SIAM J. SCI. COMPUT.
Vol. 22, No. 5, pp. 1840–1864

KRYLOV SUBSPACE ESTIMATION*

MICHAEL K. SCHNEIDER[†] AND ALAN S. WILLSKY[†]

Abstract. Computing the linear least-squares estimate of a high-dimensional random quantity given noisy data requires solving a large system of linear equations. In many situations, one can solve this system efficiently using a Krylov subspace method, such as the conjugate gradient (CG) algorithm. Computing the estimation error variances is a more intricate task. It is difficult because the error variances are the diagonal elements of a matrix expression involving the inverse of a given matrix. This paper presents a method for using the conjugate search directions generated by the CG algorithm to obtain a convergent approximation to the estimation error variances. The algorithm for computing the error variances falls out naturally from a new estimation-theoretic interpretation of the CG algorithm. This paper discusses this interpretation and convergence issues and presents numerical examples. The examples include a 10^5-dimensional estimation problem from oceanography.

Key words. Krylov subspaces, linear least-squares estimation, error variances, conjugate gradient

AMS subject classifications. 65U05, 65F10, 93E10

PII. S1064827599357292

1. Introduction.

The goal of finite-dimensional linear least-squares estimation is to estimate an l-dimensional random vector x with a linear function of another m-dimensional random vector y so as to minimize the mean squared error [11, Chapter 4]. That is, one minimizes $\mathrm{E}[\|x-\hat{x}(y)\|^2]$ over $\hat{x}(y)$ to find the linear least-squares estimate (LLSE).

Write the relationship between x and y as $y = z+n$, where n is noise, uncorrelated with x, and

$$z = Cx \tag{1.1}$$

for a matrix C reflecting the type of measurements of x. In the Bayesian framework, x, z, and n have known means and covariances. The covariance matrices are denoted by Λ_x, Λ_z, and Λ_n, respectively, and, without loss of generality, the means are assumed to be zero. The LLSE of x given y is

$$\hat{x}(y) = \Lambda_x C^T \Lambda_y^{-1} y, \tag{1.2}$$

where $\Lambda_y = \Lambda_z + \Lambda_n = C\Lambda_x C^T + \Lambda_n$ is the covariance of y.

Direct computation of $\hat{x}(y)$ is difficult if x and y are of high dimension. In particular, the work in this paper was motivated by problems in which x represents a spatially distributed phenomenon and y represents measurements encountered in applications ranging from image processing to remote sensing. For example, when x and y represent natural images, they typically consist of $256 \times 256 = 65536$ pixels.

*Received by the editors June 17, 1999; accepted for publication (in revised form) August 30, 2000; published electronically January 31, 2001. This material is based upon work supported by a National Science Foundation graduate research fellowship, by ONR grant N00014-00-1-1004, and by AFOSR grant F49620-00-0362. A very preliminary paper describing the Krylov subspace estimation algorithm appeared as *A Krylov subspace method for large estimation problems* in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE, New York, NY, 1999.

http://www.siam.org/journals/sisc/22-5/35729.html

[†] Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Office 35-439, Cambridge, MA 02139 (mikesch@mit.edu, willsky@mit.edu).

In problems from physical oceanography, the dimensions of x and y are typically upwards of 10^5 and 10^4, respectively (e.g., see [7]). Furthermore, in applications such as remote sensing in which the measurement sampling pattern is highly irregular, Λ_z is typically a full matrix that is far from Toeplitz. This prevents one from solving the linear system (1.2) by spectral or sparse matrix methods. However, Λ_y often has a considerable amount of structure. For example, the covariance, Λ_x, of the full spatial field, is often either Toeplitz or well-approximated by a very sparse matrix in an appropriate basis, such as a local cosine basis [12]. The measurement matrix C is often sparse, and the noise covariance Λ_n is often a multiple of the identity. Thus, multiplying vectors by Λ_y is often efficient, and an iterative method for solving linear systems that makes use of multiplications by Λ_y, such as a Krylov subspace method, could be used to compute $\hat{x}(y)$.

For linear least-squares estimation problems, one is interested not only in computing the estimates but also some portion of the estimation error covariance matrix. The covariance of the estimation error,

$$\Lambda_{e_x}(y) = \Lambda_x - \Lambda_x C^T \Lambda_y^{-1} C \Lambda_x, \tag{1.3}$$

is the difference between the prior covariance and the error reduction. The terms on the diagonal of this matrix are the estimation error variances, the quantities most sought after for characterizing the errors in the linear estimate. A natural question to ask is whether a Krylov subspace method for computing the linear estimate $\hat{x}(y)$, such as the method of conjugate gradients (CG), could be adapted for computing portions of the error covariance matrix. This paper presents an interpretation of CG in the context of linear least-squares estimation that leads to a new algorithm for computing estimation error variances.

Many researchers in the geosciences have used CG for computing LLSEs. In particular, Bennett, Chua, and Leslie [1, 2, 3] and da Silva and Guo [4] use CG for computing LLSEs of atmospheric variables. The structures of these estimation problems are very similar to the ones considered here. In particular, the quantities to be estimated, x, are spatially varying processes, and the measurement matrices, C, are sparse. However, they do not consider using a Krylov subspace method for the computation of error variances. We not only propose such a method in this paper but also provide a detailed convergence analysis.

Paige and Saunders [13] and Xu et al. [19, 20, 21, 22] have developed Krylov subspace methods for solving statistical problems that are closely related to linear least-squares estimation. The LSQR algorithm of Paige and Saunders solves a regression problem and can compute approximations to the standard errors. The regression problem is a more general version of linear least-squares estimation in which a prior model is not necessarily specified. In the special case of linear least-squares estimation, the standard errors of the regression problem are the estimation error variances. Thus, LSQR can compute approximations to the error variances. The novelty of our work is that it focuses specifically on linear least-squares estimation and takes advantage of the structure inherent in many prior models for image processing problems. In particular, many such prior models imply a covariance of the data, $\Lambda_y = \Lambda_z + \Lambda_n$, in which the signal covariance matrix, Λ_z, has eigenvalues that decay rapidly to zero and the noise covariance matrix, Λ_n, is a multiple of the identity. Such properties are exploited by our algorithm. These assumptions were also made in the work of Xu, Kailath, et al. for signal subspace tracking. For that problem, one is interested in computing the dominant eigenvectors and eigenvalues of Λ_z. Although computing the dominant eigenvectors and eigenvalues of Λ_z is sufficient to compute an approximation to the estimation error variances, it is not necessary. We do not explicitly compute eigenvectors or eigenvalues. This provides us with the opportunity to exploit preconditioning techniques in a very efficient manner.

Section 2 discusses our interpretation of CG as used to compute LLSEs. This naturally leads to the presentation of a new iterative algorithm for computing estimation error variances. Section 3 proposes two alternative stopping criteria. The main convergence result is presented in section 4. Techniques for accelerating convergence, including preconditioned and block algorithmic forms, are discussed in section 5. The main convergence result is proved in section 6. Finally, section 7 illustrates the proposed techniques with various numerical examples.

2. The estimation algorithm.

The primary difficulty in computing the LLSE $\hat{x}(y)$ in (1.2) is the large dimension of the data y. The signal in the data, however, typically lies in a much lower dimensional subspace. One can take advantage of this fact to compute an approximation to $\hat{x}(y)$ by computing, instead of $\hat{x}(y)$, the LLSE of x given a small number of linear functionals of the data, $p_1^T y, p_2^T y, \ldots, p_k^T y$. For a particular sequence of linearly independent linear functionals, $p_1^T, p_2^T, \ldots, p_k^T$, let $\hat{x}_k(y)$ denote the LLSE of x given $p_1^T y, p_2^T y, \ldots, p_k^T y$. If most of the signal components in y lie in the span of $p_1, p_2, \ldots, p_k$, then the estimate $\hat{x}_k(y)$ approximates $\hat{x}(y)$. In this case, the covariance of the error in the estimate $\hat{x}_k(y)$, $\Lambda_{e_x,k}(y) \triangleq \mathrm{Cov}(x - \hat{x}_k(y))$, approximates the optimal error covariance, $\Lambda_{e_x}(y) \triangleq \mathrm{Cov}(x - \hat{x}(y))$.

The principal novelty of the algorithm we propose in this paper is the use of linear functionals that form bases for Krylov subspaces. The use of Krylov subspaces for solving linear algebra problems is not new, but the application of Krylov subspaces to the computation of error covariances is new. A Krylov subspace of dimension k, generated by a vector s and the matrix Λ_y, is the span of $s, \Lambda_y s, \ldots, \Lambda_y^{k-1} s$ and is denoted by $\mathcal{K}(\Lambda_y, s, k)$ [8, section 9.1.1]. The advantage of using linear functionals that form bases for Krylov subspaces is twofold. One reason is theoretical. Specifically, one can consider the behavior of the angles between $\mathcal{K}(\Lambda_y, s, k)$ and the dominant eigenvectors, u_i, of Λ_y: $\arcsin \|(I - \pi_k)u_i\| / \|u_i\|$, where π_k is the orthogonal projection onto $\mathcal{K}(\Lambda_y, s, k)$. As noted in [16], these angles are rapidly decreasing as k increases. Thus, linear functionals from Krylov subspaces will capture most of the dominant components of the data. Another reason for using functionals from Krylov subspaces is computational. As discussed in the introduction, the structure of Λ_y in many problems is such that multiplying a vector by Λ_y is efficient. A consequence of this fact is that one can generate bases for the Krylov subspaces efficiently.

The specific linear functionals used in this paper are the search directions generated by standard CG for solving a linear system of equations involving the matrix Λ_y. The conjugate search directions, $p_1, \ldots, p_k$, form a basis for $\mathcal{K}(\Lambda_y, s, k)$ and are Λ_y-conjugate [8, section 10.2]. The Λ_y-conjugacy of the search directions implies that $\mathrm{Cov}(p_i^T y, p_j^T y) = \delta_{ij}$; so, these linear functionals of the data are white. Thus, we can draw the novel conclusion that CG whitens the data. The whiteness of the linear functionals of the data allows one to write

$$\hat{x}_k(y) = \sum_{j=1}^{k} \left(\Lambda_x C^T p_i\right) p_i^T y, \tag{2.1}$$

$$\Lambda_{e_x,k}(y) = \Lambda_x - \sum_{j=1}^{k} \left(\Lambda_x C^T p_i\right) \left(\Lambda_x C^T p_i\right)^T, \tag{2.2}$$

which follows from $\mathrm{Cov}(p_1^T y, \ldots, p_k^T y) = I$.[1] One can now write recursions for the estimates and error variances in terms of the quantities $b_{y,k} = \Lambda_x C^T p_k$. We call these the *filtered backprojected* search directions because the prior covariance matrix Λ_x typically acts as a low-pass filter and C^T is a backprojection (as the term is used in tomography) since C is a measurement matrix. In terms of the $b_{y,k}$, the recursions have the following form:

$$\hat{x}_k(y) = \hat{x}_{k-1}(y) + b_{y,k} p_k^T y, \tag{2.3}$$

$$(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - ((b_{y,k})_i)^2, \tag{2.4}$$

with initial conditions

$$\hat{x}_0(y) = 0, \tag{2.5}$$

$$(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii}, \tag{2.6}$$

where $i = 1, \ldots, l$. Unfortunately, the vectors $p_1, p_2, \ldots$ generated by standard CG are not Λ_y-conjugate to a reasonable degree of precision because of the numerical properties of the method.

