2016

A Novel Iterative Algorithm for Solving Nonlinear Inverse Scattering Problems

Howard Levinson
University of Pennsylvania, howielevinson@gmail.com

A Novel Iterative Algorithm for Solving Nonlinear Inverse Scattering Problems

Abstract
We introduce a novel iterative method for solving nonlinear inverse scattering problems. Inspired by the theory of nonlocality, we formulate the inverse scattering problem in terms of reconstructing the nonlocal unknown scattering potential V from scattered field measurements made outside a sample. Utilizing the one-to-one correspondence between V and T, the T-matrix, we iteratively search for a diagonally dominated scattering potential V corresponding to a data compatible T-matrix T. This formulation only explicitly uses the data measurements when initializing the iterations, and the size of the data set is not a limiting factor. After introducing this method, named data-compatible T-matrix completion (DCTMC), we detail numerous improvements the speed up convergence. Numerical simulations are conducted that provide evidence that DCTMC is a viable method for solving strongly nonlinear ill-posed inverse problems

with large data sets. These simulations model both scalar wave diffraction and diffuse optical tomography in three dimensions. Finally, numerical comparisons with the commonly used nonlinear iterative methods Gauss-Newton and Levenburg-Marquardt are provided.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Applied Mathematics

First Advisor
Vadim A. Markel

Keywords
Inverse Problems, Nonlinear Iterations, Scattering

Subject Categories
Applied Mathematics

A NOVEL ITERATIVE ALGORITHM FOR SOLVING NONLINEAR INVERSE SCATTERING PROBLEMS

Howard Levinson

A DISSERTATION

in

Applied Mathematics and Computational Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2016

Supervisor of Dissertation

_____

Vadim A. Markel, Associate Professor of Radiology and Bioengineering

Graduate Group Chairperson

_____

Charles L. Epstein, Thomas A. Scott Professor of Mathematics

Dissertation Committee:
Philip T. Gressman, Professor of Mathematics
Vadim A. Markel, Associate Professor of Radiology and Bioengineering
John C. Schotland, Professor of Mathematics, Professor of Physics, University of Michigan

ABSTRACT

A NOVEL ITERATIVE ALGORITHM FOR SOLVING NONLINEAR INVERSE SCATTERING PROBLEMS

Howard Levinson

Vadim Markel

We introduce a novel iterative method for solving nonlinear inverse scattering problems. Inspired by the theory of nonlocality, we formulate the inverse scattering problem in terms of reconstructing the nonlocal unknown scattering potential V from scattered field measurements made outside a sample. Utilizing the one-to-one correspondence between V and T, the T-matrix, we iteratively search for a diagonally dominated scattering potential V corresponding to a data compatible T-matrix T. This formulation only explicitly uses the data measurements when initializing the iterations, and the size of the data set is not a limiting factor. After introducing this method, named data-compatible T-matrix completion (DCTMC), we detail numerous improvements the speed up convergence. Numerical simulations are conducted that provide evidence that DCTMC is a viable method for solving strongly nonlinear ill-posed inverse problems with large data sets. These simulations model both scalar wave diffraction and diffuse optical tomography in three dimensions. Finally, numerical comparisons with the commonly used nonlinear iterative methods Gauss-Newton and Levenburg-Marquardt are provided.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Inverse scattering problems (ISPs) are a topic of interest in many fields. With applications from geophysics to medical imaging, the common goal of reconstructing one or more unknown characteristics from the produced scattered field of an object allows one to gain knowledge of what is inside an opaque region in a nondestructive manner. This can be extremely valuable information, and modern examples include diffuse optical tomography [3, 11], diffraction tomography [15, 21], electrical impedance tomography [2, 13, 27], electromagnetic imaging (near-field [7, 8, 16] and far-field [9, 53]), and seismic tomography [28, 29]. All of these methods share the restriction that scattered field measurements can only be made on the boundary or exterior to the object of interest.

In theory, if the scattering properties of the object are known ahead of time, it is relatively simple to predict the resulting scattered field when a source wave or

multiple source waves scatter after making contact. This is the forward scattering problem, which is of limited use in tomography, as the scattering effect cannot be known ahead of time. The inverse problem is more critical, and is in general much more difficult.

Inverse scattering problems are well known to be ill-posed [40]. As defined by Hadamard, this implies that the inverse problem fails to have a unique solution, fails to have a solution at all, or the solution does not depend continuously on the measured data. It is common for an ill posed problem to suffer from more than one of these deficiencies. When one cannot make direct measurements inside the sample (as is the case for ISPs), it is typically impossible to uniquely determine a solution. Moreover, once one takes into account the potentially robust noise present in any scattering data measurements, it is clear that finding a reasonable solution is no easy task.

To handle this ill-posedness, suitable regularization techniques must be applied. Popular choices such as Tikhonov regularization sacrifice exactness to restore existence, uniqueness, and stability of solutions [18]. With an appropriate regularization scheme, a reasonably precise solution can be found for many difficult ISPs.

Further complicating matters is the fact that many ISPs are nonlinear [44, 47]. A major contributing factor to this nonlinearity is the presence of multiple scatterings. One cannot blindly use the superposition principle to independently reconstruct two nearby scatterers if the field is sufficiently strong. One would be ignoring the

potentially strong scattering effect between the two scatterers. This nonlinearity adds significant difficulties to any solution process, as the well-developed class of linear solvers may not be applicable.

Linear approximations can be useful in many instances of ISPs, but many problems contain large levels of nonlinearity, and true nonlinear methods must be employed. While there are several analytic inversion approaches (such as the inverse Born series [38, 39]) and Bayesian inference nondeterministic methods [51] that have their merits, the class of nonlinear iterative algorithms are a popular choice for approaching ISPs. With an iterative approach, regularization can be added as a realization of continuous regularization strategies, or as a rule for determining the stopping index. Three-dimensional inverse scattering problems can be a significant computational undertaking, and iterative methods often can provide reasonable results in a reasonable amount of computation time.

The most conventional choices of nonlinear iterative algorithms are the family of variants on Newton's method [25]. These methods search for a descent direction at each iteration that moves closer to the desired result. Unfortunately, this search for a descent direction has the serious drawback of requiring access and the use of all available data points at each iteration. With the heavy computational workload associated with many of these problems, one does not want to take too many data measurements that will significantly slow down the solution process.

But in direct competition with reducing computing time is the need to obtain

accurate results. As mentioned there is strong ill-posedness present due to the restriction on where data measurements can be made. One of the key tools to combat this ill-posedness is the ability to make a significant number of data measurements. In diffuse optical tomography for example, it is not totally uncommon to expect data sets on the order of 10^9 to obtain acceptable resolution [12, 31, 32, 50]. Thus with these nonlinear iterative techniques, it seems one must always balance the need for additional data measurements to supply enough information, with the associated increase in computation time.

This thesis introduces a novel nonlinear iterative algorithm named data-compatible T-matrix completion (DCTMC). Motivated by the need for efficient algorithms that can solve large three-dimensional ill-posed ISPs in a reasonable amount of time, DCTMC is not limited by the number of data measurements taken. That is, adding additional data points does not significantly increase the computational load of each iteration. In fact, the data set is only explicitly used once to initialize the iterations, and the subsequent processes that utilize the data are inconsequential operation-wise in comparison with the other operations. Thus, DCTMC has the potential to be faster and more accurate than the current choices for nonlinear iterative methods.

The remainder of this thesis is organized as follows. We begin in Chapter 2 with a brief overview of scattering theory and standard nonlinear iterative approaches

to solve the related inverse problem. Next in Chapter 3 we introduce the data-compatible T-matrix completion algorithm, as well as many computational shortcuts and improvements. In Chapter 4, we provide several important results associated with the linearized DCTMC algorithm. We test the DCTMC algorithm with substantial simulations in Chapter 5 that model scalar wave diffraction and diffuse optical tomography. Lastly, we compare DCTMC analytically and numerically to standard nonlinear iterative methods in Chapter 6 followed by a final summary in Chapter 7.

Chapter 2

Theory

2.1 Scattering Theory

The background information on scattering theory in this section is based on the work in [20, 41]. The goal of any forward scattering problem is to compute the resulting field produced by the interaction of an incident wave and an inhomogeneous object compared to background. We state this general physical situation by

$$\mathscr{L}u(\mathbf{r}) = q(\mathbf{r}) \ , \tag{2.1.1}$$

where $\mathscr{L}$ is a linear operator, $u(\mathbf{r})$ is the physical field, and $q(\mathbf{r})$ is the induced source at a location $\mathbf{r}$. We are working in the frequency domain, and while both u and q contain the frequency ω as an argument, this argument is dropped as we consider the case of fixed frequency. The dependence between the field and source and frequency is purely parametric and can be reintroduced if necessary.

For example, varying the frequency is important in several types of time domain problems, but for our purposes it will be ignored.

We can write $\mathscr{L} = \mathscr{L}_0 - V$, where $\mathscr{L}_0$ is the linear operator governing the field without an added inhomogeneous scattering region. Here, V is the scattering potential, or interaction operator, which models the interaction between the field and the obstacle. It is assumed that V is compactly supported in a finite scattering volume.

Thus, absorbing the known term $\mathscr{L}_0 u$ into the induced source term q, we can write the relationship between the field and the induced source by

$$Q(\mathbf{r}) = V(\mathbf{r})u(\mathbf{r}) \ . \tag{2.1.2}$$

In this sense, the incident wavefield interacts with the potential generating the induced source Q. While scattering theory can be very general, potential scattering describes many problems of interest in electromagnetics, optics, and acoustics. Ignoring the vectorial nature of electromagnetic wave fields, the literature provides the general form for the scattering potential

$$V(\mathbf{r}) = k_0^2[1 - n_r^2(\mathbf{r})] \ , \tag{2.1.3}$$

where $n_r(\mathbf{r})$ is the relative index of refraction which measures the ratio of the index of refraction of the scatter to the index of refraction of the background material. That is,

$$n_r(\mathbf{r}) = \frac{n(\mathbf{r})}{n_0}. \tag{2.1.4}$$

Keep in mind that we are still suppressing the frequency argument, which is present iin all terms in the equation above. For a fixed frequency ω, the constant wavenumber in (2.1.3) k_0 is defined as $k_0 = (\omega/c)n_0(\omega)$. In this view, the scattering potential is completely defined by its complex index of refraction.

The field produced from the forward scattering problem must satisfy the inhomogeneous Helmholtz equation, that is

$$[\nabla^2 + k_0^2]u(\mathbf{r}) = V(\mathbf{r})u(\mathbf{r}) \ , \tag{2.1.5}$$

where we have replaced the source term on the right-hand side by (2.1.2). Reducing the field u into a decomposition of the incident and scattered fields, we obtain

$$u = u_{\text{inc}} + u_{\text{scatt}} \ , \tag{2.1.6}$$

where we conclude that the incident field satisfies the homogeneous Helmholtz equation. This is assumed despite the fact that the incident field will in all likelihood be propagated through a background medium that is not equal to free space. In this sense, the incident field will satisfy the inhomogeneous Helmholtz equation. However, as the source is separated from the scatter, the incident field u_{inc} satisfies the homogeneous Helmholtz equation within the compact scattering region. An explicit example of solving the inhomogeneous Helmholtz equation for the scattered field will be provided in Section 5.1.1. For now, we will return to symbolic notation to maintain generality.

We write $u_{\text{inc}} = G_0 q$, where G_0 is the unperturbed Green's function which is the inverse of $\mathscr{L}_0$. This corresponds correctly to the case where there is no scattering, i.e.

$V = 0$. We will denote the complete Green's function of the system by $G = \mathscr{L}^{-1}$. Thus, the total field can be written as $u = Gq$. It is worth highlighting the fact that G exists whenever the forward problem has a solution. Now, acting on both sides of the equation (2.1.1) by the unperturbed Green's function G_0, we obtain the Lippman-Schwinger integral equation in symbolic form

$$u = u_0 + G_0 V u \ . \tag{2.1.7}$$

Rearranging terms, we can then obtain the solution for the scattered field by

$$u_{\text{scatt}} = (G - G_0)q = G_0(I - VG_0)^{-1}VG_0 q \ . \tag{2.1.8}$$

We now turn to the inverse scattering problem where we are interested in reconstructing the scattering potential V. To have a chance of reconstructing V with a reasonable amount of accuracy, it is imperative to have multiple views of the scattering potential. That is, multiple scattering experiments must be performed with different incident fields. Single experiments can be used for inverse source problems, but the extra information provided from multiple scattering experiments is crucial for the reconstruction process. To that end, we obtain data measurements by measuring the scattered field at detector locations $\mathbf{r}_d$. Then, to obtain multiple views, we can propagate waves through the medium from localized sources $q(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_s)$, changing the location of $\mathbf{r}_s$ for each experiment. We will denote the set of sources by Σ_s and the set of detectors used by Σ_d. Then, by collecting the measurements made at each location $\mathbf{r}_d \in \Sigma_d$ for each source $\mathbf{r}_s \in \Sigma_s$, we obtain a data function

of two variables $\Phi(\mathbf{r}_d, \mathbf{r}_s)$. Looking back at equation (2.1.8), we see that replacing q by each point source, the equation

$$G_0(I - VG_0)^{-1}VG_0 = \Phi \tag{2.1.9}$$

holds for restricted values of $\mathbf{r}$ and $\mathbf{r}'$. That is, the operator G_0 is the same operator all three times in the equation above, but each kernel $G_0(\mathbf{r}, \mathbf{r}')$ has different restrictions. For the first operator multiplication, $\mathbf{r} \in \Sigma_d$ and $\mathbf{r}' \in \Omega$, where Ω is the scattering sample. The last operator multiplication is restricted by $\mathbf{r} \in \Omega$, $\mathbf{r}' \in \Sigma_s$. Finally, the operator G_0 that both has an operator multiplication and inversion is restricted solely within the sample, $\mathbf{r}, \mathbf{r}' \in \Omega$. Thus, the inverse scattering problem can be stated as reconstructing the unknown interaction operator V from a data function Φ measured outside the sample. The inverse problem is clearly nonlinear.

While Φ is theoretically measured continuously across the surfaces Σ_s and Σ_d, it is impossible to do this in practice. One can only measure the scattered field at a finite number of points, and likewise can only illuminate the sample from a finite number of point sources. In this regard, the operator equation (2.1.9) must be discretized to allow any numerical or analytical solving to take place. Thus, after discretizing the sample, we consider the matrix equation

$$A(I - V\Gamma)^{-1}VB = \Phi \ . \tag{2.1.10}$$

In equation (2.1.10), we have replaced the three instances of G_0 by matrices of different notation, to highlight the fact that they are each restricted in a different

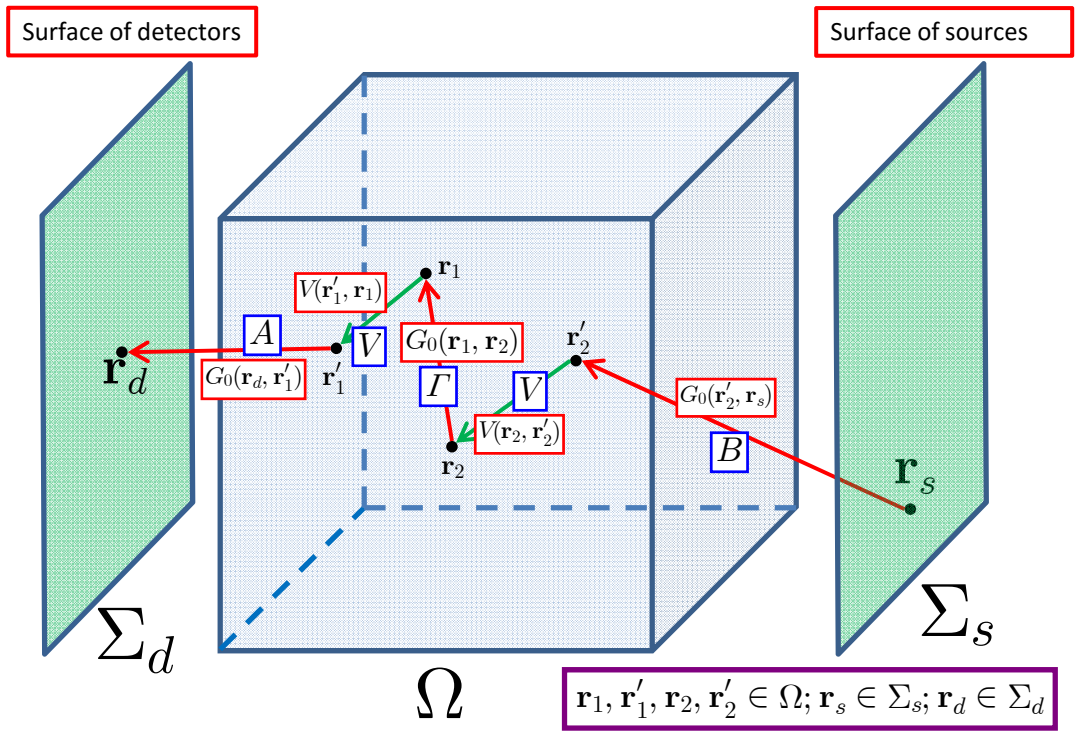


Figure 2.1: Illustration of the imaging geometry. The matrices A, B, Γ and V are the discretized operators in (2.1.10) and are depicted in blue frames. The restricting and sampling of the kernels $G_0(\mathbf{r}, \mathbf{r}')$ and $V(\mathbf{r}, \mathbf{r}')$ are demonstrated by the endpoints of the arrows. The multiple scattering depicted corresponds to the second order term $G_0VG_0VG_0$ in the formal power-series expansion of the left-hand side in (2.1.9). Note that in the local limit of the potential $V(\mathbf{r}, \mathbf{r}')$ the two green arrows contract to two vertexes at $\mathbf{r}_1 = \mathbf{r}_1'$ and $\mathbf{r}_2 = \mathbf{r}_2'$.

manner. A is restricted from the detectors to the sample, B is restricted from the sources to the sample, and Γ is restricted within the sample itself. An example of this is shown in Figure 2.1. While the slab geometry example in this figure is the main geometric setup we will be investigating in our later simulations, the above formulation is general and does not require any specific shape for the surface of detectors or sources, or the sample. The only physical requirement is that there is no intersection between Σ_s or Σ_d and the sample Ω.

Equation (2.1.10) is not only general with respect to the geometric setup of the problem and the discretization choices, but is very general in regards to its application to scattering theory. The geometry matrices A, B, and Γ are all theoretically known, and encompass all of the information of the physical model. Since the elements of the matrix Φ are all acquired experimentally, the only unknown in (2.1.10) is the scattering potential V. It is important to note that this equation cannot be solved simply by matrix inversion, as the matrices A and B are typically of low rank. In fact, the invertibility of these matrices coincides with performing measurements within the sample, which is a strict violation of the problem. We now turn to methods for recovering the interaction operator V.

2.2 Nonlinear Reconstructions

Oftentimes, a linearization of the ISP is an acceptable method for solving this inverse scattering problem. For these methods, a linearizing transformation is applied to the data function on the right-hand side of (2.1.10) to simplify the forward problem to be linear in terms of the unknown interaction matrix V, namely

$$AVB = L[\Phi] \ . \tag{2.2.1}$$

Moreover, it is conventional to assume that the interaction matrix is strictly diagonal, and such by combining the geometry matrices A and B into the matrix K by the entrywise formula $K_{(mn),j} = A_{mj}B_{jn}$, and unrolling the matrix $L[\Phi]$ into a

vector ψ, one obtains the linear equation

$$Kv = \psi \ . \tag{2.2.2}$$

This equation can then be solved for v using any linear equation solver. We will discuss more linearizing techniques in Sections 4.1 and 5.1.

Another approach to solving inverse scattering problems is to use one of several known nonlinear techniques for reconstructing the interaction matrix. These approaches intend to reconstruct the images with greater accuracy as compared to linear solvers which make approximations that can significantly reduce the accuracy of the model. However, nonlinear methods are more computationally intensive than linear solvers. For that reason, there are many practical scenarios when a linear reconstructions are acceptable. But when the results obtained from the linearization of the ISP is not worthy of its application, one must increase the computational load and use a nonlinear method.

The most popular approach to solving nonlinear inverse scattering problems is to use Newton's method or one of its variants. This section will review the idea behind these mainstream approaches. These methods are reviewed in more detail in [1, 17, 30]. However, it is worth keeping in mind alternative nonlinear methods such as the inverse Born series, where the reconstruction is an analytically computable result directly from the data, and several non-deterministic approaches based on Bayesian inference.

For these mainstream approaches, it is common to ignore the algebraic structure

of equation (2.1.10) and rewrite the ISP in the form $F[v] = 0$, where v are the diagonal entries of V and F is the nonlinear functional that relates the scattering potential to the data. This can be alternatively written as $ATB = \tilde{F}[v]$ which can more clearly elucidate the relationship the nonlinear functional $\tilde{F}$ has with the data. From here, one thinks of the inverse scattering problem as an optimization problem. The solution $\hat{v}$ is found by minimizing an objective function

$$\hat{v} = \arg\min_{v} \Psi(v) \ . \tag{2.2.3}$$

In most cases, it is common to treat this objective function Ψ as the statement of a nonlinear least squares problem, that is

$$\Psi(v) = \sum_{i=1}^{N_s} \sum_{j=1}^{N_d} \left[\Phi^c(v) - \Phi^m\right] \ , \tag{2.2.4}$$

where Φ^m is the measured data for specific source and detector pair, and $\Phi^c(v)$ is the calculated data for that same pair given a scattering potential within the vector v. We denote the number of sources and detectors used as N_s and N_d respectively. Then starting from an initial guess v_0, the forward problem for $\Phi^c(v_0)$ is calculated, and then based on the specific minimization scheme, our guess is updated by $v_{k+1} = v_k + \gamma_k d_k$, where d_k is a descent direction, and γ_k is a step size. Oftentimes in the literature $\gamma_k = 1$, as the step size is typically independent of the method used to determine the descent direction. However, one can always add an optimization step that conducts a line search for a useful value of γ_k. That is,

$$\gamma_k = \min_{\gamma} \Psi(v_k + \gamma d_k) \ . \tag{2.2.5}$$

We will now mention some of the most commonly used methods to determine the direction d_k.

2.2.1 Landweber Iteration

Landweber iteration is a special case of steepest descent, in which constraints are placed on the step size γ_k. Thus, $d_k = -\nabla\Psi(v)$. Numerically, defining the residual column vector $R(v)$ with entries

$$r_i(v) = \Phi_i^c(v) - \Phi_i^m \ , \tag{2.2.6}$$

allows us to express the gradient of the objective function as the product

$$\nabla\Psi(v) = J(v)^T R(v), \tag{2.2.7}$$

where J is the Jacobian matrix defined as

$$J(v) = \begin{pmatrix} \frac{\partial r_1(v)}{\partial v_1} & \cdots & \frac{\partial r_1(v)}{\partial v_N} \\ \vdots & \ddots & \cdots \\ \frac{\partial r_M(v)}{\partial v_1} & \cdots & \frac{\partial r_M(v)}{\partial v_N} \end{pmatrix} \ . \tag{2.2.8}$$

Here, N is the number of discretized elements in the sample, and $M = N_d N_s$ is the total number of data points. Thus, Landweber iteration can be succinctly summarized as

$$v_{k+1} = v_k - J(v)^T R(v) \ . \tag{2.2.9}$$

While steepest descent algorithms are well known for converging from very far away initial guesses, this convergence can be extremely slow. Thus, this algorithm

is rarely used directly in practice.

2.2.2 Gauss-Newton Method

A faster and more popular method for nonlinear optimization is the Gauss-Newton iteration method. While steepest descent used only first-order derivatives, Gauss-Newton uses an approximation to the second derivative. We begin with the first-order Taylor approximation to the gradient of the objective function

$$\nabla\Psi(v_{k+1}) = \nabla\Psi(v_k + d_k) \approx \nabla\Psi(v_k) + \nabla^2\Psi(v_k)d_k \ . \tag{2.2.10}$$

Thus as the objective function is minimized when this equation is equal to zero, we are interested in solving the set of equations

$$\nabla^2\Psi(v_k)d_k = -\nabla\Psi(v_k) \ , \tag{2.2.11}$$

for the Gauss-Newton direction d_k. The term $\nabla^2\Psi(v_k)$ is the Hessian operator and can be calculated as

$$\nabla^2\Psi(v_k) = 2J(v_k)^T J(v_k) + \sum_{m=1}^{M}\sum_{i=1}^{N_v}\sum_{j=1}^{N_v} r_i(v)\frac{\partial r_m(v)}{\partial v_i \partial v_j} \ . \tag{2.2.12}$$

In practice, the second term requires a very lengthy calculation, and is in fact typically much smaller than the first. Therefore, Gauss-Newton method uses the approximation

$$\nabla^2\Psi(v_k) \approx 2J(v_k)^T J(v_k) \ . \tag{2.2.13}$$

This combined with the previous result $\nabla\Psi(v_k) = 2J(v_k)^T R(v_k)$, we obtain a set of linear equations to solve for the search direction, namely

$$J(v_k)^T J(v_k) d_k = J(v_k)^T R(v_k) \ . \tag{2.2.14}$$

Gauss-Newton can converge much quicker than steepest descent, but fails to converge for initial guesses not close to the desired result. However if an initial guess is sufficiently close, convergence is quadratic.

2.2.3 Levenburg-Marquardt Method

The Levenburg-Marquardt method is a popular nonlinear iterative algorithm that balances the benefits of both steepest descent iteration and Gauss-Newton method. In its purest form, a positive diagonal matrix is added to the approximation to the Hessian on the left-hand side of (2.2.14). Most commonly this diagonal matrix is chosen to be a multiple of the identity matrix, giving the set of linear equations governing this method to be

$$(J(v_k)^T J(v_k) + \lambda_k I) d_k = J(v_k)^T R(v_k) \ . \tag{2.2.15}$$

This choice closely resembles the well-known Tikhonov regularization and is in fact equivalent to iteratively regularized Gauss-Newton. For very small values of the damping parameter λ_k, the direction is clearly very close to the direction obtained by pure Gauss-Newton. However, larger values of λ_k suppresses the second derivatives and as the left-hand side behaves more like the identity matrix, the search direction

is closer to steepest descent. Thus, the choice of λ_k can be modified each iteration – if it is known that our intermediate result is reasonably close to correct, λ_k should be made small to have near quadratic convergence behavior a la Gauss-Newton. If we can be reasonably sure we are far away, a larger value of λ_k would be preferred for a larger convergence radius.

2.2.4 Nonlinear Conjugate Gradient

The last common nonlinear reconstruction technique we will review is nonlinear conjugate gradient. Regular conjugate gradient is a linear iterative method for solving symmetric positive definite linear systems. Convergence is guaranteed for the linear case in at most n iterations, where n is the size of the system, and acceptable results can be found much faster depending on the spectrum of the matrix involved. For completeness sake (and as this algorithm is used later in the simulated linear reconstructions) this algorithm for solving the equation $Kv = \psi$ goes as follows for an initial guess v_0:

1. $r_0 = Kv_0 - \psi$

   $\Delta v_0 = -r_0$

2. While $\|r_k\| > \epsilon$ do

   (a) $\alpha_k = \frac{r_k^T r_k}{\Delta v_k^T K \Delta v_k}$

   (b) $v_{k+1} = v_k + \alpha_k \Delta v_k$

(c) $r_{k+1} = r_k + \alpha_k K \Delta v_k$

(d) $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$

(e) $\Delta v_{k+1} = -r_{k+1} + \beta_{k+1} \Delta v_k$; k =k +1

Nonlinear conjugate gradient method can then be directly obtained from its linear counterpart by removing all instances of the residual r_k from the above algorithm and replacing it with $\nabla \Psi(v_k)$.

2.3 T-matrix

The T-matrix is defined as the operator that relates the complete and unperturbed Green's functions through the Dyson equation

$$G = G_0 + G_0 T G_0 \ . \tag{2.3.1}$$

A simple inspection of equation (2.1.8) solves for this operator as

$$T = (I - V G_0)^{-1} V \ . \tag{2.3.2}$$

Another interesting way of looking at the T-matrix is to define the transition operator which maps incident waves into the product of the interaction operator and the total wavefield [14, 46]. That is,

$$T u_0 = V u \ , \tag{2.3.3}$$

and if we multiply both sides of the Lippman Schwinger (2.1.7) equation by the scattering potential V, we obtain

$$Tu_0 = Vu_0 + VG_0Tu_0 \ , \tag{2.3.4}$$

or equivalently

$$T = V + VG_0T \tag{2.3.5}$$

Again solving this equation for T we arrive precisely at the definition in (2.3.2). From the definition of the transition operator, one can conclude that the scattering amplitude outside of the sample is proportional to the boundary value of the T-matrix over a sphere. Thus, scattering amplitude can be determined directly from the T-matrix, but not vice versa. Clearly the T-matrix completely determines the scattering operator through the one-to-one correspondence in (2.3.2), but merely knowing the scattering amplitudes only determines the scattered field outside of the sample. Thus, one can think of the inverse scattering problem as computing the T-matrix from some analytic continuation of the scattering amplitudes. However, there is no stable method to accomplish this.

As an actual discretized matrix, the relationship between the T-matrix and V is written as

$$T = (I - V\Gamma)^{-1}V = V(I - \Gamma V)^{-1} \ . \tag{2.3.6}$$

Clearly, the T-matrix is symmetric. The inverse operation in equation (2.3.6) is known to exist if V is physically admissible. We can also write V in terms of T.

From (2.3.6), we can calculate that

$$V = (I + T\Gamma)^{-1}T = T(I + \Gamma T)^{-1} \ . \tag{2.3.7}$$

Thus, we have a one-to-one correspondence between the transition matrix and the interaction matrix. Knowledge of either of these matrices fully determines the forward problem, but express the properties of the scattered field in different manners.

Chapter 3

The Data-Compatible T-matrix Completion Algorithm

3.1 Motivation

We now turn towards the crux of this these – introducing the novel nonlinear iterative method, data compatible T-matrix completion. Before detailing the specifics of the algorithm, it is worth discussing the desire for additional methods to approach nonlinear inverse scattering problems. All of the methods reviewed in Section 2.2 calculate an objective function which depends on all available data measurements, and is subsequently minimized using one of the previous schemes. In its naive form, computationally implementing any of these methods at least require storage of the Jacobian matrix which is of size $M \times N_v$, where M is the number of data elements

and N_v is the number of discretized elements one wants to reconstruct. And then for say Gauss-Newton, one also requires the matrix multiplication $J^T J$, which requires $O(M^2 N_v)$ operations. This calculation quickly becomes unwieldy as the number of data measurements increases. This is clearly unwanted, as one of the major tools we have to solve inverse scattering problems is the ability to conduct multiple experiments and obtain very large data sets. However, with traditional methods one must balance the benefits from additional data measurements versus the increased computational workload.

It would be dishonest to leave out the fact that there exist more efficient methods for generating and multiplying these large Jacobian matrices, and thus run faster than $O(M^2 N_v)$. Adjoint methods such as in [4, 5, 43] can take advantage of the sparse nature of the Jacobian matrix in a manner where the Jacobian matrix is never explicitly computed. But while there are certainly computational improvements one can make when using these Newton methods, one cannot escape the fact that the computational workload increases as our data set increases as well.

There is substantial evidence that inverse scattering problems of interest require strongly overdetermined data sets in order to obtain accurate results [36, 37]. For example, to obtain optimal lateral resolution for diffuse optical tomography in the slab geometry with 100×100 grid cross-sections, one needs on the order of 300×300 panels of sources and detectors on either side of the sample. This set up produces roughly 10^{10} data measurements. Thus, the size of the data set overshadows the

size of the discretization mesh, and is the limiting factor for mainstream nonlinear approaches. This relative magnitude of the size of the overdetermined data set holds for many other ISPs of interest.

There has been a great deal of interest in both acquiring these large data sets, and experimentally determining the optimal size of data sets for DOT [6, 31, 50]. Noise and severe ill-posedness can warrant a reduction in the optimal size of the data set, as one can reach the limit of resolution, and especially noisy data measurements are better off left discarded. However, the majority of these works use linearized reconstruction techniques to be able to handle the large data sets efficiently. Thus, it is certainly desirable to develop nonlinear techniques that will forgo these linear approximations that reduce accuracy, but can still reconstruct from large data sets with reasonable computing time.

This is the main goal of DCTMC – to present a nonlinear solver in which increasing the size of the data set has a negligible impact on computation time. The descriptions of the Newton type solvers in Section 2.2 are very clear in that the unknown interaction matrix from the forward matrix equation (2.1.10) is treated as being strictly diagonal and can thus be reduced to a column vector. This comes from the theory of locality, which states that certain physical properties or events at a specified point $\mathbf{r}$ are only influenced by the field present within a finite radius ℓ of that point (as esoterically explained in [49]). While the degree of locality is

never zero, the radius of influence ℓ is typically small enough (often on the atomic scale which is much smaller than any discretization mesh), that the forced diagonal structure of V is accurate.

DCTMC relaxes this locality restriction, and allows our iterative process to search for nonlocal interaction operators as well. For example, Ohm's law in local electrodynamics is $\mathbf{J}(\mathbf{r}) = \sigma(\mathbf{r})\mathbf{E}(\mathbf{r})$. Relaxing this to nonlocal electrodynamics, the current density $\mathbf{J}(\mathbf{r})$ is given by

$$\mathbf{J}(\mathbf{r}) = \int V(\mathbf{r}, \mathbf{r}')\mathbf{E}(\mathbf{r}')d^3r' \ , \tag{3.1.1}$$

where V is the integral interaction operator as in (2.1.2). Now we consider the Calderon problem where we want to find the conductivity $\sigma(\mathbf{r})$ from voltage drop measurements taken after two electrodes inject direct current. Finding a nonlocal kernel $V(\mathbf{r}, \mathbf{r}')$ from 3.1.1 that is consistent with the voltage drop measurements can be simple as this problem is very underdetermined – the degrees of freedom of $V(\mathbf{r}, \mathbf{r}')$ is much larger than the size of the data set. But for the exact same reason, $V(\mathbf{r}, \mathbf{r}')$ cannot be determined uniquely. What we have accomplished so far by generalizing the linear relationship between the current density and electric field to a nonlocal setting is the ability to find many solutions that are compatible with the data. The question remains, how can we narrow down to our desired solution? Keep in mind that our generalization to the nonlocality of V was only a mathematical trick, we still expect V to be local to some degree, that is $V(\mathbf{r}, \mathbf{r}') \to 0$ when $|\mathbf{r}-\mathbf{r}'| > \ell$. Thus, we can safely assume that V is approximately diagonal. Furthermore, local

conductivity can be obtained from the integral operator by $\sigma(\mathbf{r}) = \int V(\mathbf{r}, \mathbf{r}')d^3r'$. Our search out of our data consistent solutions is narrowed down to a search for an approximately diagonal nonlocal kernel $V(\mathbf{r}, \mathbf{r}')$. The concept for DCTMC can be summarized as:

(1) An initialization step where a class of kernels $V(\mathbf{r}, \mathbf{r}')$ that are compatible with the data is formed. This initialization is the only place when the data measurements are explicitly used. Moreover, the size of the data set is not a limiting factor for defining this class of data compatible solutions.

