Synthetic P-P and P-SV cross sections

Gary F. Margrave and Darren S. Foltinek

ABSTRACT:

A new numerical seismic modeling facility has been developed which implements the SYNTH algorithm (Howell et al) to create P-P or P-SV offset gathers from well log information. The new facility extends the previous SYNTH functionality in that a cross section of well logs may be input (P sonic, S sonic, and density), a gather is created at each log location, the gathers are then stacked to produce a P-P or P-SV cross section. The full Zoeppritz equations, with raytraced incident angles, are used to model the offset dependent reflectivity effects. A great variety of wavelet options allow simulation of various source and bandwidth effects and near surface ghosting. This facility has been developed within the Matlab scientific programming environment which means that the code is relatively robust and easily ported to almost any hardware platform. Logedit and Logsec, which are commercially available Matlab packages for editing well logs and interpolating logs along cross sections, are used to prepare the well log cross sections

INTRODUCTION

The construction of synthetic seismograms is a popular, nearly essential, step in the interpretation of seismic data. For ordinary (vertical component) seismic data there are many options available for the construction of suitable seismograms with the most popular being those which build simple convolutional seismograms from well logs. Many variations on the basic algorithms exist (see Waters, 1981 for an overview) but the general process is simply to compute normal incidence reflection coefficients and convolve them with a suitable wavelet (source/receiver response). The realism inherent in such models comes largely from the richness of the well logs while the seismic modeling technology is quite rudimentary. Typically, no raytracing is involved, no attempt is made to model mode conversions, coherent noise, attenuation, or any other 2-D or 3-D effects. That the models often prove strikingly similar to migrated seismic sections is a testament to the effectiveness of seismic processing at eliminating unwanted effects and producing a close approximation to band limited reflection coefficients. Often entire cross sections are modeled by first producing a cross section of well logs and then running the convolutional model on each well log in the cross section.

For P-SV data, the ordinary convolutional seismograms do not suffice because they assume normal incidence where the P-SV reflection coefficient is zero. Prior work in the CREWES project (Lawton and Howell, 1992) has shown how, with a slight complication in technology, simple converted wave seismograms can be built by using offset raytracing and the Zoeppritz equations (for reflection coefficients) to compute a synthetic common reflection point (crp) gather and stacking the gather. These seismograms have proven useful not only for P-SV synthetics but also to create more realistic P-P (and S-S) synthetics in settings where there is significant amplitude variation with offset (AVO).

This paper reports on the implementation of the SYNTH algorithm in the MATLAB scientific programming environment and its integration into a new facility to compute synthetic seismic cross sections. Though still under development, this facility has already been used to create some very useful seismic cross sections for the Blackfoot interpretation. The creation of well log cross sections is a difficult and essential part of this process and is handled by a separate, commercial, MATLAB package called Logsec.

ALGORITHM DESCRIPTION

The essential steps of the SYNTH synthetic seismogram algorithm , as implemented in this work, are shown in figure 1. Required inputs are P-wave sonic, S-wave sonic, and density logs together with a specification of the gather geometry and a wavelet. The program can then compute either a P-P offset gather or a P-S offset gather. (Note that the FORTRAN version can also compute an S-S gather. See figure 5.) Either gather contains only the appropriate primary events with no other mode conversions or multiples.

The first step is to resample the logs into layers of thickness, Δz_j, such that the P-SV or P-P vertical traveltime, Δt_j, is the same for all layers. This simplifies the generation of the output traces by ensuring that each computed reflection coefficient maps to an integral sample number in two-way vertical traveltime. A similar step is a normal part of the generation of 1-D synthetic P-P seismograms (Waters, 1981) which are a standard industry tool. Unlike the 1-D case, this log averaging serves a second purpose here by stabilizing the raytracing. A typical well log shows many thin layers with very rapid velocity fluctuations as well as the blocky behavior associated with formation boundaries. A propagating wavefield will not be sensitive to rapid fluctuations that occur on a length scale much less than the shortest wavelengths involved (see figure 10). However, raytracing corresponds to the zero wavelength limit of wavefield propagation and so is destabilized by these rapid fluctuations. The log resampling amounts to a time averaging of the logs over a traveltime equal to the intended sample rate. As a result, high velocity log segments will be averaged more than low velocity ones; and since $\lambda=v/f$, this means that large wavelengths are averaged more.

The log resampling determines the number of layers in the model that must be raytraced. These are augmented by a constant velocity overburden chosen to bring the event times, and hopefully the incidence angles, into rough correspondence with observed ones. The next stage of the algorithm loops over these layers and generates one event across all offsets for each iteration.

The first step in the k^{th} loop iteration is to trace P-P rays or P-S rays from the source to each receiver with a single reflection off the k^{th} interface. The raytracing iterates indefinitely until a ray pair is found with the required offset within a specified capture radius.

Having obtained a ray parameter for each desired offset, the Zoeppritz equations can then be used to compute complex reflection coefficients. The Zoeppritz implementation used is that of Aki and Richards (1980) (the same as was used for the FORTRAN version) and was written by Dr. E. Krebes of the University of Calgary. For this work, Dr. Krebes' code was ported to MATLAB. The Zoeppritz formulation assumes incident plane waves and, since a seismic record is probably better modeled with spherical waves, represents a significant approximation. It is expected that this approximation is reasonable for incident angles not near the critical angle (C.W. Frasier, personal communication.)

The Zoeppritz reflection coefficient is then combined with the expected effect for a P or S wave incident on a free surface. This is simply a second application of the Zoeppritz equations to compute the reflected P and S waves and then the entire horizontal or vertical displacement component. The method is that of Dankbaar (1985) and was originally coded by D.W. Eaton.

