**COMPUTING REFLECTOR CURVATURE, ROTATION, CONVERGENCE, and ABERRANCY – PROGRAM *curvature3d*****Contents**

Overview	2
Definition of azimuths and strikes	3
Approximations in computing curvature from time-migrated data volumes	3
Computation flow chart for <i>structural</i> curvature	3
Output file naming convention for structural curvature	5
Invoking the curvature3d GUI and computing <i>structural</i> curvature	6
Horizon-based curvature	10
Volumetric curvature	Error! Bookmark not defined.
Defining and displaying operators and operator spectra used in curvature3d	18
Fractional Derivative Operators.....	25
Implementation of derivative operators in 3D	31
Implementation of Operators in 3D	31
Compressing the vertical operator to reduce vertical mixing	33
Computation flow chart for <i>amplitude</i> curvature	37
Output file naming convention for amplitude curvature	38
Invoking the curvature3d GUI and computing amplitude curvature	39
Plotting shape components	42
Curvedness and the Shape Index.....	44
Plotting shape components	50
Plotting lineament intensity vs. lineament strike	52
Reflector Convergence.....	56
Reflector Convergence.....	64
Reflector Rotation about the Normal	65
Reflector Rotation.....	65
Aberrancy	67
Aberrancy and the 2 nd derivatives of the dip vector, p	69
Aberrancy computation	70
More Examples of Curvature	75
Examples showing most-positive and most-negative principal curvatures	75

Geometric Attributes: Program **curvature3d**

Examples showing Curvedness and Shape Index and Shape Components.....	76
Examples of Reflector Convergence	82
Examples showing rotation about the normal	87
Examples showing structural lineaments	88
Example showing very long curvature anomalies	88
References	91

Overview

Program **curvature3d** has two modes – computing second derivatives of the two-way travel time or depth of seismic waveforms (1^{st} derivatives of structural dip components) which we call structural curvature, and computing 2^{nd} derivatives of amplitude, envelope, or impedance (1^{st} derivatives of amplitude gradients) which we call *amplitude* curvature. In addition to plotting the various attribute volumes, we are also able to plot the differential operators and their corresponding wavenumber spectra. In general, second derivatives increase the high frequency (short-wavelength), noisier components of the seismic data. Program **curvature3d** differs from most other curvature computations in that it provides the means to enhance the lower-frequency (long-wavelength) less-noisy components of the seismic data. The original implementation described by al-Dossary and Marfurt (2006) was achieved by replacing the first-derivative operator (ik in the wavenumber domain where $i = (-1)^{1/2}$ and k is the wavenumber) by a fractional-derivative-like operator (ik^α in the wavenumber domain). An astute mathematician will note that this is not exactly a fractional derivative, since we use ik^α rather $(ik)^\alpha$, such that a more accurate way of viewing the ‘fractional derivative’ is as a 1^{st} derivative followed by a low-pass filter ($ik/k^{1-\alpha}$ in the wavenumber domain). Applying an explicitly defined band-pass filter to the derivative operators is more easily understood by seismic processors than the somewhat heuristic fractional derivative operator.

For structural curvature the derivatives are computing in 3D using cascaded convolutional operator in the in the inline and crossline directions and Fourier operators along the vertical direction. This generalization provides significantly improved vertical images of the curvature attributes by eliminating vertical discontinuities due to changes in dip associated with the dominant reflection event. The 3D generalization also facilitates the introduction of several new attributes that require computation of the vertical derivatives including reflector rotation about the normal, and reflector convergence that maps lateral changes in thickness as well as angular unconformities. The amplitude curvature inline and crossline derivative operators are also 3D; however, there are no vertical derivatives because the axis of the vertical axis is either time or depth rather than amplitude.

Geometric Attributes: Program **curvature3d**

We break the discussion of the **curvature3d** GUI into three parts: (a) computing structural curvature, (b) defining and displaying the derivative operators and their spectra, and (c) computing amplitude curvature.

Definition of azimuths and strikes

Like all other AASPI programs, azimuth and strike are measured from the North where an E-W strike is at $\pm 90^\circ$, a NE-SW strike at $+45^\circ$, a N-S strike at 0° , and a NW-SE strike at -45° . Because of the discontinuity at $\pm 90^\circ$, you can not flatten or stratal slice a strike volume by itself; rather you must pair the k1_strike (ψ_1) with k1 (k_1) and the k2_strike (ψ_2) with k2 (k_2) and use programs **vector_flatten** or **vector_stratal_slice**.

The reflector convergence and aberrancy azimuths (the direction of a pinch out for convergence or a downward-oriented flexure for aberrancy) are defined in the same manner as the dip azimuth, ranging between -180° and $+180^\circ$ and is defined clockwise from North, where North is at 0° , East at 90° , West at -90° , and South at $\pm 180^\circ$. The discontinuity in the azimuth about $\pm 180^\circ$ is best addressed through the use of a cyclical color bar. Like strike, it needs to be interpolated as a vector using programs **vector_flatten** or **vector_stratal_slice**.

Approximations in computing curvature from time-migrated data volumes

Because the principal curvatures k_1 and k_2 are defined as the inverse of radii of curvature that best fit the local structural. For this reason, these curvature computations need to be computed on depth data. If the seismic amplitude data are depth migrated or stretched to depth from a time-migrated volume, the computational results represent the features seen in the subsurface seismic image. If the seismic amplitude data are time migrated, the input inline and crossline dip components are internally (and crudely) converted to measures in m/m or ft/ft using the same single constant conversion velocity provided as input to the program **dip3d** GUI. These inaccuracies in dip are carried over into subsequent curvature, reflector convergence, and aberrancy magnitude values but have minimal impact on their azimuth or strike.

The historical measures of most-positive and most-negative structural curvature k_{pos} (k_{pos}) and k_{neg} (k_{neg}) do not compensate for the average dip at each voxel and are better interpreted as a 2nd derivative map of structure. Likewise, because the vertical axis is time or depth but the input derivatives are amplitude gradients, amplitude curvatures e_{pos} (e_{pos}) and e_{neg} (e_{neg}) should be interpreted as a 2nd derivative map of amplitude.

Computation flow chart for *structural* curvature

The input to program **curvature3d** will be the inline and crossline components of dip computed using programs **dip3d** or **filter_dip_components**. The basic curvature outputs include the value

Geometric Attributes: Program **curvature3d**

and strike (eigenvalue and eigenvector) of the most-positive and most-negative principal curvatures, k_1 and k_2 . Internal to the program, these principal curvatures are combined to construct the curvedness and shape index, which in turn can be used to generate the dome, ridge, saddle, valley, and bowl shape components. Measures of a non-quadratic surface includes the reflector rotation about the smoothed normal which can indicate wrench faults and lateral changes in reflector thickening, and the magnitude and azimuth of reflector convergence which can be used to map clinoforms, fans, levees, and angular unconformities. A suite of volumes of historical interest can be optionally generated including the most-positive and most-negative, mean, strike, dip, and Gaussian curvatures. These volumes may be useful in comparing the results of **curvature3d** to other software implementation packages but in general provide less useful results than the two principal curvatures.

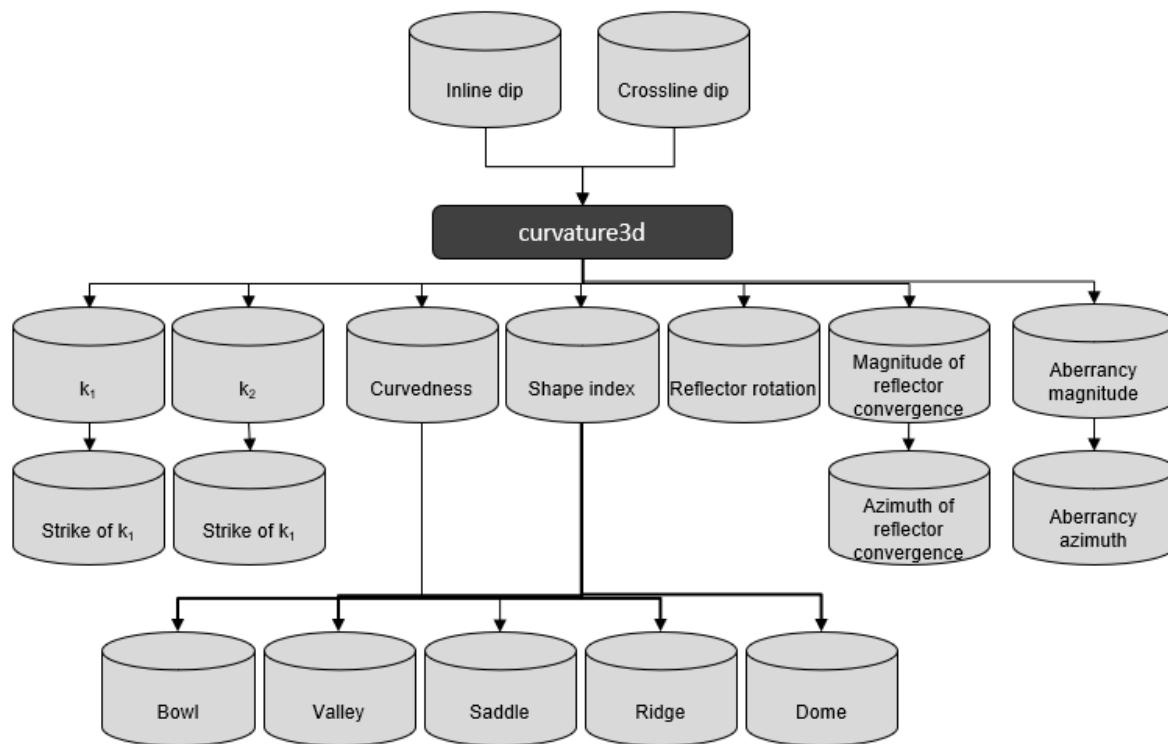


Figure 1.

Geometric Attributes: Program **curvature3d**

Output file naming convention for structural curvature

By default, program **curvature3d** will generate the following output files:

Output file description	File name syntax
Program log information	curvature3d_ <i>unique_project_name_suffix</i> .log
Program error/completion information	curvature3d_ <i>unique_project_name_suffix</i> .err
Filtered curvature derivative	d_dr_curvature_operator_ <i>unique_project_name_suffix</i> .H
Inline curvature derivative operator	d_dx_curvature_operator3d_ <i>unique_project_name_suffix</i> .H
Crossline curvature derivative operator	d_dy_curvature_operator3d_ <i>unique_project_name_suffix</i> .H
Vertical curvature derivative operator	d_dz_curvature_operator3d_ <i>unique_project_name_suffix</i> .H
Most-positive principal curvature value	k1_ <i>unique_project_name_suffix</i> .H
Strike of most-positive principal curvature	k1_strike_ <i>unique_project_name_suffix</i> .H
Most-negative principal curvature value	k2_ <i>unique_project_name_suffix</i> .H
Strike of most-negative principal curvature	k2_strike_ <i>unique_project_name_suffix</i> .H

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.

If selected, you can also output the following volumes:

Output file description	File name syntax
Reflector rotation about the normal	k_rot_normal_ <i>unique_project_name_suffix</i> .H
Reflector convergence magnitude	k_converge_mag_ <i>unique_project_name_suffix</i> .H
Reflector convergence azimuth	k_converge_azim_ <i>unique_project_name_suffix</i> .H
Curvedness	k_curvedness_ <i>unique_project_name_suffix</i> .H
Shape index	k_shape_index_ <i>unique_project_name_suffix</i> .H
Dome shape component	k_dome_ <i>unique_project_name_suffix</i> .H
Ridge shape component	k_ridge_ <i>unique_project_name_suffix</i> .H
Saddle shape component	k_saddle_ <i>unique_project_name_suffix</i> .H
Valley shape component	k_valley_ <i>unique_project_name_suffix</i> .H
Bowl shape component	k_bowl_ <i>unique_project_name_suffix</i> .H

Geometric Attributes: Program **curvature3d**

If you choose to generate aberrancy, you will obtain the following files:

Output file description	File name syntax
Filtered aberrancy derivative	d_dr_aberrancy_operator_unique_project_name_suffix.H
Inline aberrancy derivative operator	d_dx_aberrancy_operator3d_unique_project_name_suffix.H
Crossline aberrancy derivative operator	d_dy_aberrancy_operator3d_unique_project_name_suffix.H
Vertical aberrancy derivative operator	d_dz_aberrancy_operator3d_unique_project_name_suffix.H
Total aberrancy magnitude	ab_tot_mag_unique_project_name_suffix.H
Total aberrancy azimuth	ab_tot_azim_unique_project_name_suffix.H

