PAPER

Fast linear inversion for highly overdetermined inverse scattering problems

IOP Publishing Inverse Problems

Inverse Problems **35** (2019) 124002 (22pp) https://doi.org/10.1088/1361-6420/ab44e7

Fast linear inversion for highly overdetermined inverse scattering problems

Vadim A Markel[1], **Howard Levinson**[2] **and John C Schotland**[3]

[1] Department of Radiology, University of Pennsylvania, Philadelphia, PA 19104, United States of America
[2] Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, United States of America
[3] Department of Mathematics and Department of Physics, University of Michigan, Ann Arbor, MI 48109, United States of America

E-mail: vmarkel@pennmedicine.upenn.edu, levh@umich.edu and schotland@umich.edu

Received 27 March 2019, revised 16 August 2019
Accepted for publication 16 September 2019
Published 19 November 2019

Abstract

In this paper we present a fast numerical method for solving large-scale inverse scattering problems. The computational efficiency of the proposed method stems from the utilization of the special structure of the linear forward scattering operator, and does not require or assume any symmetries of the measurement geometry. The described approach is especially useful for inverse problems involving large data sets. As an illustration, we have performed direct numerical inversions for the problem of diffuse optical tomography in measurement geometries with up to $\sim 10^8$ independent data points and $\sim 7 \cdot 10^5$ unknowns.

Keywords: pseudo-inverse, inverse scattering, singular-value decomposition, large data sets

(Some figures may appear in colour only in the online journal)

1. Introduction

The computation of the singular-value decomposition (SVD) or the pseudo-inverse of a matrix is a well-researched subject. The common approach to solving the problem numerically is based on the Golub–Reinsch (GR) algorithm [1, 2], which consists of two main steps. In the first step, the matrix is transformed to a bidiagonal form by a series of unitary Householder

transformations, say $B = PAQ^*$, where A is an $M \times N$ complex matrix whose SVD we wish to compute, B is an upper bidiagonal $N \times N$ matrix, P and Q are products of unitary Householder matrices, and it is usually assumed that $M \geqslant N$. In the second step, the SVD of B is computed by an iterative method. Then, if the SVD of B is of the form $B = U\Sigma V^*$, the SVD of A is $A = (P^*U)\Sigma(V^*Q)$.

The computational complexity of various modifications of the GR algorithm has been reviewed in [3]. In the case of highly overdetermined problems with $M \gg N$, which are of primary interest in this paper, the dominant cost is that of the first step, whose computational complexity is $O(MN^2)$, whereas the cost of the second step is negligible. Indeed, the first J singular values and singular vectors of a bidiagonal matrix can be computed using $O(JN)$ floating-point operations [4–6]. Alternatively, one may not need the full SVD of A and seek instead the least-squares solution of a set of M linear equations in N unknowns, $Ax = b$. This can be achieved by converting the above equation to the form $By = c$ where $y = Qx$ and $c = Pb$. Since B is a bidiagonal $N \times N$ matrix, one can find the least squares solution to the above in $O(JN)$ operations, where J is the number of iterations, for instance, of a conjugate gradient descent algorithm.

All this is well known and used in most contemporary linear algebra computational libraries such as LAPACK. However, the direct application of this approach to the problems in inverse scattering is often impossible. The reason is that the computational complexity of the first step in the GR algorithm, $O(MN^2)$, can easily become prohibitive. In some applications, M is so large that it is not even possible to store A in memory. For example, in some experimental realizations of optical tomography [7–9], $M \gtrsim 10^8$ and $N \gtrsim 10^4$, which translates into $\gtrsim 10^{12}$ matrix elements. Efficient manipulation of matrices of this size requires many terabytes of memory.

However, linearized inverse scattering problems often involve matrices of a special form. This fact can be exploited to avoid the high computational complexity of the first step in the GR algorithm. In fact, the algebraic problem can be formulated so that the large number M never enters the estimates of computational complexity. This observation is very simple, and we have briefly remarked upon it in appendix B of [10]. We also draw the reader's attention to closely related earlier work of Lev-Ari [11]. This paper builds upon and further develops the theoretical insights of [11]. In particular, we provide the motivation and full details of a fast inversion method, introduce a new algorithm based on preconditioning by diagonal scaling, describe two specific computational algorithms, and give several numerical examples which illustrate the power of the approach.

The rest of this paper is organized as follows. In section 2 we give the algebraic formulation of the inverse problem and derive a simple but very efficient algorithm for its solution. In section 3 we describe a more intricate algorithm, which involves preconditioning by diagonal scaling (it is this version that was described in [10]). In section 4 we define the basic steps of two computational algorithms and analyze their computational complexity. In section 5, several numerical examples with ideal data are given. In section 6, we show examples with noisy data. Here we illustrate how the precondition approach can be used to achieve an efficient data reduction in the presence of noise.

Finally, section 7 contains a brief discussion of our results.

2. Algebraic formulation

We start with the algebraic formulation of the linearized inverse scattering problem. As is well-known, the scattering potential of interest $V(\mathbf{r})$ is related to measurements $\Phi(\mathbf{r}_d, \mathbf{r}_s)$ by the integral equation

$$\int_{\Omega} G_0(\mathbf{r}_d, \mathbf{r}) V(\mathbf{r}) G_0(\mathbf{r}, \mathbf{r}_s) \mathrm{d}^3 r = \Phi(\mathbf{r}_d, \mathbf{r}_s). \tag{1}$$

Here $\mathbf{r}_d$ and $\mathbf{r}_s$ are the positions of the detector and the source, which are located outside of the spatial region Ω in which $V(\mathbf{r})$ is nonvanishing, and $G_0(\mathbf{r}, \mathbf{r}')$ is the unperturbed Green's function for the underlying differential equation. The specific form of $G_0(\mathbf{r}, \mathbf{r}')$ is not important for us now, although some specific examples are given below. What is important is that (1) can be discretized and written in the form

$$\sum_{n=1}^{N_v} A_{in} x_n B_{nj} = \Phi_{ij}, \quad 1 \leqslant i \leqslant N_d, \quad 1 \leqslant j \leqslant N_s. \tag{2}$$

Here $A_{in} = G_0(\mathbf{r}_{di}, \mathbf{r}_n)$, $B_{nj} = G_0(\mathbf{r}_n, \mathbf{r}_{sj})$, $x_n = V(\mathbf{r}_n)$, $\mathbf{r}_{di}$ is the location of the ith detector and $\mathbf{r}_{sj}$ is the location of the jth source, $\mathbf{r}_n$ is the center of nth voxel, and N_d, N_s and N_v are the numbers of detectors, sources and voxels, respectively. There are many different discretization schemes utilizing point sources and detectors, or phased arrays, or incident and outgoing plane waves, etc, but all of them result in algebraic equations of the form (2).

Matrix equations of the form $AXB = C$, where X is unknown, have been extensively studied in the literature [11–24]. The problem in which X is restricted to be diagonal is encountered less frequently [11, 13, 21]. A key insight upon which the forthcoming discussion is based was made in 2005 by Lev-Ari [11]. Namely it was noted that the computationally-costly operation of vectorization is not required to solve the equation (in the minimum L^2-norm sense) and an alternative approach was suggested. However, numerical methods for inverse scattering largely rely on vectorization. In this traditional approach, (2) is re-written in the form

$$Kx = b, \tag{3}$$

where K is an $M \times N$ matrix with $M = N_d N_s$ and $N = N_v$, x is the vector of unknowns of length N and b is the vector of data of length M. The components of K and b are given by

$$K_{(ij),n} = A_{in} B_{nj}, \quad b_{(ij)} = \Phi_{ij}. \tag{4}$$

Here (ij) is a composite index; we can replace it with a single index using the operation of matrix unrolling, e.g. by writing

$$m = (i-1)N_s + j, \quad 1 \leqslant i \leqslant N_d, \quad 1 \leqslant j \leqslant N_s.$$

It can be seen that m takes values from 1 to $M = N_d N_s$ and there exists a one-to-one correspondence between (ij) and m. We note that K can be viewed as the Khatri–Rao product [25] of A and B^*. From this point on, the traditional approach treats K as a generic $M \times N$ matrix. In particular, equation (2) with a generic K has been used in diffuse optical tomography [26] and in diffraction tomography [27].

In this paper we make the simple observation that K is not generic, but is given by (4). The number of degrees of freedom in K is therefore $(N_d + N_s)N_v$, which is much smaller than the number of elements of K, that is, $NM = N_d N_s N_v$. We can use this observation to derive inversion methods that are much more computationally efficient than seeking the pseudoinverse of K.

As a first elementary example, assume that we have decided to compute K^*K instead of using the GR algorithm. Generically, the computational complexity of computing the matrix-matrix product K^*K is $O(MN^2)$, the same as the complexity of the first step in the GR algorithm. However, if we use the specific representation (4), we immediately notice that

$$(K^*K)_{nm} = (A^*A)_{nm}(BB^*)_{mn}. \tag{5}$$

The complexity of computing A^*A and BB^* is obviously much smaller than $O(MN^2)$. Assuming $N_s \sim N_d < N_v$, the number of required operations to evaluate the right-hand side of (5) is smaller by the factor $\sim 2/N_v$, and if $N_s \sim N_d > N_v$ then it is smaller by the factor $\sim 2/N_s$. There is also no need to store the large matrix K in memory, which in many cases is not feasible.