The numerical difficulties associated with standard CG can be circumvented using a Lanczos iteration, combined with some form of reorthogonalization, to generate the conjugate search directions [8, sections 9.1 and 9.2]. The Lanczos iteration generates a sequence of vectors according to the following recursion:

$$\alpha_k = q_k^T \Lambda_y q_k, \tag{2.7}$$

$$h_k = \Lambda_y q_k - \alpha_k q_k - \beta_k q_{k-1}, \tag{2.8}$$

$$\beta_{k+1} = \|h_k\|, \tag{2.9}$$

$$q_{k+1} = \frac{h_k}{\beta_{k+1}}, \tag{2.10}$$

which is initialized by setting q_1 equal to the starting vector s, $q_0 = 0$, and $\beta_1 = 0$. The Lanczos vectors, $q_1, q_2, \ldots$, are orthonormal and such that

$$\begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix}^T \Lambda_y \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} \tag{2.11}$$

is tridiagonal $\forall k$. Let $T_{y,k}$ denote this tridiagonal matrix, and let $L_{y,k}$ denote the lower bidiagonal Cholesky factor. Then, the vectors defined by

$$\begin{bmatrix} p_1 & p_2 & \cdots & p_k \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} L_{y,k}^{-T} \tag{2.12}$$

are equal, up to a sign, to the conjugate search directions generated by CG in exact arithmetic. That $L_{y,k}$ is lower bidiagonal allows one to use a simple one-step recursion to compute the p_i from the q_i. Note also that the $b_{y,k} = \Lambda_x C^T p_i$ can be computed easily in terms of a recursion in $\Lambda_x C^T q_i$. These latter quantities are available since the computation of q_{k+1} requires the product $\Lambda_y q_k = C(\Lambda_x C^T) q_k + \Lambda_n q_k$.

One of the main advantages to using the Lanczos iteration followed by the Cholesky factorization is that one can use a variety of reorthogonalization schemes to ensure

[1] Specifically, (2.1) and (2.2) follow from (1.2) and (1.3) with the substitution of I for Λ_y and $p_1^T C, \ldots, p_k^T C$ for the rows of C.

that the Lanczos vectors remain orthogonal and, in turn, that the associated conjugate search directions are Λ_y-conjugate. The simplest scheme is full orthogonalization [5, section 7.4]. This just recomputes h_k as

$$h_k := h_k - \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix} \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}^T h_k \tag{2.13}$$

between the steps in (2.8) and (2.9). This is typically sufficient to ensure orthogonality among the q_i. However, one can also use more complicated schemes that are more efficient such as selective orthogonalization [15]. A discussion of the details can be found in [18, Appendix B]. We have found that the type of orthogonalization used does not significantly affect the quality of the results.

Although one must use an orthogonalization scheme in conjunction with the Lanczos iteration, the added complexity is not prohibitive. Specifically, consider counting the number of floating point operations (flops) required to perform k iterations. We will assume that full orthogonalization is used and that the number of flops required to multiply vectors by Λ_y is linear in either the dimension m of the data or the dimension l of the estimate. Then, the only contribution to the flop count that is second order or higher in k, l, and m is from the orthogonalization, $2mk^2$. For comparison, consider a direct method for computing the error variances that uses Gaussian elimination to invert the symmetric positive definite Λ_y. The flop count is dominated by the elimination, which requires $m^3/3$ flops [8, p. 146]. Thus, our algorithm typically provides a gain if $k < m/6$. For many estimation problems, a reasonable degree of accuracy is attained for $k \ll m$. Some examples are given in section 7.

A summary of the steps outlined above to compute an approximation to the optimal linear least-squares estimate and associated estimation error variances is as follows.

Algorithm 2.1.

```
1. Initialize $\hat{x}_0(y) = 0$, $(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii}$ for $i = 1, \ldots, l$.
2. Generate a random vector $s$ to initialize the Lanczos iteration.
3. At each step $k$,
    (a) compute the conjugate search direction $p_k$ and filtered backprojection $b_{y,k}$
        using a reorthogonalized Lanczos iteration, and
    (b) update
        (2.14)  $\hat{x}_k(y) = \hat{x}_{k-1}(y) + b_{y,k} p_k^T y,$
        (2.15)  $(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - ((b_{y,k})_i)^2$  for $i = 1, \ldots, l$.
```

3. Stopping criteria.

A stopping criterion is needed to determine when a sufficient number of iterations has been run to obtain an adequate approximation to the error variances. Two alternative stopping criteria are proposed in this section. The first is a simple scheme that we have found works well. However, there is no systematic method for setting the parameters of the criterion to guarantee that a specified level of accuracy is achieved. The second stopping criterion is a more complicated scheme for which one can establish bounds on the approximation error. However, the criterion tends to be overly conservative in establishing the number of iterations needed to achieve a specified level of accuracy.

3.1. Windowed-maximal-error criterion.

Under this first criterion, the algorithm stops iterating after k steps if

$$\tau_{k,\varepsilon_{\min}} \triangleq \max_{k-K_{\text{win}} \le j \le k} \max_i \frac{((b_{y,j})_i)^2}{\max((\Lambda_{e_x,k}(y))_{ii}, \varepsilon_{\min})} < \varepsilon_{\text{tol}}, \tag{3.1}$$

where K_{win}, $\varepsilon_{\min}$, and $\varepsilon_{\mathrm{tol}}$ are parameters. This criterion guarantees that no components of the error variances have been altered over the last $K_{\mathrm{win}}+1$ iterations by more than $\varepsilon_{\mathrm{tol}}$ relative to the current approximation to the error variances. The motivation for this criterion is the theorem in section 4 which implies that the vectors $b_{y,k}$, representing the contribution to error reduction from $p_k^T y$, get smaller as k increases. However, this behavior is not always monotone; so, the criterion takes into account gains over a window of the last few iterations.

3.2. Noiseless-estimation-error criterion. The second stopping criterion examines how well the Krylov subspace at the kth step, $\mathcal{K}(\Lambda_y, s, k-1)$, captures the significant components of the signal z, as defined in (1.1). The motivation for such a criterion is the convergence analysis of section 4. A portion of the analysis examines the optimal error covariance for estimating z from y, $\Lambda_{e_z}(y)$, and its relation to the optimal error covariance for estimating z from $p_1^T y, \ldots, p_k^T y$, $\Lambda_{e_z,k}(y)$. The implication is that as $\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y)$ gets smaller, the difference between $\Lambda_{e_x,k}(y)$ and $\Lambda_{e_x}(y)$ also decreases, albeit possibly at a slower rate. So, a relatively small difference between $\Lambda_{e_z,k}(y)$ and $\Lambda_{e_z}(y)$ implies a relatively small difference between $\Lambda_{e_x,k}(y)$ and $\Lambda_{e_x}(y)$. This fact motivates the interest in efficiently computable bounds for $\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y)$. One such bound can be written, as follows, in terms of the error covariance for the noiseless estimation problem of estimating x from z.

PROPOSITION 3.1. *Suppose $\Lambda_n = \sigma^2 I$ for $\sigma^2 > 0$. Let $\Lambda_{e_z,k}(z)$ be the optimal estimation error covariance for estimating z from $p_1^T z, \ldots, p_k^T z$. Then, the difference between the error covariance for estimating z from y and z from $p_1^T y, \ldots, p_k^T y$ is bounded by*

$$\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y) \leq \Lambda_{e_z,k}(z) + f_k f_k^T, \tag{3.2}$$

where

$$\|f_k\|^2 \leq \|\Lambda_z p_{k-1}\|^2 + \|\Lambda_z p_k\|^2 + \|\Lambda_z p_{k+1}\|^2 + \|\Lambda_z p_{k+2}\|^2. \tag{3.3}$$

Proof. The proof makes use of the Lanczos vectors q_i discussed at the end of section 2. The Lanczos vectors are useful because they form bases for the Krylov subspaces, and they tridiagonalize both Λ_y and Λ_z since $\Lambda_n = \sigma^2 I$, by assumption. Since the Lanczos vectors tridiagonalize Λ_y, $q_i^T y$ is correlated with $q_j^T y$ if and only if i and j differ by at most one. Let $\Lambda_{r_z,k+1}(y)$ denote the error reduction obtained from estimating z with $q_{k+2}^T y, q_{k+3}^T y, \ldots$. Furthermore, let $\Lambda_{r_z,k+1}^{\perp}(y)$ denote the error reduction obtained from estimating z with the random variable formed by making $q_{k+1}^T y$ uncorrelated with $q_i^T y$ for $i \neq k+1$. Then,

$$\Lambda_{e_z}(y) - \Lambda_{e_z,k}(y) = \Lambda_{r_z,k+1}(y) + \Lambda_{r_z,k+1}^{\perp}(y). \tag{3.4}$$

Since y is simply a noisy version of z, $\Lambda_{r_z,k+1}(y) \leq \Lambda_{r_z,k+1}(z)$, where $\Lambda_{r_z,k+1}(z)$ is the error reduction obtained from estimating z with $q_{k+2}^T z, q_{k+3}^T z, \ldots$. Furthermore, $\Lambda_{r_z,k+1}(z) \leq \Lambda_{e_z,k}(z)$ because $\Lambda_{e_z}(z) = 0$ and $q_i^T z$ is uncorrelated with $q_j^T z$ if i and j differ by more than one. Combining the last two inequalities with (3.4) yields

$$\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y) \leq \Lambda_{e_z,k}(z) + \Lambda_{r_z,k+1}^{\perp}(y). \tag{3.5}$$

The matrix $\Lambda_{r_z,k+1}^{\perp}(y)$ in (3.5) is bounded above by the optimal error reduction for estimating z from $q_k^T y$, $q_{k+1}^T y$, and $q_{k+2}^T y$ since $\Lambda_{r_z,k+1}^{\perp}(y)$ is the error reduction for an estimator that is linear in these three functionals of y. Furthermore, $\Lambda^{\perp}_{r_z,k+1}(y)$ is bounded above by the optimal error reduction for estimating z from $p_{k-1}^T y, \ldots, p_{k+2}^T y$ since q_k, q_{k+1}, and q_{k+2} are linear combinations of $p_{k-1}, \ldots, p_{k+2}$. Now, write the rank-one matrix $\Lambda^{\perp}_{r_z,k+1}(y)$ as $f_k f_k^T$. Then, the latter bound on $\Lambda^{\perp}_{r_z,k+1}(y)$ implies (3.3). □

Although Proposition 3.1 provides a bound on $\|f_k\|^2$, the argument in the proof suggests that the bound is very weak. Recall from the proof that $f_k f_k^T = \Lambda^{\perp}_{r_z,k+1}(y)$, the error reduction obtained for estimating z from the random variable formed by making $q_{k+1}^T y$ uncorrelated with $q_k^T y$ and $q_{k+2}^T y$. Both q_k and q_{k+2}, as vectors from a Krylov subspace generated by Λ_y, are such that $q_k^T y$ and $q_{k+2}^T y$ are significantly correlated with z. Thus, making $q_{k+1}^T y$ uncorrelated with $q_k^T y$ and $q_{k+2}^T y$ will often significantly reduce the correlation of the resulting quantity with z. As a result, $\Lambda^{\perp}_{r_z,k+1}(y)$ is typically much smaller than the error reduction for estimating z from $q_{k+1}^T y$ alone, which, in turn, is smaller than the right-hand side of (3.3). Thus, the bound on $\|f_k\|^2$ is weak, and $\Lambda_{e_z,k}(z)$, the dominant term in (3.2), could be used alone as the basis of a stopping criterion.