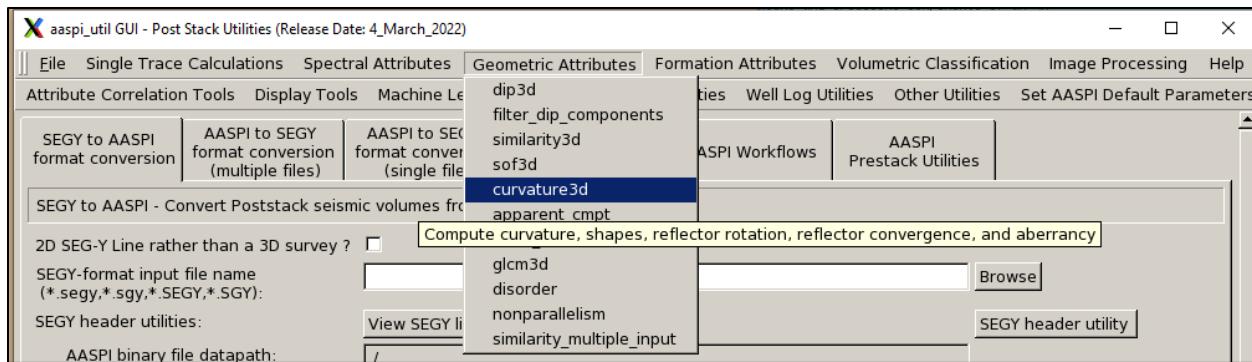
Finally, you also have the option to generate some (less useful) historical attributes, as well as the three components of aberrancy that when vector summed, form the total aberrancy vector:

Output file description	File name syntax
Maximum curvature value	k_max_unique_project_name_suffix.H
Maximum curvature azimuth	k_max_azim_unique_project_name_suffix.H
Minimum curvature value	k_min_unique_project_name_suffix.H
Minimum curvature azimuth	k_min_azim_unique_project_name_suffix.H
Positive curvature (uncorrected for dip)	k_pos_unique_project_name_suffix.H
Strike of positive curvature	k_pos_strike_unique_project_name_suffix.H
Negative curvature (uncorrected for dip)	k_neg_unique_project_name_suffix.H
Strike of negative curvature	k_neg_strike_unique_project_name_suffix.H
Maximum aberrancy_magnitude	ab_max_mag_unique_project_name_suffix.H
Maximum aberrancy_azimuth	ab_max_azim_unique_project_name_suffix.H
Intermediate aberrancy_magnitude	ab_int_mag_unique_project_name_suffix.H
Intermediate aberrancy_azimuth	ab_int_azim_unique_project_name_suffix.H
Minimum aberrancy_magnitude	ab_min_mag_unique_project_name_suffix.H
Minimum aberrancy_azimuth	ab_min_azim_unique_project_name_suffix.H

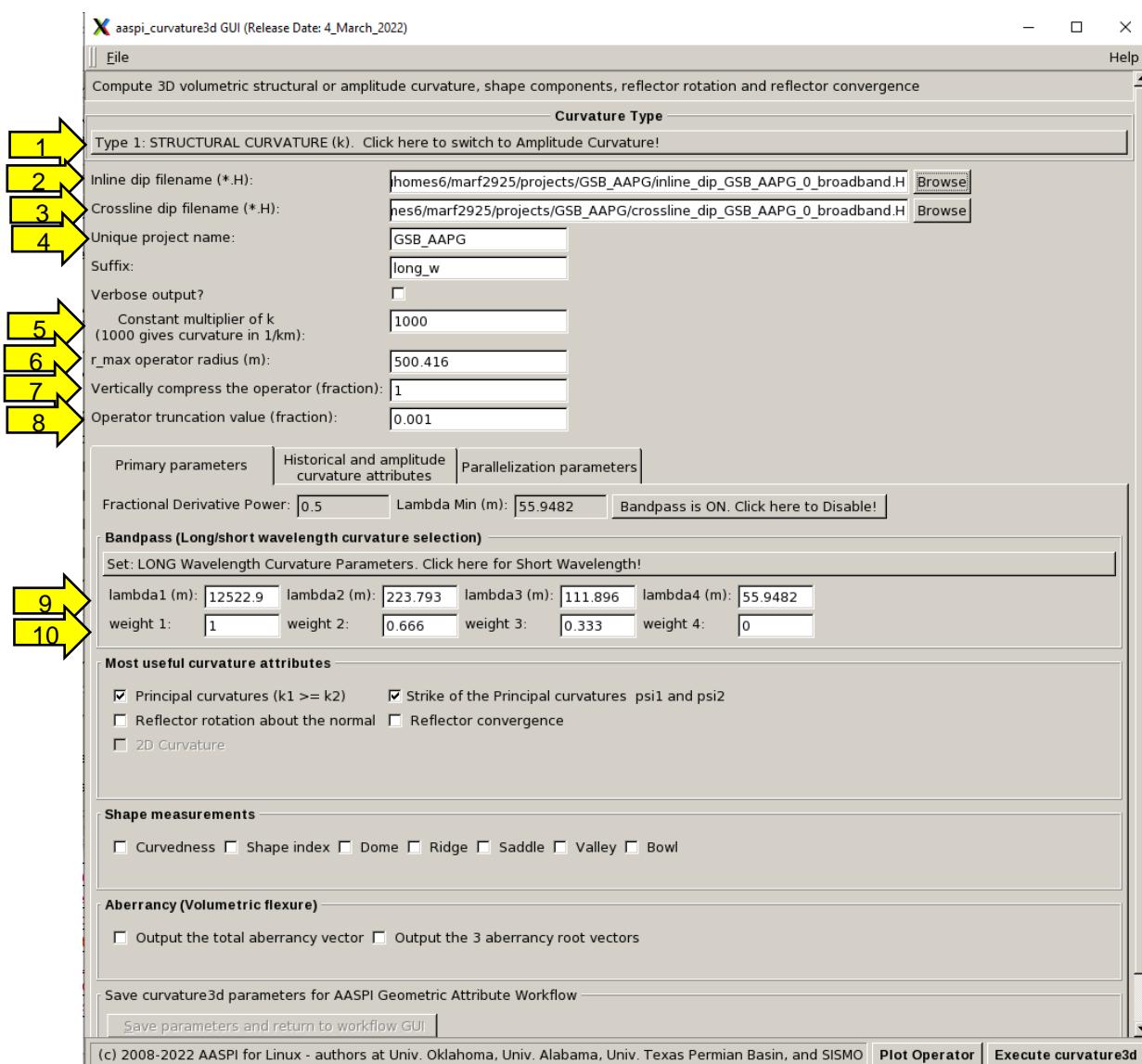
Invoking the **curvature3d** GUI and computing *structural* curvature

Now we are ready to run curvature. Select ‘**curvature3d**’ under the **aaspi_util Geometric Attributes** tab as shown below:

Geometric Attributes: Program curvature3d



The following GUI should appear:



Geometric Attributes: Program **curvature3d**

curvature3d has two modes of operation. The easier-to-understand mode is to (1) compute *structural curvature*. All output attribute files will begin with the character string 'k'. We will cover amplitude curvature later in this document. Our input files are (2) *inline_dip_GSB_AAPG_0_broadband.H* and (3) *crossline_dip_GSB_AAPG_0_broadband.H* computed from program **dip3d**. By default (4) the *Suffix* will read 'long_w' and (5) the *Constant multiplier of k* is 1000, giving us curvature measurements in kft or km. Normally, you should not need to change (6) the *r_max operator radius*. Improved computation speed can occur by reducing them, but only do so after inspecting the operator and spectrum after a first attempt with the default parameters. By default, the derivative operators applied in **curvature3d** will be isotropic, exhibiting amplitudes that decay the same in the x, y, and z directions. In some cases, such as shale resource plays in much of the USA, the regional dip is less than 10°. In this case, one can improve the vertical resolution of the curvature computation by defining an anisotropic (shorter vertically than laterally) operator, say by setting the value (7) to be 0.5. In theory, these operators are infinitely long. Option (8) allows one to truncate the computation (the default is 0.001 of the maximum operator value) to provide more efficient computation when the operator window defined by r_{\max} is larger than needed.

Next you see the (9) and (10) parameters that define the filter applied to the curvature operator defined in terms of wavelengths rather than wavenumbers. The weights are interpolated linearly between the four filter definition points.

Set: LONG Wavelength Curvature Parameters. Click here for Short Wavelength!			
lambda1 (m):	12522.9	lambda2 (m):	223.793
weight 1:	1	weight 2:	0.666
lambda3 (m):	111.896	lambda4 (m):	55.9482
weight 3:	0.333	weight 4:	0

Do not change (9) the default value of *lambda4* to be less than 56 m. Since the GSB data were sampled using 25 m x 12.5 m bins, Nyquist sampling criteria requires two samples per wavelength ($2 \times [50^2 + 25^2]^{1/2} \approx 56$ m for a structural feature aligned diagonal to the inline and crossline axes). Keep the weight parameters (10) as default for this case. By default, *lambda1* is set to be the shortest width of the survey, or the longest wavelength that can be supported by the data. As described later in the documentation, the derivative operator before filtering behaves as ik , such that the resulting operator components at *lambda1* after filtering will be near zero. If you have two or more surveys with mixed sampling (say 12.5 m, 25 m, and 37.5 m) you may wish to apply the same value of *lambda4* to obtain consistent wavenumbers in the output curvature images.

After execution, I can list the files that have the string *_long_w*.H:

Geometric Attributes: Program **curvature3d**

```
kmarfurt@ediacaran:/projects/GSB_AAFC$ ls -ltr *long_w*H
-rw-rw-r-- 1 kmarfurt kmarfurt 304 Jan 25 10:00 d_dr_curvature_operator_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 305 Jan 25 10:00 d_dx_curvature_spectrum_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 5477 Jan 25 10:00 d_dx_curvature_operator3d_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 5477 Jan 25 10:00 d_dy_curvature_operator3d_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 5477 Jan 25 10:00 d_dz_curvature_operator3d_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 5539 Jan 25 10:10 k2_strike_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 5518 Jan 25 10:10 k2_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 5539 Jan 25 10:10 k1_strike_GSB_long_w.H
-rw-rw-r-- 1 kmarfurt kmarfurt 5518 Jan 25 10:10 k1_GSB_long_w.H
kmarfurt@ediacaran:/projects/GSB_AAFC$
```

Files beginning with the letter ‘K’ correspond to structural curvature (vs. amplitude or energy curvature, which will start with the letter ‘e’). *k1_GSB_long_w.H* and *k2_GSB_long_w.H* are the most-positive and most negative principal curvatures computed for the GSB survey using ‘long wavelength’ parameters (the four values of lambda1, lambda2, lambda3, and lambda4) from the filtered data. The *d_dr*.H*, *d_dx*.H*, *d_dy*.H*, and *d_dz*.H* are the first derivative operators computed by applying the 4-point filter to the derivative *ik* in the wavenumber domain. We will examine these later in this document.

Horizon-based curvature

Curvature requires the computation of second derivatives of a real or implied surface. For horizon-based curvature (e.g. Roberts, 2000), one locally fits a 2nd degree polynomial of the form

$$z(x, y) = a \frac{\partial^2 z}{\partial x^2} + b \frac{\partial^2 z}{\partial x \partial y} + c \frac{\partial^2 z}{\partial y^2} + d \frac{\partial z}{\partial x} + e \frac{\partial z}{\partial y} + f, \quad (1)$$

to the picked horizon using nine or more grid points, where in general the use of more points provides a smoother curvature estimate. Rich (2008) shows that the curvature calculation is computed from the eigenvalues and eigenvectors of a Hessian matrix (Rich, 2008), resulting in eigenvalues k_1 and k_2 given by:

$$k_{1,2} = \frac{cde - a(1 + e^2) - b(1 + d^2) \pm (\alpha - \beta)^{1/2}}{(1 + d^2 + e^2)^{3/2}}, \quad (2)$$

Where

$$\alpha = [a(1 + e^2) - b(1 + d^2)]^2, \text{ and} \quad (3a)$$

$$\beta = [2bde - c(1 + e^2)][2ade - c(1 + d^2)]. \quad (3b)$$

The corresponding eigenvectors (defining the azimuth perpendicular to a curvature lineament) are:

$$\tan \theta_{1,2} = \frac{k_{1,2} - cde + 2b(1 + d^2)}{2ade - c(1 + d^2)}. \quad (4)$$

k_1 and k_2 are the most-positive and most-negative principal curvatures. As the names imply, there are no apparent curvature that can be more positive or more negative. These other apparent (or Euler) curvature $k_E(\psi)$ at a given azimuth ψ are defined as:

$$k_E(\psi) = k_1 \cos^2(\psi - \theta_1) + k_2 \sin^2(\psi - \theta_2). \quad (5)$$

Conflicting definitions of k_{\min} and k_{\max}

When volumetric curvature was first introduced, there was (and still remains) some confusion on the definition of maximum and minimum curvature. The AASPI software will apply the convention used by the majority of geophysical workers (e.g. Sigismundi and Soldo, 2003) and almost all of the differential geometry community for k_{\max} and k_{\min} :

$$k_{\max} = \begin{cases} k_1 & \text{where } k_1 \geq k_2 \\ k_2 & \text{where } k_1 < k_2 \end{cases}$$
$$k_{\min} = \begin{cases} k_2 & \text{where } k_1 \geq k_2 \\ k_1 & \text{where } k_1 < k_2 \end{cases}$$

where k_1 and k_2 are the most-positive principal curvature and most-negative principal curvature. An example using a conflicting definition can be found in Roberts (2001). We believe this to be a typographic error since his maximum curvature figure shows values both larger and smaller in signed amplitude than his minimum curvature figure, which is consistent with the definition above. To avoid this confusion, al-Dossary and Marfurt (2006) and almost all our team's subsequent publications avoid using k_{\max} and k_{\min} and instead use the more clearly defined most-positive and most-negative principal curvatures k_1 and k_2 .

