SIAM J. Sci. Comput.
Vol. 27, No. 6, pp. 2140–2166

RECYCLING SUBSPACE INFORMATION FOR DIFFUSE OPTICAL TOMOGRAPHY*

MISHA E. KILMER† AND ERIC DE STURLER‡

Abstract. We discuss the efficient solution of a long sequence of slowly varying linear systems arising in computations for diffuse optical tomographic imaging.

The reconstruction of three-dimensional absorption and scattering information by matching computed solutions from a parameterized model to measured data leads to a nonlinear least squares problem that we solve using the Gauss–Newton method with a line search. This algorithm requires the solution of a long sequence of linear systems. Each choice of parameters in the nonlinear least squares algorithm results in a different matrix describing the optical properties of the medium. These matrices change slowly from one step to the next, but may change significantly over many steps. For each matrix we must solve a set of linear systems with multiple shifts and multiple right-hand sides.

For this problem, we derive strategies for recycling Krylov subspace information that exploit properties of the application and the nonlinear optimization algorithm to significantly reduce the total number of iterations over all linear systems. Furthermore, we introduce variants of GCRO that exploit symmetry and that allow simultaneous solution of multiple shifted systems using a single Krylov subspace in combination with recycling. Although we focus on a particular application and optimization algorithm, our approach is applicable generally to problems where sequences of linear systems must be solved. This may guide other researchers to exploit the opportunities of tunable solvers.

We provide results for two sets of numerical experiments to demonstrate the effectiveness of the resulting method.

Key words. Krylov subspace, GCRO, recycle, MINRES, eigenvalue, invariant subspace

AMS subject classifications. 65F10, 65N22

DOI. 10.1137/040610271

1. Introduction.

In diffuse optical tomography (DOT), data is obtained by transmitting near-infrared light into a highly absorbing and scattering medium and then recording the photon flux. The goal is to use the diffuse optical data measured on the surface to reconstruct three-dimensional (3D) images of the absorption and "reduced scattering" functions in the medium. In the case of breast tissue imaging, differences in the absorption and scattering may indicate the presence of a tumor or other anomaly.

The *forward problem* is the determination of synthetic data (photon flux) for given absorption and scattering functions from some mathematical model. A number of mathematical models have been proposed in the literature [2]. We focus on the frequency-domain diffusion model in which the data is a nonlinear function of the absorption and scattering functions. In order to solve the imaging problem—the determination of the absorption and reduced scattering functions—one must solve many instances of the forward problem. This fact implies a huge computational

*Received by the editors June 21, 2004; accepted for publication (in revised form) July 8, 2005; published electronically February 21, 2006. This work was supported by NSF grants 0208548 and 0139968 and CenSSIS (the Center for Subsurface Sensing and Imaging Systems) under the ERC Program of the NSF (EEC-9986821), grant DOE LLNL B341494, and grants NSF DMR-9976550 and NSF DMR-0325939.

http://www.siam.org/journals/sisc/27-6/61027.html

†Department of Mathematics, Tufts University, 503 Boston Ave., Medford, MA, 02155 (misha.kilmer@tufts.edu).

‡The Thomas M. Siebel Center for Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin, Urbana, IL 61801-2302 (sturler@cs.uiuc.edu).

bottleneck for the imaging problem. The goal of this paper is to discuss techniques for reducing the computational complexity of forward solves, thereby greatly improving the execution time of the nonlinear imaging problem.

In [26], we recycle the subspaces that are generated automatically by the methods used. In the present paper, we analyze the application, its solutions, and the nonlinear optimization algorithm to derive strategies for selecting subspaces for recycling that enhance convergence even further. In particular, we explore which information to keep and which to save as the damped Gauss–Newton (GN) method progresses. Furthermore, we present a variant of GCRO [10] that exploits the symmetry of the matrix and that can solve multiple shifted complex systems simultaneously with a single Krylov subspace in combination with recycling. The symmetry means that there is no need to restart for a single linear system to save on storage.

We consider the solution of a sequence of linear systems of the form

$$(A(p_j) + i\gamma I)x_{s,\omega}^{(j)} = b_s \tag{1.1}$$

that arise in a 3D imaging algorithm for DOT. Here, $i = \sqrt{-1}$, γ is a positive constant that depends on the frequency, ω, and the vector p_j denotes the vector of parameters that define the diffusion and absorption in tissue at the jth step of a GN iteration to solve the nonlinear least squares problem (defined below) for an optimal set of parameters. The sparse, symmetric positive definite matrix A depends on the parameters p_j and arises from the discretization of the diffusion-absorption equation with mixed boundary conditions and some further matrix manipulation (see section 3). Therefore, Krylov iterative solvers such as MINRES [25, 4] are good candidates for solving these linear systems.

We observe that in our application the matrices $A(p_j)$ vary slowly from one GN step or line search step to the next, but they change significantly over multiple GN steps. In addition, we need to solve for multiple (complex) shifts and multiple right-hand sides for each matrix. In order to solve this problem in the most efficient way we must exploit all these features. For each of these features separately, suggestions have been made to reduce the overall cost (for example, see [19, 18, 22] for solving for a group of matrices that differ by a constant times the identity and [17, 31, 9, 26, 21] for solving for multiple right-hand sides). However, the various methods have not been combined to address all these features at once, and this is not a trivial issue.

The problem of solving a sequence of systems where the matrix changes slowly is the most complicated feature to exploit. In [26] we propose to *recycle* from one linear system to the next the Krylov subspaces that solvers like GCRO [10], GCROT [11], and GMRESDR [24] retain to improve the convergence for a single linear system. GMRESDR cannot recycle a subspace for a subsequent linear system, as it requires a Krylov space to work with; therefore, we introduced the variant GCRODR [26].

Other approaches have been proposed as well. If all matrices in a set of symmetric positive definite linear systems are pairwise close to each other and all right-hand sides are available simultaneously, the methods proposed by Chan and Ng [7] can be used. However, this is not the case for our application. For the Hermitian positive definite case, Rey and Risler have proposed to reduce the effective condition number by augmenting the search space either with all converged Ritz vectors from previous CG iterations or with all previously generated Krylov spaces for a few iterations to implicitly approximate dominant eigenvectors [27, 28]. Clearly, both in memory and floating point operations these methods are very expensive. Moreover, in their approach, although they use CG, full recurrences are required. Finally, both approaches lack the possibility to gradually improve the recycled space and to adapt it for a sequence of linear systems that change gradually but substantially over many steps. For this reason, Rey and Risler also make the assumption that all matrices remain close. Fischer has proposed to project right-hand sides from subsequent time steps onto the space of previous right-hand sides and to solve only for the remainder [16]. This leads to a better starting guess if the right-hand sides are correlated, but his algorithm does not maintain orthogonality to this subspace and, in general, does not improve the rate of convergence.

Multigrid methods might provide an alternative for the methods proposed here, as they have lower complexity than Krylov methods for diffusion type problems, and several papers have discussed how to deal effectively with large jumps in the diffusion and absorption coefficients [1, 5, 34]. However, it is not clear that the convergence rate of multigrid methods can be significantly improved for a sequence of matrices and/or right-hand sides when subsequent solutions differ significantly, as is accomplished by the methods proposed here (see also [26]). Adaptations of multigrid methods for sequences of problems mainly involve the judicious choice on which grids to update an old solution [5]. This approach is not suitable if the new solution differs significantly from the previous solution because it does not improve convergence in general. It just reduces the work in updating a slightly changed solution. In addition, our algorithms compute solutions for multiple shifts more or less for free; in a multigrid setting this is not possible. Finally, we can improve the convergence of the algorithm over multiple right-hand sides, which is not straightforward with multigrid methods.

As mentioned above, our problem includes the solution of a small set of right-hand sides for each matrix (with fixed p_j and γ). As was shown in [9, 26], subspace recycling is quite effective for this problem. However, other approaches or variations have been successful as well, in particular block methods [17, 31] and seed methods [8, 21]. Subspace recycling is the easiest to implement, as it can solve the right-hand sides one by one as individual linear systems, simply recycling the selected search space [26]. This avoids changes to the program to deal with varying block sizes and deflation if the vectors inside a block become dependent.

Finally, we also want to solve for multiple shifts using a single Krylov space. This is not complicated in itself, as the Krylov space for a shifted problem equals the Krylov space for the original problem. However, it is not easily combined with recycling Krylov subspaces, because the images of recycled spaces under matrices with different shifts are not the same. We derive an extension to the GCRO method in section 5 to deal with this problem.

The paper is organized as follows. Section 2 gives some background on GCRO and subspace recycling. In section 3, we give background information for the imaging problem in DOT, and we derive the sequence of linear systems of the form (1.1) that we wish to solve. In section 4, we discuss characteristics of the system that favor the use of recycling. We describe our algorithm in section 5, and give numerical results in section 6. Conclusions and future work are the subject of section 7.

2. Recycling Krylov subspaces.

The ideas we exploit in this paper find their origin in attempts to improve the convergence of restarted and truncated Krylov subspace methods for a single, nonsymmetric, linear system. Restarting GMRES [30] may lead to poor convergence and even stagnation. Therefore, recent research has focused on truncated methods that improve convergence by retaining a selected subspace when they restart [3, 10, 11, 24, 26, 29]. We refer to this subspace as the *recycled* subspace. These methods aim to maintain convergence close to that of full

GMRES while significantly reducing memory and CPU costs. A taxonomy of popular choices is given in [15], and approaches to convergence theory for GMRES that are relevant here can be found in [32, 35].

An important choice for the present paper is to recycle an approximate invariant subspace [23, 24]. An alternative method that adapts the recycled space based on its effectiveness is given in [11]. In [10] the updates to the solution (residual) are recycled. This is also proposed in [3], but the recycled space is used slightly differently. In this paper, we exploit knowledge of the physics underlying DOT, the solutions in a general sense, and the nonlinear optimization algorithm to combine the strategies from [10, 24, 16, 26] for a sequence of linear systems.

The GCRO method provides a general mechanism for Krylov methods to include arbitrary additional subspaces in the search space. We explain briefly how GCRO combines the recycled space with a newly generated Krylov subspace to obtain an optimal approximation over the sum of these spaces. To prepare for the extension to the simultaneous solution of a set of linear systems with matrices that differ only by a constant times the identity using a single Krylov subspace, we present the basic steps from [10] in a slightly different way.

Consider the linear system $Ax = b$, where $A \in \mathbb{R}^{N \times N}$ and $b \in \mathbb{R}^N$. Assume we have the matrix $U \in \mathbb{R}^{N \times n_c}$, such that $AU = C$ and $C^T C = I$. There are no other restrictions on the matrix U. The approximate solution in range(U) that minimizes the 2-norm of the residual yields the residual $r = b - CC^T b$ that is orthogonal to range(C). If this solution is not adequate, we expand the subspace as follows [10]. Let $v_1 = (I - CC^T)b/\|(I - CC^T)b\|_2$. We use an Arnoldi recurrence with $(I - CC^T)A$ and v_1 to generate the recurrence relation

$$(2.1) \qquad \begin{aligned} (I - CC^T)AV_m &= V_{m+1}\underline{H}_m \Leftrightarrow \\ AV_m &= CC^T AV_m + V_{m+1}\underline{H}_m. \end{aligned}$$

Next, we compute the approximate solution in range($[V_m \ U]$) that minimizes the 2-norm of the residual, $\|b - A(V_m y + Uz)\|_2$, as follows:

$$(2.2) \qquad \begin{aligned} &\min_{y,z} \left\| b - A[U \ V_m] \begin{bmatrix} z \\ y \end{bmatrix} \right\|_2 \\ = &\min_{y,z} \left\| b - [C \ V_{m+1}] \begin{bmatrix} I & C^T AV_m \\ 0 & \underline{H}_m \end{bmatrix} \begin{bmatrix} z \\ y \end{bmatrix} \right\|_2 \\ = &\min_{y,z} \left\| \begin{bmatrix} C^T b \\ \xi e_1 \end{bmatrix} - \begin{bmatrix} I & C^T AV_m \\ 0 & \underline{H}_m \end{bmatrix} \begin{bmatrix} z \\ y \end{bmatrix} \right\|_2, \end{aligned}$$

where e_1 denotes the first Cartesian basis vector in $\mathbb{R}^{m+1}$ and $\xi = \|(I - CC^T)b\|_2$. The minimization in (2.2) corresponds to a small least squares problem, whose solution requires only the QR decomposition of the submatrix $\underline{H}_m$ using Givens rotations as in [30]. We will show below that this approach is also extended easily to a set of matrices $A + i\gamma I$. In this case, we need to deal with the problem that range($(A + i\gamma I)U$) $\not\subset$ range(C), and of course range($(A + i\gamma I)U$) depends on γ.

