UCRL-JC-135513 Pt 2 Version date: September 7, 1999

Analysis of Approximate Inverses in Tomography
II. Iterative Inverses

James G. Berryman
Lawrence Livermore National Laboratory
P. O. Box 808 L-200
Livermore, CA 94551-9900

Abstract

This paper continues the analysis of approximate inverses from Part I by concentrating on iterative inverses in linear tomographic applications. The importance of differentiating between the ideal resolution of the operator/matrix to be inverted and the actual or effective resolution obtained by the approximate inverse in a iterative procedure is stressed. Means of obtaining the effective resolution operator for standard iterative procedures such as conjugate gradients, Lanczos, and LSQR are provided while circumventing the usual need to produce a singular-value decomposition of the operator being inverted. The methods discussed produce very simple results in calculations with infinite precision, but require reorthogonalization of the Krylov vectors/operators produced by the iterative procedures in finite precision. Although this need for reorthogonalization increases the expense of the procedure somewhat, it still produces the desired results much more efficiently than what could be obtained using a full singular-value decomposition of the operator.

1 Introduction

This paper is a continuation of Part I [Berryman, 1999]. We focus here on applications of our ideas concerning tomographic resolution to numerical methods and expecially to approximate iterative inverses of matrices that are either hard or impossible to invert by traditional methods. We discuss many of the standard methods such as conjugate gradients [Hestenes and Stiefel, 1952], Lanczos [Lanczos, 1950], LSQR [Paige and Saunders, 1982], etc., but our focus is on understanding how the fact that the matrix in question is usually not positive definite, and often very large and rectangular, affects the analysis. We would also like to have some means of comparing the various approximate inverses (at least qualitatively), especially when the iterative inversion procedure is terminated (as it usually is in the context of large tomography/imaging applications) before convergence. The means we chose in Part I and that we will continue to stress here is the concept of the "effective resolution matrix." As we will show, this matrix can usually be computed at little extra cost in either time or storage, but gives a realistic assessment of the progress of the inversion algorithm, and a straightforward means of comparing different approximation/numerical schemes.

Recent work of a similar nature on computation of resolution matrices has been published by Berryman [1994a,b], Zhang and McMechan [1995] and Minkoff [1996]. A different approach to analysis of approximate inverses has been presented recently by Vasco *et al.* [1996]. Some of the work presented here concerning the method of conjugate gradients is joint work with Fomel [Berryman and Fomel, 1996].

2 Iterative Methods

We will now study iterative methods of solution to the basic inversion problem for $\mathbf{Ms} = \mathbf{t}$. We refer to the model abstractly as a vector $\mathbf{s}$ in a vector space. For a block model with n blocks, the model becomes a vector in R^n, the n-dimensional Euclidean vector space. Similarly, the data vector $\mathbf{t}$ is a vector in the m-dimensional data space. The matrix $\mathbf{M}$ is $m \times n$ and is probably not of full rank in most problems of interest. Applications of interest include both overdetermined $(m > n)$ and underdetermined $(m < n)$ problems.

First consider the full rank case for overdetermined problems in which $r = n < m$. The least-squares solution is then given by

$$\hat{\mathbf{s}}_{\text{LS}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{t}. \qquad (1)$$

Thus, we begin by summarizing the main ideas behind a matrix inversion method that works if the symmetric matrix $\mathbf{A} = \mathbf{M}^T\mathbf{M}$ (the normal matrix for overdetermined least squares) is invertible. Then, we discuss other methods applicable to more realistic problems in tomography when the rank r satisfies either $r < n < m$ or $r < m < n$.

2.1 Richardson iteration

In Richardson iteration [Varga, 1962], we start with an initial model $\mathbf{s}^{(0)}$ and iteratively generate a sequence $\mathbf{s}^{(k)}$, $k = 1, 2, \ldots$using

$$\mathbf{s}^{(k+1)} = \mathbf{s}^{(k)} + \mathbf{M}^T(\mathbf{t} - \mathbf{M}\mathbf{s}^{(k)}). \tag{2}$$

In terms of singular vector expansion coefficients given by

$$\mathbf{M} = \sum_{i=1}^{r} \lambda_i \mathbf{y}_i \mathbf{z}_i^T, \quad \mathbf{s} = \sum_{i=1}^{r} \sigma_i \mathbf{z}_i, \quad \text{and} \quad \mathbf{t} = \sum_{i=1}^{r} \tau_i \mathbf{y}_i, \tag{3}$$

the iteration sequence becomes

$$\sigma_i^{(k+1)} = \sigma_i^{(k)} + \lambda_i(\tau_i - \lambda_i \sigma_i^{(k)}). \tag{4}$$

To solve this equation, note that it can be rewritten as

$$\sigma_i^{(k+1)} = \lambda_i\tau_i + (1 - \lambda_i^2)\sigma_i^{(k)} = \lambda_i\tau_i + (1 - \lambda_i^2)[\lambda_i\tau_i + (1 - \lambda_i^2)\sigma_i^{(k-1)}]. \tag{5}$$

Computing and then rrarranging the resulting series, we find

$$\sigma_i^{(k+1)} = [1 + (1 - \lambda^2) + (1 - \lambda^2)^2 + \ldots + (1 - \lambda^2)^k]\lambda_i\tau_i + (1 - \lambda^2)^{k+1}\sigma_i^{(0)}. \tag{6}$$

The series multiplying τ_i can be summed exactly for any value of $\lambda_i \neq 0$ as

$$[1 + (1 - \lambda_i^2) + (1 - \lambda_i^2)^2 + \ldots + (1 - \lambda_i^2)^k] = \frac{1 - (1 - \lambda_i^2)^{k+1}}{1 - (1 - \lambda_i^2)} \tag{7}$$

from which it follows that

$$\sigma_i^{(k+1)} = \left[\frac{1 - (1 - \lambda_i^2)^{k+1}}{\lambda_i}\right]\tau_i + (1 - \lambda_i^2)^{k+1}\sigma_i^{(0)}. \tag{8}$$

If $\lambda_i = 0$, equation (4) shows that $\sigma_i^{(k+1)} = \sigma_i^{(0)}$. If we assume that the singular values, λ_i, are all between $-\sqrt{2}$ and $\sqrt{2}$, the iteration sequence converges for $\lambda_i \neq 0$. The condition $0 < \lambda_i^2 < 2$ thus implies that $\sigma_i^{(k)} \rightarrow \sigma_i^{(\infty)} \equiv \tau_i/\lambda_i$ as $k \rightarrow \infty$. That is, the iteration converges to the minimum-norm least-squares model.

The stronger condition $-1 \leq \lambda_i \leq 1$ can be guaranteed with an appropriate preconditioning (prescaling) of the matrix $\mathbf{M}$.

Richardson iteration is a simple method for solving linear tomography problems, and is somewhat easier to implement than other methods such as conjugate directions or conjugate gradients since no orthogonalization (or conjugacy producing) step is required. This method has some computational advantages when the data dimension m of $\mathbf{M}$ is much larger than the model dimension n. The method is also closely related to the well-known tomographic inversion technique SIRT (Simultaneous Iterative Reconstruction Technique) [Gilbert, 1972; Dines and Lytle, 1979; Ivansson, 1983].

If we want to compute the model and data resolution for this method, the advantages of Richardson's iteration quickly turn into disadvantages. Since the method does not store any vectors except the most recent estimate of the model vector, we must generate a set of linearly independent vectors from which to compute the resolution matrices. To do this, we consider the update vectors

$$\Delta\mathbf{s}^{(k)} \equiv \mathbf{s}^{(k+1)} - \mathbf{s}^{(k)} = \mathbf{M}^T \left(\mathbf{t} - \mathbf{M}\mathbf{s}^{(k)}\right). \tag{9}$$

Substituting (8) into (9), we find that

$$\Delta\mathbf{s}^{(k)} = \sum_{i=1}^{r} (1 - \lambda_i^2)^k \lambda_i^2 \left(\sigma_i^{(\infty)} - \sigma_i^{(0)}\right) \mathbf{z}_i, \tag{10}$$

showing that the first r updates will be linearly independent. But this approach does not give a procedure for determining the value $r < n$. Once we have these vectors, they can be used in the conjugate directions method to produce a set of conjugate vectors in the model space and a set of orthogonal vectors in the data space, as we will elaborate in Section 2.3. Then, the procedure for computing the resolution matrices is the same as discussed for conjugate directions. This modified Richardson iteration generally constructs symmetric data resolution matrices and nonsymmetric model resolution matrices. These results are essentially equivalent to those of other conjugate directions algorithms as we will show. Even the convergence properties are similar, since the approximate inverse may, in principle, be constructed using only $r < n$ of the update vectors.

2.2 Singular-value decomposition

Another approach to solving the problem $\mathbf{M}\mathbf{s} = \mathbf{t}$ that is available but which we will not pursue at much length here, involves constructing the singular value decomposition of $\mathbf{M}$. We alluded to this previously and displayed the formal representation of the singular value decomposition as

$$\mathbf{M} = \sum_{i=1}^{r} \lambda_i \mathbf{y}_i \mathbf{z}_i^T, \tag{11}$$

where the $\mathbf{z}_i$'s are the eigenvectors of $\mathbf{M}^T\mathbf{M}$, the $\mathbf{y}_i$'s are the eigenvectors of $\mathbf{M}\mathbf{M}^T$, and the nonzero eigenvalues of both matrices are given by λ_i^2.

Once the SVD is known it is straightforward to construct the pseudoinverse of $\mathbf{M}$ as

$$\mathbf{M}^\dagger = \sum_{i=1}^{r} \lambda_i^{-1} \mathbf{z}_i \mathbf{y}_i^T, \tag{12}$$

or, what is more likely to happen in practice, to construct a truncated version of the SVD. The truncated SVD is normally constructed by first ordering the eigenvalues and eigenvectors (taken to be positive) so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$, where $k < r$. Then the truncated SVD is given by

$$X_k = \sum_{i=1}^{k} \lambda_i^{-1} \mathbf{z}_i \mathbf{y}_i^T, \tag{13}$$

for some convenient choice of k. The advantage of using either the SVD or the truncated SVD is that minimal damage is done to the most significant part of the spectrum of the inverse operator. Methods using damped least-squares or other regularization methods not based on the truncated SVD typically alter the spectrum in order to remove noisy structure from the final result. The truncated SVD accomplishes this by eliminating the components with the lowest singular values and therefore the ones that are most likely to introduce noisy structures into the inversion.

The disadvantage of using the SVD or the truncated SVD lies in the expense of finding the eigenvectors and eigenvalues. In some sense, by constructing the full SVD we have obtained a lot of information about the operator we are trying to inverse: more information than we really need to do the inversion in a satisfactory way in most cases. We might therefore think of the SVD as the ideal means of constructing the inverse, but nevertheless avoid this ideal approach for the sake of its often prohibitive computational expense.

For the sake of argument, we will talk about the truncated SVD as if it is also an iterative method that can be used to compute an approximate inverse of $\mathbf{M}$ by successively including more of the smaller singular values. It may not be practical to compute the singular values successively in precisely this way, but it will help the discussion in the next section if we imagine this might be possible.

2.3 Intuitive definition of resolution in tomography

Now it is a general characteristic of iterative methods (and we include the truncated SVD in this category as discussed in the last subsection) that each step of the iteration is centered around finding a new vector to include in the analysis. Let us suppose that the vectors found in some iterative method so far are $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, ..., $\mathbf{v}_k$, where $k \leq r$, forming a Krylov subspace of dimension k. We assume that these vectors are rigorously orthogonal (which is a nontrivial assumption for most iterative methods) and also normalized, so that $\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}$. This assumption will be relaxed and the consequences then treated later in the paper. For what follows, the orthogonality is necessary and normalization is convenient. We would like to have some means of characterizing the subspace of the full vector space that is spanned by these vectors. There is a well-known means of doing this, namely the projection operator

$$\mathbf{P}_k \equiv \sum_{i=1}^{k} \mathbf{v}_i\mathbf{v}_i^T. \qquad (14)$$

Using the orthonormal properties of the vector sequence, it is easy to see that any vector lying in the subspace formed by these vectors will be preserved when operated upon by $\mathbf{P}_k$, while any components lying outside of this subspace will not survive the operation:

$$\mathbf{P}_k \left(\sum_{i=1}^{k} a_i\mathbf{v}_i + \sum_{i=k+1}^{r} a_i\mathbf{v}_i \right) = \sum_{i=1}^{k} a_i\mathbf{v}_i. \qquad (15)$$

But this operation is easily seen to be precisely the same as the concept of the resolution operator in tomography as discussed in Part I. In particular, it agrees precisely with the resolution operator one would obtain from a truncated singular-value decomposition of $\mathbf{M}$.

