2008-07-15

Inverse Scattering Image Quality with Noisy Forward Data

Thomas J. Sorensen
Brigham Young University - Provo

INVERSE SCATTERING IMAGE QUALITY WITH NOISY FORWARD DATA

by

Thomas Jay Sorensen

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Brigham Young University

August 2008

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Thomas Jay Sorensen

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Karl F. Warnick, Chair
Date	David G. Long
Date	Brian D. Jeffs

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Thomas Jay Sorensen in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Karl F. Warnick
Chair, Graduate Committee

Accepted for the Department

Michael J. Wirthin
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of
Engineering and Technology

ABSTRACT

INVERSE SCATTERING IMAGE QUALITY WITH NOISY FORWARD DATA

Thomas Jay Sorensen

Department of Electrical and Computer Engineering

Master of Science

Image quality metrics for several inverse scattering methods and algorithms are presented. Analytical estimates and numerical simulations provide a basis for poor image quality diagnostics. The limitations and noise behavior of reconstructed images are explored analytically and empirically using a contrast ratio. Theoretical contrast ratio estimates using the canonical PEC circular cylinder are derived. Empirical studies are conducted to confirm theoretical estimates and to provide examples of image quality vs SNR for more complex scatterer profiles. Regularized sampling is shown to be more noise sensitive than tomographic reconstructive methods.

ACKNOWLEDGMENTS

I offer my graditude and acknowledgement to several influencial people involved in this work:

Dr. Karl Warnick, my academic advisor, for his knowledge, guidance, patience, and willingness to always help.

Dr. Brian Jeffs for his adoption of me into the Radio Astronomy group.

Dr. David Long for allowing me to take up space in the MERS lab, and for his desire to help students achieve.

Dr. Michael Rice for soccer.

Jacob Waldon, Jonathon Landon, Alan Stemmons, Mike Elmer, David Jones and other members of the Radio Astronomy for their assistance and friendship, and for allowing me to distract them.

All in the MERS lab for good conversations and answers to my questions.

To my family and friends for their encouragement, example, knowledge and money.

Table of Contents

List of Figures

Chapter 1

Introduction

Scientific exploration of solutions to the inverse scattering problem has increased due to its importance in a wide range of applications such as target detection, medical imaging, geophysical exploration, and non-destructive testing. As technology progresses the desire and ability to achieve more accurate solutions to the inverse scattering problem grows. Quantitative comparisons between different algorithms and classes of algorithms help in identification of the cause of poor reconstruction image quality. By diagnosing the causes of poor image quality, inverse scattering methods may be improved and better employed. To aid in the understanding of the causes of image quality, this thesis explores analytical and empirical behavior of several inverse scattering methods with varying levels of noise in the forward data.

Generally, two classes of inverse scattering algorithms exist: those of nonlinear optimization and those of linear approximations. The first class uses complex, time-consuming optimization procedures to iterate toward a solution. These methods may rely on some *a priori* knowledge of the scatterer, such as boundary condition. The second approach makes assumptions about the scatterer in order to make approximations that linearize the problem. Of course, not all methods fit into these two categories. A recent approach that has garnered attention is linear sampling [1, 2, 3]. Linear sampling has gained notoriety for being a simple method that does not rely on any approximations nor assumptions about the scatterer.

Theory predicts that perfect reconstruction can be obtained in an ideal setting. Previous research has been performed that explains under what conditions the inverse scattering problem is uniquely solvable. Under surprisingly weak assumptions, and for a variety of boundary value problems, perfect reconstruction can be obtained

with noise-free forward data. In reality, obtaining a unique solution, and thus perfect reconstruction image quality, is restricted by noise and limitations on the aperture.

Previous empirical work and theoretical proofs begin to explain the quality of reconstruction that is obtainable under non-ideal or noisy conditions. Image resolution for specific linear and non-linear inverse methods has been studied empirically [4, 5]. A comparison of noise propagation and image variances in linear tomographic methods [6] provides a coupling of analytical and empirical analysis of some classical imaging methods. Bucci *et al.* [7] considered inherent limitations on the ability to obtain uniqueness of scattered fields in the presence of noise. ϵ-uniqueness theorems introduced by Potthast [8] state that with limited or noisy data, solutions can be obtained to within a fixed error bound. Image reconstruction accuracy in the low noise regime has been provided for electromagnetic regularized sampling and linear holographic backpropagation tomography [9]. Several studies provide noise characteristics for some different variations of diffraction tomography and non-linear inverse scattering for the acoustic case, but not for linear sampling. Anastacio *et al.* [6] derived analytical image variances and covariances for direct Fourier and filtered backpropagated diffraction tomography as well as for a new reconstruction algorithm. It is shown, through numerical simulations of inverse scattering of a Shepp-Logan phantom, that backpropagated diffraction tomography is less noise sensitive than direct Fourier diffraction tomography and the new reconstruction algorithm. The difference in noise sensitivity is mainly attributed to the ability of filtered backpropagation to use sampling redundancies and the direct Fourier diffraction tomography's reliance on interpolation. Lastly, an analysis of the noise propagation characteristics for acoustic diffraction tomography and non-linear inverse scattering [10] defined image covariance matrices and image pixel variances. Specifically, for linear Born approximated inverse scattering the noise variance is constant over the whole image and the off-diagonal elements of the covariance matrix are peaked around the diagonal and decay rapidly while oscillating. This is in qualitative agreement with filtered backpropagation tomography described in [6]. The noise characteristics of diffraction tomography and its variants are relatively well understood.

Since perfect reconstruction cannot be obtained in practice, it is important to know how specific methods behave in the presence of noise. Knowing how specific solutions are affected at different SNR values provides a better understanding of which methods may be useful for a specific application, and can lead to better algorithms for the inverse problem.

The noise sensitivity of many methods, including regularized sampling, are not yet fully documented. The main purpose of this work is to create a basis for analyzing inverse scattering image quality as a function of forward scattering SNR. This is done by developing an analytical and empirical contrast ratio which compares different classes of inverse scattering reconstructed images equally. Specifically using the PEC circular cylinder, analytical estimates of the contrast ratio for regularized sampling and holographic backpropagation tomography are obtained. Numerical contrast ratio curves are simulated using noisy forward data from PEC scatterers for regularized sampling, holographic backpropagation tomography, and diffraction tomography in order to confirm theoretical estimates and compare noise sensitivity between algorithms.

1.1 Thesis Contributions

The main objectives and contributions contained in this thesis are

- Creation of a measure designed to compare inverse scattering images as a function of SNR.
- Theoretical image quality estimates via the contrast ratio for the PEC circular cylinder.
- Numerical implementation of several inverse scattering methods.
- Comparison of noise sensitivity of regularized sampling, holographic backpropagation tomography, and diffraction tomography.
- Holographic backpropagation tomography is the least sensitive to noise and provides better reconstruction image quality under noisy conditions.

1.2 Publications

Parts of this thesis have been presented and published in a conference and also have been submitted to a peer-reviewed journal. The publications and submissions associated with this work are

- T. Sorensen and K. F. Warnick,"Inverse scattering image quality as a function of SNR," *Proceedings of IEEE Antennas and Propagation Society International Symposium*, Honolulu, HI, June 10-15, 2007.

- T. Sorensen and K.F. Warnick, "Image Reconstruction Quality for Regularized Sampling and Linearized Tomography with Noisy Data," *Inverse Problems*, Submitted May 2008

Chapter 2

Background

This chapter provides background information about the specific inverse scattering problem under analysis. First, the inverse scattering problem is set up. The inverse scattering methods under investigation are derived or outlined next. The contrast ratio used for image quality comparisons is defined. A noise model for the system is described. Lastly, analytical eigenvalues for the scattering amplitudes of a PEC circular cylinder, which prove crucial in determining contrast ratio estimates, are found.

In this thesis, two-dimensional TM-polarized wave scattering is considered. For 2D inverse scattering the scatterer is assumed to be infinitely long and uniform in one direction. The scatterer is probed by incident waves and scattered fields from the object are measured in the far field at various observation points. Multiple transmitters and receivers are placed equidistant in a circle around the region which is of interest. The different methods under consideration use the scattered fields measured at the receivers to reconstruct an approximation to the scatterer. The bistatic scattering amplitudes are denoted as $u_{\mathrm{s}}(\theta, \phi)$, where ϕ is the observation angle and θ is the incident wave angle. The incident field wavenumber is k_0. Incident and scattering unit vectors are defined as $\hat{\boldsymbol{\rho}}(\theta)$, $\hat{\boldsymbol{\rho}}(\phi)$ respectively. Position vectors take the form $\boldsymbol{r} = \hat{\boldsymbol{x}}x + \hat{\boldsymbol{y}}y = \hat{\boldsymbol{\rho}}\rho + \hat{\boldsymbol{\phi}}\phi_r$.

In Figure 2.1 a 3D image of the 2D inverse scattering problem is shown. The scatterer is an infinite cylinder in the z direction. The scatterer cross section in the x,y plane is S. The antennas are placed equidistant on D which is in the far field. Figure 2.2 shows the 2D scattering setup with illustrations of the different vectors.

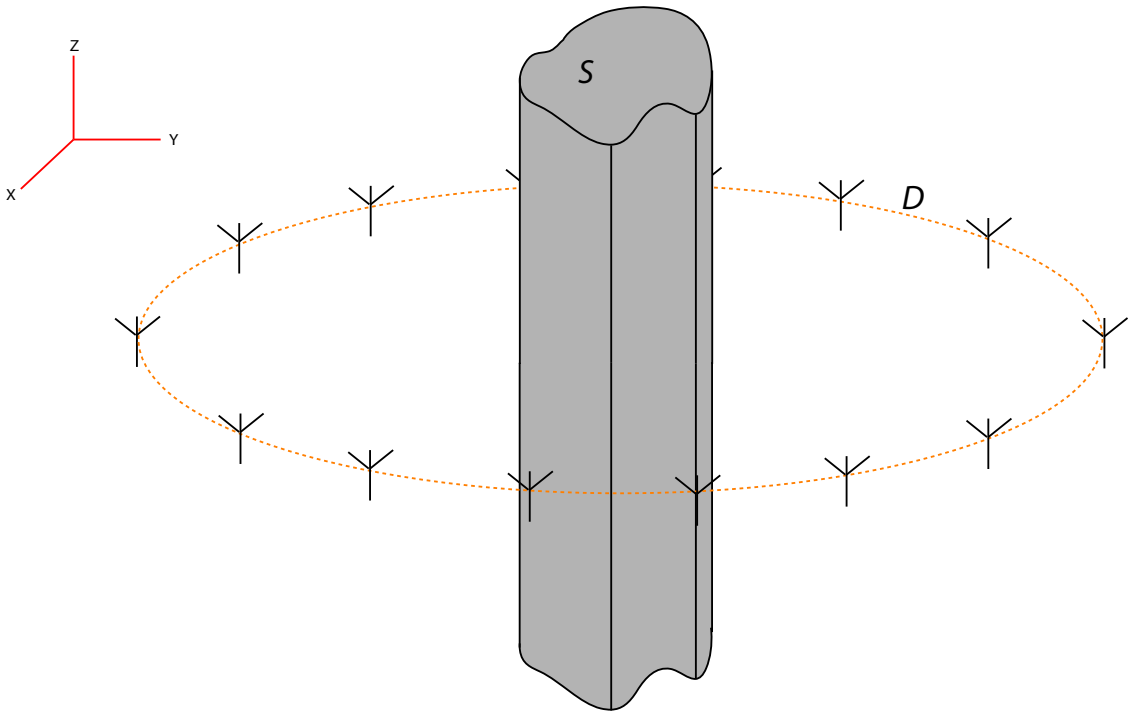


Figure 2.1: Setup for inverse scattering where the scatterer is an infinite cylinder in the z direction.

Obviously there are a wide range of inverse scattering problem scenarios but this case has been used for analytical and numerical studies because of ease and generality.

2.1 Diffraction Tomography

Diffraction tomography was first presented in [11] for acoustic waves. Diffraction tomography has been widely studied and used in practice. Thus it has been utilized in many different scenarios and assumes different forms and variations. The derivation presented here is for far field inverse scattering and follows that of [12].

A two-dimensional scatterer in free space illuminated by an incident scalar field, $E^i(\boldsymbol{r}_t)$, leads to the scattered fields

$$E^s(\boldsymbol{r}) = k_0^2 \int_V \mathrm{d}\boldsymbol{r}' g(\boldsymbol{r}, \boldsymbol{r}') O(\boldsymbol{r}') E(\boldsymbol{r}') \tag{2.1}$$

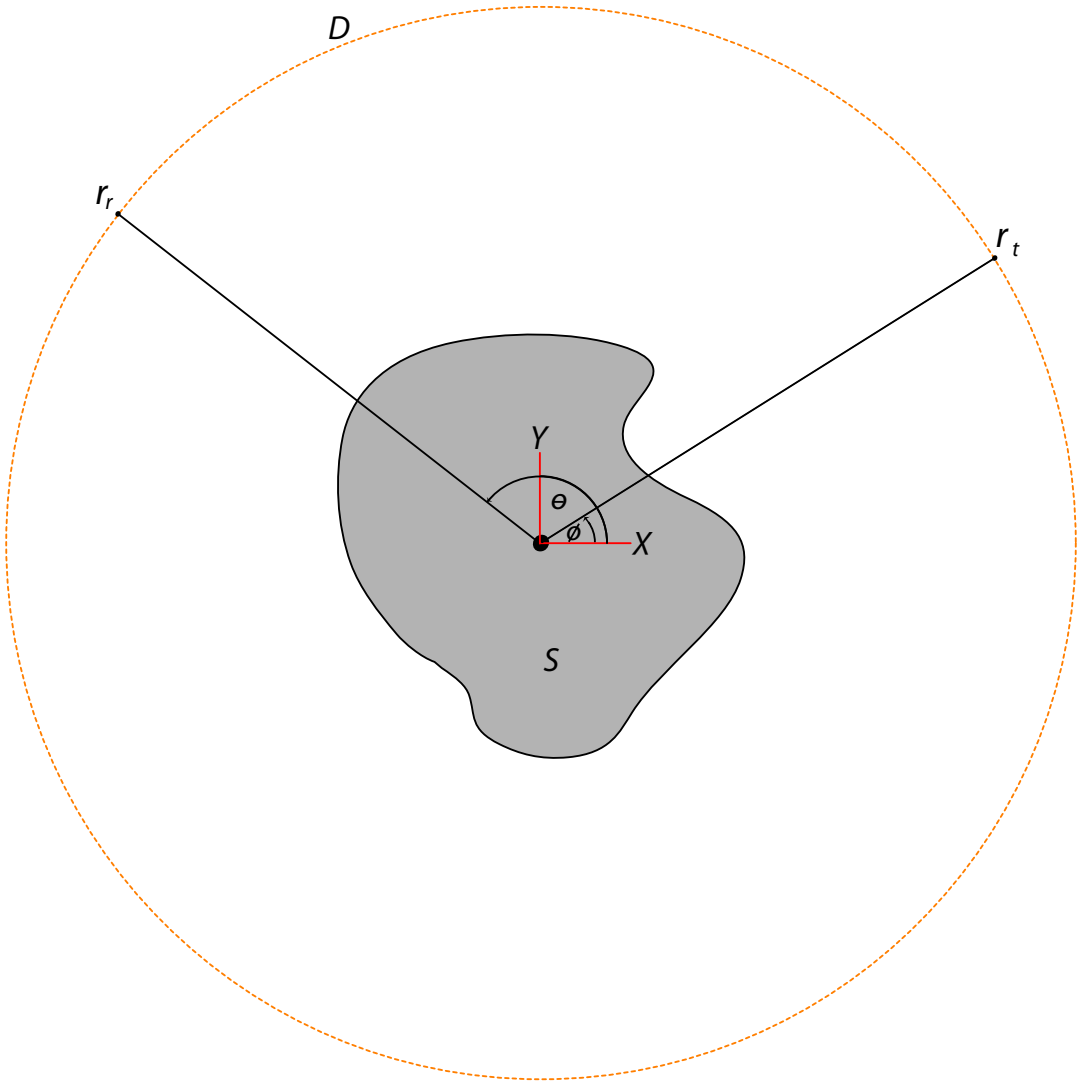


Figure 2.2: A 2D inverse scattering problem with illustration of vectors and angles included.

where the total field $E(\boldsymbol{r}) = E^s(\boldsymbol{r}) + E^i(\boldsymbol{r})$ and $O(\boldsymbol{r})$ is the permittivity profile of the scatterer. With a weakly scattering assumption about the scatterer, the Born approximation $E(\boldsymbol{r'}) = E^i(\boldsymbol{r'})$ linearizes the problem. The 2D Green's function for receive point $\boldsymbol{r_r}$ and integration point $\boldsymbol{r'}$ is

$$g(\boldsymbol{r_r}, \boldsymbol{r'}) = \frac{i}{4} H_0^{(1)}(k_0 |\boldsymbol{r_r} - \boldsymbol{r'}|) \tag{2.2}$$

and the electric field incident from a line source (in 3D) at $\boldsymbol{r_t}$ is

$$E^i(\boldsymbol{r}) = \frac{i}{4} H_0^{(1)}(k_0|\boldsymbol{r} - \boldsymbol{r_t}|). \tag{2.3}$$

In the far field these functions become

$$g(\boldsymbol{r_r}, \boldsymbol{r'}) \sim \frac{i}{4}\sqrt{\frac{2}{\pi k_0 r_r}} e^{ik_0 r_r - ik_0 \hat{\boldsymbol{\rho}}(\phi)\cdot\boldsymbol{r'} - \pi/4} \tag{2.4}$$

and

$$E^i(\boldsymbol{r}) \sim \frac{i}{4}\sqrt{\frac{2}{\pi k_0 r_t}} e^{ik_0 r_t - ik_0 \hat{\boldsymbol{\rho}}(\theta)\cdot\boldsymbol{r'} - \pi/4}. \tag{2.5}$$

The quantities r_r and r_t are the distances from the origin to the transmitters and receivers. Using these three approximations (2.1) is

$$E^s(\boldsymbol{r_r}) \sim \frac{i}{8\pi k_0 \sqrt{r_r r_t}} e^{ik_0(r_r + r_t)} \int d\boldsymbol{r'} e^{-i(\boldsymbol{k^s} - \boldsymbol{k^i})\cdot\boldsymbol{r'}} O(\boldsymbol{r'}) \tag{2.6}$$

where $\boldsymbol{k^i} = -k_0\hat{\boldsymbol{\rho}}(\theta)$ and $\boldsymbol{k^s} = k_0\hat{\boldsymbol{\rho}}(\phi)$. This describes a Fourier transform relation between the scattered fields and the object function.

Since measurements of the scattering amplitude are obtained in simulation and experiment it is beneficial to use the relationship

$$u_s(\phi, \theta) = \lim_{r\to\infty} \sqrt{\frac{\pi k_0 r}{2i}} e^{ik_0 r} E^s(\boldsymbol{r_r}, \phi). \tag{2.7}$$

The scattering amplitudes are useful because they are independent of distance from the scatterer and bulk far field phase.

After substitution of the scattering amplitudes, the Fourier relationship with the object function is

$$u_{\mathrm{s}}(\theta, \phi) \sim \int \mathrm{d}\boldsymbol{r'} \mathrm{e}^{\mathrm{i}(\boldsymbol{k}^{\mathrm{s}} - \boldsymbol{k}^{\mathrm{i}})\cdot\boldsymbol{r'}} \mathrm{O}(\boldsymbol{r'}). \tag{2.8}$$

For far field diffraction tomography, far field approximations for the Green's function and incident fields and the Born approximation lead to a simple relationship between the object function and the scattering amplitudes. In the Fourier domain the data is known only on circular arcs. To obtain the object function it is necessary to interpolate the data in the Fourier domain to a rectangular reconstruction grid and take the inverse Fourier transform.

This method relies on the linear Born approximation but otherwise makes no assumptions about scatterer properties, boundary conditions, or shape. It is fast but limited by the finite span of $\boldsymbol{k}^{\mathrm{s}} - \boldsymbol{k}^{\mathrm{i}}$ in the Fourier domain and consequently only a low-pass band-limited version of the object function is retrieved. This form of diffraction tomography is considered a direct Fourier version.

2.2 Holographic Backpropagation Tomography

Holographic backpropagation tomography is similar to the filtered backpropagation method of diffraction tomography [11]. It extends filtered backpropagation diffraction tomography by using holographic principles.

Holographic principles first applied to imaging and inverse scattering can be found in [13, 14, 15, 16, 17]. Holographic principles were first applied to electromagnetic inverse scattering in [18, 19, 20]. The specific method of holographic backpropagation tomography was derived in [9]. Below I do not reproduce the derivation completely because of its complexity but provide a general overview of the principles that allow this method to work for inverse scattering. The treatment follows that of [9].

