Velocity analysis by nonlinear optimization of phase-contoured shot profiles

Jos van Trier

ABSTRACT

The curvature of reflections from planar beds in migrated shot profiles depends on the migration velocity that is used. Overmigrated reflections curve downwards, while undermigrated ones curve upwards. Thus, curvature can be used as an objective function in an optimization scheme to determine the correct migration velocity.

Curvature is estimated from lines of equal instantaneous phase in a shot profile by taking the second derivative of these phase contours. Phase contouring can also be used to calculate local dip and find static corrections.

In the optimization, the velocity model is parametrized into basisfunctions, so that the number of unknowns is reduced considerably. The optimization method is a non-derivative method, i.e., no gradient of the objective function needs to be calculated. The optimization is also nonlinear: no linear approximations are made to relate model and data. Each iteration in the optimization consists of a shot-profile migration with the current velocity model. The estimation of the curvature is independent of the type of migration used in the optimization. So, by using a depth migration, the model can be expressed directly in interval velocities.

For general geology (i.e., curved reflectors), several shot profiles are needed to determine the velocity model. The curvature analysis in each shot profile (along the geophone direction) then has to be combined with a curvature estimation in the shot direction.

INTRODUCTION

Velocity analysis through stacking is normally done before migration. However, errors in velocity analysis cause artifacts in migration, and stacking velocities are not necessarily the same as migration velocities. By combining velocity analysis with migration, these interdependency problems can be eliminated: a recorded dataset may be migrated with a guessed velocity model, and the resulting artifacts used to update the guess.

In the last SEP-report (Van Trier and Rocca (1986)), I described a method to estimate velocities from unfocused diffraction events in migrated stacked sections. The shape of

a diffraction event in the migrated section depends on the migration velocity. When the migration velocity is too high, the shape is a "frown," when the velocity is too low, it is a "smile."

Reflection events in shot profiles display a similar behavior: a reflection from a flat (horizontal or dipping) bed shows up either curved upwards or downwards in the migrated profile, depending on the migration velocity that was used. This is illustrated in Figure 1. Figure 1a shows a reflection from a dipping bed; Figures 1b, c, and d display migrations of the event with velocities of 2.0 km/s (the correct velocity), 1.8 and 2.2 km/s.

Estimation of the curvature in a shot profile makes it possible to determine if the right velocity was used in the migration of that profile. A measure of the curvature can be used as an objective function in a nonlinear optimization scheme that iteratively finds the correct velocity model, where each iteration consists of a single shot profile migration. Traditionally, such optimization methods use the power or the semblance of the stacked section as an objective function. Using curvature has the advantage that only one shot profile is needed for the velocity estimation, while still handling more general geometry than the NMO-velocity analysis, in which the layers have to be flat *and* horizontal. Also, the curvature is sensitive to migration velocity over a wider range of velocities.

The curvature estimation method is independent of the type of migration used in the optimization. So, by using a depth migration, the velocity model can be directly expressed in interval velocities instead of stacking or rms-velocities.

The velocity model is described in terms of basisfunctions (like, e.g., Nolet, Van Trier and Huisman (1986)). This limits the number of parameters drastically, enabling the use of a non-derivative optimization method. Furthermore, the velocity model need not be constrained by adding a model-norm (of the model and/or its derivative) to the objective function: the smoothness of the model can automatically be accomplished by using smooth basisfunctions.

Of course the velocity estimation is not limited to one profile; in fact, one wants to use the full (and partly redundant) information in all the shot profiles. I will describe in a later section how to integrate the information of several shot profiles into one velocity model and how to handle curved reflectors.

The curvature estimation is based on picking events by phase contouring. Reflection events have continuous instantaneous phase across the profile. Lines of equal instantaneous phase are extracted by a contouring algorithm. An event is then represented by a contour line, and the curvature of the event is estimated by taking second derivatives of the contour line. In the last section I will discuss some other applications of the phase contouring method.

CURVATURE

Automatic picking by phase contouring

The human eye easily picks out reflections and their curvature on a shot record. It detects the strong amplitude events and then follows each event along lines of equal phase. This process can be mimicked with the help of complex seismic trace analysis.

A discrete complex trace, F_t (t is time), has as its real part the seismic trace f_t. Its

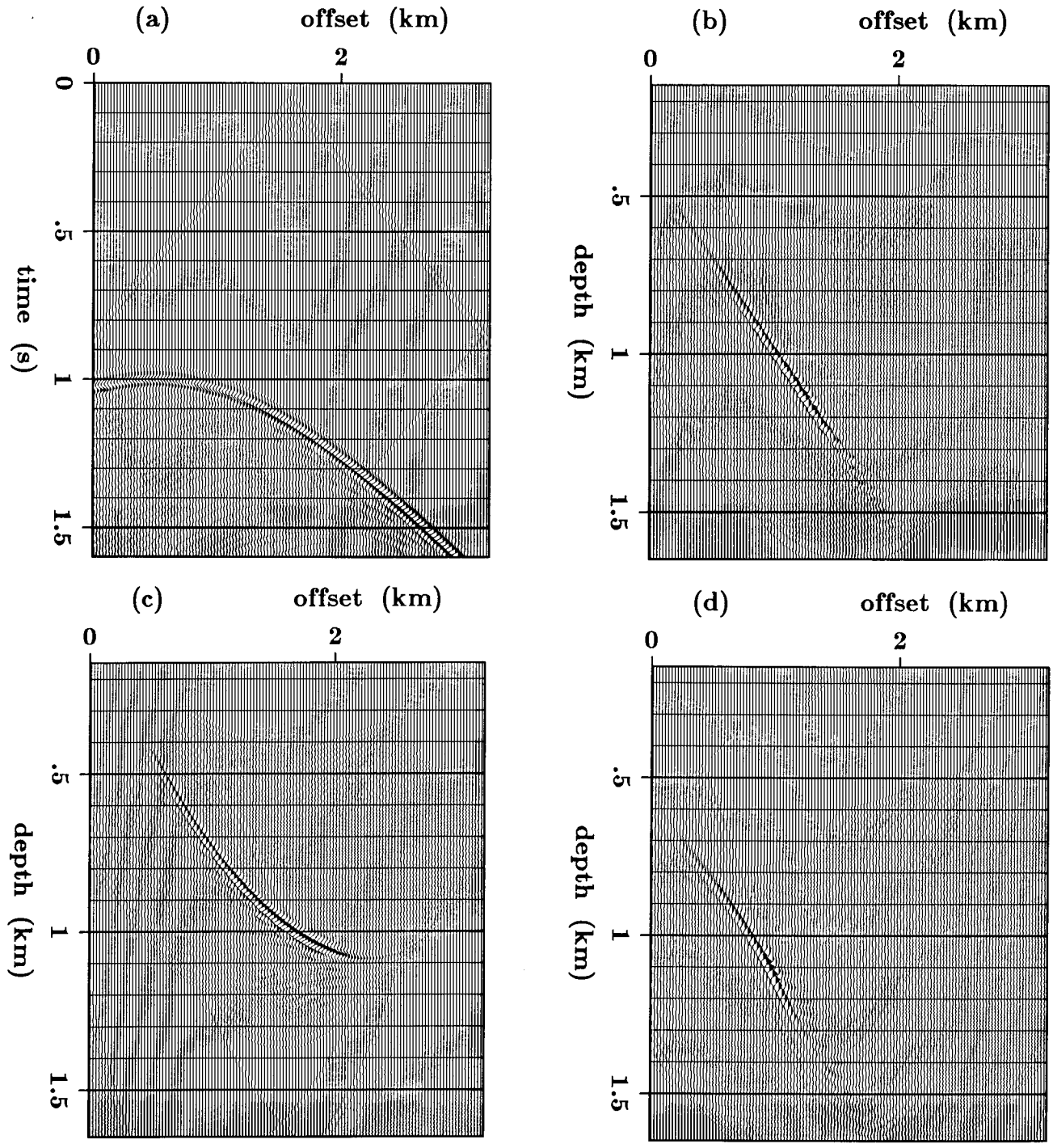

FIG. 1. Reflection from a dipping bed (a), migrated with 2.0 (b), 1.8 (c), and 2.2 km/s (d). The correct velocity is 2.0 km/s.

imaginary part, f_t^*, consists of the Hilbert transform of f_t. So, F_t can be written as:

$$F_t = f_t + i f_t^*, \quad f_t^* = \mathcal{H}(f_t), \tag{1}$$

where $\mathcal{H}$ denotes the Hilbert transform. The Hilbert transform can be implemented either in the time or Fourier domain (see Taner, Koehler and Sheriff (1979)). In this paper I used a time domain operator with a length of 21 points. The instantaneous phase, ϕ_t, and envelope, a_t, are now defined as:

$$\phi_t = \tan^{-1}\left(\frac{f_t^*}{f_t}\right) \tag{2}$$

$$a_t = |F_t| = \sqrt{f_t^2 + f_t^{*2}} \tag{3}$$

Figure 2 shows part of a seismic trace and its instantaneous phase and envelope. The reflection events in a shot profile can be traced along lines of equal instantaneous phase. However, because phase is independent of reflection strength, the phase image of a shot profile will be heavily distorted by the influence of noise. Therefore, the instantaneous phase is weighted by the envelope to select out the major events:

$$w_t = a_t \, \phi_t \tag{4}$$

Figure 2d shows the weighted instantaneous phase of the seismic trace in Figure 2a. The trace is taken from dataset 27 of the field profiles of Yilmaz and Cumro (1983). Figure 3 displays this field profile, deconvolved with Claerbout's simultaneous-deconvolution method.

