Obtaining smooth solutions to large, linear, inverse problems

John C. VanDecar* and Roel Snieder‡

ABSTRACT

It is not uncommon now for geophysical inverse problems to be parameterized by 10^4 to 10^5 unknowns associated with upwards of 10^6 to 10^7 data constraints. The matrix problem defining the linearization of such a system (e.g., $\underline{A}m = b$) is usually solved with a least-squares criterion ($m = (\underline{A}^t\underline{A})^{-1}\underline{A}^tb$). The size of the matrix, however, discourages the direct solution of the system and researchers often turn to iterative techniques such as the method of conjugate gradients to obtain an estimate of the least-squares solution. These iterative methods take advantage of the sparseness of $\underline{A}$, which often has as few as 2–3 percent of its elements nonzero, and do not require the calculation (or storage) of the matrix $\underline{A}^t\underline{A}$. Although there are usually many more data constraints than unknowns, these problems are, in general, underdetermined and therefore require some sort of regularization to obtain a solution. When the regularization is simple damping, the conjugate gradients method tends to converge in relatively few iterations. However, when derivative-type regularization is applied (first derivative constraints to obtain the *flattest* model that fits the data; second derivative to obtain the *smoothest*), the convergence of parts of the solution may be drastically inhibited. In a series of 1-D examples and a synthetic 2-D crosshole tomography example, we demonstrate this problem and also suggest a method of accelerating the convergence through the preconditioning of the conjugate gradient search directions. We derive a 1-D preconditioning operator for the case of first derivative regularization using a WKBJ approximation. We have found that preconditioning can reduce the number of iterations necessary to obtain satisfactory convergence by up to an order of magnitude. The conclusions we present are also relevant to Bayesian inversion, where a smoothness constraint is imposed through an a priori covariance of the model.

INTRODUCTION

A common problem in geophysics is to find a model of physical parameters m (e.g., density, elastic parameters) that predicts a set of measurements d (e.g., gravity anomalies, the traveltimes of elastic waves), given a physical theory relating the two:

$$f(m) = d. \qquad (1)$$

Usually, however, the data are inconsistent (because of measurement errors) and a wide range of models can explain the data equally well. If very few parameters are necessary to describe m, it is possible to perform a search of all realistic combinations of the parameters and present all those models that fit the measurements within a prescribed tolerance (usually set by the estimated level of data variance). If many parameters are necessary, as is normally the case with 2- or 3-D models, we usually wish to present a single "preferred model," along with measures of our confidence in this model and the degree of correlation between different parameters (Backus and Gilbert, 1967, 1968, 1970; Parker, 1977; Tarantola and Valette, 1982). The criteria we use to select this preferred model are necessarily subjective because of the ill-posed nature of the inverse problem.

If we have an a priori independent estimate of the model parameters, we may wish to choose the model predicting the measurements that is in some way closest to our initial model estimate (e.g., Tarantola, 1987). Unless this a priori model is well characterized (i.e., unless we have accurate estimates of the a priori model probability distribution), such criteria may result in models that contain more structure

than is necessary to explain the data. A second approach is to obtain the model predicting the measurements that contains the "least amount of structure" or in some way is the "least complicated" (Constable et al., 1987; Tarantola, 1987; Sambridge, 1990; VanDecar, 1991). Constable et al. (1987) appropriately termed the application of such constraints "Occam's Inversion," in reference to the principle that "it is vain to do with more what can be done with fewer." The amount of structure a model contains is often defined by its first or second derivative. If we choose the model with the smallest overall first derivative (in practice, first differences), then we obtain the "flattest" model that predicts the measurements, while with the second derivative (second differences) we obtain the "smoothest."

Up until this point, we have stated nothing about how we obtain this preferred model. If the physical relationship between model and data can be linearized, then equation (1) can be represented as a matrix product,

$$\underline{F}m = d. \tag{2}$$

Since it is likely that many different models m will predict the measurements d to within the desired accuracy, we need to add auxiliary information (e.g., in the form of linear equations) to equation (2) to obtain a single preferred solution. Our system then becomes,

$$\begin{pmatrix} F \\ \kappa \underline{D} \end{pmatrix} m = \begin{pmatrix} d \\ -\kappa \underline{D}m_0 \end{pmatrix}, \tag{3}$$

where $\underline{D}$ is a derivative operator in the case of a "minimum structure" inversion, or simply the identity matrix $\underline{I}$ in the case of damping toward some prescribed value. The constant κ controls the weight of these auxiliary constraints relative to the data equations. On the right-hand side, $\underline{D}m_0$ represents the degree to which the initial or reference model m_0 satisfies the auxiliary constraints (note that $m_0 = 0$ if m represents the total model, and not a perturbation).

If the data errors have zero mean and Gaussian distribution, we can obtain the maximum likelihood solution to equation (3) through the efficient method of least squares. This leads to the *normal equations*,

$$(\underline{F}'\underline{C}_d^{-1}\underline{F} + \kappa^2\underline{D}'\underline{D})m = \underline{F}'\underline{C}_d^{-1}d - \kappa^2\underline{D}'\underline{D}m_0, \tag{4}$$

where $\underline{C}_d$ represents the data covariance matrix. The solution to equation (4) is then,

$$m = (\underline{F}'\underline{C}_d^{-1}\underline{F} + \kappa^2\underline{D}'\underline{D})^{-1}(\underline{F}'\underline{C}_d^{-1}d - \kappa^2\underline{D}'\underline{D}m_0). \tag{5}$$

A comparison with equation (1.89) in Tarantola (1987) shows that the constraint $\kappa\underline{D}$ corresponds to $\underline{C}_m^{-1/2}$ in the formulation in Tarantola (1987), where $\underline{C}_m$ is the a priori model covariance matrix. In our formulation, we place constraints upon the *total* final model ($m_{total} = m + m_0$) rather than on perturbations from the reference model m_0 and therefore have the difference of a negative sign from the formulation of Tarantola (1987). The definitions,

$$\underline{A} = \begin{pmatrix} \underline{C}_d^{-1/2}\underline{F} \\ \kappa\underline{D} \end{pmatrix}$$

and $\tag{6}$

$$b = \begin{pmatrix} \underline{C}_d^{-1/2}d \\ -\kappa\underline{D}m_0 \end{pmatrix},$$

allow us to compactly represent equations (4) and (5) as,

$$\underline{A}'\underline{A}m = \underline{A}'b, \tag{7}$$

and

$$m = (\underline{A}'\underline{A})^{-1}\underline{A}'b, \tag{8}$$

for future reference.

The use of conjugate-gradients (CG) algorithms to solve least-squares inverse problems has expanded rapidly over the past decade in the field of geophysics (Nolet, 1985, 1987; Scales, 1987; Spakman and Nolet, 1988; Nolet and Snieder, 1990; VanDecar, 1991; van der Hilst et al., 1991). When faced with large sparse systems of linear equations, iterative methods such as CG have the advantage of being able to avoid the computation and storage of the normal equations in equation (4), thereby allowing much larger problems to be solved than would otherwise be computationally feasible (other row-action algorithms such as the back-projection methods, algebraic reconstruction technique (ART), and simultaneous iterative reconstruction technique (SIRT) also share this advantage). In geophysical tomography applications, iterative methods such as CG have been found to converge quickly when the regularization imposed is simple damping (Nolet, 1987). If, however, we wish to apply a criterion such as finding the flattest or smoothest model that satisfies the data, then the convergence may be severely inhibited. Since this "least structure" criterion would seem desirable (Constable et al., 1987; VanDecar, 1991; Sambridge, 1990; Neele et al., 1993), in this paper we investigate the effect of this criterion on solutions obtained via the CG algorithm. As noted in Stork (1992), velocity-reflector depth tradeoff in reflection seismology may also cause slow CG convergence because of the large ratio of largest to smallest eigenvalues in the resulting system.