We also note that

$$c_n \equiv (K^*b)_n = (A^*\Phi B^*)_{nn}. \tag{6}$$

Computing N elements of the vector c takes $O(MN)$ operations, which is again much smaller than $O(MN^2)$. Therefore, one can compute K^*K according to (5), compute c according to (6) and then solve the regularized equation

$$(K^*K + \lambda^2 I)x = c, \quad c = K^*b, \tag{7}$$

I is the $N \times N$ identity matrix, and λ is the regularization parameter. Since the matrix in (7) is symmetric and positive-definite, x can be found iteratively by the conjugate gradient descent method in $O(JN^2)$ operations, where $J \leqslant N$ is the number of iterations required for convergence. A related observation was also made in [11], where the traditional approach to solving the least squares problem is replaced by a reduced-order vector form involving the Khatri–Rao product.

3. An algorithm with preconditioning by diagonal scaling

In principle, we can apply the procedure outlined in the previous section to solve highly over-determined linear inverse problems in a way that avoids dealing directly with a generic matrix K of the size $M \times N$. Importantly, none of the numerical steps involved in this procedure scale as $O(MN^2)$. There is also no need to store a large matrix K in memory. However, we now wish to further explore the properties of the matrices A and B. The motivation is that we expect the ranks of A and B can be smaller or even much smaller than N_d and N_s, respectively, where we have assumed that $N_d, N_s < N_v$ for simplicity.

Let the SVDs of A and B be given by

$$A = \sum_{\mu=1}^{N_d} \sigma_\mu^A f_\mu^A g_\mu^{A*}, \quad B = \sum_{\mu=1}^{N_s} \sigma_\mu^B f_\mu^B g_\mu^{B*}. \tag{8}$$

Here σ_μ^A $(1 \leqslant \mu \leqslant N_d)$ and σ_μ^B $(1 \leqslant \mu \leqslant N_s)$ are the singular values of A and B, and f_μ^A, g_μ^A, f_μ^B, g_μ^B are the corresponding singular vectors. Note that f_μ^A are of the length N_d, g_μ^B are of the length N_s, and g_μ^A, f_μ^B are of the length N_v. Star denotes transposition and complex conjugation of all elements. Then we have

$$K_{(ij),n} = \sum_{\mu=1}^{N_d} \sum_{\nu=1}^{N_s} \sigma_\mu^A \sigma_\nu^B f_{\mu i}^A \, g_{\mu n}^{A*} f_{\nu n}^B \, g_{\nu j}^{B*}. \tag{9}$$

Let us now introduce a unitary matrix U with elements $U_{(\mu\nu),(ij)} = f_{\mu i}^{A*} g_{\nu j}^B$. We multiply (3) by U on the left to obtain

$$(UK)x = Ub. \tag{10}$$

Note that multiplication of a linear set of equations by a unitary matrix does not change the pseudoinverse solution. However, the expressions for UK and Ub in terms of the singular vectors of A and B are of a very simple form:

$$(UK)_{(\mu\nu),n} = \sigma^A_\mu \sigma^B_\nu \, g^{A*}_{\mu n} f^B_{\nu n}, \tag{11a}$$

$$(Ub)_{(\mu\nu)} = \sum_{(ij)} f^{A*}_{\mu i} \Phi_{ij} g^B_{\nu j} = (f^A_\mu, \Phi g^B_\nu) \equiv \tilde{\Phi}_{\mu\nu} \, . \tag{11b}$$

Here the scalar product of two complex vectors of the same length, x and y, is defined as $(x, y) = \sum_n x^*_n y_n$. Now, with the use of (11), we can re-write (10) in components as

$$\sigma^A_\mu \sigma^B_\nu \sum_{n=1}^{N} g^{A*}_{\mu n} f^B_{\nu n} \, x_n = \tilde{\Phi}_{\mu\nu}. \tag{12}$$

Equation (12) is equivalent to (3). However, we now see that each line in this equation contains an overall factor $\sigma^A_\mu \sigma^B_\nu$, which can be very small or zero for some $(\mu\nu)$. Some equations in (12) can therefore be irrelevant; they do not contain any meaningful information about x. This fact is not evident when the equation is written in the form (3).

We now introduce the operation of diagonal scaling. Consider the diagonal matrix D defined by

$$D_{(\mu\nu),(\mu'\nu')} = \delta_{(\mu\nu),(\mu'\nu')} \begin{cases} \frac{1}{\sigma^A_\mu \sigma^B_\nu}, & \sigma^A_\mu, \, \sigma^B_\nu > \epsilon \\ 0, & \sigma^A_\mu, \, \sigma^B_\nu \leqslant \epsilon \end{cases} \tag{13}$$

where ϵ is a cut-off parameter. Then multiplication of (10) by D results in

$$(DUK)x = (DU)b \tag{14}$$

or, in components,

$$\sum_{n=1}^{N} g^{A*}_{\mu n} f^B_{\nu n} \, x_n = \frac{1}{\sigma^A_\mu \sigma^B_\nu} \tilde{\Phi}_{\mu\nu}. \tag{15}$$

The set (15) contains equations with μ and ν such that $\sigma^A_\mu > \epsilon$ and $\sigma^B_\nu > \epsilon$. If these conditions hold for all μ and ν in the range $1 \leqslant \mu \leqslant N_d$ and $1 \leqslant \nu \leqslant N_s$, then (15) and (12) contain the same number of equations $M = N_s N_d$. In this case, D is invertible. This does not mean however that (15) is equivalent to (12). The equivalence holds if K is also invertible (U is unitary and therefore invertible). If both D and K are both invertible, (15) and (12) have the same unique solution. However, we do not generally know whether K is invertible and, in many practical problems, it is not.

Let us assume that the inequality $\sigma^A_\mu > \epsilon$ holds for $1 \leqslant \mu \leqslant M_A \leqslant N_d$ and similarly the inequality $\sigma^B_\nu > \epsilon$ holds for $1 \leqslant \mu \leqslant M_B \leqslant N_s$. Then (15) contains $M_A M_B$ equations. This number can be smaller or even much smaller than $M = N_d N_s$ but still significant and, more importantly, (15) is not square. However, we can compute $W \equiv (DUK)^*(DUK)$ using the same approach as in section 2, i.e.

$$W_{mn} = \sum_{\mu=1}^{M_A} (g^A_{\mu m} g^{A*}_{\mu n}) \sum_{\nu=1}^{M_B} (f^B_{\nu n} f^{B*}_{\nu m}) \tag{16a}$$

$$= (A^+ A)_{mn} (BB^+)_{nm}, \tag{16b}$$

where we have introduced the pseudo-inverses of A and B,

$$A^+ = \sum_{\mu=1}^{M_A} \frac{1}{\sigma_\mu^A} g_\mu^A f_\mu^{A*}, \quad B^+ = \sum_{\mu=1}^{M_B} \frac{1}{\sigma_\mu^B} g_\mu^B f_\mu^{B*}. \tag{17}$$

The convention here is that the singular values are arranged in descending order so that $\sigma_\mu^A > \epsilon$ and $\sigma_\mu^B > \epsilon$ holds for all terms in (17). Thus, W_{mn} is the direct (Hadamard) product of the matrices A^+A and BB^+ where the pseudoinverses A^+ and B^+ were defined above. By comparing this to the expression (5) for K^*K, we see that the operation of preconditioning by diagonal scaling amounts to replacing the factors A^*A and BB^* with A^+A and BB^+. Obviously, W is different from K^*K. Moreover, W and K^*K can have different rank.

With these definitions and the use of Tikhonov regularization, (15) is transformed to

$$\sum_{m=1}^{N_v} (W_{nm} + \lambda^2 \delta_{nm}) x_m = c_n, \tag{18}$$

where

$$c_n = \sum_{\mu=1}^{M_A} \sum_{\nu=1}^{M_B} \frac{g_{\mu n}^A \tilde{\Phi}_{\mu\nu} f_{\nu n}^{B*}}{\sigma_\mu^A \sigma_\nu^B} \tag{19a}$$

$$= \sum_{(ij)} \left(\sum_{\mu=1}^{M_A} \frac{g_{\mu n}^A f_{\mu i}^{A*}}{\sigma_\mu^A} \right) \Phi_{ij} \left(\sum_{\nu=1}^{M_B} \frac{g_{\nu j}^B f_{\nu n}^{B*}}{\sigma_\nu^B} \right) \tag{19b}$$

$$= (A^+ \Phi B^+)_{nn} \,. \tag{19c}$$

We could have started the derivations of this section with (16*b*) and (19*c*), which can be obtained in a straightforward manner by acting with A^+ on the left and B^+ on the right on (2). However, we have outlined above a series of steps that are better suited for numerical implementation.

4. Computational algorithms

We can now describe two algorithms for finding a solution to (2). Algorithm 1 yields the pseudoinverse of this equation; its basic steps as well as the estimates of computational complexity of each step are listed in table 1. The estimates were performed for the typical case $N_d, N_s < N_v < N_d N_s$. We do not list the memory requirements since this depends on programming. However, the memory bottleneck of algorithm 1 is roughly $2N_v^2$ words, where one word is 8 bytes for double-precision real arithmetic. This estimate arises from the requirement to have at least two matrices of size $N_v \times N_v$ allocated simultaneously at Step 4 (note that three matrices are not required). For example, if the medium is discretized on a $41 \times 41 \times 41$ grid ($N_v = 68\,921$), the memory requirement is roughly 76 Gb. For the discretization of $21 \times 21 \times 21$ voxels ($N_v = 9261$), less than 2 Gb is required. We emphasize that these estimates are independent of the numbers of sources and detectors. Also, the memory requirement can be halved at the cost of degrading the computational performance.