(2) Then we iteratively reduce the off-diagonal norm of $V(\mathbf{r}, \mathbf{r}')$ while ensuring that all iterations of $V(\mathbf{r}, \mathbf{r}')$ remain compatible with the data.

(3) Once the ratio of the off-diagonal and diagonal norms of $V(\mathbf{r}, \mathbf{r}')$ is sufficiently small, we have found our diagonally dominated data-compatible interaction operate. We then compute the local interaction $\sigma(\mathbf{r}) = \int V(\mathbf{r}, \mathbf{r}')d^3r'$. This is the final solution to the nonlinear ISP.

While this motivation was stated for electrical impedance tomography, its statement is very general and can be applied to many other inverse scattering problems. The fact that (1) above is the only time the data is used (and is not limiting) highlights the potential advantages of this algorithm.

Lastly of note, the T-matrix from Section 2.3 plays a crucial role in the DCTMC algorithm. T-matrix methods have been proven useful in solving nonlinear inverse

problems [33, 52]. These methods are source and detector independent, as the computation of the T-matrix gives the forward solution for any source and detector arrangement. The mainstream Newton approaches typically utilize finite difference and finite element methods for the forward problem, which must be run for each source independently. However, finite element methods generate sparse matrices which allow for some of the reductions mentioned required for Jacobian calculations. The trade off for working with source/detector independence is that computing the T-matrix as in (2.3.6) requires inverting a dense matrix. This one-to-one correspondence between the interaction operator V and the T-matrix plays an important role in DCTMC.

3.2 The Experimental T-matrix

We return to the discretized forward equation for the scattering problem

$$A(I - V\Gamma)^{-1}VB = \Phi \ , \tag{3.2.1}$$

but now substituting in equation (2.3.6) to obtain the forward equation

$$ATB = \Phi \ , \tag{3.2.2}$$

which is a linear equation in T. It is worth taking some time to discuss the relative sizes of all matrices in equation (3.2.2). The volume is discretized into N_v voxels, and recall we have N_s sources and N_d detectors. In the majority of practical setups,

it is clear that the inequality

$$N_s, N_d \ll N_v \ll N_s N_d \tag{3.2.3}$$

must hold if there is any hope of solving the inverse problem with reasonable accuracy. The first inequality is generally true due to the nature of most problems of interest, namely that it is impossible to take enough useful measurements to perfectly determine a solution. The second inequality states that we need an overdetermined system to handle the ill-posedness involved. So while this inequality holds true in general, it is useful to consider the order of the values used throughout this paper. As in the setup in Fig. 6.1, let the measurement planes Σ_s and Σ_d be identical, with both sources and detectors scanned on $L \times L$ square grids. The sample is discretized with the same pitch as the source/detector grids on an $L \times L \times L$ cubic grid. Then, $N_s = N_d = L^2$, $N_v = L^3$, and $N_s N_d = L^4$, which certainly satisfies condition (3.2.3) for reasonable values of L.

The dimensions of all matrices in equation (3.2.2) are shown in Fig. 3.1. For all of the reasons mentioned, we can assume that A and B are not invertible. However, since we have rewritten the forward equation to be linear equation in our fundamental unknown T, we can fully utilize the knowledge of pseudoinverses. This will lead us to the definition of the experimental T-matrix, which is the central concept to DCTMC.

The idea is to find a condition on T that completely satisifies equation (3.2.2). We begin by considering the singular value decompositions of the geometry matrices

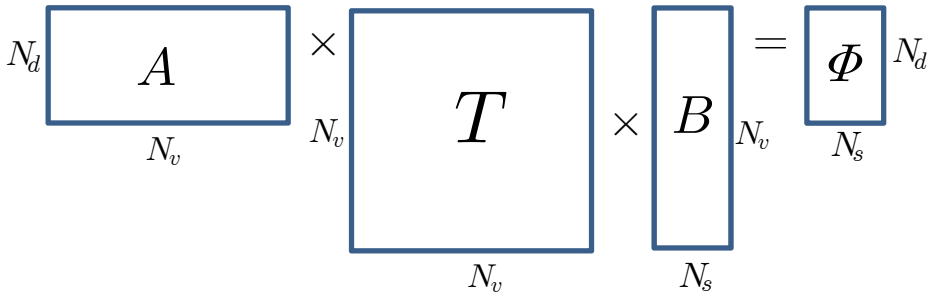


Figure 3.1: Schematic block diagram of equation (3.2.2). N_v is the number of discretized voxels, while N_d and N_s are the numbers of detectors and sources respectively.

A and B. That is,

$$A = \sum_{\mu=1}^{N_d} \sigma_\mu^A \left| f_\mu^A \right\rangle \left\langle g_\mu^A \right| \ , \quad B = \sum_{\mu=1}^{N_s} \sigma_\mu^B \left| f_\mu^B \right\rangle \left\langle g_\mu^B \right| \ , \tag{3.2.4}$$

where σ_μ^A are the singular values of A, and $\left| f_\mu^A \right\rangle$ are the left singular vectors of A of length N_d and $\left| g_\mu^A \right\rangle$ are its right singular vectors of length N_v. This is similar for B, but with the left singular vectors $\left| f_\mu^B \right\rangle$ being of length N_v and the right singular vectors $\left| g_\mu^B \right\rangle$ being of length N_s. Note that the summations in (3.2.4) have upper indicies of N_d and N_s, due to our assumption that $N_d, N_s \leq N_v$ (all singular values of larger index than N_d or N_s are identically equal to zero). Now using orthogonality of singular vectors and rearranging our forward equation, we obtain the entrywise condition

$$\sigma_\mu^A \sigma_\nu^B \tilde{T}_{\mu\nu} = \tilde{\Phi}_{\mu\nu} \ , \quad 1 \leq \mu \leq N_d \ , \ 1 \leq \nu \leq N_s \ , \tag{3.2.5}$$

where we have the following entrywise definitions:

$$\tilde{T}_{\mu\nu} \equiv \left\langle g_\mu^A | T | f_\nu^B \right\rangle \ , \quad 1 \leq \mu, \nu \leq N_v \ ; \tag{3.2.6a}$$

$$\tilde{\Phi}_{\mu\nu} \equiv \left\langle f_\mu^A | \Phi | g_\nu^B \right\rangle \ , \quad 1 \leq \mu \leq N_d \ , \quad 1 \leq \nu \leq N_s \ . \tag{3.2.6b}$$

We now let R_A be the $N_v \times N_v$ unitary matrix formed by the column singular vectors $|g_\mu^A\rangle$ and R_B be the $N_v \times N_v$ unitary matrix formed by the column singular vectors $|f_\mu^B\rangle$. Then, the first line of equation (3.2.6) implies with the unitary property that

$$\tilde{T} = R_A^* T R_B \quad \text{and} \quad T = R_A \tilde{T} R_B^* \ . \tag{3.2.7}$$

Note that even though this transformation from T to $\tilde{T}$ is invertible, it is not a conventional rotation due to the fact that $R_A \neq R_B$ in general. We now call the matrix T that has been used thus far the T-matrix in *real-space representation*, while the "rotated" matrix $\tilde{T}$ is named the T-matrix in *singular-vector representation*. We could name $\tilde{\Phi}$ and Φ similarly, albeit using different unitary matrices (of dimension $N_d \times N_d$ and $N_s \times N_s$). But to avoid confusion, since $\tilde{\Phi}$ is not a recurring aspect of the DCTMC algorithm, we refrain from explicitly defining these "rotations".

Returning to the equivalent singular-vector representation forward equation (3.2.5), we see that we can numerically reduce this constraint to

$$\tilde{T}_{\mu\nu} = \begin{cases} \dfrac{1}{\sigma_\mu^A \sigma_\nu^B} \tilde{\Phi}_{\mu\nu} & , \ \text{if } \sigma_\mu^A \sigma_\nu^B > \epsilon^2 \ ; \\ \text{unknown} & , \ \text{otherwise} \ . \end{cases} \tag{3.2.8}$$

where we have used the conventional notation that σ_μ^A and σ_μ^B are equal to zero for $\mu > N_d$ and $\nu > N_s$. We let ϵ be a small positive constant that acts as a reg-

ularization parameter to deal with numerical imprecisions. With infinite precision, equation (3.2.8) results in $N_s N_d$ known values, but for an appropriate choice of ϵ larger than the smallest positive floating-point constant of computational precision, it is certainly possible for this equation to give less than $N_s N_d$ known values in the T-matrix in singular-vector representation.

However many known entries results from equation (3.2.8) summarizes the entirety of our knowledge based solely on the data. But we know these entries are correct in singular-vector space with great certainty. In fact, since these entries fully (or nearly, depending on the choice of ϵ) represent the data, any choice or modification to the unknown values will have negligible impact on the error of the forward equation (3.2.2) when the T-matrix is rotated back to real-space. This is precisely our definition of data-compatability: a T-matrix is called *data-compatabile* if when it is rotated to the basis of singular functions, it agrees with all known values from equation (3.2.8).

In general, we can expect this equation to result in a number of known entries not much less than or equal to $N_s N_d$. However, from inequality (3.2.3), we know that $N_s N_d \ll N_v^2$, which implies that we only know a very small number of the total entries of $\tilde{T}$. For our previous estimated values of these dimensions, we have $N_s N_d / N_v^2 = 1/L^2$, which is certainly a small fraction of known entries for large L. We can arrange the singular values of A and B in descending order, and such the known elements of $\tilde{T}$ will all be contained in the upper-left submatrix of dimension

$M_A \times M_B$, where $M_A \leq N_s$ and $M_B \leq N_d$. This is schematically shown in Figs. 3.2. The region of known elements can be of a general shape contained in this rectangular block, but we will assume it is rectangular. Furthermore, in all numerical simulations, the region was indeed rectangular. It is not complicated to include irregular shapes (see Figure 3.3), but it adds nothing to the discussion.

We can now define the experimental T-matrix T_{exp} as the matrix that satisfies the forward equation (3.2.2) in the minimum norm sense with smallest norm $\|T\|_2$. We can calculate T_{exp} in one of two ways. The first method, which was hinted at before, is to calculate

$$T_{\text{exp}} = A^{+} \Phi B^{+} \ , \tag{3.2.9}$$

where A^{+} and B^{+} are the Moore-Penrose pseudoinverses. Then to obtain $\tilde{T}_{\text{exp}}$ in singular-vector space, one needs to perform the necessary rotation as in equation (3.2.7). An equivalent method is to work directly in singular-vector representation, defining

$$\left(\tilde{T}_{\text{exp}}\right)_{\mu\nu} = \begin{cases} \dfrac{1}{\sigma_\mu^A \sigma_\nu^B} \tilde{\Phi}_{\mu\nu} & , \ \text{if } \sigma_\mu^A \sigma_\nu^B > \epsilon^2 \ ; \\ 0 & , \ \text{otherwise} \ . \end{cases} \tag{3.2.10}$$

which is setting all unknown entries to be identically equal to zero. It is worth noting that even though $\tilde{T}_{\text{exp}}$ is sparse, the same is not necessarily true of T_{exp}. Perhaps even more important to note, is the fact that by this definition T_{exp} is not necessarily symmetric, even though it is theoretically known that the correct T-matrix should be symmetric. However, with later modifications that are not

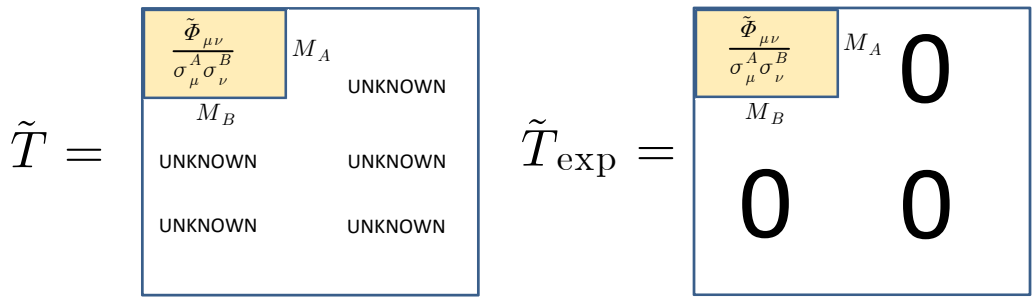


Figure 3.2: **Left panel**: Known elements of $\tilde{T}$ in singular-vector representation computed from the data by using (3.2.8) are organized inside the shaded block in the upper left-hand corner. Elements outside of this shaded block are not known and completely independent of the data. **Right panel**: The experimental T-matrix, T_{exp}, the minimum norm solution to (3.2.2). This is equivalent to setting the unknown elements of $\tilde{T}$ to zero.

inherent to the algorithm, we can easily ensure symmetry. We are now ready to proceed to defining the basics of the iterations.

3.3 Iteration Cycle

There are two main conditions to be met:

(i) The T-matrix must be data-compatible

(ii) The corresponding V-matrix must be diagonally dominated

Our goal is to "complete" the matrix $\tilde{T}$ in singular-vector space by filling in the unknown elements in a way that the corresponding interaction matrix V in real space is diagonally dominant. By using the one-to-one correspondence between the T-matrix and the interaction matrix V, and the invertible rotations between real space and singular-vector space, we can iteratively update an intermediate results

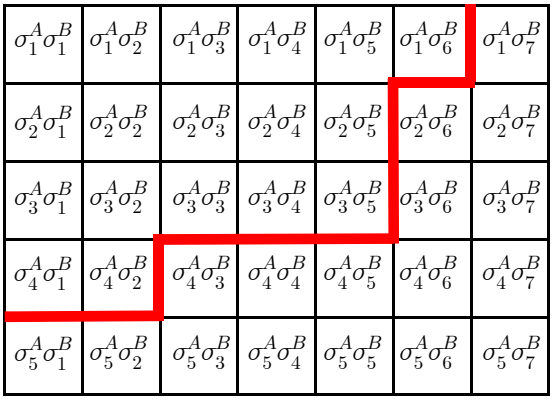


Figure 3.3: A more general shape of known elements of the experimental T-matrix, as compared with Figure 3.2. The elements above the thick red line satisfy the condition $\sigma_\mu^A\sigma_\nu^B > \epsilon^2$. Assuming that the singular values σ_μ^A and σ_μ^B are arranged in the descending order, the boundary line can only go from left to right and from bottom to top if followed from the left-most boundary of the matrix. One can easily obtain a rectangular shape by excluding the elements $\sigma_4^A\sigma_1^B$, $\sigma_4^A\sigma_2^B$, and $\sigma_1^A\sigma_6^B$, but this is not necessary.

by alternatively ensuring conditions (i) and (ii).

We now explicitly define a number of operators that will ease in the description of the algorithm. We let $\mathcal{T}[\cdot]$ be the nonlinear operators that computes the T-matrix from a given interaction matrix by the previous definition (2.3.6)

$$\mathcal{T}[V] = (I - V\Gamma)^{-1}V \ , \tag{3.3.1}$$

which is also invertible (as discussed in Section 2.3) with the form

$$\mathcal{T}^{-1}[T] = (I + T\Gamma)^{-1}T \ . \tag{3.3.2}$$

Note that both of these functionals have Γ as a parameter, and can act on any $N_v \times N_v$ matrix. However, they are only intended to be used for the appropriate V or T-matrix. We also define the rotation transformation $\mathcal{R}[\cdot]$ between real-space

and singular-vector representations and its inverse $\mathcal{R}^{-1}[\cdot]$ by

$$\mathcal{R}[T] = R_A^* T R_B \qquad , \qquad \mathcal{R}^{-1}[\tilde{T}] = R_A \tilde{T} R_B^* \ . \tag{3.3.3}$$

Again, these operators can act on any $N_v \times N_v$ matrix, but have been written with the parameter of intended use. The definition of this operator also presupposes that the rotation matrices have already been calculated, and thus the SVD representation of A and B. With these same assumptions, we define the masking operators $\mathcal{M}[\cdot]$ and $\mathcal{N}[\cdot]$:

$$\left(\mathcal{M}[\tilde{T}]\right)_{\mu\nu} \equiv \begin{cases} 0 \ , & \sigma_\mu^A \sigma_\nu^B > \epsilon^2 \ ; \\ \tilde{T}_{\mu\nu} \ , & \text{otherwise} \ . \end{cases} \qquad \left(\mathcal{N}[\tilde{T}]\right)_{\mu\nu} \equiv \begin{cases} \tilde{T}_{\mu\nu} \ , & \sigma_\mu^A \sigma_\nu^B > \epsilon^2 \ ; \\ 0 \ , & \text{otherwise} \ . \end{cases} \tag{3.3.4}$$

with the point that $\mathcal{M}[\tilde{T}] + \mathcal{N}[\tilde{T}] = \tilde{T}$. Then the method of enforcing data-compatibility of $\tilde{T}$ can be defined through the overwriting operator $\mathcal{O}[\cdot]$ by

$$\mathcal{O}[\tilde{T}] \equiv \mathcal{M}[\tilde{T}] + \tilde{T}_{\text{exp}} = \tilde{T} - \mathcal{N}[\tilde{T}] + \tilde{T}_{\text{exp}} \ . \tag{3.3.5}$$

The operator literally leaves all "unknown" elements unchanged, and forcibly overwrites the "known" entries.

Finally, we define the operator $\mathcal{D}[\cdot]$, which calculates the diagonal approximation to an $N_v \times N_v$ matrix. We will begin by defining this operator by

$$D = (\mathcal{D}[V])_{ij} \equiv \delta_{ij} V_{ii} \ , \tag{3.3.6}$$

which is perhaps the simplest method of defining a related diagonal matrix by setting all off diagonal terms to zero. The specific overwriting operator $\mathcal{O}[\cdot]$ was chosen

in a similar manner. We can certainly choose alternate definitions for $\mathcal{D}[\cdot]$ and $\mathcal{O}[\cdot]$, which we will later discuss and justify our final choices. For now, overwriting and diagonalizing are done by forcibly changing any entry that is not desired.

With these definitions in hand, we can now elaborate on the iterative algorithm DCTMC. We assume that the SVD decompositions of the geometry matrices A and B have been precomputed, along with the experimental T-matrix $\tilde{T}_{\text{exp}}$. Let $k = 1$, and our initial guess $\tilde{T}_1 = \tilde{T}_{\text{exp}}$. Then the algorithm runs as follows:

1: $T_k = \mathcal{R}^{-1}[\tilde{T}_k]$

This transforms the T-matrix from singular-vector to real-space representation. Both $\tilde{T}_k$ and T_k are data-compatible.

2: $V_k = \mathcal{T}^{-1}[T_k]$

This gives k-th approximation to the interaction matrix V. V_k is data-compatible but not diagonal. Compute the off-diagonal and diagonal norms of V_k. If the ratio of the two is smaller than a predetermined threshold, exit; otherwise, continue to the next step.

3: $D_k = \mathcal{D}[V_k]$

Compute the diagonal approximation to V_k, denoted here by D_k. D_k is diagonal but not data-compatible.

4: $T'_k = \mathcal{T}[D_k]$

Compute the T-matrix that corresponds to the diagonal matrix D_k. Unlike

T_k, T'_k is no longer data-compatible.

5: $\tilde{T}'_k = \mathcal{R}[T'_k]$

Transform T'_k to singular-vector representation. Here $\tilde{T}'_k$ is still not data-compatible.

6: $\tilde{T}_{k+1} = \mathcal{O}[\tilde{T}'_k]$

Advance the iteration index by one and overwrite the elements of $\tilde{T}'_k$ that are known from data with the corresponding elements of $\tilde{T}_{\text{exp}}$. This will restore data-compatibility of $\tilde{T}_{k+1}$. Then go to Step 1.

These steps illustrate a method for iteratively ensuring data-compatibility of the T-matrix and diagonal dominance of the interaction matrix V. A flowchart of these iterations is shown in Figure 3.4. While these enforcements are "exclusive or" in the boolean sense that only one of them is guaranteed to be true at any point in the algorithm, the goal is that convergence will lead to an acceptable result on both fronts.

3.4 Computational Complexity and Shortcuts

In terms of computational complexity, Steps 1, 2, 4, and 5 are dominant, all with complexity $O(N_v^3)$. In this section however, we will present computational shortcuts to reduce the number of steps with complexity $O(N_v^3)$. The first shortcut for fast rotations results in no loss of information, and there is no reason not to use its

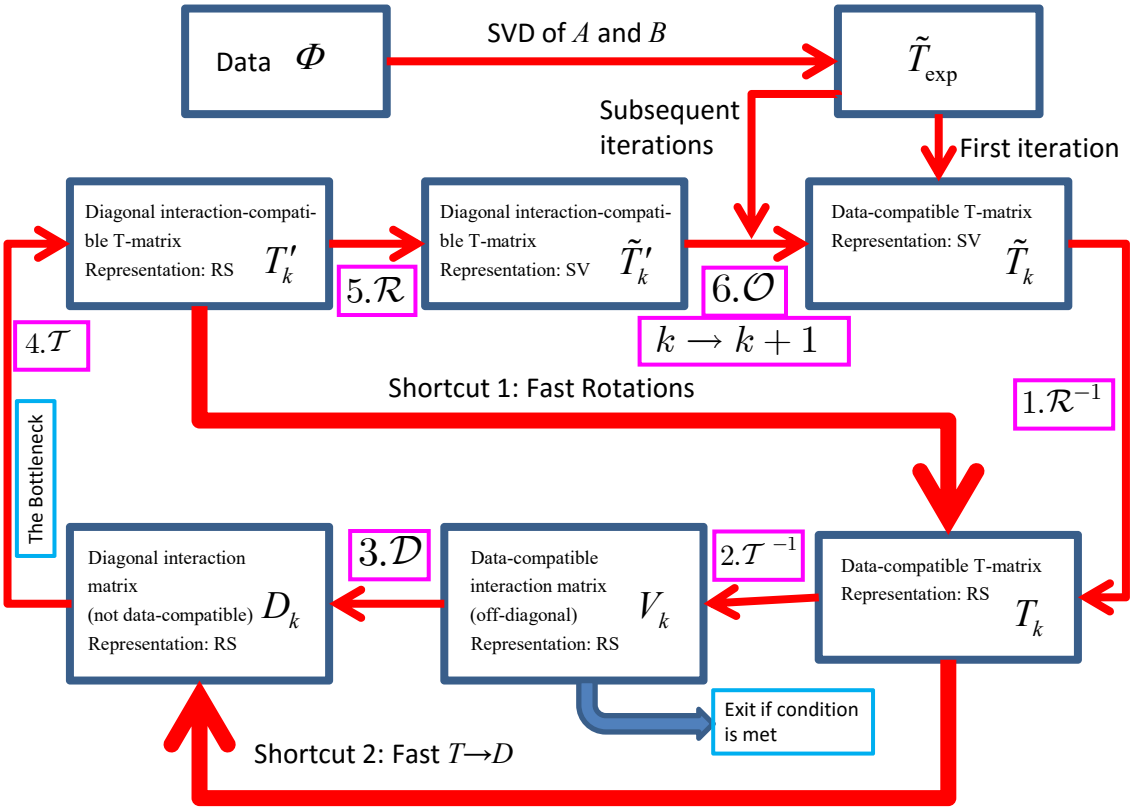


Figure 3.4: Basic flowchart of the DCTMC iteration process for the case when the iterations start with an initial guess of $\tilde{T}_{\mathrm{exp}}$. The operations for the numbered iteration steps are defined in Section 3.3. Matrix representations are abbreviated by SV (singular-vector) and RS (real-space). The computational shortcuts depicted are described in Sec. 3.4. The fact that the size of the data set is not a limiting factor is show by the arrow in the top-left corner.

implementation. The second shortcut which reduces computation time from the T-matrix to the interaction matrix V is certainly useful for the definition presently used of $\mathcal{D}[\cdot]$ in (3.3.6), but may be needed to be eschewed when using alternative definitions for the diagonal matrix approximation operator.

3.4.1 Fast Rotations and Data-Compatibility

The first shortcut combines steps 5, 6, and 1 into one step that enforces data-compatibility by overwriting without rotating all entries to singular-vector space and back. These steps are listed below, with their computational complexity as well as their complexity based on the estimated values with discretization into grid size L.

$$\begin{aligned} 5: &\qquad \tilde{T}'_k = \mathcal{R}[T'_k] &&\qquad O(N_v^3) = O(L^9)\ , \\ 6: &\qquad \tilde{T}_{k+1} = \mathcal{O}[\tilde{T}'_k] &&\qquad \leq O(N_d N_s) = O(L^4)\ , \\ 1: &\qquad T_{k+1} = \mathcal{R}^{-1}[\tilde{T}_{k+1}] &&\qquad O(N_v^3) = O(L^9)\ . \end{aligned}$$

Steps 1 and 5 are obviously the dominant steps, with Step 6 only needing to access and overwrite a maximum of $N_s N_d$ entries (and potentially less depending on desired precision). It is natural to combine these three steps, due to the fact that the bookend rotations are linear. Combining these three steps in one results in

$$T_{k+1} = \mathcal{R}^{-1}\left[\mathcal{O}\left[\mathcal{R}[T'_k]\right]\right] = \mathcal{R}^{-1}\left[\mathcal{R}[T'_k] - \mathcal{N}\left[\mathcal{R}[T'_k]\right] + \tilde{T}_{\text{exp}}\right]\ , \tag{3.4.1}$$

where the second equality has inputted the second definition of $\mathcal{O}[\cdot]$ in (3.3.5) and the fact that $\mathcal{R}[T_{\text{exp}}] = \tilde{T}_{\text{exp}}$. Now we can distribute the operator $\mathcal{R}^{-1}$ across due to linearity, which results in the expression

$$T_{k+1} = T'_k + T_{\text{exp}} - \mathcal{R}^{-1}\left[\mathcal{N}\left[\mathcal{R}[T'_k]\right]\right] \ . \tag{3.4.2}$$

This expression is identical to Steps 5, 6, and 1, but we still need to be able to reduce $\mathcal{R}^{-1}\left[\mathcal{N}\left[\mathcal{R}[T'_k]\right]\right]$ to show any computational improvements. This is not difficult, as (3.4.2) has replaced $\mathcal{O}[\cdot]$ in (3.4.1) with $\mathcal{N}[\cdot]$, which turns any matrix into a sparse matrix. Therefore, it is reasonable to expect that $\mathcal{R}^{-1}\left[\mathcal{N}\left[\mathcal{R}[T'_k]\right]\right]$ can be computed in fewer than $O(N_v^3)$ operations.

Let us first consider the operation $\mathcal{N}[\mathcal{R}[T]]$. As the masking operator $\mathcal{N}$ sets all entries not in the upper left $M_A \times M_B$ submatrix to zero, there is no need to calculate all entries in the rotation $\mathcal{R}[T]$. Therefore we define the $N_v \times M_A$ matrix P_A by the first M_A columns of R_A, and similarly P_B by the first M_B columns of R_B. Then, $\mathcal{N}[\mathcal{R}[T]] = P_A^* T P_B$ which now has computational complexity of $O(\min(M_A, M_B)N_v^2)$ which is a reduction over the original computational complexity of $O(N_v^3)$ by a minimum of $O(L)$ under our estimates. Identical reasoning allows us to expand to the full expression

$$\mathcal{R}^{-1}[\mathcal{N}[\mathcal{R}[T'_k]]] = P_A\left(P_A^* T P_B\right) P_B^* \ , \tag{3.4.3}$$

which has the same computational complexity of $O(\min(M_A, M_B)N_v^2)$. It is clear that $P_A P_A^*$ and $P_B P_B^*$ are not equal to the identity matrix, and are not sparse in

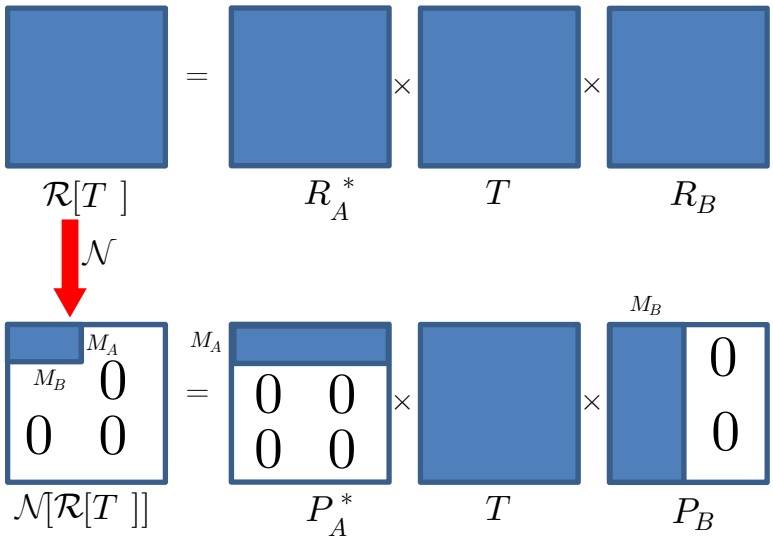


Figure 3.5: Schematics of computing $\mathcal{N}[\mathcal{R}[T]]$. Matrices P_A and P_B are obtained from R_A and R_B by setting all columns to zero except for the first M_A and M_B columns, respectively.

any sense. Therefore, it would be inefficient to premultiply these matrices and set $Q_A = P_A P_A^*$ and $Q_B = P_B P_B^*$ and calculate $Q_A T Q_B$. Using this premultiplication would clearly still have complexity $O(N_v^3)$. Therefore, the evaluation of (3.4.3) should be calculated in the order implied by the parentheses to gain an improvement. Putting all of this together, we obtain the shortcut for fast rotations combing Steps 5, 6, and 1 by

$$T_{k+1} = T'_k + T_{\text{exp}} - P_A \left(P_A^* T'_k P_B\right) P_B^* \ . \tag{3.4.4}$$

With this shortcut, it is no longer necessary to store the full matrices of R_A and R_B in memory. We only need to precompute and store P_A and P_B. Since T_{exp} is also precomputed, this shortcut only requires computing the last term.

3.4.2 Fast $T \to D$ Transformation (Option 1)

We now consider another shortcut that will quickly calculate the related interaction matrix V to the T-matrix and simultaneously find its diagonal approximation D. This shortcut can be thought of as not a pure shortcut to the algorithm, but as an alternative definition to the closest diagonal approximation operator $\mathcal{D}[\cdot]$ that runs faster. Our previous definition was to compute V that perfectly corresponds to T, but is most likely not diagonally dominated. Then we set all off diagonal terms to zero to arrive at our diagonal approximation. Put alternatively, we seek a diagonal matrix D that minimizes $\|V - D\|_2$. However, we can also search directly for the nearest diagonal matrix D that corresponds to T. From (2.3.7), we have

$$T = D + D\Gamma T \ , \tag{3.4.5}$$

which we can solve as a classical minimization problems, stated as

$$\min_{D \text{ diagonal}} \|T - D - D\Gamma T\|_2 \ . \tag{3.4.6}$$

We can explicitly solve problem (3.4.6) to arrive at the solution:

$$D_{ij} = \delta_{ij} \frac{T_{ii} + [(\Gamma T)^* T]_{ii}}{1 + [(\Gamma T)^* + (\Gamma T) + (\Gamma T)^* (\Gamma T)]_{ii}} \ . \tag{3.4.7}$$

While we have arrived at a nice closed-form solution to (3.4.6), at first glance it seems like this evaluation still requires $O(N_v^3)$ operations, which is not an improvement over the original method which required the inverse an $N_v \times N_v$ matrix. The issue seems to be the calculation of the product ΓT, however, we are able to iteratively update this product, so as to not require full calculation each step. We define

the matrix $\Lambda = \Gamma T$, and look back at the computational shortcut for fast rotations as stated in (3.4.4). We can multiply both sides of the equation by Γ to obtain

$$\Gamma T_{k+1} = \Gamma T'_k + \Gamma T_{\text{exp}} - \Gamma P_A \left(P_A^* T'_k P_B\right) P_B^* \ , \tag{3.4.8}$$

or using the natural extension of the definition of Λ,

$$\Lambda_{k+1} = \Lambda'_k + \Lambda_{\text{exp}} - \left(\Gamma P_A\right) \left(P_A^* T'_k P_B\right) P_B^* \ . \tag{3.4.9}$$

The matrix ΓP_A can be precomputed at the start, as well as the matrix $\Lambda_{\text{exp}} = \Gamma T_{\text{exp}}$, so the only remaining issues is how to obtain Λ'_k. If this matrix can be easily obtained, then the computational complexity of (3.4.9) is identical to the complexity of the previous shortcut, namely $O(\min(M_A, M_b)N_v^2)$. It is a reasonable question if the computation of Λ'_k requires a new matrix multiplication, due to the fact that $\Gamma T'_k$ has not been used previously. However, it can be precomputed without any additional cost in the previous step of the algorithm. From the definition of $\mathcal{T}[\cdot]$ in (3.3.3), we have

$$T'_k = D_k(I - \Gamma D_k)^{-1} \ . \tag{3.4.10}$$

which results in

$$\Lambda'_k = \Gamma D_k(I - \Gamma D_k)^{-1} = (I - \Gamma D_k)^{-1} - I \ , \tag{3.4.11}$$

when both sides of (3.4.10) are left-multiplied by Γ. Thus,as we already need to calculate this inverse matrix, we are able to obtain Λ'_k simply by subtracting off of the diagonal. We spell out this procedure to obtain Λ'_k exactly as follows:

1: Compute the product $\Delta_k \equiv \Gamma D_k$, which is fast because D_k is diagonal.

2: Compute the inverse $S_k \equiv (I - \Delta_k)^{-1}$, which has the complexity of N_v^3.

3: Compute $\Lambda'_k = S_k - I$ [as follows from (3.4.11)].

4: Compute $T'_k = D_k S_k$ [as follows from (3.4.10)], which is again fast because D_k is diagonal.

Thus, in this shortcut to go directly from T to D, we have eliminated all steps in this process requiring N_v^3 operations, leaving only the calculation in item 2 inverting $(I - \Delta_k)$ as required to go from the interaction matrix V to the T-matrix.

3.4.3 Fast $T \to D$ Transformation (Option 2)

We now present another alternative for the diagonal approximation D_k to an interaction matrix that also allows for computational improvements over the original method. We define the operator $\mathcal{D}[\cdot]$ as:

$$D_{ij} = \begin{cases} \sum_{k=1}^{N_v} V_{ik} & i = j \\ 0 & i \neq j \ , \end{cases} \tag{3.4.12}$$

which sums over all rows to the diagonal. This fits in with the conceptual understanding of DCTMC in terms of working in a nonlocal framework. Returning to the example used in 3.1, the nonlocal version of Ohm's law is generalized from the linear $\mathbf{J}(\mathbf{r}) = \sigma(\mathbf{r})\mathbf{E}(\mathbf{r})$ to the nonlinear $\mathbf{J}(\mathbf{r}) = \int V(\mathbf{r}, \mathbf{r}')\mathbf{E}(\mathbf{r}')d^3r'$. To now return

to *local* conductivity leads to $\sigma(\mathbf{r}) = \int V(\mathbf{r}, \mathbf{r}')d^3r'$. In a discretized setting, this corresponds to summing over rows in the V-matrix.

