

Anisotropic pre-stack depth migration of complex geological structures

M. Graziella Kirtland Grech, P. F. Daley, and Don C. Lawton

ABSTRACT

An anisotropic pre-stack depth migration algorithm for *P-P* and *P-S* wave data has been developed and successfully tested on numerical and field data from the Rocky Mountain foothills of southern Alberta, Canada. The migration is diffraction stack type and uses an interface-based raytracer to compute traveltimes for traveltime table generation. When the exact migration-velocity model was known, the quality of the migrated image depended only on the robustness of the algorithm used to interpolate these ray-traced traveltimes onto a regular grid. Proper selection of the gridding routine resulted in smooth traveltime contours and good quality images that compare well with results obtained using industry standard production migration code.

INTRODUCTION

Standard 2D Born-Kirchhoff migration (Bleistein, 2000) usually starts with a smoothed velocity grid, which is assumed to approximate the velocity structure in the zone of interest. Ray fans are shot at a set of points along the surface of the model and the computed traveltimes through the model are placed on a grid, usually coincident with the velocity grid. Points that do not coincide with a grid node, but lie within a given aperture, are interpolated onto the traveltime grid using some form of nearest neighbour weighted-average.

In the migration developed in this work, the velocity model is defined by a number of polygons, each having a set of material properties. For the anisotropic case each polygon is assigned four parameters: the bedding-perpendicular velocity (V_0), the anisotropy parameters ε and δ (Thomsen, 1986), and the stratigraphic dip, θ , of the beds within each polygon, from which the anisotropy symmetry axis is then calculated. These properties can be different for each polygon. Since the ray tracer is interface based, ray fans are shot for a set of surface locations and the x - and z - coordinates and the traveltime (x,z,t) are recorded each time a ray crosses a boundary between two polygons. Additional points along a ray between polygon boundaries are also calculated to give a denser coverage of (x,z,t) points. Spacing of the extra points along a ray is a function of the frequency content of the data and the velocity within the particular polygon through which the ray is travelling. The raytracing approach uses first arrival times. The distribution of scattered points for each surface location is then interpolated onto a regular grid to generate the traveltime tables required by the migration program. The lateral and vertical grid spacing is specified by the user, with finer spacing resulting in better structural definition but also longer computation time. It has been found that the gridding algorithm used determines the smoothness of the traveltime contours and hence the quality of the migrated image. Consequently a robust gridding algorithm is essential for this method of traveltime table computation. The migration itself is pre-stack depth and diffraction-stack type. It can accomodate

vertical and tilted transverse isotropy (VTI and TTI), as well as converted waves and multiple mode conversions. For each time sample in an input trace, it sums the corresponding amplitude at matching traveltimes locations on the output depth grid, giving the migrated depth image. Migration can be carried out from topography or from datum. A version of the code for pre-stack depth migration of VSP data has also been developed.

This paper compares the migration results obtained using a standard library gridding routine with one that was designed specially for this work. The image quality obtained by using the latter routine for gridding was superior to that obtained when the library routine was used. Also, the resulting migrated sections compare well with those obtained using industry standard production migration code.

