Tomographic resolution without singular value decomposition

James G. Berryman

Lawrence Livermore National Laboratory
P. O. Box 808 L-202
Livermore, CA 94551-9900

ABSTRACT

An explicit procedure is presented for computing both model and data resolution matrices within a Paige-Saunders LSQR algorithm for iterative inversion in seismic tomography. These methods are designed to avoid the need for an additional singular value decomposition of the ray-path matrix. The techniques discussed are completely general since they are based on the multiplicity of equivalent exact formulas that may be used to define the resolution matrices. Thus, resolution matrices may also be computed for a wide variety of iterative inversion algorithms using the same ideas.

Keywords: seismic tomography, inversion, resolution matrices

1 INTRODUCTION

Linear tomographic reconstruction schemes have well-defined resolution properties.[1] Yet, the resolution matrices summarizing these properties are not computed as often as they might be — at least in part — because of the common misconception that resolution matrices can only be found using singular value decomposition (SVD). Since SVD is generally the most computationally intensive of all commonly used matrix inversion methods, this approach is often prohibited by the time or expense involved, and in some cases it is rendered impossible by the size of the inversion problem. Other matrix inversion methods such as conjugate gradients, conjugate directions, Lanczos, and LSQR[2−5] are used more often in tomography codes, because of their smaller storage and computing requirements. In his recent review article, Nolet[6] states that "The LSQR method, due to Paige and Saunders (1982), is the most efficient method so far available to solve linear tomographic systems ...." It would therefore clearly improve the state of our practical knowledge about tomographic resolution if methods for computing the resolution matrices were available for all (or at least for the most highly regarded) iterative matrix inversion schemes. Furthermore, since these iterative methods are almost always terminated long before the full range of the original matrix has been probed, a method of computing resolution for iterative procedures directly (without SVD) is essential if we are to understand the resolution characteristics of the actual numerical inversion procedures we use.

2 RESOLUTION MATRICES

The central role of resolution in geophysical inversion problems has been emphasized by Backus and Gilbert,[7] who base their general inversion methods on techniques designed to optimize the resolution of the resulting geophysical model obtained from processed data. Discussions of resolution also played a central role in the classic paper of Aki *et al.*[8] on large scale seismic tomography. Such discussions continue to play a key role in the interpretation of inversion results in seismology. For example, in their recent review article — which also includes an extensive discussion of resolution in seismic inverse problems — Evans and Achauer[9] state that "... evaluating the whole resolution matrix, not just its diagonal elements, is a required part of interpreting [the reconstructed model]." Indeed, any inverse method based on incomplete or imperfect data should be expected to produce a filtered or blurred image of the object or region of interest. Apparent resolution is therefore an important figure of merit for practical imaging and inversion schemes, since the user of such schemes will eventually want to know what the smallest object is that can be distinguished. Judgments concerning reliability of features observed in the reconstructed image depend strongly on our understanding of inherent resolution capabilities of the inverse method used to produce the image. Therefore, a convenient quantitative measure of pointwise or cellwise model reliability is certainly helpful and perhaps essential for subsequent interpretation.

Resolution matrices for linear inversion problems can be understood most easily by considering a matrix equation of the form

$$\mathbf{M}\mathbf{s} = \mathbf{t} \tag{1}$$

and first asking the question: Given matrix $\mathbf{M}$ and data vector $\mathbf{t}$, what model vector $\mathbf{s}$ solves this equation? When the matrix $\mathbf{M}$ is square and invertible, the answer to the question is relatively easy: $\mathbf{s} = \mathbf{M}^{-1}\mathbf{t}$, with $\mathbf{M}^{-1}$ being the usual matrix inverse of $\mathbf{M}$. However, it often happens in geophysical inversion problems that $\mathbf{M}$ is not square, or not invertible even if it is square. In these situations, the least-squares method is often used, resulting in the normal equations

$$\mathbf{M}^T\mathbf{M}\mathbf{s} = \mathbf{M}^T\mathbf{t}, \tag{2}$$

which can often be solved approximately for $\mathbf{s}$ since the normal matrix $\mathbf{M}^T\mathbf{M}$ is square and symmetric — although it may still be singular. It proves convenient now to introduce an approximate inverse $\mathbf{M}^\dagger$ called the Moore-Penrose pseudoinverse.[10–12] This generalized inverse is the unique matrix that satisfies the four conditions: $\mathbf{M}\mathbf{M}^\dagger\mathbf{M} = \mathbf{M}$, $\mathbf{M}^\dagger\mathbf{M}\mathbf{M}^\dagger = \mathbf{M}^\dagger$, $\mathbf{M}^\dagger\mathbf{M} = (\mathbf{M}^\dagger\mathbf{M})^T$, and $\mathbf{M}\mathbf{M}^\dagger = (\mathbf{M}\mathbf{M}^\dagger)^T$. Although other choices for the approximate inverse are known (for example, see Rao[13]), we will restrict discussion here to this best known approximate inverse. Then, after multiplying (1) on the left by $\mathbf{M}^\dagger$, we find

$$\mathbf{M}^\dagger\mathbf{M}\mathbf{s} = \mathbf{M}^\dagger\mathbf{t}. \tag{3}$$

