Received
DEC 1 4 199

DOE/ER/14079--21

DE93 004454

Dip-Moveout Processing for Depth-Variable Velocity

Craig T. Artley

— Master of Science Thesis —

Center for Wave Phenomena
Colorado School of Mines
Golden, Colorado 80401
(303) 273-3557

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Abstract

Dip-moveout correction (DMO) has become commonplace in the seismic processing flow. The goal of DMO processing is to transform the NMO-corrected data to zero-offset, so that the application of zero-offset (poststack) migration is equivalent to full prestack migration of the recorded data. Nearly all DMO implementations assume that the seismic velocity is constant. Usually, this is an acceptable trade-off because of the tremendous cost savings of DMO and poststack migration versus prestack migration. Where the velocity changes rapidly with depth, however, this constant velocity theory can yield inadequate results.

For many areas, such as the Gulf Coast, a velocity function that varies with depth is a reasonable approximation to the true velocity field. Using ray tracing, I find the raypaths from the source and receiver to the reflection point with the given recording time. The time along the corresponding zero-offset ray gives the DMO correction. The relationships between the three rays are expressed by a system of nonlinear equations. By simultaneously solving the equations via Newton-Raphson iteration, I determine the mapping that transforms nonzero-offset data to zero-offset.

Unlike previous schemes that approximately handle vertical velocity variation, this method makes no assumptions about the offset, dip, or hyperbolic moveout. A characteristic of this exact method is that even for a linear increase of velocity with depth, the DMO operator is multi-valued. Rather than the familiar elliptical impulse response of constant-velocity DMO, the DMO operator now has multiple branches.

Tests using synthetic data as well as recorded seismic data demonstrate the effectiveness of this variable-velocity DMO. These tests show that this method accurately handles vertical velocity variation, while using constant-velocity DMO can lead to significant errors. Comparing this technique to a formulation which approximately handles velocity variation, however, suggests that the improved accuracy of the exact technique may not be justified due to uncertainty in the velocity model and increased cost.

While improved accuracy alone may not justify the use of this method in 2-D, its flexibility may in other cases. Straightforward changes could be made to handle 3-D DMO, DMO for mode-converted waves, DMO in anisotropic media, or amplitude effects, for example.

Introduction

Since moveout velocity is dip-dependent (Levin, 1971), stacking emphasizes reflections whose moveout velocity is near that used for normal moveout correction (NMO). Conversely, reflections with other slopes, and hence different moveout velocities, are attenuated. Therefore, NMO acts as a dip filter applied to the stacked data that is biased against dips with moveout velocities different from the NMO velocity.

Dip-moveout processing (DMO) attempts to increase the dip bandwidth of stacked data by making stacking velocities independent of dip (Deregowski, 1982, 1986). DMO processing enables common-midpoint stacking to simultaneously treat all dips accurately. The result is a stacked section that is equivalent to a zero-offset section, as long as certain assumptions are satisfied.

Hale's DMO by Fourier transform algorithm (1983, 1984) and most other DMO implementations assume that velocity is constant. In practice, this is found to be an acceptable limitation because of the practical advantages in computational speed of DMO and zero-offset migration over prestack migration. This is particularly true in three dimensions, where the constant-velocity DMO operator remains two-dimensional, while the prestack migration operator is three-dimensional (Hale, 1988).

When the velocity changes rapidly with depth, however, Black et al. (1985) and Hale (1988) showed that constant-velocity DMO can be worse than not applying any DMO. Furthermore, I presented numerical examples of the errors resulting from applying conventional NMO and DMO correction to data from a vertical reflector in a medium where the velocity varies only with depth (Artley, 1990). I considered both constant-velocity DMO and the first-order depth-variable velocity DMO correction published in Hale's thesis (1983), and found that while the correction is an improvement, it still leads to significant degradation in the resolution of the stack.

Several authors (Perkins and French, 1990; Dietrich and Cohen, 1992) have calculated the shape of the 3-D DMO operator for media where the velocity increases linearly with depth. While the constant-velocity 3-D DMO operator remains two-dimensional, the variable-velocity operator has support in three dimensions. The operator is saddle-shaped: concave-up in the dip-direction and concave-down in the strike direction. These issues highlight the need for an accurate method of determining the DMO mapping for media where the velocity varies with depth. I use the term $v(z)$ *DMO* for such a process, because the velocity v is a function of depth z.

Hale and Artley (1991) tuned Hale's (1983) original $v(z)$ correction to better handle vertical velocity variation by squeezing the constant-velocity DMO operator. Witte (1991) and Meinardus and Schleicher (1991) computed $v(z)$ DMO corrections for arbitrary velocity fields by assuming hyperbolic moveout and using well-known expressions for the dip-dependent moveout velocity. While these approximations may be sufficiently accurate in practice, there is still a desire to determine the exact $v(z)$ DMO correction.

Figure 1 compares CMP gathers processed with conventional constant-velocity DMO and the $v(z)$ DMO method described in later chapters. The model itself is described in detail in the following section on Applications, but at the present it is sufficient to note that the velocity increases linearly with depth and that it contains several horizontal reflectors as well as reflectors dipping at angles from 30 to 90 degrees. The deepest reflection in each gather is from the vertical reflector.

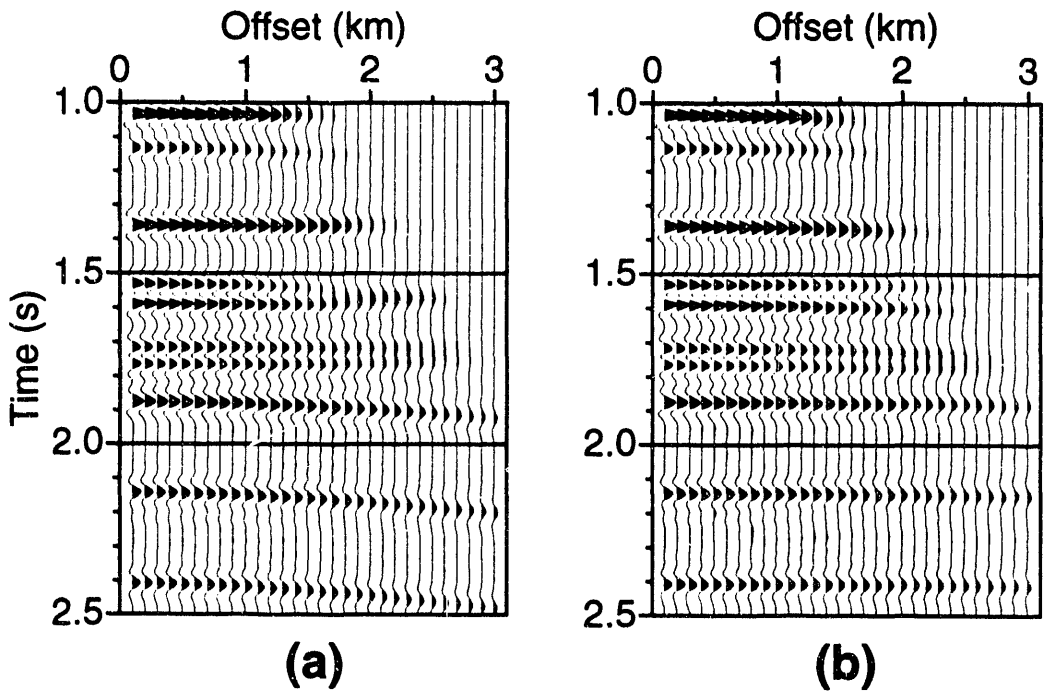

FIG. 1. Synthetic CMP gathers from a medium where velocity increases linearly with depth. (a) After constant-velocity DMO. (b) After $v(z)$ DMO correction. The last reflection in each gather corresponds to a dip of 90 degrees. Constant-velocity DMO has failed to align energy from steeply dipping reflectors.

Energy from the horizontal and gently dipping reflectors has been flattened across both gathers. Note, however, that the deeper reflections, from the steeply dipping reflectors, still show residual moveout across the gather processed with constant-velocity DMO. Obviously, since the reflections are not aligned, stacking these data attenuates the energy from the steeply dipping reflectors. In contrast, $v(z)$ DMO has successfully removed the moveout from all of the reflections, including that from the vertical reflector. Stacking these data would yield a section with a wide bandwidth of reflection slopes that is equivalent (in a traveltime sense) to the corresponding zero-offset section.

The method described here makes no assumptions about the velocity function, other than its lateral invariance. Unlike others, this formulation does not make small-offset or small-dip approximations. Thus it is accurate for long offsets and high dips and arbitrary velocity layering, where the moveout may be significantly non-

hyperbolic. Furthermore, this work has already been extended to three dimensions by Godfrey (1992), though it is discussed here only in the context of 2-D DMO. Indeed, the flexibility of this method could enable the processing of mode-converted data, or handling the effects of anisotropy.

The following section shows more examples of 2-D $v(z)$ DMO processing applied to both synthetic and field data and discusses some characteristics of the $v(z)$ DMO operator. The sections on DMO by Dip Decompositon and on Computing the $v(z)$ DMO Mapping describe in detail my approach to computing and applying the $v(z)$ DMO correction. The Conclusion contains some notes on possible future applications.

Applications

To demonstrate the effectiveness of this $v(z)$ DMO method, I tested it using both synthetic data and marine seismic data from the Gulf of Mexico. These tests show that $v(z)$ DMO is a significant improvement over constant-velocity DMO. When $v(z)$ DMO is compared to an approximate technique for handling velocity variation, the differences are still visible, but may be less significant.

Synthetic data

With noiseless synthetic data, the effect of DMO correction can be clearly judged by viewing CMP gathers. NMO alone can only align energy associated with a single moveout velocity at a given time. To simultaneously align reflections having different moveout velocities, DMO must be applied. After NMO and DMO correction, all reflection events in a CMP gather, regardless of the dip of the corresponding reflector, should be free from residual moveout.

Constant velocity gradient

For the first test, I generated synthetic data for a model in which the medium velocity increases linearly with depth. The velocity is given by $v(z) = v_0 + az$ where v_0, the velocity at the surface, is 1.5 km/s and a, the velocity gradient, is $0.8\,\mathrm{s}^{-1}$. The model contains five reflectors, and each reflector is composed of a dipping segment and a horizontal segment, as shown in Figure 2. The dipping segments range in dip from 30 degrees to 90 degrees in 15 degree increments.

Figure 3 shows a CMP gather from the dataset after various phases of processing. The gathers are taken from the location marked with the triangle in Figure 2. Each gather contains ten reflection events; one from each of the dipping and horizontal segments in the model. The last reflection in each gather is from the vertical reflector.

Figure 3.a shows the uncorrected synthetic data. Note that the reflection from the vertical reflector exhibits no moveout, as expected in a $v(z)$ medium. After NMO correction with the rms velocity function (Figure 3.b), the reflections from the horizontal segments are flat, while the reflections from the dipping segments show increasing amounts of over-correction with dip. Constant-velocity DMO (Figure 3.c) has successfully aligned the energy for the gently dipping segments, but the reflections from the 60, 75, and 90 degree reflectors now appear to be under-corrected. Constant-velocity DMO has restored too much moveout to the steeply dipping reflectors. In contrast, the exact $v(z)$ DMO process has properly flattened all reflections in the gather, including the energy from the vertical segment.

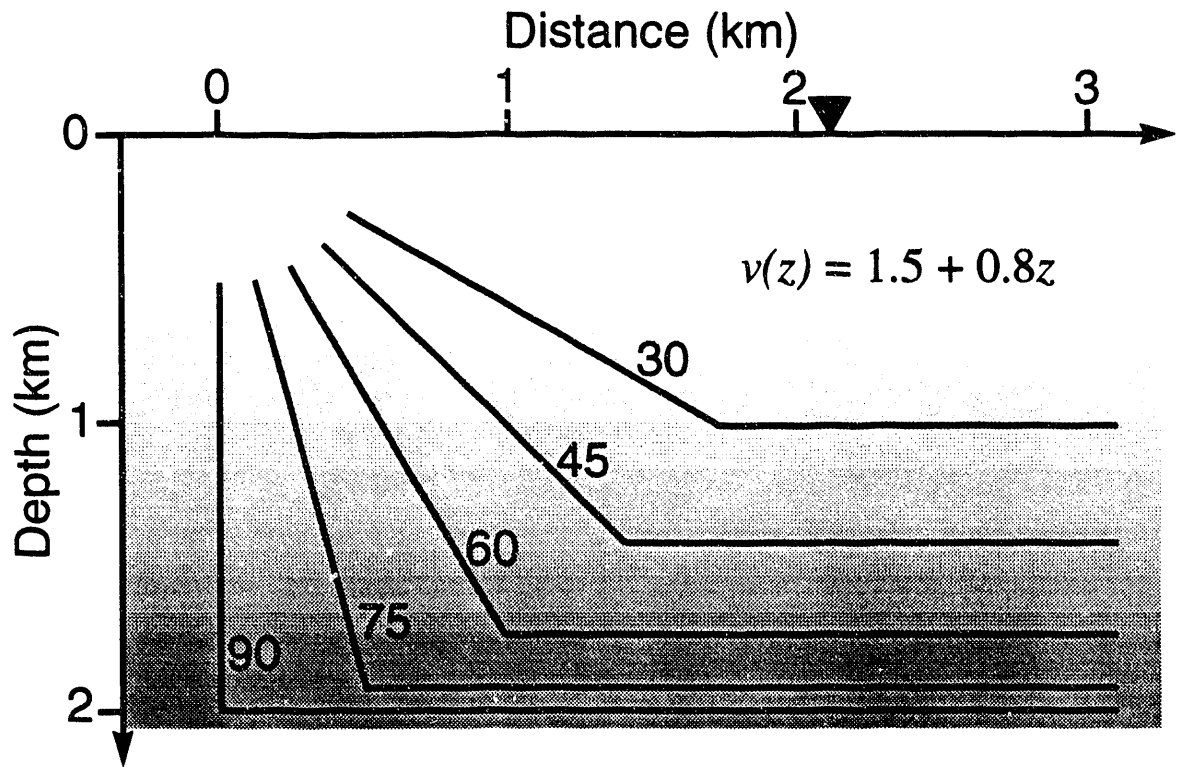

FIG. 2. The model used to generate the synthetic seismic data. The CMP gathers shown in the following examples are taken from the surface position marked with the triangle. The velocity of the medium varies according to $v(z) = 1.5 + 0.8\,z$ km/s, where the depth z is measured in kilometers.

Figure 4 shows the kinematic DMO impulse response for time of 2.4 s and source-receiver offset of 3.0 km. These values correspond to the time of the vertical reflector and the farthest offset in the data. For comparison, the impulse responses of constant-velocity DMO and Hale's first-order $v(z)$ correction are also shown in the figure.

The most striking differences between the exact and approximate curves are the cusps and multiple branches of the exact operator. The reflector dip angle at the cusp is 105 degrees, while the dip at the lower end of the second branch is 115 degrees. For these parameters, dips beyond 115 degrees correspond to reflection points above the surface of the earth. The details of the cusp and tail are dependent on the velocity function, time, and offset. In some configurations the cusp occurs before 90 degrees. I shall discuss the physical meaning of the cusp and second branch in more detail after noting some other significant differences in the curves.

While not as visually dramatic, the most important difference in practice between the impulse response curves is their width. The chief effect of vertical velocity variation is the narrowing of the operator. The constant-velocity curve deviates from the exact at relatively low dips, resulting in the significant errors shown in Figure 3. Hale's original $v(z)$ correction (1983) lies much closer to the exact curve.