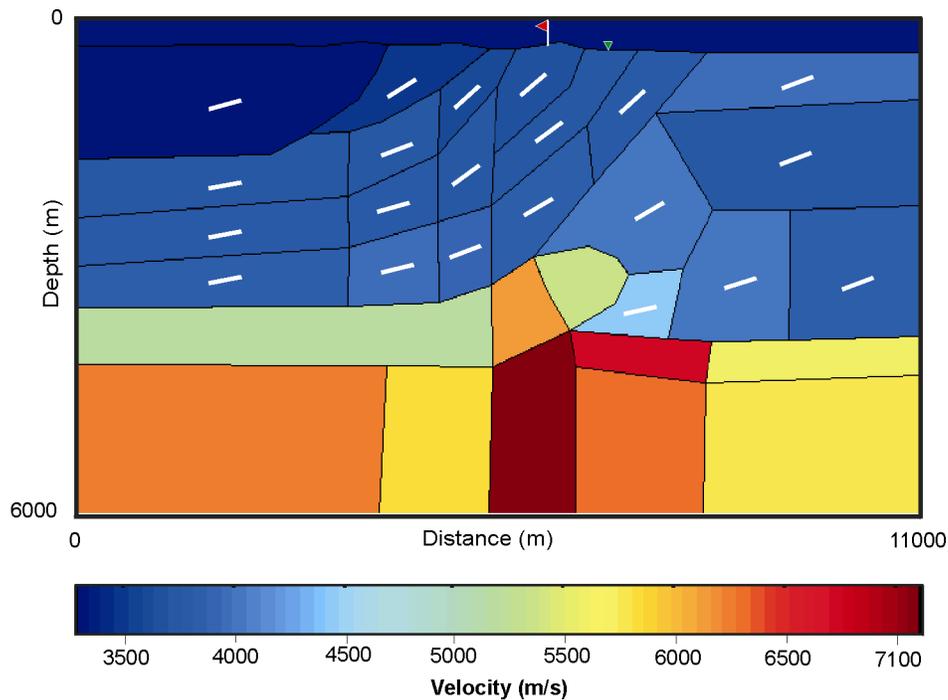


FIG. 1. A typical anisotropic migration-velocity model made up of a set of polygons. In this example, the model is colour-coded to represent the low velocity (V_0). The short white lines show the dip direction (θ) of the beds within each polygon that has anisotropic properties. A value of 0.1 for epsilon and 0.05 for delta was used for all the anisotropic blocks. The other polygons are isotropic, in which only the isotropic velocity needs to be specified.

METHOD

Gridding of traveltimes data using *Triangle*

The 2-D model space is assumed to be covered by homogeneous polygons of anisotropic composition with variable orientation of the anisotropic axes with respect to the model coordinates. This allows for both vertical and tilted transverse isotropy to be taken into account during the ray tracing. Rays fans are shot from points on the surface and the progress of each ray is determined at say N-1 time intervals yielding a

$3 \times N$ matrix, (x_i, z_i, t_i) , for each ray with x and z indicating the spatial position and t , the current ray travel time. If M rays are shot in each fan there could be a maximum of $3 \times N \times M$ scattered points in a specified subset of the model space, many of which are colinear. These scattered data points, however, have to be organised in some manner if they are to be used to create traveltimes tables for depth migration purposes. This usually takes the form of a rectangular $m \times n$ grid over the model space

$$\{(x_j, z_k, t_{jk}), j = 1, m, k = 1, n\}$$

After a number of directions for accomplishing this were investigated, it was decided that triangulation of the scattered points followed by a linear interpolation of the time values onto the uniform grid would produce the desired result.

Although many algorithms and public domain software packages are available, the program *Triangle*¹ by J.R. Shewchuk (1996), based on a *Delaunay* triangulation was chosen, due mostly to the manner in which it handled colinear points without any difficulty. *Triangle* is a *C* program which has gone through several generations of modifications and allows the user numerous options. Although originally designed as a standalone program module it has been successfully implemented for use in *FORTRAN* by adding a transparent *FORTRAN* callable *C* interface routine to extend its usefulness to a range of other problems involving scattered data points such as model building.

The author of *Triangle* has set a number of default values, which may be overwritten by the user. These default values produce what he calls a *quality* triangulation. If data points are sparse in a subset of the model space, there is the possibility of areas of triangles becoming greater than some default values, or one of the angles of a triangle becoming overly obtuse. In these areas, triangles are not generated. This effect can be seen in the lower corners of Figure 3d.²

The gridding aspect of this problem, in its initial version, consisted of considering each grid point in sequence, determining in which triangle it lay, and using the three nodes of the enclosing triangle to linearly interpolate the travel time value at the grid point. Needless to say, this process is agonizingly slow but does produce acceptable results. Other gridding methods incorporating an initial *Delaunay* triangulation are in the process of being developed. A random-walk-type of approach has shown to improve computation speed significantly, but at present suffers from a significant amount of bookkeeping overhead and missed grid points, which have to be sought out and computed by the original method or by some nearest-neighbour average.

¹ J.R. Shewchuk holds the copyright for the program *Triangle*. It may be used for educational purposes without compensation but its use in any commercial venture is subject to certain conditions.

² This brief description of *Triangle* in no way does it justice. Interested readers are referred to Shewchuk's paper.

Gridding of traveltime data using *IPgXYZToGrid*

Another gridding approach involved the *Winteracter*³ subroutine *IPgXYZToGrid*, which was used to convert the set of random (x, z, t) data-points from each ray fan to a rectangular grid. The output grid has $nx \times nz$ nodes, as specified by the user. The routine calculates a weighted average of the t values for all the points that lie within a specified search box around each output grid node. Increasing the size of the search box may increase the accuracy of the gridded output. However, it was found that a larger search box, although it gave smoother traveltime contours, resulted in the data being smeared in the migrated output, hence creating artificially continuous events. There is also a limit of 4095 points to a search box. If more points are supplied, they are ignored in the computation. The speed of the algorithm decreases with an increase in the number of input data points supplied and search box size. The quality of the compiler and power of the hardware are two other factors that affect the speed of the algorithm. More details about this algorithm can be found in the *Winteracter 2.20 Subroutine Reference* manual.