THE CONJUGATE GRADIENTS ALGORITHM

For many geophysical problems, obtaining the least-squares solution to equation (3) as shown in equation (8) in a direct fashion would require a prohibitive amount of computer memory and time. This is because of the fact that although the matrix $\underline{A}$ is sparse (few of its elements are nonzero), the matrix $\underline{A}'\underline{A}$ is relatively dense, and matrix-matrix multiplication, even for sparse matrices, is expensive. Therefore, to take advantage of the sparseness of our system, we will turn to a CG method of solution.

The CG method we use was originally developed for the solution of large sparse systems by Hestenes and Stiefel (1952). Golub and Van Loan (1989) and Scales (1987) provide reviews of its derivation and use. In one form or another at present, the CG method is used widely to solve large geophysical inverse problems. The LSQR algorithm, devel-

oped by Paige and Saunders (1982), is a popular derivative of the method (Nolet, 1987; van der Sluis and van der Vorst, 1987). The Hestenes and Stiefel (1952) CG algorithm represents an acceleration of the well-known method of steepest descent (Golub and Van Loan, 1989). Rather than iteratively searching for a solution in purely the gradient directions, the CG method searches in the subspace spanned by the current residual direction and all previous directions. In ill-conditioned systems (where the ratio of the largest to smallest eigenvalues is large), the method of steepest descent may converge very slowly, while the CG method must converge to the least-squares solution in at most n iterations, n being the number of unknowns. Remarkably, this acceleration is obtained with little extra effort.

The conjugate gradient algorithm for solving problems as defined by equation (7) is simply:

$k=0$; $\mathbf{m}_0$ = initial solution estimate; $\mathbf{r}_0 = \underline{\mathbf{A}}'(\mathbf{b} - \underline{\mathbf{A}}\mathbf{m}_0)$,

while $\mathbf{r}_k \neq 0$

$$k = k + 1,$$
$$\text{if } k = 1$$

$$\mathbf{p}_1 = \mathbf{r}_0, \tag{9a}$$

else

$$\beta_k = \mathbf{r}_{k-1}^t \mathbf{r}_{k-1} / \mathbf{r}_{k-2}^t \mathbf{r}_{k-2}$$

$$\mathbf{p}_k = \mathbf{r}_{k-1} + \beta_k \mathbf{p}_{k-1} \tag{9b}$$

end

$$\mathbf{q}_k = \underline{\mathbf{A}}\mathbf{p}_k$$

$$\alpha_k = \mathbf{r}_{k-1}^t \mathbf{r}_{k-1} / \mathbf{q}_k^t \mathbf{q}_k$$

$$\mathbf{m}_k = \mathbf{m}_{k-1} + \alpha_k \mathbf{p}_k$$

$$\mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k \underline{\mathbf{A}}'\mathbf{q}_k$$

end

$\mathbf{m} = \mathbf{m}_k$.

[Compare this with algorithm 10.2.1 in Golub and Van Loan (1989), which is appropriate for square matrices]. Notice that only matrix-vector and vector-vector products are performed, and that at each iteration we simply minimize the residual $\mathbf{r}$ by moving a distance α in search direction $\mathbf{p}$. We are guaranteed to reduce this residual at each step unless we have reached the least-squares solution. In practice, as a result of both computer round-off errors and the limited amount of time available, we do not perform n iterations. Then the CG algorithm can be thought of as a truly iterative method (Golub and Van Loan, 1989).

1-D EXAMPLE

To illustrate the type of convergence problems encountered when applying derivative regularization, we will first consider the simple 1-D example diagrammed in Figures 1 and 2. Our problem consists of fitting a model line $\mathbf{m}$ sampled

at a regular series of points x_i, $i = 1$, n, constrained by data at various points along the x-axis $\mathbf{m}(x_j) = d_j$, $j = 1$, m. Our goal is to find the line going through the cluster of data that contains the least amount of structure. If we choose, for instance, that the equations represented by $\underline{\mathbf{D}}$ constrain the first differences of the 1-D model $\mathbf{m}$ (i.e., we have chosen gradient damping as the form of regularization), we then have

$$\underline{\mathbf{D}} = \frac{1}{\Delta}\begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & \cdot & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & \cdot & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & \cdot & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & \cdots & \cdot & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \cdot & 0 & -1 & 1 \end{pmatrix} \tag{10}$$

$\kappa = 0.03$

- - - - - without preconditioning
- - - - 1/2 with, 1/2 without preconditioning
.... with preconditioning
———— true least-squares solution

(a) 2 iterations
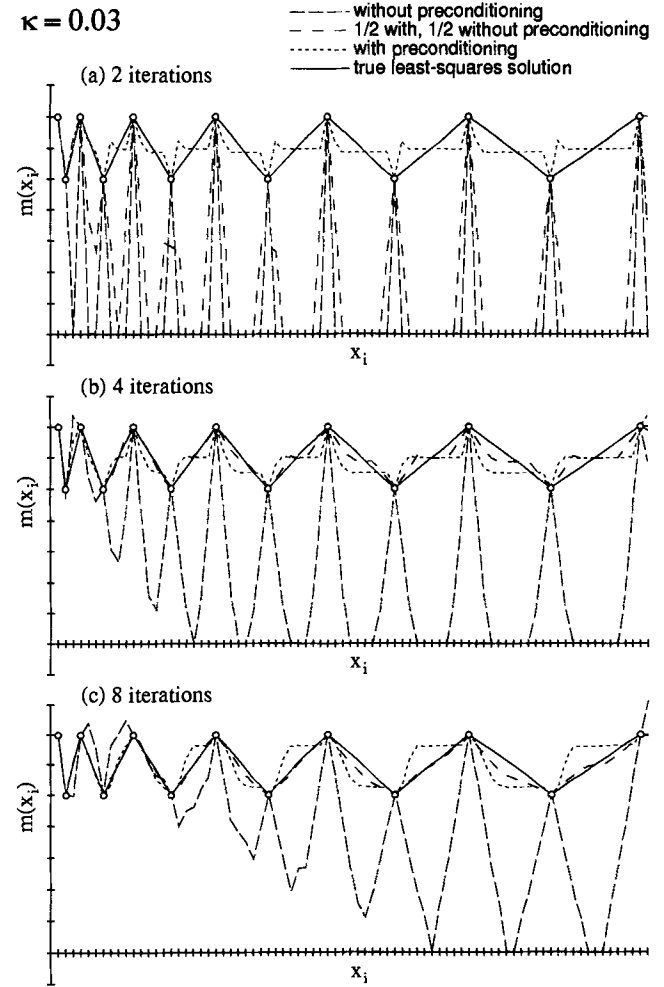
(b) 4 iterations

(c) 8 iterations

FIG. 1. One-dimensional example of the convergence of CG inversion with a low level of first-difference regularization ($\kappa = 0.03$) at (a) iteration 2, (b) iteration 4, and (c) iteration 8. The solid line represents the true least-squares solution. The long dashed lines are the CG solution without preconditioning, while the shorter dashed lines are the solutions with half or with all iterations preconditioned, as indicated.

for model elements with spacing Δ. We have chosen our data sampling so as to provide a varying level of underdetermined behavior, with the far left-hand portion of the curve sampled at every element, and a large spacing between data sampling on the right-hand side. In total, there are 13 data points (shown as open circles in each Figure) and 80 model samples (shown on the x-axis). With a high degree of regularization (large κ), this results in a flat line at the mean of the points, and with low regularization (small κ) we obtain a line going nearly through each point, connected by straight lines. The proper level of regularization depends on the accuracy of the data, determining how much structure is warranted. If we had chosen a second-difference regularization instead, a large κ would produce a linear regression (since a straight line has zero second derivative).

