^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_{(N^J-1)N^O+1}^s}{\partial \epsilon_{N'}^I} \\ \vdots & \dots & \vdots \\ \frac{\partial E_{N^J N^O}^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_{N^J N^O}^s}{\partial \epsilon_{N'}^I} \end{bmatrix} \\ \vdots \\ k_1 \begin{bmatrix} \frac{\partial E_1^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_1^s}{\partial \epsilon_{N'}^I} \\ \vdots & \dots & \vdots \\ \frac{\partial E_{N^O}^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_{N^O}^s}{\partial \epsilon_{N'}^I} \end{bmatrix} \\ \vdots \\ k_{N^J} \begin{bmatrix} \frac{\partial E_{(N^J-1)N^O+1}^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_{(N^J-1)N^O+1}^s}{\partial \epsilon_{N'}^I} \\ \vdots & \dots & \vdots \\ \frac{\partial E_{N^J N^O}^s}{\partial \epsilon_1^I} & \dots & \frac{\partial E_{N^J N^O}^s}{\partial \epsilon_{N'}^I} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \Delta \epsilon_1^I \\ \Delta \epsilon_2^I \\ \vdots \\ \Delta \epsilon_{N'}^I \end{bmatrix}$$

Figure 3.3: Structure of the multi-frequency Jacobian.

3.9 The flatness term

There are two ways to handle the flatness term, the one-sided difference method and the central difference method. The flatness term is defined as the first order approximation of the derivative of the permittivity distribution along the x and y directions at a discrete set of points.

In this thesis the flatness matrix uses the one-sided difference method to approximate the first order derivatives. The derivative with respect to x , and y , occur

between the centers of the inversion cells. $\underline{\underline{F}}$, the flatness matrix, encodes Equations 3.17 and 3.18.

$$\frac{\partial \varepsilon(x, y)}{\partial x} = \frac{\varepsilon(x + h, y) - \varepsilon(x, y)}{h} + O(h^2) \quad (3.17)$$

$$\frac{\partial \varepsilon(x, y)}{\partial y} = \frac{\varepsilon(x, y + h) - \varepsilon(x, y)}{h} + O(h^2) \quad (3.18)$$

Since the grid is regularly spaced, and the derivative approximation only involves its neighbors, the flatness matrix is sparse.

The central difference approximation is not used because it does not constrain the neighbors, but every other neighbor. This leads to two independent sheets of permittivity values connected at the edge of the cylinder. Examining Equation 3.19a and Equation 3.19b it is clear that $\varepsilon(x, y)$ is not in the central difference equation, and therefore not linked directly to its neighbors.

$$\frac{\partial \varepsilon(x, y)}{\partial x} = \frac{\varepsilon(x + h, y) - \varepsilon(x - h, y)}{2h} + O(h^2) \quad (3.19a)$$

$$\frac{\partial \varepsilon(x, y)}{\partial y} = \frac{\varepsilon(x, y + h) - \varepsilon(x, y - h)}{2h} + O(h^2) \quad (3.19b)$$

3.10 The N^I system of linear equations

From Equation 3.1, N^I linear equations can be derived to find the minimum of Φ . If Equation 3.1 is linearized with respect to $\Delta \bar{\varepsilon}'$, the system becomes quadratic with respect to the variable $\Delta \bar{\varepsilon}'$. There is only one minimum in a quadratic system and the minimum occurs at the point where the gradient vector is zero. N^I linear equations are produced, by taking the derivative of Equation 3.1 with respect to every $\Delta \varepsilon'_k$ and setting

every equations equal to zero. Using matrix-vector notation and a linear approximation of the term $\vec{d}(\bar{\epsilon}' + \Delta\bar{\epsilon}')$, Equation 3.20 can be written. The Jacobian matrix represents the first order derivatives of the term $\vec{d}(\bar{\epsilon}')$. The foregoing are shown in the equations that follow.

$$\min_{\Delta\bar{\epsilon}'} \left\{ \Phi = \frac{\|\vec{d}^{ref} - \vec{d}(\bar{\epsilon}' + \Delta\bar{\epsilon}')\|^2}{N^f N^o} + \frac{\beta \|F(\bar{\epsilon}' + \Delta\bar{\epsilon}')\|^2}{N^l} \right\} \quad (3.1a)$$

$$\min_{\Delta\bar{\epsilon}'} \left\{ \Phi = \frac{(\vec{d}^{ref} - \vec{d}(\bar{\epsilon}')) - J\Delta\bar{\epsilon}'}{N^f N^o} \left(\vec{d}^{ref} - \vec{d} - J\Delta\bar{\epsilon}' \right) + \frac{\beta (F(\bar{\epsilon}' + \Delta\bar{\epsilon}'))^T (F(\bar{\epsilon}' + \Delta\bar{\epsilon}'))}{N^l} \right\} \quad (3.20)$$

$$\begin{bmatrix} \frac{\partial}{\partial \Delta\epsilon'_1} \\ \frac{\partial}{\partial \Delta\epsilon'_2} \\ \vdots \\ \frac{\partial}{\partial \Delta\epsilon'_{N^l}} \end{bmatrix} \Phi = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (3.21)$$

The minimization problem stated in Equation 3.20, satisfies Equation 3.21 and the solution can be written as Equation 3.22. Equation 3.23 is the equation for updating the epsilon values for the next iteration. The algorithm is shown below:

1. Start with an initial guess to the permittivity
2. Calculate $\Delta\bar{\epsilon}^{new}$ from Equation 3.22

$$\Delta\bar{\epsilon}' = \left(\frac{J^T J}{N^f N^o} + \frac{\beta F^T F}{N^l} \right)^{-1} \left(\frac{J^T (\vec{d}^{ref} - \vec{d})}{N^f N^o} - \frac{\beta F^T F \bar{\epsilon}'}{N^l} \right) \quad (3.22)$$

3. Update the permittivity with Equation 3.23.

$$\bar{\epsilon}^{new} = \bar{\epsilon}' + \Delta\bar{\epsilon}' \quad (3.23)$$

4. Repeat from 2, until the stopping condition is reached.

Equation 3.22 is repeated, using $\bar{\epsilon}^{new}$ as the starting point for the linear approximation each time. The Jacobian matrix and the observations need to be recalculated given the new permittivity, $\bar{\epsilon}^{new}$.

3.11 Conclusion

In this chapter an inversion technique is developed. The technique is based on least squares minimization. The system is minimized with respect to $\Delta\bar{\epsilon}'$ and a linear model for scattering. The technique is iterative in nature and successive iterations yield better solutions.

This chapter demonstrates how to calculate the Jacobian matrix for the scattered field from Richmond's Method for dielectric modeling and how to incorporate a flatness term into the least squares minimization. The Jacobian calculation requires 2 stages. The first stage is to calculate a Jacobian for the internal field, and the second stage is to calculate the Jacobian matrix for the scattered field. The Jacobian calculation for the scattered field depends only on the internal field parameters.

Chapter 4: Implementation of the tomography algorithm

4.0 Introduction

The detailed mapping between the code and the equations presented in Chapter 3 is shown in Chapter 4.

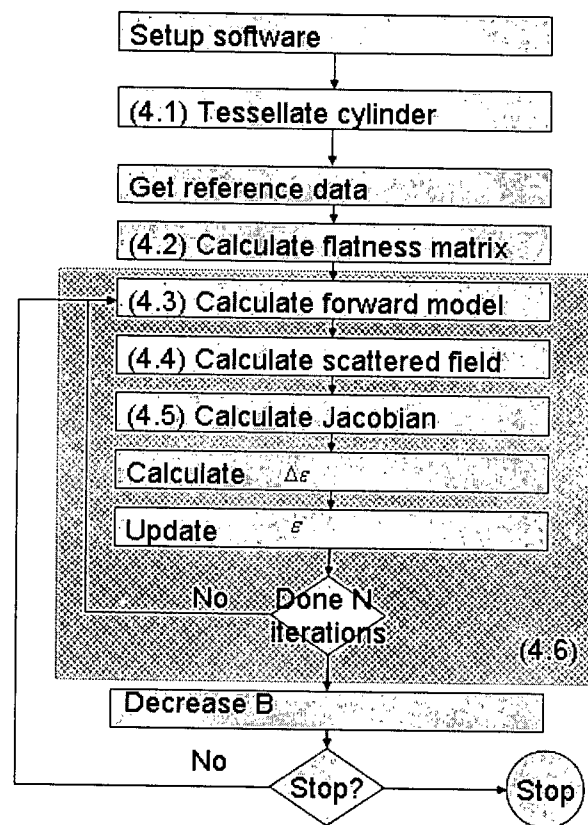


Figure 4.1: Flow chart for the complete inversion system. The numbers in brackets represent the section each calculation is presented. Section 4.6 represents the complete non-cooled system.

The flow chart for the algorithm is shown in Figure 4.1. This chapter explains how the equations in Chapter 3 are implemented in MatLab code. The code presented works with MatLab 5.3 and is upward compatible. The code is presented in the same order as in the flow chart. The code is presented in the sections that follow:

4.1) Tessellation

- 4.2) Calculating the flatness matrix
- 4.3) Calculating the internal field
- 4.4) Generation the observation matrix
- 4.5) The Jacobian calculation
- 4.6) Inversion function with a single β value
- 4.7) Inversion function with multiple β values

4.1 Tessellating the cylinder

Since the problem is two dimensional in nature (infinite cylinder) the tessellation of a circle is sufficient. The circle radius and grid spacing are the only two parameters needed for the automated generation of a uniformly tessellated circle. To create the discretization, a regularly spaced set of squares is superimposed onto a circle. If the center of the square lies within the circle then the square is considered part of the set of squares that make up the circle. This mapping results in two types of tessellation errors. The first error, labeled A in Figure 4.2, is the inclusion of areas that are not part of the circle. The second error, labeled B in Figure 4.2, is the exclusion of areas that are part of the circle. The circle is centered at the origin. Equation 4.1 is used as the test for membership for the set of squares that make up the circle.

$$x^2 + y^2 < \rho^2 \quad (4.1)$$

Figure 4.2 describes the foregoing concept graphically. Listing 4.1, `square_grid.m`, is the MatLab implementation of this mapping. The parameter `Circle_radius` is the radius of the desired circle, and `T_rad` is half the length of the edge of the square.

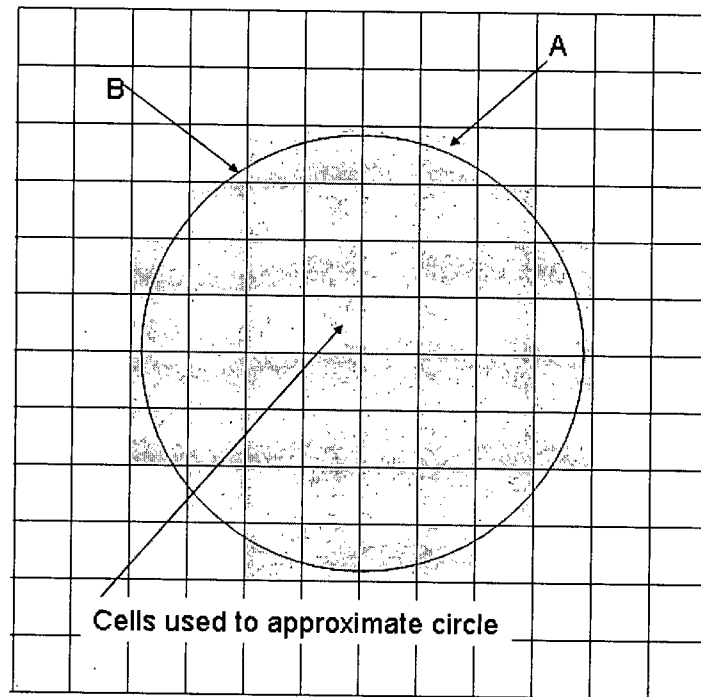


Figure 4.2: Illustrative diagram of the discretization concept. The cells in grey are used to approximate the circle. The centers of the grey cells all lie within the circle. Notice the staircase effect at the edges. With finer spacing this staircase can be reduced, but never eliminated. This causes a modeling error because approximate tessellation is never a perfect circle.

Listing 4.1 square_grid.m

```
function [Grid_points] = SQUARE_GRID(Circle_radius, T_rad)
Y=(-Circle_radius+T_rad):T_rad*2:(Circle_radius+T_rad);
%X and Y axis
X=Y;
N = size(X,2); %find out how many divisions on the axis
Grid = zeros(N,N); %Large square Grid
for LX = 1:N      %test if the point is on the circle
    for LY = 1:N
        if ((X(LX)^2+Y(LY)^2) < Circle_radius^2)
            Grid(LX,LY) = 1;
        end
    end
end
[I J] = ind2sub(size(Grid), find(Grid));
Grid_points = [X(I)' Y(J)'];
```

Related to the `square_grid` function is the `neighbors` function, shown in Listing 4.2. The `neighbors` function generates a list of neighbors for each cell. This function is essential for generating the flatness matrix, \underline{F} . The `neighbors` function takes in the set of squares (`Src`) and the grid spacing (`Trad`) as parameters. The output is an $N \times 4$ array. N is the number of squares. Column 1 to 4 represent the neighbors. The first column represents the neighbor to the left ($x - \Delta x$), the second column represents the neighbor to the right ($x + \Delta x$), the third column represents the neighbor on top ($y + \Delta y$) and the forth column represents the neighbor below ($y - \Delta y$).

The `neighbors` function determines if a cell is a neighboring cell by computing the distance vector to that cell. The algorithm eliminates the cells where the x or y coordinate differs from the reference cell by more than one grid spacing. This leaves 5 candidate cells, one cell for each neighbor and the reference cell. The algorithm then determines the neighbor by examining the direction of the distance vector.

Listing 4. 2 Neighbor.m

```
% this function determines the nearest neighbors to a
%single point.
function NeighList = neighbors(Src, Trad)

N = size(Src,1);
NeighList = zeros(N, 4);
epsilon = 0.01*Trad;

for loop = 1:N
%This is a coarse filter that eliminates alot of the
%calculations
    List1 = find ( abs(Src(:,1) - Src(loop,1)) <= ...
        (2*Trad+epsilon));
    List2temp = find ( abs(Src(List1,2) - Src(loop,2))...
        <= (2*Trad+epsilon));
    List2 = List1(List2temp);

    for loop2 = List2'
```


Listing 4. 2 Neighbor.m

```
deltax = Src(loop,1) - Src(loop2,1);
deltay = Src(loop,2) - Src(loop2,2);
r = sqrt(deltax^2 + deltay^2);
if ( r < (2*Trad + epsilon))
    if (deltax > epsilon)
        NeighList(loop,1) = loop2;
    elseif (deltax < -epsilon)
        NeighList(loop,2) = loop2;
    elseif( deltay > epsilon)
        NeighList(loop,3) = loop2;
    elseif(deltay < -epsilon)
        NeighList(loop,4) = loop2;
    end
end
end
end
end
```

4.2 Calculating the flatness matrix

Once the grid is generated the flatness matrix can be calculated. The flatness matrix implements Equation 3.17 and Equation 3.18, the one-sided difference equations. It requires a list of neighbors generated with the previous function and the grid spacing as parameters. The neighbors list allows for quick identification of the cell to the right of the current cell, and on top of the cell. The x derivatives are encoded in the top half of the matrix and the y derivatives are encoded in the bottom half of the matrix.

$$\frac{\partial \varepsilon(x,y)}{\partial x} = \frac{\varepsilon(x+h,y) - \varepsilon(x,y)}{h} + O(h^2) \quad (3.17)$$

$$\frac{\partial \varepsilon(x,y)}{\partial y} = \frac{\varepsilon(x,y+h) - \varepsilon(x,y)}{h} + O(h^2) \quad (3.18)$$

Listing 4. 3 gradientonesided.m

```
%This function generates
function Generate = gradientonesided(NeighList, Trad)
N = size(NeighList,1);
Matrix = zeros(2*N, N); %top N for x direction,
                        % bottom N for y direction
```

Listing 4.3 gradientonesided.m

```

for loop = 1:N          %This is the x loop
    index = loop;
    if (NeighList(index,2) ~= 0)
        Matrix(index,NeighList(index,2)) = 1;
        Matrix(index,index) = -1;
    end
end

for loop = (N+1):(2*N) %This is the y loop
    index = loop - N;
    if (NeighList(index,3) ~= 0)
        Matrix(loop,NeighList(index,3)) = 1;
        Matrix(loop,index) = -1;
    end
end
Generate =Matrix / (2*Trad);

```

4.3 Calculating the internal field using Richmond's method

The internal field solution is a two-step procedure. The first part is the generation of the dielectric permittivity independent portion of the matrix. The second part is the generation of the complete matrix followed by the solution for E_n .

$$\sum_{n=1}^{N^m} C_{mn} E_n = E_m^i \quad (3.8a)$$

$\underline{\underline{C}}$ in Equation 3.8a can be expanded so the permittivity values are explicit in the equation, resulting in Equation 4.3, obtained from Equation 3.8a.

$$(\underline{\underline{I}} + \underline{\underline{D}} \text{diag}(\vec{\epsilon} - 1))E = E^i \quad (4.3)$$

$\underline{\underline{D}}$ is the permittivity independent matrix and only needs to be calculated once for the algorithm. This saves time by not requiring N^2 Bessel functions to be calculated each

algorithm iteration. This first step, `init_block_matrix_gen.m`, is implemented in listing 4.4.