EXAMPLES

The following examples illustrate the effect of the gridding routine on the migrated-image quality. Figure 2 shows the model that was used to test the gridding routines and the migration code.

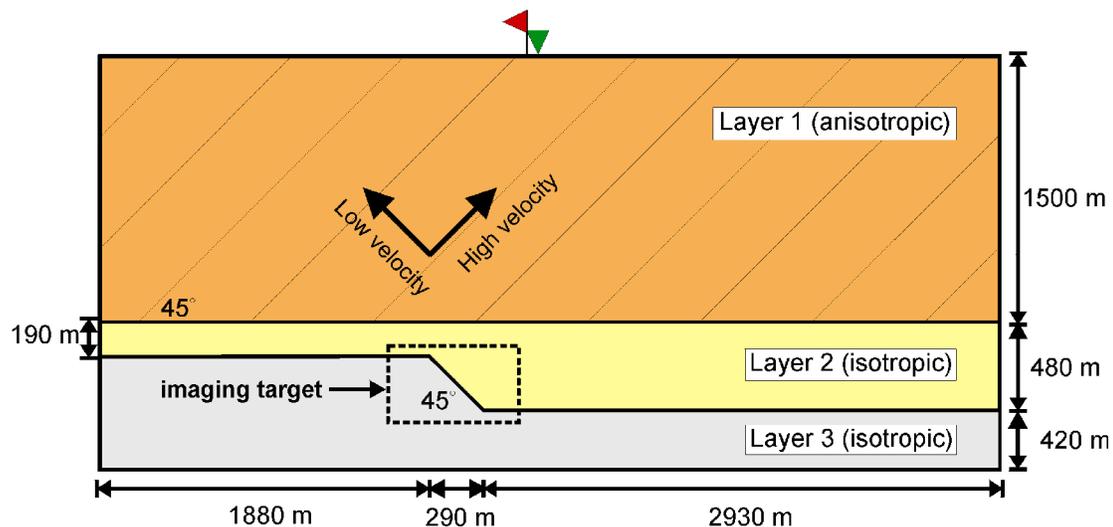
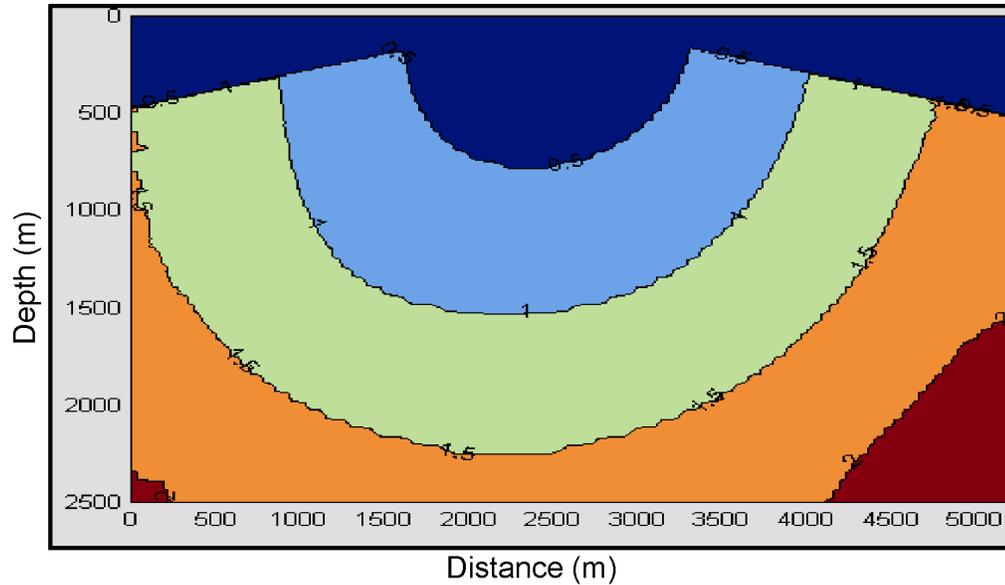


FIG. 2. The model used to test the gridding routines and the migration program. The first layer is anisotropic with the axis of symmetry tilted at 45°. The second and third layers are isotropic. The imaging target is the step in layer 2.

