Interval velocity estimation from beam-stacked data — an improved method

Biondo Biondi

ABSTRACT

Low-wavenumber components of the velocity function can be tomographically estimated from beam-stacked data. The velocity estimation method does not require data picking; rather it maximizes beam stack energy at traveltimes and surface locations predicted by the velocity model.

The gradient of the objective function with respect to the velocity model is explicitly computed using ray tracing. The objective function is maximized by applying a conjugate gradient algorithm and a quasi-Newton algorithm sequentially. The former converges to a robust estimate of the low-wavenumber components of the velocity model; while the latter resolves the finer components of the velocity model determined from the beam stacks.

The inversion method is successful in estimating the velocity function from synthetic data in a horizontally-layered medium.

INTRODUCTION

In a previous report (Biondi, 1987) I presented a method for estimating the velocity model in the presence of lateral variations in velocity as well as geological structure. The method I described is based upon a tomographical inversion of beam-stacked data. The inversion did not utilize picked data, like most tomographic methods (Bishop et al., 1985; Sword, 1986; Sword, 1987); it ,by contrast, maximized the beam-stack energy at traveltimes and surface locations predicted by the velocity model. In this respect my inversion method resembled the approaches of Toldi (1985) and Fowler (1985), who maximize stacking power as a function of stacking velocity and of migration velocity. Picking avoidance renders the algorithm more robust with respect to multiples and noise in data. In particular, multiples create local maxima of the objective function but do not influence the global solution. On the other hand, using all the transformed data for the inversion, without picking events, increases CPU-time and memory requirements of the algorithm.

The objective function I used in the previous report is too sensitive to the beam stack artifacts, causing oscillatory behavior of the velocity model. Here I define a new less sensitive objective function with more stable maximization. Therefore I can partially relax the smoothness constraints on the velocity model, and increase the resolution of the resulting velocity estimates.

I use a gradient algorithm to maximize the objective function. For this reason I have developed a scheme for explicit computation of the gradient of the objective function with respect to the velocity model. Explicit computation of the gradient is significantly more efficient than the computation of the gradient using finite difference that I used in the previous report.

THE INVERSION METHOD

The proposed velocity estimation is carried out in beam-stack domain. Beam stacks are a variation of local slant stacks (Harlan and Burridge, 1983; Sword, 1984), and are described in SEP 51 (Kostov and Biondi, 1987). Beam stacks differ from local slant stacks because the stack-trajectory is hyperbolic instead of being a straight line.

Beam stacks are function of five variables: midpoint y, half offset h, midpoint ray parameter p_y, offset ray parameter p_h and travel time t. The amplitude of beam stack can be simply the square of the sum of the amplitudes along the stacking trajectory or the value of a coherency function, such as semblance, computed on the stacking trajectory. The amplitude of $Beam(y, h, t, p_y, p_h)$ is proportional to the energy of the reflected waves propagating with ray parameters p_y and p_h, recorded at midpoint y, offset h and traveltime t.

Figure 1 shows a synthetic common midpoint gather generated by way of an acoustic finite difference modeling program. The earth model is a stratified medium composed of six layers. Only five of the reflections are primaries; all the others are multiples. Figure 2 shows the value of the semblance of a beam-stack decomposition of the gather shown in Figure 1 for a fixed offset ray parameter and null midpoint ray parameter. Energy is localized in a wedge-shaped region around the peaks; this area corresponds to reflections traveling with the specified ray parameter. The traveltime and the offset of the peaks is dependent upon the velocity model.

Modeling beam stacked data

The inversion principle is a tomographic fitting of traveltimes along the beams to beam-stacked data . The basic procedure of the inversion is therefore modeling the propagation of beams in a given velocity model $\mathbf{m}$. I use ray tracing to propagate beams. Simplicity and lower computational cost comprise the rationale for using ray

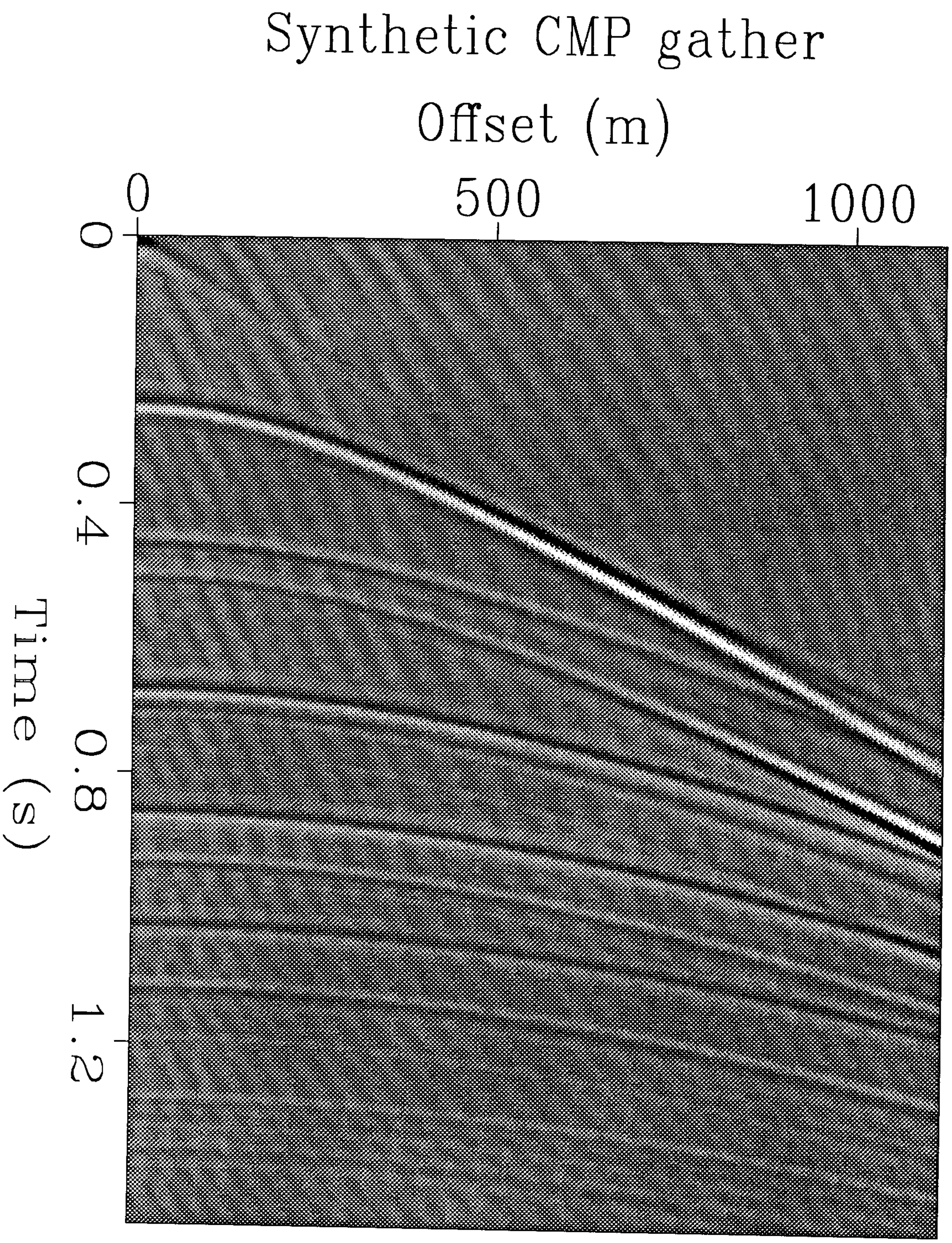

FIG. 1. Synthetic common-midpoint gather generated assuming a stratified medium composed of six layers. Only five reflections are primaries, all the others are multiples.