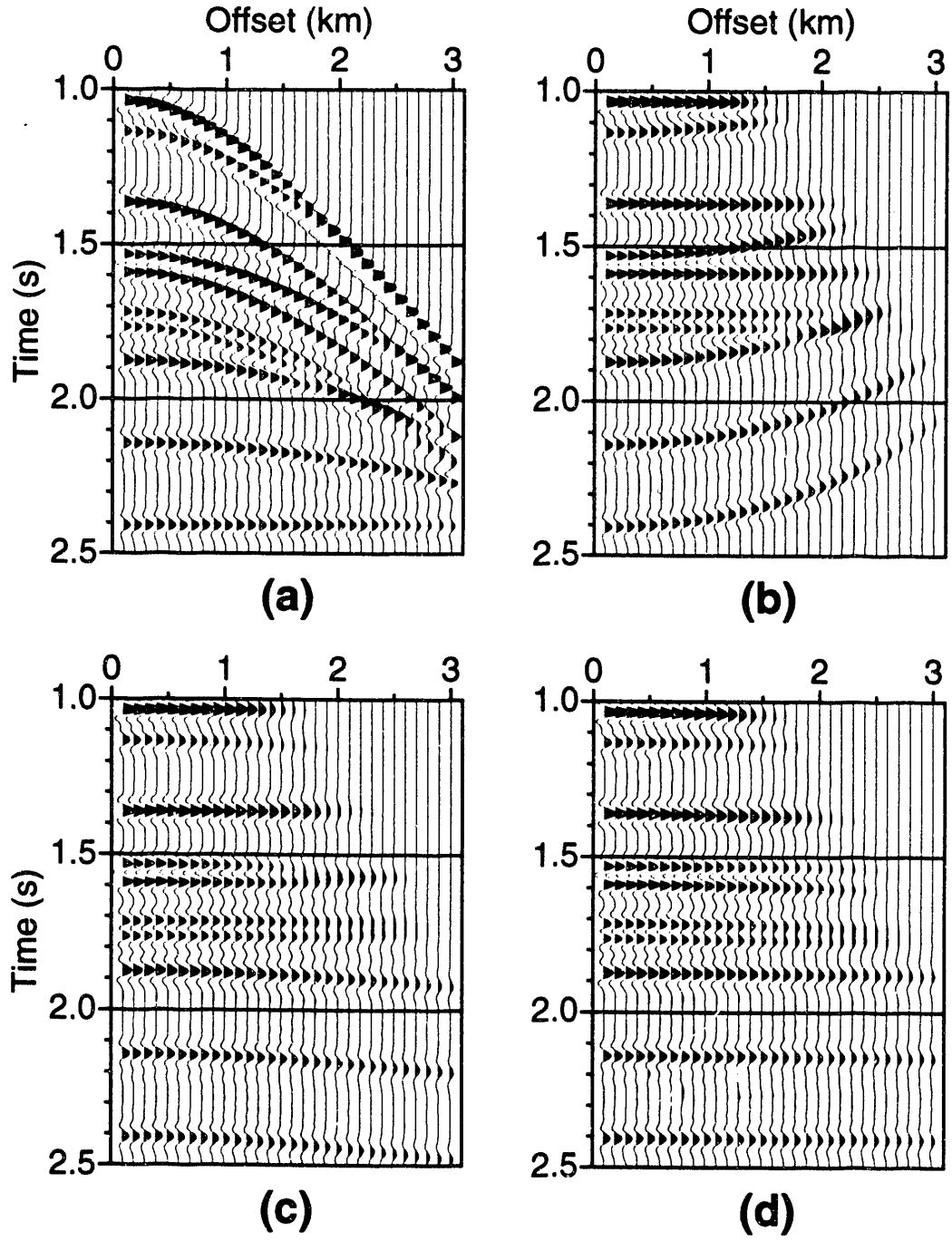

FIG. 3. (a) Uncorrected CMP gather from the model shown in Figure 2. (b) After NMO correction. (c) After constant-velocity DMO. (d) After $v(z)$ DMO correction. The five reflections in (b) that are over-corrected by NMO correspond to dipping reflectors. The last reflection in each gather corresponds to a dip of 90 degrees.

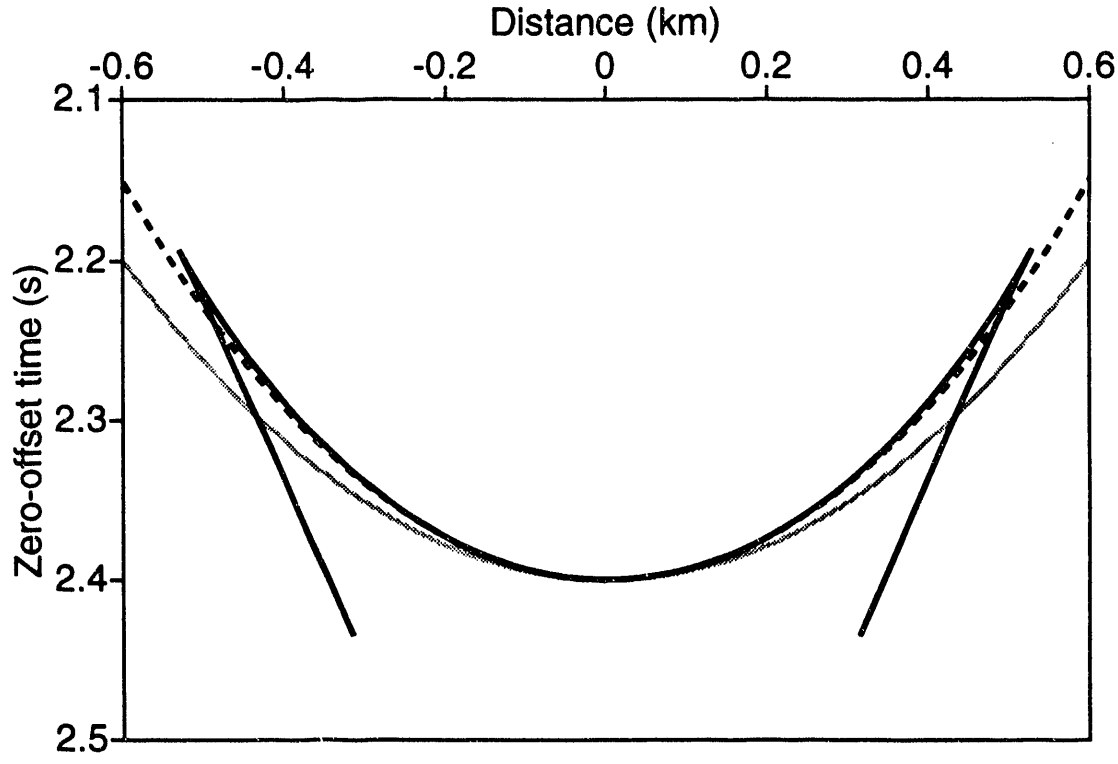

FIG. 4. Zero-offset time t_0 as a function of distance x, for NMO time $t_n = 2.4$ s for the farthest offset (3 km) in the CMP gathers of Figure 3. Times are plotted for three DMO methods: (gray) constant-velocity DMO, (dotted) Hale's original $v(z)$ approximation, and (black) the exact DMO method described here. The cusps of the exact curve correspond to reflector dips of ± 105 degrees, while the tips of the tails correspond to dips of ± 115 degrees.

Hale and Artley (1991) proposed a simple squeezing of the constant velocity DMO operator to approximately handle the effects of velocity variation. The squeezed DMO operator is not shown, but it can be made to lie arbitrarily close to the primary branch of the exact curve by adjusting the squeeze factor S.

Figure 5 compares squeezed DMO ($S = 0.6$) and the exact $v(z)$ DMO for the same synthetic data shown in Figure 3. The squeezed DMO result (Figure 5.a) shows much better alignment of the reflections across the gather than the constant-velocity DMO result (Figure 3.c). Squeezed DMO, however, still results in a slight under-correction of the 75 and 90 degree reflection events at the farthest offsets when compared with exact $v(z)$ DMO.

Decreasing the squeeze factor to $S = 0.4$ flattens even the vertical reflection, though it begins to distort other reflections. The value $S = 0.6$ has been found empirically to work well for most velocity functions (Hale and Artley, 1991), but there is no formula for estimating the optimal value. Hence while it can be adjusted to flatten a given reflection, in practice there is no convenient method of choosing S.

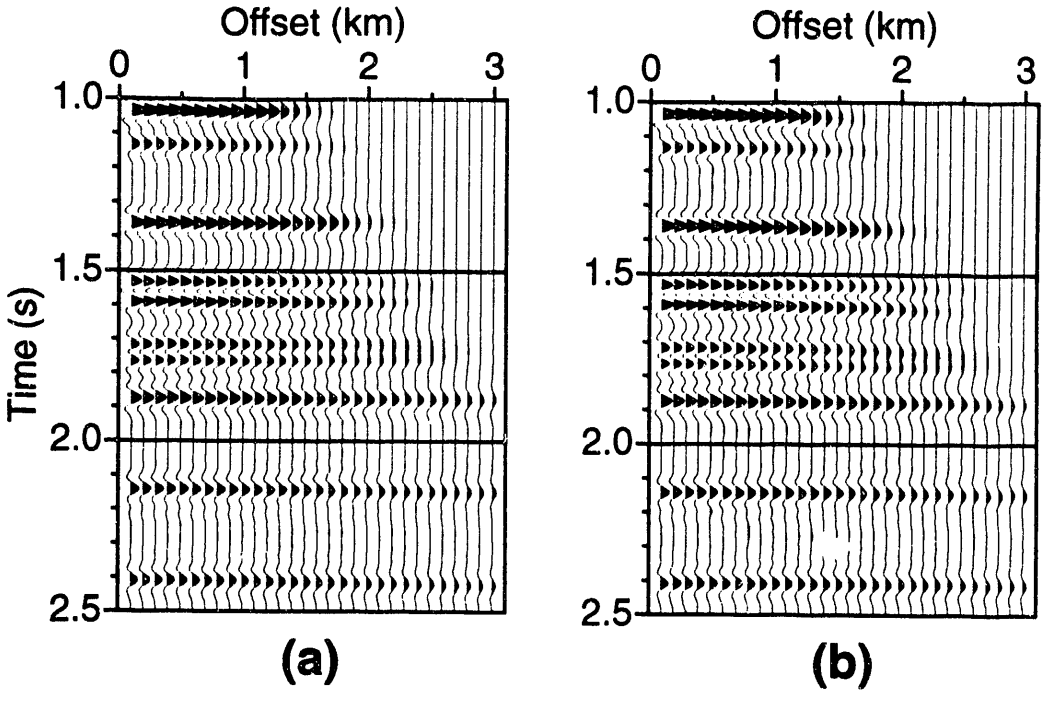

FIG. 5. (a) Comparison of squeezed DMO with (b) exact $v(z)$ DMO correction. Squeezed DMO has performed essentially as well as $v(z)$ DMO.

Also, it is important to note that the squeezed operator will never show the multiple branches of the exact operator. $v(z)$ DMO, on the other hand, will properly handle all reflections, given the correct velocity model.

Returning to the discussion of the cusp and second branch, the duplication of zero-offset times at a given midpoint shift is easier to understand through Figure 6. Figure 6.a shows the raypaths associated with the primary branch of the operator. The ray diagram shows raypaths from the source (not shown) to the receiver such that the traveltime is constant. The tips of the rays hence trace out the constant-traveltime surface. This surface is the prestack migration impulse response for this experiment. For the case of constant velocity, the rays are straight and the reflecting surface is an ellipse. For the case shown, however, the velocity increases linearly with depth according to $v(z) = 1.5 + 0.3\,z\,\mathrm{km/s}$, the source-receiver offset is 3.0 km, and the constant-traveltime surface corresponds to an NMO time of 3.0 s.

Note that the angle formed by the source and receiver rays at each reflection point is bisected by a third ray. This third ray is the *equivalent zero-offset ray*, and the time along this ray and its emergence point give the DMO mapping. The zero-offset ray is normal to the reflector at each reflection point, and each point corresponds to a different reflector dip.

Shown to the right of the rays is a plot of the emergence point of the zero-offset ray versus the time along that ray. This curve forms one-half of the DMO impulse

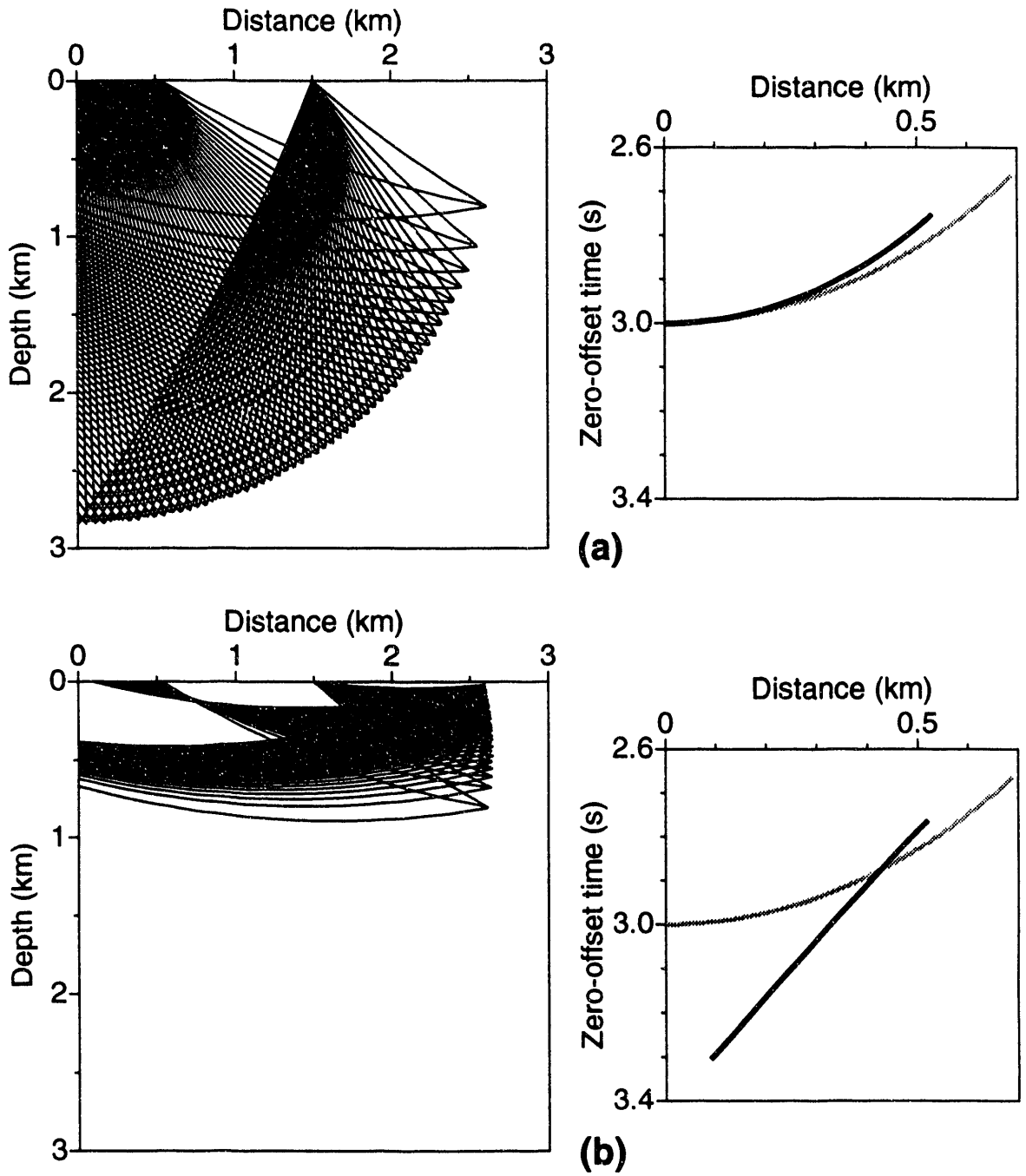

FIG. 6. Physical interpretation of the multi-valued DMO operator. (a) Rays corresponding to the primary branch of the operator. (b) Rays corresponding to the second branch. Dips beyond 80 degrees map to this branch. The constant-velocity DMO operator is shown in gray.

response curve. (Since the curve is symmetric about the source-receiver midpoint, the other half is simply the reflection of this curve.) As the dip of the reflector increases, the emergence point moves to the right and the time along the zero-offset ray decreases. These rays correspond to the primary branch of the DMO operator.

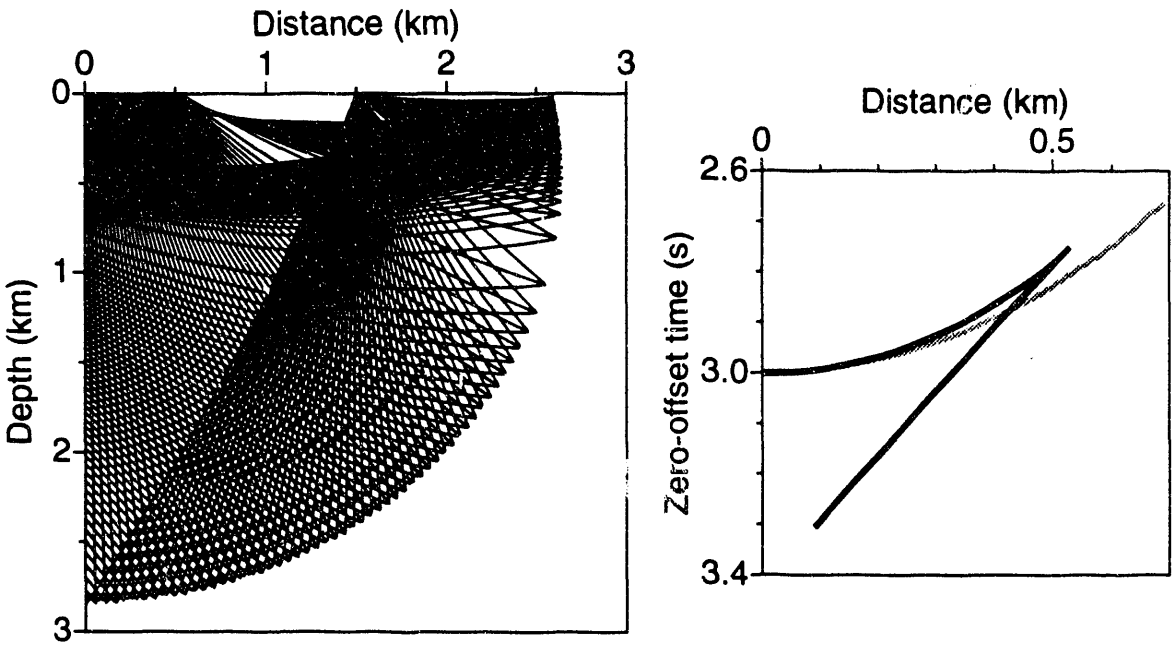

FIG. 7. The entire reflecting surface and DMO operator are formed by superimposing Figures 6.a and 6.b.

At a dip of approximately 80 degrees, however, the behavior reverses. As the dip continues to increase, the zero-offset emergence point moves *back toward the midpoint* and the time along the ray now *increases.* These rays are shown in Figure 6.b and correspond to the second, backward branch of the operator. Since this branch corresponds to steeply dipping and overturned reflectors, properly handling it in DMO processing may be important when imaging turning waves.

