RECYCLING SUBSPACE INFORMATION FOR DIFFUSE OPTICAL TOMOGRAPHY*

MISHA KILMER† AND ERIC DE STURLER‡

Abstract.
We discuss the efficient solution of a large sequence of slowly varying linear systems arising in computations for diffuse optical tomographic imaging. In particular, we analyze a number of strategies for recycling Krylov subspace information for the most efficient solution.

We reconstruct three-dimensional absorption and scattering information by matching computed solutions from a parameterized model to measured data. For this nonlinear least squares problem we use the Gauss-Newton method with a line search. This algorithm requires the solution of a large 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 involving both multiple shifts and multiple right-hand sides. We discuss strategies that minimize the overall solution time. In particular, we show how we can tune the linear solver for both the nonlinear optimization algorithm and the underlying application. Although we focus on a particular application and optimization algorithm, we feel that our approach is applicable generally to problems where many linear systems must be solved.

We describe extensions to the GCRO algorithm to deal efficiently with symmetric problems and to combine subspace recycling with solving for multiple shifts using a single Krylov subspace. 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

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 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 [1]. We focus on the frequency-domain diffusion model in which the data is a non-linear 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 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.

Specifically, we consider the solution of a sequence of linear systems of the form

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

*This work was supported by NSF grants 0208548 and 0139968, DOE LLNL B341494, and NSF DMR-9976550 and DMR-0325939.

†Mathematics Department, 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.

that arise in a 3D imaging algorithm for diffuse optical tomography. 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 damped Gauss-Newton iteration (GN) to solve the nonlinear least squares problem for an optimal set of parameters. As suggested by the notation, the matrix depends on the current values of the parameters. The matrix $A(p_j)$ is sparse and symmetric, which means that Krylov iterative solvers such as MINRES [3] are good candidates for solving these linear systems.

We observe that in our application the matrices $A(p_j)$ vary slowly from one Gauss-Newton step or line search step to the next, and they change significantly over multiple Gauss-Newton 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, suggestions have been made to reduce the overall cost (for example, see [16, 15, 19] for solving for a group of matrices that differ by a constant times the identity and [14, 29, 7, 22, 18] for solving for multiple right-hand sides). However, the various methods have not been combined to address all these features at once, and we will see that 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. Some approaches have been proposed, though most of them for systems that are special in some sense.

In [22] we propose to *recycle* from one linear system to the next the Krylov subspaces that solvers like GCRO [8], GCROT [9], and GMRESDR [21] 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 [22]. Recycling selected subspaces leads to significant improvements in the convergence of subsequent systems.

Other approaches have been proposed as well. If all matrices in a set of symmetric positive definite linear systems are pair-wise close to each other and all right-hand sides are available simultaneously, the methods proposed by Chan and Ng [5] can be used. However, this is not the case for our application. For a sequence of real, symmetric positive definite systems, Rey and Risler have proposed to retain all converged Ritz vectors from a previous CG iteration to reduce the effective condition number for a new system [23, 24, 25]. In general, this requires excessive storage. Moreover, they lose the advantage of a short recurrence, as they keep the *full* recurrence during the solution of a single system. Since they focus on the finite element tearing and interconnecting (FETI) method [12], this is less of a drawback, because the interface problem is small relative to the overall problem, and it is common to use a full recurrence in FETI. However, for more general problems, such as the present one, it is paramount to be more selective in deciding which subspace should be recycled for subsequent linear systems. Finally, Fischer 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 [13]. Note that his algorithm does not maintain orthogonality to the subspace of previous right hand sides over the iteration.

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 [7, 22], subspace recycling is quite effective for this problem. However, other approaches or variations have been successful as well, in particular block methods [14, 29] and seed methods [6, 18]. As we will see, 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 [22]. This avoids the need to change 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 without (re)generating additional Krylov subspaces. 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 a simple extension to the GCRO method in Section 5 to deal with this problem.

In [22] we select and use the subspaces for recycling in a fairly straightforward way for each particular method. In the present paper we aim to derive specific strategies related to the application and the nonlinear optimization algorithm to improve convergence even further. In particular, we explore which information to keep and which to save as the damped GN method progresses. Furthermore, we present a variant of GCRO that exploits the symmetry of the matrix and the fact that we want to solve shifted complex systems simultaneously. The symmetry means that there is no need to restart for a single linear system to save on storage.

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 diffuse optical tomography. In addition, we derive the sequence of linear systems of the form (1.1) that we wish to solve, and in Section 4 we discuss some characteristics of the system that allow us to use 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 here find their origin in attempts to improve the convergence of Krylov subspace methods for a single linear system, in particular restarted and truncated methods. We briefly discuss these ideas first. Restarting GMRES [28] 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 [2, 8, 9, 21, 22, 27]. A taxonomy of popular choices is given in [11] and various approaches to convergence theory for GMRES that are relevant here can be found in [30, 32].

As discussed in [22], two aspects play a role here. The first aspect is which subspace to retain to maintain convergence close to that of full GMRES – we will refer to this as the *recycled* subspace. The second aspect is how to use that subspace. In an *augmentation* approach, we append additional vectors at the end of the Arnoldi recurrence, in the manner of FGMRES [26], such that an Arnoldi-like relation is formed [27]. In an *orthogonalization* approach, we first minimize the residual over the recycled subspace, and then maintain orthogonality with the image of this space in the Arnoldi recurrence, see, e.g., GCRO [8]. As argued in [8, 20, 22] the the orthogonalization approach generally leads to better convergence.

Several choices have been suggested regarding the subspace to recycle after a restart for a single linear system. One important choice is to recycle an approximate invariant subspace, typically associated with the eigenvalues closest to the origin, but other approximate invariant subspaces can be used as well [20, 21]. An alternative choice is to retain the subspace that minimizes the loss of orthogonality with respect to the discarded subspace [9]. This tends to work well for nonsymmetric problems. In [8] the updates to the solution (residual) are recycled. This is also proposed in [2] as

an augmentation approach. Of course, we can use knowledge of the underlying physics of our problem and the nonlinear optimization algorithm to decide on subspaces to recycle. We will consider combinations of all these possible strategies.

The GCRO method provides a general mechanism to include arbitrary additional subspaces in the search space. We are not limited to recycling subspaces of previously generated Krylov spaces; in the context of the present application several choices of subspaces may be useful. We explain briefly how GCRO combines the selected recycle space with a newly generated Krylov subspace to obtain an optimal approximation over the sum of these spaces. We present the basic steps in a mathematically equivalent but slightly different way from the presentation in [8]. This different presentation is more easily generalized to solving for a set of matrices that differ only by a constant times the identity without generating another Krylov subspace.