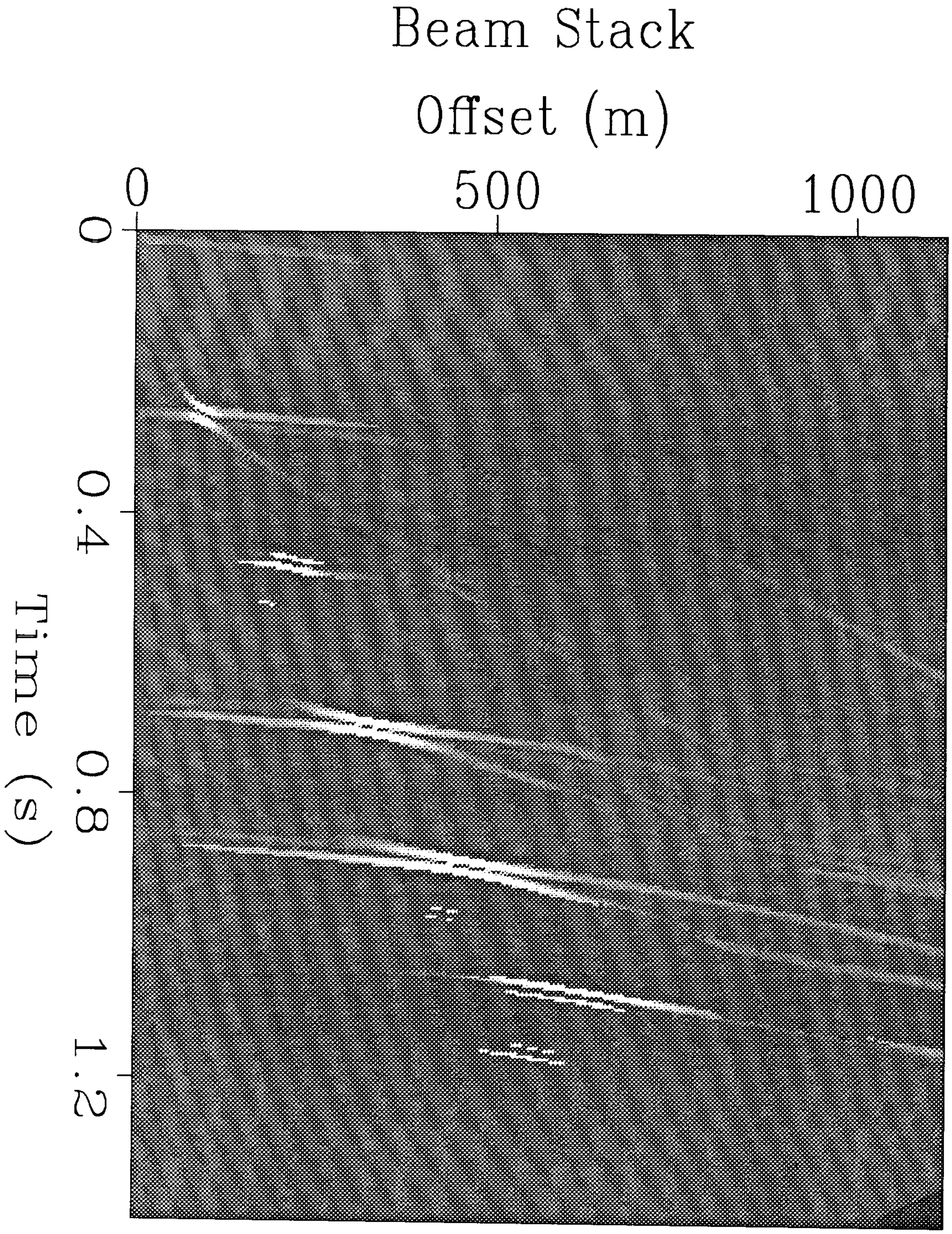

FIG. 2. Beam stacks of the gather shown in Figure 1, for a fixed offset ray parameter and null midpoint ray parameter. Energy is localized in a wedge-shaped region around the peaks; this area corresponds to reflections traveling with the specified ray parameter. The traveltime and the offset of the peaks is dependent upon the velocity model.

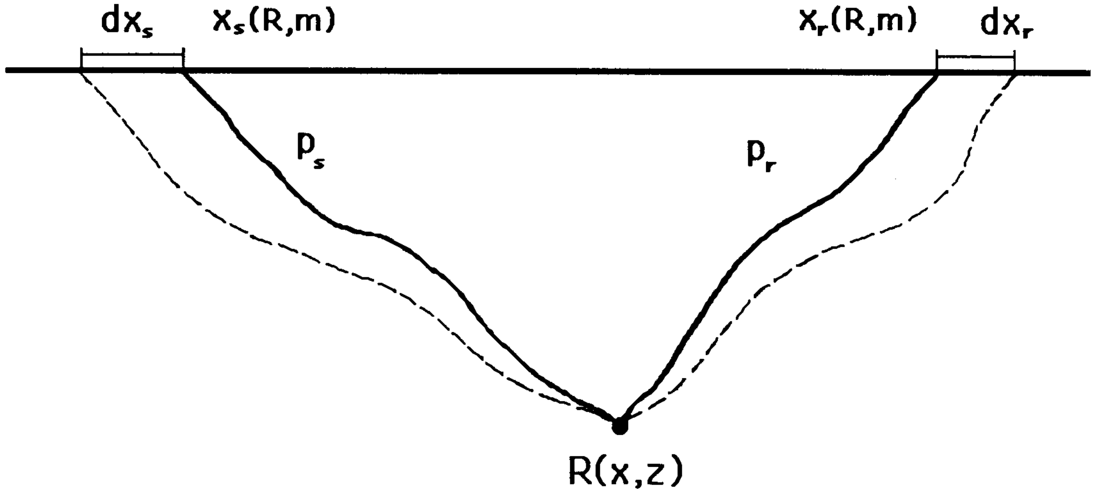

FIG. 3. The paths of the down-going ray and of the up-going ray traced from the reflector position $R(x, z)$. The dashed lines are the modified raypaths when the velocity model has been modified.

tracing. Ray theory is a high-frequency approximation of wave propagation, while the waves propagating in the Earth are band-limited in frequency. One drawback of the high-frequency approximation is that the Fresnel zones of reflections are reduced to points. I take into account the width of Fresnel zones when I compute beam stacks using many traces, but I neglect it when I model beam propagation by ray tracing. The tomographic back-projection operator derived with geometrical rays is singular. The velocity model must therefore be constrained with a smoothness condition that guards against instability during the inversion procedure. These problems could be avoided modeling beam propagation by the wave-theoretical rays proposed by Woodward (1988).