Note that for a range of zero-offset locations there are *two* possible zero-offset rays, with different zero-offset times. Superimposing both sets of rays (Figure 7) builds up the entire constant-traveltime surface, and gives the complete DMO operator. The operator is double-valued at all points for which there are two possible zero-offset rays.

As mentioned earlier, the details of the operator shape and the location of the cusp are highly dependent on the offset and velocity gradient. Figure 8 shows the effect of changing the velocity gradient. Here the offset is 3.0 km, and the NMO time is 2.5 s. The velocity at the surface is fixed at 1.5 km/s while the gradient is varied from $0.0\,\mathrm{s}^{-1}$ to $0.8\,\mathrm{s}^{-1}$. As the gradient is increased, the DMO operator becomes narrower and develops the backward branches. Also, the elliptically-shaped portion becomes more compressed in time.

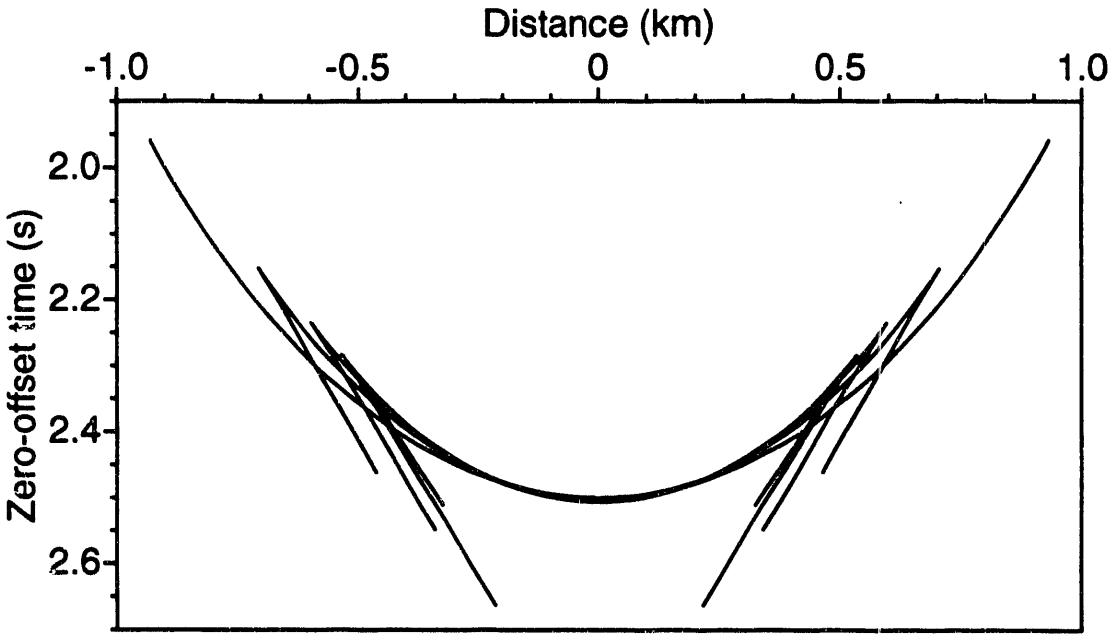

FIG. 8. $v(z)$ DMO operators for NMO time of 2.5 s and offset of 3.0 km. The velocity at the surface is 1.5 km/s and the gradient varies from 0.0 s^{-1} to 0.8 s^{-1}.

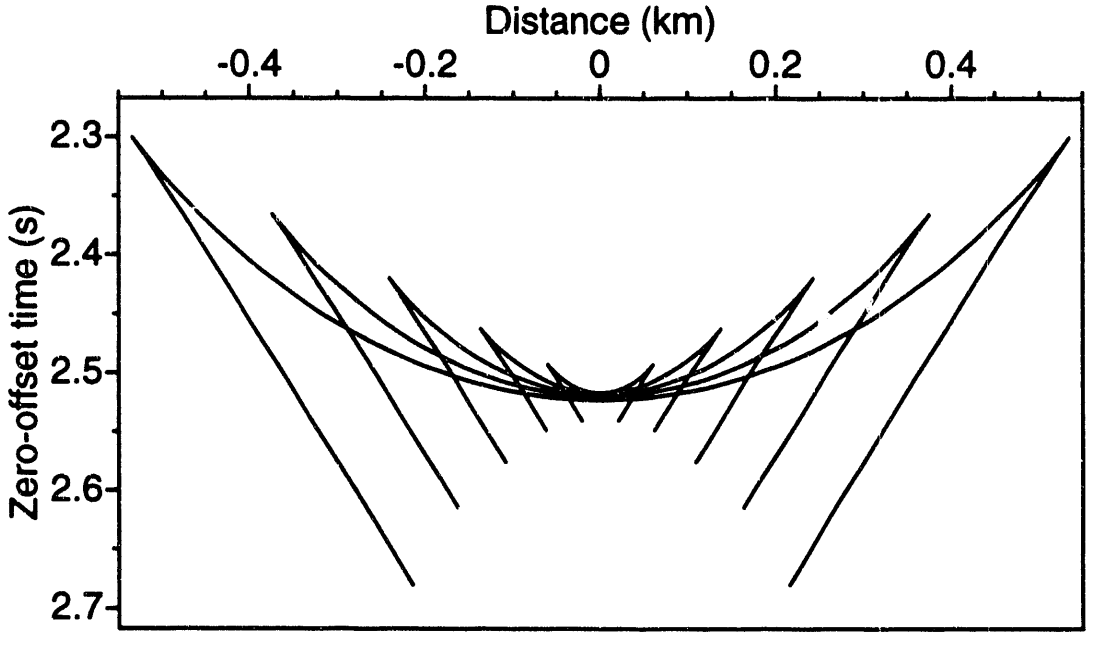

FIG. 9. $v(z)$ DMO operators for NMO time of 2.5 s and $v(z) = 1.5 + 0.6\,z$ km/s. The offset varies from 1.0 km to 3.0 km.

In Figure 9 the offset is varied from 1.0 km to 3.0 km. The velocity model is $v(z) = 1.5 + 0.6\,z$ km/s for all cases. As with constant-velocity DMO, the operator becomes narrower as the offset decreases. The backward branches also shrink with the offset. The operator is smaller because DMO has less work to perform at smaller offsets.

The final example for the constant-gradient case demonstrates the time-variant nature of DMO in a variable-velocity medium (Figure 10). Here DMO operators for several offsets are shown for several NMO times. As with constant-velocity DMO, the operators shrink at late times. So while the operator now exhibits duplications, it still behaves much like the constant-velocity DMO operator with respect to changes in offset and time.

In practice, the importance of the second branch of the DMO operator in a medium with a constant velocity gradient can perhaps be discounted. Dietrich and Cohen (1992) analytically calculated the shape and amplitude of the DMO operator for linear $v(z)$, and showed that most of the energy is concentrated along the primary branch of the curve. The amplitude along the lower branch is much smaller and can often be ignored. Also, the location of the cusp is dependent on the source-receiver offset, so energy from steeply dipping reflectors may appear on the primary branch at small offsets, but on the second branch at farther offsets. So while squeezed DMO may improperly position energy at the far offsets, stacking with the near offsets will still be a significant improvement over constant-velocity DMO.

Piecewise-constant velocity gradient

The next synthetic test uses the same five-reflector model as the previous test, but now with a layered velocity model. The velocity function is piecewise-linear with depth, as shown in Figure 11. The velocity is a constant 1.5 km/s in the first 0.3 km (0.4 s). In the second layer, velocity increases at 0.8 km/s per kilometer down to a depth of 1.2 km (1.38 s). Below this depth, the velocity gradient is $0.4\,\mathrm{s}^{-1}$. This velocity model represents a water layer over sediments. The velocity increases rapidly with depth in the shallow sediments, and then more slowly in the deeper sediments.

Figure 12 shows CMP gathers from this model after DMO processing. As in the constant-gradient example of Figure 3, constant-velocity DMO (Figure 12.a) results in under-correction at the farthest offsets, though the errors are smaller than before.

Conversely, squeezed DMO with $S = 0.6$ (Figure 12.c) results in slight *over*-correction of the 75 and 90 degree dips. Recall that in the constant-gradient example (Figure 3.c), the squeezed DMO result showed residual under-correction, as did constant-velocity DMO. Exact $v(z)$ DMO (Figures 12.b and 12.d) has properly aligned all reflections in the CMP gather, independent of dip.

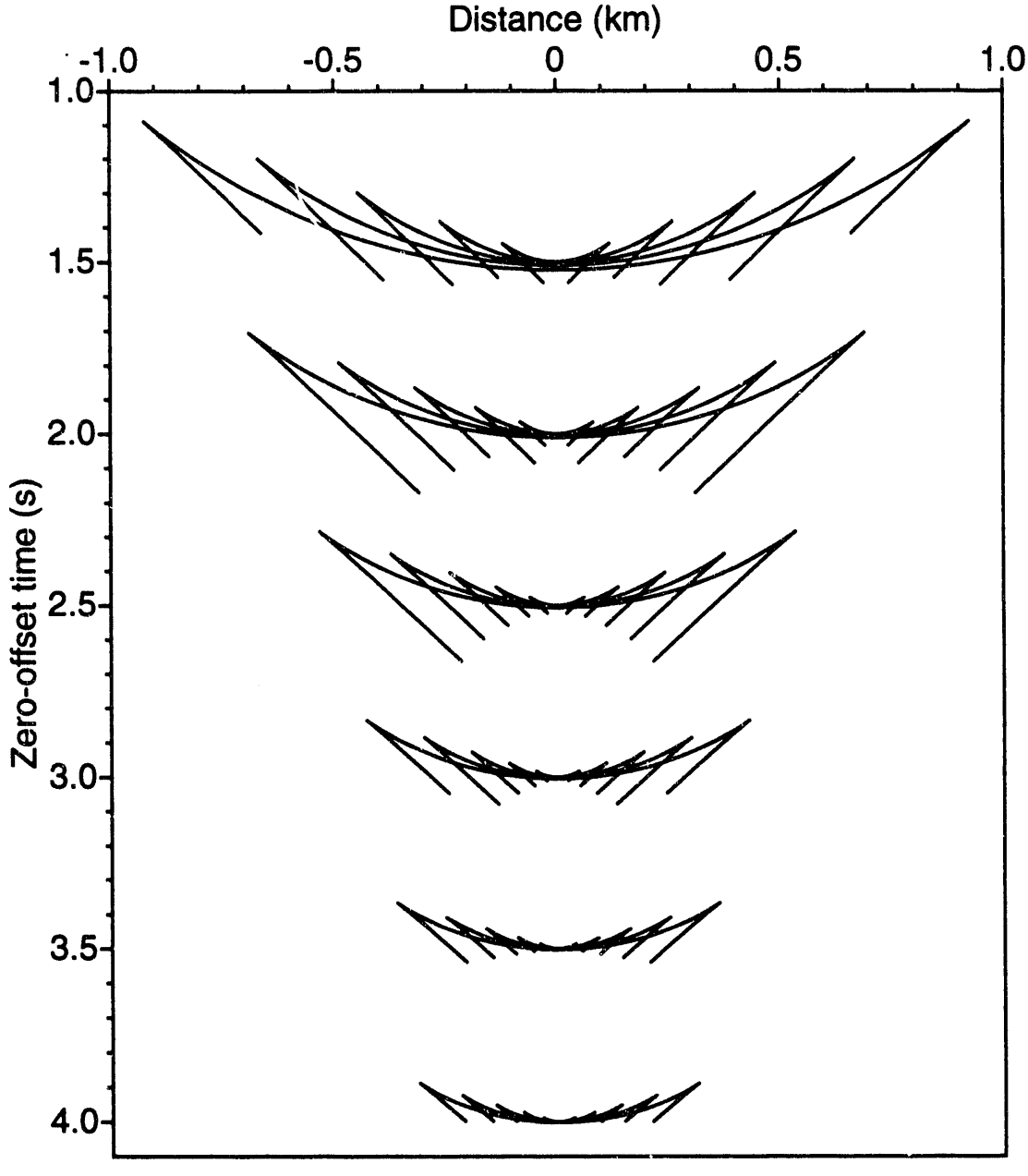

FIG. 10. Suite of $v(z)$ DMO operators for several NMO times and $v(z) = 1.5 + 0.6\,z$ km/s. The offset varies from 1.0 km to 3.0 km.

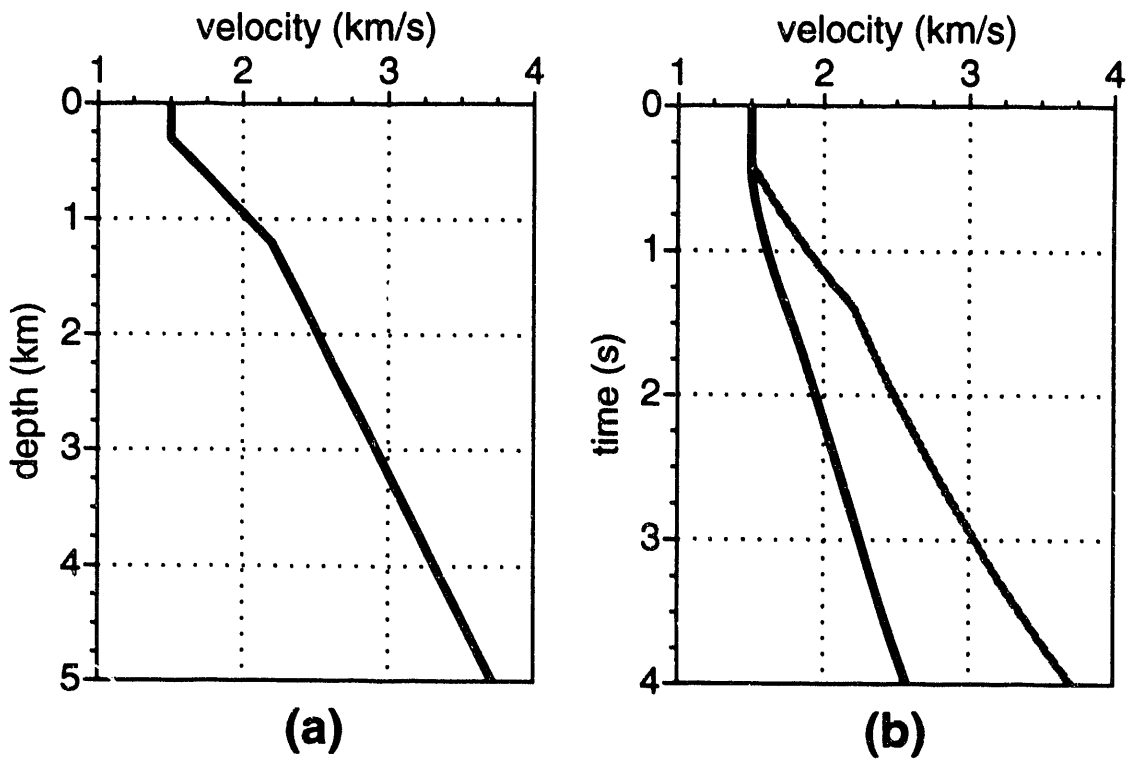

FIG. 11. Velocity function used for second synthetic data test. (a) Interval velocity versus depth. (b) Interval velocity (gray) and rms velocity (black) versus time.

The DMO operators for this velocity model show two branches, like those of the constant-gradient case. Due to their similarity to the previous series of figures, they are not shown here.

The small errors observed in this case for constant-velocity DMO and squeezed DMO can be explained by the velocity gradients used in this model. In the previous example the gradient was a steep $0.8\,\mathrm{s}^{-1}$. Here the velocity in the first layer is constant, and it increases at a comparatively gentle $0.4\,\mathrm{s}^{-1}$ in the last layer. Only in the relatively thin second layer does the velocity increase at the higher gradient. Thus, on the whole, the velocity variation is less severe in this model and we might expect conventional DMO methods to perform better here.

Gulf of Mexico data

With actual exploration seismic data, the traces are typically contaminated with noise, making it difficult to determine the effectiveness of DMO by looking at CMP gathers. Instead, the relative merits of DMO processes are judged by viewing the stacked sections and computing constant-velocity stacks.