Listing 4.4 `init_block_matrix_gen.m`

```
%returns the forward matrix dependence
function success = init_block_matrix_gen(List_C, ...
                                         k,T_rad, FName)

N=size(List_C,1);
Diag_ind = 1:N+1:N^2;
F_mat = zeros(N,N);
Lumped_Constant=j*pi*k*T_rad*0.5.*besselj(1,k*T_rad);
for loop = 1:N
    p = sqrt((List_C(:,1) - List_C(loop,1)).^2 +...
            (List_C(:,2) - List_C(loop,2)).^2);
    F_mat(loop,:) = ...
        (Lumped_Constant.*besselh(0,2,k.*p)).';
end
F_mat(Diag_ind) = (j/2)*(pi*k*T_rad*...
                    besselh(1,2,(k*T_rad))-2j);

save(FName,'F_mat');
success = 1;
```

The second step calculates the matrix $(\underline{I} + \underline{D}diag(\underline{\tilde{\epsilon}} - 1))$. This calculation is presented in listing 4.5, `block_matrix_gen.m`. After \underline{C} has been generated, the internal fields can be found. Listing 4.6, `sample1.m`, demonstrates how to calculate the internal fields. Listing 4.6 is clearly separated into 3 sections, setup, matrix generation and solving of the internal fields.

Listing 4.5 block_matrix_gen.m

```
%returns the forward matrix dependence
function [F_mat] = block_matrix_gen(epsilon, FName)

load(FName) %load parameters stored in the
              %init_block_matrix_gen function

N = size(F_mat,1);
for loop = 1:N
    F_mat(loop,:) = F_mat(loop,:).*(epsilon'-1);
end
F_mat = F_mat + eye(N); %add the identity matrix
```

Listing 4.6 sample1.mat

```
%sample script for calculating the internal fields
%Setup
C_rad = 1;          %cylinder radius
T_rad = 1/21;       %tessellation radius
k = 2*pi;           %wave number
Fname = 'temp'; %Define file name for storing stuff
Grid = square_grid(C_rad, T_rad); %generate the Grid
N = size(Grid, 1);
epsilon = 2*ones(N,1); %define permittivity distribution
Ei = PlaneWave(Grid,k); %define the incident field
init_block_matrix_gen(Grid, k,T_rad, FName);
    %calculate the epsilon independent part of Cmatrix

%generating the matrix and LU decomposing
C_Mat=block_matrix_gen(epsilon, FName);
[L,U] = lu(C_Mat);

%solving for the internal fields.
temp = L\Ei;
E     = U\temp;
```

4.4 Generation of the observation matrix

The procedure for generating the scattered field is split into step 1, a permittivity independent step, and step 2, a permittivity dependent step. To calculate the scattered field from the internal fields, Equation 3.9a is used.

$$E^s(x_m, y_m) = j(\pi k / 2) \sum_{n=1}^{N^m} (\varepsilon_n - 1) E_n a_n J_1(ka_n) H_o^{(2)}(k\rho_n(x_m, y_m)) \quad (3.9a)$$

The solution for the scattered field can be written in the form of Equation 4.4, when \underline{B} is calculated first, and the actual scattered field is calculated second.

$$\underline{\bar{E}}^s = \underline{B} \text{diag}(\underline{\bar{\varepsilon}} - 1) \underline{\bar{E}}_n \quad (4.4)$$

Listings 4.7, Init_Block_Sc_field.m, and Listings 4.8, Init_Block_Sc_Field.m, implement the two step procedure. The function Init_Block_Sc_Field, which implements step 1, takes in the grid(List_S), a list of observations points, the wave number(k), the grid spacing(T_rad), and a filename(FName) as arguments. The filename, FName, is used to store the results temporarily. The Block_Sc_Field function has the permittivity distribution(epsilon), the internal fields(E) and the filename(FName) as input arguments. Block_Sc function loads the temporary results stored by Init_Block_Sc_Field.

Listing 4.7 Init_Block_Sc_Field

```
function success = Init_Block_Sc_Field(List_S, List_O, k,
T_rad, FName);

NO = size(List_O,1);
NE = size(List_S,1);

ObsMat = zeros(NO,NE);
```

```

Lumped_Constant = -j*pi*k/2*besselj(1,k*T_rad)*T_rad;
for loop = 1:NO
    r = sqrt((List_S(:,1) - List_O(loop,1)).^2 + ...
            (List_S(:,2) - List_O(loop,2)).^2)';
    ObsMat(loop,:) = Lumped_Constant*besselh(0,2,k*r);
end
save(FName, 'ObsMat');
success =1;

```

Listing 4. 8 Block_scattered_field.m

```

% this function returns the scattered field at a given
% point
function [Es] = Block_Scattered_feild(epsilon, E,FName)
load(FName); %load stored calculations
Es = ObsMat*((epsilon-1).*E);

```

4.5 The Jacobian Calculation

The first task is the creation of a mapping between the modeling grid and the inversion grid. This mapping is referred to the \underline{G} in Equation 3.11a and presented in the output List. The columns of the matrix \underline{G} are compressed in List. The columns are stored as cell arrays, where the indices of the non-zero elements of \underline{G} form the array.

The NetList function maps the modeling cell to the closest inversion cell. It does this in two stages: i) It determines if the center of the modeling cell is within an inversion cell. This is done with the within subfunction, which checks if a modeling grid cell is within the interval of the inversion cell. The second part maps the cells that haven't been mapped to an inversion cell with within. This step finds the closest inversion cell to the given modeling grid cell.

Listing 4. 9 NetList.m

```

%This is a mapping function for a square
function List = NetList(CorseGrid, CRad, FGrid, FRad)
NC = size(CorseGrid,1);

```

Listing 4.9 NetList.m

```

NF = size(FGrid,1);
used = ones(1,NF);
List = cell(1,NC);

%generate the mapping
for loopC = 1:NC
    temp = [];
    for loopF = 1:NF
        if(Within(CorseGrid(loopC,1),CorseGrid(loopC,2),...
            FGrid(loopF,1),FGrid(loopF,2),CRad))
            if (used(loopF) == 1)
                temp = [temp loopF];
                used(loopF) =0; %prevents duplicates.
            end
        end
    end
    List{loopC} = temp;
end

%look for any points in the fine grid that
% hasn't been assigned.
for loopF = 1:NF

    if (used(loopF) ==1) % find the nearest nieghbor
        loopC =1;
        rMax = (CorseGrid(loopC,1) - FGrid(loopF,1))^2 + ...
            (CorseGrid(loopC,2) - FGrid(loopF,2))^2;
        Closest = loopC;
        for loopC = 1:NC
            r2 = (CorseGrid(loopC,1) - FGrid(loopF,1))^2 ...
                +(CorseGrid(loopC,2) - FGrid(loopF,2))^2;
            if(r2 < rMax)
                Closest = loopC;
                rMax = r2;
            end
        end
        List{Closest} = [[List{Closest}] loopF];
    end
end

%helper function
function result = Within(X1,Y1,X2,Y2,rad)
if ((X2 > (X1 - rad)) & (X2 <= (X1 + rad)) & ...
    (Y2 > (Y1 - rad)) & (Y2 <= (Y1 + rad)))
    result= 1;
end

```

Listing 4. 9 NetList.m

```

else
    result = 0;
end

```

The next task, in the Jacobian calculation, is the determination of the derivative of the internal fields as computed by Equation 3.13a.

$$\sum_{n=1}^{N^m} \frac{\partial C_{mn}}{\partial \varepsilon_k^i} E_n + \sum_{n=1}^{N^m} C_{mn} \frac{\partial E_n}{\partial \varepsilon_k^i} = 0 \quad (3.13a)$$

BlockJacNL, in Listing 4.10, has the LU decomposition of $\underline{\underline{C}}$. Since the problem involves many right hand sides for the same matrix, the LU decomposition, for a given $\underline{\underline{C}}$, can be reused.

Listing 4. 10 BlockJacNL.M

```

%This file is used for the calculation of the Jacobian for
%a mapped SRC
%list. The return type will be a Nxepsilon matrix. Were
%each column represent
%Parameters
%SrcF - location of the tessellation points
%EF - fields at the tessellation points
%LF - lower matrix of the lu(Mat) were Mat represents the
%forward problem
%UF - upper matrix of the lu(Mat) were Mat represent the
%forward problem
%epsilon - the epsilon value of the coarse grid. Should be
%stored as a column vector
%NetList -
%k - wave number 2*pi/lambda
%Crad - corrected radius sqrt(4/pi)*tessellation radius

function Jac = blockJacNL(SrcF, EF, LF,UF, epsilon,...
    Netlist, k, Crad,FName);

%step allocate space
NE = size(epsilon,1);
NF = size(SrcF,1);
Jac = zeros(NF,NE);
Map = zeros(NF,1);

```


Listing 4. 10 BlockJacNL.M

```
load(FName);
for loop = 1:NE
    A = Netlist{loop};
    Map(A) = loop;
end
for loopE = 1:NE
    rhs = zeros(NF,1);
    for loopFeqn = 1:NF
        for loopFrhs = 1:NF
            if (Map(loopFrhs) == loopE)
                rhs(loopFeqn) = rhs(loopFeqn) ...
                +F_mat(loopFeqn,loopFrhs)*EF(loopFrhs);
            end
        end
    end
    temp = -LF\rhs;
    Jac(:,loopE) = UF\temp;
end
```

The last task is the calculation of the Jacobian matrix for the scattered field in Equation 3.16. Equation 3.16 is a simple summation of terms.

$$(3.16) \quad \frac{\partial E^s(x_m, y_m)}{\partial \varepsilon_k^i} = -j(\pi k / 2) \left(\sum_{n=1}^{N^m} G_{kn} E_n a_n J_1(ka_n) H_0^{(2)}(k\rho_n) + \sum_{n=1}^{N^m} \left(\sum_{b=1}^{N^i} \varepsilon_b^i G_{nb} - 1 \right) \frac{\partial E_n}{\partial \varepsilon_k^i} a_n J_1(ka_n) H_0^{(2)}(k\rho_n(x_m, y_m)) \right)$$

The BlockassembleJacobian function produces the complex Jacobian matrix. Separating the matrix into real and imaginary parts will be done explicitly in the inversion script.

Listing 4. 11 BlockassembleJacobian .m

```
%assemble complex jacobian
function CJac = BlockAssembleJacobian(EJac, En, ...
epsilon, NetList, Src, Obs, k, Crad, FName)
NO = size(Obs,1);
NE = size(epsilon,1);
```

Listing 4. 11 BlockassembleJacobian .m

```

NF = size(Src,1);
CJac = zeros(NO,NE);
Map = zeros(NF,1);
LC = (j*pi*k/2)*Crad*besselj(1,k*Crad);
load(FName)
for loop1 = 1:NE
    temp = NetList{loop1};
    Map(temp) = loop1;
end
for loopE = 1:NE
    CJac(:,loopE) = ObsMat*((epsilon(Map)-...
1).*EJac(:,loopE));
    for loopO = 1:NO
        temp = NetList{loopE};
        for loopF2 = temp
            r = sqrt((Src(loopF2,1) -Obs(loopO,1)).^2...
+ (Src(loopF2,2) -Obs(loopO,2)).^2);
            CJac(loopO,loopE) = CJac(loopO,loopE) - ...
LC*En(loopF2)*besselh(0,2,k*r);
        end
    end
end
clear ObsMat;

```

4.6 Inversion function with a single beta value

All the steps, as shown in Figure 4.3, are now in place. All the elements of the inversion scheme can be placed in a function to simplify the inversion procedure. Listing 4.12 is the function. The function `basicInversion` essentially runs the inversion algorithm for any single beta value and for a fixed number of iterations that is specified by the user. The loop seen in Figure 4.3 is implemented as a for loop. Cascading the β is done outside this function. Figure 4.3 outlines the algorithm performed in Listing 4.12.

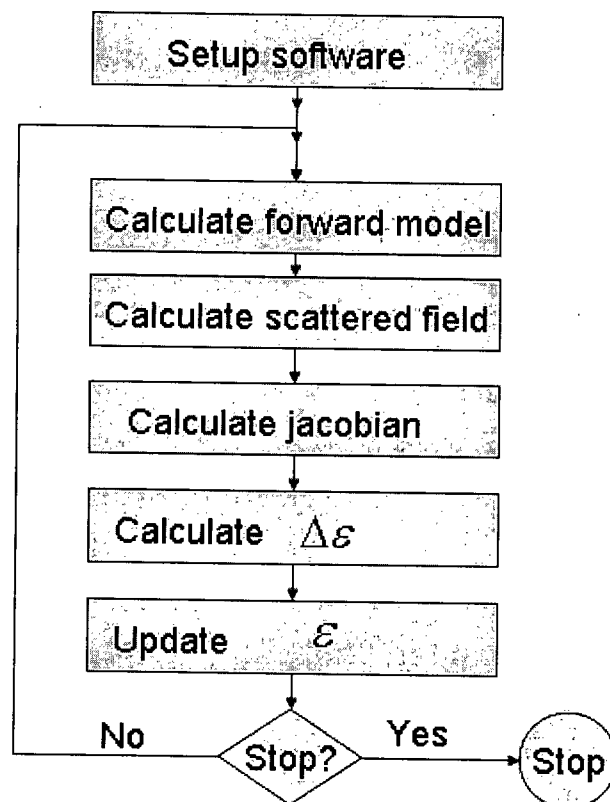


Figure 4.3: Flow chart for the `BasicInversion` function. The setup portion of the software includes the encoding of all the parameters that must be chosen. The function loops through the Jacobian calculation and updates the permittivities.

The calculation of $\Delta\bar{\epsilon}'$ is done in Listing 4.12. It is near the bottom and implements Equation 3.22.

$$\Delta \vec{\varepsilon}' = \left(\frac{\underline{\underline{J^T J}}}{N^f N^o} + \frac{\underline{\underline{\beta F^T F}}}{N^i} \right)^{-1} \left(\frac{\underline{\underline{\alpha J^T (\vec{d}^{ref} - \vec{d})}}}{N^f N^o} - \frac{\underline{\underline{\beta F^T F \vec{\varepsilon}'}}}{N^i} \right) \quad (3.22)$$

Listing 4.12 BasicInversion.m

```
%This is an attempt to abstract the inversion function to
%make it easier to run batches
%as well analyze performance

function [epsilonC, Misfit] = BasicInversion(Ref_Obs, ...
%This is the reference data
    FileNames, ... %Filename for block matrix
                    %that has been pre-calculated
    TempDirectory, ... %where to store temporary files
    CylinderRad, ... %Where to store parameters of the
                    %cylinders
    Corse_rad, ... %The size of the inversion tessellation
    Fine_rad, ... %The size of the modeling tessellation
    Corse_grid, ...
    Fine_grid, ...
    ITable, ...
    Karray, ... %an Array of wave numbers
    EpsilonC_initial, %starting model to be determine
                    %separately
    Obs_pts, ... %Observation Pts
    NIter, ... %Number of iterations
    picturename, ...
    alpha, beta, FDMatrix)

% Part 1 Figure out the size of matrixes and vectors

NK = size(Karray,1); %number of frequencies
NO = size(Obs_pts,1); %number of observations points
NEF = size(Fine_grid,1); %number of model cells
NEC = size(Corse_grid,1); %number of inversion cells
CFine_rad = sqrt(4/pi)*Fine_rad; %corrected radius for
                    %Richmond's method
epsilon = zeros(NEF,1); %Link inversion and model cells
epsilonC = EpsilonC_initial;

%Part 1.2 generate filenames and mappings for storage
% purposes

% Part 1.1 generate some filenames for temp directories
TempFile = cell(NK,1);
TempFile2 = cell(NK,1);
```

Listing 4.12 BasicInversion.m

```

for loopK = 1:NK
    TempFile{loopK} = sprintf('%s%i',TempDirectory,loopK);
end
for loopI = 1:NIter
    TempFile2{loopI} = sprintf('%sa%i'b',picturename,loopI);
end
% Part 1.2 Map the permittivity from inversion to modeling.
for loop = 1:NEC
    epsilon(IETable{loop}) = epsilonC(loop);
end

%Part 2 allocate space
Misfit = zeros(NIter,1);
IObs = zeros(NO*NK,1);
ObsJacMF = zeros(NO*NK,NEC);

% Part 3 setup the iterative algorithm
for iter = 1:NIter
    %3.1 calculate the internal fields
    for loopK = 1:NK
        Mat = block_matrix_gen(epsilon, FileNames{loopK,1});
        [L,U] = lu(Mat);
        Ei = PlaneWave(Fine_grid, Karray(loopK));
        TempE = L\Ei;
        E      = U\TempE;
        ES     = ...
        Block_Scattered_field(epsilon,E,FileNames{loopK,2});
        save(TempFile{loopK},'L','U','E','ES');
        index1 = (loopK -1)*NO+1;
        index2 = NO*loopK;
        IObs(index1:index2) =ES;
    end

    %Part 3.2 calculate the Jacobian
    for loopK = 1:NK
        load(TempFile{loopK});
        index1 = (loopK -1)*NO+1;
        index2 = NO*loopK;

        %Part 3.1.1 Jacobian stuff
        IntJac = BlockJacNL(Fine_grid, E, L,U, epsilonC,...
        IETable, Karray(loopK), CFine_rad, FileNames{loopK,1});
        ObsJac = BlockAssembleJacobian(IntJac,E, epsilonC,...
        IETable, Fine_grid, Obs_pts, Karray(loopK),...