We therefore take equation (14) as a *definition* of the resolution operator for iterative methods can be constructed simply by forming the projection operator for any orthonormal sequence of vectors generated by the iterative procedure as long as that vector sequence lies in one of the proper spaces, *i.e.*, the data space or the model space.

This intuitive approach to resolution will prove to be very helpful in understanding the iterative methods to be presented in the following sections.

2.4 Conjugate directions and conjugate gradients

This subsection summarizes joint work of Berryman and Fomel [1996].

We will continue to use a notation consistent with earlier work of Berryman [1990] on crosswell seismic tomography and with Part I in which the linear inversion problem to be solved takes the form

$$\mathbf{M}\mathbf{s} = \mathbf{t}, \tag{16}$$

where $\mathbf{s}^T = (s_1, s_2, \ldots, s_n)$ is an n-vector of wave slownesses associated in either two- or three-dimensions with cells of constant slowness, $\mathbf{M}$ is a matrix of ray-path lengths such that M_{ij} is the length of the i-th ray path through the j-th cell, and $\mathbf{t}^T = (t_1, t_2, \ldots, t_m)$ is an m-vector of the traveltimes associated with the ray paths between specified and numbered pairs of sources and receivers. We may assume that the traveltime data are given and that the model vector $\mathbf{s}$ is being sought. We further assume for the present purposes that the ray-path matrix is known. This latter assumption is optimistic, but corresponds to assuming that the full nonlinear inverse problem is being solved in an iterative fashion — in which case the ray-path matrix in question is just the one in use at the latest stage of the iteration process. We generally assume in addition that the problem is overdetermined so that $m > n$, *i.e.*, the number of data exceed the dimension of the model space, but underdetermined problems can also be treated by virtually the same methods.

Although we are presenting the analysis as if we were dealing with matrix operators operating on n-vectors and producing m-vectors as output, we want to make it clear that virtually the same analysis works also for general operators acting on model functionals and producing data functionals as output. The analysis would become rather confusing if we were to use both types of language side by side, so we will restrict the language to that of matrices and vectors but simply note that the generalization to operators is nearly immediate in most cases.

2.4.1 Conjugate directions

We want to solve the problem (16) in an iterative fashion, so we assume that the updates to the solution take the general form

$$\mathbf{s}_n = \mathbf{s}_{n-1} + \alpha_n \mathbf{p}_{n-1}, \tag{17}$$

where $\mathbf{s}_{n-1}$ is the preceding estimate of $\mathbf{s}$, $\mathbf{s}_n$ is the new estimate of $\mathbf{s}$, $\mathbf{p}_{n-1}$ is some direction to be specified in the model space, and α_n is a optimization parameter (or direction weight factor). Defining the residual data error as $\mathbf{r}_n \equiv \mathbf{t} - \mathbf{M}\mathbf{s}_n$, we find the general relation that

$$\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n \mathbf{M}\mathbf{p}_{n-1}. \tag{18}$$

One useful way to proceed is to choose the optimization parameter α_n so that the residual vector is decreased and preferably minimized at each step of the iteration scheme. Using the standard inner product notation $(\cdot,\cdot)$ and considering

$$||\mathbf{r}_n||^2 = ||\mathbf{r}_{n-1}||^2 - 2\alpha_n(\mathbf{r}_{n-1}, \mathbf{M}\mathbf{p}_{n-1}) + \alpha_n^2(\mathbf{M}\mathbf{p}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})^2, \tag{19}$$

we find easily that the optimum choice of α_n using this criterion is

$$\alpha_n = \frac{(\mathbf{r}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})}{(\mathbf{M}\mathbf{p}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})}. \tag{20}$$

This formula has the significance that, whenever the residual $\mathbf{r}_{n-1}$ has a component along the direction $\mathbf{M}\mathbf{p}_{n-1}$, α_n is chosen to scale $\mathbf{M}\mathbf{p}_{n-1}$ so that this component exactly cancels and therefore removes the contribution to $\mathbf{r}_n$ made by $\mathbf{M}\mathbf{p}_{n-1}$. This result implies therefore that, if $(\mathbf{r}_{n-1}, \mathbf{M}\mathbf{p}_{n-1}) \neq 0$, then with this choice of α_n we have

$$(\mathbf{M}\mathbf{p}_{n-1}, \mathbf{r}_n) = (\mathbf{p}_{n-1}, \mathbf{M}^T\mathbf{r}_n) = 0. \tag{21}$$

We used the adjoint property of the inner product in (21) to show that $\mathbf{p}_{n-1}$ is orthogonal to the gradient vector $\mathbf{g}_n \equiv \mathbf{M}^T\mathbf{r}_n$, so-called because it is the gradient obtained by taking the derivative with respect to $\mathbf{s}_n^T$ of the squared residual error functional associated with (16).

Thus, at each step of this iterative sequence a vector proportional to some vector $\mathbf{p}_n$ is added to the solution, while a vector proportional to $\mathbf{M}\mathbf{p}_n$ is subtracted from the residual. This sequence will be most efficient if the vectors used in decimating the residual are orthogonal, *i.e.*, if

$$(\mathbf{M}\mathbf{p}_n, \mathbf{M}\mathbf{p}_j) = 0 \qquad \text{for} \qquad j = 1, 2, \ldots, n-1. \tag{22}$$

But again using the adjoint relation for the inner product, we find that

$$(\mathbf{p}_n, \mathbf{M}^T\mathbf{M}\mathbf{p}_j) = 0 \qquad \text{for} \qquad j = 1, 2, \ldots, n-1, \tag{23}$$

which is a statement of conjugacy for the vectors $\mathbf{p}_n$. Conjugacy is just a generalization of orthogonality in which the vectors are orthogonal relative the nonstandard inner product $(\cdot, \mathbf{A}\cdot)$ – with $\mathbf{A}$ being a symmetric, positive semidefinite matrix (operator) – instead of the standard inner product given by $(\cdot,\cdot)$ with $\mathbf{A}$ replaced by the identity.

We conclude that conjugacy is a desirable property of the set of direction vectors $\mathbf{p}_n$, so our next necessary step in order to obtain a definite iterative process is to construct a convenient sequence of vectors that have this property. One set of model vectors that will be available in this iteration sequence is the set of gradient vectors themselves, where $\mathbf{g}_n = \mathbf{M}^T\mathbf{r}_n$. We show next why this set plays an important role in constructing the desired sequence.

2.4.2 Why not orthogonal residuals?

Since the direction vectors have not yet been specified, we still have several degrees of freedom that we may use to help choose an optimum method. We might suppose that it could be possible to choose the residuals themselves in a way so that they are orthogonal from one step to the next. But, we soon find this is wrong for, if we were to try this, we would find that the condition

$$(\mathbf{r}_n, \mathbf{r}_{n-1}) = 0 \tag{24}$$

implies that

$$\alpha_n = \frac{(\mathbf{r}_{n-1}, \mathbf{r}_{n-1})}{(\mathbf{r}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})} = \frac{(\mathbf{r}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})}{(\mathbf{M}\mathbf{p}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})}. \tag{25}$$

But, condition (25) is actually *impossible* because the Cauchy-Schwartz inequality for vectors states that

$$(\mathbf{r}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})^2 \le (\mathbf{r}_{n-1}, \mathbf{r}_{n-1})(\mathbf{M}\mathbf{p}_{n-1}, \mathbf{M}\mathbf{p}_{n-1}), \tag{26}$$

where the equality in (26) occurs only when the two vectors are proportional which will virtually never be true for such an iteration scheme.

2.4.3 Conjugate residuals and CG

Since orthogonality of the residuals is impossible, the next best condition we might try to impose is conjugacy of the residual vectors. This condition can be stated as

$$(\mathbf{M}^T\mathbf{r}_n, \mathbf{M}^T\mathbf{r}_j) = (\mathbf{r}_n, \mathbf{M}\mathbf{M}^T\mathbf{r}_j) = 0 \qquad \text{for} \qquad j = 1, 2, \ldots, n-1. \tag{27}$$

If this condition is met, the residuals will be orthogonal relative to the inner product $(\cdot, \mathbf{M}\mathbf{M}^T\cdot)$. The set of conditions (27) is not in conflict with (20) so we assume it holds and construct the directions $\mathbf{p}_n$ that satisfy both (23) and (27).

We will now take $\mathbf{g}_n = \mathbf{M}^T\mathbf{r}_n$ and suppose that

$$\mathbf{p}_n = \mathbf{g}_n - \beta_n \mathbf{p}_{n-1}. \tag{28}$$

[Note that this choice is not unique as additional terms in the $\mathbf{p}_i$'s could have been included on the right hand side of (28) and such choices are made by some authors.] The first term on the right hand side is natural because (20) will vanish unless $\mathbf{p}_{n-1}$ has some component along $\mathbf{g}_{n-1}$. Some additional term or terms are necessary to guarantee that (23) can be satisfied. The choice (28) is consistent with (21), which guarantees that the two terms on the right hand side of (28) are orthogonal and therefore in a sense maximally effective. Making use of the conjugacy condition $(\mathbf{p}_n, \mathbf{M}^T\mathbf{M}\mathbf{p}_{n-1}) = 0$, we find that the constant β_n takes the value

$$\beta_n = \frac{(\mathbf{g}_n, \mathbf{M}^T\mathbf{M}\mathbf{p}_{n-1})}{(\mathbf{p}_{n-1}, \mathbf{M}^T\mathbf{M}\mathbf{p}_{n-1})}. \tag{29}$$

It is not difficult to show that the resulting scheme can be expressed in other ways. For example, the computed coefficients can be rewritten as

$$\alpha_n = \frac{(\mathbf{M}^T\mathbf{r}_{n-1}, \mathbf{p}_{n-1})}{(\mathbf{M}\mathbf{p}_{n-1}, \mathbf{M}\mathbf{p}_{n-1})} = \frac{||\mathbf{M}^T\mathbf{r}_{n-1}||^2}{||\mathbf{M}\mathbf{p}_{n-1}||^2}, \tag{30}$$

and

$$\beta_n = -\frac{||\mathbf{M}^T\mathbf{r}_n||^2}{(\mathbf{M}\mathbf{p}_{n-1}, \mathbf{r}_{n-1})} = -\frac{||\mathbf{M}^T\mathbf{r}_n||^2}{||\mathbf{M}^T\mathbf{r}_{n-1}||^2} \tag{31}$$

which are forms first introduced by Hestenes and Stiefel [1952] in their well-known paper on the conjugate gradients method. It is not hard to show that all the remaining conjugacy conditions, $(\mathbf{p}_n, \mathbf{M}^T\mathbf{M}\mathbf{p}_i)$ for $i < n-1$, are satisfied with these choices for the coefficients (see for example Fomel [1996]).

The method of conjugate gradients is probably the most commonly used technique applying the method of conjugate directions to solve problems of the form $\mathbf{Ms} = \mathbf{t}$ in the least-squares sense [Luenberger, 1973; Björck and Elfving, 1979; Golub and Van Loan, 1983; Ashby *et al.*, 1990].

