SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1065–1082


# ANALYSIS AND EXPLOITATION OF MATRIX STRUCTURE ARISING IN LINEARIZED OPTICAL TOMOGRAPHIC IMAGING*

DAMON HYDE†, MISHA KILMER‡, DANA H. BROOKS†, AND ERIC MILLER§

**Abstract.** We present a novel method by which the large dense forward matrix **A** involved in a linear inverse diffusion problem can be decomposed into a number of sparse easily computed matrices. We begin by introducing an errorless decomposition which is applicable to a wide array of such imaging problems. Next, we incorporate interpolation into the construction of the matrices to reduce the computational complexity involved in the matrix-vector multiplications necessary to obtain an inverse solution. Error and computational complexity analysis are provided to support these developments. We then present numerical results that illustrate the gain in computational efficiency when the approximation is used in the Tikhonov regularized inverse problem, and show that the use of the approximation has virtually no negative effect on the quality of the reconstructed images. Finally, we discuss applicability to other imaging problems.

**Key words.** structured matrix, matrix approximation, linearized inverse scattering, Tikhonov regularization, image reconstruction

**AMS subject classifications.** 65F10, 65R32, 15A99

**DOI.** 10.1137/060657285

## 1. Introduction.

In diffuse optical tomography (DOT), near-infrared light is introduced to the body from an array of sources on the surface and collected at a number of detectors as it exits [3, 7, 8, 12, 15, 17, 26, 27]. The imaging problem consists of determining images of photon absorption and/or diffusion in the body from this measured photon fluence. In this paper, we consider the problem of efficient image reconstruction from diffuse optical data. Using a linearized model of the relationship between the data and optical absorption coefficient, the specific problem we consider is the efficient solution of the Tikhonov regularized problem:

$$\min_{\mathbf{f}} \|\mathbf{Af} - \mathbf{g}\|_2^2 + \lambda\|\mathbf{Rf}\|_2^2, \tag{1.1}$$

where the real $N_{data} \times N_{vox}$ matrix **A** is a discretization of a Born-type linearized inverse scattering operator in three dimensions (i.e., the discretization of an integral operator), **f** denotes the vectored form of the absorption image to be determined, and **g** denotes the measured data vector. The regularization term $\lambda\|\mathbf{Rf}\|_2^2$ is necessary to dampen the effects of noise on the quality of the reconstructions as well as to ensure uniqueness of the solution. We employ iterative algorithms as a computationally attractive means to solve (1.1). The nature of these methods is such that the matrix **A** need be utilized only for multiplications of the form $\mathbf{Ax}$ and $\mathbf{A}^T\mathbf{x}$ for an arbitrary

*Received by the editors April 14, 2006; accepted for publication (in revised form) by D. P. O'Leary March 27, 2007; published electronically November 2, 2007. This work was funded in part by NSF grants 0139968 and 0208548, and by the Center for Subsurface Sensing and Imaging Systems (CENSSIS).
http://www.siam.org/journals/simax/29-4/65728.html
†Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 (dhyde@ece.neu.edu, brooks@ece.neu.edu).
‡Department of Mathematics, Tufts University, Medford, MA 02155 (misha.kilmer@tufts.edu).
§Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155 (elmiller@ece.tufts.edu).

vector $\mathbf{x}$, and the problem can be solved efficiently for multiple values of $\lambda$ simultaneously. This allows us to concentrate on optimizing for matrix-vector multiplications rather than the more complicated matrix-matrix multiplications.

Though a number of different nonlinear algorithms are currently in use for DOT [6, 7, 8, 29, 25, 27], the associated data sets are generally quite small (on the order of several hundred data points). Technologically, however, sensor systems are rapidly evolving to provide greater spatial density of sources and detectors as well as finer scale sampling in time or frequency. Thus, the size of the data sets available for inversion is increasing far beyond what is typically considered in the nonlinear inversion literature. Scaling these algorithms to larger data sets, such as those considered in this paper, drastically increases the required computational power and makes application of the nonlinear inverse methods either completely infeasible or quite difficult—significant effort and new research would be required in the implementation of these methods on supercomputing-type platforms. Such efforts are indeed well beyond the capabilities of many relevant research and industrial organizations (e.g., Advanced Research Technologies (ART), who provided us with the phantom data used in this paper). In such settings the linear solution is the only feasible approach to the problem. Thus, though nonlinear methods for DOT are certainly under consideration, there remains relevance to considering the linear form of the problem as well.

Toward this end, we point to recent research using linear methods for DOT and related areas where the results presented here may prove relevant [4, 9, 14, 16, 18, 19, 30]. Moreover, we emphasize that the methods we present in this paper are really only loosely dependent upon the specific form of the Green's functions. As long as (a) the Green's functions are smooth and display some level of spatial invariance and (b) the data collection scheme is also regular, the methods detailed here will be usable. This makes our work relevant to a wide range of problems, of which DOT is but a single application. For example, the ideas here could be applicable to any large scale diffusive-type inverse problem (where data set size and voxel dimensionality prevent the use of nonlinear methods) such as diffusive-type electromagnetic induction imaging arising in geophysical applications, photothermal/photoacoustic nondestructive evaluation, bioluminescence tomography (BLT), and fluorescence molecular tomography (FMT). As a specific example, take FMT [21, 20]. When imaging fluorescence, the collected fluorescence data are approximately linearly related to the image, and using methods such as the normalized Born ratio, it is possible to minimize the effects of inhomogeneities in the background optical parameters [28]. Moreover, for this problem, the development of CCD detectors and tomographic data acquisition systems is leading to imaging problems even larger than those considered here. Thus we would anticipate that large computational gains could be achieved using the method in this paper.

The difficulty in practice with solving the Tikhonov problem is that the matrix $\mathbf{A}$ is dense and extremely large, with $N_{data}$ and $N_{vox}$ being on the order of $10^4$–$10^5$. For this work, we restricted ourselves to working with a system configuration consisting of a slab transmission geometry using time domain data collection. In this configuration, the region to be imaged is modeled as the volume contained between two parallel infinite planes. The solution volume is then a compact finite section of this infinite volume [12, 15]. Sources are located along one plane, directed into the volume, while detectors are arrayed along the other plane to collect exiting light. At each source location a picosecond laser is pulsed, and the time dependent intensity is recorded at a number of detectors for time gates of several nanoseconds. Given that each detector may collect hundreds or even thousands of time points for each source pulse,

the total data set rapidly becomes extremely large. In systems of this sort, it is not uncommon to see hundreds of thousands of data points collected in a single imaging session. Thus, even with a very small number of voxels, the size of $\mathbf{A}$ rapidly becomes prohibitive. Storing such a matrix in double precision can require gigabytes of storage space and thereby renders useless any algorithm requiring the up-front computation of $\mathbf{A}$.

Therefore, our primary goal is to represent the full matrix $\mathbf{A}$ such that both the time required for its computation and the necessary storage space are significantly reduced. In particular, we look to exploit the structure of the matrix $\mathbf{A}$ so that redundant matrix entries are not explicitly computed/stored. These redundancies arise from a combination of the regularity in the data acquisition process and the structure of the Green's functions used to compute the matrix elements. As we will show, only a relatively small amount of information is needed to implicitly represent every matrix entry, so matrix-vector products can be performed on-the-fly simply by reference to a particular source-detector pattern and the small amount of stored information. However, even with a more compact representation, a significant amount of time is still required to evaluate matrix-vector products involving these matrices. Given that these products may need to be evaluated numerous times to obtain a solution, a secondary goal of this work was the reduction of the computational complexity involved in executing matrix-vector products. To do this, we effectively replace the implicit representation of $\mathbf{A}$ by an approximation that can be applied to vectors more quickly without degradation of the reconstructed images.