If we have found a matrix U that speeds up the convergence significantly, we can recycle this matrix for the next right-hand side (with the same A and γ) and possibly improve it. This does not require any changes to the algorithm, and it allows the algorithm to learn as it computes the solutions for subsequent right-hand sides which spaces are best to recycle. This improves the convergence for multiple right-hand sides and a constant matrix without requiring extra storage. This algorithm can also be extended to a block method [37].

Now, consider a sequence of systems with coefficient matrices $A^{(j)}$. If we have found a matrix U that speeds up convergence significantly for $A^{(j)}$, and $A^{(j+1)} - A^{(j)}$ is small, it makes sense to reuse the matrix U, possibly extended with other search directions, for the linear systems with $A^{(j+1)}$. In this case, we need to update C to reflect the new operator. In many cases this can be done very cheaply. This process is not complicated, and we refer to [26] for details.

3. The DOT imaging application.

In this section, we introduce the image reconstruction problem for DOT. Here, we derive the systems of the form (1.1) that must be solved at each step of the nonlinear reconstruction algorithm.

3.1. The forward and inverse problems.

We assume that the region to be imaged is a box with a limited number of N_s sources on the top and a limited number of N_d detectors on either the top or bottom or both. We use the diffusion model [2] for photon flux/fluence $\phi_{s,\omega}(r)$ given input $f_s(r)$:

$$\begin{aligned} -\nabla D(r)\nabla\phi_{s,\omega}(r) + \mu_a(r)\phi_{s,\omega}(r) + i\frac{\omega}{\nu}\phi_{s,\omega}(r) &= f_s(r) \\ \text{for } r = (x,y,z) \text{ and } -a < x < a,\ -b < y < b,\ 0 < z &< c, \\ \phi_{s,\omega}(r) = 0 \quad \text{if } 0 \le z \le c \text{ and either } x = -a,\ x = a,\ y = -b,\ \text{or } y &= b, \\ .25\phi_{s,\omega}(r) + \frac{D(r)}{2}\frac{\partial\phi_{s,\omega}(r)}{\partial\eta} = 0 \text{ for } z = 0 \text{ or } z &= c. \end{aligned}$$

Here, $D(r)$ denotes the diffusion, which is related to the "reduced scattering" function $\mu_s'(r)$ by $D = 1/(3\mu_s'(r))$, and $\mu_a(r)$ denotes absorption [2]. We have used $i = \sqrt{-1}$, while ω represents the frequency modulation of light, and ν is the speed of light in the medium. The integer subscript s indicates the model with a single source at a known position. The function $f_s(r)$ is the source and $\phi_{s,\omega}(r)$ is the photon flux/fluence due to the source at frequency ω, given the functions $\mu_a(r)$ and $D(r)$. The function $\phi_{s,\omega}(r)$ is complex-valued if ω is nonzero. Knowing the source and the functions $\mu_a(r), D(r)$, we can compute the corresponding $\phi_s(r)$ everywhere, in particular at the detectors (i.e., at a subset of gridpoints where $z = 0$ or $z = c$).

We discretize the PDE using finite differences [6] on a uniform grid in such a way as to achieve second-order accuracy away from the boundary. The meshwidth in each direction is h centimeters. We use first-order forward or backward differences, as appropriate, on the boundary. The unknowns become $\phi_{s,\omega}(x_l, y_j, z_k)$ for $l = 1 : N_x, j = 1 : N_y, k = 1 : N_z$. We will order the unknowns so that $\phi_{s,\omega}$ values at points on the top of the box come first (i.e., let $k = 1$ and loop over all l, j), then the $\phi_{s,\omega}$ corresponding to points on the bottom (i.e., let $k = N_z$ and loop over all l, j), followed by the rest of the values by ordering in increasing l, then j, then k. We denote the corresponding vector with entries $\phi_{s,\omega}(x_l, y_j, z_k)$ as $\boldsymbol{\phi_{s,\omega}}$. Likewise, we denote the vector with entries $\mu_a(x_l, y_j, z_k)$ as $\boldsymbol{\mu_a}$. The vector $\boldsymbol{D}$ is comprised from sampling D at both whole and half integer gridpoints, because of the particular discretization we are using.

The corresponding matrix equation, after multiplication by h^2, has the following block structure:

$$\begin{bmatrix} B & D_1 \\ D_2 & (F + i\frac{\omega h^2}{\nu}I) \end{bmatrix} \begin{bmatrix} w_{s,\omega} \\ x_{s,\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_s}^{(1)} \\ \boldsymbol{f_s}^{(2)} \end{bmatrix}, \tag{3.1}$$

where $w_{s,\omega}$ and $x_{s,\omega}$ denote the discretization of $\phi_{s,\omega}(r)$ on the boundary and at internal points, respectively. The measured data *due to source s* **predicted** by this forward model is a subsampled version of the subvector $w_{s,\omega}$, which we call $\boldsymbol{\psi_{s,\omega}}$. It is important to note that in our application $\boldsymbol{f_s}^{(2)} = 0$ and $\boldsymbol{f_s}^{(1)} = h^2 e_i$, where e_i is the ith Cartesian unit vector and the position of the 1 corresponds to the location of the source.

Let $y_{s,\omega}$ denote the data subvector **measured** at all the detectors for a fixed source s and frequency ω, and let $W_{s,\omega}$ denote the diagonal weighting matrix whose diagonal entries are the inverses of the standard deviations of the noise in the data. Recall that p is a vector of parameters that describe the diffusion and absorption at all points in the region of interest. We will briefly discuss the choice of p below and refer the interested reader to [20] for more details. The 3D imaging problem then becomes one of finding the parameters that minimize the difference between the data predicted by the model and the measured data. That is, we wish to solve

$$\min_p \sum_{s,\omega} \|W_{s,\omega}(y_{s,\omega} - \boldsymbol{\psi_{s,\omega}}(p))\|_2^2 \equiv \min_p \|W(\mathbf{y} - \boldsymbol{\psi}(p))\|_2^2, \tag{3.2}$$

where $\mathbf{y}$ denotes the vector obtained by stacking the subvectors $y_{s,\omega}$, $\boldsymbol{\psi}(p)$ is the vector obtained by stacking the subvectors $\boldsymbol{\psi_{s,\omega}}(p)$, and W is the block matrix whose diagonal blocks are the matrices $W_{s,\omega}$. Let $\epsilon(p)$ denote the residual vector $W(\mathbf{y} - \boldsymbol{\psi}(p))$. We use the following damped GN iteration to solve this nonlinear least squares problem [14], where $J(p)$ denotes the Jacobian of the residual, evaluated at p:

1. compute $\epsilon(p_k)$, $J(p_k)$,
2. solve $J(p_k)^T J(p_k) s_k = -J(p_k)^T \epsilon(p_k)$,
3. $p_{k+1} = p_k + \lambda_k s_k$,

with λ_k chosen using a backtracking line search [14].

If the number of parameters used to define diffusion and absorption is small, the Jacobian will only have a small number of columns, and therefore step 2, solving for the search direction, is not computationally intensive. However, to compute $\epsilon(p_k)$ and to compute the entries in the Jacobian using an adjoint-type approach (called a "co-state" method in [36]) requires solutions of the matrix equation in (3.1) for every source and every frequency. Furthermore, each line search step requires evaluation of residual and Jacobian at a new point. Therefore, the rest of this paper is devoted to analyzing the systems (3.1) and methods for computing their solutions efficiently.

3.2. The matrix revisited. Here, we describe properties of the matrices and submatrices involved in solving for each $\boldsymbol{\phi_{s,\omega}}$. First, regarding the blocks in (3.1),

- B is an invertible diagonal matrix,
- D_1 has at most one nonzero per row, and these occur only in the first $N_x N_y$ and last $N_x N_y$ columns,
- D_2, although it has different entries, has the same sparsity pattern as D_1^T.

A MATLAB sparsity plot of the matrix in (3.1) is given in Figure 1 to give the reader a visual interpretation of the structure just mentioned.

To solve systems involving this matrix, we consider the decomposition $G = LU$, where L is block unit lower triangular, U is block upper triangular, and G represents the block matrix in (3.1). It can readily be shown that

$$G = \begin{bmatrix} I & 0 \\ D_2 B^{-1} & I \end{bmatrix} \begin{bmatrix} B & D_1 \\ 0 & F - D_2 B^{-1} D_1 + i\gamma I \end{bmatrix},$$

where $\gamma = h^2\omega/\nu$. Hence, the system $G \begin{bmatrix} w_{s,\omega} \\ x_{s,\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_s}^{(1)} \\ 0 \end{bmatrix}$ can be solved as follows.

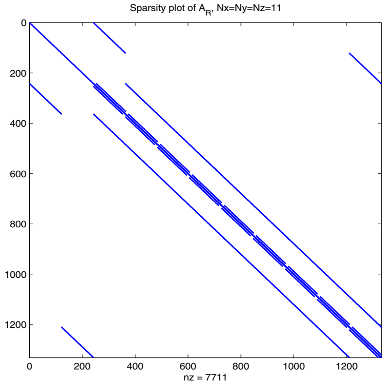

FIG. 1. *Sparsity plot of matrix G.*

Step 1. Solve the equation $L\begin{bmatrix} a_s \\ b_s \end{bmatrix} = R\begin{bmatrix} \boldsymbol{f_s}^{(1)} \\ 0 \end{bmatrix}$:

(a) $a_s = \boldsymbol{f_s}^{(1)}$,

(b) $b_s = -D_2(B^{-1}a_s)$.

Step 2. Solve the equation $U\begin{bmatrix} w_{s,\omega} \\ x_{s,\omega} \end{bmatrix} = \begin{bmatrix} a_s \\ b_s \end{bmatrix}$:

(a) solve $(F - D_2B^{-1}D_1 + i\gamma I)x_{s,\omega} = b_s$,

(b) $w_{s,\omega} = B^{-1}(a_s - D_1x_{s,\omega})$.

Multiplying with B^{-1} can be done very cheaply because B is diagonal, and D_1 and D_2 have only $(2N_xN_y)$ nonzero entries each. The computationally intensive part of this procedure is Step 2(a). Note that the system in Step 2(a) is exactly the system in (1.1).

We conclude this section with the proof that $F-D_2B^{-1}D_1$ is symmetric and positive definite. The matrix F is symmetric and positive definite because it corresponds to the finite difference discretization of the operator $-\nabla D(r)\nabla I+\mu_a(r)I$ at the internal points on the box assuming zero boundary conditions. B is a diagonal matrix of size $2N_xN_y \times 2N_xN_y$ with entries $.25h^2 + \frac{h}{2}\boldsymbol{D}_{l,j,m}$ with $m = 1$ or $m = N_z$. Due to the lexicographical ordering of the internal nodes, D_1 is $2N_xN_y \times (N_xN_y(N_z-2))$ with only one nonzero per row and D_2 is $(N_xN_y(N_z-2))\times 2N_xN_y$ with only one nonzero per column. The nonzero entries in D_1 are of the form $-\frac{h}{2}\boldsymbol{D}_{l,j,m}$ with $m = 1$ or $m = N_z$. The nonzero entries in D_2 are $-\boldsymbol{D}_{l,j,\frac{3}{2}}$ or $-\boldsymbol{D}_{l,j,N_z-\frac{1}{2}}$. Therefore, $D_2B^{-1}D_1$ is a diagonal matrix with nonzero (positive) entries only in the first and last N_xN_y positions. We now use the previous two facts to prove the following theorem.