A holographic field is defined as

$$\Theta_H(\boldsymbol{r}) = -\int_{\partial D}\left[E_s(\boldsymbol{r}',\phi)\frac{\partial g^*(\boldsymbol{r}-\boldsymbol{r}')}{\partial n'} - g^*(\boldsymbol{r}-\boldsymbol{r}')\frac{\partial E_s(\boldsymbol{r}',\phi)}{\partial n'}\right] ds(\boldsymbol{r}'). \tag{2.9}$$

The holographic surface is the defined as region D so that it corresponds to the inverse scattering problem. In theory the surface is defined as needed. The notation $ds(\boldsymbol{r}')$ refers to the fact that we are integrating around the boundary points of the region

D. Using Green's theorem and the equivalent current source

$$-ik\eta J_c(\boldsymbol{r},\phi) = k^2 O(\boldsymbol{r}) E(\boldsymbol{r},\phi) \tag{2.10}$$

where J_c is the equivalent current induced by the electric field, the holographic field is

$$\Theta_H(\boldsymbol{r}) = 2k\eta \int_D J_c(\boldsymbol{r},\phi) g_i(\boldsymbol{r}-\boldsymbol{r}')d\boldsymbol{r}'. \tag{2.11}$$

Also in the far field using the scattering amplitude and a phase term the holographic field is

$$\Theta_H(\boldsymbol{r}) = \sqrt{\frac{k}{i2\pi}} \int_\Omega u_s(\phi,\theta) e^{-ik\boldsymbol{r}\cdot\hat{\boldsymbol{\rho}}(\theta)} d\phi d\theta. \tag{2.12}$$

Equations (2.9), (2.11), (2.12) equivalently express the holographic fields. In order to image the scatterer the object function, $O(\boldsymbol{r})$, is sought. By substituting (2.10) into (2.11), approximating the total field with the Born approximation, and taking the Fourier transform it is possible to solve for the object function in k space. With the object function defined in k space, a Fourier transform back into the reconstruction domain completes a solution for the object function. The details of this process are found in [9]. The object function at a reconstruction point $\boldsymbol{r}$ is

$$O(\boldsymbol{r}) = \frac{1}{4\sqrt{\pi^3}} \Re\left\{ \sqrt{|k_0|} \int_0^{2\pi} \int_0^{2\pi} u_{\mathrm{s}}(\theta,\phi) \mathrm{e}^{-\mathrm{i}(\boldsymbol{k}^{\mathrm{s}}-\boldsymbol{k}^{\mathrm{i}})\cdot\boldsymbol{r}} |\sin(\phi-\theta)| \mathrm{d}\phi \mathrm{d}\theta \right\}. \tag{2.13}$$

The holographic backpropagation tomography method was derived for the general multiple frequency case. Only the single frequency method is shown here. This is similar in form to filtered backpropagation diffraction tomography but the underlying physical interpretation differs.

The principles of holography are most widely known because of optical holograms. Optical holography provides a good insight into how this method functions. Optical holography works by illuminating an object with two different sources of light.

The scattered light is recorded on a holographic film. The scattering from the two sources interfere at the film and a distorted image is recorded. This holographic film can be used to reproduce an image of the scatterer. When the film is illuminated by a source in the same location as an original source then an image of the scatterer appears as if it was being illuminated from the other source. This is a simplification, but it provides insight into holography and imaging methods.

The process of holographic backpropagation tomography is to first illuminate the scatterer from various angles and measure the fields on a holographic surface, in this case ∂D, and then backpropagate the fields to a reconstruction point. Looking at equation (2.13) the holographic method is intuitively explained. There are really three important mathematical constructs; two integrations and a filter. The θ integration is illuminating the holographic surface from the reconstruction point without the scatterer present. The ϕ integration then backpropagates the scattering amplitudes on the holographic surface, due to the incident waves, back to the reconstruction point. The illumination of the holographic surface with a plane wave without the presence of the scatterer is in effect removing the incident plane wave of the total field that caused the scattering. The scattering amplitudes are alone backpropagated to the reconstruction point. The filter, $|\sin(\phi - \theta)|$, is due to the limited sampling in k space. The fields are only known on circular arcs in k space. The effect is that the reconstructed image is band-limited. This filter is the manifestation of the limited sampling. By illumination and backpropagation an "image" or reconstruction of the object function is obtained at the reconstruction point. If there were no scatterer at a reconstruction point then the value of the object function will be zero or very small. When the reconstruction point is located in the support of the scatterer then a band-limited version of the object function is produced. By plotting the object function an image of the scatterer is produced. The object function also gives information about the physical characteristics of the scatterer, mainly the scatterer permittivity profile.

2.3 Regularized Sampling

Regularized sampling was originated by Colton and Kirsch [1]. Since its introduction, various extensions to the method have been derived. It has been shown to work for 3D as well as 2D scattering for acoustic and electromagnetic waves and under a variety of conditions [21, 22]. It also has been referred to by several names such as the simple method and linear sampling method. The scholarly research and publications on regularized sampling are extensive. Colton *et al.* provides a very good overview of regularized sampling and a summary of previous and current research [21]. A derivation of the regularized sampling method is given in this thesis.

2.3.1 Regularized Sampling Method

Consider the linear integral equation

$$\int_{-\pi}^{\pi} u_{\mathrm{s}}(\theta,\phi)g(\theta,\boldsymbol{r})\mathrm{d}\theta = \mathrm{e}^{\mathrm{i}\boldsymbol{k}^{\mathrm{s}}\cdot\boldsymbol{r}} \qquad \phi\in[-\pi,\pi] \tag{2.14}$$

where $\boldsymbol{k}^{\mathrm{s}} = k_0\hat{\boldsymbol{\rho}}(\phi)$. The right hand side is the far field of a cylindrical wave emanating from a point source at $\boldsymbol{r}$. The reconstruction pixel point is $\boldsymbol{r}$. The unknown function g provides information about the scatterer. The regularized sampling method relies on the fact that as $\boldsymbol{r}$ approaches the boundary of the scatterer from the inside, the solution $g(\theta,\boldsymbol{r})$, has a norm that tends to infinity. Outside the region of support of the scatterer, the norm of the solution is infinite [23, 24].

I follow an alternate derivation and a theorem from [23] in place of the original found in [1]. Consider the linear integral equation

$$L_D J = E_D \tag{2.15}$$

with D being a circle enclosing a scatterer with support S and

$$L_D J = \frac{k\mu}{4}\int_{\partial S} H_0^{(1)}(k|\boldsymbol{r}_{\boldsymbol{r}}-\boldsymbol{r}'|)J(\boldsymbol{r}')d\boldsymbol{r}' \qquad \boldsymbol{r}_{\boldsymbol{r}}\in D \tag{2.16}$$

where ∂S is the scatterer surface. Constrain E_D to be the electric field radiated by a point source at $\boldsymbol{r}_0$. The electric field is then cylindrically radiating as

$$E_D(\boldsymbol{r}) = H_0^{(1)}(k|\boldsymbol{r}_{\boldsymbol{r}} - \boldsymbol{r}|). \tag{2.17}$$

These equations begin the formulation of the derivation of regularized sampling.

The underlying theory of regularized sampling is based on Huygen's principle. Huygen's principle states that the fields outside a volume can be determined uniquely by the tangential fields on the surface of the volume. For example, if a volume encloses a source, in order to find the field outside of the volume it is only necessary to know the tangential fields on the surface of the volume and not the actual source. This is true because a finite current exists on the scatterer which radiates the exact same fields as the source.

For the regularized sampling problem, we apply Huygen's principle. There are three locations of the point source to consider: $\boldsymbol{r}$ inside the scatterer, $\boldsymbol{r}$ outside of the scatterer, and $\boldsymbol{r}$ on the surface. If $\boldsymbol{r}$ is inside of the scatterer, S, then there exists a finite current J on ∂S which radiates the same fields as E_D. Huygen's principle is satisfied. Outside the scatterer the opposite is true. The theorem in [23] states that for a circle D enclosing a current distribution J on the scatterer $\overline{S}$ there is no finite current J on $\overline{S}$ that radiates a field at D equal to a point source outside of ∂S. So if $\boldsymbol{r}$ is outside of $\overline{S}$ Huygen's principle is violated. On the surface, Huygen's principle is also violated and the surface current is singular. So, we have the final results concerning the current,

$$\|J\| < \infty, \qquad \boldsymbol{r} \in S, \tag{2.18}$$

$$\|J\| \to \infty, \qquad \boldsymbol{r} \notin S, \tag{2.19}$$

$$\|J\| \to \infty, \qquad \boldsymbol{r} \in \partial S. \tag{2.20}$$

We now specialize the formulation to the case where D is in the far field. The far field limit of the Hankel function is

$$H_0^{(1)}(x) \sim \sqrt{\frac{-2i}{\pi x}} e^{ix}, \quad x \to \infty. \tag{2.21}$$

With this approximation (2.15) is

$$\frac{k\eta}{4} \int_{\partial S} e^{-ik\hat{\boldsymbol{\rho}}(\phi)\cdot \boldsymbol{r'}} J(\boldsymbol{r'}) d\boldsymbol{r'} = e^{-ik\hat{\boldsymbol{\rho}}(\phi)\cdot \boldsymbol{r}}. \tag{2.22}$$

Let E^i be an incident field that produces a current J on the scatterer surface. The incident field is determined by

$$E^i = LJ = \frac{k\eta}{4} \int_{\partial S} H_0^{(1)}(k|\boldsymbol{r} - \boldsymbol{r'}) J(\boldsymbol{r'}) d\boldsymbol{r'}, \quad \boldsymbol{r} \in \partial S. \tag{2.23}$$

The current is then

$$J = L^{-1} E^i. \tag{2.24}$$

It is also beneficial to expand the incident field into a plane wave spectrum defined as

$$E^i(\boldsymbol{r}) = \int_0^{2\pi} g(\theta) e^{i\boldsymbol{k}^i \cdot \boldsymbol{r_t}} d\theta \tag{2.25}$$

where $\boldsymbol{k}^i$ is defined as before. The induced current due to an incident field is

$$J = L^{-1} \int_0^{2\pi} g(\theta) e^{i\boldsymbol{k}^i \cdot \boldsymbol{r_t}} d\theta. \tag{2.26}$$

The original EFIE with this induced current is

$$\frac{k\eta}{4} \int_{\partial S} e^{-ik\hat{\boldsymbol{\rho}}(\phi)\cdot \boldsymbol{r'}} L^{-1} \int_0^{2\pi} g(\theta) e^{i\boldsymbol{k}^i \cdot \boldsymbol{r_t}} d\theta d\boldsymbol{r'} = e^{-ik\hat{\boldsymbol{\rho}}(\phi)\cdot \boldsymbol{r}}. \tag{2.27}$$

Rearranging,

$$\int_0^{2\pi} \frac{k\eta}{4} \int_{\partial S} d\boldsymbol{r}' e^{-i\boldsymbol{k}^s \cdot \boldsymbol{r}'} L^{-1} e^{i\boldsymbol{k}^i \cdot \boldsymbol{r_t}} g(\theta) d\theta = e^{-ik\hat{\boldsymbol{\rho}}(\phi)\cdot \boldsymbol{r}}. \tag{2.28}$$

The scattering amplitude in the scattering direction ϕ due to an incident wave at θ is

$$F(\phi, \theta) = \frac{k\eta}{4} \int_{\partial S} d\boldsymbol{r}' e^{-i\boldsymbol{k}^s \cdot \boldsymbol{r}'} J(\boldsymbol{r}') \tag{2.29}$$

$$= \frac{k\eta}{4} \int_{\partial S} d\boldsymbol{r}' e^{-i\boldsymbol{k}^s \cdot \boldsymbol{r}'} L^{-1} e^{i\boldsymbol{k}^i \cdot \boldsymbol{r}}. \tag{2.30}$$

With the substitution of (2.29) into (2.28) we have

$$\int_0^{2\pi} F(\phi, \theta) g(\theta) d\theta = e^{-i\boldsymbol{k}^s \cdot \boldsymbol{r}} \tag{2.31}$$

which is the regularized sampling linear integral equation. Since the function $g(\theta)$ is an angular distribution which is proportional to the current we have

$$\|g\| < \infty, \qquad \boldsymbol{r} \in S, \tag{2.32}$$

$$\|g\| \to \infty, \qquad \boldsymbol{r} \notin S, \tag{2.33}$$

$$\|g\| \to \infty, \qquad \boldsymbol{r} \in \partial S. \tag{2.34}$$

The scatterer support, $D(\boldsymbol{r})$, is found by computing

$$D(\boldsymbol{r}) = \frac{1}{\|g\|} \tag{2.35}$$

and a reconstructed image is obtained by plotting the support.

This derivation motivates physical intuition into the regularized sampling method. If $\boldsymbol{r}$ is outside the scatterer or on the surface of the scatterer no fields exist that will illuminate the object so that the scattered fields appear to be emanating from a point source at $\boldsymbol{r}$. The function g can be considered a weighting function whereby the incident fields are linearly combined in such a way as to produce scattered fields that emanate from a point source at the reconstruction position

vector. If the reconstruction point is outside of the scatterer no linear combination of the scattered fields produces a point source and the norm of the solution to (2.31) is infinite since the RHS is not in the range of the integral operator. On the other hand, for a point inside the scatterer a weighting function exists and its norm is finite. According to Huygen's principle, which states that the fields inside a volume can be uniquely characterized by the tangential fields on the surface of that volume, when the reconstruction point is inside the scatterer support, Huygen's principle is valid but otherwise is invalid.

2.4 Contrast Ratio

All methods under consideration except regularized sampling return the permittivity profile of the scatterer. Regularized sampling is inherently different in that it does not return any scatterer characteristics other than the support. With this in consideration an image quality measure that compares the inverse scattering reconstruction accuracy must determine how scatterer shape is resolved. This type of measure is accomplished using a contrast ratio defined as the ratio of the norm of the pixel values inside the scatterer to the norm of the pixel values outside the scatterer. The pixel values are the object function $O(\boldsymbol{r})$ or the scatterer support $D(\boldsymbol{r})$ depending on the method.

Defining the contrast ratio in this manner provides a measure of scatterer shape resolution. When the scatterer shape is well defined the contrast ratio is large but when the reconstructed scatterer shape becomes corrupted the contrast ratio is small. As the SNR varies the reconstructed images are perturbed and scatterer shape accuracy changes. The reconstructed image quality behavior can be observed and quantified by calculating the contrast ratio at different noise levels in the forward data.

In the noise-free case, the contrast ratio does not provide a measure for comparison of image reconstruction accuracy between inverse scattering methods because they reconstruct different properties of the scatterer. The important information that

can be taken from this type of measure is the image reconstruction quality behavior as varying levels of noise are introduced into scattering data.

2.5 Noise Model

In actual experimental data noise due to amplifiers, background, interference, and thermal radiation reduce the accuracy of scattered field measurements. We model noise as a stochastic process. Noise is added to the far-field scattering amplitudes. The receivers are assumed to be spaced far enough apart that the noise is uncorrelated between receivers. A zero mean circular Gaussian random variable is added to each of the scattering amplitudes to model system noise at each receiver.

The noise matrix adds iid complex circular Gaussian noise to each element in a scattering matrix. The level of noise is determined by a desired SNR. For a given scattering amplitude matrix S of size $N \times N$ the signal power is calculated as

$$V_{rms} = \frac{\|S\|_F}{N}, \tag{2.36}$$

$$P_{sig} = \frac{\|S\|_F^2}{N^2}. \tag{2.37}$$

For a given scatterer, the signal power per channel will not change when the number of transceivers changes and matrix size changes, because of the normalization factor.

The per channel noise signal power σ^2 is calculated as

$$\sigma^2 = \frac{P_{sig}}{\mathrm{SNR}}. \tag{2.38}$$

The variance of the real random variable and the imaginary random variable is then

$$\sigma_r^2 = \sigma_i^2 = \frac{P_{sig}}{2\mathrm{SNR}}. \tag{2.39}$$

Using this variance a matrix of noise can be created which is then directly added to the matrix of forward scattering amplitudes. This results in the noise power at each

receiver being the same, with the noise power proportional to the average scattering amplitude. The noise matrix has elements given by

$$\eta_{mn} = [\nu_{\mathrm{r}} + \mathrm{i}\nu_{\mathrm{i}}] \tag{2.40}$$

where ν_{r} and ν_{i} are zero mean Gaussian random variables with variances of σ_r^2 and σ_i^2.

2.6 Sampled Far Field Data and Eigenvalues of F

The scattering amplitude is the known data to which computations are made in order to reconstruct the scatterer. These scattering amplitudes in future sections are placed in a matrix referred to as $\boldsymbol{F}$. Knowing the spectrum of the scattering amplitude matrix simplifies solutions to the analytical estimates of the contrast ratio.

The eigenvalues for a matrix of scattering amplitudes from a PEC circular cylinder with TM incident waves were found in [25] and the derivation below closely resembles it. Begin with the scattering amplitudes.

$$\boldsymbol{F}(\phi^i, \phi^s) = \sum_l (-1)^{-l} \frac{J_l(ka)}{H_l^2(ka)} e^{il(\phi^s - \phi^i)} \tag{2.41}$$

$$= \sum_l b_l e^{il(\phi^s - \phi^i)}. \tag{2.42}$$

Let $\phi_n = (n - 1/2)\Delta\phi,\ n = 1, \cdots, N$ and $\phi_m = (m - 1/2)\Delta\phi,\ m = 1, \cdots, N$ and the sampled the far field data is

$$F_{mn} = \sum_l b_l e^{il(\phi_m - \phi_n)}. \tag{2.43}$$

Assume that the eigenvectors of $\boldsymbol{F}$ are of the form $e^{iq\phi_n}$. Then

$$\boldsymbol{F}[e^{iq\phi_n}] = \sum_{n=1}^{N} F_{mn} e^{iq\phi_n} \tag{2.44}$$

$$= \sum_{n} \sum_{l} b_l e^{il(\phi_m - \phi_n)} e^{iq\phi_n} \tag{2.45}$$

$$= \sum_{l} b_l e^{il\phi_m} \sum_{n} e^{i(q-l)\phi_n}. \tag{2.46}$$

The second summation is a sinc function,

$$\sum_{n} e^{i(q-l)\phi_n} = (-1)^{q-l} \frac{\sin(\pi(q-l))}{\sin(\pi(q-l)/N)} = \begin{cases} (-1)^s N & q = sN \\ 0 & \text{otherwise} \end{cases} \tag{2.47}$$

where s is an integer. Let $q - l = sN$ and $l = q - sN$. Equation (2.46) now becomes

$$N \sum_{s} b_{q+sN} (-1)^s e^{i(q-sN)\phi_m}. \tag{2.48}$$

The exponentials simplify to

$$e^{-isN\phi_m} = e^{-is(n-1/2)2\pi} = e^{-is\pi} e^{-is2\pi n} \tag{2.49}$$

$$= (-1)^s \tag{2.50}$$

and (2.48) becomes

$$= N \sum_{s} b_{q+sN} e^{iq\phi_m} = N \sum_{s} b_{q+sN} \boldsymbol{u}_q. \tag{2.51}$$

The eigenvalues are

$$\lambda_q = N \sum_{s} b_{q+sN}. \tag{2.52}$$

Knowing the analytical eigenvalues for the scattering amplitude matrix allows us to estimate contrast ratios.

Chapter 3

Contrast Ratio vs SNR

In this chapter, analytical contrast ratio estimates for regularized sampling and holographic backpropagation tomography are derived in order to analyze image quality. This is a deviation from previous work where noise sensitivity of several inverse scattering methods was quantified using pixel variances and covariances and not measured as a function of SNR. The contrast ratio provides a good technique for analytical and numerical analysis of changing image quality with varying SNR. The contrast ratio estimates and simulations provide information about the noise characteristics of the several methods under investigation.

As already stated, the noise characteristics of diffraction tomography, filtered backpropagation diffraction tomography and non-linear inverse scattering have been studied previously. These studies measured noise characteristics of reconstructed images by deriving and simulating image pixel variances and covariances. Little has been shown as to how these statistics change and how image quality differs with varying SNR.

The approach used in deriving image quality metrics for inverse scattering is an extension of accuracy estimate derivations for numerical algorithms that compute forward scattered fields [25, 26]. Namely, analytical estimates are formed using the PEC circular cylinder as a canonical scatterer.

As background, the following expansions are used throughout the derivation of the contrast ratio estimates:

$$u_s(\phi,\theta) = -\sum_{r=-\infty}^{\infty} \frac{(-1)^r J_r(ka)}{H_r^{(2)}(ka)} e^{ir(\phi-\theta)} = \sum_{r=-\infty}^{\infty} b_r e^{ir(\phi-\theta)}, \tag{3.1}$$

$$e^{i\boldsymbol{k}^{\mathrm{s}}\cdot\boldsymbol{r}} = \sum_{r=-\infty}^{\infty} i^{-r} J_p(k\rho) e^{ir\phi} = \sum_{r=-\infty}^{\infty} d_r e^{ir\phi}, \tag{3.2}$$

$$g(\theta;\boldsymbol{r}) = \sum_{p=-\infty}^{\infty} a_p e^{ip\theta}, \tag{3.3}$$

where b_r and d_r are coefficients containing Bessel functions and a_p is unknown.

