Stanford Exploration Project, Report 80, May 15, 2001, pages 1–175

Speed and precision of dip moveout

Patrick Blondel[1]

ABSTRACT

The oil industry's need for high-quality images of the Earth's interior has become more and more pressing in last ten years. In practice, the quality of seismic imaging increases with the amount of data acquired. Therefore, seismic surveys often include three-dimensional and high-coverage acquisitions that result in a huge quantity of data to process. For timely results each step of the processing sequence must be as fast as possible. A time-consuming step in a standard seismic imaging flow is dip moveout correction. Therefore, enhancing the speed of this step without sacrificing precision is essential to improve the state of seismic imaging.

The rapid processing of a huge amount of data requires us to simplify complex algorithms. Under the constant velocity assumption, the formulation of the dip moveout correction reduces to the equation for an ellipse. Because its expression remains simple in three dimensions, the process is computationally efficient. However, in the case of an irregular data acquisition geometry, the chaotic spatial spreading forbids a trace-parallel implementation. This problem is solved by a time-parallel implementation that allows a fast processing for any azimuthal distribution in the data.

There are two ways to improve precision in amplitude balancing and focusing of the seismic images with the dip moveout process. The first is to apply a proper weight along the operator, which is achieved at almost no extra computational cost. The second way is to consider depth-variable velocity. The dip moveout correction then becomes computationally expensive. However, an approximation valid for gently dipping reflectors allows the variable-velocity process to be almost as fast as the constant-velocity process while improving the focusing of seismic events. This method is easily applicable in three dimensions as a first-order approximation of the theoretic "saddle" operator.

INTRODUCTION

The oil industry has become more reliant on seismic data interpretation than in the past. Two decades ago, only basic processing was required to find a potential oil trap. Gradually, the processing sequence has been refined to detect and outline new traps that were previously considered not interesting or not even considered at all. However, seismic acquisition and processing can be improved to more accurately reflect the Earth's structure. Currently, we enhance the acquisition by densifying the shot and geophone distribution or shooting three-

[1]**email:** not available

dimensional surveys and we refine the imaging tools by taking into account variable velocity and amplitude effects. Consequently, the processing industry needs to develop fast and accurate three-dimensional processing tools. Nowadays, a standard industrial seismic processing sequence almost always involves the dip moveout correction. When a constant-velocity Earth model is considered, the speed of this process makes it very attractive, even for three-dimensional data. However, the imaging quality may be poor in the case of three-dimensional data, when the amplitude and aliasing effects are not accurately considered. On the other hand, the dip moveout process can gain in precision by considering variable velocity, but it then loses rapidity. In this paper, I address the problems of speed and precision of the dip moveout process in a three-dimensional Earth model. Part I reports on research in which I make the limiting assumption of a constant velocity model in order to design a simple, fast dip moveout process. After an analysis of the amplitude and phase of the dip moveout operator, I present a parallel implementation of the process. In the work described in Part II, I make the assumption of a depth-variable velocity model in order to increase the precision of that process. This part compares the results of two different methods of $v(z)$ dip moveout in two dimensions (Artley, 1992; Castle, 1993) after post-stack migration. Finally, it includes a formulation for an accurate three-dimensional dip moveout process in a depth-variable velocity medium that Craig Artley, Alexander Popovici, Matthias Schwab, and I derived in a previous paper (Artley et al., 1993).

Chapter 1

Part I: Speed for 3-D constant-velocity dip moveout

The rapid processing of a huge amount of data requires us to simplify complex algorithms. Under the constant velocity assumption, the formulation of the dip moveout correction reduces to the equation for an ellipse. Because its expression remains simple in three dimensions, the process is computationally efficient. However, in the case of an irregular data acquisition geometry, the chaotic spatial spreading forbids a trace-parallel implementation. After discussing the amplitude and phase of the elliptic dip moveout operator, I propose an efficient time-parallel implementation that allows a fast processing for any azimuthal distribution in the data.

AMPLITUDE ANALYSIS OF THE CONSTANT-VELOCITY ANTI-ALIASING THREE-DIMENSIONAL INTEGRAL DIP MOVEOUT OPERATOR

A method of three-dimensional integral dip moveout processing for constant-velocity media must cope with problems related to amplitude and aliasing. The convolution of the dip moveout operator with triangle functions avoids the aliasing effect. A study of different amplitude weightings leads to the choice of the weighting scheme derived from a Fourier domain expression of dip moveout (Black et al., 1993). Testing the method on a 3-D synthetic data set shows the conservation of the amplitude-versus-offset (AVO) effect throughout the dip moveout (DMO) process.

Introduction

The Fourier domain DMO methods give appropriate results for 2-D lines but are time-consuming. In 3-D surveys, their implementation becomes more difficult because the spatial sampling rate as a function of azimuth is variable and irregular. In contrast to Fourier domain DMO methods, the integral DMO method is not affected by this problem and provides cheap processing

because of the limited extent of the operator (which is two-dimensional and dip-limited). However, the implementation of an integral method requires explicit knowledge of the operator as well as of the weight applied to each operation. The next section describes the shape of the operator, and the second section states three rules that the DMO operator must satisfy. **The shape of the operator**

Deregowski (1981) thoroughly described the elliptic DMO operator; I include Figure (1) only for the sake of introducing the terminology used throughout this paper. The operator is a

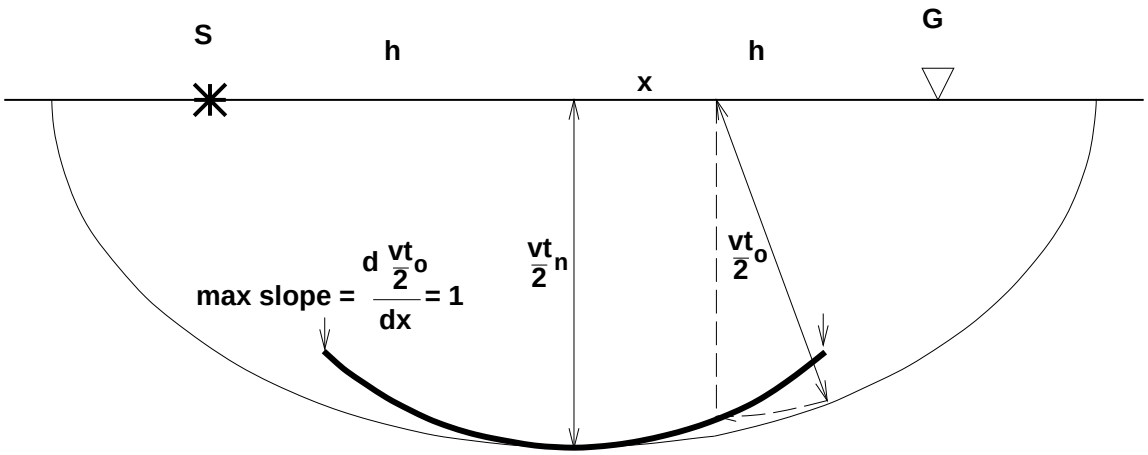

Figure 1: The DMO operator (bold line) is geometrically constructed from the elliptical reflector corresponding to an impulse in a (time, midpoint) section. Notice that in the (t_0, x) space, the maximum slope is $2/v$. patrick-Shape [NR]

dip-limited ellipse defined by the equation

$$\frac{t_0^2}{t_n^2} + \frac{x^2}{h^2} = 1. \tag{1}$$

Because it is kinematically impossible to have reflectors dipping more than the slope $2/v$ in a zero-offset constant-velocity time section, the DMO operator is dip-limited. **The rules the operator must obey**

The primary attribute of an integral method is to respect the kinematic component of the process. However, in order to yield a *consistent* stack of the operators illuminating a given location, the integration should be a weighted sum. In other words, an amplitude function should be applied along the operator. The integral DMO process will be *consistent* if it obeys the following rules.

Rule 1. According to Hale (1991), "The impulse responses [obtained by Fourier Transform DMO] may be used as a standard by which to judge integral DMO methods". Because (f,k) DMO methods have a perfect behavior with respect to amplitude, the integral DMO operator should be as close as possible to the (f,k) DMO operator in amplitude and phase. Thus, we expect the integral impulse response to have a low amplitude and a high-frequency content near $x = 0$ and a high amplitude and a low-frequency content when the slope of the operator becomes steeper.

