GENERATING A SYNTHETIC VOLUME FROM A SUITE OF PICKED HORIZONS – PROGRAM convolutional_modeling

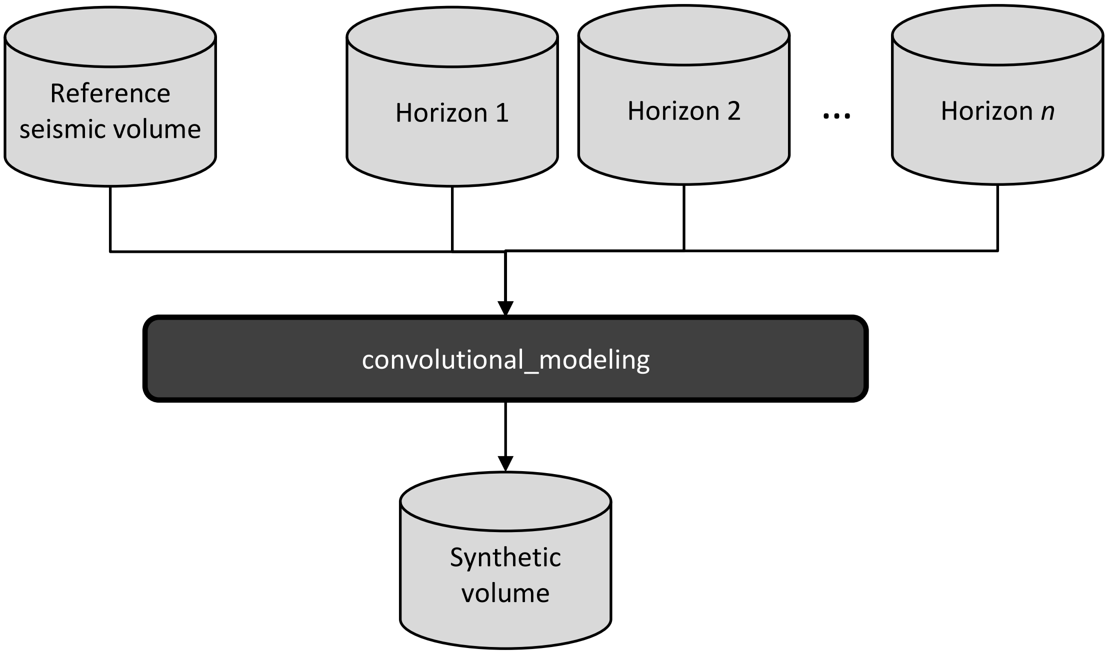
Contents

Overview

Synthetic seismic data generated from a known model are critical in both understanding and calibrating the response of new attribute algorithms. Although the response of geometric attributes such as coherence and curvature are usually intuitive, sometimes the response of specific cases such as attribute expression of box faults and wrench faults justifies additional analysis. Program **computational_modeling** reduces the events to be analyzed to those of interest. Program **convolutional_modeling** generates synthetic data by convolving a simple zero-phase source wavelet with a discrete number of reflecting horizons. The input and model can be developed from scratch, or alternatively, generated from the geometry and horizons picked on a 3D seismic survey. **convolutional_modeling** does not model diffractions, multiples, footprint, coherent noise, or other phenomena modeled by the orders of magnitude more expensive 3D wave equation methods such as finite differences.

Computation Flow Chart

Program **convolutional_modeling** reads in a reference seismic or attribute volume as well as a suite of picked horizons and generates a seismic wavelet at each pick to produce an output synthetic data volume. The input horizons may be produced by picking events in an interpretation workstation software or through the construction of idealized surfaces using a program such as **Matlab**.



Output file naming convention

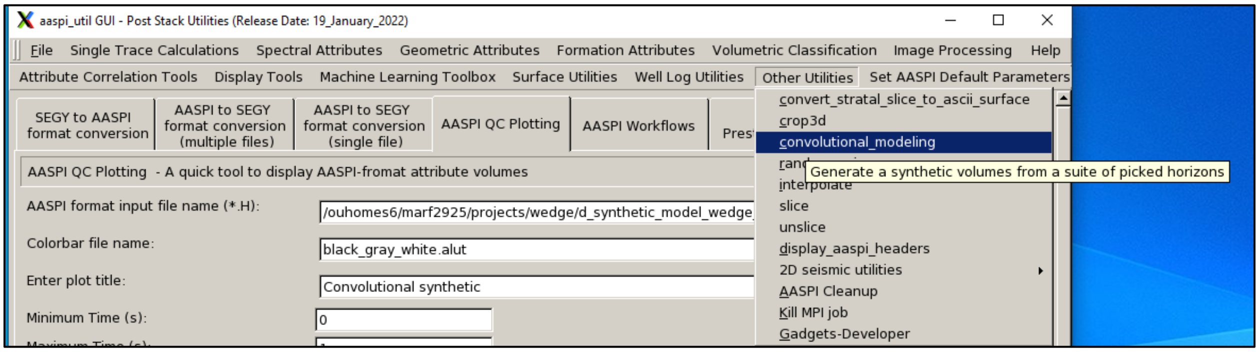
Program **convolutional_modeling** will always generate the following output files:

Output file description	File name syntax
synthetic model	d_synthetic_model_*unique_project_name_suffix*.H
program log information	convolutional_modeling_*unique_project_name_suffix*.log
program error/completion information	convolutional_modeling_*unique_project_name_suffix*.err

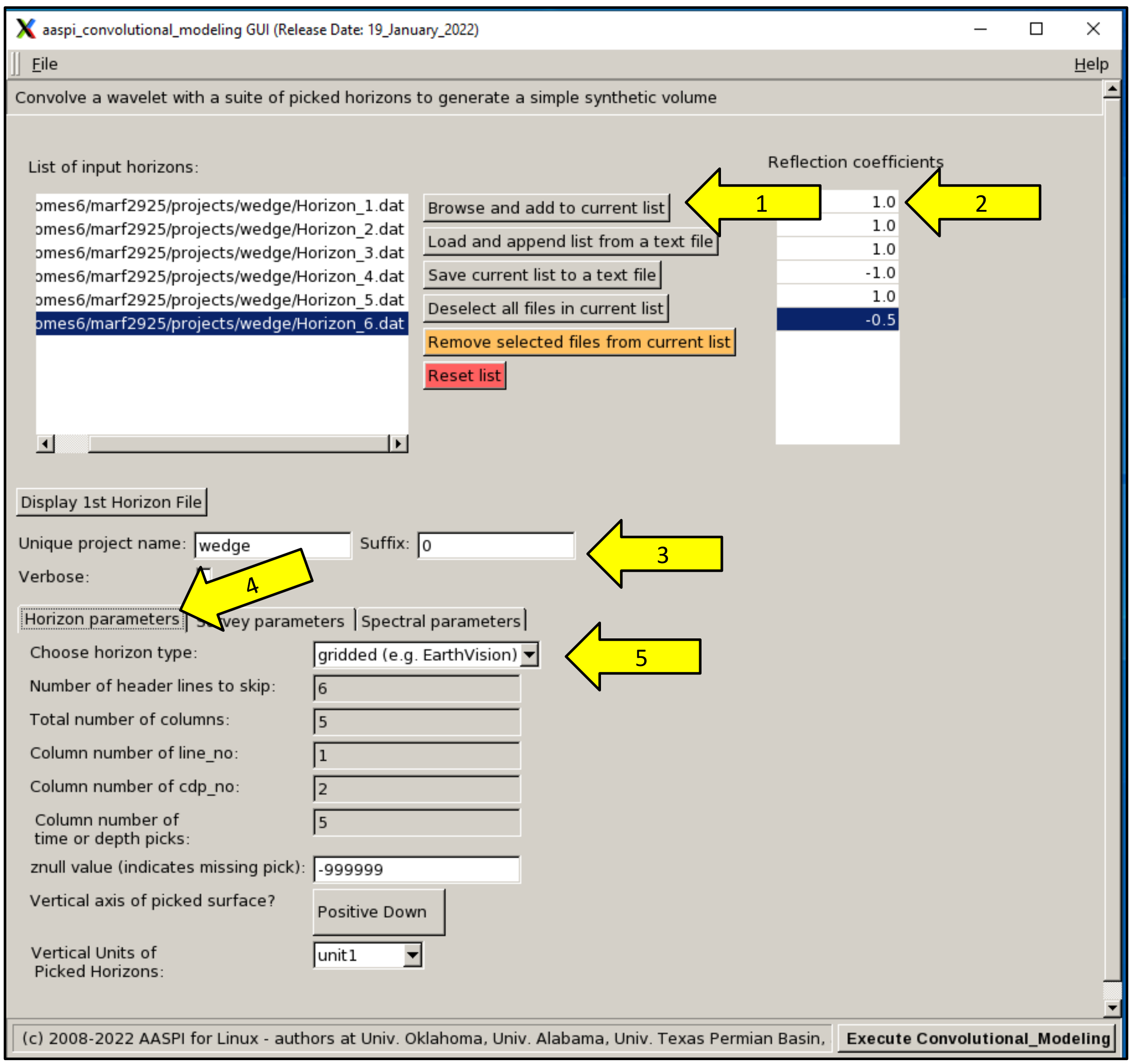
where the values in red are defined by the program GUI. The errors we anticipated will be written to the **.err* file and be displayed in a pop-up window upon program termination. These errors, much of the input information, a description of intermediate variables, and any software trace-back errors will be contained in the **.log* file.