To achieve the first goal, we take into account the spatial invariance of the integral equation kernel from which $\mathbf{A}$ is derived. Using a change of variables, we are able to exploit the matrix redundancies more readily. This allows us to represent $\mathbf{A}$ in terms of the product of a single small matrix and a collection of sparse, easily computed matrices. This decomposition is made possible by a regular sampling pattern and planar shift invariance in the kernel.

To achieve the second goal, we introduce an interpolation approach, applied in the aforementioned coordinate system, to further reduce the number of matrix components that must be explicitly computed. The utility of the interpolation approach arises from the smoothness of the kernel of the integral equation from which $\mathbf{A}$ is derived. In turn, this suggests applicability to other diffusing imaging problems with similarly smooth kernels. By choosing an interpolation method which is expressible in matrix form, we are then able to achieve reduction in the amount of computation required to implement the matrix vector products $\mathbf{Ax}$ and $\mathbf{A}^T\mathbf{x}$. These gains are shown to be directly proportional to the number of nodes used in the interpolation scheme, allowing for a direct tradeoff between computation time and accuracy.

Applying these two steps to the overall problem at hand, we are able to obtain a dramatic decrease in the amount of time required to obtain a solution to the problem (1.1) using an iterative algorithm. We present several sets of numerical results, both with and without the use of interpolation, to approximate $\mathbf{A}$. Using phantom simulations, we show that while the two solutions are not identical, visually they are very similar, and mathematically they have effectively the same mean squared error with respect to the true image.

This paper is organized as follows. In section 2, we provide an overview of the physics and mathematics behind the construction of the matrix $\mathbf{A}$. In section 3.1, we show how to represent the matrix in compact form. Section 3.2 is devoted to presenting an interpolation-based approximation to the matrix. Section 4 details the

precise reductions in computational complexity. Inversion results using the algorithm in [13] on a simulated data set are presented in section 5 along with an analysis of the error introduced by the interpolation. Finally, in section 6, we summarize our results and outline potential further extensions.

**2. Problem description.** The diffusion approximation model arises as an approximation to the radiative transport equation [5], and takes the form [2]

$$\left(\nabla \cdot \gamma^2 \nabla - \mu_a - \frac{1}{v}\frac{\partial}{\partial t}\right)\Phi(\mathbf{r},t) = -q(\mathbf{r},t), \qquad \gamma^2 = \frac{1}{3[\mu_a + (1-\overline{p})\mu_s]}. \tag{2.1}$$

Here, $\Phi(\mathbf{r},t)$ represents the photon density at a location $\mathbf{r}$ and time $t$. Sources are represented by $q(\mathbf{r},t)$. The two physical parameters of interest are $\mu_a$ and $\mu_s$, the absorption and scattering parameters, respectively. Additionally, $v$ is the speed of light in the medium, and $\bar{p}$ is the mean cosine of the scattering angle.

The goal in this work is to recover $\mu_a$, assuming that $\mu_s$ is constant, given knowledge of the sources. Since it is clearly a nonlinear problem to recover $\mu_a$ from (2.1), the equation is frequently linearized by assuming that the overall system is approximately homogeneous. One can then use Green's functions for the homogeneous case to reformulate the imaging problem as one of finding the perturbation about some known background absorption level. Therefore, we let $\mu_a$, $\mu_s$ denote the known background values of absorption and scattering, and we use $\eta(\mathbf{r}')$ to denote the unknown perturbations of absorption about the known background value.

Assuming that the source term $q(\mathbf{r},t)$ is a delta function located at position $\mathbf{r}$ and time $t$, a solution $\Phi$ in the form of a Green's function can be derived. For the slab transmission geometry that we consider in this paper, the two-point time domain Green's function is [1]

$$\begin{aligned} g_{slab}^{(\Phi)}(\mathbf{r},\mathbf{r}',t,t_0) &= \frac{\exp\left\{-\left[\mu_a c(t-t_0) + \frac{d^2}{4\gamma^2(t-t_0)}\right]\right\}}{[4\pi\gamma^2(t-t_0)]^{3/2}} \\ &\quad \times \sum_{n=-\infty}^{\infty}\left[\exp\left(\frac{-(z-2z_d n - z_0)^2}{4\gamma^2(t-t_0)}\right)\right. \\ &\qquad \left. - \exp\left(\frac{-(z-2z_d n + z_0)^2}{4\gamma^2(t-t_0)}\right)\right], \\ &\quad d = \sqrt{x^2+y^2}, \quad \text{where } \mathbf{r}-\mathbf{r}' = (x,y,z), \\ &\quad z_o = [(1-\bar{\rho})\mu_s]^{-1}. \end{aligned} \tag{2.2}$$

This equation models the transmission of light from point $\mathbf{r}$, leaving at time $t_0$, and arriving at point $\mathbf{r}'$ at time $t$. The constant $z_d$ represents the thickness of the slab in question and is used to generate the multiple image sources needed to satisfy the boundary conditions of the system [11, 24]. The placement of these image sources results in the Green's function taking a value of zero at the boundary of the diffusive medium. Finally, the distance $z_0$ represents the source depth; because we use a slab geometry model, all sources are located at this height. It is presumed that all light from the source travels a short distance into the medium before proceeding to scatter randomly. This is modeled by assuming the sources to be isotropic and placing them one mean scattering length into the medium.

These Green's functions are used in a model of the sensing system based on the first order Born approximation [22]. This approximation assumes that the total received signal is the sum of the signal for a homogeneous system and a perturbation due to $\eta(\mathbf{r}')$, the inhomogeneities in $\mu_a$:

$$\Gamma_{total} = \Gamma_{homog} + \Delta\Gamma(\eta(\mathbf{r}')). \tag{2.3}$$

Using the background optical properties, $\Gamma_{homog}$ can be computed and subtracted from $\Gamma_{total}$, leaving $\Delta\Gamma$. We now concentrate on obtaining a description of $\Delta\Gamma$ as a linear function of $\eta(\mathbf{r}')$.

Under the first order Born approximation, $\Delta\Gamma$ is dependent only on first order scattering; thus by integrating across $\Omega$, the volume to be imaged, an equation for $\Omega$ can be written as

$$\Delta\Gamma(\mathbf{s},\mathbf{d},t,t_0) \approx -\int_\Omega \int_{-\infty}^{\infty} [g_{slab}^{(\Gamma)}(\mathbf{s},\mathbf{r}',t',t_0)\eta(\mathbf{r}')g_{slab}^{(\Phi)}(\mathbf{r}',\mathbf{d},t-t',t_0)]dt'd\mathbf{r}'. \tag{2.4}$$

Here, $\triangle\Gamma(\mathbf{s},\mathbf{d},t,t_o)$, the change in the photon fluence measured at location $\mathbf{d}$ at time $t$ due to inhomogeneities in the background absorption for a source at location $\mathbf{s}$, is equal to the integral of all first order scattering throughout the volume. The Green's function $g_{slab}^{(\Gamma)}(\mathbf{r},\mathbf{r}',t,t_0)$ is the spatial gradient of $g_{slab}^{(\Phi)}(\mathbf{r},\mathbf{r}',t,t_0)$ with respect to a unit normal extending out of the solution volume. This gives $g_{slab}^{(\Gamma)}(\mathbf{r},\mathbf{r}',t,t_0)$ the form

$$\begin{aligned} g_{slab}^{(\Gamma)}(\mathbf{r},\mathbf{r}',t,t_0) = {} & \frac{\exp\left\{-\left[\mu_a c(t-t_0) + \frac{d^2}{4\gamma^2(t-t_0)}\right]\right\}}{[4\pi\gamma^2(t-t_0)]^{3/2}} \\ & \times \sum_{n=-\infty}^{\infty}\left[\frac{-2\,(z-2z_dn-z_0)}{4\gamma^2(t-t_0)}\exp\left(\frac{-(z-2z_dn-z_0)^2}{4\gamma^2(t-t_0)}\right)\right. \\ & \left. + \frac{2(z-2z_dn+z_0)}{4\gamma^2(t-t_0)}\exp\left(\frac{-(z-2z_dn+z_0)^2}{4\gamma^2(t-t_0)}\right)\right], \\ & d = \sqrt{x^2+y^2}, \quad \text{where } \mathbf{r}-\mathbf{r}' = (x,y,z), \\ & z_o = [(1-\bar{\rho})\mu_s]^{-1}. \end{aligned} \tag{2.5}$$

This relationship is necessary, as the photon density is not a directly measurable quantity. By taking the gradient of the photon density, we obtain the photon fluence, the intensity of the light exiting from the boundary at the location of the detector. This fluence is a quantity which we are capable of measuring with detectors placed on the surface.