The w_t-traces are calculated for each offset h in the shot profile, resulting in a two-dimensional function $w_{h,t}$. The w-map of Figure 3 is shown in Figure 4. Once the $w_{h,t}$-map is calculated for the entire shot profile, the lines of equal phase can be extracted by a contouring algorithm. There are many efficient contouring algorithms available, I chose one that has been in use at SEP for some years now, and that has been developed by Cottafava and Le Moli (1969). The algorithm "hooks on" onto a point of given amplitude, and then traces all neighboring points of equal amplitude until it reaches a boundary or the starting point again. Linear interpolations are performed when the amplitude of the contour is not exactly equal to the amplitude of a grid point. The contour map of Figure 4 is shown in Figure 5.

Picking events by phase contouring is more robust than picking peaks in the actual seismic trace, because instantaneous phase emphasizes the continuity of events, as observed by Taner et al (1979). Also, the "spikiness" of the weighted phase (see Figure 2d) sharply defines the event. It is not necessary to pick peaks; slicing through the spiked events by contouring is almost as accurate as selecting the peaks themselves.

Curvature estimation

Assume that one reflection event is represented by a certain contour $t(h)$. Note that the dimensionality is reduced by one. Curvature is now estimated by a 3-point second-derivative operator:

$$c(t(h),h) = \frac{\partial^2}{\partial h^2}\, t(h) = \frac{t(h-\Delta h) - 2t(h) + t(h+\Delta h)}{\Delta h^2} \tag{5}$$

There are some practical matters to consider:

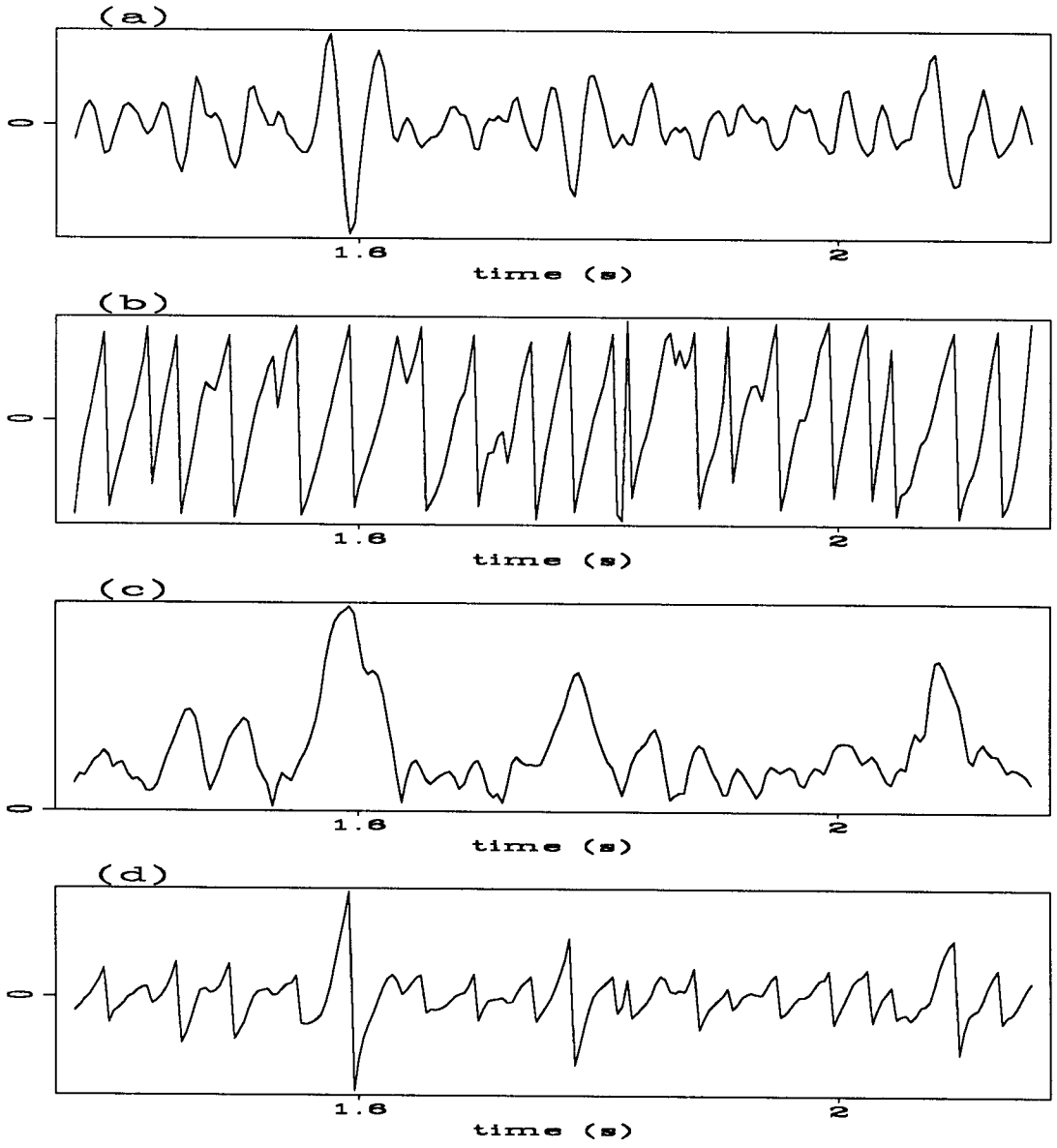

FIG. 2. Seismic trace (a), instantaneous phase (b), envelope (c), and weighted instantaneous phase (d).

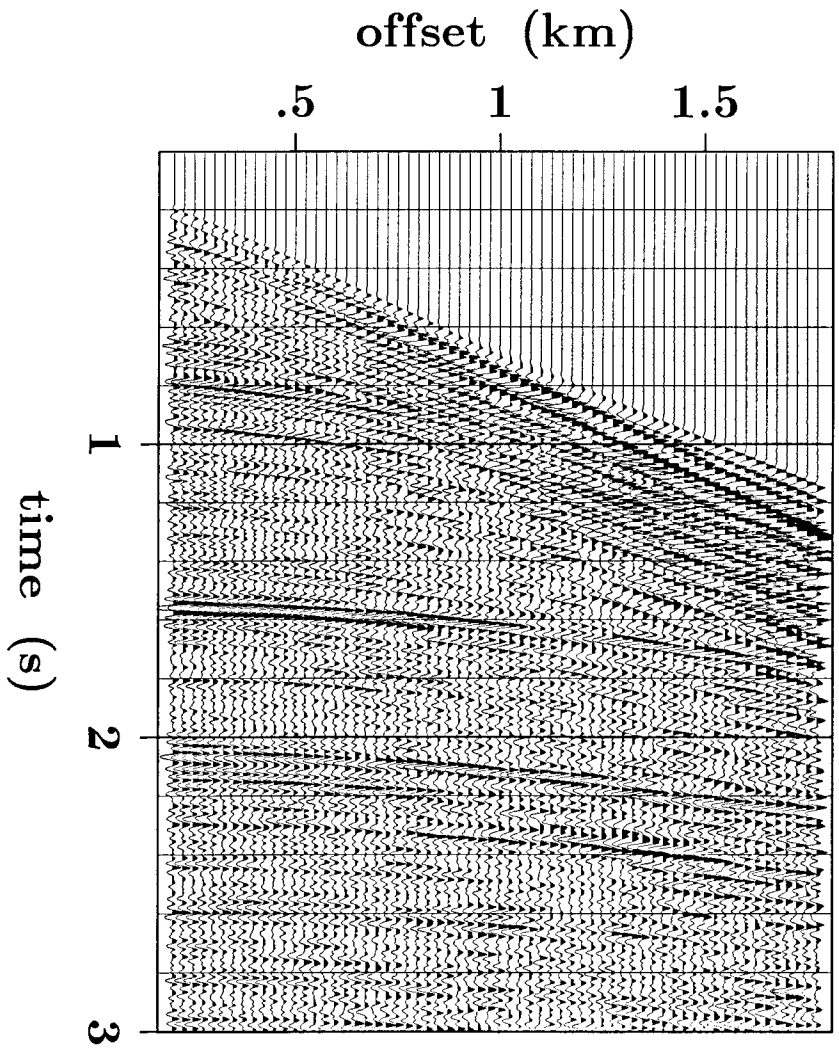

FIG. 3. Deconvolved profile 27 from Yilmaz and Cumro (1983).