2.4.4 Resolution operators for both model and data

From (17), it follows easily that the model estimate at the k-th iteration must be of the form

$$\mathbf{s}_k = \sum_{i=1}^{k-1} \alpha_{i+1}\mathbf{p}_i. \tag{32}$$

Then substituting (20) — or more directly the first ratio in (30) — for the α_i's shows that the k-th iterate is given explicitly by

$$\mathbf{s}_k = \sum_{j=1}^{k-1} \frac{\mathbf{p}_j\mathbf{p}_j^T}{(\mathbf{p}_j, \mathbf{M}^T\mathbf{M}\mathbf{p}_j)}\mathbf{g}_j = \sum_{j=1}^{k-1} \frac{\mathbf{p}_j\mathbf{p}_j^T}{(\mathbf{p}_j, \mathbf{M}^T\mathbf{M}\mathbf{p}_j)}\mathbf{M}^T\mathbf{t} \tag{33}$$

for this scheme. Since the least-squares solution is of the form $\mathbf{s} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{t}$, the resulting approximate inverse operator for the normal matrix is therefore

$$\left(\mathbf{M}^T\mathbf{M}\right)^\dagger \simeq \sum_{j=1}^{k-1} \frac{\mathbf{p}_j\mathbf{p}_j^T}{(\mathbf{p}_j, \mathbf{M}^T\mathbf{M}\mathbf{p}_j)}, \tag{34}$$

which form we will now study. We use the dagger notation to indicate the pseudoinverse. The expression in (34) is approximating a pseudoinverse, because it may happen that the normal matrix $\mathbf{M}^T\mathbf{M}$ is singular in which case the usual inverse does not exist.

First, note that although (34) might appear to be in the form of a singualr-value decomposition, it definitely is not an SVD. The $\mathbf{p}_j$'s are not orthogonal and the denominators of these terms are not eigenvalues. If we define the matrix composed of direction vectors at the k-th iteration to be

$$\mathbf{P}_k = \left(\, \mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_k \,\right), \tag{35}$$

then the approximate inverse operator can be rewritten as

$$\left(\mathbf{M}^T\mathbf{M}\right)^\dagger \simeq \mathbf{P}_k\mathbf{D}_P^{-1}\mathbf{P}_k^T, \tag{36}$$

where the matrix $\mathbf{D}_P$ is a diagonal matrix whose diagonal elements are given by $D_{jj} = (\mathbf{p}_j, \mathbf{M}^T\mathbf{M}\mathbf{p}_j)$. In fact the entire matrix is given directly by

$$\mathbf{D}_P \equiv \mathbf{P}_k^T\mathbf{M}^T\mathbf{M}\mathbf{P}_k, \tag{37}$$

because of the conjugacy of the $\mathbf{p}$'s composing $\mathbf{P}_k$. Now equation (28) shows that

$$\mathbf{P}_k\mathbf{B}_k = \mathbf{G}_k, \tag{38}$$

where

$$\mathbf{G}_k = \left(\mathbf{g}_1 \quad \mathbf{g}_2 \quad \cdots \quad \mathbf{g}_k \right), \tag{39}$$

and the matrix $\mathbf{B}_k$ is bidiagonal with units along the main diagonal and β_k's along the upper diagonal. Multiplying (38) on the right by the inverse of $\mathbf{B}_k$ and then substituting into (36), we find that

$$\left(\mathbf{M}^T\mathbf{M}\right)^\dagger \simeq \mathbf{G}_k\mathbf{B}_k^{-1}\mathbf{D}_P^{-1}\left(\mathbf{B}_k^T\right)^{-1}\mathbf{G}_k^T. \tag{40}$$

Thus, the approximate inverse is seen to have the general form

$$\left(\mathbf{M}^T\mathbf{M}\right)^\dagger \simeq \mathbf{G}_k\mathbf{T}_k^{-1}\mathbf{G}_k^T, \tag{41}$$

where $\mathbf{T}_k$ is the tridiagonal matrix

$$\mathbf{T}_k = \mathbf{B}_k^T\mathbf{D}_P\mathbf{B}_k. \tag{42}$$

This result highlights the similarities between the CG method and that of other iterative methods we treat later such as Lanczos [Lanczos, 1950] and LSQR [Paige and Saunders, 1982], also producing tridiagonal respresentations of the matrix to be inverted.

2.4.5 Model resolution for CG

Although the tridiagonal form found in (42) is interesting in its own right, the more important result contained in (41) is the fact that this analysis has resulted in a decomposition in terms of orthogonal (rather than merely conjugate) vectors. This result allows us to obtain the resolution matrix quickly for the model space from this form. In particular, if we define the diagonal matrix

$$\mathbf{D}_G = \mathbf{G}_k^T\mathbf{G}_k, \tag{43}$$

we see that

$$\mathbf{M}^T\mathbf{M} \simeq \mathbf{G}_k\mathbf{D}_G^{-1}\mathbf{T}_k\mathbf{D}_G^{-1}\mathbf{G}_k^T, \tag{44}$$

and therefore, since

$$\mathcal{R}_{model} \equiv \left(\mathbf{M}^T\mathbf{M}\right)^\dagger\mathbf{M}^T\mathbf{M} = \mathbf{M}^T\mathbf{M}\left(\mathbf{M}^T\mathbf{M}\right)^\dagger, \tag{45}$$

we find easily that

$$\mathcal{R}_{model} = \mathbf{G}_k\mathbf{D}_G^{-1}\mathbf{G}_k^T = \sum_{i=1}^{k}\frac{\mathbf{g}_i\mathbf{g}_i^T}{(\mathbf{g}_i, \mathbf{g}_i)}. \tag{46}$$

These are effective resolution operators until $k = r$, at which time they become true resolution operators.

2.4.6 Data resolution for CG

The data resolution is known to be related to the operator

$$\mathcal{R}_{data} = \mathbf{M}\left(\mathbf{M}^T\mathbf{M}\right)^{\dagger}\mathbf{M}^T. \tag{47}$$

Substituting (34) for the pseudoinverse and then defining

$$\mathbf{q}_i \equiv \mathbf{M}\mathbf{p}_i, \tag{48}$$

we find that the resolution operator for the data space is

$$\mathcal{R}_{data} = \sum_{i=1}^{k} \frac{\mathbf{q}_i\mathbf{q}_i^T}{(\mathbf{q}_i, \mathbf{q}_i)}, \tag{49}$$

a form completely analogous to that in (46). These are effective resolution operators until $k = r$, at which time they become true (ideal) resolution operators.

2.5 Tridiagonalization and the method of Lanczos

The main results of this subsection were published previously in Berryman [1994a].

Another common method of solving linear inversion problems of the form $\mathbf{Ms} = \mathbf{t}$ is Lanczos's method [Lanczos, 1950; Golub & Van Loan, 1983; van der Sluis & van der Vorst, 1987]. This method introduces a sequence of orthonormal vectors $\mathbf{z}^{(k)}$ through a process of tridiagonalization, and it must be applied to a square matrix (although there are tricks used to complete the square with rectangular matrices in a way that permits Lanczos to be applied to rectangular matrices, as in LSQR). Applying this method to the normal equations $\mathbf{M}^T\mathbf{Ms} = \mathbf{M}^T\mathbf{t}$, Lanczos's method is a projection procedure equivalent to the following:

$$\mathbf{z}^{(1)}\left(\mathbf{z}^{(1)}\right)^T \mathbf{M}^T\mathbf{t} = \mathbf{M}^T\mathbf{t}, \tag{50}$$

$$\left[\mathbf{z}^{(1)}\left(\mathbf{z}^{(1)}\right)^T + \mathbf{z}^{(2)}\left(\mathbf{z}^{(2)}\right)^T\right]\mathbf{M}^T\mathbf{M}\mathbf{z}^{(1)} = \mathbf{M}^T\mathbf{M}\mathbf{z}^{(1)}, \tag{51}$$

and, for $k \geq 2$,

$$\left[\mathbf{z}^{(k-1)}\left(\mathbf{z}^{(k-1)}\right)^T + \mathbf{z}^{(k)}\left(\mathbf{z}^{(k)}\right)^T + \mathbf{z}^{(k+1)}\left(\mathbf{z}^{(k+1)}\right)^T\right]\mathbf{M}^T\mathbf{M}\mathbf{z}^{(k)} = \mathbf{M}^T\mathbf{M}\mathbf{z}^{(k)}. \tag{52}$$

We see that (50) defines $\mathbf{z}^{(1)}$ as the unit vector in the direction of $\mathbf{M}^T\mathbf{t}$, and can have no terms from the right null space of $\mathbf{M}$. Equation (51) defines $\mathbf{z}^{(2)}$ as the unit vector found by removing the component of $\mathbf{M}^T\mathbf{M}\mathbf{z}^{(1)}$ in the direction of $\mathbf{z}^{(1)}$ and then normalizing. Equation (52) determines $\mathbf{z}^{(k+1)}$ as the unit vector along the component of $\mathbf{M}^T\mathbf{M}\mathbf{z}^{(k)}$ that is orthogonal to both $\mathbf{z}^{(k)}$ and $\mathbf{z}^{(k-1)}$. By construction,

$$\left(\mathbf{z}^{(i)}\right)^T\mathbf{z}^{(j)} = \delta_{ij}, \tag{53}$$

so the vectors are orthonormal (in infinite precision but not in practice).

Defining the constants

$$N_1 = \left|\mathbf{M}^T\mathbf{t}\right| = \left(\mathbf{z}^{(1)}\right)^T \mathbf{M}^T\mathbf{t}, \tag{54}$$

$$D_k = \left(\mathbf{z}^{(k)}\right)^T \mathbf{M}^T\mathbf{M}\mathbf{z}^{(k)} \quad \text{for} \quad k = 1, 2, \ldots, \tag{55}$$

and

$$N_{k+1} = \left(\mathbf{z}^{(k+1)}\right)^T \mathbf{M}^T\mathbf{M}\mathbf{z}^{(k)} \quad \text{for} \quad k = 1, 2, \ldots, \tag{56}$$

we then see that the equations (50)–(52) determine a tridiagonal system of the form

$$\left(\, 0 \quad \ldots \quad 0 \quad \mathbf{z}^{(k+1)} \,\right) N_{k+1} + \mathbf{Z}_k \begin{pmatrix} D_1 & N_2 & & & \\ N_2 & D_2 & N_3 & & \\ & N_3 & D_3 & N_4 & \\ & & \ddots & \ddots & \ddots \\ & & & N_k & D_k \end{pmatrix} = \mathbf{M}^T\mathbf{M}\mathbf{Z}_k \quad \text{for} \quad 2 \le k \le r, \tag{57}$$

where the orthogonal matrix composed of the resulting orthonormal vectors is

$$\mathbf{Z}_k = \left(\, \mathbf{z}^{(1)} \quad \mathbf{z}^{(2)} \quad \mathbf{z}^{(3)} \quad \ldots \quad \mathbf{z}^{(k)} \,\right). \tag{58}$$

The process stops (in principle) when $k = r$ (the rank of the matrix) because then $N_{r+1} = 0$, or is numerically negligible. It follows that this tridiagonalization process results in the identity

$$\mathbf{M}^T\mathbf{M} = \mathbf{Z}_r\mathbf{T}_r\mathbf{Z}_r^T, \tag{59}$$

where the tridiagonal matrix of coefficients is defined by

$$\mathbf{T}_k = \begin{pmatrix} D_1 & N_2 & & & \\ N_2 & D_2 & N_3 & & \\ & N_3 & D_3 & N_4 & \\ & & \ddots & \ddots & \ddots \\ & & & N_k & D_k \end{pmatrix} \quad \text{for} \quad 2 \le k \le r. \tag{60}$$

Since $\mathbf{T}_r$ is invertible (by the definition of the rank r of $\mathbf{M}$),

$$\left(\mathbf{M}^T\mathbf{M}\right)^\dagger = \mathbf{Z}_r \left(\mathbf{T}_r\right)^{-1} \mathbf{Z}_r^T. \tag{61}$$

The solution to the least-squares inversion problem may therefore be written as

$$\mathbf{s} = \mathbf{Z}_r \left(\mathbf{T}_r\right)^{-1} \mathbf{Z}_r^T\mathbf{M}^T\mathbf{t} = N_1\mathbf{Z}_r \left(\mathbf{T}_r\right)^{-1} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{62}$$

where we used (50), (54), and the fact that

$$\mathbf{Z}_r^T\mathbf{z}^{(1)} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{63}$$

It follows that we need only the first column of the inverse of $\mathbf{T}_r$ to solve this inversion problem.