Figure 3 shows ray tracing geometry. Rays are traced from each reflector point $R_j(x_j, z_j)$ to the surface, fixing both the midpoint ray parameter p_y, and the offset ray parameter p_h. The down-going ray has ray parameter $p_s = (p_y - p_h)/2$, and the up-going ray has ray parameter $p_r = (p_y + p_x)/2$. Tracing the up-going ray yields the shot horizontal position $x_s(R_j, \mathrm{m})$ and the traveltime $t_s(R_j, \mathrm{m})$. Tracing the down-going ray yields the receiver horizontal position $x_r(R_j, \mathrm{m})$ and the traveltime $t_r(R_j, \mathrm{m})$. The total travel time t, midpoint y, and half-offset h corresponding to the reflector R_j are, respectively,

$$t_j(R_j, \mathrm{m}) = t_s(R_j, \mathrm{m}) + t_r(R_j, \mathrm{m}); \tag{1a}$$

$$y_j(R_j, \mathrm{m}) = \frac{1}{2}\left(x_s(R_j, \mathrm{m}) + x_r(R_j, \mathrm{m})\right); \tag{1b}$$

and

$$h_j(R_j, \mathrm{m}) = \frac{1}{2}\left(x_r(R_j, \mathrm{m}) - x_s(R_j, \mathrm{m})\right). \tag{1c}$$

These relations implicitly define the function $t = t(y(R, \mathrm{m}), h(R, \mathrm{m}), \mathrm{m})$, that represents a manifold in the data space.

If velocity along the beam path is correct, and a reflector exists at R_j, the beam stack value $Beam(y_j, h_j, t_j(y_j, h_j, \mathrm{m}), p_y, p_h)$ is not zero. If the velocity along the beam path is wrong, by contrast, the value of the semblance of beam stack at predicted traveltime and surface location is about zero. The goal of inversion is to determine that velocity model that maximizes energy of beam-stacked data along the manifold $t(y, h, \mathrm{m})$. The inversion can therefore be formulated as maximization of the total energy

$$E(\mathrm{m}) = \sum_{p_y}\sum_{p_h}\sum_{j} Beam(y_j, h_j, t_j(y_j, h_j, \mathrm{m}), p_y, p_h). \tag{2}$$

Robust objective function

My 1987 report presented a similar formulation of the inversion as a simple fitting of traveltimes. This approach has a major drawback; the objective function is overly sensitive to artifacts of beam stacks. This problem can be avoided by devising a more robust objective function. The time axis is transformed accordingly to the relation

$$\tau = t - p_h h. \tag{3}$$

Offsets at constant τ are fitted instead of fitting traveltimes at constant offset.

A change of variables of the offset axis of beam-stacked data is convenient for saving memory needed to store the data themselves. Such a change of variables is

$$\xi = h - \frac{p_h V_0^2}{4} t, \tag{4}$$

where V_0 is a constant average velocity. The purpose of this change of variables is to localize non-zero data around $\xi = 0$ and then safely discard many zeros.

Given transformations (3) and (4) as well as the results of modeling by ray tracing (1), we can compute the relations

$$\tau_j(R_j, \mathrm{m}) = t(R_j, \mathrm{m}) - p_h h(R_j, \mathrm{m}) \tag{5a}$$

and

$$\xi_j(R_j, \mathrm{m}) = h(R_j, \mathrm{m}) - \frac{p_h V_0^2}{4} t(R_j, \mathrm{m}). \tag{5b}$$

Together with the ray tracing results (equation 1b), these relations define the function $\xi = \xi(y(R,\mathrm{m}), \tau(R,\mathrm{m}), \mathrm{m})$ implicitly. The value of semblance for a particular reflector point, and for fixed ray parameters, is $\overline{Beam}(y_j, \xi(y_j, \tau_j, \mathrm{m}), \tau_j, p_y, p_h)$, where the bar is a reminder that the data has been transformed following equations (3) and (4). When all reflector points and all ray parameters are considered, inversion can be formulated as the solution of the non-quadratic optimization problem of finding the maximum with respect to the velocity model m of the total energy

$$E(\mathrm{m}) = \sum_{p_y} \sum_{p_h} \sum_{j} \overline{Beam}(y_j, \xi(y_j, \tau_j, \mathrm{m}), \tau_j, p_y, p_h), \tag{6}$$

or in more compact notation

$$E(\mathrm{m}) = \sum_i B_i(\xi_i(\mathrm{m})), \tag{7}$$

where i is the index of the data points used, including all reflector locations R_j and all ray parameters p_y and p_h.

When an a priori estimate $\mathbf{m_0}$ of the velocity model and of its covariance matrix $\mathbf{C_m}$ exists, the objective function becomes

$$Q(\mathrm{m}) = \sum_i B_i(\xi_i(\mathrm{m})) - (\mathbf{m} - \mathbf{m_0})^T \mathbf{C_m}^{-1} (\mathbf{m} - \mathbf{m_0}). \tag{8}$$

Computing the gradient of the objective function

The maximum of $Q(\mathrm{m})$ can be found using a gradient algorithm. Implementation of a gradient algorithm requires that the gradient of the objective function be computed with respect to the model. The gradient can be expressed as

$$\nabla Q_{\mathbf{m}} = \sum_i \frac{\partial B_i(\xi_i(\mathrm{m}))}{\partial \mathrm{m}} + 2\mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_0}) = \sum_i \frac{\partial \xi_i}{\partial \mathrm{m}} \frac{\partial B_i(\xi_i)}{\partial \xi_i} + 2\mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_0}) =$$

$$\mathbf{G}^T\, \mathbf{D}_1 - 2\mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_0}), \tag{9}$$

where the derivatives are computed at fixed y, τ, p_y and p_h.

The vector $\mathbf{D}_1$ is easily computed from beam-stacked data with a finite-difference approximation of the derivative operator. This vector represents the interaction of the inversion algorithm with the actual data.