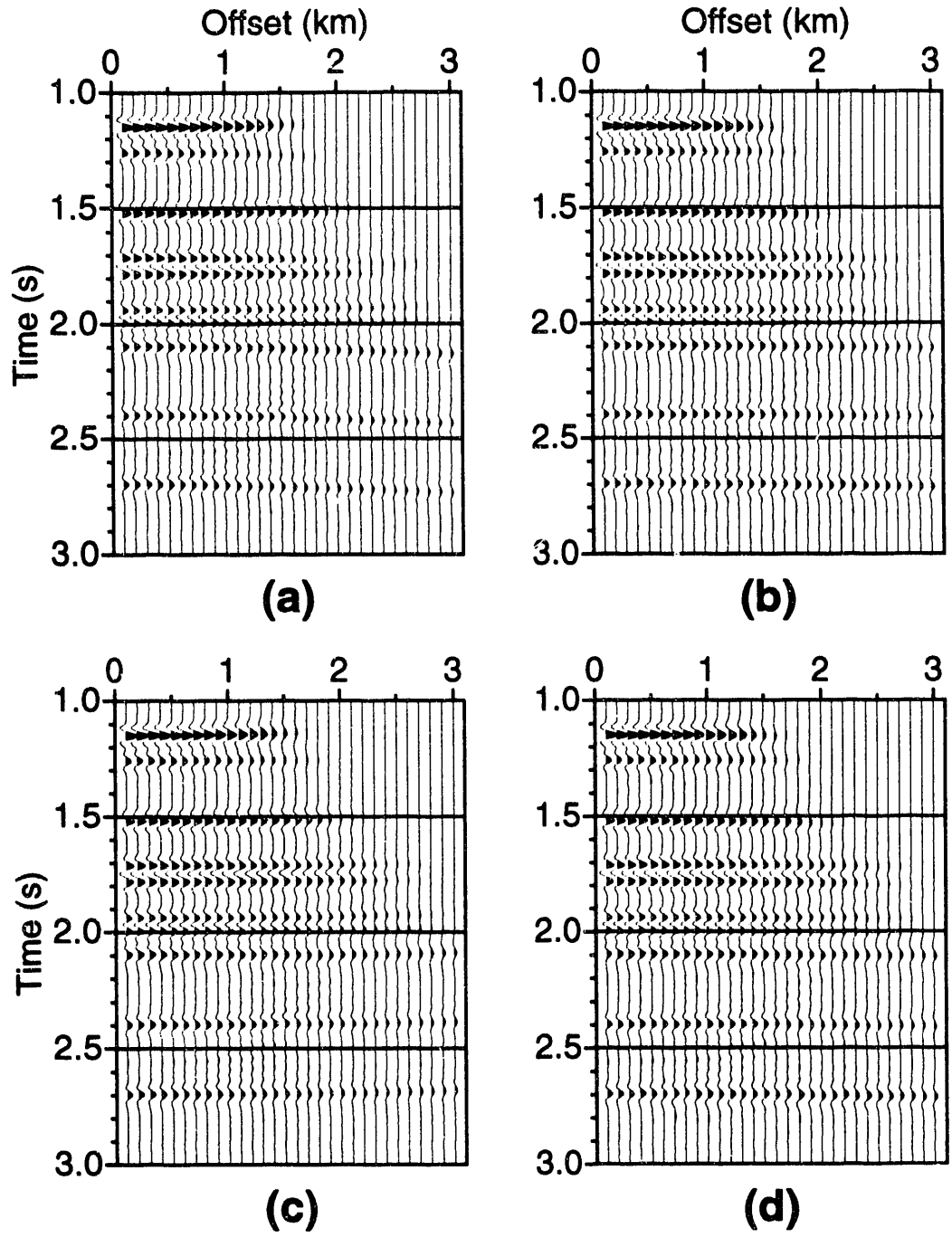

FIG. 12. Synthetic CMP gathers from a medium where the velocity gradient is piecewise-constant with depth (Figure 11). (a) After constant-velocity DMO. (c) After squeezed DMO. (b) and (d) After $v(z)$ DMO. Constant-velocity and squeezed DMO have under- and over-corrected the steep dips, respectively.

Figure 13 shows stacked sections of seismic data recorded in the Gulf of Mexico after constant-velocity and exact $v(z)$ DMO processing. The steeply sloping reflection is energy from the vertical flank of a salt dome.

The test of any DMO process is how well both dipping and horizontal reflectors are preserved in the stack. By adjusting the stacking velocities, any NMO-DMO-stack process can be tuned to enhance one particular dip at the expense of the others, but the goal is to enhance all dips simultaneously. The $v(z)$ DMO result shows slightly better continuity of amplitude along the sloping reflector than the constant-velocity result, without degrading the flat-lying sediments.

The point is perhaps more clearly seen in the constant-velocity stacks of Figure 14. These panels have been computed for a small window taken from the center of the stacked sections of Figure 13. While stacking velocities vary with time, at any particular time only a single velocity may be used. Therefore DMO can be judged by how well it equalizes the stacking velocities for dipping and horizontal reflections at a given time.

Consider the steeply sloping salt flank reflection in the stacked sections. This corresponds to the sloping reflection seen in the constant-velocity stack panels at about 2.6 s. The horizontally-lying sediments adjacent to the salt appear as the flat reflections cutting through the sloping reflection. The strongest and most continuous stack of the horizontal reflections is seen when the velocity is somewhere between 2.10 and 2.15 km/s.

The constant-gradient tests suggest that constant-velocity DMO removes too much moveout from sloping reflections. Therefore, using a slightly lower velocity might be expected to better stack the sloping reflection. Figure 14.a confirms this hypothesis: the sloping reflection is best stacked with a velocity somewhere between 2.00 and 2.05 km/s. Unfortunately, using this velocity would over-correct the horizontal reflections and result in their degradation through mis-stacking.

Exact $v(z)$ DMO (Figure 14.b) has equalized the stacking velocities of the dipping and horizontal reflectors. In particular, the reflections at 2.6 s stack best at a velocity between 2.10 and 2.15 km/s, regardless of dip. Thus, $v(z)$ DMO has successfully made the stacking velocities independent of dip. The result is the improved continuity of the sloping reflection seen in the stacked section of Figure 13.b.

Careful comparison of the constant-velocity stacks reveals that $v(z)$ DMO has altered the horizontal reflections as well as the sloping reflections. Unlike other DMO implementations, this formulation *does* affect reflections from horizontal layers. As described in a later section, the differences are due to errors in the Dix approximation used to apply NMO (see, e.g., Yilmaz, 1987, pp. 159–160). The approximation breaks down at long offsets and small times when the velocity varies with depth. Since the $v(z)$ DMO method used here employs ray tracing through the stratified medium to determine traveltimes, it in effect removes the NMO applied with the Dix approxi-

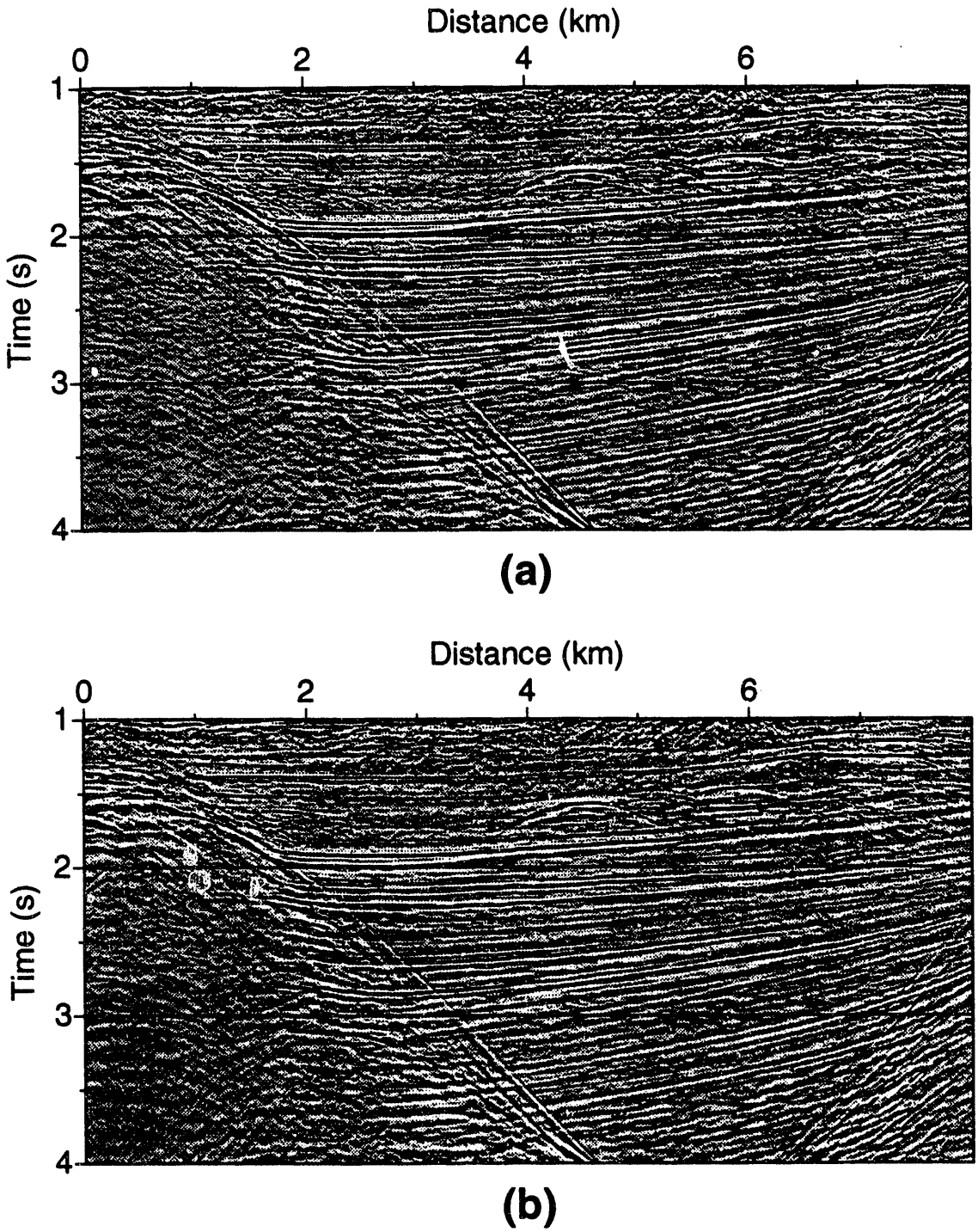

FIG. 13. Stacked sections near a salt dome processed with (a) constant-velocity DMO and (b) $v(z)$ DMO.

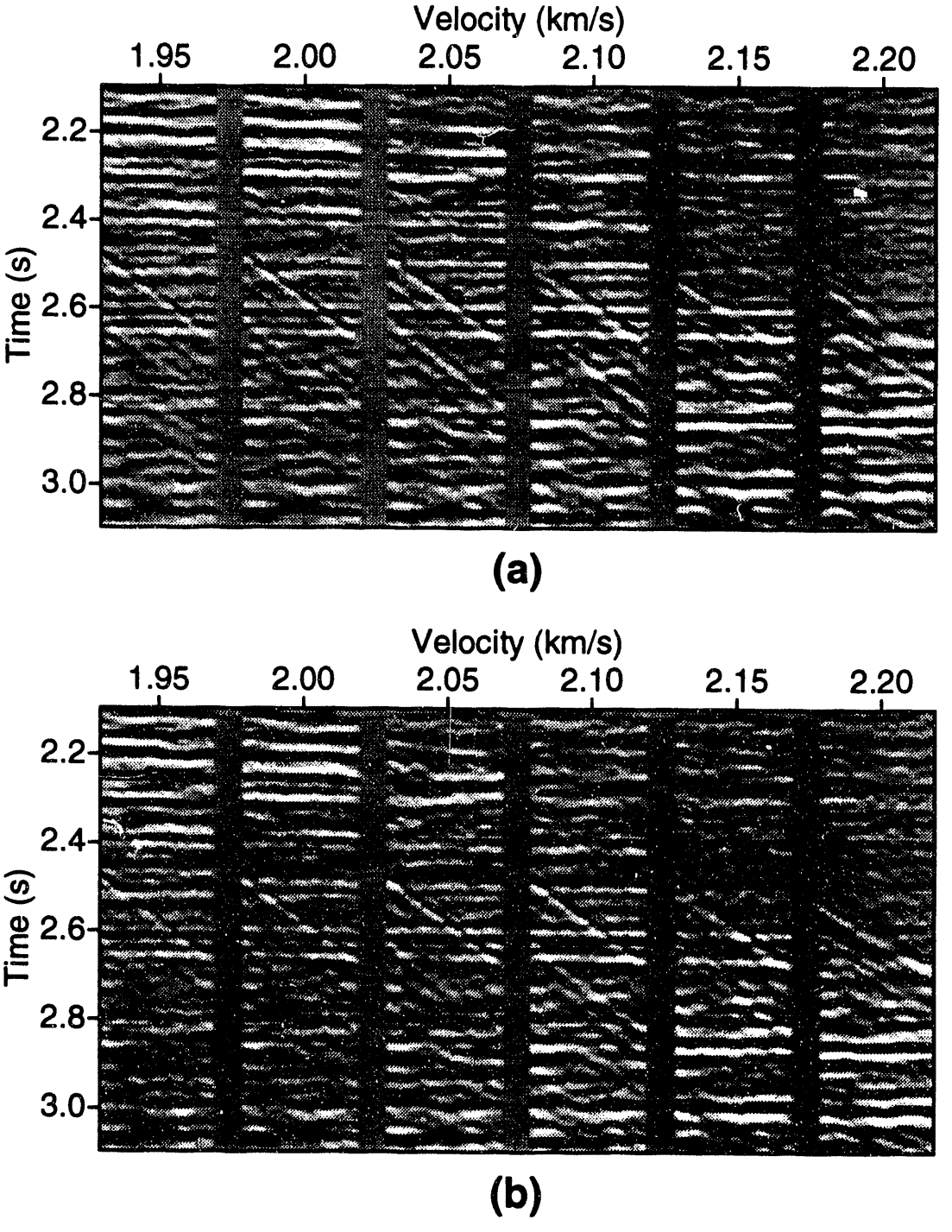

FIG. 14. Constant-velocity stacks for a window extracted from the stacks of Figure 13. (a) Constant-velocity DMO. (b) $v(z)$ DMO.

mation and reapplies the exact, ray-traced NMO for flat layers at the same time it applies the DMO correction.

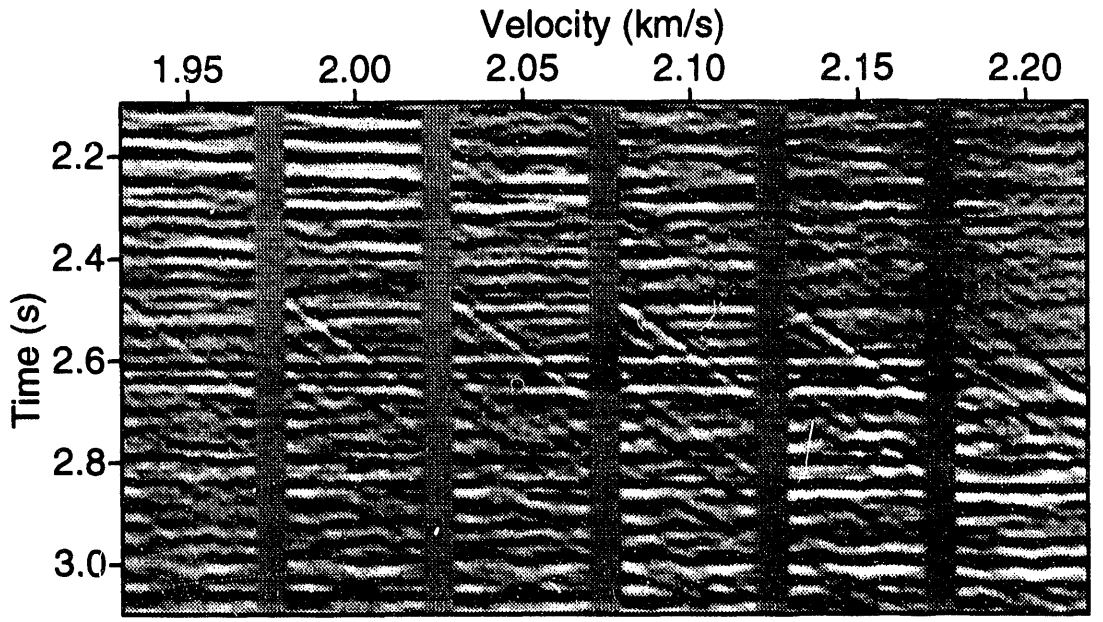

FIG. 15. Constant-velocity stacks computed using squeezed DMO.

Figure 15 shows the corresponding constant-velocity stacks for squeezed DMO. Like $v(z)$ DMO, squeezed DMO has equalized the stacking velocities of the dipping and horizontal reflections at 2.6 s. The result is very similar to the previous $v(z)$ DMO result, though horizontal reflections have not been altered.

Figure 16 shows the interval velocity function as well as the corresponding rms velocities used for $v(z)$ DMO correction. Figure 17 shows a DMO impulse response for 3.0 km offset, superimposed on the corresponding constant-velocity operator for an NMO time of 6.0 s. Unlike the previous operators, this curve now shows *four* branches. The constant-velocity operator matches the actual operator only in the small curved region near the apex. As the velocity layering becomes more general, the shape of the impulse response becomes more complicated.

Figure 18 shows several DMO impulse responses for 3.0 km offset, superimposed on the corresponding constant-velocity operators. As might be expected the errors are most severe at early times. Also note that the width of the operator decreases with time until 4.0 s, but then begins to increase again. This is in contrast with the constant-velocity operator, which decreases monotonically in width with time.