THEOREM 3.1. *The matrix $F - D_2B^{-1}D_1$ is symmetric and positive definite.*

Proof. Symmetry follows from the symmetry of F and $D_2B^{-1}D_1$. Matrices $F-D_2B^{-1}D_1$ and F differ only in the first and last N_xN_y components on the diagonal. Since the nonzeros in the second matrix are strictly positive, the first and last N_xN_y diagonal entries of $F - D_2B^{-1}D_1$ are smaller than the corresponding diagonal entries of F. Therefore, it is sufficient to consider the Gershgorin disks corresponding to these rows. Consider the first N_xN_y rows (the argument for the last N_xN_y is analogous). From Gershgorin's theorem we observe

$$\left(-\frac{\boldsymbol{D}_{i,j,1.5}\boldsymbol{D}_{i,j,1}}{0.5h + \boldsymbol{D}_{i,j,1}} + \boldsymbol{D}_{i,j,1.5}\right) + h^2\boldsymbol{\mu_a}_{i,j,1} \leq \lambda,$$

where λ is an eigenvalue. Since the first term on the left in parentheses is positive, together with lower bounds from all other Gershgorin disks, it follows that the eigenvalues are greater than zero. □

4. Summary of system properties. In the previous section, we observed that solving the forward problem efficiently boils down to solving the systems in Step 2(a) efficiently. The remainder of this paper is therefore devoted to this cause.

In this section, we discuss those system properties that can be exploited to develop efficient recycling Krylov methods for solving the systems in the previous section. We repeat the form of these systems here for convenience:

$$(\underbrace{F - D_2 B^{-1} D_1}_{A^{(j)}} + i\gamma I) x_{s,\omega}^{(j)} = b_s. \tag{4.1}$$

The superscript is used to denote dependence on the parameter vector, p_j. To simplify notation, we use $x_s^{(j)}$ for the case when $\omega = 0$.

The parametric model. As in [20], we use a piecewise continuous model for both the diffusion and absorption. In particular, we have

$$\boldsymbol{D} = \alpha_1 \Xi_1 + (\mathbf{1} - \Xi_1) B_1 \beta_1 \qquad \text{and} \qquad \boldsymbol{\mu_a} = \alpha_2 \Xi_2 + (\mathbf{1} - \Xi_2) B_2 \beta_2.$$

The vectors Ξ_i are discrete characteristic functions, having a "1" in a position corresponding to an anomaly and a 0 otherwise. The matrices B_1, B_2 are known and contain "basis" vectors while β_1 and β_2 are the unknown expansion coefficients. In a piecewise constant model, for instance, B_1 and B_2 would be vectors of all ones and the β_i would give the background value of diffusion and absorption, respectively. A more realistic model, however, accounts for the fact that tissue is not homogeneous, and in this case B_1, B_2 would correspond to vectorized "images" of a lumpy background. The vectors Ξ_i are unknown, but we assume that anomalies are modeled by ellipsoids. In this case, the entries in the vectors are determined by the parameters specifying center location, rotation, and axis lengths. Therefore, the list of unknown parameters includes $\alpha_i, \beta_i, i = 1, 2$, and up to 6 length-3 vectors specifying the locations of the 2 ellipsoids. For more details, see [20].

Diffusion, absorption, and matrix updates. Typical values for μ_a in our application range from .005 to .3 cm^{-1} whereas typical values for D range from 1/6 to 1/45 cm^{-1}. Usually, we have or can obtain good approximations to the average background values of diffusion and absorption and use these as starting guesses for the GN iteration [20]. As the GN iteration progresses, we begin to localize and characterize anomalous regions of absorption and diffusion whereas the background values become well resolved early on. This is due to the fact that the anomalies are so small relative to the size of the background that the data contains primarily information about the average background values. This means that during a single line search or when moving from one GN step to the next, $A^{(j+1)} = A^{(j)} + E_1 + E_2$, where $\|E_1\|$ is small (corresponding to a slight change in the background parameters) and E_2 has small relative rank (corresponding to a change in the shape of the object and values inside the object).

Eigenvalues and invariant subspaces. We are interested most in the smallest eigenvalues of the $A^{(j)}$. It appears that, for our examples, the matrices $A^{(j)}$ have a number of small eigenvalues occurring in clusters that remain disjoint from one system to the next, even if the eigenvalues themselves differ from one matrix to the next (see Figure 4, for example). This suggests that the corresponding invariant subspaces for subsequent matrices remain close. We discuss this further in the next subsection.

Similarity among right-hand sides. Recall that $\boldsymbol{f_s}$ has only one nonzero coefficient in, say, position m_s. It follows that $b_s^{(j)} = -\frac{(D_2)_{m_s,m_s}h^2}{B_{m_s,m_s}} e_{m_s}$, and so the $b_s^{(j)}$ are the same up to a scalar factor. The latter has no influence on the solver. Therefore, in the remaining discussion we study the convergence of iterative methods applied to the systems

$$(A^{(j)} + i\gamma I)x_{s,\omega}^{(j)} = e_{m_s}. \tag{4.2}$$

Similarity among solutions. From (4.2), it is easy to see that the solutions to any pair of (nonshifted) systems, say systems k and j, during the GN iteration are related by

$$x_s^{(k)} = (A^{(k)})^{-1}A^{(j)}x_s^{(j)}.$$

Thus, the solutions do not change much as long as the matrices $A^{(j)}$ and $A^{(k)}$ remain close, especially with respect to the smallest eigenvalues and corresponding eigenvectors; see also the following discussion on smooth solutions. The reason why we expect, in particular, the smooth components of the matrices to remain close is discussed in section 4.1. This expectation is borne out in the numerical experiments. It is important to note that the indices j and k need not be consecutive, nor must they refer to systems from within the same line search. See section 5 for details.

Smooth solutions. The vector e_{m_s} is comprised mostly of high frequency Fourier components. Furthermore, the eigenvectors corresponding to the smallest eigenvalues of $A^{(j)}$ represent low frequency (smooth) modes while the eigenvectors corresponding to the largest eigenvalues represent high frequency modes. The matrix $(A^{(j)})^{-1}$ corresponds to an integration operator that acts as a blurring operator. Therefore, we expect $x_s^{(j)}$ to be well represented in terms of the eigenvectors of $A^{(j)}$ that correspond to the smallest eigenvalues.

Multiple frequencies. The Krylov vectors generated to solve $A^{(j)}x_s^{(j)} = e_{m_s}$ also span the Krylov subspace generated by the shifted matrix $(A^{(j)} + i\gamma I)$ and e_{m_s} (see [22] and the references cited therein). When subspace recycling is used, however, it is not straightforward to produce solutions to the complex shifted systems. We discuss the extension of the algorithm to this case in section 5.4.

4.1. Invariant subspaces. In deciding whether or not it is worthwhile to recycle an approximate invariant subspace corresponding to small eigenvalues obtained from previous matrices, we must explore the relationships among these invariant subspaces of a sequence of matrices. In the previous discussion, we noted that from experiments it appears that the smallest eigenvalues of the $A^{(j)}$ do not change much and remain in more or less disjoint clusters. For a small enough perturbation $E = A^{(k)} - A^{(j)}$, this shows that the corresponding invariant subspaces from these matrices must remain close. Unfortunately, $\|E\|$ is too large to assume this without considering further details. However, recall the observation above that $E = E_1 + E_2$, where E_1 corresponds to very small changes in the background parameters and E_2 corresponds to a small rank update describing the shape of the object. From this we conjecture that most of the changes in the matrix correspond to the high frequency components and larger eigenvalues. Given this conjecture, we now show under which conditions the invariant subspaces corresponding to the smallest eigenvalues remain about the same even if the corresponding eigenvalues are not very well separated from the remaining eigenvalues.

To simplify notation, we remove all subscript and superscript notation and deal specifically with a symmetric and positive definite matrix A and a corresponding symmetric perturbation E.

Let A be a symmetric positive definite matrix, and let A have the eigendecomposition

$$A = [Q_1\ Q_2\ Q_3]\,\mathrm{diag}(\Lambda_1, \Lambda_2, \Lambda_3)\,[Q_1\ Q_2\ Q_3]^T, \tag{4.3}$$

where $Q = [Q_1\ Q_2\ Q_3]$ is an orthogonal matrix, $\Lambda_1 = \mathrm{diag}(\lambda_1^{(1)}, \ldots, \lambda_{k1}^{(1)})$, and Λ_2 and Λ_3 are defined analogously. Furthermore,

$$\lambda_1^{(1)} \leq \cdots \leq \lambda_{k1}^{(1)} < \lambda_1^{(2)} \leq \cdots \leq \lambda_{k2}^{(2)} < \lambda_1^{(3)} \leq \cdots \leq \lambda_{k3}^{(3)}.$$

In relation to our algorithm, Q_1 corresponds to the recycled invariant subspace (or a subspace of that), but Q_2 and Q_3 can be chosen to fit the theorem. Now we consider the changes in the invariant subspace range(Q_1) and the eigenvalues $\lambda_i^{(1)}$ under a symmetric perturbation E of A, where E is not small, but the projection of E onto the subspace range$([Q_1\ Q_2])$ is small, say $\|E[Q_1\ Q_2]\|_F \leq \varepsilon$, and $\|EQ_3\|_F = \eta \approx \|E\|_F$. We also assume that $\|E\|_F$ is small relative to $\mathrm{sep}(\Lambda_1, \Lambda_3) = \lambda_1^{(3)} - \lambda_{k1}^{(1)}$ and that ε is small relative to $\mathrm{sep}(\Lambda_1, \Lambda_2) = \lambda_1^{(2)} - \lambda_{k1}^{(1)}$. However, we do not need to assume that $\mathrm{sep}(\Lambda_1, \Lambda_2)$ is large. We now prove that the matrix $A + E$ has an invariant subspace range$(\hat{Q}_1)$ such that the canonical angles between range(Q_1) and range$(\hat{Q}_1)$ are small. This result shows that an invariant subspace whose associated eigenvalues are not well separated from the remaining eigenvalues is still insensitive to perturbations that are concentrated in an invariant subspace whose eigenvalues are sufficiently far removed.

We define the following notation. For two matrices $Y, Z \in \mathbb{R}^{N\times m}$, where $N \geq m$, $\Theta(\mathrm{range}(Y), \mathrm{range}(Z))$ denotes the diagonal matrix with the canonical angles between range(Z) and range(Y) as coefficients, and $\theta_1(\mathrm{range}(Y), \mathrm{range}(Z))$ denotes the largest canonical angle between range(Z) and range(Y). We use $\mathcal{L}(A)$ to denote the set of eigenvalues of A, and $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ to denote $\max \mathcal{L}(A)$ and $\min \mathcal{L}(A)$, respectively.

Furthermore, we assume that

$$\delta \equiv \min(\lambda_1^{(2)} - \varepsilon, \lambda_1^{(3)} - \eta) - 2\varepsilon - (\lambda_{k1}^{(1)} + \varepsilon) \gg \varepsilon, \tag{4.4}$$

$$\hat{\delta} = \delta\left(1 - \frac{2\varepsilon^2}{\delta^2}\right) + \lambda_{k1}^{(1)} + \varepsilon, \tag{4.5}$$

and as a consequence of (4.4) that $\delta > 2\varepsilon$.