Computing the matrix of the Frechet derivatives $\mathbf{G}^T$ is more complicated. The total derivative of ξ with respect to the velocity model, computed at fixed reflector point $\bar{R}$ and velocity model $\bar{\mathrm{m}}$, can be derived applying the chain rule to the implicit function $\xi = \xi(y(R,\mathrm{m}), \tau(R,\mathrm{m}), \mathrm{m})$, and its result is

$$\left.\frac{\delta \xi}{\delta \mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} = \left.\frac{\delta y}{\delta \mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} \left.\frac{\partial \xi}{\partial y}\right|_{(\bar{y},\bar{\tau},\bar{\mathrm{m}})} + \left.\frac{\delta \tau}{\delta \mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} \left.\frac{\partial \xi}{\partial \tau}\right|_{(\bar{y},\bar{\tau},\bar{\mathrm{m}})} + \left.\frac{\partial \xi}{\partial \mathrm{m}}\right|_{(\bar{y},\bar{\tau},\bar{\mathrm{m}})}, \tag{10}$$

where $\bar{y} = y(\bar{R}, \bar{\mathrm{m}})$, $\bar{\tau} = \tau(\bar{R}, \bar{\mathrm{m}})$. The only unknowns in this expression are the Frechet derivatives $\partial\xi/\partial\mathrm{m}$, which can be evaluated rewriting the previous formula as

$$\left.\frac{\partial\xi}{\partial\mathrm{m}}\right|_{(\bar{y},\bar{\tau},\bar{\mathrm{m}})} = \left.\frac{\delta\xi}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} - \left.\frac{\delta y}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} \left.\frac{\partial\xi}{\partial y}\right|_{(\bar{y},\bar{\tau},\bar{\mathrm{m}})} - \left.\frac{\delta\tau}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} \left.\frac{\partial\xi}{\partial\tau}\right|_{(\bar{y},\bar{\tau},\bar{\mathrm{m}})} . \qquad (11)$$

Partial derivatives $\partial\xi/\partial y$ and $\partial\xi/\partial\tau$ can be evaluated using finite difference on the manifold defined by $\xi = \xi(y(R,\mathrm{m}), \tau(R,\mathrm{m}), \mathrm{m})$.

Total derivatives, with respect to the velocity model, of the transformed offset ξ, of the midpoint y, and of the transformed travel time τ, are related to the total derivatives, with respect to the velocity model, of the travel times t_s and t_r, and of the arrival locations x_s and x_r of the down-going and up-going rays. These derivatives can be evaluated by ray tracing as presented in Appendix A. The formulae relating total derivatives are easily drawn from equations (1) and (5). They are:

$$\left.\frac{\delta\xi}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} = \left.\frac{\delta h}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} - \frac{p_h V_0^2}{4} \left.\frac{\delta t}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} =$$

$$\frac{1}{2}\left(\left.\frac{\delta x_r}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} - \left.\frac{\delta x_s}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})}\right) - \frac{p_h V_0^2}{4}\left(\left.\frac{\delta t_s}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} + \left.\frac{\delta t_r}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})}\right); \qquad (12a)$$

$$\left.\frac{\delta y}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} = \frac{1}{2}\left(\left.\frac{\delta x_r}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} + \left.\frac{\delta x_s}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})}\right); \qquad (12b)$$

and

$$\left.\frac{\delta\tau}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} = \left.\frac{\delta t}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} - p_h \left.\frac{\delta h}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} =$$

$$\left(\left.\frac{\delta t_s}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} + \left.\frac{\delta t_r}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})}\right) - p_h\frac{1}{2}\left(\left.\frac{\delta x_r}{\delta\mathrm{m}}\right|_{(\bar{R},\bar{\mathrm{m}})} - \left.\frac{\delta x_s}{\delta\mathrm{m}}\right|_{\bar{R},\bar{\mathrm{m}}}\right). \qquad (12c)$$

A priori assumption of velocity-model smoothness

Tomographic inversion only resolves the low-wavenumbers of the velocity model because it is based on traveltimes, integral measures of velocity. Furthermore ray-tracing modeling renders the back-projection operator $\mathbf{G}$ singular. The poorly resolved components of the velocity model must be constrained so as to avoid instability during the inversion process. A common approach for constraining poorly determined components of the velocity model is to impose a smoothness condition using a derivative operator as the inverse of the model covariance matrix $\mathbf{C_m}$

(Toldi, 1985). This approach is straightforward and easyly implemented but it has the disadvantage that the optimization problem is solved for many more unknown parameters than are needed to describe a smooth velocity model. The cost of each iteration is therefore higher than necessary.

I use a parametrization of the velocity function that takes the smoothness condition into account implicitly. I parametrized the velocity function with B-spline basis functions that have a continuous second derivative (Diercks, 1975).

OPTIMIZATION ALGORITHMS

The optimization problem of finding the maximum of the objective function (8) can be solved sequentially using two different algorithms. The first converges to a robust estimate of the low-wavenumber components of the velocity model starting from an initial model that is generally different from the true model. The second converges to the best estimates of the velocity model that is consistent with the beam stacked data; starting from a nearly correct initial model. The first algorithm is more reliable in solving a global convergence problem, while the second is more efficient in solving one of local convergence.

Global convergence to the maximum of the objective function

The objective function of equation (8) is highly non-quadratic. The fitting functions $B_i(\xi_i)$ are non-quadratic and often they are not even unimodal because of local maxima caused by multiples and artifacts. Further, the modeling functions $\xi_i(\mathbf{m})$ are not linear because traveltime is not a linear function of velocity.

For highly non-quadratic optimization problems algorithms with the best global convergence properties are those based on the first derivatives of the objective function. Among the several gradient algorithms I choose a version of the conjugate gradient algorithm derived by Polak-Ribiere (Luenberger, 1984) that is particularly efficient in solving non-quadratic problems.

The objective function has many local maxima that must be avoided so as to converge towards the global maximum. Some of these local maxima result from the dependence of the raypath upon velocity. Raypaths are most strongly influenced by low-wavenumber components of the velocity model. It therefore is fundamental to determine these components before attempting to resolve higher wavenumber components of the velocity model. One heuristic solution to the problem of local maxima is thus starting the optimization from the low-wavenumber components of the velocity model and slowly increasing the bandwidth of the model. Following this procedure also prevents from converging towards the local maxima caused by the multiples. These local maxima usually correspond to interval velocity functions varying rapidly. Constraining the solution to be stiff until the algorithm has converged to a model that approximately predicts the traveltimes of the primaries, avoids the following of local maxima caused by multiples. In other words, I constrain the velocity model to predict ξ_i that are close to the maxima of the fitting functions $B_i(\xi_i)$ correspondent to the primaries, preventing the solution from following the ξ_i close to local maxima resulting from multiples.

In practice, I start the inversion algorithm using few B-spline basis functions for describing the velocity model. When the conjugate gradient algorithm has converged to a maximum for the current parametrization I increment the number of basis functions and restart the iterative algorithm. This procedure can be repeated until the solution has the desired bandwidth. The scheme of the algorithm that I used for achieving global convergence is thus

Set initial parametrization and starting model $\mathbf{m}_0 = \bar{\mathbf{m}}$.