Algorithm 2 yields the pseudoinverse of (2) after preconditioning by diagonal scaling. The basic steps of this algorithm are listed in table 2. A few differences relative to algorithm 1 can be noted. First, algorithm 2 computes the matrices AA^* and B^*B, which are smaller than A^*A and BB^* that are required by algorithm 1. Second, the memory bottleneck of algorithm 2 is

Table 1. Steps involved in algorithm 1. Here 5a and 5b are alternative steps. Step 5a entails computing the pseudoinverse x^+ by an iterative method such as the conjugate gradient descent and J is the number of iterations (typically, $J \ll N_v$). Step 5b entails computing the full SVD decomposition of K^*K.

#	Operation	Complexity $[O(\cdot)]$
1	Compute or read from disk A, B, Φ	$(N_d + N_s)N_v + N_sN_d$
2	Compute c_n by (6)	$N_vN_dN_s$
3	Compute A^*A and BB^*	$(N_d + N_s)N_v^2$
4	Compute K^*K by (5)	N_v^2
5a	Compute $x^+ = (\lambda^2 I + K^*K)^{-1}c$	JN_v^2
5b	Compute full SVD of K^*K	N_v^3

$(M_A + M_B)N_v$ words, which can be significantly smaller than the respective bottleneck $2N_v^2$ words of algorithm 1. Algorithm 2 is more intricate and involves more steps. One needs to pay close attention to memory management since algorithm 2 requires temporary storage that can be allocated and deallocated when needed. Overall, algorithm 2 is slightly faster than algorithm 1 and can be significantly less memory intensive. The reason for this improvement is the useful data reduction that is involved in the operation of diagonal scaling (14). In some cases, algorithm 2 also provides better reconstructions, as is illustrated in the numerical examples below. Finally, we note that all computationally expensive steps of both algorithms are easily amendable to parallelization.

5. Numerical examples with ideal data

To illustrate the algorithms described in section 4, we have performed numerical simulations for the inverse problem of reconstructing a three-dimensional cubic sample represented by an array of $L \times L \times L$ voxels. In this section, we show inversions with ideal (noiseless) data. Choosing the most efficient algorithm in the presence of noise in the data, and the associated problem of data reduction are discussed in section 6 below.

We introduce an unknown x_n for each voxel where $1 \leqslant n \leqslant N_v = L^3$. The source and detector positions are shown schematically in figure 1. We emphasize that the positions of sources and detectors must be independent sets for the methods of this paper to be applicable. In other words, we require that all elements of the data matrix Φ in (2) be known. In the numerical examples shown below, the sets of positions of sources and detectors are the same. In particular, it is assumed that measurements can be performed for a source and a detector located at the same point, i.e. if the physical device used for the measurements can function as a source and a detector concurrently. Additional details as well as the definitions of the sampling frequency F and the imaging window W are given in the caption to figure 1.

Further, we choose the form of the Green's function in (1) to be

$$G_0(\mathbf{r}, \mathbf{r}') = \frac{1}{|\mathbf{r} - \mathbf{r}'|} \mathrm{e}^{-k_d|\mathbf{r}-\mathbf{r}'|}. \tag{20}$$

This, together with the locations of the voxel centers, sources and detectors, determine the matrices A and B in (2). Specifically, we have

Table 2. Steps involved in algorithm 2. Same comment regarding the alternative steps 14a and 14b as in the caption for 1 applies. $\tilde{x}^+$ denotes pseudo-inverse after preconditioning by diagonal scaling (it is in general different from the pseudo-inverse of (3)).

#	Operation	Complexity $[O(\cdot)]$
1	Compute or read from disk A, B, Φ	$(N_d+N_s)N_v+N_sN_d$
2	Compute AA^*	$N_d^2N_v$
3	Diagonalize AA^*, compute f_μ^A, σ_μ^A, $1\leqslant\mu\leqslant N_d$	N_d^3
4	Define M_A; truncate f^A according to (13)	—
6	Compute g_μ^A, $1\leqslant\mu\leqslant M_A$ by $g_\mu^A=\frac{1}{\sigma_\mu^A}A^*f_\mu^A$	$M_AN_dN_v$
7	Compute B^*B	$N_s^2N_v$
8	Diagonalize B^*B, compute g_μ^B, σ_μ^B, $1\leqslant\mu\leqslant N_s$	N_s^3
9	Define M_B; truncate g^B according to (13)	—
10	Compute f_μ^B, $1\leqslant\mu\leqslant M_B$ by $g_\mu^B=\frac{1}{\sigma_\mu^B}Bf_\mu^B$	$M_BN_sN_v$
11	Compute $\tilde{\Phi}$ by (11*b*)	$M_BN_d(N_s+M_A)$
12	Compute c_n by (19*a*)	$M_AM_BN_v$
13	Compute W by (16*a*)	$(M_A+M_B)N_v^2$
14a	Compute $\tilde{x}^+=(\lambda^2I+W)^{-1}c$	JN_v^2
14b	Compute full SVD of W	N_v^3

$$A_{in}=\frac{1}{|\mathbf{r}_{di}-\mathbf{r}_n|}\mathrm{e}^{-k_d|\mathbf{r}_{di}-\mathbf{r}_n|} \tag{21}$$

where $\mathbf{r}_{di}$ is the location of the ith detector and $\mathbf{r}_n$ is the center of the nth voxel. A similar expression can be written for B (involving locations of the sources). However, since the sets of the sources and detectors are the same in the simulations performed here, we have $B=A^T$. We emphasize that this is not a general property of A and B and, in general, these two matrices are independent. In the simulations shown below, this property was not used (it can be used to save computation time). We also note that (20) is, up to a constant, the free -space Green's function of the diffusion equation, $\frac{1}{4\pi D_0|\mathbf{r}-\mathbf{r}'|}\mathrm{e}^{-k_d|\mathbf{r}-\mathbf{r}'|}$, where $k_d=\sqrt{\alpha_0/D_0}$, and α_0 and D_0 are the absorption and diffusion coefficients of the background medium. In the simulations, we chose the units of length so that $k_d=1$ and renormalized the contrast of the medium so that $4\pi D_0=1$. Since in many applications $k_d\sim 1\ \mathrm{cm}^{-1}$, one can say that the distance in (20) is measured in centimeters.

Thus, the problem considered here is that of linearized diffuse optical tomography with free boundaries [28, 29]. Methods of linearization that can be used to arrive at this mathematical formulation are summarized in the above reference. The problem is notoriously ill-posed and the mathematical origin of the ill-posedness is similar to that of numerical inversion of Laplace transform [30]. However, we can make use of the three-dimensional nature of the problem to try to reduce the ill-posedness as much as possible. In the spirit of [31, 32], we can place the planes of sources and detectors adjacent to each face of the cubic sample. It is known that, in transmission-type measurements, wherein there is a plane of sources on one side of the sample and a plane of detectors on the opposite side, the transverse resolution (in the directions parallel to the planes) is much better than depth resolution. In the geometrical arrangement of this paper, we place the planes of sources and detectors adjacent to each face of the cubic sample so that *any* direction can be considered transverse with respect to

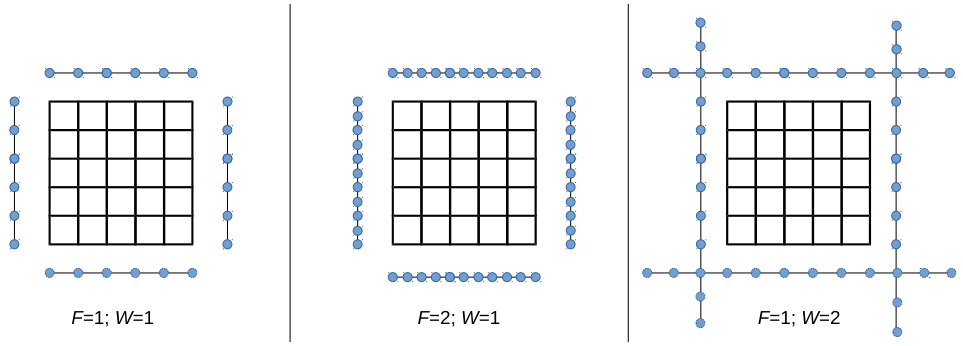

Figure 1. Illustration of the measurement scheme used in the simulations. The actual sample is a three-dimensional cube and the measurement surfaces are planes located near each of the six sides of the cube. The centers of the voxels are located at the intersections of black lines; the sample shown is a two-dimensional projection of a $6 \times 6 \times 6$ three-dimensional sample. The sets of sources and detectors are the same; each blue point can represent either a source or a detector and all source-detector pairs are used in the measurements, including the cases wherein the location of the source and the detector is the same. Arrangements with $W = 1$ correspond to the imaging window being the same size as the side of the cube (or square in 2D). The case $W = 2$ corresponds to the window being twice as large as the side of the cube. Sampling frequency $F = 1$ corresponds to the inter-source (inter-detector) separations being equal to the voxel size. The case $F = 2$ corresponds to twice larger sampling frequency. The case $W = F = 2$ is considered in the paper but not shown in this sketch. Finally, the measurement planes are displaced from the centers of nearest voxles by a distance equal to the size of one voxel.

some transmission and/or reflection measurement, which are all included in the data set. This arrangement will indeed allow us to overcome some of the ill-posedness at the cost of having to deal with extremely large data sets. The size of the data sets would make it hard or impossible to perform image reconstruction using conventional methods. However, the methods described in this paper will prove to be efficient.

In the examples shown below, the side of the cube measured from a voxel center in a surface layer of a cube to the voxel center at the opposite surface layer is $H = 5$. The physical size of the sample (surface to surface) is $H + h$ where $h = H/(L - 1)$ is the voxel size. The distance between opposing measurement planes is $H + 2h$. In the typical setup of optical tomography with the diffuse wave number $k_d \approx 1$ cm^{-1}, this corresponds to a physical sample being a cube with sides $\sim$5 cm. The cube however will be differently discretized. We will be looking for geometrical features on the scale of a few millimeters.