We consider solving the linear system $Ax = b$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Assume we have the matrices $U_k \in \mathbb{R}^{n\times k}$ and $C_k \in \mathbb{R}^{n\times k}$, such that $AU_k = C_k$ and $C_k^T C_k = I_k$. If we choose an approximate solution to our system in $Range(U_k)$ that minimizes the 2-norm of the residual, then the corresponding residual, $r_k = b - C_k C_k^T b$, will be orthogonal to $Range(C_k)$. There are no restrictions on the matrix U_k; clearly, the columns of U_k should be chosen so that a reasonable approximate solution can be found for small k. (For example, the columns of U_k could contain a basis for the Krylov subspace generated by A and b, but they might also contain previous approximate solutions or corrections to previous approximate solutions.) If the solution so obtained is not adequate, we expand the subspace in which we look for solutions according to the discussion in [8]. Now let $v_1 = (I - C_k C_k^T)b/\|(I - C_k C_k^T)b\|$. We use an Arnoldi recurrence with $(I - C_k C_k^T)A$ and v_1. This gives the following recurrence relation

$$(I - C_k C_k^T)AV_m = V_{m+1}\underline{\mathrm{H}}_m \Leftrightarrow \tag{2.1}$$

$$AV_m = C_k C_k^T AV_m + V_{m+1}\underline{\mathrm{H}}_m. \tag{2.2}$$

Next, we want to find the approximation in $Range([V_m\ U_k])$ that minimizes the two-norm of the residual, $\|b - A(V_m y + U_k z)\|_2$. This can be reduced to a simpler problem as follows.

$$\min_{y,z} \left\| b - A[V_m\ U_k] \begin{bmatrix} y \\ z \end{bmatrix} \right\|_2 = \tag{2.3}$$

$$\min_{y,z} \left\| b - [V_{m+1}; C_k] \begin{bmatrix} \underline{\mathrm{H}}_m & 0 \\ C_k^T AV_m & I \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \right\|_2 = \tag{2.4}$$

$$\min_{y,z} \left\| \begin{bmatrix} \xi e_1 \\ C_k^T b \end{bmatrix} - \begin{bmatrix} \underline{\mathrm{H}}_m & 0 \\ C_k^T AV_m & I \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \right\|_2, \tag{2.5}$$

where e_1 denotes the first Cartesian basis vector in $\mathbb{R}^{m+1}$ and $\xi = \|(I - C_k C_k^T)b\|$. The minimization in (2.5) corresponds to a small least squares problem that can be solved by standard methods. An efficient implementation would follow the approach suggested in [28]. Note that no assumptions have been made on the space $Range(C_k)$. 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 $Range((A + i\gamma I)U_k) \not\subset Range(C_k)$ and of course $Range((A + i\gamma I)U_k)$ depends on γ.

If we have found a matrix U_k that speeds up the convergence significantly, we can also reuse this matrix for the next right-hand side (with the same A and γ). This does not require any changes in the algorithm, and allows the algorithm to learn which spaces are best to remove over the solution of multiple right-hand sides. This approach

to improve the convergence for multiple right-hand sides and a constant matrix does not require any extra storage beyond what is needed for a single right-hand side.

This method can also be combined with a block method [34]. Extending this in turn to incorporate multiple shifts is possible too. However, the use of block methods increases the required memory significantly, and complicates the resulting program, especially when using deflation [14]. Since we showed that subspace recycling is quite effective for solving multiple right-hand sides, we will not include the use of block techniques here. It may be a worthwhile strategy to pursue later.

Again if we have found a matrix U_k that speeds up convergence significantly for $A(p_j)$ and $A(p_{j+1}) - A(p_j)$ is small it makes sense to try to reuse the same search space U_k possibly extended with other search directions for the linear systems with $A(p_{j+1})$ as well. We need to update C_k in this case to reflect the new operator. In many cases this can be done very cheaply. This process is not complicated, and we refer to [22] for details.

3. The DOT Imaging Application. In this section, we introduce the image reconstruction problem for diffuse optical tomography. In the course of the discussion, 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. A limited number of N_s sources will be present on the top, and a limited number of N_d detectors will be located on the either the top or bottom or both. We use the diffusion model [1] 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 [1]. 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 subscript s is an integer index which indicates that this is the model corresponding to 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 $D(r)$. Knowing the source and the functions $\mu_a(r), D(r)$, we could 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 [4] 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 $\phi_s(x_l, y_j, z_k)$ for $l = 1{:}N_x, j = 1{:}N_y, k = 1{:}N_z$. We will order the unknowns so that ϕ_s values at points on the top of the box come first (i.e. let $k = 1$ and loop over all l, j), then the ϕ_s 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. The corresponding vector with entries $\phi_s(x_l, y_j, z_k)$ we will call $\boldsymbol{\phi_{s,\omega}}$. Likewise, we will call the vector with

entries $\mu_a(x_l, y_j, z_k)$ $\boldsymbol{\mu_a}$. The vector $\boldsymbol{D}$ will have entries at half-integer grid points (i.e. $(x_{l\pm 1/2}, y_j, z_k), (x_l, y_{j\pm 1/2}, z_k)$, etc.) as well as whole integer grid points because of the particular discretization we are using.

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

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

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* that is **predicted** by this forward model is a subsampled version of the subvector $w_{\omega,s}$, which we call $\boldsymbol{\psi_{s,\omega}}$.

It is important to note that in our application $\boldsymbol{f_s}^{(2)} = 0$ and

$$\boldsymbol{f_s}^{(1)} = h^2[0, 0, \ldots 0, 1, 0, \ldots 0]^T,$$

where 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. 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 [17] for more details. The 3D imaging problem then becomes one of finding the optimal parameters such that the data predicted by the diffusion model is well matched by the data. That is, we wish to solve

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

where $\mathbf{y}$ denotes the vector obtained by stacking the subvectors $y_{s,\omega}$. We use the following damped Gauss-Newton iteration (GN) to solve this non-linear least squares problem [10]:

1. Compute $\boldsymbol{\psi}(p_k)$, $J(p_k)$,
2. Solve $J(p_k)^T J(p_k) s_k = -J(p_k)^T \epsilon(p_k)$,
3. $p_{k+1} = p_k + \lambda_k s_k$,

where λ_k is chosen using a backtracking line search [10] and $\epsilon(p_k)$ denotes the weighted residual in the right side of the equation above evaluated at parameter vector p_k.

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 very computationally intensive. However, to compute $\boldsymbol{\psi}(p_k)$, to compute the entries in the Jacobian using an adjoint-type approach (called a "co-state" method in [33]), and to find the best search direction requires solutions of the matrix equation in (3.1) for every source and every frequency. Therefore, the rest of this paper is devoted to analyzing the systems themselves 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 $\boldsymbol{\phi_{s,\omega}}$. The following are important facts about the structure of the blocks in (3.1):

- B is an invertible diagonal matrix.

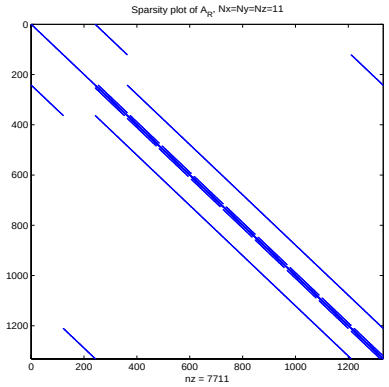

FIG. 3.1. *Sparsity plot of matrix G.*

- D_1 has at most one non-zero per row, and these occur only in the first N_xN_y and last N_xN_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 3.1 to give the reader a visual interpretation of the structure just mentioned.