```

Listing 4.12 BasicInversion.m

```
CFine_rad, FileNames{loopK,2});
ObsJacMF(index1:index2,:) = ObsJac;
end
%3.3 now to separate into real and imaginary parts
delta_obs = Ref_Obs-IObs;
Misfit(iter) = sum(abs(delta_obs).^2)/(NO*NK);

%Part 3.4 separate Jacobian into real and
%imaginary parts.
RMObs = zeros(2*NO*NK,1);
RMJac = zeros(2*NO*NK,NEC);
RMObs(1:(NO*NK),:) = real(delta_obs);
RMObs((NO*NK+1):(2*NO*NK),:) = imag(delta_obs);
RMJac(1:(NO*NK),:) = real(ObsJacMF);
RMJac((NO*NK+1):(2*NO*NK),:) = imag(ObsJacMF);

%3.5 Delta epsilon calculated here. delta_epsilon =
(alpha*RMJac'*RMJac/(NO*NK)+beta*FDmatrix'*FDmatrix/NEC)\...
(alpha*RMJac'*RMObs/(NO*NK)-...
(beta*FDmatrix'*FDmatrix*epsilonC)/NEC);

%3.6 permittivity updated here.
epsilonC = delta_epsilon+epsilonC;
for loop = 1:NEC
    epsilon(IETable{loop}) = epsilonC(loop);
end
save(TempFile2{iter},'epsilonC')
end
```

4.7 Inversion function with multiple beta values

Listing 4.13 performs the algorithm shown in Figure 4.1. It uses the

BasicInversion function for the greyed out box in Figure 4.1.

Listing 4.13 complete script

```
cellN = 'cascade3';
%load the reference observations
searchstring = sprintf('c:\\model\\fourcellobs2.mat')
if (~exist(searchstring))
    load4_parameters2;
end
load(searchstring)
%generate Grids
```

Listing 4. 13 complete script

```

radius =1;
Irad= 1/9;
Mrad=1/41;
Igrid = square_grid(radius, Irad);
Mgrid = square_grid(radius, Mrad);
Correct_rad = sqrt(4/pi)*Mrad;
ETable = NetList(Igrid, Irad, Mgrid, Mrad);
NF= size(Mgrid,1);
NC= size(Igrid,1);

%setup matrixes
%generate some filenames
filename = cell(NK,2);
for loopK = 1:NK
    filename{loopK,1} =...
    sprintf('c:\\model\\FWMAT%02i',loopK);
    filename{loopK,2} =...
    sprintf('c:\\model\\OBMAT%02i',loopK);
end

%Initialize function for calculating scattered
%and internal fields
for loopK = 1:NK
    Init_Block_matrix_gen(Mgrid, Karray(loopK),...
    Correct_rad,filename{loopK,1});
    Init_Block_Sc_field(Mgrid, ObsPts,...
    Karray(loopK), Correct_rad, filename{loopK,2});
end

% Declare beta values to use
BV = [ 10*(8/9).^[0:60] 0 0 0 ];
NB = size(BV,2);
epsilonStart = 2.5*ones(NC,1);
%Generate Flatness matrix
NeiList = Neighbors(Igrid,Irad);
FDmatrix = gradientonesided(NeiList, Irad);
NIter =1;
alpha = 1;

for loopB =1:NB
    beta = BV(loopB);
    epsilonStore = ...
    sprintf('C:\\%sOS\\epsilon%05i_',cellN,loopB);
    MisfitStore = ...
    sprintf('C:\\%sOS\\Misfit%05i',cellN,loopB);

```

Listing 4. 13 complete script

```
%perform the inversion for a single beta
[epsilonC, Misfit]...
=InversionWithFlattessModel(Ref_Obs,...% reference data
    filename, ... %Filenames for matrix that has
                    %been pre-calculated
    'c:\\temo',...%Where to store temporary files
    radius,...     %Where to store parameters of the
cylinders
    Irad,...       %The size of the coarse tessellation
    Mrad,...       %The size of the fine tessellation
    Igrid,...
    Mgrid,...
    ETable, ...
    Karray, ...    %An Array of wave numbers
    epsilonStart,...%Starting model to be
                    %determine separately
    ObsPts, ...    %Observation Pts
    Niter,...
    epsilonStore,...%Number of interactions
    alpha, beta, FDmatrix);
epsilonStart = epsilonC; %cascading
save(MisfitStore,'Misfit'); %store the misfit function
                    %for analysis
end
```

4.8 Summary

In this chapter the implementation of the inversion algorithm is presented. The implementation is separated into several functions. The separation of the program into several functions allows for easy debugging and testing of the algorithm. The program has not been optimized for speed or memory, but attempts to reuse calculations where possible.

Chapter 5: Numerical experiments using the tomography algorithm

5.0 Introduction

In this chapter an investigation of the functional minimization defined in Equation 3.1 is explored. The parameter, β , in this equation is varied in magnitude, as is the number of iterations. β represents the relative weighting between the data misfit term and the flatness in the predicted model. The investigation is separated into experiments.

The experiments are organized into 4 sets, each exploring a different aspect of the inversion algorithm. The first three sets of experiments share common parameters, as presented in Section 5.1. The first two sets of experiments will share 3 common test cases. The first set of experiments keeps β constant, as a function of iteration. The second set of experiments decreases β as a function of iteration. Decreasing β is analogous to cooling in simulated annealing. The third set of experiments introduces noise to the most complex test cases of the second set of experiments. The fourth set of experiments explores the issue of moving to a case where there are more inversion cells than observations. This is known as the under-determined case.

Table 5.1: Summary of experiments for chapter 5

Experimental set	Permittivity distributions	Number of experiments	Description
1	Figures 5.4, 5.5, 5.6	21	β is fixed in these experiments.
2	Figures 5.4, 5.5, 5.6	6	β is decreasing in these experiments
3	Figure 5.6	1	Noise is added to the signal
4	Figure 5.22	1	β is decreasing in these experiments, and a finer tessellation is used.
5	Figure	1	Same as above, but with 2 blobs.

5.1 Setup for the experimental procedure

For the experiments in Chapter 5, the \vec{d}^{ref} vectors are numerically generated, using Richmond's Method. The permittivity distributions for the numerically generated scattered fields are chosen such that the permittivity distribution can be described exactly by the inversion tessellations. That is the tessellation is a region consisting only of a collection of squares, approximating a circle. This eliminates any noise that arises from errors caused by the inability of the tessellation cells to model the permittivity distribution exactly.

For all experiments, 6 frequencies are used; a circular cylinder with a radius of 1λ forms the outer boundary of the object. Table 5.2 summarizes the properties in experimental sets 1, 2 and 3. The properties for experimental sets 4 and 5 are presented in Section 5.7 and Section 5.8. Note that all these experiments do not represent far-field scattering.

Table 5.2: Parameters for the experimental sets 1, 2 and 3

Parameter	Value
Cylinder radius	1λ
k values	$0.80*2\pi$ $0.88*2\pi$ $0.96*2\pi$ $1.04*2\pi$ $1.12*2\pi$ $1.20*2\pi$
Inversion cell tessellation radius	$1/9\lambda$
Modeling cell tessellation radius	$1/41\lambda$
Incident wave	e^{jkx}
Observation locations	-90 to 90 degrees with 6 degrees of separation. $\rho = 1.5\lambda$

The relevant geometry is shown in Figure 5.1, which represents the cells corresponding to possible permittivity distributions. The permittivity value is assumed to be constant within each region. Each square region represents an inversion cell, with length $2/9\lambda$. The regions are generated with the `square_grid` function described in Chapter 4.

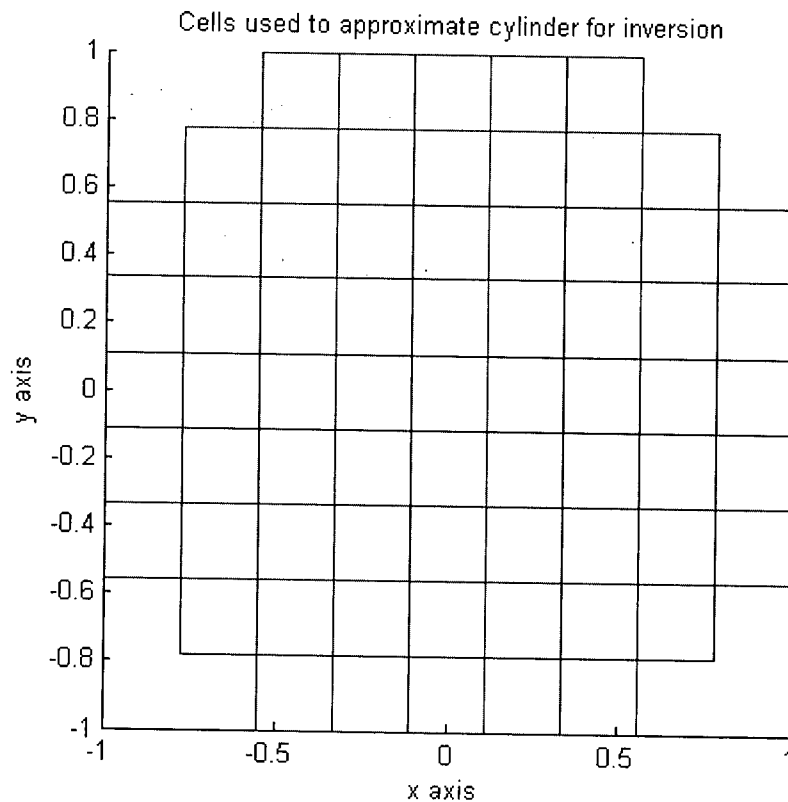


Figure 5.1: Inversion cells for experimental sets 1, 2 and 3.

Figure 5.2 represents the same region as Figure 5.1. The cells in Figure 5.2 are used for Richmond's Method for forward modeling. The permittivity and electric field are assumed to be constant in each region. Each square region represents a modeling cell, with length $2/41\lambda$. The regions are generated with the `square_grid` function as described in Chapter 4.

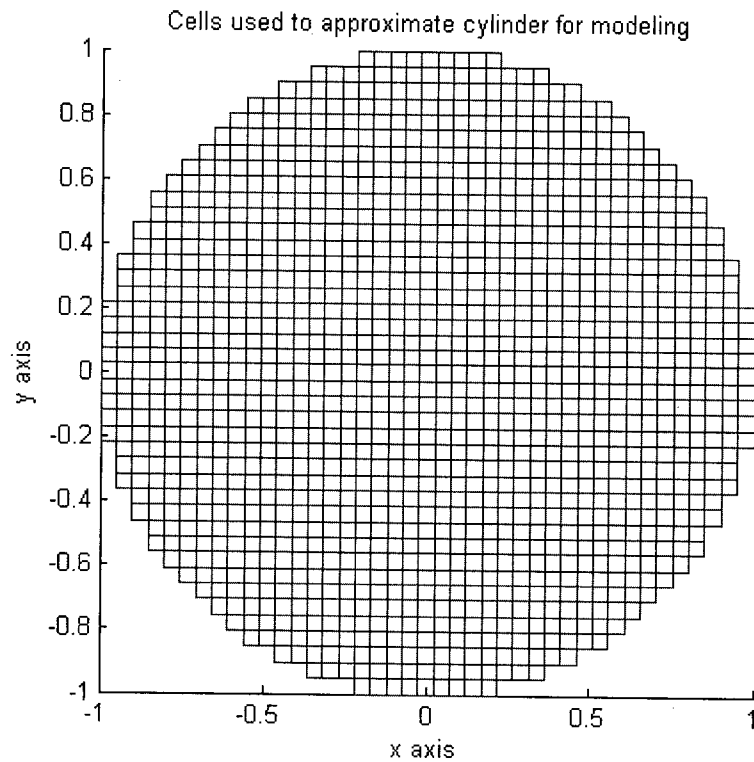


Figure 5.2: Modeling cells for experimental sets 1, 2 and 3.

Figure 5.3 illustrates the superposition of the inversion regions and modeling regions. The figure represents the mapping concept presented in Section 4.1. The square regions represent the inversion cells and the x 's represent the centers of the modeling cells. The x 's within a given square all share the same permittivity value.

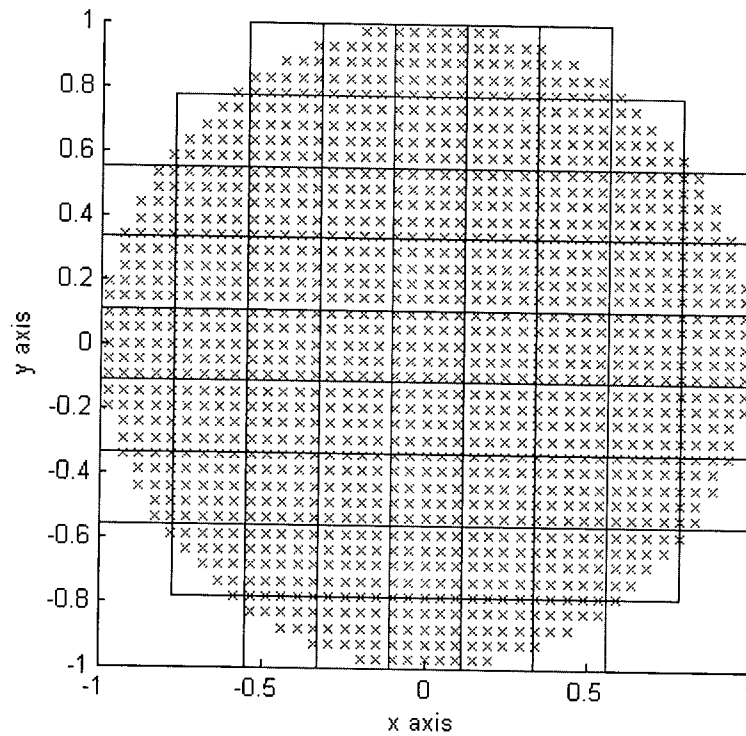


Figure 5.3: Superposition of modeling regions and inversion regions

5.2 Test cases 1,2 and 3

Three test cases are used for experimental sets 1 and 2. The properties of the test case are summarized in Table 5.1. The first test case is shown in Figure 5.4. It has 1 cell with a permittivity of 1, and 68 cells with a permittivity of 2.5. The distribution is asymmetric. The test case will be used in experimental sets 1, 2, and 3.

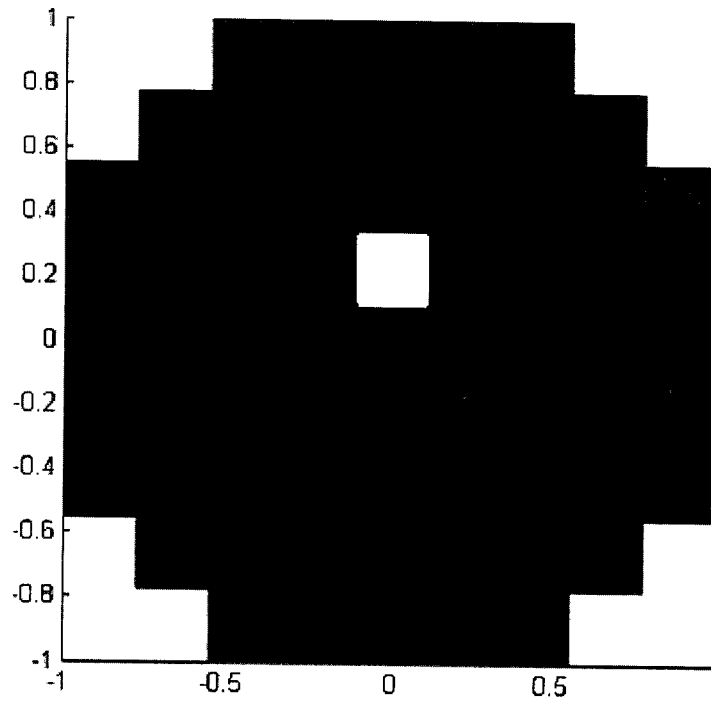


Figure 5.4: Test case 1, background has the permittivity of 2.5 and perturbation has a permittivity of 1.

The second test case is shown in Figure 5.5. It has 4 cells with a permittivity of 1, and 65 cells with a permittivity of 2.5. The distribution is asymmetric. The test case will be used in experimental sets 1, 2, and 3.