5.1. Effects of imaging window and sampling frequency

In figure 2, we plot the squared singular values of K (eigenvalues of K^*K) for a cube of side $L = 21$. The eigenvalues were computed by algorithm 1. The purpose of this simulation is to show that there is virtually no benefit in oversampling the sources and detectors relative to the voxels of the medium or in using imaging windows larger than the face of the cube. The geometrical quantities F (the sampling frequency) and W (the imaging window) are illustrated in figure 1 and explained in detail in the caption to that figure. We have investigated different combinations of parameters with F and W taking the values 1 or 2.

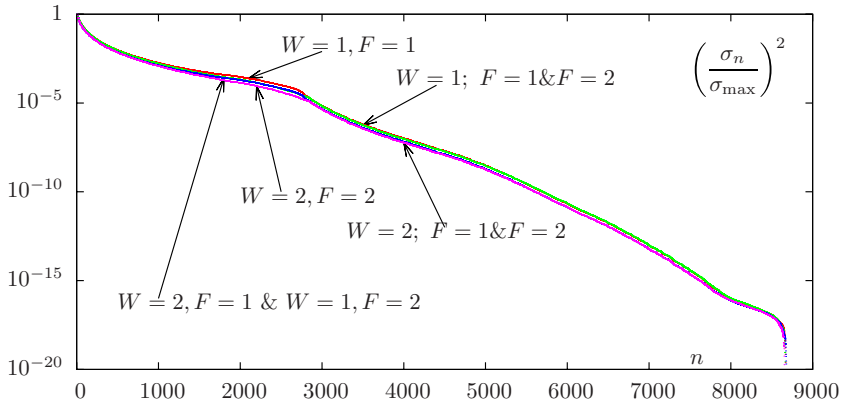

Figure 2. Squared singular values of the matrix K for a $21 \times 21 \times 21$ cube of voxels and various source-detector arrangements, as labeled. The total number of singular values is $21^3 = 9261$.

It can be seen that there exist two distinct regions of singular values. For the indexes less than $\sim$2800, the data points with $(F = 1, W = 1)$ are located somewhat higher than the data points for $(F = 2, W = 2)$. The data points with either $(F = 1, W = 2)$ or $(F = 2, W = 1)$ are located roughly in the middle and are visually indistinguishable from each other. Note that these singular values (and the corresponding singular vectors) contribute mostly to stable reconstructions while the smaller singular values (and the corresponding singular vectors) are typically ignored in computing the pseudo-inverse solution (due to regularization). We thus conclude that the choice $(F = 1, W = 1)$ yields the most stable reconstructions, although the differences are really very minor.

In the second region where the singular value indexes are between $\sim$2800 and $\sim$8000 shows a slight benefit of oversampling but not of using a larger window. However, the singular values in this region are already so small that this improvement is not expected to matter in any realistic reconstruction. In order for the benefit of oversampling to be noticeable, the data must be known with an unrealistically high precision.

We note that the data of figure 2 display no instabilities associated with diagonalization of K^*K rather than seeking the singular values of K directly, i.e. by the GR algorithm. It is widely believed that diagonalization of K^*K is not the preferred method since the singular values thus computed suffer from loss of relative precision related to computing the square root of a small number. In our simulations, the only points affected by this instability are those with indexes $\gtrsim$8000. In this case, the singular values are already so small that they are definitely discarded by any regularized inversion. Note that diagonalization of K^*K produces eigenvalues that are very small (on the order of the machine precision) and sign indefinite while it is known theoretically that all eigenvalues of K^*K are non-negative. Therefore, for such small eigenvalues we have a total loss of precision. The GR algorithm produces only positive singular values but they are limited from below by the machine precision and can not be smaller than a certain machine-defined number. The actual singular values can be many orders of magnitude smaller. Therefore, even in the GR algorithm computation of these small singular values is characterized by a total loss of precision. In figure 2, the singular values that are affected by this instability are seen as the data points of the order of $\sim 10^{-20}$ that descend with an almost vertical slope, at the very end of the 'curves'. In fact, there are also negative eigenvalues in that region of indexes but they can not be displayed using the vertical logarithmic scale. The

conclusion is that the singular values affected by the instability are so small that they can not be computed reliably by any method.

Another comment needs to be made regarding the imaging windows. That taking imaging windows larger than the side of the cube does not provide any additional information about the sample is not unexpected. In the case $W = 1$, the sources and detectors sample a closed surface enclosing the sample. Any data points outside of these closed surface are not independent. However, in purely transmission measurement schemes, wherein there is one plane of sources on one side of the sample and one plane of sources on the other side, taking larger windows (than the discretized region) actually makes sense. In this case, the source-detector planes do not form a closed surface and using data points in a large window affords additional information by looking at the sample 'from the side'. The effects of imaging windows and sampling in purely transmission measurements have been investigated by us in [28, 29].

Since we have confirmed that the source-detector arrangement with $F = W = 1$ is optimal and no further image quality improvement can be obtained by over-sapling or using larger windows, most reconstructions below are performed for $F = W = 1$, except for the very last figure below in which we compare the influence of various source-detector arrangements in order to illustrate the usefulness of large data sets.

5.2. Cube with $L = 21$

Reconstruction of three different targets contained within a cubic sample discretized on a $L \times L \times L$ grid with $L = 21$ are shown in figure 3. For this arrangement, the total number of voxels is $N_v = 21^3 = 9261$ ans the numbers of sources and detectors are $N_s = N_d = 2646$. The total number of source-detector pairs (the leading size of K) is $7001\,316$. In spite of the relatively small discretization of the sample, this already can be viewed as a formidable computational problem: the matrix K contains $\sim 6.5 \cdot 10^{10}$ elements; only storage of its elements in memory requires about 0.5 Tb of memory (in double precision). However, both algorithms 1 and 2 require less than 2 Gb of memory and complete in under one minute of wall-clock time on a 16-thread entry-level workstation. Note that this included computation of full SVD of K^*K (for algorithm 1) or W (for algorithm 2) and scanning over 10^3 different values of λ to determine the optimal value of this regularization parameter.

We now discuss the results of figure 3 in more detail. First, the large and small target consisted of three cubic shells. The cross section of these shells by the central plane is shown in the top row of images. The contrast was equal to 2 in the outer-most shell, to -1 in the middle shell and to 1 in the inner-most shell (actually, a small cube). The sizes of the shells were 17, 9 and 5 for the large target, and 9, 5 and 3 for the small target. The asymmetric target shown in the middle column is the large target in which parts of the outer and middle shells were removed (more precisely, set to zero). Specifically, assuming that the XY central plane is shown in the reconstructions, all values to the left of the XZ central plane of voxels were set to zero for the outer shell and all values below the YZ central plane were set to zero for the middle shell; the innermost shell was unchanged.

Reconstructions were performed by both algorithms with comparable time and results, although the reconstructions according to algorithm 2 are slightly faster, less memory-demanding and produce visibly better results for the large target. The cut-off parameter ϵ that is used in the operation of diagonal scaling (13) was set to $10^{-3} \max_\mu [\sigma_\mu^A]$. This choice was made by analyzing the singular values of the matrix A (identical to those of B). These singular values are shown in figure 4. It can be see that there exists a natural gap between the singular values. The singular vectors g_μ^A that correspond to the singular values below the gap are *non-radiating*

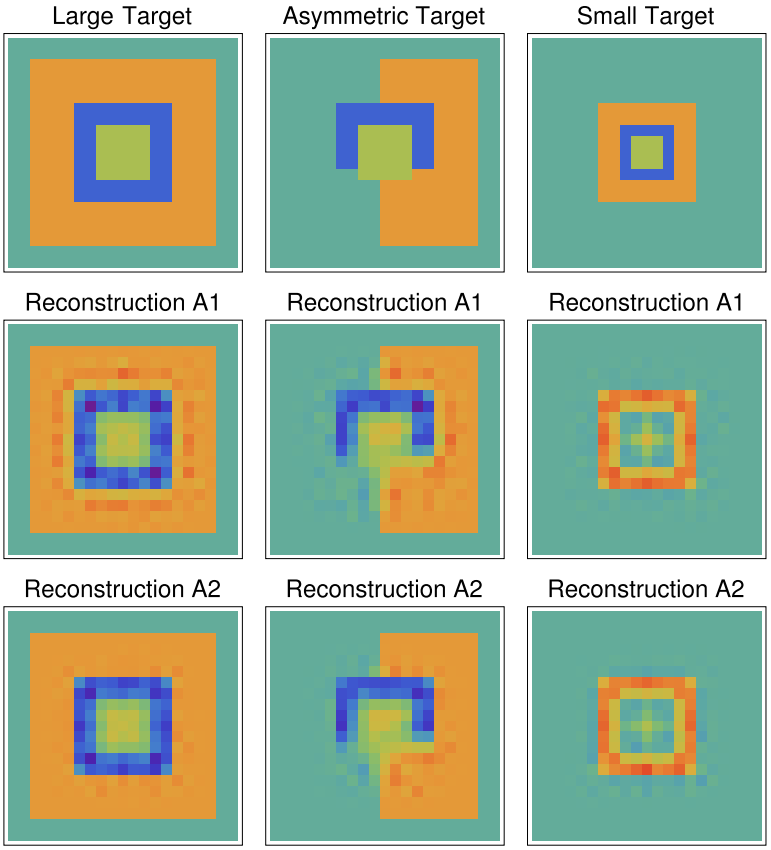

Figure 3. Reconstructions of three different targets for the medium discretized on a 21^3 cube. Central slice for the cube is shown. A1 and A2 denote reconstructions by algorithms 1 and 2. For algorithm 2, we have used $\epsilon = 10^{-3}\max_\mu[\sigma_\mu^A]$ (this small parameter appears in (13)). Same color scale is used in all plots with dark blue representing the value of -2 and red representing $+3$ (rainbow color scheme). The values of the contrast in the actual target are form -1 to $+2$; the extra range was used to depict reconstructed values that are either larger or smaller than the theoretical bounds (due to the imprecision of reconstructions).

states; they represent the states that do not produce any noticeable signal for any detectors located on a surface completely enclosing the sample; therefore, these states do not generate signal that is measurable by any detector or combination of detectors. These singular vectors can be safely discarded, and the operation of diagonal scaling achieves exactly that.