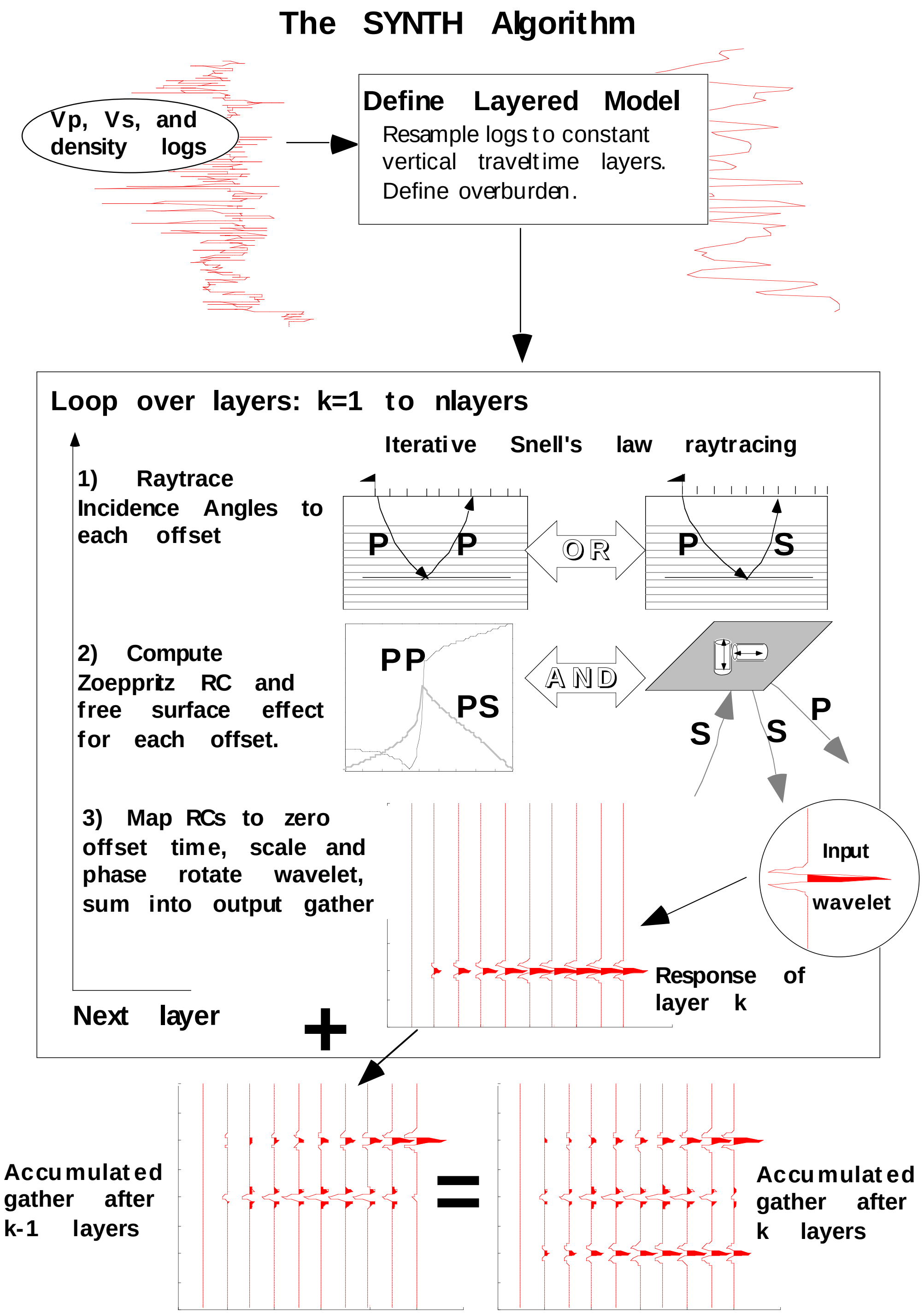


Fig. 1. The SYNTH algorithm

Though most reflections do not generate complex coefficients, this possibility is allowed for and amounts to a phase rotation as well as a scaling of the wavelet. The final step of each iteration is to sum a scaled, and possibly phase rotated, wavelet into the output gather at each offset. Each event is mapped directly into vertical traveltime rather than the correct raytraced traveltime. Thus the generated gather is called "pseudo zero offset" and simulates a perfect removal of normal moveout with no nmo stretch.

COMPARISON WITH THE ORIGINAL SYNTH

In order to test the new MATLAB implementation, a simple test case was developed and run through both the FORTRAN (original) SYNTH and the new version. The model used for testing is shown in figure 2 and consists of four discrete reflections with alternating v_p contrasts, a v_p/v_s ratio of 2, and constant density. On the right hand side of figure 2 are the actual Zoeppritz equation reflection coefficients plotted versus source-receiver offset.