Since Lanczos's method directly produces a sequence of orthonormal vectors in the model space, it is straightforward to see that the model resolution matrix for this method is given by

$$\mathcal{R}_{model} = \mathbf{M}^T\mathbf{M}\left(\mathbf{M}^T\mathbf{M}\right)^\dagger = \mathbf{Z}_r\mathbf{Z}_r^T = \sum_{k=1}^{r} \mathbf{z}^{(k)}\left(\mathbf{z}^{(k)}\right)^T, \tag{64}$$

which is also clearly symmetric. It is however more difficult to compute the data resolution matrix. Using the fact that

$$\mathcal{R}_{data} = \mathbf{M}\mathbf{M}^\dagger = \mathbf{M}\left(\mathbf{M}^T\mathbf{M}\right)^\dagger\mathbf{M}^T \tag{65}$$

together with (61), we find that

$$\mathcal{R}_{data} = \mathbf{M}\mathbf{Z}_r\left(\mathbf{T}_r\right)^{-1}\mathbf{Z}_r^T\mathbf{M}^T. \tag{66}$$

It is clear that both of the resolution matrices are symmetric if we compute the full Lanczos inverse. If the process is terminated early so that $k < r$ by setting the constant N_{k+1} equal to zero, then a Lanczos approximate inverse is given by

$$\mathbf{X}^{(k)} = \mathbf{Z}_k\left(\mathbf{T}_k\right)^{-1}\mathbf{Z}_k^T\mathbf{M}^T \tag{67}$$

so that $\mathbf{X}^{(k)}\mathbf{M}\mathbf{X}^{(k)} = \mathbf{X}^{(k)}$ for all k, but $\mathbf{M}\mathbf{X}^{(k)}\mathbf{M} \neq \mathbf{M}$ if $k < r$. The effective resolution matrices are given by $\mathcal{E}_{model} = \mathbf{X}^{(k)}\mathbf{M}$ and $\mathcal{E}_{data} = \mathbf{M}\mathbf{X}^{(k)}$, or are found by replacing $\mathbf{Z}_r$ and $\mathbf{T}_r$ in (64) and (66) by $\mathbf{Z}_k$ and $\mathbf{T}_k$, respectively. Clearly, the effective resolutions are also symmetric. Thus, three out of four Moore-Penrose conditions are satisfied by (67), and all four are satisfied (for infinite precision) when $k = r$.

An alternative method of computing the resolution matrices involves noting that the vector sequence $\mathbf{z}^{(k)}$ may also be used as the starting set of vectors for the method of conjugate directions. Then, we can produce a set of conjugate vectors from this orthogonal set and easily compute both the model resolution matrix and the data resolution matrix if desired. This alternative requires additional work, however, to produce the sequence of conjugate vectors and it also generally produces a model resolution that is not symmetric, and therefore difficult to interpret. Thus, the Lanczos procedure appears to be superior to conjugate directions/gradients from this point of view. (But we have demonstrated alternatives for computing resolution with conjugate gradients in the preceding subsection.) The main disadvantage of the Lanczos method is that the amount of storage increases with the iteration number k, since all the vectors $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k)}$ must be stored until the final calculation of $\mathbf{s}^{(k)} = \mathbf{X}^{(k)}\mathbf{t}$. In contrast, conjugate gradients requires a fixed amount of storage, but we must sacrifice an easily interpretable model resolution to gain this advantage.

2.6 Bidiagonalization, LSQR, and the modified LSQR algorithm

The main results of this subsection were published previously in Berryman [1994b].

The method of Lanczos [1950] may be applied to any square, symmetric matrix inversion problem. To solve the least-squares inversion problem, the method may therefore be applied to the normal matrix $\mathbf{M}^T\mathbf{M}$. However, dynamic range problems due to poor conditioning of $\mathbf{M}$ worsen when the Lanczos method is applied directly to the normal equations since the singular values are squared in $\mathbf{M}^T\mathbf{M}$. It would therefore be preferable to avoid squaring the singular values if possible. Another serious drawback of this procedure is that, whereas $\mathbf{M}$ is always a sparse matrix in seismic tomography (each ray path traverses only a small number of cells in the model), $\mathbf{M}^T\mathbf{M}$ is nevertheless a full matrix. Thus, the straightforward application of the Lanczos method to the least-squares estimation problem immediately destroys one of the key simplifying features of the matrix $\mathbf{M}$.

To take full advantage of the sparsity of $\mathbf{M}$, we can "complete the square" with $\mathbf{M}$ and consider either the problem

$$\begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{t} \\ \mathbf{0} \end{pmatrix}, \tag{68}$$

where $\mathbf{r} = \mathbf{t} - \mathbf{M}\mathbf{s}$ is the residual vector, or the alternate form

$$\begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{M}^T\mathbf{t} \end{pmatrix}, \tag{69}$$

where -$\mathbf{p} = \mathbf{M}\mathbf{s}$ is the predicted traveltime. The idea of completing the square this way appears to be due Lanczos [1960], and the application of the idea to computations such as bidiagonalization and singular-value decomposition is due to Golub and Kahan [1965].

The first approach (68) is the one on which LSQR [Paige and Saunders, 1982] is based, and it has been used extensively in seismic tomography applications as discussed recently by van der Sluis and van der Vorst [1987; 1990]. The second approach (69) has been named LSCG by Paige [1974] and the approach is discussed but rejected by Paige and Saunders [1982] on the grounds that it must have inferior numerical performance compared to LSQR. We wish to revisit (69) in the present work, because it has special properties that make it the preferred form when trying to analyze the progress of the inversion process using resolution matrices.

Equations (68) and (69) again both have the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ with square symmetric matrix $\mathbf{A} = \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix}$, unknown vectors $\mathbf{x}^T = (\,\mathbf{r}^T \quad \mathbf{s}^T\,)$ or $\mathbf{x}^T = (\,\mathbf{p}^T \quad \mathbf{s}^T\,)$, and data vectors $\mathbf{b}^T = (\,\mathbf{t}^T \quad \mathbf{0}^T\,)$ or $\mathbf{b}^T = (\,\mathbf{0}^T \quad -\mathbf{t}^T\mathbf{M}\,)$, respectively. The LSQR algorithm of Paige and Saunders [1982] can be obtained in a relatively straightforward manner from (68) by applying the Lanczos method to the system as we demonstrate in the sequel. Another somewhat older approach called Craig's method [Craig, 1955] has been compared and contrasted with LSQR recently by Saunders [1995]. LSQR was designed originally for overdetermined problems, whereas Craig's method was designed for underdetermined or compatible problems. But Saunders [1995] shows that the two methods, although not precisely equivalent, do solve the same problems, especially when damping is incorporated as we will do later. We refer the interested reader to Saunders' paper and to his references for further information about technical issues concerning relationships between LSQR and Craig that are beyond our present scope.

In this discussion of LSQR, we continue to assume infinite precision in the computations in order to develop the main ideas. We discuss practical consequences of finite precision in Section 3.

2.6.1 Choice of starting vector

Relatively little attention has been paid to the choice of starting vector for LSQR and the other variants of this method such as Craig's method [Craig, 1955], LSCG [Paige, 1974] and LSLQ [Paige and Saunders, 1982]. Discussions have naturally centered around issues of the numerical performance of these algorithms for solving the problem $\mathbf{Ms} = \mathbf{t}$ in the least-squares or some other sense. But as we have been stressing in this article, it is also important to evaluate what has been accomplished once the procedure terminates. To do this requires either the computation of the resolution matrices or something similar. As we will now explain, the choice of starting vector has a very important effect on our ability to compute the resolution matrix for the data space.

The choices of starting vector in the two cases (68) and (69) are very different. These two vectors are orthogonal, but more importantly their nonzero components lie in different vector spaces. The choice in (68) is all in the data space, whereas the choice in (69) is all in the model space. The distinction is important both for resolution analysis and for numerical reasons. Paige and Saunders [1982] show that even though the two methods appear to be analytically equivalent, the choice in (69) is expected to be numerically inferior. The algorithm resulting from applying the Lanczos procedure to (69) is effectively solving the normal equations $\mathbf{M}^T\mathbf{Ms} = \mathbf{M}^T\mathbf{t}$, whereas the algorithm resulting from (68) is instead solving $\mathbf{Ms} = \mathbf{t}$ in the pseudo-inverse (Moore-Penrose) sense.

Although these conclusions are no doubt correct when the system is compatible, a problem can arise with (68) when the system is inconsistent. In particular, if the data vector $\mathbf{t}$ contains components in the null space of $\mathbf{M}^T$, then LSQR — when used without modification — will produce a sequence of vectors in the data space having some very strange characteristics. Since the very first vector in the sequence is the data vector itself, it will appear that we have perfectly resolved the data at the very first iteration step, which is clearly incorrect. Furthermore, all subsequent vectors generated in the data space will be orthogonal (at least in principle) to the first vector, and therefore cannot be used in any way to shed light on the true resolution of the data that has been achieved.

If we use (69) instead of (68), then the problem described in the preceding paragraph cannot occur. The starting vector is $\mathbf{M}^T\mathbf{t}$, and any components of $\mathbf{t}$ that might lie in the null space of $\mathbf{M}^T$ are projected out of the problem by the operator itself. Thus, by using (69), we gain most of the advantages of LSQR, but have the added advantage that we can compute both $\mathcal{R}_{model}$ and $\mathcal{R}_{data}$ in a very natural way, as we will see in the following discussion. We will show later that the numerically unfavorable performance of the choice (69) affects the calculation by forcing us to reorthogonalize the resulting Krylov sequence, but this is usually unavoidable anyway whenever the resolution matrices must be computed.

2.6.2 A variant of LSQR

As we have already discussed, the LSQR method is most commonly based on (68). However, for our present purposes, (69) is the superior choice for reasons having to do with the data resolution matrix as we discussed in the preceding subsection. This variant of LSQR is very closely related to LSCG [Paige, 1974] and to LSLQ [Paige and Saunders, 1982]. We will refer to it as the "modified LSQR," and the main modification is just the choice of starting vector.

We will now provide a derivation of the modified LSQR algorithm. First, consider a vector of the form

$$\mathbf{z}^{(k)} = \begin{pmatrix} \mathbf{f}^{(k)} \\ \mathbf{h}^{(k+1)} \end{pmatrix}. \tag{70}$$

Then, the Lanczos tridiagonalization process applied to (69) takes the form

$$\mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T \begin{pmatrix} \mathbf{0} \\ \mathbf{M}^T\mathbf{t} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{M}^T\mathbf{t} \end{pmatrix}, \tag{71}$$

together with equations

$$\left[\mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T + \mathbf{z}^{(2)} \left(\mathbf{z}^{(2)}\right)^T\right] \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(1)} = \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(1)}, \tag{72}$$

and, for $k > 2$,

$$\begin{aligned}\left[\mathbf{z}^{(k-1)} \left(\mathbf{z}^{(k-1)}\right)^T + \mathbf{z}^{(k)} \left(\mathbf{z}^{(k)}\right)^T + \mathbf{z}^{(k+1)} \left(\mathbf{z}^{(k+1)}\right)^T\right] & \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(k)} \\ &= \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(k)}.\end{aligned} \tag{73}$$

We first set $\mathbf{h}^{(1)} = \mathbf{0}$. Then, substituting (70) into (71), (72), and (73) gives

$$\mathbf{f}^{(1)} = \mathbf{0} \quad \text{and} \quad \mathbf{h}^{(2)} \left(\mathbf{h}^{(2)}\right)^T \mathbf{M}^T\mathbf{t} = \mathbf{M}^T\mathbf{t}, \tag{74}$$

$$\mathbf{f}^{(2)} \left(\mathbf{f}^{(2)}\right)^T \mathbf{M}\mathbf{h}^{(2)} = \mathbf{M}\mathbf{h}^{(2)} \quad \text{and} \quad \mathbf{h}^{(3)} = \mathbf{0}, \tag{75}$$

and, for $k \geq 1$,

$$\mathbf{f}^{(2k+1)} = \mathbf{0} \quad \text{and} \quad \left[\mathbf{h}^{(2k)} \left(\mathbf{h}^{(2k)}\right)^T + \mathbf{h}^{(2k+2)} \left(\mathbf{h}^{(2k+2)}\right)^T\right] \mathbf{M}^T\mathbf{f}^{(2k)} = \mathbf{M}^T\mathbf{f}^{(2k)}, \tag{76}$$

and

$$\left[\mathbf{f}^{(2k)} \left(\mathbf{f}^{(2k)}\right)^T + \mathbf{f}^{(2k+2)} \left(\mathbf{f}^{(2k+2)}\right)^T\right] \mathbf{M}\mathbf{h}^{(2k+2)} = \mathbf{M}\mathbf{h}^{(2k+2)} \quad \text{and} \quad \mathbf{h}^{(2k+3)} = \mathbf{0}. \tag{77}$$