Now how can this method of finding the diagonal approximation lead to any improvement in computational speed? It seems like we still need to calculate the entire V matrix and then sum over rows as opposed to just discarding all off-diagonal terms. However, we can exploit the fact that we can compress the diagonal matrix D into a column vector containing its diagonal entries. We can obtain the values along the diagonal of D, $|D\rangle$ from V by

$$|D\rangle = V|\mathbf{1}\rangle \ , \tag{3.4.13}$$

where $|\mathbf{1}\rangle$ is the $N_v \times 1$ column vector with all entries equal to one. Thus, the process going from T to D can be written as

$$|D\rangle = (I + T\Gamma)^{-1}T|\mathbf{1}\rangle \ , \tag{3.4.14}$$

or equivalently as

$$(I + T\Gamma)|D\rangle = T|\mathbf{1}\rangle \ , \tag{3.4.15}$$

which can then be solved by a linear solver. This is both faster than explicitly calculating the inverse matrix, as well as being much more stable. We can also remove the matrix multiplication $T\Gamma$ by factoring out Γ and then left-multiplying our result $|D\rangle$ by Γ^{-1}. In all tested applications, the matrix Γ has been computationally invertible (and thus can be precomputed). It is a moot point if Γ is not invertible or of very low rank, as then the inverse matrices can be calculated easily anyways

as in our sparse methods of Section 3.5.5.

3.4.4 Streamlined Iteration Cycles

This section integrates the computational shortcuts into the DCTMC algorithm. In what follows, Option 1 refers to the method in Section 3.4.2, while Option 2 refers to the method explained in Section 3.4.3. The shortcut for fast rotations is always used, which as mentioned, is a pure improvment of DCTMC with no downsides.

Initial setup:

a: Permanently store in memory the analytically-known matrix Γ. **Option 2 only**: Calculate and store in memory Γ^{-1}.

b: Compute the SVD decomposition (3.2.4) of A and B. This will yield a set of singular values σ_μ^A, σ_μ^B (some of which are identically zero) and singular vectors $|f_\mu^A\rangle$, $|f_\mu^B\rangle$, $|g_\mu^A\rangle$, $|g_\mu^B\rangle$.

c: Use the previous result to construct R_A and R_B, and permanently store in memory the submatrices P_A and P_B. R_A and R_B can be discarded and deallocated. **Option 1 only**: Construct and store $\tilde{P}_A = \Gamma P_A$.

d: Compute $\tilde{\Phi}_{\mu\nu}$ according to (3.2.6) and $\tilde{T}_{\text{exp}}$ according to (3.2.10). Discard the real-space data function, the singular values and singular vectors, and deallocate the associated memory.

e: Compute and store permanently in memory $T_{\rm exp} = P_A \tilde{T}_{\rm exp} P_B^*$ and **Option 1 only**: $\mathit{\Lambda}_{\rm exp} = \mathit{\Gamma} T_{\rm exp}$.

f: Initialize iterations by setting $T_1 = T_{\rm exp}$ and **Option 1 only**: $\mathit{\Lambda}_1 = \mathit{\Lambda}_{\rm exp}$.

Main iteration (Option 1): For $k = 1, 2, \ldots$, perform the following computations:

1: $(D_k)_{ij} = \delta_{ij} \dfrac{(T_k)_{ii} + (\mathit{\Lambda}_k^* T_k)_{ii}}{1 + (\mathit{\Lambda}_k^* + \mathit{\Lambda}_k + \mathit{\Lambda}_k^* \mathit{\Lambda}_k)_{ii}}$;

2: $\mathit{\Delta}_k = \mathit{\Gamma} D_k$;

3: $S_k = (I - \mathit{\Delta}_k)^{-1}$;

4: $T'_k = D_k S_k \quad , \quad \mathit{\Lambda}'_k = S_k - I$;

5: $T_{k+1} = T'_k + T_{\rm exp} - P_A \left(P_A^* T'_k P_B\right) P_B^* \quad , \quad \mathit{\Lambda}_{k+1} = \mathit{\Lambda}'_k + \mathit{\Lambda}_{\rm exp} - Q_A \left(P_A^* T'_k P_B\right) P_B^*$

.

Main iteration (Option 2): For $k = 1, 2, \ldots$, perform the following computations:

1: Solve $(\mathit{\Gamma}^{-1} + T_k)|D'_k\rangle = T_k|\mathbf{1}\rangle$ for $|D'_k\rangle$;

2: $|D_k\rangle \ = \mathit{\Gamma}|D'_k\rangle$ Place values onto diagonal of D_k ;

3: $\mathit{\Delta}_k = \mathit{\Gamma} D_k$;

4: $S_k = (I - \mathit{\Delta}_k)^{-1}$;

5: $T'_k = D_k S_k$;

6: $T_{k+1} = T'_k + T_{\exp} - P_A \left(P_A^* T'_k P_B\right) P_B^*$.

Some operations are placed on the same line to emphasize the fact that their execution is insignificant and can run in parallel. A few comments on these iterations:

For Option 1 [1:], the calculation of $(\Lambda_k T_k)_{ii}$ and $(\Lambda_k^* \Lambda_k)_{ii}$ for $i = 1, \ldots, N_v$ has the computational complexity of only $O(N_v^2)$. As noted previously, the overwriting step [5:] has computational complexity of $O(\min(M_A, M_b) N_v^2)$, while steps [2:] and [4:] are fast as D_k is diagonal. Therefore, the only bottleneck is the operation of matrix inversion in [3:], with complexity $O(N_v^3)$. Iterations can be stopped in this case when the norm $\|T_k - D_k - D_k \Gamma T_k\|_2$ is smaller than a predetermined threshold.

For Option 2, both [1:] and [4:] have complexity $O(N_v^3)$, but step [1:] is still faster and more stable as mentioned. All are other steps are identical to Option 1, or inconsequential. Iterations can again be stopped when the norm $\|T_k - D_k - D_k \Gamma T_k\|_2$ is smaller than a predetermined threshold. It is worth noting that both options avoided the troublesome direct inversion required to calculate V from T, a marked improvement over the original statement of DCTMC. The runtime per iteration on a slower workstation for $N_v = 2304$ and $M_A = M_B = 954$ are shown in Table **??**. The relative difference between these methods scales linearly, that is Option 2 runs about twice as fast as not using a faster diagonalization method (but still using fast rotations), while Option 1 always runs a bit (1.25 times) slower than Option 2.

Just Rotation Shortcut	≈ 58s
Option 1 Shortcut	≈ 39s
Option 2 Shortcut	≈ 31s

Table 3.1: Runtimes for different computational shortcuts

3.5 Variations and Improvements

3.5.1 Starting From an Initial Guess

The iteration cycles were written from a starting point of $T_1 = T_{\text{exp}}$. This is in no way a requirement for DCTMC, it is merely the most convenient option as T_{exp} must already be computed. If one has an initial guess for V based on some a priori knowledge, it makes sense to start from this initial guess and continue the algorithm from the appropriate starting point D_1. One option is to use the linearized solution is the initial guess, which can be obtained quickly through (4.1.6). This gives the linearized solution $|v_L\rangle$ where we then set $D_1 = D_L$ that contains the elements of $|v_L\rangle$ on its diagonal. While one must be careful to avoid local minimums using this method, it was shown to be highly effective in our simulations.

3.5.2 Reciprocity

From the general reciprocity principle, data measurements are unchanged if the locations of sources and detectors are interchanged. This is typically an unimportant point for most reconstruction algorithms, as adding these duplicate data points does not contribute any information to the Jacobian and only adds computation time.

In T-matrix methods the reciprocity principle is evident through the symmetry of the T-matrix. However, as noted in Section 3.2, our definition of the experimental T-matrix T_{exp} does not enforce symmetry. Thus, including these data points from interchanging sources and detectors is not redundant, and in fact enforces symmetry of the experimental T-matrix. This in turn also forces all data compatible T-matrices obtained in the iterations to be symmetric. This is clearly useful information for the DCTMC algorithm – we must explicitly enforce the reciprocity principle. Note that these "interchanged" data measurements are trivially obtained, and do not require any additional experimental work. Moreover, while enforcing symmetry of the T-matrix will add some computation time, it is small as the number of data measurements is not a limiting factor in this method. That is, only the values of M_A and M_B will change (the number of known elements in the T-matrix), which will add some extra work to the overwriting step, but is still dwarfed by the full matrix inversions required to obtain T from V. This more complete data set was shown to be crucial in our simulations with noise in Section 5.3.

3.5.3 Regularization

It is well known that inverse scattering problems of interests can be severely ill-posed and noisy. Suitable regularization is typically required to produce reasonable results. There are two main forms of regularization for DCTMC – a linear based version of Tikhonov regularization and regularization by enforcing known physical

constraints. The linear-inspired Tikhonov regularization will be discussed in Section 4.1, whose main idea is regularizing the matrix W in (4.1.3). In combination with this or separately, imposing physical constraints can be applied to the elements of D_k each iteration. This is a way of applying nonlinear information, and is certainly a type or regularization that can prevent convergence towards unwanted solutions. Examples of a priori physical information that can be applied iteratively include knowledge that all elements of D_k be real, be nonnegative, or have nonnegative imaginary part. All of these constraints can easily be applied to each obtained diagonal interaction matrix.

It is also worth pointing out that the choice of ϵ in equation (3.2.10) plays an important role in suppressing noise in the data. This is akin to truncated SVD and can also be viewed as a form of regularization. The choice of ϵ regulates how much we want our solution to perfectly fit the data. Larger values of epsilon reduce the number of known entries in the experimental T-matrix, and allow entries that could be perfectly determined by the data to change. This can be especially valuable in the presence of noisy measurements. In the simulations from Section 5.1, a relatively large value of ϵ could be used as the simulations were without noise. However, as the problem became more nonlinear, smaller values of ϵ were required to prevent unwanted convergence. The simulations in Section 5.3 required much smaller values of ϵ in the presence of noise. Thus, this truncation of the experimental T-matrix is key to regularizing against both noise and instances of strong nonlinearity.

3.5.4 Choice of Diagonal Approximation

We have previously discussed three main options for obtaining the diagonal approximation to the nonlocal interaction matrix V, setting all off diagonal terms to zero, summing over all rows to the diagonal, or solving the minimization problem (3.4.6). In this section, we will focus our attention on the second option, summing over rows as

$$D_{ij} = \begin{cases} \sum_{k=1}^{N_v} V_{ik} & i = j \\ 0 & i \neq j \ . \end{cases} \tag{3.5.1}$$

While this operator is physically inspired by the nonlocal approach of DCTMC, in practice, it is necessary to apply some type of weight function to the sum. We can reasonably expect that if two voxels are far away from each other, there should not be large interaction between them. Thus it can be prudent to forgo adding these elements in the row sum in (3.5.1), or at least heavily suppressing them to avoid adding unwanted computational noise to the diagonal. One wants to define a weight function $w(\mathbf{r}_i, \mathbf{r}_k)$ such that $w(\mathbf{r}_i, \mathbf{r}_i) = 1$ and $w(\mathbf{r}_i, \mathbf{r}_k)$ goes to zero sufficiently fast as $\mathbf{r}_i$ and $\mathbf{r}_k$ move away from each other. Then this diagonal approximation is given by

$$D_{ij} = \begin{cases} \sum_{k=1}^{N_v} w(\mathbf{r}_i, \mathbf{r}_k) V_{ik} & i = j \\ 0 & i \neq j \ . \end{cases} \tag{3.5.2}$$

The addition of this weight function w has no strong impact on the computation time of the full algorithm, but using Option 2 of the streamlined iteration cycle must

be modified. One no longer wants to obtain D from V by right-multiplying V by the column vector of all ones, as this ignores any weighting that takes place. Thus, one must instead weight the T-matrix in equation (3.4.15) to be more diagonally dominated. That is, define a corresponding weighting operator $\mathcal{W}$ and then solve

$$(I + \mathcal{W}[T]\Gamma)|D\rangle = \mathcal{W}[T]|\mathbf{1}\rangle \ . \tag{3.5.3}$$

It is clear that there is no easy correspondence between the choice of w and $\mathcal{W}$, but there is evidence that this second approach using the shortcut of Option 2 can be effective. The choice of $\mathcal{W}$ should be such that the off diagonal terms of T are slightly reduced in a relative manner to the desired weighting definition of w.

3.5.5 Accounting for Sparsity

It is clear that no matter which computational improvement of DCTMC one chooses, the inversion $(I - \Delta_k)^{-1}$ is the computational bottleneck. If some of the elements of D_k are zero, the matrix $\Delta_k = \Gamma D_k$ will have corresponding zero columns, and the matrix $(I - \Delta_k)$ all have corresponding identity columns. The inversion of this matrix can ignore these columns, and if there are p elements of D_k that are zero, computation of $(I - \Delta_k)^{-1}$ requires $O((N_v - p)^3)$ operations.

We cannot reasonably expect in practice that any diagonal elements will be precisely equal to zero, but we can use a threshold to automatically set absolutely small or relatively small diagonal elements of D_k to zero and then invert the subsequently "smaller" matrix. Moreover, if one has a priori knowledge that the target

is p-sparse, one could set the smallest p (or some number less than p) entries of D_k to be identically equal to zero.

3.5.6 DCTMC in the Inverse Regime

DCTMC is a nonlinear solver as it fully utilises the nonlinear relationship between T and V to iteratively ensure data-compatibility of the T-matrix and locality of the interaction matrix V. However, it is interesting to reformulate the problem in terms of V^{-1} and T^{-1}. It is important to highlight that the work in this section presupposes that all inverses will exist (which might not be true), and has less rationale for convergence. Nonetheless, it is an interesting variation under these assumptions, with a potential to be much faster per iteration.

The benefit of working in the inverse regime is that the relationship between V^{-1} and T^{-1} is linear. Inverting (2.3.7), we can obtain

$$T^{-1} = V^{-1} - \Gamma \ , \tag{3.5.4}$$

or written in terms of operators

$$\mathcal{T}[V^{-1}] = T^{-1} + \Gamma \qquad , \qquad \mathcal{T}^{-1}[T^{-1}] = V^{-1} - \Gamma \ . \tag{3.5.5}$$

This gives us an easy method for transforming between V^{-1} and T^{-1}. As we want V to be diagonal, it is clear that the corresponding condition in the inverse regime is for V^{-1} to be diagonal as well. However, it is not obvious how we can define data-compatibility for T^{-1}. In what follows as our definition of data-compatibility

for T^{-1}, it is clear that it will not be the same definition of data-compatibility for T, and these definitions do not correlate perfectly. The formulation of DCTMC in the inverse regime is in no way an identical formulation to the original DCTMC.

First we introduce some notation. We denote by $\tilde{\tau}$ the $M_A \times M_B$ upper left submatrix of the known entries of the experimental T-matrix. We will further assume that $M_A = M_B$. This is not an unrealistic assumption as using reciprocity as in Section 3.5.2 ensures $M_A = M_B$. Nevertheless, a more rudimentary solution for the case $M_A \neq M_B$ is to set $M_A = M_B = \min\{M_A, M_B\}$. Then it is clear that for an appropriate choice of ϵ, we can assume that $\tilde{\tau}$ is invertible.

For data compatibility, we still need to rotate to the singular-vector representation – we want the upper left submatrix of $(\tilde{T}^{-1})^{-1}$ to agree with $\tilde{\tau}$. We note that in the inverse regime

$$\tilde{T}^{-1} = (R_A^* T R_B)^{-1} = R_B^* T^{-1} R_A \ . \tag{3.5.6}$$

Therefore, we redefine the rotations between singular vector and real representations by

$$\mathcal{R}[T^{-1}] = R_B^* T^{-1} R_A \qquad , \qquad \mathcal{R}^{-1}[\tilde{T}^{-1}] = R_B \tilde{T}^{-1} R_A^* \ . \tag{3.5.7}$$

But since we are working in this inverse regime, it is greatly preferred to not actually invert the matrix back, as then this method is no different than the original DCTMC. Instead, we will treat the fundamental unknown $\tilde{T}^{-1}$ as a block matrix which in turn allows us to use known expressions for inverting block matrices. We

write $\tilde{T}^{-1}$ as:

$$\tilde{T}^{-1} = \left[\begin{array}{c|c} X_1 & X_2 \\ \hline X_3 & X_4 \end{array}\right] , \tag{3.5.8}$$

where X_1 is $M_A \times M_A$, X_2 is $M_A \times (N_v - M_A)$, X_3 is $(N_v - M_A) \times M_A$, and X_4 is $(N_v - M_A) \times (N_v - M_A)$ in dimension. Then we can write the inverse as

$$\tilde{T} = (\tilde{T}^{-1})^{-1} = \left[\begin{array}{c|c} C & -CX_2X_4^{-1} \\ \hline -X_4^{-1}X_3C & X_4^{-1} + X_4^{-1}X_3CX_2X_4^{-1} \end{array}\right] , \tag{3.5.9}$$

where $C = (X_1 - X_2X_4^{-1}X_3)^{-1}$. Therefore, we need $C = \tilde{\tau}$, or equivalently

$$X_1 = \tilde{\tau}^{-1} + X_2X_4^{-1}X_3 , \tag{3.5.10}$$

where $\tilde{\tau}^{-1}$ can be precomputed and the inversion X_4^{-1} is faster than full inversion as it is a submatrix. Data compatibility in the inverse regime requires overwriting the upper left submatrix in a nonlinear fashion, which makes sense as we have removed the nonlinearity present between T and V. We have not removed all the nonlinearity from the problem, but the idea is that the computations required in inverse regime require less computation time.

To fully take advantage of all computational improvements in the inverse regime, we will exploit the linear relationship between T^{-1} and V^{-1} to perform all calculations solely on V^{-1}. Conceptually, it makes sense to still think of T^{-1} as the fundamental unknown due to its relationship with data compatibility, however it is not needed to ever explicitly compute the inverse of the T-matrix in this algorithm.

For example, let us look at the steps involved in enforcing data compatibility. This is the computation of $\mathcal{O}[\mathcal{R}[T^{-1}]]$. Using (3.5.4) and (3.5.7), we obtain

$$\begin{aligned}\mathcal{O}[\mathcal{R}[T^{-1}]] =& \mathcal{O}[R_B^* T^{-1} R_A] \\ =& \mathcal{O}[R_B^* V^{-1} R_A - R_B^* \Gamma R_A] \ ,\end{aligned}$$

where it is clear that we only want to overwrite the entries of $R_B^* V^{-1} R_A$, as the other term is constant. Hence, writing Γ and $\tilde{V}^{-1}$ as

$$\Gamma = \left[\begin{array}{c|c} \Gamma_1 & \Gamma_2 \\ \hline \Gamma_3 & \Gamma_4 \end{array}\right] \quad , \quad \tilde{V}^{-1} = \left[\begin{array}{c|c} Y_1 & Y_2 \\ \hline Y_3 & Y_4 \end{array}\right] \ , \tag{3.5.11}$$

it is clear that we can write the overwriting operation purely in terms of V^{-1} as

$$\mathcal{O}[\tilde{V}^{-1}] = \left[\begin{array}{c|c} \tilde{\tau}^{-1} + \Gamma_1 + (Y_2 - \Gamma_2)(Y_4 - \Gamma_4)^{-1}(Y_3 - \Gamma_3) & Y_2 \\ \hline Y_3 & Y_4 \end{array}\right] \ , \tag{3.5.12}$$

where we denote $\tilde{Y}_1 = \tilde{\tau}^{-1} + \Gamma_1 + (Y_2 - \Gamma_2)(Y_4 - \Gamma_4)^{-1}(Y_3 - \Gamma_3)$. The iteration cycle can be written purely in terms of V^{-1} as follows. Starting from an initial guess of D_1^{-1}:

1: $\tilde{V}_k^{-1} = \mathcal{R}[D_k^{-1}]$

This transforms the diagonal inverse V-matrix from real-space representation to singular vector representation. $\tilde{V}_k^{-1}$ is no longer diagonal.

2: $(\tilde{V}_k')^{-1} = \mathcal{O}[\tilde{V}_k^{-1}]$

This overwrites the elements of $\tilde{V}_k^{-1}$ to restore data compatibility.This gives

k-th approximation to the interaction matrix V. V_k is data-compatible but not diagonal.

3: $(V_k')^{-1} = \mathcal{R}^{-1}[(\tilde{V}_k')^{-1}]$

Transforms $(\tilde{V}_k')^{-1}$ back to real space representation. Note that this matrix is data compatible, but not diagonal. Compute the off-diagonal and diagonal norms of V_k. If the ratio of the two is smaller than a predetermined threshold, exit; otherwise, continue to the next step.

4: $D_k^{-1} = \mathcal{D}[(V_k')^{-1}]$

Compute the diagonal approximation to $(V_k')^{-1}$. D_k^{-1} is diagonal but not data-compatible. Replace k with $k+1$ and return to Step 1.

This algorithm in the inverse regime has fewer steps in each iteration as we can define data compatibility directly for the interaction matrix due to its linear relation with the T-matrix.

In terms of actual implementation of this algorithm, the rotation in Step 1 must be performed to obtain Y_2, Y_3, and Y_4 in order to pass along to the overwriting operation. This rotation is the bottleneck of this algorithm. The computation of $R_B^* D^{-1} R_A$ has one fast matrix multiplication as D^{-1} is diagonal, but the second multiplication runs at $O(N_v^3)$. The dominating part of the overwriting in Step 2 is the inversion of $Y_4 - \Gamma_4$ which requires $O((N_v - M_A)^3)$ operations. A similar shortcut to the fast rotations in Section 3.4.1 can be used for Step 3 to dramatically

reduce computation time. Writing this rotation as

$$\begin{aligned}\mathcal{R}^{-1}[(\tilde{V}_k')^{-1}] =& R_B \left[\begin{array}{c|c} Y_1 & Y_2 \\ \hline Y_3 & Y_4 \end{array}\right] R_A^* + R_B \left[\begin{array}{c|c} \tilde{Y_1} - Y_1 & 0 \\ \hline 0 & 0 \end{array}\right] R_A^* \\ =& \tilde{V}_k^{-1} + R_B \left[\begin{array}{c|c} \tilde{Y_1} - Y_1 & 0 \\ \hline 0 & 0 \end{array}\right] R_A^* \ ,\end{aligned}$$

is beneficial as this multiplication is fast due to the sparsity of the remaining matrix. It is worth reminding that there is no shortcut for the initial rotation for DCTMC in the inverse regime as we do not know what we will be overwriting with until after singular vector representation has been obtained. Therefore $R_B^* D^{-1} R_A$ is the bottleneck, but in some ways is a nicer bottleneck then the full inversion required in DCTMC. It is much simpler to make approximations for this matrix multiplication (such as using rank one updates to D^{-1} if not many entries change dramatically each iteration) than it is to approximate the nonlinear relationship between interaction matrix and the T-matrix without fundamentally changing the problem. To make clear the computation time advantages to using this method, compared to the results in Table 3.1, DCTMC in the inverse regime only takes about 23 seconds per iteration. This is nearly three times faster than the original method. However, the advantages or disadvantages of using DCTMC in the inverse regime in terms of convergence are unexplored.

Chapter 4

DCTMC in the Linear Regime

4.1 Formulation of Linearized DCTMC

We now consider the DCTMC algorithm in the linear regime. By linear regime, we imply that $V = T$, or put alternatively, $\Gamma \to 0$. Returning to the iteration cycle described in Section 3.3, we see that by sending $\Gamma \to 0$, steps 2 and 4 are eliminated, and combined with fast rotations from Section 3.4.1 the iteration cycle is reduced to only two steps: diagonalizing and overwriting. In terms of operators, the algorithm in the linear regime looks like:

1: $D_k = \mathcal{D}[T_k]$;

2: $T_{k+1} = D_k + T_{\exp} - (P_A P_A^*) D_k (P_B P_B^*)$.

We first look at the diagonalizing operator $\mathcal{D}[\cdot]$ which discards all off diagonal terms as defined in (3.3.6). Then by applying this operator to Step 2 in the iteration

scheme above combined with the definition $D_{\text{exp}} = \mathcal{D}[T_{\text{exp}}]$, we obtain the one line algorithm

$$D_{k+1} = D_k + D_{\text{exp}} - \mathcal{D}[(P_A P_A^*) D_k (P_B P_B^*)] \ . \tag{4.1.1}$$

This last term can be simplified due to the diagonal nature of D_k where the diagonal entries of $\mathcal{D}[(P_A P_A^*) D_k (P_B P_B^*)]$ can be obtained in vector form by

$$|\mathcal{D}[(P_A P_A^*) D_k (P_B P_B^*)]\rangle = |v_k'\rangle = W|v_k\rangle \ , \tag{4.1.2}$$

where $|v_k\rangle$ contains the diagonal entries of D_k and W is formed by

$$W = (P_A P_A^*) \circ (P_B P_B^*)^T \ , \tag{4.1.3}$$

where $\circ$ represents the Hadamard product (entrywise multiplication). Equivalently stated as $W_{ij} = (P_A P_A^*)_{ij} (P_B P_B^*)_{ji}$. Now (4.1.1) can be reduced to the vector form

$$|v_{k+1}\rangle = |v_{\text{exp}}\rangle + (I - W)|v_k\rangle \ . \tag{4.1.4}$$

Started from $|v_1\rangle = |v_{\text{exp}}\rangle$, this iteration can be written as

$$|v_{k+1}\rangle = \sum_{j=0}^{k-1} (I - W)^j |v_{\text{exp}}\rangle \tag{4.1.5}$$

This is Richardson first-order iteration which clearly converges if $|1 - w_n| < 1$ for all n where w_n are the eigenvalues of W. If this condition holds, the iteration converges to $|v_\infty\rangle = W^{-1}|v_{\text{exp}}\rangle$. Thus, DCTMC in the linear regime provides an iterative scheme for solving the linear equation

$$W|v\rangle = |v_{\text{exp}}\rangle \ . \tag{4.1.6}$$

4.2 Analysis of Linearized DCTMC

An interesting question is how is this equation relates to the standard method of linearizing these ISPs. Recall that the typical linearization procedures assume that V is strictly diagonal and also place the diagonal entries of V into the vector $|v\rangle$. The data matrix Ψ is then unrolled into a vector $|\psi\rangle$ by stacking the columns into one vector of length $N_d N_s$. The main equation to be solved is

$$K|v\rangle = |\psi\rangle \ , \tag{4.2.1}$$

where we have used the first Born approximation (see Section 5.1), and we recall that K is formed by combining the matrices A and B in the manner of $K_{(mn),j} = A_{mj}B_{jn}$ where (mn) is a composite index. DCTMC already takes into account this separable structure of K, as W is formed by already separating K into A and B. But it is interesting to investigate how we can derive W from K.

Using the singular value decompositions of A and B from (3.2.2), we can express the elements of K as

$$K_{(mn),j} = \sum_{\mu=1}^{N_d} \sum_{\nu=1}^{N_s} \sigma_\mu^A \sigma_\nu^B \langle m|f_\mu^A\rangle \langle g_\mu^A|j\rangle \langle j|f_\nu^B\rangle \langle g_\nu^B|n\rangle \ . \tag{4.2.2}$$

We also define the unitary matrix U by the entrywise definition

$$U_{(\mu\nu),(mn)} = \langle f_\mu^A|m\rangle \langle n|g_\nu^B\rangle \ , \quad 1 \le \mu, m \le N_d \ , \ 1 \le \nu, n \le N_s \ . \tag{4.2.3}$$

Left-multiplying the linear equation (4.2.1) by this unitary matrix will not change its Tikhonov regularized pseudoinverse solution. This is a direct consequence of the

fact that $(UK)^*(UK) = K^*K$ and $(UK)^*U = K^*$. Therefore we are now interested in the linear equation

$$(UK)|\upsilon\rangle = U|\phi\rangle \ , \tag{4.2.4}$$

where we can express the elements of the matrix UK as

$$(UK)_{(\mu\nu),j} = \sigma_\mu^A \sigma_\nu^B \langle g_\mu^A|j\rangle\langle j|f_\nu^B\rangle \ . \tag{4.2.5}$$

Returning back to our work on the experimental T-matrix in (3.2.6), we can also see that $\langle(\mu\nu)|U|\phi\rangle = \tilde{\Phi}_{\mu\nu}$ which allows us to make numerous substitutions to (4.2.4) to obtain

$$\sigma_\mu^A \sigma_\nu^B \sum_{j=1}^{N_v} \langle g_\mu^A|j\rangle\langle j|f_\nu^B\rangle\langle j|\upsilon\rangle = \tilde{\Phi}_{\mu\nu} \ , \quad 1 \leq \mu \leq N_d \ , \quad 1 \leq \nu \leq N_s \ . \tag{4.2.6}$$

Now just as in (3.2.8) where we had to introduce the thresholding parameter of ϵ to account for the fact that very small singular values may create numerical instabilities, we will equally discard equations from (4.2.6) with zero or very small coefficients. We will only keep the equations for which $\sigma_\mu^A \sigma_\nu^B > \epsilon^2$ which will restrict the range on μ and ν by

$$\sigma_\mu^A \sigma_\nu^B \sum_{j=1}^{N_v} \langle g_\mu^A|j\rangle\langle j|f_\nu^B\rangle\langle j|\upsilon\rangle = \tilde{\Phi}_{\mu\nu} \ , \quad 1 \leq \mu \leq M_A \ , \quad 1 \leq \nu \leq M_B \ , \tag{4.2.7}$$

where M_A and M_B are identical to their definitions in Section 3.2. We highlight that this threshold discarding does not computationally affect the pseudoinverse solution. Up to this point, the numerical solution of (4.2.7) is equivalent to the numerical solution of (4.2.1). We now introduce a transformation that can alter

the solution. We precondition (4.2.7) by diagonal scaling by dividing each equation by $\sigma_\mu^A \sigma_\nu^B$. Thus, using the definition of $\tilde{T}_{\exp}$ as in (3.2.8), we obtain

$$\sum_{j=1}^{N_v} \langle g_\mu^A | j \rangle \langle j | f_\nu^B \rangle \langle j | v \rangle = \left(\tilde{T}_{\exp} \right)_{\mu\nu} , \quad 1 \le \mu \le M_A , \quad 1 \le \nu \le M_B . \tag{4.2.8}$$

Note that due to our previous thresholding, this diagonal scaling is stable and is in fact invertible. Therefore, if the original linear equation (4.2.1) is invertible (which as previously mentioned is most certainly not for problems of interest), then its solution and the solution of (4.2.8) are identical. However, if (4.2.1) is not invertible, then these two equations no longer have equivalent pseudoinverse solutions. That is, this last diagonal scaling operation has invalidated the exact equivalence between traditional linearization and DCTMC in the linear regime.

We continue our search for the exact relationship between W amd K, by defining the matrix Q by $Q_{(\mu\nu),j} = \langle g_\mu^A | j \rangle \langle j | f_\nu^B \rangle$ and the column vector τ by $\langle (\mu\nu) | \tau \rangle = \left(\tilde{T}_{\exp} \right)_{\mu\nu}$. We substitute these expressions into (4.2.8) to obtain the neatly written linear equation

$$Q | v \rangle = | \tau \rangle . \tag{4.2.9}$$

Now recall that the columns of P_A in (4.1.3) are the singular vectors $|g_\mu^A\rangle$ for $\mu = 1, \ldots, M_A$ and likewise the columns of P_B are the singular vectors $|f_\nu^B\rangle$ for $\nu = 1, \ldots, M_B$. Thus, we can write the entrywise definition of W by

$$W_{ij} = \sum_{\mu=1}^{M_A} \sum_{\nu=1}^{M_B} \langle i | g_\mu^A \rangle \langle g_\mu^A | j \rangle \langle j | f_\nu^B \rangle \langle f_\nu^B | i \rangle . \tag{4.2.10}$$

Comparing this to the definition of Q, we see the important fact that $Q^* Q = W$.

Similarly, we can see that

$$\langle i|Q^*|\tau\rangle = \sum_{\mu_1}^{M_A}\sum_{\nu=1}^{M_B} Q^*_{i,(\mu\nu)}\left(\tilde{T}_{\exp}\right)_{\mu\nu} = (P_A\tilde{T}_{\exp}P_B^*)_{ii} = (T_{\exp})_{ii} = \langle i|v_{\exp}\rangle \ , \quad (4.2.11)$$

or $Q^*|\tau\rangle = |v_{\exp}\rangle$. Therefore left multiplying both sides of (4.2.9) by Q^* results in the linear equation (4.2.1) from DCTMC in the linear regime. Letting Θ be the diagonal matrix with entries $1/(\sigma_\mu^A\sigma_\nu^B)$ for composite index $1 \leq (\mu\nu) \leq M_AM_B$ and zeros elsewhere, we see that $Q = \Theta UK$. Thus, the relationship between W and K is given by

$$W = Q^*Q = (\Theta UK)^*(\Theta UK) = K^*U^{-1}\Theta^2UK \ , \quad (4.2.12)$$

which, as previously mentioned, is a preconditioned form of the original linear equation.

Also noteworthy is the fact that as $W = Q^*Q$, Tikhonov regularization of (4.1.6) is accomplished by the transformation $W \to W + \lambda^2 I$. It is worth reminding again that this Tikhonov regularization is not equivalent to Tikhonov regularization of (4.2.1). However, this is a reasonable method of solving the linearized problem.

Now how does this Tikhonov regularization relate to the actual iterations of DCTMC in the linear regime? The substitution $W \to W + \lambda^2 I$ changes the over-writing step (Step 2 at the beginning of this section) by

2: $T_{k+1} = D_k - \lambda^2 D_k + T_{\exp} - (P_AP_A^*)D_k(P_BP_B^*)$.

This regularization of the DCTMC algorithm in the linear regime can be naturally extended to the general nonlinear case by replacing the linear transformation in

the overwriting operation $(P_A P_A^*)T_k(P_B P_B^*)$ by $(P_A P_A^*)T_k(P_B P_B^*) + \lambda^2 \mathcal{D}[X]$. This regularization alters Step 5 of the general algorithm for Option 1 (Step 6 for Option 2) in Section 3.4.4 by

$$\begin{aligned} 5:\ T_{k+1} &= T'_k - \lambda^2 \mathcal{D}\left[T'_k\right] + T_{\exp} - P_A \left(P_A^* T'_k P_B\right) P_B^* \ , \\ \Lambda_{k+1} &= \Lambda'_k - \lambda^2 \mathcal{D}\left[\Lambda'_k\right] + \Lambda_{\exp} - Q_A \left(P_A^* T'_k P_B\right) P_B^* \ . \end{aligned}$$

We have so far answered the question of regularization of DCTMC in the linear regime and its relationship to the standard linearization method. We now return to the question of convergence of equation (4.1.6) using the Richardson first-order iteration in (4.1.4). We remark that this numerical method of solving (4.1.6) is not the most efficient one as conjugate gradient descent would be expected to perform better. So while DCTMC in the linear regime is not expected to be used in practice, the analysis of convergence and regularization give important insight into these properties for the nonlinear case.