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

$$G = \begin{bmatrix} I & 0 \\ D_2B^{-1} & I \end{bmatrix} \begin{bmatrix} B & D_1 \\ 0 & C - D_2B^{-1}D_1 + i\gamma I \end{bmatrix},$$

where $\gamma = h^2\omega/\nu$. Thus, any system involving G, say $G \begin{bmatrix} w_{\omega,s} \\ x_{\omega,s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_s}^{(1)} \\ 0 \end{bmatrix}$, can be solved according to the following steps:

1. Step 1: Solve the equation $L \begin{bmatrix} a_s \\ b_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_s}^{(1)} \\ 0 \end{bmatrix}$:
   (a) $a_s = \boldsymbol{f_s}^{(1)}$
   (b) $b_s = -D_2(B^{-1}a_s)$
2. 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 $(C - D_2B^{-1}D_1 + i\gamma I)x_{s,\omega} = b_s$
   (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 only have $(2N_xN_y)$ non-zero entries each. The computationally intensive part of this procedure is Step 2a. Note that the system in Step 2a is exactly the system in (1.1), except that we have ignored the superscript on $x_{s,\omega}$ for simplicity.

We conclude this section with the proof that $C - D_2B^{-1}D_1$ is symmetric and positive definite. Note that C 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. It follows that the matrix C is symmetric and positive definite.

Recall that B is a diagonal, $2N_xN_y \times 2N_xN_y$ matrix 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 non-zero per row and D_2 is $(N_xN_y(N_z -$

2)) $\times 2N_xN_y$ with only one non-zero per column. The non-zero entries in D_1 are of the form $-\frac{h}{2}\boldsymbol{D}_{l,j,m}$ with $m = 1$ or $m = N_z$. The non-zero entries in D_2 are $-\boldsymbol{D}_{l,j,\frac{3}{2}}$ or $-\boldsymbol{D}_{l,j,Nz-\frac{1}{2}}$. From this we deduce that the matrix $D_2B^{-1}D_1$ is a diagonal matrix with positive, non-zero 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 $C - D_2B^{-1}D_1$ is symmetric and positive definite.*

Proof. Symmetry follows from the symmetry of C and $D_2B^{-1}D_1$. Matrices $C-D_2B^{-1}D_1$ and C only differ in the first and last N_xN_y components on the diagonal. Since the non-zeros in the second matrix are strictly positive, the first and last N_xN_y diagonal entries of $C - D_2B^{-1}D_1$ are smaller than the corresponding diagonal entries of C. 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 (2a) 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:

$$(\underbrace{C - D_2B^{-1}D_1}_{A^{(j)}} + i\gamma I)x^{(j)}_{s,\omega} = b_s. \tag{4.1}$$

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 [17], 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 - they have 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. For example, an entry in $\boldsymbol{D}$ has a value of α_1 if that entry corresponds to a voxel inside the diffusion anomaly, otherwise, its value is determined by the corresponding component of the vector $B_1\beta_1$. 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 include $\alpha_i, \beta_i, i = 1, 2$, and up to 6 length-3 vectors specifying the locations of the 2 ellipsoids. For more details, see [17].

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 $1/45\,\text{cm}^{-1}$. It is usually the case that we have or can obtain good approximations to the average background values of diffusion and absorption and use these for starting guesses for the GN iteration [17]. As the GN iterations progress, 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 disjunct from one system to the next, even if the eigenvalues themselves differ from one matrix to the next (see Figure 6.2, 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: Recalling that $\boldsymbol{f_s}$ has only one nonzero coefficient in, say, position m_s, it follows that $b_s^{(j)} = -\frac{(D_2)_{m_s,m_s}h^2}{B_{m_s,m_s}} e_{m_s}$. Since blocks D_1, D_2, B and vector $\boldsymbol{f_s}^{(1)}$ do not depend on frequency, $b_s^{(j)}$ is independent of frequency. As long as the GN iteration is converging, the values of $(D_2)_{k,m_s}$ and B_{m_s,m_s} do not change much in the course of the inversion because these indices refer to positions near the boundary where the values of absorption and diffusion are already accurately captured early in the GN process. Therefore, in the remaining discussion, we consider iterative methods applied to the approximate systems

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

in order to study convergence.

Similarity among Solutions: From (4.2), it is easy to see that the solutions to any pair of systems, say systems k and j, during the damped Gauss-Newton iteration are related by

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

So, the solutions do not change much as long as the matrices $A^{(j)}$ and $A^{(k)}$ remain close.

Smooth Solutions: The vector e_{m_s} is comprised mostly of high-frequency Fourier components. It is well known that the eigenvectors corresponding to the smallest eigenvalues of $A^{(j)}$ are smooth (or low frequency) while the eigenvectors corresponding to the largest eigenvalues represent high frequency. We can think of $(A^{(j)})^{-1}$ as a discretization of an integration operator that when applied to e_{m_s} acts as a blurring operator. For this reason, we expect the solutions $x_s^{(j)}$ to be smooth: that is, 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: It is well known (see, for example, [19] and the references therein) that the Krylov vectors that are generated when solving $A^{(j)}x_s^{(j)} = e_{m_s}$

also span the Krylov subpace generated by the shifted matrix $(A^{(j)} + i\gamma I)$ and e_{m_s}. Therefore, without storing or generating extra Krylov vectors, there exist short term recurrences for solving the systems of non-zero γ with little extra work. In our case, since $\gamma > 0$ is not too large relative to the eigenvalues of $A^{(j)}$, the convergence rate should also be about the same.

When subspace recycling is used, however, it is not straightforward to produce solutions to the complex shifted systems. We discuss this further 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 runs, we must explore the relationships of the corresponding invariant subspaces from one matrix to the next. 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 disjunct 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 much 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. Next, we 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. Although we used V previously to denote Krylov vectors, in this discussion, the matrix V is used to denote the eigenvector matrix corresponding to A.