Set $k = 0$.

Compute the gradient $\mathbf{g}_0 = \nabla Q(\mathbf{m}_0) = \mathbf{G}_0^T \mathbf{D}_1 - 2\mathbf{C_m}^{-1}(\mathbf{m}_0 - \bar{\mathbf{m}})$.

Set the search direction $\delta \mathbf{m}_0 = \mathbf{g}_0$.

Find α_0 *that maximizes* $Q(\mathbf{m}_0 + \alpha_0 \delta \mathbf{m}_0)$.

Update the model $\mathbf{m}_1 = \mathbf{m}_0 + \alpha_0 \delta \mathbf{m}_0$.

Set $k = k + 1$.

Compute the gradient $\mathbf{g}_k = \nabla Q(\mathbf{m}_k) = \mathbf{G}_k^T \mathbf{D}_1 - 2\mathbf{C_m}^{-1}(\mathbf{m}_k - \bar{\mathbf{m}})$.

Compute the search direction $\delta \mathbf{m}_k = \mathbf{g}_k + \frac{(\mathbf{g}_k - \mathbf{g}_{k-1})^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \delta \mathbf{m}_{k-1}$.

Find α_k *that maximizes* $Q(\mathbf{m}_k + \alpha_k \delta \mathbf{m}_k)$.

Update the model $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k \delta \mathbf{m}_k$.

Check for convergence.

Increase the bandwidth of the parametrization.

Set $\mathbf{m}_0 = \mathbf{m}_{k+1}$.

Check for convergence.

The line search is an expensive part of this algorithm as it requires tracing the rays from all the reflectors for each evaluation of the objective function. There are many algorithms for accurately estimating the stepsize that require a limited number of function evaluations. I used an iterative line search algorithm that fits a quadratic function to the objective function and evaluates the objective function at the maximum of this quadratic curve (Luenberger, 1984).

Local convergence and resolution

The previous algorithm converges to a model close to the true model; however near the solution its rate of convergence can be improved using the second derivatives of the objective function with respect to the velocity model. Improving the convergence rate of the algorithm saves a considerable amount of computation time: each iteration requires many ray tracings through the entire model.

The Hessian of the objective function can be derived from the gradient in equation (9):

$$\mathbf{H}(\mathbf{m}) = \frac{\partial^2 Q}{\partial \mathbf{m}^2} = \sum_i \frac{\partial^2 B_i(\xi_i(\mathbf{m}))}{\partial \mathbf{m}^2} - 2\mathbf{C_m}^{-1} =$$

$$\sum_i \frac{\partial^2 \xi_i}{\partial \mathbf{m}^2} \frac{\partial B_i(\xi_i)}{\partial \xi_i} + \sum_i \frac{\partial \xi_i}{\partial \mathbf{m}} \frac{\partial^2 B_i(\xi_i)}{\partial \xi_i^2} \frac{\partial \xi_i}{\partial \mathbf{m}} - 2\mathbf{C_m}^{-1} =$$

$$\mathbf{S}\,\mathbf{D}_1 + \mathbf{G}^T \mathbf{D}_2 \mathbf{G} - 2\mathbf{C_m}^{-1}. \tag{13}$$

The first term in the Hessian depends on the second derivatives of ray tracing with respect to the velocity model. I drop this term because it is too expensive to compute. Dropping this term implies the approximation of ray tracing with a linear function. Another reason for neglecting the first term is that it also depends on $\mathbf{D}_1$, that decreases as the velocity model approaches the true velocity model.

The search direction of the quasi-Newton method I used is the solution of the linear system

$$\left[\mathbf{G}^T(-\mathbf{D}_2)\mathbf{G} + 2\mathbf{C_m}^{-1}\right] \delta\mathbf{m} = \mathbf{G}^T\mathbf{D}_1 - 2\mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m}_0). \tag{14}$$

The solution of this system is equivalent to the least-squares solution of the system

$$\begin{pmatrix} \sqrt{-\mathbf{D}_2}\mathbf{G} \\ \sqrt{2\mathbf{C_m}^{-1}} \end{pmatrix} \delta\mathbf{m} \simeq \begin{pmatrix} (\sqrt{-\mathbf{D}_2})^{-1}\mathbf{D}_1 \\ -\sqrt{2\mathbf{C_m}^{-1}}(\mathbf{m} - \mathbf{m}_0) \end{pmatrix}, \tag{15}$$

that can be efficiently solved using a conjugate gradient algorithm as LSQR (Paige and Saunders, 1982). To solve the system in (15) it is necessary to use the covariance matrix $\mathbf{C_m}$ to damp the solution because the back-projection matrix $\mathbf{G}$ is singular.

The second derivatives $\mathbf{D}_2$ estimated from the data are not reliable enough to be used in equation (15). Generating the example that I present in the following section I used a constant value for the second derivatives. Probably a better solution is to set the second derivatives to be inversely proportional to the depth of the reflectors, because the width of the peaks in the beam stacks is proportional to the width of the Fresnel zones of the reflections.

The scheme of the quasi-Newton method that I used is

Set starting model $\mathbf{m}_0$.
Set $k = 0$.

Compute the search direction

$$\delta\mathbf{m}_k = \left[\mathbf{G}_k^T(-\mathbf{D}_2)\mathbf{G}_k + 2\mathbf{C_m}^{-1}\right]^{-1}\left[\mathbf{G}_k^T\mathbf{D}_1 - 2\mathbf{C_m}^{-1}(\mathbf{m}_k - \mathbf{m}_0)\right].$$

Find α_k *that maximizes* $Q(\mathbf{m}_k + \alpha_k\delta\mathbf{m}_k)$.

Update the model $\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\delta\mathbf{m}_k$.

Check for convergence.

Set $k = k + 1$.

In the quasi-Newton algorithm I use the same line-search scheme that I used in the conjugate gradient algorithm. At this stage of the inversion, when the current model is close to the true model, I could save computer time evaluating the objective function using the gradient of ray tracing with respect to the velocity model instead of tracing rays for each α_k. Therefore, to evaluate the objective function I could use the approximation

$$Q(\mathbf{m}_k + \alpha_k\delta\mathbf{m}_k) \simeq \sum_i B_i\left(\xi_i(\mathbf{m}_k) + \alpha_k\frac{\partial\xi_i}{\partial\mathbf{m}}\delta\mathbf{m}_k\right) - (\mathbf{m}_k + \alpha_k\delta\mathbf{m}_k - \mathbf{m}_0)^T\mathbf{C_m}^{-1}(\mathbf{m}_k + \alpha_k\delta\mathbf{m}_k - \mathbf{m}_0). \qquad (16)$$

RESULTS OF 1-D INVERSION OF SYNTHETIC DATA

I tested the inversion scheme of the previous sections using the synthetic CMP gather shown in Figure 1. I computed the beam stacks for 10 values of the offset ray parameter p_h; doing this only for a null value of the midpoint ray parameter p_y, because the medium is laterally homogeneous. The beam stack were slightly smoothed in time and space before being utilized by the inversion algorithm.