In figure 5 we plot the L^2 norm of the reconstruction error (for the large target) as a function of the Tikhonov regularization parameter λ. Tikhonov regularization is employed at Step 5 of algorithm 1 or Step 14 of algorithm 2. It can be seen that algorithm 2 has indeed a slightly deeper minimum of the function $\chi(\lambda)$. The optimal value of λ was used in the reconstructions of figure 3.

We finally note that the sum of the unknowns, $X = \sum_n x_n$, was reconstructed with very high precision in all cases. Apparently, the measurements are very sensitive to this integral characteristic. However small features buried deep inside the sample proved to be difficult to reconstruct. This is not a shortcoming of the algebraic method proposed here but rather the consequence of the intrinsic ill-posedness of the inverse problem.

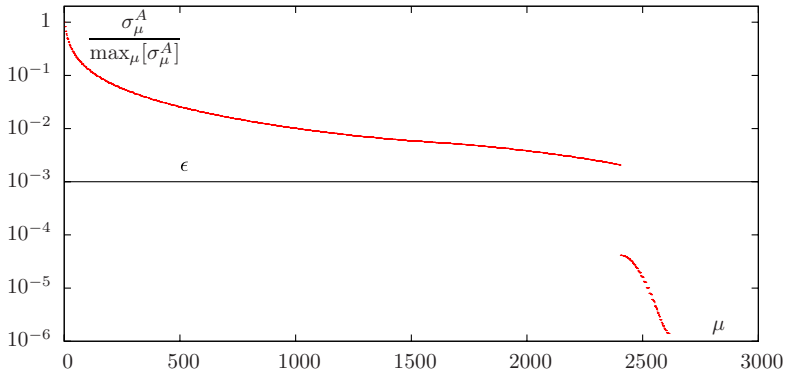

Figure 4. Singular values of A scaled by the maximum singular value and the cut-off parameter ϵ used in (13) and in the reconstructions of figure 3.

5.3. Cube with L = 35

We now show reconstructions for a cube of the same physical size but discretized on an L^3 grid where $L = 35$. We now have $N_v = 42\,875$; $N_s = N_d = 7350$ and the total number of source-detector pairs (the leading dimension of K) is 54 022 500. The matrix K now has $\sim 2.3 \cdot 10^{12}$ elements, which, under normal circumstances, can not be stored in computer memory (about 18 Tb is required). However, the reconstructions by algorithm 2 can still be performed on an entry-level workstation in about 90 min of wall-clock time (using 16 threads). The time is mostly used for the operations of complexity $O(N_v^3)$. Even though this computational time is longer than what one would hope for, inversion of K by standard methods is simply unfeasible in this case.

Reconstructions obtained by algorithm 2 are shown in figure 6. Algorithm 1 was not used for $L = 35$ as it does not provide any improvement and requires more memory. As in the previous case, the total contrast $X = \sum_n x_n$ is reconstructed with very high precision, but small features buried deep inside the sample are difficult to resolve. We do not display the data similar to those shown in figures 4 and 5 as they all look similar and do not convey new qualitative information.

5.4. Cube with L = 41

We now show our most ambitious reconstructions with $L = 41$. Indeed, we have in this case $N_v = 68\,921$, $N_s = N_d = 10\,086$, and the total number of source-detector pairs (the leading dimension of K) is 101 727 396. The matrix K has $\sim 7 \cdot 10^{12}$ elements. The computational bottleneck for this sample is not the number of data points (which can easily be increased) but the number of voxels. Still, even for $L = 41$, computational time is not a significant concern if modern multi-threaded computers are used. Rather, the method is limited by the available memory. An efficient numerical implementation requires simultaneous allocation of two matrices of the size N_v^2 for algorithm 1 and of the sizes $M_A \times N_v$ and $N_v \times N_B$ for algorithm 2. This means that about 76 Gb of memory are required for algorithm 1 to run in two to three hours of wall-clock time. The memory requirements of algorithm 2 can be significantly lower (see below) but are still significant. The required amount of memory is currently not typical in entry- to moderate-level workstations. In principle, one can program both algorithms so that they require much less memory, but then the implementation becomes much slower. For this reason, the reconstructions with $L = 41$ reported here were performed on a supercomputer.

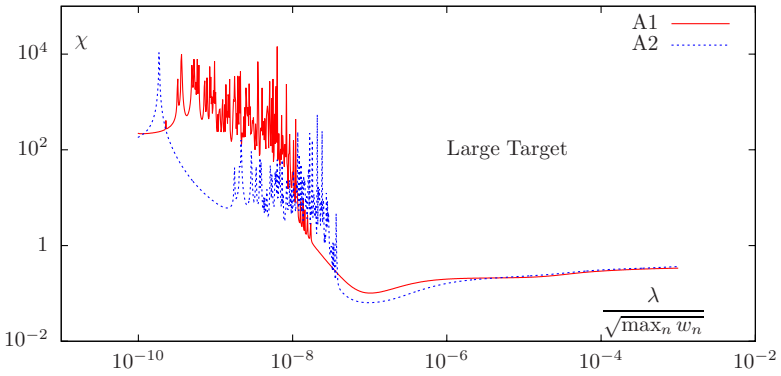

Figure 5. L^2 norm of the reconstruction error χ for the large target as a function of the Tikhonov regularization parameter λ. Here $\max_n[w_n]$ is the maximum eigenvalue of K^*K for algorithm 1 and of W for algorithm 2.

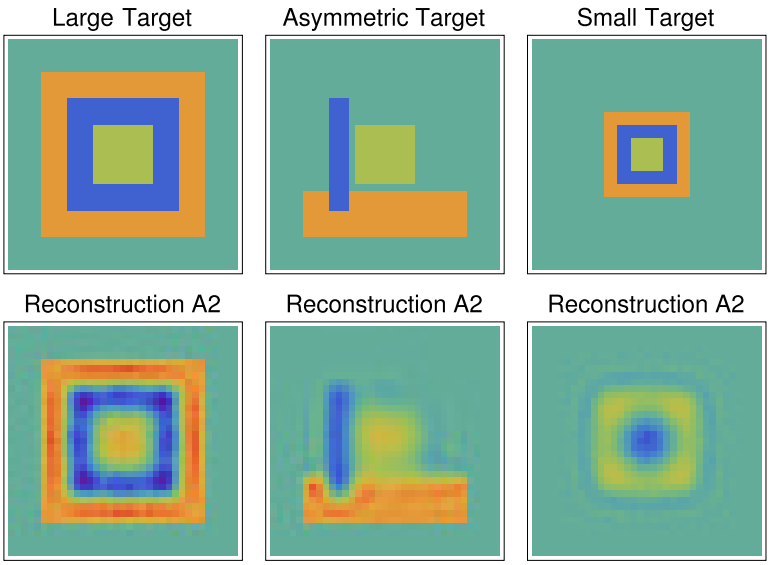

Figure 6. Same as in figure 3 but for a cubic sample discretized on a $35 \times 35 \times 35$ grid. Definitions of the model targets is slightly different and only reconstructions by algorithm 2 are shown as algorithm 1 requires in this geometry significantly more memory. Same value $\epsilon = 10^{-3} \max_\mu[\sigma^A_\mu]$ is used as in figure 3 (it is still located in the gap of singular values).

Reconstructions obtained by both algorithms are shown in figure 7 and appear to be of comparable quality. However, we note that the reconstructions by algorithm 2 required approximately 6 times less memory. This is so because the cut-off parameter ϵ that was used in the operation of diagonal scaling (13) was set to $10^{-2} \max_\mu[\sigma^A_\mu]$, which is well above the gap in the singular values of A (or B) as is shown in figure 8. In this case, $M_A = M_B = 1724 \approx N_v/6$. We note that, in the case of a very large sample considered here, this choice is justified; reducing ϵ to a value inside the gap (i.e. $\epsilon \sim 10^{-4}$), as it was done in the case of $L = 21$, does not result in any improvement of the image quality. The first M_A singular vectors of A already

probe the target with near-optimal precision and keeping more of these singular vectors, even if the corresponding singular values are not too small, is not useful for solving the inverse problem. Therefore, the operation of diagonal scaling achieves a useful *data reduction* which, in turn, allows one to significantly reduce the memory requirements of the associated numerical algorithm. This does not mean that not all data were used or that one can achieve a similar image quality with less real-space data points, and this point is illustrated below. Rather, the diagonal scaling selects the linear combinations of the data points that are informative; it is not practically possible or at least very difficult to realize such linear combinations in a physical experiment directly.

It can be concluded that the use of algorithm 2 can be justified by computational efficiency if not by superior quality of reconstructions. However, a more systematic comparison of the two methods is required to understand the subtle interplay of the operation of diagonal scaling, ill-posedness of the problem and round-off errors that are characteristic of any numerical solution of a large-scale algebraic problem. As above, the total contrast $X = \sum_n x_n$ is reproduced by both algorithms very well but small deep-buried features are elusive.