The curves with long-dashed lines in Figures 1a–1c (labeled "without preconditioning") show the result of 2, 4, and 8 CG iterations for a low level of regularization (corresponding to low data variance). The solid line in each figure represents the true least-squares solution. The early iterations reflect the fact that the first difference information requires one iteration to propagate over each model element. This is because the regularization equations between data

points are completely satisfied until a neighboring element is disturbed. This can be observed clearly in the CG algorithm given above since only multiplications with $\underline{A}$ and $\underline{A}^t$ occur at each iteration. It is an inherent feature of gradient-type algorithms that information can take many iterations to propagate over multiple model elements when local constraints are imposed. Since the data equations are given the largest weight in this problem, they are fit first at the expense of large gradients being introduced into the model.

Figures 2a–2c represent the same set of iterations when a higher level of regularization has been chosen (appropriate for large data variance). Now the regularization severely inhibits the data fitting, resulting in a low amount of structure, as defined by the first difference constraints, but severe biases exist throughout the entire curve. In neither of these circumstances does the CG algorithm produce satisfactory results in 8 ($n/10$) iterations; in fact, it requires closer to $n/2$ iterations to obtain reasonable solutions. To understand these effects better, it is instructive to examine the eigenvalue-eigenvector decomposition of the problems (Wiggins, 1972).

The eigenvalue distributions of these two examples, plotted in Figure 3, vary significantly from one another. In the two cases with regularization weights (κ) of 0.03 and 3.00, however, significant features obtained with later iterations are associated with small eigenvalues of the associated SVD solution. This is illustrated in Figure 4. While in general it is thought that the inclusion of information related to small eigenvalues is undesirable since these represent aspects of the model poorly constrained by the equations, our mixture of data and regularization constraints must change this argument. This is explained by the regularization constraints being associated with zero variance. Unlike the situation when normally applying an singular value decomposition (SVD) (Wiggins, 1972) analysis (when all constraints have some data variance associated with them), in this type of problem small eigenvalues associated with the regularization equations will not be swamped by noise because these equations are not subject to a measurement error. While it is still true that the changes in the model caused by the inclusion of these eigenvalues have small effect on the
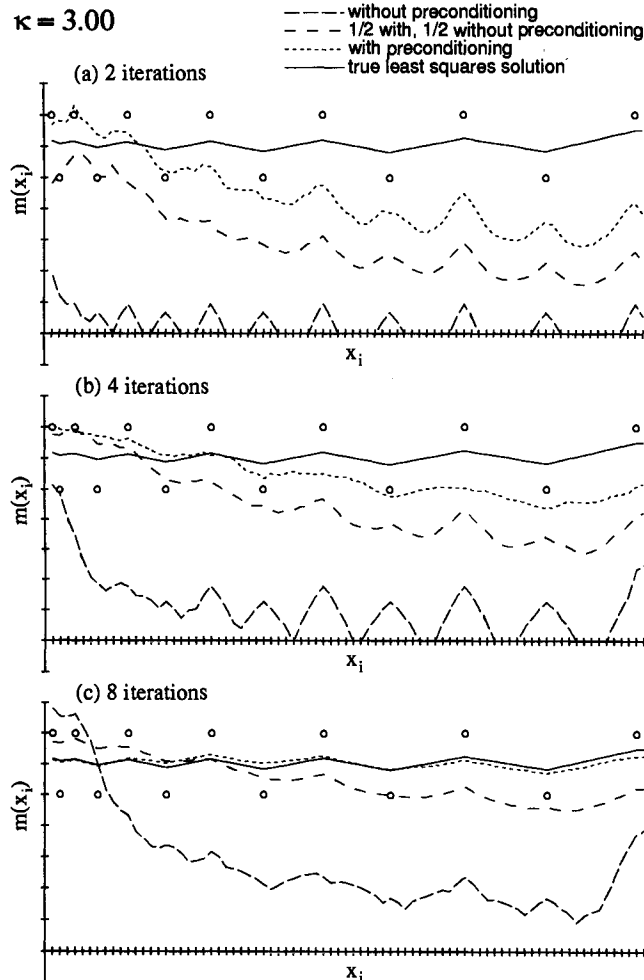

FIG. 2.   Same as Figure 1 except now for a high level of first-difference regularization ($\kappa = 3.00$).
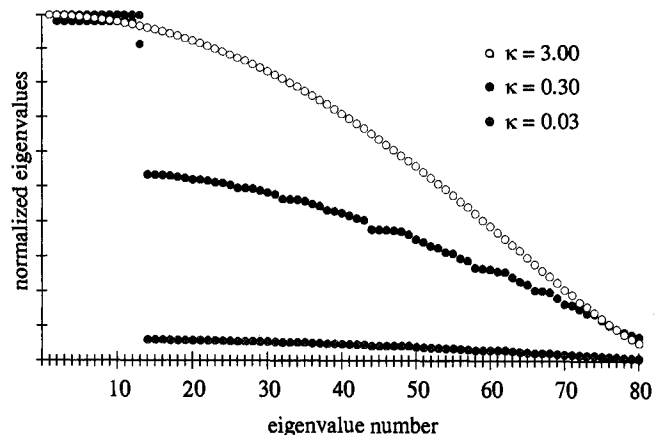

FIG. 3.   Normalized eigenvalue spectra for 1-D examples shown in Figures 1 and 2, along with spectrum from the same problem but with an intermediate regularization ($\kappa = 0.30$).

overall residual, their exclusion produces models with *greater* structure rather than the desired flat or smooth solution (Wiggins, 1972). Such systems are therefore not amenable to SVD analysis with a simple eigenvalue truncation.

Another way of looking at this problem is to realize that through the addition of the regularization constraints $\underline{D}'\underline{D}$ we have created a new system $\underline{A}'\underline{A}$ that is of full rank. Therefore, at least to numerical precision, $(\underline{A}'\underline{A})^{-1}$ exists and should be our goal (to obtain a model with the absolute least amount of structure necessary to explain the data) rather than some *generalized* inverse of $\underline{A}'\underline{A}$, such as an eigenvalue truncation would produce. The inverse $(\underline{A}'\underline{A})^{-1}$ can already be considered a generalized inverse of the matrix $\underline{F}'\underline{C}_d^{-1}\underline{F}$.

THE PRECONDITIONING OPERATOR

The search directions $\mathbf{p}$ defined in equations (9a) and (9b) of the CG algorithm are constructed without the use of any a priori knowledge of regular structure within matrix $\underline{A}$ (recall that $\underline{A}$ represents a set of data equations $\underline{C}_d^{-1/2}\underline{F}$ supplemented by regularization equations $\kappa\underline{D}$). Often, however, a portion or all of $\underline{A}$ will contain regular patterns that could potentially guide us in our selection of optimum search directions. Consider inserting this additional information through a preconditioning of the search direction by an

(a) $\kappa = 0.03$

—— 13 largest eigenvalues
— — 60 largest eigenvalues
········· 70 largest eigenvalues
——— true least-squares solution (all 80)