3.1 Discrete Regularized Sampling for a PEC Cylinder Noise Free Case

The regularized sampling equation is

$$\int_{-\pi}^{\pi} u_{\mathrm{s}}(\theta,\phi) g(\theta,\boldsymbol{r}) \mathrm{d}\theta = \mathrm{e}^{\mathrm{i}\boldsymbol{k}^{\mathrm{s}}\cdot\boldsymbol{r}} \qquad \phi \in [-\pi,\pi]. \tag{3.4}$$

Substituting the expansions (3.1)-(3.3) leads to

$$\int_{-\pi}^{\pi} \sum_{r=-\infty}^{\infty} b_r e^{ir(\phi-\theta)} \sum_{p=-\infty}^{\infty} a_p e^{ip\theta} d\theta = \sum_{r=-\infty}^{\infty} d_r e^{ir\phi}. \tag{3.5}$$

The equation can be discretized in scattering and incident angle. This is done by the midpoint rule letting $\phi_m = (m-\frac{1}{2})\Delta\phi$ and $\phi_n = (n-\frac{1}{2})\Delta\theta$. In discrete form (3.4) becomes

$$\sum_{n=1}^{N} u_s(\phi_m,\theta_n) g(\theta_n) = e^{-ik\hat{\boldsymbol{\rho}}(\phi_m)\cdot\boldsymbol{r}} \tag{3.6}$$

where $m = 1,\cdots,M$ and $n = 1,\cdots,N$. An equal discretization size is used for both angles so $\Delta\phi = \Delta\theta$ and $N = M$. Now the complete discretized equation with series

expansions is

$$\sum_{n=1}^{N}\sum_{r=-\infty}^{\infty} b_r e^{ir(\phi_m-\theta_n)} \sum_{p=-\infty}^{\infty} a_p e^{ip\theta_n} = \sum_{r=-\infty}^{\infty} d_r e^{ir\phi_m}. \tag{3.7}$$

Instead of using summation notation equation (3.7) can be written as a matrix equation

$$\boldsymbol{F}\boldsymbol{g} = \boldsymbol{d} \tag{3.8}$$

where

$$F_{mn} = \sum_{r=-\infty}^{\infty} b_r e^{ir(\phi_m-\theta_n)}, \tag{3.9}$$

$$d_m = \sum_{r=-\infty}^{\infty} d_r e^{ir\phi_m}, \tag{3.10}$$

$$g_n = \sum_{p=-\infty}^{\infty} a_p e^{ip\theta_n}. \tag{3.11}$$

Knowing the eigenvalues of $\boldsymbol{F}$ aids in finding a solution to g. Using the vectors $\boldsymbol{\phi} = [\phi_1 \cdots \phi_N] = \boldsymbol{\theta} = [\theta_1 \cdots \theta_M]$ which are equivalent because of the equal discretation size , the eigenvectors are assumed to be of the form $e^{ip\boldsymbol{\phi}}$ and therefore normal to each other. This holds true when using transceivers or colocated receivers and transmitters, which is the setup used in this thesis. Using the eigenvalues of Section 2.6, and defining the eigenvectors

$$\boldsymbol{v}_p = e^{ip\boldsymbol{\theta}} = \boldsymbol{u}_p = e^{ip\boldsymbol{\phi}} \tag{3.12}$$

equation (3.8) becomes, with suppressed summations,

$$\boldsymbol{F} a_p \boldsymbol{u}_p = d_r \boldsymbol{u}_r \tag{3.13}$$

$$a_p \lambda_p \boldsymbol{u}_p = d_r \boldsymbol{u}_r. \tag{3.14}$$

The linear system of equations (3.8) and (3.13) is ill-posed, because $\boldsymbol{F}$ is ill-conditioned due to its small or nonzeros eigenvalues, so it is necessary to regularize the scattering amplitude matrix. There are various methods to do this. A very simple method is to add a small value to the diagonal of $\boldsymbol{F}$ which in turn eliminates the nonzero or very small eigenvalues by adding a constant ϵ to every eigenvalue. This can be done by adding $\epsilon \boldsymbol{I}$ to $\boldsymbol{F}$ where ϵ is the regularization parameter. Although effective, simple regularization usually requires that ϵ be determined by inspection of the eigenvalues or reconstructed image in order obtain the least error.

The unknown function $\boldsymbol{g}$ is known when the coefficients are determined. The coefficients on both sides of (3.14) must be equal and the unknown coefficients are easily solved as

$$a_p = \frac{d_p}{\lambda_p + \epsilon} \tag{3.15}$$

and

$$\boldsymbol{g} = \sum_p \frac{d_p}{\lambda_p + \epsilon} \boldsymbol{u}_p \tag{3.16}$$

where simple regularization is employed. The pixel value follows as

$$D(\rho) = \left(\sum_p \frac{|d_p(\rho)|^2}{|\lambda_p + \epsilon|^2} \right)^{-1}. \tag{3.17}$$

From this point it is straightforward to include noise in the forward data.

It is easy to see the need for regularization. As some of the eigenvalues are very small, $\boldsymbol{g}$, without the addition of ϵ to the eigenvalues, would be infinitely large and unstable. A regularization scheme must be added to stabilize the solution.

3.2 Discrete Regularized Sampling for a PEC Cylinder With Noise

The solution to the pixel value for the noise free case with simple regularization is straightforward. Now consider the case with noisy forward scattering data. The

regularized sampling matrix equation with noise is

$$(\boldsymbol{F} + \boldsymbol{\eta})\boldsymbol{g} = \boldsymbol{d} \tag{3.18}$$

where $\boldsymbol{\eta}$ is the noise matrix of section 2.5. By adding simple regularization (3.18) is now

$$(\boldsymbol{F} + \boldsymbol{\eta} + \epsilon\boldsymbol{I})\boldsymbol{g} = \boldsymbol{d}. \tag{3.19}$$

Expand $\boldsymbol{g}$ and $\boldsymbol{b}$ into its Fourier series representation and substitute it into the previous equation

$$\sum_p (\boldsymbol{F} + \boldsymbol{\eta} + \epsilon\boldsymbol{I})a_p\boldsymbol{v}_p = \sum_p d_p\boldsymbol{u}_p. \tag{3.20}$$

Using the eigenvectors (3.12) and the eigenvalues (2.52) leads to

$$\sum_p (\lambda_p\boldsymbol{u}_p + \boldsymbol{\eta}\boldsymbol{u}_p + \epsilon\boldsymbol{u}_p)a_p = \sum_p d_p\boldsymbol{u}_p. \tag{3.21}$$

The noise matrix can be simplified. We are interested in the perturbations of the reconstructed image caused by the noise. The effects of noise can be encapsulated by how the noise perturbs the eigenvalues of the scattering matrix. Therefore it is necessary to examine the eigenvalues of the noise matrix. Girko's circle theorem states that for a square matrix, with a zero mean and unit variance complex random variable for each entry, the eigenvalues when scaled by $1/\sqrt{2N}$ all lie in the unit circle in the complex plane. As the noise power changes the variance of the random variables in the matrix also changes. With variance of the noise defined as (2.38) the scaling factor is $1/\sqrt{2N\sigma^2}$. By scaling all the eigenvalues by this amount they lie in the unit circle. The scaling factor corresponds to the approximate magnitude of the largest eigenvalue which implies that the largest eigenvalue magnitude for a complex random matrix of zero mean and an arbitrary variance defined by the SNR

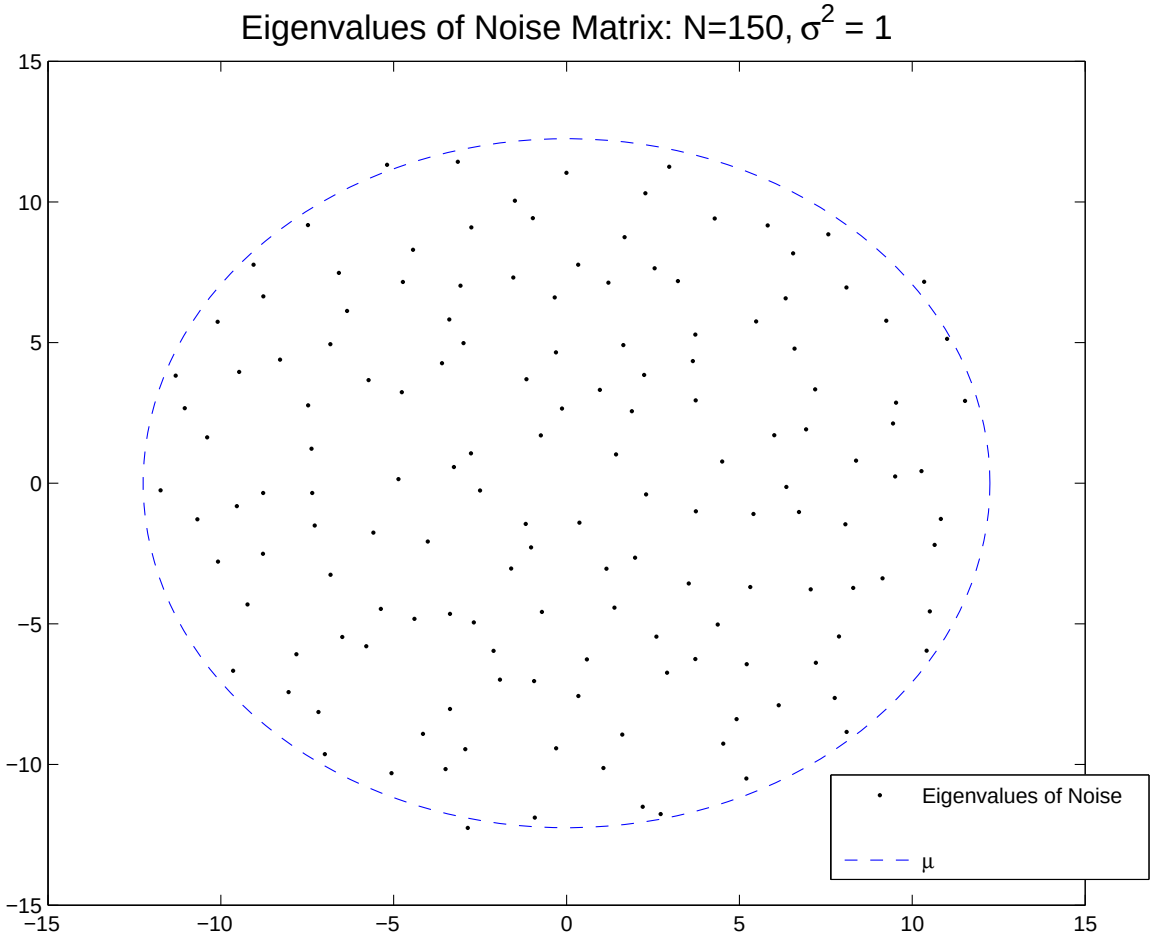


Figure 3.1: Eigenvalues of noise matrix of size N=150 and unit variance with the largest eigenvalue approximation

is approximately

$$\mu = \sqrt{\frac{2NP_{sig}}{SNR}} \tag{3.22}$$

where N is the number of transceivers. Figure 3.1 displays this approximation.

The noise matrix can be approximated using μ. Since the goal is to observe how noise perturbs or blurs a reconstructed image, focus is on the largest noise matrix eigenvalue. The largest eigenvalue has the greatest effect on the reconstructed image. By this argument, the noise matrix can be simply approximated by the scaled identity

matrix

$$\boldsymbol{\eta} \simeq \mu \boldsymbol{I}. \tag{3.23}$$

With this approximation the large perturbations in the reconstructed image, caused by the noise, are considered and the small effects neglected. The noise matrix approximation simplifies the system of equations to

$$\sum_p (\lambda_p \boldsymbol{u}_p + \mu \boldsymbol{u}_p + \epsilon \boldsymbol{u}_p) a_p = \sum_p d_p \boldsymbol{u}_p. \tag{3.24}$$

It may be noted that for low SNR the noise itself acts like a regularizing matrix. This will be shown in more detail later. Rearranging,

$$\sum_p (\lambda_p + \mu + \epsilon) \boldsymbol{u}_p a_p = \sum_p d_p \boldsymbol{u}_p. \tag{3.25}$$

The coefficients of both sides are equivalent which leads to

$$(\lambda_p + \mu + \epsilon) a_p = d_p, \tag{3.26}$$

$$a_p = \frac{d_p}{\lambda_p + \mu + \epsilon}. \tag{3.27}$$

The pixel value is determined by $\|\boldsymbol{g}\|$ given as

$$\boldsymbol{g} = \sum_p \frac{d_p}{\lambda_p + \mu + \epsilon} \boldsymbol{u}_p, \tag{3.28}$$

$$\|\boldsymbol{g}\|^2 = \sum_p N a_p^* a_p = N \boldsymbol{a}^* \boldsymbol{a} = N \|\boldsymbol{a}\|^2. \tag{3.29}$$

The contrast ratio is the norm of the pixel values inside the scatterer divided by the norm of the pixel values outside, where the pixel value is

$$O(\rho) = \frac{1}{\|\boldsymbol{g}(\rho)\|} \tag{3.30}$$

$$= \frac{1}{\sqrt{\sum_p N a_p(\rho)^* a_p(\rho)}} \tag{3.31}$$

$$= \frac{1}{\sqrt{N \sum_p \frac{|d_p(\rho)|^2}{|\lambda_p+\mu+\epsilon|^2}}}. \tag{3.32}$$

The norm of the pixel values inside reduces to

$$\|O(\rho)\|^2_{\rho \in scatterer} = 2\pi \int_0^a \frac{1}{N \sum_p \frac{|d_p(\rho)|^2}{|\lambda_p+\mu+\epsilon|^2}} \rho d\rho. \tag{3.33}$$

This is true because the problem is rotationally symmetric, and ρ is independent of θ. The norm of the pixel values outside the scatterer is

$$\|O(\rho)\|^2_{\rho \notin scatterer} = 2\pi \int_a^b \frac{1}{N \sum_p \frac{|d_p(\rho)|^2}{|\lambda_p+\mu+\epsilon|^2}} \rho d\rho \tag{3.34}$$

where a is the radius of the cylinder and b is the edge of the image. The contrast ratio becomes

$$CR = \frac{\int_0^a \frac{1}{\sum_p \frac{|d_p(\rho)|^2}{|\lambda_p+\mu+\epsilon|^2}} \rho d\rho}{\int_a^b \frac{1}{\sum_p \frac{|d_p(\rho)|^2}{|\lambda_p+\mu+\epsilon|^2}} \rho d\rho}. \tag{3.35}$$

No formal solution to this integral is obtainable. Thus this expression does not lend insight into the image reconstruction quality. An asymptotic approach to solving the contrast ratio leads to an approximate solution to this integration and is explored in the section below.

A more robust regularization scheme is that of Tikhonov regularization with Morozov's discrepancy rule. Tikhonov regularization provides the least squares estimate of the solution. Tikhonov regularization for a general system of equations

$\boldsymbol{Ax} = \boldsymbol{b}$ is

$$(\boldsymbol{A}^H\boldsymbol{A} + \epsilon\boldsymbol{I})\boldsymbol{x} = \boldsymbol{A}^H\boldsymbol{b} \tag{3.36}$$

with ϵ being the regularizing parameter. ϵ can be determined by various methods but Morozov's discrepancy gives a regularization parameter satisfying the minimun mean square error. This form of regularization was shown to be the most effective at reproducing accurate images with the least noise sensitivity. It is Tikhonov regularization in conjuction with Morozov's discrepancy rule that is used in numerical simulations found in the next chapter. The specifics for Morozov's discrepancy as used in regularized sampling is outlined in [2, 27].

Applying Tikhonov regularization to regularized sampling

$$[(\boldsymbol{F} + \boldsymbol{\eta})^H(\boldsymbol{F} + \boldsymbol{\eta}) + \epsilon\boldsymbol{I}]\boldsymbol{g} = (\boldsymbol{F} + \boldsymbol{\eta})^H\boldsymbol{b}. \tag{3.37}$$

The noise matrix once again can be approximated as it was in the previous section, and we have

$$[(\boldsymbol{F} + \mu\boldsymbol{I})^H(\boldsymbol{F} + \mu\boldsymbol{I}) + \epsilon\boldsymbol{I}]\boldsymbol{g} = (\boldsymbol{F} + \mu\boldsymbol{I})^H\boldsymbol{b} \tag{3.38}$$

where μ is as defined above. Simplifying,

$$(\boldsymbol{F}^H\boldsymbol{F} + \mu\boldsymbol{F}^H + \mu\boldsymbol{F} + (\mu^2 + \epsilon)\boldsymbol{I})\boldsymbol{g} = (\boldsymbol{F}^H + \mu\boldsymbol{I})\boldsymbol{b}, \tag{3.39}$$

$$(\boldsymbol{F}^H\boldsymbol{F}\boldsymbol{v}_p + \mu\boldsymbol{F}^H\boldsymbol{v}_p + \mu\boldsymbol{F}\boldsymbol{v}_p + (\mu^2 + \epsilon)\boldsymbol{v}_p)a_p = (\boldsymbol{F}^H + \mu)d_p\boldsymbol{u}_p, \tag{3.40}$$

$$(\lambda_p^*\lambda_p\boldsymbol{u}_p + \mu\lambda_p\boldsymbol{u}_p + \mu\lambda_p\boldsymbol{u}_p + \mu^2\boldsymbol{u}_p + \epsilon\boldsymbol{u}_p)a_p = (\lambda_p^* + \mu)d_p\boldsymbol{u}_p, \tag{3.41}$$

$$((\lambda_p + \mu)^*(\lambda_p + \mu) + \epsilon)\boldsymbol{u}_p a_p = (\lambda_p^* + \mu)d_p\boldsymbol{u}_p, \tag{3.42}$$

$$a_p = \frac{(\lambda_p^* + \mu)d_p}{|\lambda_p + \mu|^2 + \epsilon}. \tag{3.43}$$

Finally an estimate for $\boldsymbol{g}$ is available,

$$\boldsymbol{g} = \sum_p \frac{(\lambda_p^* + \mu)d_p}{|\lambda_p + \mu|^2 + \epsilon}\boldsymbol{u}_p. \tag{3.44}$$

The norm is

$$\|\boldsymbol{g}\|^2 = N\sum_p \left(\frac{(\lambda_p^* + \mu)d_p}{|\lambda_p + \mu|^2 + \epsilon)}\right)^* \left(\frac{(\lambda_p^* + \mu)d_p}{|\lambda_p + \mu|^2 + \epsilon)}\right) \tag{3.45}$$

$$= N\sum_p \frac{|\lambda_p^* + \mu|^2|d_p|^2}{(|\lambda_p + \mu|^2 + \epsilon)^2}. \tag{3.46}$$

The pixel value is

$$O(\rho) = \frac{1}{\sqrt{N\sum_p \frac{|\lambda_p^*+\mu|^2|d_p(\rho)|^2}{(|\lambda_p+\mu|^2+\epsilon)^2}}}. \tag{3.47}$$

The contrast ratio contains two integrations, so that

$$CR = \frac{\int_0^a \frac{1}{\sum_p \frac{|\lambda_p^*+\mu|^2|d_p(\rho)|^2}{(|\lambda_p+\mu|^2+\epsilon)^2}}\rho d\rho d\theta}{\int_a^b \frac{1}{\sum_p \frac{|\lambda_p^*+\mu|^2|d_p(\rho)|^2}{(|\lambda_p+\mu|^2+\epsilon)^2}}\rho d\rho d\theta}. \tag{3.48}$$

Once again an analytical expression for this integral is not obtainable and we turn to an asymptotic expression for the contrast ratio.

3.3 Discrete Regularized Sampling for a PEC Cylinder Contrast Ratio Estimate

The Fourier series coefficients of $\boldsymbol{g}$ are

$$a_p = \frac{d_p}{\lambda_p + \mu + \epsilon}, \qquad a_p = \frac{(\lambda_p^* + \mu)d_p}{|\lambda_p + \mu|^2 + \epsilon} \tag{3.49}$$

for simple regularization and Tikhonov regularization respectively. While it is more common to use Tikhonov regularization in practice, in order to determine the con-

trast ratio simple regularization is used as it provides similar results but with less complexity.