THEOREM 4.1. *Let A be s.p.d. and have the eigendecomposition given in* (4.3), *and let E, ε, η, δ, and $\hat{\delta}$ be defined as above. Then, there exists a matrix $\hat{Q}_1$ conforming to Q_1 such that* range$(\hat{Q}_1)$ *is a simple invariant subspace of $A + E$, and*

$$\tan \theta_1\left(\mathrm{range}(Q_1), \mathrm{range}(\hat{Q}_1)\right) \leq \frac{\varepsilon}{\hat{\delta}}.$$

Furthermore, for each eigenvalue $\hat{\lambda}_j^{(1)}$ of $A+E$ corresponding to the invariant subspace range$(\hat{Q}_1)$, *there exists a value $\lambda_i^{(1)}$ such that*

$$|\hat{\lambda}_j^{(1)} - \lambda_i^{(1)}| \leq \varepsilon + \frac{2\varepsilon^2}{\delta},$$

and in particular,

$$\lambda_{\max}(\,\hat{Q}_1^T(A+E)\hat{Q}_1\,) \leq \lambda_{k1}^{(1)} + \varepsilon + \frac{2\varepsilon^2}{\delta}.$$

Proof. We consider the perturbation E, such that

$$Q^T(A+E)Q = \begin{pmatrix} \Lambda_1 + \mathcal{E}_{11} & 0 & 0 \\ 0 & \Lambda_2 + \mathcal{E}_{22} & \mathcal{E}_{32}^T \\ 0 & \mathcal{E}_{32} & \Lambda_3 + \mathcal{E}_{33} \end{pmatrix} + \begin{pmatrix} 0 & \mathcal{E}_{21}^T & \mathcal{E}_{31}^T \\ \mathcal{E}_{21} & 0 & 0 \\ \mathcal{E}_{31} & 0 & 0 \end{pmatrix}. \tag{4.6}$$

By the assumptions above we also have

$$\left\| \begin{pmatrix} \mathcal{E}_{21} \\ \mathcal{E}_{31} \end{pmatrix} \right\|_F \leq \varepsilon,$$

$\|\mathcal{E}_{11}\|_F \leq \varepsilon$, $\|\mathcal{E}_{22}\|_F \leq \varepsilon$, and $\|\mathcal{E}_{33}\|_F \leq \eta$. From (4.6) we see that

$$\begin{aligned} L_1 \;&\equiv\; Q_1^T(A+E)Q_1 = \Lambda_1 + \mathcal{E}_{11}, \\ L_{23} \;&\equiv\; [Q_2\; Q_3]^T(A+E)[Q_2\; Q_3] = \begin{pmatrix} \Lambda_2 + \mathcal{E}_{22} & \mathcal{E}_{32}^T \\ \mathcal{E}_{32} & \Lambda_3 + \mathcal{E}_{33} \end{pmatrix}. \end{aligned}$$

From [33, Corollary IV.3.4] it follows that

$$\lambda_{\max}(\Lambda_1 + \mathcal{E}_{11}) \leq \lambda_{k1}^{(1)} + \|\mathcal{E}_{11}\| \leq \lambda_{k1}^{(1)} + \varepsilon, \tag{4.7}$$

$$\lambda_{\min}(\Lambda_2 + \mathcal{E}_{22}) \geq \lambda_1^{(2)} - \|\mathcal{E}_{22}\| \geq \lambda_1^{(2)} - \varepsilon, \tag{4.8}$$

$$\lambda_{\min}(\Lambda_3 + \mathcal{E}_{33}) \geq \lambda_1^{(3)} - \|\mathcal{E}_{33}\| \geq \lambda_1^{(3)} - \eta. \tag{4.9}$$

Now we can apply [33, Corollary IV.3.4] once more to obtain

$$\lambda_{\min}(L_{23}) \geq \min(\lambda_1^{(2)} - \varepsilon, \lambda_1^{(3)} - \eta) - 2\varepsilon. \tag{4.10}$$

From (4.7)–(4.10) we have $\mathrm{sep}(L_1, L_{23}) > \delta$. Furthermore, let $R \equiv (A+E)Q_1 - Q_1L_1 = Q_2\mathcal{E}_{21} + Q_3\mathcal{E}_{31}$. Then, from symmetry it follows that $Q_1^T(A+E) - L_1Q_1^T = R^T$, and we have $\|R\|_F = \|R^T\|_F \leq \varepsilon$. Finally, we have

$$\frac{\|R\|_F\|R^T\|_F}{\mathrm{sep}(L_1, L_{23})^2} \leq \frac{\varepsilon^2}{\delta^2} < \frac{1}{4},$$

and by [33, Corollary V.2.2] we know there exists a matrix $\hat{Q}_1$ conforming to Q_1 such that $\mathrm{range}(\hat{Q}_1)$ is a simple invariant subspace of $A+E$, and

$$\tan\theta_1\left(\mathrm{range}(Q_1), \mathrm{range}(\hat{Q}_1)\right) \;\leq\; \|\tan\Theta\left(\mathrm{range}(Q_1), \mathrm{range}(\hat{Q}_1)\right)\|_F \;\leq\; 2\frac{\varepsilon}{\delta}. \tag{4.11}$$

Since A and $A+E$ are symmetric and we have established the existence of $\hat{Q}_1$, we can obtain a better bound using [33, Theorem V.3.10]. This theorem conforms nicely to our special case. However, we need to establish the minimal distance between eigenvalues of L_1 and the eigenvalues of $\hat{L}_{23} \equiv \hat{Q}_{23}^T(A+E)\hat{Q}_{23}$, where $\mathrm{range}(\hat{Q}_{23}) = \mathrm{range}(\hat{Q}_1)^\perp$ and $\hat{Q}_{23}$ has orthonormal columns. From [33, Theorem V.2.1] specialized to the symmetric case, we know there exists a matrix P, such that $\|P\|_F \leq 2\varepsilon/\delta$ and

$\mathcal{L}(\,\hat{L}_{23}\,) = \mathcal{L}(L_{23} - P[\mathcal{E}_{21}^T\ \mathcal{E}_{31}^T])$. From $\|P[\mathcal{E}_{21}^T\ \mathcal{E}_{31}^T]\| \leq 2\varepsilon^2/\delta$ and [33, Corollary IV.3.4], we have the following bound (with $\hat{\delta}$ defined in (4.5)):

$$\lambda_{\min}(\hat{L}_{23}) \geq \min(\lambda_1^{(2)} - \varepsilon, \lambda_1^{(3)} - \eta) - 2\varepsilon - \frac{2\varepsilon^2}{\delta} = \hat{\delta}. \tag{4.12}$$

Finally, we obtain from [33, Theorem V.3.10]

$$\tan\theta_1(\text{range}(Q_1), \text{range}(\hat{Q}_1)) \leq \|\tan\Theta(\text{range}(Q_1), \text{range}(\hat{Q}_1))\|_F \leq \frac{\varepsilon}{\hat{\delta}},$$

which is better than (4.11) by about a factor 2 or more. Analogously to (4.12) we observe from [33, Theorem V.3.10] that for each eigenvalue $\hat{\lambda}_j^{(1)}$ of $\hat{Q}_1^T(A+E)\hat{Q}_1$ there exists a number, $\lambda_i^{(1)}$, such that

$$|\hat{\lambda}_j^{(1)} - \lambda_i^{(1)}| \leq \varepsilon + \frac{2\varepsilon^2}{\delta}.$$

In particular, this gives

$$\lambda_{\max}(\,\hat{Q}_1^T(A+E)\hat{Q}_1\,) \leq \lambda_{k1}^{(1)} + \varepsilon + \frac{2\varepsilon^2}{\delta}. \qquad \square$$

Our numerical experiments confirm our conjecture except for a few GN steps, when the new matrix is quite far from previous ones. In those cases, the projection of E on the (smooth) invariant subspace corresponding to the smallest eigenvalues is still small, but $\|E\|$ is large enough that the invariant subspaces corresponding to large eigenvalues might perturb those corresponding to the smallest ones. Because of the problem size we cannot check the projection of E on invariant subspaces corresponding to medium or larger eigenvalues. However, Figures 4 and 9 include examples with relatively large canonical angles between invariant subspaces associated with the smallest eigenvalues for systems at the start of a line search.

5. Algorithm.

Let us outline some aspects related to the optimization algorithm. We combine the GN algorithm with a line search. As we will see in the numerical results section, the final steps in each line search tend to be small. Therefore, the solutions obtained toward the end of each line search are not too different for a few GN iterations. However, over many GN steps they tend to differ more significantly; see Figure 2. An obvious way to exploit this is by using the solution of the previous step as a starting guess. However, this will work better or worse depending on whether the previous step was the first step of a line search, towards the start or the end of a line search, and so on. In practice, it is not easy to choose the best among several previous solutions as the best choice is governed by the progression of the algorithm (it is not necessarily the latest one). However, recycling a small subspace of previous solutions for the search space relieves this problem. In fact, we can vary this additional subspace depending on whether we are at the start of a line search or near the end and on how large the line search parameter is. We can also update this additional search space as we go. Strategies based on this approach turn out to be very effective; we will give more details later in this section when we discuss the application and the optimization algorithm. The idea to use previous solutions was also proposed in [16], although in [16] they were only used to provide a better initial guess.

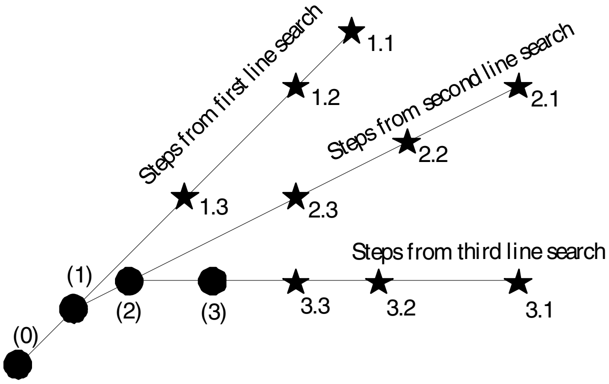
Steps from first line search
Steps from second line search
Steps from third line search
1.1
1.2
1.3
2.1
2.2
2.3
3.1
3.2
3.3
(0)
(1)
(2)
(3)

FIG. 2. *Conceptual overview of three complete steps of the GN with line search algorithm. The black circle marked* (0) *indicates the start of the overall procedure. The other black circles denote the end of a line search (the end of the first line search is marked* (1)*, etc.). The black stars denote the first and intermediate steps from each line search as indicated (the first step of line search* 1 *is indicated by* (1.1)*, etc.).*

The recycling strategy that we propose is based on the observations in the previous section. In particular, we use the proximity of certain invariant subspaces and tune which old solutions to recycle to the phase of the GN algorithm with line search.

5.1. Recycled GCRO for DOT.

Consider the sequence of systems

$$A^{(j)}x_1^{(j)} = e_{m_1},$$

for the source $s = 1$. We will always recycle, in a matrix U, the most recent solution that occurred at the end of a line search. We may recycle previous solutions from within a particular line search. We do not know whether a line search is complete until after the corresponding system has been solved. However, before we solve a system, we can test whether we are "close" to the end of a line search. We assume this is the case if the relative residual norm of the current system for the solution at the end of the previous line search is below a certain threshold. This information helps to keep the smallest possible recycling space. It may not be necessary to keep a large approximate invariant subspace to reduce the initial residual over that part of the spectrum when we keep solutions in the recycle space that have the effect of reducing the residual over the same part of spectrum anyway (see the discussion in section 5.2).

To keep the notation as simple as possible, we begin by reviewing a basic GCRO algorithm that implements the recycling strategy mentioned above; then we discuss modifications that allow us to solve for multiple right-hand sides and multiple frequencies. Below, the MATLAB notation $[C, R] = qr(C, 0)$ is used to mean that C is overwritten with the orthogonal matrix from its compact QR factorization and R is the upper triangular factor.