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

$$A = [V_1\ V_2\ V_3]\mathrm{diag}(\Lambda_1, \Lambda_2, \Lambda_3)[V_1\ V_2\ V_3]^T, \tag{4.3}$$

where $V = [V_1\ V_2\ V_3]$ 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,

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

Now we consider the changes in the invariant subspace range(V_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$([V_1\ V_2])$ is small, say $\|E[V_1\ V_2]\|_F \leq \varepsilon$, and $\|EV_3\|_F = \eta \approx \|E\|_F$. 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$(\hat{V}_1)$ such that the canonical angles between range(V_1) and range$(\hat{V}_1)$ 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 $n \geq m$, $\Theta(\text{range}(Y), \text{range}(Z))$ denotes the diagonal matrix with the canonical angles between range(Z) and range(Y) as coefficients, and $\theta_1(\text{range}(Y), \text{range}(Z))$ 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

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

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

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

THEOREM 4.1. *Let A be SPD 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 $\hat{V}_1$ conforming to V_1 such that* range$(\hat{V}_1)$ *is a simple invariant subspace of $A + E$, and*

$$\tan\theta_1\left(\text{range}(V_1), \text{range}(\hat{V}_1)\right) \leq \frac{\varepsilon}{\hat{\delta}}.$$

Furthermore, the eigenvalues $\hat{\lambda}_j^{(1)}$ of $A + E$ corresponding to the invariant subspace range$(\hat{V}_1)$ *satisfy*

$$\forall\, \hat{\lambda}_j^{(1)} :\ \exists\, \lambda_i^{(1)} \ \text{ such that } \ |\hat{\lambda}_j^{(1)} - \lambda_i^{(1)}| \leq \varepsilon + \frac{2\varepsilon^2}{\delta}, \tag{4.6}$$

and in particular,

$$\lambda_{\max}(\,\hat{V}_1^T(A+E)\hat{V}_1\,) \leq \lambda_{k1}^{(1)} + \varepsilon + \frac{2\varepsilon^2}{\delta}. \tag{4.7}$$

Proof. We consider the perturbation E, such that

$$V^T(A+E)V = \begin{pmatrix} \Lambda_1 + \mathcal{E}_{11} & 0 & 0 \\ 0 & \Lambda_2 + \mathcal{E}_{22} & \mathcal{E}_{32}^T \\ 0 & \mathcal{E}_{32} & \Lambda_3 + \mathcal{E}_{33} \end{pmatrix} + \begin{pmatrix} 0 & \mathcal{E}_{21}^T & \mathcal{E}_{31}^T \\ \mathcal{E}_{21} & 0 & 0 \\ \mathcal{E}_{31} & 0 & 0 \end{pmatrix}. \tag{4.8}$$

By the assumptions above we also have

$$\left\| \begin{pmatrix} \mathcal{E}_{21} \\ \mathcal{E}_{31} \end{pmatrix} \right\|_F \leq \varepsilon, \tag{4.9}$$

$\|\mathcal{E}_{11}\|_F \leq \varepsilon$, $\|\mathcal{E}_{22}\|_F \leq \varepsilon$ and $\|\mathcal{E}_{33}\|_F \leq \eta$. From (4.8) we see that

$$L_1 \ \equiv\ V_1^T(A+E)V_1 = \Lambda_1 + \mathcal{E}_{11}, \tag{4.10}$$

$$L_{23} \ \equiv\ [V_2\ V_3]^T(A+E)[V_2\ V_3] = \begin{pmatrix} \Lambda_2 + \mathcal{E}_{22} & \mathcal{E}_{32}^T \\ \mathcal{E}_{32} & \Lambda_3 + \mathcal{E}_{33} \end{pmatrix}. \tag{4.11}$$

From [31, Corollary IV.3.4] it follows that

$$\lambda_{\max}(\Lambda_1 + \mathcal{E}_{11}) \leq \lambda_{k1}^{(1)} + \|\mathcal{E}_{11}\| \leq \lambda_{k1}^{(1)} + \varepsilon, \tag{4.12}$$

$$\lambda_{\min}(\Lambda_2 + \mathcal{E}_{22}) \geq \lambda_1^{(2)} - \|\mathcal{E}_{22}\| \geq \lambda_1^{(2)} - \varepsilon, \tag{4.13}$$

$$\lambda_{\min}(\Lambda_3 + \mathcal{E}_{33}) \geq \lambda_1^{(3)} - \|\mathcal{E}_{33}\| \geq \lambda_1^{(3)} - \eta. \tag{4.14}$$

Now we can apply [31, Corollary IV.3.4] once more to obtain

$$\lambda_{\min}(L_{23}) \geq \min(\lambda_1^{(2)} - \varepsilon, \lambda_1^{(3)} - \eta) - 2\varepsilon. \tag{4.15}$$

From (4.12)–(4.15) we have $\mathrm{sep}(L_1, L_{23}) > \delta$. Furthermore, let $R \equiv (A+E)V_1 - V_1 L_1 = V_2\mathcal{E}_{21} + V_3\mathcal{E}_{31}$. Then, from symmetry it follows that $V_1^T(A+E) - L_1 V_1^T = R^T$, 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 [31, Corollary V.2.2] we know there exists a matrix $\hat{V}_1$ conforming to V_1 such that $\mathrm{range}(\hat{V}_1)$ is a simple invariant subspace of $A+E$, and

$$\tan\theta_1\left(\mathrm{range}(V_1), \mathrm{range}(\hat{V}_1)\right) \;\leq\; \|\tan\Theta\left(\mathrm{range}(V_1), \mathrm{range}(\hat{V}_1)\right)\|_F \;\leq\; 2\frac{\varepsilon}{\delta}. \tag{4.16}$$

Since A and $A+E$ are symmetric and we have established the existence of $\hat{V}_1$, we can obtain an better bound using [31, 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 $\hat{L}_{23} \equiv \hat{V}_{23}^T(A+E)\hat{V}_{23}$, where $\mathrm{range}(\hat{V}_{23}) = \mathrm{range}(\hat{V}_1)^\perp$ and $\hat{V}_{23}$ has orthonormal columns. From [31, 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 [31, Corollary IV.3.4], we have the following bound

$$\lambda_{\min}(\hat{L}_{23}) \geq \min(\lambda_1^{(2)} - \varepsilon, \lambda_1^{(3)} - \eta) - 2\varepsilon - \frac{2\varepsilon^2}{\delta} = \hat{\delta}. \tag{4.17}$$

Finally, we obtain from [31, Theorem V.3.10]

$$\tan\theta_1(\mathrm{range}(V_1), \mathrm{range}(\hat{V}_1)) \leq \|\tan\Theta(\mathrm{range}(V_1), \mathrm{range}(\hat{V}_1))\|_F \leq \frac{\varepsilon}{\hat{\delta}}, \tag{4.18}$$

which is about a factor 2 better than (4.16). Analogously to (4.17) we have from [31, Theorem V.3.10] for each eigenvalue $\hat{\lambda}_j^{(1)}$ of $\hat{V}_1^T(A+E)\hat{V}_1$ that

$$\exists\, \lambda_i^{(1)} \ \text{ such that } \ |\hat{\lambda}_j^{(1)} - \lambda_i^{(1)}| \leq \varepsilon + \frac{2\varepsilon^2}{\delta}. \tag{4.19}$$

In particular, this gives

$$\lambda_{\max}(\,\hat{V}_1^T(A+E)\hat{V}_1\,) \leq \lambda_{k1}^{(1)} + \varepsilon + \frac{2\varepsilon^2}{\delta}. \tag{4.20}$$

□

Our numerical experiments confirm our conjecture except for a few Gauss-Newton 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, though not always small enough to guarantee that the individual clusters of those eigenvalues do not merge. However, more importantly, the $\|E\|$ is sufficiently large that we cannot preclude the invariant subspaces corresponding to large eigenvalues from perturbing those corresponding to the smallest ones. Note that the problem is too large to actually check the projection of E on invariant subspaces corresponding to medium or larger eigenvalues. However, Figures 6.2 and 6.7 include examples with relatively large canonical angles corresponding to systems at the start of a line search.