Rule 2. Flat events must not be affected in amplitude and phase by the DMO process. This rule, clearly stated by Hale (1991), is perfectly respected by any (f,k) DMO process (Hale, 1983a; Liner, 1990), but it represents a challenging test for integral DMO processes.

Rule 3. Events of a given reflectivity must show balanced amplitude after the DMO process, whatever their dip. This rule is essential in order not to spoil the data for a possible AVO study.

The first section of this part explains how to avoid the aliasing of the operator. In the next section, the three rules stated above help us choose the most convenient weighting among three amplitude schemes selected from the literature. Finally, a brief section discusses how to apply the operator on a 3-D grid.

The triangle as an anti-aliasing structure

In the Fourier domain, the operator is not aliased. However, space-time integral methods must be applied carefully to account for operator aliasing. For a given temporal frequency of the data, the increase in the dip of the operator produces an increase in the spatial frequency until it reaches the Nyquist frequency (two points per wavelength). Beyond that point, the operator is aliased. Claerbout (1992) introduced an efficient technique to avoid the aliasing of the operator with Kirchhoff methods. Instead of spreading a simple spike along the operator, a dip-dependent triangle is effectively convolved with the operator. Assuming spatial spacing of Δx, the width of the triangle at a point of the operator is determined by equation (2)

$$\Delta_t = \left(\frac{dt_0}{dx}\right) \Delta x, \tag{2}$$

where $\left(\frac{dt_0}{dx}\right)$ is the operator dip. It is therefore assured that the operator always has at least two points per wavelength on the spatial axis, even when the time frequency of the data is Nyquist. Figure (2) shows the impulse response of a spike when the triangle anti-aliasing method is used. The phase shift caused by the half differential filter (usual in 2-D Kirchhoff methods) makes the triangles look like the teeth in a shark's jaw. The triangular weight can be built with three spikes $\frac{1}{\Delta_t}(-1,\ldots,+2,\ldots,-1)$ separated by Δ_t and submitted to both causal and anticausal integration. As Figure (3) shows, the anti-aliasing process is cheap because each output trace is double integrated only once.

Amplitudes along the DMO operator

A rich literature has been published describing "true" DMO amplitudes, which suggests that the truth is variable! In this section, I examine three propositions for amplitude weighting (Beasley and Mobley, 1988; Gardner and Forel, 1990; Black et al., 1993) to find the one that best verifies the rules stated in the introduction. I then discuss the inclusion of factors accounting for the spherical divergence and the effect of anti-aliasing on amplitudes. **Different weighting schemes**

Figure 2: Impulse response of the anti-aliasing integral DMO using triangles. Input spike: 1.0 s; velocity: 2000 ms^{-1}.

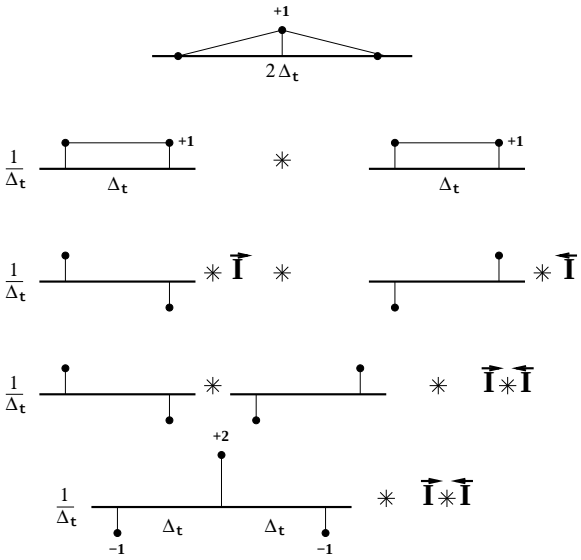

Figure 3: Decomposition of the method for building triangles. All lines are equivalent: a triangle function is the convolution of two box functions (line 1 to 2); a box function is the integration of a two-opposite-spike signal (line 2 to 3); the convolution is commutative (line 3 to 4); the convolution of two two-opposite-spike signals is the final three-spike signal (line 4 to 5). The width Δ_t of the triangle depends on both time and space variables. However, the causal ($\vec{\mathbf{I}}$) and anticausal ($\overleftarrow{\mathbf{I}}$) integrations are independent of Δ_t and can be applied only once after all convolutions are performed. patrick-triangle [NR]

The weighting scheme proposed by Beasley (1988) is based on a heuristic approach. Starting from the dip-domain formulation of DMO (Jakubowicz, 1990), he showed that the amplitudes along the DMO operator are proportional to its curvature $\frac{d^2t_0}{dx^2}$. Then he constrained the amplitudes in time and offset to verify Rule 2, assuming a dominant frequency of the data. He came up with the following expression:

$$A_1(t_n,h,x) = \frac{\Delta x}{\Delta x + 2h\sqrt{2s-s^2}} \frac{1}{\left(1-\frac{x^2}{h^2}\right)^{\frac{3}{2}}}, \tag{3}$$

where $s = (2\,f_d t_n)^{-1}$ and f_d is the dominant frequency. Gardner (1990) introduced the DMO-NMO method and derived a DMO amplitude factor that is directly related to the curvature of the operator through the expression

$$A_2(t_n,h,x) = \left(-\left(\frac{d^2t_0}{dx^2}\right)\right)^{\frac{1}{2}} = \sqrt{t_n}\frac{1}{h}\frac{1}{\left(1-\frac{x^2}{h^2}\right)^{\frac{3}{4}}}. \tag{4}$$

Recently, Black et al. (1993) derived the following mathematical expression for integral DMO from the (f,k) DMO expression:

$$A_3(t_n,h,x) = \sqrt{t_n}\frac{1}{h}\frac{1+\frac{x^2}{h^2}}{\left(1-\frac{x^2}{h^2}\right)^{\frac{3}{4}}}. \tag{5}$$

I now apply the rules to the three weighting schemes and compare their results. **Rule 1.** Figure (4) shows the comparison of the different weighting schemes normalized by their value at $x = 0$ with the amplitude picked along the impulse response of an (f,k) DMO program (Zhang's formulation). The different curves agree in shape, their concavity turned upward. However, Black's and Beasley's schemes are closer to the (f,k) DMO amplitudes. The decrease in amplitude of the (f,k) amplitude when x increases is caused by the velocity-dependent dip limitation of the operator. The three weighting functions have been plotted

Figure 4: Normalized weighting schemes compared to (f,k) DMO amplitudes. Dotted line: Beasley; Dashed line: Gardner; Dashed-dotted line: Black; Solid line: picked amplitudes on the (f,k) DMO operator.

up to $x = h$ without taking the limitation into account. **Rule 2.** Figure (5) shows impulse responses that have been stacked along the x-axis. This stack simulates the contribution of impulse responses along a horizontal planar reflector to a single trace. Thus, to obey the second rule, the stacked impulse response should yield the input impulse. The three weighting schemes restore the balanced amplitudes of the impulse rather well. However, Gardner's and Black's schemes show a better stack, especially at earlier times, where Beasley's weighting produces strongest artifacts. **Rule 3.** In order to verify the third rule, I used a 3-D synthetic

Figure 5: Impulse responses stacked along the x-axis. The top left plot is the input impulse. Three impulse responses have been generated with three weighting schemes and then stacked along the x-axis.

dataset created by David Lumley. Figure (6) shows an NMO-corrected section of the model for a shot located at x = 0. The structure is composed of three planar reflectors: a horizontal reflector showing some amplitude effect (the amplitudes increase away from the source) and two shallow, dipping reflectors (15 and 30 degrees). Figure (7) shows the result on the same section after the 3-D DMO. Some strong artifacts appear above the horizontal reflector, and the amplitudes are not well balanced along the reflectors or in relation to each other. This poor result is mainly caused by omission of spherical spreading and the amplitude-boosting effect of the anti-aliasing triangles. **Spherical divergence**

Figure 6: A slice in the 3-D data after NMO. The horizontal plane shows some amplitude effect. The two other planes dip at a 15-degree and a 30-degree angle, respectively.

Figure 7: A slice in the 3-D data after DMO with Black's amplitudes, without spherical spreading and without triangle effect corrections.