ALGORITHM 1.

```
1. Solve $A^{(1)}x_1^{(1)} = e_{m_1}$ with MINRES. Set $x^{(curr)} = x_1^{(1)}, x^{(beg)} = []$.
2. Form approximate eigenvectors for $A^{(1)}$ from information generated from the
   MINRES run. Save these eigenvectors in the matrix $W$.
```

```
3. For $j = 2, \ldots$
   (a) If $(\|A^{(j)}x^{(curr)} - e_{m_1}\|/\|e_{m_1}\| \leq tol)$ and not at beginning of LS
           $U = [W_{ind}, x^{(curr)}]$,
       Else
           $U = [W, x^{(curr)}, x^{(beg)}]$.
   (b) $A^{(j)}U = C$, $[C, R] = qr(C, 0)$, set[1] $U = UR^{-1}$.
   (c) Compute $x_0 = U(C^T e_{m_1})$. (Ensures $x_0$ is optimal in the sense that the
       residual is minimized over all solutions in range$(U)$.)
   (d) Set $P = (I - CC^T)$. Compute $r_0 = Pb$.
   (e) Solve $PA^{(j)}Pv = r_0$ by MINRES.[2]
   (f) Update $x, r$.
   (g) If (at end of LS), $x^{(curr)} = x_1^{(j)}$.
   (h) If (at beginning of LS), $x^{(beg)} = x_1^{(j)}$.
```

The index j is the index for the system within the GN algorithm, and therefore the loop on j terminates when GN converges for (3.2). Here, W_{ind} indicates that we may wish to keep fewer approximate eigenvectors according to the discussion preceding the algorithm (see also the discussion in section 5.2 and numerical example 1). The logic for tailoring the choice of the columns of U to the GN process comes from the discussion in the previous section. First, based on our observations for this application, we expect the approximate invariant subspace corresponding to the smallest eigenvalues of the first matrix to be close to an invariant subspace of other matrices in the GN sequence corresponding to the smallest eigenvalues. If the GN iteration is converging, we expect that at the end of two consecutive line searches the corresponding matrices will be related since the purpose of the line search is to produce a parameter update vector that allows the GN process to converge. Likewise, if we are at the beginning of a line search sequence, matrices from that sequence should be related, too. An analysis of the effect of these choices on the convergence of the MINRES steps is provided in the next subsection.

5.2. Algorithm analysis.

Given that U always contains the matrix W, which we assume is a good approximation for the invariant subspace corresponding to the smallest eigenvalues for all the systems, we expect the systems in 3(e) to converge as if the smallest eigenvalues have been deflated from $A^{(j)}$. This is the case, in fact, as Theorem 4.1 from [26] (which closely follows Theorem 2.1 in [32]) implies.

Next, we consider the effect of keeping a previous solution, $x^{(curr)}$, in U. Let the current matrix U have n_c columns. For ease of discussion, we may assume that the first column of U is $x^{(curr)}$, since this does not change the orthogonal projector. Now, at the end of 3(b), we observe that $A^{(j)}x^{(curr)} = \rho_1 c_1$, where c_1 is the first column of C and ρ_1 an appropriate scaling. However, $x^{(curr)} = (A^{(k)})^{-1}e_{m_s}$ for some previous index k, and hence $A^{(j)}(A^{(k)})^{-1}e_{m_s} = \rho_1 c_1$. We have $A^{(j)}(A^{(k)})^{-1} = I + \tilde{E}$ for error matrix $\tilde{E} = -E_1(A^{(k)})^{-1} - E_2(A^{(k)})^{-1}$, with E_1, E_2 as the small norm and relatively small rank terms defined previously. Therefore,

$$\tilde{v} \equiv (I - c_1 c_1^T)e_{m_s} = e_{m_s} - c_1 c_1^T(\rho_1 c_1 - \tilde{E}e_{m_s}) = e_{m_s} - \rho_1 c_1 + c_1(c_1^T \tilde{E} e_{m_s}).$$

[1] From an implementation standpoint, we would not perform the matrix-matrix product UR^{-1}. Rather, we would keep R around, and when we needed to perform UR^{-1} times a vector, as in the next step, we would do backward substitution with R followed by multiplication with U. However, this explicit notation simplifies the introduction of the algorithm.

[2] Since all the Krylov vectors in the MINRES iteration are orthogonal to C, we do not need to apply P on the right. This choice of notation serves only to illustrate the symmetry of the matrix operator and therefore the applicability of MINRES.

Then the initial residual is $r_0 = (I - C_{n_c-1}C_{n_c-1}^T)\tilde{v}$, where C_{n_c-1} contains the remaining columns in C. Thus we have

$$\begin{aligned} r_0 &= (I - C_{n_c-1}C_{n_c-1}^T)(e_{m_s} - \rho_1 c_1) + (c_1^T \tilde{E} e_{m_s})c_1 \\ &= (I - C_{n_c-1}C_{n_c-1}^T)(A^{(k)} - A^{(j)})x^{(curr)} + (c_1^T \tilde{E} e_{m_s})c_1. \end{aligned}$$

The solution $x^{(curr)}$ is smooth and, as seen earlier, $A^{(k)} - A^{(j)}$ is small over the invariant subspace of $A^{(j)}$ corresponding to the smallest eigenvalues. Thus, the vector $e_{m_s} - \rho_1 c_1$ should already be small in norm. Since C_{n_c-1} contains approximate eigenvectors corresponding to the smallest (and smoothest) eigenvectors, then clearly r_0 will be even smaller in norm, particularly if $(e_{m_s} - \rho_1 c_1)$ lies predominately in the direction of these eigenvectors anyway. Furthermore, we have $c_1^T \tilde{E} e_{m_s} = -c_1^T E_1 x^{(curr)} - c_1^T E_2 x^{(curr)}$. The first term in this expression is small in norm. The second term must also be small, since E_2 lies predominantly in the direction corresponding to larger magnitude eigenvalues whereas $x^{(curr)}$ is smooth. In summary, not only do we observe that the norm of r_0 is small, but the smoothness properties ensure it is smallest in directions corresponding to the larger magnitude eigenvalues and that it has been reduced in directions corresponding to the smallest magnitude eigenvalues. This is demonstrated clearly by the reduction of the norm of the initial residual as the GN iteration progresses in Figures 7 and 12 and also by the eigenvector components of $r_0^{(j)}$ given in Figure 5. Hence, corrections to the residual occur primarily over the remaining subspace, which accounts for the convergence behavior observed in our numerical examples.

5.3. Multiple right-hand sides. The next consideration is the solution for multiple right-hand sides. First, we expand step 2 in Algorithm 1 to accommodate the right-hand sides. Once the first system for the first source has been solved, we have approximate eigenvector information. We can use this eigenvector information when we solve for the remaining right-hand sides, and we can also collect additional eigenvector information. Thus, the new step 2 becomes the following:

- Form approximate eigenvectors for $A^{(1)}$ from information generated from the MINRES run. Save these eigenvectors in the matrix W.
- **For** $s = 2, \ldots, N_s$
  - Set $U = W$.
  - Perform steps 3b–3e of Algorithm 1 (only updates of U and C).
  - Update $x_s^{(1)}$ with this information.
  - Add columns to W if desired.

Additional (approximate) invariant subspace information might be computed either from range(V_m) or from range($[U\,V_m]$). Then, to solve for $x_s^{(j)}$ for the remaining sources, we insert a loop over the remaining sources just after step 3 which would run over all s and contain steps 3(a)–3(h).

In this way, all the right-hand sides use a different last column (or two) in the U matrix, depending on the source. However, for every right-hand side, the first several columns of the U matrix are comprised of the (final) approximate eigenvector matrix W. Therefore, step 3(b) is cheaper for sources $s \geq 2$, since all but the last column (or two) of C were determined during the run on the first source.

5.4. Multiple shifts γ. To account for multiple frequencies, in step 3(e), we have $v_1 = (I - CC)^T e_{m_s} / \|(I - CC^T)e_{m_s}\|_2$ and the matrix recurrence (cf. (2.1))

$$AV_m = CB_m + V_{m+1}\underline{T}_m, \tag{5.1}$$

where $B_m = C^T A V_m$ and the leading $m \times m$ submatrix of $\underline{T}_m$ is symmetric and positive definite. From this recurrence, we obtain

$$\begin{aligned}
(A + i\gamma I)[V_m \; U] &= [V_{m+1}(\underline{T}_m + i\gamma \underline{I}_m) + CB_m \;\; C + i\gamma U] \\
&= [V_{m+1} \;\; C \;\; U] \begin{bmatrix} \underline{T}_m + i\gamma \underline{I}_m & 0 \\ B_m & I \\ 0 & i\gamma I \end{bmatrix} \\
&= [V_{m+1} \;\; C \;\; \hat{U}] \begin{bmatrix} I & 0 & V_{m+1}^T \hat{U} \\ 0 & I & C^T U \\ 0 & 0 & N \end{bmatrix} \begin{bmatrix} \underline{T}_m + i\gamma \underline{I}_m & 0 \\ B_m & I \\ 0 & i\gamma I \end{bmatrix},
\end{aligned}$$

where the last step involves the reduced QR decomposition of $[V_{m+1} \; C \; U]$, so that $[V_{m+1} \; C \; \hat{U}]$ has orthonormal columns. Notice that $[V_{m+1} \; C]$ is already an orthogonal matrix. Restricting our approximate solutions to be in range$(U) \oplus$ range(V_m), we need to solve the small least squares problem

$$\begin{bmatrix} I & 0 & V_{m+1}^T U \\ 0 & I & C^T U \\ 0 & 0 & N \end{bmatrix} \begin{bmatrix} \underline{T}_m + \gamma \underline{I}_m & 0 \\ B_m & I \\ 0 & \gamma I \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \approx \begin{bmatrix} \xi e_1 \\ C^T e_{m_s} \\ 0 \end{bmatrix}$$

for every choice of γ, and put $x_{s,\omega}^{(j)} = V_m y + Uz$. This is equivalent to solving

$$\begin{bmatrix} \underline{T}_m + i\gamma \underline{I}_m & i\gamma V_{m+1}^T U \\ B_m & I + i\gamma C^T U \\ 0 & i\gamma N \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \approx \begin{bmatrix} \xi e_1 \\ C^T e_{m_s} \\ 0 \end{bmatrix}. \tag{5.2}$$

This equation leads to an algorithm for updating the solutions to systems in which $\gamma \neq 0$. After step 3(e) in Algorithm 1, we insert the following piece of code:

```
ALGORITHM 2a: based on solving (5.2)
For each $\gamma$
  • Solve (5.2) for $\left[\begin{smallmatrix} y \\ z \end{smallmatrix}\right]$.
  • Set $x_{s,\omega}^{(j)} = V_m y + Uz$.
```

Care must be taken in solving (5.2) if a subspace of range(U) is very close to an invariant subspace of $A^{(j)}$. In particular, if range(U) is an invariant subspace of $A^{(j)}$, then $U = C\Delta$, for some Δ. In this case, the least squares problems simplify considerably as both $V_{m+1}^T U = 0$ and $N = 0$. Hence, we need to solve only the following least squares problems (compare to (2.2)):

$$\begin{bmatrix} I + i\gamma\Delta & B_m \\ 0 & \underline{T}_m + i\gamma \underline{I}_m \end{bmatrix} \begin{bmatrix} z \\ y \end{bmatrix} \approx \begin{bmatrix} C^T e_{m_s} \\ \xi e_1 \end{bmatrix}. \tag{5.3}$$

Having to solve only (5.3) is the ideal situation because this problem can be solved as two separate minimization problems: the first to compute y, the second to compute z. However, we do not need y explicitly, only the products $V_m y$ and $B_m y = C^T A V_m y$, which can be obtained from short-term recurrences using a MINRES-type approach without keeping the vectors V_m around. The short-term recurrences are similar to the MINRES algorithm and therefore their derivation is left out for brevity.

```
ALGORITHM 2b: based on solving (5.3)
For each $\gamma$
  • Determine $V_m y$ using short-term recurrences.
```

- Determine z.
- Set $x_{s,\omega}^{(j)} = V_m y + Uz$.