Invoking the convolutional_modeling GUI

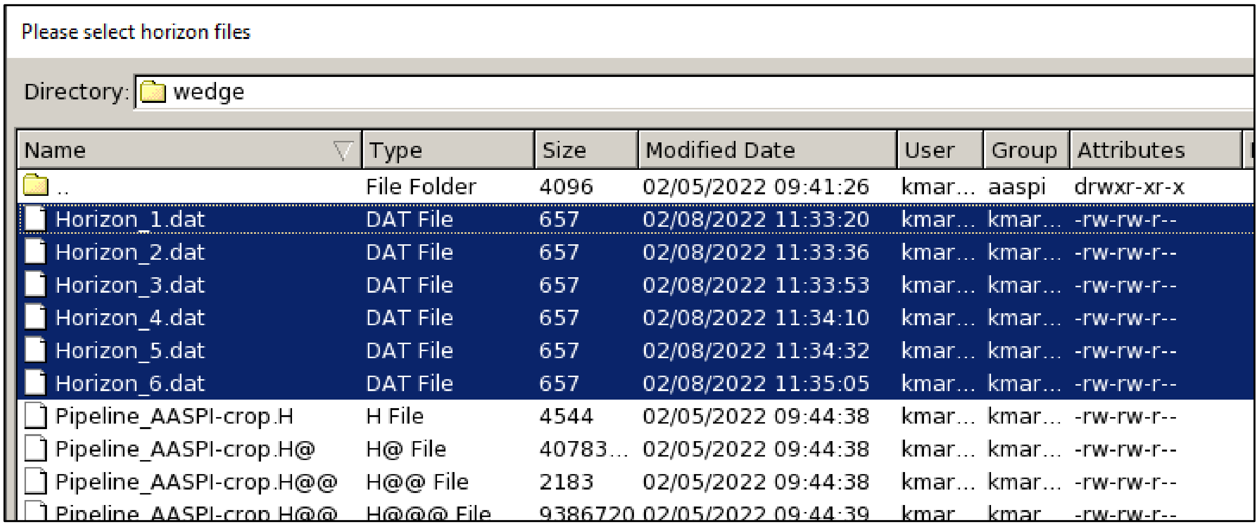
The **convolutional_modeling** GUI is found on the **aaspi_util** GUI under the *Other Utilities* tab:



The following GUI appears:



The first step is to (1) selected the desired seismic horizons. Here, I've selected six simple horizons that I constructed to define three wedges:


Please select horizon files

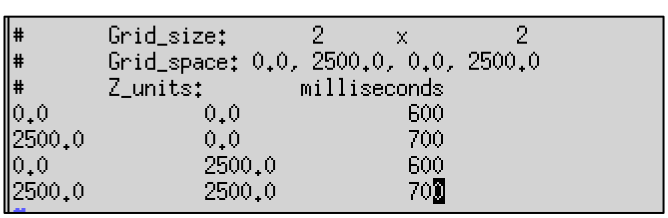
Directory: wedge

Name	Type	Size	Modified Date	User	Group	Attributes
..	File Folder	4096	02/05/2022 09:41:26	kmar...	aaspi	drwxr-xr-x
Horizon_1.dat	DAT File	657	02/08/2022 11:33:20	kmar...	kmar...	-rw-rw-r--
Horizon_2.dat	DAT File	657	02/08/2022 11:33:36	kmar...	kmar...	-rw-rw-r--
Horizon_3.dat	DAT File	657	02/08/2022 11:33:53	kmar...	kmar...	-rw-rw-r--
Horizon_4.dat	DAT File	657	02/08/2022 11:34:10	kmar...	kmar...	-rw-rw-r--
Horizon_5.dat	DAT File	657	02/08/2022 11:34:32	kmar...	kmar...	-rw-rw-r--
Horizon_6.dat	DAT File	657	02/08/2022 11:35:05	kmar...	kmar...	-rw-rw-r--
Pipeline_AASPI-crop.H	H File	4544	02/05/2022 09:44:38	kmar...	kmar...	-rw-rw-r--
Pipeline_AASPI-crop.H@	H@ File	40783...	02/05/2022 09:44:38	kmar...	kmar...	-rw-rw-r--
Pipeline_AASPI-crop.H@@	H@@ File	2183	02/05/2022 09:44:38	kmar...	kmar...	-rw-rw-r--
Pipeline_AASPI-crop.H@@	H@@@ File	9386720	02/05/2022 09:44:39	kmar...	kmar...	-rw-rw-r--

Next, (2) I assign reflection coefficients to each selected horizons. In this example, I first highlight Horizon_6.dat and then type in the number -0.5 (also highlighted in blue). As with almost all AASPI programs, I define a (3) *Unique project name* and *Suffix* that are used in constructing the output file names.