Because the system is causal, (2.2) is zero for $t < t_0$. Additionally, presuming that the timescale can be adjusted such that $t_0 = 0$, the second integral in (2.4) will have support only for $t'$ such that $0 \le t' \le t$, and the dependence upon $t_0$ can be dropped from (2.4). Discretizing (2.4) in piecewise constant fashion for each voxel converts the spatial integration into a summation. Combining these modifications results in

$$\triangle\Gamma(\mathbf{s},\mathbf{d},t) \approx -\sum_{i=1}^{Nvox} dV_i \int_0^t [g_{slab}^{(\Gamma)}(\mathbf{s},\mathbf{r}_i',t')g_{slab}^{(\Phi)}(\mathbf{r}_i',\mathbf{d},t-t')\eta(\mathbf{r}_i')]dt', \tag{2.6}$$

where $dV_i$ is the volume of the $i$th voxel and the $r_i'$'s are locations of voxel centers. For simplicity and maximum computational gain, we assume that $dV_i$ is constant for

all voxels and simply note it as $dV$. The above equation then serves as a basis from which to construct the discrete linear model $\mathbf{Af} \approx \mathbf{g}$, where $\mathbf{f}$ is the vector of unknown absorption values at each of the voxels in the image. This equation is ill-posed in the sense that a least-squares solution to the system would be hopelessly contaminated by noise. Therefore, we solve instead the Tikhonov regularized problem (1.1).

From (2.6), we see that the entry in the matrix $\mathbf{A}$ associated with voxel (column) $i$ and row corresponding to source $\mathbf{s}$ and detector $\mathbf{d}$ at time $t$ is

$$J_i^{(\Gamma)}(\mathbf{s},\mathbf{d},t) \approx -dV \left( \sum_{j=1}^{T_t} w_j \left[ g_{slab}^{(\Gamma)}\left(\mathbf{s},\mathbf{r}_i',t_j'\right) g_{slab}^{(\Phi)}\left(\mathbf{r}_i',\mathbf{d},t-t_j'\right)\right]\right), \tag{2.7}$$

where in this case the approximation notation conveys the fact that the integral was evaluated numerically using the composite trapezoid rule on a regular grid, and the $w_j$ denote the weights of the composite trapezoid rule.

As mentioned in the introduction, it is not feasible to naively construct and store each entry in $\mathbf{A}$. However, it is clear from the above Green's functions that in a slab geometry there is some degree of spatial invariance in the kernels. In the following section, we describe how to utilize the invariance to store only a minimum of information to represent every entry in $\mathbf{A}$, and to utilize the stored information to perform the matrix-vector products necessary to employ an iterative algorithm for solving (1.1).

**3. Exploiting matrix structure.** There is a significant amount of redundancy in the forward matrix. By eliminating the excesses involved in computing the same value multiple times, we can reduce the time required to generate the matrix. We are able to store each computed value only once and reuse it as needed. This reuse takes the guise of a series of selection matrices: extremely sparse, easily formed matrices consisting entirely of ones and zeros.

**3.1. Change of coordinates.** In (2.2), (2.5), and (2.7), the X-Y coordinates of the source-voxel and detector-voxel differences enter into the equation only as radial distances $\sqrt{x^2+y^2}$. Because of this, the absolute X-Y locations involved are irrelevant to the computation, and it is possible to change from the original absolute Cartesian coordinates to a different coordinate system based on these radial distances. Equation (2.7) expresses each matrix component as the convolution of two Green's functions with respect to time. Given the convolution involved in obtaining the matrix components, this new coordinate system can be seen as two joined cylindrical coordinate systems, with the central axis of one lying upon the radial boundary of the other. To see this, let $(X_s, Y_s, Z_s)$ represent the absolute location of the source and $(X_r, Y_r, Z_r)$ and $(X_v, Y_v, Z_v)$ represent the absolute locations for the detector and voxel, respectively. Define the new variables:

$$\begin{aligned} D_1 &= ((X_v - X_s)^2 + (Y_v - Y_s)^2)^{1/2}, \\ Z_1 &= Z_v - Z_s, \\ D_2 &= ((X_v - X_r)^2 + (Y_v - Y_r)^2)^{1/2}, \\ Z_2 &= Z_r - Z_v. \end{aligned} \tag{3.1}$$

These four values are sufficient for computing the value of (2.7), regardless of the absolute position of the three initial sets of $(x, y, z)$ coordinates. Further, because we are modeling a slab geometry, the $z$-coordinates for the sources are fixed and known,

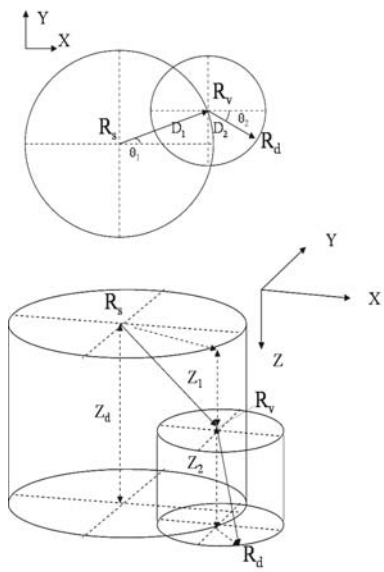



FIG. 3.1. *Dual cylindrical coordinate system. This illustration depicts visually the spatial invariance of* (2.7)*, which expresses the matrix values as the numerical convolution of two Green's functions. Here, the source is located at $R_s$, the center of the top of the larger cylinder; the voxel under consideration is at $R_v$, the top center of the smaller cylinder; and the detector is located at $R_d$, a point on the lower rim of the cylinder. Given this arrangement, the radial location of the small cylinder with respect to the larger, and the radial location of the detector with respect to the small cylinder, can both be changed arbitrarily without affecting the resulting value of* (2.7)*. Additionally, given a slab geometry of fixed thickness $Z_d$, the entire system is shift invariant to changes along the X-Y plane.*

as are the $z$-coordinates for the detectors. Therefore, the only time that $Z_1$ and $Z_2$ change is when $Z_v$ changes, and thus only one of $Z_1, Z_2, Z_v$ is needed in order to compute the other two. As the locations of the source, detector, and voxel are allowed to vary, these changes will be reflected in changes to the triple $(D_1, D_2, Z_v)$. This dual cylindrical coordinate system is shown in Figure 3.1. Note that the two radial angles $\theta_1$ and $\theta_2$ do not appear in (3.1). Clearly, (2.7) is independent of $\theta_1$ and $\theta_2$ and is therefore invariant to changes in the angles. Therefore, given fixed $t$, for each source-voxel/detector-voxel pairing in XYZ space which maps to the same $(D_1, D_2, Z_v)$ triple, the corresponding matrix entry will be the same. Note that this means that all sets of three points with the same voxel height and the same length x-y projections of the source-voxel and voxel-detector distances require identical computations.