Defining the constants

$$q_1 = |\mathbf{M}^T\mathbf{t}| = \mathbf{h}^{(2)}\mathbf{M}^T\mathbf{t}, \tag{78}$$

and, for $k \geq 1$,

$$q_{2k} = \left(\mathbf{h}^{(2k)}\right)^T \mathbf{M}^T \mathbf{f}^{(2k)} = \left(\mathbf{f}^{(2k)}\right)^T \mathbf{M}\mathbf{h}^{(2k)} \tag{79}$$

and

$$q_{2k+1} = \left(\mathbf{h}^{(2k+2)}\right)^T \mathbf{M}^T \mathbf{f}^{(2k)} = \left(\mathbf{f}^{(2k)}\right)^T \mathbf{M}\mathbf{h}^{(2k+2)}. \tag{80}$$

The orthogonal matrix of the vectors $\mathbf{f}^{(2k)}$ is defined by

$$\mathbf{F}_k = (\, \mathbf{f}^{(2)} \quad \mathbf{f}^{(4)} \quad \ldots \quad \mathbf{f}^{(2k)} \,), \tag{81}$$

while the corresponding matrix $\mathbf{H}_k$ of the vectors $\mathbf{h}^{(2k)}$ is defined by

$$\mathbf{H}_k = (\, \mathbf{h}^{(2)} \quad \mathbf{h}^{(4)} \quad \ldots \quad \mathbf{h}^{(2k)} \,). \tag{82}$$

Now we introduce the $k \times k$ upper bidiagonal matrix

$$\mathbf{Q}_k = \begin{pmatrix} q_2 & q_3 & & & \\ & q_4 & q_5 & & \\ & & q_6 & q_7 & \\ & & & \ddots & \ddots \\ & & & & q_{2k} \end{pmatrix}, \tag{83}$$

and the related $k \times (k+1)$ matrix

$$\overline{\mathbf{Q}}_k = \begin{pmatrix} q_2 & q_3 & & & & \\ & q_4 & q_5 & & & \\ & & q_6 & q_7 & & \\ & & & \ddots & \ddots & \\ & & & & q_{2k} & q_{2k+1} \end{pmatrix}, \tag{84}$$

obtained from $\mathbf{Q}_k$ by adding another column with just one nonzero element q_{2k+1}. The two bidiagonal systems may be written, for $2 \leq k \leq r$, as

$$\mathbf{H}_{k+1}\overline{\mathbf{Q}}_k^T = \mathbf{M}^T \mathbf{F}_k, \tag{85}$$

and

$$\mathbf{f}^{(2k+2)}\mathbf{e}_{k+1}^T q_{2k+2} + \mathbf{F}_k \overline{\mathbf{Q}}_k = \mathbf{M}\mathbf{H}_{k+1}. \tag{86}$$

We see that, when $k = r$ with r determined by the condition that $q_{2r+1} = 0$ (or is numerically negligible), multiplying (85) on the right by $\mathbf{F}_r^T$ and then taking the transpose yields the relation

$$\mathbf{M} = \mathbf{F}_r \mathbf{Q}_r \mathbf{H}_r^T. \tag{87}$$

This equality is exact for infinite precision and takes its usual approximate meaning in finite precision. The corresponding approximate inverse is

$$\mathbf{X}_r = \mathbf{H}_r \left(\mathbf{Q}_r\right)^{-1} \mathbf{F}_r^T. \tag{88}$$

Finally, the solution is given by

$$\mathbf{s} = \mathbf{X}_r\mathbf{t} = \mathbf{H}_r \left(\mathbf{Q}_r\right)^{-1} \mathbf{F}_r^T\mathbf{t}, \tag{89}$$

which cannot be simplified directly since, for the sequence following from (69), $\mathbf{t}$ is not orthogonal to any of the vectors $\mathbf{f}^{(2k)}$. Nevertheless, two further simplifications can be obtained.

To improve the efficiency of the computation, first note that the inverse matrix is upper triangular and has the simple form

$$\left(\mathbf{Q}_r\right)^{-1} = \begin{pmatrix} 1/q_2 & -q_3/q_2q_4 & q_3q_5/q_2q_4q_6 & \cdots \\ & 1/q_4 & -q_5/q_4q_6 & \cdots \\ & & 1/q_6 & \cdots \\ & & & \ddots \end{pmatrix}. \tag{90}$$

Although it appears that all elements of $\mathbf{H}_r$ must be retained to the end in order to compute $\mathbf{s}$, it turns out that if we define the $n \times r$ matrix

$$\mathbf{W}_r \equiv \mathbf{H}_r \left(\mathbf{Q}_r\right)^{-1} = \left(\mathbf{w}^{(2)} \quad \mathbf{w}^{(4)} \quad \ldots \quad \mathbf{w}^{(2r)} \right), \tag{91}$$

then the n-vectors $\mathbf{w}^{(2k)}$ for $k = 1, \ldots, r$ can be computed recursively according to

$$\mathbf{w}^{(2k)} = \frac{1}{q_{2k}}\mathbf{h}^{(2k)} - \frac{q_{2k-1}}{q_{2k}}\mathbf{w}^{(2k-2)} \quad \text{for} \quad k \geq 1. \tag{92}$$

The initial vector in the sequence is defined by

$$\mathbf{w}^{(0)} = \mathbf{0}. \tag{93}$$

The need for retaining earlier vectors $\mathbf{h}^{(2j)}$ for $j < k$, is eliminated using this recursion, since the pertinent information in the required form is already imbedded in the vectors $\mathbf{w}^{(2k)}$.

The second simplification results when we note that

$$\mathbf{F}_r\mathbf{Q}_r = \mathbf{G}_{r+1}\overline{\mathbf{B}}_r. \tag{94}$$

From (94), it follows that

$$\mathbf{f}^{(2k)}q_{2k} = \mathbf{g}^{(2k)}b_{2k} + \mathbf{g}^{(2k+2)}b_{2k+1} - \mathbf{f}^{(2k-2)}q_{2k-1}. \tag{95}$$

Using the facts that $\mathbf{t}^T \cdot \mathbf{g}^{(2)} = b_1$ and that $\mathbf{t}$ is orthogonal to all other $\mathbf{g}$s, we find the general result that

$$\mathbf{t}^T \cdot \mathbf{f}^{(2k)} = -\mathbf{t}^T \cdot \mathbf{f}^{(2k-2)}q_{2k-1}/q_{2k} = (-1)^k \frac{q_{2k-1}\cdots q_1}{q_{2k}\cdots q_2}. \tag{96}$$

Thus, we have found an explicit expression for the elements of the vector $\mathbf{F}_r^T\mathbf{t}$ appearing in (89).

The resulting algorithm is very similar to the one used in LSQR, although the starting point and some of the details of the derivation are slightly different.

The main point of this effort is that we end up with virtually the same algorithm as LSQR — apparently having all its good properties — but we can also compute the data resolution safely and easily if we choose to do so.

2.6.3 Formulas for resolution using LSQR

Our main conclusion is that the model resolution for this approach is given by

$$\mathcal{R}_{model} = \mathbf{M}^\dagger\mathbf{M} = \mathbf{X}_r\mathbf{M} = \mathbf{H}_r\mathbf{H}_r^T, \tag{97}$$

while the data resolution is

$$\mathcal{R}_{data} = \mathbf{M}\mathbf{M}^\dagger = \mathbf{M}\mathbf{X}_r = \mathbf{F}_r\mathbf{F}_r^T. \tag{98}$$

Equations (97) and (98) are both exact (at least in principle). The usual caveats apply if the process is terminated early, so the final iteration number is $k < r$. See Section 3 for further discussion of numerical issues, and discussion of reorthogonalization.

2.6.4 Diagonal resolution

If — as is often the case — the only resolution matrix desired is the model resolution, then it may also suffice to limit consideration to the diagonal components of the effective model resolution matrix

$$diag(\mathcal{E}^{(k)}_{model}) = \sum_{i=1}^{k} \begin{pmatrix} (h_1^{(2i)})^2 \\ (h_2^{(2i)})^2 \\ \vdots \\ (h_n^{(2i)})^2 \end{pmatrix}, \quad \text{for} \quad k = 1, \ldots, r. \tag{99}$$

This vector can be accumulated easily without subsequently retaining the components of the vectors $\mathbf{h}^{(2i)}$ for all $i = 1, \ldots, k$. The diagonal components are also the ones that are normally the easiest to interpret, since they are positive and generally lie between 0 and 1 and therefore may be treated as measures of the probability of having correctly resolved the model slowness value in a given cell. However, it should be emphasized that various authors, including Evans and Achauer [1993], caution that diagonal resolution alone can be misleading.

Of the methods commonly used to compute iterative inverses, this variant of LSQR is the only one known to the author that may be used to compute diagonal components of the data resolution (98) in analogy to (99). This result is another factor favoring the use of LSQR-type algorithms for seismic tomography problems.

2.7 LSQR for damped least-squares

If the ray-path matrix $\mathbf{M}$ is very poorly conditioned, it may be advisable to modify the normal equations with a damping term. Then, (69) changes to

$$\begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & -\mu\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} 0 \\ -\mathbf{M}^T\mathbf{t} \end{pmatrix}, \tag{100}$$

where $-\mathbf{p} = \mathbf{Ms}$ is still the predicted traveltime vector, but now the solution vector satisfies

$$\mathbf{s} = \left(\mathbf{M}^T\mathbf{M} + \mu\mathbf{I}\right)^{-1} \mathbf{M}^T\mathbf{t}. \tag{101}$$

The inverse matrix in (101) exists for any value of $\mu > 0$, since the eigenvalues of $\mathbf{M}^T\mathbf{M}$ are nonnegative and the addition of the diagonal term μ shifts all the eigenvalues up by μ.

Applying the Lanczos method to (100), we soon discover that the terms proportional to μ always cancel out of the equations so the hard part of the analysis is the same as for the undamped problem. Thus, with r being the rank of $\mathbf{M}$, we find

$$\mathbf{M} = \mathbf{G}_r\mathbf{B}_r\mathbf{H}_r^T \quad \text{and} \quad \mathbf{M}^\dagger = \mathbf{H}_r\mathbf{B}_r^{-1}\mathbf{G}_r^T, \tag{102}$$

as before. For infinite precision, the orthogonal matrices $\mathbf{G}_k$ and $\mathbf{H}_k$ for all k, in the range $1 \le k \le r$, are all identical to those for the undamped case.