As suggested by the preceding result, spherical divergence should be included in the DMO process. Indeed, for a given shot-geophone traveltime, the zero-offset ray path when the reflector is dipping is shorter than when the reflector is horizontal. Since the decrease in amplitude of a spherical wave is inversely proportional to the distance traveled, the spherical spreading is clearly related to the dip of the structures and, thus, must be included in the DMO process. Gardner (1990) expressed the spherical spreading factor as a function of k ($\sqrt{h^2 - x^2}$) and t_2 (t_0). As a function of h, t_n, and x, it becomes

$$F_{ss} = h^2 \left(1 + \frac{v^2 t_n^2}{4h^2}\right)\left(1 - \frac{x^2}{h^2}\right). \tag{6}$$

The two first terms of expression (6) account for the offset and the depth of the reflector, whereas the third term clearly accounts for its dip. **The effect of the anti-aliasing triangle width**

The anti-aliasing is performed by effectively convolving a triangle function with the output trace. The width of the triangle, Δ_t, usually exceeds the time sampling rate, Δt, for the steepest dips. Therefore, the triangles overlap and produce an undesirable amplitude increase. The number of overlapping triangles is proportional to their width. Thus, a reasonable method of amplitude correction is to divide the convolution triangles by Δ_t, as follows:

$$F_{aa} = \frac{1}{\Delta_t}. \tag{7}$$

Figure (8) shows the same section of the 3-D cube as Figure (7), but the DMO process now involves the two correction factors for spherical spreading and triangle width. The amplitude effect along the horizontal reflector is restored, and the relative amplitude of the different reflectors is better balanced and closer to the amplitude distribution in Figure (6). The third rule is now more completely verified although some singularities occur at the near offset traces (close to $x = 0$).

Figure 8: A slice in the 3-D data after DMO with Black's amplitudes, corrected for spherical spreading and the triangle effect.

The DMO operator in three dimensions

As described by Hale (1991), the zero offset rays bouncing off an ellipsoidal reflector of foci S (source) and G (geophone) emerge on the segment $[SG]$. Therefore, the DMO operator is really a 2-D operator working along the source-geophone line, even in a 3-D space. Consequently, applying the operator in three dimensions is not much different than in two dimensions except that the trace smearing is performed for an irregular spatial sampling according to the azimuth. The technique used in this 3-D DMO code consists of computing the bins affected by the segment $[SG]$. More precisely, whenever the center of a bin is closer to the $[SG]$ segment than half the bin size, the bin receives an output trace. This operation is repeated for all input traces, gradually filling the output space. This technique is equivalent to the nearest neighbor interpolation in space and linear interpolation in time described by Nichols (1993). This algorithm requires some evenly distributed data so that the fold over any bin is nearly constant. Dividing the trace amplitude by the fold of the corresponding bin seems an attractive solution, but it may spoil the amplitude properties of the data.

Conclusion

I implement a three-dimensional integral dip moveout processing algorithm for constant-velocity media that prevents operator aliasing. The study of amplitudes led me to choose Black's formulation as the most satisfactory in regard to three rules that assure the proper behavior of the DMO process. The method also corrects for spherical spreading. Although the program has not been tested on real 3-D data, the results with synthetic data, especially regarding amplitude restoration, are encouraging. The next step consists of implementing the algorithm on the Connection Machine.

A PARALLEL IMPLEMENTATION OF KIRCHHOFF DMO

In this section I compare two parallel algorithms for three-dimensional Kirchhoff dip moveout in a constant velocity medium. In the case of 3-D land data, an algorithm where data are processed in time slices allows highly irregular offset geometry. This mapping of the data into processor memory minimizes the cost of communication, which is reduced to nearest neighbor communication of time slices, and achieves load balance, keeping 80 percent of the processors busy throughout the process. The time aliasing of the operator is solved by a spatial convolution with dip-dependent triangles in two dimensions and with dip-dependent pyramids in three dimensions.

Introduction

Over the last years, DMO has become a standard step in seismic processing flows. The main reason for this is that it clearly improves the stack for little extra computational cost. Recently,

some research showed that the improvement is even more distinct when depth-variable velocity is considered in the DMO process (Godfrey, 1992; Artley, 1992; Meinardus and Schleicher, 1993). However, these new methods are computationally expensive and a constant-velocity DMO is often enough to give an idea of the geological structures involved or improve the velocity analysis. Therefore, a parallel implementation of constant-velocity DMO is natural to reduce the processing time cost. In a 3-D constant-velocity medium, the DMO process uses a line operator (Hale, 1991). It convolves the input data with elliptical impulse responses, and sums the convolved data into the final stacked volume. The single dimension of the operator as well as its dip limitation are two reasons to consider 3-D constant-velocity DMO as a fast process whatever the geometry of the data acquisition. Unfortunately, the multi-azimuthal distribution of 3-D land data is a major obstacle for parallel implementation (Biondi and Moorhead, 1992). This section details two possible parallel algorithms for 3-D constant-velocity DMO. The first algorithm involves a spiral data movement that does not use the full computational capabilities of the Connection Machine (CM5). The second algorithm reduces the communication cost by processing time slices but requires additional care to avoid temporal aliasing of the operator.

First algorithm: DMO by spiral data movement

The parallel implementation of DMO in a two-dimensional space is straightforward. Biondi (1991) describes a simple algorithm where the traces are laid out local to each processor. The DMO process then consists of a trace stretch and a nearest neighbor shift along the midpoint axis. In three dimensions, the data shift is not restricted to a single axis but requires a movement of the input traces across a two-dimensional space. Biondi (1992) uses the same layout to apply DMO to 3-D marine data. The algorithm takes advantage of the regular data movement in the in-line direction because of the uniform sampling of the shot locations. However, 3-D land surveys record shot profiles of varying geometry (an example is shown in Figure (9)), and thus, the shot axis can not be processed in parallel.

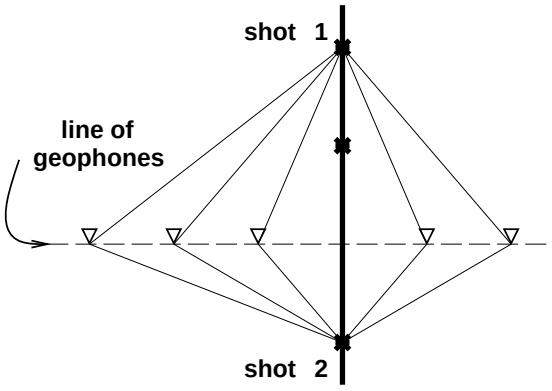

Figure 9: This simple 3-D land acquisition geometry exhibits the non-recurrence of the shot profiles. As opposed to marine surveys, the geophone line does not move along with the shot location. patrick-land3d [NR]

Memory layout

Since the irregularity of 3-D land data appears in both azimuth and offset distributions, there is no preferential direction for data movement. Hence, Biondi had the idea of a spiral data movement alternating the shifts along the x- and y- axes. The algorithm can be outlined as follows:

1. The data are sorted by offset length into classes.
2. For each bin of the output space, the trace of the current class whose midpoint falls into the bin is loaded in-processor.
3. The DMO stretch is performed in parallel over output space.
4. The traces are moved to the neighboring processor according to a spiral pattern illustrated in Figure (10).

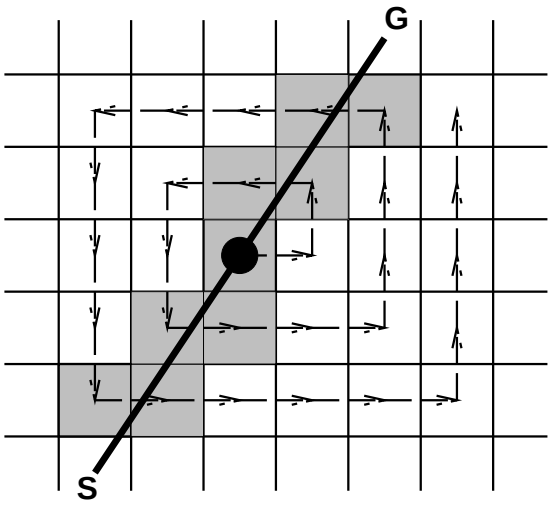

Figure 10: Spiral pattern for data movement over output space. The shaded squares represent the processors that actually perform the stretch and the stack to the output. The white squares represent the processors that stay idle during the movement of the trace. patrick-spiral [NR]

Load balance

By sorting the data into classes of offset, we can restrict the size of the spiral to the maximum offset of each class. The spiral data movement allows any azimuth distribution of the input traces. However, because a trace contributes to the stack only along its original source-geophone line, some processors are left idle during the spiral trip of the trace. Figure (10) shows the working processors in grey and the idle processors in white. In this example, the computer load, defined as the ratio of the number of working processors to the total number of processors, is thirty percent. For larger offsets, the load balance is expected to be worse because the number of working processors increases in proportion to the offset length whereas the total number of processors increases in proportion to the square of the offset. The unoptimized use of processing power makes the algorithm inefficient for 3-D land data.