Using this new coordinate system, we now return to the original problem, with all of the source and detector positions, and examine those positions within this dual-cylindrical system. In practice, many source-detector configurations fall into one of two categories: fixed array or raster scanned. For the fixed array case, two grids are defined, one for the sources and one for the detectors. For each source location, data are collected at all of the detectors. In a raster scanned system, a source grid is defined, along with a number of detector locations, fixed relative to the source. In both cases, high levels of redundancy in the $(D_1, D_2, Z)$ triplets will be present. This means that a large number of repeated operations are performed if each component of the matrix $\mathbf{A}$ is explicitly computed. Table 1 shows redundancy levels for several common source-detector configurations based on a raster scan and two uniform grids

TABLE 1

*Computational/storage improvements for a variety of source-detector configurations. This table reflects the amount of redundancy present for three examples of common source-detector geometries. The reduction is presented as the ratio of total to unique number of matrix elements. Thus, for the second example given, explicitly computing every matrix element results in each unique computation being done an average of* 690 *times. By eliminating these excess computations and simply computing each unique element once, we are able to significantly reduce the amount of time required to compute the forward matrix.*

| SD configuration | Computational/storage reduction |
|---|---|
| $7 \times 7$ Raster scanned | $49\times$ |
| $5 \times 5$ Sources over $5 \times 5$ Detectors $10 \times 10 \times 10$ Voxels | $690\times$ |
| $10 \times 10$ Sources over $10 \times 10$ Detectors $10 \times 10 \times 10$ Voxels | $3844\times$ |

TABLE 2

*Summary of index notation. Note that* $N_{pts} \leq N_{vox} N_D N_S$.

| Symbol | Meaning |
|---|---|
| $N_z$ | No. grid pts in z-direction |
| $N_{vox}$ | No. of voxels |
| $N_{pts}$ | No. of unique $(D_1, D_2, Z)$ triples |
| $N_S$ | No. of sources |
| $N_D$ | No. of detectors |
| $N_t$ | No. of time pts |
| $N_{comp}$ | No. of interp. nodes in 3-space |

of sources and detectors.

Now we are ready to consider taking advantage of this redundancy to represent the matrix $\mathbf{A}$ (see Table 2 for definitions of the dimension notation). The matrix $\mathbf{A}$ has $N_t N_D N_S$ rows and $N_{vox}$ columns. We order the rows of $\mathbf{A}$ such that the inner loop is over time, then detectors, then sources. It will be convenient to consider the structure of $\mathbf{A}^T$ instead of $\mathbf{A}$. Using $\mathbf{A}_{ij}$ to denote the $N_{vox} \times N_t$ submatrix holding the entries given by (2.7) for the $i$th source and the $j$th detector, the matrix $\mathbf{A}^T$ has the block structure

$$\mathbf{A}^T = [\mathbf{A}_{11} \quad \dots \quad \mathbf{A}_{1N_D} \quad \mathbf{A}_{21} \quad \dots \quad \mathbf{A}_{N_S N_D}]. \tag{3.2}$$

However, given the redundancy noted above, explicit computation of each $A_{ij}$ is unnecessary. Let us assume that there are $N_{pts}$ unique $(D_1, D_2, Z)$ coordinate triplets, given all source-voxel-detector combinations. Clearly, $N_{pts} \leq N_{vox} N_D N_S$. Each of these triplets is encountered again at each time step, for a total of $N_{pts} N_t$ unique evaluations of (2.7). Let $\mathbf{A}_s$ denote the $N_{pts} \times N_t$ matrix containing these values. Thus instead of computing and storing all $N_{vox} N_D N_S N_t$ entries in $\mathbf{A}$, we will only need to evaluate and store the $N_{pts} N_t \leq N_{vox} N_D N_S N_t$ unique entries of $\mathbf{A}_s$.

We can represent $\mathbf{A}$ in terms of $\mathbf{A}_s$ using a series of selection matrices of size $N_{vox} \times N_{pts}$, where each row consists of all zeros, except for a single "1" to select the appropriate row from $\mathbf{A}_s$. Placing these selection matrices into the previous expression for $\mathbf{A}$ results in

$$\mathbf{A}^T = [\mathbf{S}_{11}\mathbf{A}_s \quad \dots \quad \mathbf{S}_{1N_D}\mathbf{A}_s \quad \mathbf{S}_{21}\mathbf{A}_s \quad \dots \quad \mathbf{S}_{N_S N_D}\mathbf{A}_s]. \tag{3.3}$$

It is possible to rewrite (3.3) in a form which exploits the underlying Kronecker

structure of the matrix:

$$\mathbf{A}^T = [\mathbf{S}_{11} \quad \dots \quad \mathbf{S}_{1N_D} \quad \mathbf{S}_{21} \quad \dots \quad \mathbf{S}_{N_S N_D}][\mathbf{I}_{N_S * N_D} \otimes \mathbf{A}_s]. \tag{3.4}$$

We now have a compact representation of the matrix $\mathbf{A}$ that may be used inside the iterative solver to produce the necessary matrix-vector products $\mathbf{A}x$ or $\mathbf{A}^T x$. All entries of the matrix $\mathbf{A}$ need not be explicitly formed.

Furthermore, there is structure present within the selection matrices themselves. If $\mathbf{A}_s$ is arranged such that it has block structure,

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_{N_z} \end{bmatrix}, \tag{3.5}$$

where each block corresponds to the set of $(D_1, D_2, Z)$ triplets sharing a particular $Z$ value, the selection matrices themselves have a Kronecker structure. This is because, assuming a uniform grid for the voxels, the same set of $(D_1, D_2)$ values will be required from each Z-slice. Thus we have

$$\mathbf{S}_{ij} = \mathbf{I}_{N_z} \otimes \tilde{\mathbf{S}}_{ij}, \tag{3.6}$$

where the matrix $\tilde{\mathbf{S}}_{ij}$ is the selection matrix required to extract the appropriate rows from any of the $Z_k$ matrices.

**3.2. Interpolation.** While this change-of-coordinates representation provides a significant reduction in the amount of overhead required to compute and store the matrix $\mathbf{A}$, it does not provide any gains when that matrix is used in matrix-vector multiplications. While operations on $\mathbf{A}$ can be done block by block, the overall size of $\mathbf{A}$ is still exceedingly large. When used in an iterative scheme where multiple matrix-vector products are required for each iteration, the time involved in each product becomes a limiting factor. The desire to accelerate these products, as well as further reduce the required initial computation, motivates the next step in our method.

Recall that even though we have reduced the number of distinct entries that need to be computed to represent $\mathbf{A}$, each of these distinct entries requires the evaluation of the expression (2.7). These are clearly expensive to compute because of the numerous evaluations of the Green's functions and multiple summations. It is this function evaluation whose explicit calculation we hope to minimize. Therefore, we propose to use interpolation to aid in the function evaluation. Not only does this reduce the overall amount of initial computation required to approximate each matrix entry, but as we will illustrate shortly, it has the added benefit of speeding up matrix-vector products.

We utilized interpolation expressible in the linear form

$$\mathbf{A}_s \approx \mathbf{Q}\mathbf{V}. \tag{3.7}$$

Here, $\mathbf{Q}$ is of size $N_{pts} \times N_{comp}$ and is the interpolation matrix, while $\mathbf{V}$ is $N_{comp} \times N_t$, consisting of the smaller set of values which must be explicitly computed. $N_{comp}$ is the number of nodes to be computed for use in the interpolation scheme and is chosen to be significantly smaller than $N_{pts}$. While all further results, including computational complexities, will be shown with respect to the specific linear interpolation scheme

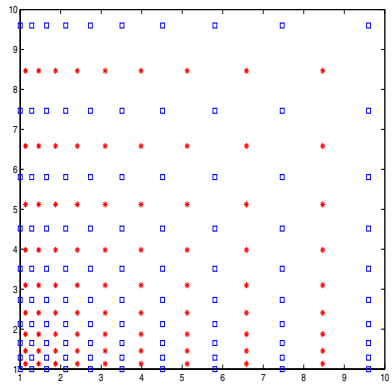


FIG. 3.2. *Graphical description of the sampling technique used with the interpolation. The sets of red and blue points (asterisks and squares, respectively) show the two grids which were used to sample $D_1 - D_2$ slices of the sample space. For each Z-value, one of the two grids was selected, alternating as the Z-value was stepped from one value to the next.*

we chose to use, this process could be used with any interpolation method that can be expressed in matrix format.