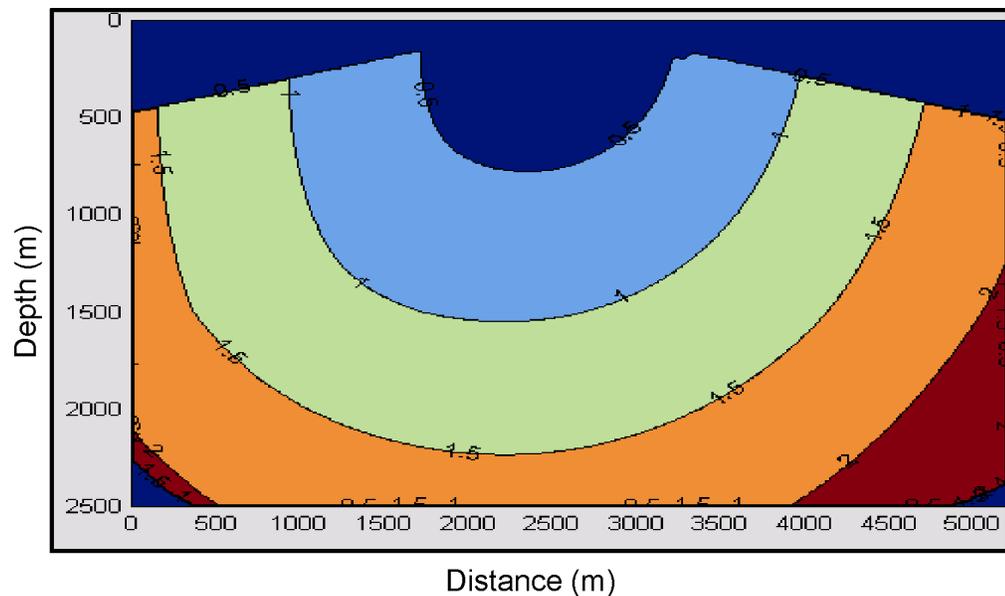
It consists of three layers: the first layer exhibits seismic velocity anisotropy with tilted axis of symmetry. The second layer is isotropic and has a step function, which can represent a fault or reef-edge. The step is used to check for positioning accuracy

³ *Winteracter* is a portable user interface and graphics toolset for *FORTRAN* 9x developers. More information can be obtained from *Interactive Software Services* in the United Kingdom.

when imaging underneath an anisotropic overburden. The third layer is also isotropic and is simply used to obtain a reflection off the top and base of the step in layer 2. Finite difference modelling code at Veritas GeoServices was used to create synthetic P - P and P - S data sets. The acquisition was split-spread with a shot interval of 20 m and a receiver interval of 80 m.



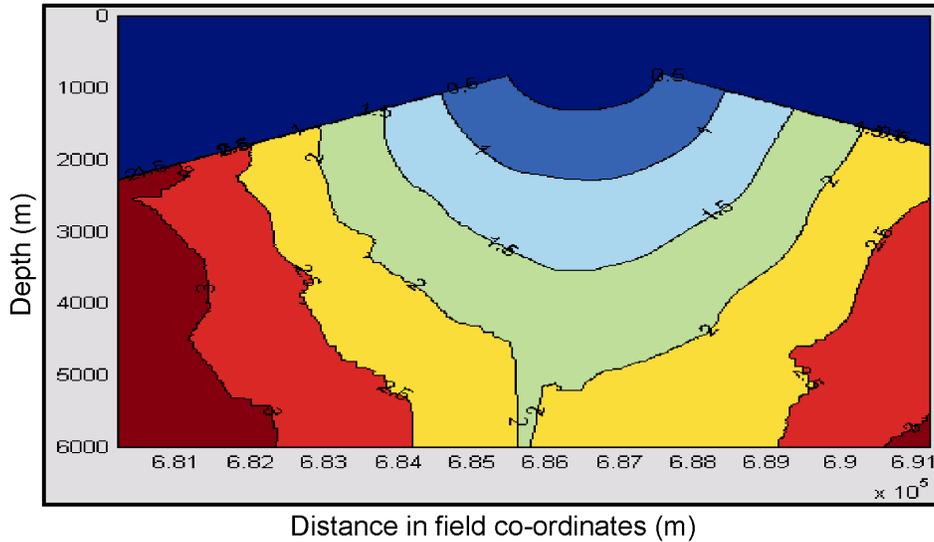
(a)