Begin by simplifying the eigenvalues to

$$\lambda_q \simeq -N \frac{J_q(ka)}{H_q^{(2)}(ka)}. \tag{3.50}$$

The higher order terms of (2.52) are neglected because they decrease exponentially and are negligibly small. Inspecting equation (3.49) the μ and ϵ are being added to the eigenvalue. Thus both are acting as a regularization parameter. Except for very high SNR values the μ term is considered larger than the regularization parameter and thus ϵ is disregarded. The image pixel value depends on the function $\boldsymbol{g}$ only, and upon removing ϵ we have

$$\|\boldsymbol{g}\| \simeq \sqrt{N} \left[\sum_q \left| \frac{d_q}{\lambda_q + \mu} \right|^2 \right]^{1/2}. \tag{3.51}$$

Recall that d_q is a function of ρ, or the distance from the origin. To find an approximation to the contrast ratio it is beneficial to plot a slice of the reconstructed image for different SNR. For a circular cylinder of radius $k_0 a = \pi$, Figure 3.2 shows a slice through the center of the 2D image from the origin to the image boundary for varying SNR. The blue lines are pixel values inside the scatterer and green lines are pixel values outside the scatterer. By (3.35) the contrast ratio is the area under the curve inside the scatterer divided by the area under the curve outside the scatterer. The area under the curve inside the scatterer does not vary significantly. So the contrast ratio will follow as the inverse of the area under the curve outside the scatterer. This area is proportional to the pixel value at the edge of the cylinder. The pixel value at the edge of the cylinder depends on

$$\|g(a,0)\| \simeq \sqrt{N} \left[\sum_q \left| \frac{\frac{1}{2\pi} i^{-q} J_q(ka)}{-N \frac{J_q(ka)}{H_q^{(2)}(ka)} + \mu} \right|^2 \right]^{1/2}. \tag{3.52}$$

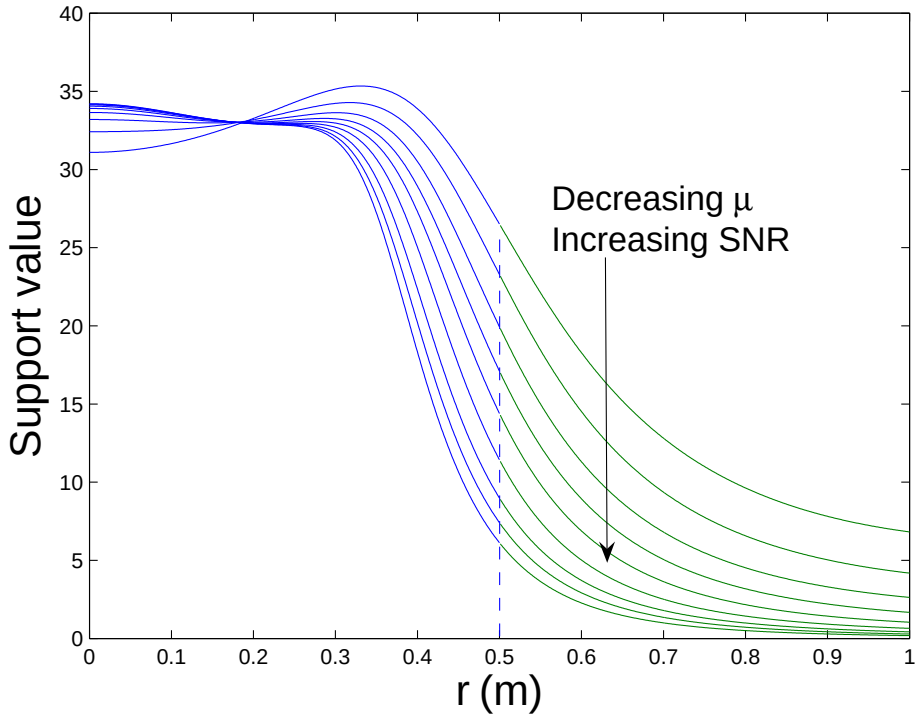


Figure 3.2: Slice of reconstructed cylinder of radius $k_0 a = \pi$ for varying SNR

The pixel value is the inverse of (3.52). The inverse is taken later to make calculations easier. The Bessel and Hankel function ratio has an asymptotic behavior given by

$$J_q(ka) \simeq \frac{e^{q(1-log\frac{2q}{ka})}}{\sqrt{2\pi q}}, \quad q \to \infty \tag{3.53}$$

$$\simeq \frac{e^{-q}}{\sqrt{2\pi q}} \tag{3.54}$$

and

$$H_q^{(2)}(ka) \simeq -iJ_q(ka) \sim \frac{e^{q(log\frac{2q}{ka}-1)}}{\sqrt{2\pi q}}, \quad q \to \infty \tag{3.55}$$

$$\simeq \frac{e^{q}}{\sqrt{q\pi/2}}. \tag{3.56}$$

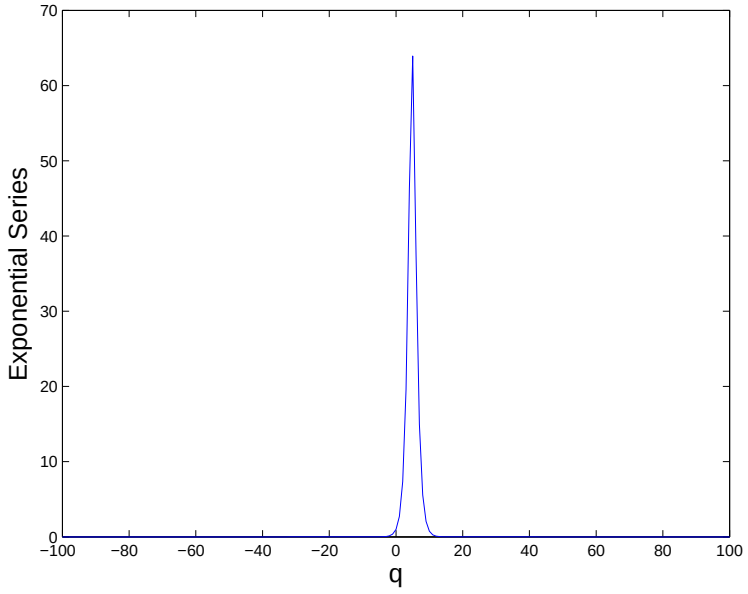


Figure 3.3: Terms in exponential series

Substituting the approximations into (3.52) leads to

$$\|g(a,0)\| \simeq \sqrt{N}\left[\sum\left|\frac{e^{-q}}{-Ne^{-2q}+\mu}\right|^2\right]^{1/2} \tag{3.57}$$

$$\simeq \sqrt{N}\left[\sum\left|\frac{e^{-q}}{-e^{-2q}+\mu'}\frac{1}{N}\right|^2\right]^{1/2} \tag{3.58}$$

where $\mu' = \mu/N$. Figure 3.3 shows a plot of the terms of (3.58). It is seen that there is one peak value in this series. The value of q at which this peak occurs is found by taking the derivative of the series terms with respect to q and setting it to 0.

$$0 = \frac{d}{dq}\frac{-e^{-q}}{e^{-2q}+\mu'} = e^{-2q}+\mu'-2e^{-2q} \tag{3.59}$$

$$= e^{-2q}-\mu' \tag{3.60}$$

and the dominant series term is

$$q_o = -\frac{1}{2}\log\mu'. \tag{3.61}$$

By only using the peak value the series in (3.58) reduces to the single term

$$\|g(a,0)\| \simeq \sqrt{N}\frac{e^{\frac{1}{2}\log\mu'}}{-e^{\log\mu'}+\mu'}\frac{1}{N} \sim \frac{\sqrt{\mu'}}{\sqrt{N}2\mu'} \sim \frac{1}{\sqrt{N\mu'}} \tag{3.62}$$

and the pixel value is

$$D \sim \sqrt{N\mu'}. \tag{3.63}$$

The contrast ratio is now just the inverse of the pixel value, and we have

$$CR \sim \frac{1}{\sqrt{N\mu'}}. \tag{3.64}$$

Making the substitution for μ' leads to

$$CR \sim \frac{1}{\sqrt{N\mu'}} \sim \frac{1}{\sqrt{N\sqrt{\frac{2NP_s}{SNR}}}} \sim SNR^{1/4}. \tag{3.65}$$

This contrast ratio estimate provides a theoretical image quality measure as a function of forward scattering data SNR for regularized sampling.

3.4 Discrete Holographic Backpropagation Tomography for a PEC Cylinder Noise Free Case

For single frequency operation, holographic backpropagation tomography is

$$O(\boldsymbol{r}) = \frac{1}{4\sqrt{\pi^3}}\Re\left\{\sqrt{|k_0|}\int_0^{2\pi}\int_0^{2\pi} u_{\mathrm{s}}(\theta,\phi)\mathrm{e}^{-\mathrm{i}(\boldsymbol{k}^{\mathrm{s}}-\boldsymbol{k}^{\mathrm{i}})\cdot\boldsymbol{r}}|\sin(\phi-\theta)|\mathrm{d}\phi\mathrm{d}\theta\right\} \tag{3.66}$$

where ϕ is the scattering angle and θ is the incident angle. Just as in regularized sampling we must consider the discrete version that correctly replicates how holographic backpropagation tomography is used in practice. Using the midpoint rule

and discretizing in angle,

$$O(\boldsymbol{r}) = \frac{1}{4\sqrt{\pi^3}} \Re \left\{ \sqrt{|k_0|} \sum_n \sum_m u_s(\theta_m, \phi_n) e^{-i(\boldsymbol{k}_n^s - \boldsymbol{k}_m^i)} |\sin(\phi_n - \theta_m)| \right\}. \tag{3.67}$$

This can be made into a matrix equation by defining the vectors

$$\boldsymbol{c} = [e^{-i\boldsymbol{k}_1^s \cdot \boldsymbol{r}} \cdots e^{-i\boldsymbol{k}_M^s \cdot \boldsymbol{r}}]^T = [e^{-ik_0 \hat{\boldsymbol{\rho}}(\phi_1) \cdot \boldsymbol{r}} \cdots e^{-ik_0 \hat{\boldsymbol{\rho}}(\phi_N) \cdot \boldsymbol{r}}]^T, \tag{3.68}$$

$$\boldsymbol{d} = [e^{i\boldsymbol{k}_1^i \cdot \boldsymbol{r}} \cdots e^{-i\boldsymbol{k}_N^i \cdot \boldsymbol{r}}]^T = [e^{ik_0 \hat{\boldsymbol{\rho}}(\theta_1) \cdot \boldsymbol{r}} \cdots e^{ik_0 \hat{\boldsymbol{\rho}}(\theta_N) \cdot \boldsymbol{r}}]^T \tag{3.69}$$

and the matrices

$$F_{mn} = u_s(\theta_m, \phi_n), \tag{3.70}$$

$$P_{mn} = |\sin(\phi_n - \theta_m)|. \tag{3.71}$$

With these definitions the pixel value is

$$O(\boldsymbol{r}) = \frac{\sqrt{|k_0|}}{4\sqrt{\pi^3}} \Re \left\{ \boldsymbol{c}^H (\boldsymbol{F} \odot \boldsymbol{P}) \boldsymbol{d} \right\} = \frac{\sqrt{|k_0|}}{4\sqrt{\pi^3}} \Re \{ < (\boldsymbol{F} \odot \boldsymbol{P}) \boldsymbol{d}, \boldsymbol{c} > \} \tag{3.72}$$

where $< \cdot, \cdot >$ is the L_2 inner product and $\odot$ is the Hadamard product. Thus a scatterer image can be found by matrix vector multiplication and inner product for each reconstruction point in the image. Holographic backpropagation tomography in matrix form allows for simple fast numerical implementation.

With a discrete version of holographic backpropagation, an analytical solution to pixel values for the PEC circular cylinder is easily found following a similar procedure as with regularized sampling.

For the case when the transmitters and receivers are the same, we have

$$\boldsymbol{\phi} = [\phi_1 \cdots \phi_N] = \boldsymbol{\theta} = [\theta_1 \cdots \theta_M] \tag{3.73}$$

where N is the number of transmitters and M the number of receivers. The vectors $\boldsymbol{c}$ and $\boldsymbol{d}$ can be expanded using the cylindrical wave expansion

$$\boldsymbol{d} = \sum_q i^q J_q(k_0\rho)e^{-iq\boldsymbol{\varphi}}, \tag{3.74}$$

$$\boldsymbol{c} = \sum_q i^q J_q(k_0)\rho)e^{-iq\boldsymbol{\varphi}}. \tag{3.75}$$

The cylindrical expansion was derived for a plane wave propagating in the x direction. In this case we have a plane wave propagating in a direction denoted by the elements of $\boldsymbol{\phi}$. Therefore the argument of the expansion is

$$\boldsymbol{\varphi} = \phi_r - \boldsymbol{\phi} \tag{3.76}$$

and

$$\boldsymbol{d} = \sum_q i^q J_q(k_0\rho)e^{-iq(\phi_r-\boldsymbol{\phi})} \tag{3.77}$$

$$= \sum_q i^q J_q(k_0\rho)e^{-iq\phi_r}e^{jq\boldsymbol{\phi}} \tag{3.78}$$

$$= \sum_q d_q e^{iq\boldsymbol{\phi}}. \tag{3.79}$$

The matrix equation (3.72) contains a Hadamard product which can be regarded as a filter. Since the contrast ratio is used to determine image quality, constants are neglected. Also for contrast ratio calculations the filter is left out because its effects are similar at any SNR. This allows for an analytical contrast ratio to be determined for holographic backpropagation tomography. With these approximations the pixel value becomes

$$O(\boldsymbol{r}) = \Re\left\{\boldsymbol{c}^H\boldsymbol{F}\boldsymbol{d}\right\} = \Re\{< \boldsymbol{F}\boldsymbol{d}, \boldsymbol{c} >\}. \tag{3.80}$$

Once again it is useful to employ the eigenvalues of the scattering amplitude matrix. The eigenvalue problem for holographic backpropagation tomography is

$$O(\boldsymbol{r}) = \Re\left\{\sum_q d_q \boldsymbol{c}^H \boldsymbol{F} e^{iq\boldsymbol{\phi}}\right\} \tag{3.81}$$

$$= \Re\left\{\sum_q d_q \lambda_q \boldsymbol{c}^H e^{iq\boldsymbol{\phi}}\right\} \tag{3.82}$$

which reduces the pixel value to an inner product. Also for the case of when $\boldsymbol{\phi} = \boldsymbol{\theta}$ then $\boldsymbol{c} = \boldsymbol{d}$ and

$$O(\boldsymbol{r}) = \Re\left\{\sum_q d_q \lambda_q \boldsymbol{d}^H e^{iq\boldsymbol{\phi}}\right\} \tag{3.83}$$

$$= \Re\left\{\sum_q \lambda_q d_q e^{iq\boldsymbol{\phi}} \sum_r d_r e^{ir\boldsymbol{\phi}}\right\} \tag{3.84}$$

$$= N\Re\left\{\sum_q |d_q|^2 \lambda_q\right\} \tag{3.85}$$

$$= N\sum_q |d_q|^2 \Re\left\{\lambda_q\right\}. \tag{3.86}$$

3.5 Discrete Holographic Backpropagation Tomography for a PEC Cylinder With Noise

The noise matrix is added to the scattering amplitude matrix and the holographic backpropagation tomography method with noise is

$$O(\boldsymbol{r}) = \Re\left\{\boldsymbol{c}^H(\boldsymbol{F} + \boldsymbol{\eta})\boldsymbol{d}\right\}. \tag{3.87}$$

By using the expanded version of $\boldsymbol{d}$ the pixel value is simplified by the eigenvalue problem and noise matrix approximation to

$$O(\boldsymbol{r}) = \Re\left\{\sum_q d_q \boldsymbol{c}^H(\boldsymbol{F}+\boldsymbol{\eta})e^{iq\boldsymbol{\phi}}\right\} \tag{3.88}$$

$$= \Re\left\{\sum_q d_q \boldsymbol{c}^H(\lambda_q+\mu)e^{iq\boldsymbol{\phi}}\right\}. \tag{3.89}$$

Following the same steps as the previous section the analytical pixel value with noise is

$$O(\boldsymbol{r}) = N\sum_q |d_q|^2(\mu + \Re\{\lambda_q\}) \tag{3.90}$$

$$= -N\sum_q (-1)^q J_q^2(k_0\rho)\Re\{\lambda_q+\mu\}. \tag{3.91}$$

The contrast ratio is then

$$CR = \frac{\sum_q (-1)^q \Re\{\lambda_q+\mu\}\int_o^a J_q^2(k_0\rho)d\rho}{\sum_q (-1)^q \Re\{\lambda_q+\mu\}\int_a^b J_q^2(k_0\rho)d\rho}. \tag{3.92}$$

Obtaining a solution to these integrals is difficult and does not lead to a simple expression or intuition as how the contrast ratio changes with SNR.

3.6 Discrete Holographic Backpropagation Tomography for a PEC Cylinder SNR Transition Estimate

As no exact solution to the contrast ratio exists a simplification is sought. Looking at equation (3.91) the Bessel function stays constant as the SNR varies and is considered insignificant. The relation between μ and λ_q describes a SNR transition point. For low SNR, noise dominates reconstructed images, and for high SNR the images are essentially noise free. A transition point occurs between the two states of images. This transition point is when $\mu = \Re\{\lambda_q\}$. Recall that the simplified eigenvalues without the higher order terms of (2.52) are $\lambda_q = -NJ_q(k_0a)/H_q^{(2)}(k_0a)$. Using the estimate

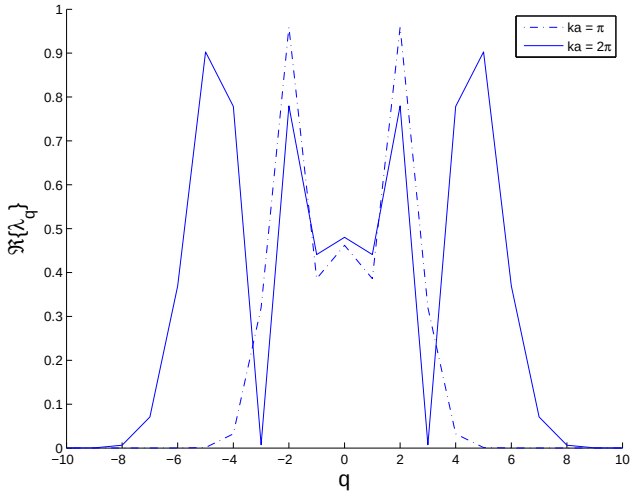


Figure 3.4: The real part of the scattering amplitude eigenvalue for ka of two different values

$$\max_n \frac{J_q(k_0 a)}{H_q^{(2)}(k_0 a)} \simeq 1 \tag{3.93}$$

an approximation to the largest eigenvalue is N. Figure 3.4 exhibits this approximation. This approximation is especially true for small cylinders. The transition occurs when $\mu \simeq N$ and using (3.22) the SNR transition point is

$$\text{SNR} \simeq \frac{2P_{sig}}{N}. \tag{3.94}$$

The signal power is verified as $P_{sig} \simeq k_0 a$ by direct calculation. An estimate of the SNR transition is now

$$\text{SNR} \simeq \frac{2k_0 a}{N}. \tag{3.95}$$

This SNR is the point where good image resolution is obtainable. Lower SNR forward data produces a noise dominated image, and at higher SNR noise free images are expected. It is also important to note that the SNR transition point improves with increasing N. This is unlike regularized sampling whose contrast ratio stays constant with changing N. In Chapter 4 Figure 4.8 provides simulated confirmation of this

effect. Theorectically and in practice increasing the number of transcievers allows for better reconstructed image quality with holographic backpropagation but not regularized sampling.

Chapter 4

Simulations

As verification of the theoretical estimates of the previous chapter, numerical simulations of several inverse scattering methods are given to quantify image quality as a function of SNR. Forward scattering data was computed using a boundary element method. To ensure data accuracy a discretization of at least 10 mesh elements per wavelength was used. Bistatic scattering amplitudes were computed at 64 evenly spaced incident and receive angles. Realizations of complex circular Gaussian random variables of noise power determined by (2.38) were added to the scattering data to model system noise. The contrast ratio versus SNR computations were performed using Monte Carlo simulations.

To provide insight into the noise effects on image quality it is beneficial to first look at some reconstructed images produced at various SNR values. Figures 4.1-4.3 show reconstructed images of a PEC circular cylinder using holographic backpropagation tomography, diffraction tomography, and regularized sampling methods. Figures 4.4-4.6 show reconstructed images of a F15 aircraft profile for the same three algorithms. These reconstructed images provide a good visualization of noise behavior. The noise perturbations in regularized sampling and holographic backpropagation tomography are spatially correlated whereas the noise in diffraction tomography is speckled. Image shape begins to be discernible between -10dB and -5dB for both holographic backpropagation tomography and regularized sampling. It is between -5dB and 0dB that the circular cylinder is distinguishable in diffraction tomography.