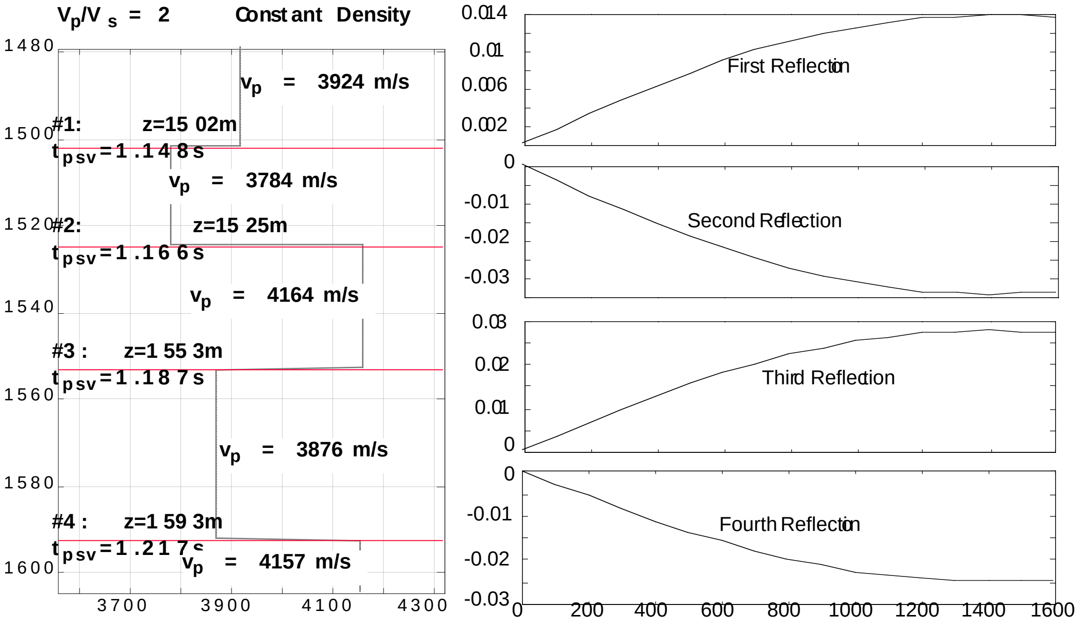


Fig. 2. Simple model used for testing MATLAB version of SYNTH. On the left is a synthetic Vp log showing four reflectors. On the right are reflection coefficients plotted versus offset for each reflector.

Running this model through both versions of SYNTH to create a P-S gather and using comparable program options resulted in the gathers shown in figure 3. Overall the two gathers are quite similar and both compare very favorably to the expected AVO (amplitude variation with offset) shown in figure 2.

A detailed comparison of the far offset traces is shown in figure 4. As can be seen, the results are very similar but not exactly the same. It is felt that the small "glitches" surrounding the fourth reflection from the FORTRAN version are artifacts of the log resampling algorithm. The other differences are not as easy to explained but could result from differences in the raytracing algorithms used (and hence small differences in the incident angles) and perhaps differences in mapping reflection coefficients to specific time samples. In any case, it is felt that the results are sufficiently similar to lend condifence to the use of either code. (The new version is a complete rewrite of the original with the only shared code being the Zoeppritz and free surface effect routines.)

Figure 5 lists and compares the features of the two programs. The FORTRAN version is much more full featured in the number of physical effects that it models while the MATLAB version provides more in the way of user and external interfaces. It is intended to gradually implement all of the features of the original SYNTH in the new version. The graphical interface will also be evolved to facilitate increased understanding of the physical effects which contribute to the seismograms.

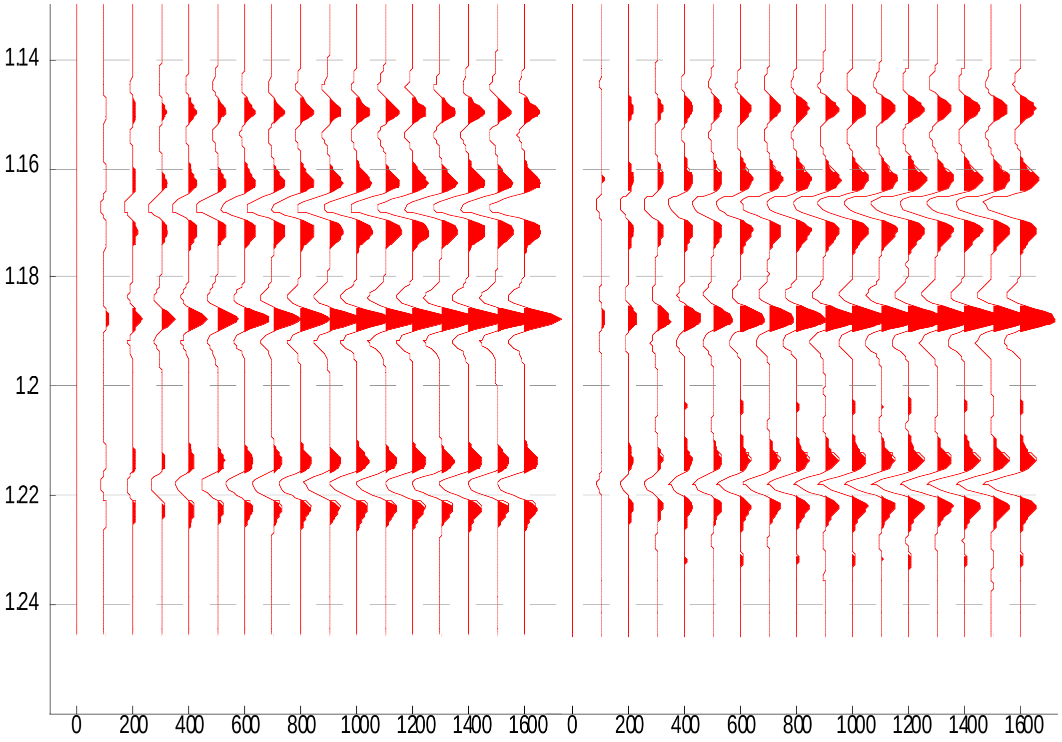


Fig. 3. Comparison of the new MATLAB version of SYNTH (left) with the previous FORTRAN version for the model of figure 2.