If it were true that $\mathbf{M}^\dagger\mathbf{M} = \mathbf{I}$ (the identity matrix), then we would have solved the inversion problem exactly, and also have perfect resolution. But it is precisely in those problems for which no such inverse exists that we need to consider the analysis that follows. In particular, we define the matrix coefficient of $\mathbf{s}$ in (3) as the resolution matrix

$$\mathcal{R} \equiv \mathbf{M}^\dagger\mathbf{M}. \tag{4}$$

The deviations of $\mathcal{R}$ from the identity matrix $\mathbf{I}$, *i.e.,* the components of the difference matrix $\mathbf{I} - \mathcal{R}$, determine the degree of distrust we should have in the components of the solution vector $\mathbf{s}$ that are most poorly resolved.

For definiteness, consider the seismic tomography problem (see Figure 1): $\mathbf{M}$ is an $m \times n$ ray-path matrix, $\mathbf{t}$ is a data m-vector of first arrival traveltimes, and $\mathbf{s}$ is the model n-vector for (possibly rectangular) cells of constant slowness (inverse velocity). We seek the slownesses $\mathbf{s}$ given the measured traveltimes in $\mathbf{t}$ and the estimates of the ray paths between source and receiver locations contained in the matrix $\mathbf{M}$ (see Berryman[14] and references therein). Then, the resolution matrix defined in (4) is the model resolution, since the slowness vector is the desired model of acoustic wave slowness. We can also define a data resolution matrix. First, multiply (3) on the left by $\mathbf{M}$ so

$$\mathbf{M}\mathbf{M}^\dagger\mathbf{M}\mathbf{s} = \mathbf{M}\mathbf{s} = \mathbf{M}\mathbf{M}^\dagger\mathbf{t}, \tag{5}$$

and then compare (5) to (1), noting that the matrix product multiplying $\mathbf{t}$ should equal the identity matrix if the approximate inverse $\mathbf{M}^\dagger$ is a true inverse. Again, deviations of this matrix from the identity provide information

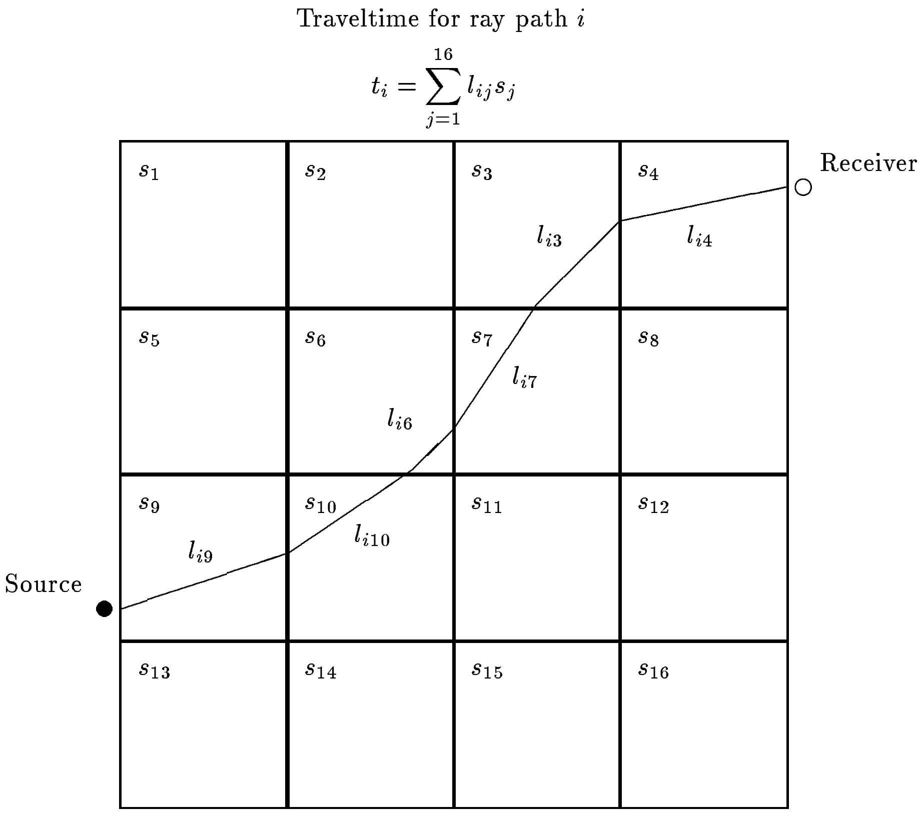

Figure 1: Schematic illustration of ray paths through a slowness model with rectangular cells.