Let's plot *k2_GSB_small_long_w.H* (we will call this the most-negative *principal* curvature) using the *AASPI QC Plotting tab* and capture the following image delineating the hanging wall side of the normal faults (see next page):

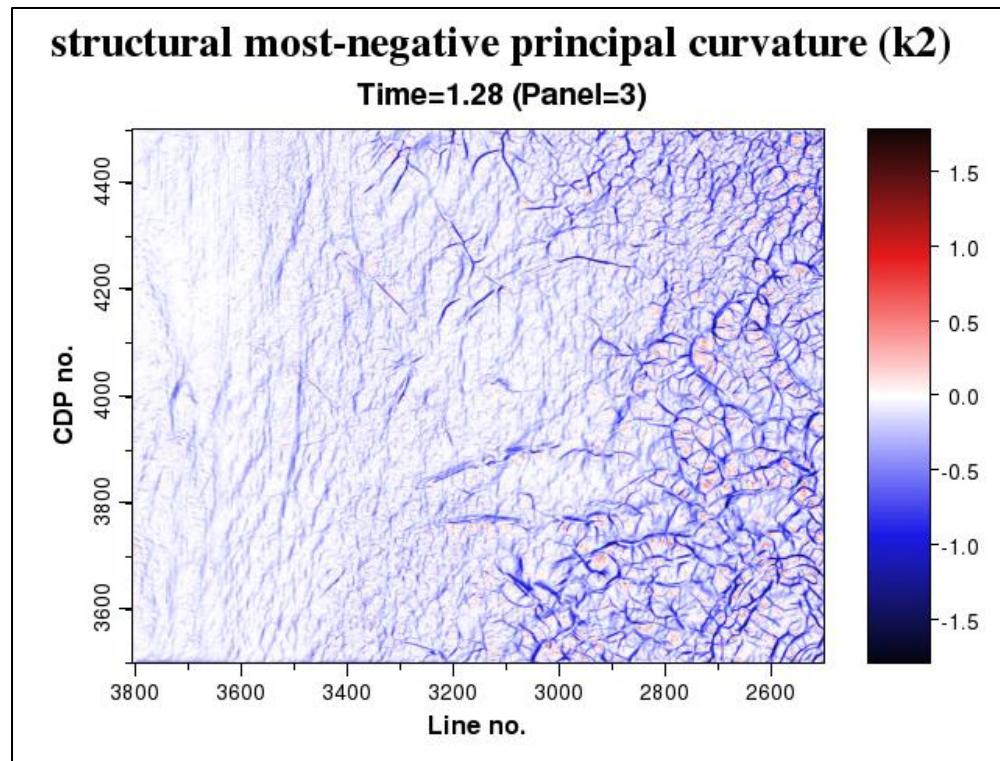


Figure 1.

The corresponding most positive curvature delineates the footwall side of the faults:

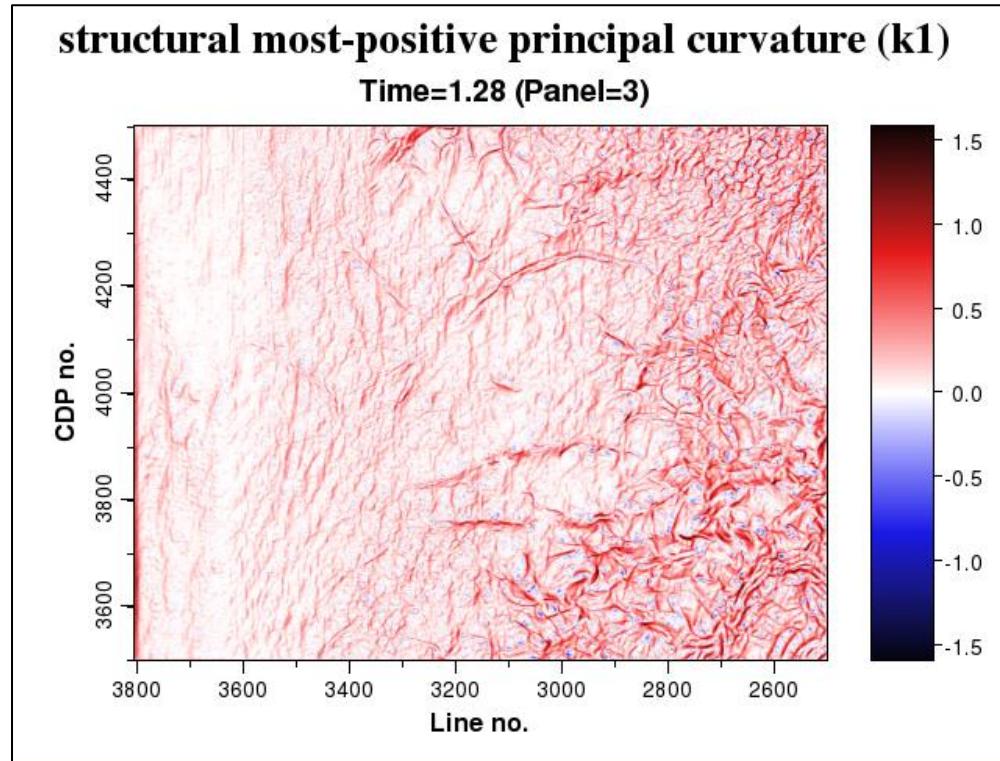
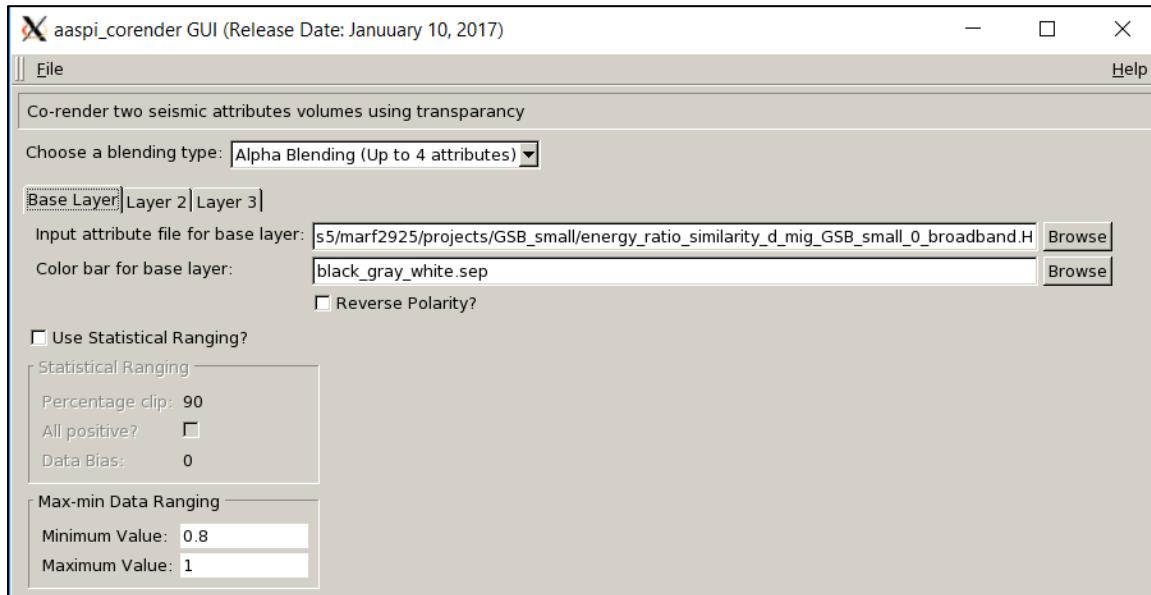


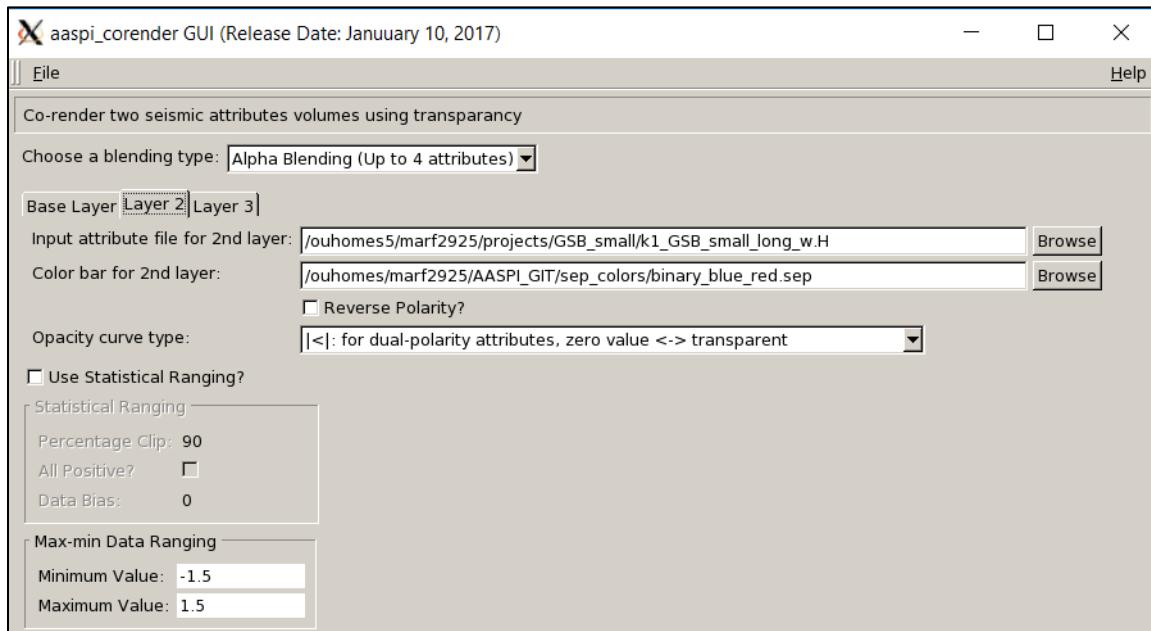
Figure 2.

Geometric Attributes: Program **curvature3d**

Let us co-render both k_1 and k_2 with energy-ratio coherence using program **corender**, using the energy ratio coherence for the base layer,

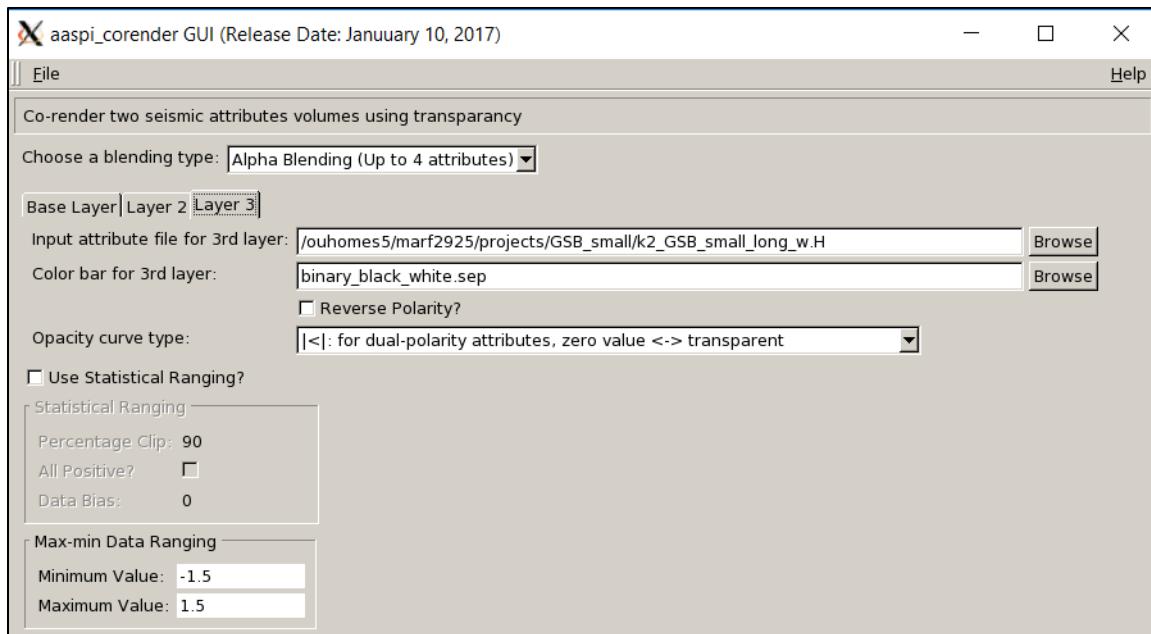


most-positive curvature for the 2nd layer,



and most-negative curvature for the 3rd layer.