(b) $\kappa = 3.00$

—— 60 largest eigenvalues
— — 70 largest eigenvalues
········· 75 largest eigenvalues
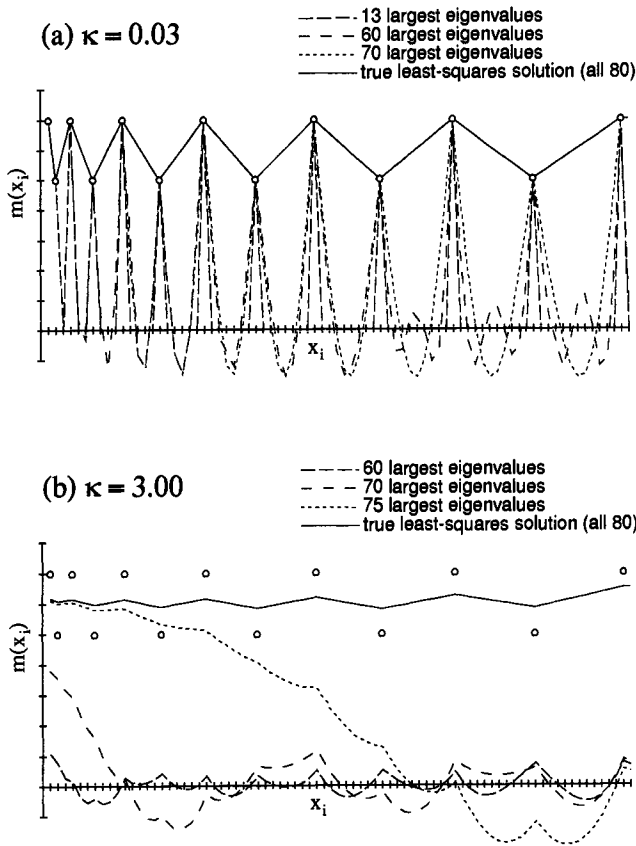——— true least-squares solution (all 80)

FIG. 4. Partial solutions to the examples shown in Figures 1 and 2 found by performing an SVD with eigenvalue truncations at the number of eigenvalues indicated.

operator $\underline{S}$ to obtain an improved direction $\mathbf{z}$ (i.e., $\mathbf{z} = \underline{S}\mathbf{p}$). The preconditioned CG algorithm is then:

$k = 0$; $\mathbf{m}_0$ = initial solution estimate; $\mathbf{r}_0 = \underline{A}'(\mathbf{b} - \underline{A}\mathbf{m}_0)$,

while $\mathbf{r}_k \neq \mathbf{0}$

$\quad \mathbf{z}_k = \underline{S}\mathbf{r}$
$\quad k = k + 1$
$\quad$ if $k = 1$
$\quad\quad \mathbf{p}_1 = \mathbf{z}_0$,
$\quad$ else
$\quad\quad \beta_k = \mathbf{r}_{k-1}^t\mathbf{z}_{k-1}/\mathbf{r}_{k-2}^t\mathbf{z}_{k-2}$
$\quad\quad \mathbf{p}_k = \mathbf{z}_{k-1} + \beta_k\mathbf{p}_{k-1}$
$\quad$ end

$\quad \mathbf{q}_k = \underline{A}\mathbf{p}_k$

$\quad \alpha_k = \mathbf{r}_{k-1}^t\mathbf{z}_{k-1}/\mathbf{q}_k^t\mathbf{q}_k$
$\quad \mathbf{m}_k = \mathbf{m}_{k-1} + \alpha_k\mathbf{p}_k$
$\quad \mathbf{r}_k = \mathbf{r}_{k-1} - \alpha_k\underline{A}'\mathbf{q}_k$

end

$\mathbf{m} = \mathbf{m}_k$.

[Compare this with algorithm 10.3.1 of Golub and Van Loan (1989) which is appropriate for square matrices]. The optimum preconditioning,

$$z = (\underline{A}'\underline{A})^{-1}\mathbf{p} = (\underline{F}'\underline{C}_d^{-1}\underline{F} + \kappa^2\underline{D}'\underline{D})^{-1}\mathbf{p} \quad (11)$$

would, of course, produce the least-squares solution to equation (3) in a single iteration. Since we are unable to form this inverse directly, we will use our knowledge of the regular structure of $(\underline{F}'\underline{C}_d^{-1}\underline{F} + \kappa^2\underline{D}'\underline{D})$ to form a simple and computationally inexpensive approximation to the inverse $(\underline{F}'\underline{C}_d^{-1}\underline{F} + \kappa^2\underline{D}'\underline{D})^{-1}$. We now derive a preconditioning operator $\underline{S}$ for derivative damping in one dimension appropriate to the example shown in the previous section.

The matrix $\underline{D}$ is defined in equation (10) for equidistant model elements with spacing Δ. For $\underline{F}$ we will make the approximation that $(\underline{F}'\underline{C}_d^{-1}\underline{F}) = diag\,(\underline{F}'\underline{C}_d^{-1}\underline{F})$, and define (with no implicit summation),

$$h_i^2 \equiv (\underline{F}'\underline{C}_d^{-1}\underline{F})_{ii}. \quad (12)$$

Generalizing to continuous functions, we find

$$\underline{D}'\underline{D} \rightarrow -\nabla^2 \quad \text{and} \quad h_i^2 \rightarrow h^2(\mathbf{x}), \quad (13)$$

and therefore equation (11) can be approximated by

$$z(\mathbf{x}) = (h^2(\mathbf{x}) - \kappa^2\nabla^2)^{-1}p(\mathbf{x}), \quad (14)$$

or

$$(h^2(\mathbf{x}) - \kappa^2\nabla^2)z(\mathbf{x}) = p(\mathbf{x}). \quad (15)$$

The WKBJ Green's function to equation (15) is (Bender and Orszag, 1978),

$$G(\mathbf{x}, \mathbf{x}') \approx \frac{1}{2} \frac{1}{\sqrt{\dfrac{h(\mathbf{x})}{\kappa}\dfrac{h(\mathbf{x}')}{\kappa}}} \exp - \left| \int_{\mathbf{x}'}^{\mathbf{x}} \frac{h(\zeta)}{\kappa} d\zeta \right| , \quad (16)$$

and from the definition of the Green's function $z(\mathbf{x}) = \int G(\mathbf{x}, \mathbf{x}')p(\mathbf{x}')\, d\mathbf{x}'$ we have,

$$z(\mathbf{x}) \approx \frac{\kappa}{2} \int \left[\frac{1}{\sqrt{h(\mathbf{x})h(\mathbf{x}')}} \right.$$

$$\left. \times \left(\exp - \left| \int_{\mathbf{x}'}^{\mathbf{x}} \frac{h(\zeta)}{\kappa} d\zeta \right| \right) p(\mathbf{x}') \right] d\mathbf{x}'. \quad (17)$$

Now, reverting to a discretized system, we let $\int d\mathbf{x}' \to \Sigma_j \Delta$ and $\mathbf{x} \to i$

and obtain,

$$z_i \approx \sum_j \frac{1}{\sqrt{h_i h_j}} \left(\exp - \frac{\Delta}{\kappa} \left| \sum_{k=j}^{i} h_k \right| \right) p_j, \quad (18)$$

where we have dropped the arbitrary multiplicative constant $(\kappa/2)$.

We can see immediately that this operator contains the general features that we desire. In regions of high data density (h_i large), the function is sharply peaked, becoming closer to a delta function as h_i increases. In that case, z is parallel to p, except for a scale factor to normalize each column of $\mathbf{A}$. Conversely in areas of low data density (h_i small) the exponential function falls off slowly, averaging out rough structure in the search direction, and thereby imposing our a priori knowledge of what type of solution is "preferred" by the regularization constraint $\mathbf{D}$. In the case of the existence of a model element with zero data coverage, our formulation breaks down, since the WKBJ solution used breaks down for h_i close to zero. Therefore, we must impose a special condition. We have chosen to simply set a base level for h_i (e.g., $\mathrm{MIN}(h_i) = 0.1$) to satisfy this condition. Instead of deriving the approximate analytic solution in equation (18), another way of computing the preconditioned search direction z would be to derive an approximate inverse of equation (11) through an incomplete LU decomposition (Meijerink and van der Vorst, 1977) or other numerical approximation (H. A. van der Vorst, personal communication, 1993). The application of equation (18) is less expensive than might appear at first sight. This is because if we calculate the operator from position i outward independently in the positive and negative directions, then the sum in the exponent may be computed recursively at each step.