One of the main advantages of the bound in Proposition 3.1 is that the diagonal elements of $\Lambda_{e_z,k}(z)$ are easily computable. As discussed in the proof of Proposition 3.1, the Lanczos vectors $q_1, q_2, \ldots$ generated by Algorithm 2.1 not only tridiagonalize Λ_y; they also tridiagonalize Λ_z:

$$\begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix}^T \Lambda_z \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} = T_{z,k}. \tag{3.6}$$

Let $L_{z,k}$ be the lower bidiagonal Cholesky factor of $T_{z,k}$, and let the vectors $r_1, r_2, \ldots$ be defined by

$$\begin{bmatrix} r_1 & r_2 & \cdots & r_k \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} L_{z,k}^{-T}. \tag{3.7}$$

Then, the linear functionals of the signal $r_1^T z, r_2^T z, \ldots$ are white. So, a simple recursion can be used to compute $\Lambda_{e_z,k}(z)$:

$$(\Lambda_{e_z,k}(z))_{ii} = (\Lambda_{e_z,k-1}(z))_{ii} - ((b_{z,k})_i)^2 \tag{3.8}$$

with the initialization

$$(\Lambda_{e_z,0}(z))_{ii} = (\Lambda_z)_{ii}, \tag{3.9}$$

where $i = 1, \ldots, m$ and $b_{z,k} = \Lambda_z r_k$. Note that $b_{z,k}$ can be computed without an additional multiplication by Λ_z since Algorithm 2.1 computes $\Lambda_z q_i$. The computations for calculating $\Lambda_{e_z,k}(z)$ are summarized as follows.

```
Algorithm 3.2.
  1. Initialize $(\Lambda_{e_z,0}(z))_{ii} = (\Lambda_z)_{ii}$.
  2. At each iteration $k$:
      (a) compute $b_{z,k}$ using $q_k$ and the one-step recursion specified by $L_{z,k}^T$, and
      (b) update
          (3.10)    $(\Lambda_{e_z,k}(z))_{ii} = (\Lambda_{e_z,k-1}(z))_{ii} - ((b_{z,k})_i)^2$.
```

Stopping Algorithm 2.1 when a function of $(\Lambda_{e_z,k}(z))_{ii}$ falls below some threshold has a variety of advantages and disadvantages. Although it may appear that one of the main disadvantages is the requirement that Λ_n must be a multiple of the identity, this is not the case. There is an extension to the nonwhite case that makes use of preconditioning ideas, as discussed in section 5. In fact, the main disadvantage stems from the bound in Proposition 3.1 being based on the noiseless estimation problem (i.e., $\Lambda_n = 0$). If Λ_n is not small, the bound may not be tight. Thus, a stopping criterion based on $\Lambda_{e_z,k}(z)$ may be conservative in determining the number of iterations needed to guarantee a specified level of accuracy. On the other hand, the bound is easy to compute and provides a good indication of the fraction of error reduction that has been attained by a specific iteration.

4. The main convergence result. In this section, we state the main convergence result. It establishes a bound on the rate at which the approximation to the error variances, in exact arithmetic, converges to the optimal estimation error variances. The result leads naturally to a consideration of the two acceleration techniques discussed in the next section. The proof of the main result is left for section 6.

Establishing the convergence result requires making a few assumptions concerning the estimation problem and starting vector for the algorithm. The first is that the starting vector s in Algorithm 2.1 is a zero-mean Gaussian random vector. This assumption is needed to guarantee the independence of uncorrelated components of s. The covariance matrix of s, Λ_s, is assumed to equal Λ_y or to be proportional to the identity. As regards the estimation problem for the purposes of this section, Λ_n is not necessarily a multiple of the identity. However, we do assume that Λ_y and Λ_z have the same eigenvectors $u_1, u_2, \ldots, u_m$ and that the corresponding eigenvalues $\lambda_{y,1} \geq \lambda_{y,2} \geq \cdots \geq \lambda_{y,m}$ and $\lambda_{z,1} \geq \lambda_{z,2} \geq \cdots \geq \lambda_{z,m}$ satisfy the inequality, $\lambda_{z,i}/\lambda_{y,i} \leq \bar{\lambda}_i/\sigma^2$ for some $\sigma^2 > 0$ and sequence $\bar{\lambda}_i$. Note that both of these statements would hold for $\bar{\lambda}_i = \lambda_{z,i}$ if Λ_n were $\sigma^2 I$. The conditions are stated this generally because Λ_n may not be a multiple of the identity if some of the preconditioning techniques of section 5.1 are used. We also assume that the eigenvalues of Λ_y are distinct and have a relative separation $(\lambda_{y,i} - \lambda_{y,i+1})/(\lambda_{y,i+1} - \lambda_{y,m})$ that is bounded below by a constant $\lambda_{\text{sep}} > 0$. Furthermore, the $\lambda_{y,i}$ are assumed to decrease slowly enough (not faster than a geometric decay) that one can find constants $\zeta > 0$ and $0 < \Gamma < 1$ of reasonable magnitude (ζ not much larger than $\|\Lambda_y\|$) for which $1/(\lambda_{y,k}\gamma^k) < \zeta\Gamma^k$, where

$$\gamma \triangleq 1 + 2\left(\lambda_{\text{sep}} + \sqrt{\lambda_{\text{sep}} + \lambda_{\text{sep}}^2}\right). \tag{4.1}$$

This last assumption is a very weak assumption that is almost never violated. All of these assumptions concerning the estimation problem are not restrictive because they can be guaranteed using appropriate preconditioning techniques, as described in section 5. The assumptions are summarized as follows.

Assumptions.

1. The starting vector s in Algorithm 2.1 is a zero-mean Gaussian random vector, and $\Lambda_s = \Lambda_y$ or $\Lambda_s \propto I$.
2. There exist constants $\zeta > 0$ and $0 < \Gamma < 1$ such that $1/(\lambda_{y,k}\gamma^k) < \zeta\Gamma^k$.
3. Λ_y and Λ_z have the same eigenvectors.
4. There exist a constant $\sigma^2 > 0$ and a sequence $\bar{\lambda}_i$ such that $\lambda_{z,i}/\lambda_{y,i} \leq \bar{\lambda}_i/\sigma^2$.
5. There exists a constant $\lambda_{\text{sep}} > 0$ such that $(\lambda_{y,i} - \lambda_{y,i+1})/(\lambda_{y,i+1} - \lambda_{y,m}) \geq \lambda_{\text{sep}} > 0$.

These assumptions lead to the main convergence result, as stated next in Theorem 4.1. The theorem consists of two bounds, one concerning the error variances for estimating x, and one concerning the error variances for estimating only the measured components of x, $z = Cx$. Two bounds are given because one may need fewer iterations to obtain a good estimate of z than of x. Moreover, the rate of convergence of the error variance for estimating z is of interest since z is often a subsampled version of x.[2]

THEOREM 4.1. *If Assumptions* 1–5 *hold, then*

$$\sum_{j=1}^{m} (\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{jj} \leq \frac{\|s\|^2 \zeta \eta \|\Lambda_x\| \|\Lambda_y\|}{\sigma^2 (1 - \frac{1}{\gamma^2})(1 - \frac{1}{\sqrt[4]{\gamma}})} \gamma^{-k/4} + \frac{\|\Lambda_x\|}{\sigma^2} \sum_{i=k}^{m-1} (i - k + 4) \bar{\lambda}_{\lfloor \frac{i}{4} \rfloor} \tag{4.2}$$

and

$$\begin{aligned} \sum_{j=1}^{m} (\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y))_{jj} \leq {} & \frac{\|s\|^2 \zeta \eta \|\Lambda_y\|}{(1 - \frac{1}{\gamma^2})(1 - \frac{1}{\sqrt{\gamma}})} \gamma^{-k/2} \\ & + \sum_{i=k}^{m-1} (i - k + 4) \min \left(\frac{\bar{\lambda}_{\lfloor \frac{i}{4} \rfloor} \lambda_{z, \lfloor \frac{i}{4} \rfloor}}{\sigma^2}, \bar{\lambda}_{\lfloor \frac{i}{4} \rfloor} \right), \end{aligned} \tag{4.3}$$

where γ is given by (4.1) *and η is a random variable whose statistics depend only on* λ_{sep}, γ, *and* Γ.

The bounds in Theorem 4.1 provide a characterization of the difference between the optimal error variances and the computed approximation. The bounds are a sum of two terms. The first terms on the right-hand sides of (4.2) and (4.3) characterize how well the Krylov subspaces have captured the dominant components of Λ_y. The bigger λ_{sep} is, the larger γ is, and the smaller the first terms in (4.2) and (4.3) become. Thus, the more separated the eigenvalues (as measured by λ_{sep}) are, the better the algorithm will perform. The second term is a sum of bounds $\bar{\lambda}_i$ on the ratio of eigenvalues $\lambda_{z,i}/\lambda_{y,i}$. The sum is over those $\bar{\lambda}_i$ corresponding to eigenvectors of Λ_z that are not well captured by the Krylov subspaces at step k. Note that the sum is over the more rapidly decreasing $\bar{\lambda}_i \lambda_{z,i}$ in (4.3).

The bounds are useful principally for two reasons. First, they indicate how the errors will scale as s, σ^2, $\|\Lambda_x\|$, $\|\Lambda_y\|$, and the eigenvalues of Λ_z change. In particular, note that the only dependence on the starting vector s is through the norm $\|s\|$. Thus, the performance of the algorithm does not depend strongly on s. Second, the bounds indicate that the rate of convergence can be increased by transforming the estimation problem in order to make γ big enough so that the second terms in (4.2) and (4.3) dominate. Such transformations are discussed next in section 5.1.

5. Techniques for improving convergence properties.

This section presents two different techniques for improving the convergence properties of the proposed algorithm for computing error variances. These techniques can be used to guarantee convergence in the case that a given estimation problem violates any of the assumptions of Theorem 4.1. One can also use these techniques to increase γ so as to improve the theoretical convergence rates.

5.1. Preconditioning.

In the estimation context, preconditioning consists of determining an invertible transformation B such that estimating x from the transformed data By can be theoretically done more efficiently by the proposed algorithm

[2] That the two bounds differ is a consequence of the fact that, for a given number of iterations k, we are not computing the best k linear functionals of the data for estimating x.

than estimating x directly from y. This will be the case if the covariances of the transformed data, $B\Lambda_y B^T$, and of the transformed signal, $B\Lambda_z B^T$, satisfy Assumptions 3 and 5 of Theorem 4.1 but Λ_y and Λ_z do not. The convergence properties will also be improved if γ for the transformed problem is higher than for the untransformed problem. The principal novelty of the preconditioning approaches described here is that they focus on these particular goals, which are very different than those of standard CG preconditioning and differ significantly from those of preconditioning for eigenvector algorithms [17, Chapter 8]. Although the goals of the preconditioning discussed here are different than for standard CG, the implementation details are very similar. In particular, explicit specification of a transformation B is not necessarily required for preconditioning techniques because preconditioning can be implemented in such a way that only multiplications by $B^T B$ are needed instead of multiplications by B and B^T.