about the degree to which the solution we compute makes use of all the data in $\mathbf{t}$. Thus, we find that the data resolution matrix[1,15] is defined by

$$\mathcal{R}_{data} \equiv \mathbf{M}\mathbf{M}^{\dagger}, \tag{6}$$

while the resolution matrix defined previously in (4) is the model resolution[1,7]

$$\mathcal{R}_{model} \equiv \mathbf{M}^{\dagger}\mathbf{M}. \tag{7}$$

Furthermore, for seismic inversion, we must also concern ourselves with mathematial nonlinearities involved in the process of finding a ray-path matrix $\mathbf{M}$ that is consistent with the model $\mathbf{s}$. For the present purposes, we assume that $\mathbf{M}$ and $\mathbf{s}$ are the final (and mutually consistent) products of an iterative algorithm.[14] Then, the question of resolution needs to be studied carefully in order to explore fully the range of possible solutions resulting from inherent nonuniqueness of the inverse problem.

We can better understand the significance of these two resolution matrices by considering the singular value decomposition (SVD) of the matrix $\mathbf{M}$, given by

$$\mathbf{M} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \mathbf{v}_i^T, \tag{8}$$

where the m-vectors $\mathbf{u}_i$ and n-vectors $\mathbf{v}_i$ are the eigenvectors of $\mathbf{M}$ determined by $\mathbf{M}\mathbf{v}_i = \lambda_i \mathbf{u}_i$ and $\mathbf{u}_i^T\mathbf{M} = \lambda_i \mathbf{v}_i^T$ and the λ_is are the eigenvalues. The eigenvectors are also assumed to satisfy orthonormality conditions $\mathbf{u}_i^T\mathbf{u}_j = \delta_{ij}$ and $\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}$. The rank of $\mathbf{M}$ (the number of nonzero eigenvalues) has a value $r \leq \min(m,n)$. The Moore-Penrose pseudoinverse is then known to be given by

$$\mathbf{M}^\dagger = \sum_{i=1}^{r} \lambda_i^{-1} \mathbf{v}_i \mathbf{u}_i^T, \tag{9}$$

so the resolution matrices are written explicitly in terms of sums of the outer products of the eigenvectors as

$$\mathcal{R}_{model} = \sum_{i=1}^{r} \mathbf{v}_i \mathbf{v}_i^T \tag{10}$$

and

$$\mathcal{R}_{data} = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{u}_i^T. \tag{11}$$

When displayed in this form, it is clear that the resolution matrices simply express the completeness of the resolved model or data spaces respectively. They are projection operators on the span of the resolved parts of the model and data vector spaces.

3 COMPUTING RESOLUTION

Now it is important to recognize that, although the resolution matrices have generally been *defined* by equations (6) and (7) — or by (10) and (11) (which implicitly assume that a singular value decomposition has been performed), it may nevertheless be possible *to compute* these matrices in other ways. Of particular importance is the computation of an effective inverse matrix $\mathbf{X}$ generated by an iterative inversion procedure.

To establish the plausibility of computing resolution without singular value decomposition, we first consider a simple pedagogical example that would not be ideal for computations. For convenience we define η to be a parameter having the significance of a continuous iteration number and let $\mathbf{X}(\eta)$ be the current approximation to the pseudoinverse $\mathbf{M}^\dagger$. Then the current value of the approximate solution vector is given by $\mathbf{s}(\eta) = \mathbf{X}(\eta)\mathbf{t}$ and the unresolved part of the data vector is clearly given by the difference vector $\Delta\mathbf{t} = \mathbf{t} - \mathbf{M}\mathbf{X}(\eta)\mathbf{t}$. The length of this vector is a scalar measure of the unresolved portion of the data. If we design the iterative inversion scheme to decrease the length of this vector progressively as $\eta \to \infty$, we note that the derivative of its square with respect to the continuous iteration number η is given by

$$\frac{\partial}{\partial\eta}\mathbf{t}^T(\mathbf{I} - \mathbf{X}^T\mathbf{M}^T)(\mathbf{I} - \mathbf{M}\mathbf{X})\mathbf{t} = -\mathbf{t}^T\frac{\partial\mathbf{X}^T}{\partial\eta}\mathbf{M}^T(\mathbf{I} - \mathbf{M}\mathbf{X})\mathbf{t} - \mathbf{t}^T(\mathbf{I} - \mathbf{X}^T\mathbf{M}^T)\mathbf{M}\frac{\partial\mathbf{X}}{\partial\eta}\mathbf{t}. \tag{12}$$