The result of the application of the preconditioning operator derived above to the 1-D example previously considered is shown in Figures 1 and 2. In each figure, the preconditioned solutions are represented by curves with shorter dashed lines, showing both all iterations with preconditioning applied and also the case where only the first half of the iterations were performed with preconditioning. It is clear that for both the case of low regularization (Figure 1) and high regularization (Figure 2), the solutions that took advantage of our a priori knowledge of the normal equations

through preconditioning the search directions converge more rapidly to the least-squares solution. In the case of low regularization, switching halfway to iterations without preconditioning performs better than the case with all iterations preconditioned, since at some point we wish to allow a degree of roughness into our model (because of our low a priori data variance estimates predicting that this level of roughness is required by the measurements).

CROSSHOLE TOMOGRAPHY EXAMPLE

We now turn to a more realistic application of the preconditioning operator, a 2-D crosshole geometry. Figure 5 represents the velocity field that we use to generate the synthetic data. The model consists of a series of square blocks (100×60) with sources and receivers as shown and raypaths connecting a random 50 percent of the possible source-receiver pairs. To isolate the effect of the inversion procedure, we will consider only a single linear iteration of the traveltime inversion problem, using the true raypaths through the structure (Figure 6) to simulate the type of ray coverage obtained in realistic situations. We used the shortest path method of Moser (1991) to obtain the raypaths shown in Figure 6. The parameterization, ray geometry and velocity perturbations used in this example are realistic for a crosshole experiment (Bregman et. al, 1989). Figure 7 shows the weighting factor (h_i) used in the preconditioning operation, indicating the regions of low path coverage where the operator will have the strongest effect. To implement the operator in two dimensions at each iteration we simply apply the 1-D operator of equation (18) independently in each direction.