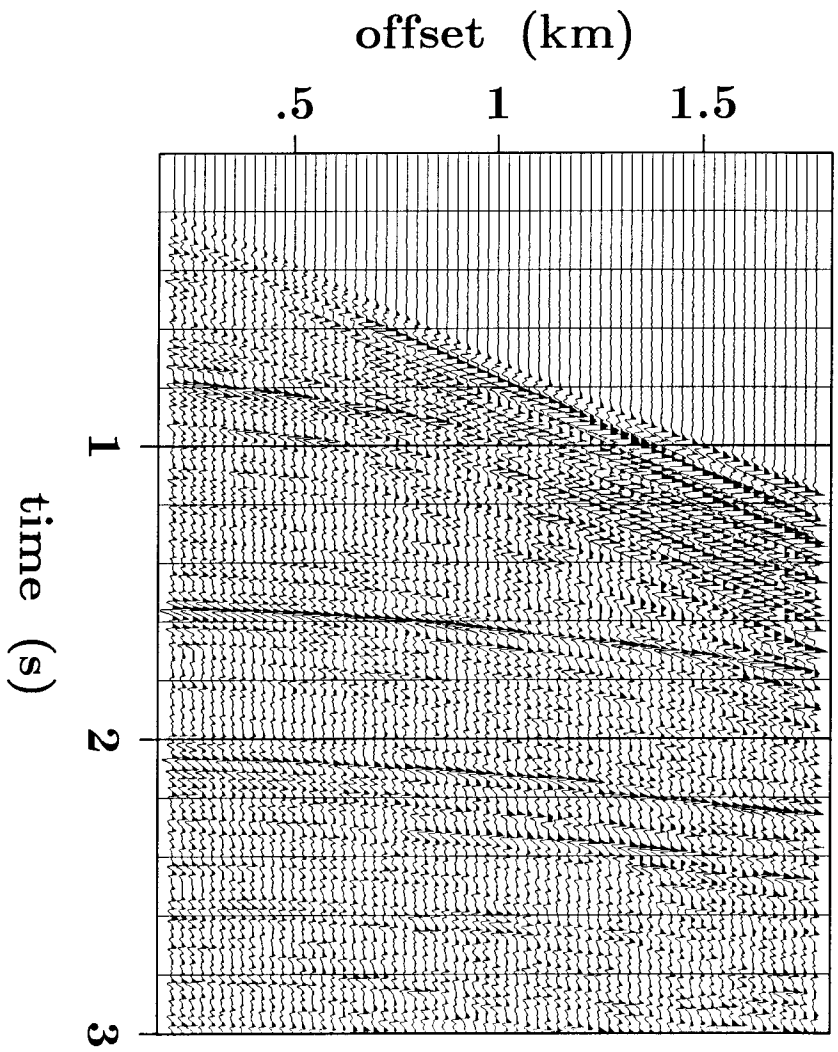

FIG. 4. Weighted instantaneous phase section of the 27-profile.

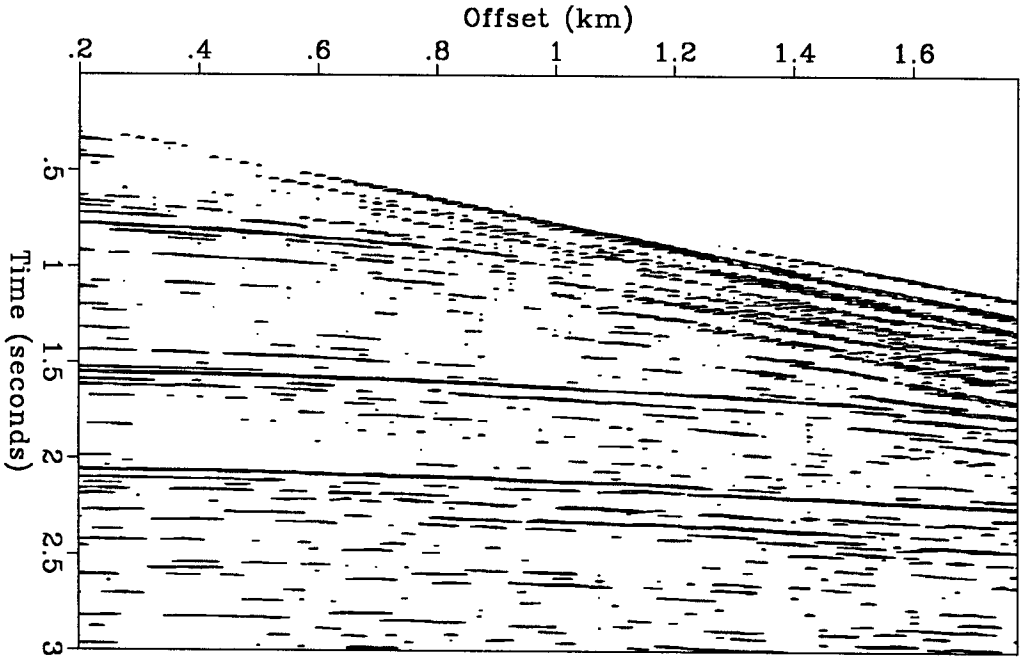

FIG. 5. Phase contour map of the 27-profile.

1. The linear interpolation causes a zig-zag pattern in the contour. To smooth out the resulting fluctuation in the curvature, the finite-difference interval Δh is chosen to be larger than the offset sampling rate and a moving average filter is applied.

2. The "turning points" of the contour have to be discarded, because the curvature is of course incorrectly calculated in those points.

3. As can be seen in Figure 5, many small contours are caused by noise, or they are too small to determine reliably the curvature. These contours are discarded by setting some minimum value for the number of points in a contour.

Figure 6 shows the curvature of the largest contours in Figure 5. Although the figure is not conclusive about the absolute values of the curvature in the shot profile, it illustrates that all the events have a positive curvature.

VELOCITY INVERSION

Objective function

The squares of the curvature can be summed over the whole shot profile and used as

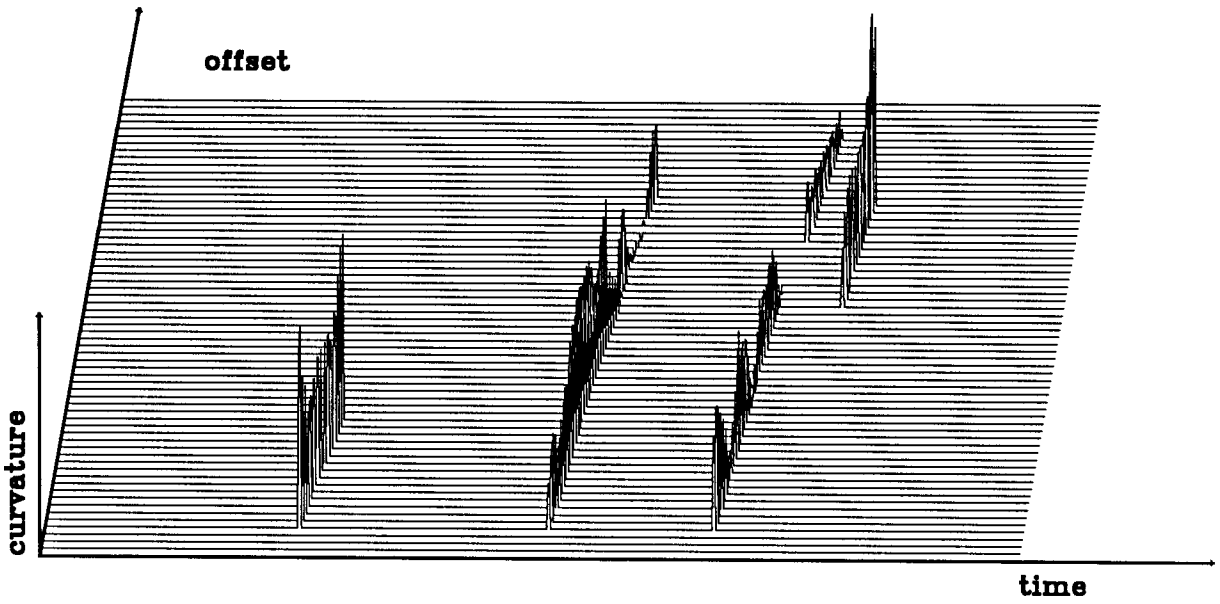

FIG. 6. Curvature of the main events in Figure 5.

objective function in an optimization scheme that minimizes this objective function:

$$\mathbf{F}_2(v(x,z)) \;=\; \frac{1}{N} \sum_h \sum_t \left(c_{h,t}\right)^2, \tag{6}$$

where $\mathbf{F}_2$ is the objective function, which depends on the velocity model $v(x,z)$, a function of x, the lateral direction, and z, the depth. The objective function is normalized by N, the number of points for which the curvature is determined.