Then, we can write the regularized normal matrix in terms of these orthogonal matrices and find its inverse. The matrix is given by

$$\mathbf{M}^T\mathbf{M} + \mu\mathbf{I} = \mathbf{H}_r\mathbf{B}_r^T\mathbf{B}_r\mathbf{H}^T + \mu\mathbf{I} = \mathbf{H}_r\left(\mathbf{B}_r^T\mathbf{B}_r + \mu\mathbf{I}\right)\mathbf{H}_r^T + \mu\left(\mathbf{I} - \mathbf{H}_r\mathbf{H}_r^T\right). \tag{103}$$

The inverse matrix is easily shown to be

$$\left(\mathbf{M}^T\mathbf{M} + \mu\mathbf{I}\right)^{-1} = \mathbf{H}_r\left(\mathbf{B}_r^T\mathbf{B}_r + \mu\mathbf{I}\right)^{-1}\mathbf{H}_r^T + \frac{1}{\mu}\left(\mathbf{I} - \mathbf{H}_r\mathbf{H}_r^T\right). \tag{104}$$

Substituting this result into (101) gives

$$\mathbf{s} = \mathbf{H}_r\left(\mathbf{B}_r^T\mathbf{B}_r + \mu\mathbf{I}\right)^{-1}\mathbf{B}_r^T\mathbf{G}_r^T\mathbf{t} = b_1\mathbf{H}_r\left(\mathbf{B}_r^T\mathbf{B}_r + \mu\mathbf{I}\right)^{-1}\mathbf{B}_r^T\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{105}$$

which should be compared to (89). Following the same analytical path used in the Lanczos method for damped least-squares, the result is

$$\mathbf{s} = \mathbf{Z}_r\left(\mathbf{T}_r + \mu\mathbf{I}\right)^{-1}\mathbf{Z}_r^T\mathbf{M}^T\mathbf{t} = D_1\mathbf{Z}_r\left(\mathbf{T}_r + \mu\mathbf{I}\right)^{-1}\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{106}$$

which should be compared to (62).

Using (102), it is straightforward to check that these two formulas have the same Moore-Penrose characteristics as other damped approximate inverses. The resulting resolution matrices are symmetric, but the two remaining uniqueness conditions are both violated for any $\mu > 0$. The same results hold for partial versions of these formulas, where the process is terminated early at some $k < r$ and rs are replaced by ks everywhere in the formulas (105) and (106).

2.8 Summary

The Table summarizes the results obtained for iterative inverses.

TABLE. Qualitative comparison of various iterative inverses.

	Moore-Penrose Conditions			
Iterative Inverse	$\mathbf{MXM}$ $= \mathbf{M}$	$\mathbf{XMX}$ $= \mathbf{X}$	$(\mathbf{XM})^T$ $= \mathbf{XM}$	$(\mathbf{MX})^T$ $= \mathbf{MX}$
Fully iterated CD, CG, Lanczos, or Modified LSQR	yes	yes	yes	yes
Full LSQR Inverse	yes	no	yes	yes
Partial Lanczos, or Modified LSQR	no	yes	yes	yes
Partial Conjugate Directions/gradients	no	yes	no	yes
Partial LSQR Inverse	no	no	yes	yes
Damped Lanczos or LSQR	no	no	yes	yes

3 Computational Issues: Necessity of Reorthogonalization

Suppose we want to invert a square, $n \times n$ symmetric matrix $\mathbf{A}$ and we want to do so using some iterative method like Lanczos, LSQR, conjugate gradients, etc. Then the iterative algorithm generally may be expressed in the form

$$\mathbf{A}\mathbf{Z}_k = \mathbf{Z}_k\mathbf{T}_k + N_{k+1}\mathbf{z}_{k+1}\mathbf{e}_k^T \tag{107}$$

for $k \leq r$ where r is the rank of the matrix $\mathbf{A}$ and the matrix $\mathbf{Z}_k$, given by

$$\mathbf{Z}_k = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \ldots & \mathbf{z}_k \end{pmatrix}, \tag{108}$$

is composed of the column vectors $\mathbf{z}_i$ for $i = 1, \ldots, k$ that have been generated so far in the iterative process. The other terms in (107) are the unit vector $\mathbf{e}_k$, with its single component in the kth position, and a norming constant N_{k+1}. The formula (107) with $\mathbf{T}$ being a tridiagonal matrix is just the iterative scheme of Lanczos.

Now the model resolution matrix $\mathcal{R}_{model}$ for the iterative scheme at the kth iteration is given by

$$\mathcal{R}_k = \mathbf{Z}_k \mathbf{Z}_k^T. \tag{109}$$

However, care must be taken to make sure that the $\mathbf{z}_i$s are really orthogonal as they are expected to be in this scheme. The tridiagonalization process produces a sequence of orthogonal vectors in principle, but in practice the orthogonality may break down after several iterations when the calculations are performed at finite precision. To demonstrate how the lack of orthogonality affects the process consider the following facts. Let the eigenvalues of the matrix $\mathbf{A}$ be ordered so that $\lambda_i^2 \neq 0$ for $i = 1, \ldots, r$ and $\lambda_i^2 = 0$ for $i = r+1, \ldots, n$. (We write the eigenvalues as squares because $\mathbf{A}$ often takes the form of a normal matrix $\mathbf{A} = \mathbf{M}^T\mathbf{M}$, in which case the singular values of $\mathbf{M}$ are the λ_is.) Then the trace of the matrix is just

$$\mathrm{Tr}\mathbf{A} = \sum_{i=1}^{r} \lambda_i^2. \tag{110}$$

The model resolution $\mathcal{R}_k$ is a projection operator onto a k-dimensional Krylov subspace (the one explored so far by the iterative method) of the n-dimensional space that is both the range and domain of $\mathbf{A}$. Taking the trace of $\mathcal{R}_k\mathbf{A}$ shows that

$$\mathrm{Tr}\left[\mathbf{Z}_k \mathbf{Z}_k^T \mathbf{A}\right] = \mathrm{Tr}\left[\mathbf{Z}_k^T \mathbf{A} \mathbf{Z}_k\right] = \sum_{i=1}^{k} \mathbf{z}_i^T \mathbf{A} \mathbf{z}_i. \tag{111}$$

Let $\mathbf{v}_i$ for $i = 1, \ldots, r$ be the normalized eigenvectors associated with the eigenvalues λ_i^2. Then, assuming only that $\mathbf{z}_1$ has no components in the null-space of $\mathbf{A}$, each of the iteration vectors $\mathbf{z}_i$ can be expanded in terms of these eigenvectors with the result that

$$\mathbf{z}_i = \sum_{j=1}^{r} \zeta_{ij} \mathbf{v}_j. \tag{112}$$

Similarly, the eigenvectors can be expanded in terms of the full set of normalized iteration vectors $\mathbf{z}_i$ according to

$$\mathbf{v}_j = \sum_{i=1}^{r} \zeta_{ij} \mathbf{z}_i, \tag{113}$$

where the same set of coefficients ζ_{ij} is used in both

$$\sum_{i=1}^{r} \zeta_{ij}^2 = 1 = \sum_{j=1}^{r} \zeta_{ij}^2 \tag{114}$$

in order for both sets of vectors to be normalized to unity.

Substituting (112) into (111), we find easily that

$$\mathrm{Tr}\left[\mathbf{Z}_k\mathbf{Z}_k^T\mathbf{A}\right] = \sum_{j=1}^{r}\lambda_j^2\left[\sum_{i=1}^{k}\zeta_{ij}^2\right] \leq \sum_{j=1}^{r}\lambda_j^2. \tag{115}$$

The inequality follows, since the coefficients ζ_{ij} are all real, and therefore

$$\sum_{j=1}^{k}\zeta_{ij}^2 \leq \sum_{j=1}^{r}\zeta_{ij}^2 \equiv 1. \tag{116}$$

Thus, we can expect that as long as the vectors returned by the iterative scheme are orthogonal, the effective trace of the operator $\mathbf{T}$ must remain smaller than that of the original operator $\mathbf{A}$. If this constraint is violated (we provide examples of this in the next section), then we know that the $\mathbf{z}_i$s generated so far are not orthogonal and furthermore that this lack of orthogonality is having serious negative consequences on our ability to compute the resolution operator correctly.

For a more extensive discussion of the effects of finite precision on such computations, the reader is referred to Greenbaum and Strakos [1992].

4 Crosswell Tomography Application

Figures 1-4 provide some numerical examples comparing and contrasting the results obtained using standard SVD resolution calculations with the LSQR [Paige and Saunders, 1982] resolution calculations as described in an earlier paper [Berryman, 1994b]. We consider a 4×4 model using strictly crosswell data, so there are 16 source/receiver pairs as well as 16 cells in 2D. Model slowness values are shown in the upper block of each figure, while diagonal resolution values are shown in the lower block. The first two examples (Figures 1 and 2) show results for the actual model used to compute the traveltime data (see Berryman [1990] for a description of the code used to generate both the forward and inverse solutions). The second two examples (Figures 3 and 4) show results obtained after 15 iterations of the reconstruction code of Berryman [1990]. The LSQR resolution examples (Figures 2 and 4) were computed using ten iterations of the LSQR algorithm, so the maximum size of the resolved model vector space has dimension ten. To aid in the comparison, the SVD resolution examples use only the 10 eigenvectors associated with the 10 largest eigenvalues of the ray-path matrix. We find the results are in qualitatively good agreement. Better quantitative agreement is not anticipated because the 10-dimensional vector spaces spanned by these two approximations, although having large regions of overlap, will nevertheless almost always differ to some degree.

4.1 Tests of orthogonalization

For the LSQR algorithm, it is easy to check the trace of the effective operator since it is just the sum of the squares of the elements in the resulting bidiagonal matrix. To test the ideas of the preceding section, we first perform an LSQR inversion while imposing full reorthogonalization [Arnoldi, 1951] at each step. Doing so, we obtain Figure 5, showing that the effective operator trace converges monotonically to the true trace from below as expected.

In contrast, Figure 6 shows that without reorthogonalization the trace of the effective operator has already exceeded the upper bound — thus demonstrating that the eighth direction vector found by LSQR has at least some component parallel to one of the earlier vectors and also along some singular vector with large singular value. Observing the later behavior of this curve, regions of small slopes may correspond to allowable (orthogonal) direction vectors, but the regions of large slope must be resampling the regions of the vector space with the largest singular values. This observation shows that the rebirth of nonorthogonal vectors does not happen just at the eighth iteration, but is a recurring problem.

These two examples show clearly that full reorthogonalization works very well and that failure to do any reorthogonalization can lead to serious problems with the set of direction vectors generated by such schemes. To make progress, we want to know whether full reorthogonalization is required, or whether some type of partial reorthogonalization might be as effective as full reorthogonalization. We want to explore the tradeoffs between cost of the partial reorthogonalization and the benefits to be derived from it. Figures 7–11 explore these issues.

Figures 7–9 refer to the same model considered in Figures 1-6. Figure 7 shows that orthogonalizing against the first 1, 2, or 3 vectors improves the results progressively, the more vectors are used for the reorthogonalization. Figure 8 shows that reorthogonalization against the 1, 2, or 3 most recently generated vectors does not work as well as the previous Figure. Reorthogonalizing against both 1, 2, or 3 early and late vectors gives virtually identical results in Figure 9 as those results observed in Figure 7.

Figures 10 and 11 refer to a model of 16×8 cells similar to that considered earlier by Berryman [1990]. Figure 10 shows that reorthogonalizing against only the first and last vectors generated in the iteration sequence is quite ineffective at reducing the nonorthogonal vectors generated. Figure 11 shows that reorthogonalizing against the first 35 vectors produces a major improvement, without significant orthogonalization problems out to 90 iterations, instead of the less than 35 iterations found for serious errors to arise with only one vector reorthogonalization.

We conclude that reorthogonalization is effective and the partial reorthogonalization is most effective when the vectors chosen for the reorthogonalization set are those from the early part of the iteration sequence. The reason that these vectors are best to use is presumably because they correspond to directions that have components parallel to directions in the space that are singular vectors of the operator being inverted with largest singular values. These vectors like to be reborn in this process and reorthogonalization is an effective means of preventing multiply copies of the same dominant vectors from recurring in the iteration sequence.

5 Conclusion

The methods of computing resolution matrices that have been illustrated here may be easily generalized to a variety of other iterative and approximate inversion methods. We have explored partial reorthogonalization methods for iterative methods and have found that using a subset of the early vectors generated in the iteration sequence is most effective at reducing unwanted occurrences of nonorthogonal vectors in the later parts of the iteration sequence. These early vectors correspond to directions that have components along the singular vectors with the largest singular values, and these are precisely the vectors we most need to exclude from the later iterations. Such recurrences may not adversely affect the inversion itself, but do make the computation of the resolution matrices (operators) much more complicated than if the orthogonality is enforced.