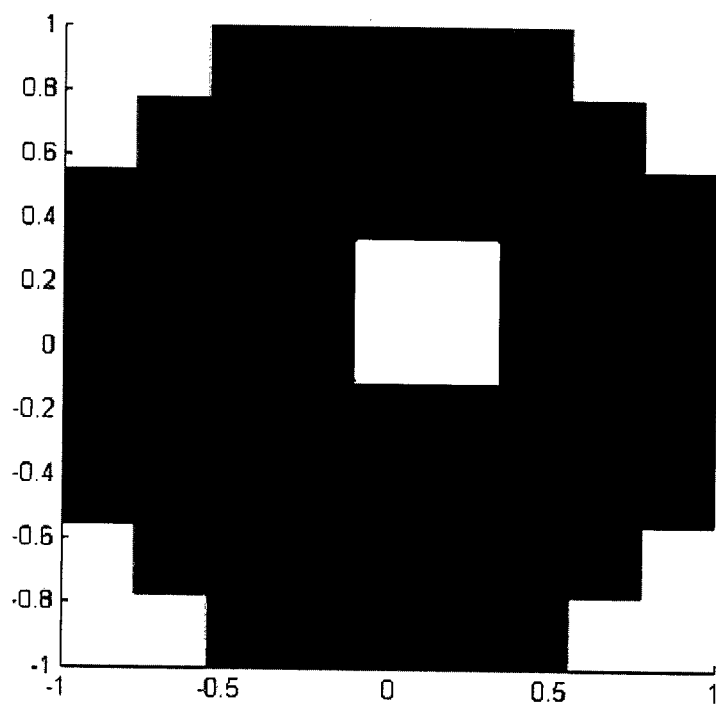


Figure 5.5: Test case 2, background has the permittivity of 2.5 and perturbation has a permittivity of 1.

Test case 3 is shown in Figure 5.6. It has 9 cells with a permittivity of 1, and 60 cells with a permittivity of 2.5. The distribution is symmetric. The distribution is used in experimental sets 1, 2, 3 and 4.

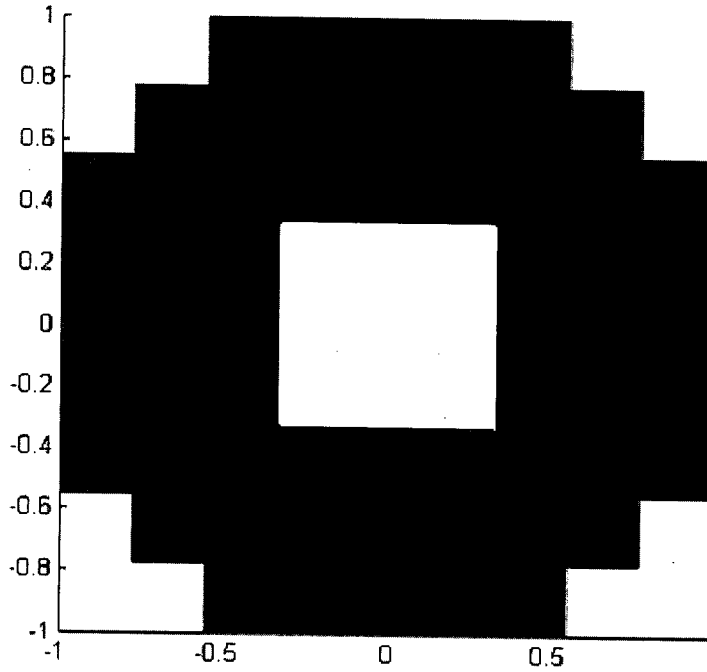


Figure 5.6: Test case 3, background has the permittivity of 2.5 and perturbation has a permittivity of 1.

5.3 Norms for assessing the quality of the solution

Two norms can be defined to help assess the quality of the solution. Because the measured scattered field is simulated, the exact permittivity distribution corresponding to the scattered field is known. By knowing the permittivity distribution, Equation 5.1 can be defined.

$$\Phi_2 = \frac{\|\vec{\epsilon}^{ref} - \vec{\epsilon}\|^2}{N'} \quad (5.1)$$

Equation 5.1 is a measure of the difference between the recovered permittivity distribution and the exact permittivity distribution. Ideally the value of Φ_2 is zero at the end of the algorithm. This term is called the modeling error.

The second norm is a measure of the difference between the measured scattered field and the predicted scattered field. This is known as the variation and is shown in Equation 5.2.

$$\Phi_M = \frac{\|\vec{d}^{ref} - \vec{d}\|^2}{N^o N^f} \quad (5.2)$$

5.4 Experimental set 1

For the experiments in experimental set 1, β is held constant as a function of iteration and the algorithm runs for 9 iterations. The parameters in Table 5.2 are used, the modeling regions are discretized according to Figure 5.2, and the inversion cells are discretized according to Figure 5.1. Seven values of β are used for the three test cases described in Section 5.2 for a total of 21 experiments. The values of β used are 100, 10, 1, 0.1, 0.01, 0.001 and 0. For all the cases a flat starting model of permittivity of 2.5 is assumed.

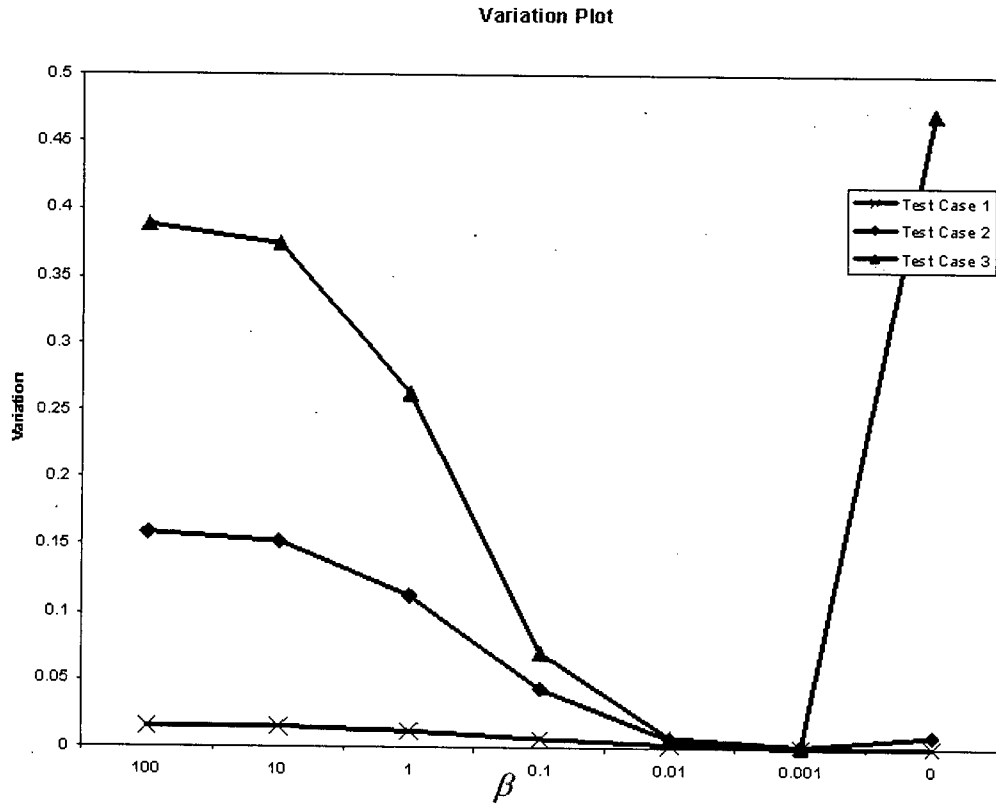


Figure 5.7: Variation plot after 9 iterations as a function β for experimental set 1, test cases 1, 2 and 3.

The graph in Figure 5.7 is the plot of the variation for the experiments in experimental set 1 after the ninth iteration. The line connects experiments that share the same test case. The significant features of the graph include the decreasing variation with decreasing β , and the sudden increase in the variation value when β is equal to zero.

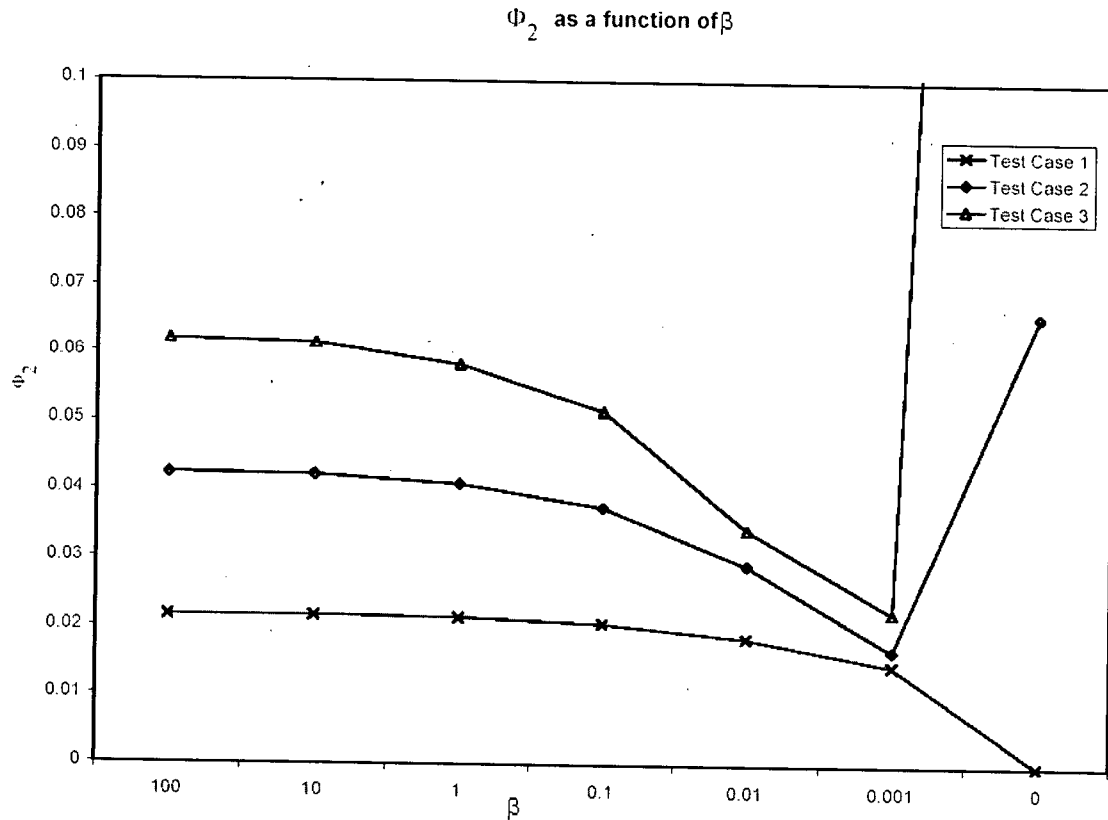


Figure 5.8: Modeling error after 9 iterations as a function β for experimental set 1, test cases 1, 2 and 3.

Figure 5.8 is a plot of the modeling error at the ninth iteration of each experiment. The lines connect experiments that share the same test case. Note the fact that the modeling error decreases as β decreases, and then suddenly increases when β is set to zero.

Examination of test case 3 indicates that the variation is not a good measure of divergence. The variation at $\beta = 0$, is large but finite. The magnitude is of the same order as the value of the variation at $\beta = 100$. The modeling error at $\beta = 0$ is very large. Clearly, the solution has diverged, yet the variation value remains finite. When $\beta = 0$ the inverse problem is ill-posed. Without regularization the inversion algorithm is expected to diverge when the initial model is significantly different from the true model. This

implies that the inclusion of the flatness term in the minimization helps find a stable solution to test case 3.

For test case 2, the values of the variation and modeling error are studied as a function of iteration for $\beta = 0$ and $\beta = 0.001$. This case is examined closely because of the sudden increase in modeling error. It is not clear from looking at Figure 5.7 and 5.8, whether test case 2 is converging at $\beta = 0$.

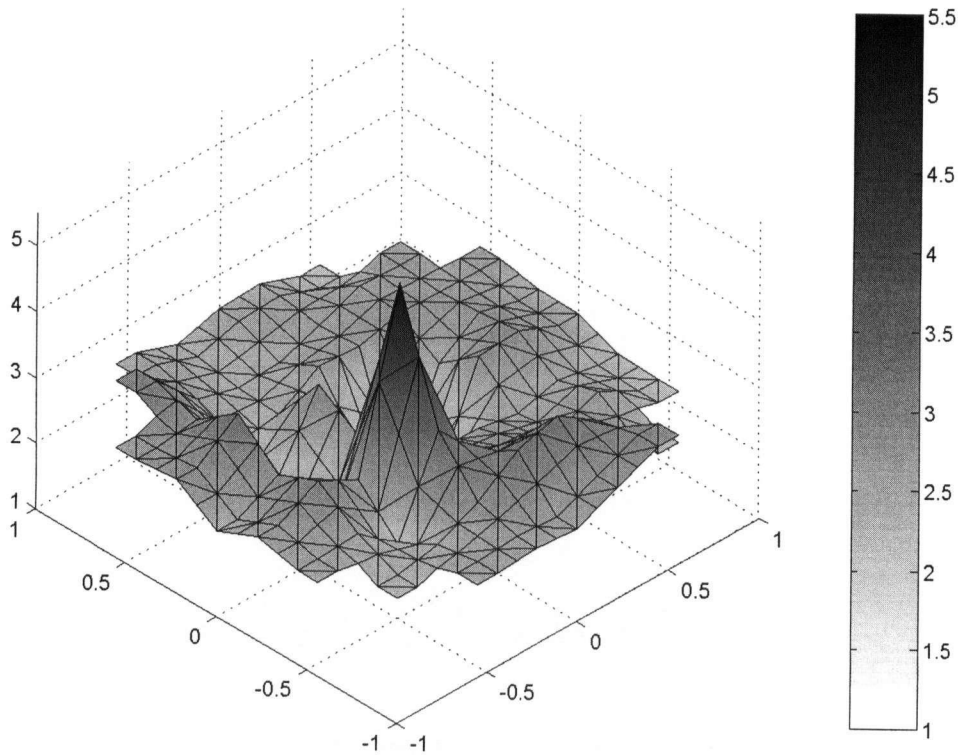


Figure 5.9: Recovered permittivity distribution for experimental set 1, test case 2 and $\beta = 0$.

Figure 5.9 shows the permittivity distribution recovered in test case 2 at the ninth iteration. The surface is bumpy and the peak permittivity value is about 5. This is quite different than that of the actual permittivity distribution.

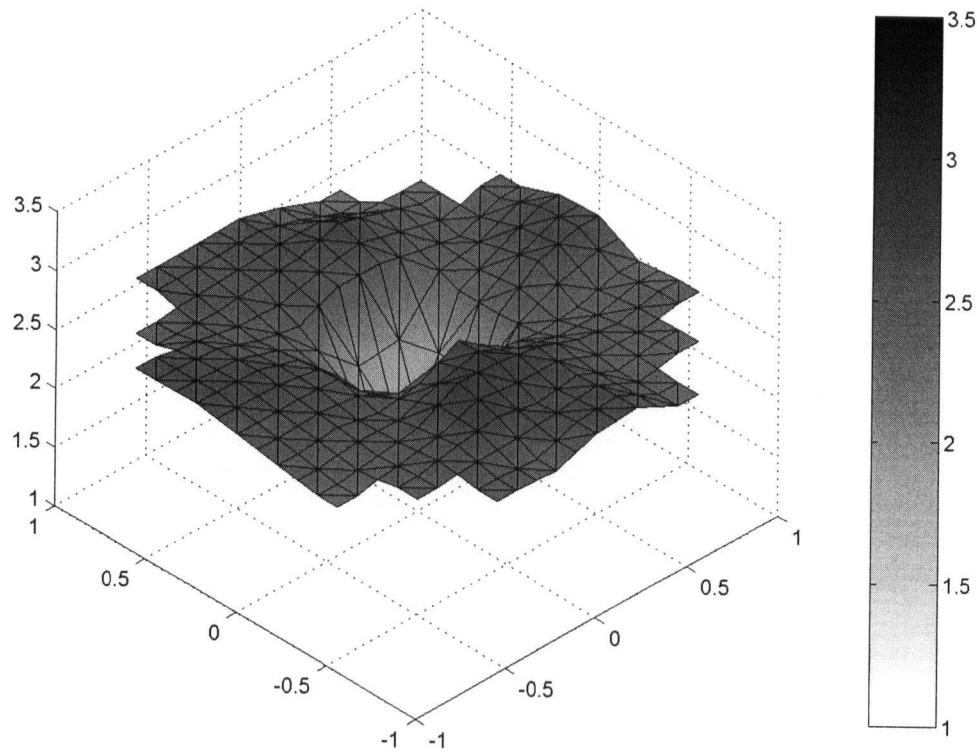


Figure 5.10: Recovered permittivity distribution for experimental set 1, test case 2 and $\beta = 0.001$.