Geometric Attributes: Program **curvature3d**



This process gives the following image where we note that the two curvature anomalies “bracket” the normal fault coherence anomaly between them:

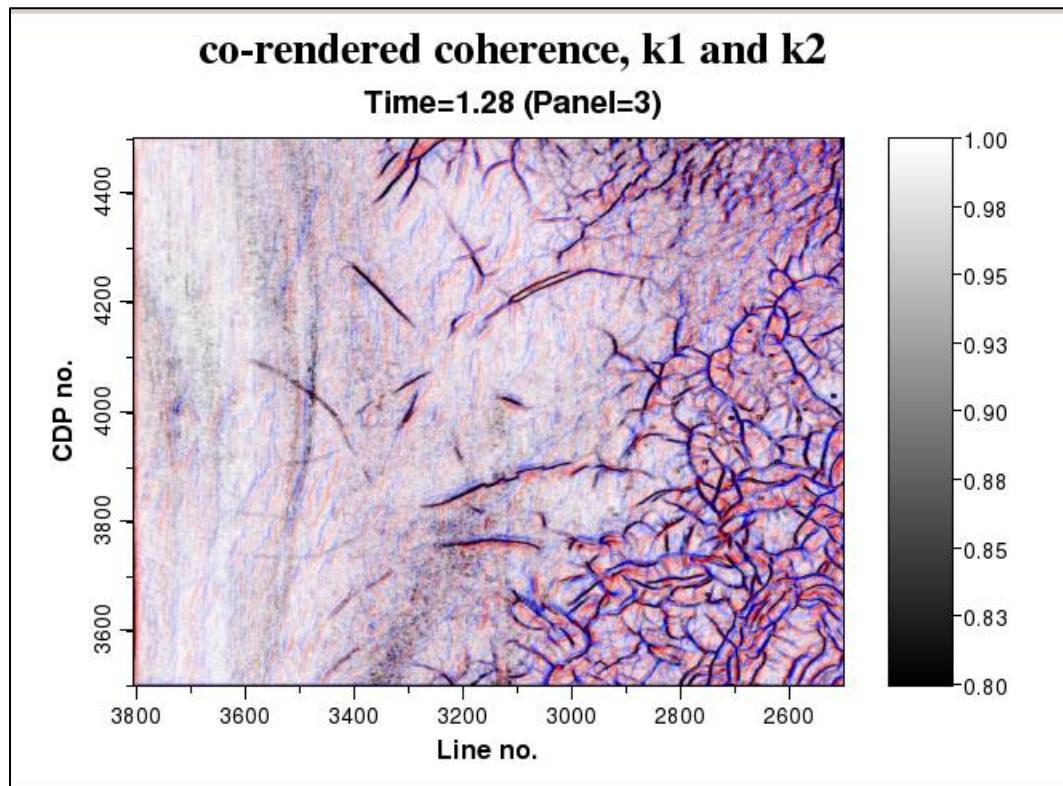


Figure 3.

Geometric Attributes: Program **curvature3d**

Long wavelength vs short wavelength curvature

Examining the theory for both horizon-based and volumetric curvature above, you will note that the curvature estimation requires computing spatial derivatives of the inline and crossline components (p and q) of the vector dip field. Computing the derivative acts as a filter that increases the contribution of the low wavenumber (long wavelength) and decrease the contribution of the high wavenumber (short wavelength) components of the data, thereby exacerbating any short wavelength noise in the data. Common short wavelength noise (or jitters) include acquisition footprint, suboptimum statics, and migration operator aliasing. For horizon-based curvature, one can apply a low-pass smoothing filter to the time-structure map itself or to the inline and crossline components of dip prior to computing coherence. Such application is more difficult volumetrically because there are no explicitly picked horizons to smooth along. If we call this smoothing filter F_{smooth} , by the associative law of algebra we see that

$$\frac{\partial}{\partial x} p_{\text{smooth}} = \frac{\partial}{\partial x} (F_{\text{smooth}} * p) = \left(\frac{\partial}{\partial x} F_{\text{smooth}} \right) * p$$

Such that we can compute the derivative of the smoothing operator and apply the results to the unsmoothed dip and obtain the same results. Such an operation is not only easier to implement but also computationally more efficient since we can precompute the smoothed the derivative operator once for the entire survey rather than smooth each voxel in the data volume.

The curvature3d GUI defines a filter that allows the user to precisely define the low-pass filter shape and thus control output curvature wavelengths, with more aggressive hi-cut filters obviously leading to smoother, or longer-wavelength, curvature results.

Example 1. Long-wavelength and short-wavelength curvature for the Boonsville survey

Error! Reference source not found. (top) shows the filter GUI in its default “long-wavelength” state, while **Error! Reference source not found.** (bottom) displays the amplitude spectrum of the corresponding derivative operator after multiplication by “ $1/k$ ”. Note that this $1/k$ -normalization serves to isolate the action of the low-pass filter (the full derivative operator being the complex product of the theoretical “ ik ” derivative operator and the zero-phase low-pass filter—it follows that the derivative operator’s amplitude spectrum is the product of k times the magnitude of the low-pass filter, and thus division by $1/k$ gives the spectrum of the low-pass filter). **Error! Reference source not found.** shows that the shape of the low-pass filter reduces linearly from a maximum near DC down to zero at spatial Nyquist. Despite the fact that this filter’s linear-ramp shape winds up attenuating much of the higher-wavenumber energy—especially compared to a more common high-cut filter such as Butterworth or Ormsby (both of which have a flat spectrum up to some corner point)—experience shows that it tends to produce very useful curvature results. From the above discussion it should be obvious that

Geometric Attributes: Program **curvature3d**

these results tend to emphasize the low spatial wavenumbers, ergo the “long-wavelength curvature” description.

While the k-axis annotation in **Error! Reference source not found.** (bottom) might suggest that curvature wavelengths as long as the survey dimension can be reliably estimated, it is worth noting that in practice the longest computed wavelength will be significantly smaller. Precisely, this longest computed wavelength will be controlled either by (a) the maximum operator radius, R_{\max} (i.e., 7th parameter from top in Primary Parameters tab shown in **Error! Reference source not found.**), in which case λ_{\max} will be on the order of R_{\max} , or (b) by the operator truncation parameter (i.e., 9th parameter from top in Primary Parameters tab), in which case λ_{\max} will be on the order of the truncated operator width (this width is not explicitly reported by the algorithm). Fortunately, this potentially confusing situation isn’t particularly relevant in day-to-day usage, as the very long wavelengths on the order of R_{\max} or larger are not ones of primary interest to the interpreter.

In order to generate a curvature result which is richer in higher wavenumbers, the user can flip the GUI to short-wavelength mode (i.e., toggle green box, **Error! Reference source not found.**). Though not shown here, this action alters the filter coefficients at the kneepoints shown in **Error! Reference source not found.** (top) from {1, 0.66, 0.33, 0} to {1, 1, 0.5, 0}. Experience shows that the short wavelength curvature tends to have more background noise, and as such it is not used as much as the long wavelength curvature. Beyond these two notional choices of output long/short wavelength curvature, the user is obviously free to experiment with various shapes of low-cut filter in order to accentuate whatever wavelength of geological feature he/she chooses.

To obtain a shorter-wavelength version of the most negative principal curvature (1) set the *Suffix* to read *short_wavelength_filt*, (2) keep the values of *lambda1* through *lambda4* the same for now, and (3) modify *weight 1* through *weight 4* to be 1, 1, 0.5, and 0 as shown in the image on the next page:

Geometric Attributes: Program **curvature3d**

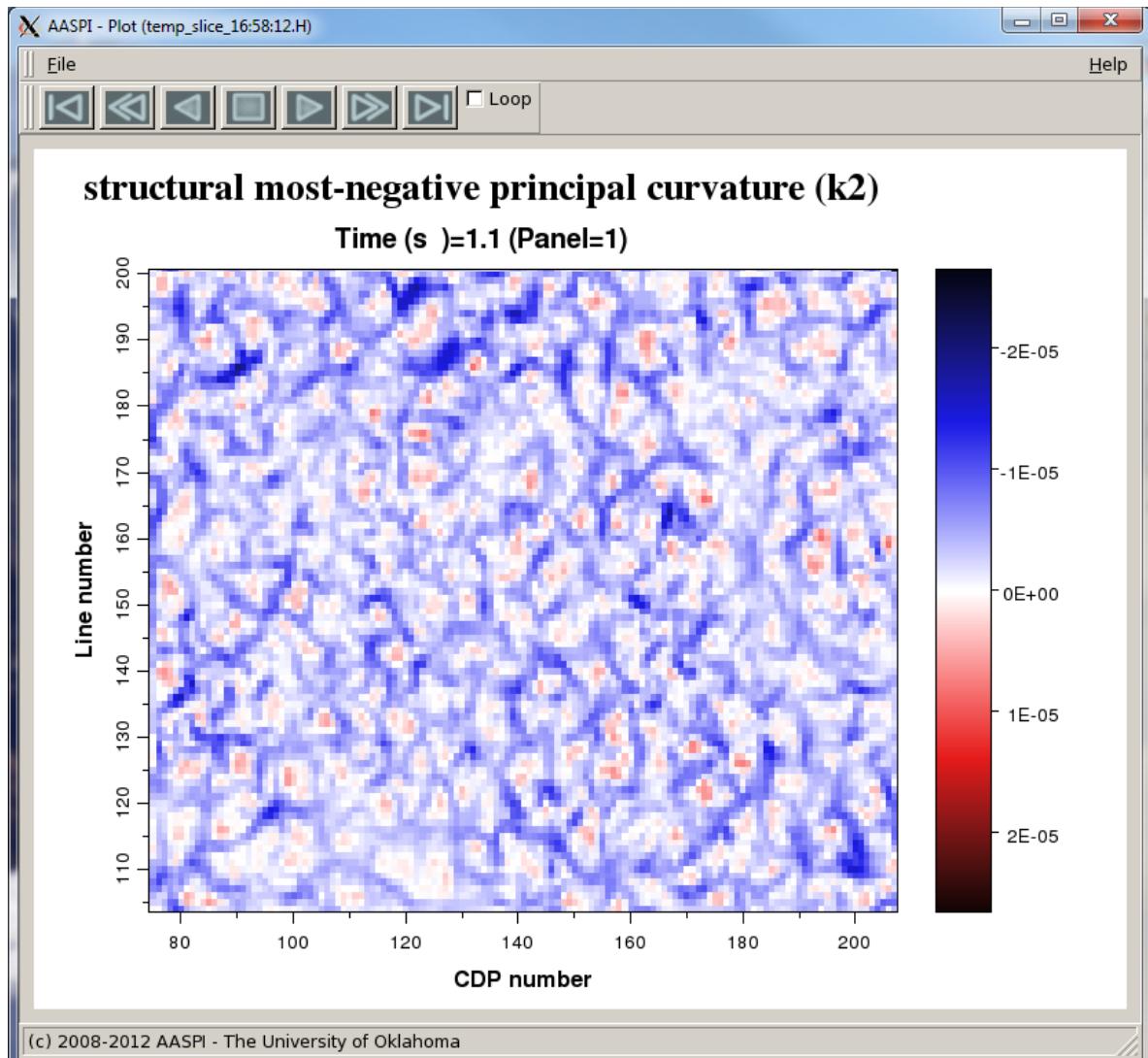


Figure 6.

The long-wavelength most-negative curvature computed from the filtered dip volumes for the Boonsville survey at $t=1.1$ s is almost indistinguishable from the previously displayed image of k_2 , the most-negative *principal* curvature:

Geometric Attributes: Program **curvature3d**

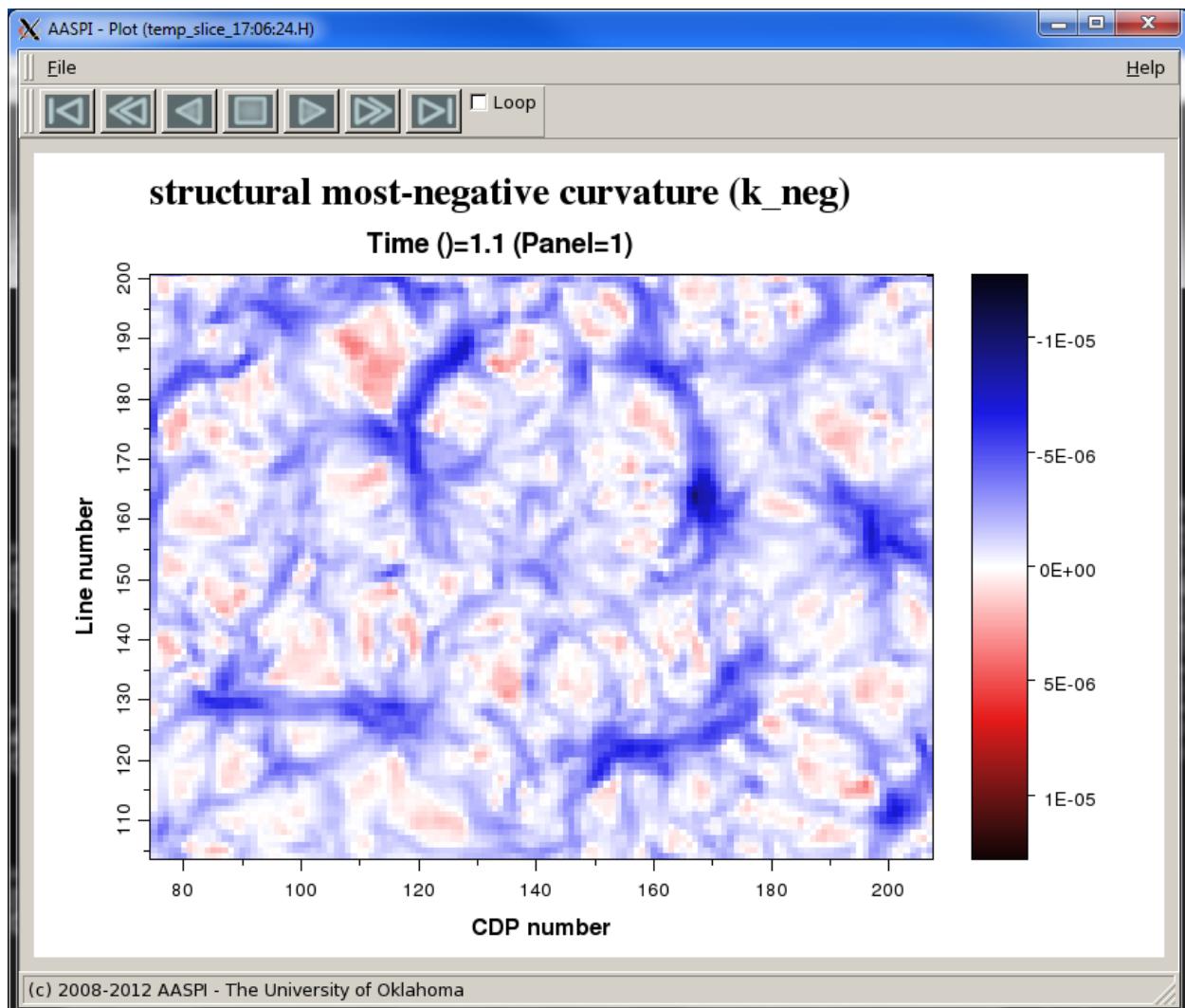


Figure 7.