Finally, we provide a demonstration that very large data sets can indeed be useful for obtaining the best possible quality of reconstructions. To this end, we compare the reconstructions of the large target obtained in the surrounding measurement geometry shown in figure 1 with the sampling frequencies $F = 1$ and $F = 0.5$ (so far we have used only the 'perfect' sampling frequency $F = 1$) with the transmission measurement geometry wherein the sources are located close to one face of the cube and the detectors are placed at the opposite face, again, for $F = 1$ and $F = 0.5$. To avoid making an additional choice associated with adjustable the cut-off parameter ϵ, the reconstructions in this figure were computed by algorithm 1. The results are shown in figure 9 where we show *XY* and *XZ* slices. Both slices are drawn through the center of the cube, but the *XY* slices are parallel to the planes of sources and detectors in the case of transmission measurements while the *XZ* slices are perpendicular to these planes and, in the latter case, the direction of the *Z*-axis (the horizontal direction in the right column of images) can be regarded as 'depth' while the *X* and *Y* directions are lateral. In the case of surrounding measurements, the *XY* and *XZ* slices are equivalent, as can indeed be seen from the figure.

We can conclude from the data of figure 9 that reduction of the sampling frequency from 1 to 0.5 in the surrounding measurement geometry results in a modest loss of quality. This is illustrated in a more quantitative way in figure 10. However, transitioning from the surrounding measurement scheme to the transmission scheme leads to a dramatic loss of precision. This can be understood by noting that the resulting resolution in the depth direction becomes quite poor and the lateral resolution has also deteriorated. These results are consistent with the theoretical predictions of optimized resolution that can be achieved by employing 'multiple projection' measurement schemes [31, 32]. In fact, the surrounding measurement geometry realizes these 'multiple projections' so that every direction inside the sample can be regarded as lateral with respect to some pair of source-detector planes. However, when we move to the transmission measurement geometry, there is virtually no difference between the $F = 1$ and $F = 0.5$ cases. The reason is that the inverse problem is now so ill-posed that the advantages of using the perfect sampling frequency are very insignificant. We emphasize that this is not a general result: for a $L = 21$ cube, there is substantial difference between $F = 1$ and $F = 0.5$ transmission-geometry reconstructions (results not shown). In any event, the data of figures 9 and 10 confirms that the large data sets can be useful even in inverse problems that are as ill-posed as the inverse problem of diffusion tomography.

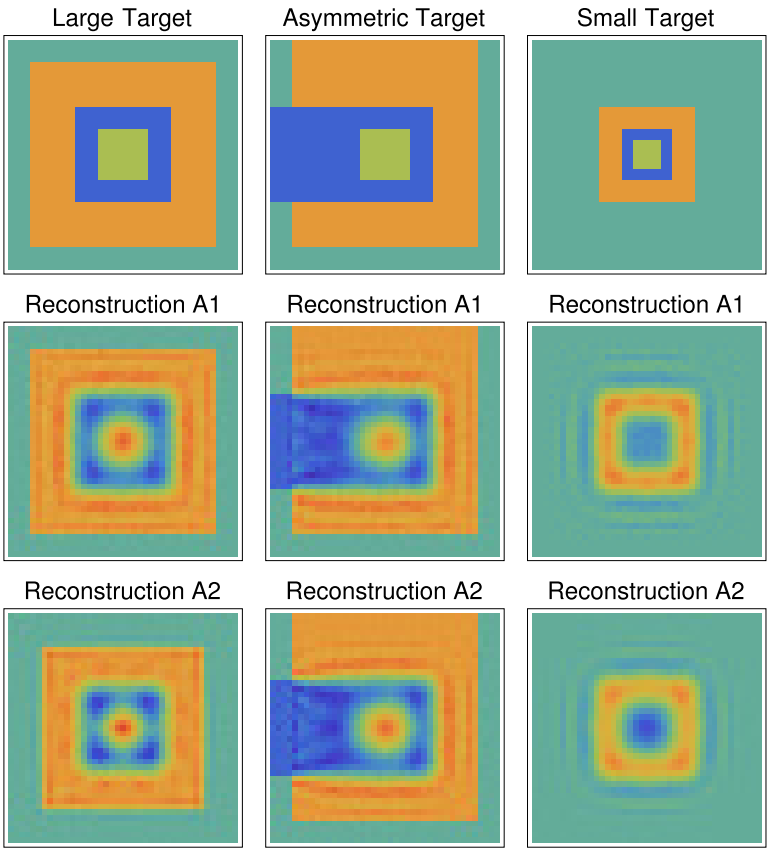

Figure 7. Same as in figures 3 and 6 but for a cubic sample discretized on a $41 \times 41 \times 41$ grid. Definitions of the model targets is slightly different from the previously-considered cases. For reconstructions by algorithm 2, the small parameter of diagonal scaling is $\epsilon = 10^{-2} \max_\mu[\sigma_\mu^A]$.

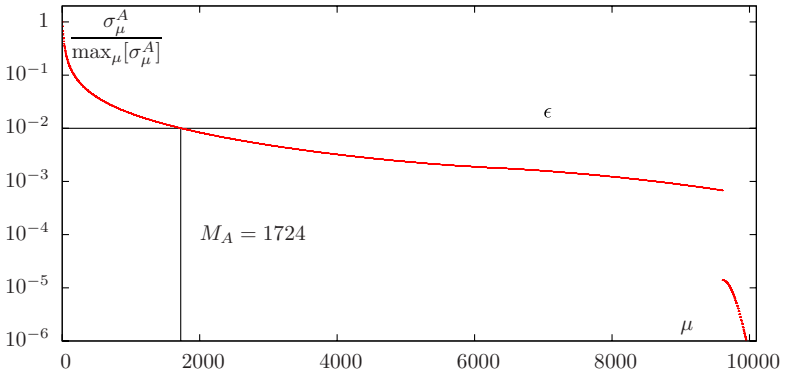

Figure 8. Singular values of A scaled by the maximum singular value and the cut-off parameter ϵ used in the A2 reconstructions of figure 7.

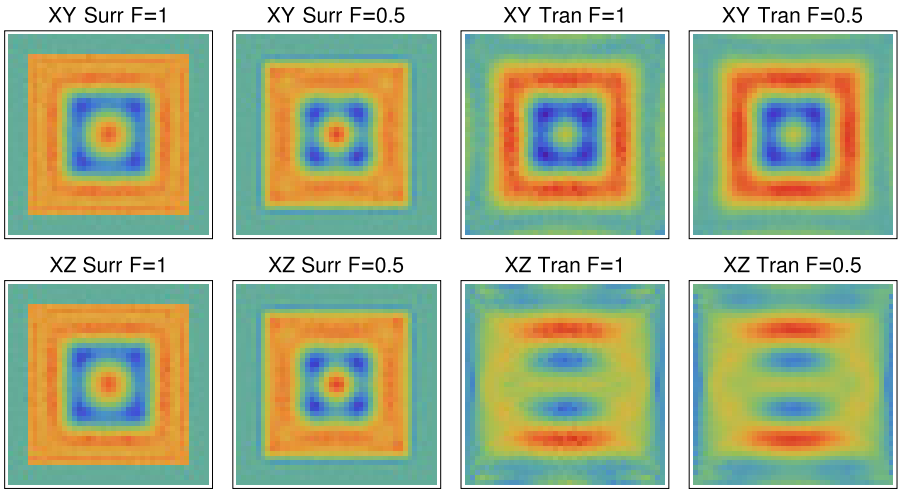

Figure 9. Reconstructions of the large target discretized on a $41 \times 41 \times 41$ cubic grid for various arrangements of sources and detectors. 'Surr' denotes the surrounding arrangement as is illustrated in figure 1. 'Tran' denotes transmission measurements with the plane of sources on one side of the cube and plane of detectors on the other side. The sampling frequency F is defined in the caption of figure 1. Two orthogonal central cross sections through the cube are shown. The left column shows the cross section by an XY-plane, which is parallel to the planes of sources and detectors in the case of transmission measurements. The right column shows cross sections by the XZ plane, which is perpendicular to these two planes. In the case of transmission measurements, reconstructions in the XY and XZ cross sections are not equivalent. Reconstructions obtained by algorithm 1.

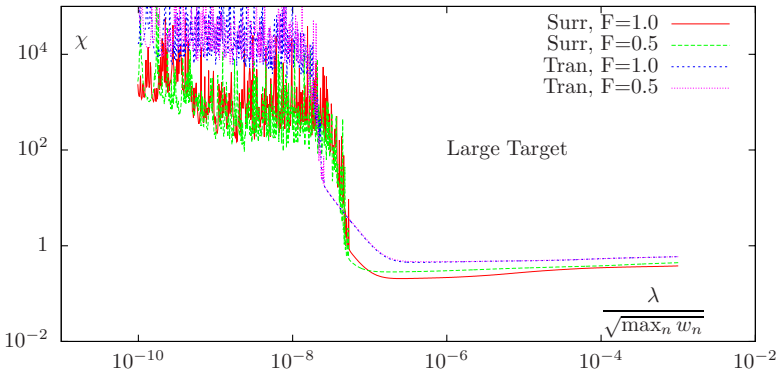

Figure 10. L^2 norm of the reconstruction error χ for the large target whose reconstructions are shown in figure 9 as a function of the Tikhonov regularization parameter λ. Here $\max_n[w_n]$ is the maximum eigenvalue of K^*K for algorithm 1.