Figure 4.7 shows the average image contrast ratio for the three different methods for various PEC cylinder profiles along with the image quality estimate for regularized sampling. The same behavior noticed in reconstructed images is once again

seen in this figure. For the circular cylinder, the numerical contrast ratio is very close the theoretical image quality estimate (3.65) of regularized sampling. Holographic backpropagation tomography with $k_0 a = \pi$ and $N = 64$ predicts a transition SNR of -10 dB. The knee of the holographic backpropagation tomography contrast ratio curve of figure 4.7 is approximately -10 dB. Therefore the SNR transition estimate predicts a good image quality at -10 dB and the numerical simulation confirms the prediction. The contrast ratio for holographic backpropagation tomography is very near its peak at 0dB, after which the small increase in contrast ratio is due to an increase in edge sharpness. Diffraction tomography exhibits the same behavior as holographic backpropagation which is expected as both methods rely on the linearizing of the Born approximation. It does have a higher SNR transition which is an agreement with the observation of [6] that diffraction tomography is more noise sensitive than filtered backpropagation tomography. Regularized sampling makes its transition from a noise dominated image beginning at 0 dB and does not reach a highly resolved image until much higher SNR.

The transition from noise dominated images to higher quality images of more complex shaped scatterers occur at different SNR than the circular cylinder case but the same qualitative image characteristics are observed. Mainly, the tomographic methods are less sensitive to noise than regularized sampling. Higher quality images of complex scatterers are obtained using holographic backpropagation at low SNR.

Figure 4.8 displays numerical contrast ratios for images reconstructed with different values of N. Regularized sampling noise sensitivity remains constant as a function of N. This agrees with the analytical image quality estimate. The knee of the holographic backpropagation tomography curves change as N varies. With increasing N the transition from a noise dominated image to a resolved reconstruction occurs at lower SNR. The analytical SNR transition estimate matches the numerical observations. The knee of the contrast ratio curves for diffraction tomography does not change with N. The contrast ratio of the noise dominated images decreases as the number of transceivers increase. Reconstructed images of high quality using diffraction tomography are still expected to occur at the same SNR regardless of N.

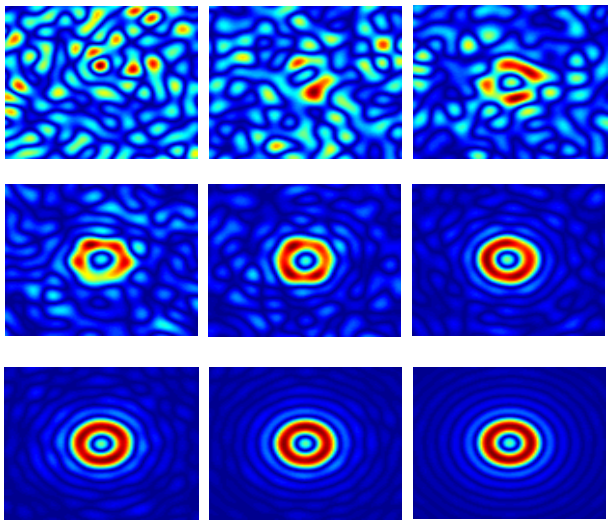


Figure 4.1: Reconstructed images for a PEC circular cylinder using holographic backpropagation tomography at varying SNR. SNR values are -25dB, -15dB, -10dB, -5dB, 0dB, 5dB, 10dB, 15dB, 25dB from the top left image to the bottom right.

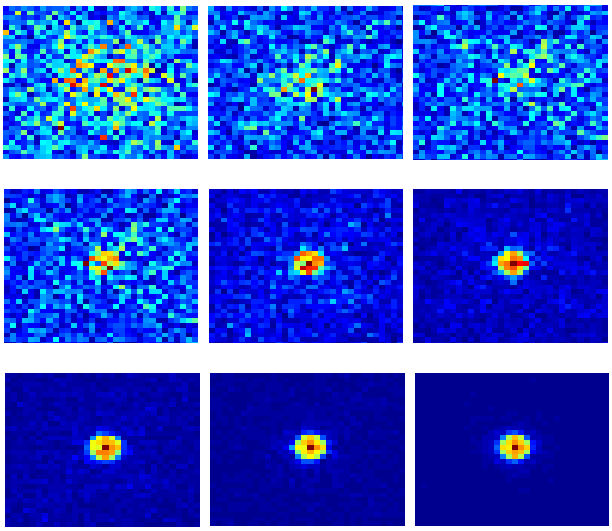


Figure 4.2: Reconstructed images for a PEC circular cylinder using diffraction tomography at varying SNR. SNR values are -25dB, -15dB, -10dB, -5dB, 0dB, 5dB, 10dB, 15dB, 25dB from the top left image to the bottom right.

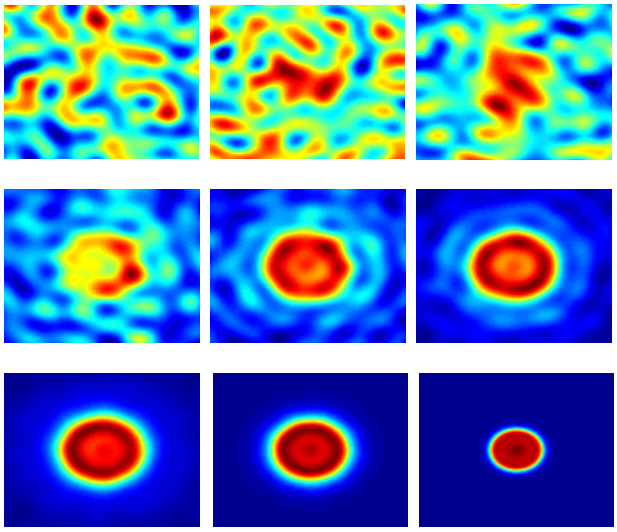


Figure 4.3: Reconstructed images for a PEC circular cylinder using regularized sampling at varying SNR. SNR values are -25dB, -15dB, -10dB, -5dB, 0dB, 5dB, 15dB, 25dB, and Noise Free starting from the top left image to the bottom right.

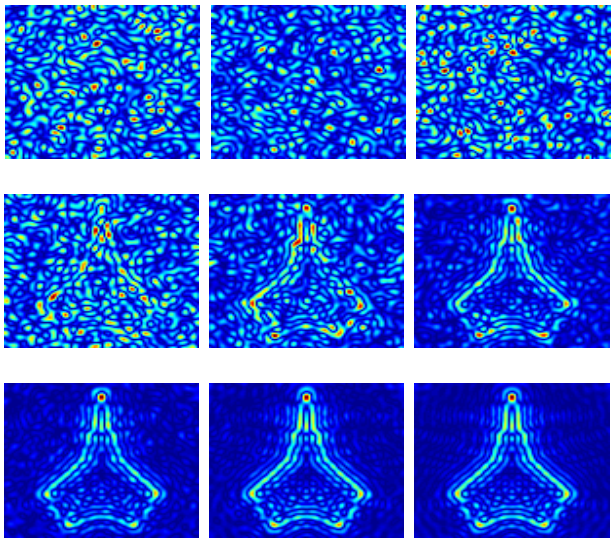


Figure 4.4: Reconstructed images for a PEC F15 shaped cylinder using holographic backpropagation tomography at varying SNR. SNR values are -25dB, -15dB, -10dB, -5dB, 0dB, 5dB, 10dB, 15dB, 25dB from the top left image to the bottom right.

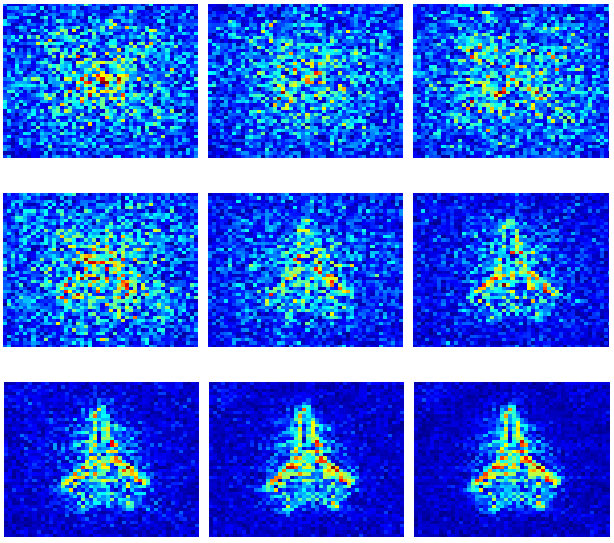


Figure 4.5: Reconstructed images for a PEC F15 cylinder using diffraction tomography at varying SNR. SNR values are -25dB, -15dB, -10dB, -5dB, 0dB, 5dB, 10dB, 15dB, 25dB from the top left image to the bottom right.

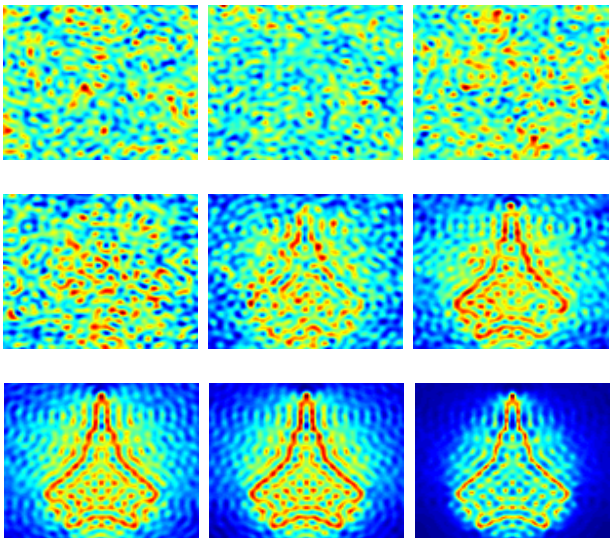
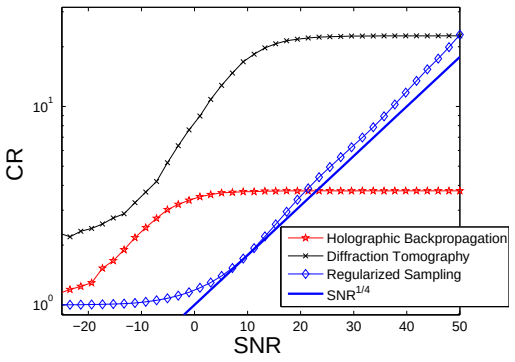
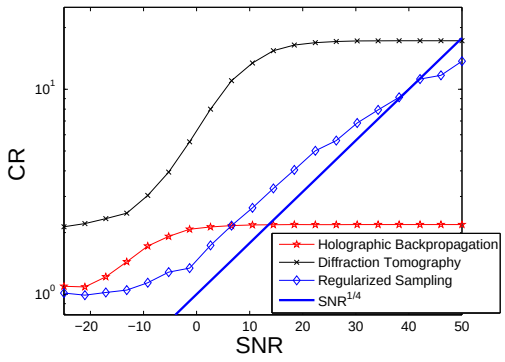


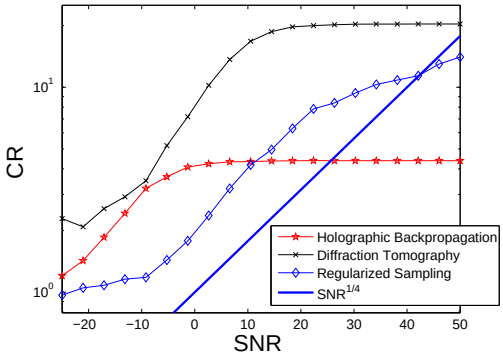
Figure 4.6: Reconstructed images for a PEC F15 shaped cylinder using linear sampling at varying SNR. SNR values are -25dB, -15dB, -10dB, -5dB, 0dB, 5dB, 10dB, 15dB, and 25dB starting from the top left image to the bottom right.



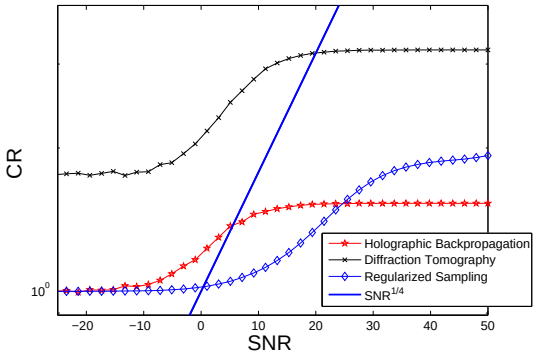
(a) Cylinder



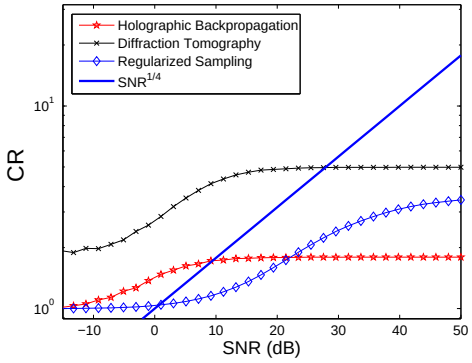
(b) Square



(c) Triangle



(d) F15 profile



(e) B2 profile

Figure 4.7: Contrast ratio curves as a function of SNR for various scatterer profiles.

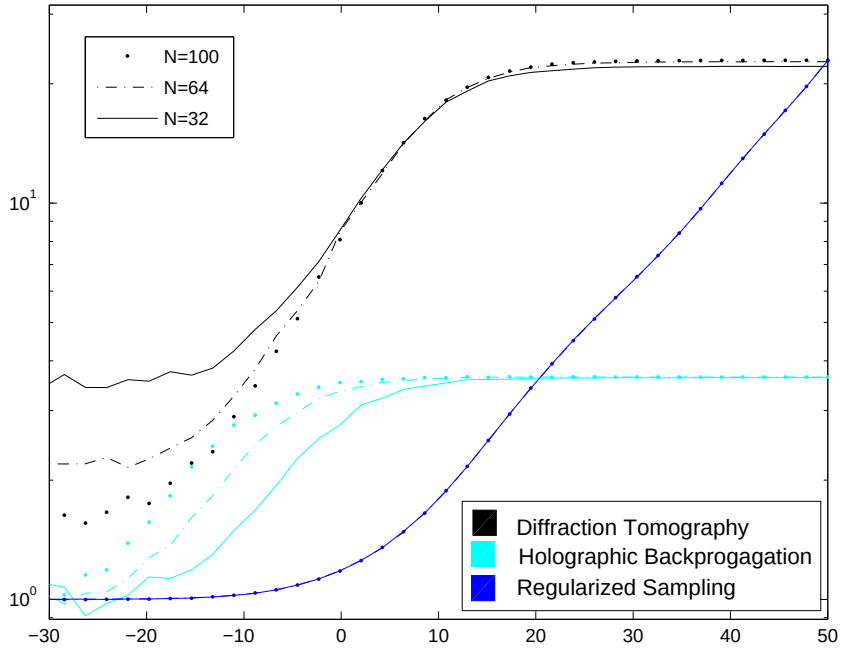


Figure 4.8: Contrast ratio for circular cylinder as the number of transceivers varies.

Chapter 5

Conclusion

The noise behavior of holographic backpropagation tomography and Colton-Kirsch regularized sampling has been studied analytically and empirically using a simple but effective metric for quantifying reconstruction image quality as a function of forward scattering data SNR. Of the tomographic methods considered, holographic backpropagation is the least sensitive to noise which coincides with previous literature. The approached used in this thesis to measure noise sensitivity allows for comparisons between algorithms that reconstruct different properties of the scatterer. It has been shown that regularized sampling is more noise sensitive than tomographic methods.

Noise sensitivity of holographic backpropagation also depends on the number of transceivers. Image quality improves as the number of transceivers increase. Regularized sampling, on the other hand, is independent of transceiver density. The contrast ratio of noise dominated images using diffraction tomography depends on the number of transceivers but the SNR transition from noise dominated to resolved images is unaffected by transceiver number.

These conclusions hold for canonical scatterers, from which analytical estimate are derived, as well as more complex shaped or electrically large scatterers.

5.1 Future Work

This work provides a beginning understanding of the noise characteristics for a few inverse scattering methods, and could be expanded to include other common methods of inverse scattering. Theoretical estimates and numerical simulations could be performed for such methods as the distorted born iterative method, time reversal,

or other algorithms. The contrast ratio can be used for comparison of image qualities as a function of SNR for any type of inverse scattering method.

The dependence of image quality on transceiver number was not fully explained in this thesis. Each method of inverse scattering investigated produces reconstructed images using the solution to three different mathematical operations. Diffraction tomography uses a FFT. Holographic backpropagation tomography computes a inner product and matrix multiplication. Regularized sampling relies on a matrix inversion and regularization. The difference of dependence on N seems to be attributed to the difference of these underlying mathematics operations, but further investigation and explanation is still needed.

The noise in reconstructed images is spatially correlated for regularized sampling and holographic backpropagation tomography but speckled for diffraction tomography. This observation has not been given sufficient attention and requires further study.

Ultimately, these studies aim to improve the accuracy of algorithms of inverse scattering and more generally the inverse problem. The observation and measures made here can be used to improve existing algorithms or develop new ones.

Appendix A

Appendix: Matlab Code Used in Simulations

A.1 Forward Scattering Amplitudes

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                            Thomas Sorensen
%
% Forward Scattering problem
%
%    Implement the 2D MOM to compute the surface current on a PEC cylinder
%    of a half wavelength radius due to a TM polarized incident plane wave
%    traveling in the multiple angles
%
%    Convert surface currents to Numerical Scattering Amplitudes
%
%    Compare to Analytical Solution
%
%    Polygon representations of scatterers are taken from Dr. Warnick's
%    forward code
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Es San S] = ScatAmp( Npoints, scat_type,SC,snr,f,NTran);
%function MOMforwardScat
% input Npoints: the number of points per wavelength along PEC surface
% output S: the scattering amplitude
%
%
% computes the scattering amplitude from a 2D cylinder due to multiple
% incident plane waves
% Calculates the surface current on a PEC Cylinder
% Implements a 2D MOM
% half wavelengths radius
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   Constants   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
Npoints   = 50;                     % Number of points per Wavelength
scat_type = 'cylinder';             % along the surface of the PEC
SC        = .5 % scales the big scatterers down
f         = 300e6;
NTran     = 64;
%% Constants

Eo      = 8.854e-12;                  % Electric permeability of free space
MUo     = 4*pi*1e-7;                  % Magnetic permeability of free space
Co      = 1/sqrt(MUo*Eo);             % Speed of light in free space
ETAo    = sqrt(MUo/Eo);               % Wave Impedence
e       = exp(1);
GAMMA   = 1.781;
E       = 1;                          % E field amplitude
w       = 2*pi*f;
k       = 2*pi*f/Co;
wlength = Co/f;
h       = wlength/Npoints;

%% Physical Parameters

rad     = SC;
rho     = 100000;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%   Problem   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% The the polarc representation of the scatterer
[Uppoly x y] = PolyMeshScatFull(scat_type,SC,Npoints);
% [Uppoly x y] = PolygonScat(scat_type,SC,Npoints);
% polyarc mesh plot

x1p = Uppoly(:,1);
y1p = Uppoly(:,2);
x2p = Uppoly(:,3);
y2p = Uppoly(:,4);

x1p = x;
y1p = y;
```

```
Nx = length(x);
Ny = length(y);

%% plot scatterer
figure(2)
clf
plot(x1p, y1p,'-')
% axis([min(x1p) max(x1p) min(y1p) max(y1p)])
% hold on
% grid on
title('PEC Scatterer Mesh')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                          Fill A(m,n) matrix

%% Revamp the code to have for loops for the A matrix
A = zeros(Nx,Ny);

Adiag = sub2ind([Nx Ny], 1:Nx,1:Nx);

xm = x;
xm = xm(:,ones(1,Nx));
xn = x.';
xn = xn(ones(Nx,1),:);
ym = y;
ym = ym(:,ones(1,Nx));
yn = y.';
yn = yn(ones(Nx,1),:);

A        = (ETAo*k*h/4)*besselh(0,2,k*sqrt((xm-xn).^2 + (ym-yn).^2));
A(Adiag) = (ETAo*k*h/4)*(1 - (2*i/pi)*log(GAMMA*k*h/(4*e)));
[L U P]  = lu(A);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                Scattering Amplitude and Surface Currents

dPhi = 2*pi/(NTran);
PhiScat  = 0: dPhi : 2*pi-dPhi;

Ns = length(PhiScat);
Cn = zeros(Nx,NTran);
C = zeros(Nx,1);
S = zeros(NTran,NTran);
% Numerically Calculate the Scattering Amplitude
```

```
for m = 1:NTran

    % Calculate the E field for incident angles on range of 0 to 2 pi
    kx  = k*cos(PhiScat(m));
    ky  = k*sin(PhiScat(m));
    Ez  = E*exp(-i*(ky*y + kx*x));    % E field for incident angle PhiScat
    J   = U\(L\(P*Ez));               % Unknown currents on scatterer for incident field
    if (m == 1)
       J0 = J;
    end
       % Fill the Cn Matrix; and calc
       for n = 1:NTran