Indeed, k_{neg} and k_2 will produce nearly identical results if the overall dip is relatively flat. In the Boonsville survey, the average dip is less than 5° . For more tectonically deformed terrains, the images can be significantly different, as described by Mai et al. (2009).

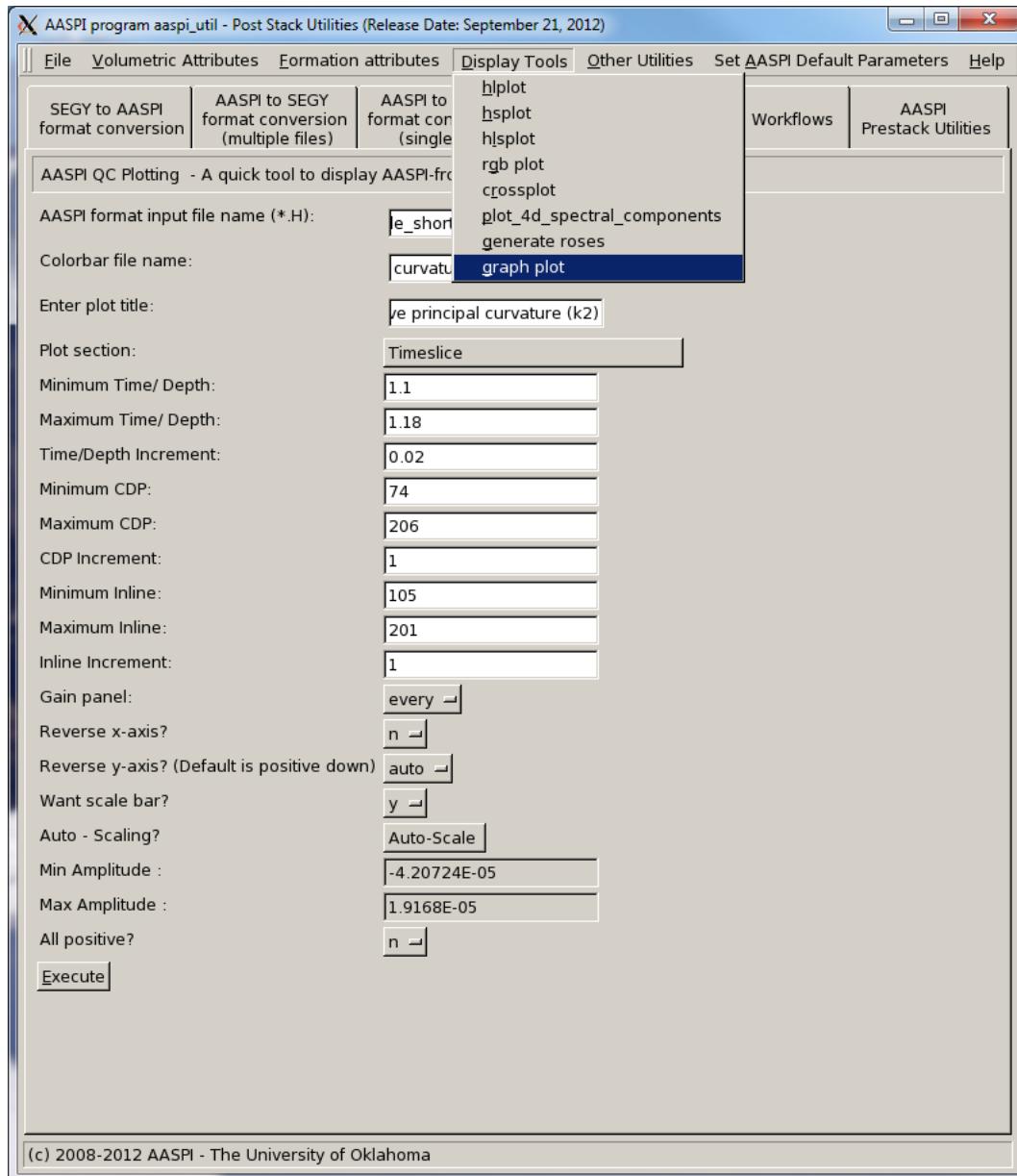
Defining and displaying operators and operator spectra used in **curvature3d**

If you are an interpreter, you will want to understand the degree of mixing that takes place with what we have called ‘long-’ and ‘short-’ wavelength operators above. If you wish to be more precise, you may wish to design different operators to enhance features of interest in your curvature computation. With the 2010 release, we have made this ‘precise’ definition the

Geometric Attributes: Program **curvature3d**

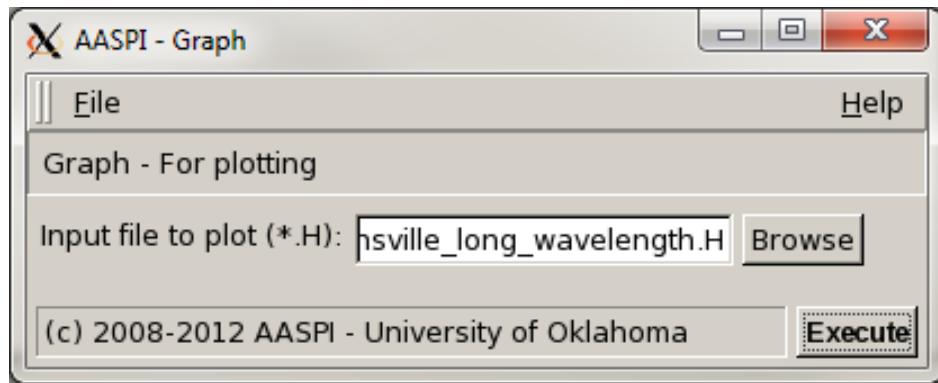
default. However, the ‘fractional derivative’ definition of the operator is still available with which we will start our discussion.

To facilitate graphing these operators, we run a very simple GUI to display them, found under the Display tools tab. Let us go under *Display Tools* in **aaspi_util** and invoke **graph plot**:



The following GUI appears:

Geometric Attributes: Program **curvature3d**



Select *d_dr_operator_boonsville_long_wavelength_filt.H* and obtain the following image:

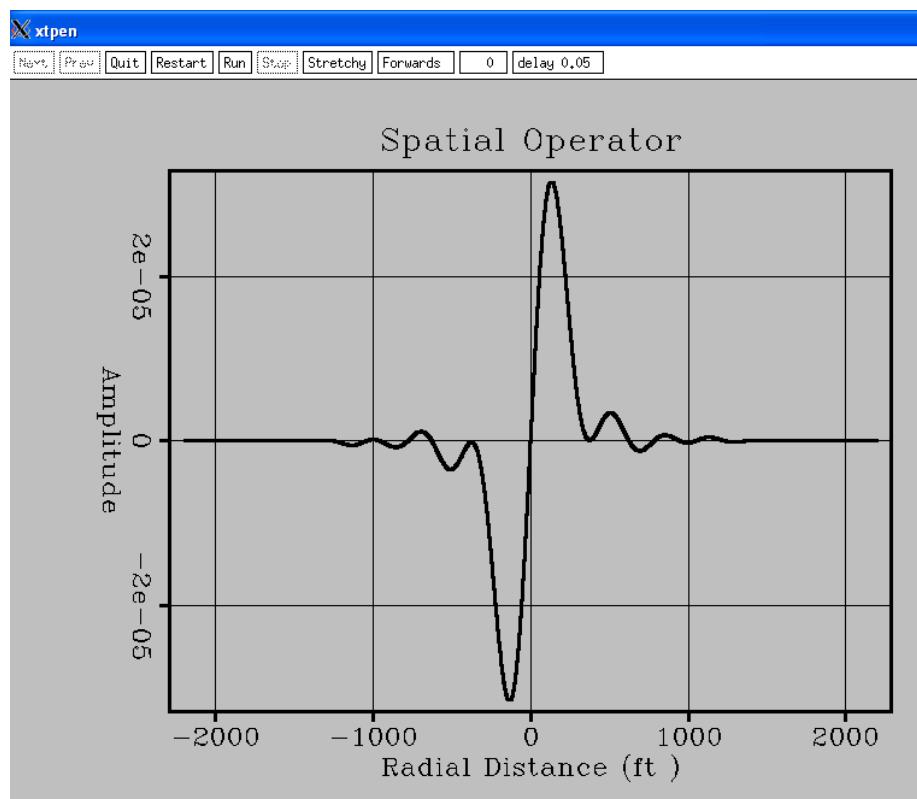


Figure 8.

The units of horizontal axis are that of *unit2='ft'*. Repeat the process for the short wavelength operator and obtain:

Geometric Attributes: Program **curvature3d**

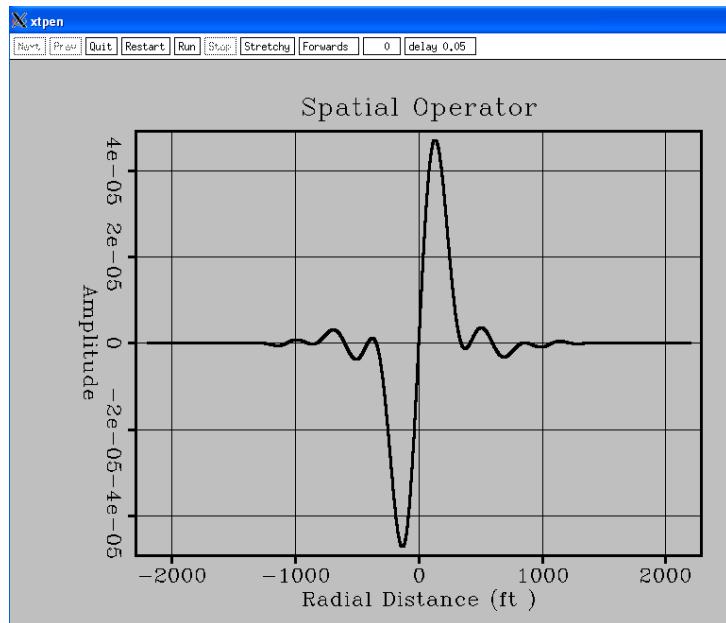


Figure 9.

The short wavelength operator was computed using values of *lambda1* through *lambda4* of 1.0, 1.0, 0.5, and 0.0. Let us rerun **curvature3d** without 'any' filter using values of *lambda1* through *lambda4* of 1.0, 1.0, 1.0, and 1.0. Recall that this would run the filter right up against the Nyquist limit of 331 ft at 45 degrees to the acquisition grid. This operator looks like the following:

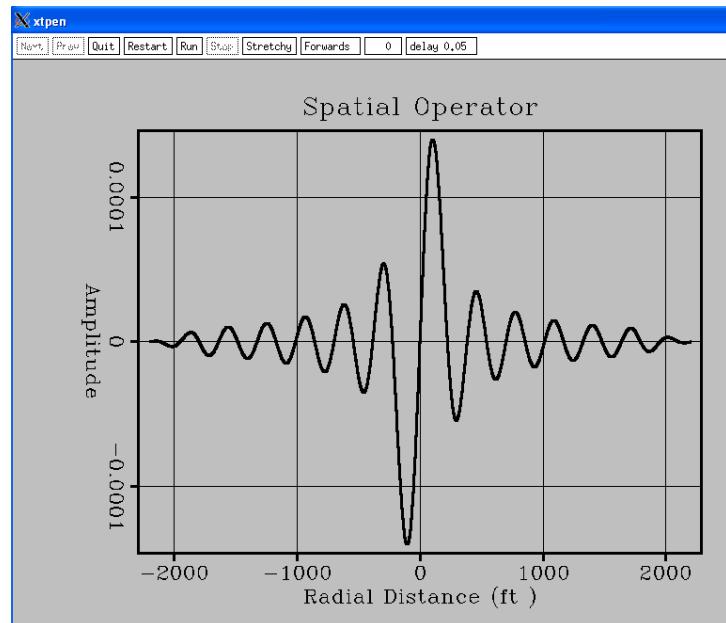


Figure 10.