On the other hand, if the columns of U do not span an invariant subspace of $A^{(j)}$, we should solve (5.2). Unfortunately, there exists no short-term recurrence for $V_m y$ in this case, and thus we are forced to save the V_m in order to form $x_{s,\omega}^{(j)}$. However, for a fixed source, only one set of vectors V_m needs to be saved from which solutions at all other frequencies can be computed. Moreover, the generation of V_m itself requires only short-term recurrences, so no extra work is required.

We advocate Algorithm 2b when storage is at a premium and/or when we know that U is a good approximate invariant subspace, and Algorithm 2a otherwise.

We make one further modification to Algorithm 1 to solve these shifted systems simultaneously with the nonshifted systems. We change step 3(a) to append to U also the column imag$(x_{s,\omega}^{(curr)})$ and possibly imag$(x_{s,\omega}^{(beg)})$ depending on which part of the conditional statement is executed. The justification is as follows. Let $E = A^{(j)} - A^{(k)}$, and therefore $E = A_\omega^{(j)} - A_\omega^{(k)}$ as well. One can show that $A^{(j)}(x_s^{(j)} - x_s^{(k)}) = Ex_s^{(k)}$ and $A_\omega^{(j)}(x_{s,\omega}^{(j)} - x_{s,\omega}^{(k)}) = Ex_{s,\omega}^{(k)}$. Given the eigendecomposition $A^{(j)} = Q\Lambda Q^T$, we have

$$x_{s,\omega}^{(k)} = (A^{(k)} + i\gamma I)^{-1} e_{m_s} = \sum_\ell q_\ell \frac{q_\ell^T e_{m_s}}{\lambda_\ell + i\gamma},$$

where γ is reasonably small compared to the (real) eigenvalues λ_ℓ. Hence, the real part of $x_{s,\omega}^{(k)}$ is close to $x_s^{(k)}$, and the difference between $Ex_s^{(k)}$ and $Ex_{s,\omega}^{(k)}$ is primarily due to the (small) imaginary part of $x_{s,\omega}^{(k)}$. Since we look for solutions to the complex system in range$(U) \oplus$ range(V_m), it makes sense to include imag$(x_{s,\omega}^{(k)})$ in U.

We expect either algorithm to perform sufficiently well when $\gamma = h^2\omega/\nu$ is not too large in an absolute sense. In our application, γ will typically be less than or equal to $O(10^{-4})$, and we will not be solving the system for very many values of γ. This means that if MINRES had been applied directly to the shifted system, the number of iterations required for convergence would be about the same as the number of iterations required for the unshifted system. However, if γ is much larger, neither algorithm will necessarily produce solutions with a small relative residual norm. This stems from the fact that in solving the projected problem, we may leave out directions from the Krylov subspace in which the solutions to the complex systems have large components. On the other hand, if γ is very large relative to the smallest eigenvalue of the $A^{(j)}$ (which is not the case in our application), the eigenvalues of the shifted matrix will be well clustered. In the latter case, MINRES applied to this shifted problem would converge quickly, and recycling may not be necessary.

5.5. Computational issues. In this subsection, we briefly outline the computational complexity and storage issues associated with Algorithm 1 and its variants. We focus on overhead introduced by our algorithmic changes compared with MINRES. This overhead is offset by the reduction in iterations. Clearly, due to storage limitations we cannot have too many columns for U if the dimension of the problem is large. In addition, we must balance the cost of orthogonalizing the Krylov vectors against the columns of C with the reduction in the total number of iterations times the cost per iteration. If the cost of the matrix-vector product and preconditioning is large we can invest more time in orthogonalizations to reduce the number of iterations. In general, the idea is to find U with as few columns as possible such that the number of MINRES iterations is significantly reduced.

Computational cost. We consider only the complexity of computational steps that are linear in the size of the system matrix, i.e., $O(N)$. The costs of the various computations on small matrices, such as the solution of least squares problems and the computation of (selected) eigenvalues and eigenvectors, are negligible.

The main overhead introduced by our method compared with standard MINRES is the orthogonalization of each Krylov vector against the matrix C, indicated by the left multiplication by P in step 3(e) in Algorithm 1. Let the number of columns in C be n_c. Then the orthogonalizations introduce a cost of $4n_cN$ floating point operations (flops) per iteration. For each linear system, the initialization of x_0 and r_0 in 3(c) and 3(d) takes about $4n_cN$ flops, and the correction to the solution with the term $U(B_my)$, $x = (x_0 + V_my) - UB_my$ (see [10]), after the MINRES iteration, costs an additional $2n_cN$ flops. For each new matrix (but not for each right-hand side) we have the cost of n_c matrix-vector products and the cost of the QR decomposition of an N-by-n_c matrix (about $2n_c^2N$ flops) in 3(b). If k indicates the average number of nonzeros per row in the matrix, the cost for the matrix-vector products from which C is formed is $2n_ckN$ flops. However, these matrix-vector products have been accounted for in Figures 6 and 11, whereas they do not correspond to iterations. In that respect they actually represent a reduction in computational cost compared with MINRES. Note that, in general, the multiplication of the matrix times a block of vectors takes less time than multiplying each vector separately. Since we vary one or two columns in C for each right-hand side there are an additional one or two matrix-vector products per right-hand side.

Finally, we use the information generated by the solver for the first system matrix and a few right-hand sides to approximate invariant subspace information. If we approximate eigenvectors over range(V_m), the only $O(N)$ cost arises from constructing approximate eigenvectors for $A^{(1)}$ from the eigenvectors of the $m \times m$ leading principal submatrix of $\underline{T}_m$ (5.1) generated in the MINRES iteration. This costs approximately $2mN$ flops per approximate eigenvector, where m is the length of the MINRES run at the time the invariant subspace is computed. This may vary and need not be at the end of the MINRES run. Approximating eigenvectors over range($[V_m\ U]$) takes approximately $2(m + n_c)N$ flops per eigenvector plus $2(n_c^2 + mn_c)N$ flops to set up the generalized eigenvalue problem. Since the invariant subspace associated with the smallest eigenvalues changes only little in our application and high accuracy is not needed, this is done typically only for the first few eigenvectors for one of the matrices early in the GN iteration. Thus, these costs are incurred only once.

For Algorithm 2a, the only additional cost of $O(N)$ arises from the orthogonalization of the U matrix against the matrices V_{m+1} and C and an additional $2n_cN$ flops for updating the residual with the $\hat{U}$ component. For the solution there is no change in cost with respect to Algorithm 1. The additional cost of the orthogonalization is $2n_c(2m+3n_c)N$. Note that there is significant overlap with the computation of eigenvectors. Thus, in steps where eigenvectors are computed or updated the additional cost is negligible. Finally, note that the significant reduction in the number of iterations per right-hand side helps in reducing the actual overhead. For Algorithm 2b no additional costs arise. The matrix Δ is available from the computation of the invariant subspace.

Parallel implementation. Realistic problems from DOT are very large and therefore well suited to straightforward parallelization by distributing each matrix row-wise following a good domain partitioning. The special structure of the matrix (see section 4) allows this in spite of the factored form of the matrix. Therefore, the parallel implementation of the algorithms outlined above can follow the discussion in, for example, [13, 12]. In addition, we could parallelize over the right-hand sides. This can be done with minor variations of the algorithms presented here. Since we can solve for multiple shifts with a single Krylov subspace, there is little benefit in parallelizing over the shifts.

6. Numerical results.

In this section, we give the results of our proposed algorithm on two sequences of matrices generated from two different runs of the parametric nonlinear inversion scheme outlined previously. In the first experiment, a piecewise constant model for diffusion and absorption was used. In the second, we used a nonhomogenous model for the diffusion and absorption in the background whereas the diffusion and absorption inside the anomalies were constant. There were 16 sources and 32 detectors. In both experiments, the nonlinear inversion scheme was run using only data for the 0 frequency case; however, we apply our algorithm to both the 0 frequency case and shifted systems at 5 MHz in order to test our algorithm. The region was discretized[3] into $31 \times 31 \times 21$ voxels of volume h^3, with $h = .2$ cm. The sizes of the matrices in both experiments were 18,259 × 18,259. The sources and detectors were located in a 3 cm × 3 cm plane over the center of the grid. The starting guesses for the ellipsoids describing the anomaly were the largest possible ellipsoids fitting in the 3 cm × 3 cm × 4 cm region under the sources. Starting guesses for the other parameters were then obtained by fixing the shape parameters and using 1–5 GN steps to find the best values for those starting ellipsoids.

All experiments were conducted in MATLAB using IEEE double precision floating point arithmetic.

6.1. Experiment 1.

We ran our algorithm on the first 40 systems that were generated by a damped GN run trying to reconstruct piecewise constant absorption and diffusion images. Systems numbered 2, 5, 7, 10, 13–19 (odd), 22–40 (even) correspond to the beginning of a line search; systems numbered 4, 6, 9, 12, 14–18 (even), 21–41 (odd) correspond to the end of a line search; and the remaining systems correspond to the middle of a line search.

First, we test our hypothesis that the invariant subspaces corresponding to the smallest eigenvalues of these matrices do not change much, whether we compare within a line search or across line searches. Recall that if the columns of $W^{(j)}$ form an orthonormal basis for the eigenspace associated with the smallest M eigenvalues for matrix $A^{(j)}$, and the same holds for $W^{(k)}$ and $A^{(k)}$, then the cosines of the canonical angles between range$(W^{(k)})$ and range$(W^{(j)})$ are given by [33, Corollary I.5.4]

$$\cos \Theta[\text{range}(W^{(j)}), \text{range}(W^{(k)})] = \Sigma[(W^{(j)})^T W^{(k)}],$$

where $\Sigma[V]$ denotes the singular values of the argument V. The sines of the canonical angles are therefore $\sqrt{1-\sigma_i^2}$, where the σ_i denote the cosines of the canonical angles. In Figure 3, we display the sines of the canonical angles between pairs of subspaces corresponding to the smallest eigenvalues of matrices. We observe that the invariant subspaces corresponding to the smallest eigenvalues are in fact relatively close, as predicted by Theorem 4.1. The plots for the sines with $M = 3$ and $M = 12$ illustrate the relative insensitivity of the invariant subspaces corresponding to the smallest 3 and smallest 12 eigenvalues, respectively. The indices have no correspondence with

[3] These experiments represent small test cases designed to test the regularization scheme itself. Ideal practical implementations of the inversion routine, which are not feasible without fast forward solvers such as those we present here, will require voxelations giving millions of unknowns.

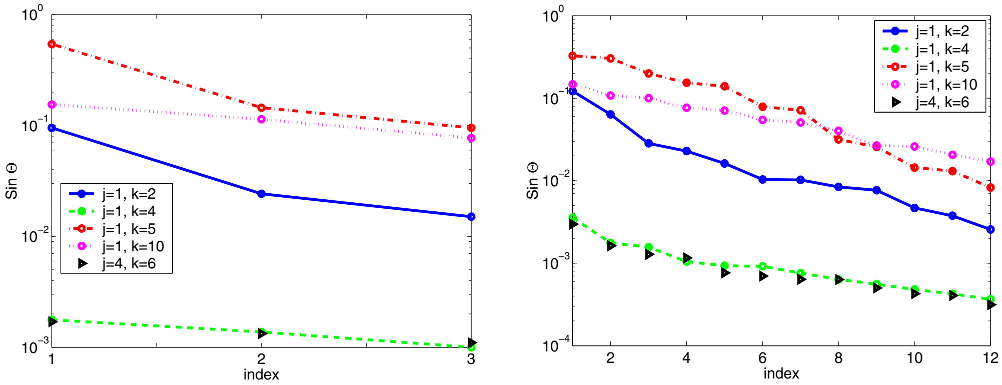

FIG. 3. *Experiment* 1. *Plots of* $\sin\Theta[\mathrm{range}(W^{(j)}), \mathrm{range}(W^{(k)})]$ *for various* (j,k) *for subspaces of dimension* $M = 3$ *(left) and* $M = 12$ *(right).*

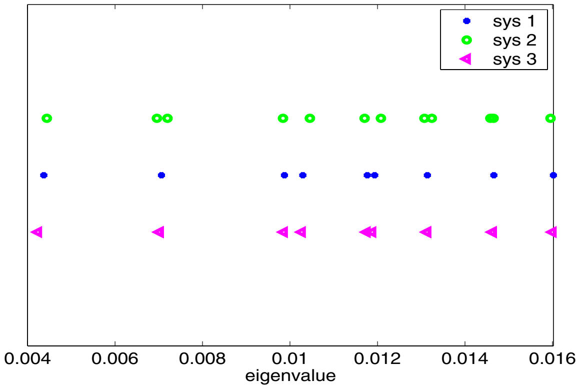

FIG. 4. *Experiment* 1. *Smallest* 12 *eigenvalues of* $A^{(j)}$, $j = 1, 2, 3$.

the eigenvectors themselves, nor do the canonical angles reflect the angles between corresponding eigenvectors. The values of j and k in the pictures were selected to illustrate the fact that the relevant invariant subspaces of matrices corresponding to (final) updates in the GN process remain fairly close to each other (e.g., 1 and 4, 4 and 6), whereas those from matrices from distinct phases in the line search differ more (e.g., 1 and 5, 1 and 10). Nevertheless, even these do not differ that much, particularly if a larger dimensional invariant subspace is used. The smallest 12 eigenvalues of $A^{(j)}$ for $j = 1, 2, 3$ are given in Figure 4.