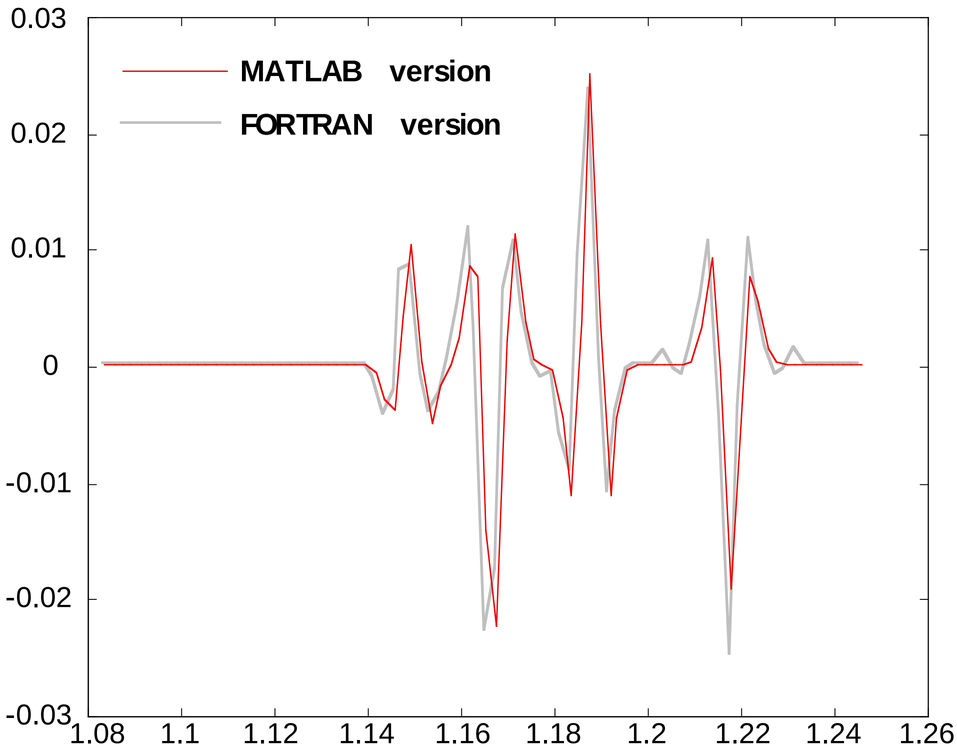


Fig. 4. Detailed comparison of far offset traces from figure 3

Features	FORTRAN SYNTH	MATLAB SYNTH
Raytraced incidence angles	✓	✓
Full Zoeppritz equations	✓	✓
Transmission losses	✓	
NMO in or removed	✓	
Pseudo zero offset	✓	✓
P-P and P-S gathers	✓	✓
S-S gathers	✓	
Q attenuation		✓*
Geologic cross sections		✓
LAS log import		✓
GMA log import	✓	✓
SEGY export	✓	✓
Flexible wavelet options		✓
Flexible graphics		✓*
GUI	✓	✓*

* currently under development

Fig. 5. The features of the FORTRAN and MATLAB versions of SYNTH compared.

A graphical interface (GUI) has been developed (in MATLAB) for the simple offset gather version of SYNTH and is planned for the more complex cross section version. Figure 6 shows the GUI with a P-SV offset gather for the 8-08 Blackfoot well. It offers complete access to all of the program parameters, log import in either GMA or LAS formats, scaled hardcopy, and SEGY seismic output.

IMPLEMENTATION IN MATLAB

In addition to having a very powerful set of tools for numerical manipulation, it is possible, within the Matlab programming environment, to create a complete application program, with a modern graphical user interface and all the data input / output capabilities that are needed. Another advantage of the Matlab environment is ease with which different software modules can be linked together. In this release of SYNTH, we have included a link to a Matlab wavelet editing program (WAVELETED) which allows wavelets to be designed and manipulated and then used in SYNTH to construct the seismogram.

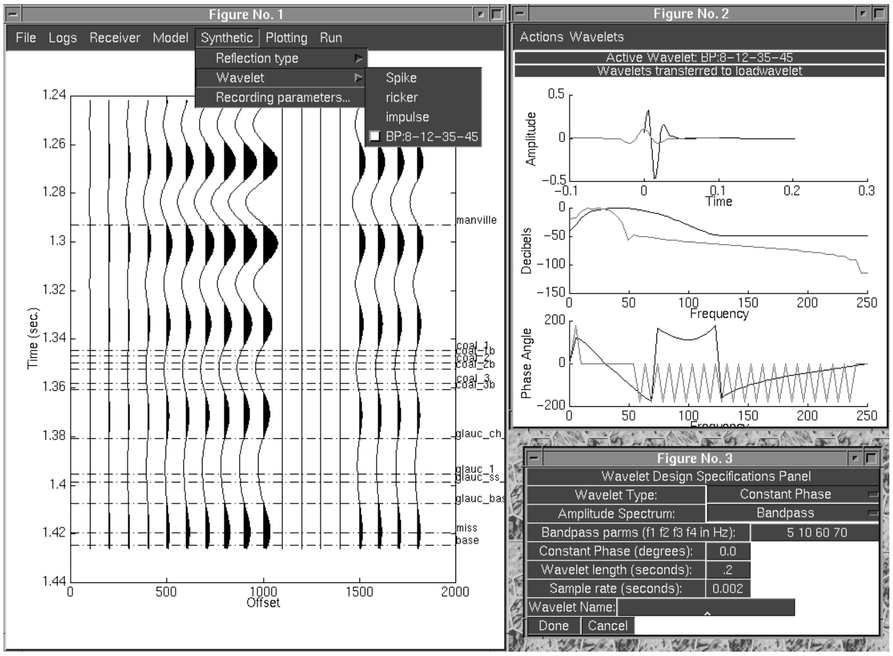
Figure 6 shows the currently implemented SYNTH graphical user interface which allows the generation of an offset gather and stack of either a P-P or P-S seismogram. The following parameters are available through the menus:

Input log types: P sonic, shear sonic and density
Receiver type: Vertical, Horizontal, or total displacement.
Surface to start of log: Vs, Vs and density
Vp/Vs ratio (if no shear log is available)
Reflection type: PP or PS
Wavelet type: spike, or any wavelet designed by the integrated wavelet editor
Recording parameters: trace length in seconds
Plotting parameters: offset seismogram and stack or just stack, formation tops (if available from logs) and plot scale

BLACKFOOT MODELS

As an illustration of the cross section building capabilities of the new MATLAB facility, P-P and P-SV cross sections were built for a simple model across a hypothetical glauconite channel sand as might be expected at Blackfoot. Three of the Blackfoot wells have a complete set of logs: P sonic, S sonic, and density, and they were used to form the well log cross section. 8-08 is a producing oil well, 12-16 is also in an apparent channel but encountered a shale plug, while 9-17 is regional.

Logsec was used to build a cross section with these logs to show the transition from full channel to regional. Figures 7 and 8 show the algorithm used by Logsec to synthesize new logs along a cross section given a set of real logs. Essentially, boundary conditions can be prescribed along each interface in the cross section to control the lateral propagation of log samples as layers thick, thin, and pinch out (figure 6). As shown in figure 7, logs synthesized intermediate to real wells on a cross section are formed by inverse distance weighted contributions from the closest left and right logs in each layer.



Wavelet Type: Constant Phase
Amplitude Spectrum: Bandpass
Bandpass parms (f1 f2 f3 f4 in Hz): 5 10 60 70
Constant Phase (degrees): 0.0
Wavelet length (seconds): .2
Sample rate (seconds): 0.002
Wavelet Name:

Figure 6: SYNTH and integrated WAVELETED user interfaces

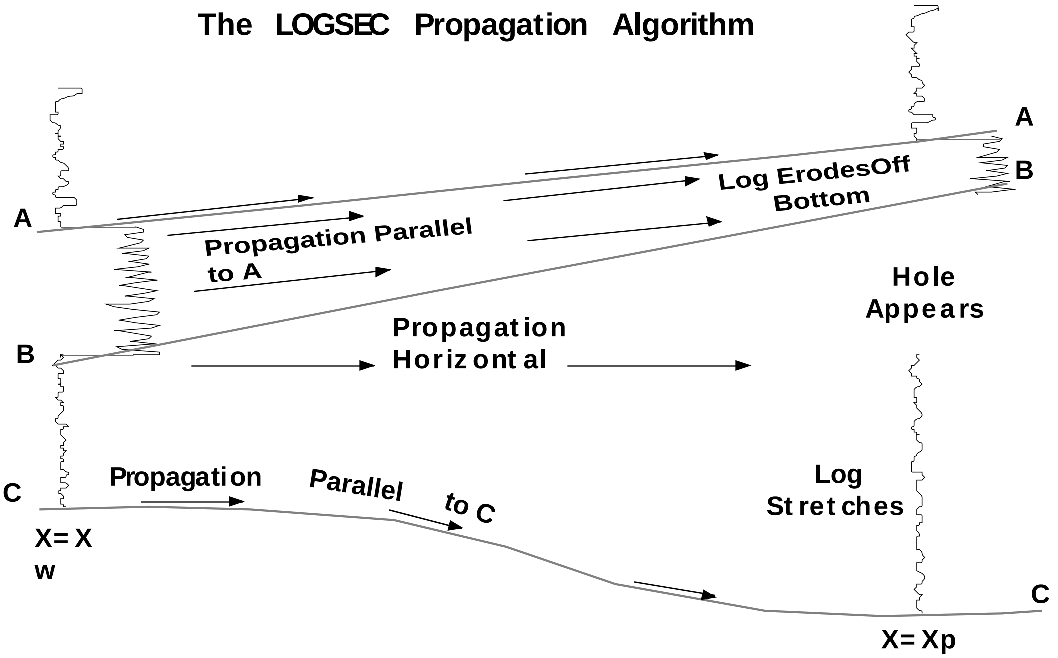


Fig. 7. Propagation is independently specifiable on top and bottom of each horizon. Horizontal propagation or parallel to any horizon is supported.

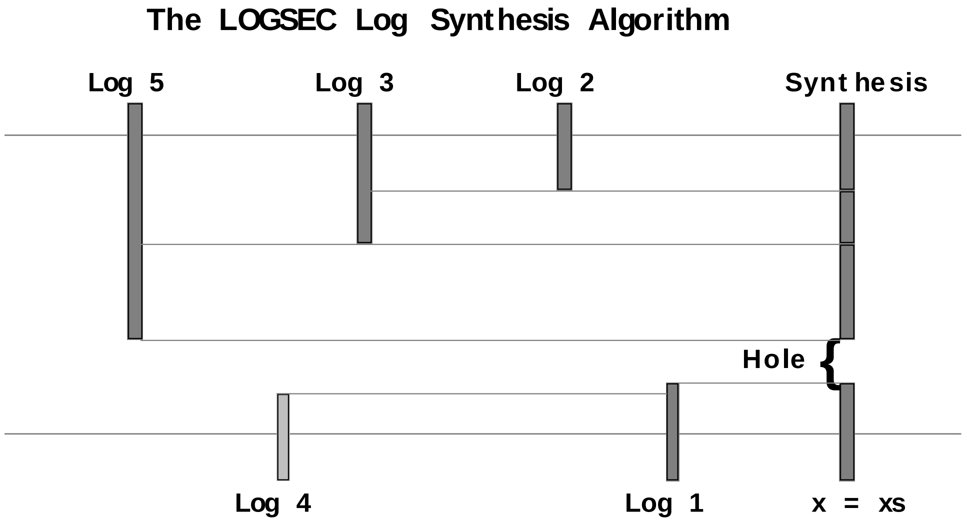


Fig. 8. Formation of a one-sided synthesis at x=xs is illustrated assuming horizontal propagation. Closer logs screen more distant logs from contributing. Final synthesis is the linear combination of left and right sided contributions weighted by their inverse distances.