A sufficient condition for the traveltime data resolution to improve continuously as $\eta \to \infty$ is then (see Lu and Berryman[16]) the equation of motion for $\mathbf{X}$ given by

$$\frac{\partial\mathbf{X}}{\partial\eta} = \gamma\mathbf{M}^T(\mathbf{I} - \mathbf{M}\mathbf{X}), \tag{13}$$

where $\gamma > 0$ is some arbitrary scalar that determines the rate of convergence. It follows by construction that the right hand side of (12) is always negative or zero. Thus, the nonnegative length $|\mathbf{t} - \mathbf{MX}(\eta)\mathbf{t}|$ is a continuously decreasing function of the iteration parameter η as long as $\mathbf{X}(\eta)$ has the equation of motion given by (13). Clearly, the right hand side of (13) vanishes if and only if the approximate inverse matrix satisfies

$$\mathbf{M}^T\mathbf{MX}(\infty) = \mathbf{M}^T, \tag{14}$$

which is equivalent to the normal equations of least-squares, so $\mathbf{X}(\infty) = \mathbf{M}^\dagger$ as expected.[12]

If we define effective resolution matrices $\mathcal{E}_{model}(\eta) \equiv \mathbf{X}(\eta)\mathbf{M}$ and $\mathcal{E}_{data}(\eta) \equiv \mathbf{MX}(\eta)$, then it follows from (13) that

$$\frac{\partial \mathcal{E}_{model}}{\partial \eta} = \gamma \mathbf{M}^T\mathbf{M}(\mathbf{I} - \mathcal{E}_{model}) \tag{15}$$

and

$$\frac{\partial \mathcal{E}_{data}}{\partial \eta} = \gamma \mathbf{MM}^T(\mathbf{I} - \mathcal{E}_{data}). \tag{16}$$

Assuming initial conditions for the integration are given by $\mathcal{E}_{model}(0) = \mathbf{O}$ and $\mathcal{E}_{data}(0) = \mathbf{O}$, it is easy to see that the solutions of these equations will be symmetric matrices as desired. This approach becomes an iterative procedure when we solve the equations numerically by discretizing the independent parameter η and stepping from one discrete value of η to the next.

The procedure just outlined establishes that iterative procedures for computing resolution matrices are certainly possible. However, this particular method is less than ideal because it would be computationally intensive. So next we consider a general procedure that could be applied to many iterative methods that are used in practice.

To find a more efficient approach that is also inherently symmetric, first analyze the SVD of the normal matrix, *i.e.*,

$$\mathbf{M}^T\mathbf{M} = \sum_{i=1}^r \lambda_i \mathbf{v}_i \mathbf{u}_i^T \sum_{j=1}^r \lambda_j \mathbf{u}_j \mathbf{v}_j^T = \sum_{i=1}^r \lambda_i^2 \mathbf{v}_i \mathbf{v}_i^T. \tag{17}$$

Then, we can easily show that

$$\left(\mathbf{M}^T\mathbf{M}\right)^\dagger \mathbf{M}^T\mathbf{M} = \sum_{i=1}^r \mathbf{v}_i\mathbf{v}_i^T = \mathcal{R}_{model}. \tag{18}$$

It is equally straightforward to show that an alternative is given by

$$\mathbf{M}^T\left(\mathbf{MM}^T\right)^\dagger \mathbf{M} = \sum_{i=1}^r \mathbf{v}_i\mathbf{v}_i^T = \mathcal{R}_{model}, \tag{19}$$

a formula that is automatically symmetric. Similarly, we find that the data resolution is given by the various alternative forms

$$\mathbf{MM}^\dagger = \mathbf{MM}^T\left(\mathbf{MM}^T\right)^\dagger = \mathbf{M}\left(\mathbf{M}^T\mathbf{M}\right)^\dagger \mathbf{M}^T = \sum_{j=1}^r \mathbf{u}_j\mathbf{u}_j^T = \mathcal{R}_{data}. \tag{20}$$

Iterative methods such as the method of Lanczos[17] for solving (1) in the least-squares sense often compute the pseudoinverse — not of $\mathbf{M}$ itself but rather — of the normal matrix $\mathbf{M}^T\mathbf{M}$. The appearance of the pseudoinverse $(\mathbf{M}^T\mathbf{M})^\dagger$ of the normal matrix in these alternative expressions for the resolution matrices (18)–(20) then provides both a motivation and a clue to constructing methods for computing resolution with such iterative methods. Thus, in the following discussion, we are able to show how these various equivalent formulas for the resolution matrices may be used efficiently, in the course of a routine computation whose primary goal is to find an approximate solution to $\mathbf{Ms} = \mathbf{t}$ by iterative methods.