The Horizon parameters tab

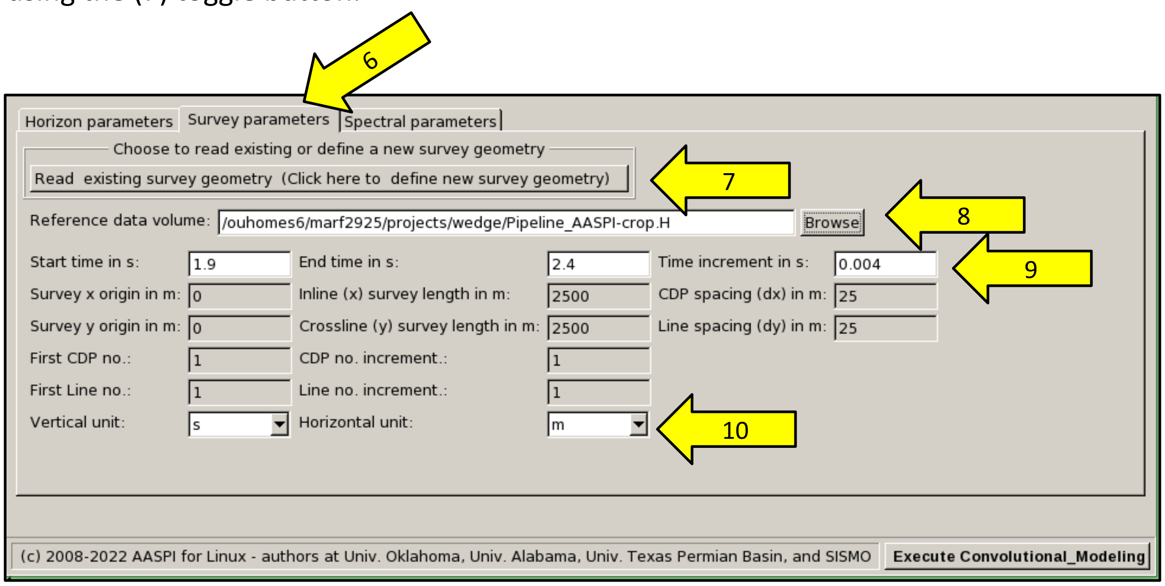
Next, I select the (4) *Horizon parameters* tab and then (5) choose the horizon type as being either *gridded* or *interpolated*. A *gridded* horizon (e.g. *Earthvision* format) forms a rectangular mesh of points on a North-South East-West grid, regardless of the orientation of the seismic survey. Such gridded horizons work not only for seismic surveys, but also for acreage when only irregularly spaced well tops are available. In contrast, an *interpolated* horizon (e.g., *SeisX* format) provides a pick for each seismic trace for which the horizon exists, where the coordinates of the horizon are defined by inline and crossline (CDP) numbers. In all cases, the input parameters can be delimited using either spaces or commas. Not all of the *Earthvision* header information or columns are needed for program convolutional_modeling, which is useful if you wish to create a model by typing, by using **MatLab**, or even **Excel.** For the 3-wedge problem, Horizon_6a.dat looks like this:


```
#       Grid_size:       2       x         2
#       Grid_space: 0.0, 2500.0, 0.0, 2500.0
#       Z_units:         milliseconds
0.0             0.0             600
2500.0          0.0             700
0.0             2500.0          600
2500.0          2500.0          700
```

where the #-sign indicates a comment card in *Earthvision* format. Here, I am defining a 2x2 grid (a rectangle) that ranges from 0.0 to 2500.0 m in the N-S direction and 0.0 to 2500.0 m in the E-W direction. Program convolutional_modeling only uses the first three columns to define the horizon as (x, y, z) triplets where x is northing and y is easting. The grids are defined South-to-North column by South-to-North column, with the first one being the most westerly and the last the most easterly. In this example Horizon_6.dat defines a planar reflector dipping from 600 ms in the South to 700 ms in the North.

The Survey parameters tab

Next, select the (6) Survey parameters tab. There are two ways to define the survey parameters: by reading an existing survey geometry, or by defining a new survey geometry, which selected using the (7) toggle button.



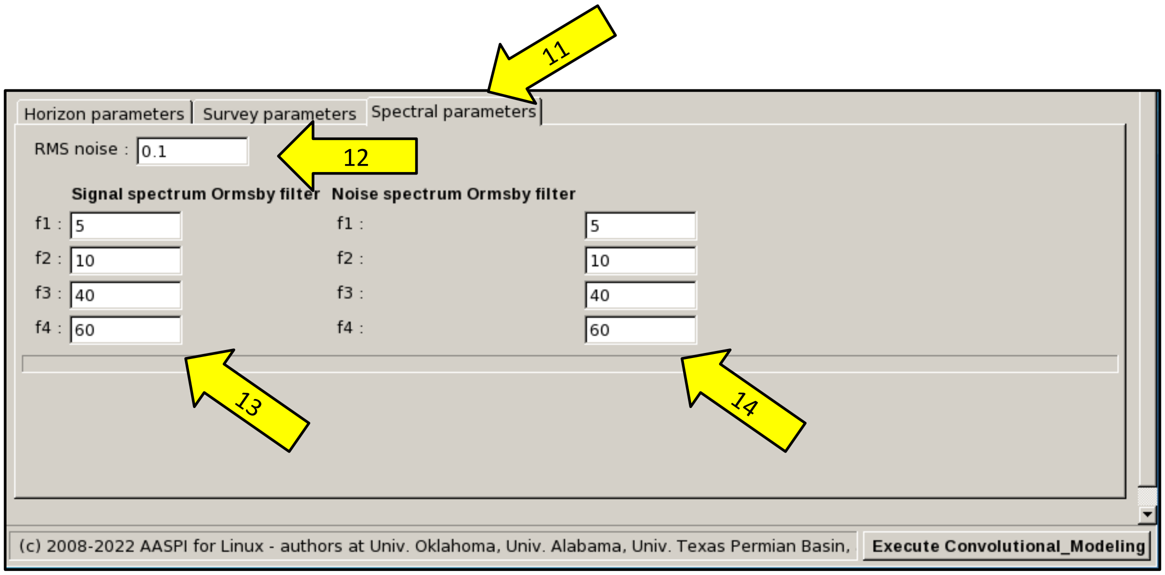
If I choose to read an existing survey geometry, the area containing the (8) *Reference data volume* bane will be white, allowing me to *Browse* and fill it in. I am also able to change the (9) vertical limits and the (10) units of the output synthetic. In contrast, the entries in gray, although defaulted will not be used in defining the geometry of the survey.

For the wedge problem, I have much more control by explicitly defining the geometry of the synthetic volume to meet my needs. I therefore (7) toggle to choose to Define new survey geometry. The Reference data volume area is now gray and disabled whereas the other parameters are now white and enabled. There are a few constraints that are familiar to data loaders, but perhaps not to less experienced interpreters. First, all inline and crossline (CDP) indices must be greater than 0. They can be numbered backwards. The length in the x (N-S, or inline) direction and in the y (E-W or crossline) direction must be positive. The CDP no. and Line no. increments must be integers, with 1 being the most common value, but 2, 3, and 4 seen in many marine surveys. The CDP spacing and the Line spacing is the distance between adjacent

CDP no. and Line no. values. Thus, if the *Line no. increment* is 2 and the *Line spacing* is 25 m, the spacing between traces in the crossline direction will be 50 m.