Figure 9 shows the synthesized well log cross sections that were made for this example. The locations of the various horizons are taken form tops in the wells and vary according to interpretation in between. As is evident, 8-08 encountered full channel above a relatively low Mississippian and is positioned at coordinate 100 in the model. 12-16 encountered only partial channel and a locally high Mississippian and is positioned at coordinate 200. Finally 9-17 encountered no channel, a low Mississippian, and is at coordinate 300. Notice the considerable lateral variation of log values in the three cross sections particularly between the coals and the Mannville.

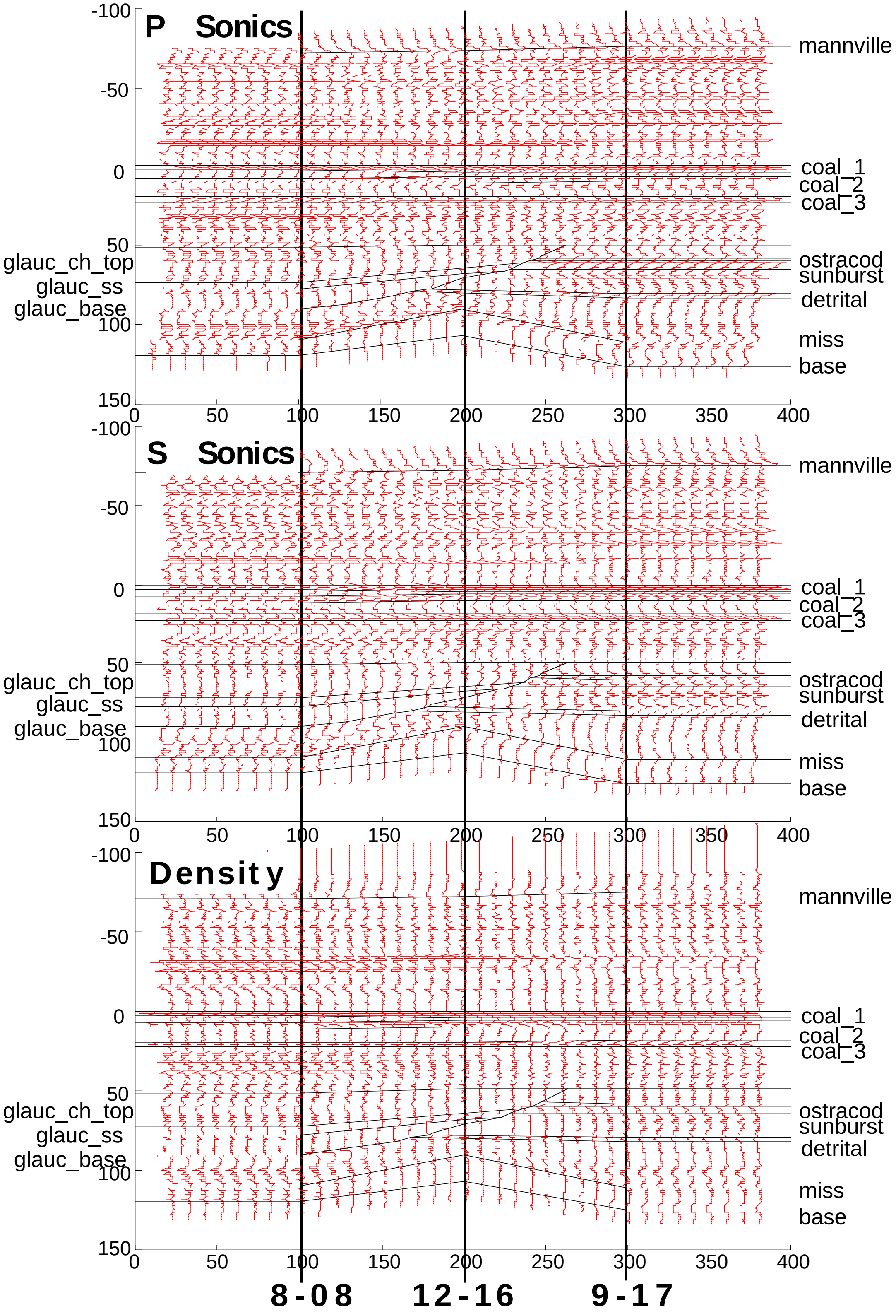


Fig. 9. P sonic, S sonic, and density log sections for Blackfoot channel model (from LOGSEC)

Figure 10 shows three synthetic cross sections created from the well logs of figure 9. Uppermost is an ordinary P-P convolutional model generated with an 8-12-75-85 zero phase wavelet, in the center is the model created by the stacked P-P gathers using the algorithm of this paper and the same wavelet, and on the bottom is a P-SV section created from stacked P-SV gathers and an 8-12-45-55 wavelet. The wavelets used are simply meant to be rough guesses of appropriate wavelets and are not intended to represent the Blackfoot data for interpretation purposes. (See Miller et. al. in this volume for more appropriate models generated with this algorithm.)