Figure 4 shows the result of 11 iterations applying the conjugate gradient algorithm in the inversion of data shown in Figure 1. The true model is indicated by a dashed line; the result of the final iteration is shown by a thicker line. I started the algorithm using one basis function every 240 meters; the final solution was found using a basis function every 120 meters. I tried the procedure with different starting velocity models and the algorithm always converged to a solution similar to that of Figure 4. The final iteration does not resolve the low velocity layer well. However the final model is a good approximation of the true velocity and it could be used as starting model for a more powerful, but also less robust, algorithm.

Figure 5 shows the contour plot of a slice taken at constant offset ray parameter of beam-stacked data I used for inversion. Traveltime curves as a function of transformed offset are superimposed on the contour plot. The traveltime curve correspondent to the last iteration is drawn using a thick line. The algorithm converged from a velocity model that did not predict the traveltimes of the beams to a solution that predicts the position of the peaks in the data fairly well. The slow increase of the model bandwidth has prevented the iterative inversion from being influenced by the data peaks caused by multiples. Nevertheless, the final solution did not accurately predict the position of the maxima corresponding to the first and second reflectors. The data contains more information on the velocity within the first and second layers than what the inversion algorithm was able to extract.

Figure 6 shows the result of 3 iterations of the quasi-Newton algorithm starting from the final iteration result of the conjugate gradient algorithm shown in Figure 4. The true model is indicated by a dashed line and the result of the last iteration is shown by a thicker line. The velocity model was parametrized using a basis function every 120 meters. The algorithm converged to a good, smooth approximation of the true velocity model.

Figure 7 shows the contour plot of a slice of beam stack taken at constant offset ray parameter. The traveltime curves as a function of the transformed offset are drawn for each iteration. The traveltime curve correspondent to the final iteration is indicated by a thick line. The final solution predicts the peaks position of beam stacks almost perfectly.

CONCLUSIONS

I have presented a theory for estimating the lower wavenumber of the velocity model from beam-stacked data. The method is valid in presence of geological structure and lateral velocity variations and does not require to pick events in the data.

Synthetic data results show the validity of the theory in the simple case of a layered medium. So as to I need to test the method with field data and to generalize the implementation of the algorithm to the 2-D case.

ACKNOWLEDGMENTS

I thank my officemates John Etgen and Jos Van Trier for their many discussion about the velocity estimation problem.

REFERENCES

Biondi, B., 1987, Interval velocity estimation from beam-stacked data: SEP–**51**,13–27.

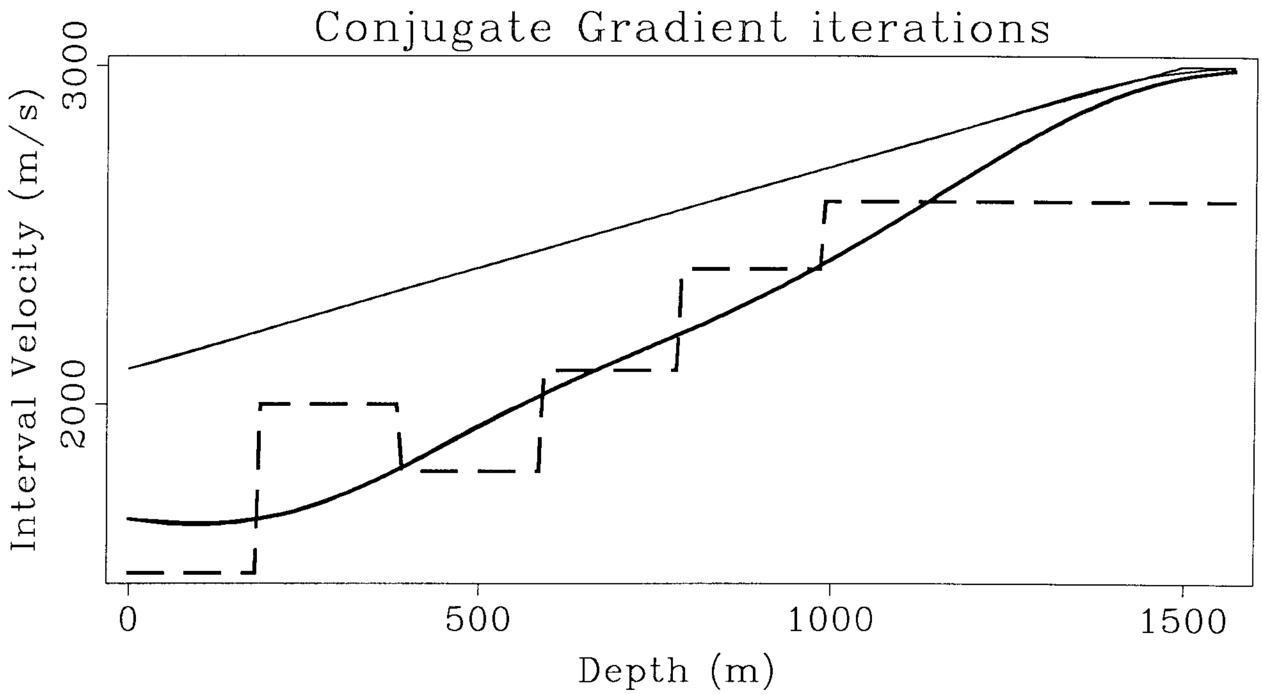

FIG. 4. The result of 11 iterations of the conjugate gradient algorithm applied to the inversion of data shown in Figure 1. The true model is drawn with a dashed line; the result of the final iteration is shown by a thicker line. The inversion started using one basis function every 240 meters to describe the velocity model; it ended with one basis function every 120 meters. The final iteration did not resolve the low velocity layer well.