Part of the oscillation is associated with the software attempting to avoid going beyond Nyquist by applying a 'boxcar' filter in the wavenumber domain that transforms to a sinc function in the

Geometric Attributes: Program **curvature3d**

space domain. Even so, recall that the 10th order finite difference operator (five points on either side of the analysis point) oscillates like this:

$$\frac{-2}{2520} \quad \frac{25}{2520} \quad \frac{-150}{2520} \quad \frac{600}{2520} \quad \frac{-2100}{2520} \quad 0 \quad \frac{2100}{2520} \quad \frac{-600}{2520} \quad \frac{150}{2520} \quad \frac{-25}{2520} \quad \frac{2}{2520} .$$

Plotting the spectrum of these three operators (*d_dr_spectrum_boonsville_long_wavelength.H*, *d_dr_spectrum_boonsville_short_wavelength.H*, and *d_dr_spectrum_boonsville_all_wavelength.H*) gives the following three images:

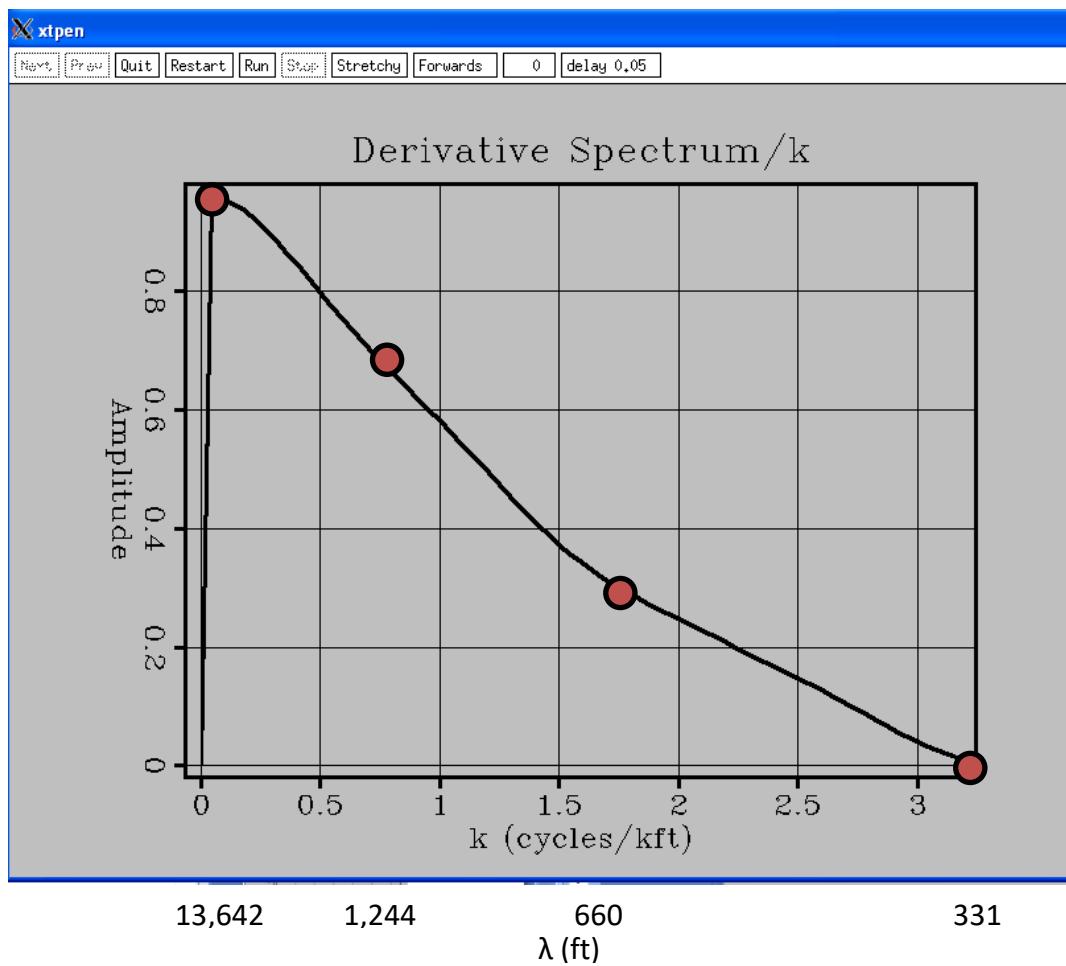


Figure 11.

Geometric Attributes: Program **curvature3d**

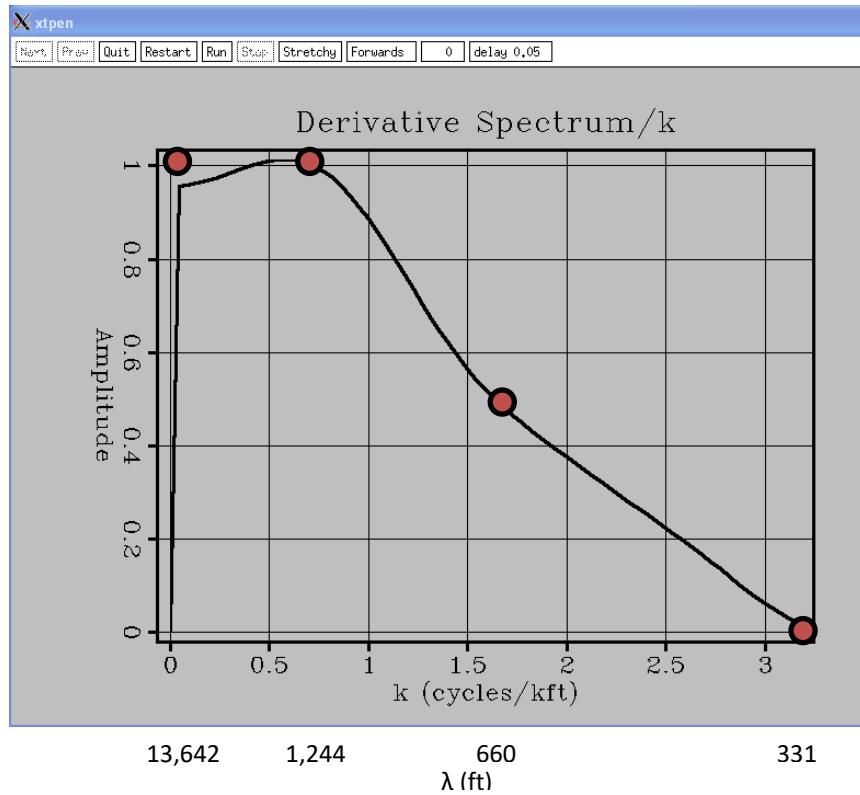


Figure 12.

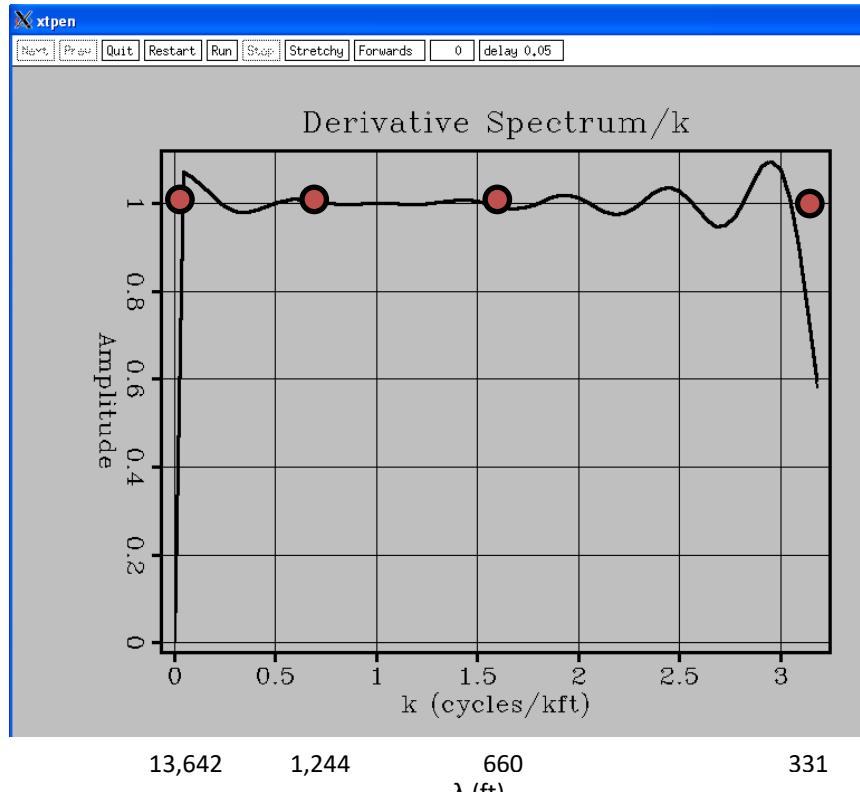
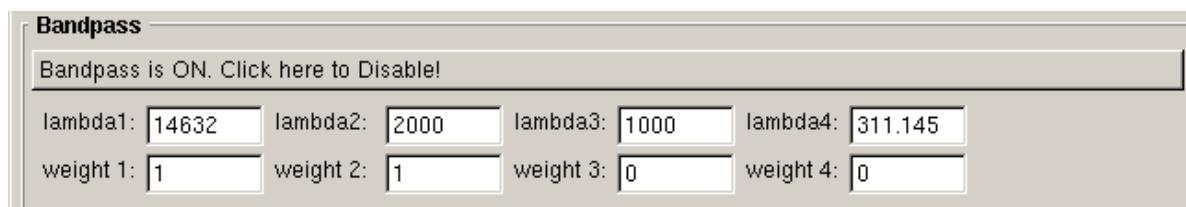


Figure 13.

Geometric Attributes: Program **curvature3d**

In all three of these images I have added the corresponding wavelength λ along the horizontal axis and the weights entered at *lambda1* through *lambda4* as red circles. Each spectrum has been normalized by $1/k$. Since the first derivative of a continuous media can be perfectly approximated in the frequency domain by the operator ik , its magnitude spectrum (normalized by $1.0/k$) should be identically 1.0. Thus, we can interpret our long wavelength operator as being equivalent to computing the conventional curvature with a simple first derivative operator and following the computation by applying a low-pass filter.

We can further demonstrate this concept by explicitly defining a more traditional low-pass filter:



where now, no wavelengths are allowed smaller than 1000 ft. (the *weight3*=0.0 and *lambda3*=1000).

The spectrum looks like this:

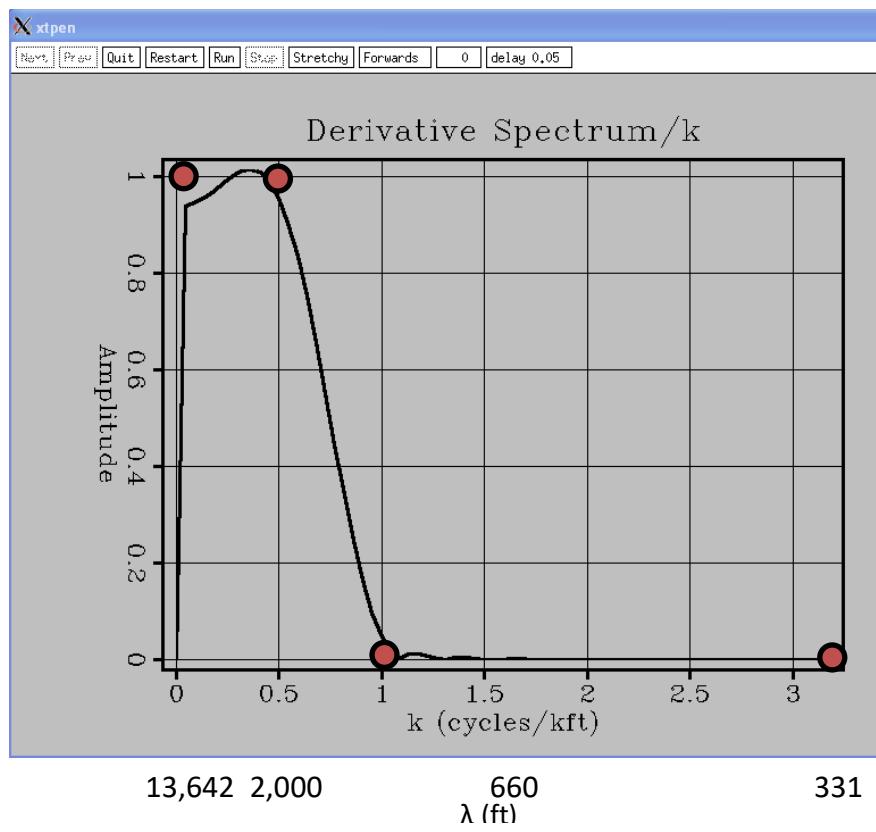


Figure 14.