The spectral parameters tab

Now turn to the (11) Spectral parameters tab



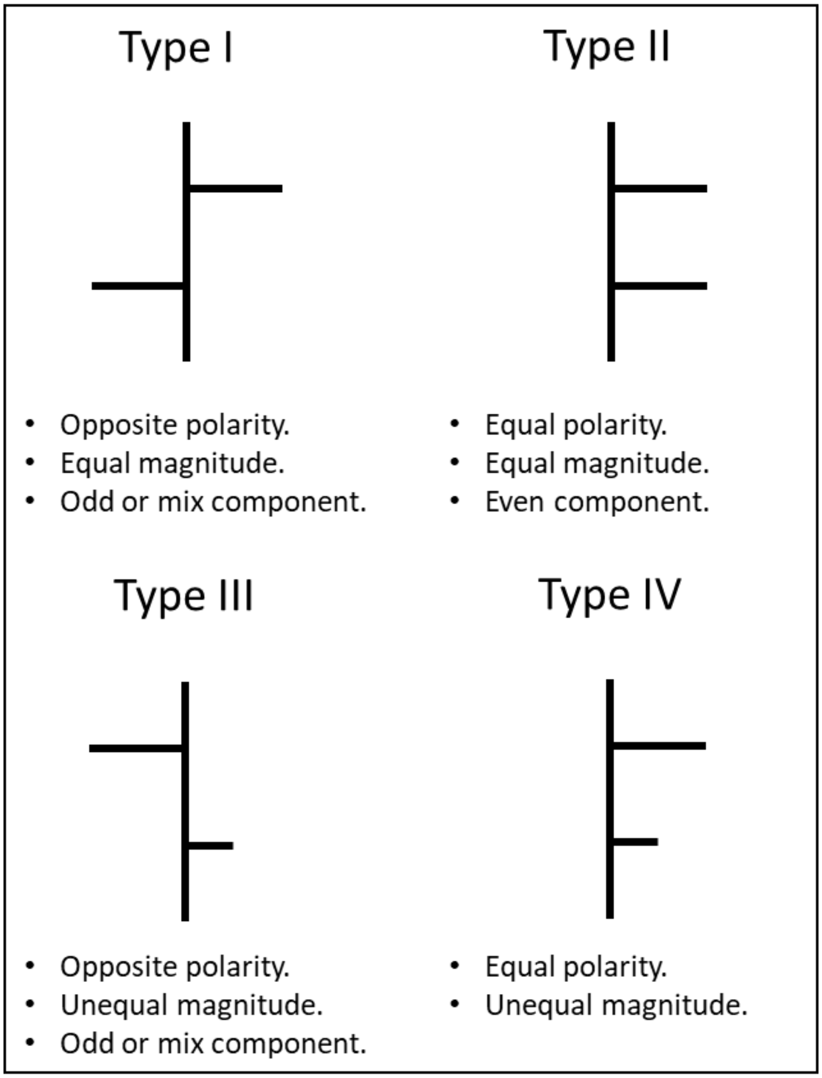
First, define the (12) *RMS noise* level for each trace. Then define the spectrum of the (13) signal and filter (14) applied to the random noise. All seismic data suffer from some level of seismic noise, where the (not modeled!) coherent noise is the most problematic. In the case of random noise, the range of the noise is usually similar to that of the signal, in that both signal and noise were filtered using the same processing parameters. In contrast, high frequency random noise going to Nyquist is rarely seen on real seismic data. The source wavelet will be a zero phase Ormsby wavelet defined by the four corner points. Side lobes diminish if the distance between f_1 and f_2 and between f_3 and f_4 are an octave or greater. In this example the difference between f_3 and f_4 is less than an octave (where we would need to set f_4 to be $2f_3$=80 Hz).

Once all parameters have been selected, drop down and click *Execute Convolution_Modeling* to execute the program

Examples

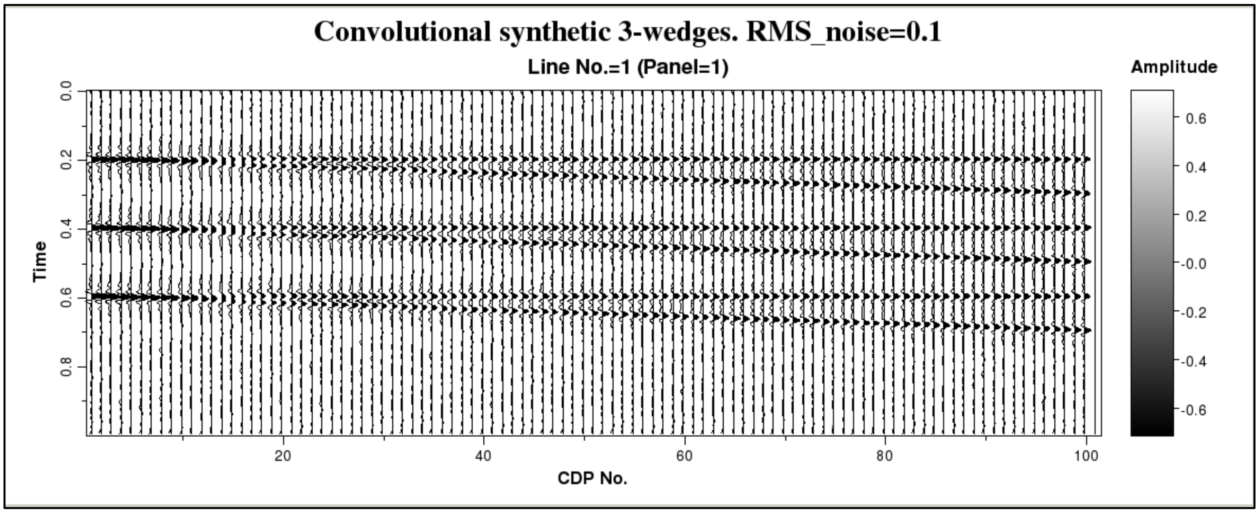
Example 1: 3 wedges

The wedge model is one of the most common models used to calibrate problems of seismic resolution. Here I defined three wedges based on Chung and Lawtons's (1995) thin bed reflectivity types I, II, and IV:

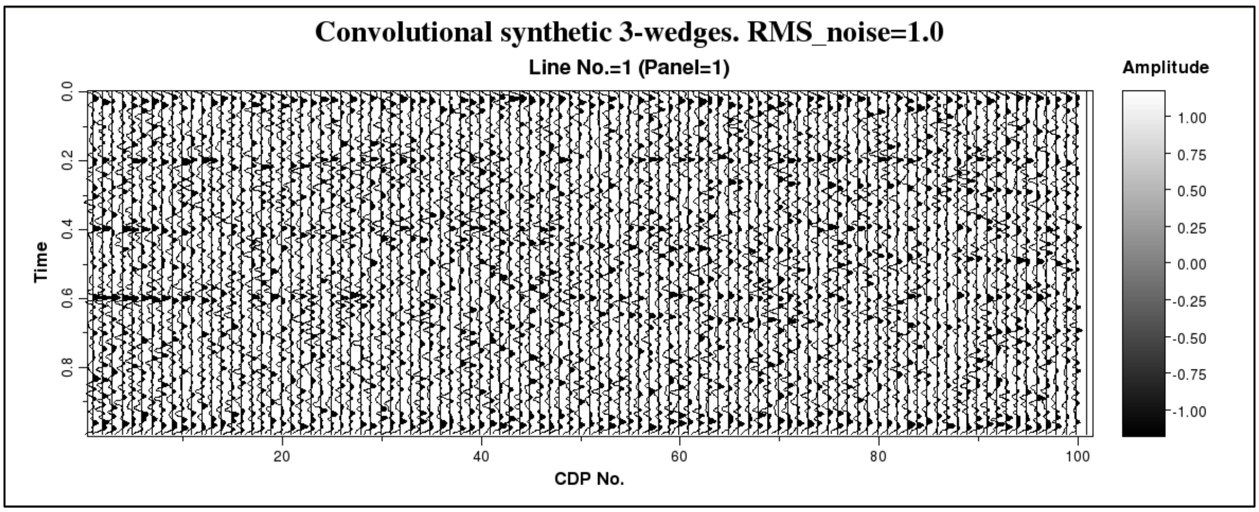


Reflectivity types defined by Chung and Lawton (1995).