6. Reconstructions with noise and reduction of noisy data

So far, we have considered reconstruction with ideal data; the only cause of imprecision in the above simulations are round-off errors. In practice, however, all measurements are affected by noise. In this section, we will show that algorithm 2 (with preconditioning by diagonal scaling) is well-suited for reconstructions of noisy data as it allows one to discard efficiently the

equations that are hopelessly corrupted by noise and do not carry useful information about the target. A similar data reduction technique can be applied in the traditional approach as well. One can, for example, exclude from the set (3) the equations for which the data point $b_{(ij)}$ is expected to be small and therefore overshadowed by noise [33], or is affected by some systematic error of the model [9]. However, these algorithms do not reduce the data set dramatically. In contrast, algorithm 2 allows one to reduce the number of equations well beyond the number of unknowns while keeping the image quality (attainable at a given noise level) almost unaffected. This may seem counter-intuitive since under-determined problems do not have unique solutions. However, we will show that the underdetermined problem obtained by selecting a large cut-off parameter ϵ in algorithm 2 contains almost the same information as all the original equations. Of course, some image degradation when equations are discarded can be expected but we will see that this degradation is modest.

The noise model we use is the following. We note that the data point $\Phi(\mathbf{r}_d, \mathbf{r}_s)$ in (1) is obtained by a differential measurement of some intensity, i.e, $\Phi(\mathbf{r}_d, \mathbf{r}_s) = C[I_0(\mathbf{r}_d, \mathbf{r}_s) - I(\mathbf{r}_d, \mathbf{r}_s)]$, where I is the measurement obtained for the actual sample and I_0 is obtained using a reference homogeneous sample, and C is some overall factor [29]. We assume that each measurement is affected by positive-definite shot noise. This is typical of CCD cameras that are used for detection in optical tomography [7–9]. For 16-bit cameras operating at the limit of their dynamic range, the average amplitude of the shot noise is approximately 100 counts, which should be compared to the maximum possible value of 65 536 counts. That is, the noise amplitude for the intensity measurements is $\sim$0.15% of the maximum measurable intensity. For differential measurements, the relative noise amplitude can be higher. In the below simulations, we assumed that each ideal data point Φ_{ij} is modified by noise as follows:

$$\Phi_{ij} \longrightarrow \Phi_{ij} + \frac{\gamma}{\mu}(n_1 - n_2), \tag{22}$$

where n_1 and n_2 are two statistically independent integer random numbers selected from the Poisson distribution $P(n) = \mu^n \mathrm{e}^{-\mu}/n!$ and the overall factor γ was varied in the simulations. Note that $\langle n_1 - n_2 \rangle = 0$ and $\langle (n_1 - n_2)^2 \rangle = 2\mu$. Since the average number of counts in typical shot noise is $\sim$100, we have selected $\mu = 100$.

The sample and the measurement geometry were exactly the same as in section 5.2 (the large target was used). The maximum data point in this geometry is $\Phi_{\max} \approx 37$ and the minimum is $\Phi_{\min} \approx 0.026$. If we take $\gamma = 1$, the maximum data point is well above the noise level. However, the weakest data points are completely overshadowed by noise.

Reconstructions are shown in figure 11. The left column was obtained by algorithm 1. The other columns were obtained by algorithm 2. Optimal regularization parameter was used at each noise level. In the second column, we show reconstructions with $\epsilon = 10^{-6}$. As expected, these reconstructions are inaccurate except in the case of noiseless data. Indeed, the operation of diagonal scaling multiplies the equations that are most affected by noise by large constants and therefore makes them more significant. As we increase the cut-off parameter ϵ, the reconstructions are more clear. At the largest value of noise used, $\gamma = 1$, the reconstructions by algorithms 1 and 2 with $\epsilon = 0.1$ are approximately of the same quality. The outer shell is clearly visible and is reconstructed more-or-less correctly, but the inner shell is not visible. However, the algorithm 2 reconstruction is much more efficient. Indeed, at $\epsilon = 0.1$, we have $M_A = M_B = 28$. The total number of equations retained in this case is $28^2 = 784$. This is significantly less than the total number of unknowns, which is $N_v = 21^3 = 9261$. The computational bottleneck is reduced by the factor $(9261/784)^3 \approx 1650$. This is the approximate computational speed-up that one obtains for the last (fifth) column of figure 11 relative

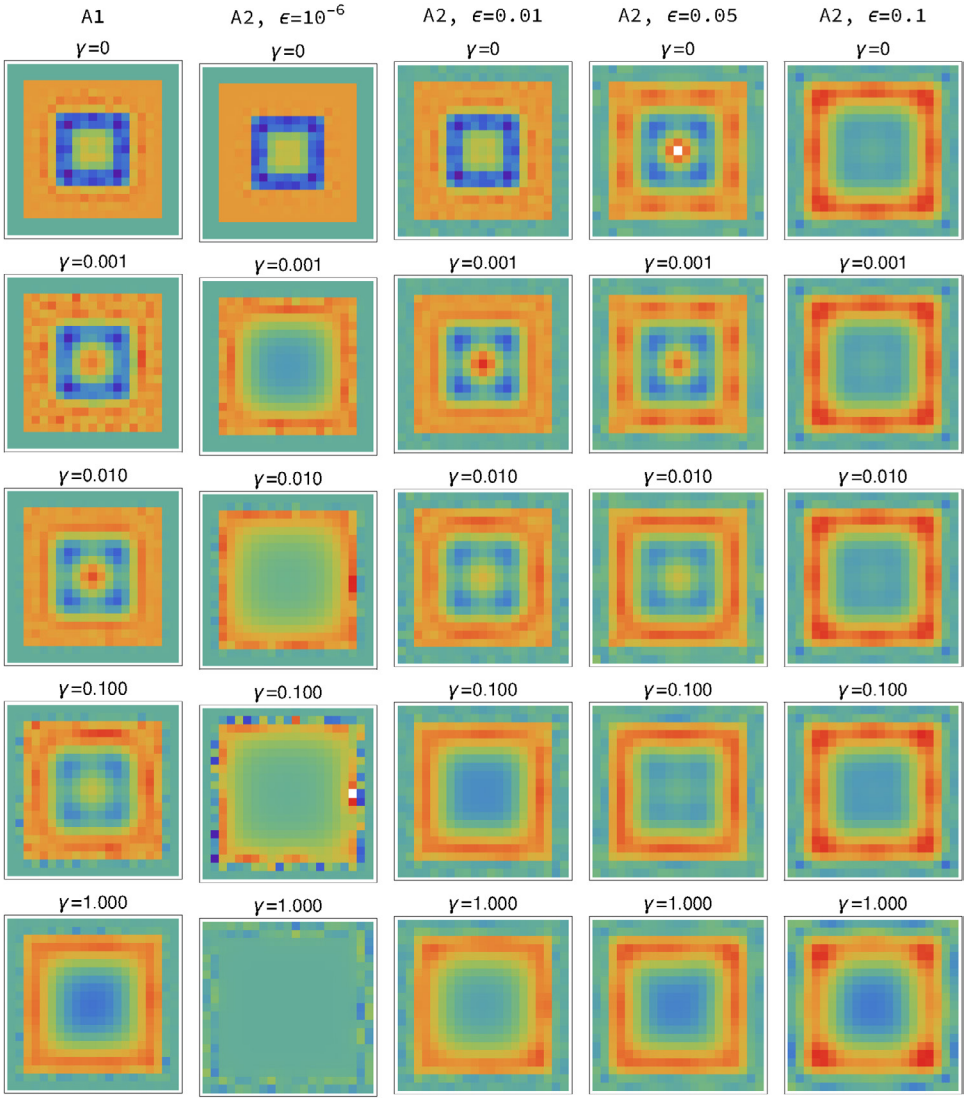

Figure 11. Reconstructions for the large target and the same measurement geometry as in section 5.2 obtained by algorithm (left column) and by algorithm 2 with different values of the cut-off parameter ϵ. Reconstructions were obtained at different levels of noise as quantified by the parameter γ. Areas where the reconstruction exceeds the maximum value used for the color map (which represents values from -2 to 3) are shown as white.

to the computations of the first column. Nevertheless, the results are approximately the same at $\gamma = 1$. Thus, we have obtained a useful data reduction.

We note that, when we transition from $\epsilon = 0.1$ (third column) to $\epsilon = 0.5$ (forth column), the problem becomes underdetermined. This is why the quality of the ideal data reconstruction in the forth column is visibly worse than that in the third column. However, in the presence of noise, this quality deterioration is not as pronounced.