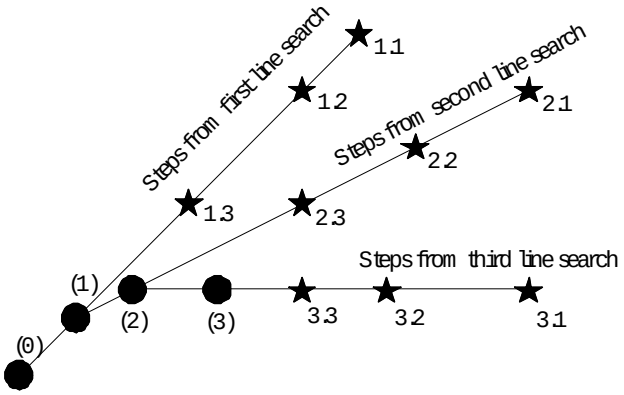

FIG. 5.1. *Several steps of the Gauss-Newton with Line Search algorithm. The black circles denote the end of a line search (and one the start of the overall procedure); the black stars denote the first and intermediate steps from the line search.*

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 5.1. 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, to-wards 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 how large the line search parameter is. We can also update this additional search space as we go. In future work, we will explore the use of GCROT-like techniques to better measure the effectiveness of subspaces. Strategies based on this approach turn out to be very effective; we will give more details in this section where we discuss the application and the optimization algorithm. The idea to use previous solutions to accelerate convergence of a Krylov method was also proposed in [13], although in [13] they were only used to provide a better initial guess.

The recycling algorithm 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. To keep the notation as simple as possible, we begin by reviewing the basic GCRO algorithm, then we discuss modifications that allow us to solve for multiple right-hand sides and multiple frequencies.

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 (c.f. discussion in Section 5.2). The recycled-GCRO method proceeds as follows.

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.
   (f) update x, r
   (g) **If** (at end of LS), $x^{(curr)} = x_1^{(j)}$
   (h) **If** (at beginning of LS), $x^{(beg)} = x_1^{(j)}$

Here, W_{ind} indicates that we may wish to keep fewer approximate eigenvectors according to the discussion preceding the algorithm (see 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 the 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 (3e) to converge as if the smallest eigenvalues have been deflated from $A^{(j)}$. In fact, Theorem 4.1 from

[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.

[22] (which closely follows Theorem 2.1 in [30]) says that this should be the case.

Next, we consider the effect of keeping a previous solution $x^{(curr)}$, in U. Assume the number of columns in the current U matrix is J. For ease of discussion, let us permute the columns of U so that the first column of U is $x^{(curr)}$ (notice the ordering is unimportant, since it does not change the orthogonal projector).

At the end of (3b), we observe that

$$\frac{1}{\rho_1} A^{(j)} x^{(curr)} = c_1,$$

where c_1 is the first column of C. However, $x^{(curr)} = (A^{(k)})^{-1} e_{m_s}$ for some previous index k. Therefore,

$$\frac{1}{\rho_1} A^{(j)} (A^{(k)})^{-1} e_{m_s} = c_1.$$

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,

$$\begin{aligned} \tilde{v} &:= (I - c_1 c_1^T) e_{m_s} = e_{m_s} - c_1 c_1^T (\rho_1 c_1 - \tilde{E} e_{m_s}) \\ &= e_{m_s} - \rho_1 c_1 + c_1 (c_1^T \tilde{E} e_{m_s}) \end{aligned}$$

Then the initial residual is

$$r_0 = (I - C_{J-1} C_{J-1}^T) \tilde{v}$$

where C_{J-1} represents all the remaining columns in C. Thus we have

$$r_0 = (I - C_{J-1} C_{J-1}^T)(e_{m_s} - \rho_1 c_1) + (c_1^T \tilde{E} e_{m_s}) c_1.$$

But it is readily shown that

$$e_{m_s} - \rho_1 c_1 = (A^{(k)} - A^{(j)}) x^{(curr)}.$$

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. The vector $e_{m_s} - \rho_1 c_1$ should already be small in norm. Since C_{J-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, the term $c_1^T \tilde{E} e_{m_s} = -c_1^T E_1 x^{(curr)} - c_1^T E_2 x^{(curr)}$. The first term in this expression should be small in norm. Since E_2 lies predominantly in the direction corresponding to larger magnitude eigenvalues but $x^{(curr)}$ is smooth, the second term must also be small. 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. Hence, corrections to the residual occur primarily over the remaining subspace, which accounts 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. We need to solve

$$A^{(j)} x_s^{(j)} = e_{m_s}, \qquad s > 1.$$

First, we replace Step (2) 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 as we solve those eigensystems. Thus, the new step becomes

- 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
  - Update $x_s^{(1)}$ with this information
  - Add columns to W if desired.

Then, to solve for $x_s^{(j)}$ for the remaining sources, we insert a loop over the remaining sources after Step (3). 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 (3b) 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. Finally, we discuss the solution for multiple frequencies using a single Krylov subspace. During Step 3e, we have
$v_1 = (I - CC)^T e_{m_s} / \|(I - CC^T) e_{m_s}\|$ and the matrix recurrence (cf. (2.1))

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

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

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

where the last step involves the reduced QR decomposition of $[V_{m+1} \,|\, C \,|\, U]$, so that $[V_{m+1,s} \,|\, C \,|\, \hat{U}]$ is an orthogonal matrix. Notice that $[V_{m+1} \,|\, C]$ is already an orthogonal matrix.

If we restrict our approximate solutions to be in $\mathit{Range}(U) \oplus \mathit{Range}(V_m)$, then we need to solve a least squares problem

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

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

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

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

Algorithm 2a: based on solving (5.2)

For each γ:

- Solve (5.2) for $\begin{bmatrix} y \\ z \end{bmatrix}$.
- Set $x_{s,\omega}^{(j)} = V_m y + Uz$.

Care must be taken in solving the least squares problem if range(U) is very close to an invariant subspace of $A^{(j)}$. In particular, if range(C) does contain eigenvectors of $A^{(j)}$, then range(U) is an invariant subspace of $A^{(j)}$ and so $U = C\Delta$, for some Δ. In this case, the least squares problems simplifies considerably as $V_{m+1}^T U = 0$. Thus, we need to solve the least squares problem (compare to (2.5))

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

Having only to solve (5.3) is the ideal situation because then the problem can separated into two minimization problems: one for which y is solved, followed by one for which z is solved. But the solution for y requires $V_m y$, rather than y explicitly, and the term $V_m y$ can be obtained from short-term recurrences without keeping the vectors V_m around using a MINRES-type approach.