There are three different implementations of preconditioning, each of which is mathematically equivalent in exact arithmetic. Symmetric preconditioning simply consists of applying the Krylov subspace algorithm to estimating x from $By = BCx + Bn$. Essentially, x is estimated given linear functionals from Krylov subspaces $\mathcal{K}(B\Lambda_y B^T, Bs, k)$ applied to By. There are also left and right preconditioning techniques. The following discussion focuses on right preconditioning, and analogous statements can be made concerning left preconditioning. Right preconditioning differs from symmetric preconditioning in that it involves estimating x given linear functionals from the Krylov subspaces $\mathcal{K}(\Lambda_y B^T B, s, k)$ applied to $B^T By$. Note that this is equivalent to the estimation performed in the case of symmetric preconditioning. Although $\Lambda_y B^T B$ is not symmetric, it is self-adjoint with respect to the $B^T B$ inner product. As in Algorithm 2.1, we do not compute the conjugate search directions for the preconditioned estimation problem using a standard preconditioned CG iteration. Instead, we use Lanczos iterations that compute a series of $B^T B$-conjugate vectors that tridiagonalize $B^T B\Lambda_y B^T B$, as follows:

$$\alpha_k = t_k^T \Lambda_y t_k, \tag{5.1}$$

$$h_k = \Lambda_y t_k - \alpha_k q_k - \beta_k q_{k-1}, \tag{5.2}$$

$$d_k = B^T B h_k, \tag{5.3}$$

$$\beta_{k+1} = \sqrt{d_k^T h_k}, \tag{5.4}$$

$$q_{k+1} = \frac{h_k}{\beta_{k+1}}, \tag{5.5}$$

$$t_{k+1} = \frac{d_k}{\beta_{k+1}}, \tag{5.6}$$

where $t_1 = B^T Bs$, $q_1 = s$, $q_0 = 0$, and $\beta_1 = 0$. The q_k are the $B^T B$-conjugate Lanczos vectors that tridiagonalize $B^T B\Lambda_y B^T B$, and the $t_k = B^T Bq_k$ tridiagonalize Λ_y. This latter tridiagonal matrix can be factored, as in Algorithm 2.1, to compute the Λ_y-conjugate search directions p_k. The only difference is that the t_k replace the q_k in (2.11) and (2.12). Moreover, one can compute the filtered backprojected search directions $b_{y,k} = \Lambda_x C^T p_k$ as a by-product. Overall, the steps of the preconditioned Krylov subspace algorithm are the same as those in Algorithm 2.1 except that a preconditioned Lanczos iteration replaces the normal Lanczos iteration. Note that the Lanczos method for tridiagonalizing a left-preconditioned system is the same as the generalized Lanczos algorithm for solving generalized eigenvalue problems [14, section 15.11]. What follows are some examples of preconditioners in squared up form, B^TB, that one can consider using in various contexts.

One choice for a preconditioner when the noise covariance Λ_n is not a multiple of the identity but is invertible is to choose $B^TB = \Lambda_n^{-1}$. This choice of preconditioner will effectively shape the noise covariance to be a multiple of the identity. The transformed data covariance, $B\Lambda_yB^T$, and signal covariance, $B\Lambda_zB^T$, will then satisfy Assumption 3. Multiplying a vector by Λ_n^{-1} is often easy because Λ_n is often diagonal.

If the noise covariance is, or has been transformed to be, a multiple of the identity, one can consider preconditioners that will maximally separate the eigenvalues of Λ_y. Such preconditioners can guarantee that the transformed data covariance, $B\Lambda_yB^T$, satisfies Assumption 5 and can increase γ to improve the bounds in Theorem 4.1. Note that such preconditioning will do little to change the bound $\bar{\lambda}_i$ on $\lambda_{z,i}/\lambda_{y,i}$ in Assumption 4 because the preconditioner will transform both $\lambda_{z,i}$ and $\lambda_{y,i}$. The ideal preconditioner would simply operate on the spectrum of Λ_y and force a geometric decay in the eigenvalues to the noise level σ^2. The geometric decay guarantees a constant relative separation in the eigenvalues as measured by the ratio in Assumption 5. However, operating on the spectrum is difficult because one doesn't know the eigendecomposition of Λ_y. When the rows of C are orthogonal (which is often the case in the applications mentioned in the introduction) and the eigendecomposition of Λ_x is known, one practical preconditioner is the following. Let Λ_p be a matrix whose eigenvectors are the same as those of Λ_x and whose eigenvalues decay geometrically. Then, let the preconditioner be given by $B^TB = C\Lambda_pC^T$. Although this preconditioner has worked well in practice, as described in section 7, we have no theoretical results concerning the properties of the transformed estimation problem.

One can use extensions of each of the stopping criteria of section 3 in conjunction with preconditioning; however, the preconditioner must satisfy certain assumptions for the extension of the noiseless-estimation stopping criterion of section 3.2 to be used. What follows is a discussion of the extension and the underlying assumptions concerning the preconditioner for the right-preconditioned case. Recall that the discussion in section 3.2 assumes that the noise covariance is a multiple of the identity. This assumption ensures that the Lanczos vectors tridiagonalize both Λ_y and Λ_z so that one can compute $\Lambda_{e_z,k}(z)$ efficiently. Now, suppose one is using a preconditioning transformation B. Let $\Lambda_{n'} = \Lambda_n - (B^TB)^{-1}$. Assume that $\Lambda_{n'}$ is positive semidefinite so that it is a valid covariance matrix. Let n' be a random vector with covariance $\Lambda_{n'}$ and uncorrelated with z. Then, $z' = z + n'$ has covariance $\Lambda_{z'} = \Lambda_z + \Lambda_{n'}$. One can compute $\Lambda_{e_z,k}(z')$ efficiently because the t_k in (5.1)–(5.6) tridiagonalize both Λ_y and $\Lambda_{z'}$. For $\Lambda_{e_z,k}(z')$ to be useful, the pseudosignal z' should not have any significant components not in z. Note that an example of a preconditioner satisfying the above two assumptions is given by $B^TB = \Lambda_n^{-1}$. For this preconditioner, $\Lambda_{n'} = 0$; so, $\Lambda_{e_z,k}(z) = \Lambda_{e_z,k}(z')$. Thus, one can use $\Lambda_{e_z,k}(z')$ as part of a stopping criterion in conjunction with preconditioning provided that the preconditioner satisfies the two assumptions outlined above.

5.2. Using multiple starting vectors.

Another technique for improving convergence properties in the case where Λ_y has repeated eigenvalues is to use a block form of Algorithm 2.1. Block Krylov subspace algorithms have been developed for other computations, particularly eigendecompositions [8, section 9.2.6]. The principal novelty of the algorithm we present here is the application to estimation.

Now consider the subspace spanned by the columns of

$$\begin{bmatrix} S & \Lambda_y S & \Lambda_y^2 S & \cdots & \Lambda_y^{k-1} S \end{bmatrix}, \tag{5.7}$$

where S is an $m \times r$ matrix of independent identically distributed random starting vectors whose marginal statistics satisfy the restrictions for Algorithm 2.1 starting vectors. Denote this subspace by $\mathcal{K}(\Lambda_y, S, k)$. Then, one can consider forming $m \times r$ matrices $P_1, \ldots, P_k$ whose columns form bases for $\mathcal{K}(\Lambda_y, S, k)$ and which satisfy $P_i^T \Lambda_y P_j = \delta_{ij} I$. As for the single starting vector case in section 2, the LLSE of x given the random vectors $P_1^T y, \ldots, P_k^T y$ and the associated error variances can be computing using a recursion:

$$\hat{x}_k(y) = \hat{x}_{k-1}(y) + B_{y,k} P_k^T y, \tag{5.8}$$

$$(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - \sum_{j=1}^{r} ((B_{y,k})_{ij})^2, \tag{5.9}$$

with initial conditions

$$\hat{x}_0(y) = 0, \tag{5.10}$$

$$(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii}, \tag{5.11}$$

where $i = 1, \ldots, l$ and $B_{y,k} = \Lambda_x C^T P_k$.

The P_i and $B_{y,i}$ can be computed using a reorthogonalized block Lanczos algorithm [8, section 9.2.6]. The block Lanczos iteration generates, according to the following recursions, a sequence of orthogonal matrices Q_i that are orthogonal to each other:

$$A_k = Q_k^T \Lambda_y Q_k, \tag{5.12}$$

$$H_k = \Lambda_y Q_k - Q_k A_k - Q_{k-1} R_k, \tag{5.13}$$

$$Q_{k+1} R_{k+1} = H_k \quad \text{(QR factorization of } H_k\text{)}, \tag{5.14}$$

where Q_1 and R_1 are a QR factorization of the starting matrix S, and $Q_0 = 0$. The Q_i block tridiagonalize Λ_y; so, one can write

$$\begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix}^T \Lambda_y \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix} = T_{y,k}, \tag{5.15}$$

where $T_{y,k}$ is a block tridiagonal matrix. Let $L_{y,k}$ be the lower block bidiagonal Cholesky factor of $T_{y,k}$. Then, the P_i are defined by

$$\begin{bmatrix} P_1 & \cdots & P_k \end{bmatrix} \triangleq \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix} L_{y,k}^{-T}. \tag{5.16}$$

Thus, the P_i can be computed from the Q_i using a one-step recursion. Moreover, the $B_i = \Lambda_x C^T P_i$ can be computed as a by-product, as with a single starting vector.

As for the single starting vector case in section 2, the block Lanczos iteration must be combined with some form of reorthogonalization. Unlike the previous case, however, there are not as many methods for reorthogonalizing the block Lanczos iteration. Full orthogonalization is very common and is the method we have used. This simply recomputes H_k as

$$H_k := H_k - \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix} \begin{bmatrix} Q_1 & \cdots & Q_k \end{bmatrix}^T H_k \tag{5.17}$$

between steps (5.12) and (5.13).

The algorithm is summarized as follows.

Algorithm 5.1.