Evaluation of the function in (2.7) will occur at a set of $N_{comp}$ interpolation nodes in $(D_1, D_2, Z)$ space at each of the $N_t$ values $\{t_1, \ldots, t_{N_t}\}$. The linear interpolation we use is based on a Delaunay tessellation of the $(D_1, D_2, Z)$ space. This tessellation uses the interpolation nodes as vertices of a tetrahedral mesh. To determine the value at each point on the more dense grid to which we interpolate, we first determine inside which tetrahedron the point lies. Barycentric coordinates of the desired point are then computed with respect to the vertices of the enclosing tetrahedron, and those coordinates are used as the weights of the interpolation. For example, if the coordinates of the four interpolation nodes comprising the encircling tetrahedron are given by $(D_1^{(i)}, D_2^{(i)}, Z^{(i)})$, $i = [1, \ldots, 4]$, with corresponding function values given by $v_i$, $i = [1, \ldots, 4]$, then the barycentric coordinates of the desired point $(D_1, D_2, Z)$ can be determined as

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} D_1^{(1)} & D_1^{(2)} & D_1^{(3)} & D_1^{(4)} \\ D_2^{(1)} & D_2^{(2)} & D_2^{(3)} & D_2^{(4)} \\ Z^{(1)} & Z^{(2)} & Z^{(3)} & Z^{(4)} \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} D_1 \\ D_2 \\ Z \\ 1 \end{bmatrix}. \tag{3.8}$$

These barycentric coordinates become the weights of the interpolation, to give an approximate value $v$ at the point $(D_1, D_2, Z)$ of

$$v = a_1 v_1 + a_2 v_2 + a_3 v_3 + a_4 v_4. \tag{3.9}$$

Arranging the weights in $\mathbf{Q}$ and the coarse grid values in $\mathbf{V}$, the $N_{pts} \times N_{comp}$ matrix $\mathbf{Q}$ will be sparse with only four nonzero entries per row, while $\mathbf{V}$ will be dense. It remains to decide how to choose the position of the interpolation nodes.

Looking again at (2.2), there is an exponential relationship between the calculated value and the values of $D_1$ and $D_2$. Because of this, a linear sampling method is unlikely to give acceptable results when combined with a linear interpolation method. This is especially true when $D_1$ and $D_2$ are close to zero, where the value of (2.2)

TABLE 3

*Number of floating point operations for various matrix-vector products. This assumes that multiplication by 1 can be done at no cost. The vector $w$ is the component of $v$ corresponding to the data points from a single source-detector pair. Note that the use of $A_s$ alone does not provide computational gains when computing matrix-vector products. It is the combination of the selection and interpolation matrices (the $S_{ij}Q$ product) and the initial $Vw$ multiplication that provide the reduction in total required computation.*

| Product | Flops (in big-Oh) |
|---|---|
| $A^T v$, $A$ dense | $O(N_D N_S N_t N_{vox})$ |
| $A_s w$ | $O(N_{pts} \times N_t)$ |
| $A^T v = S(I_{N_D N_S} \otimes A_s)v$ | $O(N_D N_S N_t N_{vox})$ |
| $(S_{ij}Q)(Vw)$ | $O(4N_{vox} + N_{comp}N_t)$ |
| Approx. $A$ using $A_s w \approx QVw$ | $O(N_D N_S(4N_{vox} + N_{comp}N_t))$ |

is changing very rapidly. Initial experimentation confirmed that this held true in practice. As an alternative sampling method, a grid with exponential spacing in $D_1$ and $D_2$ was developed, as illustrated in Figure 3.2. By clustering a larger number of sample points near the origin, it was possible to achieve significantly better interpolation results for the same values of $N_{pts}$, with acceptable error levels in terms of the quality of the resulting reconstructions. To efficiently sample the space, two grids, red and blue, were established, with the nodes of one grid centered between the nodes of the other. We alternated between the two grids as we stepped down the Z-axis, as indicated by the red and blue grids in Figure 3.2. In order to ensure that the grid covered all of the necessary space, the final slice always used the blue grid, even if this meant that two adjacent slices utilized the same grid.

**4. Computational complexity.** To this point, we have been primarily concerned with the up-front costs in terms of storage and computation time for representing the matrix. We decreased the storage for the matrix by exploiting redundancy. With the interpolation approximation, we also reduced the initial time needed to represent entries in the (approximate) matrix, and the storage requirement for $\mathbf{A}_s$ has decreased from $N_{pts}N_t$ to $4N_{pts} + N_{comp}N_t$ double values stored in memory. Further, the addition of the interpolation step adds the potential to decrease the overall computational complexity of executing each matrix-vector product involved in solving the minimization problem. Because the computations in the matrix-vector product that correspond to individual source-detector pairs are independent, we will simply examine the computation required for a single source-detector pair.

As a benchmark, the number of floating point operations (i.e., multiplications and additions) required for matrix-vector products using the dense formulation and also the formulation in section 3.1 are given in Table 3. Note that the flop count is not reduced over the dense formulation when using the sparse representation in section 3.1. (Note: multiplication by the selection matrix requires no flops.)

The situation changes when utilizing interpolation. At each source-detector pair, the equation to be evaluated is

$$\mathbf{S}_{ij}\mathbf{QVx}. \tag{4.1}$$

Here, the first step is to combine the selection and interpolation matrices. Again, the product of these two matrices can be computed at no cost. The reason for taking this "product" first is that in general $N_{vox} < N_{pts}$, and thus the resulting sparse matrix will have only $4N_{vox}$ nonzeros as opposed to the $4N_{pts}$ nonzeros that are in $Q$.

Therefore, we compute $S_{ij}QVw$ in three steps:

TABLE 4

*The various matrices used in our approximate forward model, their sizes, and their number of nonzero values. $A_s$ is the small version of the forward matrix containing only the unique matrix elements. $S_{ij}$ are the selection matrices used to retrieve the blocks of the full matrix from $A_s$. $A$ is the full weight matrix. $Q$ is the matrix responsible for implementing the linear interpolation, and $V$ is a small dense matrix of unique values such that $A_s \simeq QV$. For descriptions of other variables used, see Table 2.*

| Matrix | Dimensions | Nonzeros |
|---|---|---|
| $A_s$ | $N_{pts} \times N_t$ | $N_{pts}N_t$ |
| $S_{ij} = I_{N_z} \otimes \tilde{S}_{ij}$ | $N_{vox} \times N_{pts}$ | $N_{vox}$ |
| $A$ | $N_D N_S N_t \times N_{vox}$ | (Sparse rep.) $N_D N_S N_{vox} + N_{pts}N_t$ |
| $Q$ | $N_{pts} \times N_{comp}$ | $4N_{pts}$ |
| $V$ | $N_{comp} \times N_t$ | $N_{comp}N_t$ |
| Approx. $A$ | | $N_D N_S N_{vox} + 4N_{pts} + N_{comp}N_t$ |

- Form the product $W_{ij} = S_{ij}Q$.
- Compute the matrix-vector product $Vw$.
- Compute the matrix-vector product $W_{ij}(Vw)$.

The first step is "free." The second step requires $O(N_{comp}N_t)$ flops. Since $W_{ij}$ is $N_{vox} \times N_{comp}$ but has only four nonzero entries per row, the product $W_{ij}(Vw)$ requires an additional $O(4N_{vox})$ flops. Thus, the cost of the product $S_{ij}QVw$ is $O(4N_{vox} + N_{comp}N_t)$ flops. There is one such product for every source-detector pair, and therefore products with the approximation to $A$ formed by using $A_s \approx QV$ cost $O(N_D N_S(4N_{vox} + N_{comp}N_t))$ flops. Comparing this to the total number of flops required for the dense formulation, we can see that the reduction in required computation is dependent upon the number of interpolation nodes $N_{comp}$ and the total number of voxels $N_{vox}$. The computational costs and storage requirements for the various steps are detailed in Tables 3 and 4.