Algorithm 2b: based on solving (5.3) For each γ:

- 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 C do not span an invariant subspace of $A^{(j)}$, then we really should solve (5.2). Unfortunately, there exists no short-term recurrence for $V_m y$ in this case, so 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.

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.

One other element in Algorithm 1 must be changed to accommodate additional γ. We must change Step 3a to include one or two new columns, imag($x_{s,\omega}^{(curr)}$) (and 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 note that we also have $E = A_\omega^{(j)} - A_\omega^{(k)}$. One can show that $A^{(j)}(x_s^{(j)} - x_s^{(k)}) = Ex_s^{(k)}$ while $A_\omega^{(j)}(x_{s,\omega}^{(j)} - x_{s,\omega}^{(k)}) = Ex_{s,\omega}^{(k)}$. Assuming the real part of $x_{s,\omega}^{(k)}$ is close to $x_s^{(k)}$, the difference between these right-hand sides is primarily due to the 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. However, if γ is very large, 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. Future research includes plans for avoiding this dilemma.

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 parameteric nonlinear inversion scheme outlined previously. In the first experiment, a piecewise constant model for diffusion and absorption was used. In the second, a piecewise continuous model was used. 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[2] into 31x31x21 voxels of volume h^3, with $h = .2$cm. The sizes of the matrices in both experiments were 18,259 x 18,259. The sources and detectors were located in a 3cm x 3cm plane over the center of the grid. The starting guesses for the ellipsoids describing the anomaly were the largest possible ellipsoids fitting in the 3x3x4cm 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 [31, Corollary I.5.4],

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

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 6.1, 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 plot for the sines with $M = 3$ and $M = 12$ illustrates the relative insensitivity of the eigenspace corresponding to the smallest 3 and smallest 12 eigenvalues, respectively. The indices have no correspondence with

[2]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.

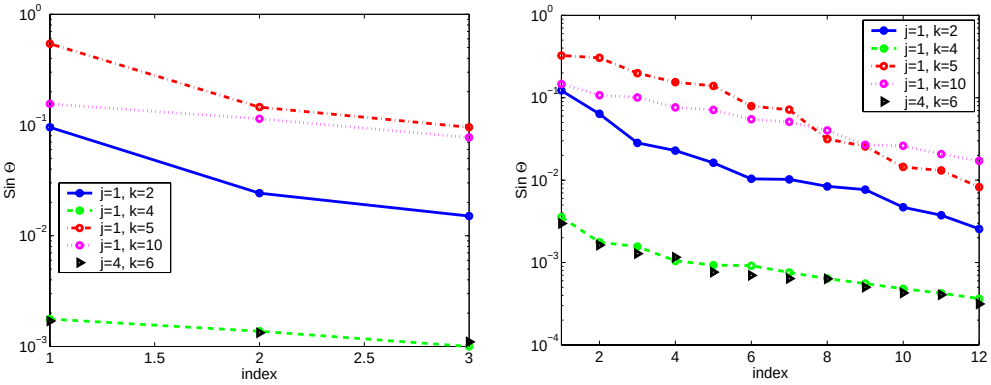

FIG. 6.1. *Experiment 1. Plots of* $\sin\Theta[Range(W^{(j)}), Range(W^{(k)})]$ *for various* (j,k) *for subspaces of dimension* $M = 3$ *(left) and* $M = 12$ *(right).*

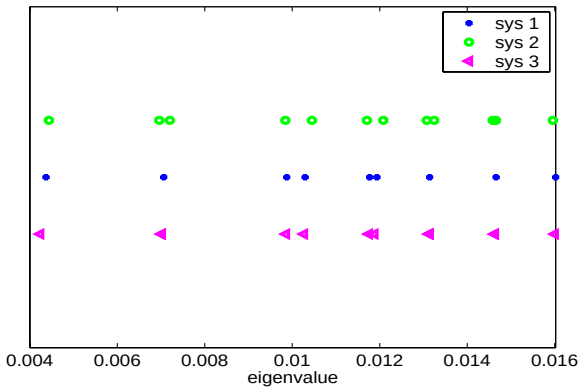

FIG. 6.2. *Experiment 1. Smallest 12 eigenvalues of* $A^{(j)}$, $j = 1, 2, 3$.

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 updates in the GN process remain fairly close to each other (e.g. 1 and 4, 4 and 6, 1 and 14), whereas those from matrices from distinct line searches (e.g. 1 and 3, 1 and 5, 1 and 13) differ more. 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 6.2.

We ran Algorithm 1, adjusted as in Section 5.3, for the multiple right-hand side problem for zero frequency and adjusted for an additional non-zero frequency of ω=5 MHz using Algorithms 2a and 2b. We saved 2 harmonic Ritz vectors from each of the first 6 right-hand sides in the initial phase of Algorithm 1 in order to try to capture an invariant subspace of $A^{(1)}$ of dimension 12 corresponding to small 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 6.3 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

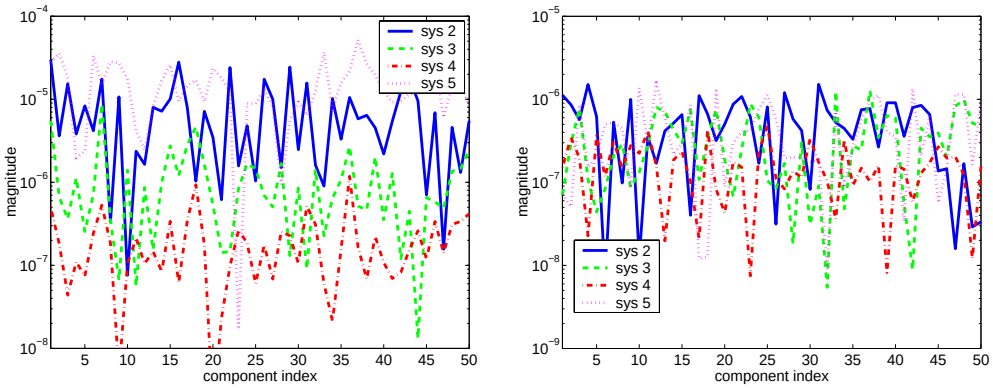

FIG. 6.3. *Experiment 1. Left: Plots of* $(W^{(j)})^T r_0$ *for* $j = 2, 3, 4, 5$ *for source 1 where* $W^{(j)}$ *corresponds to the eigenvector matrix associated with the 50 smallest eigenvalues of* $A^{(j)}$. *Right: Same, except for 50 largest eigenvalues of* $A^{(j)}$.

all from the same line search, and we observe that the spectral 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 6.4 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. 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 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 needed on the projected system. We think it is possible to design further tuning strategies to reduce the peaks in the curve for source 1 while maintaining the reduction in matrix-vector products for the other systems; this is a subject for future research.