The precise shape of the DMO impulse response is highly dependent on the details of the velocity function. DMO, however, is generally applied early in the seismic processing flow, before detailed knowledge of velocity function is available. Because the constant-velocity stacks for $v(z)$ DMO and squeezed DMO, as well as the correspond-

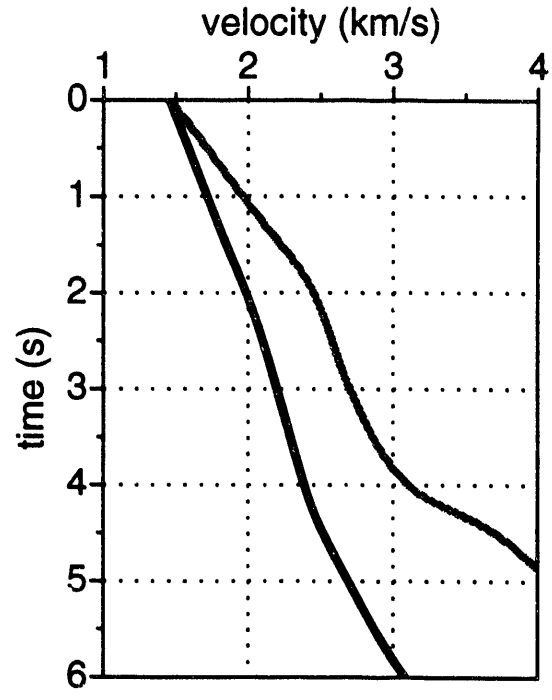

FIG. 16. The interval velocity function (gray) and corresponding rms velocities (black) used to process the data of Figure 13.

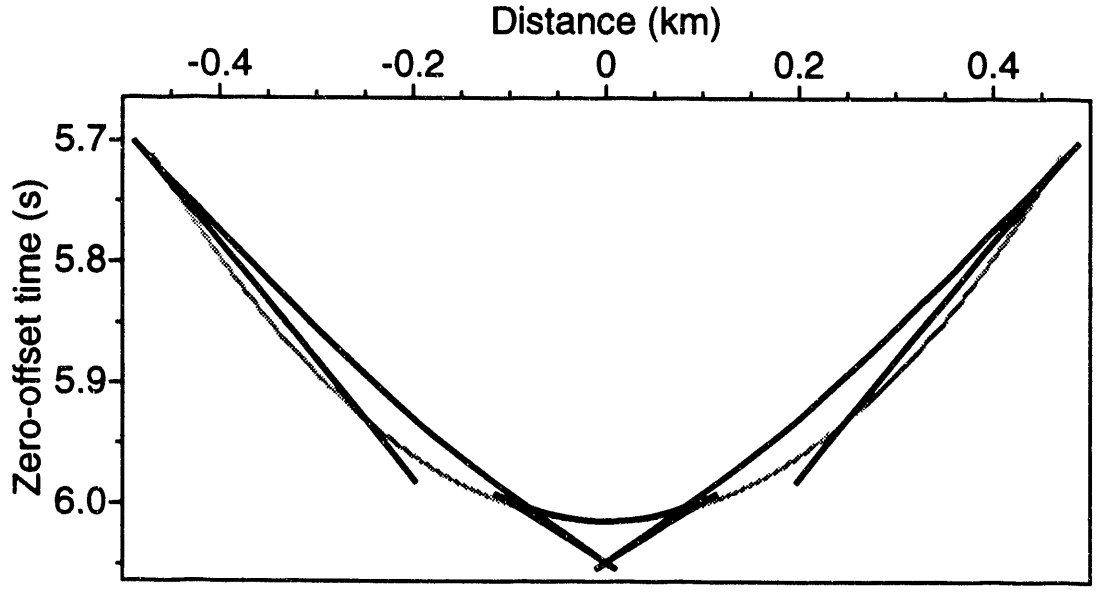

FIG. 17. DMO operator for 3.0 km offset and NMO time of 6.0 s. This operator has four branches. The constant-velocity DMO operator is shown in gray.

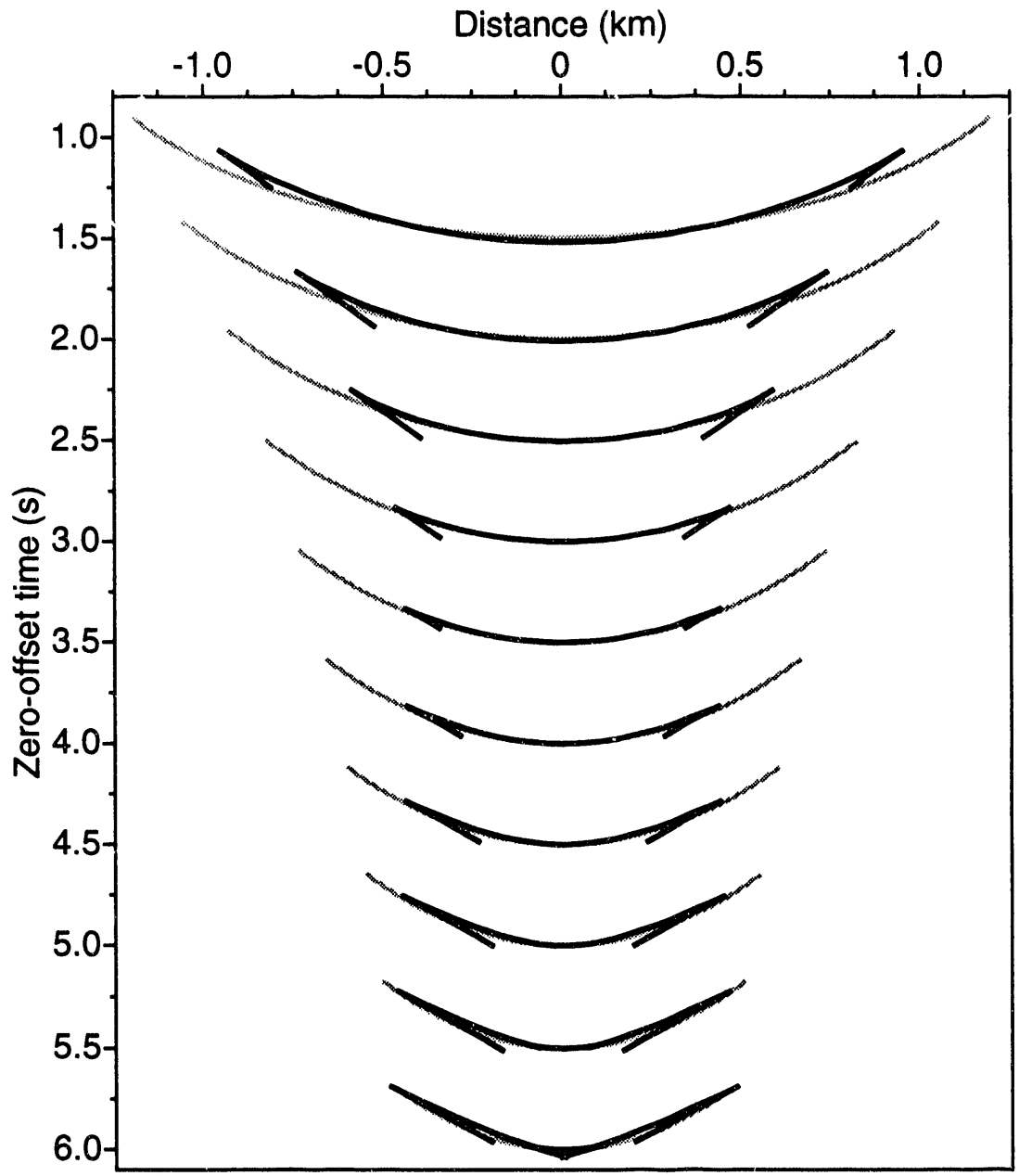

FIG. 18. Several DMO operators for 3.0 km offset and the velocity function of Figure 16. The corresponding constant-velocity DMO operators are shown in gray.

ing stacked sections, are so similar for these data, there is little benefit in performing the more complicated $v(z)$ DMO correction. The approximate solution offered by squeezed DMO is adequate and probably more appropriate, since it requires only an estimate of the velocity profile, not a detailed interval velocity function.

Cost

Even when the velocities are well-determined, the improved accuracy of $v(z)$ DMO comes at the cost of increased compute time. Compared to conventional and squeezed DMO applied via fast Fourier transform through logarithmic stretching of time (Notfors and Godfrey, 1987), this method takes anywhere from twice to ten times as long, depending on the parameters of the problem.

All the data were processed on an IBM RS/6000 Model 520 workstation. The synthetic data examples consisted of 360 CMP gathers containing 30 offsets for a total of 10 800 traces. Each trace was 251 samples long, and the total size of the dataset was approximately 10 megabytes. The processing times are given in Table 1. Comparing CPU times shows that $v(z)$ DMO took over four times as long to complete as standard DMO. If wall clock times are compared, however, $v(z)$ DMO is only twice as expensive. I believe the wall clock times are closer to each other because of the significant amount of time spent reading from and writing to disk. Both methods, of course, spend roughly the same amount of time performing disk I/O, but it is a larger fraction of the total time for conventional DMO.

Table 1. Compute times in minutes and seconds for synthetic data tests.

	CPU time	wall time
$v(z)$ DMO	4:53	9:33
DMO	1:10	4:57
ratio	4.2	1.9

Table 2. Compute times in hours and minutes for Gulf of Mexico data.

	CPU time	wall time
$v(z)$ DMO	4:33	5:56
DMO	0:31	0:52
ratio	8.8	6.8

The Gulf of Mexico data consisted of 600 CMPs of 30 offsets each, giving 18 000 traces. The traces were 751 samples long and the dataset was approximately 51 megabytes in size. Table 2 shows the CPU and wall clock times for DMO processing on the same workstation. The exact $v(z)$ DMO is now approximately seven to nine times

as expensive. Note that for this dataset, the time spent reading from and writing to disk is less significant, so the wall clock and CPU time ratios are more consistent with each other.

The survey parameters of the Gulf of Mexico data are, of course, more typical of exploration seismic data than those of the synthetic data. Therefore I maintain that $v(z)$ DMO can be characterized as being five to ten times as expensive as log-stretched DMO by Fourier transform for realistic datasets. Whether the added cost results in significantly improved stacked data depends on the particular dataset and velocity model. In this case, it did not.

These examples have demonstrated the effectiveness of $v(z)$ DMO processing over constant-velocity DMO and, in some cases, squeezed DMO. The next chapters describe how I compute and apply the $v(z)$ DMO correction for arbitrarily layered velocity fields.

DMO by Dip Decomposition

There are many ways of applying DMO correction to seismic data. For example, the DMO process can be applied in the time–space domain via finite-difference methods (Bolondi et al., 1982; Li, 1992) or with integral (i.e., Kirchhoff) schemes (Deregowski, 1985, 1987). Alternatively, DMO can be applied in the frequency–wavenumber domain. The frequency–wavenumber methods include Hale's (1983, 1984) original DMO by Fourier transform method, as well as its log-stretch variants (see, e.g., Notfors and Godfrey, 1987; Liner, 1990), and the *dip-decomposition* formulation of Jakubowicz (1984, 1990).

The $v(z)$ DMO correction developed in the section on Computing the $v(z)$ DMO Mapping is most naturally applied through an extension of this last technique. Meinardus and Schleicher (1991) and Witte (1991) also use Jakubowicz's method to apply approximate $v(z)$ DMO correction. Here I describe DMO by dip decomposition for constant velocity and lay the groundwork for its extension to variable-velocity media.

DMO for a single slope

Hale's (1984) method transforms NMO-corrected, common-offset seismic data $q_n(t_n, x)$ to the equivalent zero-offset section $q_0(t_0, x)$ in four steps. First, a Fourier transform over space takes the data from $q_n(t_n, x)$ to $q_n(t_n, k)$. Next, the frequency–wavenumber representation of the zero-offset data is computed by

$$Q_0(\omega_0, k) = \int_0^\infty dt_n \, e^{i\omega_0 t_n A} A^{-1} q_n(t_n, k), \tag{1}$$

where $2h$ is the source-receiver offset and

$$A = \left(1 + \frac{k^2 h^2}{\omega_0^2 t_n^2}\right)^{1/2}.$$

Finally, inverse Fourier transforms over frequency and wavenumber give the zero-offset data $q_0(t_0, x)$.

Note that if the data were recorded at zero offset ($h = 0$), $A = 1$ and the integral transform of equation (1) reduces to a Fourier transform over time. In this case DMO does nothing because the data are already at zero-offset.

In general, however, $A \neq 1$ and equation (1) is not a Fourier transform. This integral transform then becomes computationally intensive because it cannot be performed by FFT. Hoping to reduce the cost of DMO processing, Jakubowicz (1984, 1990) developed a clever technique for applying the moveout correction to the data one slope at a time.

For a flat reflector dipping at an angle θ in a medium of constant velocity v, the relationship between zero-offset time t_0 and NMO time t_n at a common midpoint is

$$t_0^2 = t_n^2 + \frac{4h^2 \sin^2 \theta}{v^2}$$

(Hale, 1984). By noting that $p_0 = 2\sin\theta/v = k/\omega_0$, where $p_0 \equiv \partial t_0/\partial x$ is the reflection slope measured on a zero-offset section, Jakubowicz (1990) wrote this relationship as

$$t_n^2 = t_0^2 - p_0^2 h^2. \tag{2}$$

Note that for a single reflection slope, DMO reduces to a simple slope-dependent moveout operation.

As a result, Hale's DMO formula (1) can be written for a single slope $p_0 = k/\omega_0$ as

$$\tilde{Q}_0(\omega_0, k) = \int_0^\infty dt_0\, e^{i\omega_0 t_0} q_n[t_n(t_0, p_0), k]. \tag{3}$$

By defining

$$\tilde{q}_0(t_0, k) \equiv q_n[t_n(t_0, p_0), k], \tag{4}$$

equation (3) becomes

$$\tilde{Q}_0(\omega_0, k) = \int_0^\infty dt_0\, e^{i\omega_0 t_0} \tilde{q}_0(t_0, k), \tag{5}$$

which is simply the Fourier transform of the interpolated data $\tilde{q}_0(t_0, k)$.

Equations (4) and (5) are equivalent to Hale's formula (1), but only for the particular value of reflection slope p_0 used in computing the moveout in equation (2). The moveout is incorrect for all other slopes.

Combining the slopes

Jakubowicz's method proceeds by *decomposing* the zero-offset section $Q_0(\omega_0, k)$ into its component slopes. The contribution of each slope component is computed separately via equations (4) and (5). For this reason, the method is sometimes referred to as *DMO by dip-decomposition* (Hale, 1988), or, perhaps more precisely, reflection *slope*-decomposition.

Since each component $\tilde{Q}_0(\omega_0, k)$ is only correct for the particular value of p_0 used in computing the slope-dependent moveout, the components must be filtered to remove other slopes before they are recombined to form the zero-offset section. This filtering can easily be performed in the frequency–wavenumber domain. Each slope component $\tilde{Q}_0(\omega_0, k)$ contributes to the zero-offset data $Q_0(\omega_0, k)$ only for those frequencies and wavenumbers such that k/ω_0 is near the particular slope p_0 used in equation (4). Thus, each component contributes to the zero-offset section along a wedge-shaped region centered about the line $k/\omega_0 = p_0$. The portion of $\tilde{Q}_0(\omega_0, k)$ outside this

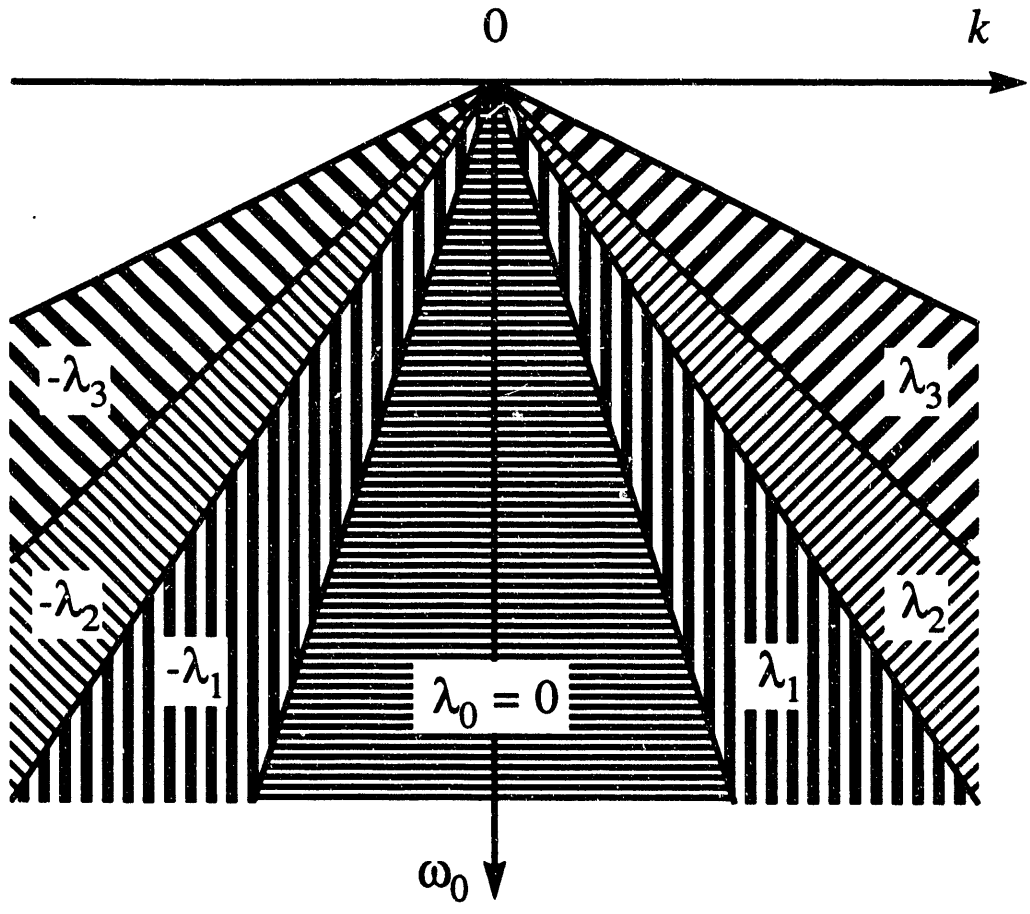

FIG. 19. Each slope component $\tilde{Q}_i(\omega_0, k)$ contributes to the zero-offset section only for those slopes near the reference slope $p_0 = \lambda_i$ used to compute that component. After Hale (1988).

wedge is discarded. Since the slope-dependent moveout function (2) is symmetric about $p_0 = 0$, both positive and negative slopes can be processed together.