1. *Initialize* $\hat{x}_0(y) = 0$, $(\Lambda_{e_x,0}(y))_{ii} = (\Lambda_x)_{ii}$ *for* $i = 1, \dots, l$.
2. *Generate a random* $m \times r$ *matrix* S *to initialize the block Lanczos iteration.*
3. *At each step* k,
    - (a) *compute the block of search directions* P_k *and filtered backprojections* $B_{y,k}$ *using a reorthogonalized block Lanczos iteration, and*
    - (b) *update*

$$\hat{x}_k(y) = \hat{x}_{k-1}(y) + B_{y,k} P_k^T y, \tag{5.18}$$

$$(\Lambda_{e_x,k}(y))_{ii} = (\Lambda_{e_x,k-1}(y))_{ii} - \sum_{j=1}^{r} ((B_{y,k})_{ij})^2 \quad \text{for } i = 1, \dots, l. \tag{5.19}$$

The advantage of using the block form is that there may be small angles between the subspaces $\mathcal{K}(\Lambda_y, S, k)$ and multiple orthogonal eigenvectors of Λ_y associated with the same repeated eigenvalue, even in exact arithmetic. This is because each of the columns of S may have linearly independent projections onto the eigenspace associated with a repeated eigenvalue. The following theorem establishes convergence rates for the block case when there may be repeated eigenvalues. It is an extension of Theorem 4.1 to the block case. The proofs of both theorems are very similar, so the proof of Theorem 5.2 is omitted.[3]

Theorem 5.2. *Suppose the following.*

1. *There exists a constant* $\lambda_{\text{sep},r} > 0$ *such that* $(\lambda_{y,i} - \lambda_{y,i+r})/(\lambda_{y,i+r} - \lambda_{y,m}) \geq \lambda_{\text{sep},r}$.
2. *There exist constants* $\zeta > 0$ *and* $0 < \Gamma < 1$ *such that* $1/(\lambda_{y,i}\gamma_r^i) < \zeta\Gamma^i$, *where*

$$\gamma_r \triangleq 1 + 2\left(\lambda_{\text{sep},r} + \sqrt{\lambda_{\text{sep},r} + \lambda_{\text{sep},r}^2}\right). \tag{5.20}$$

3. Λ_y *and* Λ_z *have the same eigenvectors.*
4. *There exist a constant* $\sigma^2 > 0$ *and a sequence* $\bar{\lambda}_i$ *such that* $\lambda_{z,i}/\lambda_{y,i} \leq \bar{\lambda}_i/\sigma^2$.
5. $(\lambda_{y,i} - \lambda_{y,i_+})/(\lambda_{y,i_+} - \lambda_{y,m})$ *is bounded away from zero, where* i_+ *is the index of the next smallest distinct eigenvalue of* Λ_y *after* i, *and then,*

$$\sum_{j=1}^{m} (\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{jj} \leq \frac{\eta \|\Lambda_x\| \|\Lambda_y\|}{\sigma^2 (1 - \frac{1}{\gamma_r^2})(1 - \frac{1}{\sqrt[4]{\gamma_r}})} \gamma_r^{-k/4} + \frac{\|\Lambda_x\|}{\sigma^2} \sum_{i=k}^{m-1} (i - k + 4) \bar{\lambda}_{\lfloor \frac{i}{4} \rfloor} \tag{5.21}$$

and

$$\sum_{j=1}^{m} (\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y))_{jj} \leq \frac{\eta \|\Lambda_y\|}{(1 - \frac{1}{\gamma_r^2})(1 - \frac{1}{\sqrt{\gamma_r}})} \gamma_r^{-k/2} + \sum_{i=k}^{m-1} (i - k + 4) \min\left(\frac{\bar{\lambda}_{\lfloor \frac{i}{4} \rfloor} \lambda_{z,\lfloor \frac{i}{4} \rfloor}^2}{\sigma^2}, \bar{\lambda}_{\lfloor \frac{i}{4} \rfloor} \right), \tag{5.22}$$

where the statistics of the random variable η *depend on the starting matrix* S.

There are two key differences between the statements of Theorems 4.1 and 5.2. The first addresses the possibility of repeated eigenvalues. Specifically, the bounds in

[3] A proof may be found in [18, Appendix A].

Theorem 5.2 depend on the eigenvalue separation through $\lambda_{\text{sep},r}$, which measures the relative separation between eigenvalues whose indices differ by r. Thus, the proposition establishes a convergence rate in the case where there may be groups of up to r repeated or clustered eigenvalues. The second key difference is that the bounds in Theorem 5.2 may have a strong dependence on the starting matrix. This contrasts with the bounds in Theorem 4.1 which depend on the starting vector s only through the norm $\|s\|$. However, our numerical results have not indicated that the block algorithm's performance depends strongly on the starting matrix S.

One can use natural extensions of the preconditioning techniques and either of the stopping criteria of section 3 with Algorithm 5.1. Thus, Algorithm 5.1 is a simple replacement for Algorithm 2.1 that can be used to obtain better convergence properties when Λ_y has repeated eigenvalues.

6. Convergence analysis.

The bounds in Theorem 4.1 are proved in this section in several steps. The first few steps place bounds on the norms of the filtered backprojected conjugate search directions, $\|\Lambda_x C^T p_i\|$ and $\|C\Lambda_x C^T p_i\|$. The bounds are proved using Saad's convergence theory for the Lanczos algorithm [16]. These bounds are stated in terms of an extremum of independent random variables. The extremum arises because the starting vector affects the angles between the Krylov subspaces and the dominant components of Λ_y. However, we prove that the extremum is part of a sequence of extrema that are converging in probability to a finite random variable (η in Theorem 4.1). Thus, the starting vector has no strong effect on the quality of the approximation to the error variances. This result is the principal novelty of our convergence analysis. After establishing the convergence of the extrema, we prove Theorem 4.1.

6.1. Bounds on the filtered backprojected search directions.

One is interested in bounding the norms of the filtered backprojected search directions because the quality of the approximation to the error variances depends on the norms as follows:

$$\sum_{j=1}^{l}(\Lambda_{e_x,k}(y)-\Lambda_{e_x}(y))_{jj} = \sum_{i=k+1}^{l}\|\Lambda_x C^T p_i\|^2, \tag{6.1}$$

$$\sum_{j=1}^{l}(\Lambda_{e_z,k}(y)-\Lambda_{e_z}(y))_{jj} = \sum_{i=k+1}^{l}\|C\Lambda_x C^T p_i\|^2. \tag{6.2}$$

PROPOSITION 6.1. *Write the conjugate search directions in the basis of eigenvectors of Λ_y as follows:*

$$p_i = v_{i,1}u_1 + \cdots + v_{i,m}u_m. \tag{6.3}$$

Then

$$\|\Lambda_x C^T p_i\|^2 \le \|\Lambda_x\|\sum_{j=1}^{m}\lambda_{z,j}v_{i,j}^2, \tag{6.4}$$

and

$$\|C\Lambda_x C^T p_i\|^2 = \sum_{j=1}^{m}\lambda_{z,j}^2 v_{i,j}^2. \tag{6.5}$$

Proof. $\|\Lambda_x C^T p_i\|^2 \leq \|\Lambda_x\| \|\Lambda_x^{1/2} C^T p_i\|^2 = \|\Lambda_x\| \sum_{j=1}^m \lambda_{z,j} \upsilon_{i,j}^2$. This proves the first inequality. The second inequality follows from Parseval's theorem. □

As we now show, one can bound the coefficients $\upsilon_{i,j}$ in terms of $\|(I-\pi_i)u_j\|/\|\pi_i u_j\|$, where π_i is the operator that produces the orthogonal projection onto $\mathcal{K}(\Lambda_y, s, i)$ with respect to the standard inner product.

Proposition 6.2. *Write* $p_i = \upsilon_{i,1}u_1 + \cdots + \upsilon_{i,m}u_m$ *as in Proposition* 6.1. *Then*

$$|\upsilon_{i+1,j}| \leq \frac{\|\Lambda_y\|^{1/2}}{\lambda_{y,j}} \frac{\|(I-\pi_i)u_j\|}{\|\pi_i u_j\|}. \tag{6.6}$$

Proof. Note that

$$\begin{aligned} \lambda_{y,j}|\upsilon_{i+1,j}| &= |p_{i+1}^T \Lambda_y u_j| \\ &= |p_{i+1}^T \Lambda_y \pi_i u_j + p_{i+1}^T \Lambda_y (I-\pi_i) u_j| \\ &= |p_{i+1}^T \Lambda_y (I-\pi_i) u_j| \end{aligned} \tag{6.7}$$

since p_{i+1} is Λ_y-conjugate to vectors in the range of π_i. Thus, $\lambda_{y,j}|\upsilon_{i+1,j}| \leq \|\Lambda_y p_{i+1}\| \cdot \|(I-\pi_i)u_j\| \leq \|\Lambda_y\|^{1/2}\|(I-\pi_i)u_j\|$ because of the Cauchy–Schwarz inequality and the fact that p_{i+1} is Λ_y-normal. The inequality in (6.6) then follows from $\|\pi_i u_j\| \leq 1$. □

The bound in Proposition 6.2 can be refined. In particular, a theorem due to Saad [16, Theorem 1] implies the following result concerning the ratio $\|(I-\pi_i)u_j\|/\|\pi_i u_j\|$, which we state without proof.

Theorem 6.3. *Let* γ *be defined by* (4.1), *and let* K_j *be defined by*

$$K_j \triangleq \begin{cases} \prod_{k=1}^{j-1} \frac{\lambda_{y,k}-\lambda_{y,m}}{\lambda_{y,k}-\lambda_{y,j}} & \textit{if } j \neq 1, \\ 1 & \textit{if } j = 1. \end{cases} \tag{6.8}$$

Then

$$\frac{\|(I-\pi_i)u_j\|}{\|\pi_i u_j\|} \leq \frac{2K_j}{\gamma^{i-j}} \frac{1}{\|\pi_1 u_j\|}. \tag{6.9}$$

Recall, from the definition of angles between subspaces given in section 2, that $\|(I-\pi_i)u_j\|/\|\pi_i u_j\|$ is the tangent of the angle between the Krylov subspace $\mathcal{K}(\Lambda_y, s, i)$ and the eigenvector u_j. Theorem 6.3 bounds the rate at which these angles decrease as the subspace dimension i increases. The bound has three components. The rate of decay is γ, the relative separation between eigenvalues as defined in (4.1). The constant in the numerator, $2K_j$, depends on the eigenvalues according to (6.8). The numerator, $\|\pi_1 u_j\|$, is the norm of the projection of the starting vector, s, onto u_j. The primary importance of the theorem is that it establishes the decay rate γ.

One can refine the bound in Proposition 6.2 by splitting the coefficients $\upsilon_{i,j}$ into two groups: those that are getting small by Proposition 6.2 and Theorem 6.3 and those that may be large but do not significantly affect $\|\Lambda_x C^T p_i\|$ because the corresponding eigenvalues of Λ_z are small. This idea leads to the following proposition.

Proposition 6.4.

$$\|\Lambda_x C^T p_{i+1}\|^2 \leq 4\|\Lambda_x\|\|\Lambda_y\| \sum_{j=1}^{\lfloor \frac{i}{4} \rfloor -1} K_j^2 \frac{1}{\gamma^{2(i-j)}\|\pi_1 u_j\|^2} \frac{\lambda_{z,j}}{\lambda_{y,j}^2} + \|\Lambda_x\| \sum_{j=\lfloor \frac{i}{4} \rfloor}^{\infty} \frac{\lambda_{z,j}}{\lambda_{y,j}}, \tag{6.10}$$

and

$$\|C\Lambda_x C^T p_{i+1}\|^2 \leq 4\|\Lambda_y\| \sum_{j=1}^{\lfloor \frac{i}{4} \rfloor - 1} K_j^2 \frac{1}{\gamma^{2(i-j)}\|\pi_1 u_j\|^2} \frac{\lambda_{z,j}^2}{\lambda_{y,j}^2} + \sum_{j=\lfloor \frac{i}{4} \rfloor}^{\infty} \frac{\lambda_{z,j}^2}{\lambda_{y,j}}. \tag{6.11}$$

Proof. The first term in each of (6.10) and (6.11) follows immediately from Propositions 6.1 and 6.2 and Theorem 6.3. The second term follows from Proposition 6.1 and the fact that $p_{i+1}^T \Lambda_y p_{i+1} = \sum_{j=1}^m \lambda_{y,j} v_{i+1,j}^2 = 1$. □

The first terms in the bounds of Proposition 6.4 may get large if $1/(\gamma^i\|\pi_1 u_j\|^2)$ or K_j are not well behaved. However, the standing assumptions concerning the eigenvalues of Λ_y, Λ_z, and Λ_s imply that K_j and $1/(\gamma^i\|\pi_1 u_j\|^2)$ are bounded by quantities of a reasonable magnitude, as we now show.