            C        = h*exp(i*k*(x*cos(PhiScat(n)) + y*sin(PhiScat(n))));
            S(m,n)   = (-k*ETAo/4)*(C)'*J;%      E^s = sqrt(-2*i/(pi*kem*rho)) exp(i*kem*rho) S
%

        end

end

% Add complex gaussian white noise to scattering amplitude
% S = awgn(S,snr);
% save ScatAmpCyl S;

%--- Calculate the Efields
%      E^s = sqrt(-2*i/(pi*kem*rho)) exp(i*kem*rho) S

Es =  sqrt(2*i/(pi*k*rho))*exp(-i*k*rho).*S;


%-------------------------------------------------------------------------
%%                  Calculate the Analytical Scat. Amps.
% % Pn = 1;
% % San = zeros(NTran,NTran);
% %
% % for mm = 1:NTran
% %
% %     for m = 1:NTran
% %
% %         n = -50:50;
% %
% %             San(m,mm) = sum(((-1).^n).*besselj(n,k*rad).*exp(i.*n.*(PhiScat(m)
% %                     .. - (PhiScat(mm)))./besselh(n,2,k.*rad));
```

```
% %
% %     end
% %
% % end
San = 0;

%-------------------------------------------------------------------------
%%                 Calculate Numerical and Analytical RCS

RCSn = (4/k)*(abs(S)).^2;
% RCSa = (4/k)*(abs(San)).^2;

%-------------------------------------------------------------------------
%%                  Calculate Analytical Surface currents
% n = -20:20
% NJ0 = length(J0);
% PhiScat1 = linspace(-pi,pi,NJ0);
% for m = 1:NJ0
% Ja(m)  = sum(2*E/(pi*k*rad*ETAo).*i.^n.*exp(i.*n.*PhiScat1(m))./(besselh(n,2,k*rad)));
% end

% % figure(7)
% % plot(PhiScat1+pi,imag(Ja),'-',PhiScat1+pi,imag(J0),'--')
% % xlabel('Scattering Angle')
% % ylabel('Current Amplitude')
% % title('Scattering Currents')
% % legend('Analytical','Numerical')



%-------------------------------------------------------------------------
%%                          Plot Scattering Amps
%
% figure(3)
% semilogy (PhiScat,abs(S(1,:)))
% hold on
% semilogy (PhiScat,abs(San(1,:)),'*r')
% grid on
% hold off
% % title('Scattering Amplitude')
% % xlabel('Scattering Angle')
% % ylabel('Scattering Amplitude')
% % legend('MOM','Analytical')

%-------------------------------------------------------------------------
```

```
%%                          Plot Scattering Widths(RCS)
%
% figure(4)
% plot(PhiScat,RCSn(Pn,:), '-b', PhiScat,RCSa(Pn,:),'.r')
% % semilogy(PhiScat,(RCSn))
% grid on
% title('RCS 2D MOM')
% legend('MOM','Analytical')
% xlabel('Scattering Angle')
% ylabel('RCS')
% size(S)
save ScatAmpCyl S

% diff = RCSn.' - RCSa;
```

A.2 Holographic Backpropagation Tomography

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                                    Thomas Sorensen
%                                 Linearized  Tomography
%
%                    Described in M Brandfass et al "A comparison
%                     of the Colton-Kirsch inverse scattering
%                     methods with linearized tomographic inverse
%                     scattering" see below
%
% Takes scattering amplitudes calculated in ScatAmp.m and creates a image
% of the scatterer.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [object xRrec   yRrec] = BrandfassTomo(scat_type,Er,xmax,ymax,SC,snr,saved,f,N)
% Constants %%%%%%%%
if 1==1
scat_type = 'cylinder';
Er        = 1;
xmax      = 1;
ymax      = 1;
SC        = .5;
snr       = 25;
saved     = 2;
```

```
f         = 300e6;
N         = 64;
end
%
% N            = 32;
NTran        = N;
NRec         = N;
Npoints      = 20;

Eo           = 8.854e-12;              % Electric permeability of free space
Escat        = 1 * Eo;
MUo          = 4*pi*1e-7;             % Magnetic permeability of free space
Co           = 1/sqrt(MUo*Eo);      % Speed of light in free space
ETAo         = sqrt(MUo/Eo);        % Wave Impedence
Eta          = sqrt(MUo/Escat);
C            = 1/sqrt(MUo*Escat);
e            = exp(1);
GAMMA        = 1.781;
E            = 1.0;                            % E field amplitude
w            = 2*pi*f;
k            = w/C;
wlength      = C/f;


% Read in Scattering Amplitudes %%%%%%%%
%[S,sx,sy,phi_i,phi_s,j1] = Dr_Warnick_Forward(13,0)
if(Er == 1)

    if(saved == 1)

        if (strcmp(scat_type, 'B2Bomber'))
                 load ScatAmpB2
%                  S = awgn(S,snr,'measured');

        end

        if (strcmp(scat_type, 'square'))
               load ScatAmpSq
        end

        if (strcmp(scat_type, 'triangle'))
                load ScatAmpTri
        end

        if (strcmp(scat_type, 'f15'))
```

```
                load ScatAmpf15;
%                 S = awgn(S,snr,'measured');

        end

        if (strcmp(scat_type, 'cylinder'))
                load ScatAmpCylAn;
%                 if N==64
%                 load ScatAmpCylAn64;
%                 end
%                 if N==100
%                 load ScatAmpCylAn100;
%                 end
%                 if N==32
%                 load ScatAmpCylAn32;
%                 end

                S = San;
%                 S = awgn(S,snr,'measured');
        end

        if (strcmp(scat_type, 'cresent'))
                load ScatAmpCres;
        end

        if (strcmp(scat_type, 'vfy218'))
                load ScatAmpfVF;

%                 S = awgn(S,snr,'measured');
                %S = awgn(S,snr);
% my own noise
%                 snr1 = 10^(snr);
%                 [ssr ssc] = size(S);
%                 Sigrms = (norm(S,'fro'));
%                 alpha = sqrt(Sigrms^2/snr1);
%                 noise = randn(ssr);
%                 S = S + alpha*noise;
        end

          if (strcmp(scat_type, 'C'))
                load ScatAmpC;
%                 S = awgn(S,snr,'measured');
                %S = awgn(S,snr);
% my own noise
%                 snr1 = 10^(snr);
```

```
%                  [ssr ssc] = size(S);
%                  Sigrms = (norm(S,'fro'));
%                  alpha = sqrt(Sigrms^2/snr1);
%                  noise = randn(ssr);
%                  S = S + alpha*noise;
        end


    else

    %[S,sx,sy,phi_i,phi_s,j1] = Dr_Warnick_Forward(13,0)

    [S] = ScatAmp(15,scat_type,SC,snr,f,NTran); %scat amp matrix-input type of scatterer

    end


else

    if(saved == 1)

        if (strcmp(scat_type, 'B2Bomber'))
                 load ScatAmpB2DE2
%                  S = awgn(S,snr,'measured');


        end

        if (strcmp(scat_type, 'f15'))
                 load ScatAmpf15DE;
%                  S = awgn(S,snr,'measured');

        end

        if (strcmp(scat_type, 'cylinder'))
                  load ScatAmpCylDE2;
%                   S = awgn(S,snr,'measured');
        end

        if (strcmp(scat_type, 'C'))
                  load ScatAmpCDE2;
%                  S = awgn(S,snr,'measured');
        end
```

```
    else
    Er = Er
    SC = SC
    [S] = VolumeMOM(Npoints,scat_type,Er,SC,NTran,NRec);
    end

end

% bypass all the above forward codes
if (1==2)
    S =Dr_Warnick_Forward(12,0,300e6);
end

%% Analytical Scattering Amplitudes
if 1==5
 [eigSan S] = AnalyticalScatAmpsPECCyl(NTran,f);
end




%% add noise

if (1==1)

%%%% Correct Noise %%%%%%%%%%%%%
SNRt = snr;
snr     = 10.^(snr/10);
if 1==1 % add noise matrix

  display('Noise Matrix')
  sp      = sum(abs(S(:)).^2)/length(S(:));
  sn      = sqrt(sp/snr);
  Snoise  = sn.*(randn(size(S)) + i*randn(size(S)));
  S       = S + Snoise;
end

if 1==2

  display('Mu Identity')
  sp      = sum(abs(S(:)).^2)/length(S(:))-Delta;
  mu      = sqrt(sp/snr)*sqrt(2*NTran);
  Snoise  = mu*eye(size(S));
  S       = S + Snoise;
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end %% add noise

%% plot the scattering amplitudes
%
% figure(9876)
% semilogy(abs(S).^2)

% determine number of receivers, transmitters and set up discrete angles

[sr st]     = size(S);
dPhit       = 2*pi/(st);
dPhir       = 2*pi/(sr);
PhiScatt    = 0: dPhit : 2*pi-dPhit;
PhiScatr    = 0: dPhir : 2*pi-dPhir;
Nst         = length(PhiScatt);
Nsr         = length(PhiScatr);
% Set up the reconstruction rectangular grid

if(1==18) % Load scattering amplitudes

sa       = load('scat3_sa.dat');
F        = sa(:,3)+j*sa(:,4);
PhiScatt = sa(1:64,2)*pi/180;
NA       = sqrt(length(F));
F        = reshape(F,NA,NA)
S        = F;
PhiScatt = PhiScatt.';
PhiScatr = PhiScatt;

end

gridsz = 100;
xRrecmax       = xmax+1;
yRrecmax       = xRrecmax;
dxRrec         = 2*xRrecmax/(gridsz-1);
dyRrec         = 2*yRrecmax/(gridsz-1);
xRrec          = -xRrecmax: dxRrec : xRrecmax;
yRrec          = -yRrecmax: dyRrec : yRrecmax;
NxRrec         = length(xRrec);
NyRrec         = length(yRrec);
```

```
% Frequency spectral vector - we only excite with one frequency so this is
% a constant
Fw =1;

% set up the rotating coordinate sytstem(see Brandfass et al)
Rphy     = cos(PhiScatr); % Rph- R prime hat - the receive direction
Rphx     = sin(PhiScatr);
Kihy     = -cos(PhiScatt);
Kihx     = -sin(PhiScatt);

% Kih = K incident hat - the incendent direction - by convention the + Kih
% is pointing away from the scatterer therefore it must be negated in the
% code to make the wave direction pointing to the scatterer



kiperpx  = -cos(PhiScatt + pi/2);  % kiperp is offset 90 degrees from Kih
kiperpy  = -sin(PhiScatt + pi/2);


scatterer = zeros(NxRrec,NyRrec);

%% matrisize it

Phis = PhiScatr(ones(st,1),:); %% Row has changing angle
Phii = PhiScatt.';
Phii = Phii(:,ones(1,sr)); %% Row has constant angle

% absKiRs = ((abs(-sin(Phii).*cos(Phis) - cos(Phii).*sin(Phis))));
% absKiRs = ((abs(-sin(Phii).*cos(Phis) - cos(Phii).*sin(Phis))));


% Reconstruct the scatterer
ORec  = zeros(sr,st);
OTran = zeros(sr,st);
OCon  = zeros(sr,st);
Odiag = sub2ind([sr st],1:sr,1:st);
sinpips = abs(sin(-Phis+Phii));
sinpips = abs(sin(Phis - Phii)./2 - sin(Phii-Phis)./2);
for mm = 1:NxRrec
                                  % cycle through each reconstruction point
        for l = 1:NyRrec

             Rx = xRrec(mm);
```

```
            Ry  = yRrec(l);

% %             %%%%%%%%%%%% slow and done pointwise %%%%%%%%%%%%%%%%%%%
% %          for m = 1:Ns
% %                                  % cycling through each transmit and
% %                                  % receive angle
% %               for n = 1:Nsr
% %
% %
% %                     % Integrating by the midpint rule
% %
% %                 O(:,n)  =  S(n,:)  * (exp(i .*k.*(Rx.*(Rphx(n)-Kihx) + Ry.*(Rphy(n)-Kihy))) ...
% %                     .* abs((kiperpx .* Rphx(n)) + (kiperpy .* Rphy(n)))).' .* dPhir^2;
% %
% %               end
% %
% %           end
% %           % adding up the discrete pieces of the integral
% %           scatterer(j,l) = (1/( 4 * sqrt(pi^3) )) * real (( sqrt(2*i/pi)/Fw)  * sum(sum(O)));
% %           scatterer(j,l) = (1/( 8*pi )) * real ( 1/( sqrt(i)*Fw)  * sum(sum(O)));
% %
% %             %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      %%%%%%%%%%%%%%%%%%%  matricised  %%%%%%%%%%%%%%%%%%%%%%%%
%              ORec(Odiag)  = exp(-i*k*(Rx.*Rphx + Ry.*Rphy ));
%              OTran(Odiag) = exp(i*k*(Rx.*Kihx.' + Ry.*Kihy.' ));

%      O1  =  exp(-i*k*(Rx.*Kihx + Ry.*Kihy )) *S.'* exp(i*k*(Rx.*Rphx.' + Ry.*Rphy.' )).*absKiRs;
%      O1  = exp(i*k*(Rx.*Rphx + Ry.*Rphy )) *S.'* exp(-i*k*(Rx.*Kihx.' + Ry.*Kihy.' )).*absKiRs;
%      O1  = exp(-i*k*(Rx.*Rphx + Ry.*Rphy )) *S* exp(i*k*(Rx.*Kihx.' + Ry.*Kihy.' ));
%       O  = sum(sum(ORec*(S.*OTran)*sinpips));
%      O1  = sum(sum(ORec*S*OTran.*sinpips));

        g     = exp(-i*k*(Rx.*Rphx + Ry.*Rphy )).'; %% column
%               h     = exp(i*k*(-Rx.*Rphx + Ry.*Rphy )).';

        h     = exp(-i*k*(Rx.*Kihx.' + Ry.*Kihy.' )); %% column
%               h     = conj(g);
        F     = S;
%               F = S.*sinpips;
        O     = h'*(S.*sinpips)*g;
        O1    = g'*(S)*h;

        scatterer1(mm,l) = (1/( 4 * sqrt(pi^3) )) * real ((sqrt((k)))*O);
        scatterer11(mm,l) = (1/( 4 * sqrt(pi^3) ))*(.5*g'*F*h + .5*h'*F'*g);
```

```
%               scatterer11(mm,l) =  real(O1);
        scatterer2(mm,l) = (1/(4 * sqrt(pi^3) )) * (.5*g'*F*h + .5*g.'*conj(F) *conj(h));
     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        end
%%
end
object  = rot90((((scatterer1))));
object1 = rot90(((real(scatterer11))));
object2 = rot90(((real(scatterer2))));
x       = xRrec;
y       = yRrec;
size(x);
%object = rot90(log10(abs(scatterer)));

% generate the image

%%  Scale object
object = real(object);

%%
% figure(30)
% imagesc(x, y,rot90(fliplr(object)))
% axis off



%% Plots of object
if 1==1
% figure(30)
% imagesc(x, y,abs(object))
% xlabel('x','FontSize',20)
% ylabel('y','FontSize',20)
% titstr = ['Holographic Backprojection Tomography: circular cylinder, SNR = ', num2str(SNRt), 'dB']
% title(titstr,'FontSize',16)

figure(30)
imagesc(x, y,(rot90((object))))
% axis off
% % generate the image
% figure(31)
% % imagesc(x, y,(object1)/max(max(object1)))
% imagesc(x, y,(object1))
%
% figure(32)
```

```
% imagesc(x, y,(object2))
% % generate slice of image
% figure(1)
% hold on
% slice_ind = gridsz/2+1:gridsz;
% xslice    = (object(:,50));
% plot(x,(xslice));
% hold off

end %%
%%
% xlabel('Meters')
% ylabel('Meters')
% title('Backprojection Tomography: B2 Bomber')
% size(object)

% save X x
% save Y y
% save ERcyl object

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%% Bibliography %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Michael Brandfass , Aaron D Lanterman and Karl FWarnick. "A comparison of the Colton
% Kirsch inverse scattering methods with linearized tomographic inverse scattering."
% INSTITUTE OF PHYSICS PUBLISHING. Inverse Problems 17 (2001) 1797-1816
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A.3 Diffraction Tomography

```
%                                        Thomas Sorensen
%                                    Diffraction Tomography
%
%   Computes an image of a scatterer given measured scattering amplitudes
%   and known source.  Assumes the known source is actually various line
%   sources positioned on a circle at equidistant angles.  This function allows a
%   user to pick a scatterer, the Er of the scatterer, the scale, the SNR, and the acuracy of the
%   forward code.
%
%   Input: scat_type = the type of scatterer to be used in generating the
```

```
%                               forward scattering amplitudes
%            Er = the dielectric constant of the scatterer.
%
%            SC = the scale of the scatterer
%            SNR = the signal to noise ratio(in dB)
%            Npionts = the number of points per wavelength: used in generating
%                            polyarc representation of PEC scatterer: 15
%                            results in suficient resolution
%
%   Output: object = The scatterer object function. Can be used to create image
%                                using the imagesc function
%
%               x ,y = the x and y values over which the object function is known.
%
%   Usage: [object,x,y] = Diffraction_Tomography('B2Bomber',1,xmax,ymax,.4,120,15)
%    a matrix object is created  that represents the object function.  It will be of a PEC
%    B2Bomber scaled to .4 the original.  The SNR will be 120 dB.  the max x and y values
%    of the scatterer are xmax and ymax respectively.  When generating the forward scatterering
%    amplitudes 15 points per wavelength will be used.
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                 Notes
% If PEC(Er=1) the forward scattering code is done by method of moments.
% If scatterer is dielectric then volume method of moments is used to
% calculate scattering amplitudes
%
% When PEC, Npoints can be large. A good value that provides accurate
% results is 15.  When dielectric, Npoints must be smaller.  Depending on the scatterer
% it should be around 7-8.  The larger the Npoints the more accurate but
% the longer it will take to calculate.  When this function is run, for
% Nfor will be printed to the terminal.  If it is above 1 million lower the
% Npoints.
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [object, x, y] = Diffraction_Tomography(scat_type,Er,SC,snr,f,saved,NTran);
% function DF
clear
if 1==1
scat_type = 'cylinder';
Er = 1;
SC = .5;
snr = 25;
saved = 2;
```

```
f = 300e6;
NTran = 64;
end
Npoints = 50;
% NTran    = 64;

%constants

Eo      = 8.854e-12;               % Electric permeability of free space
%Er       = 1;                            % Dielectric constant of scatterer
Escat   = 1 * Eo;                     %
MUo     = 4*pi*1e-7;              % Magnetic permeability of free space
Co      = 1/sqrt(MUo*Eo);        % Speed of light in free space
ETAo    = sqrt(MUo/Eo);          % Wave Impedence
Eta     = sqrt(MUo/Escat);
C       = 1/sqrt(MUo*Escat);
e       = exp(1);
GAMMA   = 1.781;
E       = 1.0;                            % E field amplitude
%f       = 300e6;
w       = 2*pi*f;
k       =  w/C;
wlength = C/f;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Read in Scattering Amplitudes %%%%%%%%s
%[S,sx,sy,phi_i,phi_s,j1] = Dr_Warnick_Forward(13,0)
if(Er == 1)

    if(saved == 1)

        if (strcmp(scat_type, 'B2Bomber'))
                 load ScatAmpB2
%                   S = awgn(S,snr,'measured');
                 % S = awgn(S,snr);
% my own noise
%                   snr1 = 10^(snr);
%                   [ssr ssc] = size(S);
%                   Sigrms = (norm(S,'fro'));
%                   alpha = sqrt(Sigrms^2/snr1);
%                   S = S + alpha*randn(ssr);
%
        end
```

```
        if (strcmp(scat_type, 'square'))
               load ScatAmpSq
        end

        if (strcmp(scat_type, 'triangle'))
                load ScatAmpTri
        end


        if (strcmp(scat_type, 'f15'))
                 load ScatAmpf15;
%                  S = awgn(S,snr,'measured');

        end

        if (strcmp(scat_type, 'cylinder'))
                  load ScatAmpCylAn;
%                   if NTran==64
%                   load ScatAmpCylAn64;
%                   end
%                   if NTran==100
%                   load ScatAmpCylAn100;
%                   end
%                   if NTran==32
%                   load ScatAmpCylAn32;
%                   end
                  S = San;
%                   S2 = awgn(S,snr,'measured');
%                   S2 = awgn(S,snr);

% my own noise
%                   snr1 = 10^(snr);
%                  [ssr ssc] = size(S);
%
%                  Sigrms = (norm(S.^2,'fro'));
%                  alpha = sqrt(Sigrms/snr1);
%                  noise = randn(ssr);
%                  S1 = S + alpha*noise;
%                  S = S1;

        end

         if (strcmp(scat_type, 'cresent'))
                  load ScatAmpCres;
```

```
%                   S = awgn(S,snr,'measured');
                  % S = awgn(S,snr);
% my own noise
%                   snr1 = 10^(snr);
%                   [ssr ssc] = size(S);
%                   Sigrms = (norm(S,'fro'));
%                   alpha = sqrt(Sigrms^2/snr1);
%                   noise = randn(ssr);
%                   S = S + alpha*noise;


         end

            if (strcmp(scat_type, 'C'))
                  load ScatAmpC;