As previously mentioned, the iterations converge if the eigenvalues of W, w_n, all satisfy the inequality $|1 - w_n| < 1$. As W is Hermitian, all of its eigenvalues are real which reduces the convergence criteria to $0 < w_n < 2$. We prove that in fact the eigenvalues of W satisfy the inequality

$$0 \leq w_n \leq 1 \ . \tag{4.2.13}$$

We return to the expression used for W in (4.2.10). We let $|x\rangle$ be an arbitrary $N_v \times 1$ nonzero vector, and X the diagonal matrix containing the elements of $|x\rangle$.

Then,

$$\langle x|W|x\rangle = \sum_{\mu=1}^{M_A}\sum_{\nu=1}^{M_B}\left|\langle g_\mu^A|X|f_\nu^B\rangle\right|^2 \geq 0\ . \tag{4.2.14}$$

Using the orthonormal properties of the singular vectors, we can also obtain

$$\langle x|x\rangle = \sum_{i=1}^{N_v}\langle i|X^*X|i\rangle = \sum_{\mu=1}^{N_v}\sum_{\nu=1}^{N_v}\left|\langle g_\mu^A|X|f_\nu^B\rangle\right|^2\ , \tag{4.2.15}$$

which shows that $\langle x|W|x\rangle \leq \langle x|x\rangle$ as $M_a, M_b \leq N_v$. Thus, as $0 \leq \langle x|W|x\rangle \leq \langle x|x\rangle$, we have proved (4.2.13).

If there exists an $|x\rangle$ such that $\langle x|W|x\rangle = \langle x|x\rangle$, this implies that $M_A = M_B = N_v$. This implies that all elements of the experimental T-matrix are determined directly from the data, which is quite unrealistic. As the T-matrix is perfectly determined, no iterations are conducted as the initial guess is the fixed point ($W = I$).

If there exists an $|x\rangle$ such that $0 = \langle x|W|x\rangle$, then clearly there will be no convergence as there is an eigenvalue exactly equal to zero. In this case, the natural substitution $W \to W + \lambda^2 I$ will obtain convergence. Furthermore, this substitution will also aid in slow convergence when W has a small eigenvalue. Recall that this substitution was previously proved to be exactly Tikhonov regularization. We can also define the characteristic overlap of singular vectors related to detectors and sources as

$$\xi = \inf_{X\neq 0}\left\{\frac{\sum_{\mu=1}^{M_A}\sum_{\nu=1}^{M_B}\left|\langle g_\mu^A|X|f_\nu^B\rangle\right|^2}{\sum_{\mu=1}^{N_v}\sum_{\nu=1}^{N_v}\left|\langle g_\mu^A|X|f_\nu^B\rangle\right|^2}\right\}\ , \tag{4.2.16}$$

where the iterations in (4.1.4) will converge at least as fast as the power series $\sum_n(1-\xi)^n$. This shows how Tikhonov regularization can increase convergence

speed. The application of the extension of this Tikhonov regularization to nonlinear DCTMC was shown to be effective in simulations.

Chapter 5

Simulations and Results

While the previous chapters have developed and analyzed the DCTMC algorithm, we now investigate its use in several large-scale simulations. We begin in Section 5.1 with simulations of three-dimensional scalar wave diffraction experiments, and then move on to significantly improving these results in Section 5.2. Finally, we tackle a more difficult problem that simulates experimental diffuse optical tomography data in Section 5.3. The overall conclusion from this chapter is that DCTMC is a viable nonlinear solver that is able to handle large noisy data sets.

5.1 Three-dimensional Scalar Wave Diffraction

A first investigates three-dimensional scalar wave diffraction simulations, that are applicable to ultrasound imaging [15, 26] and seismic wave imaging [28, 29, 45]. As mentioned previously, the method of DCTMC can be applied to a wide range of

inverse problems, and is not limited to inverse scattering for scalar waves. However the first tests of DCTMC were conducted for this problem to study its capabilities and merits without the complications associated with other imaging modalities. For example, focusing on scalar propagating waves ignores the vector representations necessary in the electromagnetic scattering. In Section 5.3, we will investigate DCTMC in a more complicated setting, namely diffuse optical tomography. Those simulations incur more severe ill posedness due to the exponential decay of the waves involved. In that section, we will also add noise to further complicate the simulations. But for now, we are merely interested in whether DCTMC will converge, and how it will behave as a solver of inverse scattering problems with significant nonlinearity.

To that end, the reconstructions in this section will mainly be compared to their linear counterparts. We are interested if DCTMC is a viable nonlinear solver – which is only valuable if a linear solution cannot produce acceptable results. As discussed in Chapter 4, we have two methods for obtaining linearized solutions. We can either solve equation (4.2.1) using the matrix K, or we can solve the preconditioned equation (4.1.6) using W. In both cases, the equation will be Tikhonov-regularized and then solved by conjugate gradient descent. Recall that (4.1.6) is inspired by DCTMC in the linear regime, but we will not actually solve it using its slower Richardson iterative process.

Using the traditional approach of computing the pseudoinverse of K can be

problematic as the size of the problem increases. For the largest reconstructions we conducted, the matrix K contains 28,149,336,000 entries which requires at least 224Gb of RAM in double precision (which was used for all reconstructions). There are methods which can compute the product K^*K without storing the entire matrix K (as in [6]), but this multiplication can still take a considerable amount of time. From here, the Tikhonov regularization substitution $K^*K \to K^*K+\lambda^2 I$ and solving the resulting system of equations is reasonable.

However, directly forming the positive-definite matrix W and Tikhonov regularizing it by the operation $W \to W + \lambda^2 I$ is a much faster process. One never has to deal with any matrices that are larger than $N_v \times N_v$. "Separating" the matrix K into the smaller matrices A and B is more computationally efficient, as computing W only requires the singular vectors from these matrices of smaller dimension. Thus, this DCTMC inspired linear approach can be significantly advantageous over the traditional approach. While this preconditioning does change the results, our tests show that any difference between these linearized results is negligible. This evidence suggests that DCTMC in the linear regime has merits. All shown linearized reconstructions are the result of using this DCTMC inspired method with the matrix W.

As a last note about the linear reconstructions used, we tested three conventional linearization methods – the first Born, first Rytov, and mean-field approximations. This has no impact on the choice of using K or W, these approximations only affect

the data matrix, by the transformation

$$AVB = \Psi[\Phi] \ , \tag{5.1.1}$$

as in (5.1.1). These linearization approaches are all reviewed in [10]. The most natural approach to linearizing (as was done previously for DCTMC in the linear regime) is to send $\Gamma \to 0$. This is precisely the first Born approximation which is valid if $\|V\Gamma\|_2 \ll 1$ and makes the approximation

$$A(I - V\Gamma)^{-1}VB \approx AVB \ . \tag{5.1.2}$$

This is trivial in terms of linearizing the data function, so in this case $\Psi = \Phi$.

Neither the first Rytov nor the mean-field approximation are as simple to write as general expressions. For these, we will define a third geometry matrix C which is restricted directly between source and detector pairs. That is, $C_{ij} = G_0(\mathbf{r}_i, \mathbf{r}_j)$ and $\mathbf{r}_i \in \Sigma_d$, $\mathbf{r}_j \in \Sigma_s$ This restriction is simply the incident field produced by a source located at $\mathbf{r}_i$ and measured at $\mathbf{r}_j$ (i.e. $V = 0$). The first Rytov approximation is then

$$\Psi_{ij} = C_{ij} \log\left(1 + \Phi_{ij}/C_{ij}\right) \ , \tag{5.1.3}$$

while the mean-field approximation uses entrywise harmonic average as

$$\Psi_{ij} = \frac{1}{1/\Phi_{ij} + 1/C_{ij}} \ . \tag{5.1.4}$$

In general, there was no distinct advantage seen using any particular one of these three approximations, so the majority of cases only show the first Born approximated results to ease calculation.

5.1.1 Discretization

We will first detail the discretization process for the scalar wave equation, and how it fits into the DCTMC framework. We begin with a scalar field $u(\mathbf{r})$ and the wave equation that it satisfies

$$\left[\nabla^2 + k^2\epsilon(\mathbf{r})\right] u(\mathbf{r}) = -4\pi k^2 q(\mathbf{r}) \ . \tag{5.1.5}$$

Here $q(\mathbf{r})$ is the source and assume $\epsilon(\mathbf{r}) = 1$ outside of the bounded domain of our sample, Ω. We further make the assumption that $k = \omega/c$ is fixed, where c is the scalar wave velocity in free space. We are thus working in the frequency domain. Note that the factor $-4\pi k^2$ has been added to the standard wave equation to ease our future calculations. Our goal is to reconstruct the values of $\epsilon(\mathbf{r})$ from measurements of the field $u(\mathbf{r})$ taken outside of Ω. Our discretization process is based on the discrete dipole approximation for Maxwell's equations [23, 42]. A similar method will be used for the scalar diffusion equation in Section 5.3.

We define the susceptibility of the medium by

$$\chi(\mathbf{r}) \equiv \frac{\epsilon(\mathbf{r}) - 1}{4\pi} \ , \tag{5.1.6}$$

which transforms (5.1.5) to

$$\left(\nabla^2 + k^2\right) u(\mathbf{r}) = -4\pi k^2 \left[\chi(\mathbf{r})u(\mathbf{r}) + q(\mathbf{r})\right] \ . \tag{5.1.7}$$

Now we can obtain the well-known Lippmann-Schwinger equation by inverting the operator on the left-hand side of (5.1.7), namely

$$u(\mathbf{r}) = \int G_0(\mathbf{r}, \mathbf{r}')q(\mathbf{r}')d^3r' + \int G_0(\mathbf{r}, \mathbf{r}')\chi(\mathbf{r}')u(\mathbf{r}')d^3r' \ , \tag{5.1.8}$$

where $G_0(\mathbf{r}, \mathbf{r}')$ is the free-space Green's function of the wave equation which solves

$$\left(\nabla^2 + k^2\right) G_0(\mathbf{r}, \mathbf{r}') = -4\pi k^2 \delta(\mathbf{r} - \mathbf{r}') \ . \tag{5.1.9}$$

This solution can be explicitly computed through contour integration to be

$$G_0(\mathbf{r}, \mathbf{r}') = k^2 \frac{\exp\left(ik|\mathbf{r} - \mathbf{r}'|\right)}{|\mathbf{r} - \mathbf{r}'|} \ . \tag{5.1.10}$$

Note that in (5.1.8), the first term is the incident field while the second term is the scattered field. That is, $u = u_{\text{inc}} + u_{\text{scatt}}$, where

$$u_{\text{inc}}(\mathbf{r}) = \int G_0(\mathbf{r}, \mathbf{r}') q(\mathbf{r}') d^3 r' \ , \tag{5.1.11}$$

and

$$u_{\text{scatt}}(\mathbf{r}) = \int G_0(\mathbf{r}, \mathbf{r}') \chi(\mathbf{r}') u(\mathbf{r}') d^3 r' \ . \tag{5.1.12}$$

Discretizing the Lippman-Schwinger equation (5.1.8) into N_v cubic voxels, each of volume h^3, allows us to reduce the problem to having finite number of degrees of freedom. We will assume our sample is rectangular (as in Figure 2.1) and denote our collection of N_v voxels by $\mathbb{V}_n$. To now be able to solve the Lippman-Schwinger equation, we will further assume that the susceptibility quantities $\chi(\mathbf{r})$ and the field $u(\mathbf{r})$ are constant across each voxel. That is,

$$\chi(\mathbf{r}) = \chi_n \quad \text{AND} \quad u(\mathbf{r}) = u_n \quad \text{IF} \quad \mathbf{r} \in \mathbb{V}_n \ . \tag{5.1.13}$$

This is a reasonable approximation, but certainly never exact. This approximation allows us to extract a finite set of equations from (5.1.8) by focusing on the center

of each voxel $\mathbf{r}_n$. As the value of interest is constant in each voxel, we can solve for the susceptibility for each $\mathbf{r}_n$ ($n = 1, \ldots, N_v$). The obtained equations are of the form

$$u_n = u_{\text{inc}}(\mathbf{r}_n) + \sum_{m=1}^{N_v} \chi_m u_m \int_{\mathbf{r}\in\mathbb{C}_m} G_0(\mathbf{r}_n, \mathbf{r}) d^3r \ . \tag{5.1.14}$$

Under our reasonable assumptions, we expect (5.1.14) to have a solution for the discrete field u_n. What is left to evaluate in (5.1.14) are the integrals on the right-hand side. There are two methods to calculate these integrals. The less-popular method is to directly compute each integral numerically. While this is certainly possible, the previous approximations render this extra work of limited practical use. Much more common is to use the known analytical expressions for G_0 (as in (5.1.10)) to further approximate the integrals in a discrete process.

For $\mathbf{r}_n \notin \mathbb{V}_m$ (i.e. $n \neq m$), we use the conventional approximation

$$\int_{\mathbf{r}\in\mathbb{C}_m} G_0(\mathbf{r}_n, \mathbf{r}) d^3r \approx h^3 G_0(\mathbf{r}_n, \mathbf{r}_m) \ , \quad n \neq m \ . \tag{5.1.15}$$

However, the case $n = m$ must be treated separately as the integrand contains a singularity when the denominator in (5.1.10) equals zero. We will thus evaluate the integral under the approximation

$$\exp\left(ik|\mathbf{r} - \mathbf{r}'|\right) \approx 1 + ik|\mathbf{r} - \mathbf{r}'| \ . \tag{5.1.16}$$

The validity of using this approximation requires the assumption that $kh \ll 1$. However, if kh is not small, than all of our previous assumptions are invalid as the field will not be relatively constant across each voxel. Substituting (5.1.16) into

(5.1.10) now allows us to evaluate the integral for the case $n = m$ as

$$\int_{\mathbf{r}\in\mathbb{C}_n} G_0(\mathbf{r}_n, \mathbf{r}) d^3r \approx \int_{-h/2}^{h/2} dx \int_{-h/2}^{h/2} dy \int_{-h/2}^{h/2} dz \left(\frac{k^2}{\sqrt{x^2+y^2+z^2}} + ik^3 \right)$$
$$= (kh)^2 (\xi + ikh) \ , \tag{5.1.17}$$

where the value of ξ is

$$\xi = \log\left(26 + 15\sqrt{3}\right) - \pi/2 \approx 2.38 \ . \tag{5.1.18}$$

This "volume" parameter ξ is completely dependent on the shape of the voxels used. While we are focused on a cubic voxelization scheme, this integration would be evaluated over a ball if it were more appropriate to discretize the sample into spherical regions. In this case, $\xi = (9\pi/2)^{1/3} \approx 2.42$ (which can clearly be calculated from a much simpler integral).

Note that the value of ξ has no impact on the imaginary part of the integral, $(kh)^3$. This term is known as the first non-vanishing radiative correction. Including higher terms in the approximation of (5.1.16) would result in higher order dynamic corrections. This first correction is the most important as it is well known in electromagnetics that it ensures energy conservation of the scattering process [22, 24, 34].

With these discretized integral calculations, we are now able to rewrite (5.1.14) in a fully discretized manner. We first introduce the "moments" d_n, where

$$d_n \equiv h^3 \chi_n u_n \ , \tag{5.1.19}$$

and also denote the discretized incident field by e_n:

$$e_n \equiv u_{\text{inc}}(\mathbf{r}_n) \ . \tag{5.1.20}$$

Now, (5.1.14) takes on the especially tidy form

$$d_n = \alpha_n \left(e_n + \sum_{m=1}^{N_v} \Gamma_{nm} d_m \right) \ , \tag{5.1.21}$$

where the quantity of interest is now the polarizability of each voxel α_n, where

$$\alpha_n = \frac{h^3 \chi_n}{1 - (kh)^2(\xi + ikh)\chi_n} \ . \tag{5.1.22}$$

The interaction matrix Γ contains the approximated integral values with zeros on its diagonal. It is entrywise defined as

$$\Gamma_{nm} = (1 - \delta_{nm})\, G_0(\mathbf{r}_n, \mathbf{r}_m) \ . \tag{5.1.23}$$

Equation (5.1.21) is our discretized forward problem. The relationship between the moments d_n and the scattered field u_{scatt} is the discretization of definition (5.1.12), namely

$$u_{\text{scatt}}(\mathbf{r}_d) = \sum_{n=1}^{N_v} G_0(\mathbf{r}_d, \mathbf{r}_n) d_n \ . \tag{5.1.24}$$

Thus, if the values of the polarizabilities α_n are known, the forward problems states that given the incident field at each voxel e_n, the moments are given by (5.1.21). Then, using the relationship between the moments and the scattered field in (5.1.24), the scattered field at any detector location $\mathbf{r}_d$ can be calculated. This is the (linear) forward problem that gives the scattered field.

The inverse problem is to use measurements of the scattered field from outside the sample to recover the values of α_n. Then (5.1.22) can be inverted to obtain χ_n by

$$\chi_n = \frac{\alpha_n/h^3}{1 + (kh)^2(\xi + ikh)(\alpha_n/h^3)} \ , \tag{5.1.25}$$

with the value of ϵ_n obtained with one more simple step. Before continuing with the manipulations necessary to state the inverse problem, let us first rationalize the choice of the fundamental unknown α_n as opposed to using χ_n or ϵ_n. It is true that the voxel permittivity or voxel susceptibility are the actual physical quantities of interest, but there are advantages to treating the polarizabilities as the fundamental unknown. The preference of χ_n over ϵ_n is clear as this transition was necessary to obtain a neat version of the Lippman-Schwinger equation. Moreover, since the relationship between χ_n and α_n is nonlinear, we are in essence removing some of the nonlinearity from the problem analytically. In fact, (5.1.25) is a scalar wave form of the Maxwell-Garnett formula, while its inverse (5.1.22) is an analog to the Clausius-Mossotti relation [48]. The nonlinearity we are removing is the self-interaction present in a fixed voxel, which was partially removed by renormalizing the polarizability. That is, if the corrective term from the denominator of (5.1.22) vanished, we would be left with just $h^3\chi_n$ which is the bare polarizability. Adding in this corrective term (which was calculated from the integrand of self-interaction) renormalizes the polarizability and removes some of the nonlinearity. It is clear that the nonlinearity of the ISP that is caused by interaction across different voxels can

not be removed, which is why we still require nonlinear solvers.

As a simple example of how the renormalization removes some nonlinearity, let us look at a sample of only one voxel. Even though we would not require an iterative method to determine the polarizability or susceptibility of this voxel, we will consider the case of using such a scheme. Treating the polarizability α as the fundamental unknown results in $A\alpha = b$ (where A and b are known), which is a linear equation that only requires one iteration. Treating the more direct χ as the fundamental unknown actually results in a nonlinear equation, namely $A\chi/(1 - \beta\chi) = b$, which can require several iterations.

Thus, we have given some justification for the choice of the polarizabilities as the fundamental unknown. We now return to the full statement of the discretized inverse problem. We acquire multiple measurements of the scattered field $u_{\text{scatt}}(\mathbf{r}_d)$ at N_d detector locations ($\mathbf{r}_{dk}$ for $k = 1, \ldots, N_d$). These measurements are obtained for each of N_s point source locations given by $\mathbf{r}_{sl}$ ($l = 1, \ldots, N_s$). By measuring at all detector locations $\mathbf{r}_{dk}$ for each source independently, we obtain the data matrix elements Φ_{kl}. We will also let $|f\rangle$ be the column vector of length N_d whose entries are $f_k = u_{\text{scatt}}(\mathbf{r}_{dk})$, and $|q\rangle$ be the column vector of length N_s whose entries are the point sources.

Similarly, we will let $|d\rangle$ and $|e\rangle$ be vectors of length N_v that contain the moments d_n and incident fields e_n respectively. Then, letting V be the $N_v \times N_v$ interaction matrix that contains the polarizabilites α_n on its diagonal and zeros everywhere

else, equation (5.1.21) can be rewritten in matrix form

$$|d\rangle = V\left(|e\rangle + \Gamma|d\rangle\right) \ . \tag{5.1.26}$$

Solving (5.1.26) gives us the desired T-matrix, where

$$|d\rangle = (I - V\Gamma)^{-1}V|e\rangle = T[V]|e\rangle \ . \tag{5.1.27}$$

Now, to finish it off, we define the $N_d \times N_v$ matrix A by $A_{kn} = G_0(\mathbf{r}_{dk}, \mathbf{r}_n)$, and the $N_v \times N_s$ matrix B by $B_{nl} = G_0(\mathbf{r}_n, \mathbf{r}_{sl})$ and $\mathbf{r}_{sl}$. Then by (5.1.24), $|f\rangle = A|d\rangle$, and by (5.1.11), $|e\rangle = B|q\rangle$. Thus, (5.1.27) can be transformed to the familiar equation

$$AT[V]B = \Phi \ . \tag{5.1.28}$$

Exactly as in (3.2.2), the geometry matrices A and B are obtained by direct sampling of the Green's function whose definition (in this case (5.1.10)) is dependent on the physics of the problem of interest. Note that while the off-diagonal terms of Γ are also obtained by direct sampling, this will not work for the diagonal terms due to the singularity of G_0. As seen in (5.1.23), the discretization process defined $\Gamma_{nn} = 0$ when using the renormalized polarizabilities. This local-field correction and renormalization are thus related to the singularity of the Green's function, as it only affects the diagonal entries of Γ. Algebraically, for an invertible matrix P, this is just the transformation $V' = PV$, $\Gamma' = \Gamma - V^{-1}(I - P^{-1})$ so that

$$(I - V\Gamma)^{-1}V = (I - V'\Gamma')^{-1}V', \tag{5.1.29}$$

where we can specifically take $P = (I - VD)^{-1}$, where D is the diagonal matrix containing the diagonal elements of Γ, so that Γ'_{nn}=0.

5.1.2 Iteration Process

Our reconstructions will deal with sparse targets. This is mathematically equivalent to experiments with known background as detailed in Section 2.1. This allows us to take advantage of sparsity to increase the speed of convergence for a most basic form of DCTMC. Then we will show how many of the improvements from 3.5 can significantly quicken convergence. As mentioned previously, our first test is to show proof of concept for DCTMC as a nonlinear solver.

We consider two targets in these sparse simulations, one "large" and one "small". For the small target, we have discretized the sample into 2,304 voxels ($N_v = 2,304$) on a $16 \times 16 \times 9$ grid. With a homogeneous background of zero, there are two rectangular inclusions of contrast. We parameterize these contrast inclusions by varying the susceptibility χ_0. Then, the first rectangular inclusion is of voxel size $6 \times 6 \times 3$ with susceptibility χ_0 while the second inclusion is of voxel size $5 \times 5 \times 2$ with susceptibility $0.857\chi_0$. These two inclusions touch at one corner only. Thus we can model the target by its susceptibility $\chi(\mathbf{r})$ where

$$\chi(\mathbf{r}) = \chi_0 \Theta(\mathbf{r}) \ , \tag{5.1.30}$$

in which $\Theta(\mathbf{r})$ is a shape function bounded between zero and one. For the small target, $\Theta(\mathbf{r}) = 1$ across the first inclusion, $\Theta(\mathbf{r}) = 0.857$ across the second inclusion, and $\Theta(\mathbf{r}) = 0$ everywhere else.

The large target is discretized into 13,500 voxels on a $30 \times 30 \times 15$ grid and also contains two rectangular inclusions. One inclusion is of voxel size $6 \times 6 \times 4$ and

has $\Theta(\mathbf{r}) = 1$ while the other inclusion is of voxel size $10 \times 10 \times 6$ and has reduced contrast of $\Theta(\mathbf{r}) = 0.75$. Again, $\Theta(\mathbf{r}) = 0$ everywhere else. For the large target, the two rectangles come close to one another, but do not touch anywhere. Both of these targets are displayed in Figure 5.1.

These shapes hold true for all small and large targets in this section, with the only varying parameter the largest contrast susceptibility χ_0. As we increase the value of χ_0, it is clear that the overall nonlinearity of the ISP also increases. As we are working in the frequency domain, the wavenumber was fixed such that $kh = 0.2$. This small value is fairly realistic, and validates the discretization scheme used. In general, very strong nonlinearity is present if the phase shift between an incident wave not passing through any inclusions and one propagating through the inhomogeneity is $\pi/2$ or larger. The phase shift can be calculated by

$$\Delta\varphi = (kh)n\left(\sqrt{1 + 4\pi\chi_0\Theta_i} - 1\right) \ , \tag{5.1.31}$$

where n represents the voxel depth of the inclusion. This rough analysis severely underestimates of the amount of nonlinearity present, as it ignores multiple scattering between inclusions. Table 5.1 gives the varying values of susceptibility used, and its equivalence in terms of permittivity as well as the expected phase shift ignoring multiple scattering for the small target, with identical information for the large target in Table 5.2. As one can see, for either size target, a susceptibility of $\chi_0 \geq 0.875$ leads to strong nonlinearity with accordingly sizable phase shifts. As we will later see, the linear reconstructions will break down for smaller values of χ_0

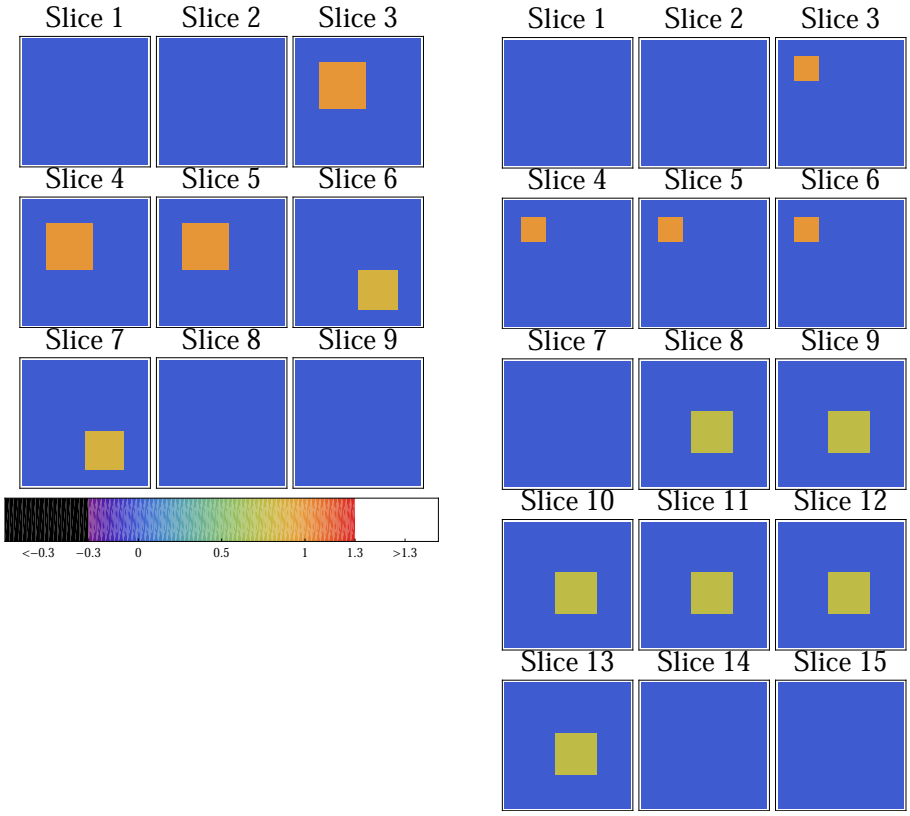


Figure 5.1: The shapes of the small (left) and the large (right) targets. The small target consists of $16 \times 16 \times 9$ voxels and is represented by 9 slices, each slice being of the dimension 16×16. These slices are shown in the figure consecutively. Similarly, the large target consists of 15 slices of the size 30×30 each. The color scale ranges over $[-0.3, 1.3]$ to allow slight deviations of the ratio χ_n/χ_0. There are two cutoffs: values that are smaller than -0.3 are displayed as black and values that are larger than 1.3 are displayed as white.

	Inclusion 1		Inclusion 2	
χ_0	ϵ_n	$\Delta\varphi$	ϵ_n	$\Delta\varphi$
0.00175	1.02	0.004π	1.01	0.002π
0.0175	1.22	0.04π	1.19	0.02π
0.175	3.20	0.30π	2.88	0.13π
0.875	12.00	0.94π	10.42	0.43π
1.75	22.99	1.45π	19.85	0.66π

Table 5.1: Small target susceptibilities and estimated phase shifts

	Inclusion 1		Inclusion 2	
χ_0	ϵ_n	$\Delta\varphi$	ϵ_n	$\Delta\varphi$
0.002	1.03	0.007π	1.02	0.005π
0.02	1.25	0.07π	1.18	0.05π
0.2	3.51	0.52π	2.88	0.36π
1.0	13.57	1.61π	10.42	1.14π
2.0	26.13	2.47π	19.85	1.76π

Table 5.2: Large target susceptibilities and estimated phase shifts

due to the effects of multiple scattering. Thus the order of χ_0 gives us an idea of the amount of nonlinearity present, but is in no way precise. However, to be able to compare reconstructions across varying degrees of χ_0, all reconstructions plot the ratio of χ_n/χ_0. This normalization is done after the fact – no *a priori* information of the value of χ_0 was used during the reconstruction process.

Source and detector grids were placed on either side of the sample as in Figure 2.1. For the small target, we have considered near-field, intermediate-field, and far-field zone cases. For the near-field zone arrangement, two identical 22×22 panels of sources and detectors were used, each with pitch h. The sample is placed

symmetrically between these grids, so that the grids extend beyond the sample by $3h$ in all directions. These panels are placed a distance of $h/2$ away from the sample. With $N_s = N_d = 484$, there are a total of 234,256 data points.

For the intermediate-field zone arrangement, the grids of sources and detectors are moved farther away so that they are $5h$ away from the sample. To account for this distance, extra rows and columns of sources and detectors were added so that these grids wer of size 40×40. The sample is still symmetrically positioned between these panels. For this case, $N_s = N_d = 1,600$ with a total of 2,256,000 data points.

The far-field zone arrangement increases the separation between the sample and source/detector grids to $50h$. The grids of sources and detectors also increase in size to 46×46. For this case, $N_s = N_d = 2,116$ leading to $4,477,456$ data points.

We only consider the case of the near-field zone for the large target. Two panels of sources and detectors of size 38×38 are placed a distance of $h/2$ away on either side. The data size is then 2,085,136 with $N_s = N_d = 1,444$.

The reconstructions in this section follow this streamlined iteration cycle as described in 3.4.3. As previously mentioned, these initial simulations were conducted to test DCTMC's ability as a nonlinear solver, and forgo the computational improvements discussed in Section 3.5. However, there are two specific modifications made to the algorithm for the simulations. One is regularization to account for physical constraints as mentioned in Section **??**, and the other is to take advantage

of the sparse target. Before displaying the reconstructions, we will first provide a bit of detail into these two additions.

Physical Constraint Regularization

We can apply physical constraints if we have some a priori knowledge of the properties of the target. An example of this is if we know that the medium is passive, meaning that the imaginary part of the susceptibility is nonnegative. An even stricter condition would be if we knew that the medium is non-absorbing or transparent, in which case $\mathrm{Im}\chi_n = 0$. As we have chosen the polarizabilities α_n as the fundamental unknowns as opposed to the susceptibilities χ_n, it is imperative to convert either the non-amplifying or non-absorbing conditions in terms of α_n. To do this, we notice that the quantity

$$\mathrm{Im}\left(\frac{h^3}{\alpha_n}\right) = -\left[\frac{\mathrm{Im}\chi_n}{|\chi_n|^2} + (kh)^3\right] \ , \tag{5.1.32}$$

where if $\mathrm{Im}\chi_n \geq 0$, the first term on the right-hand side is strictly positive. Therefore, $\mathrm{Im}(h^2/\alpha_n) \leq -(kh)^3$. For a given intermediate result α_n, this condition can be enforced by the transformation

$$\alpha_n \longrightarrow \frac{1}{\mathrm{Re}(1/\alpha_n) - i\max\left[-\mathrm{Im}(1/\alpha_n), k^3\right]} \ . \tag{5.1.33}$$

Moreover, if we have the stricter non-absorbing condition ($\mathrm{Im}\chi_n = 0$), then we know that $\mathrm{Im}(h^2/\alpha_n) = -(kh)^3$. In this case, the transformation

$$\alpha_n \longrightarrow \frac{1}{\mathrm{Re}(1/\alpha_n) - ik^3} \ . \tag{5.1.34}$$

ensures that all results satisfy this physical condition. To this end, the simulations in this section were assumed a priori to be transparent, and the transformation (5.1.34) was applied each iteration after diagonalizing the interaction matrix. It is worth noting that this regularization slightly improved convergence speed, and was needed in strongly nonlinear cases to prevent convergence towards unwanted results.

Sparsity

We are assuming that our targets are sparse. This means that we have a homogeneous background of zero. This comes from our mathematical discretization of the problem, where we are using the free space Green's function calculated in (5.1.10). Assuming that the background is zero is not an unreasonable assumption, as in most imaging, it is conventional to assume that the value of the homogeneous background is known. In the unlikely event that the background is not free space, computation of a more appropriate Green's function can be analytically performed for a number of standard geometrical setups. In this case, the updated Green's function transforms the unknown χ in the Lippman-Schwinger equation (5.1.8) to be $\chi - \chi_0$ where χ_0 is the background susceptibility. Thus the procedure described in this section still applies even though the background is not free space. We have chosen to use free space background for these targets, so that we are able to use the simple expression (5.1.10) for the Green's function. But in terms of the algorithm,

this is mathematically equivalent to calculating the appropriate Green's function for a known non-free space background.

We account for sparsity using a primitive type of adaptive mesh that roughens the target to force entries close to background values to remain as the background. In our statement of the problem, this refers to identifying susceptibilities χ_n that are negligibly small, and can thus be set to the background value of zero. We call these background voxels "noninteracting", as they do not contribute in any meaningful way to the scattered field.

If we know for certain that a voxel is "noninteracting", then recalculating its value every iteration is a waste of computational effort. To this end, removing this discretized voxel from the computational domain can dramatically increase reconstruction speed. This is because the bottleneck obtaining the matrix $S_k = (I - \Delta_k)^{-1}$ can be reduced from $O(N_v^3)$ to $O((N_v - p)^3)$, where p is the number of found "noninteracting" voxels. An even more dramatic approach that improves convergence speed is to recalculate the experimental T-matrix and its associated matrices under this reduced computational domain. This allows the knowledge of a "noninteracting" voxel to be integrated into the algorithm beyond just quicker matrix inversion. In a sense, since if $(D_k)_{ii} = 0$, then the i-th column and row of T_k are identically equal to zero, the effective size of the T-matrix is of smaller dimension.

We have accounted for sparsity using this more intense method of using "nonin-

teracting voxels" by recalculating geometry and experimental matrices whenever a background voxel has been found. To this end, we use a modified algorithm that has strict conditions for declaring a given voxel to be "noninteracting". The algorithm runs as follows:

1. Run 50 iterations normally.

2. Then every 20 iterations check whether some susceptibilities χ_n satisfy $|\chi_n| < \chi_{\max}/100$, where $\chi_{\max} = \max_n |\chi_n|$. This identifies susceptibilities that could negligibly contribute to the scattered field.