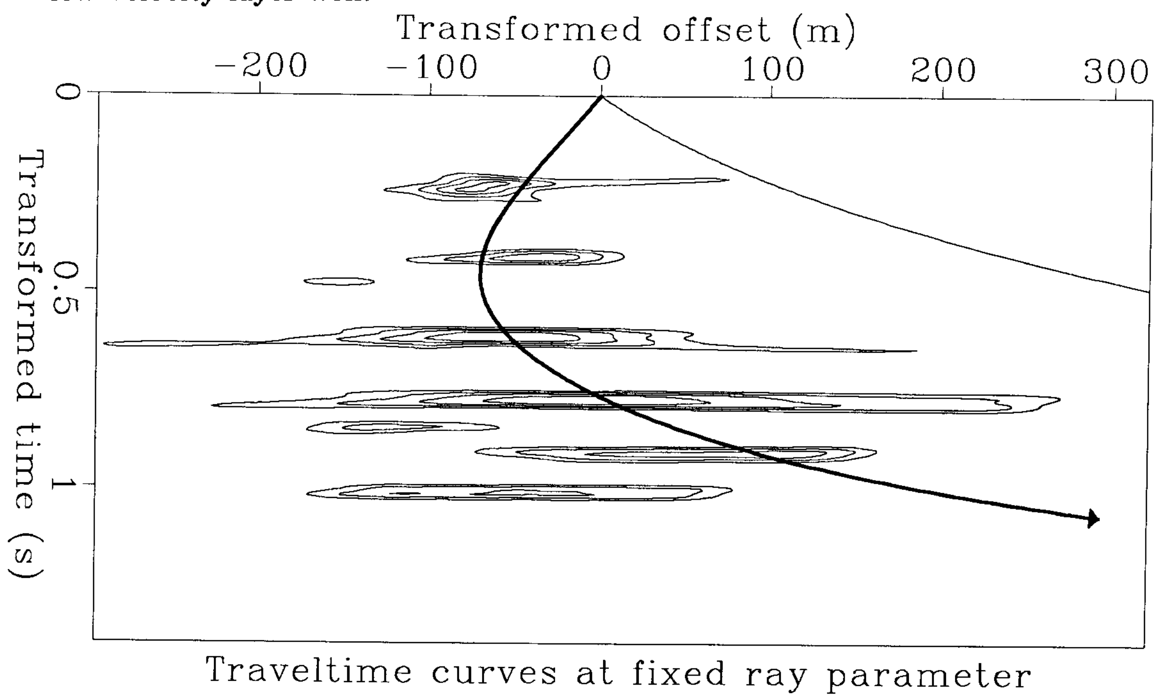

FIG. 5. Contour plot of a slice of the beam-stacked data taken at constant offset ray parameter. The traveltime curves as a function of the transformed offset are superimposed on the contour plot. The traveltime curve correspondent to the last iteration is drawn using a thicker line. The inversion converged to a solution that predicts fairly well the position of the peaks in the data starting from a velocity function that is distant from the true model.

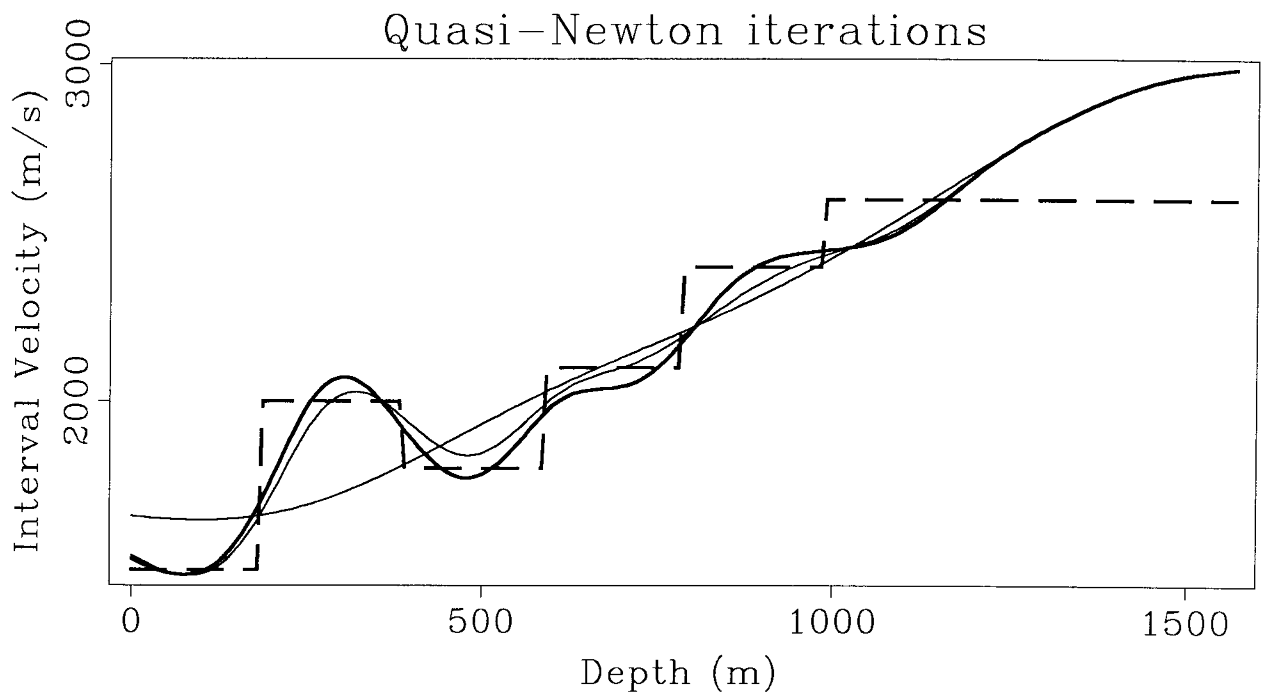

FIG. 6. The result of 3 iterations of the quasi-Newton algorithm applied to the data shown in Figure 1. The true model is indicated with a dashed line; the result of the final iteration is shown with a thicker line. The velocity model was parametrized using a basis function every 120 meters. The inversion converged to a good smooth approximation of the true velocity model.