%                   S = awgn(S,snr,'measured');
                  % S = awgn(S,snr);
% my own noise
%                   snr1 = 10^(snr);
%                   [ssr ssc] = size(S);
%                   Sigrms = (norm(S,'fro'));
%                   alpha = sqrt(Sigrms^2/snr1);
%                   noise = randn(ssr);
%                   S = S + alpha*noise;


         end



    else

    %[S,sx,sy,phi_i,phi_s,j1] = Dr_Warnick_Forward(13,0)
%     [S] = ScatAmp(15,scat_type,SC,snr,f); % gives scattering amp matrix - input type of scatterer
    [Es_X San_X S] = ScatAmp( Npoints, scat_type,SC,snr,f,NTran);
    end


else

    if(saved == 1)
```

```
        if (strcmp(scat_type, 'B2Bomber'))
                load ScatAmpB2DE2
%                 S = awgn(S,snr,'measured');


        end

        if (strcmp(scat_type, 'f15'))
                load ScatAmpf15DE;
%                 S = awgn(S,snr,'measured');

        end

        if (strcmp(scat_type, 'cylinder'))
                 load ScatAmpCylDE2;
%                  S = awgn(S,snr,'measured');
        end

        if (strcmp(scat_type, 'C'))
                load ScatAmpCDE2;
%                 S = awgn(S,snr,'measured');
        end


    else
    Er = Er

    [S] = VolumeMOM(Npoints,scat_type,Er,SC);
    end

end
% Bypass above forward codes
if (1==2)
    S =Dr_Warnick_Forward(11,0)
end
%%%%%%%%%%%%%%%%%%%%
% overide the Scattering Amplitudes with save SA matrix
if(1==3)
    load CylDE;
end
% overide the Scattering Amplitudes by calling Dr. Warnick's forward code
if(1==3)
    [S,sx,sy,phi_i,phi_s,j1] =Dr_Warnick_Forward(11,0)
end
% add noise to S
```

```
if(1==1)

%%%% Correct Noise %%%%%%%%%%%%
SNRt = snr;
snr     = 10.^(snr/10);
Stest = S;
if 1==1 % add noise matrix

  display('Noise Matrix')
  sp      = sum(abs(S(:)).^2)/length(S(:));
%   sp      = norm(S,'fro')/sqrt(length(S(:)))
%     sp      = sum(abs(S(:)).^2)/NTran;
  sn      = sqrt(sp/(2*snr));
  Snoise  = sn.*(randn(size(S)) + i*randn(size(S)));
  S       = S + Snoise;

  snr     = sqrt(sp/snr);

end

if 1==2

  display('Mu Identity')
  sp      = sum(abs(S(:)).^2)/length(S(:));
  mu      = sqrt(sp/(snr))*sqrt(2*NTran);
  Snoise  = mu*eye(size(S));
  S       = S + Snoise;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end % add noise

[sr sc] = size(S);
% array and matix initializations
Phi   = linspace(0,2*pi,NTran);
Nphi  = length(Phi);
krtx  = zeros(Nphi,Nphi);
krty  = zeros(Nphi,Nphi);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%      Set up grids: K space and Cartesian   %%%%%%%%%%%%%%%%%%%

% first for loop will cycle transmit angles
```

```
% second for loop will cycle scatter angles
% these loops define the trasnmit and receive points in k

% calculate the kr - kt which is the space upon which o(the scatterer volume)
% will be known

for m = 1:Nphi % transmit angles

    for n = 1:Nphi % recieve angles

        krtx(m,n) = k*cos(Phi(n)) + k*cos(Phi(m));
        krty(m,n) = k*sin(Phi(n)) + k*sin(Phi(m));

    end

end

%plot the k space grid

% figure(111223)
% plot(krtx , krty,'.')
% title('Kr - Kt')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% set up the cartesian space grid
kmax = 2*k;

Nx = 64;
Ny = Nx;
dkx = 2*kmax/Nx;
dky = dkx;
kx    = (-kmax + dkx/2):dkx:(kmax-dkx/2);
ky    = (-kmax + dky/2):dkx:(kmax-dky/2);
Or = zeros(Nx,Ny);

[kX kY] = meshgrid(kx,ky);

NX = length(kX);
NY = length(kY);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%  find the scatter   %%%%%%%%%%

% Refering to eq. 3.76 from Dr. Warnicks CEM notes, the scatter field O can
% be found from the scattering amplitude from the function call to ScatAmp
```

```
%
% The incident field magnitude from a line source used to compute the scattered field
%  is found on pg 49 of Dr. Jensens notes, neglect the plane wave
%
% There will be no dependence in the far field on the transmit and recieve
% points.

O = S*4*i;
% O = krtx.^2 +krty.^2;
%O  = cos(krtx);

% figure(3)
% plot3(krtx,krty,real(O),'.')
% title('bigO')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%interpolate the kr-kt to the kx, y grid:
for m = 1:Nx

    for n = 1:Ny

        kxm = kx(m);
        kyn = ky(n);
        dist2xy = (kxm - krtx).^2 + (kyn - krty).^2;

                [v1,i1] = min(dist2xy);
                [v2,i2] = min(v1);
                val = v2;
                ind(1) = i1(i2);    % j
                ind(2) = i2;        % d
                Or(m , n) = O(ind(1) , ind(2));

    end

end

o = ifft2(Or);
ob = (fftshift(o));

[sco sro] = size(o);

dx = 2*pi/(2*kmax);
dy = 2*pi/(2*kmax);

xmax = dx*(sro/2);
ymax = dy*(sco/2);
```

```
x    = (-xmax+dx/2):dx:(xmax-dx/2);
y    = (-ymax + dy/2):dy:(ymax-dy/2);

Nx = length(x);
Ny = length(y);

for n = 1:Nx

    for m = 1:Ny

%          o(n,m) = o(n,m)*dx*exp(-i*2*pi*(n-1)/(xmax+xmax))*exp(i*dx*k/2);
       o(n,m) = o(n,m)*exp(-i*2*pi*(n-1)/(2*xmax))*exp(i*dx*k/2);
    end

end

[X Y] = meshgrid(x,y);

% figure(6)
% plot3(X,Y,abs(fftshift(o)),'.')
% title('scaled')

object = ((abs(fftshift(o))));



% plot the scatterer
% figure(2666)
% imagesc(x,y ,rot90(object));
% xlabel('x','FontSize',20)
% ylabel('y','FontSize',20)
% titstr = ['Diffraction Tomography: circular cylinder, SNR =', num2str(SNRt),'dB' ]
% title(titstr,'FontSize',16)
%
% figure(2666)
% imagesc(x,y ,rot90(object));
% axis([-6 6 -6 6])
% axis off




% xlabel('meters')
% ylabel('meters')
% title('Diffraction Tomography: cylinder')

% set(get(1, 'CurrentAxes'),'ydir','normal')
```

```
% %% plot slice
% slice = object(32,:);
% figure(90)
% plot(x, slice)



% save ERcyl object
% save X x
% save Y y
```

A.4 Regularized Sampling

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                Thomas Sorensen
%                      Inverse Scatter Regularized Sampling
%
% Takes Scattering amplitude from an unknown scatterer and uses the
% regularized sampling method to unveil the scatterer.
%
% This function is designed to be used in conjunction with forward
% scattering code, Image contrast functions.  To produce an image of a
% scatterer when all that is known and available use InverseScat.m
%
%   Input: scat_type = type of scatterer
%             Er = dielectric permiability
%             xmax , ymax = the max x an y dimensions of the scatterer
%             SC = the scale of the scatterer: use to scale the scatterer
%             down inorder for the image to be created using only 64
%             recievers
%             snr = signal to noise ratio
%             saved = used saved scattering amps or generate new ones
%                           1: uses saved
%                           2: generates new ones
%             f = frequency of incident field
%
%   output: object = matrix of scatterer data that will produce image
%                           of scatterer, plot with imagesc
%               x,y = x and y values on which the scatterer is know and will be plotted
%
%   usage: [object,x,y] = RegSamp('B2Bomber,1,3,3,.4,120,1,300e6)
%             image data over x and y will be produced for a PEC B2Bomber
```

```
%              with max x and y values of 3, scaled down to .4, snr of 120,
%             using saved scattering amplitude data, at a frequency of 300e6;
%
%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% function [eigMy eigDr eigSan object x y] = RegSamp(scat_type,Er,xmax,ymax,SC,snr,saved,f);
function [ object x y] = RegSampTinkReg(scat_type,Er,xmax,ymax,SC,snr,saved,f,reg,NTran);

if 1==1
clear
Er        = 1;
xmax      = 1;
ymax      = 1;
scat_type = 'cylinder';
SC        = .5;
snr       = 5;
saved     = 1;
f         = 300e6;
reg       = 10e-8;
Npoints   = 15;
N         = 100
NTran     = N;
upperbound = 10e1;
end
% NTran     = N;
snrtitle = snr;

%% --- Regularization Method
% 1 = Tikhonov Fixed Reg Parameter
% 2 = Tikhonov with Morozovs Discrepency Rule

regmeth   = 2;

%% --- Bounds for zero find function

ub = 10^14;
lb = 0;


%% --- load scattering amplitudes or call the scattering amplitude function to
% generate the amplitudes for given scatterer and freq
if(Er == 1)
```

```
    if(saved == 1)

        if (strcmp(scat_type, 'B2Bomber'))
                 load ScatAmpB2

        end

        if (strcmp(scat_type, 'square'))
               load ScatAmpSq
        end

        if (strcmp(scat_type, 'triangle'))
                load ScatAmpTri
        end

        if (strcmp(scat_type, 'f15'))
                 load ScatAmpf15;
        end

        if (strcmp(scat_type, 'cylinder'))
                  load ScatAmpCylAn;
% %                   if NTran==64
% %                   load ScatAmpCylAn64;
% %                   end
% %                   if NTran==100
% %                   load ScatAmpCylAn100;
% %                   end
% %                   if NTran==32
% %                   load ScatAmpCylAn32;
% %                   end
                  S = San;

        end

        if (strcmp(scat_type, 'cresent'))
                  load ScatAmpCres;

        end

        if (strcmp(scat_type, 'C'))
                  load ScatAmpC;
        end

        if (strcmp(scat_type, 'vfy218'))
```

```
                load ScatAmpfVF;
        end
    else
    jkl = 11
    [Es San S] = ScatAmp(Npoints,scat_type,SC,snr,f,NTran);
    eigMy = eig(S);
    MyS   = S;
    end

else

    if(saved == 1)

        if (strcmp(scat_type, 'B2Bomber'))
                load ScatAmpB2DE2
        end

        if (strcmp(scat_type, 'f15'))
                load ScatAmpf15DE;
        end

        if (strcmp(scat_type, 'cylinder'))
                 load ScatAmpCylDE2;
        end

        if (strcmp(scat_type, 'C'))
                 load ScatAmpCDE2;
        end
    else

    Er = Er
    [S] = VolumeMOM(Npoints,scat_type,Er,SC);

    end

end

%% ---Dr Warnick Forward Code

if (1==2)

 S = Dr_Warnick_Forward(1,0,f,NTran,NTran,0);
%  eigDr = eig(S);
 DrS   = S;
```

```
end

%% --- Load in Scattering Amplitudes

if(1==2) % Load scattering amplitudes

 sa      = load('scat1_sa.dat');
 F       = sa(:,3)+j*sa(:,4);
 PhiScat = sa(1:64,2)*pi/180;
 NA      = sqrt(length(F));
 F       = reshape(F,NA,NA)
 S       = F;

end

%% --- Analytical Scattering Amplitudes
if 1==2

 [eigSan S] = AnalyticalScatAmpsPECCyl(NTran,f,SC);
 AnS = S;
 save ScatAmpCyl S
end

if 1==2
  load ScatAmpCylAn
  S = San;
end

%% --- Numerical noise level

% Serror = San - S;
% delta = norm(Serror(:))%12106216426505
% Delta  = var(Serror(:));
delta = .12106216426505;

%% --- Constants

Eo        = 8.854e-12;              % Electric permeability of free space
Escat     = Er * Eo;
MUo       = 4*pi*1e-7;              % Magnetic permeability of free space
Co        = 1/sqrt(MUo*Eo);      % Speed of light in free space
ETAo      = sqrt(MUo/Eo);        % Wave Impedence
Eta       = sqrt(MUo/Escat);
C         = 1/sqrt(MUo*Escat);
e         = exp(1);
```

```
GAMMA     = 1.781;
E         = 1.0;                          % E field amplitude
%f            = 300e6;
w         = 2*pi*f;
k         =  w/C;
wlength   = C/f;

%% --- Reconstruction Grig and Incident/Scattering Angles

Gridsz    = 100;
[ssr ssc] = size(S);
dPhi      = 2*pi/(ssr);
PhiScat   = 0: dPhi : 2*pi-dPhi;
Ns        = length(PhiScat);
xmax      = xmax+1;
ymax      = ymax +1;
dx        = 2*xmax/(Gridsz-1);
dy        = 2*ymax/(Gridsz-1);  % the dx and dy are not the same. This might cause problems???
x         = -xmax: dx : xmax;
y         = -ymax: dy : ymax;
Nx        = length(x);
Ny        = length(y);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                     Tikhonov Regularization
%%%                      Fixed Reg Parameter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if regmeth==1
  regmeth
%% --- Add Noise Noise Matrix
%
snr     = 10.^(snr/10);

if 1==2 % add noise matrix

  display('Noise Matrix')
  sp      = sum(abs(S(:)).^2)/length(S(:))-delta;
  sn      = sqrt(sp/snr);
  Snoise  = sn.*(randn(size(S)) + i*randn(size(S)));
%   S       = awgn(S,snr,'measured');
  F       = S + Snoise;
end

if 1==1
```

```
  display('Mu Identity')
  sp      = sum(abs(S(:)).^2)/length(S(:))-delta;
  mu      = sqrt(sp/snr)*sqrt(2*NTran);
  Snoise  = mu*eye(size(S));
  F       = S + Snoise;

end


%% --- find the regularization parameter
% sp      = sum(abs(S(:)).^2)/length(S(:))-Delta;
mu      = sqrt(sp/snr)*sqrt(2*NTran);
epsilon = 10e-5;

%% --- regularize the Scattering Matrix

Fmu  = S + mu*eye(size(S));
Fer   = S + epsilon*eye(size(S));


%% --- Kirsh Factorization method

KF           = (S'*S).^.25 + epsilon*eye(size(S));

%% --- A*A
FsF       = F'*F + epsilon*eye(size(S));;
Fonly     = F + epsilon*eye(size(S));
FersFer   = Fer'*Fer;
%% --- decompose scattering amplitude matrix

[L, U, P]                      = lu(Fonly);
% [LFer, UFer, PFer]             = lu(Fer);
% [LFersFer, UFersFer, PFersFer] = lu(FersFer);
% [LsF, UsF, PsF]                = lu(FsF);

% [LKF, UKF, PKF]                = lu(KF);

%% --- solve g for range of points on grid epsilon regularization

for m = 1:Nx

    for n = 1:Ny

        kx     = k*cos(PhiScat);
```

```
        ky     = k*sin(PhiScat);
        xi     = x(m);
        yi     = y(n);
        rho    = sqrt(xi^2+yi^2);
        phirho = atan2(yi,xi);
        dS     = F'*exp(-i*k*rho*cos(PhiScat - phirho)).';
%         d      = S'*(E*exp(i*(kx*xi + ky*yi))).';
        d      = exp(-i*k*rho*cos(PhiScat - phirho)).';

        gF       = U\(L\(P*(d)));
%         gFer     = UFer\(LFer\(PFer*(d)));
%         gFsF     = UsF\(LsF\(PsF*(dS)));
%         gFersFer = UFersFer\(LFersFer\(PFersFer*(dS)));

        AF(m,n)       = 1/norm(gF);
%         AFer(m,n)     = 1/norm(gFer);
%         AFsF(m,n)     = 1/norm(gFsF);
%         AFersFer(m,n) = 1/norm(gFersFer);


    end

end

object = AF;
% object0  = AFsF;
% object1 = AFer;
% object3 = AFersFer;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%              Tikhonov/Morozov regularization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if regmeth == 2

snr     = 10.^(snr/10);

if 1==1 % add noise matrix

  display('Noise Matrix')
  sp      = sum(abs(S(:)).^2)/length(S(:))-delta;
  sn      = sqrt(sp/snr);
  Snoise  = sn.*(randn(size(S)) + i*randn(size(S)));
```

```
%   S       = awgn(S,snr,'measured');
  S       = S + Snoise;
end
%   mu      = sqrt(sp/snr)*sqrt(2*NTran);
if 1==2

  display('Mu Identity')
  sp      = sum(abs(S(:)).^2)/length(S(:))-delta;
  mu      = sqrt(sp/snr)*sqrt(2*NTran);
  Snoise  = mu*eye(size(S));
  S       = S + Snoise;

end

%% --- Noise level
delta = delta
delta2   = norm(Snoise,'fro');
% delta2 = 0;
if delta2>delta
  delta = delta2;
end

%% --- Singular Values

[U Ssv V ]      = svd(S);
Sn              = diag(Ssv);

  for m = 1:Nx

      for n = 1:Ny

        kx     = k*cos(PhiScat);
        ky     = k*sin(PhiScat);
        xi     = x(m);
        yi     = y(n);
        rho    = sqrt(xi^2+yi^2);
        phirho = atan2(yi,xi);
        d      = exp(-i*k*rho*cos(PhiScat - phirho)).';
        un     = abs(U'*d).^2;
        vn     = abs(V'*d).^2;
        regalphaCK(m,n)  = fzero(@(x)sum(un.*(delta^2*Sn.^2 - x^2)...
                           ./(x +Sn.^2).^2),[lb,ub]);
%
%         regalphaCKF(m,n) = fzero(@(x)sum(vn.*(delta^2*Sn - x^2)...
%                             ./(x +Sn.^2).^2),[lb,ub]);
```

```
%          regalphaCK(m,n) = 10e-5;
         regalphaCKF(m,n) = 0;
         normgCK  = sum(un.*Sn.^2./(regalphaCK(m,n) + Sn.^2).^2);
         normgCKF = sum(vn.*Sn./(regalphaCKF(m,n) + Sn).^2);

         A(m,n)  = 1/normgCK;
         AF(m,n) = 1/normgCKF;

       end

  end

object  = (((((A)))));
object1 = AF;
object2 = regalphaCK;
object3 = regalphaCKF;

end

%%

%% Create Images

% clims  = [0 .0015];
%
% figure(12)
% clf
% img = imagesc(x,y,(rot90(object)));
% imagesc(x, y,(object))
% xlabel('x','FontSize',20)
% ylabel('y','FontSize',20)
% titstr = ['Linear Sampling: circular cylinder, Noise Free']
% % titstr = ['Linear Sampling: circular cylinder, SNR = ', num2str(snrtitle), 'dB' ]
% title(titstr,'FontSize',16)
%%
% figure(12)
% clf
% img = imagesc(x,y,(rot90(object)));
% axis off

% figure(13)
% clf
% img = imagesc(x,y,((object2)));
```

```
% title(snrtitle)
% xlabel(snrtitle)
% figure(13)
% [X Y]  = meshgrid(x,y);
% surf(X,Y,object)

% figure(21)
% clf
% imagesc(x,y,((object1)));
% title('CK-RSF')
%
%
%  figure(22)
% clf
% imagesc(x,y,((object2)));
% title('CK-RS Reg Parameter')
%
%  figure(23)
% clf
% imagesc(x,y,(abs(object3)));
% title('CK-RSF Reg parameter')

%
%  figure(13)
%  imagesc(x,y,abs(A1))
% img = imagesc(x,y,log10(abs(object)))
%
% %  hold on
% %  rad = .5;
% %  x = rad*cos(PhiScat);
% %  y = rad*sin(PhiScat);
%
% %  plot(x,y)
% %  hold off
% xlabel('x')
% ylabel('y')
% title('PEC Cylinder')
% hold off

%% Image Cut

% %  cutnum = object(32,:);
% %  figure(65)
% %  hold on
% %  plot(x,abs(cutnum),'--')
```

```
% % legend('Eigenvalue Analytical','Numerical')
% % xlabel('x')
% % xlabel('Pixel Value')
% % title('CK-RS comparison of analytical and numerical solutions for Tikhonov regularization')
% % grid on
% %   hold on

%% blah

% xlabel('Meters')
% ylabel('Meters')
% title('Regularized Sampling: B2 Bomber')

% imageC = get(img,'CData');
% level = graythresh(imageC)
% BW = im2bw(imageC,level);
%
% figure(98723)
% imshow(BW,'XData',x,'YData',y)

% set(get(1, 'CurrentAxes'),'ydir','normal')


%% Comparisons of Scattering Amplitudes
% % dPhi        = 2*pi/(NTran-1);
% % dPhi1        = 2*pi/(NTran);
% % PhiScat     = 0: dPhi : 2*pi;
% % PhiScat1     = 0: dPhi1 : 2*pi-dPhi;
% %
% % figure(2222)
% % in = 1;
% % semilogy(PhiScat, imag(MyS(in,:)),'-ob')
% % hold on
% % semilogy(PhiScat, imag(DrS(in,:)),'-.r')
% % hold on
% % semilogy(PhiScat, imag(AnS(in,:)),'-hg')
% % hold off
% % legend('Analytical Analytical','Numerical My','Numerical Dr','Numerical Analytical')
% %
% % figure(2223)
% % in = 1;
% % plot(PhiScat, real(MyS(in,:)),'-ob')
% % hold on
% % plot(PhiScat, real(DrS(in,:)),'-.r')
% % hold on
```

```
% % plot(PhiScat, real(AnS(in,:)),'-hg')
% % hold off
% % legend('Analytical Analytical','Numerical My','Numerical Dr','Numerical Analytical')
% %
```