6.2. Convergence of infinite products and extrema of independent sequences. The main result regarding the convergence of infinite products and extrema of independent sequences is the following.

PROPOSITION 6.5. *Let $F_i(v)$, $i = 1, 2, \ldots,$ be a sequence of functions such that*

1. $1 - F_i(v)$ *is a cumulative distribution function, i.e., right-continuous and monotonically increasing from zero to one;*
2. *for every interval $[V, \infty)$ over which $1 - F_i(v)$ are positive, there exist a constant $A(V)$ and an absolutely summable sequence $\bar{F}_i(V)$ such that $F_i(V) \leq \bar{F}_i(V) \leq A(V) < 1$ $\forall i$; and*
3. $\lim_{v\to\infty} \sum_{i=1}^{\infty} F_i(v) = 0$.

Then, $F(v) = \prod_{i=1}^{\infty}(1 - F_i(v))$ is a distribution function. Moreover, $F(v)$ is positive over every interval such that $1 - F_i(v)$ is positive $\forall i$.

Proof. For $F(v)$ to be a distribution function, it must be right-continuous and monotonically increasing from zero to one.

Consider the interval $[V, \infty)$. Now, $\sum_{i=1}^{I} \log(1 - F_i(v))$ is right-continuous for each I since each $F_i(v)$ is right-continuous. Furthermore,

$$\begin{aligned} \left|\log(F(v)) - \sum_{i=1}^{I} \log(1 - F_i(v))\right| &= \left|\sum_{i=I+1}^{\infty} \log(1 - F_i(v))\right| \leq \left|\sum_{i=I+1}^{\infty} \log(1 - \bar{F}_i(V))\right| \\ &= \left|\sum_{i=I+1}^{\infty}\sum_{j=1}^{\infty} \frac{\bar{F}_i^j(V)}{j}\right| \leq \left|\sum_{i=I+1}^{\infty} \frac{\bar{F}_i(V)}{1 - A(V)}\right|. \end{aligned} \tag{6.12}$$

Since $\bar{F}_i(V)$ is absolutely summable, $\sum_{i=1}^{I} \log(1 - F_i(v))$ converges to $\log(F(v))$ uniformly for $v \in [V, \infty)$. Thus, $\log(F(v))$ and, in turn, $F(v)$ are right-continuous.

That $F(v)$ is monotonic follows from the monotonicity of the $1 - F_i(v)$. Now, $\lim_{v\to-\infty} F(v) = 0$ since $\lim_{v\to-\infty}(1 - F_1(v)) = 0$. Moreover,

$$\lim_{v\to\infty} \log(F(v)) \geq \lim_{v\to\infty} \sum_{i=1}^{\infty} \frac{-F_i(v)}{1 - A(V)} = 0, \tag{6.13}$$

where V is such that $1 - F_i(v)$ is positive over $[V, \infty)$ $\forall i$. So, $\lim_{v\to\infty} F(v) = 1$.

Furthermore, if $1 - F_i(v)$ is positive $\forall i$ over an interval $[V, \infty)$, then

$$\log(F(v)) \geq \sum_{i=1}^{\infty} \frac{-\bar{F}_i(V)}{1 - A(V)} > -\infty. \tag{6.14}$$

Hence, $F(v)$ is positive over every interval such that $1 - F_i(v)$ is positive $\forall i$. ◻

A particular example of such a sequence of functions $F_i(v)$ satisfying the assumptions of Proposition 6.5 is

$$F_i(v) = \begin{cases} 1, & v < 0, \\ (1-v)^i, & 0 \le v \le 1, \\ 0, & v > 1. \end{cases} \tag{6.15}$$

Thus, any product of numbers converging geometrically fast towards one is bounded away from zero, and the product is continuously varying from zero to one as the geometric rate changes from one to zero. This fact is used in the proof of the following proposition, which bounds the constants K_j.

Proposition 6.6. *There exists a function $K(v)$ which is continuous and monotonically decreasing from infinity to one as v ranges from zero to infinity and satisfies*

$$K_j \le K(\lambda_{\text{sep}}). \tag{6.16}$$

Proof.

$$\begin{aligned} \frac{1}{K_j} &= \prod_{k=1}^{j-1} \frac{\lambda_{y,k} - \lambda_{y,j}}{\lambda_{y,k} - \lambda_{y,m}} \\ &\ge \prod_{k=1}^{j-1} \left(1 - \left(\frac{1}{1+\lambda_{\text{sep}}} \right)^k \right), \end{aligned} \tag{6.17}$$

where the inequality follows from Assumption 5. By Proposition 6.5, the product is monotonically decreasing to a limit as j tends to infinity. The limit is a continuous function of λ_{sep} that varies monotonically from zero to one as λ_{sep} increases from zero to infinity. Denote the limit by $1/K(\lambda_{\text{sep}})$. Then, $K_j \le K(\lambda_{\text{sep}})$, as desired. ◻

The bound on $1/(\gamma^i \|\pi_1 u_j\|^2)$ is stochastic because $\pi_1 = s^T/\|s\|$, where s is the starting vector. By Assumption 1, one can write $\|\pi_1 u_j\|^2 = \lambda_{s,j}|w_j|^2/\|s\|^2$, where $\lambda_{s,j}$ are eigenvalues of Λ_s and w_j are independent, zero-mean, unit variance Gaussian random variables. Thus,

$$\frac{1}{\gamma^i \|\pi_1 u_j\|^2} \le \|s\|^2 \max_{1 \le k \le m} \frac{1}{\lambda_{s,k} \gamma^k |w_k|^2} \tag{6.18}$$

for $m \ge i \ge j$. Suppose that the $\lambda_{y,k}$ satisfy

$$\frac{1}{\lambda_{y,k}\gamma^k} < \zeta \Gamma^k \tag{6.19}$$

for constants $\zeta > 0$ and $0 < \Gamma < 1$. Then, (6.19) holds for $\lambda_{s,k}$ for the same ζ and Γ if $\Lambda_s = \Lambda_y$ and for a different ζ and $\Gamma = 1/\gamma$ if $\Lambda_s \propto I$. Let

$$\mu_k = \max_{1 \le j \le k} \frac{\Gamma^j}{|w_j|^2}. \tag{6.20}$$

The quantity μ_k is an extremum of an independent, nonidentically distributed sequence of random variables. Bounding the rate at which extrema grow is a classic problem in statistics [10]. The following result states that the μ_k do not grow without bound but converge in probability.

PROPOSITION 6.7. *Suppose $w_1, w_2, w_3, \dots$ is an independent sequence of zero-mean, unit variance Gaussian random variables. Let μ_k be as in* (6.20). *Then, the μ_k converge in probability to a finite-valued random variable.*

Proof. First, we show the μ_k converge in distribution.

$$\mathrm{P}\{\mu_k \le M\} = \prod_{i=1}^{k} \mathrm{P}\left\{ |w_i| \ge \sqrt{\frac{\Gamma^i}{M}} \right\}. \tag{6.21}$$

Let

$$F_i(M) = \mathrm{P}\left\{ |w_i| \ge \sqrt{\frac{\Gamma^i}{M}} \right\}. \tag{6.22}$$

Then

$$F_i(M) \le \sqrt{\frac{2}{\pi}} \sqrt{\frac{\Gamma^i}{M}}, \tag{6.23}$$

which satisfies the conditions of Proposition 6.5. Thus, $\lim_{k\to\infty} \mathrm{P}\{\mu_k \le M\} = F(M)$ for some distribution function F.

To show that the μ_k converge in probability, consider the following. For $n > k$ and $\varepsilon > 0$,

$$\mathrm{P}\{\mu_n - \mu_k > \varepsilon\} = \int \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v), \tag{6.24}$$

where G_k is the distribution of μ_k. Now

$$\begin{aligned} \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} &= \mathrm{P}\left\{ \max_{1\le j \le n-k+1} \frac{\Gamma^j}{|w_j|^2} > \frac{\varepsilon + v}{\Gamma^{k-1}} \right\} \\ &\le 1 - F\left(\frac{\varepsilon + v}{\Gamma^{k-1}}\right). \end{aligned} \tag{6.25}$$

Let V be such that $1 - F(V) < \varepsilon/2$, and let N be such that

$$1 - F\left(\frac{\varepsilon + v}{\Gamma^{k-1}}\right) < \frac{\varepsilon}{2} \quad \text{for } k \ge N. \tag{6.26}$$

For $n > k \ge N$,

$$\begin{aligned} \int \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v) &= \int_0^V \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v) \\ &\quad + \int_V^\infty \mathrm{P}\{\mu_n > \varepsilon + v | \mu_k = v\} dG_k(v) \\ &\le \int_0^V \frac{\varepsilon}{2} dG_k(v) + \int_V^\infty dG_k(v) < \varepsilon. \end{aligned} \tag{6.27}$$

Thus, the μ_k satisfy the Cauchy criterion and converge in probability to a random variable whose distribution function is F [6, pp. 226–227]. □

6.3. Proof of Theorem 4.1.

The results of the preceding two subsections combine to form a proof of Theorem 4.1 as follows.

Proof. By Propositions 6.4 and 6.6,

$$\begin{aligned}\sum_{j=1}^{m}(\Lambda_{e_x}(p_1^T y,\dots,p_k^T y))_{jj}-(\Lambda_{e_x}(y))_{jj} &= \sum_{i=k+1}^{m}\|\Lambda_x C^T p_i\|^2\\ &\leq 4\|\Lambda_x\|\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}})\zeta\mu_m\sum_{i=k}^{m-1}\sum_{j=1}^{\lfloor\frac{i}{4}\rfloor-1}\frac{\lambda_{z,j}}{\lambda_{y,j}^2}\frac{1}{\gamma^{(i-2j)}}+\|\Lambda_x\|\sum_{i=k}^{m-1}\sum_{j=\lfloor\frac{i}{4}\rfloor}^{m}\frac{\lambda_{z,j}}{\lambda_{y,j}},\end{aligned}\tag{6.28}$$

and

$$\begin{aligned}\sum_{j=1}^{m}(\Lambda_{e_z}(p_1^T y,\dots,p_k^T y))_{jj}-(\Lambda_{e_z}(y))_{jj} &= \sum_{i=k+1}^{m}\|\Lambda_x C^T p_i\|^2\\ &\leq 4\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}})\zeta\mu_m\sum_{i=k}^{m-1}\sum_{j=1}^{\lfloor\frac{i}{4}\rfloor-1}\frac{\lambda_{z,j}^2}{\lambda_{y,j}^2}\frac{1}{\gamma^{(i-2j)}}+\sum_{i=k}^{m-1}\sum_{j=\lfloor\frac{i}{4}\rfloor}^{m}\frac{\lambda_{z,j}^2}{\lambda_{y,j}}.\end{aligned}\tag{6.29}$$