4 BIDIAGONALIZATION AND THE LSQR ALGORITHM

The method of Lanczos[17] may be applied to any square, symmetric matrix inversion problem. To solve the least-squares inversion problem, the method may 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 eigenvalues are squared in $\mathbf{M}^T\mathbf{M}$. It would therefore be preferable to avoid squaring the eigenvalues 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 a full matrix. Thus, the straightforward application of Lanczos's 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 could have "completed the square" with $\mathbf{M}$ and considered instead either the problem

$$\begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{t} \\ \mathbf{0} \end{pmatrix}, \tag{21}$$

where $\mathbf{r} = \mathbf{t} - \mathbf{Ms}$ is the residual vector, or the alternate problem

$$\begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\mathbf{M}^T\mathbf{t} \end{pmatrix}, \tag{22}$$

where -$\mathbf{p} = \mathbf{Ms}$ is the predicted traveltime. Equations (21) and (22) again both have the form $\mathbf{Ax} = \mathbf{b}$ 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 first approach has been discussed recently by van der Sluis and van der Vorst,[2,3] while the second approach has better numerical properties and has been treated by both Golub and Kahan[4] and Paige and Saunders.[5] The modification of Lanczos's method resulting from the use of (22) is now generally known as the LSQR algorithm of Paige and Saunders.[5]

In this Section, we assume infinite precision in the computations in order to develop the main ideas. We discuss practical consequences of finite precision briefly in Section 5.

4.1 A variant of LSQR

To provide a somewhat different derivation of the LSQR algorithm, consider

$$\mathbf{z}^{(k)} = \begin{pmatrix} \mathbf{f}^{(k)} \\ \mathbf{h}^{(k+1)} \end{pmatrix}. \tag{23}$$

Then, the tridiagonalization process takes the form

$$\mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T \begin{pmatrix} \mathbf{0} \\ \mathbf{M}^T\mathbf{t} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{M}^T\mathbf{t} \end{pmatrix}, \tag{24}$$

together with equations

$$\left[\mathbf{z}^{(1)} \left(\mathbf{z}^{(1)}\right)^T + \mathbf{z}^{(2)} \left(\mathbf{z}^{(2)}\right)^T\right] \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(1)} = \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(1)}, \tag{25}$$

and, for $k > 2$,

$$\begin{aligned} &\left[\mathbf{z}^{(k-1)} \left(\mathbf{z}^{(k-1)}\right)^T + \mathbf{z}^{(k)} \left(\mathbf{z}^{(k)}\right)^T + \mathbf{z}^{(k+1)} \left(\mathbf{z}^{(k+1)}\right)^T\right] \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(k)} \\ &\qquad = \begin{pmatrix} \mathbf{I} & \mathbf{M} \\ \mathbf{M}^T & \mathbf{O} \end{pmatrix} \mathbf{z}^{(k)}. \end{aligned} \tag{26}$$

We first set $\mathbf{h}^{(1)} = \mathbf{0}$. Then, substituting (23) into (24), (25), and (26) gives

$$\mathbf{f}^{(1)} = \mathbf{0} \quad \text{and} \quad \mathbf{h}^{(2)} \left(\mathbf{h}^{(2)}\right)^T \mathbf{M}^T \mathbf{t} = \mathbf{M}^T \mathbf{t}, \tag{27}$$

$$\mathbf{f}^{(2)} \left(\mathbf{f}^{(2)}\right)^T \mathbf{M}\mathbf{h}^{(2)} = \mathbf{M}\mathbf{h}^{(2)} \quad \text{and} \quad \mathbf{h}^{(3)} = \mathbf{0}, \tag{28}$$

and, for $k \geq 1$,

$$\mathbf{f}^{(2k+1)} = \mathbf{0} \quad \text{and} \quad \left[\mathbf{h}^{(2k)} \left(\mathbf{h}^{(2k)}\right)^T + \mathbf{h}^{(2k+2)} \left(\mathbf{h}^{(2k+2)}\right)^T\right] \mathbf{M}^T \mathbf{f}^{(2k)} = \mathbf{M}^T \mathbf{f}^{(2k)}, \tag{29}$$

and

$$\left[\mathbf{f}^{(2k)} \left(\mathbf{f}^{(2k)}\right)^T + \mathbf{f}^{(2k+2)} \left(\mathbf{f}^{(2k+2)}\right)^T\right] \mathbf{M}\mathbf{h}^{(2k+2)} = \mathbf{M}\mathbf{h}^{(2k+2)} \quad \text{and} \quad \mathbf{h}^{(2k+3)} = \mathbf{0}. \tag{30}$$