We tested Algorithm 1, adjusted according to section 5.3 for the multiple right-hand side problem for zero frequency, and also adjusted for an additional nonzero frequency of $\omega = 5$ MHz using Algorithms 2a and 2b. For the first 6 right-hand sides in the initial phase of Algorithm 1, we saved 2 harmonic Ritz vectors at the end of each MINRES run to approximate the invariant subspace of $A^{(1)}$ corresponding to the 12 smallest eigenvalues. We used a threshold value of 10^{-3}, derived by trial and error, to distinguish between the beginning of a line search step and steps near the end. The left plot in Figure 5 shows the magnitudes of the spectral coefficients of the initial residuals for systems 2 through 5 which correspond to the 50 smallest eigenvalues, while the right plot gives the magnitudes corresponding to the 50 largest eigenvalues. Systems 2, 3, and 4 are all from the same line search, and we observe that the spectral

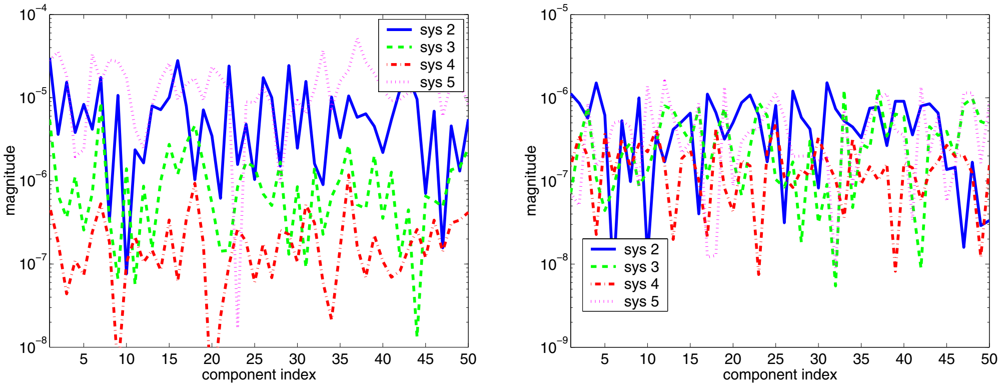

FIG. 5. *Experiment* 1. *Left: Plots of* $(W^{(j)})^T r_0$ *for* $j = 2, 3, 4, 5$ *for source* 1*, where* $W^{(j)}$ *corresponds to the eigenvector matrix associated with the* 50 *smallest eigenvalues of* $A^{(j)}$*. Right: Same, except for* 50 *largest eigenvalues of* $A^{(j)}$*.*

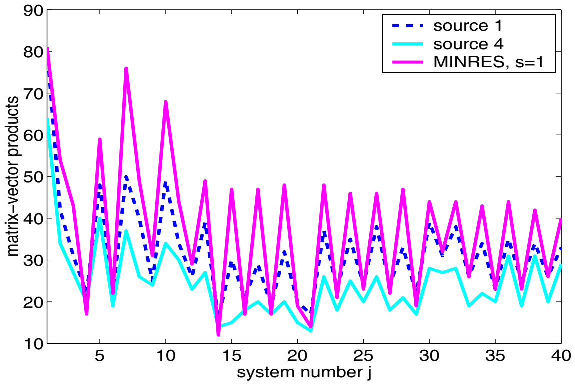

FIG. 6. *Experiment* 1*. Number of matrix-vector products per system solve for systems* 1:40 *and sources* 1 *and* 4*. The number of matrix-vector products for source* 4 *is representative of all sources after source* 1*. For comparison we give the number of matrix-vector products required for MINRES on the systems for source* 1*, where the starting guesses for each system were the solutions at the end of the most recent line search. Without subspace recycling, these numbers for source* 1 *are representative for all sources.*

components decrease by roughly one order of magnitude with each system. System 5 corresponds to the beginning of a new line search whose matrix and solution are not as close to those from the first line search. We observe a corresponding increase in the magnitudes of the spectral coefficients over the small eigenvalues, whereas a comparison of the figures shows that the initial residual is typically smaller over the subspace corresponding to the largest eigenvalues. This behavior is consistent both with the analysis of the initial residual in section 5.2 and with the observed convergence behavior for system 5 in the sense that the solver must work harder to reduce the residual significantly over these components.

Figure 6 gives the total number of matrix-vector products to solve each system using our recycling algorithm. This number includes the matrix-vector products required to compute the columns of C. Note that these represent a lower computational cost than actual iterations. The residuals for all the real systems were required to have a relative norm of 10^{-6}. Note the savings in matrix-vector products for the

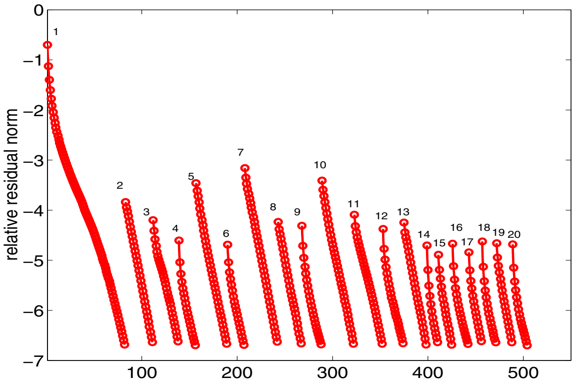

FIG. 7. *Experiment* 1. *Relative residual norms are displayed for the first* 20 *systems, source* 1.

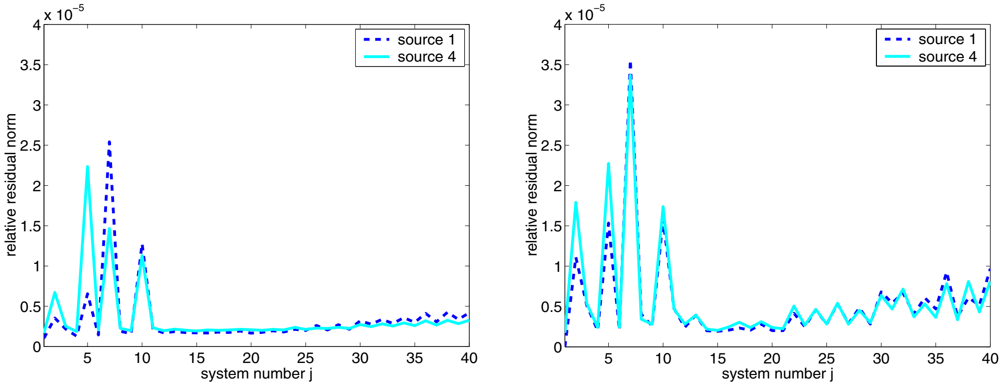

FIG. 8. *Experiment* 1. *Left: Relative residual norm per system solve, systems* 1:40, *sources* 1 *and* 4, *for* $\omega = 5$ *MHz, results computed using Algorithm* 2a. *Right: Relative residual norm per system solve, systems* 1:40, *sources* 1 *and* 4, *for* $\omega = 5$ *MHz computed using Algorithm* 2b.

right-hand sides other than the first of a single system, because most of the columns of C are computed only once for each s. Moreover, the projection has the desired effect of reducing the total number of iterations required for the projected system.

For comparison purposes, we note that if we used MINRES with a zero starting guess for every system and right-hand side, the number of matrix-vector products would be roughly constant at about 81 iterations for each system. Even MINRES with the solution at the end of the most recent line search as a starting guess could not achieve the reduction in the number of iterations we achieve with our algorithm, as demonstrated in Figure 6. Additionally, the matrix-matrix product $(A^{(j)}U)$ performed prior to running MINRES on the projected system runs faster than the equivalent number of matrix-vector products performed inside (unprojected) MINRES.

The relative residual norms for the first 20 systems, for source 1, are given in Figure 7. Note the convergence rate becomes higher and the initial relative residual norm becomes smaller as we move through one sequence of systems in a line search.

The plot in Figure 8 illustrates the relative residual norms that are achieved when solving the complex system with $\omega = 5$ MHz when Algorithms 2a and 2b are used to update the complex solution vectors. In neither case do we exactly attain a relative residual norm of 10^{-6}, the stopping criterion for the corresponding real

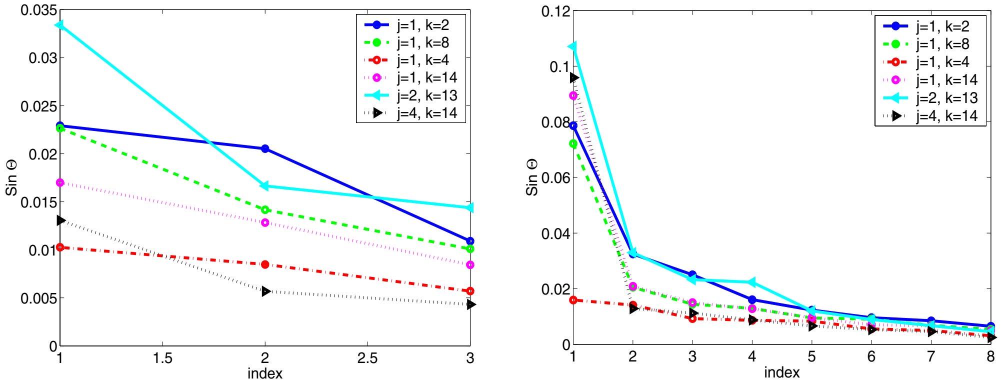

FIG. 9. *Experiment* 2. *Left: Plots of* $\sin\Theta[\text{range}(W^{(j)}), \text{range}(W^{(k)})]$ *for various* (j,k) *assuming a subspace dimension of* 3. *Right: Assuming a subspace dimension of* 8.

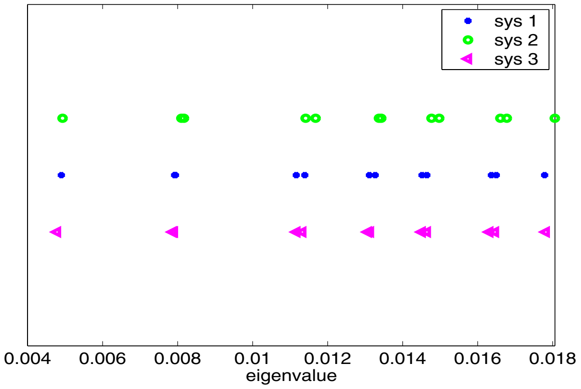

FIG. 10. *Experiment* 2. *Plot of smallest* 12 *eigenvalues for* $A^{(j)}$, $j = 1, 2, 3$. *Note that the smallest magnitude eigenvalues remain in clusters and do not cross clusters.*

system. However, for our application we feel this is sufficient, and perhaps more than sufficient since the measured data that defines the least squares problem contains noise at a level several orders of magnitude larger than 10^{-6}. On the other hand, we need to take the (unknown) conditioning of the algebraic system into account. Since previous solutions occur in the U matrix, this accounts for the slight upward creep of the graphs toward the end of the sequence of systems.