The curvature-norm can be compared to the power of the stacked trace, a conventional norm in velocity analysis. Consider a reflection from a flat bed. The event is migrated with a range of velocities and its curvature estimated for each velocity. Alternatively, the traces in the profile can be summed over offset, and the power of the stacked trace can be used as objective function. Figure 7 displays the two norms for a reflection event with a velocity of 2.0 km/s. The two norms are strikingly different: the stack-norm is narrow around 2.0 km/s and flat at the sides, the curvature-norm is broad, and shaped like a parabola. This means that the curvature-norm is well-suited for nonlinear problems, where the starting velocity model is far from the real velocity. The stack-norm is probably preferred in the final stages of the optimization, when the velocity model converges to the earth velocity. To combine these effects, the two norms can be summed (the power of the stacked trace actually has to be subtracted from the curvature-norm, if an optimization is used that *minimizes* the objective function).

Some information is lost by taking the square of the curvature: it is not possible any more to distinguish between up- and downward curved events. Therefore, one might consider just using the absolute value of the sum of the curvatures as objective function:

$$\mathbf{F}_1(v(x,z)) \;=\; \frac{1}{N} \left| \sum_h \sum_t c_{h,t} \right|. \tag{7}$$

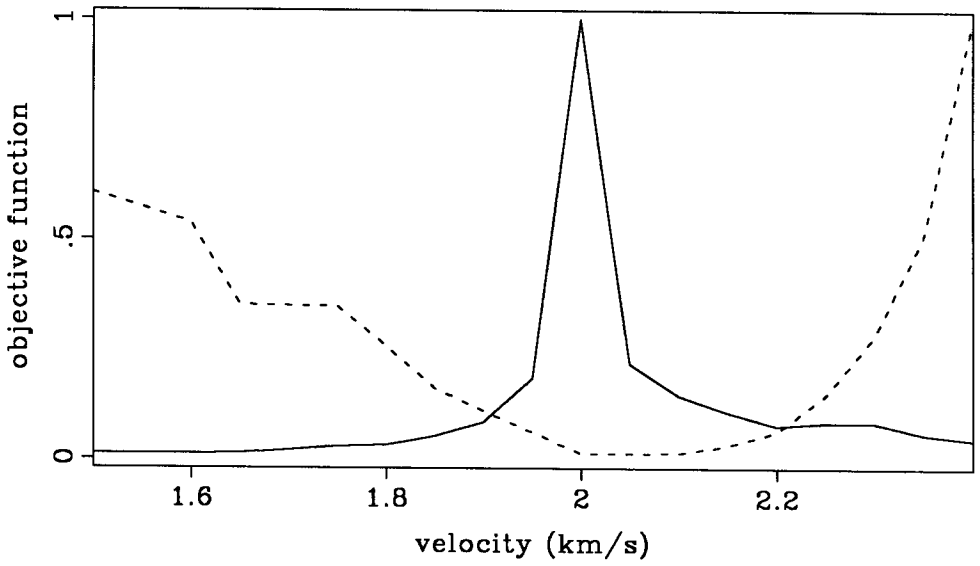

FIG. 7. Power of the stacked trace (solid line) and curvature (dotted line) as a function of migration velocity for a flat reflector model with velocity of 2.0 km/s.

The problem with this norm is that, if one event has some upward curvature and another event has the same curvature, but downwards, the $\mathbf{F}_1$-norm will be zero. The optimization routine will think it has found the minimum (uncurved events), and will stop. Again, an obvious solution is to take the sum of the $\mathbf{F}_1$ and $\mathbf{F}_2$-norms,

$$\mathbf{F}(v(x,z)) \;=\; \mathbf{F}_1(v(x,z)) \;+\; \mathbf{F}_2(v(x,z)). \tag{8}$$

Now, for the two-event example mentioned above, this norm is larger than zero when one event is curved upwards and the other downwards. When both events are curved either upwards or downwards, the $\mathbf{F}$-norm is even larger, a desirable feature because the velocity model is obviously more incorrect in this case.

Parametrization

Instead of sampling the velocity at each depth point (for a 1-D velocity model) or at each grid point (for a 2-D model) and inverting for many unknowns, we can reduce the number of unknowns considerably by describing the velocity function in terms of basisfunctions in a limited number of layers (1-D) or cells (2-D):

$$v(x,z) \;=\; \sum_{i=1}^{NE}\sum_{j=1}^{NG} p_{ij}\; e_i(x)\; g_j(z), \tag{9}$$

where NE is the number of basisfunctions in the lateral direction, and NG the number of basisfunctions in the depth direction, $e_i(x)$ is the i-th lateral basisfunction, $g_j(z)$ is the j-th depth basisfunction. The unknowns are the parameters p_{ij}.

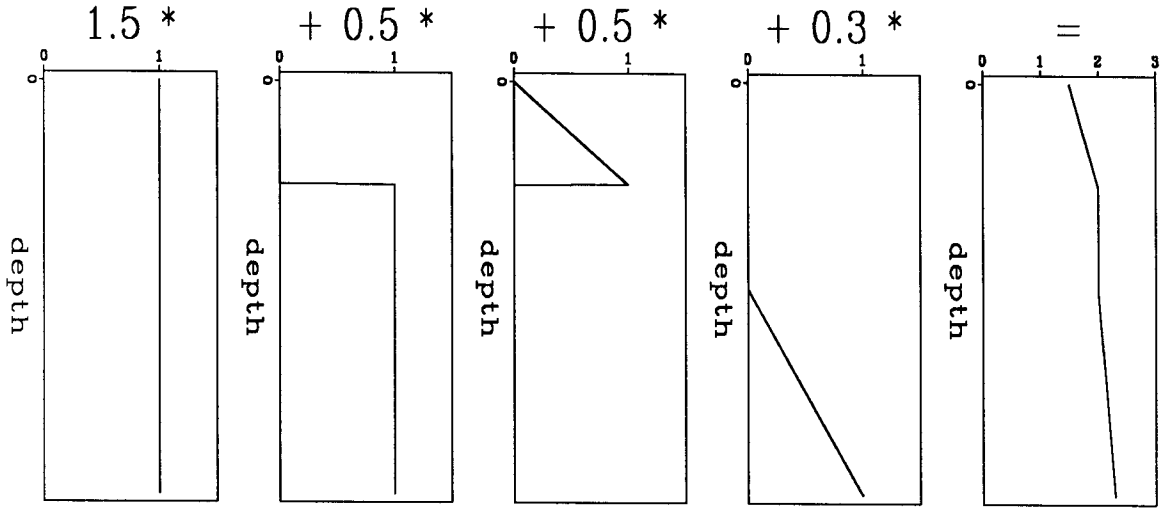

FIG. 8. 1-D parametrization: the velocity function on the right is the sum of the weighted basisfunctions on the left. The weights are shown above the functions. Given the set of basisfunctions, the velocity function is uniquely defined by 4 parameters: 1.5, 0.5, 0.5, and 0.3.

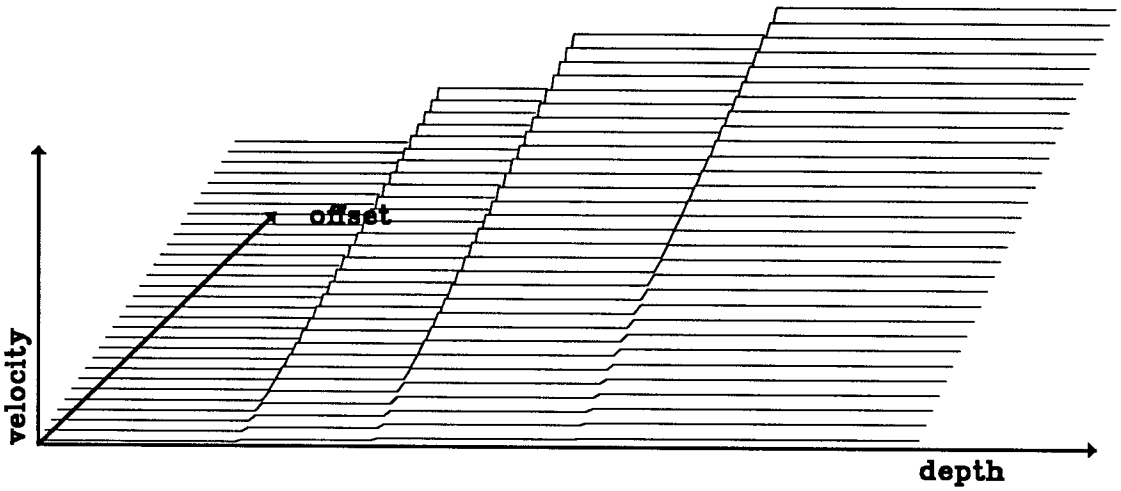

FIG. 9. 2-D parametrization: constant basisfunctions in the layers and a linear basisfunction superimposed in the lateral direction. There are 5 parameters here: 4 parameters in the depth direction and 1, the slope, in the offset direction.