Figure 8 shows the true least-squares solution to our inverse problem where once again we have chosen first-difference regularization. The "true" least-squares solution is found by performing 6000 CG iterations. Overall, this solution reproduces many of the important features from the synthetic model without producing much in the way of "phantom" structures not necessitated by the data. We will now examine how partial reconstructions of the model compare to the true solution, evaluating the result obtained by starting with three different initial models (note that the true solution to our problem is independent of starting model). Figures 9a–9c show what would be the projected borehole velocity logs at the positions indicated on Figure 8, as a function of the number of CG iterations (the solid line in each figure represents the true least-squares solution). The three different solutions shown are computed with homogeneous starting models of 4.6, 5.0, and 5.4 km/s. Notice that at 10 and 50 iterations, the curves contain a large amount of structure that is clearly not necessary to explain the data (in fact the final smooth curve explains the data significantly better). Also, it is clear that depending on which starting velocity was used, our interpretations of these curves would vary dramatically, and simple a posteriori smoothing of the curves would not bring them into agreement. At 100 iterations, the smoothness constraints are certainly coming more into play, yet there remain significant biases between the curves, and any quantitative analysis of these values would not be reliable. It is not until 500 CG iterations of this system (Figure 9d) that these biases are effectively removed, and even then a low-frequency component yet to be resolved remains
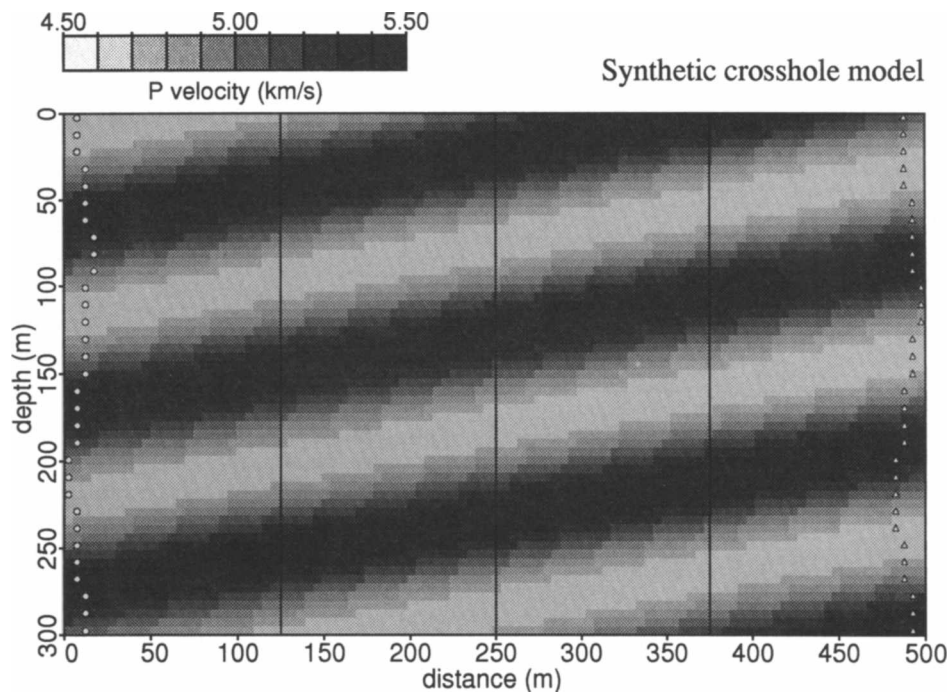
FIG. 5.   Synthetic crosshole model used for 2-D example. Open circles and triangles represent downhole source and geophone locations. There is no vertical exaggeration.
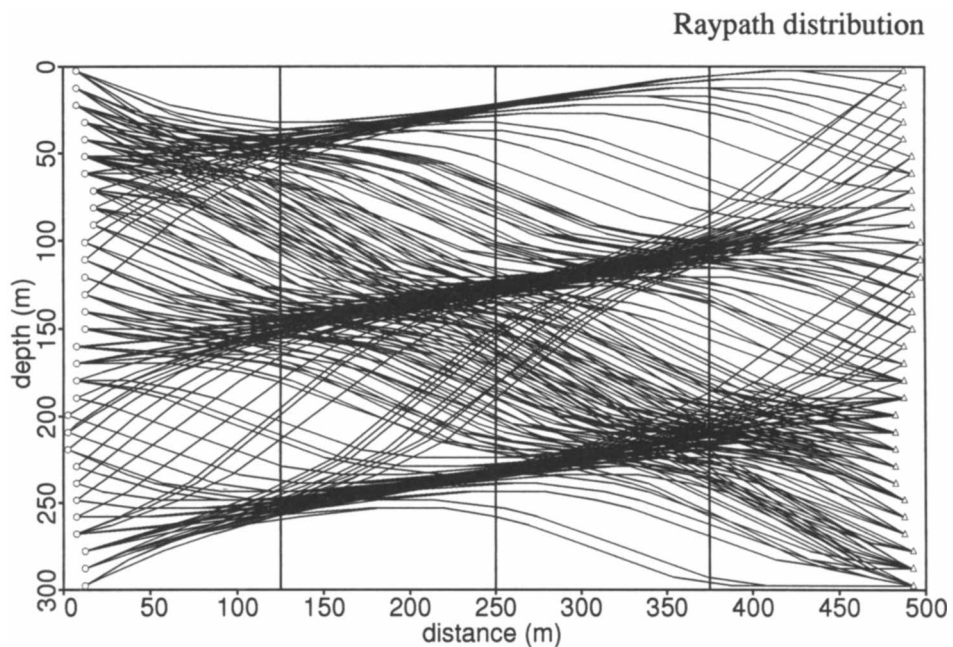
FIG. 6.   Raypaths through synthetic model shown in Figure 5. A random 50 percent of the possible source-receiver combinations were used to simulate the type of coverage found in real data studies (Bregman et al., 1989). The rays were calculated using the shortest path method of Moser (1991).
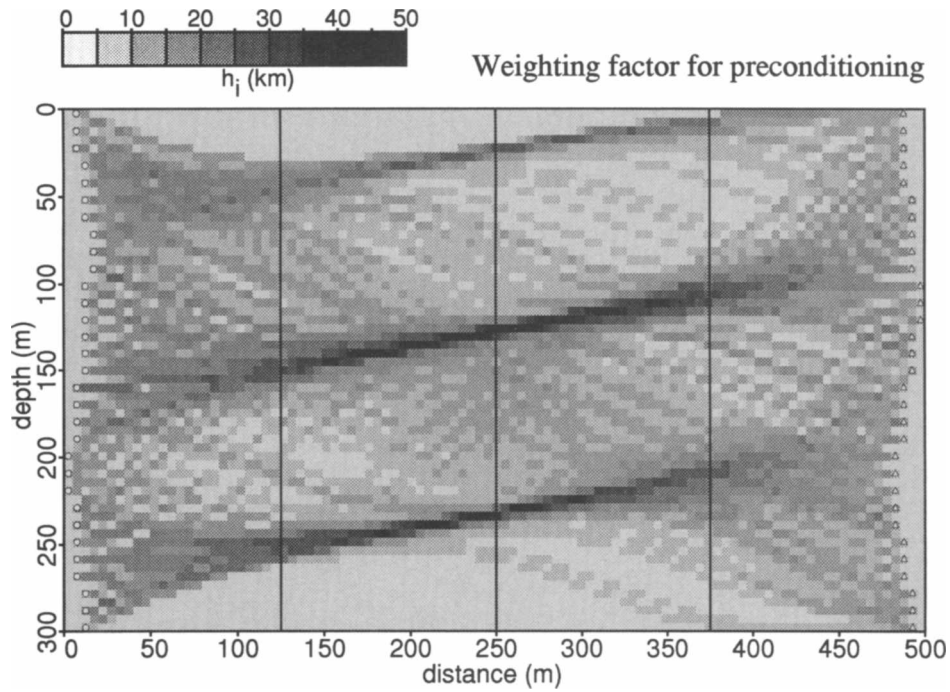
FIG. 7.   The weighting function h_i used in equation (18) for preconditioning of CG search directions.
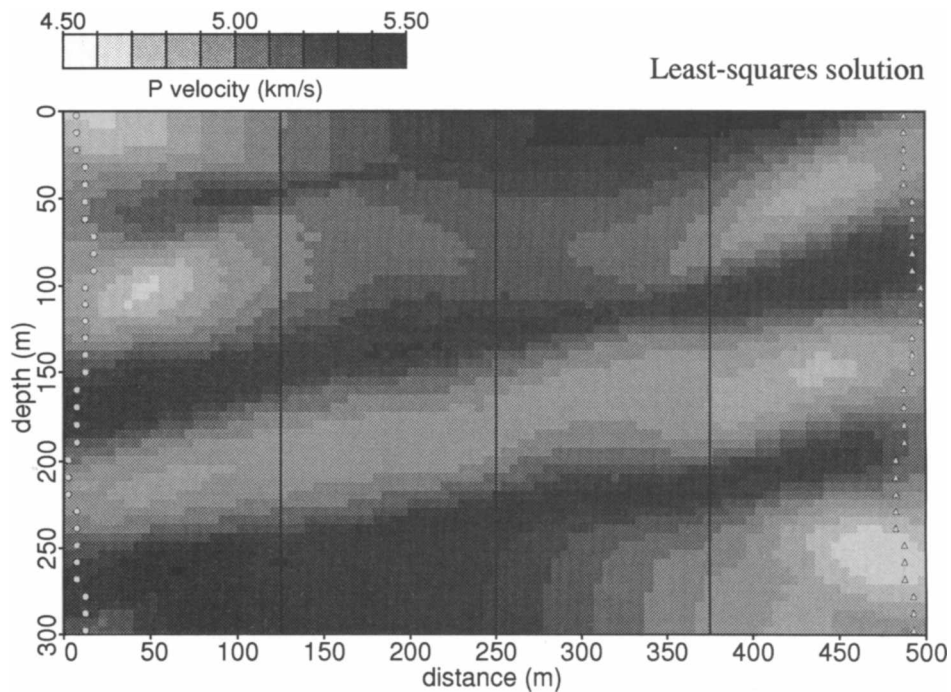
FIG. 8.   The least-squares solution to the crosshole problem (solution after 6000 CG iterations). "Potential borehole" locations shown as black lines at 1/4, 1/2, and 3/4 distance across model.
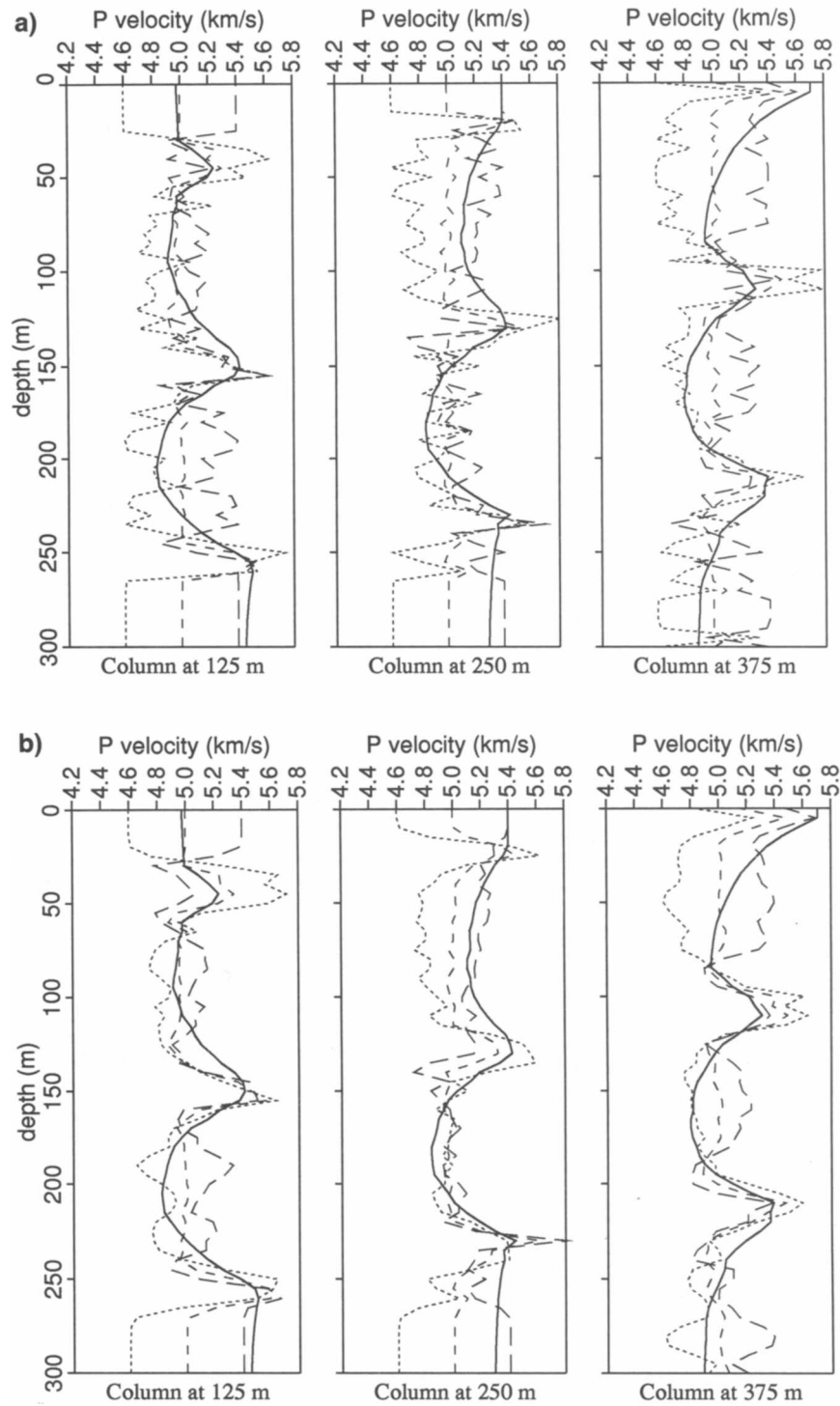
FIG. 9. Velocity as a function of depth at potential borehole locations shown in Figure 8 for CG inversions without preconditioning. The solid line is the least-squares solution shown in Figure 8. The dashed lines represent partial CG solutions at (a) iteration 10, (b) iteration 50, (c) iteration 100, and (d) iteration 500, using three different homogeneous starting models. The short-, medium-, and long-dashed curves represent starting models of 4.6, 5.0, and 5.4 km/s, respectively. All starting models were within the range of perturbations of the synthetic model (from 4.6–5.4 km/s).