Figure 5.10 is the recovered permittivity distribution for $\beta = 0.001$ at the ninth iteration. The scaling of the graph is different than that of Figure 5.9 to accentuate the features of each graph.

Figure 5.10 looks much more similar to Figure 5.6, than Figure 5.9. This implies that the solution produced with a little regularization is better than the solution with no regularization for test case 2.

Figure 5.11 examines the behavior of the modeling as a function of iteration for the case of $\beta = 0$ and $\beta = 0.001$ for test case 2.

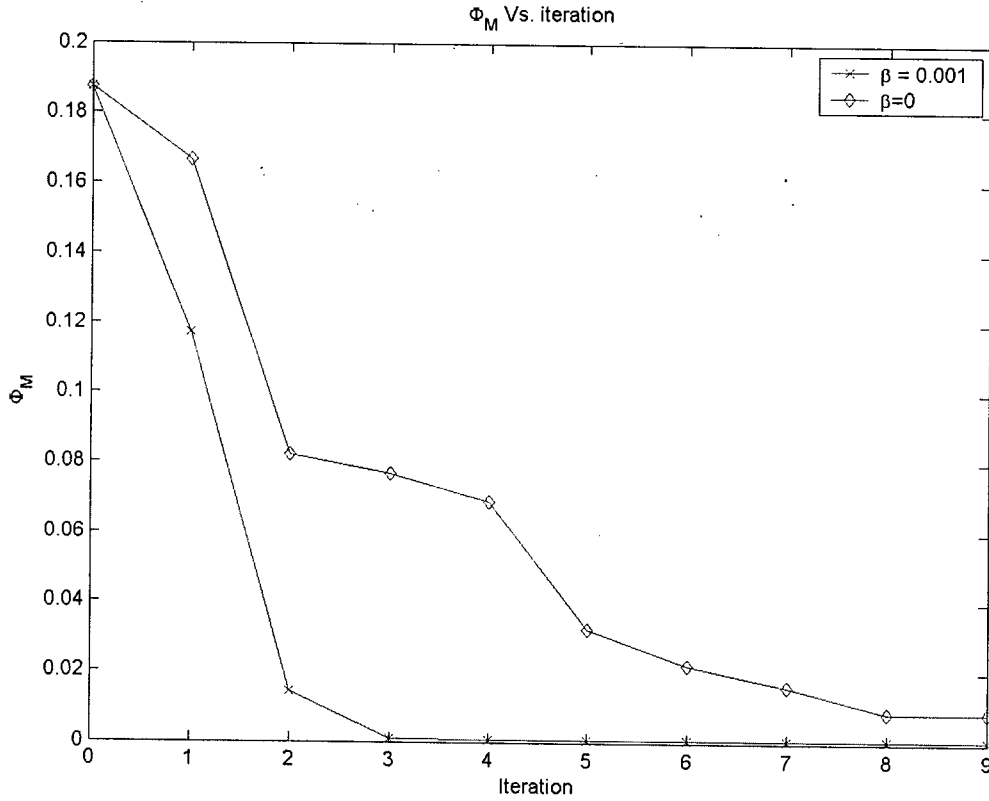


Figure 5.11: Variation as a function of iteration for experimental set 1, test case 2, $\beta = 0$ and $\beta = 0.001$

For test case 2, the $\beta = 0$ line does converge, however the $\beta = 0.001$ line converges much more rapidly in Figure 5.11 and the final variation value is lower than that of the $\beta = 0$ line. This further implies that, when $\beta = 0$, a local minimum has been reached. The regularization helps avoid the local minimum because the local minimum has significantly more structure than that of the global minimum.

The behavior of the variation suggests that slowly decreasing β leads to a better solution for test case 2 and test case 3 compared to when $\beta = 0$.

5.5 Experimental set 2

For the second set of experiments β decreases as a function of iteration. For each of the three test cases, β is decreased in 2 ways. For the first method, β is decreased after every iteration. For the second method, β is decreased after every third iteration. There are a total of six experiments. The values of β used in both methods are 100, 10, 1, 0.1, 0.01, 0.001 and 0. The parameters in Table 5.2 are used for both methods. A flat starting model of permittivity 2.5 is used.

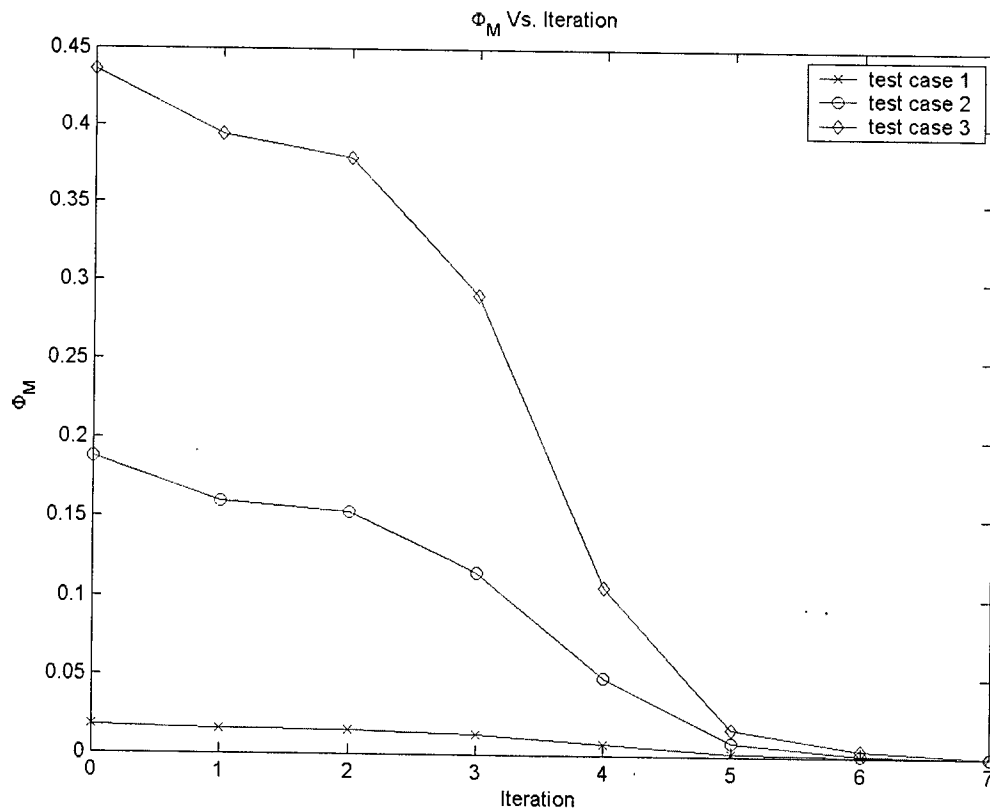


Figure 5.12: Variation plot for experimental set 2, method 1 (decreasing β after every iteration).

Figure 5.12 is a plot of the variation for the first method of decreasing β . The variation decreases after every iteration. At the 7th iteration, the variation is not zero, although from the graph it may appear so.

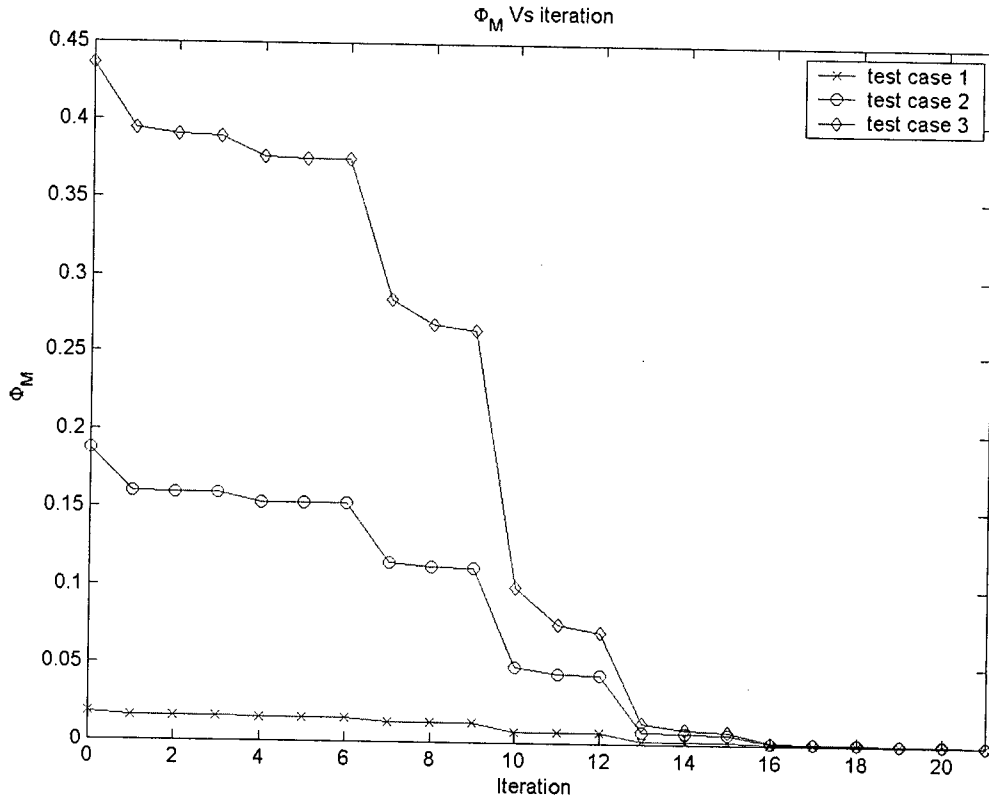


Figure 5.13: Variation plot for experimental set 2, method 2(decreasing β after every third iteration).

Figure 5.15 is a plot of the variation for the second method of decreasing β . The lines represent a specific test case. Similar to Figure 5.14, the variation decreases after every iteration. The variation exhibits large decreases in variation when the value of β is changed. The variation at the 20th iteration is nearly zero for all three test cases.

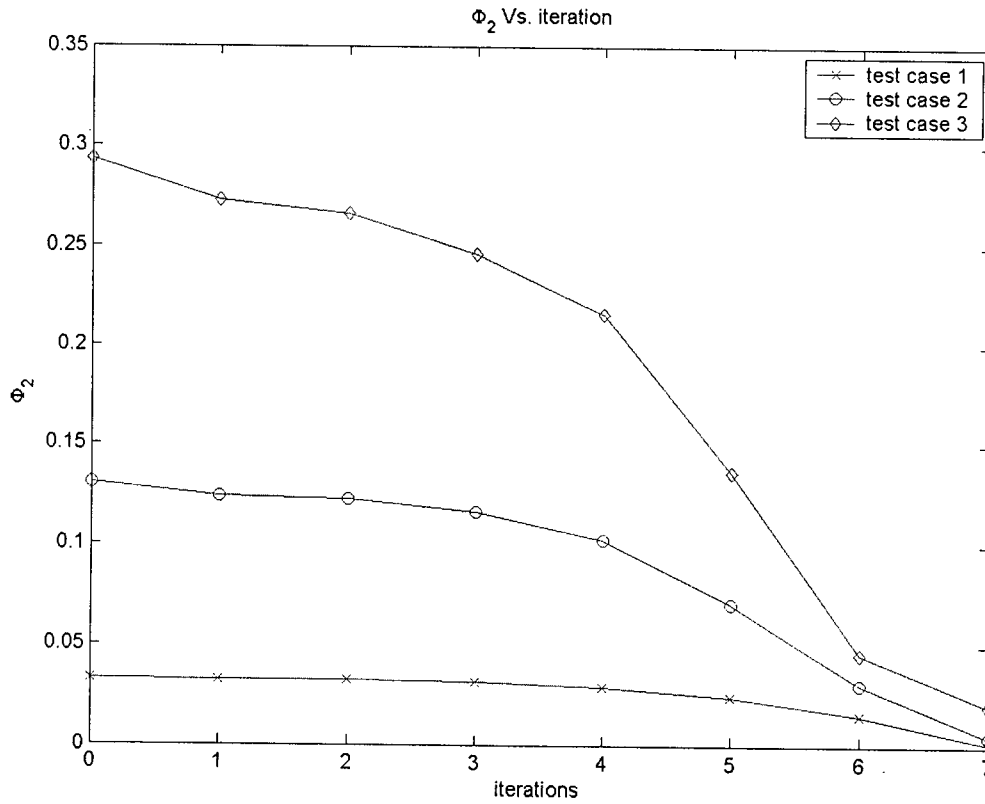


Figure 5.14: Modeling for experimental set 2, method 1(decreasing β after every iteration).

Figure 5.14 plots the modeling error for the case where β is decreased after every iteration. The modeling error decreases after every iteration. The misfit value at the seventh iteration is clearly non-zero for test case three.

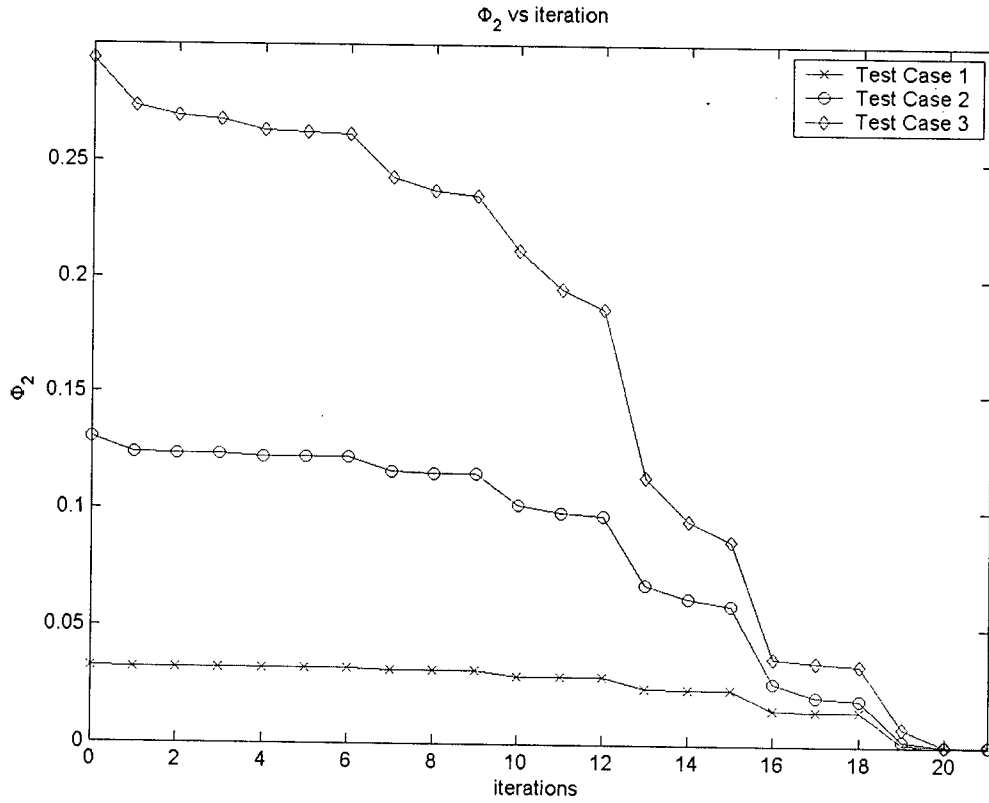


Figure 5.15: Modeling error for experimental set 2, method 2(decreasing β after every third iteration).

Figure 5.15 is a plot of the modeling error for the case where β is decreased after every third iteration.

Both methods of decreasing β improve the variation and modeling error when the final solution is generated. For method 2, when β is decreased after every third iteration, the computed solution comes very close the exact solution. The average error in test case 1 is on the order of 10^{-4} , with similarly small errors for test cases 2 and 3.

The variation and modeling error terms also exhibit the steepest drop when the β value is changed. The model error changes very little in Figure 5.15 when β is held constant. This implies that the system reaches a minimum value for that particular β quickly.

The first method, where the value of β decreases after every iteration, has a larger modeling error. Note that the first method requires 3 times less computation than the second method. The solution to method 1 is not as good as the solution to method 2. Method 1 corresponds to cooling the system too fast. Examining the variation and modeling error, these experiments show the same behavior as the first set of experiments, that is the variation and modeling error decreases after every iteration. The final value of modeling error for test case 3 is lower in method 2 than method 1.