**5. Simulation results.** For the regularization matrix $\mathbf{R}$, a first order approximation to the gradient was utilized, generated as

$$\mathbf{R} = \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}, \tag{5.1}$$

where each of the three submatrices are discrete approximations to the first order derivative along the associated axis.

Rather than simply run an iterative algorithm such as LSQR [23] with an augmented matrix once for each trial value of $\lambda$ in (1.1), we used the algorithm in [13] so that the results could be run simultaneously on an array of $\lambda$'s. The number of iterations was fixed at fifty, chosen by first running the algorithm without any regularization and performing an L-curve analysis [10] across the iterations. It is reasonable to presume that a given regularized system should have sufficiently converged by several iterations past the corner of the L-curve for the unregularized problem. For the regularized problem, selection of the appropriate regularization parameter was done through the use of an L-curve analysis at the final iteration.

After some experimentation, it was found that for our first data set, where the ground truth was known, the minimum error solution was consistently at a point which would be considered underregularized according to the L-curve. Visual analysis of a second data set, provided by ART, suggested that a similar situation existed with that data. As such, the results shown for the known phantom are those at the

TABLE 5

*Interpolation levels, with corresponding number of computed nodes and the resulting error induced in the solution for data set* 1. *Error is computed as* $\frac{\|Computed-Actual\|_2}{\|Actual\|_2}$. *The interpolation level noted in column* 1 *denotes the initial number of interpolation nodes along each dimension of the space. To eliminate unnecessary evaluations of* (2.7), *only those nodes needed to approximate* $A_s$ *were computed.*

| Interpolation level | Number of computed points | Induced error |
|---|---|---|
| None | 11760 | 0.7336 |
| (40,40,40) | 7188 | 0.7344 |
| (30,30,30) | 4659 | 0.7352 |
| (20,20,20) | 2109 | 0.7362 |
| (15,15,15) | 1098 | 0.7409 |
| (10,10,15) | 578 | 0.7294 |
| (10,10,10) | 402 | 0.7404 |

minimum error point, while those shown for the second data set were chosen to be "underregularized" by a similar order of magnitude.

Of course, critical to determining the utility of the interpolation is evaluation of the error introduced into the reconstructions. Interpolation levels are denoted in what follows as a triplet (a,b,c), where the values define the number of grid points along each of the $(D_1, D_2, Z)$ axes, respectively. However, the rectangular grid computed using the values will contain some nodes which will not be needed during the interpolation step. Rather than compute their values and not use them, we simply eliminate these nodes. The numbers in the second column of Table 5 give the number of nodes remaining after this elimination.

We show results for two simulated data sets, each using a number of different interpolation levels. For our first data set, where ground truth was known, we present reconstruction images for two interpolation levels as well as the fully computed result. Additionally, we report error levels for a further four interpolation levels. For our second data set we present three reconstructions, two using interpolation and one without, and give an analysis of relative error levels.

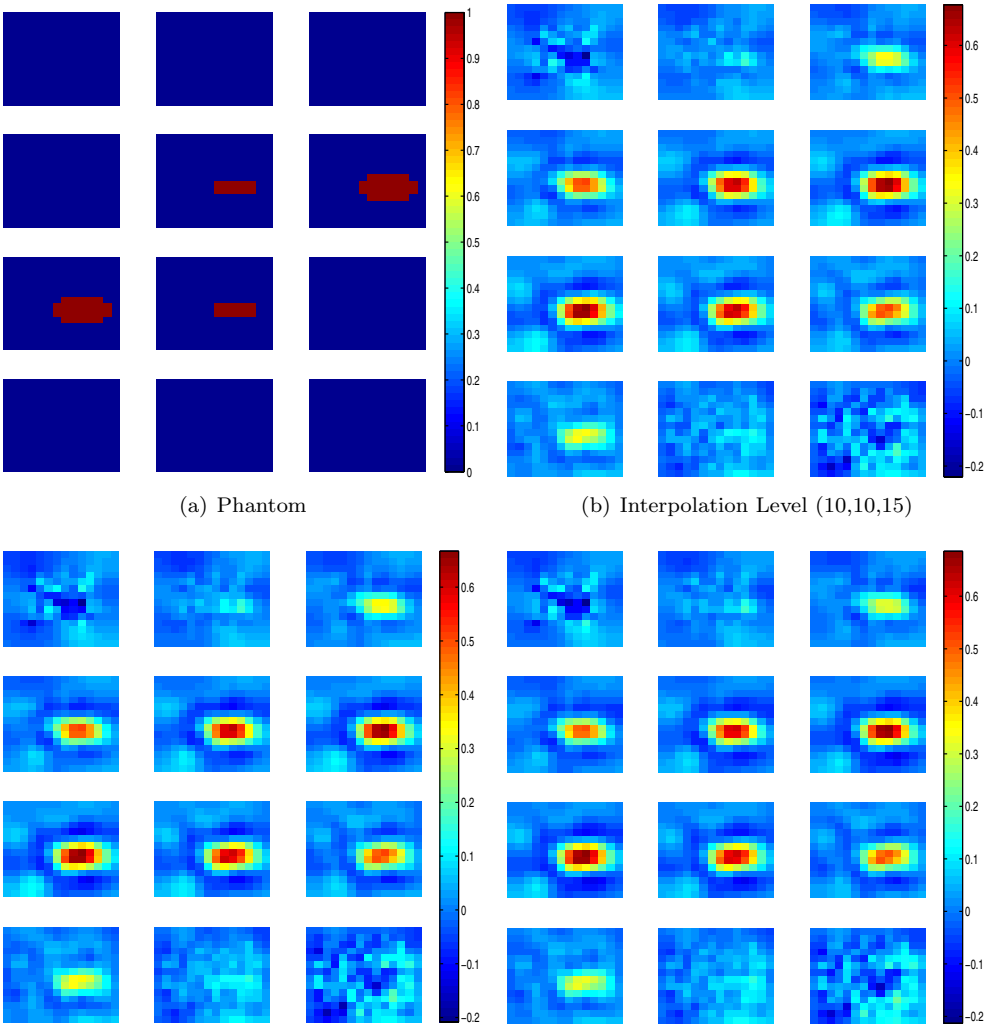
For each data set, relative error levels are computed with respect to the other reconstructions and with respect to ground truth in the case it is known. All error levels are computed as

$$E_{\mathbf{f}_b}(\mathbf{f}_a) = \frac{\|\mathbf{f}_b - \mathbf{f}_a\|_2}{\|\mathbf{f}_b\|_2}, \tag{5.2}$$

where $E_{\mathbf{f}_b}(\mathbf{f}_a)$ is the error in a reconstruction $\mathbf{f}_a$ with respect to a reconstruction $\mathbf{f}_b$, using the standard 2-norm.

**5.1. Data set 1.** In order to determine how the interpolation error affects the resulting solutions, we first used a simulated data set generated using a known phantom. The data was generated using the image in Figure 5.1(a) as $\mathbf{f}_{real}$ to get $\mathbf{g} = \mathbf{A}\mathbf{f}_{real}$, with random noise added by the Matlab awgn() function at a signal-to-noise ratio of 10dB. Inversions were then run using six different interpolation levels with a number of nodes ranging from 61% $N_{pts}$ down to 3.4% $N_{pts}$. The specific interpolation levels, and number of points computed, are shown in Table 5, along with the relative error of each solution with respect to ground truth.

Images of reconstructions obtained using the (10,10,15) and (15,15,15) interpolation levels, as well as the fully computed matrix, can be seen in Figures 5.1(b)–(d).