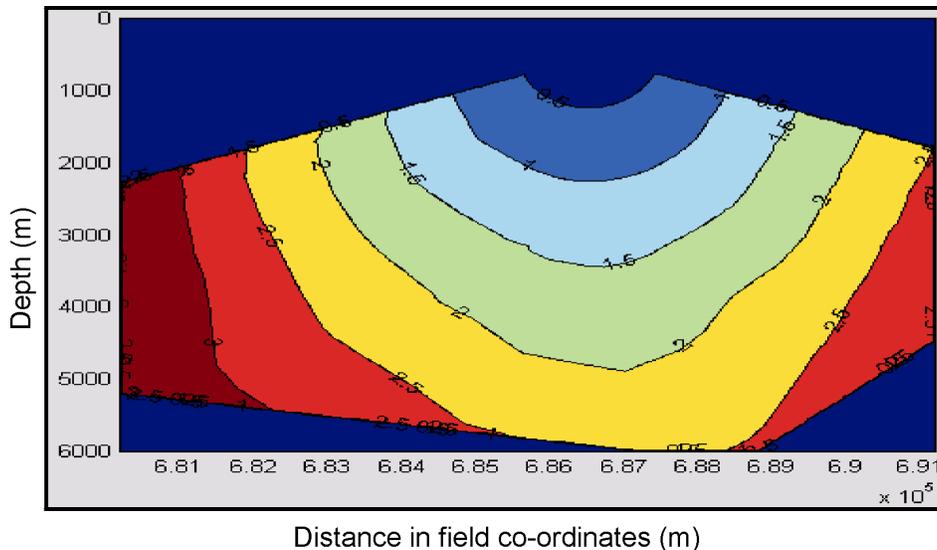
(b)

FIG. 3. Traveltime tables for the shot and receiver pair shown on the model in Figure 2. (a) The gridding was done using *lpgXYZtograd*. (b) The gridding was done using *Triangle*. The rough contour surfaces in (a) degrade the quality of the migrated image.

The two gridding routines described in the previous section were then used to create traveltimes tables for migration. The grid spacing was 10m in both the x - and z -direction in each case. Typical source-receiver traveltimes tables for a source at 2460m and a receiver at 2520m (Figure 2) from each approach are shown in Figures 3a and 3b. Gridding using *Triangle* resulted in smoother traveltimes contours (b), which give a better depth image.



(c)



(d)

FIG. 3 (continued). Traveltimes tables for source and receiver pair from the model shown in Figure 1. (c) The gridding was done using *IPgXYZtoGrid*, and (d) the gridding of traveltimes data was done using *Triangle*.

The robustness of the gridding algorithm becomes more important when the velocity model is complex, such as that shown in Figure 1. Figures 3c and 3d show

source-receiver traveltimes for a source and receiver pair located at 5896m and 6746m respectively. The problems with the gridding when the routine is not robust enough are now more obvious (Figure 3c). The traveltimes from *Triangle* (Figure 3d), on the other hand, are still smooth and continuous.

The migrated image quality obtained using *Triangle* to do the gridding during the anisotropic PSDM of the *P-P* and *P-S* finite-difference model datasets is shown in Figures 4 and 5 respectively. In each case, (a) shows the anisotropic PSDM result using anisotropic pre-stack depth migration code developed and used in production at Veritas GeoServices, and (b) shows the anisotropic PSDM result using the migration program developed in this work. These images show that the results obtained using the code developed in this work compare well with those obtained using the migration code. The first breaks were muted in the datasets used in Figures 4b and 5b while they were not in the case of Figures 4a and 5a, hence the noise between 0m and 1000m depth. Another difference was in the migration process, since the production code used a smoothed migration-velocity model instead of a polygon-based one as used in this work. The weak event of the step on the *P-S* migrated images (indicated by the arrows on Figures 5a and b) is attributed to the fact that polarity-reversal was assumed to occur at the centre of the spread, and hence trace polarity was reversed for all traces on one side of the spread. This assumption does not hold if the axis of symmetry is tilted, as it was in this case. This led to a second weaker image of the step at the wrong location.

Figure 6 shows the difference in migrated image quality of field data as a result of using different gridding algorithms in the migration. The data in this case comes from the Rocky Mountain Foothills of southern Alberta, Canada. The migration-velocity model used for anisotropic pre-stack depth migration (PSDM) is shown in Figure 1. The grid spacing was 25m in the x-direction and 15m in the z-direction. When the gridding algorithm used to create the traveltimes for the migration was not robust enough (Figure 4a), the events appear to be discontinuous and the image is difficult to interpret, in particular the circled event on Figure 6a. On the other hand, the use of *Triangle* as a gridding routine (Figure 6b) resulted in a superior image which was easier to interpret. The events are more continuous and the structural style is defined better. The circled event on (Figure 6a) can now be interpreted as a pop-up structure (Figure 6b). The hanging wall of the thrust (indicated by the arrow on Figure 6a) is more continuous and there is also a continuous reflection from the basement (indicated by the double-headed arrow), which is not imaged on Figure 6a. The interpretation of Figure 6b is shown in Figure 6c. This was constrained by lithological tops from 7 wells (shown in white on Figure 6c). Major faults are marked in black and dominant horizons in colour. Faults and horizons are shown as continuous lines where known and as dashed lines where assumed.