By Assumptions 4 and 2, $\lambda_{z,j}/\lambda_{y,j}\leq\bar{\lambda}_j/\sigma^2$ and $\bar{\lambda}_j/(\gamma^j\lambda_{y,j})\leq\xi$ for a constant ξ. Moreover, $\lambda_{z,i}/\lambda_{y,j}\leq 1$, in general. Thus

$$\begin{aligned}\sum_{j=1}^{m}(\Lambda_{e_x}(p_1^T y,\dots,p_k^T y))_{jj}-(\Lambda_{e_x}(y))_{jj} &= \sum_{i=k+1}^{m}\|\Lambda_x C^T p_i\|^2\\ &\leq \frac{4\|\Lambda_x\|\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}})\zeta\mu_m\xi}{\sigma^2(1-\frac{1}{\gamma^2})}\sum_{i=k}^{m-1}\frac{1}{\gamma^{i/4}}+\frac{\|\Lambda_x\|}{\sigma^2}\sum_{i=k}^{m-1}(i-k+4)\bar{\lambda}_{\lfloor\frac{i}{4}\rfloor},\end{aligned}\tag{6.30}$$

and

$$\begin{aligned}&\sum_{j=1}^{m}(\Lambda_{e_z}(p_1^T y,\dots,p_k^T y))_{jj}-(\Lambda_{e_z}(y))_{jj} = \sum_{i=k+1}^{m}\|C\Lambda_x C^T p_i\|^2\\ &\leq \frac{4\|\Lambda_y\|\|s\|^2 K^2(\lambda_{\text{sep}})\zeta\mu_m}{(1-\frac{1}{\gamma^2})}\sum_{i=k}^{m-1}\frac{1}{\gamma^{i/2}}+\sum_{i=k}^{m-1}(i-k+4)\min\left(\frac{\bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\lambda_{z,\lfloor\frac{i}{4}\rfloor}}{\sigma^2},\bar{\lambda}_{\lfloor\frac{i}{4}\rfloor}\right).\end{aligned}\tag{6.31}$$

The increasing μ_m converge in probability to a random variable μ by Proposition 6.7. Equations (4.2) and (4.3) follow immediately from (6.30) and (6.31). □

The analysis presented here predicts actual convergence behaviors, as illustrated next with the numerical examples in section 7.

7. Numerical examples.

The following numerical examples illustrate the actual performance of the algorithm in relation to the theory of the previous sections. There are four different examples. Each one illustrates a different aspect of the theory. The estimation problems in each of the examples is different. The breadth of estimation problems provides a glimpse at the range of applicability of the Krylov subspace estimation algorithm. For each of the following problems, full orthogonalization was used, except as noted.

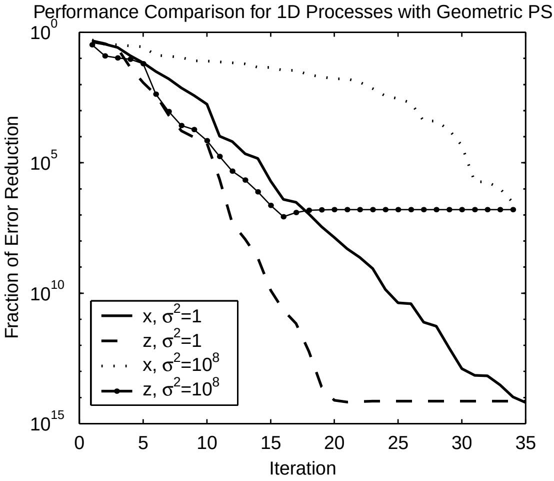

FIG. 7.1. *The four curves plotted here show the convergence behaviors when computing error variances for estimating two different quantities in two slightly different estimation problems. One of the quantities to be estimated is a* 1*-D process, x, and the other is a subsampled version of the same process, z. Both quantities are estimated from measurements consisting of z embedded in additive noise. The only difference between the two estimation problems is the variance of the noise,* σ^2*, which is* 1 *in one case and* 10^{-8} *in the other. The curves indicate that convergence is slower for lower* σ^2 *and for estimating x, as predicted by Theorem* 4.1.

The results in Figure 7.1 illustrate the relationship between the actual performance of the algorithm and that predicted by Theorem 4.1. The estimation problem consists of estimating 1024 samples of a stationary process, x, on a 1-D torus from 512 consecutive point measurements, y. The power spectral density (PSD) of x has a geometric decay, $S_{xx}(\omega) \propto (0.3)^{|\omega|}$, and is normalized so that the variance of x is one. Depicted in Figure 7.1 are the fractions of error reduction obtained for estimating x,

$$\frac{\sum_{i=1}^{l}(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}}{\sum_{i=1}^{l}(\Lambda_x - \Lambda_{e_x}(y))_{ii}}, \tag{7.1}$$

and z,

$$\frac{\sum_{i=1}^{l}(\Lambda_{e_z,k}(y) - \Lambda_{e_z}(y))_{ii}}{\sum_{i=1}^{l}(\Lambda_z - \Lambda_{e_z}(y))_{ii}}, \tag{7.2}$$

where $\Lambda_n = \sigma^2 I$ for $\sigma^2 = 1$ and $\sigma^2 = 10^{-8}$. Note that the numerators in (7.1) and (7.2) are the terms bounded in Theorem 4.1 and that the denominators are independent of the iteration index, k. The reference values $\Lambda_{e_x}(y)$ and $\Lambda_{e_z}(y)$ are computed using direct methods in MATLAB. The numerical errors in these direct methods tend to dominate after several iterations especially for $\sigma^2 = 10^{-8}$. Note that the eigenvalues of Λ_x and Λ_z satisfy $\lambda_{x,i} \geq \lambda_{z,i} \geq \lambda_{x,l-m+i}$ as a consequence of Cauchy's interlace theorem [9, Theorem 4.3.15] and the rows of the measurement matrix C being orthogonal. Since the PSD (collection of eigenvalues) display a two-sided geometric decay, Λ_z and, in turn, $\Lambda_y = \Lambda_z + \sigma^2 I$ may have eigenvalue multiplicities of order two. However, the plots show a geometric rate of convergence consistent with a geometrical decay of Λ_y despite the fact that the block form of the algorithm is not used. A block form is not necessary because roundoff error will introduce components of the

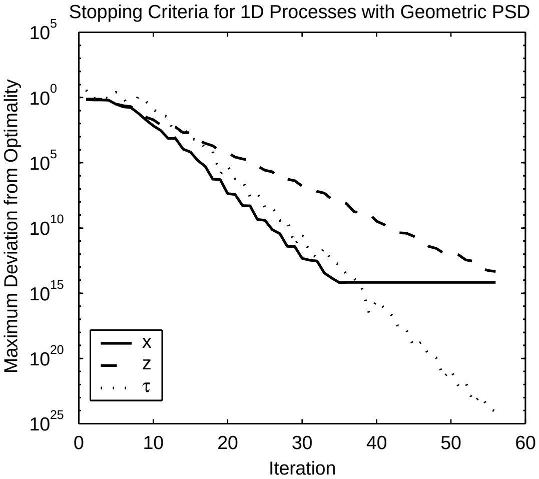

FIG. 7.2. *The results plotted here indicate how the computable quantities making up the two stopping criteria of section* 3 *relate to the difference between the computed approximation to the error covariance for estimating x at iteration k and the optimal error covariance, $\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y)$. The solid line is the maximal difference between the computed and optimal error variances for estimating x, $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$. Each of the other two curves plot the quantities making up the two stopping criteria. The dashed line is the maximal error variance for estimating z, $\max_i(\Lambda_{e_z,k}(z))_{ii}$, and the dotted line is the maximum change made to the error variances at the current iteration, $\tau_{k,0}$, as defined in* (3.1) *for $K_{\mathrm{win}} = 0$.*

eigenvectors of Λ_y into the Krylov subspaces that are not present in the starting vector [15, p. 228]. Note also that, as suggested by Theorem 4.1, the rate of convergence is faster for the error variances at measurement locations, i.e., for estimates of z, than away from measurement locations, i.e., for estimates of all of x. The theorem also suggests that convergence is slower for smaller σ^2, which is evident in Figure 7.1. Thus, Theorem 4.1 accurately predicts convergence behavior.

Figure 7.2 depicts how the two stopping criteria relate to the difference between the computed approximation to the error covariance for estimating x at iteration k and the optimal error covariance, $\Lambda_{e_x,k}(y)-\Lambda_{e_x}(y)$. The process to be estimated is the same one previously described. The measurement locations are chosen randomly. At any given location, the chance that there is a measurement is 50% and is independent of there being a measurement at any other sample point. The measurement noise covariance matrix is a diagonal matrix whose elements vary according to the following triangle function:

$$(\Lambda_n)_{ii} = \begin{cases} 9\frac{i-1}{\lfloor m/2 \rfloor - 1} + 1 & \text{for } 1 \le i \le \lfloor m/2 \rfloor, \\ 9\frac{m-i}{m-\lfloor m/2 \rfloor - 1} + 1 & \text{for } \lfloor m/2 \rfloor + 1 \le i \le m. \end{cases} \tag{7.3}$$

A whitening preconditioner, Λ_n^{-1}, is used. The figure contains plots of the maximal difference between the computed and optimal error variances for estimating x, $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$. There are also plots of the two quantities making up each of the two stopping criteria. One is of the maximal error variance for estimating z, $\max_i(\Lambda_{e_z,k}(z))_{ii}$, and the other is of the maximum change made to the error variances at the current iteration, $\tau_{k,0}$, as defined in (3.1). Note that $\Lambda_{e_z,k}(z)$ is a bound on $\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y)$, but that the rates of convergence of these two quantities are different. The $\tau_{k,0}$, on the other hand, are more erratic but decrease at a rate close

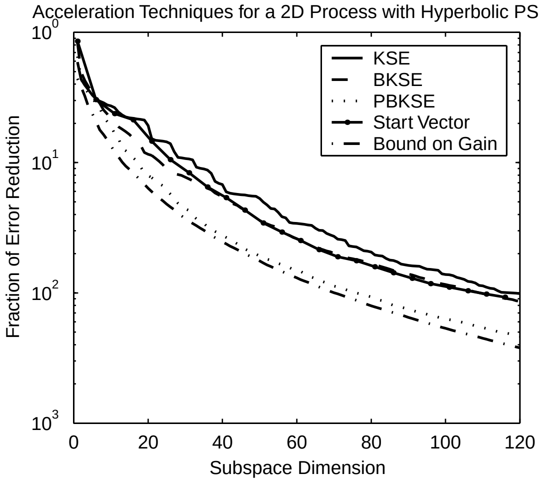

FIG. 7.3. *The results plotted here indicate that various acceleration techniques can be used to achieve nearly optimal performance. The curves depict the fraction of error reduction for estimating x for different methods of choosing linear functionals of the data. The figure shows the results for the standard Krylov subspace estimation algorithm (KSE), a block form with a block size of* 2 *(BKSE), and a preconditioned block form (PBKSE), also with a block size of* 2*. For comparison, the figure shows two additional curves. One (Start Vector) is of the results for Algorithm* 2.1 *modified to start with a linear combination of the first* 60 *eigenvectors of* Λ_y*. The other (Bound on Gain) is of the fraction of error reduction attained by using the optimal linear functionals of the data.*

to $\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y)$. Stopping when $\tau_{k,\varepsilon_{\min}}$ falls below a threshold has been the most successful criterion because the $\tau_{k,\varepsilon_{\min}}$ give a good indication of the rate of decrease of $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$. However, stopping when $\max_i(\Lambda_{e_z,k}(z))_{ii}$ falls below a threshold is a preferable criterion when the noise intensity is small primarily because $\max_i(\Lambda_{e_z,k}(z))_{ii}$ provides a tight bound on $\max_i(\Lambda_{e_x,k}(y) - \Lambda_{e_x}(y))_{ii}$.