(although this component is probably much less critical to our analysis).

Figure 10 illustrates how these biases can be substantially reduced in many fewer iterations with the use of a preconditioning operator. In this example, we have applied preconditioning to the first 25 CG iterations and not to the second 25. The models we obtain are much less dependent on initial

values, with the exception of the outskirts of the model where there is no data coverage (areas we presumedly would not interpret in any case). The cost of the preconditioning depends heavily on the regularization level imposed (as this controls the relative widths of the convolution operators, and therefore directly the number of additional multiplications involved in the operation). However, in this example a
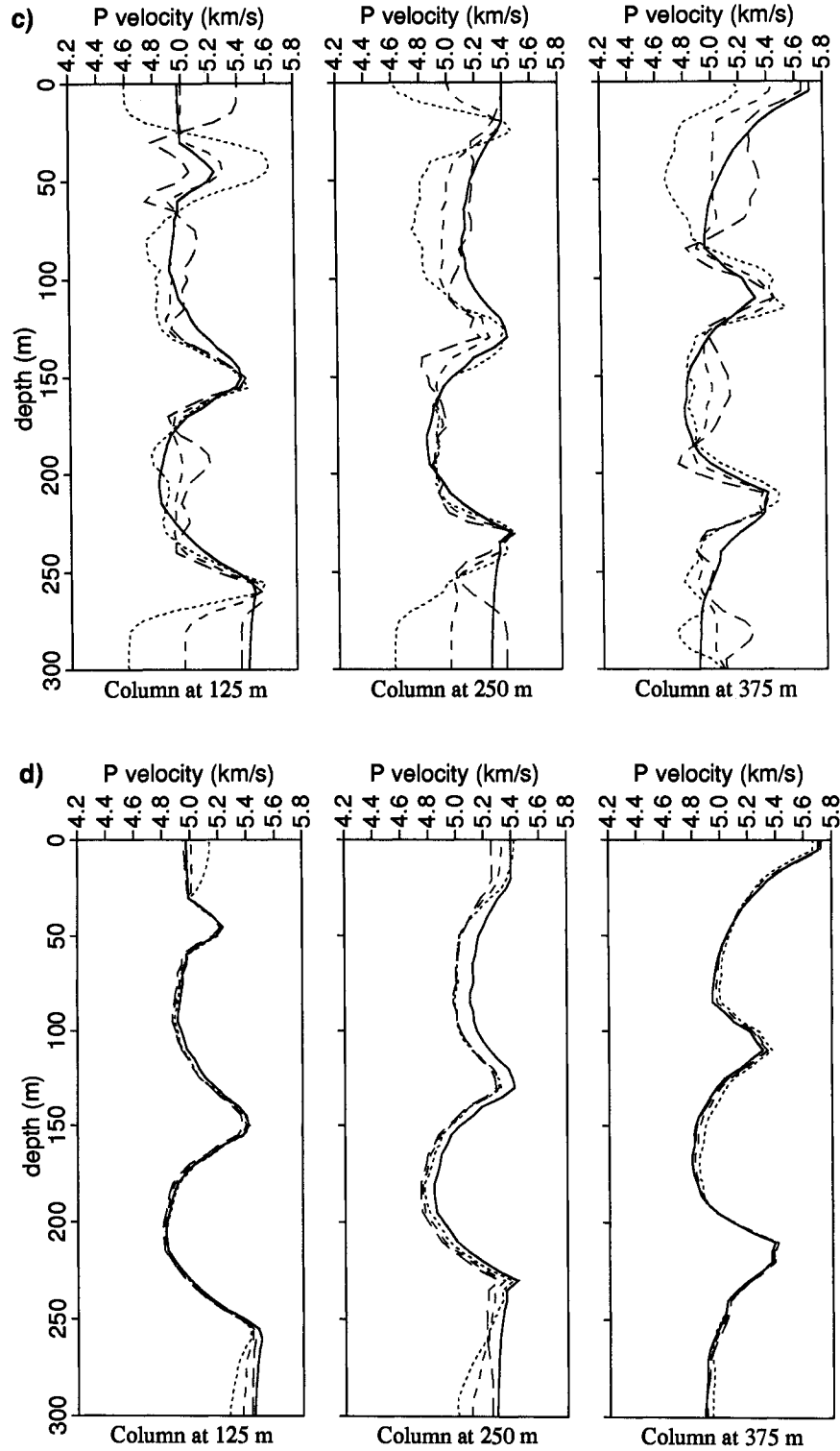

FIG. 9. continued

preconditioned iteration took approximately three times as long as a normal CG iteration.

The extra cost of preconditioning, however, also depends on the number of data equations versus number of model elements. It is clear from equation (11) that the preconditioning operation depends on only the number of model elements (since $\underline{S}$ acts only upon the gradient operator g which is of length n) while the inversion itself depends on the number of data equations m. The preconditioning, therefore, becomes relatively more efficient as the number of data equations rises. It is of little use to give quantitative numbers as to the efficiency of the preconditioning operation, because of the high dependence on application. In all the tests we have run to date, however, the accelerated convergence obtained though the use of this operator has produced comparable results by a factor of at least three and often 10 or more times the rate obtained without preconditioning. Depending upon memory versus time constraints, the preconditioning matrix $\underline{S}$ may be either precalculated and stored, or its elements calculated recursively at each iteration (as we have chosen to do).

In large-scale tomographic applications the number of data equations can reach a factor of 10^2 to 10^3 greater than the number of model parameters (Spakman, 1991), leading to the situation where only a few tens of iterations are computationally feasible. In such situations, using a preconditioned CG iteration would require virtually no additional cost over a standard CG algorithm. In such large inverse problems (where $O(10^5)$ model elements are used) CG methods such as the Hestene and Stiefel's (1952) algorithm (VanDecar, 1991) or the LSQR algorithm of Paige and Saunders (1982)

(Zielhuis, 1992) have been found to require $O(10^2)$ to $O(10^3)$ iterations to converge under derivative regularization. The benefits of applying a preconditioning algorithm to such problems are expected to be high.

CONCLUSIONS

We have shown that the use of CG algorithms to solve least-squares inverse problems under derivative regularization can result in the need to perform many more iterations than in the case of simple damping. For some of the inverse problems being performed today, a large number of iterations are neither computationally feasible nor desirable. Performing too few iterations, however, may result in models that contain unwarranted structure. To attack this problem, we have developed a method of preconditioning the CG search directions that takes advantage of a priori knowlege of the structure of the regularization equations. This approximate operator depends only on the number of model elements and not on the number of data (often order of magnitudes greater) making it feasible to implement in even the largest of studies. The operation can be implemented recursively thereby increasing its efficiency dramatically.