Defining the constants

$$q_1 = |\mathbf{M}^T \mathbf{t}| = \mathbf{h}^{(2)} \mathbf{M}^T \mathbf{t}, \tag{31}$$

and, for $k \geq 1$,

$$q_{2k} = \left(\mathbf{h}^{(2k)}\right)^T \mathbf{M}^T \mathbf{f}^{(2k)} = \left(\mathbf{f}^{(2k)}\right)^T \mathbf{M}\mathbf{h}^{(2k)} \tag{32}$$

and

$$q_{2k+1} = \left(\mathbf{h}^{(2k+2)}\right)^T \mathbf{M}^T \mathbf{f}^{(2k)} = \left(\mathbf{f}^{(2k)}\right)^T \mathbf{M}\mathbf{h}^{(2k+2)}. \tag{33}$$

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

$$\mathbf{F}_k = \left(\, \mathbf{f}^{(2)} \quad \mathbf{f}^{(4)} \quad \ldots \quad \mathbf{f}^{(2k)} \,\right), \tag{34}$$

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

$$\mathbf{H}_k = \left(\, \mathbf{h}^{(2)} \quad \mathbf{h}^{(4)} \quad \ldots \quad \mathbf{h}^{(2k)} \,\right). \tag{35}$$

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

$$\mathbf{Q}_k = \begin{pmatrix} q_2 & q_3 & & & \\ & q_4 & q_5 & & \\ & & q_6 & q_7 & \\ & & & \ddots & \ddots \\ & & & & q_{2k} \end{pmatrix}, \tag{36}$$

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

$$\overline{\mathbf{Q}}_k = \begin{pmatrix} q_2 & q_3 & & & & \\ & q_4 & q_5 & & & \\ & & q_6 & q_7 & & \\ & & & \ddots & \ddots & \\ & & & & q_{2k} & q_{2k+1} \end{pmatrix}, \tag{37}$$

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 $2 \le k \le r$, as

$$\mathbf{H}_{k+1}\overline{\mathbf{Q}}_k^T = \mathbf{M}^T\mathbf{F}_k, \tag{38}$$

and

$$\mathbf{f}^{(2k+2)}\mathbf{e}_{k+1}^T q_{2k+2} + \mathbf{F}_k\overline{\mathbf{Q}}_k = \mathbf{M}\mathbf{H}_{k+1}. \tag{39}$$

We see that, when $k = r$ with r determined by the condition that $q_{2r+1} = 0$ (or is numerically negligible), multiplying (38) on the right by $\mathbf{F}_r^T$ and then taking the transpose yields the relation

$$\mathbf{M} = \mathbf{F}_r\mathbf{Q}_r\mathbf{H}_r^T. \tag{40}$$

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

$$\mathbf{X}_r = \mathbf{H}_r\left(\mathbf{Q}_r\right)^{-1}\mathbf{F}_r^T. \tag{41}$$

Finally, the solution is given by

$$\mathbf{s} = \mathbf{X}_r\mathbf{t} = \mathbf{H}_r\left(\mathbf{Q}_r\right)^{-1}\mathbf{F}_r^T\mathbf{t}, \tag{42}$$

which cannot be simplified directly since $\mathbf{t}$ 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

$$\left(\mathbf{Q}_r\right)^{-1} = \begin{pmatrix} 1/q_2 & -q_3/q_2q_4 & q_3q_5/q_2q_4q_6 & \cdots \\ & 1/q_4 & -q_5/q_4q_6 & \cdots \\ & & 1/q_6 & \cdots \\ & & & \ddots \end{pmatrix}. \tag{43}$$

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

$$\mathbf{W}_r \equiv \mathbf{H}_r\left(\mathbf{Q}_r\right)^{-1} = \begin{pmatrix}\mathbf{w}^{(2)} & \mathbf{w}^{(4)} & \dots & \mathbf{w}^{(2r)}\end{pmatrix}, \tag{44}$$

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

$$\mathbf{w}^{(2k)} = \frac{1}{q_{2k}}\mathbf{h}^{(2k)} - \frac{q_{2k-1}}{q_{2k}}\mathbf{w}^{(2k-2)} \quad \text{for} \quad k \ge 1. \tag{45}$$

The initial vector in the sequence is defined by

$$\mathbf{w}^{(0)} = \mathbf{0}. \tag{46}$$

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

The second simplification results when we note that

$$\mathbf{F}_r\mathbf{Q}_r = \mathbf{G}_{r+1}\overline{\mathbf{B}}_r. \tag{47}$$

From (47), it follows that

$$\mathbf{f}^{(2k)}q_{2k} = \mathbf{g}^{(2k)}b_{2k} + \mathbf{g}^{(2k+2)}b_{2k+1} - \mathbf{f}^{(2k-2)}q_{2k-1}. \tag{48}$$

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

$$\mathbf{t}^T \cdot \mathbf{f}^{(2k)} = -\mathbf{t}^T \cdot \mathbf{f}^{(2k-2)} q_{2k-1}/q_{2k} = (-1)^k \frac{q_{2k-1} \cdots q_1}{q_{2k} \cdots q_2}. \tag{49}$$

Thus, we have found an explicit expression for the elements of the vector $\mathbf{F}_r^T \mathbf{t}$ appearing in (42).