A comparison among various techniques to accelerate convergence is provided in Figure 7.3. The estimation problem consists of estimating a stationary random field, x, on a 32×32 toroidal grid from point measurements, y, of equal quality taken over one 32×16 rectangle. The PSD of x is proportional to $1/(|\omega|+1)^3$ and is normalized so that the variance of x is one. The measurement noise covariance matrix Λ_n is $4I$. The plots are of the fraction of error reduction attained for estimating x, as defined by (7.1), versus the Krylov subspace dimensions. Both a right-preconditioned and a block form are considered. The preconditioner has the form $C\Lambda_p C^T$, as described in section 5.1. A simple block algorithm (BKSE) with a block size of 2 does not do much better than the standard algorithm (KSE). However, a preconditioned block form (PBKSE) requires considerably fewer iterations to achieve a given level of accuracy than the standard algorithm. The error reduction attained by using the optimal linear functionals of the data is also plotted in Figure 7.3. The performance of PBKSE is close to the optimal performance. Figure 7.3 also shows the results of an experiment to determine whether one can gain much by picking a good starting vector. A starting vector with components in each of the first 60 eigenvectors of Λ_y was used to start a run. The results are plotted in Figure 7.3 and are comparable to those of BKSE, indicating that one does not gain much by picking a good starting vector. That the choice of starting vector should have little impact on the results is a consequence of Proposition 6.7.

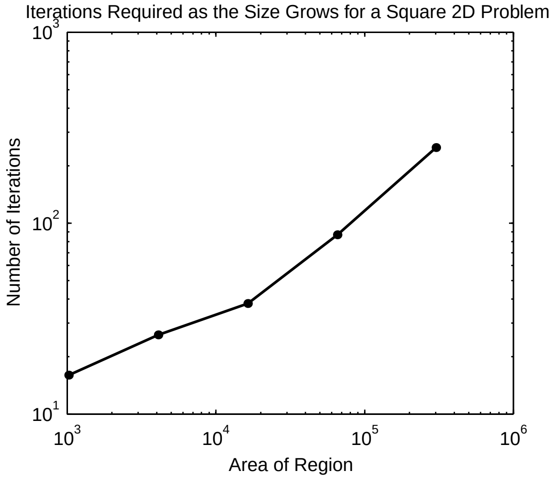

FIG. 7.4. *The number of iterations required for a practical 2-D problem of interest is not very large and grows no more than linearly with the area of the region of interest.*

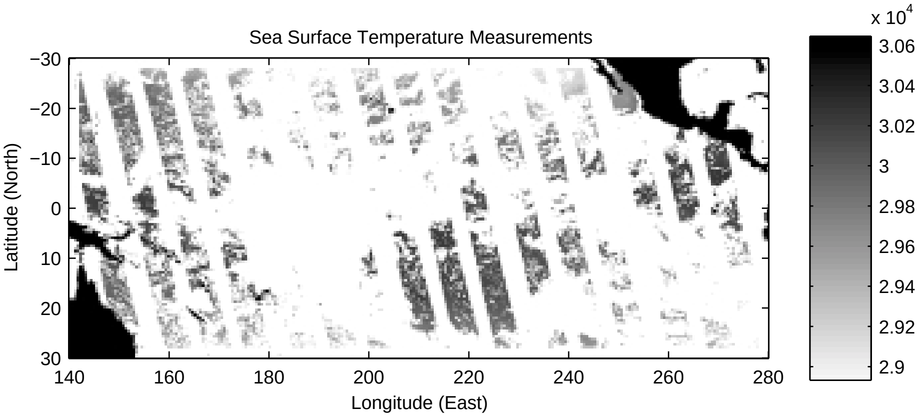

FIG. 7.5. *These data are satellite measurements of sea surface temperature. Measurements are taken only along satellite tracks with no obscuring cloud cover.*

Lastly, Figure 7.4 shows how the number of iterations grows with the region size for the problem of estimating deviations from mean sea surface temperature, x, from the satellite data, y, in Figure 7.5 [7]. The temperature deviations are estimated on a rectangular grid and are assumed to be stationary with a Gaussian-shaped covariance function. The width of the Gaussian is 60 pixels, and the height is 9×10^4. The measurements are very scattered because they exist only along the satellite tracks where there is no obscuring cloud cover (see Figure 7.5). The measurement noise covariance Λ_n is $400I$. Figure 7.4 shows how the number of iterations needed to satisfy $\tau_{k,10^{-2}} < 10^{-2}$ for $K_{\text{win}} = 8$ grows as a region of interest grows. Note that the measurement density in these regions varies from approximately $10 - 20\%$. The growth in the number of iterations is less than linear as the area of the region grows. One expects this behavior because one should need an increasing number of linear functionals as the region grows, but the growth should be no more than linear in the

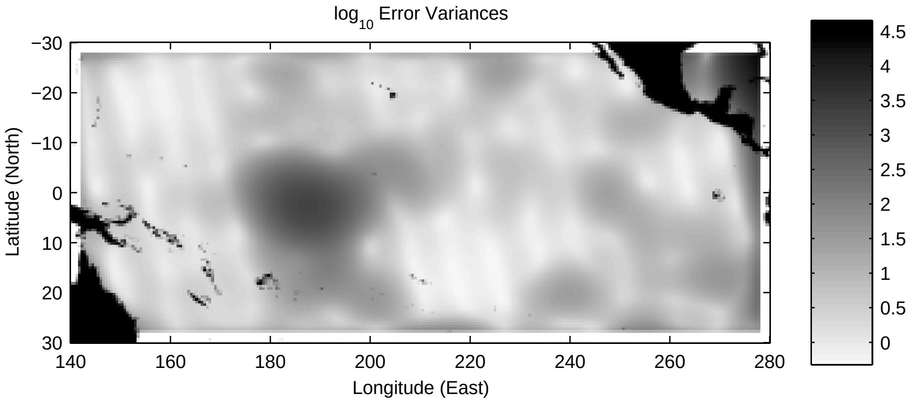

FIG. 7.6. *The Krylov subspace estimation algorithm generated these error variances on a* 1/6*-degree grid.*

area, provided that the process is stationary (as it is in this case). Figure 7.6 shows the error variances for estimating sea surface temperature given all 42,298 measurements in Figure 7.5. A selective orthogonalization scheme was used to generate this result [18, Appendix B]. Although the number of iterations is growing with problem size, the number of iterations needed for this moderately large 320,400-dimensional estimation problem is 249. That only a relatively small number of iterations was used indicates that the algorithm has found a very low-rank but very good estimator. Hence, the algorithm described here can be used to solve high-dimensional, practical problems with relatively few iterations.

8. Conclusion. In this paper, a statistical interpretation of CG has been used to derive a Krylov subspace estimation algorithm. The algorithm computes a low-rank approximation to the linear least-squares error reduction term which can be used to recursively compute LLSEs *and* error variances. An analysis of the convergence properties explains behaviors of the algorithm. In particular, convergence is more rapid at measurement locations than away from them when there are scattered point measurements. Furthermore, the analysis indicates that a randomly generated vector is a good starting vector. The theory also suggests preconditioning methods for accelerating convergence. Preconditioning has been found to increase the rate of convergence in those cases where convergence is not already rapid.

The low-rank approximation to the error reduction term is a very useful statistical object. The computation of estimates and error variances is just one application. Another is the simulation of Gaussian random processes. Simulation typically requires the computation of the square root of the covariance matrix of the process, a potentially costly procedure. However, the Krylov subspace estimation algorithm can be adapted to generate a low-rank approximation to the square root of the covariance matrix. Yet another application is the fusion of existing estimates with those generated by additional data. The resulting fusion algorithm can also be used as the engine of a Kalman filtering routine, thereby allowing the computation of estimates of quantities evolving in time. This is the subject of ongoing research.

Acknowledgments. The authors wish to thank Jacob White, Gilbert Strang, and Hamid Krim for helpful discussions, Paul Fieguth for his insight and for providing the sea surface temperature data, and the reviewers for their thoughtful suggestions.

REFERENCES

[1] A. F. Bennett, *Inverse Methods and Data Assimilation*, Lecture notes from the summer school at the College of Oceanic and Atmospheric Sciences, Oregon State University, Corvallis, OR, 1999.

[2] A. F. Bennett, B. S. Chua, and L. M. Leslie, *Generalized inversion of a global numerical weather prediction model*, Meteorology Atmospheric Phys., 60 (1996), pp. 165–178.

[3] A. F. Bennett, B. S. Chua, and L. M. Leslie, *Generalized inversion of a global numerical weather prediction model* II: *Analysis and implementation*, Meteorology Atmospheric Phys., 62 (1997), pp. 129–140.

[4] A. da Silva and J. Guo, *Documentation of the Physical-space Statistical Analysis System (PSAS) Part* I: *The Conjugate Gradient Solver Version PSAS*-1.00, DAO Note 96-02, Data Assimilation Office, Goddard Laboratory for Atmospheres, NASA, 1996; also available online from ftp://dao.gsfc.nasa.gov/pub/office_notes/on9602.ps.Z.

[5] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[6] R. M. Dudley, *Real Analysis and Probability*, Chapman and Hall, New York, 1989.

[7] P. W. Fieguth, M. R. Allen, and M. J. Murray, *Hierarchical methods for global-scale estimation problems*, in Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, IEEE, New York, NY, 1998, pp. 161–164.

[8] G. Golub and C. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1996.

[9] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[10] M. R. Leadbetter, G. Lindgren, and H. Rootzen, *Extremes and Related Properties of Random Sequences and Processes*, Springer-Verlag, New York, 1983.

[11] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley and Sons, New York, 1969.

[12] S. Mallat, G. Papanicolau, and Z. Zhang, *Adaptive covariance estimation of locally stationary processes*, Ann. Statist., 26 (1998), pp. 1–47.

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

[14] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[15] B. N. Parlett and D. S. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.

[16] Y. Saad, *On the rates of convergence of the Lanczos and the block-Lanczos method*, SIAM J. Numer. Anal., 17 (1980), pp. 687–706.

[17] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.

[18] M. K. Schneider, *Krylov Subspace Estimation*, Ph.D. thesis, MIT, Cambridge, MA, 2001.

[19] G. Xu, Y. Cho, and T. Kailath, *Application of fast subspace decomposition to signal processing and communication problems*, IEEE Trans. Signal Process., 42 (1994), pp. 1453–1461.

[20] G. Xu and T. Kailath, *Fast estimation of principal eigenspace using Lanczos algorithm*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 974–994.

[21] G. Xu and T. Kailath, *Fast subspace decomposition*, IEEE Trans. Signal Process., 42 (1994), pp. 539–551.

[22] G. Xu, H. Zha, G. Golub, and T. Kailath, *Fast algorithms for updating signal subspaces*, IEEE Trans. Circuits Systems II Analog Digital Signal Process., 41 (1994), pp. 537–549.