This slope-filtering procedure is illustrated in Figure 19. Here the frequency–wavenumber plane has been decomposed into four reference slopes λ_i. The contribution of each slope to the zero-offset section is computed via equations (4) and (5), and the portions of the ω_0–k plane for which k/ω_0 is near $\pm\lambda_i$ are summed into the zero-offset data. Figure 20 summarizes the entire DMO by dip-decomposition algorithm.

Jakubowicz (1990) showed that the dip-decomposition technique is equivalent to Hale's (1984) DMO by Fourier transform formulation provided that the sampling of reflection slope is sufficiently fine. Hale (1988) derived a sampling criterion that avoids spatial aliasing of the zero-offset data.

DMO by dip-decomposition:

Fourier transform NMO-corrected data $q_n(t_n, x)$ *to* $q_n(t_n, k)$
Zero $Q_0(w_0, k)$ *accumulator*
For all wavenumbers k {
    For all reflection slopes p_0 {
        Compute $\tilde{q}_0(t_0, k) = q_n[t_n(t_0, p_0), k]$
            by interpolation
        Fourier transform $\tilde{q}_0(t_0, k)$ *to* $\tilde{Q}_0(\omega_0, k)$
        Sum $\tilde{Q}_0(\omega_0, k)$ *into* $Q_0(\omega_0, k)$
            for $|k/\omega_0|$ *near* p_0
    }
    Inverse Fourier transform $Q_0(w_0, k)$ *to* $q_0(t_0, k)$
}
Inverse Fourier transform $q_0(t_0, k)$ *to* $q_0(t_0, x)$

FIG. 20. Dip-decomposition algorithm for computing the DMO-corrected zero-offset data $q_0(t_0, x)$ from the input NMO-corrected data $q_n(t_n, x)$ for one common-offset section.

While Jakubowicz originally developed the dip-decomposition strategy to reduce the cost of DMO processing, several authors (e.g., Notfors and Godfrey, 1987; Liner, 1990) have since published techniques that approximate Hale's DMO integral (1) by a Fourier transform through logarithmic stretches of time. Thus, DMO can be applied entirely by fast Fourier transform, making it even more efficient than the dip-decomposition approach. Dip-decomposition, however, can easily be extended to $v(z)$ DMO processing. Meinardus and Schleicher (1991), Witte (1991), and I all apply $v(z)$ DMO via Jakubowicz's technique.

When the velocity varies, the DMO mapping $t_n(t_0, p_0)$ can no longer be calculated from equation (2). For simple velocity functions, one could hope to derive an analytic expression for the DMO mapping (see, e.g., Dietrich and Cohen, 1992). Rather than restrict myself to a few specialized velocity functions, in the next chapter I show how to compute the DMO mapping for *any* velocity function which varies *arbitrarily* with depth. Once the DMO mapping has been tabulated by this procedure, it is a simple matter to use it in the DMO by dip-decomposition formula of equation (4).

Computing the $v(z)$ DMO Mapping

As shown in the previous chapter, to perform $v(z)$ DMO by dip-decomposition one must first compute the slope-dependent relationship between NMO time t_n and zero-offset time t_0. I refer to this relationship as the DMO mapping $t_n(t_0, p_0)$, noting that it is, of course, also dependent on the source-receiver half-offset h. In an effort to simplify the notation, this dependence is not explicitly shown. In this chapter I describe a method of determining this mapping for an arbitrary depth-dependent velocity function.

Raypath geometry

I use a ray-tracing method (Slotnick, 1959) to determine this mapping for the variable velocity problem. The details of the ray-tracing are described in Appendix A. Since the velocity of the medium varies with depth only, the ray-tracing need only be performed for a single surface location. Rays leaving from other surface locations are simple lateral translations of the traced rays. Since rays do not need to be traced from every surface location, the $v(z)$ method is potentially efficient.

Each ray is tabled as three functions of the ray parameter p and the *two-way* traveltime along the ray t, as shown in Figure 21. The computed quantities are the lateral distance traveled by the ray $x(p,t)$, the *two-way* vertical traveltime $\tau(p,t)$, and the propagation angle $\theta(p,t)$. The vertical traveltime $\tau(p,t)$ is analogous to the depth of the ray, stretched according to $d\tau/dz = 2/v(z)$. Since seismic data are recorded as functions of time, it is more convenient to work with τ than the depth z. The interval velocity function of the medium $v(\tau)$ is assumed to be known.

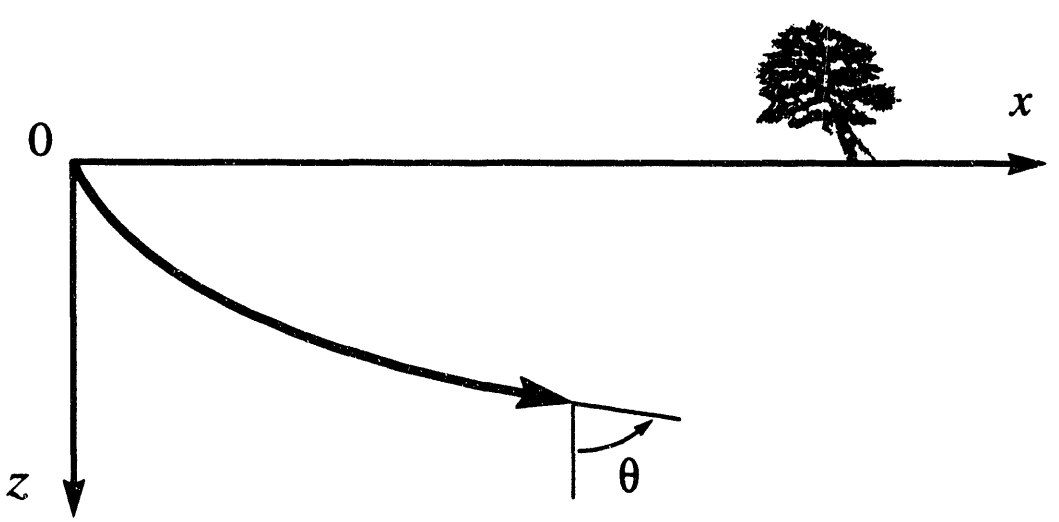

FIG. 21. Rays are traced through the medium and x, z (or τ), and θ are tabulated as functions of the ray parameter and the time along the ray.

Now that the ray tables have been constructed, the task is to find combinations of rays that meet several conditions that describe the DMO transformation. One such combination (or *trio*) of rays is shown in Figure 22. Consider a sample of data recorded at time $t = t_{sg}$ with source-receiver offset $2h$. The ray leaves the source and travels downward to the reflector, where it is reflected back up to the receiver. The total time along the path is the recording time t_{sg}, and the two segments of the path are referred to as the *source ray* and the *receiver* (or *geophone*) *ray*, respectively. Together, NMO and DMO transform this sample of finite-offset data to the equivalent zero-offset data that would be recorded at the *zero-offset location* x_0 at time t_0.

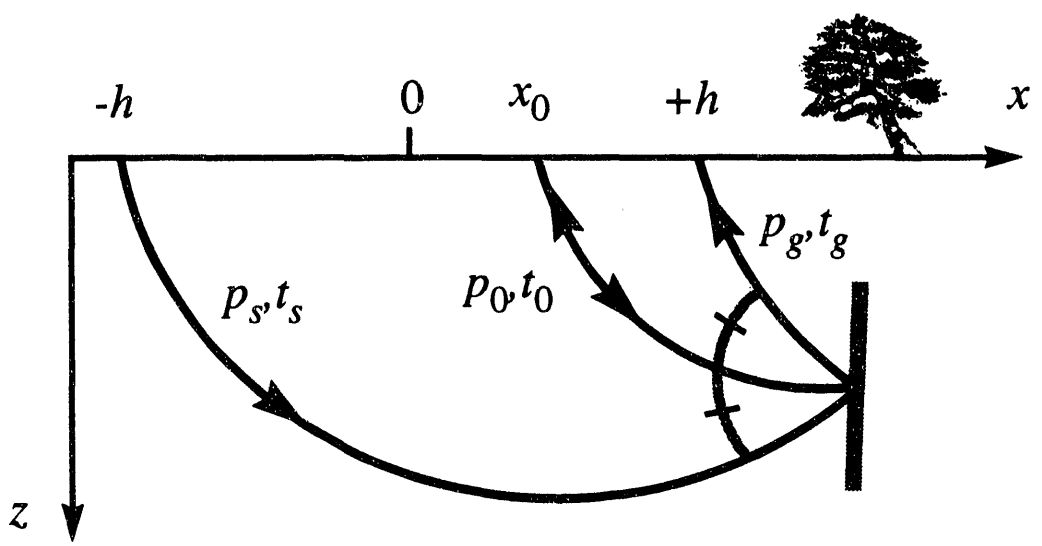

FIG. 22. Diagram showing a DMO raypath trio. All three rays must terminate at the reflection point and the zero-offset ray bisects the angle formed by the source ray and the geophone ray. The zero-offset ray is normal to the reflecting surface at the reflection point.

Initially, only the half-offset h and the recording time t_{sg} are known. For each zero-offset slope p_0, the problem is to find the corresponding trio of rays and, consequently, the two-way time along the zero-offset ray t_0 and the zero-offset location x_0. This (x_0, t_0) pair gives one point on the DMO operator. The entire operator is constructed as the locus of points (x_0, t_0), where each point corresponds to a different value of p_0. In the following sections I show how to compute these points and the DMO mapping.

System of equations

The source ray, with ray parameter p_s, leaves from $x = -h$ and terminates at the reflection point (x_r, z_r). Similarly, the geophone ray departs from $x = h$ with ray parameter p_g, while the zero-offset ray leaves from the given zero-offset location x_0 with ray parameter p_0. Both rays meet the ray from the source at the reflection point.

Take the source-receiver half-offset h, the recording time t_{sg}, and the zero-offset ray parameter p_0 as given. The ray parameters along the source and geophone rays, the departure point of the zero-offset ray, and the times along all three rays have yet to be determined. This gives a total of six unknowns. However, the rays must satisfy several conditions if they are to form a valid raypath trio.

The first requirement is that all three rays terminate at the same lateral location x_r. Consequently, the x-coordinates of the tips of the source, geophone, and zero-offset rays are, respectively,

$$\begin{aligned} x_r &= x(p_s, 2t_s) - h && (6)\\ &= x(p_g, 2t_g) + h && (7)\\ &= x(p_0, t_0) + x_0, && (8) \end{aligned}$$

where $x(p,t)$ is the lateral position function determined from ray tracing, as described above. Eliminating the unknown lateral location x_r from equations (6)–(8) gives two relationships between the six unknowns.

Furthermore, since the three rays also terminate at the same depth z_r, they must also have the same vertical two-way time τ_r,

$$\begin{aligned} \tau_r &= \tau(p_s, 2t_s) && (9)\\ &= \tau(p_g, 2t_g) && (10)\\ &= \tau(p_0, t_0). && (11) \end{aligned}$$

Again, eliminating the unknown vertical time τ_r from equations (9)–(11) gives two more relationships between the unknowns.

The final condition is that the zero-offset ray must bisect the angle formed by the intersection of the source ray and the geophone ray. This ensures that the source ray and the geophone ray give a specular reflection at the reflection point and that the zero-offset ray is normal to the reflecting surface there. The propagation angle of the zero-offset ray at the reflection point is thus the average of the propagation angles of the source and geophone rays,

$$\theta(p_0, t_0) = \frac{1}{2}\Big[\theta(p_s, 2t_s) + \theta(p_g, 2t_g)\Big]. \qquad (12)$$

Equations (6)–(12) now define a system of five nonlinear equations in the six unknowns. However, the given recording time t_{sg} yields the needed sixth equation. The recording time is simply the sum of the times along the source and geophone rays,

$$t_{sg} = t_s + t_g. \qquad (13)$$

Equation (13) can be used to immediately eliminate t_s from the remaining equations, leaving a system of five nonlinear equations,

$$0 = x(p_g, 2t_g) - x[p_s, 2(t_{sg} - t_g)] + 2h$$

$$
\begin{aligned}
0 &= x(p_g, 2t_g) - x(p_0, t_0) + h - x_0 \\
0 &= \tau(p_g, 2t_g) - \tau[p_s, 2(t_{sg} - t_g)] \\
0 &= \tau(p_g, 2t_g) - \tau(p_0, t_0), \\
0 &= \theta[p_s, 2(t_{sg} - t_g)] + \theta(p_g, 2t_g) - 2\theta(p_0, t_0)
\end{aligned}
\tag{14}
$$

in the five unknowns x_0, p_s, p_g, t_g, and t_0 and three parameters h, t_{sg}, and p_0.

Solving the system

For each set of the parameters h, t_{sg}, and p_0, I find the roots of this system of nonlinear equations using multi-dimensional Newton-Raphson iteration (see, e.g., Press et al., 1986, pp. 269–273). In this section I summarize the Newton-Raphson method.

A system of N equations in the N unknowns $u_1, u_2, \ldots, u_N$ can be written as

$$f_i(u_1, u_2, \ldots, u_N) = 0, \qquad i = 1, 2, \ldots, N. \tag{15}$$

By defining the N-element vector of functions $\mathbf{F} = (f_1, f_2, \ldots, f_N)$, and the N-element vector of unknowns $\mathbf{u} = (u_1, u_2, \ldots, u_N)$, equation (15) is compactly written

$$\mathbf{F}(\mathbf{u}) = \mathbf{0}. \tag{16}$$

Given a trial solution $\mathbf{u}_{\text{old}}$ to the system, Newton-Raphson iteratively refines the solution according to

$$\mathbf{u}_{\text{new}} = \mathbf{u}_{\text{old}} - \left[\frac{\partial \mathbf{F}}{\partial \mathbf{u}}\right]^{-1}_{\mathbf{u}_{\text{old}}} \mathbf{F}(\mathbf{u}_{\text{old}}), \tag{17}$$

where $\partial \mathbf{F}/\partial \mathbf{u}$ is the $N \times N$ matrix of partial derivatives

$$\frac{\partial \mathbf{F}}{\partial \mathbf{u}} = \begin{pmatrix} \partial f_1/\partial u_1 & \partial f_1/\partial u_2 & \ldots & \partial f_1/\partial u_N \\ \partial f_2/\partial u_1 & \partial f_2/\partial u_2 & \ldots & \partial f_2/\partial u_N \\ \vdots & \vdots & \ddots & \vdots \\ \partial f_N/\partial u_1 & \partial f_N/\partial u_2 & \ldots & \partial f_N/\partial u_N \end{pmatrix}. \tag{18}$$

Since equation (17) requires the inverse of this matrix, Newton's method will clearly fail if this matrix is singular or near-singular. Furthermore, a nonsingular matrix does not guarantee convergence. But when the method does converge, it does so quickly (Press et al., 1986, p. 270). Equation (17) is iteratively applied to update $\mathbf{u}$ until equation (16) is satisfied to within some prescribed tolerance.

In the $v(z)$ DMO application, $N = 5$, the elements of $\mathbf{F}$ are the right-hand-sides of the equations in system (14), and the elements of $\mathbf{u}$ are the unknowns x_0, p_s, p_g, t_g, and t_0. I have found that typically five iterations are sufficient to solve the system to within an acceptable tolerance, and I have not experienced convergence problems. The first partial derivatives needed for equation (17) are enumerated in Appendix B.

It is clear that the equations in system (14) have different dimensions. The first two have units of length; the next two, time; and the last, angle. In some cases it may be necessary to use dimensionless variables and a dimensionless system to improve the conditioning of matrix (18) for numerical purposes. I, however, did not encounter the need for such measures in my work. I simply used the natural units of exploration seismology: distances in meters or kilometers, times in seconds, and angles in radians.

Extracting the DMO mapping from the solution

As presented, the finite-offset recording time t_{sg} was used as a parameter of the system and the zero-offset time and location were output. Therefore this process takes finite-offset seismic data directly to zero offset, combining the operations of NMO and DMO. However, the typical seismic data processing flow performs this operation in two steps: NMO correction, followed by DMO correction.

This $v(z)$ DMO formulation is easily put in this form by making the NMO time t_n a parameter and in turn computing the recording time t_{sg} from t_n using the Dix approximation,

$$t_{sg}^2 = t_n^2 + \frac{4h^2}{V_{\text{rms}}^2(t_n)}. \tag{19}$$

Here, $V_{\text{rms}}(t)$ denotes the root mean square average of the interval velocity function of time $v(t)$,

$$V_{\text{rms}}(t) = \left[\frac{1}{t}\int_0^t ds\, v^2(s)\right]^{1/2}.$$

The resulting recording time is then used as a parameter of the system.

Figure 23 shows a schematic representation of a DMO impulse response computed by this procedure. The operator is constructed by first computing t_{sg} from the given NMO time, offset, and velocity function via equation (19). Next system (14) is repeatedly solved for particular values of the zero-offset ray parameter. The kinematic constant-offset impulse response is then formed by the locus of $(x_0(p_0), t_0(p_0))$ points swept out as p_0 is varied.