Acknowledgments

I thank S. F. Ashby, R. Burridge, J. F. Claerbout, S. Fomel, D. M. Goodman, J. B. Keller, F. Muir, G. Nolet, G. Papanicolaou, B. N. Parlett, S. Ronen, M. A. Saunders, W. W. Symes, H. A. van der Vorst, and J. J. Zucca for helpful discussions, lectures, collaborations, correspondence and/or criticism. This work was performed under the auspices of the U. S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 and supported by the Environmental Management Science Program with oversight by the Engineering Research Division of the Office of Basic Energy Sciences, Office of Energy Research, and by the Office of Science and Technology, Office of Environmental Management. Early work in this area was performed while the author was visiting the Geophysics Department at Stanford University. Partial support for this visit was provided by the Stanford Exploration Project, whose sponsors are hereby gratefully acknowledged.

References

Arnoldi, W. E., 1951, The principle of minimized iterations in the solution of the matrix eigenvalue problem, *Quart. Appl. Math.* **9**, 17–29.

Ashby, S. F., T. A. Manteuffel, and P. E. Saylor, 1990, A taxonomy for conjugate gradient methods, *SIAM J. Numer. Anal.* **27**, 1542–1568.

Berryman, J. G., 1990, Stable iterative reconstruction algorithm for nonlienar traveltime tomography, *Inverse Problems* **6**, 21–42.

Berryman, J. G., 1994a, Resolution of iterative inverses in seismic tomography, in *Proceedings of the Cornelius Lanczos International Centenary Conference*, J. D. Brown, M. T. Chu, D. C. Ellison, and R. J. Plemmons (eds.), North Carolina State University, Raleigh, NC, December 12–17, 1993 (SIAM, Philadelphia, PA), pp. 297–299.

Berryman, J. G., 1994b, Tomographic resolution without singular value decomposition, in *Mathematical Methods in Geophysical Imaging II*, S. Hassanzadeh (ed.), Proceedings of

SPIE, Volume 2301, 24–29 July, 1994, San Diego, CA (SPIE, Bellingham, Washington), pp. 2–13.

Berryman, J. G., 1999, Analysis of approximate inverses in tomography. I. Resolution analysis of common inverses, *Optimization and Engineering*, submitted.

Berryman, J. G., and S. Fomel, 1996, Iterative methods of optimization with application to crosswell tomography, *Stanford Exploration Project*, Vol. **SEP-93**, 109–132.

Björck, A. and T. Elfving, 1979, Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations, *BIT* **19**, 145–163.

Craig, J. E., 1955, The N-step iteration procedures, *J. Math. and Phys.* **34**, 64–73.

Dines, K. A., and R. J. Lytle, 1979, Computerized geophysical tomography, *Proc. IEEE* **67**, 1065–1073.

Evans, J. R., and U. Achauer, 1993, Teleseismic velocity tomography using the ACH method: Theory and application to continental-scale studies, in *Seismic Tomography: Theory and Practice*, H. M. Iyer and K. Hirahara (eds.), Chapman and Hall, London, pp. 319–360.

Fletcher, R., and C. M. Reeves, 1964, Function minimization by conjugate gradients, *Computer Journal* **7**, 149–154.

Fomel, S., 1996, Least-square inversion with inexact adjoints, Method of conjugate direction: A Tutorial, *Stanford Exploration Project*, Vol. **SEP-92**, 253–265.

Gilbert, P., 1972, Iterative methods for the three-dimensional reconstruction of an object from projections, *J. Theor. Biol.* **36**, 105–117.

Golub, G. and W. Kahan, 1965, Calculating the singular values and pseudo-inverse of a matrix, *SIAM J. Numer. Anal.* **2**, 205–224.

Golub, G. H., and C. F. Van Loan, 1989, *Matrix Computations*, Johns Hopkins University Press, Baltimore, p. 341.

Greenbaum, A., and Z. Strakos, 1992, Predicting the behavior of finite precision Lanczos and conjugate gradient computations, *SIAM J. Matrix Anal.* **13**, 121–137.

Hestenes, M. R., and E. Stiefel, 1952, Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Stan.* **B 49**, 409–436.

Ivansson, S., 1983, Remark on an earlier iterative tomographic algorithm, *Geophys. J. R. Astr. Soc.* **75**, 855–860.

Lanczos, C., 1950, An iterative method for the solution of the eigenvalue problem of lienar differential and integral operators, *J. Res. Nat. Bur. Stand.* **45**, 255–282.

Luenberger, D. G., 1973, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading Massachusetts, Chapters 7 and 8, pp. 148–155, 168–186.

Minkoff, S. E., 1996, A computationally feasible approximate resolution matrix for seismic inverse problems, *Geophys. J. Int.* **126**, 345–359.

Paige, C. C., and M. A. Saunders, 1982, LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Softw.* **8**, 43–71.

Saunders, M. A., 1995, Solution of sparse rectangular systems using LSQR and Craig, *BIT* **35**, 588–604.

van der Sluis, A., and H. A. van der Vorst, 1987, Numerical solution of large, sparse linear algebraic systems arising from tomographic problems, in *Seismic Tomogrpahy – With Applications in Global Seismology and Exploration Geophysics*, G. Nolet (ed.), Reidel, Dordrecht, Holland, Chapter 3, pp. 49–83.

van der Sluis, A., and H. A. van der Vorst, 1990, SIRT- and CG-type methods for the iterative solution of sparse linear least-squares problems, *Linear Algebra Appl.* **130**, 257–302.

Varga, R. S., 1962, *Matrix Iterative Analysis*, Prentice-Hall, Englewood, Cliffs, NJ, pp. 16–17, 141, 159.

Vasco, D. W., J. E. Peterson, Jr., and E.L. Majer, 1996, Nonuniqueness in traveltime tomography: Ensemble inference and cluster analysis, *Geophysics* **61**, 1209–1227.

Zhang, J., and G. A. McMechan, 1995, Estimation of resolution and covariance for large matrix inversions, *Geophys. J. Int.* **121**, 409–426.

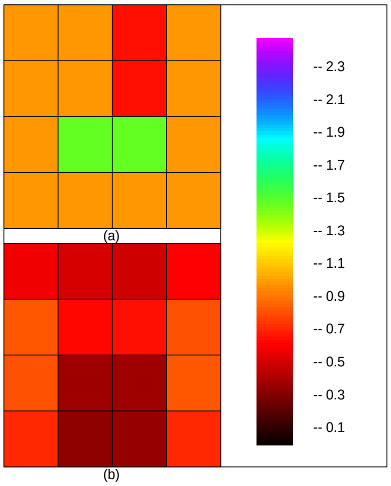

Figure 1: Target model slowness (a) and resolution (b) for truncated SVD using 10 largest eigenvalues.

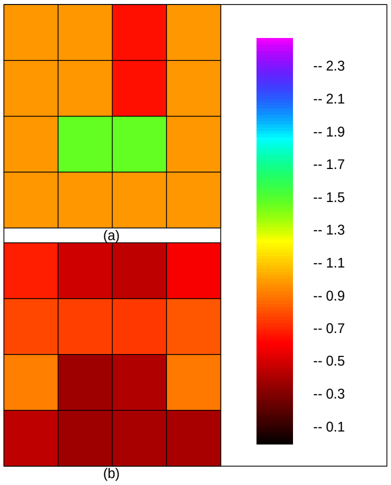

Figure 2: Target model slowness (a) and resolution (b) for LSQR after 10 iteratons.

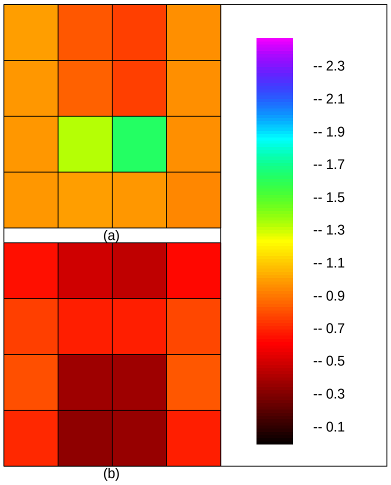

Figure 3: Slowness (a) and resolution (b) for truncated SVD using 10 largest eigenvalues.

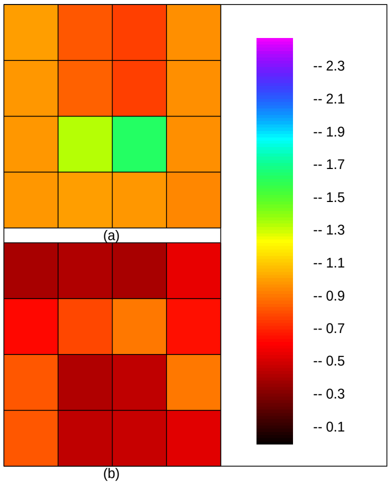

Figure 4: Reconstructed slowness (a) and resolution (b) for LSQR after 10 iterations.

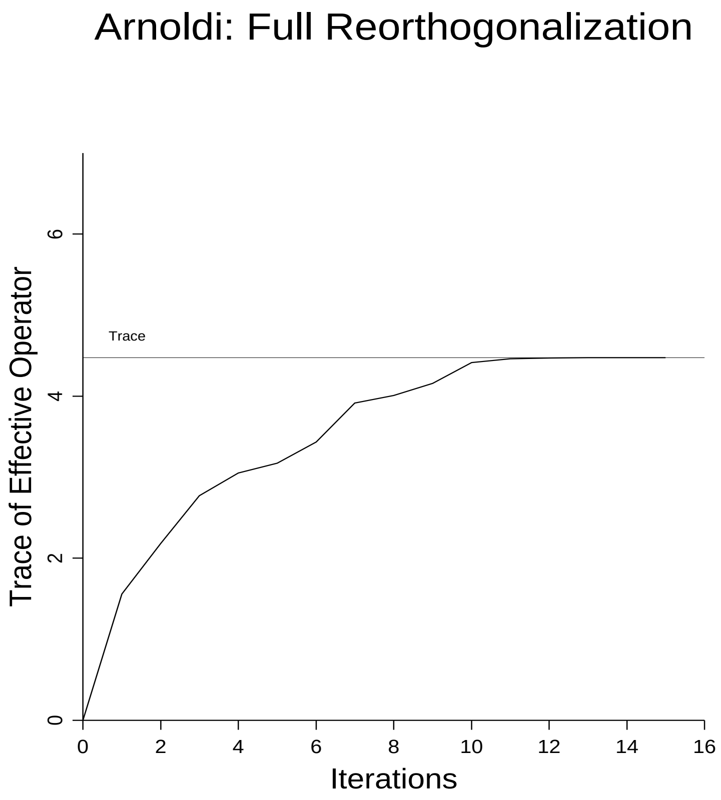

Figure 5: LSQR with full (Arnoldi) reorthogonalization.

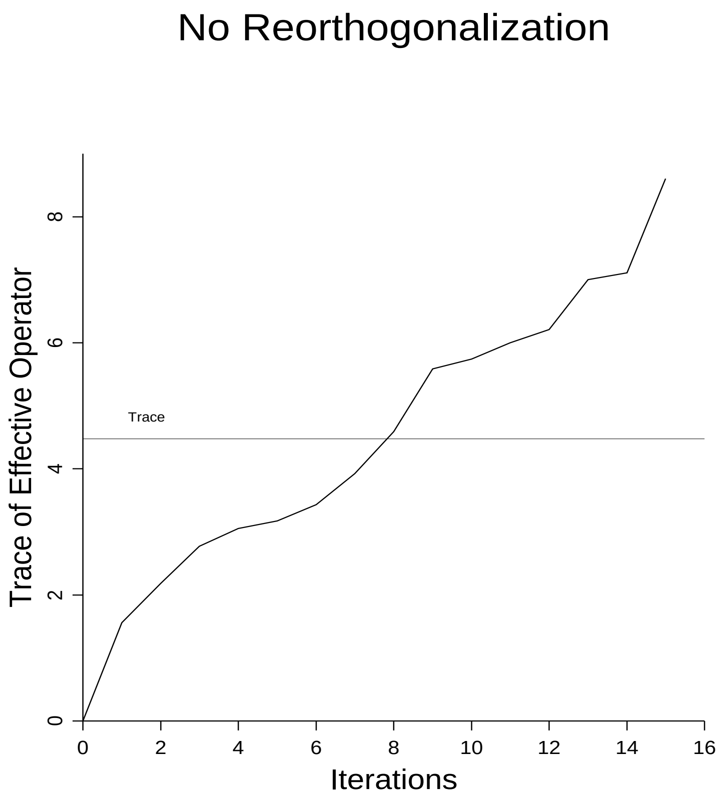

Figure 6: LSQR with no reorthogonalization.