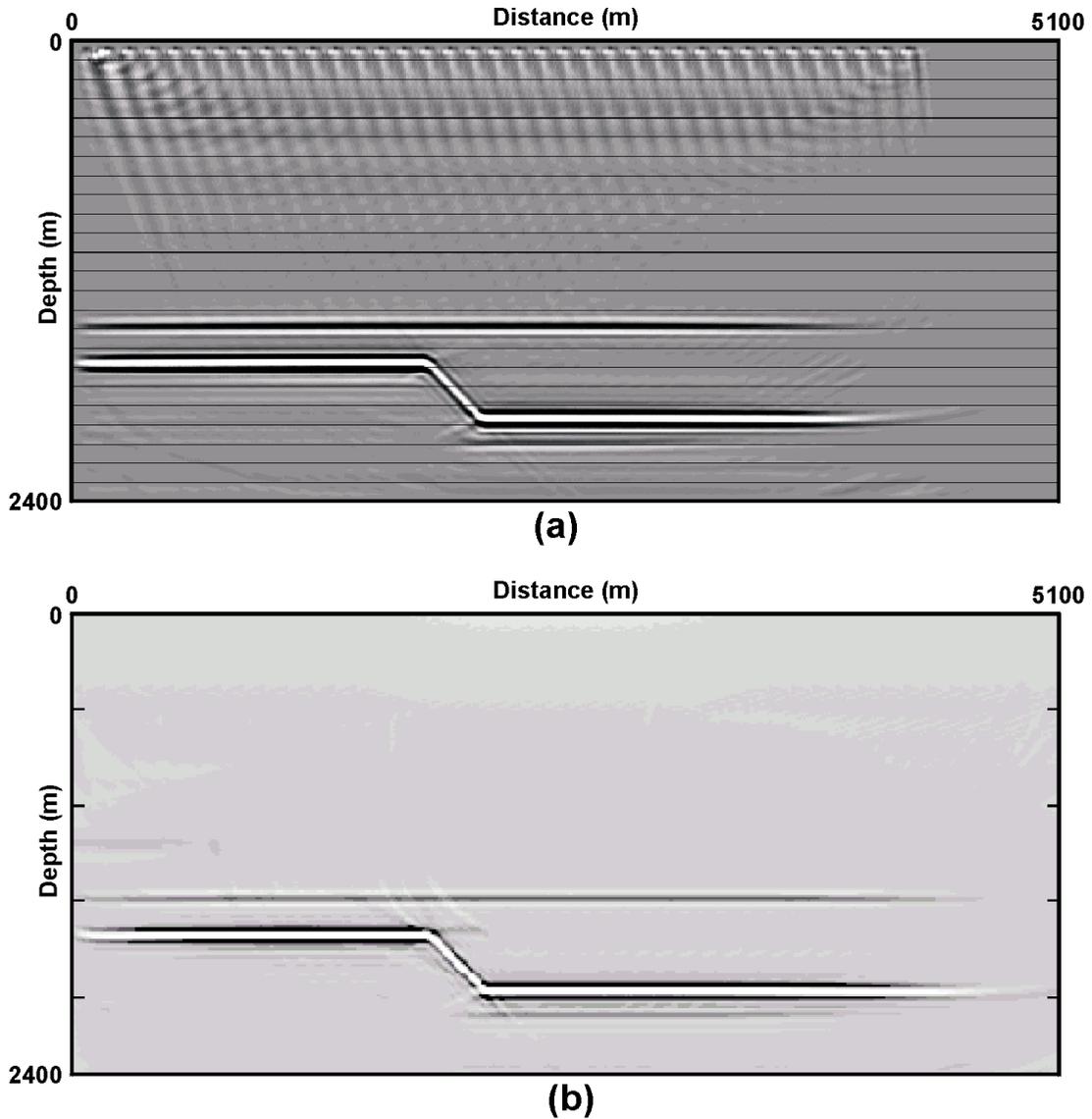


FIG. 4. Anisotropic PSDM of P - P data from the step model. (a) Image obtained using standard production code, and (b) image obtained using the code developed in this work. AGC has been applied to (a) on display. The migration-velocity model was smoothed in (a) and made up of polygons in (b). The sample rate of the input data was 1ms in (a) and 4ms in (b). The first breaks were muted in (b) only.