An impulse at time t_n on the input NMO-corrected common-offset section is spread out along the DMO operator defined by these points. Note that the operator is shifted below the input time t_n, implying that $v(z)$ DMO moves data for which $p_0 = 0$. While constant-velocity DMO has no effect on horizontal reflections, this formulation *does* because it automatically corrects for the error in the small offset approximation in the Dix equation (19) used to compute NMO. The error is most significant at small times and large offsets. This $v(z)$ DMO formulation first removes the approximate NMO correction from the data then applies the *exact* transformation to zero offset (simultaneous NMO and DMO correction).

To get the slope-dependent moveout required for DMO by dip-decomposition, the zero-offset time t_0 corresponding to a particular slope p_0 must be projected back to

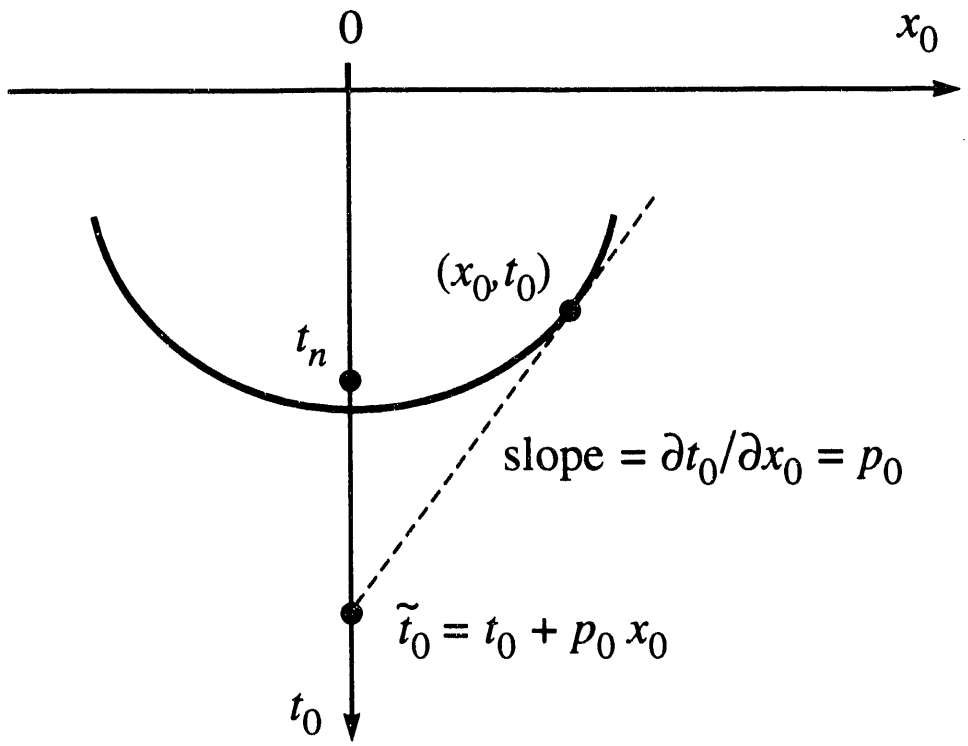

FIG. 23. The kinematic DMO impulse response, shown here as the heavy elliptical arc, is found by solving system (14) and parametrically plotting $(x_0(p_0), t_0(p_0))$ pairs for fixed t_n and h. To find the slope-dependent moveout needed for DMO by dip-decomposition, t_0 is projected back to the midpoint along a line with slope p_0.

the midpoint along the line tangent to the DMO operator at (x_0, t_0). By definition, the slope of this tangent is $\partial t_0/\partial x_0 = p_0$, so the projection is

$$\tilde{t}_0 = t_0 + p_0 x_0.$$

It is actually this $\tilde{t}_0$ that is referred to as simply t_0 in the DMO mapping $t_n(t_0, p_0)$ required for applying DMO by dip-decomposition. At this point, however, t_n is an independent variable and $t_0 = t_0(t_n, p_0)$. Inverse interpolation is used to make t_0 the independent variable. For a fixed value of p_0, assume that $t_0(t_n)$ is a monotonically increasing one function of t_n. This ensures that no two NMO times t_n correspond to the same zero-offset time t_0. Then linear interpolation can be used to find $t_n(t_0)$ for that slope, and $t_n(t_0, p_0)$ in general.

The assumption that $t_0(t_n)$ is monotonic has not proven to be a serious limitation of this scheme. In practice I have found that it tends to be violated only for large offsets and small times. Such data are often muted as part of the NMO process anyway. To avoid these troublesome areas, I compute the mapping only for times that survive the user-specified NMO stretch mute. When the problem of non-monotonicity

arises elsewhere, I simply use linear extrapolation to complete $t_n(t_0)$. This happens so infrequently that I have observed no obvious adverse effect on the data.

Summary and algorithm

The above sections describe a method for computing the $v(z)$ DMO mapping. Combined with the previously described dip-decomposition method, this yields a technique for applying $v(z)$ DMO correction to seismic data. At this point I shall summarize the steps of the algorithm and address a few related computational issues.

The scheme for computing the $v(z)$ DMO mapping $t_n(t_0, p_0)$ is outlined in the algorithm of Figure 24. Note that rays are only traced through the medium once, due to the laterally homogeneous velocity field. Next, working with one offset at a time, the DMO mapping is found for all NMO times and zero-offset slopes by solving system (14) using Newton-Raphson iteration (equation (17)).

Newton-Raphson requires an initial guess at the solution to start the iteration. For each value of h and t_n, the trial solution to system (14) for $p_0 = 0$ is determined assuming a constant-velocity earth. For subsequent p_0, the final solution from the previous p_0 is carried forward to the new p_0 and used as the trial solution there. Referring back to Figure 23, this means that first the point at the apex of the operator is found, corresponding to $p_0 = 0$. Successive points are boot-strapped from the previous point, sweeping out one limb of the operator as p_0 is increased. (Since the operator is symmetric about the midpoint, it is not necessary to trace out the other limb, which corresponds to $p_0 < 0$.)

As p_0 increases, the depth of the reflection point gradually decreases. Eventually the reflection point reaches the surface of the earth. For a medium with constant velocity v this occurs at $p_0 = 2/v$, while it is typically met at some fraction of $2/v_0$ for depth-variable media, where v_0 is the velocity at the surface. The depth of the reflection point is monitored in this scheme, and the loop over slope is broken when this occurs. Reflections in the seismic data with slopes greater than this are evanescent and need not be processed.

The slope sampling is uniform in p_0^2 to avoid aliasing, as described by Hale (1988). He showed that sampling interval in p_0^2 must be

$$\Delta(p_0^2) < \frac{T}{Fh^2},$$

where T and F are the minimum time of interest and the maximum frequency in the data, respectively. Strictly speaking, this is only true for constant velocity, but I have found it to be an effective criterion for avoiding spatial aliasing in the variable velocity problem as well.

Since the DMO operators change slowly with time, the efficiency of the computation may be improved by sampling the NMO times at a multiple of the time sampling

Computation of the DMO mapping:

Compute ray tables $x(p,t)$, $\tau(p,t)$, $\theta(p,t)$
For all half-offsets h {
    For all NMO times $t_n = 0, \Delta t_n, 2\Delta t_n, \ldots$ {
        Compute recording time via $t_{sg}^2 = t_n^2 + 4h^2/v_2^2(t_n)$
        Compute trial solution to system for $p_0 = 0$,
            assuming $v = \text{const} = v_2(t_n)$
        For all $p_0^2 = 0, \Delta p_0^2, 2\Delta p_0^2, \ldots$ {
            Refine trial solution to system using
                Newton-Raphson iteration
            Project t_0 *back to the midpoint with*
                $t_0(t_n, p_0) := t_0(t_n, p_0) + p_0 x_0(t_n, p_0)$
            Save the inverse DMO mapping $t_0(t_n, p_0)$
            Use final solution for this p_0 *as*
                trial solution for next p_0
        }
    }
    For all $p_0^2 = 0, \Delta p_0^2, 2\Delta p_0^2, \ldots$ {
        Compute the DMO mapping $t_n(t_0, p_0)$ *from*
            $t_0(t_n, p_0)$ *via inverse interpolation*
    }
    Apply DMO via dip-decomposition to common-
        offset section (as described in the
        section on DMO by Dip Decomposition)
}

FIG. 24. Algorithm for computing and applying $v(z)$ DMO correction.

interval of the data. In practice, I have found that $\Delta t_n = 10\Delta t$ works well. The operators are linearly interpolated to the finer sampling interval of the data when performing the inverse interpolation from $t_0(t_n)$ to $t_n(t_0)$.

Conclusion

Accurate poststack imaging of steeply dipping reflectors in a medium where the velocity changes with depth requires DMO processing that faithfully honors that velocity variation. While conventional DMO correction schemes assume that velocity is constant, and while there are several approximate $v(z)$ techniques available now, the scheme presented here gives a method for *exactly* computing and applying DMO correction for arbitrary depth-variable velocity functions. Additionally, this method makes no assumptions about the offset or reflector dip. This exact DMO method properly handles energy reflected from vertical and even overhanging reflectors. An additional characteristic of this method is the removal of the errors at long offsets and small times resulting from the assumption of hyperbolic moveout in NMO correction.

Since the velocity of the medium is assumed to be laterally invariant, the cost of ray tracing need only be borne once. Thus the time spent tracing rays is an almost insignificant portion of $v(z)$ DMO processing. Similarly, the DMO mapping is computed just once per offset, and it is applied to all data with that offset. Together, these efficiencies keep the cost of computing the DMO correction low compared with the cost of actually applying it to the data via dip-decomposition, thereby making exact $v(z)$ DMO an affordable process.

The tests described in the section on Applications indicate that $v(z)$ DMO can yield significant improvements over conventional DMO. Stacking velocities are better equalized with this method, resulting in stacked sections with a wider bandwidth of reflection slopes. The price for increased accuracy is increased compute times.

When compared to an approximate $v(z)$ DMO (Hale and Artley, 1991), the differences are less significant. Given an accurate interval velocity model, exact $v(z)$ DMO can deliver higher accuracy than approximate techniques. DMO, however, is typically applied at a stage in the processing flow when detailed knowledge of the interval velocities is unavailable. In fact, due to the increased cost of exact $v(z)$ DMO and its sensitivity to the velocity model, it is probably more appropriate to use one of the approximate methods for handling vertical velocity variation.

In three dimensions, the improved accuracy of this method may prove to be more significant. Squeezing constant-velocity DMO, for example, only improves the dip component of the operator and does nothing to match the strike effects. Because the 3-D operator has support in three dimensions (e.g., Perkins and French, 1990), squeezing the (two-dimensional) constant-velocity DMO operator is ineffective. In this case, one could use this exact method, or one of the other approximate techniques (Witte, 1991; Meinardus and Schleicher, 1991).

The method of for computing the DMO correction in the previous section has already been extended to three dimensions (Godfrey, 1992). The system of equations (14) grows to include equations describing the location of the reflection point in space. Similarly, the equation describing specular reflection of the source and geophone rays in 2-D would be replaced by equations describing the azimuthal and dip

angles at the reflection point. The ray tracing tables remain unchanged, however, because the velocity field is only a function of depth.

Along with the increased accuracy of this method, it is also highly flexible. Besides the 3-D extension outlined above, changes to the ray tracing scheme and the system of equations could extend this technique to DMO for mode-converted waves. Furthermore, recent work has indicated that anisotropy may have significant effects on recorded seismic data (see, e.g., Lynn et al., 1991; Gonzalez et al., 1992; Larner and Hale, 1992). Appropriate changes to this scheme would enable it to handle DMO correction in anisotropic media.

The issue of DMO amplitude can be handled by using dynamic ray tracing (Červený, 1985). By adding the dynamic ray tracing equations to the kinematic equations of Appendix A, the geometric spreading along each ray can be stored in the ray tables along with the location and propagation angle. Then when the DMO correction is applied, the geometric spreading information could be used to correct the data for divergence. This would be a prestack analogy to the poststack dip-dependent divergence correction described by Fazzari (1992).

Acknowledgments

Thanks to my advisor, Dr. Dave Hale, for guiding me through this project. Whether at the chalkboard, the keyboard, or in the saddle, Dave leads by example and always with enviable enthusiasm

This work was supported in part by the members of the consortium project at the Center for Wave Phenomena. Other computational resources were provided by the Center for Geoscience Computing at the Colorado School of Mines. This work was also partially supported by the United States Department of Energy, grant number DE-FG02-89ER14079. (This support does not constitute an endorsement by DOE of the views expressed in this paper.)

Thanks to all of my colleagues and friends at the Center for Wave Phenomena, especially Phil Anno and Mohammed Alfaraj.

Finally, thanks to my wife, Brenda, and our parents, Dr. & Mrs. T. E. Artley and Mr. & Mrs. A. C. Bradshaw. Their unfailing love and support have given me the strength to carry forward.

References

[1] Artley, C. T., 1990, Dip-moveout for depth-variable velocity: in Center for Wave Phenomena report **CWP-097**, Colorado School of Mines.

[2] Black, J. L., Cornish, B. E., Dingwall, K. A, Gerhardstein, A. C., and Meinardus, H. A., 1985, Velocity refinement for accurate migration: preprint published by Geophysical Service Inc.

[3] Bolondi, G., Loinger, E., and Rocca, F., 1982, Offset continuation of seismic sections, Geophys. Prosp., **30**, 813–828.

[4] Červený, V., 1985, The application of ray tracing to the propagation of shear waves in complex media, *in* Dohr, G. P., Ed., Seismic shear waves, Part A: Theory; *in* Helbig, K., and Treitel, S., Eds., Handbook of geophysical exploration, Section 1: Seismic exploration, **15A**: Geophysical Press.

[5] Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell Scientific Publications.

[6] Deregowski, S. M., 1982, Dip-moveout and reflector point dispersal: Geophys. Prosp., **30**, 318–322.

[7] Deregowski, S. M., 1985, An integral implementation of dip moveout: Presented at the 47th Mtg., Eur. Assn. Expl. Geophys.

[8] Deregowski, S. M., 1986, What is DMO?: First Break, **4**, no. 7, 7–24.

[9] Deregowski, S. M., 1987, An integral implementation of dip moveout: Geophysical Trans. of the Geophys. Inst. of Hungary, **33**, 11–22.

[10] Dietrich, M., and Cohen, J. K., 1992, 3-D migration to zero offset for a constant velocity gradient: an analytical formulation: *submitted for publication in* Geophysical Prospecting; also published as Center for Wave Phenomena report **CWP-113**.

[11] Fazzari, F., 1992, A dip-dependent divergence correction: M.S. thesis, Colorado School of Mines; also published as Center for Wave Phenomena report **CWP-116**.

[12] Godfrey, R. J., 1992, DMO and $V(z)$, 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 952–954.

[13] Gonzales A., Levin., F. K., Chambers, R. E., and Mobley, E., 1992, Method of correcting 3-D DMO for the effects of wave propagation in an inhomogeneous earth, 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 966–969.

[14] Hale, D., 1983, Dip-moveout by Fourier transform: Ph.D. thesis, Stanford University; also published as Stanford Exploration Project report **SEP-36**.

[15] Hale, D., 1984, Dip-moveout by Fourier transform: Geophysics, **49**, 741–757.

[16] Hale, D., 1988, Dip moveout processing, *in* Domenico, S. N., Ed., Course notes series, **4**: Soc. Expl. Geophys.

[17] Hale, D., and Artley, C. T., 1991, Squeezing DMO for depth-variable velocity: *accepted for publication in* Geophysics; also published as Center for Wave Phenomena report **CWP-112**.

[18] Jakubowicz, H., 1984, A simple exact method of prestack partial migration: Presented at the 46th Ann. Mtg., Eur. Assn. Expl. Geophys.

[19] Jakubowicz, H., 1990, A simple efficient method of dip-moveout correction: Geophys. Prosp., **38**, 221–245.

[20] Larner K., and Hale D., 1992, Dip-moveout error in transversely isotropic media with linear velocity, 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 979–983.

[21] Levin, F. K., 1971, Apparent velocity from dipping interfaces: Geophysics, **36**, 510–516.

[22] Li, J., 1992, Finite-difference migration to zero offset, 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 962–965.

[23] Liner, C., 1990, General theory and comparative anatomy of dip moveout: Geophysics, **55**, 595–607.

[24] Lynn, W., Gonzales A., and MacKay, S., 1991, Where are the fault-plane reflections?, 61st Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1151–1154.

[25] Meinardus, H. A., and Schleicher, K., 1991, 3-D time-variant dip moveout by the *FK* Method, 61st Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1208–1210.

[26] Notfors, C. D., and Godfrey, R. J., 1987, Dip moveout in the frequency-wavenumber domain: Geophysics, **52**, 1718–1721.

[27] Perkins, W. T., and French, W. S., 1990, 3-D migration to zero-offset for a constant velocity gradient: 60th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1354–1357.

[28] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1986, Numerical recipes: Cambridge Univ. Press.

[29] Slotnick, M. M., 1959, Lessons in seismic computing: Soc. Expl. Geophys.

[30] Witte, D., 1991, Dip moveout in vertically varying media, 61st Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1181–1183.

[31] Yilmaz, Ö., 1987, Seismic data processing: Soc. Expl. Geophys.