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 6.4. Additionally, the level-3 BLAS matrix-matrix product $(A^{(j)}U)$ performed prior to running MINRES on the projected system is 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 6.5. Note that the convergence rate becomes higher and the initial relative residual norm becomes smaller as we move through one sequence of systems in a line

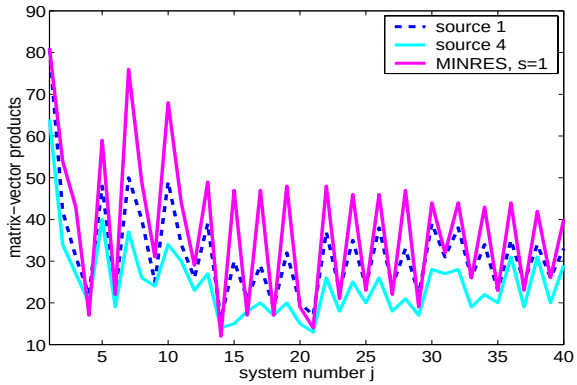

FIG. 6.4. *Experiment 1. Number of matrix-vector products per system solve for systems 1:40 and sources 1 and 4. The number of matrix-vector products for source 4 is representative of all sources after source 1. For comparison we give the number of matrix-vector products required for MINRES on the systems 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.*

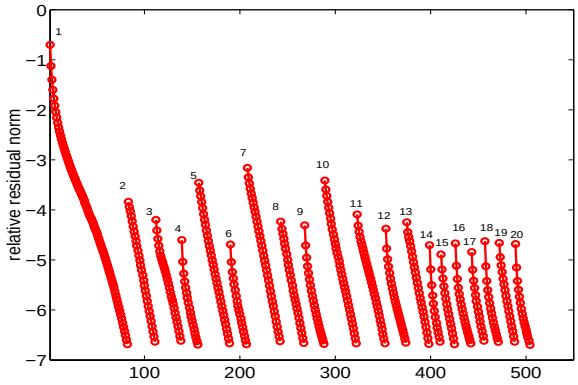

FIG. 6.5. *Experiment 1. Relative residual norms are displayed for the first 20 systems, source 1.*

search.

The plot in Figure 6.6 illustrates the relative residual norms that are achieved when solving the complex system with $\omega = 5$MHz when Algorithm 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 system. However, for our application, we feel this is sufficient. In future work, we consider alternatives for the shifted frequency case. 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 [17], so 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

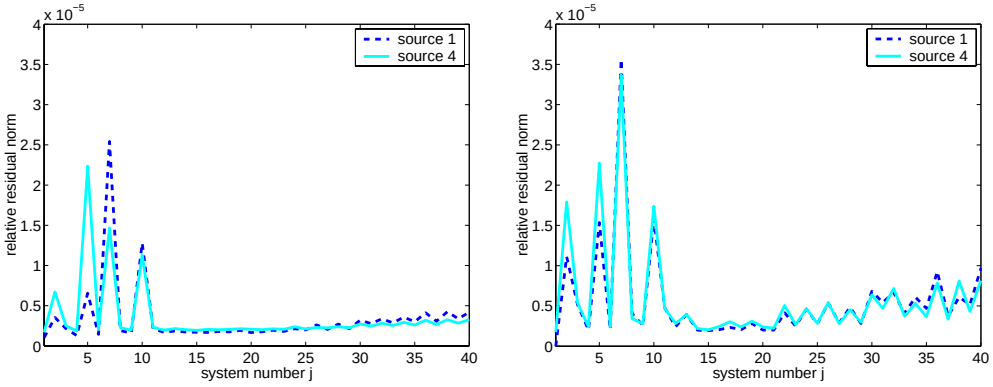

FIG. 6.6. *Experiment 1. Left: Relative residual norm per system solve, systems 1:40, sources 1 and 4, for ω=5 MHz, results computed using Algorithm 2a. Right: Relative residual norm per system solve, systems 1:40, sources 1 and 4, for ω= 5 MHz computed using Algorithm 2b.*

line search are 4,7,9,12,14-40 (even).

The sines of the canonical angles between different pairs of small-eigenvalue subspaces and for different subspace dimensions are given in Figures 6.7. In Figure 6.8 the smallest 12 eigenvalues are displayed. Consistent with Theorem 4.1 and our conjecture that the changes in the matrices are concentrated in the invariant subspaces corresponding to higher frequencies (larger eigenvalues), the smallest eigenvalues remain in disjunct clusters.

Figures 6.7 and 6.8 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 $\tilde{T}_m$ for each of the first 4 sources. In Figure 6.9, 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 system positioning 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 6.10. We observe an increased rate of convergence and smaller initial residuals in the course of a line search.

Finally, the relative residual norms for the complex systems at ω = 5MHz are given in Figure 6.11 for Algorithm 2a and 2b. In general, the behavior is more uniform for these systems as opposed to those in the first experiment due to the fact that 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 based on recycling approximate invariant subspaces and strategies based on recycling subspaces from previous solutions. Furthermore, our algorithms are based on a careful analysis which strategy is most useful at each stage of the optimization algorithm. This analysis also takes the underlying application,

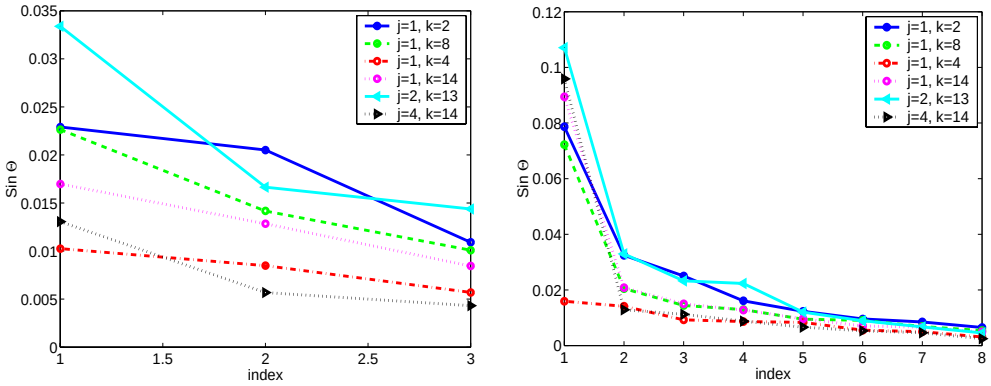

FIG. 6.7. *Experiment 2. Left: Plots of* $\sin\Theta[Range(W^{(j)}), Range(W^{(k)})]$ *for various* (j,k) *assuming a subspace dimension of* 3. *Right: Assuming a subspace dimension of* 8.