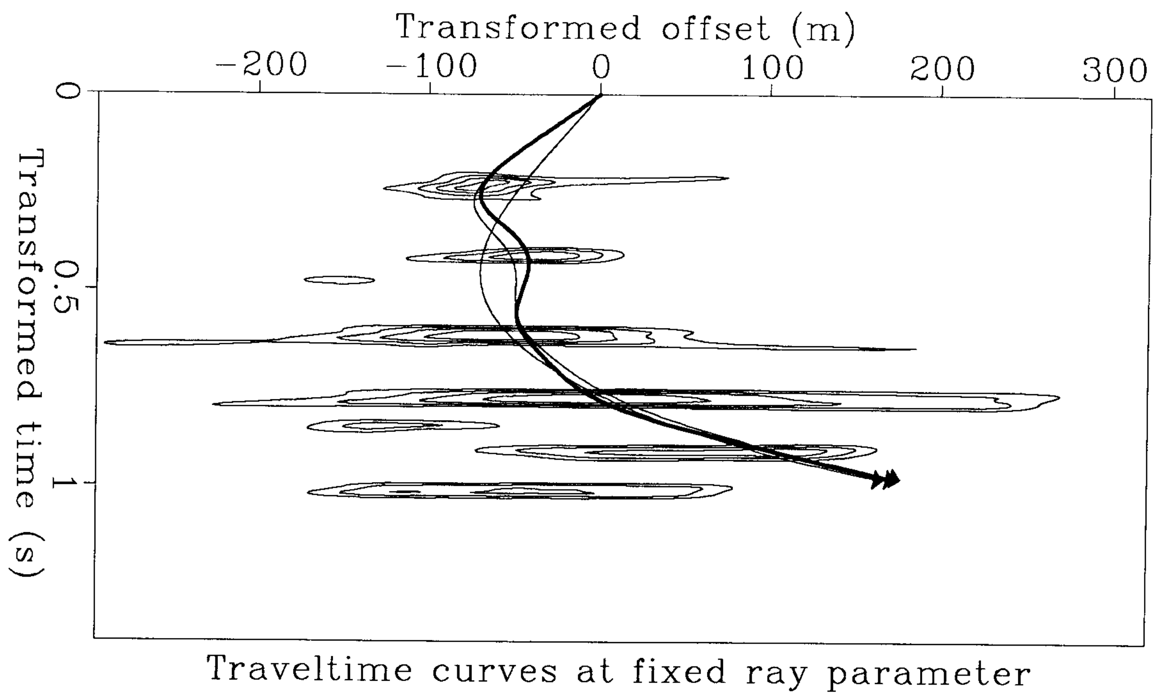

FIG. 7. Contour plot of a slice, taken at constant offset ray parameter, of the beam-stack data. The traveltime curves as a function of the transformed offset are drawn for each iteration. The traveltime curve correspondent to the last iteration is drawn with a thicker line. The result of the last iteration predicts almost perfectly the peaks of beam stacks.

Bishop, T.N., Bube, K.P., Cutler, R.T., Langan, R.T., Love, P.L., Resnick, J.R., Shuey, R.T., Spindler, D.A., and Wyld, H.W., 1985, Tomographic determination of velocity and depth in laterally varying media: Geophysics, **50**, 903–923.

Diercks, P., 1975, An algorithm for smoothing, differentiation and integration of experimental data using splines functions. J. Comp. Appl. Maths. **1**, 165–185

Fowler, P., 1985, Migration velocity analysis by optimization: linear theory: SEP–**44**, 1–20.

Harlan, B., and Burridge, R., 1983, A tomographic velocity inversion of unstacked data: SEP–**37**, 1–7.

Kostov, C., and Biondi, B., 1987, Improved resolution of slant stacks using beam stacks: SEP–**51.**, 343–350

Luenberger, D.G., 1984, Linear and nonlinear programming (2nd edition): Addison-Wesley Publishing Company Inc. .

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

Sword, C., 1984, The method of controlled directional reception: SEP–**41**, 369–395.

Sword, C., 1986, Tomographic determination of interval velocities from picked seismic data-theory and synthetic results: SEP–**48**, 1–33.

Sword, C.H., 1987, Tomographic determination of interval velocities from reflection seismic data: The method of controlled directional reception: Ph.D. thesis, Stanford University.

Toldi, J.L., 1985, Velocity analysis without picking: Ph.D. thesis, Stanford University.

Woodward, M., 1988, Wave equation tomography: I: SEP–**57**.

APPENDIX A

RAY TRACING IN A LAYERED MEDIUM

In inverting beam stacks I need to perform ray tracing as shown in Figure 3. The down-going and up-going rays are traced, starting with assigned ray parameters and from an assigned reflector point, until they reach the surface. This is not a classical "two points" ray tracing because the arrival point is unfixed. Therefore, Fermat's principle cannot be invoked for a simple computation of the gradient of the traveltimes t_s and t_r with respect to the velocity model. The effect on the traveltime of the changes of the ray path caused by velocity variations is not of the second order any longer, and therefore must be taken into account for a correct gradient computation. Further I need the gradient of the arrival points x_s and x_r with respect to the velocity model to compute the gradient of the objective function.

I developed a simple algorithm to perform the ray tracing needed by the inversion in a layered medium. The algorithm I devised for the general 2-D case is more complicated and not yet fully-tested; I postpone its presentation to a future report.

Rays are traced in a horizontally-layered medium with slowness function $S(z;\mathbf{m})$, where z is depth and $\mathbf{m}$ is the vector of velocity parameters. The slowness function, or more precisely slowness squared, is parametrized using B-spline; that renders the evaluation of the function easy and accurate. The evaluation of its derivatives with respect to the spatial axis or with respect to the model parameters is equally easy.

A ray of ray parameter p is continued upward from the location (x_{k-1}, z_{k-1}) to the location (x_k, z_k) using Snell's law. That is to say:

$$x_k = x_{k-1} + \frac{\Delta z p}{\sqrt{S_k^2 - p^2}}, \qquad (A1)$$

where $\Delta z = z_k - z_{k-1}$ is constant, and S_k is the slowness at the intermediate depth $\bar{z} = (z_{k-1} + z_k)/2$. The whole path of this ray can be traced starting from the initial reflector point $R(x_r, z_r)$.

The total traveltime of the ray $t(R, \mathbf{m})$ is the summation of the times along the ray path,

$$t(R,\mathbf{m}) = \sum_{k=1}^{N} \Delta t_k = \sum_{k=1}^{N} \frac{\Delta z S_k^2}{\sqrt{S_k^2 - p^2}}. \qquad (A2)$$

The gradient of the total traveltime with respect to the parameter m_j of the velocity model is thus

$$\frac{\partial t(R,\mathbf{m})}{\partial m_j} = \sum_{k=1}^{N} \frac{\partial \Delta t_k}{\partial S_k^2}\frac{\partial S_k^2}{\partial m_j} = \sum_{k=1}^{N} \frac{\Delta z}{\sqrt{S_k^2 - p^2}} \left[1 - \frac{S_k^2}{2(S_k^2 - p^2)}\right] \frac{\partial S_k^2}{\partial m_j}. \qquad (A3)$$

The arrival point of the ray $x(R, \mathbf{m})$ is

$$x(R,\mathbf{m}) = \sum_{k=1}^{N} \Delta x_k = \sum_{k=1}^{N} \frac{\Delta z p}{\sqrt{S_k^2 - p^2}}. \qquad (A4)$$

The gradient of the arrival point with respect to the parameter m_j of the velocity model is therefore

$$\frac{\partial x(R,\mathbf{m})}{\partial m_j} = \sum_{k=1}^{N} \frac{\partial \Delta x_k}{\partial S_k^2}\frac{\partial S_k^2}{\partial m_j} = \sum_{k=1}^{N} - \frac{\Delta z p}{2\sqrt{(S_k^2 - p^2)^3}} \frac{\partial S_k^2}{\partial m_j}. \qquad (A5)$$

Tracing a ray up from the reflector or down from the surface, yields to exactly the same raypath. However in a general 2-D medium, the gradients of traveltime and of arrival location depend upon the ray direction of propagation. In a layered medium also the gradients are independent from the direction of propagation. In a practical implementation of the ray tracing algorithm it is convenient to exploit this property and to trace the ray downward instead of upward.

REPORT AUTHORS

116