3. If a given voxel satisfies the above condition 3 checks in a row, the corresponding χ_n is set to zero. This ensures that the given voxel is not likely to be interacting.

4. The voxels with zero χ_n (as determined in the previous step) are declared to be non-interacting and are excluded from the computational domain. We then recalculate the initial setup procedure, but for a smaller number of interacting voxels N_v. This results in a smaller computational time per subsequent iteration.

5. The process is repeated with the following modifications. After 200 iterations, checks are made every 10 iterations, and after 400 iterations, the relative threshold for determining a non-interacting voxel is reduced to the factor of 60, and after 600 iterations the relative threshold is reduced to the factor of

40.

The roughening procedure described is fairly ad hoc, and the parameters or conditions used can be easily changed depending on the level of confidence one has with declaring noninteracting voxels. There is certainly a risk of incorrectly assigning noninteracting voxels, which has the potential to severely derail the reconstruction process. However our simulations show that this rarely happened, and incorrect assignments only occurred for severely nonlinear problems that would not have converged regardless. In fact, several results with incorrectly assigned noninteracting voxels were dramatically preferred over reconstructions done without taking sparsity into account. Moreover, in certain scenarios one can detect an incorrect assignment by monitoring the error of the matrix equation (5.1.36). A spike in this error after declaring a voxel to be negligible is likely a result of incorrect assignment.

It is also worth pointing out that while we are taking advantage of the sparse nature of these targets, we do not actually have to assume any a priori knowledge of sparsity (as opposed to the physical constraint regularization in the previous section). This procedure merely reduces very small values to zero. If there are no such voxels, this procedure does not force any roughening to occur.

Measure of Convergence

We use three errors to analyze the convergence of iterations. The first indicative error is normalized root mean square error of the solution η_χ. This error is defined

as

$$\eta_\chi^2 = \frac{1}{N_v \chi_0^2} \sum_{n=1}^{N_v} \left[\chi_n^{(\text{Reconstructed})} - \chi_n^{(\text{True})}\right]^2 \ . \tag{5.1.35}$$

Note that this error can only be computed as we are simulating the reconstructions and thus know the actual susceptibility values of the target. In practice, this error can not be computed. No matter which version of the DCTMC iteration cycle is being used, this error is calculated after obtaining the diagonal interaction matrix D_k (after Step 1 in Option 1).

The error that can be used to monitor convergence without any a priori knowledge of the desired result is the error of the equation η_Φ. This is defined as

$$\eta_\Phi^2 = \frac{1}{N_d N_s \chi_0^2} \sum_{i=1}^{N_d} \sum_{j=1}^{N_s} \left[\Phi_{ij} - (ATB)_{ij}\right]^2 \ . \tag{5.1.36}$$

This error is intended to be computed after obtaining the T-matrix from a diagonal interaction matrix (after Step 4). Of course this error should not be computed after overwriting the T-matrix, as then the matrix is fully data compatible and $\eta_\Phi \approx 0$ up to numerical precision. If the roughening procedure is used to take advantage of sparsity, this is the error that should be monitored for any unwanted spikes.

The last error deals with the conceptual description of DCTMC, where the goal is for the interaction matrix to become more diagonally dominated as the iterations proceed. To that end, we also calculate the ratio of the off diagonal to diagonal norms η_V of the interaction matrix V before any diagonalization takes place. That is,

$$\eta_V = \frac{\sum_{i,j}(1-\delta_{ij})|V_{ij}|^2}{\sum_i |V_{ii}|^2} \ . \tag{5.1.37}$$

This error can also be calculated without any a priori knowledge of the target. It is worth noting tht the main shortcut options for the iteration cycle directly compute a diagonal interaction matrix from a data compatible T-matrix–a non-diagonal V is never actually computed. Therefore, if one is interested in calculating this error, one must add a step where you compute a not necessarily diagonal V directly by $V_k = (I + T_k\Gamma)^{-1}T_k$. This matrix will then be discarded in terms of continuing the iterations, it is only used for calculating this error. Note that while computing the error of the solution η_χ is fast, the matrix multiplication required for computing η_Φ and the extra step required to calculate η_V does add a computational burden. Therefore, to avoid unnecessary slowdown, these errors were calculated every ten iterations.

5.1.3 Small Target Reconstructions

Using Option 1 of the streamlined iteration cycle and the add-on modifications described in this chapter, reconstructions for the small target in the near field zone are shown below in Figure 5.2. The top two rows show the linearized solutions from the first Born and first Rytov approximations. As mentioned previously, the mean field approximation was also tested, which is not shown. However, it was clear that there were no advantages to using either approximation over the first Born, so first Born was the linearizing approximation used henceforth. We display the first Rytov approximated reconstruction for this case to provide evidence that it is not superior.

It is clear that the linear solver is near perfect for $\chi_0 = 0.00175$, and still quite good for the case $\chi_0 = 0.0175$. However it begins to break down as $\chi_0 = 0.175$, and is near worthless for $\chi_0 \geq 0.875$. Thus we would be content if DCTMC performed better for $\chi_0 = 0.175$.

The DCTMC reconstructions are shown in the third row of Figure 5.2 after performing 900 iterations. For $\chi_0 \leq 0.875$, the DCTMC reconstructions are nearly perfect, and vastly outperform the linear counterparts. Even the case with strongest nonlinearity $\chi_0 = 1.75$, we were able to obtain a very reasonable result with this nonlinear algorithm. Looking closely at this particular reconstruction, it is clear that the roughening procedure incorrectly assigned several voxels as noninteracting. It is quite interesting that the reconstruction still provided some value despite being handicapped with erroneous information. Thus, tweaking the parameters and conditions for assigning noninteracting voxels can ameliorate this issue. This can include both setting the initial relative threshold to a value smaller than 1/100, or increasing the number of consecutive identifications beyond three for declaring a voxel noninteracting forever.

Figures 5.3 and 5.4 contain the reconstruction results for the intermediate field zone and the far field zone respectively. Clearly as the source and detector panels are moved farther away from the sample, the quality of the reconstruction begins to deteriorate as can be seen by looking at the linearized reconstructions. While this is less true for the DCTMC reconstructions in the intermediate field zone up

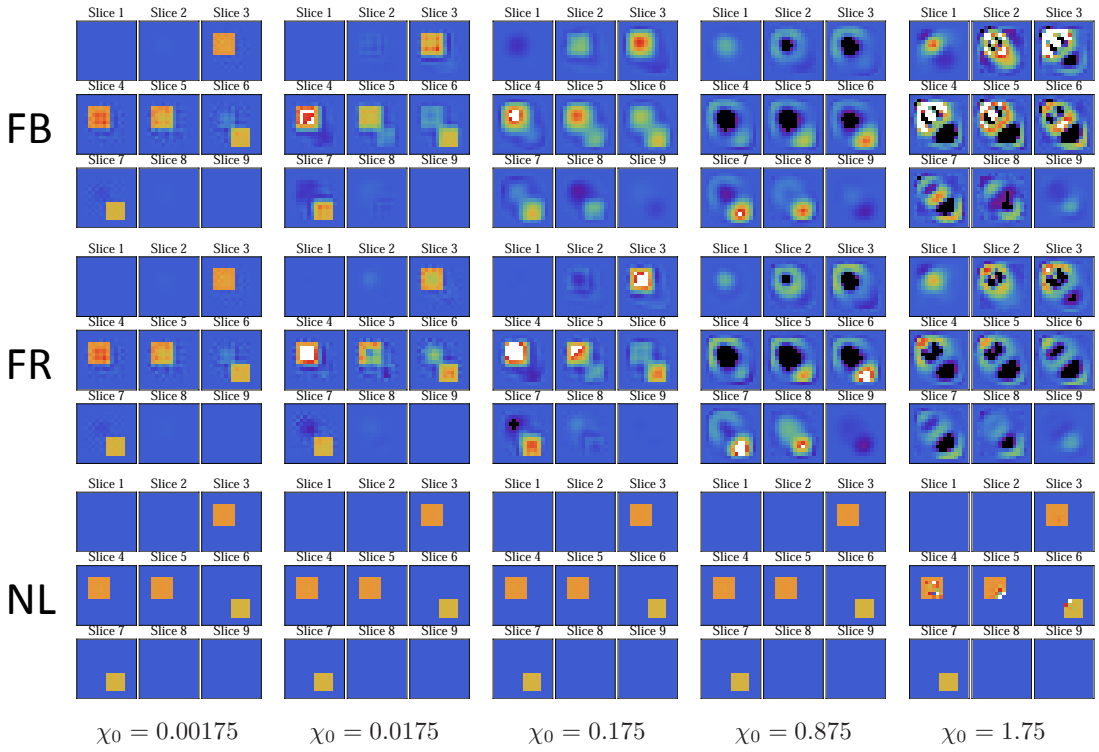


Figure 5.2: Linear (top and middle rows, marked FB for first Born and FR for first Rytov approximation) and nonlinear (bottom row, marked NL) reconstructions of the small target for varying levels of contrast χ_0. The source/detector planes are in the near-field zone of the sample. The quantity shown by the color scale in each plot is χ_n/χ_0, where χ_n is the reconstructed susceptibility of the n-th voxel and χ_0 is the amplitude of the shape function. Voxels reconstructed above a cutoff value are shown in white while those reconstructed below a lower cutoff are shown as black.

to $\chi_0 = 0.875$, where the reconstructions are still nearly perfect, it is clear that there is a complete breakdown for the strongest case. This result is definitely made to look worse by the complete assignment of incorrect noninteracting voxels. Thus this reconstruction can be made to look better by either adjusting the sparsity conditions or forgoing the roughening procedure altogether, but we have displayed all reconstructions using the same algorithm for the sake of consistency.

As we move to the far field zone,it is clear that the problem is too ill-posed to arrive at such neat images as before. That is, we have lost a significant amount of important information that is contained in the evanescent waves. It is still noteworthy that DCTMC outperforms the linear reconstructions in the less nonlinear cases ($\chi_0 \leq 0.175$), but both reconstructions fail at stronger nonlinearities. Overall, the conclusion from these reconstruction plots is that DCTMC is a viable nonlinear solver that outperforms a linear reconstruction in many important scenarios. It is reliable in the near field zone for even severely nonlinear problems, but is less effective at strong nonlinearities of sources and detectors are placed farther away. Still in these cases, a linear solver is never preferred over a DCTMC reconstruction.

We now turn to analyzing the error plots. These are all contained in Figure 5.5. The top row displays the error against the target η_χ for each iteration for all three source/detector arrangements. Let us first take a look at the case in the near field zone. Besides $\chi_0 = 1.75$, these normalized errors are nearly identical and independent of the value of χ_0. This is evidence that DCTMC is consistent in terms

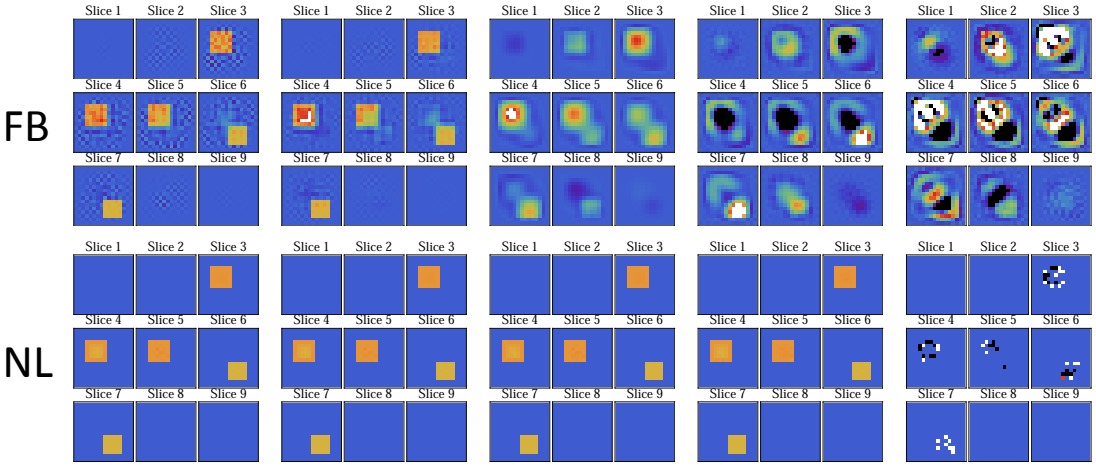


Figure 5.3: Same as in Fig. 5.2 but the source/detector planes are located in the intermediate-field zone of the sample.

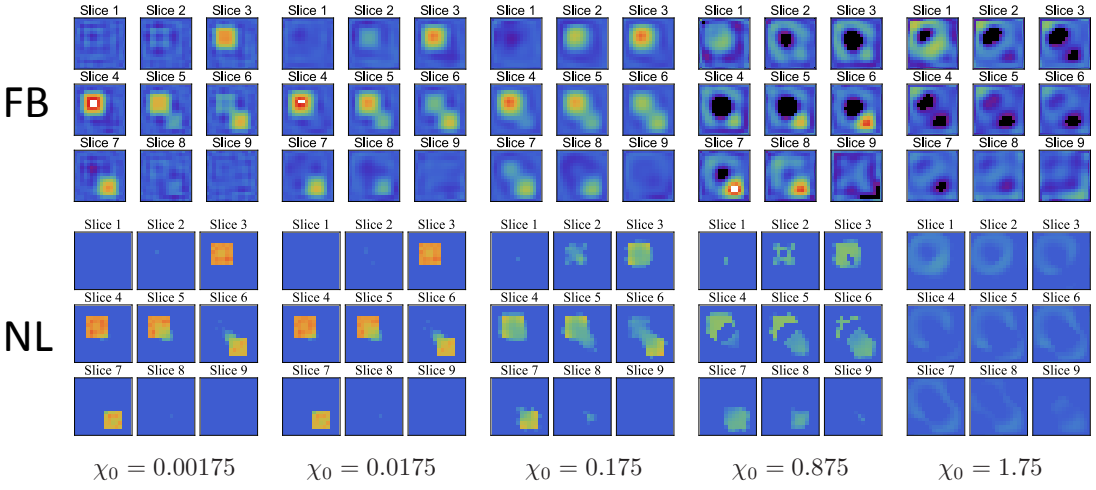


Figure 5.4: Same as in Fig. 5.2 but the source/detector planes are located in the far-field zone of the sample.

of solving nonlinear problems up until severely nonlinear situations (when we would expect many solvers to fail). For these cases, there are three convergence patterns. Up until about iteration number 200, convergence is slow. Here, the intermediate results of reconstructed values α_n are all very small, in which case the relationship between T and V is quite linear. Therefore, these iterations are proceeding similar to DCTMC in the linear regime, where it was shown that the iterations are slow as in first-order Richardson. The convergence of this region can be easily sped up by adding Tikhonov regularization, or starting from the linear solution that can be obtained quickly by conjugate gradient descent. As we will see in Section 3.5, our initial guess obtained directly from the experimental T-matrix is very close to a zero interaction matrix, and starting from the linear solution greatly reduces this region of slow convergence.

At about iteration number 200, the slope of the error curve decreases as the magnitude of the reconstructed polarizabilities has increased into a true nonlinear regime, and the algorithm has correctly assigned many noninteracting voxels. Zooming in on the errors between iteration number 200 and 400, one sees that the slope is not smooth, and there are several sharp descents when the algorithm correctly assigns noninteracting voxels.

The last region occurs around iteration 400, when the majority of noninteracting voxels have been identified and algorithm is left to run in this reduced computational domain. The error $\eta_\chi(i)$ decreases at an exponential rate as the amplitudes of the

interacting voxels are in general increased to become closer to the actual value. It is worth noting that the plot suggests that further iterations will reduce this error and improve the accuracy of the reconstruction.

However the case $\chi_0 = 1.75$ in the near field zone has a different behavior. There is still the slow convergence region, but then there is a brief spike when several voxels are incorrectly assigned as noninteracting. However, thereafter is the interesting turn where the reconstruction actually improves in a region of fast convergence. However, this is short-lived compared to the less nonlinear cases as the reconstruction hits a floor imposed by these incorrectly assigned voxels around iteration number 250.

The behavior of $\eta_\chi(i)$ in the intermediate field zone is similar, where the cases $\chi_0 < 1.75$ are all nearly identical, albeit with a slower region of fast convergence. Moreover, the case $\chi_0 = 1.75$ shows that too many voxels were incorrectly identified as noninteracting as there is a spike, but no eventual decrease as in the near field zone case. All five curves in the far field zone are fairly similar, were all iterations remain in the slow convergence zone, due to the lack of information contained in the data.

Turning to the error of the equation η_Φ, we see roughly similar behavior in all three source/detector arrangements. However, these curves are not quite as independent of the value of χ_0, as there is some separation present. The separation that is present, say in the near field zone, is mainly in the third region of exponential

convergence. This is solely a result of the amount nonlinearity present in the inverse problem. Moreover, the fact that there is some separation in the slow convergence area does not disprove the above rationale that these iterations behave like linear first-order Richardson, as the definition of this error η_{Φ} is not proportional to χ_0, as it contains the data matrix Φ. Besides the separation, the cases $\chi_0 \leq 0.875$ all exhibit similar convergence properties to the error η_{χ}. That is, in the near field zone there are three distinct regions of convergence: slow linear convergence, fast convergence as noninteracting voxels are identified, and final exponential convergence where the error continues to decrease. The slow convergence region is longer in the intermediate field zone with a fitting reduction in faster convergence time, while again the far field zone struggles to escape slow convergence.

For $\chi_0 = 1.75$, there is interesting behavior in the near field zone. Slow convergence is ended by a slight increase due to unwanted assignment of noninteracting voxels followed by fast convergence as with η_{χ}. However, instead of immediately reaching a floor, there is a region of exponential convergence until around iteration 600. We have exponential growth after this fact, which shows we would have been better off stopping the iterations at 600 as opposed to 900. Recall, η_{Φ} can be calculated without any a priori knowledge of the target, so it is viable to monitor this error and stop whenever there is no longer any decreasing behavior. We will show an example of this for the large target. It is also clear that there is a small spike when a voxel is declared noninteracting incorrectly. It is certainly possible to

detect this bump, and reset the algorithm back before this assignment and forgo any sparsity checks until a later time.

The intermediate field zone error of the equation for $\chi_0 = 1.75$ shows chaotic behavior consistent with its undesirable reconstruction. The far field zone remains in slow convergence at the strong nonlinearity as well.

Lastly, we analyze the rate of the interaction matrix becoming diagonal, quantified by η_V. The behavior across different levels of nonlinearity is analogous to η_Φ in the near field zone, albeit with a lack of exponential decay for $\chi_0 = 0.175$ and 0.875. It is worth noting that even in the intermediate field zone and far field zone, the overall behavior is similar, but in all cases there is fast convergence at the beginning. This shows that the concept of DCTMC iteratively searching for an interaction matrix that is increasingly diagonally dominated actually plays out in the reconstruction process. Thus it is interesting that the error η_V suffers mainly from slow convergence after the initial 200 iterations. If not enough noninteracting voxels are identified after that point (as in the intermediate and far field zones), this error more or less levels off.

5.1.4 Large Target Reconstructions

The large target has significantly larger dimensions than the previous reconstructions for the small target. This adds both a greater degree of ill-posedness as well as stronger nonlinearity at lower contrast levels. A contributing factor to this ill-

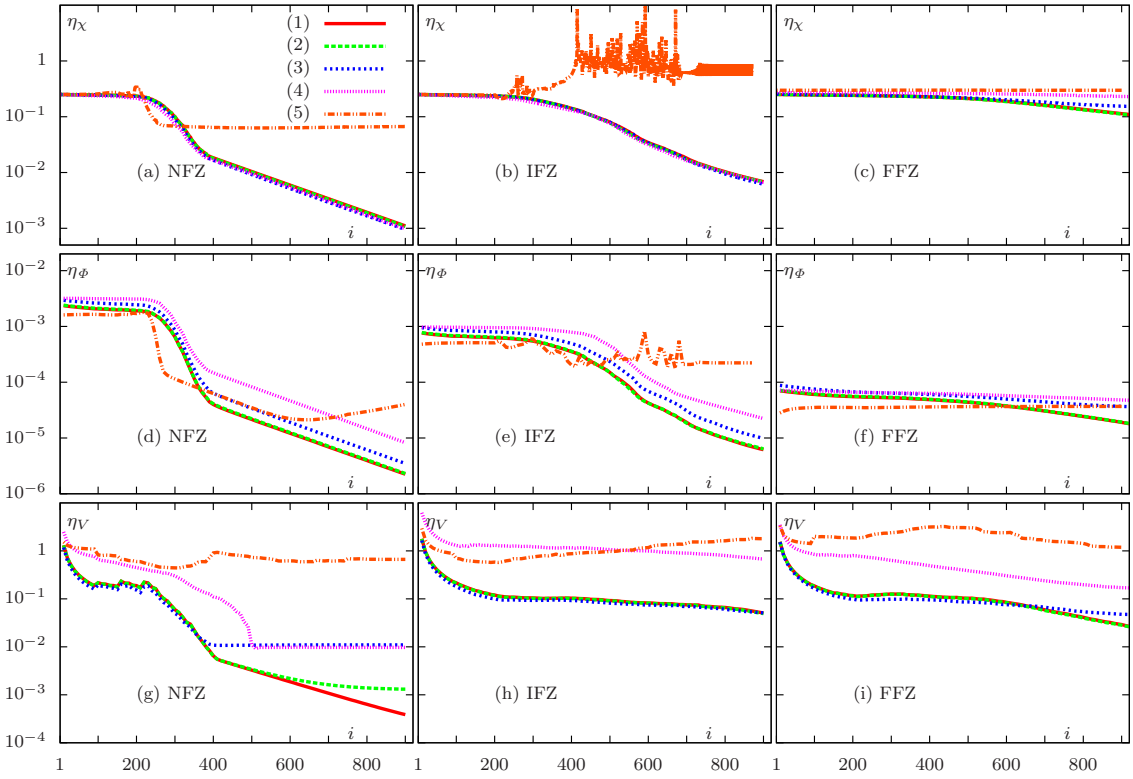


Figure 5.5: Convergence data for the small target. Errors η_χ (a,b,c), η_Φ (d,e,f) and η_D (g,h,i) are plotted vs the iteration number i for the near-field zone (NFZ: a,d,g), intermediate-field zone (IFZ: b,e,h) and far-field zone (FFZ: c,f,i) source-detector arrangements. The different curves correspond to different contrast χ_0 as follows: $\chi_0 = 0.00175$ (1), $\chi_0 = 0.0175$ (2), $\chi_0 = 0.175$ (3), $\chi_0 = 0.875$ (4), and $\chi_0 = 1.75$ (5).

posedness is the fact that there are interacting voxels that are far away from both sources and detectors. Thus, the amount of information passed along by a propagating wave is reduced. Therefore, the linearized reconstructions begin to break down a lower contrast levels then for the small target as can be seen in Figure 5.6. Even at the lowest contrast level tested $\chi_0 = 0.002$, there is significant room for improvement in the reconstruction, especially as the seventh slice contains a shadow of both phantoms where none should appear. This is a direct consequence of struggling to determine the depth with the increased distance between source and detectors. As in the small target, we have only displayed the linearized reconstructions based on the first Born approximation as first Rytov and mean field approximations were not noticeably better.

Again, the second row of Figure 5.6 shows the DCTMC reconstructions after 900 iterations. For $\chi_0 \leq 0.2$, the DCTMC reconstructions are consistent and fairly independent of χ_0. It is clear that even the DCTMC reconstruction struggles to accurately reconstruct interactive voxels at great depth. However, it is still evident that these DCTMC reconstructions are greatly preferred to their linearized counterparts. This is especially important for the case $\chi_0 = 0.2$, where the linear solution is of quite limited value. As we increase the nonlinearity to $\chi_0 = 1.0$, DCTMC begins to break down, but still provides some utility especially compared to the broken down linear solver. It is clear that incorrect assignment of noninteracting voxels has taken place precisely in the interior of the larger inclusion where voxels

had previously been underestimated at lower nonlinearity levels. However, the first inclusion is reconstructed quite nicely, and the boundaries of the second inclusion are still sharp. This is as far as DCTMC goes, as at $\chi_0 = 2.0$, the reconstruction is not useful analogous to the strongest nonlinearity for the small target in the near field zone.

We now turn to the convergence data for these reconstructions in Figure 5.7. The error plots are reminiscent of the results in the intermediate field zone for the small target. For η_χ, the three lowest levels of nonlinearity are fairly close with a region of slow convergence, followed by a region of increased convergence speed that seems to continue if we ran more than 900 iterations. The curve for $\chi_0 = 1.0$ begins to decrease until it has incorrectly assigned several voxels at which point the error sharply increases. The case $\chi_0 = 2.0$ once again has unwanted oscillatory behavior. These qualitative descriptions hold true for η_Φ and η_V. As previously mentioned, one can certainly monitor the error η_Φ and stop after there is no more decreasing behavior. This is certainly true for the $\chi_0 = 1.0$ reconstruction, in which the error plot has a distinct minimum at $i = 590$. Figure 5.8 compares the reconstructions after 590 and 900 iterations where there is a clear (but not dramatic) preference for the reconstruction if we had stopped the iterations when η_Φ increased.

Overall, the reconstructions so far provide initial proof that DCTMC is a viable iterative method for solving nonlinear inverse problems. Especially taking into account the results for the large target, it is clear that DCTMC is capable of ef-

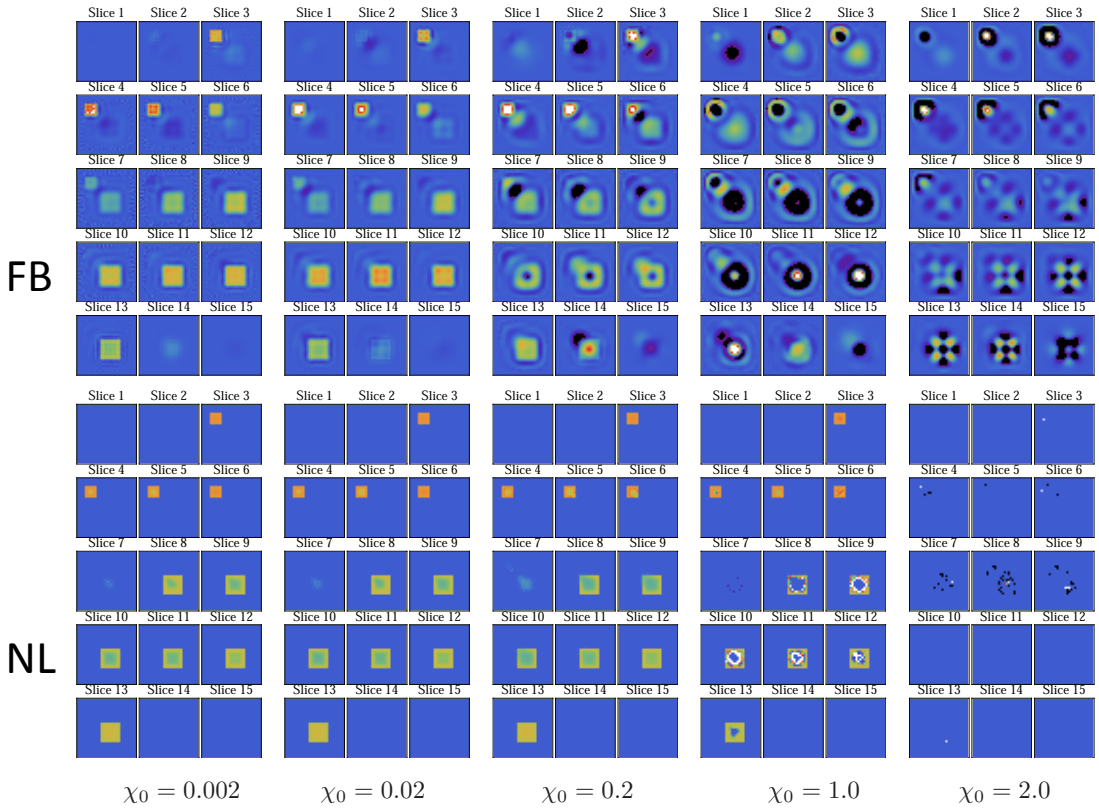


Figure 5.6: Same as in Fig. 5.2, but for the large target, near-field zone source/detector arrangement and a slightly different set of contrasts χ_0. Utilization of first Rytov approximation for linearized reconstruction does not provide any improvements over first Born approximation, and the corresponding results are not shown in this figure.

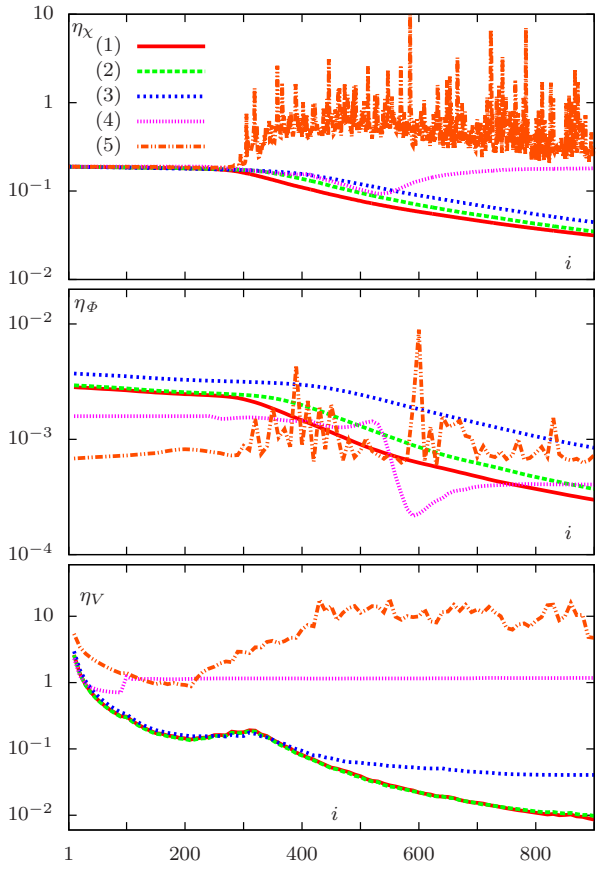


Figure 5.7: Convergence data for the large target. The different curves correspond to different contrast χ_0 as follows: $\chi_0 = 0.002$ (1), $\chi_0 = 0.02$ (2), $\chi_0 = 0.2$ (3), $\chi_0 = 1$ (4) and $\chi_0 = 2$ (5).

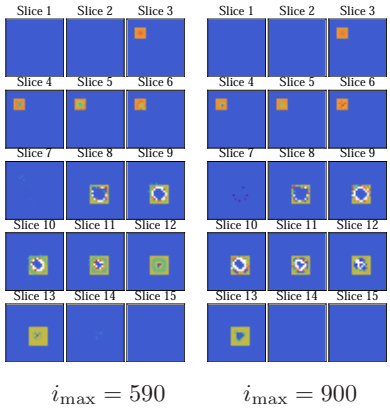


Figure 5.8: Comparison of the reconstructions of the large target with $\chi_0 = 1$ (Case 4 in Fig. 5.7) for different numbers of iterations $i_{\max}$, as labeled. Reconstruction with $i_{\max} = 490$ is marginally better than the reconstruction with $i_{\max} = 900$.

fectively handling large data sets with more than 2×10^6 data points. In all cases, the DCTMC reconstruction performed at least as well as the linear solver, and was indeed greatly preferred the majority of the time. At the strongest levels of nonlinearity, the DCTMC algorithm failed to produce useful results, but this is not too worrisome as early estimates (Table 5.2) show that these problems are severely nonlinear and troublesome for all attempts at solving. While these reconstructions were done methodically in 900 iterations, the next section will take into account many of the discussed improvements to greatly reduce the number of iterations required to fully converge.

5.2 Improved Reconstructions

This section investigates several improvements to the DCTMC algorithm, both motivated by the previous discussion in Section 3.5 and the later analysis of the toy problem in Section 6.1. The overall conclusion is that with a few easy modifications, we can substantially reduce the necessary computation time to obtain accurate results with DCTMC. The results shown are to be compared with the simulations run for the small target in the near-field zone. Similar results were obtained for the intermediate and far field zones, but the work presented will give an idea of the relative merits of these improvements. In the following section we will apply DCTMC with these proven improvements to the more difficult case of diffuse optical tomography.

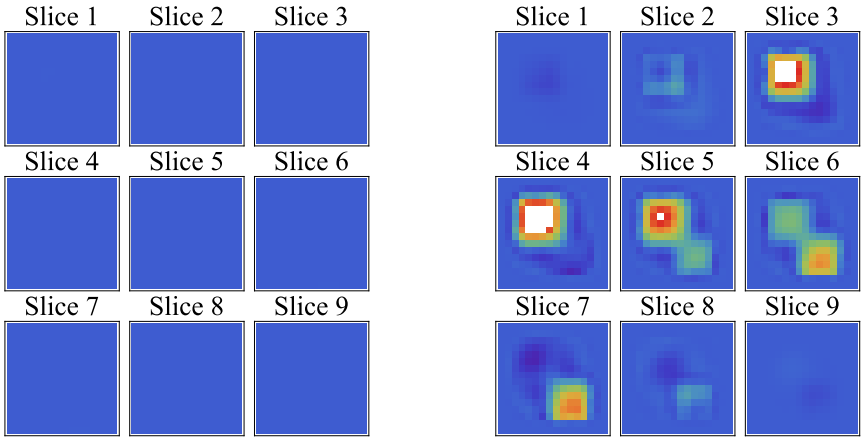


Figure 5.9: Initial starting points for DCTMC in the case $\chi_0 = 0.175$. The left figure is the V-matrix obtained directly from the experimental T-matrix. The right figure is the linear reconstruction obtained quickly.

5.2.1 DCTMC Starting from Linear Reconstruction

As previously mentioned, DCTMC in the linear regime is Richardson first-order iteration, which has slow convergence. Therefore, if our initial guess V_1 is close to 0, then the initial iterations act like the slow Richardson iterations, and take an unnecessary number of iterations to enter the "nonlinear regime". Looking back at the simulations for the small sample done in 5.1.3, we see that our initial guess corresponding to the experimental T-matrix was actually very close to 0, thus making a natural improvement to start from the linear reconstructions that were obtained quickly. For example, as seen in Figure 5.9 the initial guess used in the previous reconstruction for $\chi_0 = 0.175$ is shown in the left panel, compared to the more useful starting point of the linearized reconstruction in the right panel. The number of necessary iterations can be significantly reduced using this modified version of the algorithm:

1. Run the linear reconstruction.

2. Using the result of step 1 as the initial guess, run 5 iterations normally.

3. Then every 5 iterations check whether some susceptibilities χ_n satisfy $|\chi_n| < \chi_{max}/500$, where $\chi_{max} = \max_n |\chi_n|$.