Some simple basisfunctions are:

$$\text{constant}: \quad e(x) = \begin{cases} 1 & \text{for } x_i < x < x_{i+1}, \\ 0 & \text{elsewhere;} \end{cases}$$

$$\text{linear}: \quad e(x) = \begin{cases} \dfrac{x - x_i}{x_{i+1} - x_i} & \text{for } x_i < x < x_{i+1}, \\ 0 & \text{elsewhere;} \end{cases} \tag{10}$$

where x_i and x_{i+1} are the boundaries of layer i. One layer can have several basisfunctions to build up a complicated velocity function. Also, basisfunctions may *overlap*: e.g., one constant basisfunction can extend across the whole model to represent the "dc-component" of the model, whereas several others in smaller layers define the fine behavior of the velocity. To illustrate the concept, Figure 8 shows how one 1-D velocity function is constructed from several basisfunctions. In the 2-D case, basisfunctions are superimposed in the lateral direction (Figure 9).

Velocity is often desired to be continuous across layers. Continuity can be added as a constraint in the optimization. An alternative is to choose higher order basisfunctions, such as cubic or B-splines (Guiziou et al (1987)). Splines have the additional advantage that layer boundaries are not "rigid": the velocity function need not necessarily go through sampled points at the boundaries.

There are some good reasons for parametrizing the velocity into basisfunctions:

1. Cost: by reducing the number of parameters from thousands to dozens, the cost of optimization reduces accordingly.

2. Smoothness: velocity functions are generally smooth, unlike for example static corrections (one geophone can be placed in a ditch, whereas its neighboring geophones might be put on a flat meadow). By using smooth basisfunctions, this behavior is explicitly taken into account.

3. Resolution: velocity information comes from the arrival times of the reflected waves of a limited number of reflectors; the fine behavior of the velocity is unresolved between those reflectors. By choosing the number of layers to be about the same as the number of reflectors, we have a useful set of parameters; the null space of the inversion problem is reduced to zero.

A disadvantage of this parametrization is of course that the inverted velocity function is limited by the *a priori* chosen layer or cell boundaries and basisfunctions. However, this is the same limitation that can be found in optimization methods where the velocity function is *a posteriori* constrained by adding a model-norm (of the model and/or its derivative) to the objective function.

Nonlinear optimization

The expression for the parametrized velocity function (equation (9)) can be substituted

in equation (8):

$$\mathbf{F}(v(x,z)) \;=\; \mathbf{F}\Big(\sum_{i=1}^{NE} \sum_{j=1}^{NG} p_{ij}\; e_i(x)\; g_j(z) \Big) \;=\; \mathbf{F}(\bar{p}), \tag{11}$$

where $\bar{p}$ is a $NE \cdot NG$-dimensional vector, whose elements are the unknowns p_{ij} that control the basisfunctions.

The goal of the optimization is to find the vector $\bar{p}_m$ that minimizes $\mathbf{F}$. The optimization method I used is a non-derivative method (Powell (1964)). It iteratively builds up conjugate directions, by using only function evaluations. Each iteration consists of n searches in n parameter directions. The details of the method are explained elsewhere in this report (Van Trier (1987)). The optimization is fully nonlinear: it does not use linear approximations to evaluate the objective function or its derivatives.

Because there is no gradient information, albeit linear, gradient methods can only be used if the gradient is calculated by finite differences. For a n-dimensional objective function, this means that the function has to be evaluated n times to calculate one gradient direction. A conjugate-gradient method then becomes about expensive as Powell's method. Also, the finite difference interval for the gradient calculation is hard to determine, and generally has to be adjusted during the optimization (Van Trier (1987)). Nevertheless, when the number of parameters gets large, conjugate-gradient methods are likely to be more efficient. I have not yet tested large parameter problems to confirm this.

RESULTS

The main purpose of this section is to illustrate that the method works on field data; no attempt is made to interpret the inverted velocity models.

To compare the method with conventional velocity analysis, the optimization is first done with NMO instead of migration. A comparison is also made between optimization with curvature as objective function and optimization that maximizes the power of the stacked trace. The curvature-objective function is the $\mathbf{F}_2$-norm of equation (6), I have not yet tested the $\mathbf{F}$-norm (equation (8)) or combinations of the curvature- and stack-norm. The velocity models that are considered here are 1-D models; the velocity is not allowed to vary laterally.

NMO

In traditional velocity analysis a CMP gather is move-out corrected with a constant trial velocity. The moved-out gather is summed over offset, resulting in a stacked trace. The process is repeated for many velocities, and a contour map of the power of the stacked traces is then used to pick velocities. Such a contour map for the gather in Figure 3 is shown in Figure 10. Although the gather is a shot gather, the geometry is assumed to be flat, so that the move-out in the CMP gather is the same as in the CSP gather. The flat geometry assumption is confirmed by tests with the Overlay program (Claerbout (1986)). These tests show that the reflection events can be fitted by hyperbolas computed for horizontal geometry. The apparent decrease in velocity in the deeper region is probably caused by the left-over of a pegleg not removed by the deconvolution.

The optimization is first done with the power of the stacked trace as objective function,

to compare with traditional velocity analysis. The velocity is parametrized in linear basisfunctions with continuity across the boundaries. The boundaries are chosen at 0.8, 1.5, and 2.1 s.

After one iteration, i.e., after finding a minimum in each of the parameter directions, the moved out reflectors are more or less flat. Additional iterations hardly improve the objective function, and the optimization is stopped after 3 iterations. The result of the optimization is shown in Figure 12. Figure 11 shows how the velocity changes during the iterations. The inverted velocity profiles match well the output of conventional velocity analysis (Figure 10).

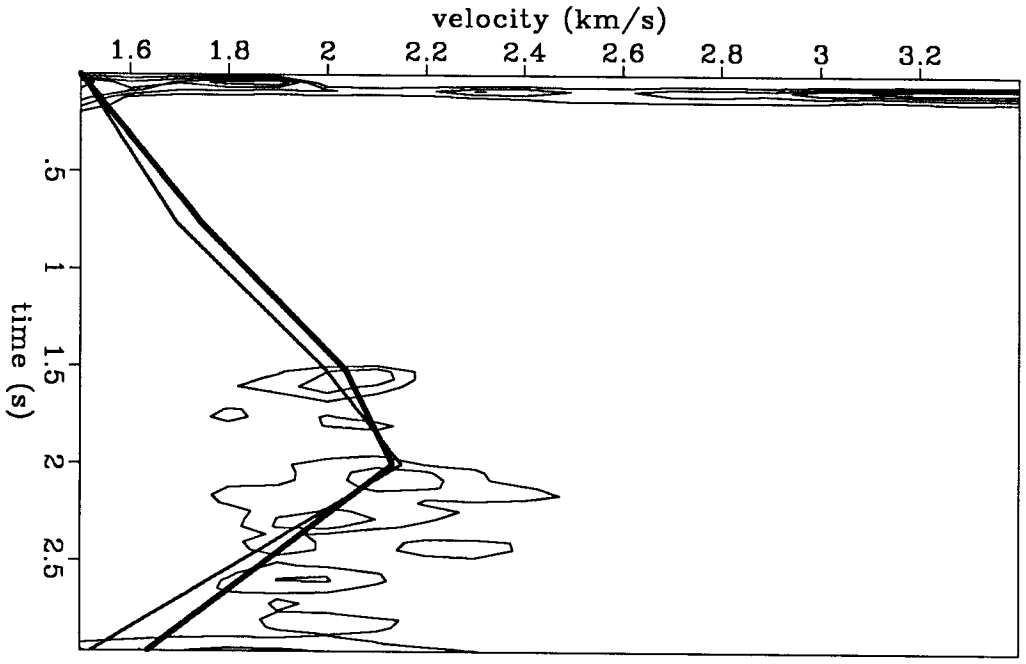

FIG. 10. Velocity panel for gather 27: some of the inverted velocity profiles of Figure 11 are plotted on top of the contour map: the thin line is the result after 1 iteration, the thick one the result after 3 iterations.