My synthetic seismogram for the three wedges with an RMS noise level of 0.1 looks like this:

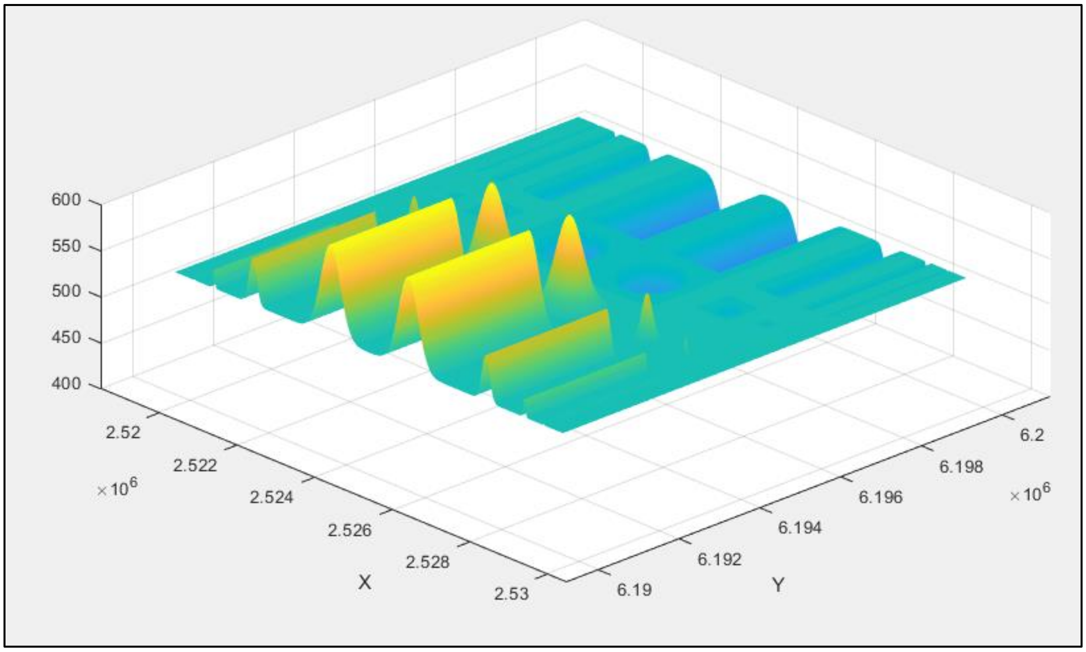


Increasing the RMS noise to be 1.0 gives this:

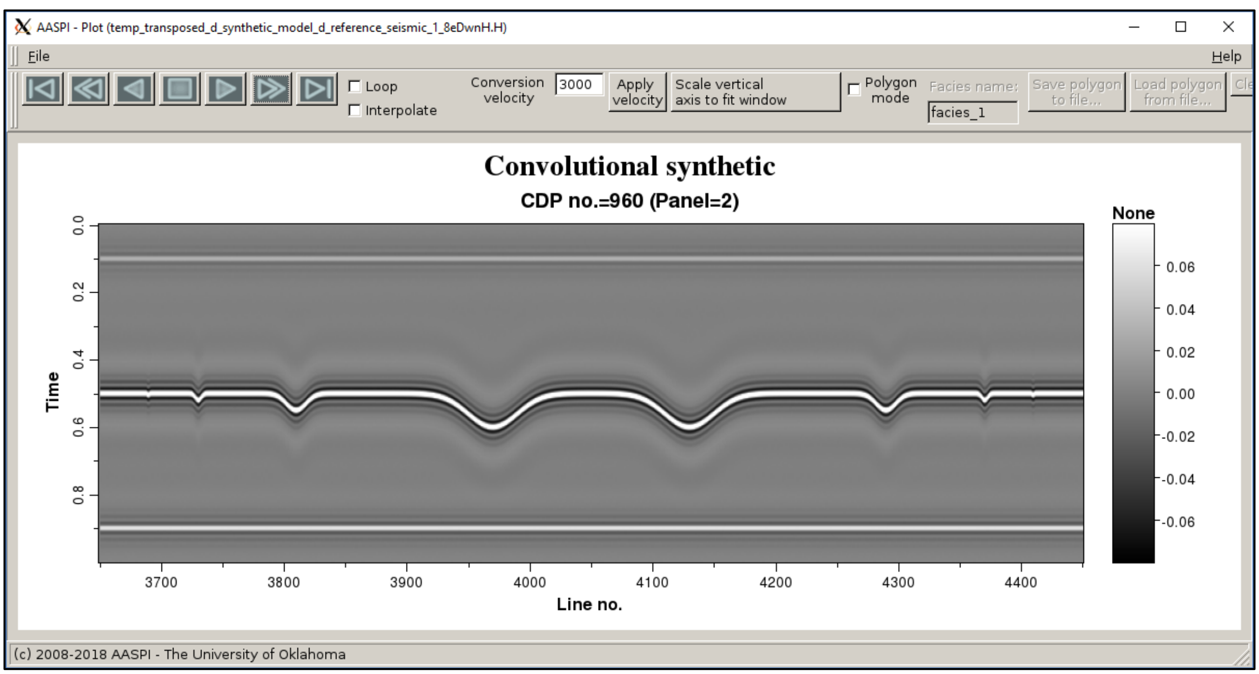


Example 2: Horizons to calibrate the response of curvature algorithms

The following horizons were constructed by Sumit Verma to better understand the behavior of 3D curvature and aberrancy to deformed surfaces. The middle, deformed surface exhibits ridges, valleys, sinkholes, and domes was constructed using MatLab and looks like this:

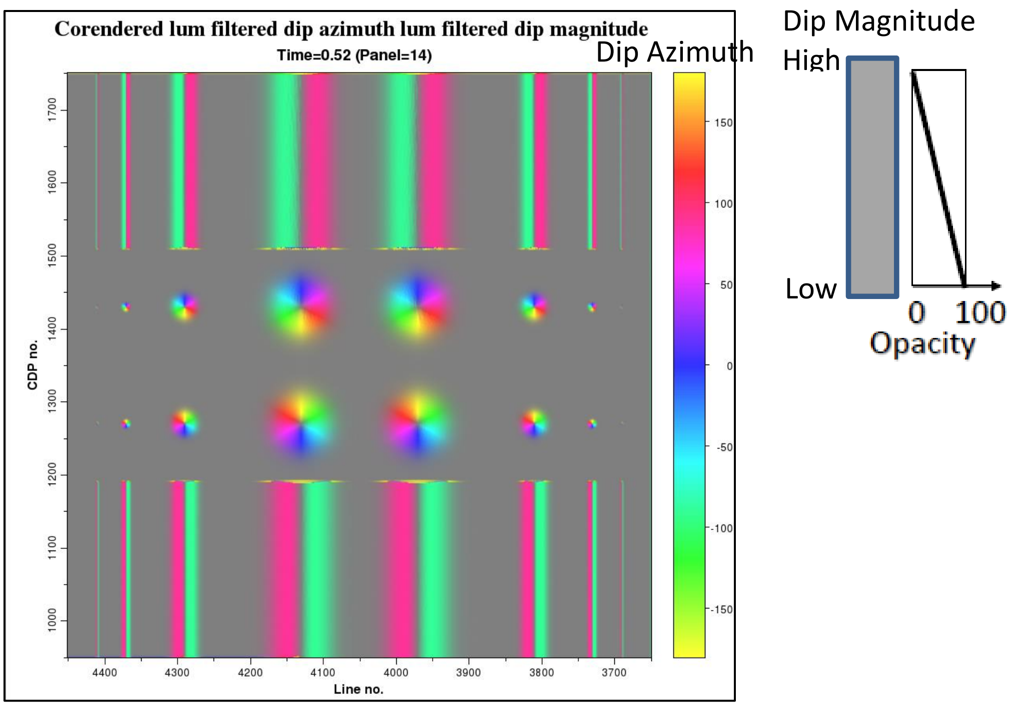


This surface was encased by two flat horizons, one at t=100 ms, the other at t=900 ms. These three surfaces were input into program convolutional_modeling, resulting in the representative vertical slice shown here:

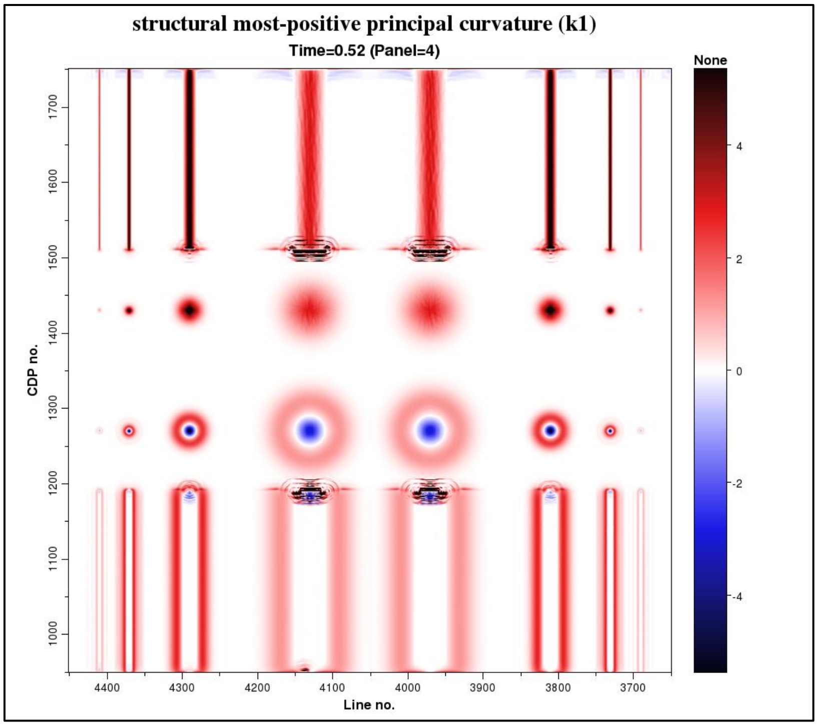


The input horizons Surface_100ms and Surface_900ms are horizons with a constant value of 100ms and 900ms respectively, whereas the Surface_500ms, has ridges, valleys, sinkholes, and domes.

The resulting synthetic *d_synthetic_ridge_valley_dome_bowl.H* was input into program **dip3d**, resulting in the following corendered time slice at *t*=0.520 s through dip azimuth and dip magnitude:

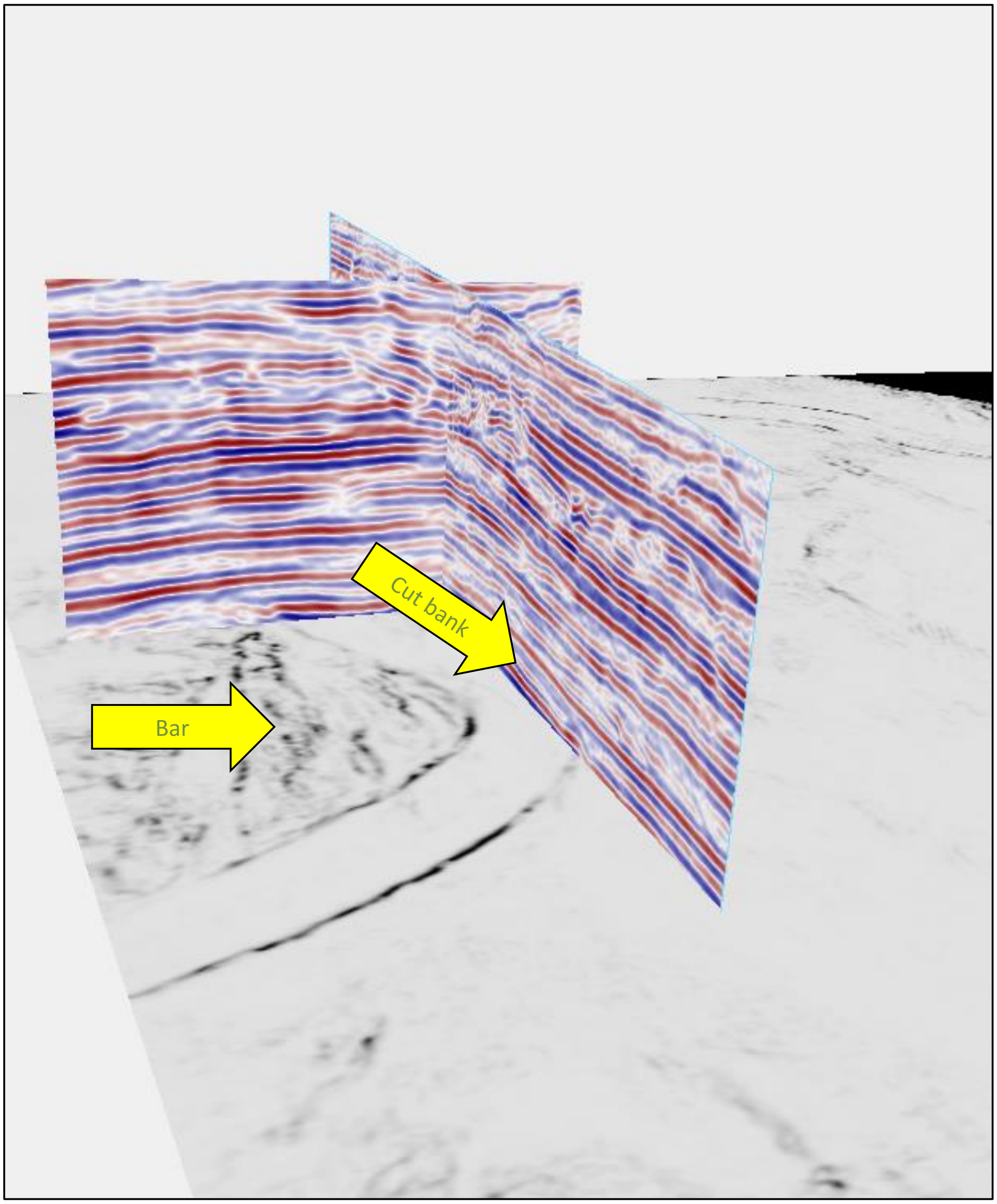


The inline and crossline dip volumes from **dip3d** were then input into program **curvature3d**, resulting in the following time slice at *t*=0.520 s through most-positive k_1 curvature:


structural most-positive principal curvature (k1)
Time=0.52 (Panel=4)
None
4
2
0
-2
-4
CDP no.
1700
1600
1500
1400
1300
1200
1100
1000
4400
4300
4200
4100
4000
3900
3800
3700
Line no.

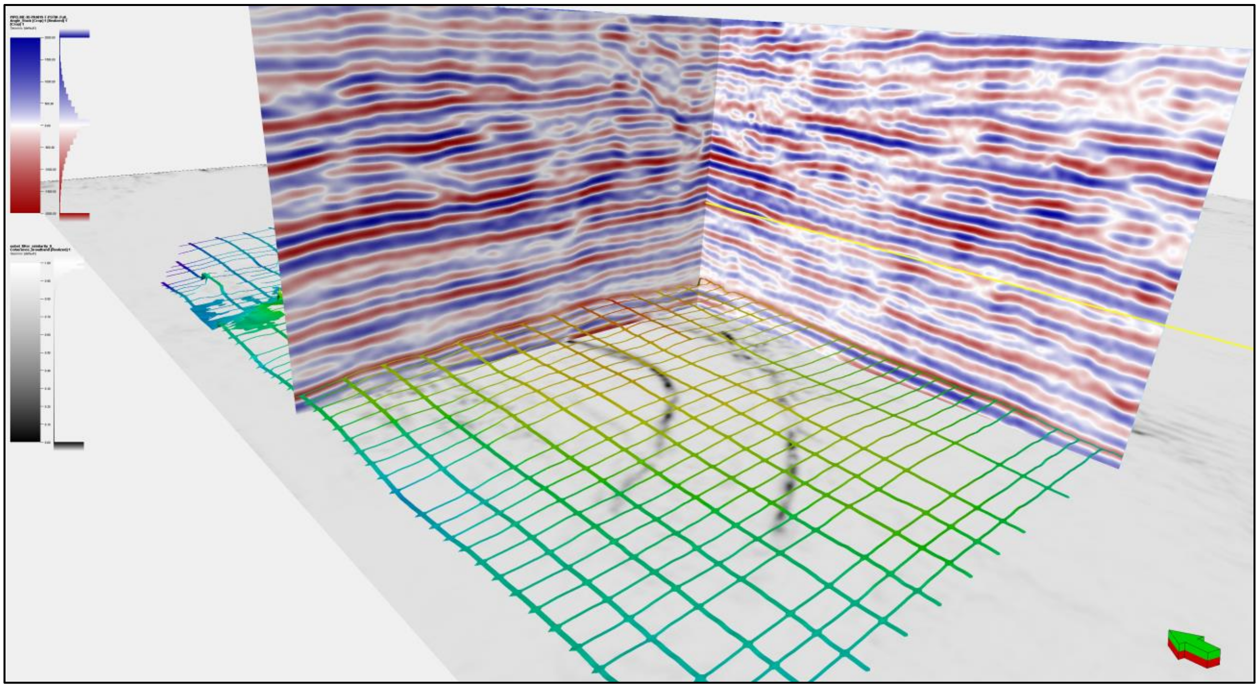
Example 3: Synthetics from horizons picked on a 3D seismic volume