Second algorithm: DMO by time slice

In order to spread the data along an elliptical path in (t,x,y) space, it is possible to either move spatially from bin to bin and stretch the traces or move the time slices upward and stretch the data in the offset direction. **Memory layout**

The memory layout for processing time slices is illustrated in Figure (11). The time slices are local to each processor and the time axis is the parallel dimension. Thus, the number of processors needed is the number of time samples, and the memory of each processor must be large enough to load and process one time slice. When the amount of data exceeds the memory available, the process can run on blocks of data. The blocks may be pieces of data cut in the (x,y) space, preserving the parallel axis and relieving the processors memory, or they may be cut in time, shortening the parallel axis but requiring a smaller area of overlap between successive lumps. Unlike the previous algorithm, data communication is performed in a single direction, up the time axis, and does not depend on the offset and azimuth distribution. The long ranging and chaotic communications in the (x,y) space take place in-processor. Thus, the processing of time slices results in a more efficient inter-processor communication than the trace processing. **Load balance**

Because the DMO elliptic operator is dip-limited, it does not extend all the way up to the Earth's surface ($t=0$). The time spread of the impulse response is given by:

$$\Delta t = t_n \left(1 - \frac{1}{\sqrt{1+\left(\frac{t_m}{t_n}\right)^2}} \right), \tag{8}$$

where t_n is the time location of the impulse in the input space, h is the half offset, v is the velocity of the medium, and $t_m = 2h/v$ is the horizontal two-way traveltime between shot and midpoint. Figure (12) is a plot of the time spread as a function of NMO time. An interesting feature of these curves is that the time spread of the impulse responses never exceeds μt_m where μ is

$$\mu = \sqrt{G}\left(\frac{G-1}{G+1}\right) \simeq .300283106, \qquad \text{with } \mathrm{G} = \frac{1+\sqrt{5}}{2}. \tag{9}$$

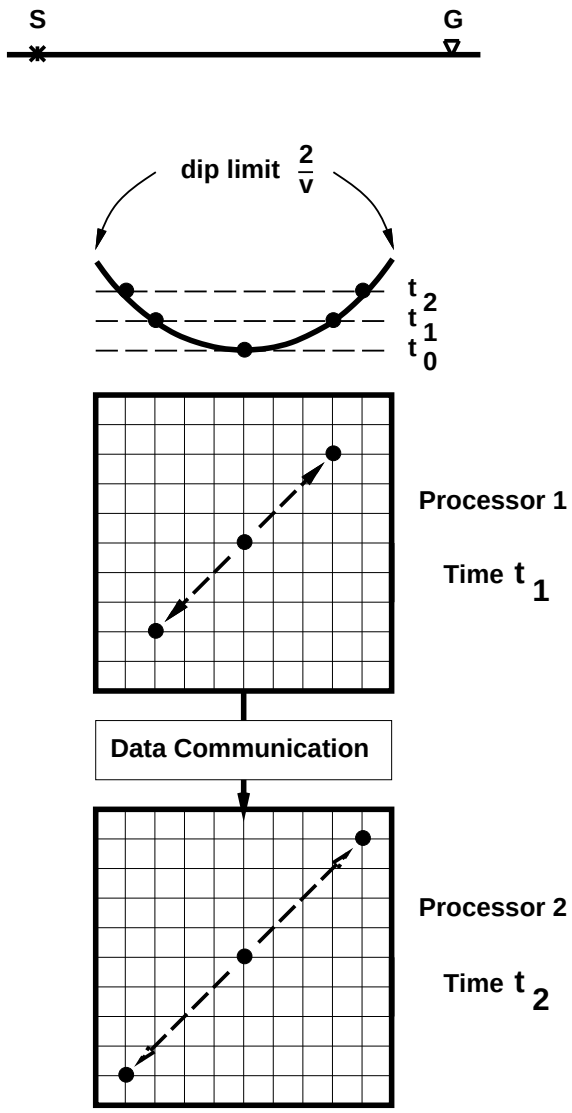

Figure 11: The top-most drawing represents the elliptic dip-limited DMO operator. I assume that the offset line bisects the x- and y- axes on the Earth's surface. Below, the two grids represent the data layout inside the processors. Each processor contains a time slice of data. Processor 1 contains the time slice at $t = t_0$ and performs the data communication across the (x, y) space that corresponds to a vertical shift from t_0 to t_1. During this time, processor 2 performs the same kind of operation for a vertical shift from t_1 to t_2. Then, processor 1 sums its output in the output volume and communicates its input to processor 2. The action is then repeated, moving the data across the (x, y) space for a vertical shift from t_0 to t_2. patrick-tsproc [NR]

../patrick/./Fig/magic.pdf

Figure 12: Time spread of the impulse responses of DMO as a function of the impulse location, t_n. The maximum time spread occurs for the input time $t_n = t_m/\sqrt{G}$ where G is the golden number. Click on the following button to see the curve for different times t_m.

Because t_m is limited by the maximum offset, the time spread is much shorter than the full trace length. During the process, the time slices are shifted upward until they reach the maximum time spread. Of course, only the time slice corresponding to the maximum time spread will have to be processed all the way. Other time slices, like for example $t_n = 5$ seconds (Figure (12)), will be processed for the first .1 second and then pass through idle processors. Obviously, the later time slices require less processing than the earlier ones, and thus represent a waste of processing capacity. The following formula gives the load balance as a function of trace length:

$$\text{Load}(t_{\max}) = \frac{1}{t_{\max}} \int_0^{t_{\max}} \frac{\Delta t(t)}{\Delta t_{\max}} dt. \tag{10}$$

This formula is derived from the two following observations. The time for which the processors are active is proportional to the area under the curve representing $\Delta t(t)$ in Figure (12), $\int \Delta t(t) dt$. The total computation time is proportional to both the trace length and the maximum time shift, $t_{\max} \Delta t_{\max}$. The computer load is then the ratio of the effective working time to the total execution time, as expressed in equation (10). As an example, for a trace length of 4 seconds and a midpoint time $t_m = 1$ second, the load balance reaches seventy percent (Figure (13)). This algorithm allows a more efficient distribution of work between processors

Figure 13: Load balance as a function of the trace length. The optimal load balance of eighty percent corresponds to a trace length which is a function of t_m ($= 2h/v$). The bigger t_m is, the later the load balance is optimal. Click on the following button to see a movie of the load balance for different values of t_m.

than the spiral trace processing described earlier. I implemented this algorithm for a two-dimensional model (Figure (14) is an output of the program) but the run time is similar to a serial implementation of DMO (in 2-D, trace processing is more straightforward than time-slice spreading). However, the real advantage of the method will appear in processing 3-D land data.

The pyramid as an anti-aliasing structure

Applying DMO in time slices assures that the operator will not introduce spatial aliasing in the data. However, the spatial spreading from one time slice to the next must never exceed the bin spacing or temporal aliasing will occur, especially for the gentle dips of the operator (Hale, 1991). A technique described by Claerbout (1992) prevents operator aliasing. The method consists of the convolution of the operator with triangles whose width depends on the dip of

the operator as described previously. However, this process differs from the trace-oriented process in that the convolution is applied horizontally in the time slices. Figure (14) shows the DMO response of several impulses corrected with anti-aliasing triangles. In three dimen-

Figure 14: Impulse responses of DMO applied to time slices. The anti aliasing triangles are wide at gentle dips and shrink towards the steepest dips.

sions, the triangles cannot be directly obtained by Claerbout's double integration technique. Instead, a similar method leads to the construction of pyramids in a three-dimensional extension of the concept of the triangle. Figure (15) shows how to decompose the construction of a pyramid. The pyramids cause a lateral expansion of the operator that does not agree with the theoretical ellipse of dip moveout. However, the operator has a cross-line component when we consider variable velocity or transversely isotropic media. Gonzalez (1992) showed that the three-dimensional DMO operator in the cross-line direction may curve either downward when the velocity increases with depth or upward when the velocity increases with angle in a transversely isotropic medium. In both cases, the first derivative of the operator with respect to the cross-line coordinate vanishes on the in-line axis. Therefore, the horizontal expansion caused by the pyramids is a first-order approximation of the operator.