In both 1-D and 2-D examples we have demonstrated the problems incurred in using CG inversion with derivative regularization and shown how this problem can be mitigated through the use of a preconditioning operation. Inversion using the "multi-grid" approach (inverting first with a coarse parameterization and then finer) or preconditioning with approximate numerical inverses (Meijerink and van der Vorst, 1977) may also be effective in addressing this prob-
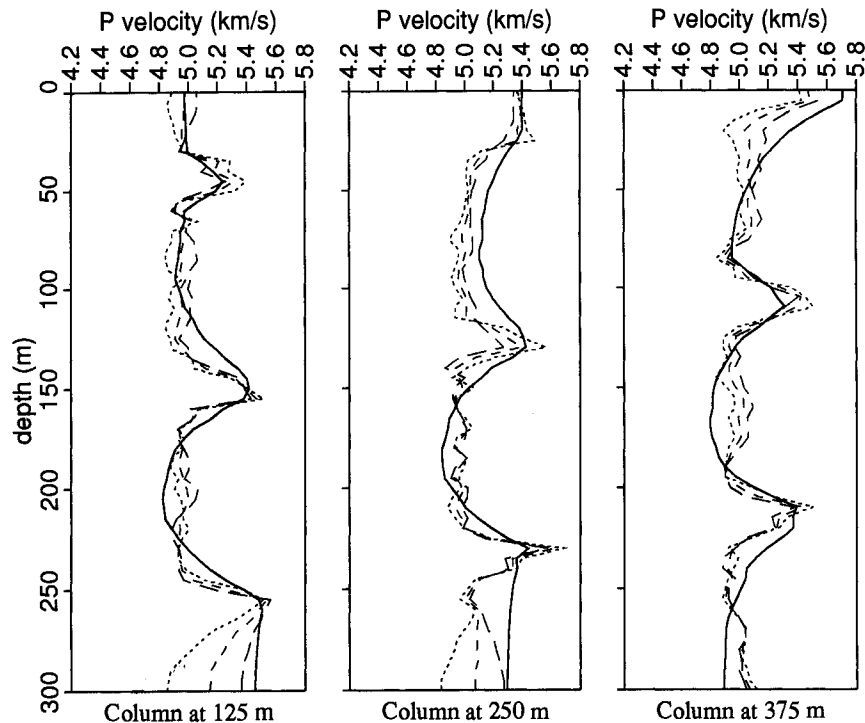

Fig. 10. Same as Figure 9, except now with 25 preconditioned CG iterations followed by 25 iterations without preconditioning. Compare with Figure 9b, which was the result of 50 iterations without preconditioning.

lem. From our examples, it was clear that our interpretation of models resulting from insufficient CG iterations was highly dependent on the choice of starting model (although the solution was not), and simple a posteriori smoothing of the curves would not bring them into agreeement.

As discussed above, the regularization equations can be related to an a priori model covariance matrix as described by Tarantola (1987). Tarantola (1987) derives the form of this constraint for the case of an exponential covariance operator in 3-D to be approximately

$$\underline{C}_m^{-1} \approx a\underline{I} + b\underline{\delta} + c\underline{\delta}^2, \tag{19}$$

where $\underline{\delta}$, and $\underline{\delta}^2$ represent first and second derivative operators, respectively, and a, b, and c are constants. It is clear from what we have learned above that for an iterative-type inversion, the damping portion of this constraint will dominate the initial iterations and therefore the overall constraint imposed by equation (19) will be unevenly applied. Therefore, to apply a priori model covariance constraints of this kind, many CG iterations may become necessary.

ACKNOWLEDGMENTS

REFERENCES

Backus, G. E., and Gilbert, J. F., 1967, Numerical applications of a formalism for geophysical inverse problems: Geophys. J. Roy. Astr. Soc., **13**, 247–276.

—— 1968, The resolving power of gross earth data: Geophys. J. Roy. Astr. Soc., **16**, 169–205.

—— 1970, Uniqueness in the inversion of inaccurate gross earth data: Proc. Roy. Soc. London, Ser. A, **266**, 123–192.

Bender, C. M., and Orszag, S. A., 1978, Advanced mathematical methods for scientists and engineers: McGraw-Hill Inc. Book Co.

Bregman, N. D., Bailey, R. C., and Chapman, C. H., 1989, Crosshole seismic tomography: Geophysics, **54**, 200–215.

Constable, S. C., Parker, R. L., and Constable, C. G., 1987, Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data: Geophysics, **52**, 289–300.

Golub, G. H., and Van Loan, C. F., 1989, Matrix computations: Johns Hopkins Univ. Press.

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

Meijerink, J. A., and van der Vorst, H. A., 1977, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix: Math. Comp., **31**, 148–162.

Moser, T. J., 1991, Shortest path calculation of seismic rays: Geophysics, **56**, 59–67.

Neele, F., VanDecar, J. C., and Snieder, R., 1993, The use of P-wave amplitude data in a joint tomographic inversion with traveltimes for upper-mantle velocity structure: J. Geophys. Res., **98**, 12,033–12,054.

Nolet, G., 1985, Solving or resolving inadequate and noisy tomographic systems: J. Comp. Physics, **61**, 463–482.

—— 1987, Seismic wave propagation and seismic tomography, in Nolet, G., Eds., Seismic tomography: D. Reidel Publishing Company, 1–23.

Nolet, G., and Snieder, R., 1990, Solving large linear inverse problems by projection: Geophys. J. Int., **103**, 565–568.

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

Parker, R. L., 1977, Understanding inverse theory: Ann Rev. Earth Planet. Sci., **5**, 35–64.

Sambridge, M. S., 1990, Non-linear arrival time inversion: Constraining velocity anomalies by seeking smooth models in 3-D: Geophys. J. Int., **102**, 653–677.

Scales, J. A., 1987, Tomographic inversion via the conjugate gradient method: Geophysics, **52**, 179–185.

Spakman, W., 1991. Delay-time tomography of the upper mantle below Europe, the Mediterranean, and Asia Minor: Geophys. J. Int., **107**, 309–332.

Spakman, W., and Nolet, G., 1988, Imaging algorithms, accuracy, and resolution in delay time tomography, in Vlaar, N. J., Nolet, G., Wortel, M. J. R., and Cloetingh, S. A. P. L., Eds., Mathematical geophysics: D. Reidel Publ. Co., 155–187.

Stork, C., 1992, Singular-value decomposition of the velocity-reflector depth tradeoff, part 2: High-resolution analysis of a generic model: Geophysics, **57**, 933–943.

Tarantola, A., 1987, Inverse problem theory: Methods for data fitting and model parameter estimation: Elsevier Science Publ. Co., Inc.

Tarantola, A., and Valette, B., 1982, Generalized nonlinear inverse problems solved using the least squares criterion: Rev. Geophys. Space Phys., **20**, 219–232.

VanDecar, J. C., 1991, Upper-mantle structure of the Cascadia subduction zone from non-linear teleseismic travel-time inversion: Ph.D. thesis, Univ. of Washington.

van der Hilst, R. D., Engdahl, E. R., Spakman, W., and Nolet, G., 1991, Tomographic imaging of subducted lithosphere below northwest Pacific island arcs: Nature, **353**, 37–43.

van der Sluis, A., and van der Vorst, H. A., 1987, Numerical solution of large sparse linear algebraic systems arising from tomographic problems: in Nolet, G., Eds., Seismic tomography: D. Reidel Publ. Co., 49–83.

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

Zielhuis, A., 1992, S-wave velocity below Europe from delay-time and waveform inversions: Ph.D. thesis, Univ. of Utrecht.