Appendix A:
$v(z)$ Ray Tracing

In this appendix I derive the $v(z)$ ray-tracing equations needed to compute the ray tables described in the section on Computing the $v(z)$ DMO Mapping. The objective is to find the horizontal distance x, depth z, and propagation angle θ at the tip of the ray as functions of the time t along the ray and the ray parameter p, as illustrated in Figure A-1. Slotnick (1959, pp. 201–204) derived similar equations, but he took

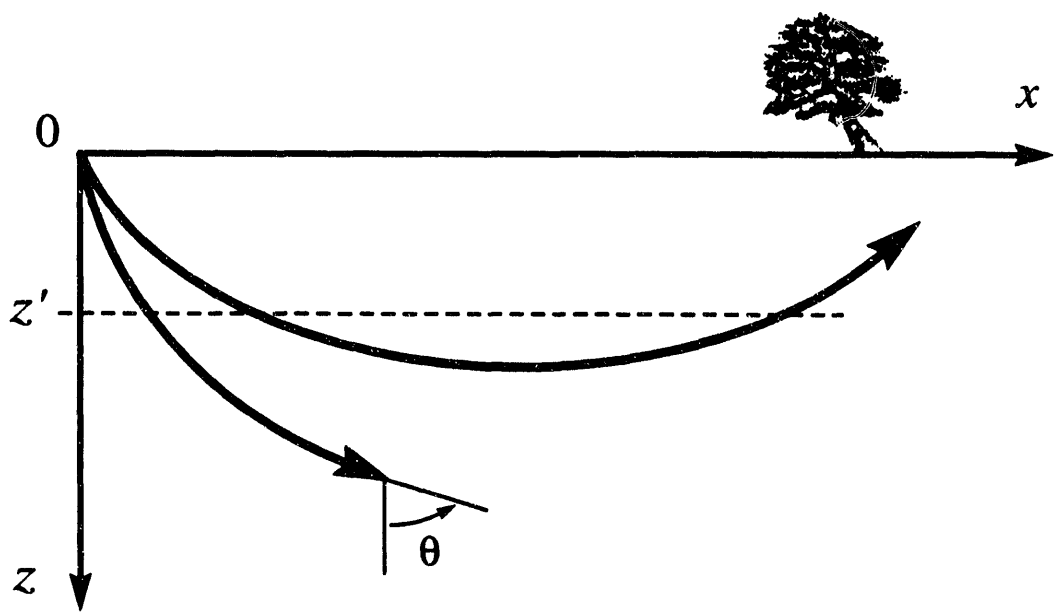

FIG. A-1. Diagram showing the ray-tracing problem. The objective is to compute x, z, and θ as functions of the ray parameter and the time. Note that the upper ray is multi-valued as a function of depth. For this reason the rays are more conveniently parameterized by their traveltime.

the time along the ray as a dependent variable and instead found x, t, and θ as functions of the depth. However, this parameterization results in functions that are double-valued at depths above the turning depth and undefined below.

For example, note that in Figure A-1 the upper ray crosses the depth z' twice. In this case, $x(z')$, $t(z')$, and $\theta(z')$, each have two distinct values corresponding to the downgoing and upgoing portions of the ray. Conversely, the ray does not penetrate below the turning depth, and thus the functions are undefined there. Parameterizing the rays as functions of the time along the ray rather than the depth avoids this computational difficulty because the rays are single-valued in these parameters.

To derive the ray-tracing equations, consider the small piece of a raypath shown in Figure A-2. The time interval Δt is taken to be small so that the velocity v can be assumed to be constant. As a result, the ray is straight over this interval. Ray

bending is handled by applying Snell's law to update the propagation angle as the ray moves into the next layer.

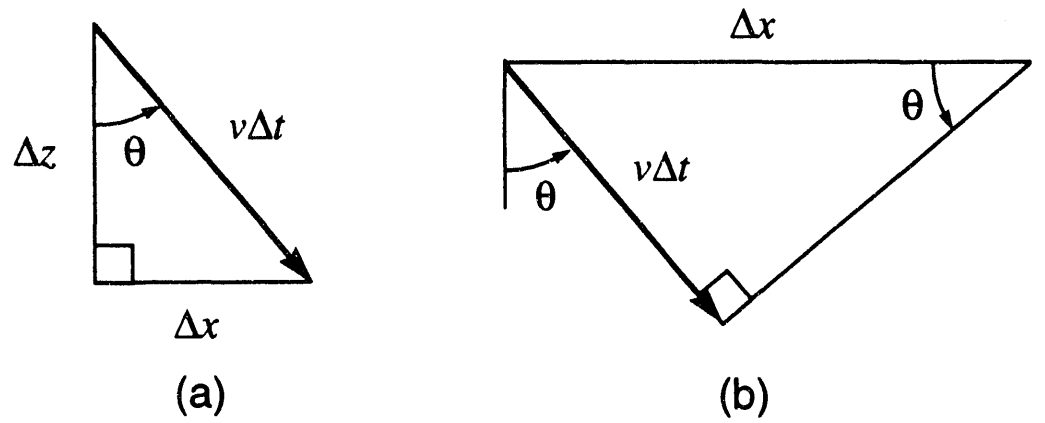

FIG. A-2. Geometrical relationships used in deriving the ray-tracing equations. (a) Triangle used to find the total derivatives dx/dt and dz/dt. (b) For $p \equiv \partial t/\partial x$, holding z constant.

Applying simple trigonometry to the triangle in Figure A-2.a gives

$$\sin\theta = \frac{\Delta x}{v\Delta t},$$

and hence, as Δt approaches zero, the total derivative dx/dt is

$$\frac{dx}{dt} = v\sin\theta. \tag{A-1}$$

Similarly,

$$\frac{dz}{dt} = v\cos\theta. \tag{A-2}$$

These two equations describe how x and z change with time and will be used to trace the ray through a layer with θ constant.

The equation used to update θ is found by differentiating Snell's law for the propagation angle. As shown in Figure A-2.b, the ray parameter $p \equiv \partial t/\partial x$ is defined as the change in arrival time of the wavefront with horizontal distance observed at a fixed depth z.

From the resulting triangle, Snell's law is

$$p = \frac{\sin\theta}{v},$$

or, rearranging,

$$\sin\theta = pv. \tag{A-3}$$

For a $v(z)$ medium the ray parameter is constant along the ray (Slotnick, 1959), so differentiating this equation with respect to time gives

$$\cos\theta \frac{d\theta}{dt} = p \frac{dz}{dt}\frac{dv}{dz}.$$

Substituting dz/dt from equation (A-2) yields

$$\frac{d\theta}{dt} = pv\frac{dv}{dz}, \tag{A-4}$$

which describes the bending of the ray as it moves through layers of differing velocity.

Equations (A-2) and (A-4) are written explicitly in terms of the depth z. As discussed earlier in the section on Computing the $v(z)$ DMO Mapping, it is generally more convenient to work with the vertical time τ, where

$$\tau = \int_0^z \frac{d\xi}{v(\xi)}$$

and hence

$$\frac{d\tau}{dz} = \frac{1}{v(z)}.$$

Using the chain rule, equations (A-2) and (A-4) become, respectively,

$$\frac{d\tau}{dt} = \cos\theta \tag{A-5}$$

and

$$\frac{d\theta}{dt} = p\frac{dv}{d\tau}. \tag{A-6}$$

Next, use Snell's law (equation (A-3)) to write equation (A-1) explicitly in terms of the ray parameter p:

$$\frac{dx}{dt} = pv^2. \tag{A-7}$$

Equations (A-5) and (A-6) together with equation (A-7) are integrated numerically to construct the ray tables $x(p,t)$, $\tau(p,t)$, and $\theta(p,t)$. These equations are the keys to the ray-tracing function used in my $v(z)$ DMO program. While they were derived here using the *one-way* traveltime along the ray t and the *one-way* vertical time τ, the equations can immediately be converted to the corresponding *two-way* times simply by using *half-velocities*, as described by, e.g., Claerbout (1985).

Appendix B: System of Equations and Their Partial Derivatives

Overview

Here I present the system of equations describing the raypath trio and the partial derivatives needed for solving the system using Newton-Raphson iteration.

As developed in the section on Computing the $v(z)$ DMO Mapping, the five equations $f_i = 0$ are

$$\begin{aligned} f_1 &= x(p_g, 2t_g) - x[p_s, 2(t_{sg} - t_g)] + 2h \\ f_2 &= x(p_g, 2t_g) - x(p_0, t_0) + h - x_0 \\ f_3 &= \tau(p_g, 2t_g) - \tau[p_s, 2(t_{sg} - t_g)] \\ f_4 &= \tau(p_g, 2t_g) - \tau(p_0, t_0) \\ f_5 &= \theta[p_s, 2(t_{sg} - t_g)] + \theta(p_g, 2t_g) - 2\theta(p_0, t_0), \end{aligned} \tag{B-1}$$

where the $f_i = f_i(x_0, p_s, p_g, t_g, t_0; h, t_{sg}, p_0)$ are functions of the five variables and the three parameters h, t_{sg}, and p_0.

Numerically solving the system with Newton's method requires that the equations and their first partial derivatives be evaluated at a given trial solution. However, the functions $x(p,t)$, $\tau(p,t)$, and $\theta(p,t)$ that appear in the equations are not computed continuously, making direct evaluation of the equations impractical. Rather, they have been previously tabled at regular intervals of the ray parameter p and the two-way time t. Linear interpolation between the nearest tabled values is used to evaluate the functions at the required points, while the partial derivatives are approximated by finite-differencing the adjacent tabled values.

One subtlety is the evaluation of the functions and their derivatives for values of $p < 0$. Consider the raypath trio for $p_0 = 0$ depicted in Figure B-1. Since the zero-offset ray travels straight down in this case, the reflection point is directly below the midpoint. Note that the reflection point is to the left of the receiver location, and hence p_g is negative. Similarly, p_g will be negative for small values of $p_0 > 0$. Therefore I require a method for evaluating the ray functions for negative p.

Because the medium is laterally invariant, the rays are symmetric with respect to their takeoff angle. The takeoff angle of a ray with ray parameter $-p$ is the negative of that of the ray with ray parameter p, so the rays are simply horizontal reflections of one another. Hence, x and θ are odd functions of p, while τ is even. Since the derivative of an odd function is even and vice versa, $\partial x/\partial p$ and $\partial\theta/\partial p$ are odd functions of p and $\partial\tau/\partial p$ is even. These relationships are used to evaluate the functions for negative p using values tabulated only for $p \geq 0$.

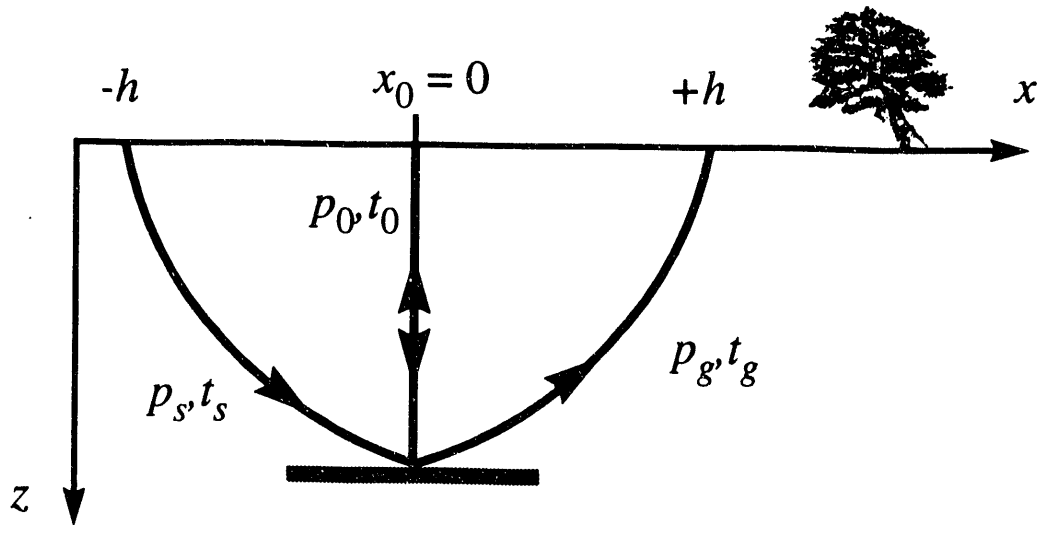

FIG. B-1. A DMO raypath trio for $p_0 = 0$. (Compare with Figure 22.) Note that $p_g < 0$ in this case because the geophone ray takes off to the left of vertical.

First partial derivatives

Partial differentiation of the equations is straight-forward, though derivatives with respect to t_g, a one-way traveltime, require the chain rule because the functions are tabled in terms of two-way traveltime. Tabulated here for completeness are the first partial derivatives of the equations of system (B-1), as required for the Newton-Raphson iteration described in the section on Computing the $v(z)$ DMO Mapping.

For the first equation of the system,

$$f_1 = x(p_g, 2t_g) - x[p_s, 2(t_{sg} - t_g)] + 2h,$$

the first partial derivatives are

$$\begin{aligned}
\frac{\partial f_1}{\partial x_0} &= 0 \\
\frac{\partial f_1}{\partial p_s} &= -\frac{\partial}{\partial p} x[p, 2(t_{sg} - t_g)]\Big|_{p=p_s} \\
\frac{\partial f_1}{\partial p_g} &= \frac{\partial}{\partial p} x(p, 2t_g)\Big|_{p=p_g} \\
\frac{\partial f_1}{\partial t_g} &= 2\,\frac{\partial}{\partial t} x(p_g, t)\Big|_{t=2t_g} + 2\,\frac{\partial}{\partial t} x(p_s, t)\Big|_{t=2(t_{sg}-t_g)} \\
\frac{\partial f_1}{\partial t_0} &= 0.
\end{aligned}$$

Similarly, the first partial derivatives for the second equation of the system,

$$f_2 = x(p_g, 2t_g) - x(p_0, t_0) + h - x_0,$$

are

$$\begin{aligned}
\frac{\partial f_2}{\partial x_0} &= -1 \\
\frac{\partial f_2}{\partial p_s} &= 0 \\
\frac{\partial f_2}{\partial p_g} &= \frac{\partial}{\partial p} x(p, 2t_g)\bigg|_{p=p_g} \\
\frac{\partial f_2}{\partial t_g} &= 2\,\frac{\partial}{\partial t} x(p_g, t)\bigg|_{t=2t_g} \\
\frac{\partial f_2}{\partial t_0} &= -\frac{\partial}{\partial t} x(p_0, t)\bigg|_{t=t_0}.
\end{aligned}$$

For the third equation of system (B-1),

$$f_3 = \tau(p_g, 2t_g) - \tau[p_s, 2(t_{sg} - t_g)],$$

the partial derivatives are

$$\begin{aligned}
\frac{\partial f_3}{\partial x_0} &= 0 \\
\frac{\partial f_3}{\partial p_s} &= -\frac{\partial}{\partial p} \tau[p, 2(t_{sg} - t_g)]\bigg|_{p=p_s} \\
\frac{\partial f_3}{\partial p_g} &= \frac{\partial}{\partial p} \tau(p, 2t_g)\bigg|_{p=p_g} \\
\frac{\partial f_3}{\partial t_g} &= 2\,\frac{\partial}{\partial t} \tau(p_g, t)\bigg|_{t=2t_g} + 2\,\frac{\partial}{\partial t} \tau(p_s, t)\bigg|_{t=2(t_{sg}-t_g)} \\
\frac{\partial f_3}{\partial t_0} &= 0.
\end{aligned}$$

Next, the fourth equation of system (B-1),

$$f_4 = \tau(p_g, 2t_g) - \tau(p_0, t_0),$$

has these first partial derivatives:

$$\frac{\partial f_4}{\partial x_0} = 0$$

$$
\begin{aligned}
\frac{\partial f_4}{\partial p_s} &= 0 \\
\frac{\partial f_4}{\partial p_g} &= \frac{\partial}{\partial p}\tau(p, 2t_g)\bigg|_{p=p_g} \\
\frac{\partial f_4}{\partial t_g} &= 2\,\frac{\partial}{\partial t}\tau(p_g, t)\bigg|_{t=2t_g} \\
\frac{\partial f_4}{\partial t_0} &= -\,\frac{\partial}{\partial t}\tau(p_0, t)\bigg|_{t=t_0}.
\end{aligned}
$$

Finally, the partial derivatives of the remaining equation of the system,

$$f_5 = \theta[p_s, 2(t_{sg} - t_g)] + \theta(p_g, 2t_g) - 2\theta(p_0, t_0),$$

are

$$
\begin{aligned}
\frac{\partial f_5}{\partial x_0} &= 0 \\
\frac{\partial f_5}{\partial p_s} &= \frac{\partial}{\partial p}\theta[p, 2(t_{sg} - t_g)]\bigg|_{p=p_s} \\
\frac{\partial f_5}{\partial p_g} &= \frac{\partial}{\partial p}\theta(p, 2t_g)\bigg|_{p=p_g} \\
\frac{\partial f_5}{\partial t_g} &= 2\,\frac{\partial}{\partial t}\theta(p_g, t)\bigg|_{t=2t_g} - 2\,\frac{\partial}{\partial t}\theta(p_s, t)\bigg|_{t=2(t_{sg}-t_g)} \\
\frac{\partial f_5}{\partial t_0} &= -2\,\frac{\partial}{\partial t}\theta(p_0, t)\bigg|_{t=t_0}.
\end{aligned}
$$

DATE
FILMED
1 / 7 / 93