A.5 Contrast Ratio

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                  Thomas Sorensen
%                                Image Contrast Ratio
%
% This function will be used to compute the contrast ratio of an image
% generated by inverse scatterering to the actual scatterer.
%
% This image will call  a inverse scattering method function to get the
% image data
%
% It also relies on the polyarc mesh generation to determine the
% scatterer polygon vectors which will be used with the inpoly function
% to determine points inside the scatterer.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ x slice middlevalue CR] = ImageContrast(scat_type,Er,SC,snr,f,saved,method,NTran);
% function test
% Call PolyMeshScat to get the polyarc representation of the scatterer
% defined by scat_type.
% %
% Er        = 1;
% scat_type = 'cylinder';
% SC        = .5;
% snr       = 180;
% f         = 300e6;
% saved     = 1;
% method    = 1;
% reg       = 10e-8;
fig       = 411;
Npoints   = 50;

% Constants ------------------------------------------------
Eo      = 8.854e-12;               % Electric permeability of free space
%Er       = 1;                           % Dielectric constant of scatterer
Escat   = Er * Eo;                    %
MUo     = 4*pi*1e-7;              % Magnetic permeability of free space
```

```
Co      = 1/sqrt(MUo*Eo);        % Speed of light in free space
ETAo    = sqrt(MUo/Eo);          % Wave Impedence
Eta     = sqrt(MUo/Escat);
C       = 1/sqrt(MUo*Escat);
e       = exp(1);
GAMMA   = 1.781;
E       = 1.0;                           % E field amplitude
%f       = 300e6;
w       = 2*pi*f;
k       =  w/C;
wlength = C/f;
wl      = (.5*wlength);
%-------------------------------------------------------
% Get points of scatterer
scat_type;
[polyarc xv yv] = PolygonScat(scat_type,SC,Npoints);
xv = xv;
yv  = -yv;
xv1 = xv;
yv1 = yv;


%___Gets points of scatterer that is .5 wavelengths larger than the
%original_____________________________________________________
if strcmp(scat_type,'clinder')

    xv = xv * (1+wl);
    yv = yv  * (1+wl);

end

if strcmp(scat_type,'crresent')



end

if strcmp( scat_type,'CC')

    C = SC *[-2-wl,-3-wl;  -2-wl,3+wl;  2+wl,3+wl;...
        2+wl,1-wl; 0+wl,1-wl; 0+wl,-1+wl; 2+wl,-1+wl; 2+wl,-3-wl; -2-wl,-3-wl];
    xv = C(:,1);
    yv = C(:,2);

end
```

```
%if(1==2)
if strcmp( scat_type,'B2BBomber')

    B2 =     [ 0,    3.0629+wl;
    7.7500+1.6*wl,   -2.3204;
    6.4970,   -3.1093-wl;
    4.0374,   -1.2994-wl;
    2.0883,   -2.7380-wl;
    1.2066,   -2.1811-wl;
              0,   -2.9237-wl;
   -1.2066,   -2.1811-wl;
   -2.0883,   -2.7380-wl;
   -4.0374,   -1.2994-wl;
   -6.4970,   -3.1093-wl;
   -7.7500-1.6*wl,   -2.3204;
               0,    3.0629+wl];
          B2 = B2*.4;
    xv = B2(:,1);
    yv = B2(:,2);

end


 %___find max x value and max y value of scatterer____________

xmax = max(abs(polyarc(:,1)));
ymax = max(abs(polyarc(:,2)));

xmax = max([xmax ymax]);
ymax = xmax;

%___call inverse scattering method to get image data__________

if (method == 1)
inverse_method =   'Reg_Samp';
end

if (method == 2)
inverse_method =   'Diffract_Tomo';
end

if (method == 3)
inverse_method =   'Brandfass';
end
```

```
if (method == 4)
inverse_method =   'RSwoNoise';
end

if (method == 5)
inverse_method =   'RSwNoise';
end

if (method == 6)
inverse_method =   'RSDiscrete';
end

switch inverse_method

    case 'Reg_Samp'


        if (1==1)
           reg = 10e-5;
%         [object x y] = RSEigFormulation(scat_type,Er,xmax,ymax,SC,snr,saved,f,reg);
        [object x y] = RegSampTinkReg(scat_type,Er,xmax,ymax,SC,snr,saved,f,reg,NTran);
%         [g2 object,x,y] = RegSamp(scat_type,Er,xmax,ymax,SC,snr,saved,f);
%
       % use the analytical case for a cylinder
        else

%             [object,x,y] = RegSampAnalytical(xmax,ymax,snr);

       end


    case 'Diffract_Tomo'

        [object, x, y] = Diffraction_Tomography(scat_type,Er,SC,snr,f,saved,NTran);

    case 'Brandfass'
         [object,x , y] = BrandfassTomo(scat_type,Er,xmax,ymax,SC,snr,saved,f,NTran);

    case 'RSwoNoise'

        [nada object x y] = RegSampAnalytical(1,1,snr);

    case 'RSwNoise'

        [object x y] = RegSampAnalyNoise(1,1,snr);
```

```
    case 'RSDiscrete'

        [object x y] = RegSampAnalyticaNoise(snr,xmax);

end

%___find points inside the scatterer_____________________________________
% the points xv yv come from the inverse scattering method and are the points
% corresponding to points of the scatterer
% variables in caps are for the points outside the scatterer, ones is lower case are
% for the points inside the scatterer

[X Y] = meshgrid(x,y);

% figure(12345)
% plot(X,Y,'.')
% title('X Y points')

% produce the image of the object and the scatterer outline
% %

% figure(fig)
% imagesc(x,y,rot90(object));
% hold on
% plot(xv,yv,'-w')
% hold off
% figure(999)
% plot(xv1,yv1,'-b')
% hold off

% plot(xv1,yv1,'-b')



% built in function to get points inside polygon
[in on] = inpolygon(X,Y,xv1,yv1);

% find indicies of points inside and on scatterer
[j,a]  = find(in);
[l,m] = find(on);
j = [j l];
a = [a m];

% get values of image at those points
inind    = sub2ind(size(object),j,a);
```

```
inval    = object(inind);
% inval    = diag(inval);

% onval   = object(l,m);
% onval   = diag(onval);

%% --- plot values inside(debug)
object = rot90(object);
if 1==2
% clim = [0 .025]
% object(inind) = 100;
figure(fig)
set(gca,'fontsize',16)
imagesc(x,y,log10(abs(object)));
hold on
plot(xv,yv,'-w')
hold off
xlabel('x (m)')
ylabel('y (m)')
% xlim([ -3 3 ])
% ylim([-3 3 ])
title('CK-LS for a B2')

end

if 1==2

    figure(fig)
    set(gca,'fontsize',20)
    plot(xv,-yv)
    xlabel('x (m)')
    ylabel('y (m)')
    xlim([ -2 2 ])
    ylim([-2 2 ])
    title('C Shape')

end
% hold on
% plot(xv,yv,'-w')
% hold off

%% --- Norm in over Norm out Contrast Ratio
if 1==1

% pick off values inside
```

```
imgval =  inval; %cat(1,inval,onval);

if (method == 6)
   imgval = sum(imgval)/length(imgval);
else
   imgvalold = sum(abs(imgval).^2)/length(imgval);
   nor  = 'norm'
%    imgval = norm(inval)/length(imgval);
   imgval = norm( inval );
   imgval = sqrt( imgvalold );
end
% get the x and y points of the points in and on the scatterer

xin = diag(X(j,a));
yin = diag(Y(j,a));

% xon = diag(X(1,m));
% yon = diag(Y(1,m));

% plot(xin,yin,'.w',xon,yon,'.w')
%  hold off
% plot points in and on the scatterer


% figure(fig)
% imagesc(x,(y),(object));
% hold on
% plot(xin,yin,'.')
% title('points in and on scatterer')
% hold off
% find the points outside the scatterer and get the image values at those
% points
%

% all the points outside the scatter

if (1==1)

[sro sco] = size(object);

XSCAT = ones(sro,sco);

OutScat = XSCAT - in ;

[L,M] = find(OutScat);
```

```
OUTdiag = sub2ind([sro sco],L,M);

IMGVAL    = object(OUTdiag);
XOUT      = X(OUTdiag);
YOUT      = Y(OUTdiag);

end

% Use all points

if (1==2)

    IMGVAL = object;

end

if (method == 6)
    IMGVAL = (sum(IMGVAL(:)))/length(IMGVAL(:));
else
    IMGVALold = sum(abs(IMGVAL(:)).^2)/length(IMGVAL(:));
    nor = 'norm'
%     IMGVAL = norm(IMGVAL)/length(IMGVAL);
      IMGVAL = norm(IMGVAL(:));
%     IMGVAL = max(IMGVAL);
      IMGVAL = sqrt(IMGVALold);
end

CR = (imgval/IMGVAL);
%cr = IMGVAL/imgval;

end %% Norm CR

%% Min max CR
if 1==2 %% Min max CR

xcut = x(x>=0);
ycut = zeros(1,length(xcut));

[in] = inpolygon(x,y,xv1,yv1);

in      = find(in==1);
edgeind = max(in);
outind  = edgeind + 1;
```

```
CR      = object(edgeind,32)/object(outind,32);

end %% Min Max CR


%% --- edge value
% [N N] = size(object);
% middlevalue = object(round(N/2),round(N/2));
middlevalue = 5;%object(28,36);

%% -- slice through image
[osx osy] = size(object);
slice = abs(object(floor(osx/2),floor(osy/2):osy));
```

A.6 Monte Carlo Simulations

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                       Thomas Sorensen
%                                   Monte Carlo CR vs SNR
%
% Uses Monte Carlo Methods to calculate the CR vs SNR.
%
% At each SNR level 20 simulations are run and the average CR is taken
% for that SNR.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
% function ImageContrastMonte
Er        = 1;
% scat_type = 'B2Bomber';
scat_type = 'cylinder';
% scat_type = 'f15';
% scat_type = 'square';
% scat_type = 'triangle';
% NTran     = 100;
SC        = .5;
f         = 300e6;
saved     = 1;

%___choose method of inverse scattering__________________________
% 1 = Regularized Sampling
% 2 = Diffraction Tomography
% 3 = Brandfass Tomography

method  = 3;
```

```
fig     = 4798;

%___set up an SNR array_____________________________________________
%

N = [16 32 64 128 192 260 324];%16:12:256;

for mm = 1:length(N)

NTran = N(mm)
snrmax = 50;
snrmin = -35;
Nsnr   = 20;
dsnr   = (snrmax-snrmin)/(Nsnr-1);
snr    = zeros(Nsnr);
snr    = snrmin: dsnr: snrmax;

% snr = 10.^(-10:1:15);
Nsnr = length(snr);

CRMean = zeros(1,Nsnr);
CRStD  = zeros(1,Nsnr);
count  = 0;
% snr    = 100;

% dNTran = (192-8)/(Nsnr-1);
% NTran  = 8:dNTran:192;


% SC = linspace(.1,7,Nsnr);
% SC = .4;
%
% df  = 8/(Nsnr-1);
% f   = 10.^(0:df:10);
% Eo  = 8.854e-12;
% MUo = 4*pi*1e-7;             % Magnetic permeability of free space
% Co  = 1/sqrt(MUo*Eo);
% w   = 2*pi*f;
% k   =  w/Co;


%% Initialize arrays
middlevalue = ones(1,NTran)
CR = ones(1,NTran)
```

```
count = 0;
%___for each SNR calculate 25 CRs and find their mean and std____
for m = 1:Nsnr
    if m==1
     Debug =  NTran
[eigSan San] =  AnalyticalScatAmpsPECCyl(NTran,f,SC);
    end
    for  n = 1:20
tic
% %         if n == 1
% %           saved = 1;
% %         else
% %            saved =1;
% %         end
      saved  = 1;
      SNRind = snr(m)
%         N = (NTran);
%        [t1 t2 t3 t4] = ImageContrast(scat_type,Er,SC,snr(m), f,saved,method,NTran);
    [x slice(n,:) middlevalue(n) CR(n)]=ImageContrast(scat_type,Er,SC,snr(m),f,saved,method,NTran);
      count  = count + 1
toc
    end
   Slicemean(m,:) = mean(slice) ;
   CRMean(m)   = mean(CR);
   CRStD(m)    = std(CR);
   mvMean(m)   = mean(middlevalue);

end

%___plot the CR mean ____________________________________

%%
if 1==1



pt = ['.' '.-' '-.' '-p' '-s' '-x' '-d' ];
figure(fig)
hold on
snrlin = 10.^(snr/10)
semilogy(snr,CRMean,pt(mm))
% semilogy(NTran,CRMean,'-')
xlabel('SNR')
% ylabel('Contrast Ratio Mean')
% title('Diffraction Tomography: PEC circular cylinder')
title('Holographic Backprojection Tomography: PEC circular cylinder')
```

```
% title('CK-RS Tik/Moro: PEC circular cylinder')
% title('CK-LS Tik\Moro: PEC square cylinder')
grid on
% xlabel('Number of T_x/R_x')
ylabel('Contrast Ratio Mean')
% axis ([-40 50 10^-.1 10^.7])
hold off
end

end

%% plot the slice
%
% figure(fig+10)
% plot(x,Slicemean)
%

%%

if 1==2

 figure(fig)
 semilogy(SC,CRMean,'-')

end

if 1==2

 figure(fig)
 loglog(f,CRMean,'-')

end

%% plot edge value
% % figure(1234)
% % snrlin = 10.^(snr/10)
% % semilogx(snrlin,(mvMean))

%%
figure(fig)
% legend(['N= ' 'N(1)'])
legend(['N=  N(1)'], ['N= ' 'N(2)'], ['N= ''N(3)'], ['N= ''N(4)'],['N=''N(5)'],['N= ''N(6)']...
  ,['N= ''N(7)']);
```

```
%%
%___plot the CR std ____________________________________________
% figure(66)
% plot(snr,CRStD)
% xlabel('SNR')
% ylabel('Contrast Ratio Standard Deviation')
% title('Contrast Ratio Standard Deviation vs SNR')
```

Bibliography

[1] D. Colton and A. Kirsch, "A simple method for solving inverse scattering problems in the resonance region," *Inverse Problems*, vol. 12, pp. 383–393, 1996. 1, 12

[2] D. Colton, M. Piana, and R. Potthast, "A simple method using Morozov's discrepancy principle for solving inverse scattering problems," *Inverse Problems*, vol. 13, pp. 1477–1493, 1997. 1, 29

[3] D. Colton, K. Giebermann, and P. Monk, "A regularized sampling method for solving three dimensional inverse scattering problems," *SIAM J. Sci. Comp.*, vol. 21, no. 6, pp. 2316–2330, 2000. 1

[4] A. A. Aydiner and W. C. Chew, "On the nature of super-resolution in inverse scattering," *Proc. IEEE Antennas and Propagation Society International Symposium*, vol. 1, pp. 507–510, June 22-27, 2003. 2

[5] T. J. Cui, W. C. Chew, X. X. Yin, and W. Hong, "Study of resolution and superresolution in electromagnetic imaging for half-space problems," *IEEE Trans. Ant. Propag.*, vol. 52, no. 6, pp. 1398–1411, June 2004. 2

[6] M. A. Anastasio, M. A. Kupinski, and X. Pan, "Noise propagation in diffraction tomography: Comparison of conventional algorithms with a new reconstruction algorithm," *IEEE Transactions on Nuclear Science*, vol. 45, no. 4, pp. 2216–2223, August 1998. 2, 42

[7] O. M. Bucci and T. Isernia, "Degrees of freedom of scattered fields and attainable resolution in electromagnetics inverse problems," in *Direct and Inverse Electromagnetic Scattering*, A. H. Serbest and S. R. Cloude, Eds. Longman, 1996, pp. 207–216. 2

[8] R. Potthast, "On a concept of uniqueness in inverse scattering for a finite number of incident waves," *SIAM J. Appl. Math.*, vol. 58, no. 2, pp. 666–682, Apr. 1998. 2

[9] M. Brandfass, A. Lanterman, and K. F. Warnick, "A comparison of the Colton-Kirsch inverse scattering method with linearized tomographic inverse scattering," *Inverse Problems*, vol. 17, no. 6, pp. 1797–1816, Dec. 2001. 2, 9, 10

[10] D. N. G. Roy, J. Roberts, and M. Schabel, "Noise propagation in linear and nonlinear inverse scattering," *The Journal of the Acoustical Society of America*, vol. 121, pp. 2743–2749, May 2007. 2

[11] A. J. Devaney, "A filtered backpropagation algorithm for diffraction tomography," *Ultrasonic Imaging*, vol. 4, pp. 336–360, 1982. 6, 9

[12] W. C. Chew, *Waves and Fields in Inhomogeneous Media*, D. G. Dudley, Ed. New York: IEEE Press, 1995. 6

[13] R. Porter, "Diffraction limited, scalar image formation with holograms of arbitrary shape," *J. Opt. Soc. Am.*, vol. 60, no. 8, pp. 1051–1059, August 1970. 9

[14] N. N. Bojarski, "Inverse scattering," *Naval Air Systems Command*, vol. Contract N00019-73-C-0312, October 1973. 9

[15] R. Porter and A. Devaney, "Holography and the inverse source problem," *J. Opt. Soc. Am.*, vol. 72, no. 3, pp. 327–330, March 1982. 9

[16] ——, "Generalized holography and computational solutions to inverse source problems," *J. Opt. Soc. Am.*, vol. 72, no. 12, pp. 1707–1713, December 1982. 9

[17] N. Bojarski, "A survey of the near-field far-field inverse scattering inverse source integral equation," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 5, pp. 975–979, September 1982. 9

[18] M. Brandfass and K. Langenberg, "Polarimetric microwave inverse scattering as applied to nondestructive testing," *Review of Progress in Quantitative Nondestructive Evaluation*, vol. 11A, pp. 741–748, 1992. 9

[19] M. Brandfass, *Inverse Beugungstheorie Elecktromagnetischer Wellen: Algorithmen und Numerische Realisierung*. Shaker, 1996. 9

[20] K. Langenburg, M. Brandfass, P. Fellinger, T. Gurke, and T. Kreutter, *A unified theory of multidimensional electromagnetic vector inverse scattering within the Kirchoff or Born approximation*, W. Boerner and H. Uberall, Eds. Springer, 1994. 9

[21] D. Colton, H. Haddar, and M. Piana, "The linear sampling method in inverse electromagnetic scattering theory," *Inverse Problems*, vol. 19, no. 6, pp. 105–137, December 2003. 12

[22] T. Arens, "Why linear sampling works," *Inverse Problems*, vol. 20, no. 1, pp. 163–173, February 2004. 12

[23] N. Shelton and K. F. Warnick, "Behavior of the regularized sampling inverse scattering method at internal resonance frequencies," *Progress in Electromagnetics Research*, vol. 38, pp. 29–45, 2002. 12, 13

[24] F. Cakoni, D. Colton, and H. Haddar, "The linear sampling method for anisotropic media," *The Journal of Computational and Applied Mathematics*, vol. 146, no. 2, pp. 285–299, 2002. 12

[25] K. Warnick and W. C. Chew, "Accuracy of the method of moments for scattering by a cylinder," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 48, no. 10, pp. 1652–1660, Oct 2000. 18, 21

[26] K. Warnick and W. Chew, "Error analysis of the moment method," *IEEE Antennas Propagation Magazine*, vol. 46, pp. 38–53, Dec. 2004. 21

[27] A. Tacchino, J. Coyle, and M. Piana, "Numerical validation of the linear sampling method," *Inverse Problems*, vol. 18, pp. 511–527, 2002. 29