Geometric Attributes: Program **curvature3d**

while the operator looks like this:

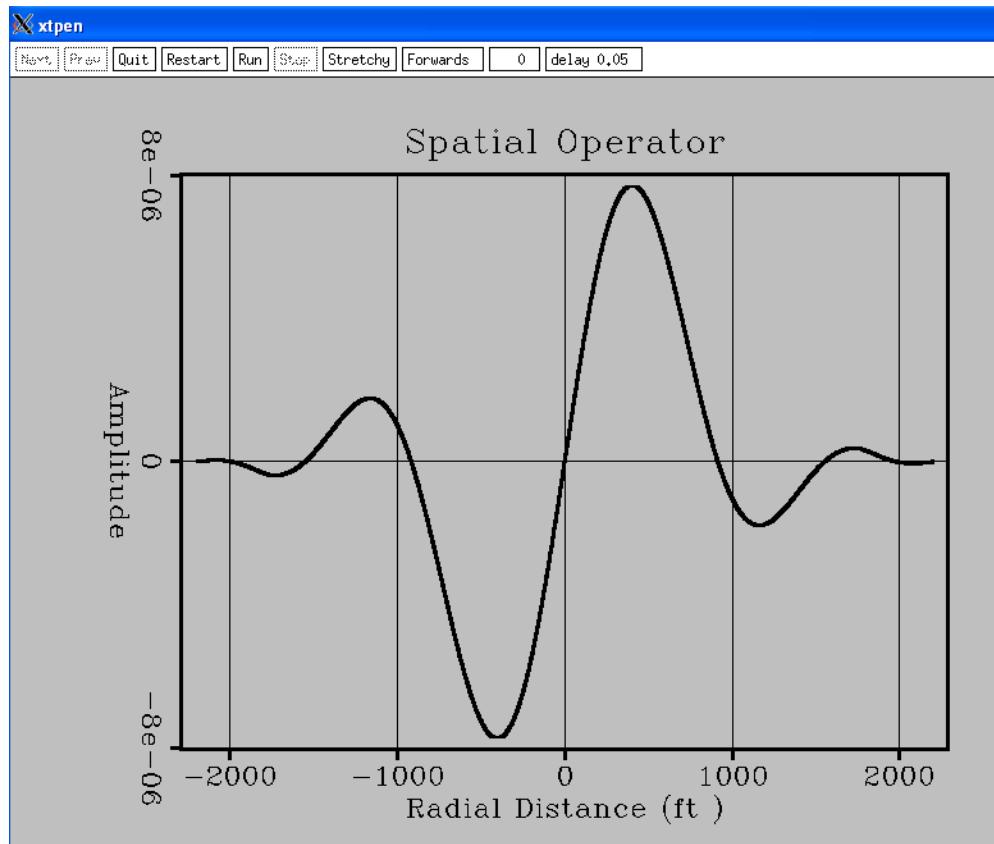


Figure 15.

giving the following image at t=1.1 s:

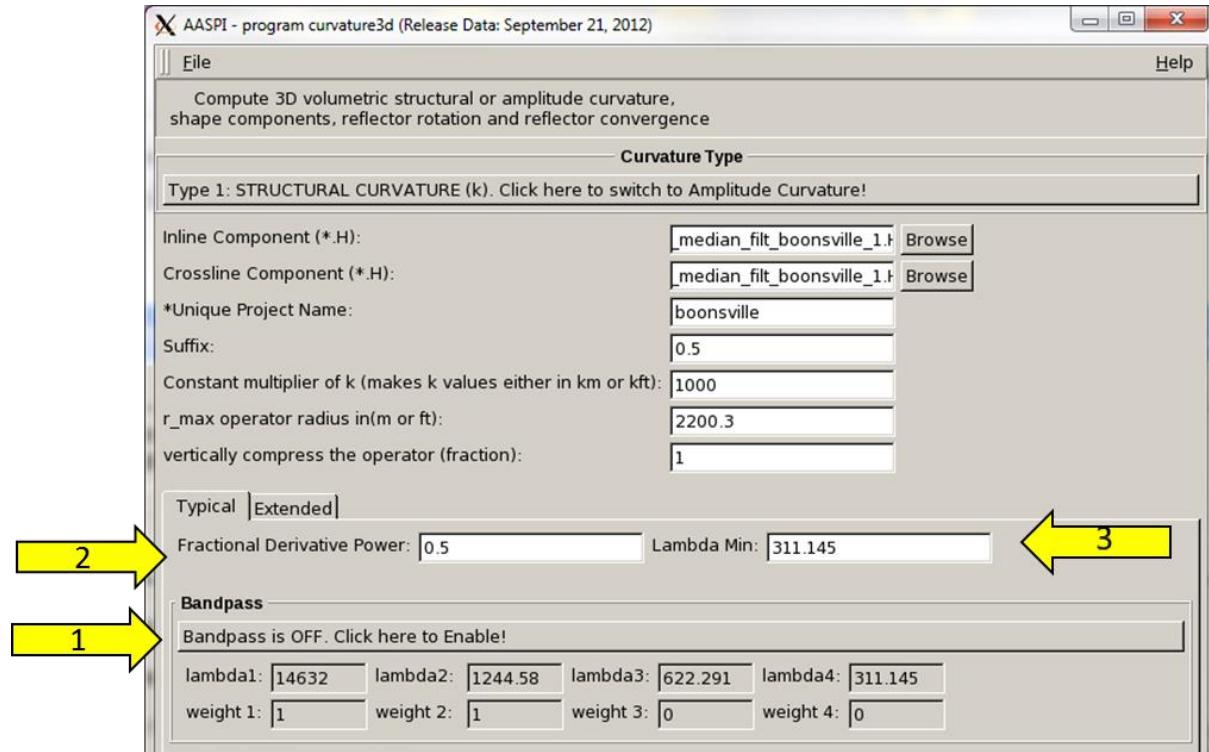
While there may be some value of such low-pass filtered data for statistical analysis, it is clearly inferior to the previous relatively broadband ‘long wavelength’ version of most-negative principal curvature for purposes of structural or stratigraphic interpretation.

Fractional Derivative Operators

The above discussion, while perhaps somewhat tedious, is very straightforward for processing geophysicists. Al-Dossary and Marfurt (2006) are perhaps responsible for introducing the concept of fractional derivatives to the curvature literature. Their work was inspired by high quality images generated by Cooper and Cowans (2003) in their analysis of potential field data. Since curvature is based on the computation of the first derivatives of inline and crossline components of structural dip, Al-Dossary and Marfurt (2006) postulated that one could improve on the calculation by generating fractional derivatives of inline and crossline components of structural dip. While these images were indeed very useful, the concept of fractional derivative added an extra degree of obfuscation to the entire workflow. For this reason, we now set the

Geometric Attributes: Program **curvature3d**

default curvature computations to be computed using the more conventional (and easier to explain) bandpass filter concepts described above.



To invoke the fractional derivative option, simply (1) click the Bandpass tab to turn it off. Note that (2) the *Fractional Derivative Power* space is now activated and set to the default of $\alpha=0.5$. Do not set (3) the value of Lambda Min to be smaller than Nyquist. The resulting spatial operator looks as follows (see next page):

Geometric Attributes: Program **curvature3d**

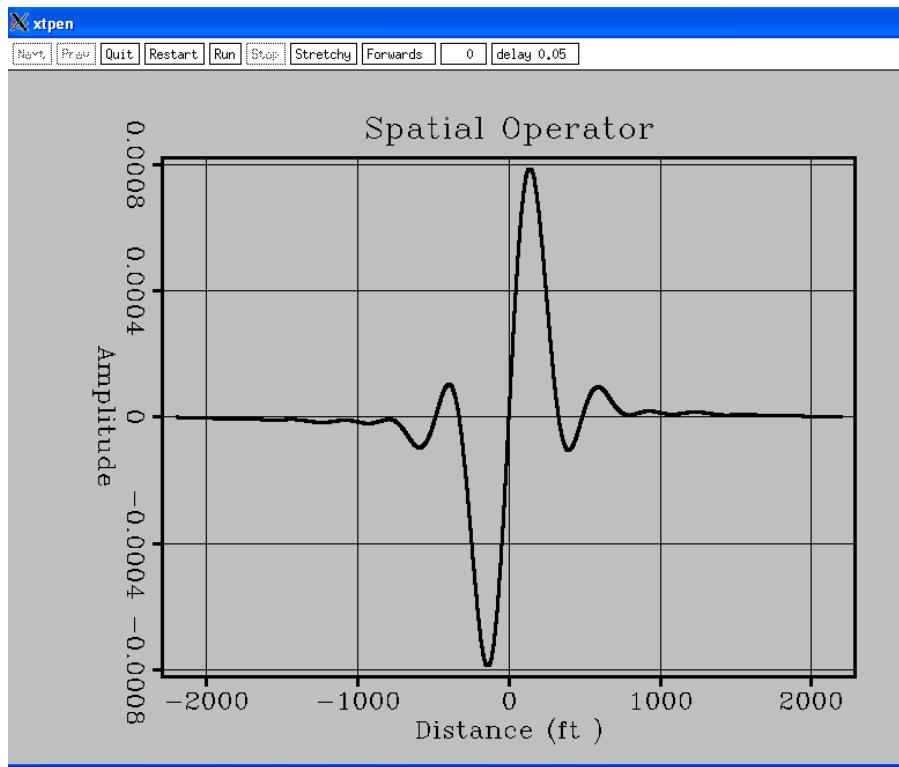


Figure 16.

while the corresponding spectrum appears like this:

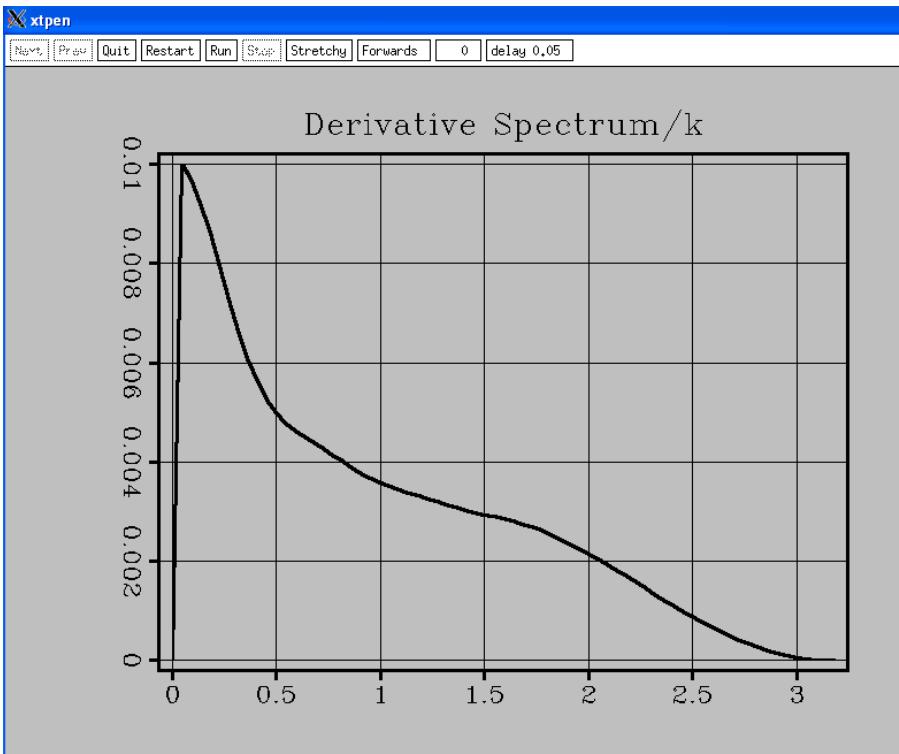


Figure 17.

Geometric Attributes: Program **curvature3d**

Note that with the exception of the vertical scale, the shape of both the spectrum and the operator look very similar to the long wavelength filter discussed earlier.