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 — having all its good properties — but can also compute the data resolution safely and easily if we so choose.

4.2 Formulas for resolution using LSQR

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

$$\mathcal{R}_{model} = \mathbf{M}^\dagger \mathbf{M} = \mathbf{X}_r \mathbf{M} = \mathbf{H}_r \mathbf{H}_r^T, \tag{50}$$

while the data resolution is

$$\mathcal{R}_{data} = \mathbf{M}\mathbf{M}^\dagger = \mathbf{M}\mathbf{X}_r = \mathbf{F}_r \mathbf{F}_r^T. \tag{51}$$

Equations (50) and (51) is 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 5 for further discussion of numerical issues.

4.3 Diagonal resolution

If — as is most 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

$$diag(\mathcal{E}_{model}^{(k)}) = \sum_{i=1}^{k} \begin{pmatrix} (h_1^{(2i)})^2 \\ (h_2^{(2i)})^2 \\ \vdots \\ (h_n^{(2i)})^2 \end{pmatrix}, \quad \text{for} \quad k = 1, \ldots, r. \tag{52}$$

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,[9] caution that diagonal resolution alone can be misleading.

Of the methods commonly used to compute iterative inverses, the variant of LSQR considered here is the only one known to the author that may be used to compute diagonal components of the data resolution (51) in analogy to (52). This result is another factor favoring the use of LSQR for seismic tomography problems.

5 EFFECTS OF NUMERICAL ROUNDING ERRORS

In practice, computer precision is finite and the idealized analysis of the preceding section must be modified to account for the influence of rounding errors. The most important consequence of these errors is the fact that, after some number of iterations (say j), the basis vectors in LSQR are no longer orthonormal[18–22] to working precision. In particular, Paige[18–19] has shown that this effect is an inevitable consequence of the convergence of the eigenvalues of the truncated tridiagonal system in Lanczos' method to the actual eigenvalues of the full matrix $\mathbf{A} = \mathbf{M}^T\mathbf{M}$.

The standard method of dealing with this lack of orthogonality is reorthogonalization — used even by Lanczos in his original paper on this subject. Reorthogonalization at each stage is costly, both in terms of the computational time lost and also due to the added storage required to maintain copies of all the vectors $\mathbf{h}^{(2k)}$ needed for the orthogonalization step. Reorthogonalization of the $\mathbf{h}$ vectors requires storage of a matrix with size $n \times k$ at the kth iteration, whereas the model resolution matrix requires a fixed storage of size $\frac{1}{2}n(n+1)$, regardless of the iteration number. If we can afford the storage to compute model resolution, we can certainly afford the storage needed to perform reorthogonalization of the $\mathbf{h}$s. In fact, computation of model resolution matrix $\mathcal{R}_{model}$ can be delayed until the end of the iteration process, since the $\mathbf{h}$ vectors themselves contain all the information required to compute $\mathcal{R}_{model}$.

Similarly, reorthogonalization of the $\mathbf{f}$ vectors requires an $m \times k$ matrix for storage at the kth iteration, compared to a region of size $\frac{1}{2}m(m+1)$ for storage of the data resolution matrix elements. When $m >> n$, the storage issues for data resolution and reorthogonalization can provide significantly more limiting constraints than those found for the model resolution and reorthogonalization.

6 EXAMPLES AND CONCLUSIONS

Figure 2 provides some numerical examples comparing and contrasting the results obtained using standard SVD resolution calculations with the new LSQR resolution calculations described in this paper. 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 value is shown in the upper left corner of each cell, while diagonal resolution value is shown in the lower right corner. The top two examples (a,b) show results for the actual model used to compute the traveltime data (see Berryman[14] for a description of the code used to generate both the forward and inverse solutions). The bottom two examples (c,d) show results obtained after 15 iterations of the reconstruction code of Berryman.[14] The LSQR resolution examples (b,d) 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.

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. These other methods will be discussed in greater detail elsewhere, as will further applications to synthetic and real data. This paper has studied means of computing resolution matrices for fixed ray-path matrices. The more interesting and more difficult problem of determining the effective resolution of a nonlinear traveltime tomography algorithm based on analysis of feasibility constraints[14] will also be explored in a later publication.