(a) Phantom (b) Interpolation Level (10,10,15)

(c) Interpolation Level (15,15,15) (d) Fully Computed

FIG. 5.1. *Results for data set* 1. *A known phantom was used to generate simulated data with* 10*dB white Gaussian noise. Inverse results were obtained for a number of different interpolation levels.* (a) *Phantom used to generate simulated data.* (b) *Result with (*10,10,15*) interpolation level (*578 *explicitly computed points). Absolute error of* 0.7295*, error relative to* (d) *of* 0.1128. (c) *Result with (*15,15,15*) interpolation level (*1098 *computed points). Absolute error of* 0.7406*, error relative to* (d) *of* 0.1035. (d) *Result with fully computed matrix (*11760 *computed points). Absolute error of* 0.7336. *Note that all three constructions are visually almost identical, and that while the relative error between the fully computed and interpolated reconstructions is greater than* 10*, the absolute error changes very little.*

Visual comparison of the three images reveals little if any difference. Analytic comparison results in relative error in the (10,10,15) solution with respect to the fully computed solution of 0.1128, while the (15,15,15) solution exhibits a relative error of 0.1035. Examining the errors with respect to ground truth, however, suggests that despite their differences, the solutions using interpolation are of nearly the same quality as the fully computed solution. Decreasing the number of points computed from

11760 (the full number) to 1098 (the (15,15,15) interpolation level) results in the relative error with respect to ground truth changing only 0.0070 from 0.7336 to 0.7406. Interestingly, the (10,10,15) case, with only 578 points computed, results in a relative error of 0.7295, which is actually a lower error than for the fully computed case. This suggests that, given the ill-posed nature of the problem and the regularization occurring in the inversion, the error induced by the interpolation has little consistent effect upon the absolute error in the solution.

**5.2. Data set 2.** The second data set was generated by ART using an undisclosed forward solver (background $\mu_a$ and $\mu_s$ were provided) and a proprietary noise model. Reconstruction images are shown in Figure 5.2 for the fully computed matrix, as well as (10,10,15) and (15,15,15) interpolation levels.

The results for this data set are similar to those of data set 1. Visually, the three inversion results are nearly identical. Because the ground truth is not known in this case, absolute error values cannot be computed. However, comparing the two interpolated solutions to the fully computed one yields relative differences with respect to the fully computed solution of 0.0947 and 0.0857 for the (15,15,15) and (10,10,15) solutions, respectively. These numbers are similar to those seen in the case of the known phantom, where it was shown that absolute error levels were only slightly perturbed by the use of the approximated $\mathbf{A}_s$.

**6. Conclusions and future work.** We have presented a method by which the forward matrix $\mathbf{A}$ associated with a certain linearized diffuse optical tomography problem can be efficiently computed and then effectively approximated. Our first step utilizes a change of variables to enable us to represent $\mathbf{A}$ as a core data matrix $\mathbf{A}_s$ and a number of selection matrices $\mathbf{S}_{ij}$. While $\mathbf{A}_s$ is small and dense, the selection matrices are extremely sparse, enabling the entire representation of $\mathbf{A}$ to be stored in significantly less memory. This decomposition also allows for matrix-vector operations upon $\mathbf{A}$ to be performed in a sequential manner, drastically reducing the amount of memory required to perform such operations.

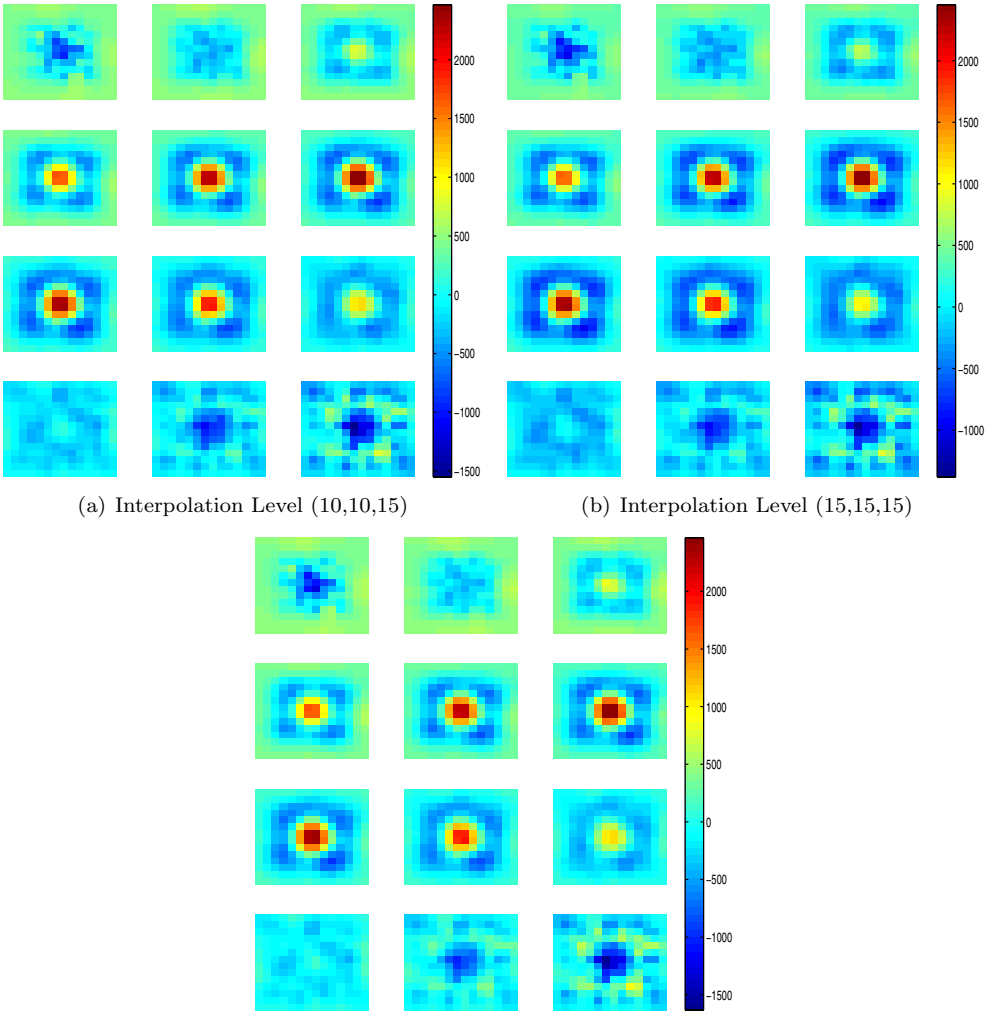
Our simulated results indicate that the use of interpolation to approximate $\mathbf{A}_s$ and thus $\mathbf{A}$ gives accurate solutions. While the solutions using interpolation result in relative errors with respect to the fully computed solution on the order of 0.10, our results indicate that these differences do not significantly affect the error with respect to ground truth.

A further test of this work would be the application of our method to experimentally collected data. Given that the mathematical models presented here are inherently an approximation of reality, there will be an increased mismatch between the data and the model. As such, the model errors introduced by our interpolation scheme should have even less of an effect than they did in the results presented here.

This method also has applications beyond diffuse optical tomography. Any system with a similar invariance to radial angle could potentially be rewritten so as to use our referencing scheme. This includes problems such as continuous wave diffusion imaging, heat transfer in solids, and other problems using omnidirectional sources.

One specific area of application is fluorescence-based optical imaging. Linear models are capable of accurately modeling such systems and have led to systems currently in use for basic in vivo research [21, 20]. Because these systems are optically based using lasers at similar wavelengths, the mathematical details of the diffusion approximation and Green's functions detailed here carry over almost unchanged.