4. If a given voxel satisfies the above condition 3 checks in a row, the corresponding χ_n is set to zero, and the computational domain is reduced.

5. The process is repeated with the following modifications. After 15 iterations, checks are made every 20 iterations and the relative threshold for determining a non-interacting voxel is reduced to the factor of 100. After 200 iterations, checks are made every 10 iterations, and after 400 iterations, the relative threshold is reduced to the factor of 60, and after 600 iterations the relative threshold is further reduced to the factor of 40.

Compared to the original algorithm used previously, the interval between checks is reduced at the beginning to utilize the non-interacting voxels found by the linear reconstruction, while simultaneously raising the relative threshold so as to not trust the linear reconstruction too much. After these 15 initial iterations, these parameters are then set back to match the original method. Error plots comparing these two methods are shown in Figures 5.10 and 5.11. There are a few immediate conclusions to be made from these error plots. First of all, as we are starting from a much more reasonable guess, our starting error is much smaller. More importantly,

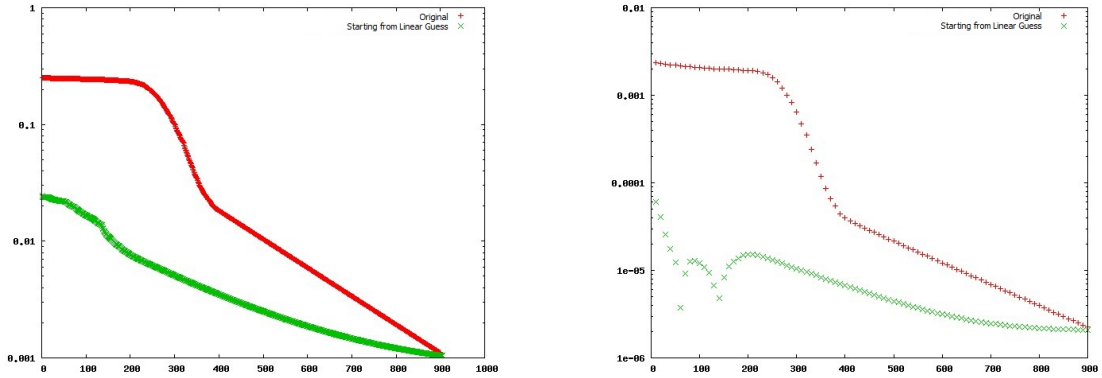


Figure 5.10: Convergence data for the case $\chi_0 = 0.00175$ comparing the original guess and the linear reconstruction guess. The left panel plots the error η_χ while the right panel plots η_Φ.

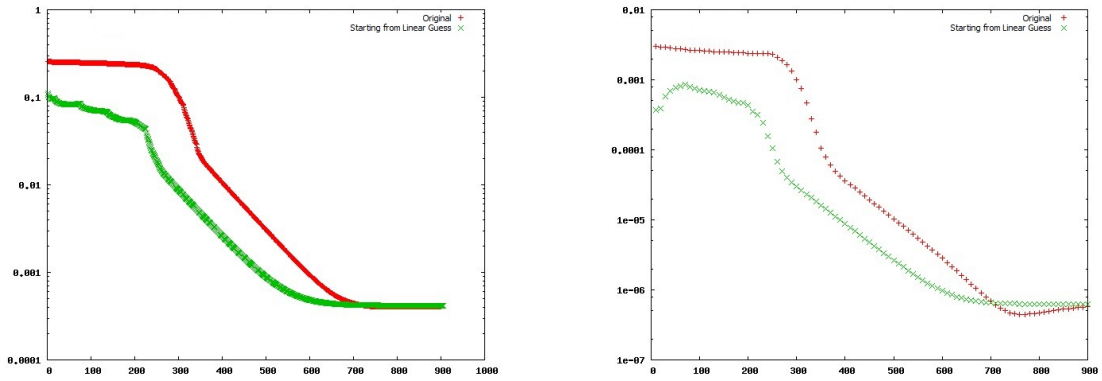


Figure 5.11: Convergence data for the case $\chi_0 = 0.175$ comparing the original guess and the linear reconstruction guess. The left panel plots the error η_χ while the right panel plots η_Φ.

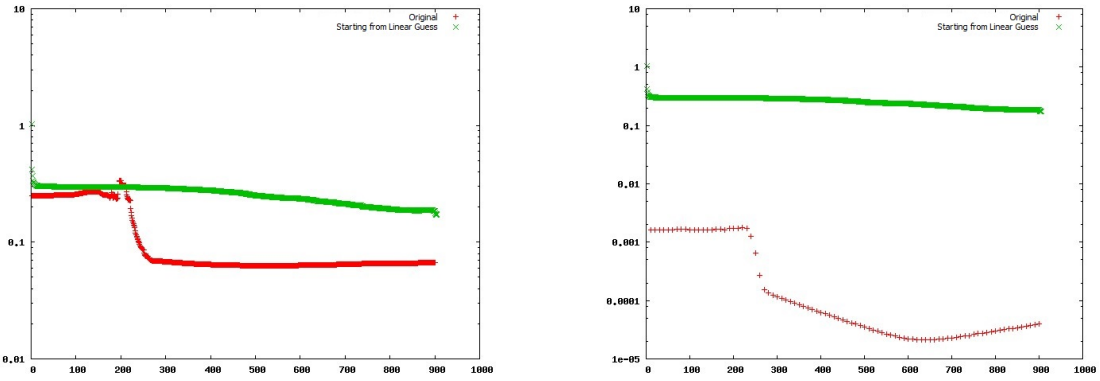


Figure 5.12: Convergence data for the case $\chi_0 = 1.75$ comparing the original guess and the linear reconstruction guess. The left panel plots the error η_χ while the right panel plots η_Φ.

as we can use sparsity conditions very quickly in the iteration process, we have faster convergence from the start, as can be seen from the slope of the error curves starting at iteration 15. For the case $\chi_0 = 0.00175$ (where the linear reconstruction is nearly perfect), and the case $\chi_0 = 0.175$ (where the linear reconstruction is still reasonable), this makes perfect logical sense – a much improved initial guess leads to improved reconstruction speed. The least nonlinear case $\chi_0 = 0.00175$ has a more dramatic improvement (see Figure 5.10), since it starts from a better guess, but even starting from the linearized guess for $\chi_0 = 0.175$ leads to an improvement by about 150 iterations (see Figure 5.11).

The most interesting case is for the strongest nonlinearity $\chi_0 = 1.75$. This case is noteworthy as our initial guess is not close to the actual result. In fact, it is farther away in L^2 error than starting from 0. But as discussed previously, this could be advantageous, namely that starting from 0 leads to slow Richardson iterations whereas an initial guess V_1 far from 0 can take advantage of the nonlinear

aspect of the algorithm. However, in this case we see a very slowly converging (but stable error) that does not outperform the original method (see Figure 5.12). This is not an ideal situation, but we will see in the last section when combining all the improvements that this is not a problem.

5.2.2 Using Reciprocity of Sources and Detectors

We now investigate the improvement of enforcing symmetry of the experimental T-matrix by including the doubled data set obtained by interchanging sources and detectors from Section 3.5.2. As can be seen from the following error plots, using this more complete information leads to slightly improved starting points for the iterations as well as much faster convergence speed. This is a dramatic improvement for a simple change that does not require any additional information or processing power. The case represented in Figure 5.13 is for $\chi_0 = 0.0175$, and all other strengths of nonlinearity exhibit similar behavior. It is noteworthy that with this "more complete" dataset, we achieve full convergence in about 500 iterations, as opposed to obtaining a satisfactory but still improving result after 900 iterations. In terms of the "known" entries in the experimental T-matrix, using reciprocity we know about 17% of the entries, whereas without taking advantage of this we only know about 4% of the entries. That is, by using reciprocity the dimension of the upper submatrix used in overwriting the T-matrix is roughly doubled, which leads to much better results.

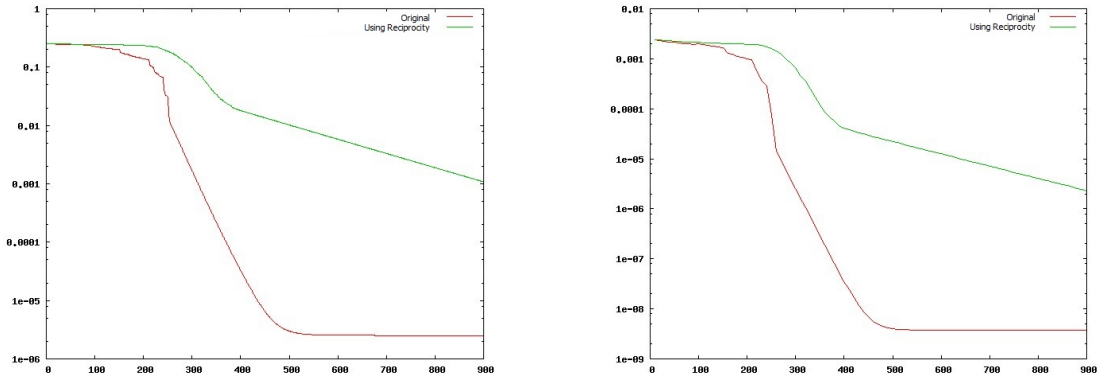


Figure 5.13: Convergence data for the case $\chi_0 = 0.0175$ comparing the original process and using reciprocity of sources and detectors. The left panel plots the error η_χ while the right panel plots η_Φ.

Summing over Rows when Diagonalizing V

As we will see in the toy problem analysis from Section 6.1, much better results were obtained when diagonalizing V by summing over rows versus just setting all off-diagonal terms to zero. The numerical examples show both an increase in convergence speed as well as being more robust in terms of larger convergence areas. This fits in with the conceptual understanding of DCTMC in terms of working in a nonlocal framework. As mentioned in Section 3.5.4, for these intermediate results it is not reasonable to sum over elements that are very far away from each other, and a weight function must be used to suppress these unwanted nonlocal interactions. The results in this section are the most promising results in terms of significantly improving the convergence rate of DCTMC. It is worth emphasizing that the results in this section are without the improvements of the previous two sections (starting from better initial guesses and interchanging source and detector), as well as any

sparsity constraints. The sparsity conditions used previously have been crucial to reasonable convergence rates in simulations thus far, but are in reality an "add-on" to the DCTMC algorithm. Adding the row-summing technique is an inherent change, and these results are a fundamental success wholly attributable to DCTMC.

To fully investigate the merits of this diagonalization scheme, we will forgo using the shortcut from Option 2 with the weighted T-matrix as described in Section 3.5.4, and conduct iterations in the full manner that directly calculate the interaction matrix from the data compatible T-matrix. We can then explicitly sum over the rows of V in the weighted manner we desire. While this increases computation time per iteration by a factor of about 2, we will demonstrate that any loss in calculation time per iteration is more than made up for with faster convergence.

The weight function used for these first examples was a simple characteristic function. That is for each voxel $\mathbf{r}_k$ and a specified radius R, let $I_R = \{j : |\mathbf{r}_j - \mathbf{r}_k| \leq R\}$ be the set of all indices j such that $\mathbf{r}_j$ is in the ball of radius R centered at $\mathbf{r}_k$. Then the act of diagonalizing to $D_k = \mathcal{D}[V_k]$ is given by the entrywise formula:

$$D_{ij} = \begin{cases} \sum_{k \in I_R} V_{ik} & i = j \\ 0 & i \neq j \end{cases} \tag{5.2.1}$$

The first results are for the case $\chi_0 = 0.0175$, where 150 iterations were performed without any adjustments or reductions for sparsity. The final result after these 150 iterations for varying values of R are shown in Figure 5.14. The immediate reaction

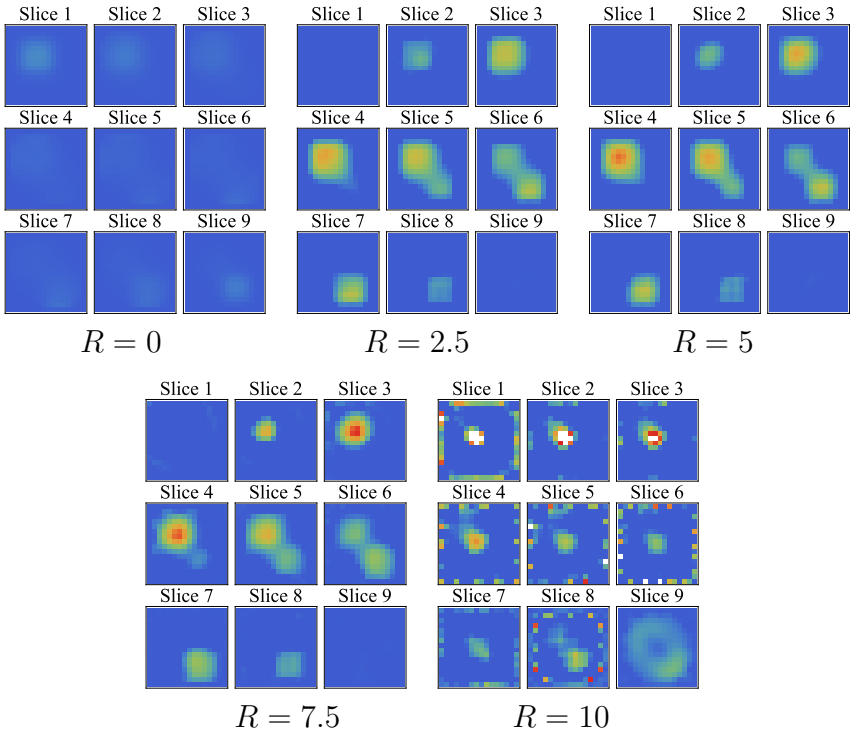


Figure 5.14: Final output of DCTMC algorithm after 150 iterations with varying degrees of R, where the row-summing is done over the radius R.

is that the cases $R = 2.5, 5$,and 7.5 are clearly much better than the original $R = 0$, where little change has happened after 150 iterations (the initial slow convergence). It is also true that summing over a radius of 10 is too large of a radius, as this result is unwanted. Looking at these five images almost looks like a type of regularization, where $R = \infty$ is no regularization, and as we send R to 0, the image "spreads" out a little until it is completely over-regularized at $R = 0$.

Looking at the error plots of η_χ in Figure 5.15, we see how dramatic of an improvement one achieves using row-summing. Using the error against target η_χ, the case $R = 5$ is preferred, followed by $R = 2.5$ and $R = 7.5$. Furthermore, it is interesting that for the case $R = 10$, the first couple of iterations are quite effective,

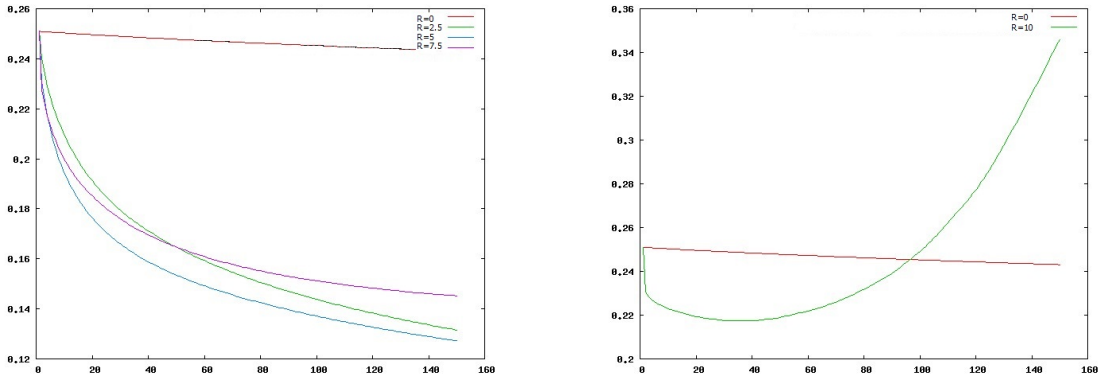


Figure 5.15: Convergence data of η_χ. The left plot compares the cases $R = 0, 2.5, 5,$ and 7.5 while the right plot compares $R = 0$ and 10.

before something goes wrong and the error curve turns sharply in the opposite direction. This is an example of not suppressing a far away nonlocal interaction that causes unwanted behavior. Note that the case $R = 0$ corresponds to no row-summing, and Shortcut 2 was used for this case. But despite the fact that the $R = 0$ case completed in about half the time, looking at the first error plot in Figure 5.15 shows that there is no confusion on the preference between 75 iterations of the positive values of R and 150 iterations of $R = 0$.

We have to be careful as we increase the nonlinearity present in the problem, as the larger magnitude in difference between the background and the inhomogeneities can cause unwanted behavior when summing farther away voxels. For the case $\chi_0 = 0.875$, we see that for $R = 4$ and $R = 6$, the error plots quickly diverge. However, if we add a simple weight function to the sum to penalize entries from farther away, we can dramatically improve the result. An extremely rudimentary

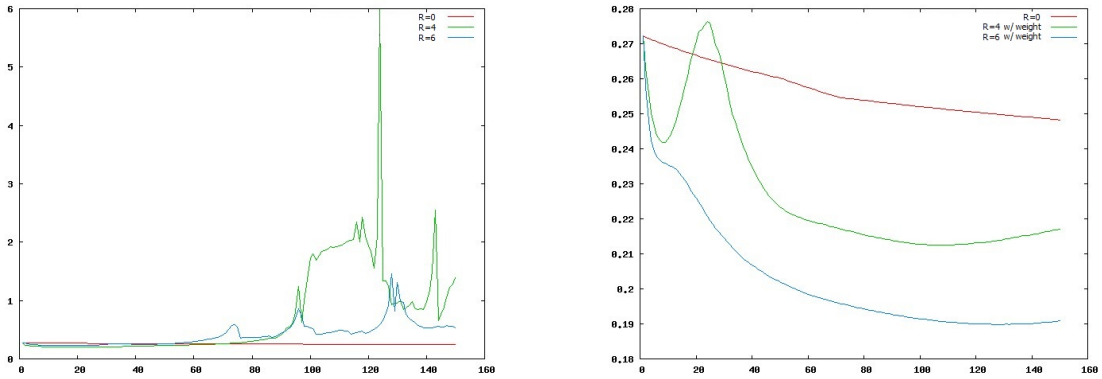


Figure 5.16: Convergence data of η_χ for the case $\chi_0 = 0.875$. The left panel plots the error without using any weight function while the right panel plots the error under the weight function Θ defined in equation (5.2.2).

weight function $w(\mathbf{r}_i, \mathbf{r}_k)$ was used such that diagonalizing is now given by:

$$D_{ij} = \begin{cases} \sum_{k \in I_R} w(\mathbf{r}_i, \mathbf{r}_k) V_{ik} & i = j \\ \\ 0 & i \neq j \ , \end{cases} \tag{5.2.2}$$

where

$$w(\mathbf{r}_i, \mathbf{r}_k) = \begin{cases} 1 & i = j \\ \\ (2|\mathbf{r}_i - \mathbf{r}_k|)^{-1} & i \neq j. \end{cases} \tag{5.2.3}$$

As can be seen in Figure 5.16, adding on this extra weight funtcion was key to improving the algorithm, and suggests that further investigation into the optimal choice of radius R and weight function is warranted. Figure 5.17 contains the final result after 150 iterations for each of these methods. Again it is clear that the reconstructions performed with weighted row-summing are superior to the reconstructions performed without any row-summing, or done with unweighted row-summing.

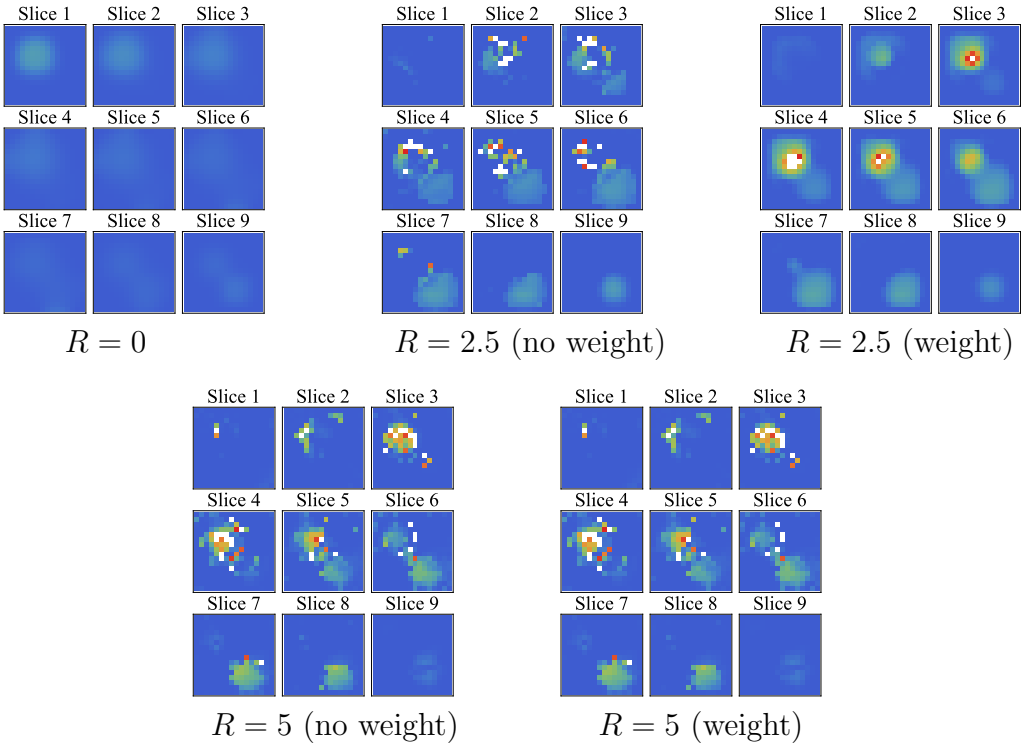


Figure 5.17: Final output of DCTMC algorithm after 150 iterations for $\chi_0 = 0.875$ with varying degrees of R, where the row-summing is done over the radius R with or without a weight function.

5.2.3 Putting it All Together

Combining these improvements, we can see significant performance enhancing of the DCTMC algorithm. Compared to the five cases used in the small sample in the near-field zone, we will demonstrate that one can achieve acceptable results in 75 iterations, compared to the 900 iterations previously performed in Section 5.1.3. For the sake of clarity, the algorithm used in this section proceeds as follows:

1. Run the linear reconstruction.

2. Using the result of step 1 as the initial guess, run 5 iterations normally.

3. Then every 5 iterations check whether some susceptibilities χ_n satisfy $|\chi_n| < \chi_{max}/500$, where $\chi_{max} = \max_n |\chi_n|$.

4. If a given voxel satisfies the above condition 3 checks in a row, the corresponding χ_n is set to zero, and the computational domain is reduced.

5. The process is repeated with the following modifications. After 20 iterations, the relative threshold for determining a non-interacting voxel is reduced to the factor of 100. After 40 iterations, the relative threshold is further reduced to the factor of 60.

Throughout the algorithm, reciprocity of sources and detectors was used, as well as the row summing described in equation (5.2.2). The final result for all levels of nonlinearity are shown in Figure 5.18. The main takeaway is that all of these

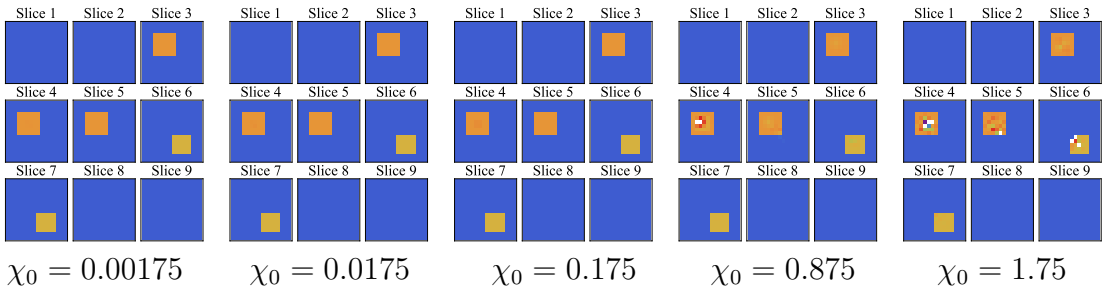


Figure 5.18: Final output of improved DCTMC algorithm after 75 for all five levels of nonlinearity.

results are perfect or at least very acceptable, and were produced in a small fraction of the time originally used to obtain the images in Figure 5.2. The most dramatic improvement are for the three lowest levels of nonlinearity, as starting from these linear reconstructions is of great assistance. For these three, we obtain more accurate results in 75 iterations, compared to the 900 used previously. For the case $\chi_0 = 0.875$, the end result is only marginally worse than the original method, but this trade-off is worth it. Moreover, we could run another 50 iterations quickly and surpass the results obtained previously. These results are contained in the error plots in Figures 5.19 and 5.20.

Again, the interesting case is for the strongest nonlinearity $\chi_0 = 1.75$, which has repeatedly been shown to be the limiting case for DCTMC and has a strong probability to break down. However, we can again obtain a stable result that is not much worse than the "safer" algorithm over 900 iterations. The error of the target fluctuates wildly at the beginning, but then starts to settle down as seen in Figure 5.21. We believe this behavior is due to the row-summing at the beginning handling a large variance across the sample. Even in this case, we can

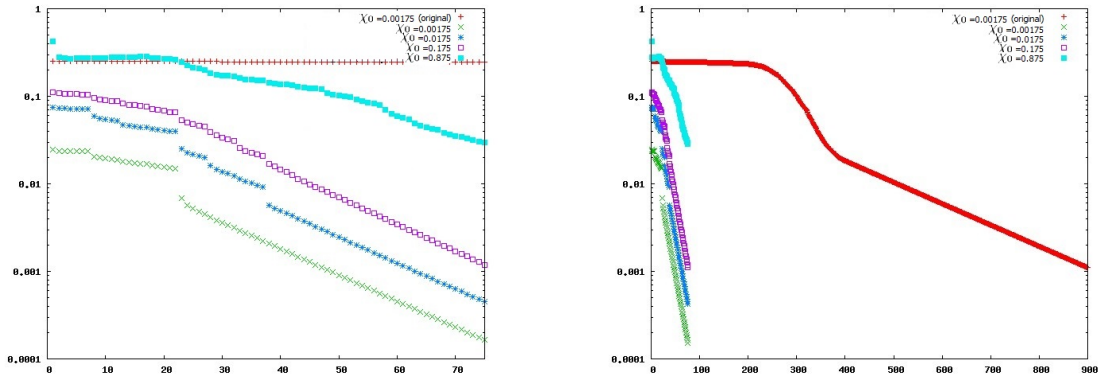


Figure 5.19: Convergence data of η_χ for the cases $\chi_0 = 0.00175, 0.0175, 0.175$, and 0.875 compared against the original results for $\chi_0 = 0.00175$ (The original plots were identical so only one shown for clarity). The left panel plots the error for the 75 iterations only, while the right panel is zoomed out to be able to compare the error of the final result.

strongly conclude that the modifications detailed in this paper result in a significant improvement in convergence speed. We were able to obtain accurate (and in some cases much improved) results in many fewer iterations, which highly increases the practical usefulness of the DCTMC algorithm.

5.3 Three-Dimensional Diffuse Optical Tomography

We now turn to reconstructions of simulated data for diffuse optical tomography. The inverse problem for diffuse optical tomography has greater ill-posedness then scalar wave diffraction due to the exponential decay of the waves. Moreover, we have added noise to the data sets in order to more closely replicate experimental data. Thus, the task of reconstructing the DOT data in the section is much more

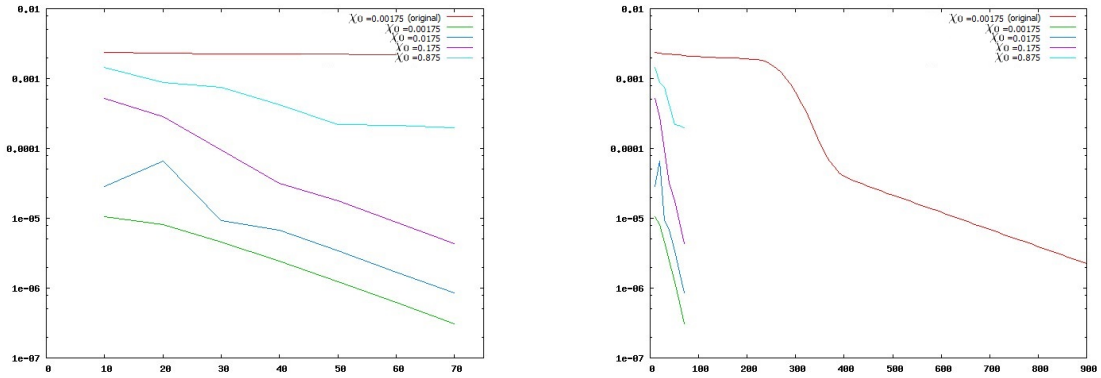


Figure 5.20: Convergence data of η_Φ for the cases $\chi_0 = 0.00175, 0.0175, 0.175$, and 0.875 compared against the original results for $\chi_0 = 0.00175$ (The original plots were identical so only one shown for clarity). The left panel plots the error for the 75 iterations only, while the right panel is zoomed out to be able to compare the error of the final result.

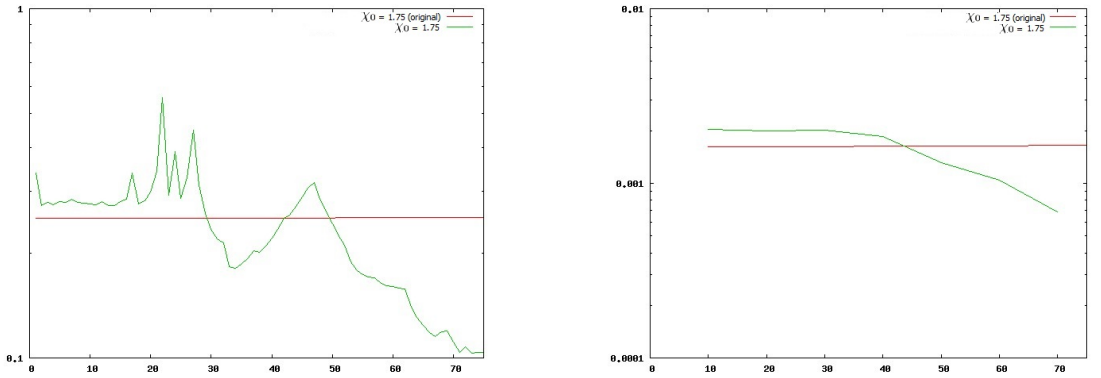


Figure 5.21: Convergence data of η_χ (left plot) for the case $\chi_0 = 1.75$ compared with original result as well as the convergence data of η_Φ (right plot) for the same case.

difficult than in the previous reconstructions. We will again compare our DCTMC obtained reconstructions against linearized reconstructions. For this case, the first Rytov approximation was the superior tested linear approximation and is shown in all plots.

The general discretization process is very similar to the detailed discretization process outlined in Section 5.1.1 and is modeled in [35]. However, we point out a few key differences. First, the free space Green's function G_0 can be calculated as

$$G_0(\mathbf{r}, \mathbf{r}') = -k^2 \frac{\exp\left(-k|\mathbf{r} - \mathbf{r}'|\right)}{|\mathbf{r} - \mathbf{r}'|} \ .$$

Our quantity of interest is now the absorption coefficient at each voxel and will be represented by α_n. We will still consider our targets to be sparse, which is mathematically equivalent to reconstructions with known backgrounds.

Two different targets were chosen for these reconstructions, both the same size. The choice of target was inspired by the work in [6], where experimentally obtained data sets modeling DOT near the chest wall were reconstructed using linear algebraic methods. This models clinical breast cancer imaging, which is a promising application for DOT due to the fact that optical methods are sensitive to several biomarkers for cancer. However in practice, the data can be inefficient due to the strong scattering by the nearby chest wall, which can distort the region of interest. Thus, our targets have placed a larger stronger absorbing material with a nearby smaller phantom of interest with a smaller absorption coefficient. Note that there is no scattering contrast between these two inclusions.

Precisely, we again discretize our sample into 2,304 voxels on a $16 \times 16 \times 9$ grid. The absorption coefficient models are target by

$$\alpha(\mathbf{r}) = \alpha_0 \Theta(\mathbf{r}) \ , \tag{5.3.1}$$

where $0 \leq \Theta(\mathbf{r}) \leq 1$ is the shape function. The "chest wall" phantom is of voxel size $12 \times 3 \times 7$, while the phantom of interest is a cross target made up of two intersecting $3 \times 7 \times 3$ rectangular inclusions. The chest wall phantom had absorption coefficient of α_0, while the cross target's absorption coefficient was halved at $0.5\alpha_0$. The difference between the two targets we consider is the lateral distance between the two phantoms. For the "far" model, this distance is $d = 5$ voxels. The "near" model reduces this distance to $d = 2$ voxels. These models are shown below in Figure 5.22. We will again only vary the value of α_0 to control the amount of nonlinearity present in the problem, and plot all reconstructions on the same color scale by plotting the ratio α_n/α_0.

Consistent with the experimental methodology for this DOT imaging [19], we only consider the near field case, where the grids of sources and detectors are placed $h/2$ away from the sample. The slab geometry consisted of two panels of 26×26 sources and detectors with the sample centered between them in all directions.

5.3.1 Regularization and Noise Suppression

In this section, we are not only testing DCTMC in a more difficult physical model, but we are also adding noise to the data. Gaussian noise at a level of 2% was added

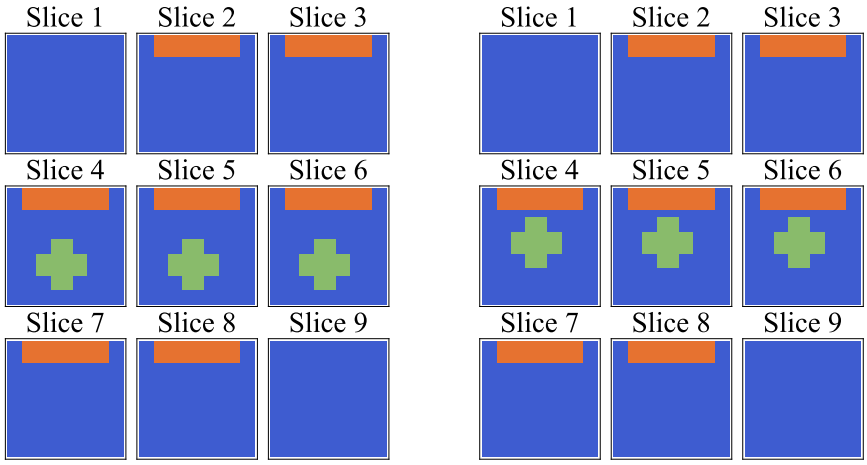


Figure 5.22: **Left**: The "far" target where the cross shaped phantom is a distance of $5h$ from the stronger absorbing rectangular inclusion.**Right:** The "near" target where the cross shaped phantom is a distance of $2h$ from the stronger inclusion.

to the data measurements to this end. There were a few parameter modifications to the algorithm to help suppress this noise.