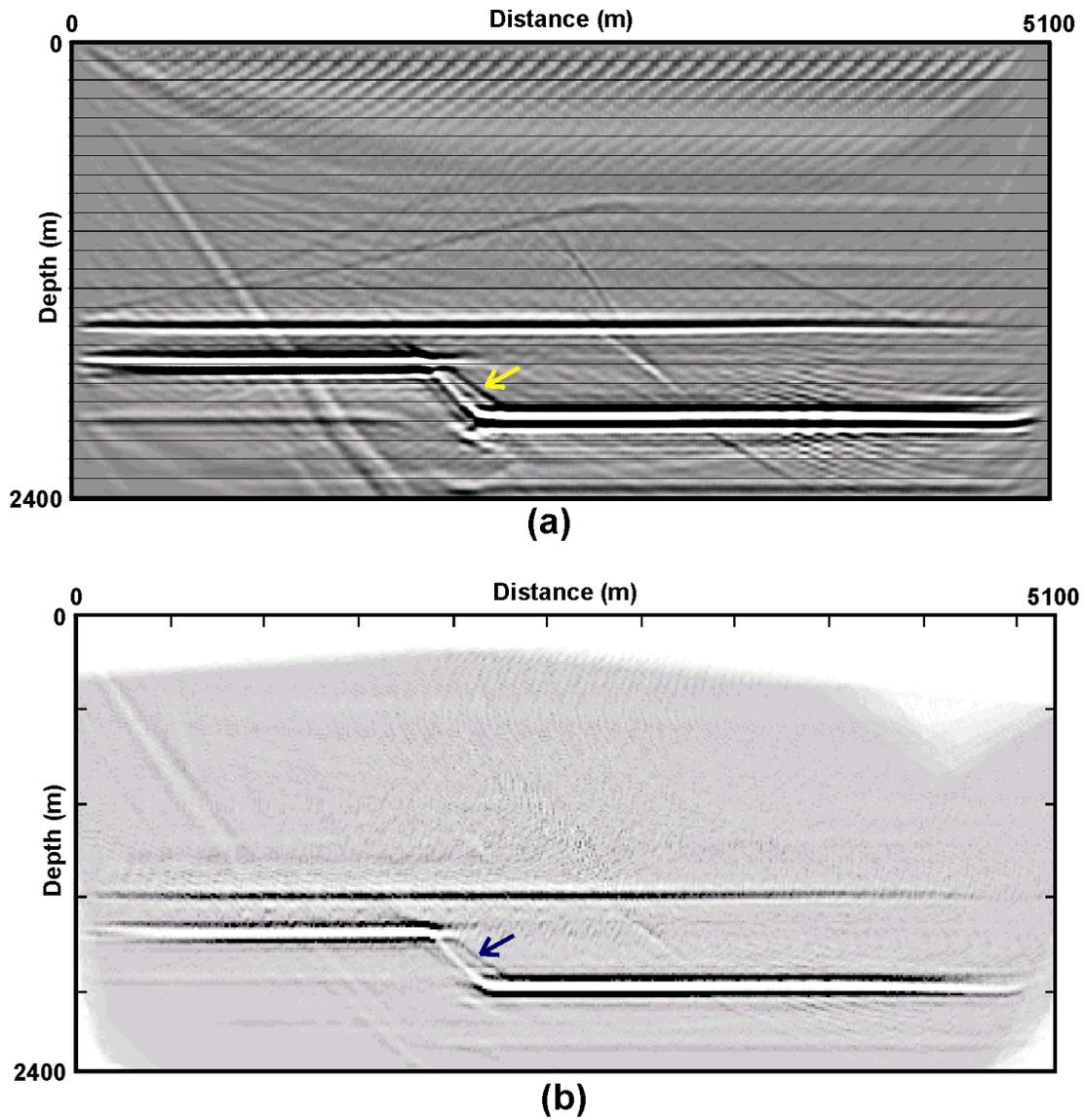


FIG. 5. Anisotropic PSDM of P - S data from the step model. (a) Image obtained using standard production code, and (b) image obtained using the code developed in this work. AGC has been applied to (a) on display. The information on the sample interval of the input data, the migration velocity model and the first breaks are the same as in Figure 4.