Conclusion

Because constant-velocity DMO in a 3-D Earth model operates along curves, care must be taken in parallel implementation in order to avoid the waste of computational capacity. Trace processing algorithms require a great deal of inter-processor communications and leave many processors idle during the process. On the other hand, the processing of time slices is well adapted to the irregular geometry of 3-D land data, attaining a load balance of eighty percent. Although the anti-aliasing convolution is more quickly performed along traces, the efficient use of computational capacity compensates for a slower convolution in the time slices. Unfortunately, the three-dimensional implementation of the time-slice algorithm has not been tested on real data. Doing so would tell us whether the lateral expansion of the operator is an undesirable artifact or a step toward the variable-velocity assumption.

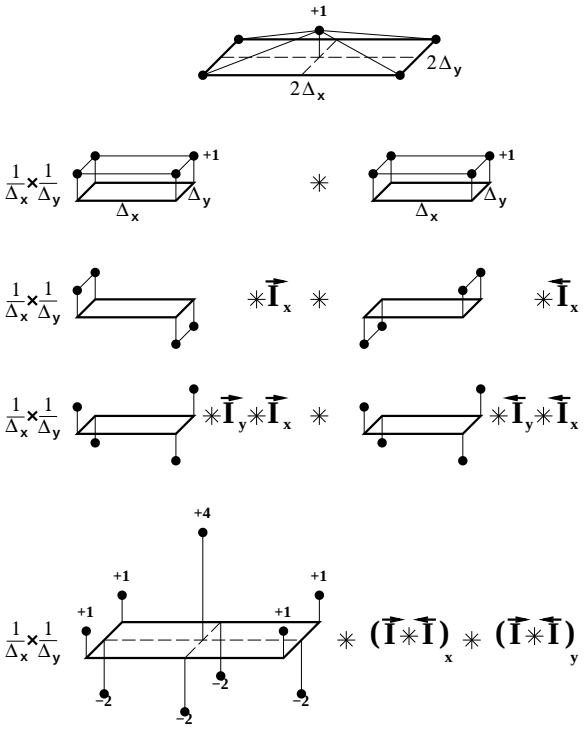

Figure 15: Decomposition of the method for building pyramids. All lines are equivalent: a pyramid function is the convolution of two box functions (line 1 to 2); a box function is the x-integration of an x-opposite-spike square signal (line 2 to 3); an x-opposite-spike square signal is the y-integration of an x-y-opposite-spike square signal (line 3 to 4); the convolution is commutative (line 4 to 5); the convolution of two x-y-opposite-spike square signals is the final nine-spike signal (line 5 to 6). The dimensions Δ_x and Δ_y of the pyramid depend on both time and space variables. However, the causal ($\vec{\mathbf{I}}_x$, $\vec{\mathbf{I}}_y$) and anticausal ($\overleftarrow{\mathbf{I}}_x$, $\overleftarrow{\mathbf{I}}_y$) integrations are independent of Δ_x and Δ_y and can be applied only once after all convolutions are performed. patrick-pyramid [NR]

Chapter 2

Part II: Attaining accuracy with depth-variable velocity dip moveout

Nowadays, a standard industrial seismic processing almost always involves the constant-velocity dip moveout correction. Two main features make the process attractive. First, the computational cost is low compared to prestack migration in both two-dimensional and three-dimensional processings. Secondly, because constant-velocity dip moveout is "independent of velocity", it may come before velocity analysis, removing the effects of dip (Forel and Gardner, 1988). However, the hypothesis of constant velocity is somewhat inconsistent with the concept of velocity analysis which results in the construction of a depth-variable velocity model. Recently, several methods for depth-variable velocity dip moveout (Godfrey, 1992; Artley, 1992; Meinardus and Schleicher, 1993) undeniably improved the zero-offset stack section. However, one can wonder if they clearly improve the result of the post-stack migrated section.

HOW VARIABLE VELOCITY DIP MOVEOUT IMPROVES POST-STACK MIGRATION

In this section I show that the depth-variable velocity dip moveout strongly differs from the constant-velocity dip moveout not by the shape of the operator (although the operator is squeezed horizontally when the velocity increases with depth (Hale, 1983b; Artley, 1992)) but by the amplitude distribution along the operator. In the second section, a synthetic data example shows how the processing flow NMO-$v(z)$ DMO-post-stack migration is comparable to pre-stack migration.

Some amplitude issues

Jakubowicz (1990) gave an elegant formulation of $f-k$ dip moveout where a finite number of dips are processed separately and stacked according to equation (1),

$$P_{\text{DMO}} = \sum_i \Delta D_i \; P_{D_i}, \tag{1}$$

where P_{D_i} is the wavefield, NMO-corrected for a single dip D_i, and ΔD_i is the width of the dip-range surrounding D_i. Obviously, if the dip sampling is irregular, the dip-decomposed wavefields stack with a different weight. Beasley (1988) derived an amplitude scheme for Kirchhoff dip moveout from Jakubowicz's weighting by using the following practical approximation:

$$\Delta D_i \approx \frac{\partial^2 t_0}{\partial x^2} \Delta x. \tag{2}$$

To the first order, the amplitude along the dip moveout operator is proportional to its time curvature. This approximation is close to the amplitude scheme mathematically derived by Black *et al.* (1993) from (f,k) dip moveout (Blondel, 1993). Thus, Jacubowicz's dip moveout by dip-decomposition, Black's Kirchhoff dip moveout, and (f,k) dip moveout have a very similar amplitude distribution. In the general case of depth-variable velocity, there is no analytic expression of the amplitude along the operator but the dip-decomposition method gives us a qualitative idea. Indeed, the operator of variable-velocity dip moveout may be build point by point for a range of dips regularly sampled. Then, a region where the points are close to each other allows a significant stack of the dipping segments of the operator, and thus corresponds to an area of high amplitude. Figure (2) shows an impulse response of dip moveout in a time-variable velocity model whose profile is represented in Figure (1). Artley's (1992) $v(z)$ dip

Figure 1: Interval velocity model.

moveout method which produced Figure (2), requires a non-linear inversion of a 4×4 system of equations for each time, offset, and dip we consider. Though it is a time consuming process, it provides us with amplitude information. First, the triplication of the impulse response have a low energy for this velocity model (a trough in the velocity profile would show a high amplitude triplication). Secondly, the amplitude varies a lot along the operator, starting high

Figure 2: Impulse response of $v(z)$ 2-D DMO for an offset of .8 km and a total two-way traveltime of 1. s. The stars along the operator correspond to a uniform sampling of dips. The denser are the points, the higher is the amplitude of the operator. The dashed line represents the elliptic support of the constant-velocity DMO impulse response for the same offset and traveltime. Both the solid and dashed lines have a similar shape in the region of high amplitudes, differing only by the triplications along the solid line. The unrealistic points beyond the offset show that there is a problem of convergence of the $v(z)$ DMO algorithm for steep dips.

at gentle dips, decreasing, and increasing again just before a triplication. The shape of a $v(z)$ DMO operator differs from the constant-velocity DMO operator essentially by a number of triplications along its branches (Figure (2)). However, since these triplications are generally low-amplitude, the shape is not the most discriminative feature of the operator. The amplitude variations along the operator strongly depend on the velocity model, and therefore play an important role in the dip moveout correction.