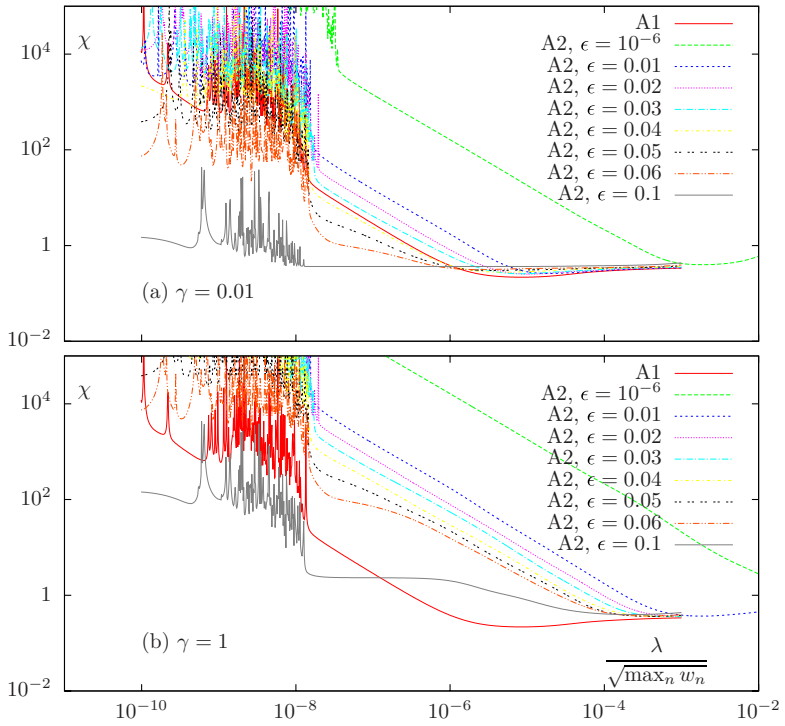

Figure 12. L^2 norm of the reconstruction error χ as functions of the regularization parameter λ for the large target discretized on a $21 \times 21 \times 21$ cubic grid reconstructed using algorithm 1 (A1) and algorithm 2 (A2) for the noise level $\gamma = 0.01$ (a) and $\gamma = 1$ (b) and for different values of the cut-off regularization parameter ϵ, as labeled.

The data reduction described above can be further illustrated if we consider the curves $\chi(\lambda)$ (χ is the L^2 norm of the reconstruction error and λ is the Tikhonov regularization parameter) for different levels of noise γ and different cut-off parameters ϵ. These curves are plotted in figure 12. Consider first the case $\gamma = 0.01$. It can be seen that the error obtained in algorithm 2 with $\epsilon = 10^{-6}$ is much larger than that for algorithm 1 for almost all values of λ considered. However, as we increase ϵ, the error curves for the two algorithms become similar. The curve with $\epsilon = 0.03$ reaches almost the same minimum value as the A1 curve. A qualitatively similar behavior is observed at the noise level $\gamma = 1$. The curves with $\epsilon = 0.05$, 0.06 and 0.1 reach a minimum that is only slightly above the minimum of the A1 curve. The related difference in the reconstruction quality mainly affects the precision of determining the boundaries of the outer shell, while the inner shell can not be reconstructed by either method at this noise level (see figure 11).

We conclude this section with a few notes. First, the reconstruction for the under-determined problem can be obtained by using algorithm 1 as described above but this is not the most computationally-efficient approach. Alternatively, one can start with (15) and compute the singular-value decomposition of the linear operator in this equation directly or by diagonalizing $W_{(\mu_1\nu_1)(\mu_2\nu_2)} = \sum_n \left(g^{A*}_{\mu_1 n} g^{A}_{\mu_2 n}\right) \left(f^{B}_{\nu_1 n} f^{B*}_{\nu_2 n}\right)$. We have implemented both algorithms and, as expected, they yield the same numerical result. The second algorithm however is much faster since its computational bottleneck has complexity $O((M_A M B)^3)$ rather than $O(N_v^3)$, and we have $M_A M_B < N_v$ in the underdetermined case. This is how the computational speed-up

mentioned above was achieved. We emphasize that this speed-up is not limited by the number of data points or the number of unknowns (voxels in our case). Rather, the limiting factor is the number of voxels that can be reconstructed reliably given a particular level of noise.

Secondly, the above numerical illustrations utilize a specific expression for the noise term. In particular, the noise is additive and affects all data points in a statistically-independent manner. A multiplicative noise of the form $\Phi \to \Phi(1+n)$, where n is a statistically-independent random variable is also frequently used in the literature. Using this form of noise can yield very different results from what is shown above.

Finally, we note that the algorithm with diagonal preconditioning can be further modified to moderate the effects of noise on the reconstructions while still achieving a useful data reduction. To this end, one can define the diagonal operator in (13) as

$$D_{(\mu\nu),(\mu'\nu')} = \delta_{(\mu\nu),(\mu'\nu')} \begin{cases} 1, & \sigma^A_\mu,\ \sigma^B_\nu > \epsilon \\ 0, & \sigma^A_\mu,\ \sigma^B_\nu \leqslant \epsilon \end{cases} \tag{23}$$

that is, with a cut-off but without scaling. In this way, we can discard unreliable equations efficiently but without amplifying the effects of noise. The involved algebra and the numerical algorithm are only trivially modified in this case, with no increases in the computational complexity. Since this paper is primarily focused on fast numerical inversion of generic equations of the form (2), we do not investigate this approach further, but we expect it to be efficient when reconstructing with noisy data.

7. Summary and discussion

The purpose of this paper is not to study image reconstruction in diffuse optical tomography but rather to present a general method of efficiently computing the pseudo-inverse of the linear operator in inverse scattering problems. The underlying idea of the method is very simple and provides a dramatic improvement in computational efficiency. We note that the idea has appeared in the literature before [11]. Here we have explained it in detail, discussed two algorithmic implementations of the method (one of them with preconditioning) and illustrated its power with numerical simulations involving both ideal and noisy data.

It is important to note that the methods described here are purely algebraic and not dependent on any symmetry of the sample. In our previous work [28, 29] and references therein, we developed fast inversion methods that require some symmetry to hold, i.e. translational symmetry in the case of an infinite slab. These methods require large imaging windows and are inapplicable to the geometry with only a few discrete symmetries that has been considered in the above numerical examples.

In fact, it can be shown that the methods of [28, 29] are related to the method described here, when the SVD decomposition of A and B can be computed analytically. The mathematical connection between the two methods will be explained elsewhere.

We finally note that the efficiency of the proposed methods make them suitable for use in iterative solvers of nonlinear inverse problems, wherein linearized inversion is required at each step. This includes, for example, the Newton-type methods.

Acknowledgments

John Schotland acknowledges support from the NSF Grant DMS-1619907. This work was performed while Vadim Markel was a Visiting Scholar in the Department of Mathematics at

the University of Michigan. We are grateful to one of the referees for drawing our attention to the work of Lev-Ari [11].

ORCID iDs

Vadim A Markel https://orcid.org/0000-0002-9748-6865
John C Schotland https://orcid.org/0000-0003-0545-1962

References

[1] Golub G and Kahan W 1965 *J. SIAM Numer. Anal. B.* **2** 205–24
[2] Golub G H and Reinsch C 1970 *Numer. Math.* **2** 403–20
[3] Chan T F 1982 *ACM Trans. Math. Soft.* **8** 72–83
[4] Grosser B and Lang B 2003 *Linear Algebr. Appl.* **358** 45–70
[5] Dhillon I S and Parlett B N 2004 *SIAM J. Matrix Anal. Appl.* **25** 858–99
[6] Dhillon I S and Parlett B N 2004 *Linear Algebr. Appl.* **387** 1–28
[7] Wang Z M, Panasyuk G Y, Markel V A and Schotland J C 2005 *Opt. Lett.* **30** 3338–40
[8] Konecky S D, Panasyuk G Y, Lee K, Markel V, Yodh A and Schotland J C 2008 *Opt. Exp.* **16** 5048–60
[9] Ban H Y, Busch D R, Pathak S, Moscatelli F A, Machida M, Schotland J C, Markel V A and Yodh A G 2013 *J. Biomed. Opt.* **18** 026016
[10] Levinson H W and Markel V A 2016 *Phys. Rev.* E **94** 043317
[11] Lev-Ari H 2005 *Commun. Inf. Syst.* **5** 123–30
[12] Khatri C G and Mitra S K 1976 *SIAM J. Appl. Math.* **31** 579–85
[13] Magnus J R 1983 *Linear Multilinear Algebra* **14** 67–88
[14] Hua D 1990 *Linear Algebr. Appl.* **131** 1–7
[15] Tian Y 2003 *Linear Multilinear Algebra* **51** 111–25
[16] Zhang X 2004 *Appl. Math.* E-Notes **4** 40–7
[17] Hou J J, Peng Z Y and Zhang X L 2006 *Numer. Algorithms* **42** 181–92
[18] Cvetkovic-Iliic D S 2006 *Comput. Math. Appl.* **51** 897–902
[19] Cvetkovic-Iliic D S 2008 *J. Aust. Math. Soc.* **84** 63–72
[20] Dehghan M and Hajarian M 2008 *Comput. Math. Appl.* **56** 3246–60
[21] Aman M 2014 *Iran. J. Math. Sci. Inf.* **9** 31–42
[22] Yuan Y and Zuo K 2015 *Appl. Math. Comput.* 905–12
[23] Mohseni Moghadam M, Rivaz A, Tajaddini A and Saberi Movahed F 2015 *Bull. Iran. Math. Soc.* **41** 981–1001
[24] Sheng X and Sun W 2017 *Comput. Math. Appl.* **74** 597–604
[25] Liu S and Trenkler G 2008 *Int. J. Inf. Syst. Sci.* **4** 160
[26] Arridge S R and Schotland J C 2009 *Inverse Problems* **25** 123010
[27] Shihab Uddin K M, Mostafa A, Anastatsio M and Zhu Q 2017 *Biomed. Opt. Exp.* **8** 5437–49
[28] Markel V A and Schotland J C 2002 *Appl. Phys. Lett.* **81** 1180–2
[29] Markel V A and Schotland J C 2004 *Phys. Rev.* E **70** 056616
[30] Epstein C L and Schotland J 2008 *SIAM Rev.* **50** 504–20
[31] Markel V A and Schotland J C 2004 *Opt. Lett.* **29** 2019–21
[32] Markel V A and Schotland J C 2005 *Phys. Med. Biol.* **50** 2351–64
[33] Markel V A, Wang Z M and Schotland J C 2005 *Photonic Applications in Biosensing and Imaging* vol 5969 (Bellingham, WA: SPIE) p 59691B