First of all, linearized DCTMC inspired regularization from Section 4.1 was used with a Tikhonov value of $\lambda^2 = 1.0 \times 10^{-7}$. Significant testing showed that this was a reasonable choice of λ, and more importantly, reconstructions were noticeably superior with the choice of $\lambda \neq 0$.

Additionally, as the choice of when to stop the iterations can also be seen as a regularization process, the error η_Φ was monitored and iterations were stopped after several consecutive increases in this error.

Lastly, and perhaps most importantly, the value of ϵ from equation (3.2.10) had to be significantly reduced compared to the noiseless reconstructions in Section 5.1. While values of $\epsilon^2 = 1.0 \times 10^{-12}$ were frequently used for the scalar wave diffraction reconstructions (which in turn trusted almost all "known" elements of

the experimental T-matrix), such small values of ϵ proved to produce unreasonable results. The optimal value of ϵ found was $\epsilon = 1.0 \times 10^{-4}$. This can be thought of as a type of data restriction, where we treat computed "known" elements of the experimental T-matrix as unknown due to the noisy data measurements that were used in the calculation.

5.3.2 Iteration Process

To have any hope of reconstructing these more difficult simulated data, we employed many of the improvements discussed in Section 3.5 and tested in Section 5.2. We will now detail the improvements used so one can contrast with the iteration cycle for the scalar wave diffraction simulations in Section 5.1.

Symmetry of the T-matrix was enforced using the data points obtained by interchanging sources and detectors. Most importantly, we used a weighted sum over rows to the diagonal to calculate the diagonal approximation D_k to the nonlocal interaction matrix V_k'. This weight was previously tested and defined in (5.2.3). For clarity, the diagonalization operator is defined entry wise as

$$D_{ij} = \begin{cases} \sum_k w(\mathbf{r}_i, \mathbf{r}_k) V_{ik} & i = j \\ 0 & i \neq j \ , \end{cases} \tag{5.3.2}$$

where

$$w(\mathbf{r}_i, \mathbf{r}_k) = \begin{cases} 1 & i = j \\ (2|\mathbf{r}_i - \mathbf{r}_k|)^{-1} & 0 < |\mathbf{r}_i - \mathbf{r}_k| \leq 6 \\ 0 & |\mathbf{r}_i - \mathbf{r}_k| > 6 \ . \end{cases} \tag{5.3.3}$$

We will still use sparsity to our advantage, and can check for noninteracting voxels both sooner in the iteration process and more frequently due to the increased convergence speed provided by the improvements used. The iteration cycles proceed as follows:

1. Run 10 iterations normally.

2. Then this iteration and every 10 iterations afterwards, check whether some values α_n satisfy $|\alpha_n| < \alpha_{\max}/200$, where $\alpha_{\max} = \max_n |\alpha|$.

3. If a given voxel satisfies the above condition 3 checks in a row, the corresponding α_n is set to zero. This ensures that the given voxel is not likely to be interacting.

4. The voxels with zero χ_n (as determined in the previous step) are declared to be non-interacting and are excluded from the computational domain. We then recalculate the initial setup procedure, but for a smaller number of interacting voxels N_v. This results in a smaller computational time per subsequent iteration.

5. The process is repeated with the following modifications. After 50 iterations,

the relative threshold for determining a non-interacting voxel is reduced to the factor of 100, and after 200 iterations the relative threshold is reduced to the factor of 60.

5.3.3 Reconstructions

We again test each target for five levels of nonlinearity: $\alpha_0 = 0.001, 0.01, 0.1, 1.0,$ and 2.0. We first look at the reconstructions of the "far" target. The top row of Figure 5.23 displays the first Rytov approximated linear reconstructions. First Born and mean field approximations were also tested, but not shown as they do not provide any more useful information. Looking at these linear reconstructions, the cases $\alpha_0 \leq 0.1$ all look similar – they do a decent job of reconstructing the chest wall phantom, and the area of the cross target is identified, but its shape is far from exact and the absorption coefficient is substantially under represented. For the case $\alpha = 1.0$, there is still some indicator of the cross target, but the chest wall phantom is extremely noisy. At the largest level of nonlinearity, $\alpha = 2.0$, the amount of regularization needed to prevent an uncomfortably noisy image suppresses any presence of the cross target. Moreover, the chest wall phantom is not reconstructed either.

Now looking at the second row of the DCTMC nonlinear reconstructions, we see a much nicer picture. The immediate reaction is that all five levels of nonlinearity look more or less identical. This is strong evidence that DCTMC is a

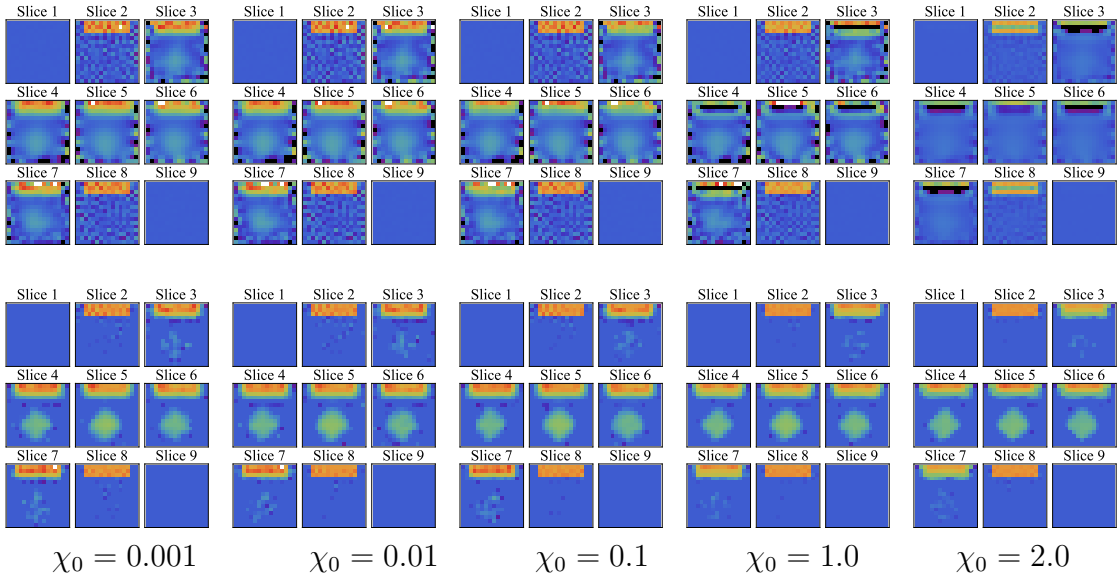


Figure 5.23: Top row first Rytov reconstructions, bottom row DCTMC reconstructions.

viable nonlinear solver. Each reconstruction does a good job of locating the chest wall phantom and correctly identifying its absorption coefficient. The cross target's absorption coefficient is also reconstructed quite accurately, but the shape of the cross is blurred. If one looks carefully, there is a cross outline in the circular region, but it is by no means sharp. However, it is clear at all levels of nonlinearity, and especially at all high levels of nonlinearity, the DCTMC reconstruction is preferred.

These reconstructions provide ample evidence that with the improvement in regularization, DCTMC can handle noisy ill-posed inverse scattering problems. Looking at the error plots of η_χ and η_Φ paints an encouraging picture. Compared with the error plots for diffraction in Figure 5.5, there is no region of slow convergence. In fact, the error curves (both error of target and error of equation) have their fastest convergence in the first few iterations, and then slow down as they converge

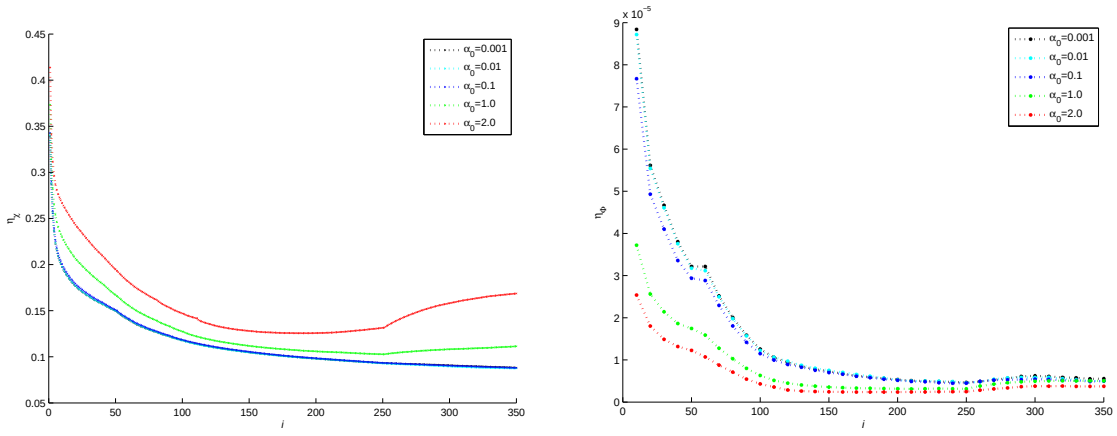


Figure 5.24: Convergence plots for the "far" target.

to the final result. The floor on the curve is much higher than before, but this is due to increased ill-posedness, noise, and regularization. For example, the strongest nonlinearity $\alpha_0 = 2.0$ initially has $\eta_\chi \approx 0.41$. After just ten iterations, this error has dropped to $\eta_\chi \approx 0.26$, a factor of about 1.5. Such steep convergence rate was unheard of in the old reconstructions until the region of fast convergence as many noninteracting voxels were found. The improved version of DCTMC has much more attractive convergence behavior, where not nearly as many iterations are needed for convergence. Perhaps more importantly, the algorithm relies less on taking advantage of sparsity, as this initial fast convergence happens before any voxels can be identified as noninteracting (which can happen at the earliest that iteration number 30 under the procedure outlined).

Note that for the stronger levels of nonlinearity, the error η_Φ begins to increase around iteration 250. Thus for these cases ($\alpha = 1.0, 2.0$), the reconstruction shown in Figure 5.23 are the results obtained after 250 iterations.

Turning to look at the second target, the "near" target, where the cross target is placed closer to the chest wall phantom, we obtain very consistent results. As displayed in Figure 5.25, the linearized reconstructions for $\alpha_0 \leq 0.1$ are all similar, albeit with the cross target place closer to the chest wall phantom, these regions blend together a bit. Again, this linear reconstruction begins to break down at $\alpha_0 = 1.0$, and provides no use at the level of $\alpha_0 = 2.0$.

The nonlinear DCTMC reconstructions are again very consistent across all levels of nonlinearity. These reconstructions and their error plots in Figure 5.26 exhibit similar behavior to the previous target. The only subtle difference is that the top of the cross target is a bit fainter due to the interaction with the chest wall phantom. Overall, the results of this section are extremely promising for DCTMC, where we were able to reconstruct images from a substantially ill posed problem, in the presence of some noise. Most importantly the convergence was fairly quick, with desirable fast initial convergence behavior which would allow one to stop the iterations much sooner if time was an issue.

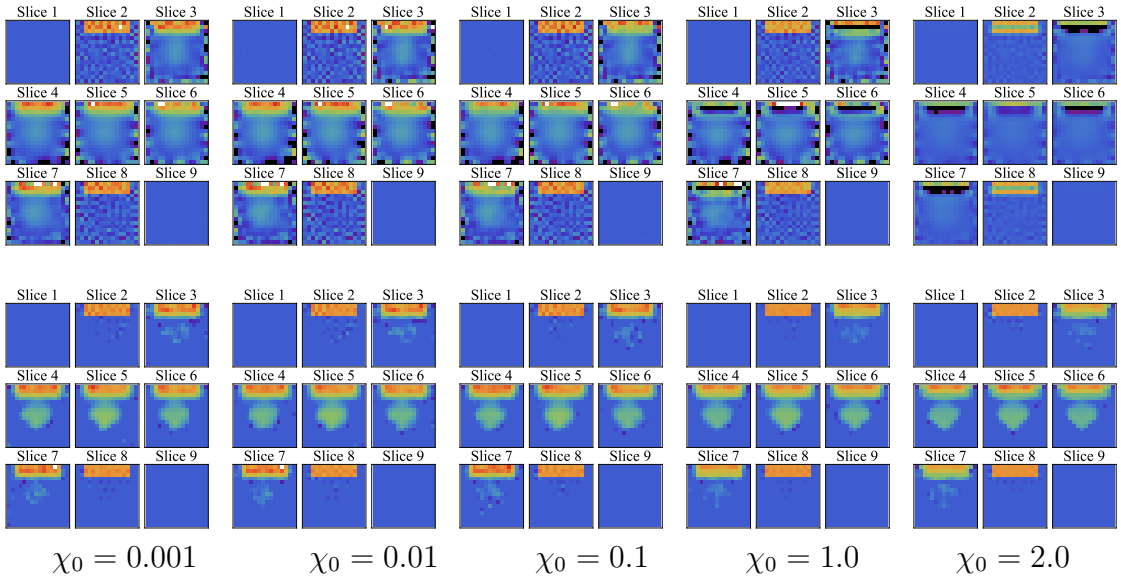


Figure 5.25: Top row first Rytov reconstructions, bottom row DCTMC reconstructions.

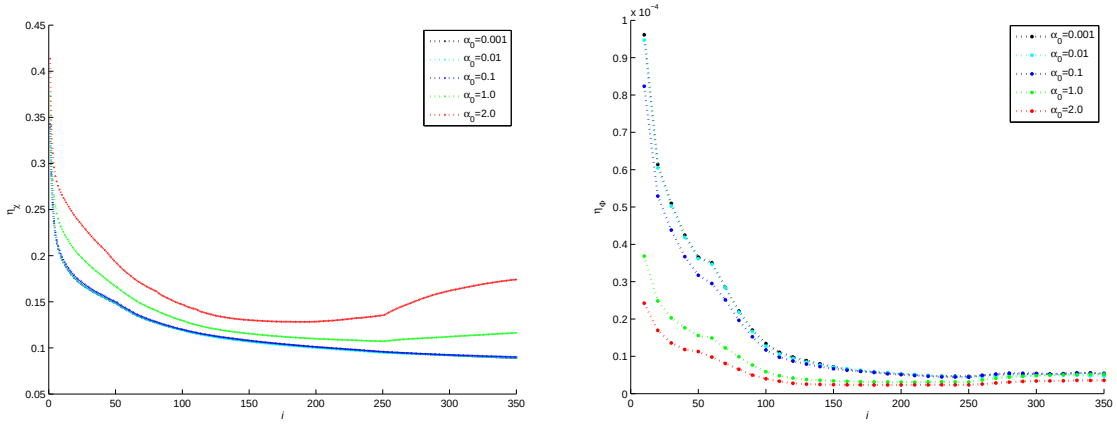


Figure 5.26: Convergence plots for the "near" target.

Chapter 6

Comparison of DCTMC and other Nonlinear Iterative Methods

6.1 Analysis of a Toy Problem

Consider the problem of reconstructing the polarizabilities α_1 and α_2 of two discrete small scatterers from the measurements produced by one source S and two detectors D_1 and D_2. Let g be the Green's function between the two scatterers, A be the Green's function from S to either of the scatterers, and B_1 and B_2 be the Green's functions from the scatterers to the detectors as is schematically illustrated in Fig. 6.1. In this case, the number of unknowns and the size of the data set are both equal to two, so that the inverse problem is perfectly determined.

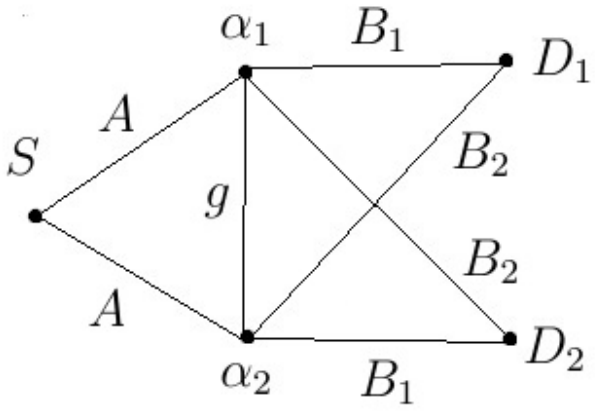


Figure 6.1: Schematic illustration of the setup of the toy problem

The forward problem for the setup considered is the set of two equations

$$d_1 = \alpha_1 (A + g d_2) \ , \tag{6.1.1a}$$

$$d_2 = \alpha_2 (A + g d_1) \ , \tag{6.1.1b}$$

where A is the incident field created by the source at the locations of the scatterers (in the case considered, the same at each scatterer) and d_1, d_2 are the induced "dipole moments" (as in the discretization process in Section 5.1.1). The detectors then measure the linear combinations

$$\phi_1 = (B_1 d_1 + B_2 d_2)/A \ \ \text{(the first detector)} \ , \tag{6.1.2a}$$

$$\phi_2 = (B_1 d_2 + B_2 d_1)/A \ \ \text{(the second detector)} \ . \tag{6.1.2b}$$

Here the normalization (division by A) is used for simplicity and does not influence any results. Our goal is to use the measurements of ϕ_1 and ϕ_2 to find α_1 and

α_2. After some manipulation, we can write the nonlinear equations coupling the unknowns α_1, α_2 to the data points ϕ_1, ϕ_2 as

$$\left(1 - g^2\alpha_1\alpha_2\right)\phi_1 = \alpha_1(1 + g\alpha_2) + \beta\alpha_2(1 + g\alpha_1) \ , \tag{6.1.3a}$$

$$\left(1 - g^2\alpha_1\alpha_2\right)\phi_2 = \alpha_2(1 + g\alpha_1) + \beta\alpha_1(1 + g\alpha_2) \ , \tag{6.1.3b}$$

where $\beta = B_2/B_1$. These equations can be solved as long as $\beta^2 \neq 1$, resulting in

$$\alpha_1 = \frac{\beta\phi_2 - \phi_1}{\beta^2 - 1 + g(\beta\phi_1 - \phi_2)} \ , \tag{6.1.4a}$$

$$\alpha_2 = \frac{\beta\phi_1 - \phi_2}{\beta^2 - 1 + g(\beta\phi_2 - \phi_1)} \ , \tag{6.1.4b}$$

as well as the spurious solution $\alpha_1 = \alpha_2 = -1/g$. If $\beta^2 = 1$, application of (6.1.4) to perfect data results in a 0/0-type uncertainty while application to noisy data will result in the unwanted solution $\alpha_1 = \alpha_2 = -1/g$. We therefore assume that $\beta^2 \neq 1$. Then the reconstructions depend on the data continuously except for the lines $\beta\phi_1 - \phi_2 = (1-\beta^2)/g$ and $\beta\phi_2 - \phi_2 = (1-\beta^2)/g$, where the inversion formulas are singular.

We will now provide the relevant formulation for this toy problem using DCTMC. First note that $\Gamma_{12} = \Gamma_{21} = g$, $\Gamma_{11} = \Gamma_{22} = 0$. The defining characteristic of DCTMC is the experimental T-matrix, which can be calculated as

$$T_{\text{exp}} = \frac{1}{2}\begin{pmatrix} \alpha_1^{(L)} & \alpha_1^{(L)} \\ \\ \alpha_2^{(L)} & \alpha_2^{(L)} \end{pmatrix}. \tag{6.1.5}$$

where $\alpha_1^{(L)}$ and $\alpha_2^{(L)}$ are the approximate "linearized" solutions to the equations in

(6.1.3) given by

$$\alpha_1^{(L)} = \frac{\beta\phi_2 - \phi_1}{\beta^2 - 1} \ , \tag{6.1.6a}$$

$$\alpha_2^{(L)} = \frac{\beta\phi_1 - \phi_2}{\beta^2 - 1} \ . \tag{6.1.6b}$$

These linearized solutions can be obtained from (6.1.4) by setting $g = 0$.

Overwriting the T-matrix in an iteration to maintain data-compatibility is now defined by the operation

$$\mathcal{O}[T] = T + \begin{pmatrix} a & a \\ & \\ b & b \end{pmatrix} \ , \tag{6.1.7}$$

where the terms a and b are selected so that the row sums of $\mathcal{O}[T]$ are equal to $\alpha_j^{(L)}$. As can be easily verified, this will enforce data compatibility of the T-matrix. The terms a and b can be written as

$$a = \frac{1}{2}\alpha_1^{(L)} - \frac{1}{2}(t_{11} + t_{12}) \ , \tag{6.1.8a}$$

$$b = \frac{1}{2}\alpha_2^{(L)} - \frac{1}{2}(t_{21} + t_{22}) \ , \tag{6.1.8b}$$

where t_{ij} are the elements of the matrix T. The first immediate remark is that this attractive overwriting scheme in the reduced toy problem is equivalent to rotating the T-matrix to singular vector representation and force overwriting the known elements (in this case $\tilde{T}_{11}$ and $\tilde{T}_{21}$). This provides evidence that our definition of overwriting is a natural choice, and is not merely the most computationally efficient choice.

With these definitions in hand, and using the "force-diagonalizing" technique of $\mathcal{D}[V]_{ij} = \delta_{ij} V_{ij}$, we can derive a mapping that is strictly equivalent to DCTMC in terms of the reconstructed values of α_j. Letting (v_1, v_2) be an initial guess or an intermediate result for the values of (α_1, α_2), the mapping that represents one iteration of DCTMC is given by the formula

$$v_1 \to v_1 - \frac{(v_1 - \alpha_1)(1 + \alpha_2 g)(g v_1 - 1)}{(1 + \alpha_2 g)(g v_1 - 1) + (1 + \alpha_1 g)(g v_2 - 1)} \ , \tag{6.1.9a}$$

$$v_2 \to v_2 - \frac{(v_2 - \alpha_2)(1 + \alpha_1 g)(g v_2 - 1)}{(1 + \alpha_2 g)(g v_1 - 1) + (1 + \alpha_1 g)(g v_2 - 1)} \ . \tag{6.1.9b}$$

In these expressions, α_1 and α_2 are the true values of the polarizabilities (not the reconstructions) and it is assumed that the data points ϕ_1, ϕ_2 represent ideal measurements that are free of noise or systematic errors.

Series Convergence

We can compare the convergence of DCTMC as a series in this simple scenario in comparision with the inverse Born series. We expand as a series the inverse solutions of (6.1.4) in powers of ϕ_1 and ϕ_2. This is the same form as under which convergence of the inverse Born series has been investigated [2, 38, 39]. We note that, for a more general ISP, a sufficient condition of convergence of inverse Born series has been obtained in [38]. For the toy problem discussed here, we can obtain a sufficient and necessary condition of convergence using these methods, which is

$$\left| \frac{g}{\beta^2 - 1} (\beta \phi_1 - \phi_2) \right| < 1 \quad \text{AND} \quad \left| \frac{g}{\beta^2 - 1} (\beta \phi_2 - \phi_1) \right| < 1 \ . \tag{6.1.10}$$

Substituting the model values of α_1, α_2 by using (6.1.3) into this condition, we can obtain a neat form. Let $x_k = g\alpha_k$, where α_k are the model values of the polarizabilities (again, not the reconstructed values). Then the convergence condition reads

$$(x_1, x_2) \in \Omega_1 \ : \ \left\{ \left| \frac{x_2(1+x_1)}{1-x_1x_2} \right| < 1 \ \text{ AND } \ \left| \frac{x_1(1+x_2)}{1-x_1x_2} \right| < 1 \right\} \ . \tag{6.1.11}$$

(Sufficient and necessary condition for convergence of inverse Born series)

This convergence stated above are illustrated in Fig. 6.2 (left panel) below.

We can now compare this convergence result for the inverse Born series to a similar result for DCTMC. However, for DCTMC we can obtain only the sufficient condition of convergence. Thus, we will compare the sufficient condition of convergence of DCTMC with the sufficient and necessary condition of convergence of the inverse Born series. It should be kept in mind that DCTMC can converge outside of the region of parameters defined by the sufficient condition of convergence. For the toy problem considered, DCTMC can be run analytically for a few iterations, and these results can be used to prove a sufficient condition for convergence of the iterations. This condition is

$$(x_1, x_2) \in \Omega_2 \ : \ \{|x_1| < 1 \ \text{ AND } \ |x_2| < 1\} \ . \tag{6.1.12}$$

(Sufficient condition for convergence of DCTMC)

This defines a square region between the lines $x_1 = \pm 1$ and $x_2 = \pm 1$, which partially

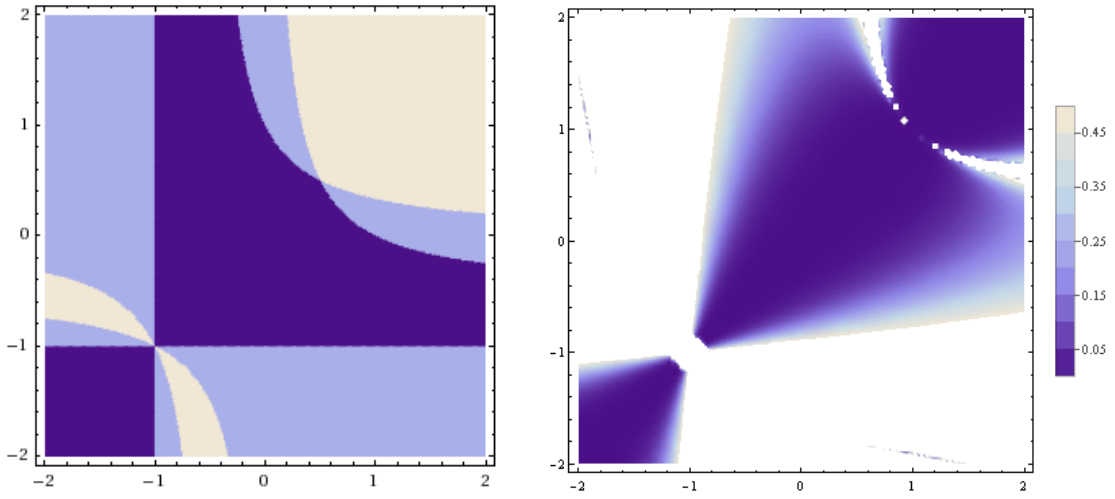


Figure 6.2: **Left:** Illustration of the region Ω_1 defined by the inequalities in (6.1.11). The axes of the plot are x_1 and x_2. Both inequalities are satisfied in the dark blue region, which is (a part of) Ω_1. Only one inequality is satisfied in the light blue region. None are satisfied outside of the light blue region. The region Ω_2 is a square with the the center at the origin and vertices at $(-1,-1)$ and $(1,1)$ (not shown explicitly in the plot). This figure illustrates convergence conditions only for purely real α_k's. **Right:** Relative error E (6.1.13) after four iterations of DCTMC for the toy problem. Same axes as in the left panel are used.

overlaps with Ω_1.

In practice, we have observed that DCTMC converges in a much wider area than the one defined by the sufficient condition (6.1.12). In particular, the method appears to converge for most positive values x_1 and x_2 in just a few iterations. This is illustrated in Fig. 6.2 (right panel), where we plot the relative error

$$E = \sqrt{\left|\alpha_1^{\text{rec}}/\alpha_1^{\text{mod}} - 1\right|^2 + \left|\alpha_2^{\text{rec}}/\alpha_2^{\text{mod}} - 1\right|^2}\,. \tag{6.1.13}$$

Here "rec" and "mod" refer to reconstruction and model.

Proof of Convergence

We can prove convergence for DCTMC for $(x_1, x_2) \in \Omega_2$ using a simple algebraic proof. Again, let v_j be the intermediate reconstruction results. We are interested in showing that the direction

$$\mathbf{u} = \frac{1}{(1+\alpha_2 g)(gv_1 - 1) + (1+\alpha_1 g)(gv_2 - 1)} \begin{pmatrix} (v_1 - \alpha_1)(1+g\alpha_2)(gv_1 - 1) \\ \\ (v_2 - \alpha_2)(1+g\alpha_1)(gv_2 - 1) \end{pmatrix} \tag{6.1.14}$$

given by the second terms in (6.1.9) is a descent direction. We will, in fact, prove a slightly stronger condition that the result of an iteration $(v_1, v_2) \to (v_1', v_2')$ satisfies

$$|v_1' - \alpha_1| \leq |v_1 - \alpha_1| \quad \text{AND} \quad |v_2' - \alpha_2| \leq |v_2 - \alpha_2| \ . \tag{6.1.15}$$

We will only prove the inequality for α_1 and the inequality for α_2 is obtained by permutation of indices. The confirmation of a descent direction implies convergence due to the fact that the only fixed point in this region is the desired solution. For simplicity we will assume that $0 < \alpha_1, \alpha_2, v_1, v_2 < 1/g$. All other configurations using combinations of negative counterparts are identical calculations.

Without loss of generality, let $\alpha_1 < v_1$. We first show that under our assumptions, the denominator in (6.1.14) is negative. That is,

$$g(v_1 + v_2\alpha_2(gv_1 - 1) + \alpha_1(gv_2 - 1)) - 2 < 0 \ .$$

since the terms $(-2/g + v_1 + v_2)$, $\alpha_1(gv_2 - 1)$, and $\alpha_2(gv_1 - 1)$ are individually negative due to the fact that $gv_j < 1$ and the positivity of all elements. The

factored numerator is clearly negative under the assumption $\alpha_1 < v_1$, which proves that $v_1' < v_1$.

Now, to show that $\alpha_1 < v_1'$, we can reduce this inequality to

$$g(v_1 + v_2\alpha_2(gv_1 - 1) + \alpha_1(gv_2 - 1)) - 2 \leq (1 + \alpha_2 g)(gv_1 - 1) \ , \tag{6.1.16}$$

and then simply by expanding out these terms and canceling repeats, we obtain the inequality

$$(1 + \alpha_1 g)(gv_2 - 1) \leq 0 \ . \tag{6.1.17}$$

This inequality $\alpha_1 < v_1' < v_1$ proves that not only does $v_1 \to \alpha_1$, but by monotone convergence that the DCTMC algorithm in this region never "overshoots" the model values of the scatterers during the iterations. Thus DCTMC is a straightforward convergent algorithm in the region $|x_1|, |x_2| < 1$.

Nonlinear Least Squares Approach

Considering the toy problem as a nonlinear least squares problem, we can naturally compare DCTMC with the Gauss-Newton method. From the data equations, we can calculate the Jacobian in terms of the model scatterers (α_1, α_2) and their reconstructed values (v_1, v_2) as

$$J = \frac{1}{1 - \alpha_1\alpha_2 g^2} \begin{pmatrix} 1 - \alpha_1\alpha_2 g^2 + g(1 + \alpha_1 g)v_2 & g(1 + \alpha_1 g)v_1 \\ g(1 + \alpha_2 g)v_2 & 1 - \alpha_1\alpha_2 g^2 + g(1 + \alpha_2 g)v_1 \end{pmatrix} . \tag{6.1.18}$$

Then, a single iteration from $\{v_1, v_2\}$ using the Gauss-Newton algorithm $J^{-1}R$ results in the mapping

$$v_1 \to v_1 - \frac{(v_1 - \alpha_1)(1 + \alpha_2 g)(1 + g v_1)}{1 + g(v_1 + \alpha_2 g v_1 + v_2 + \alpha_1 g(v_2 - \alpha_2))} , \tag{6.1.19a}$$

$$v_2 \to v_2 - \frac{(v_2 - \alpha_2)(1 + \alpha_1 g)(1 + g v_2)}{1 + g(v_1 + \alpha_2 g v_1 + v_2 + \alpha_1 g(v_2 - \alpha_2))} . \tag{6.1.19b}$$

Comparing (6.1.19) to (6.1.9) shows that the DCTMC algorithm is similar in structure to Gauss-Newton. A natural question arises, namely, is there a matrix $\tilde{J}$ such that DCTMC can be written as $y = \tilde{J}^{-1}R$? It turns out the we can explicitly calculate such a matrix, using the Gauss-Newton result as a template. We then obtain that

$$\tilde{J}^{-1} = C \begin{pmatrix} \frac{gv_1 - 1}{gv_1 + 1}(\alpha_2 g^2(\alpha_1 - v_1) - g v_1 - 1) & \frac{gv_1 - 1}{gv_1 + 1}(g v_1(1 + \alpha_1 g)) \\ \frac{gv_2 - 1}{gv_2 + 1}(g v_2(1 + \alpha_2 g)) & \frac{gv_2 - 1}{gv_2 + 1}(\alpha_1 g^2(\alpha_2 - v_2) - g v_2 - 1) \end{pmatrix} , \tag{6.1.20}$$

where

$$C = \frac{-1}{g(v_1 + v_2 + \alpha_2(g v_1 - 1) + \alpha_1(g v_2 - 1)) - 2} . \tag{6.1.21}$$

This result signifies a closed form method equivalent to DCTMC for this toy problem. That is, we have the normal equations for the DCTMC solution to nonlinear least squares. While it is useful to have a closed form algorithm for DCTMC, the ultimate question is how does the matrix $\tilde{J}$ relate to the Jacobian matrix J. As of now, the best conclusion is that there exists a diagonal scaling matrix D such that $JD = \tilde{J}$. Fully understanding the properties of $\tilde{J}$ and its relationship to J is an important line of ongoing research.

While the results thus far have used the diagonalizing operator $\mathcal{D}[V]_{ij} = \delta_{ij} V_{ij}$, it is worthwhile to consider another method of diagonalizing. We will also investigate diagonalizing by summing over rows ($\mathcal{D}[V]_{ij} = \delta_{ij} \sum_k V_{ik}$), which is inspired by the nonlocal formation of the problem used for DCTMC. This method results in a separate closed form, governed by the mapping

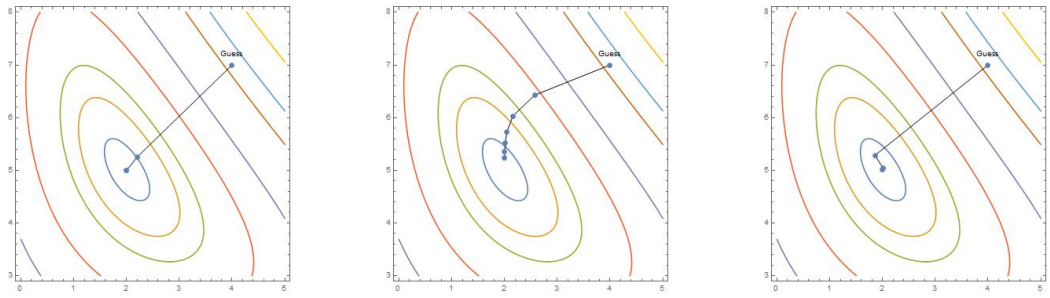
$$v_1 \to v_1 - \frac{(v_1 - \alpha_1)(1 + \alpha_2 g)(g(v_1 + v_2) - 2)}{(1 + \alpha_2 g)(g v_1 - 1) + (1 + \alpha_1 g)(g v_2 - 1)} , \tag{6.1.22a}$$

$$v_2 \to v_2 - \frac{(v_2 - \alpha_2)(1 + \alpha_1 g)(g(v_1 + v_2) - 2)}{(1 + \alpha_2 g)(g v_1 - 1) + (1 + \alpha_1 g)(g v_2 - 1)} . \tag{6.1.22b}$$

Comparing (6.1.19)-(6.1.22), we see that they all share the fixed point (α_1, α_2) (the correct solution), but differ on the spurious solutions that prevent global convergence. Gauss-Newton iteration has the fixed point $(-1/g, -1/g)$, which is the natural unwanted solution produced by the data equations, whereas DCTMC has shifted this to the fixed points $(1/g, \alpha_2)$ and $(\alpha_1, 1/g)$. The modified version of DCTMC (with row-wise summation) has the spurious solutions on the line $v_1 + v_2 = 2/g$.