Comparison with constant-velocity dip moveout

There is no doubt that replacing the constant-velocity dip moveout step by a variable-velocity dip moveout step improves the zero-offset section after stack. However, the final result of a standard processing flow is the post-stack migrated section. Thus, the idea of comparing the pre-stack migrated section to the post-stack migrated sections with and without the constant velocity assumption is natural. David Lumley provided the synthetic data example that appears in the next three figures. This synthetic model includes four diffractors at various depth and midpoint positions, two horizontal events, and a 45-degree dipping event, overlaid by a strong, constant RMS-velocity gradient ($V_{\text{RMS}}^{(\text{km/s})} = 1.5 + 2.\tau^{(\text{s})}$). **Results of constant-velocity DMO**

The processing flow involving a constant-velocity dip moveout step is displayed in Figure (3). The three deepest diffractors are unfocused whereas the shallow diffractor is focused but low-amplitude. The horizontal reflectors are well imaged but the dipping event has a lower amplitude than expected though it is not mispositioned. **Results of approximate variable-**

Figure 3: Post-stack migrated section after NMO, constant-velocity DMO and stack. The deepest diffractors are unfocused and the dipping event is low-amplitude.

velocity DMO

Castle (1993) proposes a method conceptually similar to Hale's squeezed dip moveout operator (1983b; 1993) but different in its implementation. Instead of applying a standard normal moveout correction ($t_n^2 = t^2 - x^2/v_{\text{RMS}}^2$) followed by a dip moveout correction with a squeezed operator, the method uses de Bazelaire's (1988) normal moveout correction with shifted hyperbolae:

$$t_p^2 = (t - t_n + t_p)^2 - \frac{x^2}{v^2}, \tag{3}$$

where $t_p = t_n/S$, $v^2 = S\mu_2$, S is the squeezing factor given by $S = \mu_4/\mu_2^2$, and μ_k is the order-k momentum ($\mu_2 = v_{\text{RMS}}^2$). Castle shows that applying the constant-velocity dip moveout operator in the t_p time domain and then shifting back the data according to the equation

$$t_0 = t_p S \tag{4}$$

is equivalent to squeezing the dip moveout operator. Both methods approximate the exact dip moveout operator for depth-variable velocity by wiping out the triplication apparent in Figure (2). Figure (4) results from post-stack migration after the approximate dip moveout step. The four diffractor points are better focused with this flow than with the flow that involves constant-velocity dip moveout. Unfortunately, the method, apart from approximating the shape, also simplifies the amplitude distribution along the operator. The dull appearance of the reflectors is a consequence of this improper handling of the amplitudes.

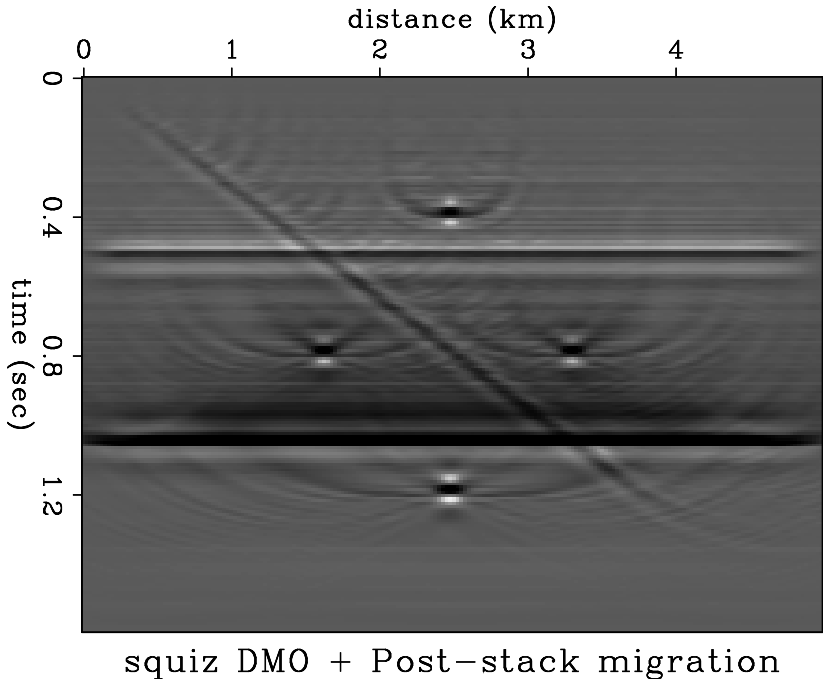

Figure 4: Post-stack migrated section after NMO, approximate variable-velocity DMO and stack. The diffractors and the dipping event are better focused than in Figure (3) but have unbalanced amplitudes. patrick-svmig [ER]

Result of variable-velocity DMO

For this processing flow, I use the variable-velocity DMO algorithm developed by Artley (1992). The method, accurate for any depth-variable velocity media, solves a system of equation accounting for the location and the dip of any reflector point. An extension of this system to three dimensions is proposed in the next section (page 25). The similarity between the pre-stack migration (Figure (6)) and the post-stack migration after depth-variable velocity dip moveout Figure (5) is not surprising because the two processing are equivalent when there is no lateral velocity variation. However, the clear improvement with respect to the standard processing makes the variable-velocity dip moveout a very useful tool.

Figure 5: Post-stack migrated section after NMO, exact variable-velocity DMO and stack. The diffractors and the dipping event are well imaged.

Conclusion

The main difference between constant-velocity and variable-velocity dip moveout is not the shape of the operator but the amplitude distribution. Because this distribution highly depends on the velocity model, variable-velocity dip moveout considerably improves the post-stack migration. However, the drawback is its high computational cost. For example, Artley's method is more costly than a pre-stack migration, because the algorithm does much more than compute the zero-offset traveltime: it performs a ray-based pre-stack migration and uses the computed zero-offset traveltime to apply the dip moveout correction. Because both Meinardus's and Artley's algorithms are based on the dip-decomposition idea, it is possible to reduce the computational cost by cutting down the number of dips to process. On the other hand,

Figure 6: Pre-stack migration. Click on the following button to see a movie of the four previous pictures.

Castle's method differs from the constant-velocity dip moveout flow by involving an extra time-variable shift of the data. At a small additional computational cost, the approximate dip moveout flow improves the quality of imaging after post-stack migration. Furthermore, this method can be adapted in three dimensions by using the constant-velocity algorithm described in the first part (page 14). Apart from all speed considerations, a precise dip moveout method requires both assumptions of a depth-variable velocity and a three-dimensional model. The next section explains how Artley's two-dimensional depth-variable velocity dip moveout method (1992) can be extended to three dimensions.

EQUATIONS FOR THREE-DIMENSIONAL DIP MOVEOUT IN DEPTH-VARIABLE VELOCITY MEDIA

Artley (1992) introduced an original method to perform $v(z)$ dip moveout in a two-dimensional earth model. The process uses ray tracing tables and solves a system of equations accounting for the location and the dip of the reflection point. This section describes an extension of Artley's method and derives a new set of equations for the 3-D case.

The ray parameter in three dimensions

In a two-dimensional earth model, the ray parameter is given at any point of the ray by the relationship

$$p = \frac{\sin\theta}{v}, \tag{5}$$

where θ is the inclination of the ray path with respect to the vertical axis and v is the local velocity. We can see that the ray parameter is simply the projection length of the ray path vector $\mathbf{r}$ ($\|\mathbf{r}\| = r = 1/v$) on the earth surface. In a 3-D v(z) model of the earth, the rays travel in a vertical plane. In this case, the ray parameter, being the projection of the ray path vector, is a two-dimensional vector that can be expressed in either cartesian or polar coordinates, as follows:

$$\mathbf{p} = \begin{pmatrix} p_x \\ p_y \end{pmatrix} = p \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix}, \tag{6}$$

where ϕ is the strike of the vertical plane containing the ray and $p = \sqrt{p_x^2 + p_y^2}$.

Deriving the system of equations

Vectorial relationships on the earth's surface

In the triangle (PSG) in Figure (7), the following expression relates the horizontal traveling of the source and geophone rays with the offset:

$$\overrightarrow{SG} = \overrightarrow{SP} + \overrightarrow{PG}\,. \tag{7}$$

The vectors $\overrightarrow{PS}$ and $\overrightarrow{PG}$ are related to the table that gives the lateral distance traveled by the ray as a function of the ray parameter and the travel time, $\xi(p,t)$[1]. The projection on the earth's surface of a ray path (of parameter p and traveltime t) is the 2-D vector $\xi(p,t)\frac{\mathbf{p}}{p}$. Thus, relation (7) yields

$$\mathbf{x}_g - \mathbf{x}_s = \xi(p_g,t_g)\frac{\mathbf{p}_g}{p_g} - \xi(p_s,t_s)\frac{\mathbf{p}_s}{p_s}, \tag{8}$$

where $\mathbf{x}_s$ and $\mathbf{x}_g$ are the 2-D vectors of the source and geophone coordinates.