ACKNOWLEDGMENTS

I thank S. F. Ashby, D. M. Goodman, F. Muir, G. Nolet, B. N. Parlett, M. A. Saunders, W. W. Symes, H. A. van der Vorst, and J. J. Zucca for helpful discussions and correspondence. 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 specifically by the Geosciences Research Program of the DOE Office of Energy Research within the Office of Basic Energy Sciences, Division of Engineering and Geosciences. Part of the work 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

1. D. D. Jackson, "Interpretation of inaccurate, insufficient and inconsistent data," *Geophys. J. R. Astron. Soc.* **28**, pp. 97–109, 1972.

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

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

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

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

6. G. Nolet, "Solving large linearized tomographic problems," in *Seismic Tomography: Theory and Practice*, H. M. Iyer and K. Hirahara (eds.), Chapter 9, pp. 227–247, Chapman and Hall, London, 1993.

7. G. Backus and F. Gilbert, "The resolving power of gross earth data," *Geophys. J. R. Astron. Soc.* **16**, pp. 169–205, 1968.

8. K. Aki, A. Christoffersson, and E. S. Husebye, "Determination of the three-dimensional structure of the lithosphere," *J. Geophys. Res.* **82**, pp. 277–296, 1977.

9. J. R. Evans and U. Achauer, "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.), pp. 319–360, Chapman and Hall, London, 1993.

10. E. H. Moore, *Bull. Amer. Math. Soc.* **26**, pp. 394–395, 1920.

11. R. Penrose, "A generalized inverse for matrices," *Proc. Cambridge Philos. Soc.* **51**, pp. 406–413, 1955.

12. R. Penrose, "On best approximation solutions of linear matrix equations," *Proc. Cambridge Philos. Soc.* **52**, pp. 17–19, 1955.

13. C. R. Rao, *Linear Statistical Inference and Its Applications*, Wiley, New York, pp. 24–26, 1965.

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

15. R. A. Wiggins, "The general linear inverse problem: Implications of surface waves and free oscillations for Earth structure," *Rev. Geophys. Space Phys.* **10**, pp. 251–285, 1972.

16. S.-Y. Lu and J. G. Berryman, "Inverse scattering, seismic traveltime tomography, and neural networks," *Int. J. Imaging Sys. Techn.* **2**, pp. 112–118, 1990.

17. C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bur. Stand.* **45**, pp. 255–282, 1950.

18. C. C. Paige, “Computational variants of the Lanczos method for the eigenproblem,” *J. Inst. Math. Appl.* **10**, pp. 373–381, 1972.

19. C. C. Paige, “Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix,” *J. Inst. Math. Appl.* **18**, pp. 341–349, 1976.

20. B. N. Parlett, *The Symmetric Eigenvalue Problem*, Chapter 13, pp. 257–287, Prentice-Hall, Englewood Cliffs, NJ, 1980.

21. J. Cullum and R. A. Willoughby, “Lanczos and the computation in specified intervals of the spectrum of large, sparse real symmetric matrices,” in *Sparse Matrix Proceedings 1978*, I. S. Duff and G. W. Stewart (eds.), pp. 220–255, SIAM, Philadelphia, 1979.

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

1.000 0.602	1.000 0.534	0.667 0.524	1.000 0.633
1.000 0.854	1.000 0.656	0.667 0.666	1.000 0.839
1.000 0.834	1.500 0.404	1.500 0.408	1.000 0.848
1.000 0.731	1.000 0.363	1.000 0.377	1.000 0.728

(a)

1.000 0.712	1.000 0.517	0.667 0.485	1.000 0.613
1.000 0.812	1.000 0.799	0.667 0.781	1.000 0.857
1.000 0.955	1.500 0.404	1.500 0.430	1.000 0.920
1.000 0.475	1.000 0.406	1.000 0.421	1.000 0.414

(b)

1.024 0.681	0.859 0.515	0.797 0.484	0.989 0.654
1.004 0.791	0.878 0.717	0.792 0.707	0.993 0.801
1.009 0.827	1.356 0.395	1.659 0.395	0.977 0.846
1.009 0.728	1.020 0.368	0.997 0.378	0.976 0.712

(c)

1.024 0.417	0.859 0.447	0.797 0.423	0.989 0.577
1.004 0.645	0.878 0.815	0.792 0.924	0.993 0.681
1.009 0.844	1.356 0.437	1.659 0.471	0.977 0.918
1.009 0.854	1.020 0.483	0.997 0.505	0.976 0.557

(d)

Figure 2: Comparison of results obtained using SVD and LSQR for an ideal model and the reconstructed model: (a) model and SVD resolution, (b) model and LSQR resolution, (c) reconstruction and SVD resolution, (d) reconstruction and LSQR resolution. In each cell, the upper left number is the slowness value and the lower right is the diagonal resolution value for the cell. The ten eigenvectors corresponding to the ten largest eigenvalues were used to compute the values of SVD resolution. Ten iterations were used to compute the values of LSQR resolution.