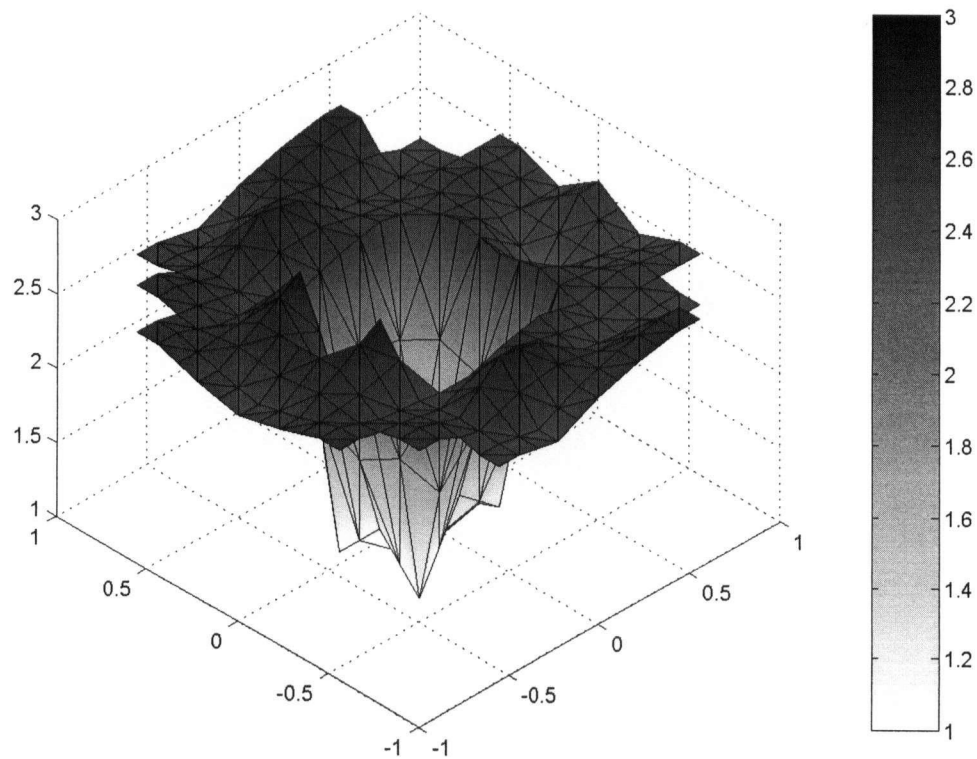


Figure 5.16: Image of permittivity distribution the experimental set 2, test case 3, method 1, and iteration 7.

Figure 5.16 is a surface plot of the permittivity distribution for test case 3 of method 1 at the seventh iteration. The surface looks very rough. The major feature of test case 3 is visible.

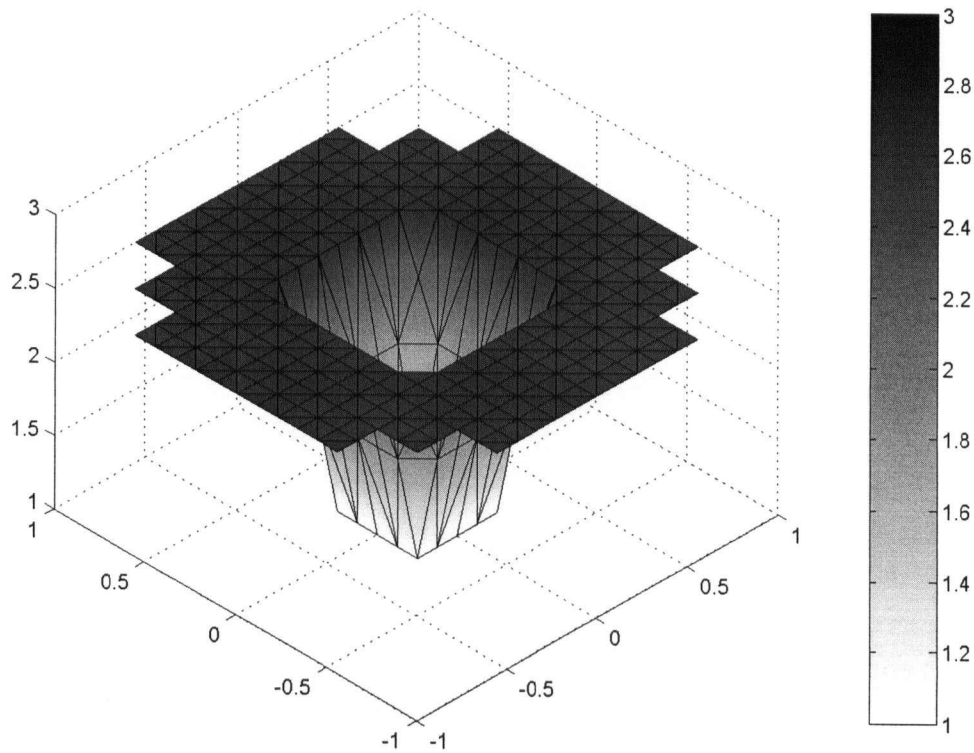


Figure 5.17: Image of permittivity distribution experimental set 2, test case 3, method 2, and iteration 21.

Figure 5.17 is a surface plot of the permittivity distribution of test case 3 of method 2 at the 21st iteration. The solution resembles the exact solution.

Figure 5.17 resembles Figure 5.6 more than Figure 5.16 and demonstrates that the method where decreasing β slowly can find a better solution than the case where β is decreased too quickly.

5.6 Experimental set 3

Experimental set 3 uses the same cooling concept as experimental set 2. There is only one experiment in experimental set 3. A noise signal with a standard deviation of 0.2 is added to the scattered field of test case 3. This is approximately 20% of the maximum

signal. In the presence of noise the cooling needs to be slower. More β values are used in this case than experimental set 2. The value of β decreases after every third iteration. The values 100, 50, 10, 5, 1, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001 and 0 are used. The parameters in Table 5.2 are used. A flat starting model of permittivity 2.5 is used.

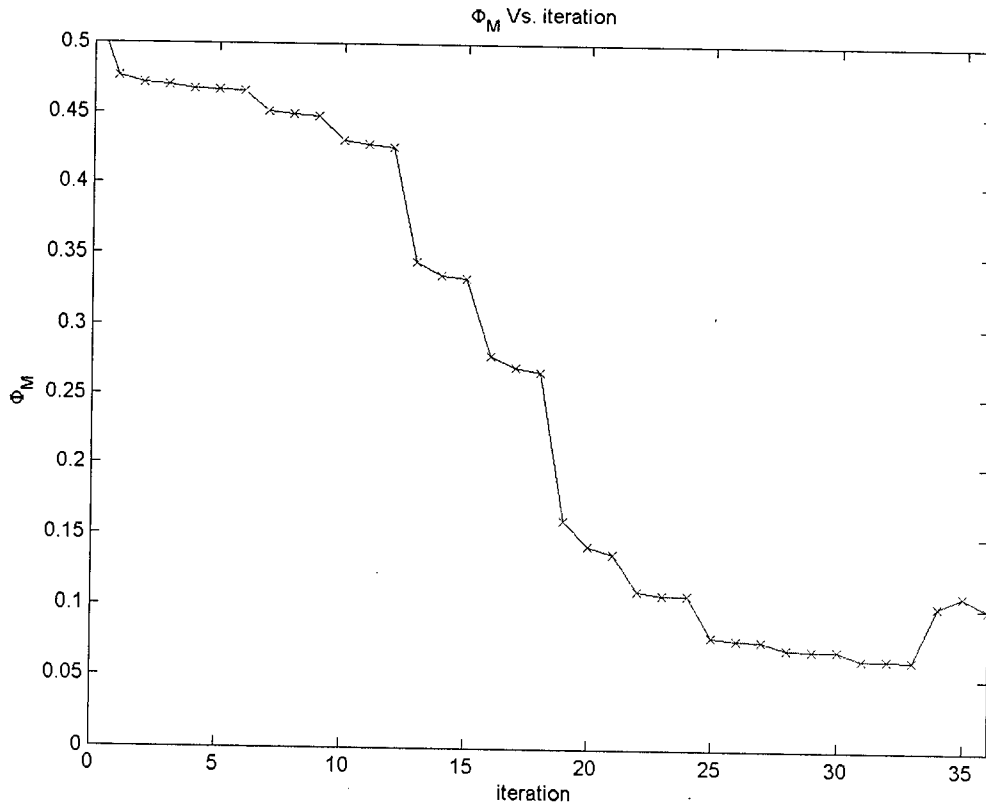


Figure 5.18: Variation plot for experimental set 2 (20% noise), test case 3. Note the nonzero variation value at the last iteration.

Figure 5.18 is a plot of the variation. The value of the variation decreases after every iteration with the sharpest decreases when the value of β changes. Because of noise the value of the variation never reaches zero. When β is set to zero, the variation exhibits a small increase. For the similar test case in experimental set 2, the variation was near zero. The noise is clearly the cause of this behavior.

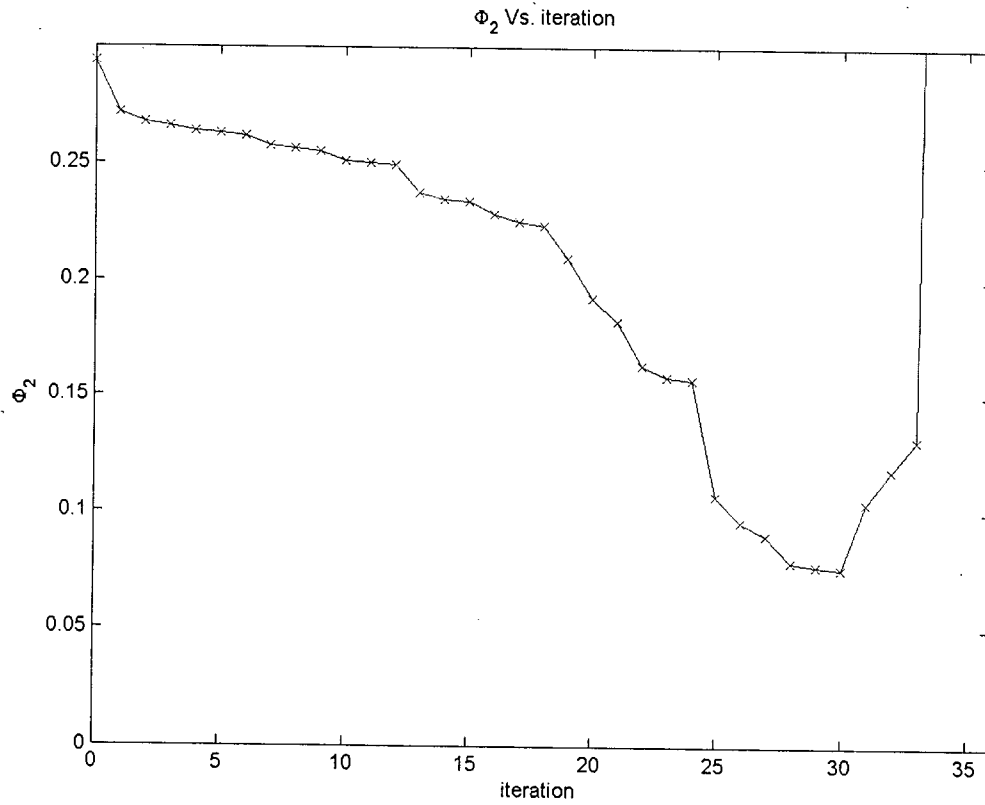


Figure 5.19: Modeling error plot for experimental set 2 (20% noise), test case 3. Note the jump in modeling error at iteration 30.

Figure 5.19 is a plot of the modeling error. When the value of β is set to zero, a sudden jump occurs in the modeling error. There is a corresponding jump in the variation plot. However, the jump in the variation plot occurs later.

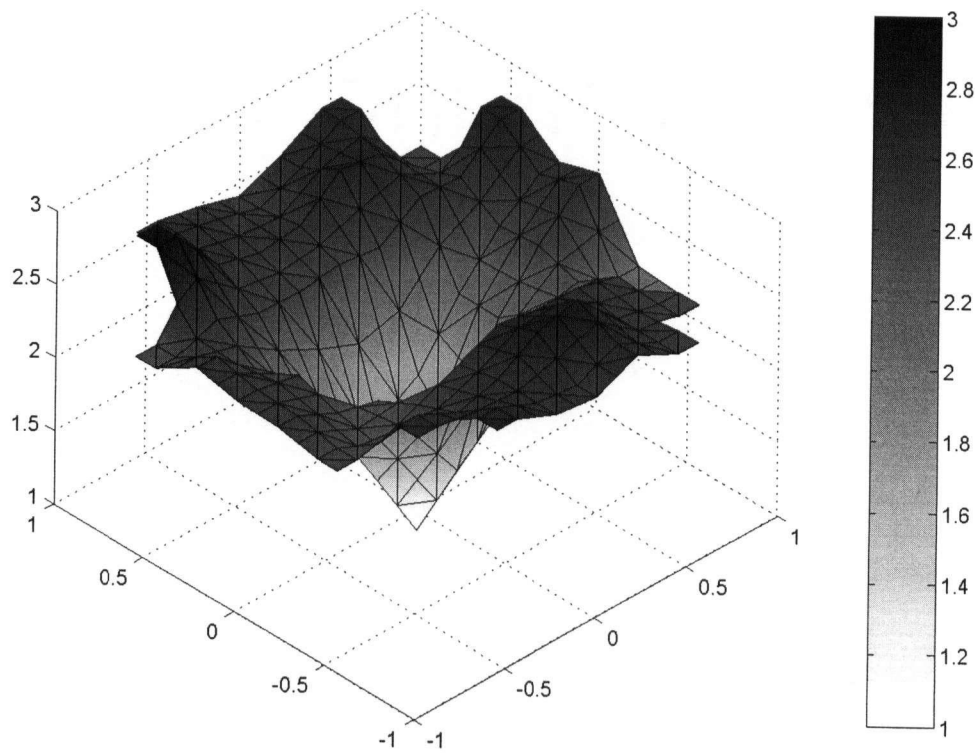


Figure 5.20: Permittivity distribution for experimental set 3, test case 3, and iteration 30.

Figure 5.20 is a plot of the permittivity distribution at the 30th iteration and Figure 5.21 is a plot of the permittivity distribution at the 31st iteration. Figure 5.20 looks much closer to Figure 5.6 than Figure 5.21. Both permittivity distributions display the major feature of a perturbation in the middle. In the presence of noise, small changes to the distribution are not recoverable.

Figure 5.19 clearly illustrates that the regularization parameter cannot be turned off. When the parameter is turned off the system converges to a system completely different from the exact solution. The function Φ_2 decreases with iteration, implying the solution is improving. Only when regularization is removed does the Φ_2 increase, which is opposite to the desired behavior. In the noiseless experiment the exact solution is found when β is set to 0; however, in this experiment the exact solution was not

recovered, due to the inability of the model to match the scattered field exactly. The regularization cannot be turned off in the presence of noise.

Examining the variation in Figure 5.18, it can be seen that the equilibrium variation term is approximately equal to the variation of the noise signal and the data misfit remains small even after β has been set to zero. This is because the permittivity distribution in Figure 5.21 produces a scattered field that matches a noisy scattered field better than that of the scattered field from the permittivity distribution in Figure 5.20. This is expected from the results of Chapter 2. The results from Chapter 2 illustrated that two homogenous cylinders with significantly different permittivities can generate very similar scattered fields. The difference in their permittivity can be smaller than the noise.

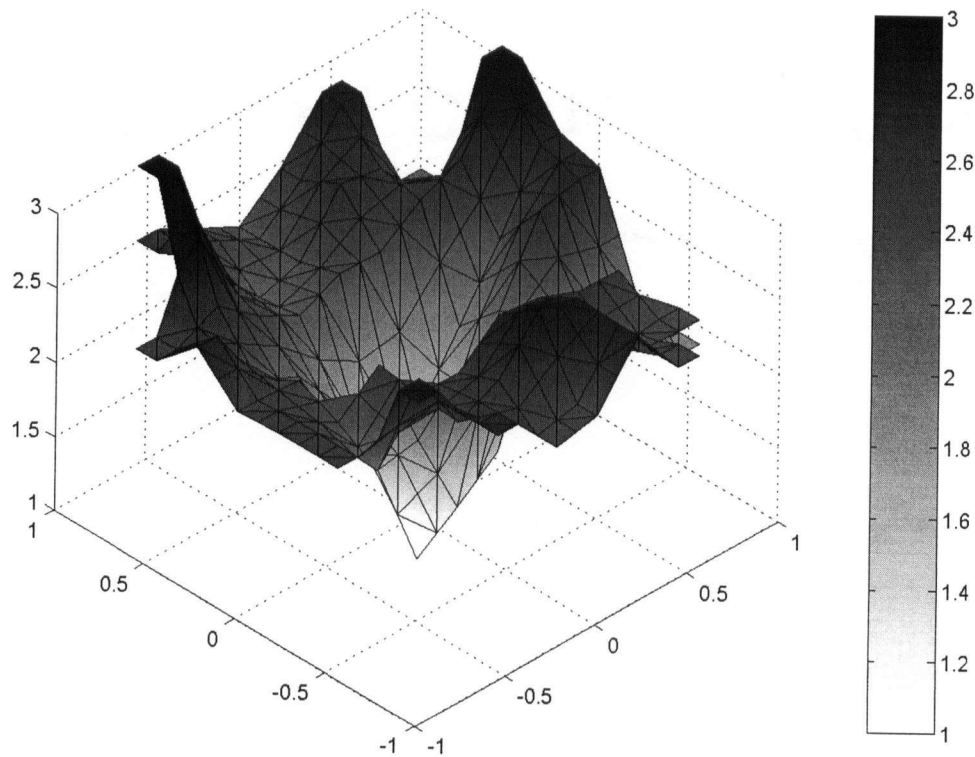


Figure 5.21: Permittivity distribution for experimental set 3, test case 3, and iteration 31.