For NMO, using the curvature as objective function is not necessarily better than using the power of the stacked trace, because NMO is limited to horizontal reflectors anyway. The curvature-norm can be more robust than the stack-norm, though, as discussed in the previous section, and our main concern here is to compare the two norms.

The "curvature optimization" also converges after only one iteration, and is stopped after three iterations. The result is shown in Figure 13, the inverted velocity profile in Figure 14. The NMOed dataset in Figure 13 looks identical to the one in Figure 12, except for the reflector at 2 s, which is still slightly curved at the wide offsets. The curvature is not estimated at the widest offsets because of end effects (see section on curvature estimation), so residual NMO at these offsets will not be detected.

The difference in the inverted velocity models is more apparent (Figure 14). This brings up some interesting questions on how well resolved the velocity is, and how the choice of basisfunctions effects the inversion. I will not discuss these issues here, but I would certainly

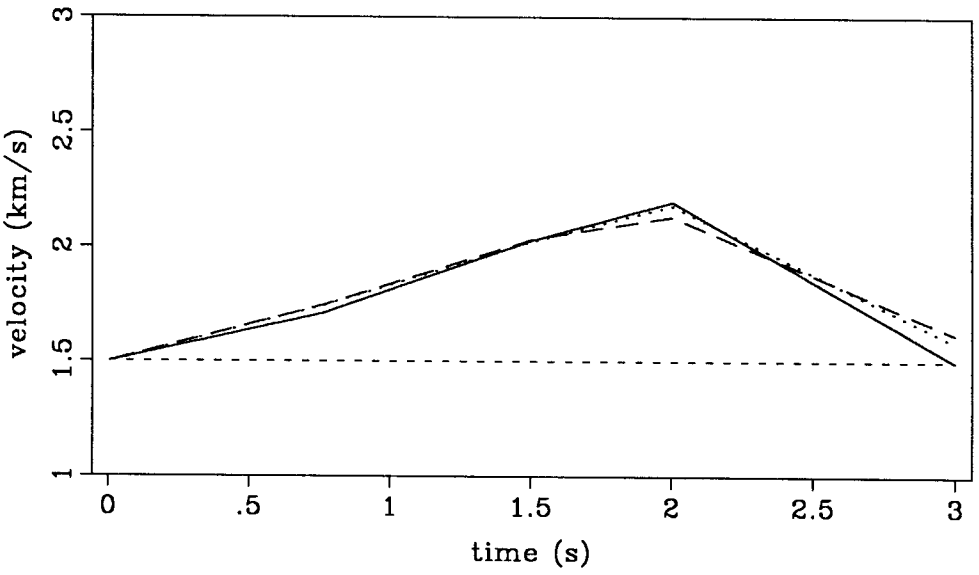

FIG. 11. NMO-velocities after each iteration. - - -: starting velocity; ———: after 1 iteration; · · ·: after 2 iterations; — — —: after 3 iterations.

like to address them in the future.

Shot profile migration

The curvature optimization is now done with shot profile migration instead of NMO. The migration is a fast Kirchhoff-integral depth migration (Etgen (1987)). Unlike NMO and time migration, layer boundaries in the velocity model are hard to determine in advance for depth migration. Therefore, only "rough" basisfunctions are used: a constant and a linear one across the whole model, a linear one in the top half of the model, and a linear one in the bottom half. The velocity is constrained to be continuous across boundaries. The optimization converges again after one iteration. The migrated shot profile is shown in Figure 15, and the result of each parameter search in the first iteration in Figure 16.

The optimization has flattened the deeper reflectors (the curved tails at the wide offsets are migration artifacts, which are not picked up by the curvature estimation because of their low amplitude), but the shallow events seem to be overmigrated. The chosen configuration of basisfunctions is too limited to build up a velocity function that can flatten all reflectors. Also, migration focusing narrows the shallow events, and curvature is harder to determine. A different set of basisfunctions needs to be chosen, and more shot profiles need to be included in the optimization. The next section deals with the latter problem.

BEYOND SHOT PROFILES

The optimization really has to be extended to include the information of an entire seismic survey, consisting of multiple shot profiles. One approach is to optimize each shot profile separately, adjusting one 2-D velocity model that extends over the whole survey. The information in each profile contributes to a different part of the model.

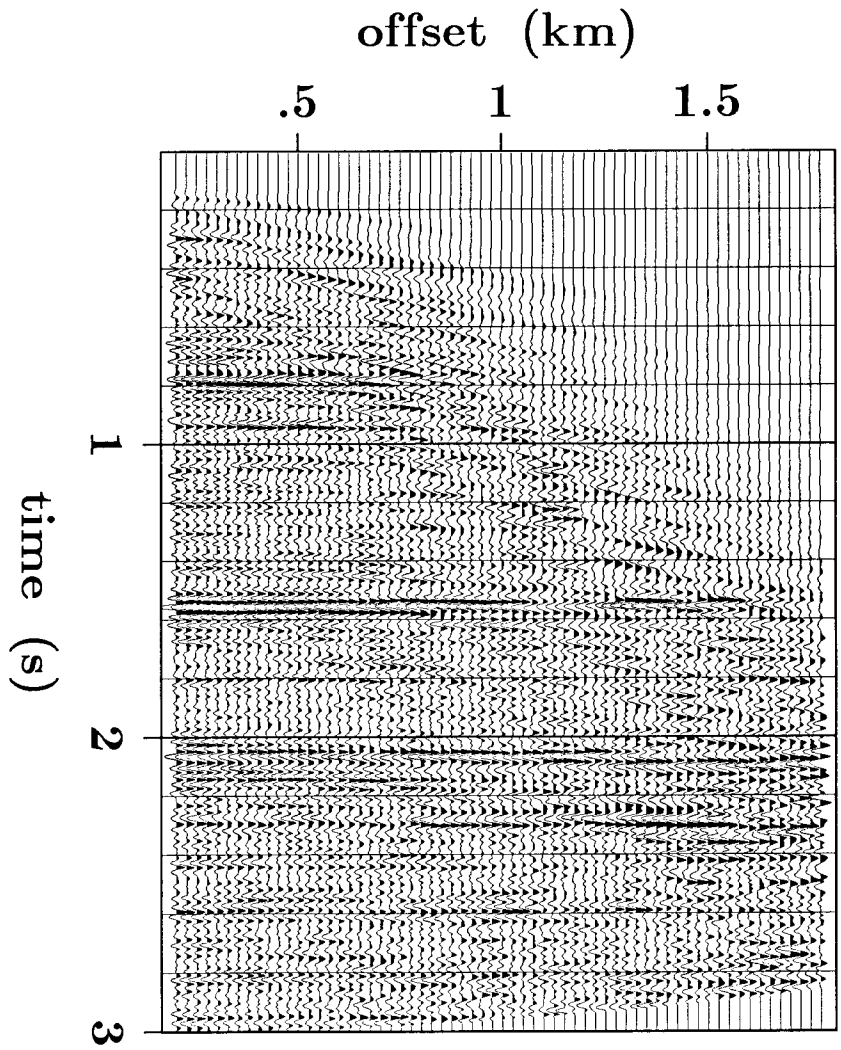

FIG. 12. NMOed data: result of the optimization after 3 iterations. The optimization uses the power of the stacked trace as objective function.