6.2. Experiment 2. In this experiment, the background diffusion and absorption were generated to have a “lumpy” variation [20], and thus the matrices that were generated correspond to piecewise continuous, rather than piecewise constant, absorption and diffusion. The total number of GN steps was 24 and the total number of system matrices was 51. For the first 40 systems the indices corresponding to the beginning of a line search are 2, 5, 8, 10, 13–39 (odd), while indices corresponding to the end of a line search are 4 ,7, 9, 12, 14–40 (even).

The sines of the canonical angles between different pairs of invariant subspaces associated with the smallest eigenvalues and for different subspace dimensions are given in Figure 9. In Figure 10 the smallest 12 eigenvalues are displayed. Consistent with Theorem 4.1 and our conjecture that the changes in the matrices are concentrated

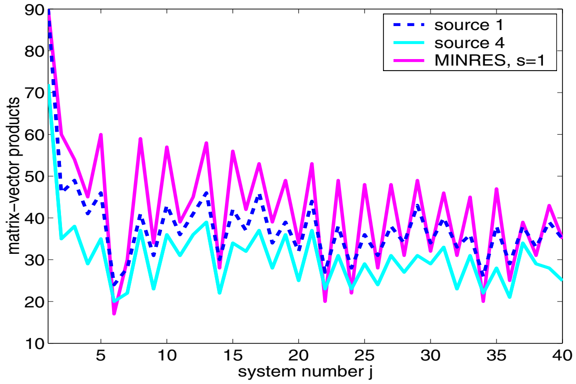

FIG. 11. *Experiment* 2. *Number of matrix-vector products per system solve for systems* 1:40 *and sources* 1 *and* 4. *The number of matrix-vector products for source* 4 *is representative of all sources after source* 1. *For comparison we give the number of matrix-vector products required for MINRES on the systems for source* 1 *where the starting guesses for each system were the solutions at the end of the most recent line search. Without subspace recycling, these numbers for source* 1 *are representative for all sources.*

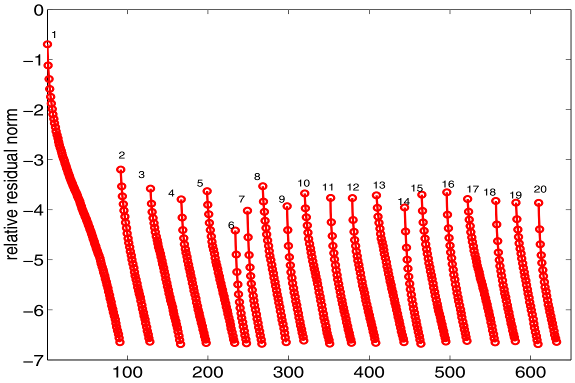

FIG. 12. *Experiment* 2. *Relative residual norms for systems* 1 *to* 20, *source* 1.

in the invariant subspaces corresponding to higher frequencies (larger eigenvalues), the smallest eigenvalues remain in disjoint clusters.

Figures 9 and 10 seem to indicate that the smallest 8 or so eigenvalues correspond to an invariant subspace that remains well separated from its orthogonal complement. Therefore, in the initialization phase of Algorithm 1, we added two vectors to W corresponding to the smallest harmonic Ritz values of $\tilde{T}_m$ (i.e., the matrix in (5.3) with $\gamma = 0$) for each of the first 4 sources. In Figure 11, we see the effect of keeping these 8 columns plus the other vectors proposed in section 5.3. In this experiment, $W_{ind} = W$, and a threshold value was used to distinguish the phase within the line search. Just as for the first experiment, note the savings in matrix-vector products for the right-hand sides after the first for a single system. Again, we compare our results to MINRES with $x^{(curr)}$ from Algorithm 1 as the starting guess.

Relative residual norms for Algorithm 1 for the first 20 systems for source 1 are given in Figure 12. We observe an increased rate of convergence and smaller initial residuals in the course of a line search.

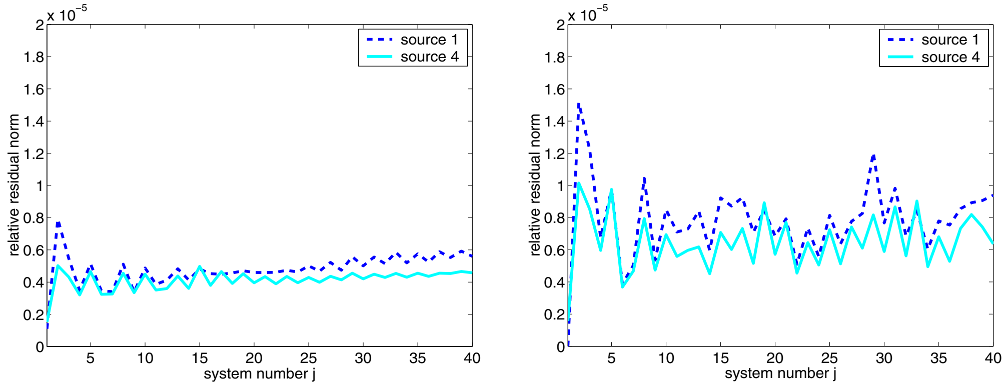

FIG. 13. *Experiment* 2. *Left: Relative residual norms for systems* 1 *to* 40*, sources* 1 *and* 4*,* $\omega = 5$ *MHz using Algorithm* 2a*. Right: Relative residual norms for systems* 1 *to* 40*, sources* 1 *and* 4*,* $\omega = 5$ *MHz using Algorithm* 2b*.*

Finally, the relative residual norms for the complex systems at $\omega = 5$ MHz are given in Figure 13 for Algorithm 2a and 2b. In general, the behavior is more uniform for these systems versus those in the first experiment since the invariant subspaces corresponding to the small eigenvalues among the matrices are more closely related.

7. Conclusions and future work. We have discussed various strategies for Krylov subspace recycling to improve the convergence of linear solvers for a sequence of slowly changing linear systems arising in computations for optical tomography. We have combined strategies of recycling approximate invariant subspaces and strategies of recycling subspaces from previous solutions. Furthermore, our algorithms are based on a careful analysis of which strategy is most useful at each stage of the optimization algorithm. This analysis also takes the underlying application, DOT, and matrix symmetry into account. Additionally, we have adapted the GCRO algorithm to combine subspace recycling with solving for multiple shifted systems using a single Krylov subspace. Our numerical results, based on two model problems for DOT, show that our strategies are quite effective. Although we have focused on a particular application and optimization algorithm, we feel that this approach to tuning the linear solver is applicable generally to problems where many linear systems must be solved.

Important future work includes the study of how characteristics of the matrices arising in DOT, such as invariant subspaces and eigenvalues, change for small changes in model parameters. This may lead to further improvements for linear solvers and also improved line search strategies for the nonlinear solver. This issue is, of course, equally important for other applications where we must solve a long sequence of slowly changing problems, such as crack propagation [26]. Future work will combine modeling aspects from applications with matrix theory. Other useful extensions that we plan to research are the use of GCROT-like techniques [11] to measure the effectiveness of the recycled subspace and update the space accordingly, to combine our current algorithms with a block approach [37], and to further tune the nonlinear algorithm, in particular the line search.

Acknowledgment. We would like to thank the referees for their comments, which helped us to improve the presentation of this paper.

REFERENCES

[1] R. E. Alcouffe, A. Brandt, J. E. Dendy, Jr., and J. W. Painter, *The multi-grid method for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 430–454.

[2] S. R. Arridge, *Optical tomography in medical imaging*, Inverse Problems, 16 (1999), pp. R41–R93.

[3] A. H. Baker, E. R. Jessup, and T. Manteuffel, *A technique for accelerating the convergence of restarted GMRES*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 962–984.

[4] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1993.

[5] A. Brandt, *Multigrid Techniques:* 1984 *Guide, with Applications to Fluid Dynamics*, GMD-Stud. 85, St. Augustin, Germany, 1984.

[6] M. Celia and W. Gray, *Numerical Methods for Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[7] T. F. Chan and M. K. Ng, *Galerkin projection methods for solving multiple linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 836–850.

[8] T. F. Chan and W. L. Wan, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM J. Sci. Comput., 18 (1997), pp. 1698–1721.

[9] E. de Sturler, *Inner-outer methods with deflation for linear systems with multiple right-hand sides*, in Householder Symposium XIII, Proceedings of the Householder Symposium on Numerical Algebra, Pontresina, Switzerland, 1996, pp. 193–196.

[10] E. de Sturler, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.

[11] E. de Sturler, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889.

[12] E. De Sturler and H. A. Van der Vorst, *Communication cost reduction for Krylov methods on parallel computers*, in High-Performance Computing and Networking, Lecture Notes in Comput. Sci. 797, W. Gentzsch and U. Harms, eds., Springer-Verlag, Berlin, 1994, pp. 190–195.

[13] E. De Sturler and H. A. Van der Vorst, *Reducing the effect of global communication in GMRES(m) and CG on parallel distributed memory computers*, Appl. Numer. Math., 18 (1995), pp. 441–459.

[14] J. E. Dennis, Jr., and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Classics Appl. Math. 16, SIAM, Philadelphia, 1996.

[15] M. Eiermann, O. G. Ernst, and O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods*, J. Comput. Appl. Math., 123 (2000), pp. 261–292.

[16] P. F. Fischer, *Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 193–204.

[17] R. Freund and M. Malhotra, *A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl., 254 (1997), pp. 197–257.

[18] A. Frommer and U. Glässner, *Restarted GMRES for shifted linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 15–26.

[19] A. Frommer and P. Maass, *Fast CG-based methods for Tikhonov–Phillips regularization*, SIAM J. Sci. Comput., 20 (1999), pp. 1831–1850.

[20] M. E. Kilmer, E. Miller, A. Barbaro, and D. Boas, *3D shape-based imaging of absorption perturbation for diffuse optical tomography*, Appl. Optics, 42 (2003), pp. 3129–3144.

[21] M. Kilmer, E. Miller, and C. Rappaport, *QMR-based projection techniques for the solution of non-Hermitian systems with multiple right-hand sides*, SIAM J. Sci. Comput., 23 (2001), pp. 761–780.

[22] M. E. Kilmer and D. P. O'Leary, *Choosing regularization parameters in iterative methods for ill-posed problems*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1204–1221.

[23] R. B. Morgan, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1112–1135.

[24] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.

[25] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[26] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov Subspaces for Sequences of Linear Systems*, Technical report UIUCDCS-R-2004-2421/UILU-ENG-2004-1722, Department of Computer Science, University of Illinois at Urbana-Champaign, 2004. SIAM J. Sci. Comput., submitted.

[27] C. Rey and F. Risler, *A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems*, Numer. Algorithms, 17 (1998), pp. 279–311.

[28] F. Risler and C. Rey, *Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems*, Numer. Algorithms, 23 (2000), pp. 1–30.

[29] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 435–449.

[30] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[31] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.

[32] V. Simoncini and D. Szyld, *On the occurrence of superlinear convergence of exact and inexact Krylov subspace methods*, SIAM Rev., 47 (2005), pp. 247–272.

[33] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[34] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2000.

[35] H. A. Van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.

[36] C. R. Vogel, *Computational Methods for Inverse Problems*, Frontiers Appl. Math. 23, SIAM, Philadelphia, 2002.

[37] R. Yu, E. de Sturler, and D. D. Johnson, *A Block Iterative Solver for Complex Non-Hermitian Systems Applied to Large-Scale Electronic-Structure Calculations*, Technical report UIUCDCS-R-2002-2299 and UILU-ENG-2002-1742, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.