A second relation expresses the coordinates of the point of emergence of the zero-offset ray, E. In the triangle (PEM), we have :

$$\overrightarrow{ME} = \overrightarrow{MP} + \overrightarrow{PE}\,. \tag{9}$$

Again, the vector $\overrightarrow{PE}$ is related to the table of lateral distances, $\xi(p,t)$, yielding

$$\mathbf{x}_0 = -\frac{1}{2}\left(\xi(p_s,t_s)\frac{\mathbf{p}_s}{p_s} + \xi(p_g,t_g)\frac{\mathbf{p}_g}{p_g}\right) + \xi(p_0,t_0)\frac{\mathbf{p}_0}{p_0}, \tag{10}$$

where $\mathbf{x}_0$ is the 2-D vector of the emergence point coordinates. **Time relationships in the**

[1]The ray tracing tables are fully described in Artley (1992)

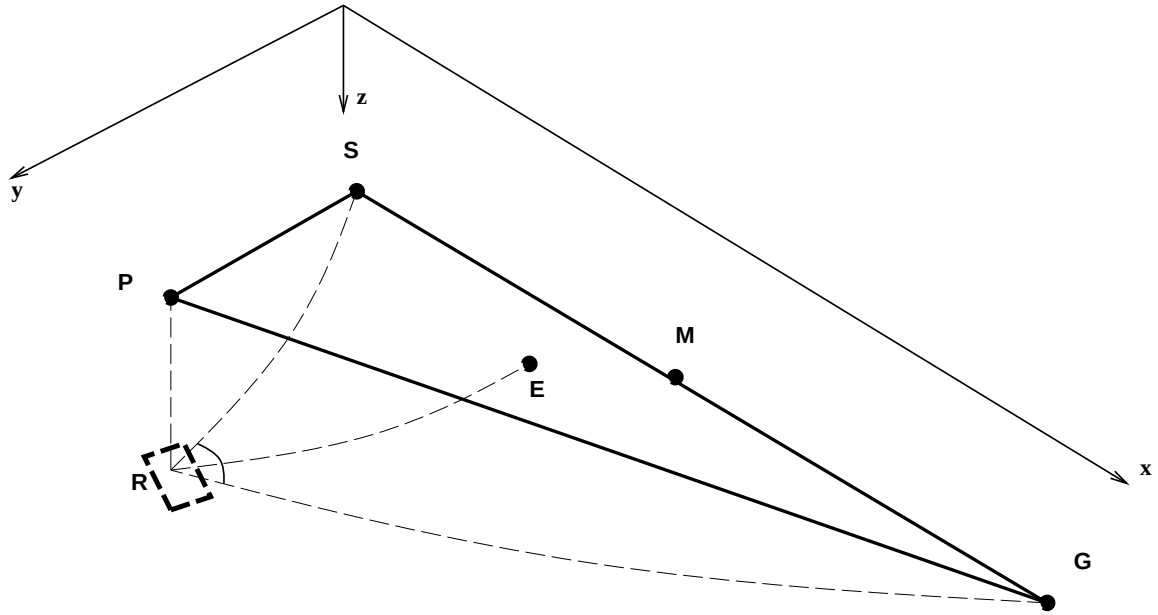

Figure 7: Three-dimensional view of the source, receiver, and reflection points for 3-D v(z) DMO. The dashed lines represent the ray paths in the earth, and the bold solid lines represent the distances on the surface of the earth. patrick-Rayp3d [NR]

earth's interior

The following obvious equation relates the traveltimes along the ray paths to source and geophone, t_s and t_g, to the given total traveltime t_{sg}:

$$t_s + t_g = t_{sg}. \tag{11}$$

We can also state two more equations to account for the fact that the three ray paths, namely the zero-offset, source, and geophone ray paths, end at the same point in depth (R):

$$\tau(p_0, t_0) = \tau(p_s, t_s) = \tau(p_g, t_g), \tag{12}$$

where $\tau(p,t)$ is the table of vertical traveltimes. **The relationship between the ray paths** At the reflector point R, the ray must obey Snell's law. In other words, the angle of incidence must equal the angle of reflection. In terms of ray paths, $\mathbf{r}_0$ fixes the position of the reflection plane, and $\mathbf{r}_s$ and $\mathbf{r}_g$ account for the angles of incidence and reflection. Since $\mathbf{r}_s$ and $\mathbf{r}_g$ have equal lengths $1/v$, Snell's law can be expressed in the following vectorial form:

$$\mathbf{r}_0 = \lambda(\mathbf{r}_s + \mathbf{r}_g), \tag{13}$$

where λ is a coefficient of proportionality. The first two coordinates are the ray parameter components, which have the following relation:

$$\mathbf{p}_0 = \lambda(\mathbf{p}_s + \mathbf{p}_g). \tag{14}$$

The third coordinate of the ray path vectors can be expressed as a function of the inclination angle and the vector length, $1/v$, as follows:

$$r_z = \frac{\cos\theta(p,t)}{v}, \tag{15}$$

where $\theta(p,t)$ is the table of angles along a ray of parameter p and time t. Then, after simplification of the $1/v$ factors (the velocity is the same for all rays at the reflector point), the third equation of relation (13) becomes

$$\cos\theta(p_0,t_0) = \lambda(\cos\theta(p_s,t_s) + \cos\theta(p_g,t_g)). \tag{16}$$

Substituting equation (16) into relation (14), we can eliminate the proportionality factor λ to obtain the following relation:

$$\mathbf{p}_0\left(\cos\theta(p_s,t_s) + \cos\theta(p_g,t_g)\right) = (\mathbf{p}_s + \mathbf{p}_g)\cos\theta(p_0,t_0). \tag{17}$$

Sets of equations and unknowns Our system of equations is constituted by collecting relations (8), (10), (11), (12), and (17):

$$\left\{\begin{array}{ccc}
\xi(p_g,t_g)\frac{\mathbf{p}_g}{p_g} - \xi(p_s,t_s)\frac{\mathbf{p}_s}{p_s} & = & \mathbf{x}_g - \mathbf{x}_s \\
\xi(p_0,t_0)\frac{\mathbf{p}_0}{p_0} - \frac{1}{2}\left(\xi(p_s,t_s)\frac{\mathbf{p}_s}{p_s} + \xi(p_g,t_g)\frac{\mathbf{p}_g}{p_g}\right) & = & \mathbf{x}_0 \\
t_s + t_g & = & t_{sg} \\
\tau(p_0,t_0) & = & \tau(p_s,t_s) \\
\tau(p_0,t_0) & = & \tau(p_g,t_g) \\
\mathbf{p}_0\left(\cos\theta(p_s,t_s) + \cos\theta(p_g,t_g)\right) & = & (\mathbf{p}_s + \mathbf{p}_g)\cos\theta(p_0,t_0)
\end{array}\right. . \tag{18}$$

Because the **x** and **p** vectors are two-dimensional, the first, second, and sixth relations of system (18) give 3×2 equations, yielding a set of nine equations. The unknowns are $\mathbf{p}_s$, $\mathbf{p}_g$, t_s, t_g, t_0, and $\mathbf{x}_0$. As described in the preamble, the **p** vectors have only two unknown parameters, p_x and p_y (or p and ϕ). Similarly, $\mathbf{x}_0$ is a two-dimensional unknown vector. Therefore, system (18) relates nine unknowns with the known parameters $\mathbf{p}_0$, t_{sg}, $\mathbf{x}_s$, $\mathbf{x}_g$, and the ray tracing tables $\xi(p,t)$, $\tau(p,t)$, and $\theta(p,t)$. **Solving the system**

The second relation of system (18) can be isolated because it simply adds two relations and two unknowns (the coordinates of $\mathbf{x}_0$). Similarly, the third equation can be substituted into the others by replacing t_s (or t_g) with $t_{sg} - t_g$ (or $t_{sg} - t_s$).