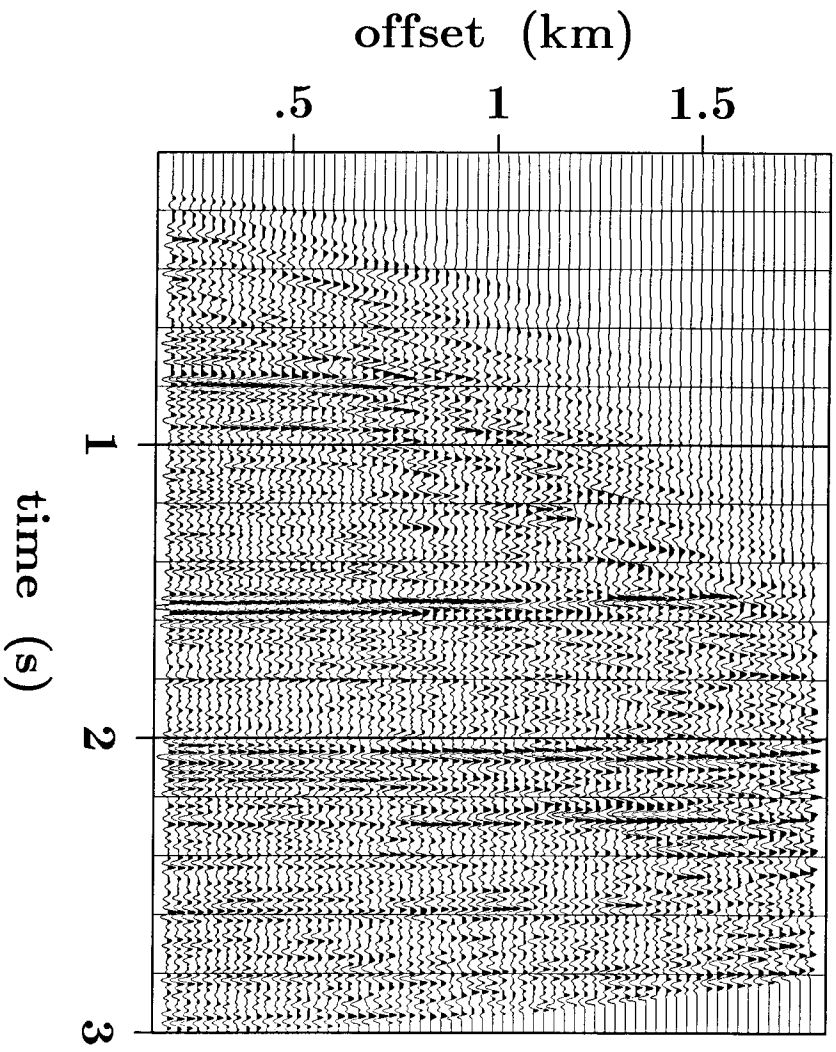

FIG. 13. NMOed data: result of the optimization after 3 iterations. The optimization uses the curvature as objective function.

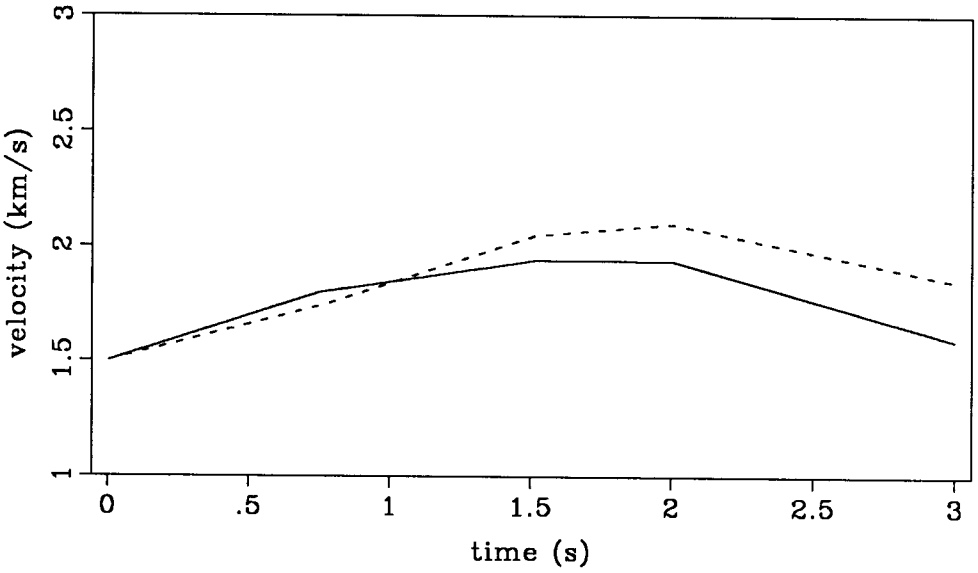

FIG. 14. The inverted velocity profiles of stack-power (dashed) and curvature optimization (solid).

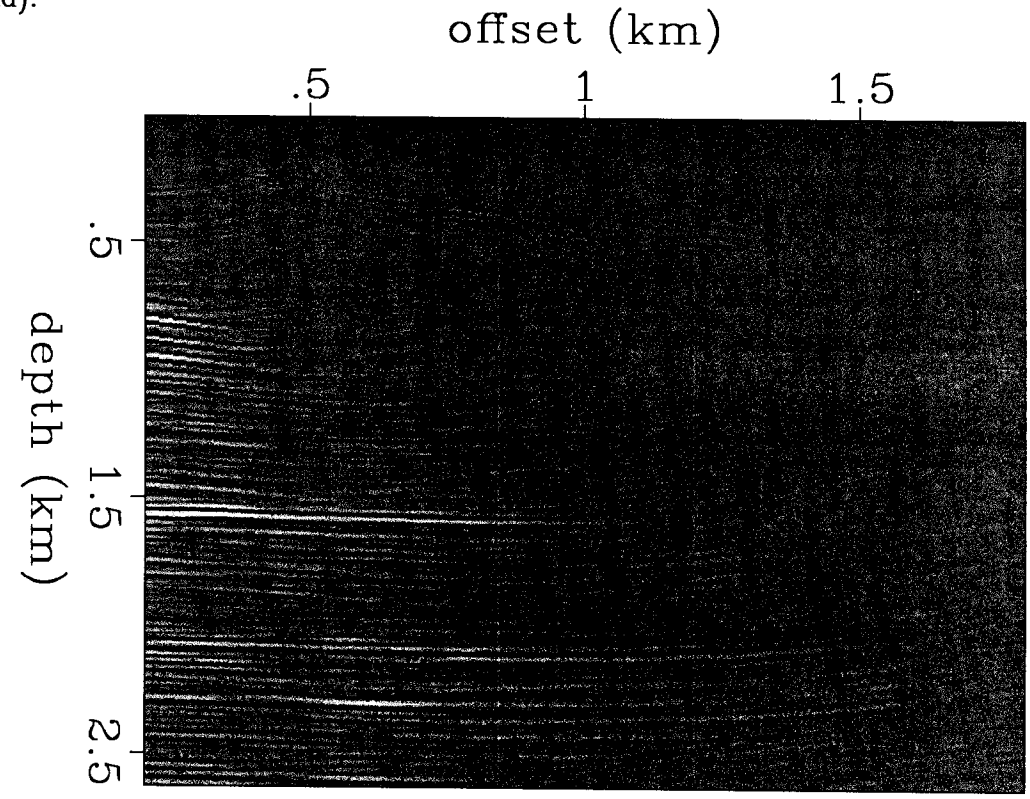

FIG. 15. Migrated data, after one iteration in the curvature optimization. The plot is "dithered" because the continuity of the reflectors can be better seen on a dithered plot than on a wiggle-trace plot.

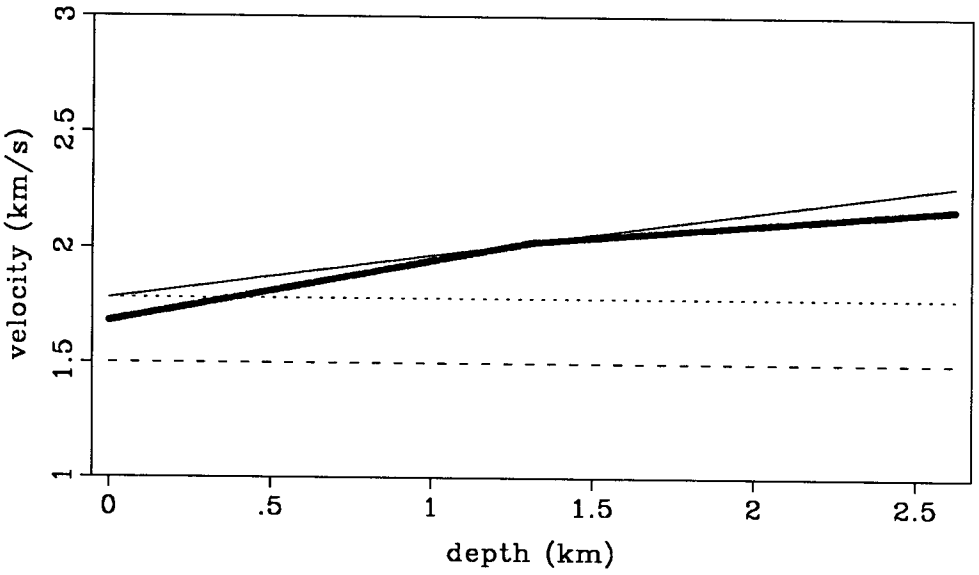

FIG. 16. Migration velocity after one iteration (thick line). The thin lines show how the velocity function is constructed from basisfunctions during the iteration. Each thin line shows the velocity function after the minimum in one parameter direction has been found. (For example, the first parameter determines the weight of the constant basisfunction across the whole model, the dc-component. The constant velocity model that fits the data best is shown by the dotted line.) - - -: starting velocity; · · ·: after first parameter search; ———: after second parameter search. The result of the third parameter search can not be seen: it is covered by the thick and the solid line.