The convolutional model and the SYNTH cross section model show reasonable agreement through most of the section though the strong event above the Mannville on the latter seems anomalous. This may be due to an unrealistic range of incidence angles at the top of the model due to the current limitation of a constant velocity overburden. Note that the section displays of figure 10 are plotted such that there is no clipping. That is, the maximum amplitude gets one trace excursion. This leads to a bit of difficulty in the comparison.

The P-SV section is broadly similar to the P-P but there are some interesting differences in the channel vicinity. Note the apparent impossibility of making a true Mannville pick consistently across the section.

Both of these cross sections are highly dependent on the degree of "log averaging" (log resampling into layers of equal traveltime thickness) which is done. In general, some resampling must be done to prevent the raytracing algorithm from finding an unrealistic number of critical angle effects. Since raytracing is a zero wavelength approximation, it is sensitive to rapid fluctuations in layer properties no matter how thin the layers are. This is not so with waves which will generally not react to material fluctuations over distances greatly smaller than the dominant wavelength. Thus the models generation involves the determination of an appropriate log averaging which stabilizes the raytracing but is not so large that the log character is compromised. Note that there is no analog to this problem for ordinary convolution seismograms which are normal incidence simulations.

That these sections seem similar in resolution despite the wavelet differences is an interesting compensation effect. Figure 11 shows the P-P impedance from 8-08 in P-P reflection time next to the 8-12-75-85 wavelet used for the synthetics and then repeats the display with S-S impedance in P-SV time with the 8-12-45-55 wavelet also used here. Note the remarkable coincidence of almost identical geologic resolution in each case.

SUMMARY AND CONCLUSIONS

The SYNTH algorithm has been ported to MATLAB and linked together with commercial log editing and log cross section code to create a facility for the generation of synthetic P-P and P-SV cross sections. Since these cross sections are generated from stacked offset gathers, the P-P models have the promise of greater realism than the conventional convolutional approach. Furthermore, the P-SV models provide a totally new functionality to create P-SV cross sections.

More work is needed to assess the level of realism in the models and to determine appropriate values for critical parameters such as the log averaging parameter. Also the GUI interface needs to be extended to include the generation of cross sections.

Additional work will involve the extension of the algorithm to include more physical effects (Q, multiples, NMO, etc...).

ACKNOWLEDGMENTS

We are grateful to the sponsors of the CREWES project for funding this work and to PanCanadian Petroleum for providing the Blackfoot well logs used. We also thank Don Lawton and Sue Miller for helpful discussions regarding the algorithm and its use in interpretation. Finally, Gary Margrave thanks Chevron Canada Resources for allowing the use and publication of the Logsec program.

REFERENCES

Dankbaar, J. W.M., 1985, Separation of P- and S-waves, Geophysical Prospecting, 33, 970-986.

Lawton, D.C., and Howell, T.C., 1992, P-P and P-SV synthetic stacks, Expanded Abstract, 62nd SEG Annual International Meeting, October 25-29, New Orleans, USA, 1344-1347

Miller, S.L.M, Aydemir, E.O., and Margrave, G.F., 1995, Preliminary interpretation of the P-P and P-S seismic data from the Blackfoot broad-band survey, CREWES Annual Research Report, Vol. 7.

Waters, K.H., Reflection Seismology, A tool for Energy Resource Exploration, J. Wiley and Sons, 1981, 453 pages