5.7 Experimental set 4

Experimental set 4 explores the case where there are more parameters than data points. There is one experiment in experimental set 4. β is decreased every iteration. Experimental set 4 uses different parameters from the previous three sets of experiments. The permittivity distribution is shown in Figure 5.22. Again, numerically calculated data are used, and the inversion tessellation can represent the distribution exactly. The parameters are presented in Table 5.3.

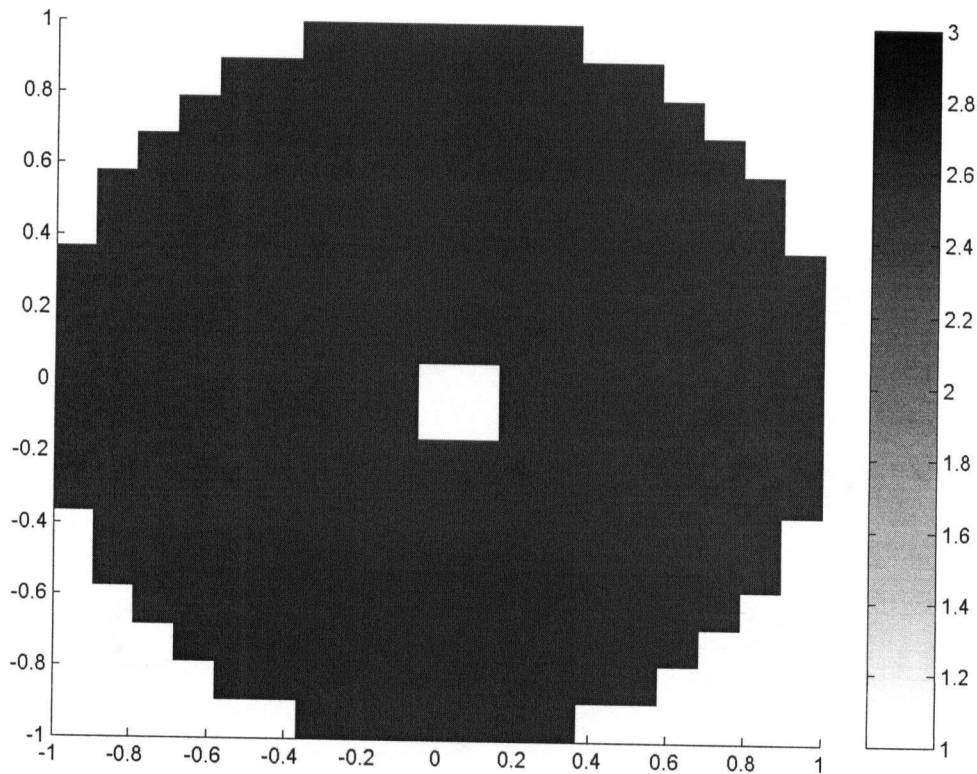


Figure 5.22: Permittivity distribution for experimental set 4. Perturbation has permittivity of 1 and background has permittivity 2.5

The value of β is decreased more slowly than before. The need for more β values is expected in this case because of the higher number of inversion cells causing the system to be more sensitive to small differences. From experimental experience it is

known that more β values need to be used. Forty-one β values are used. This provides a slow convergence to the solution.

Table 5.3: Parameters for experimental set 4

Parameter	Value
Cylinder radius	1λ
k values	$0.80*2\pi$ $0.88*2\pi$ $0.96*2\pi$ $1.04*2\pi$ $1.12*2\pi$ $1.20*2\pi$
Inversion cell tessellation radius	$1/19 \lambda$
Modeling cell tessellation radius	$1/51 \lambda$
Incident wave	e^{jkx}
Observation locations	-90 to 90 degrees with 6 degrees of separation. $\rho = 1.5 \lambda$
β values	10.0000, 8.7500, 7.6563, 6.6992, 5.8618, 5.1291, 4.4880, 3.9270, 3.4361, 3.0066, 2.6308, 2.3019, 2.0142, 1.7624, 1.5421, 1.3493, 1.1807, 1.0331, 0.9040, 0.7910, 0.6921, 0.6056, 0.5299, 0.4636, 0.4057, 0.3550, 0.3106, 0.2718, 0.2378, 0.2081, 0.1821, 0.1593, 0.1394, 0.1220, 0.1067, 0.0934, 0.0817, 0.0715, 0.0626, 0.0547, 0.0479
Iterations per β	1
Starting model	Flat model of permittivity 2.5

Figure 5.23 is a plot of variation with respect to iteration. The variation decreases with every iteration.

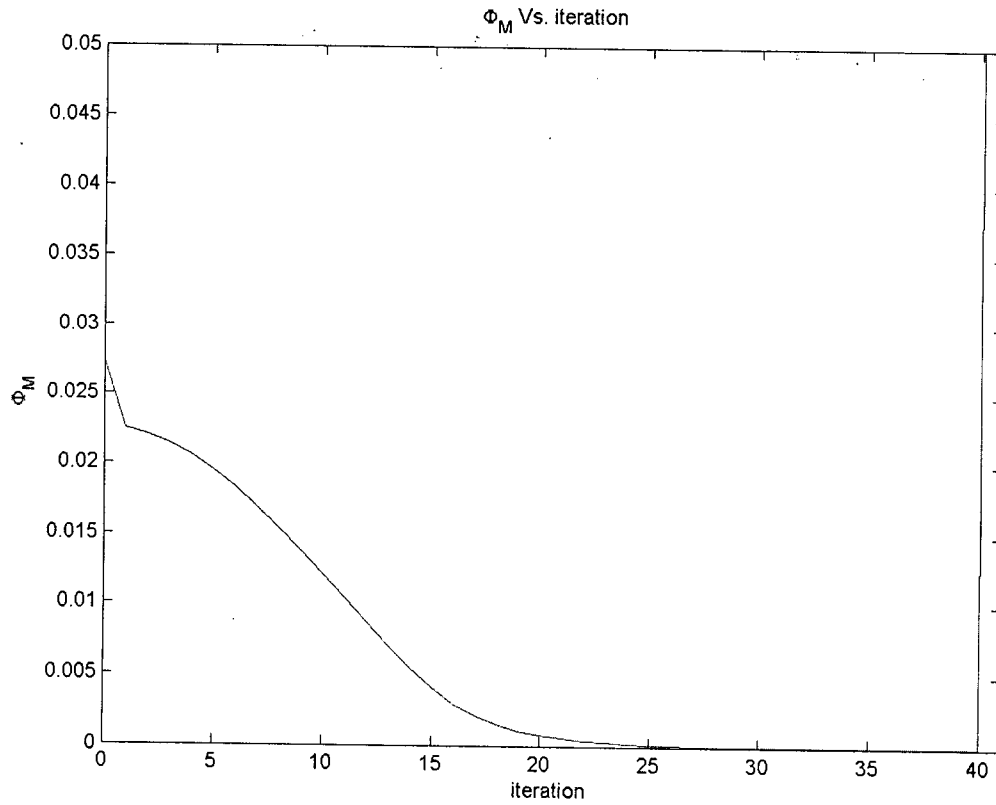


Figure 5.23: Variation plot for experimental set 4, experiment 1. Note at iteration 41, the variation value is off the scale presented, but finite and is approximately 0.5.

Figure 5.24 is a plot of the modeling error. Notice the sudden increase in the modeling error when β is set to zero.

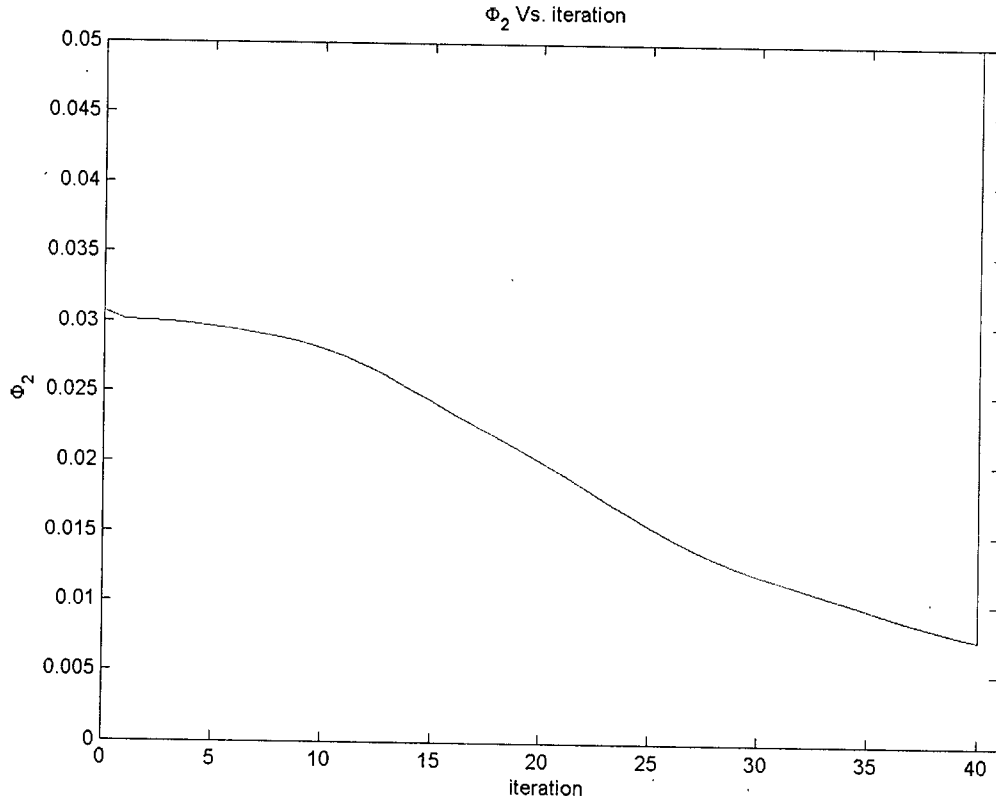


Figure 5.24: Modeling plot for experimental set 4, experiment 1.

From Figure 5.24 and 5.25, it can be inferred that the solution is getting closer with each iteration, however when the regularization is dropped to zero, the solution changes rapidly. This is due to the ill-posedness of the problem. If regularization is never turned off, an approximate solution can be recovered. When regularization is turned off the modeling error becomes large, while the data misfit remains relatively small. This is a result of the system being under-determined. The permittivity values become very large. The modeling grid is no longer adequate for the high permittivity values causing Richmond's Method to fail. This is the reason why the variation value remains small.

5.8 Experimental set 5

Experimental set 5 is the same as Experimental set 4, but with a different permittivity distribution. The permittivity distribution is shown in Figure 5.25. The permittivity distribution contains two blobs which are different than that of the background.

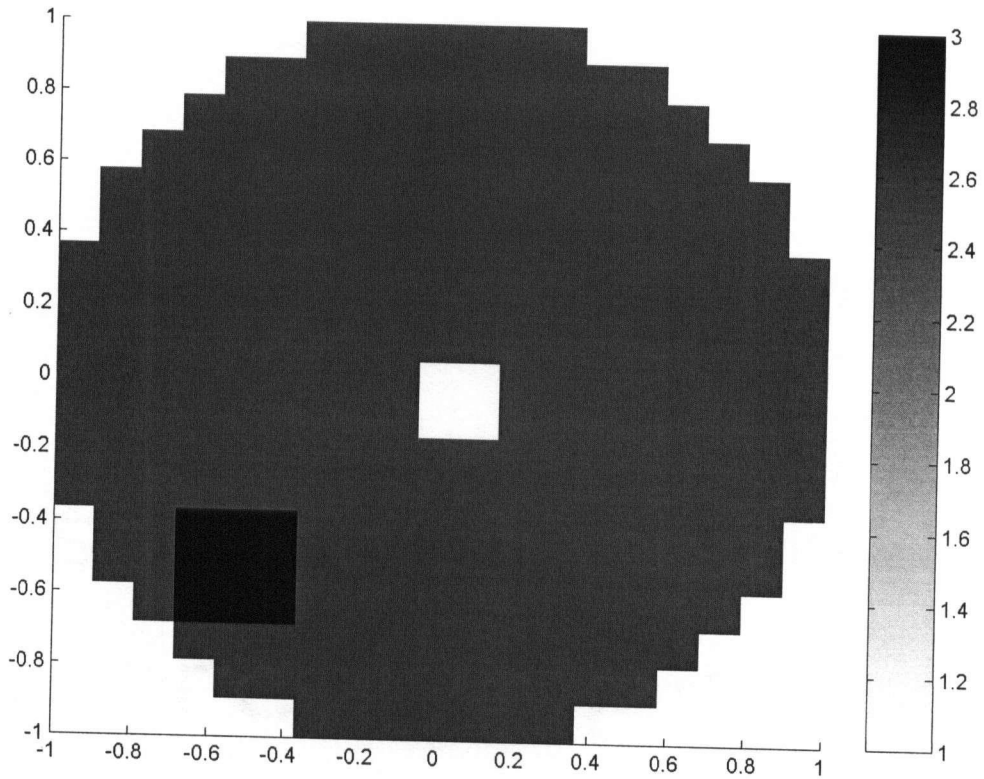


Figure 5.25: Permittivity distribution with two blobs different than the background.

Figure 5.26 the variation demonstrated the same behavior as the previous experiment. The variation decreases until β is turned off. The system diverges when β is turned off.

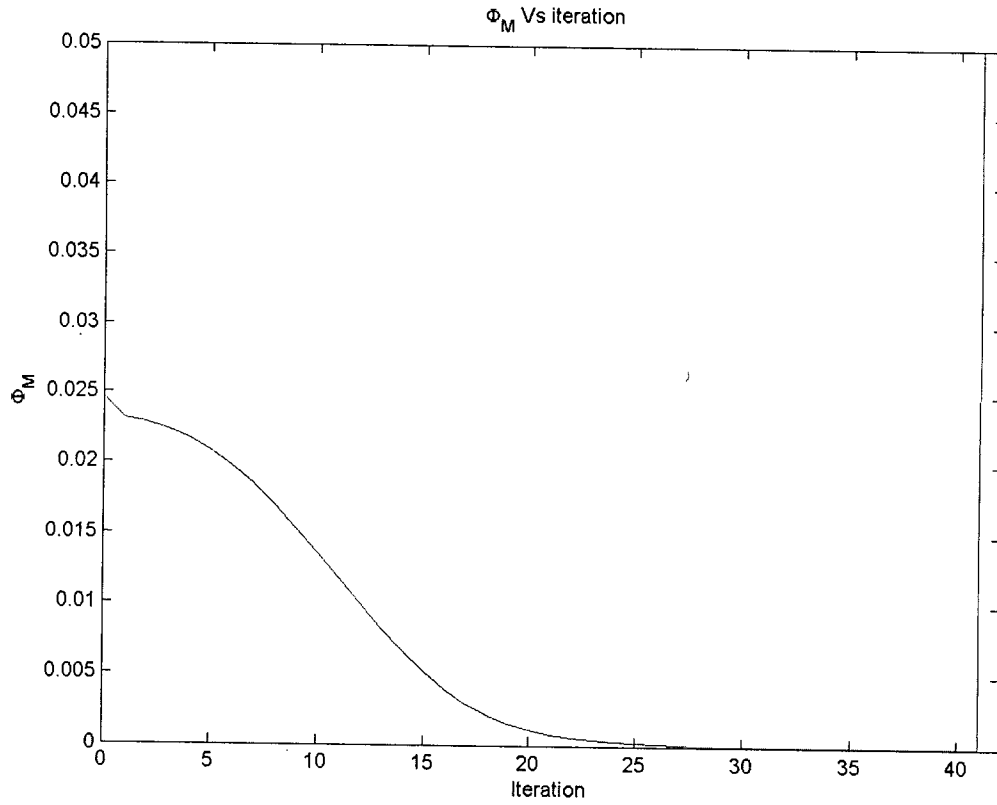


Figure 5.26: Variation plot for 2 blob case.

Figure 5.27 is a plot of the modeling error. It exhibits the same behavior as the Experimental set 4. The modeling error decreases until β is turned off.