The interpolation could also be applied to other systems, especially those which are linearizations of nonlinear systems. Presuming a reasonably smooth kernel along




(a) Interpolation Level (10,10,15)

(b) Interpolation Level (15,15,15)

(c) Fully Computed

FIG. 5.2. *Results for data set* 2. *Data set provided by ART with known background optical parameters, but unknown forward and noise models. The images compare results using different interpolation levels. Note that absolute values for the reconstructions are significantly higher, owing to a lack of information regarding source intensities. Thus the true quantitative values are all multiplied by an unknown scaling factor.* (a) *Result with (*10,10,15*) interpolation level (*578 *explicitly computed points). Error relative to* (c) *of* 0.0857. (b) *Result with (*15,15,15*) interpolation level (*1098 *computed points). Error relative to* (c) *of* 0.0947. (c) *Result with fully computed matrix (*11760 *computed points). Again, all three reconstructions are visually identical and exhibit similar degrees of relative error. The change in absolute error is likely similar to that seen with data set* 1.

some dimension, it is feasible that linear approximations of that kernel would result in similarly small changes to the resulting solutions. When computation of individual kernel values is prohibitively expensive, this could lead to significant reductions in required computation. This interpolation could also be investigated to determine its regularizing properties. While our results suggest empirically that any regularizing effect is minimal, a study on the regularizing effects of interpolation-smoothed kernels

may be of value.

Finally, further work could seek to extend this type of optimization to the case of systems with structured inhomogeneities. Layered media offer a straightforward extension of our method, while media with more complex structure would require correspondingly more effort. Both would enable our method to be used in a wider range of systems and configurations.

## REFERENCES


[1] S. R. Arridge, *The theoretical basis for the determination of optical pathlengths in tissue*, Phys. Med. Bio., 37 (1992), pp. 1531–1560.

[2] S. R. Arridge, J. C. Hebden, M. Schweiger, F. E. W. Schmidt, M. E. Fry, E. M. C. Hillman, H. Dehghani, and D. T. Delpy, *A method for three-dimensional time-resolved optical tomography*, Int. J. Imaging Systems Technol., 11 (2000), pp. 2–11.

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

[4] D. A. Boas and A. M. Dale, *Simulation study of magnetic resonance imaging-guided cortically constrained diffuse optical tomography of human brain function*, Appl. Optics, 44 (2005), pp. 1957–1968.

[5] S. Chandrasekhar, *Radiative Transfer*, Dover, New York, 1960.

[6] A. Corlu, R. Choe, T. Durduran, K. Lee, M. Schweiger, S. R. Arridge, E. M. C. Hillman, and A. G. Yodh, *Diffuse optical tomography with spectral constraints and wavelength optimization*, Appl. Optics, 44 (2005), pp. 2082–2093.

[7] A. P. Gibson, T. Austion, N. L. Everdell, M. Schweiger, S. R. Arridge, J. G. Meek, J. S. Wyatt, D. T. Delpy, and J. C. Hebden, *Three-dimensional whole-head optical tomography of passive motor evoked responses in the neonate*, Neuroimage, 30 (2006), pp. 521–528.

[8] A. P. Gibson, J. C. Hebden, and S. R. Arridge, *Recent advances in diffuse optical imaging*, Phys. Med. Bio., 50 (2005), pp. R1–R43.

[9] H. L. Graber, Y. Pei, and R. L. Barbour, *Imaging of spatiotemporal coincident states by dc optical tomography*, IEEE Trans. Med. Imag., 21 (2002), pp. 852–866.

[10] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM Monogr. Math. Model. Comput. 4, SIAM, Philadelphia, PA, 1997.

[11] R. C. Haskell, L. O. Svaasand, T.-T. Tsay, T.-C. Feng, M. S. McAdams, and B. J. Tromberg, *Boundary conditions for the diffusion equation in radiative transfer*, J. Opt. Soc. Amer. A, 11 (1994), pp. 2727–2741.

[12] J. C. Hebden and S. R. Arridge, *Imaging through scattering media by the use of an analytical model of perturbation amplitudes in the time domain*, Appl. Optics, 35 (1996), pp. 6788–6796.

[13] M. E. Kilmer, P. C. Hansen, and M. I. Español, *A projection-based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., 29 (2007), pp. 315–330.

[14] A. Li, G. Boverman, Y. Zhang, D. Brooks, E. L. Miller, M. E. Kilmer, Q. Zhang, E. M. C. Hillman, and D. A. Boas, *Optimal linear inverse solution with multiple priors in diffuse optical tomography*, Appl. Optics, 44 (2005), pp. 1948–1956.

[15] C. Lindquist, A. Pifferi, R. Berg, S. Andersson-Engels, and S. Svanberg, *Reconstruction of diffuse photon-density wave interference in turbid media from time-resolved transmittance measurements*, Appl. Phys. Lett., 69 (1996), pp. 1674–1676.

[16] V. A. Markel, V. Mital, and J. C. Schotland, *Inverse problem in optical diffusion tomography.* III. *Inversion formulas and singular value decomposition*, J. Opt. Soc. Amer. A, 20 (2003), pp. 890–902.

[17] V. A. Markel and J. C. Schotland, *Scanning paraxial optical tomography*, Optics Lett., 27 (2002), pp. 1123–1125.

[18] V. A. Markel and J. C. Schotland, *Symmetries, inversion formulas, and image reconstruction for optical tomography*, Phys. Rev. E (3), 70 (2004).

[19] V. A. Markel and J. C. Schotland, *Multiple projection optical diffusion tomography with plane wave illumination*, Phys. Med. Bio., 50 (2005), pp. 2351–2364.

[20] V. Ntziachristos, E. A. Schellenberger, J. Ripoll, D. Yessayan, E. Graves, Jr., A. Bogdanov, L. Josephson, and R. Weissleder, *Visualization of antitumor treatment by means of fluorescence molecular tomography with an annexin v-cy*5.5 *conjugate*, Proc.

Natl. Acad. Sci. USA, 101 (2004), pp. 12294–12299.

[21] V. Ntziachristos and R. Weissleder, *Experimental three-dimensional fluorescence reconstruction of diffuse media by use of the normalized Born approximation*, Optics Lett., 26 (2001), pp. 893–895.

[22] M. A. O'Leary, *Imaging with Diffuse Photon Density Waves*, Ph.D. thesis, Physics Department, University of Pennsylvania, Philadelphia, PA, 1996.

[23] C. Paige and M. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[24] M. S. Patterson, B. Chance, and B. C. Wilson, *Time resolved reflectance and transmittance for the noninvasive measurement of tissue optical properties*, Appl. Optics, 28 (1989), p. 2331 ff.

[25] C. H. Schmitz, H. L. Graber, Y. Pei, M. Farber, M. Stewart, R. D. Levina, M. B. Levin, Y. Xu, and R. L. Barbour, *Dynamic studies of small animals with a four-color diffuse optical tomography imager*, Rev. Sci. Inst., 76 (2005).

[26] J. C. Schotland, *Continuous-wave diffusion imaging*, J. Opt. Soc. Amer. A, 14 (1997), pp. 275–279.

[27] M. Schweiger, S. R. Arridge, and I. Nissila, *Gauss–Newton method for image reconstruction in diffuse optical tomography*, Phys. Med. Bio., 50 (2005), pp. 2365–2386.

[28] A. Soubret, J. Ripoll, and V. Ntziachristos, *Accuracy of fluorescent tomography in the presence of heterogeneities: Study of the normalized born ratio*, IEEE Trans. Med. Imag., 24 (2005), pp. 1377–1386.

[29] S. Srinivasan, B. W. Pogue, S. Jiang, H. Dehghani, C. Kogel, W. A. Wells, S. P. Poplack, and K. D. Paulsen, *Near-infrared characterization of breast tumors in vivo using spectrally-constrained reconstruction*, Tech. Cancer Res. Treatment, 4 (2005), pp. 513–526.

[30] 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*, Optics Lett., 30 (2005), pp. 3338–3340.