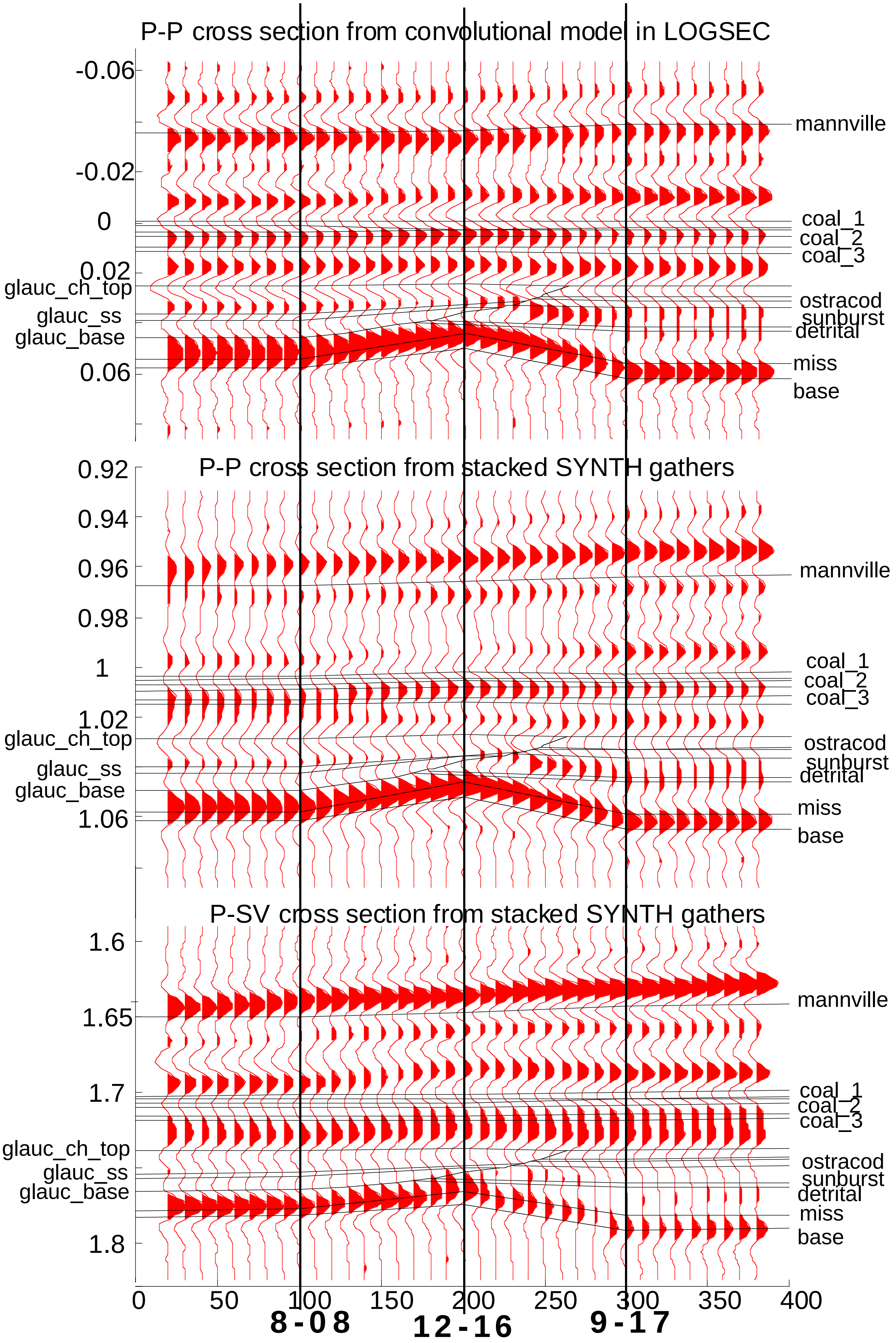


Fig. 10. Blackfoot synthetic sections: P-P convolutional (top), P-P stacked gathers (middle), P-SV stacked gathers (bottom).

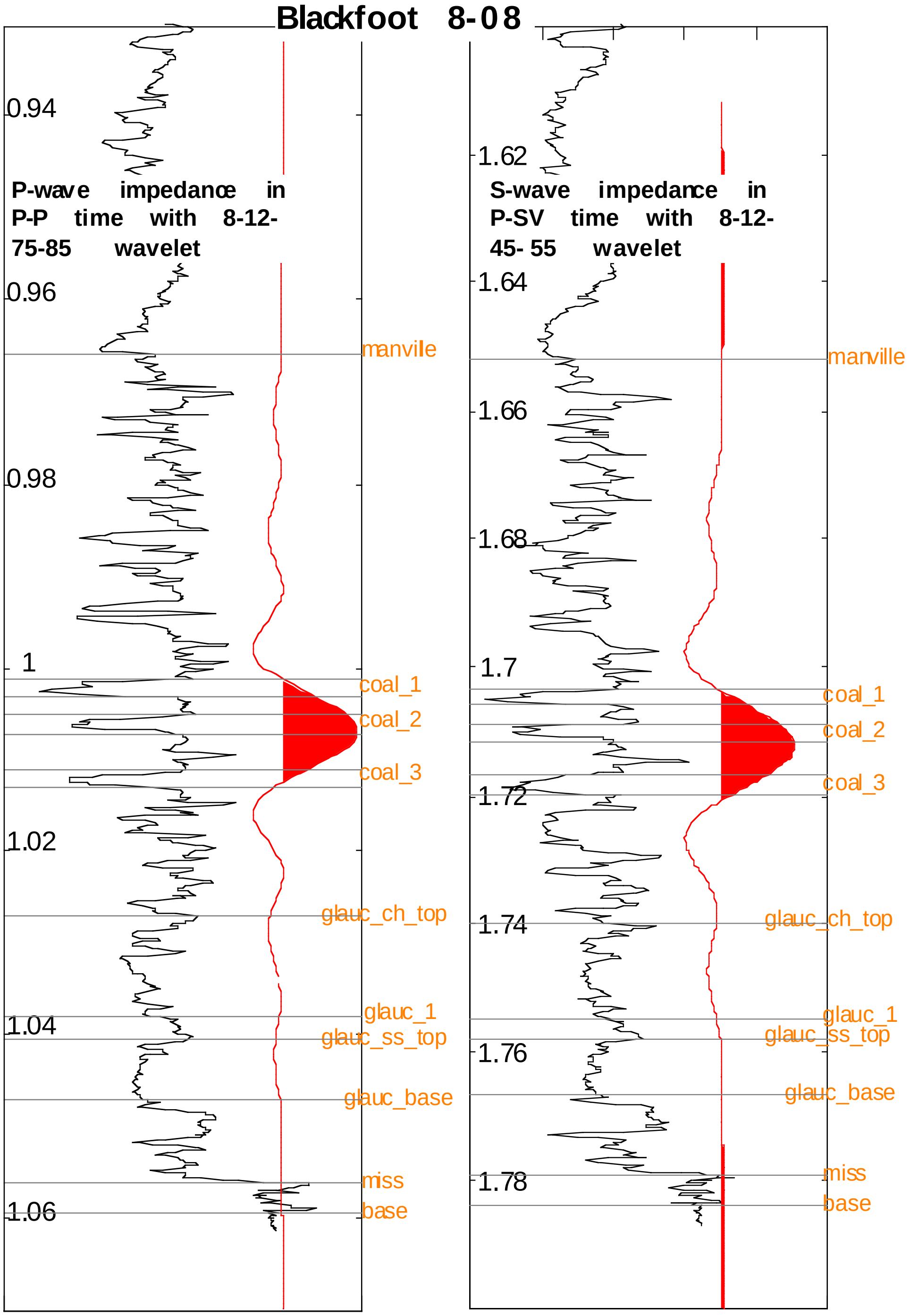


Fig. 11. The P-P impedance from 8-08 (right) is displayed in P-P reflection time next to the 8-12-75-85 wavelet used in modeling. Also the S-S impedance is shown (left) in P-SV reflection time next to the 8-12-45-55 wavelet used.