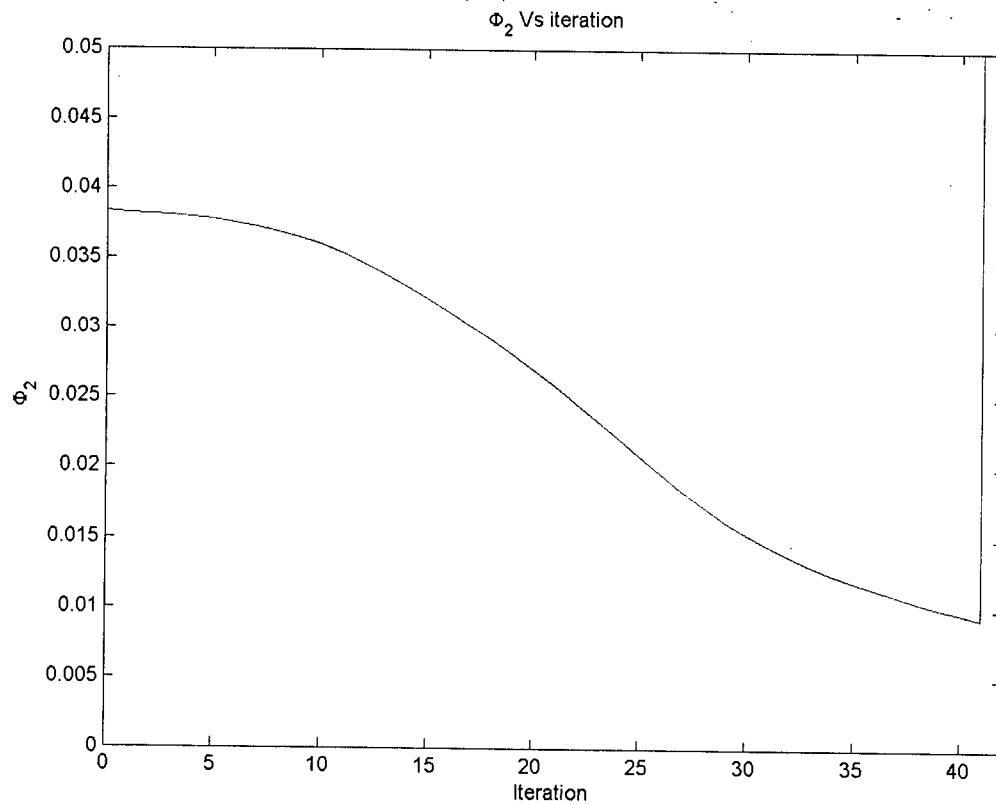


Figure 5.27: Modeling error plot for 2 blob case.

Figure 5.28 represents the recovered permittivity distribution at the 41st, one iteration before the regularization is turned off. The major features in the reference distribution are clearly visible.

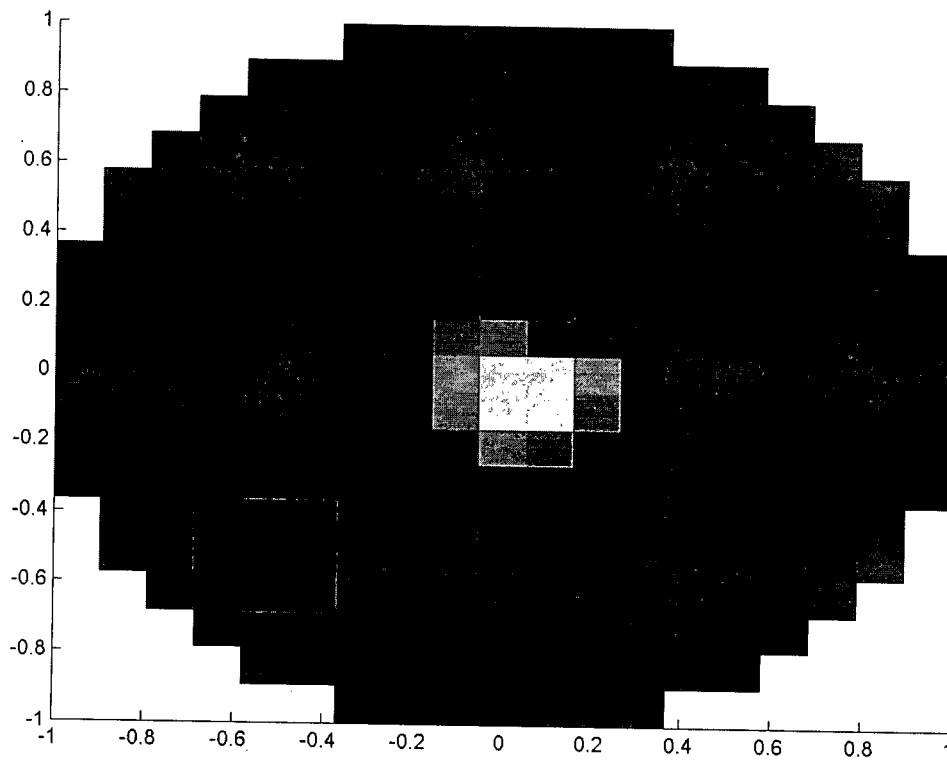


Figure 5.28: Recovered distribution at the 41st iteration

5.9 Conclusion

In this chapter the behavior of the inversion algorithm is examined. It is shown that the addition of the flatness term aids in the search for the solution. In cases where there is a large perturbation from the starting model the regularization yields a better solution than the unregularized solution.

It has also been shown that the exact solution can be found for over constrained cases if β is slowly decreased. The regularization imposed initially favors a flat starting model.

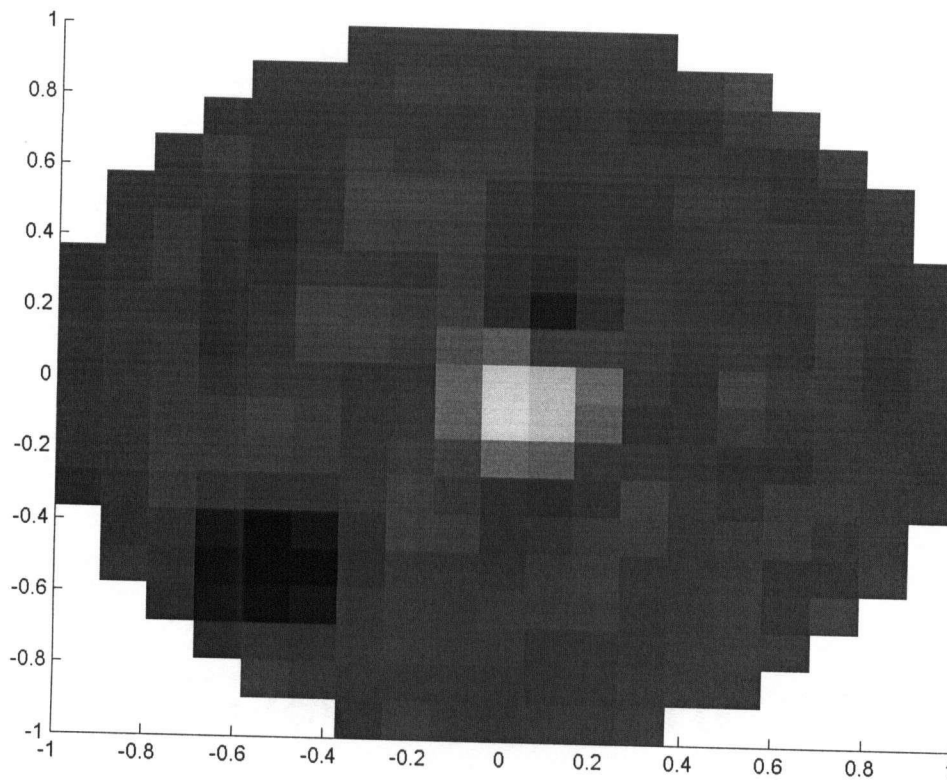


Figure 5.28: Recovered distribution at the 41st iteration

5.9 Conclusion

In this chapter the behavior of the inversion algorithm is examined. It is shown that the addition of the flatness term aids in the search for the solution. In cases where there is a large perturbation from the starting model the regularization yields a better solution than the unregularized solution.

It has also been shown that the exact solution can be found for over constrained cases if β is slowly decreased. The regularization imposed initially favors a flat starting model.

For under constrained cases, where the number of observations are less than the number of inversion cells, the algorithm is still able to recover major features in the permittivity distribution, provided a very slow cooling algorithm is used.

Chapter 6: Summary and future work

6.0 Summary

This thesis examines the problem of recovering the permittivity distribution of an infinite cylinder. The scattered field is influenced by the permittivity distribution; therefore information can be recovered from the scattered field. An optimization technique is used to recover this distribution. Many optimization techniques have been applied to the permittivity distribution recovery problem. Each technique has its strengths. The strength of the technique presented in this thesis is that there is a physical interpretation to the regularization, which differs from previous work in this area.

The inference, from scattering observations, of the parameters of a homogeneous, infinite, circular cylinder is studied in Chapter 2. To solve the scattering problem a Bessel function expansion is used, and the boundary value problem for each Bessel function mode is solved. A quasi-periodic behavior in the scattered field is discovered. This quasi-periodic behavior is problematic because for a single frequency, two cylinders with different permittivity values can cause similar scattered fields. For optimization this implies that the wrong permittivity value can be found by the algorithm. This non-uniqueness problem is solved by the use of multiple frequencies in the recovery algorithm.

In Chapter 3, Richmond's Method for forward modeling is presented and an inverse modeling algorithm is developed. Richmond's Method is used to model the scattering behavior of an inhomogeneous cylinder. From Richmond's Method the Jacobian of the predicted scattered field with respect to a guessed permittivity distribution is calculated. Knowing the Jacobian matrix, a linear approximation to the predicted

scattered field can be generated. To develop the inverse modeling algorithm an objective functional is minimized. The objective functional is the sum of two norms. The first norm term is the difference between the measured scattered field and a predicted scattered field. The second norm term is a measurement of the structure in the guessed permittivity distribution. To minimize the objective functional, the predicted scattered field is replaced by a linear approximation. The permittivity distribution that minimizes the objective function with the linearized scattered field can be found with a simple matrix inverse operation. To find the minimum of the original objective functional, this process is repeated using the permittivity distribution obtained from the previous iteration. The iteration process is repeated several times to find the minimum of the objective functional. A minimum of the objective functional occurs at the point where the exact permittivity distribution obtained and the numerically computed permittivity distribution are close to each other. The iterative algorithm is implemented in MatLab and the implementation is presented in Chapter 4.

The weighting between the structure in the permittivity distribution and the scattered field is explored in Chapter 5. A homogenous model with permittivity 2.5 is used as the starting model. A constant weighting with respect to iteration is initially explored and it reveals that regularization aids in the convergence of the solution. When the weighting is large, the resulting solution looks flat. When small, a solution close to the exact one is found by the algorithm. Next, the weighting of the model structure with respect to iteration, is decreased, which is analogous to cooling in simulated annealing. The cooled system can now recover the permittivity distribution. When the system cools too quickly, the correct solution is not recovered. Rather, a solution close to the exact

solution is found. In the presence of noise or in a case where there are more degrees of freedom than observations, the weighting of the model structure cannot be zero, implying the necessity of a norm-based regularization.

6.2 Future work

There are three directions for future work. The first direction is to implement a real system that uses the algorithm to examine long cylindrical structures, such as trees. Such a system currently exists [Dotto, 2003]. There are a number of challenges associated with the measurement of the electric field. These challenges include precision measurement of the incident and scattered fields, precision measurements of the geometry, and simulating the signal received by an antenna. A dipole antenna can be used to simplify the conversion between electric field and received signal. The current algorithm needs to be augmented by weighting the data with the measurement errors and including a χ^2 misfit criterion to terminate the inversion iteration.

The second direction is to theoretically extend the algorithm. The algorithm can be extended to work with complex permittivity as well as 3 spatial dimensions. The speed of the algorithm can also be improved. One possible way to increase the speed is to explore an iterative technique to the LU decomposition. A different forward modeling algorithm can also be used to reduce the computation time.

A third direction is the exploration of non-linear inversion and optimization in different electromagnetic problems, such as antenna design. Different model norms can be used in this type of parametric inversion.

Bibliography

K. Dotto, "Development of an ultra wide band antenna", Preliminary draft of PHD thesis, unpublished, 2003.

E. J. Bond, X. Li, S. C. Hagness, and B. D. Van Veen, "Microwave Imaging via Space-Time Beamforming for Early Detection of Breast Cancer", IEEE Transactions on Antennas and Propagation, Vol. 51, pp1690-1705, August 2003

S. Caorsi, M. Donelli, D. Franceschini, and A. Massa "A New Methodology Based on an Iterative Multiscaling for Microwave Imaging", IEEE transactions on Microwave theory and technique, vol. 51, pp 1162-1173, April 2003.

S. Y. Semenov, R.H. Svenson, A. E. Bulyshev, A. E. Souvorov, A. G. Nazarov, Y. E. Sizov, V. G. Posukh, A. Pavlovsky, P. N. Repin, A. N. Starostin, B. A. Voinov, M. Taran, G. P. Tatsis, and V. Y. Baranov, "Three-Dimensional Microwave Tomography: Initial Experiments Imaging of Animals", IEEE transactions on Biomedical Engineering, Vol. 49, pp. 55-63, January 2002.

M. Tanaka and J. Ogata, "Fast Inversion Method for Electromagnetic Imaging of Cylindrical Dielectric Objects with Optimal Regularization Parameter", vol. 84, pp2560-2565, December 2001

C. J. Lin and C. Chui "Image reconstruction of buried Dielectric Cylinder by TE Wave illumination" IEEE MTT-S IMOC 2001 Proceeding 213-216 (2001)

Semenov, S.Y.; Svenson, R.H.; Bulyshev, A.E.; Souvorov, A.E.; Nazarov, A.G.; Sizov, Y.E.; Pavlovsky, A.V.; Borisov, V.Y.; Voinov, B.A.; Simonova, G.I.; Starostin, A.N.; Posukh, V.G.; Tatsis, G.P.; Baranov, V.Y. "Three-dimensional microwave tomography: experimental prototype of the system and vector Born reconstruction method", IEEE transactions on Biomedical Engineering, Vol. 46, pp. 937-946, August 1999.

M. Pastorino "Short-Range Microwave Inverse Scattering Techniques for Image reconstruction and Applications", IEEE Transactions on Instrumentation and Measurement 47(6) 1419-1427 (1998)

C. G. Farquharson and D.W. Oldenburg, "Non-Linear Inversion Using General Measures of Data Misfit and Model Structure" International Journal of Geophysics, Vol. 134, pp213-227, 1998.

S. Hagness, A. Taflove, and J. Bridges. "Two-Dimensional FDTD Analysis of a Pulsed Microwave Confocal System for Breast Cancer Detection: Fixed-Focus and Antenna-Array Sensors", IEEE Transactions on Biomedical Engineering, Vol. 45 pp 1470-1479, December 1998

A. Souvorov, A. Bulyshev, S. Y. Semenov, R. H. Svenson, A. G. Nazarov, Y. E. Sizov, and G. P. Tatsis "Microwave Tomography: A Two-Dimensional Newton Iterative Scheme", IEEE Transactions on Microwave Theory and Technique, vol. 46 pp1654-1659, November 1998

I. R. Ciric and Y. Qin " Self-adaptive Selection of the Regularization parameter for Electromagnetic Imaging." IEEE Transactions on Magnetics, 33(3) 1556-1559 (1997)

W. R. Hendee and P. N. T. Wells, *The Perception of Visual Information*, Springer-Varlag , New York, 1997

G. H. Golub and C. F. Van Loan, *Matrix Computations*, Third edition Maryland: The John Hopkins University Press, 1996

I. R. Ciric and Y. Liu " An adaptive Algorithm for Electromagnetic Imaging", IEEE Transactions on Magnetics 32(3) 1303-1305 (1996)

W.C. Chew and J. H. Lin "A Frequency-Hopping Approach for Microwave Imaging of Large Inhomogeneous Bodies", IEEE Microwave and Guided Wave Letters 5(2) 439-441 (1995)

S. Caorsi, G.L. Gragnani, et al "Microwave Imaging Method Using a Simulated Annealing Approach", IEEE Microwave and Guide Wave Letters 1(11) 331-333 (1991)

S. Caorsi, G. L. Gragnani, et al " A Multiview Microwave Imaging System for Two-Dimensional Penetrable Objects." IEEE Transactions on Microwave Theory and Techniques 39(5) 845 – 851 (1991)

I. Joachimowicz, C. Pichot, and J. Hugonin "Inverse Scattering: An iterative Numerical Method for electromagnetic Imaging" IEEE Transactions on Antennas and Propagation, 39(12) 1742-1752 (1991)

Y.M. Wang and W.C. Chew, "An Iterative Solution of the Two-Dimensional Electromagnetic Inverse Scattering Problem" International Journal of Imaging Systems and Technology, vol. 1, no. 1, pp 100-108, 1989.

D.T. Borup and O.P. Gandhi, "Calculation of high resolution SAR distributions in biological bodies using the FFT algorithm and conjugate gradient method," IEEE transactions on microwave theory and technique, vol 33 pp 417-419, may 1985.

M. Slaney, A. C. Kak, and L. Larsen, "Limitations of Imaging with First-Order Diffraction Tomography", IEEE Transactions on Microwave Theory and Technique, vol. 32 pp860-874, August 1984

J. H. Wilkinson and G. Peters, "The Least Squares Problem and Pseudo-Inverses," The Computer Journal, Vol. 13, No. 3, August, 1970.

J. H. Richmond, "Scattering by a Dielectric Cylinder of Arbitrary Cross-Section Shape," IEEE Transactions on Antennas and Propagation, vol. 13, pp. 334-341, May 1965.

R.F. Harrington, "Time-Harmonic Electromagnetic Fields" McGraw-Hill Book Company, Toronto, 1961.