Numerical Results

The first simulation was conducted to determine if convergence was correct in the region $|x_1|, |x_2| < 1$. The model was set to $\alpha_1 = 2$ and $\alpha_2 = 5$, and $g = 0.1$. An initial guess of $(\alpha_1, \alpha_2) = (4, 7)$ was chosen which is within the bounds set by $1/g = 10$. Plots depicting the iteration results overlaid on a contour plot of the underlying residual function are shown in Fig. 6.3. The DCTMC iteration does indeed converge to the correct result, and never overshoots in any direction



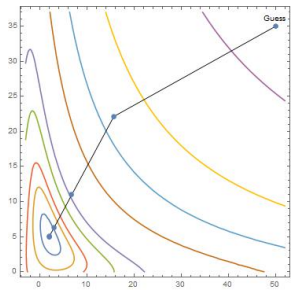
(a) Gauss-Newton (b) DCTMC (c) DCTMC with Row Sums

Figure 6.3: A comparison of iterative solvers for $\alpha_1 = 2, \alpha_2 = 5, g = 0.1$ with initial guess $v_1 = 4, v_2 = 7$. The axes show (v_1, v_2).

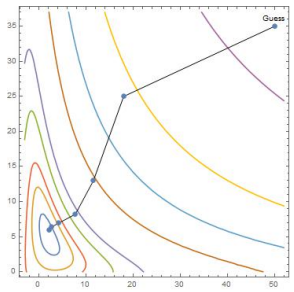
as was proved earlier. However, compared to Gauss-Newton and the modified row sums version of DCTMC, the DCTMC iterations converge the slowest, taking seven iterations as opposed to two.

Starting from an initial guess well outside the proven region of convergence, we see in Fig. 6.4 that DCTMC still converges, and again in a fairly straightforward manner. As previously mentioned, there is significant evidence that DCTMC converges in a much larger region. However, again DCTMC is the slowest to converge, requiring ten iterations to replicate the scatterers within an accepted error. Gauss-Newton method takes four iterations to converge, while the modified DCTMC algorithm converges the fastest in only three. It is worth noting that the modified DCTMC version does overshoot the model with its first iteration.

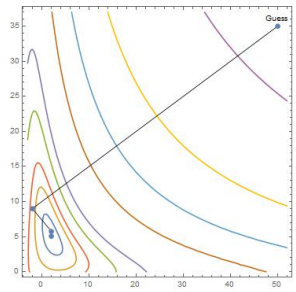
We have seen the comparatively slow convergence of DCTMC, and the perhaps faster than quadratic convergence of modified DCTMC, but what about resistance to converging to local minima (unwanted solutions)? If we take our initial guess as (-5,-5), which is relatively close to the incorrect solution of (-1/g,-1/g), we see



(a) Gauss-Newton

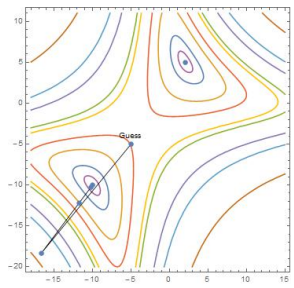


(b) DCTMC

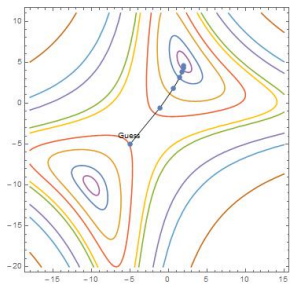


(c) DCTMC with Row Sums

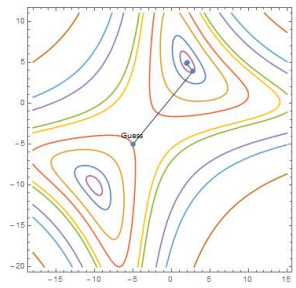
Figure 6.4: A comparison of iterative solvers for $\alpha_1 = 2, \alpha_2 = 5, g = 0.1$ with the initial guess $(v_1 = 50, v_2 = 35)$.



(a) Gauss-Newton



(b) DCTMC



(c) DCTMC with Row Sums

Figure 6.5: A comparison of iterative solvers for $\alpha_1 = 2, \alpha_2 = 5, g = 0.1$ with initial guess $(v_1 = -5, v_2 = -5)$, which is closer to the unwanted solution of $(-1/g, -1/g) = (-10, -10)$.

in Fig. 6.5 that Gauss-Newton does indeed converge to the wrong result. Both DCTMC algorithms converge correctly, with the modified version again performing much better.

But this makes perfect sense that DCTMC wouldn't converge to the minimum at $(-1/g, -1/g)$ as this is not a fixed point for the algorithm. What if our initial guess is very close to the fixed point $(1/g, \alpha_2) = (10, 5)$? Let the iterations begin with $(10.3, 4.8)$, and we do see convergence in Fig. 6.6, albeit with a very strange behavior. After a few iterations near the unwanted solution, the algorithm shoots off to around $\alpha_1 = 160$ before settling back towards the correct answer in 10 iterations. This

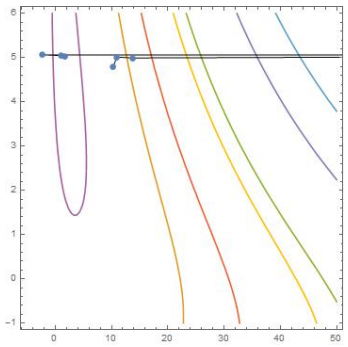


Figure 6.6: DCTMC iterations for $\alpha_1 = 2, \alpha_2 = 5, g = 0.1$ with initial guess $v_1 = 10.3, v_2 = 4.8$, which is closer to the unwanted solution of $(1/g, \alpha_2) = (10, 5)$.

provides some evidence that even though DCTMC is slower than Gauss-Newton, it is more resistant towards converging to spurious solutions.

Conclusions

Because Gauss-Newton solves the local linearization of the problem in one step, it is clear that our simulations that do not start from the linear solution are biased towards this method. As proved in Section 4.1, the linearization of DCTMC solves this problem by the Richardson iteration, which converges albeit quite slowly. Thus the slow convergence for DCTMC in these numerical examples can be greatly reduced by starting from the linearized solution. It is important to note that this linearized solution does not have to be linearized around the origin, just as in Gauss-Newton we can linearize around any reasonable initial guess. Despite this handicap, it is important to note that DCTMC still converges to the correct solution. Perhaps more remarkable is the fact that the improved version of DCTMC outperforms Gauss-Newton despite this handicap.

While the analysis, showed comparable convergence regions between DCTMC and the inverse Born series, the convergence radius using DCTMC was found to be much larger than the area defined as Ω_2. We stress that we were not able to find an initial guess for which the method diverges. There is ample evidence that there are scenarios where DCTMC appears to converge even when Gauss-Newton and/or the inverse Born series diverges.

6.2 Simulations of DCTMC vs. Newton-type Methods

In this section, we investigate diffuse optical tomography simulations in the slab geometry comparing DCTMC and the mainstream approaches based on Newton's method. We investigate a much smaller target, in order to be able to run numerous simulations to find optimal results for each method. To that end, the target was discretized on a $8 \times 8 \times 4$ grid for a total of $N_v = 256$ voxels. Surfaces of sources and detectors were placed on either side of the sample in the near-field zone at a distance of $h/2$ away. These grids were of size 10×10, ($N_d = N_s = 100$) and again centered about the sample on all sides.

To increase the nonlinearity present in the problem, we have not assumed that the background values are unknown, and have thus used the mathematically equivalent property of the background differing from free space. Thus, we do not use any

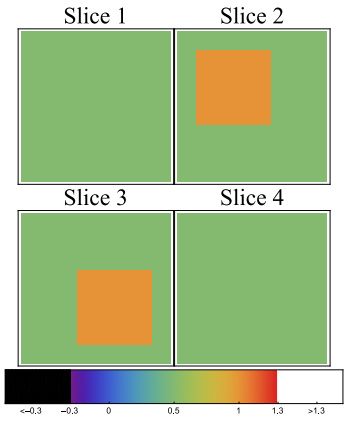


Figure 6.7: Target for the reconstructions in this section. Contrary to the previous reconstructions, the background is not equal to free space and has absorption coefficient $0.5\alpha_0$. The inhomogeneity formed by two offset square inclusions has absorption coefficient of α_0.

sparsity reduction procedures. The target had one inhomogeneity formed by two overlapping rectangles of dimension $4 \times 4 \times 1$. These two square regions overlap in the central 2×2 area. In terms of the shape function description, $\Theta(\mathbf{r}) = 1$ across the inhomogeneity and $\Theta(\mathbf{r}) = 0.5$ in the background. The target used is shown in Figure 6.7.

There were three reconstruction methods tested for this section, DCTMC (with all of its improvements), Gauss-Newton, and Levenburg-Marquardt (iteratively regularized Gauss-Newton). These two Newton-type methods were chosen as Gauss-Newton is the most natural first attempt for nonlinear solvers, and Levenburg-Marquardt is often the goto choice for DOT reconstructions [19]. For each reconstruction, extensive testing was done to optimize any regularization parameters, including number of iterations run (with a maximum of 100 needed for this small target). Between reconstructions that only differed in the method of solving, the

same initial guess was used. We did not test any linear solvers in this section, as we are interested in how DCTMC compares to other nonlinear methods. For this reason, we chose to test three substantial levels of nonlinearity, $\alpha_0 = 0.1, 1.0$ and 3.0.

6.2.1 Noiseless Reconstructions

We first test reconstructions without noise, which will allow us to compare pure convergence behavior between the methods. The reconstruction results are shown in Figure 6.8. The immediate reaction is that for the cases $\alpha_0 = 0.1$ and 1.0, the Newton type methods produce perfect reconstructions while the DCTMC reconstruction is reasonable, but contains some artifacts. Because the sample is so small, there is much less ill-posedness compared to the reconstructions in Chapter 5, and one could reasonably expect that the DCTMC algorithm should produce near flawless results. Thus, this convergence to a blurred result hints at the inherent regularization present in the DCTMC algorithm, as discussed in Section 6.1. Here, the algorithms without any regularization (Gauss-Newton) or minimal regularization (the choice of λ in (2.2.15) was very small) demonstrate that the problem is not too ill-posed to solve.

As we increase the nonlinearity to $\alpha_0 = 3.0$, we see a decrease in accuracy for all three methods, albeit each to different degrees. The Gauss-Newton method converged to a unwanted spurious solution (as can be seen in the matrix equa-

tion error) where all voxels are reconstructed much larger than their actual values. This is the case of our initial guess being too far away for Gauss-Newton to converge to the correct solution. The DCTMC algorithm reconstruction is actually quite similar to its reconstructions for the lower valleys of nonlinearity, but with lower levels of reconstructed values for the contrast. The reconstruction obtained from Levenburg-Marquardt is noticeably worse than its previous reconstructions, and thus this largest leveled nonlinearity introduces some trouble. One could argue in favor of either the DCTMC or Levenburg-Marquardt reconstruction – the Levenburg-Marquardt reconstruction comes much closer to the actual values of the absorption coefficient, but the unwanted artifact is much stronger. It is clearer where the inhomogeneity is located in the DCTMC reconstruction. Depending on the application, either choice can be warranted.

Looking at the error plots for these reconstructions paints a very similar story. It is clear for that the cases $\alpha_0 = 0.1$ and 1.0, the Newton-type methods solve the inverse problem within a few iterations and both errors (η_χ and η_Φ) rapidly go to zero. The DCTMC reconstructions for these cases converge to near zero for η_Φ but reach a floor in terms of the error of target. For the case $\alpha_0 = 3.0$, Levenburg-Marquardt converges slightly faster in terms of η_χ, and surprisingly has η_Φ converge to zero extremely fast despite its flawed reconstruction. While the final error of the Levenburg-Marquardt reconstruction is slightly lower than the DCTMC error, one could still prefer either image.

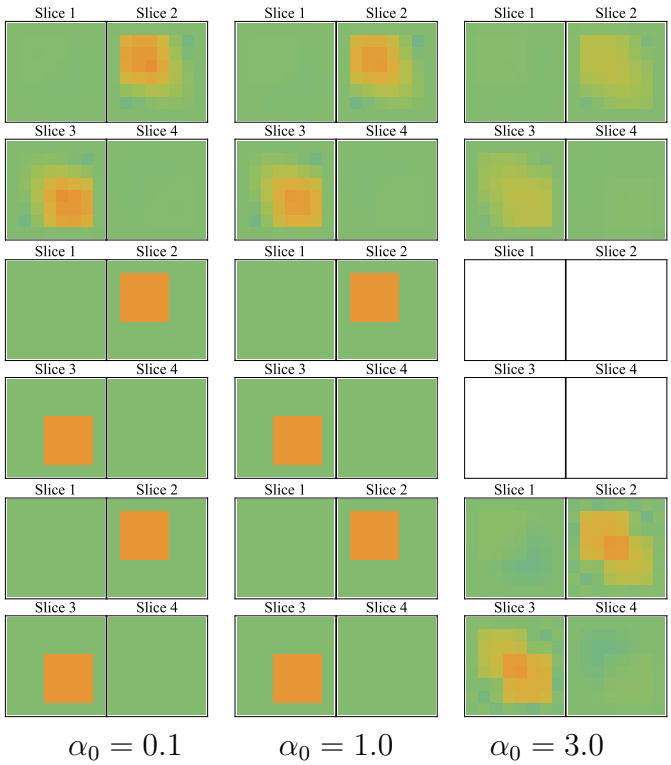


Figure 6.8: Reconstructions for the three methods with noiseless data. The top row depicts the DCTMC reconstructions, the middle row contains the reconstructions obtained from Gauss-Newton method, and the Levenburg-Marquardt reconstructions are in the bottom row. Note that the Gauss-Newton reconstruction for $\alpha_0 = 3.0$ is all white as all reconstructed values are larger than the cutoff $\alpha_n/\alpha_0 > 1.3$.

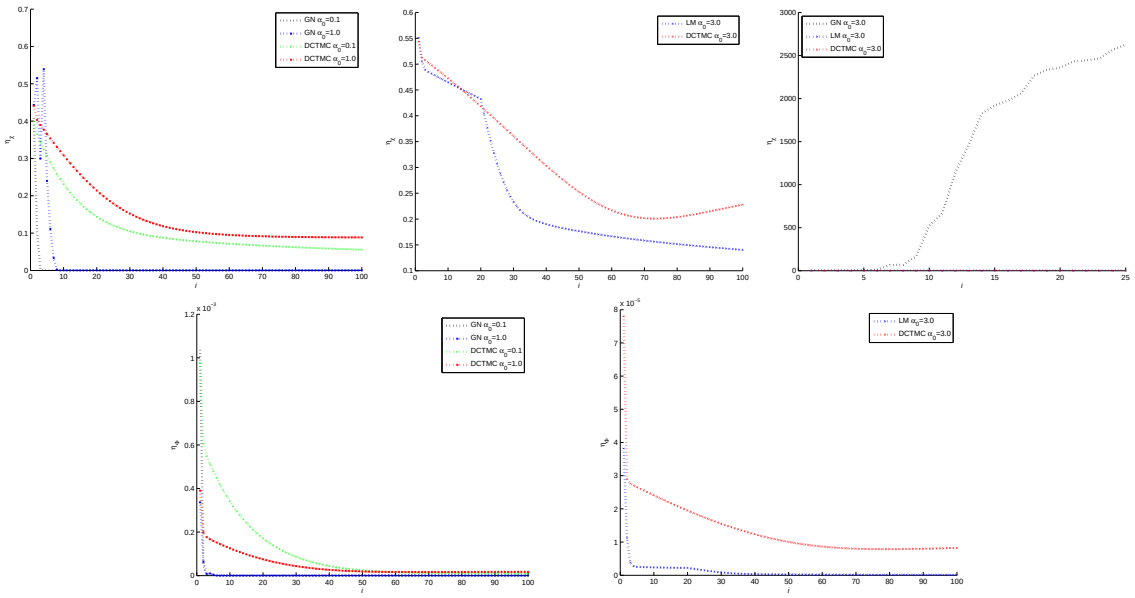


Figure 6.9: Error plots for the noiseless reconstructions. The top row plots η_χ against iterations, while the bottom row plots the error the equation η_Φ. The non-converging Gauss-Newton iterations for the case $\alpha_0 = 3.0$ are left out of the middle plot in the top row, but instead displayed in the last plot in this row.

The conclusions to be made from this section are that, in general, the Newton methods are superior to DCTMC in reconstructing these noiseless, slightly ill-posed problems. However, at strong levels of nonlinearity, DCTMC demonstrates a larger convergence radius than Gauss-Newton, as well as comparable results to Levenburg-Marquardt. For these unrealistic inverse scattering problems with perfect data, DCTMC is inferior at lower levels of nonlinearity, but we will see in the next section its merits with more realistic simulations.

6.2.2 Noisy Reconstructions

We now replicate the reconstructions from the previous section but after adding 2% Gaussian noise to the data. The results obtained are shown in Figure 6.10. Once the noise is added, it seems like DCTMC is the preferred method! Is clear that without any regularization, Gauss-Newton fails to produce any reasonable results, and is the worst of the three options. Again, at the highest level of nonlinearity tested, Gauss-Newton reconstructs all voxels above the cutoff. The Levenburg-Marquardt reconstructions are consistent across all levels of nonlinearity, and produce reasonable results that indicate both the location of the inclusion, as well as its value. However, the DCTMC reconstructions in the top row suppress the noise in a useful manner to clearly highlight the uniform contrast. However, we again have at the contrast $\alpha_0 = 3.0$ a significantly reduced reconstructed value. One might prefer the Levenburg-Marquardt reconstruction at this highest level nonlinearity, but for the other two levels, the DCTMC algorithm is preferred.

We have similar behavior in the convergence plots in Figure 6.11. The Gauss-Newton error plots either exhibit oscillatory behavior, or unwanted flatline results. While the error of the Levenburg-Marquardt method both decreases faster than DCTMC, and often to a lower absolute value, the reconstruction images show that there is no substantial advantage to this method, and can often be subjectively worse.

The results from this section are very encouraging for DCTMC and further

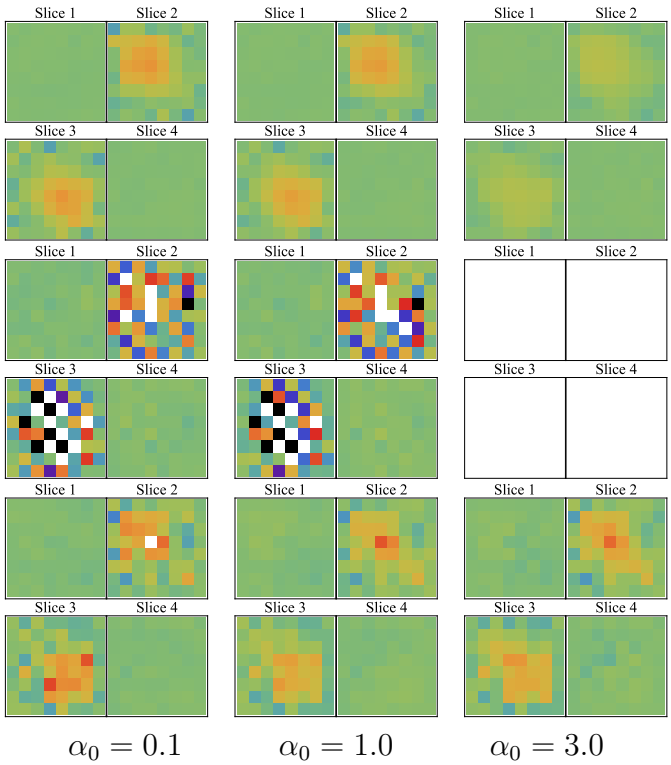


Figure 6.10: Reconstructions for the three methods with noisy data. The top row depicts the DCTMC reconstructions, the middle row contains the reconstructions obtained from Gauss-Newton method, and the Levenburg-Marquardt reconstructions are in the bottom row. Note that the Gauss-Newton reconstruction for $\alpha_0 = 3.0$ is again all white as all reconstructed values are larger than the cutoff $\alpha_n/\alpha_0 > 1.3$.

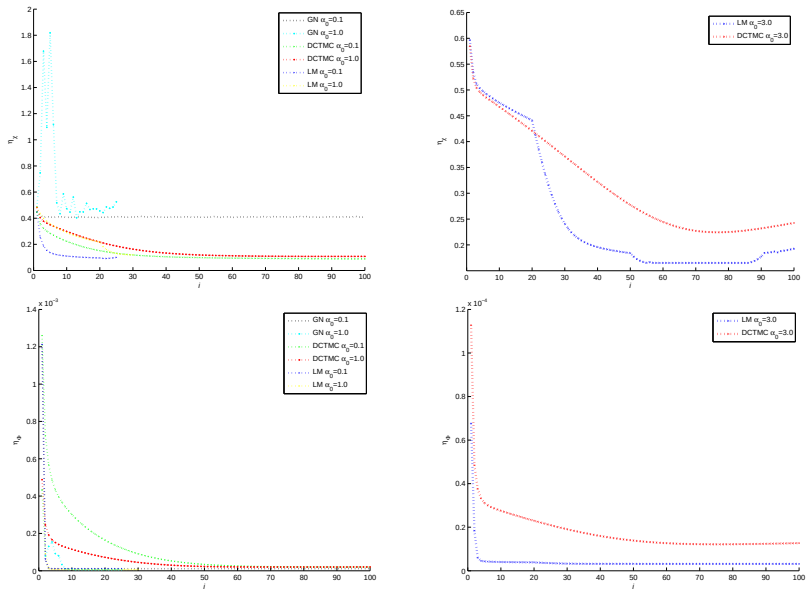


Figure 6.11: Error plots for the noisy reconstructions. The top row plots η_χ against iterations, while the bottom row plots the error the equation η_Φ. The non-converging Gauss-Newton iterations for the case $\alpha_0 = 3.0$ are left out altogether.

validates its prospect as a future nonlinear solver of choice. While previous simulations exhibited the ability of DCTMC to reconstruct ill-posed ISPs with large data sets, the investigations in this section are necessary to compare DCTMC to more than just linearized solutions. DCTMC held its own in all reconstructions against nonlinear solvers, and in several instances was slightly preferred. While both Gauss-Newton and Levenburg-Marquardt were great at reconstructing perfect data at lower levels of nonlinearity, once the simulations were modified to be more realistic, the merits of DCTMC were evident. We can conclude that DCTMC is a viable alternative to the mainstream nonlinear iterative methods.

Chapter 7

Summary and Discussion

In this thesis, we have introduced the novel iterative algorithm data compatible T-matrix completion for solving nonlinear inverse scattering problems. Motivated by the theory of nonlocality, this unique reformulation treats the T-matrix as the fundamental unknown in an underdetermined inverse problem as opposed to the conventional approach where one solves for the interaction operator in an overdetermined setting. This difference allows us to create this iterative method where the size of the data set is not a limiting factor. This has the potential to be significantly advantageous over mainstream nonlinear iterative methods, where the computational complexity grows with the data set. With modern experimental techniques that readily produce the large overdetermined data sets needed to solve inverse scattering problems with reasonable precision, DCTMC can be an important tool for solving these problems.

With Tikhonov regularization inspired by the linearized version of DCTMC, and noise suppression by limiting the known values in the experimental T-matrix, DCTMC was able to reconstruct large three-dimensional ill-posed problems with or without noise. Good results were obtained even when linear reconstructions failed, which validated the role of DCTMC as a true nonlinear solver. While accounting for sparsity was originally required to exit regions of slow convergence, enforcing symmetry of the T-matrix and improving the method of determining the local approximation to the interaction matrix V created pleasant convergence behavior with initial rapid convergence. This allowed us to cut down the required number of iterations by a factor of 10. This was key to being able to solve the strongly ill-posed problems common in diffuse optical tomography in a reasonable amount of time.

Comparisons with Gauss-Newton demonstrated a larger convergence radius for DCTMC, as well as better convergence with noisy data. The regularized Levenburg-Marquardt was on equal footing with DCTMC depending on the setting. DCTMC can be computationally advantageous over these methods with large data sets, and the results indicate that it performs at least as well. Thus, we have demonstrated that DCTMC is a viable alternative to solving nonlinear inverse scattering problems.

Bibliography

[1] L. Adams, J. Nazareth, A. Society, I. Statistics, and S. Mathematics. *Linear and Nonlinear Conjugate Gradient-related Methods*. Proceedings in Applied Mathematics Series. Society for Industrial and Applied Mathematics, 1996.

[2] S. Arridge, S. Moskow, and J. C. Schotland. Inverse born series for the calderon problem. *Inverse Problems*, 28(3):035003, 2012.

[3] S. R. Arridge and J. C. Schotland. Optical tomography: forward and inverse problems. *Inverse Problems*, 25(3):123010, 2009.

[4] S. R. Arridge and M. Schweiger. Photon-measurement density functions. part 2: Finite-element-method calculations. *Appl. Opt.*, 34(34):8026–8037, Dec 1995.

[5] S. R. Arridge, M. Schweiger, M. Hiraoka, and D. T. Delpy. A finite element approach for modeling photon transport in tissue. *Medical Physics*, 20(2), 1993.

[6] H. Y. Ban, D. R. Busch, S. Pathak, F. A. Moscatelli, M. Machida, J. C. Schotland, V. A. Markel, and A. G. Yodh. Diffuse optical tomography in the

presence of a chest wall. *Journal of Biomedical Optics*, 18(2):026016–026016, 2013.

[7] G. Bao and P. Li. Numerical solution of inverse scattering for near-field optics. *Optics letters*, 32(11):1465–1467, 2007.

[8] K. Belkebir, P. C. Chaumet, and A. Sentenac. Superresolution in total internal reflection tomography. *J. Opt. Soc. Am. A*, 22(9):1889–1897, Sep 2005.

[9] K. Belkebir, P. C. Chaumet, and A. Sentenac. Influence of multiple scattering on three-dimensional imaging with optical diffraction tomography. *J. Opt. Soc. Am. A*, 23(3):586–595, Mar 2006.

[10] W. B. Beydoun and A. Tarantola. First born and rytov approximations: Modeling and inversion conditions in a canonical example. *The Journal of the Acoustical Society of America*, 83(3), 1988.

[11] D. Boas, D. Brooks, E. Miller, C. DiMarzio, M. Kilmer, R. Gaudette, and Q. Zhang. Imaging the body with diffuse optical tomography. *Signal Processing Magazine, IEEE*, 18(6):57–75, 2001.

[12] P. Bonfert-Taylor, F. Leblond, R. W. Holt, K. Tichauer, B. W. Pogue, and E. C. Taylor. Information loss and reconstruction in diffuse fluorescence tomography. *J. Opt. Soc. Am. A*, 29(3):321–330, Mar 2012.

[13] L. Borcea. Electrical impedance tomography. *Inverse Problems*, 18(6):R99, 2002.

[14] F. Borghese, P. Denti, and R. Saija. *Scattering from Model Nonspherical Particles: Theory and Applications to Environmental Physics*, chapter Multipole Expansions and Transition Matrix, pages 73–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[15] M. M. Bronstein, A. M. Bronstein, M. Zibulevski, and H. Azhari. Reconstruction in diffraction ultrosound tomography using nonuniform FFT. 21:1395–1401, 2002.

[16] P. S. Carney, R. A. Frazin, S. I. Bozhevolnyi, V. S. Volkov, A. Boltasseva, and J. C. Schotland. Computational lens for the near field. *Physical review letters*, 92(16):163903, 2004.

[17] G. Chavent. *Nonlinear Least Squares for Inverse Problems: Theoretical Foundations and Step-by-Step Guide for Applications*. Scientific Computation. Springer Netherlands, 2010.

[18] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Applied Mathematical Sciences. Springer, 1998.

[19] H. Dehghani, M. E. Eames, P. K. Yalavarthy, S. C. Davis, S. Srinivasan, C. M. Carpenter, B. W. Pogue, and K. D. Paulsen. Near infrared optical tomogra-

phy using nirfast: Algorithm for numerical model and image reconstruction. *Communications in Numerical Methods in Engineering*, 25(6):711–732, 2009.

[20] A. Devaney. *Mathematical Foundations of Imaging, Tomography and Wavefield Inversion*. Mathematical Foundations of Imaging, Tomography and Wavefield Inversion. Cambridge University Press, 2012.

[21] A. J. Devaney. Geophysical diffraction tomography. *IEEE Trans. Geosci. Remote Sensing*, GE-22:3–13, 1984.

[22] B. T. Draine. Discrete-dipole approximation and its application to interstellar graphite grains. *Astrophysical Journal*, 333, 1988.

[23] B. T. Draine and P. J. Flatau. Discrete-dipole approximation for scattering calculations. *J. Opt. Soc. Am. A*, 11(4):1491–1499, Apr 1994.

[24] B. T. Draine and P. J. Flatau. Discrete-dipole approximation for periodic targets: theory and tests. *J. Opt. Soc. Am. A*, 25(11):2693–2703, Nov 2008.

[25] H. W. Engl and P. Kugler. *Multidisciplinary Methods for Analysis Optimization and Control of Complex Systems Mathematics in Industry*, volume 6, chapter Nonlinear inverse problems: Theoretical aspects and some industrial applications, pages 3–47. Springer, 2005.

[26] J. Espinoza, R. Romero, J. P. Kusanovic, F. Gotsch, W. Lee, L. F. Gonalves, and S. S. Hassan. Standardized views of the fetal heart using four-dimensional

sonographic and tomographic imaging. *Ultrasound in Obstetrics & Gynecology*, 31(2):233–242, 2008.

[27] D. Isaacson, J. L. Mueller, J. C. Newell, and S. Siltanen. Reconstructions of the chest phantoms by the d-bar method for electrical impedance tomography. 23:821–828, 2004.

[28] M. Jakobsen. T-matrix approach to seismic forward modelling in the acoustic approximation. *Studia Geophysica et Geodaetica*, 56(1):1–20, 2012.

[29] M. Jakobsen and B. Ursin. Full waveform inversion in the frequency domain using direct iterative t-matrix methods. *Journal of Geophysics and Engineering*, 12(3):400, 2015.

[30] C. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995.

[31] S. D. Konecky, G. Y. Panasyuk, K. Lee, V. Markel, A. G. Yodh, and J. C. Schotland. Imaging complex structures with diffuse light. *Opt. Express*, 16(7):5048–5060, Mar 2008.

[32] S. D. Konecky, G. Y. Panasyuk, K. Lee, V. A. Markel, A. G. Yodh, and J. C. Schotland. Optical tomography with large data sets and analytic reconstruction formulas. In *Biomedical Optics*, page BMC5. Optical Society of America, 2008.

[33] D. W. Mackowski and M. I. Mishchenko. Calculation of the t matrix and the scattering matrix for ensembles of spheres. *J. Opt. Soc. Am. A*, 13(11):2266–2278, Nov 1996.

[34] V. Markel'. Scattering of light from two interacting spherical particles. *Journal of Modern Optics*, 39(4):853–861, 1992.

[35] V. A. Markel. Erratum to the effects of averaging on the enhancement factor for absorption of light by carbon particles in microdroplets of water [jqsrt 72 (2002) 765]. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 103(2):428 – 429, 2007.

[36] V. A. Markel and J. C. Schotland. Effects of sampling and limited data in optical tomography. *Applied Physics Letters*, 81(7), 2002.

[37] V. A. Markel and J. C. Schotland. Symmetries, inversion formulas, and image reconstruction for optical tomography. *Phys. Rev. E*, 70:056616, Nov 2004.

[38] S. Moskow and J. C. Schotland. Convergence and stability of the inverse scattering series for diffuse waves. *Inverse Problems*, 24(6):065005, 2008.

[39] S. Moskow and J. C. Schotland. Numerical studies of the inverse born series for diffuse waves. *Inverse Problems*, 25(9):095007, 2009.

[40] J. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. Computational Science and Engineering. SIAM, 2012.

[41] R. Newton. *Scattering Theory of Waves and Particles*. Dover Books on Physics. Dover Publications, 1982.

[42] E. M. Purcell and C. R. Pennypacker. Scattering and Absorption of Light by Nonspherical Dielectric Grains. *Astrophysical Journal*, 186, 1973.

[43] M. Schweiger and S. R. Arridge. Direct calculation with a finite-element method of the laplace transform of the distribution of photon time of flight in tissue. *Appl. Opt.*, 36(34):9042–9049, Dec 1997.

[44] J. K. Seo and E. J. Woo. *Nonlinear Inverse Problems in Imaging*. Wiley, 2012.

[45] C. Shin, K. Yoon, K. J. Marfurt, K. Park, D. Yang, H. Y. Lim, S. Chung, and S. Shin. Efficient calculation of a partialderivative wavefield using reciprocity for seismic imaging and inversion. *GEOPHYSICS*, 66(6):1856–1863, 2001.

[46] A. SITENKO. Chapter 2 - the scattering matrix and transition probability. In A. SITENKO, editor, *Lectures in Scattering Theory*, volume 39 of *International Series in Natural Philosophy*, pages 10 – 31. Pergamon, 1971.

[47] R. Snieder. The role of nonlinearity in inverse problems. 14:387404, 1998.

[48] E. Talebian and M. Talebian. A general review on the derivation of clausius-mossotti relation. *Optik - International Journal for Light and Electron Optics*, 124(16):2324 – 2326, 2013.

[49] F. J. Tipler. Quantum nonlocality does not exist. *Proceedings of the National Academy of Sciences*, 111(31):11281–11286, 2014.

[50] Z.-M. Wang, G. Y. Panasyuk, V. A. Markel, and J. C. Schotland. Experimental demonstration of an analytic method for image reconstruction in optical diffusion tomography with large data sets. *Opt. Lett.*, 30(24):3338–3340, Dec 2005.

[51] D. Watzenig. *e & i Elektrotechnik und Informationstechnik*, volume 124, chapter Bayesian inference for inverse problems - statistical inversion, pages 240–247. Springer, 2007.

[52] A. B. Weglein, F. V. Arajo, P. M. Carvalho, R. H. Stolt, K. H. Matson, R. T. Coates, D. Corrigan, D. J. Foster, S. A. Shaw, and H. Zhang. Inverse scattering series and seismic exploration. *Inverse Problems*, 19(6):R27, 2003.

[53] T. Zhang, P. C. Chaumet, E. Mudry, A. Sentenac, and K. Belkebir. Electromagnetic wave imaging of targets buried in a cluttered medium using a hybrid inversion-dort method. *Inverse Problems*, 28(12):125008, 2012.