Treating each shot profile separately has some advantages: on parallel computers several shot profiles can be optimized at the same time, and each shot profile represents a single physical experiment. There are also some disadvantages. First, the optimization of one profile can destroy the results of previous optimizations. Another problem is that the curvature of events in a single profile can be caused either by incorrect migration or by curvature in the reflectors themselves. This ambiguity can be resolved by looking at the curvature in common receiver gathers (Al-Yahya and Muir (1984)): they observe that "only when imaged correctly will an event be horizontally aligned in a migrated common receiver gather."

Therefore, the optimization can be started by optimizing single shot profiles with a roughly parametrized velocity model, determining the gross structure. After that the common receiver gathers are optimized with a more finely parametrized model, to find detailed structure. The second step properly treats curved reflectors.

OTHER APPLICATIONS OF PHASE CONTOURING

Tomography

Since the phase-contouring method picks traveltime curves, it can be readily applied in tomography. The described velocity-analysis method itself is tomographic in the sense that it uses a feature of traveltime curves, the curvature, to determine velocity.

Local dip

Just as second derivatives of reflection events $t(h)$ determine local curvature (equation (5)), first derivatives give the local dip, p:

$$p(t(h),h) \;=\; \frac{\partial}{\partial h}\, t(h) \;=\; \frac{t(h+\Delta h) \;-\; t(h)}{\Delta h} \tag{12}$$

The dip can be determined very locally in this way, much more local than with local slant stacks, where the dip is found by summing over line segments that extend over a range of traces. The local-dip section of profile 27 is shown in Figure 17. To display local dip in many events, the *un*deconvolved profile, which still contains all the multiples, is used. The figure clearly shows how the dip increases as the offset increases. The resolution of the dip estimation is hard to see on the plot (a color display is useful here), but the high resolution is illustrated by the sensitivity of the dip estimation to residual statics at the offset of 1.1 km (the same statics can also be seen in Figure 3).

Statics

As mentioned above, local statics cause anomalies in the global curvature or dip of an event. Therefore, an optimization that minimizes curvature or dip can be used to find static corrections. It is not necessary that the curvature or dip has to be zero in the corrected shot record; only anomalous curvature or dip has to be detected against some background level. This means that the shot profile need not be NMO-corrected (or migrated): residual (e.g., non-hyperbolic) NMO, or residual migration will not effect the estimation of the static corrections.

CONCLUSIONS

The described method combines migration with interval velocity estimation. The concept is not new (e.g., see Al-Yahya and Muir (1984) and Fowler (1985)), but the parametrization of the velocity model into a small number of basisfunctions enables the use of an optimization that does not depend on a linearized relation between model and data. Curvature seems to be a good objective function for this nonlinear optimization scheme, because its sensitivity to migration velocity extends over a wide range of velocities.

Some improvements can be made in the curvature estimation method: the linear interpolation in the contour algorithm can be replaced by a more accurate one, a higher-order second-derivative operator can be used to calculate curvature, and, instead of smoothing by moving average filtering, smoothing can be done with a tridiagonal smoother.

As for the parametrization, the predefined boundaries of the basisfunctions pose a problem when depth migration is used in the optimization. A solution is to add the boundaries

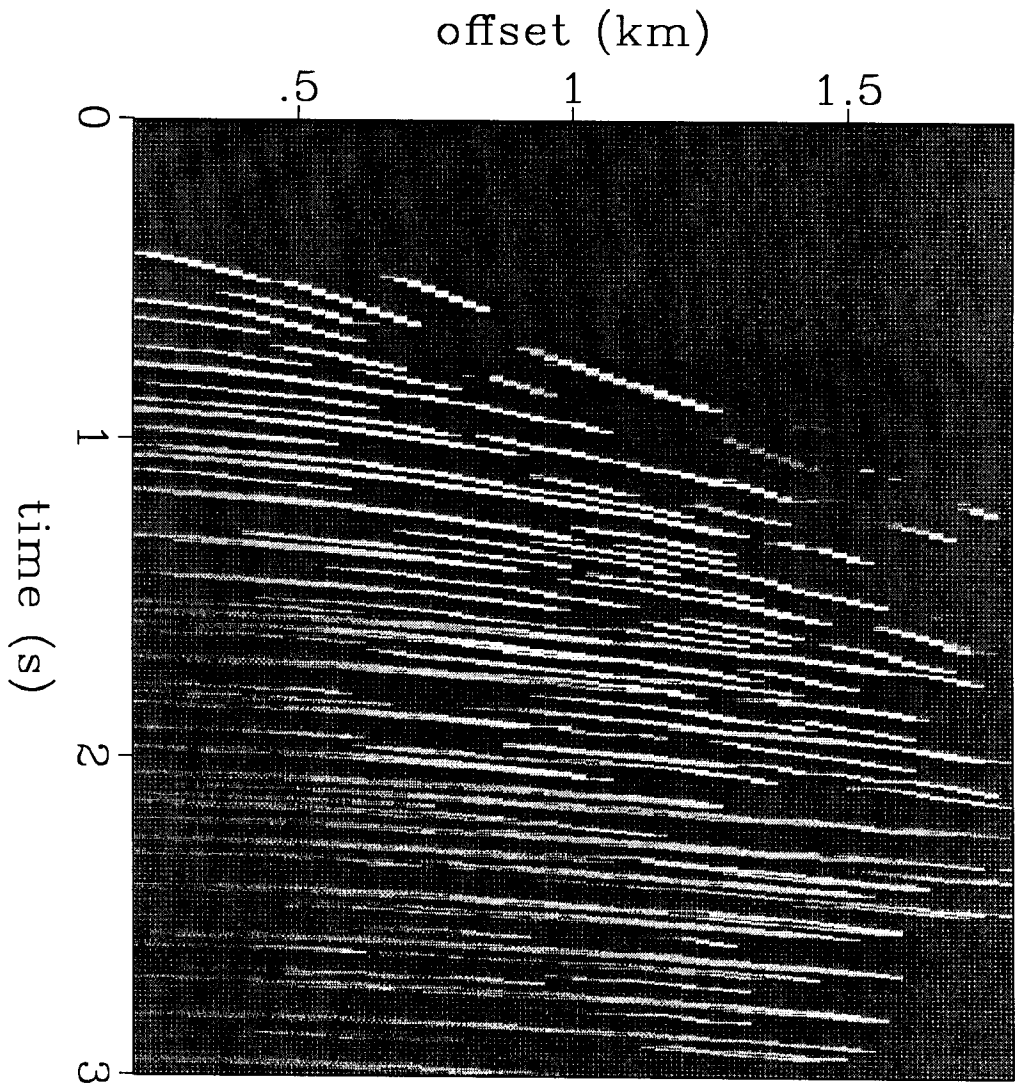

FIG. 17. Local-dip section of the undeconvolved 27-profile: grey represents low dip, white high dip.

as parameters to the optimization problem, or to use cubic- or B-splines as basisfunctions. In the optimization, the different objective functions have to be tested.

Finally, the optimization has to be extended to include more than one shot profile.

ACKNOWLEDGMENTS

I would like to thank John Etgen for the use of his prestack migration program and for many helpful discussions on migration. Paul Fowler suggested to me to look at instantaneous phase. Thanks also to Marta Woodward for many useful suggestions. Finally, I would like to thank Shell Internationale Petroleum Maatschappij B.V. for financial contributions.

REFERENCES

Al-Yahya, K., and Muir, F., 1984, Velocity analysis using prestack migration: SEP-38, 105–112.

Claerbout, J., 1986, Hyperbola overlay program development: a personal engagement: SEP-50, 269–278.

Cottafava, G., and Le Moli, G., 1969, Automatic contour map: Communications of the ACM, **12**, 386–391.

Etgen, J., 1987, Fast prestack depth migration using a Kirchhoff integral method: this report.

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

Guiziou, J. L., Dezard, Y., and Martayan, G., 1987, 2D Tomography for VSP: this report.

Nolet, G., Van Trier, J., Huisman, R., 1986, A formalism for nonlinear inversion of seismic surface waves: Geophysical Research Letters, **13**, 26–29.

Powell, M. J. D., 1964, An efficient method for finding the minimum of a function of several variables without calculating derivatives, Computer Journal **7**, 155–162.

Taner, M. T., Koehler, F., and Sheriff, R. E., 1979, Complex seismic trace analysis: Geophysics, **44**, 1041–1063.

Van Trier, J., and Rocca, F., 1986, Autofocusing of migrated data: SEP-50, 113–127.

Van Trier, J., 1987, Some nonlinear optimization methods: this report.

Yilmaz, O., and Cumro, D., 1983, Worldwide Assortment of Field Seismic Records, released by Western Geophysical Company of America, Houston.