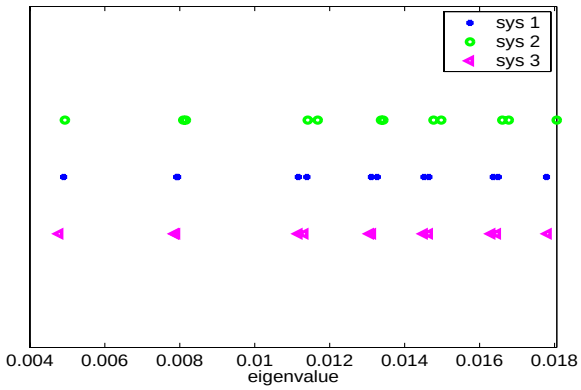

FIG. 6.8. *Experiment 2. Plot of smallest 12 eigenvalues for* $A^{(j)}$, $j = 1,2,3$. *Note that the smallest magnitude eigenvalues remain in clusters and do not cross clusters.*

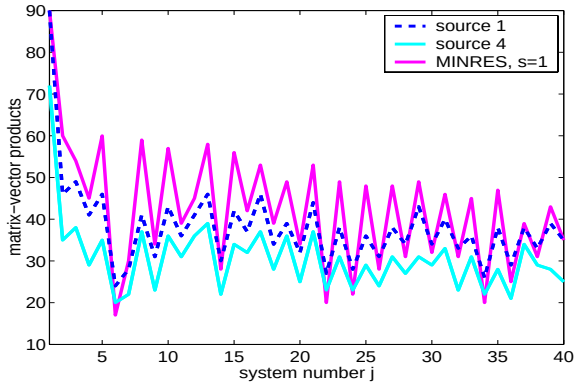

FIG. 6.9. *Experiment 2. Number of matrix-vector products per system solve for systems 1:40 and sources 1 and 4. The number of matrix-vector products for source 4 is representative of all sources after source 1. For comparison we give the number of matrix-vector products required for MINRES on the systems 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.*

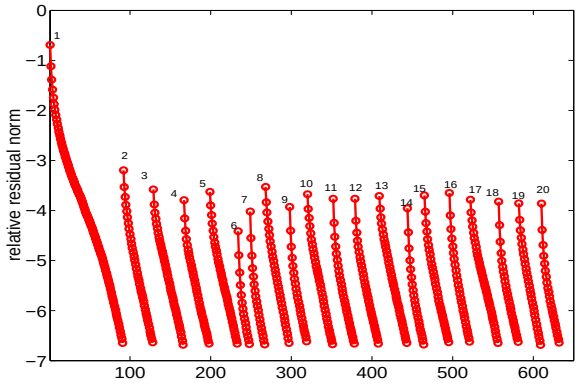

FIG. 6.10. *Experiment 2. Relative residual norms for systems 1 to 20, source 1.*

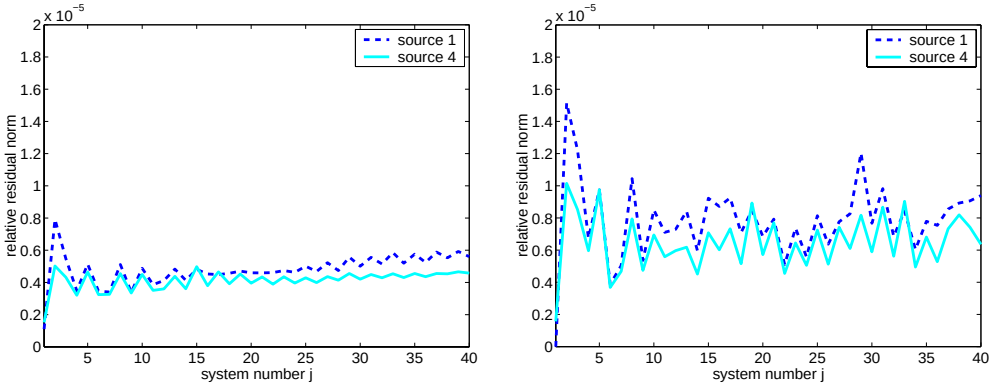

FIG. 6.11. *Experiment 2. Left: Relative residual norms for systems 1 to 40, sources 1 and 4, ω=5 MHz using Algorithm 2a. Right: Relative residual norms for systems 1 to 40, sources 1 and 4, ω=5 MHz using Algorithm 2b.*

diffuse optical tomography, and matrix symmetry into account. Furthermore, 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 diffuse optical tomography, 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 in this area is to study how characteristics of the linear systems arising in diffuse optical tomography, 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 large sequence of slowly changing problems, such as crack propagation [22]. Future work will combine modeling aspects from applications with matrix theory.

REFERENCES

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

[2] A. H. Baker, E. R. Jessup, and T. Manteuffel, *A technique for accelerating the convergence of restarted GMRES*, Tech. Report CU-CS-945-03, Department of Computer Science, University of Colorado at Boulder, 2003.

[3] 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.

[4] M. Celia and W. Gray, *Numerical Methods for Partial Differential Equations*, Prentice Hall, 1992.

[5] T. F. Chan and M. K. Ng, *Galerkin projection methods for solving multiple linear systems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 836–850.

[6] 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.

[7] 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, June 17 - June 26, 1996, Pontresina, Switzerland, 1996, pp. 193–196.

[8] ———, *Nested Krylov methods based on GCR*, Journal of Computational and Applied Mathematics, 67 (1996), pp. 15–41.

[9] ———, *Truncation strategies for optimal Krylov subspace methods*, SIAM Journal on Numerical Analysis, 36 (1999), pp. 864–889.

[10] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for unconstrained optimization and nonlinear equations*, Classics in Applied Mathematics; 16, SIAM, Philadelphia, 1996.

[11] M. Eiermann, O. G. Ernst, and O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 261–292.

[12] C. Farhat and F.-X. Roux, *Implicit parallel processing in structural mechanics*, in Computational Mechanics Advances, J. T. Oden, ed., vol. 2 (1), North-Holland, 1994, pp. 1–124.

[13] P. F. Fischer, *Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides*, Comp. Meth. in Appl. Mech. Eng., 163 (1998), pp. 193–204.

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

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

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

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

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

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

[20] R. B. Morgan, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1112–1135.

[21] ———, *GMRES with deflated restarting*, SIAM Journal on Scientific Computing, 24 (2003), pp. 20–37.

[22] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, Tech. Report UIUCDCS-R-2004-2421/UILU-ENG-2004-1722, Department of Computer Science, University of Illinois at Urbana-Champaign, March 2004. Submitted to SIAM Journal for Scientific Computing.

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

[24] F. Risler and C. Rey, *On the reuse of Ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods*, Contemporary Mathematics, 218 (1998), pp. 334–340.

[25] ———, *Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems*, Numerical Algorithms, 23 (2000), pp. 1–30.

[26] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Statist. Comput., 14 (1993), pp. 461–469.

[27] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM Journal on Matrix Analysis and Applications, 18 (1997), pp. 435–449.
[28] Y. Saad and M. H. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.
[29] 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.
[30] V. Simoncini and D. Szyld, *On the superlinear convergence of exact and inexact Krylov subspace methods*, Technical Report 03-3-13, Temple University, March 2003.
[31] G. Stewart and J. Sun, *Matrix perturbation theory*, Academic Press Inc., Boston, 1990.
[32] H. A. Van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, Journal of Computational and Applied Mathematics, 48 (1993), pp. 327–341.
[33] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
[34] 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*, Tech. Report Tech. Report No. UIUCDCS-R-2002-2299 and UILU-ENG-2002-1742, Department of Computer Science, University of Illinois at Urbana-Champaign, October 2002.