Thus, we obtain a reduced system of four relations (six equations):

$$\left\{\begin{array}{ccc}
\xi(p_g,t_g)\frac{\mathbf{p}_g}{p_g} - \xi(p_s,t_{sg}-t_g)\frac{\mathbf{p}_s}{p_s} & = & \mathbf{x}_g - \mathbf{x}_s \\
\tau(p_0,t_0) & = & \tau(p_s,t_{sg}-t_g) \\
\tau(p_0,t_0) & = & \tau(p_g,t_g) \\
\mathbf{p}_0\left(\cos\theta(p_s,t_{sg}-t_g) + \cos\theta(p_g,t_g)\right) & = & (\mathbf{p}_s + \mathbf{p}_g)\cos\theta(p_0,t_0)
\end{array}\right. . \tag{19}$$

and two additional relations to compute the remaining unknowns:

$$\mathbf{x}_0 = \xi(p_0,t_0)\frac{\mathbf{p}_0}{p_0} - \frac{1}{2}\left(\xi(p_s,t_s)\frac{\mathbf{p}_s}{p_s} + \xi(p_g,t_g)\frac{\mathbf{p}_g}{p_g}\right) \tag{20}$$

$$t_s = t_{sg} - t_g. \tag{21}$$

The six unknowns of system (19) are t_0, t_g (or t_s), $\mathbf{p}_s$ [including the two unknowns (p_{xs}, p_{ys}) or (p_s, ϕ_s)], and $\mathbf{p}_g$ [(p_{xg}, p_{yg}) or (p_g, ϕ_g)]. The parameters are t_{sg}, $\mathbf{p}_0$ [including the two parameters (p_{x0}, p_{y0}) or (p_0, ϕ_0)], $\mathbf{x}_s$, and $\mathbf{x}_g$. This system, like its 2-D counterpart, can be solved using the Newton-Raphson algorithm (Press et al., 1986).

Agreement with the equations in two dimensions

The 2-D system derived by Artley (1992) is a set of four equations with four unknowns. It can be retrieved by replacing the vectorial ray parameters with their scalar form (since the vectors are now in a common plane). The first relation of system (18) then becomes

$$\xi(p_g, t_g) - \xi(p_s, t_{sg} - t_g) = 2h, \tag{22}$$

where h is the half offset. The fourth relation becomes

$$\frac{\sin\theta(p_0,t_0)}{v}\left(\cos\theta(p_s,t_s)+\cos\theta(p_g,t_g)\right) = \left(\frac{\sin\theta(p_s,t_s)}{v}+\frac{\sin\theta(p_g,t_g)}{v}\right)\cos\theta(p_0,t_0) \tag{23}$$

and can be simplified to

$$\sin(\theta(p_0,t_0)-\theta(p_s,t_s)) = \sin(\theta(p_g,t_g)-\theta(p_0,t_0)). \tag{24}$$

Finally, we obtain the bisection condition as shown by Artley (1992), as follows:

$$\theta(p_0,t_0) = \frac{1}{2}\left[\theta(p_s,t_s)+\theta(p_g,t_g)\right]. \tag{25}$$

Conclusion

We[2] derived a set of equations that constrains the travel path of a ray in a three-dimensional earth model with a depth-variable velocity. This system can be solved for the zero-offset traveltime and the emergence point of the ray, yielding an efficient method for dip moveout processing. A successful implementation of the equations have been realized by Artley at the Colorado School of Mines, showing the expected saddle-like impulse response.

CONCLUSIONS

The dip moveout correction requires important compromises between speed and precision in two- and three-dimensional models. The conclusions to part I (page 17) propose ways of increasing the speed. The constant-velocity assumption allows the dip moveout process to be fast in both two- and three-dimensional model geometries. Some additional attributes of

[2] *We* refers to the co-authors of the paper (Artley et al., 1993)

the operator, such as weighting and anti-aliasing schemes, reduce the amplitude artifacts in a computationally efficient manner. Moreover, parallel computing can speed up the process even in the case of three-dimensional land data where the acquisition geometry is irregular. A parallel implementation of integral dip moveout in time slices proves efficient for irregular azimuthal distribution of the data. The second part of this paper proposes two ways of improving precision. The first is to apply a proper weight along the operator, which is achieved at almost no extra computational cost. The second way is to consider depth-variable velocity. The dip moveout correction then becomes computationally expensive. However, an approximation valid for gently dipping reflectors allows the variable-velocity process to be almost as fast as the constant-velocity process. Aside from the considerations of speed, the method of three-dimensional dip moveout in depth-variable velocity developed in the last section may considerably improve the imaging of complex data structures. Throughout my analysis of the dip moveout correction, I have tried to optimize both speed and precision. Unfortunately, the very fast methods prove inaccurate, whereas the very precise ones are slow. The approximate methods related to the squeezing of the dip moveout operator turn out to be fast and rather precise, even in three dimensions. This result suggests a method for three-dimensional dip moveout processing: the operator with limited cross-line extension that I introduced in the first part can be squeezed, yielding a first-order approximation of the theoretic "saddle" operator.

ACKNOWLEDGMENTS

I would like to thank David Lumley for many useful discussions and for providing me with synthetic data and code. I am grateful to Mihai Popovici and Biondo Biondi for their efforts to transmit their knowledge of DMO and for their useful hints on parallel programming. I also thank Dave Nichols for his patience in teaching me the basis of parallel processing. Dimitri Bevc and David Lumley helped clarify for me the idea of anti-aliasing with triangle convolution. In addition, I would like to thank Craig Artley, Mihai Popovici, and Matthias Schwab for working with me on the derivation of the equations of 3-D $v(z)$ DMO. Finally, I am grateful to Professor Jon Claerbout and Stew Levin for providing an excellent environment for my research, and to the Total oil company for funding it.

REFERENCES

Artley, C., Blondel, P., Popovici, A. M., and Schwab, M., 1993, Equations for three dimensional dip moveout in depth-variable velocity media: SEP–**77**, 43–48.

Artley, C. T., 1992, Dip moveout processing for depth-variable velocity: Master's thesis, Colorado School of Mines.

Beasley, C. J., and Mobley, E., 1988, Amplitude and antialiasing treatment in $(x-t)$ domain DMO: 58th Annual Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1113–1116.

Biondi, B., and Moorhead, W. D., 1992, Data parallel algorithm for Kirchhoff 3-D DMO: 62nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 330–333.

Biondi, B., 1991, Wave equation algorithms on massively parallel computers: SEP–**70**, 59–72.

Black, J. L., Schleicher, K. L., and Zhang, L., 1993, True amplitude imaging and dip moveout: Geophysics, **58**, 47–66.

Blondel, P., 1993, Constant-velocity anti-aliasing three-dimensional integral dip moveout: SEP–**77**, 49–58.

Castle, R. J., 1993, A 2-D $v(z)$ DMO algorithm: 63nd Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1130–1133.

Claerbout, J. F., 1992, Anti aliasing: SEP–**73**, 371–390.

de Bazelaire, E., 1988, Normal moveout revisited - inhomogeneous media and curved interfaces: Geophysics, **53**, no. 2, 143–157.

Deregowski, S. M., and Rocca, F., 1981, Geometrical optics and wave theory of constant offset sections in layered media: Geophys. Prosp., **29**, no. 3, 374–406.

Forel, D., and Gardner, G. H. F., 1988, A three-dimensional perspective on two-dimensional dip moveout: Geophysics, **53**, no. 5, 604–610.

Gardner, G. H. F., and Forel, D., 1990, Amplitude preservation equations for dip moveout (short note): Geophysics, **55**, no. 4, 485–487.

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

Gonzalez, 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.

Hale, D., and Artley, C., 1993, Squeezing dip moveout for depth-variable velocity: Geophysics, **58**, 257–264.

Hale, I. D., 1983a, Dip moveout by Fourier transform: SEP–**36**, 5–34.

Hale, I. D., 1983b, Dip moveout for depth variable velocity: SEP–**36**, 35–52.

Hale, D., 1991, Course notes: Dip moveout processing: Soc. Expl. Geophys., Tulsa.

Jakubowicz, H., 1990, A simple efficient method of dip-moveout correction: Geophys. Prosp., **38**, no. 3, 221–246.

Liner, C., 1990, General theory and comparative anatomy of dip moveout: Geophysics, **55**, no. 5, 595–607.

Meinardus, H. A., and Schleicher, K. L., 1993, 3-D time-variant dip moveout by the *f-k* method: Geophysics, **58**, 1030–1041.

Nichols, D., 1993, Integration along a line in a sampled space: SEP–**77**, 283–294.

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