The next example was generated by Karelia LaMarca, who hand-picked horizons defining a meandering channel in the Taranaki Basin Pipeline3d survey, New Zealand. Here is the channels imaged by Sobel filter similarity displayed in Petrel where she has identified a few of the

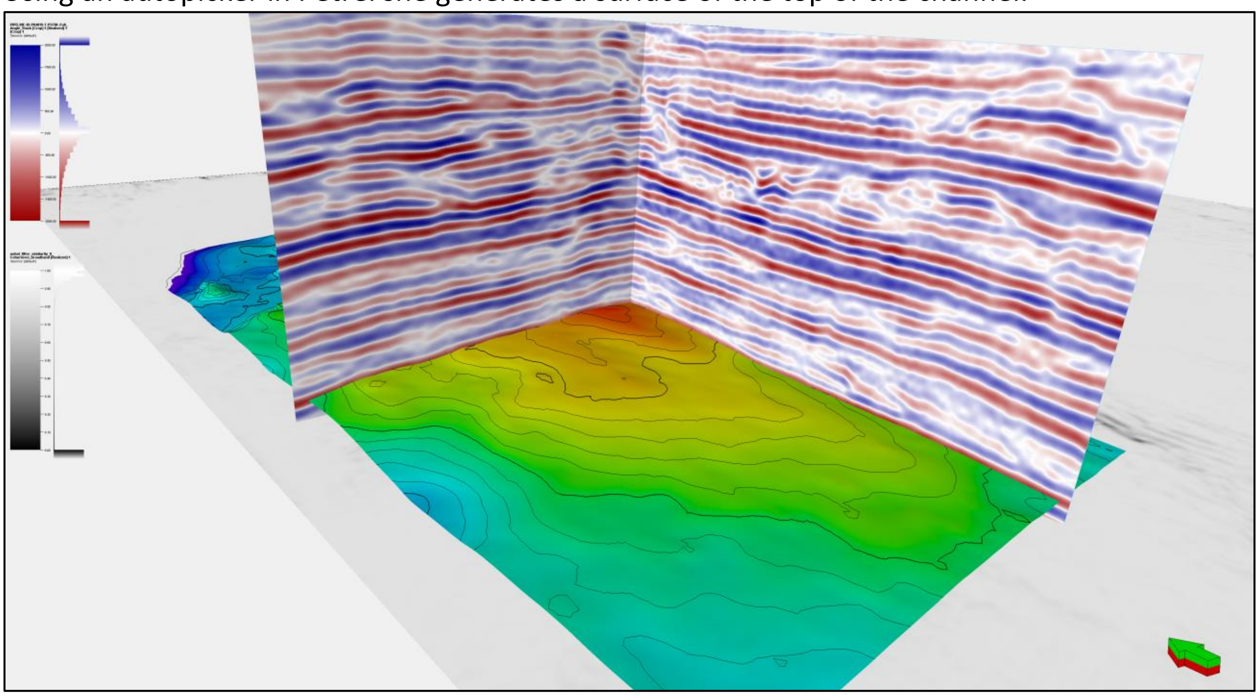


architectural elements:

Next she picks the top and base of the channel using a 10 line by 10 crossline grid:

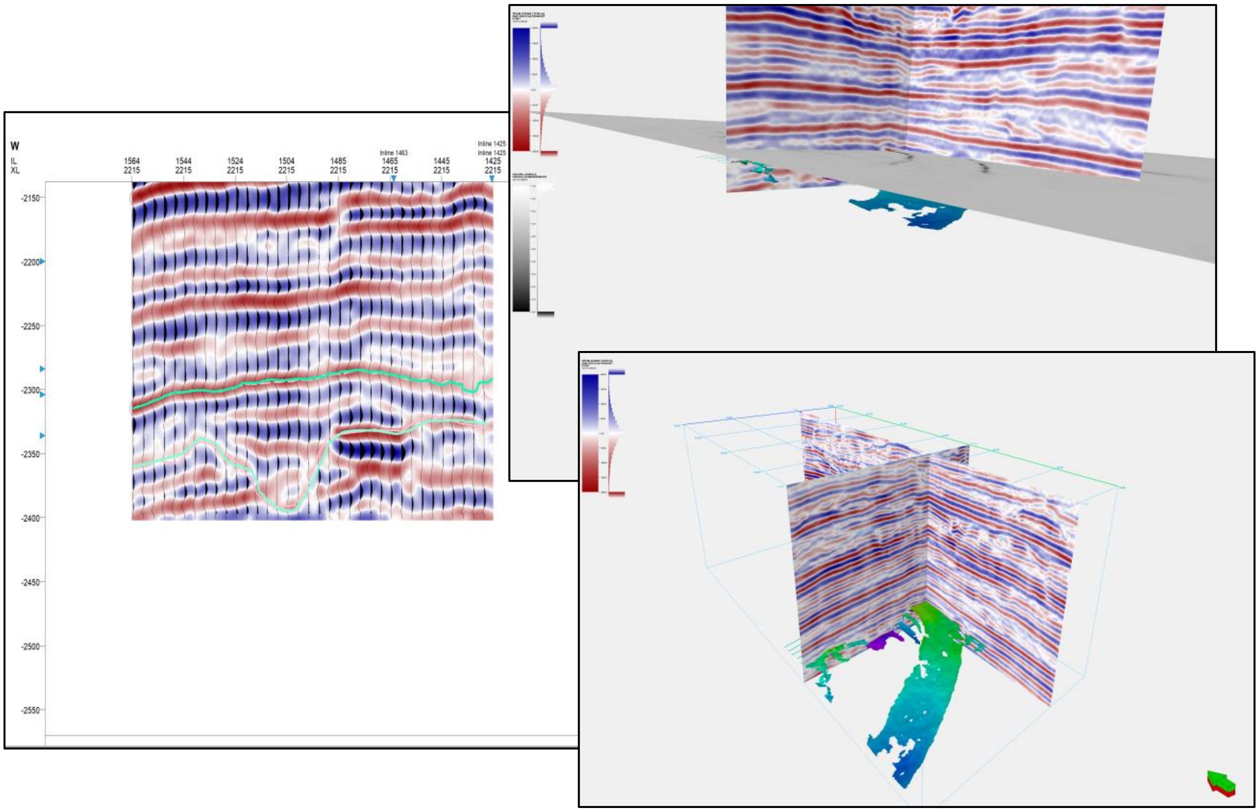


Using an autopicker in Petrel she generates a surface of the top of the channel:

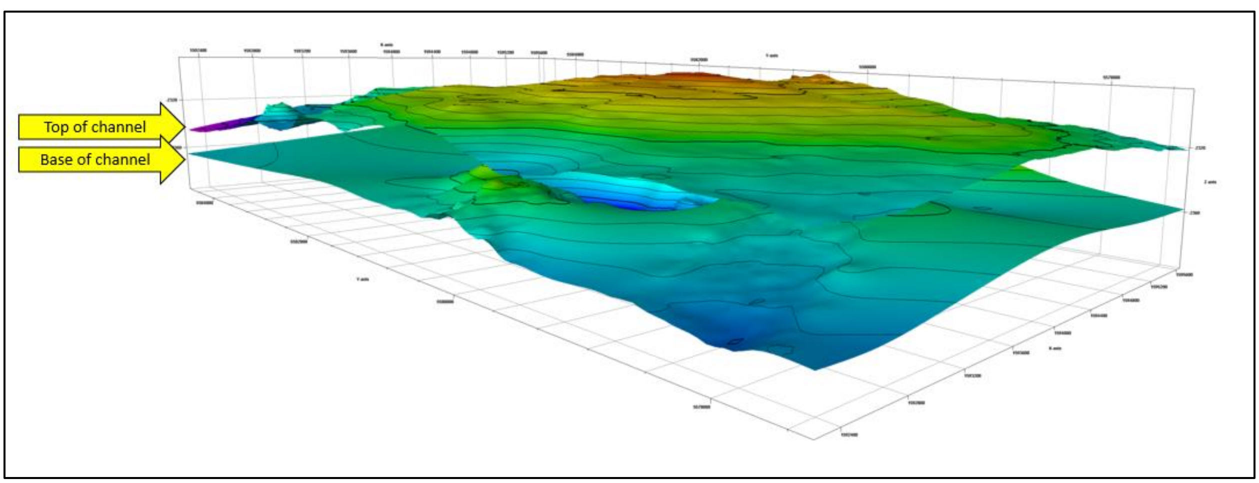


And the base of the channel:

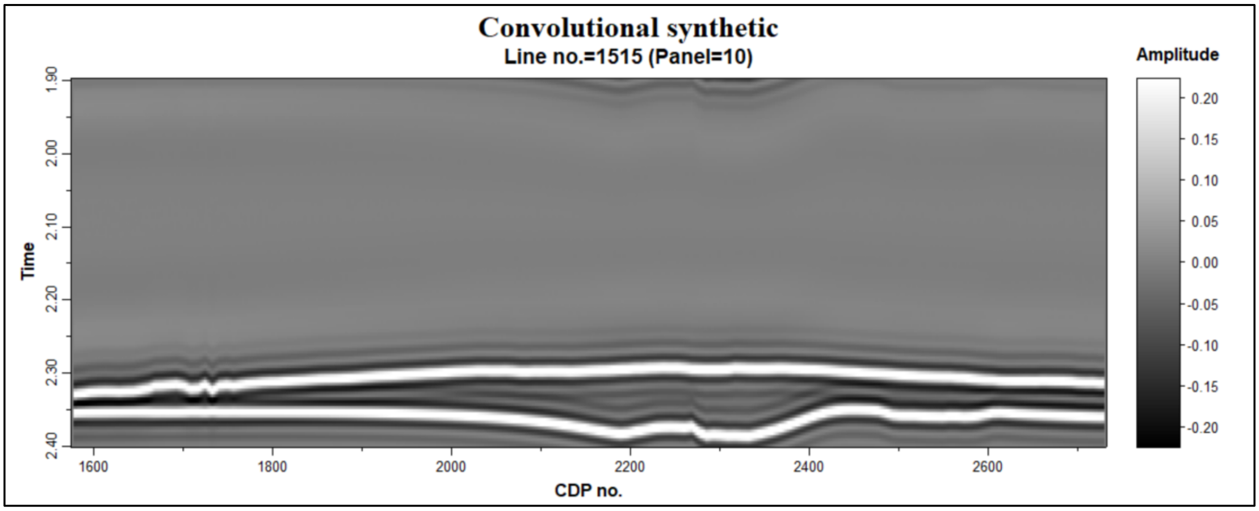
Other Utilities: Program **convolutional_modeling**



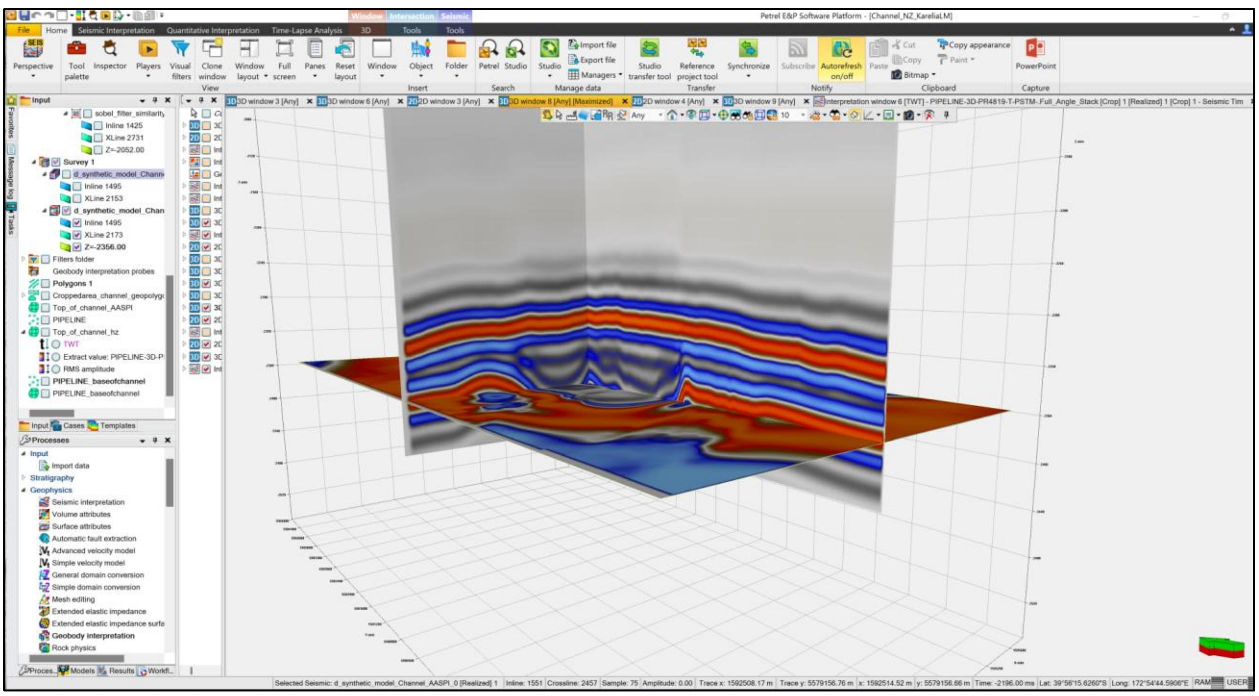
The final two surfaces look like this:



These surfaces and the original Pipeline3d survey are then used as input to convolutional_modeling resulting in the following representative vertical slice:



The synthetic can then be converted into SEGY format and reimported into Petrel, resulting in an idealized channel synthetic that will allow here to more clearly understand the limits to vertical resolution and attribute sensitivity in imaging the cut bank and point bar flanks of the channel.



References

Chung, H., and D. C. Lawton, 1995, Frequency characteristics of seismic reflections from thin beds: Canadian Society of Exploration Geophysicists Recorder, **31**, 32-37.