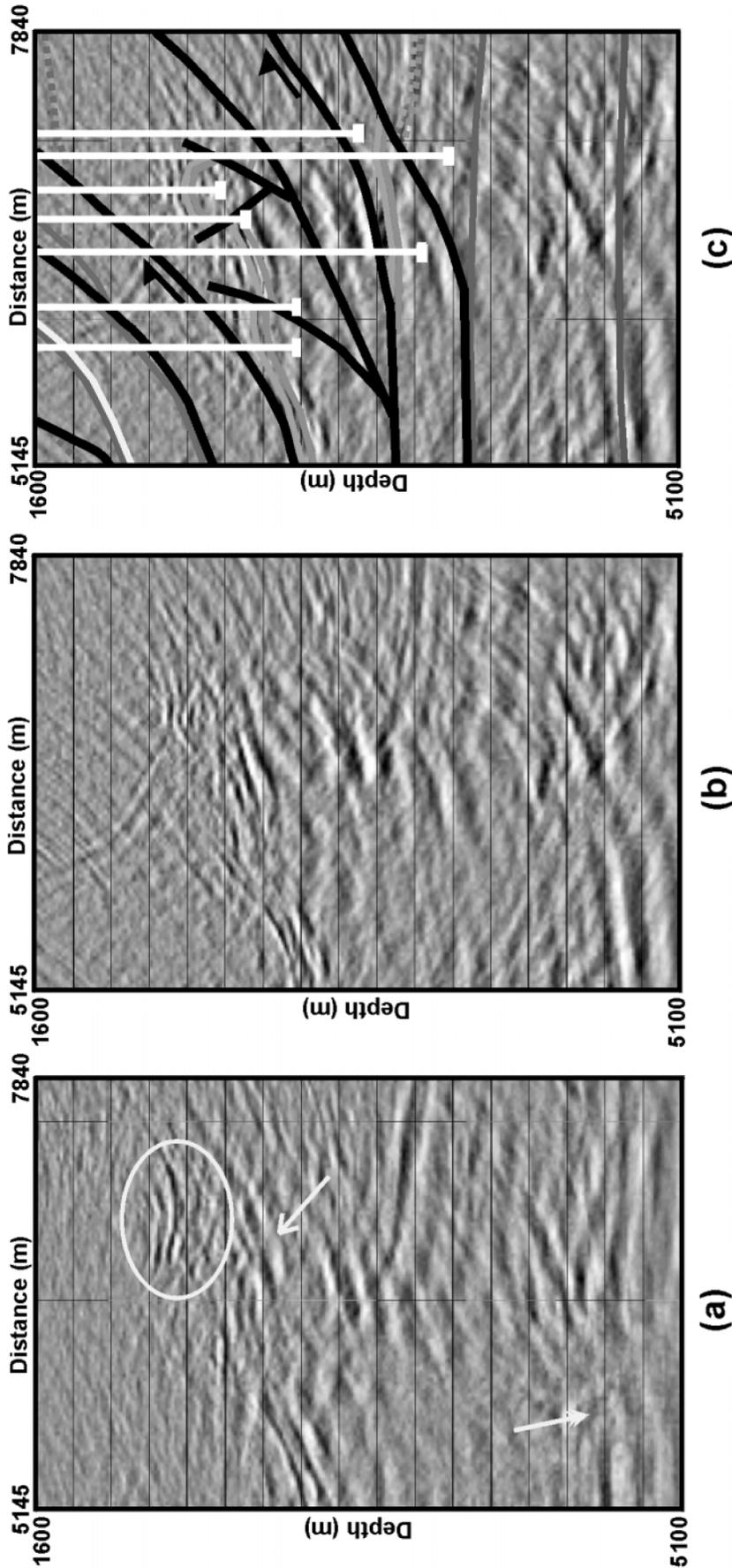


Figure 6. Anisotropic PSDM of part of a surface seismic line from the Foothills of southern Alberta. (a) Gridding done using *IPgXYZToGrid*, and (b) gridding done using *Triangle*. The latter shows better event continuity and structural definition, making it easier to interpret. (c) The interpretation of (b), which was constrained by lithological tops from nearby wells (shown in white).

DISCUSSION

This work has shown that good migrated images can be obtained using an interface-based raytracer to calculate the traveltimes tables for migration, as long as the algorithm used to convert the calculated traveltimes into a regular grid is sufficiently robust. The images obtained compare well with those obtained using industry standard processing software.

ACKNOWLEDGEMENTS

We would like to thank sponsors of CREWES, the Fold-Fault Research Project (FRP) and NSERC for their financial support. Thanks also go to Veritas GeoServices for the finite difference modelling data sets and for the use of the *SAGE* processing software to produce the images in Figures 4a and 5a.

REFERENCES

- Bleistein, N., Cohen, J.K., and Stockwell Jr., J.W., 2000: *Mathematics of Multidimensional Seismic Inversion*, Springer Verlag.
- Shewchuk, J.R., 1996, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, First Workshop on Applied Computational Geometry, ACM, May 1996.
- Thomsen, L., 1986, Weak elastic anisotropy: *Geophysics*, **51**, 1954-1966.
- Winteracter 2.20 Subroutine Reference, 1997-1999, Interactive Software Services Ltd., U.K., 5-69.