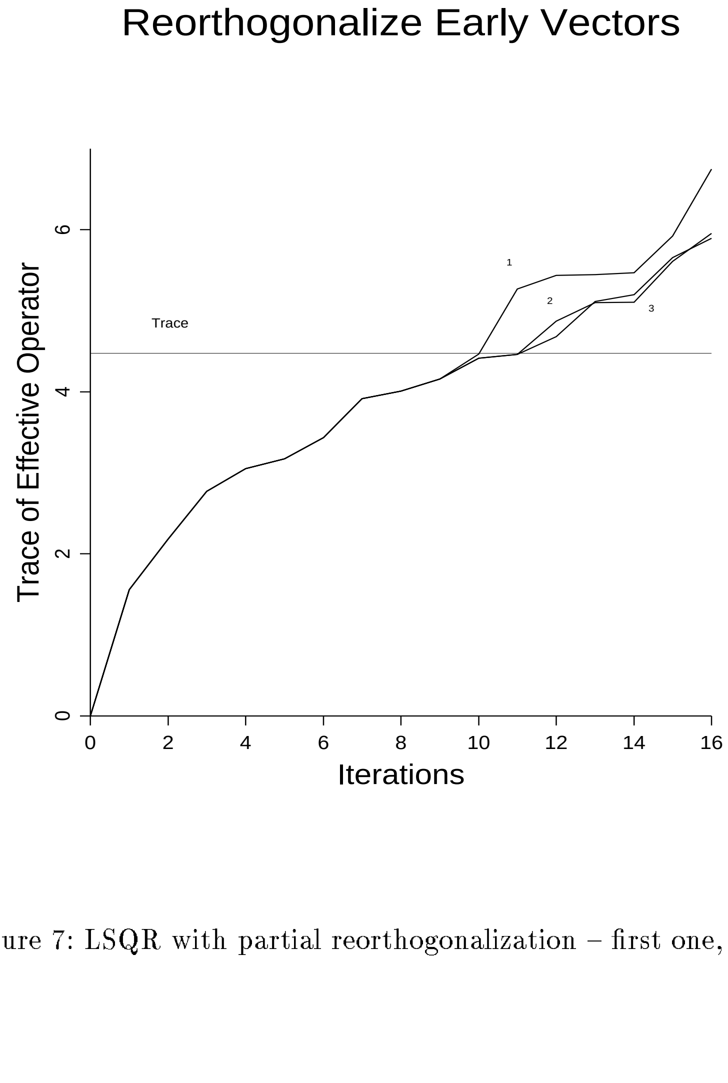

Figure 7: LSQR with partial reorthogonalization – first one, two, or three.

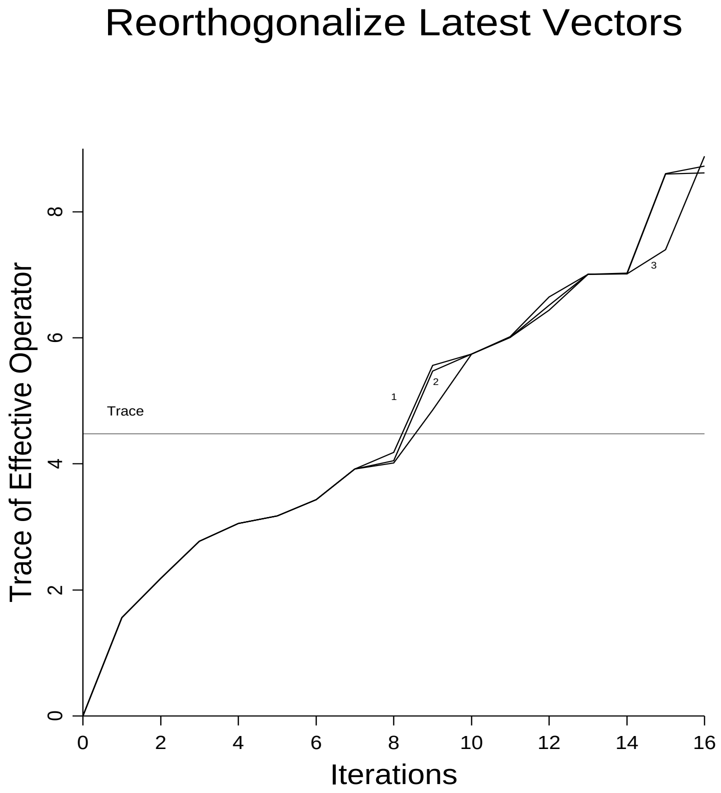

Figure 8: LSQR with partial reorthogonalization – last one, two, or three.

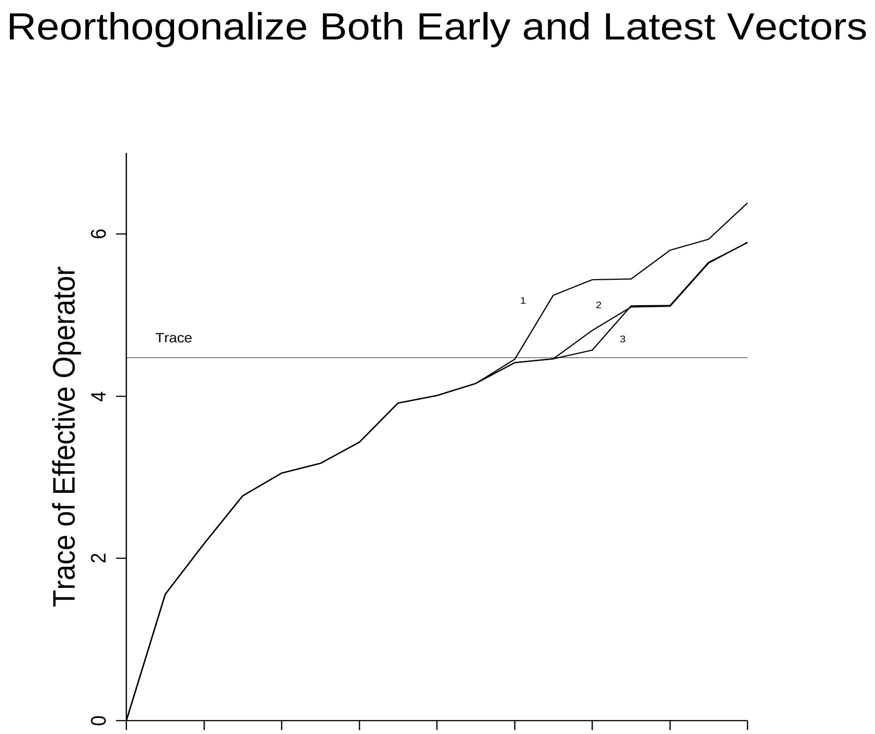

Figure 9: LSQR with partial reorthogonalization – using both first and last one, two, or three.

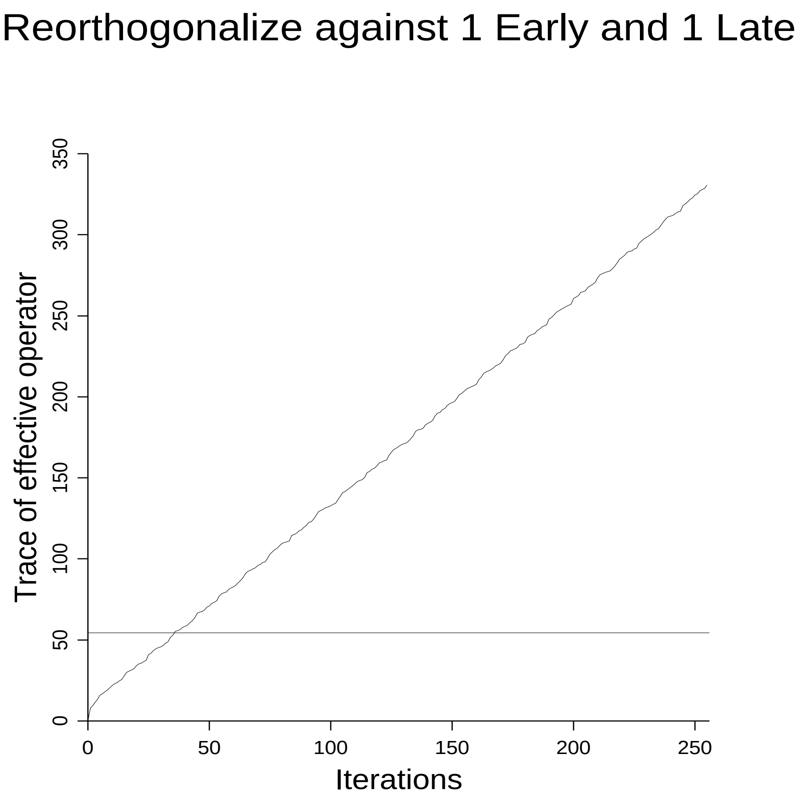

Figure 10: LSQR with partial reorthogonalization – using first and last vector.

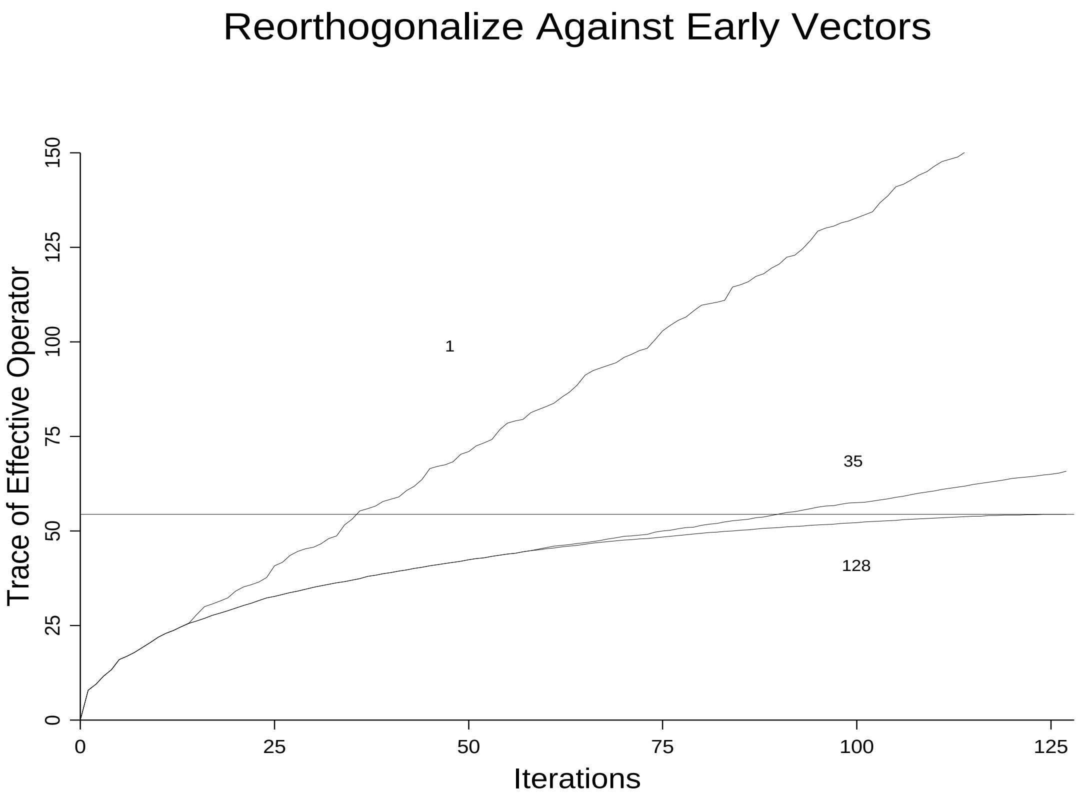

Figure 11: LSQR with partial reorthogonalization – using first, or first 35 vectors, or full reorthogonalization.