The time slice through the most-negative principal curvature at t=1.1 s looks like this:

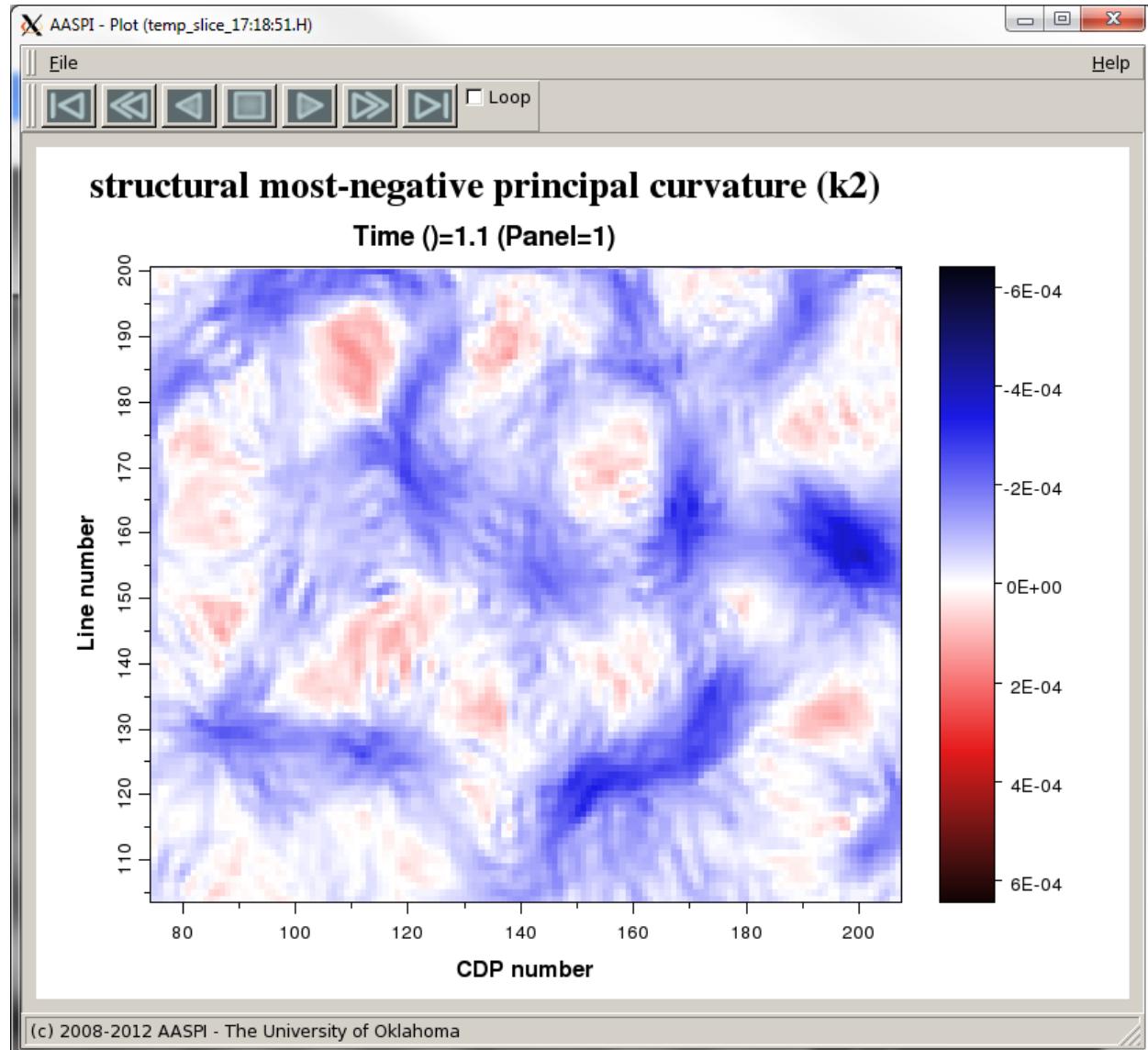


Figure 18.

If we push the fractional derivative concept to its endpoint, with the *Fractional derivative power* = 0.0, we should get an operator that looks like $ik^{0.0}=i$ which is equivalent to a Hilbert transform. Normalizing by $1/k$, we obtain the following spectrum, observed on the following image.

Geometric Attributes: Program **curvature3d**

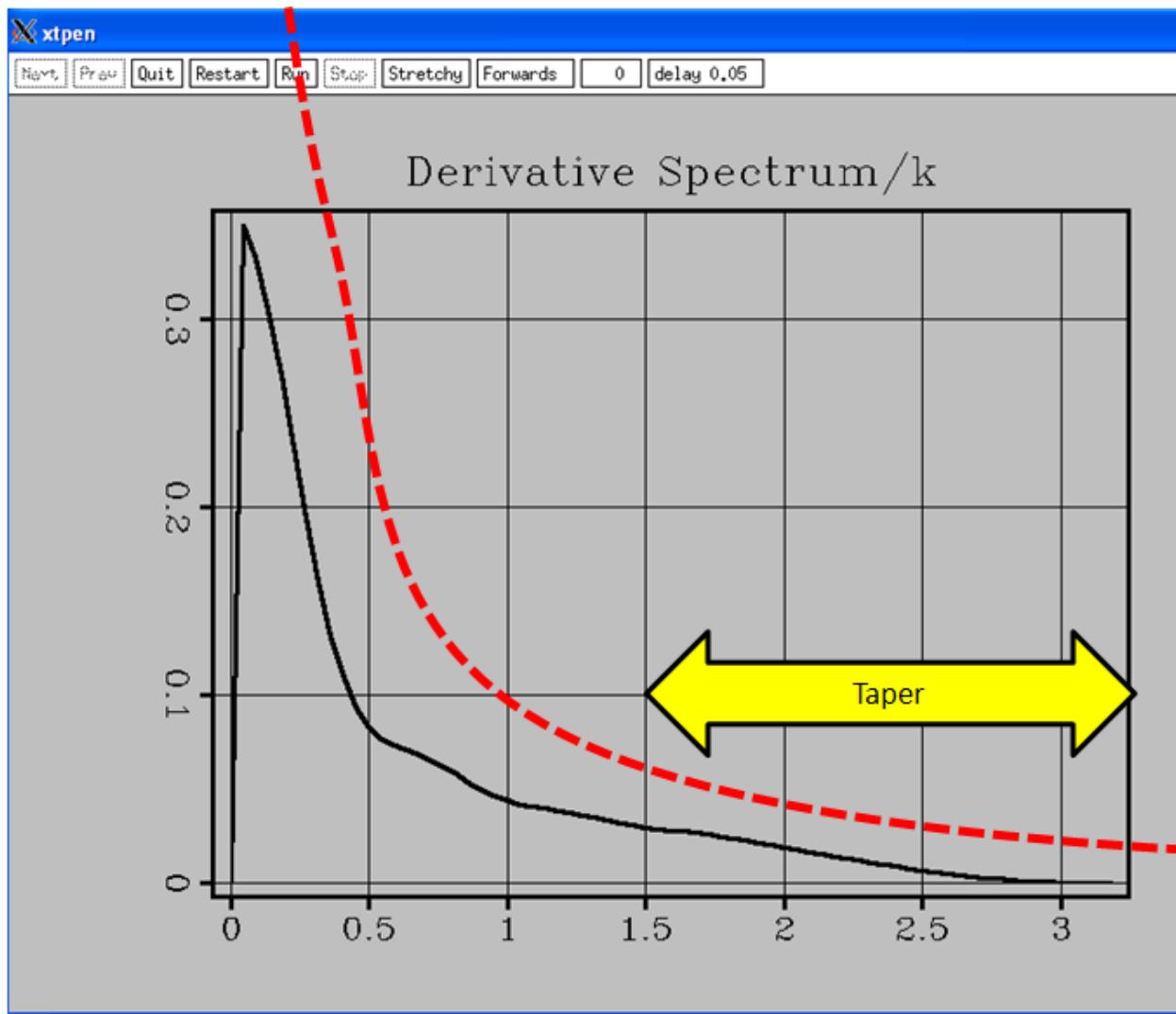


Figure 19.

The AASPI software applies two limitations to any desired spectrum. First, all data are band-limited, with the minimum wavelength for these data, $\lambda_{min} = 331$ ft, such that $\max(k)=3$ cycles/kft. An abrupt truncation of any spectrum produces a Gibbs' phenomenon, such that we apply a default taper that begins at $\lambda_{taper} = 2\lambda_{min} = 662$ ft or at $k=1.5$ cycles/kft. Second, we truncate the spatial operator at a default level of $op_clip=1.0\%$ of the maximum spatial operator value. Since the idealized Hilbert transform operator decays as $1/r$, this would happen approximately at $100*110$ ft = 11,000 ft, which would result in very long (and very slow) spatial operators. To avoid long run times, we limit and spatially taper our operators to be no longer than $\pm 20*\Delta x$ by $\pm 20*\Delta y$ (or in our case ± 2200 ft by ± 2200 ft). For this reason, the operator spectra approximates the dashed red ($1/k$) line but does not exactly equal it.

Geometric Attributes: Program **curvature3d**

The corresponding operator looks like this:

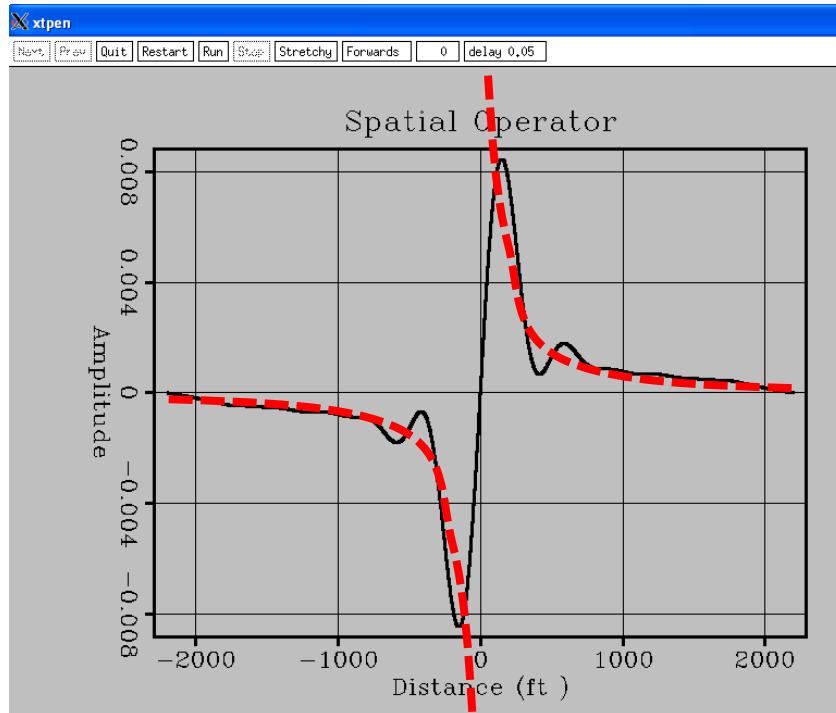


Figure 20.

where the dashed red lines are a (hand-drawn!) approximation to the non-band-limited Hilbert transform operator. The resulting k_2 most-negative principal curvature image looks like,

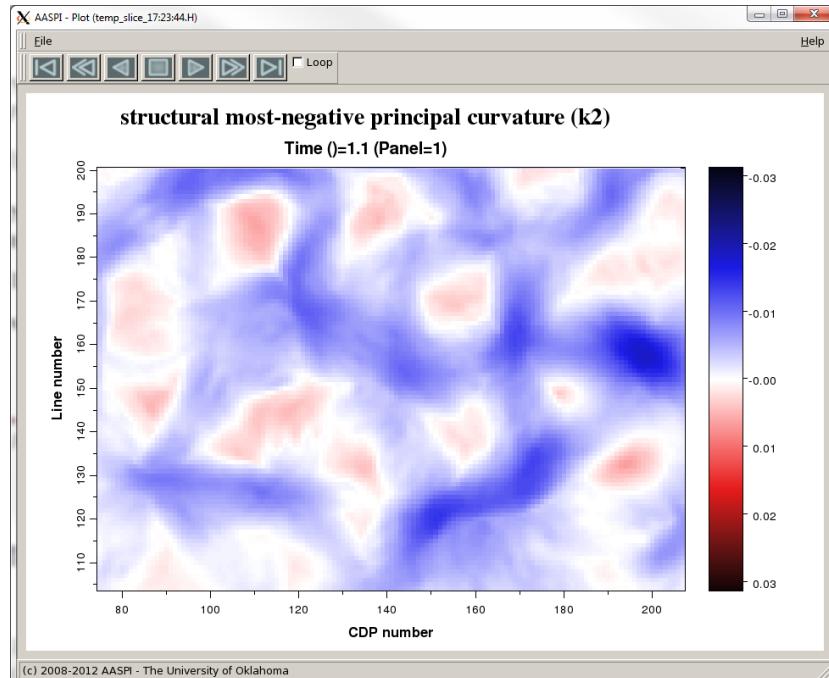


Figure 21.

Geometric Attributes: Program **curvature3d**

which is somewhat a longer wavelength than the image generated using a *Fractional derivative power*=0.50.

Implementation of derivative operators in 3D

Volumetric curvature requires the numerical computation of the derivatives of p and q along the x and y axes. Because the dip values are defined discretely at each voxel in the analysis window rather than as a continuous function, considerable care should be taken in computing their derivatives. Specifically, the numerical operator should be (for equally spaced voxels) rotationally invariant and appropriately use all the data in the analysis window rather than just those along a given axis. First, define the 1D filtered derivative

$$\frac{\partial u}{\partial r} \approx \delta r * u . \quad (14)$$

Then define the three spatial derivatives about the center voxel to be:

$$\begin{aligned}\frac{\partial}{\partial x} &= \sum_{k=1}^K \frac{x_k}{r_k} \delta r_k , \\ \frac{\partial}{\partial y} &= \sum_{k=1}^K \frac{y_k}{r_k} \delta r_k , \text{ and} \\ \frac{\partial}{\partial z} &= \sum_{k=1}^K \frac{z_k}{r_k} \delta r_k\end{aligned} \quad (15)$$

where $r_k = [x_k^2 + y_k^2 + z_k^2]^{1/2}$ is the distance of each of the K voxels in the analysis window to the center voxel.

Implementation of Operators in 3D

To examine the derivative operators, use the *AASPI QC Plotting tab* in **aaspi_util** to display the file *d_dx_operator3d_boonville_long_wavelength.H*, *d_dx_operator3d_boonville_long_wavelength.*, and *d_dx_operator3d_boonville_long_wavelength.H*.

Geometric Attributes: Program **curvature3d**

Time slices at time $t=0$ through the operator produces the following images:

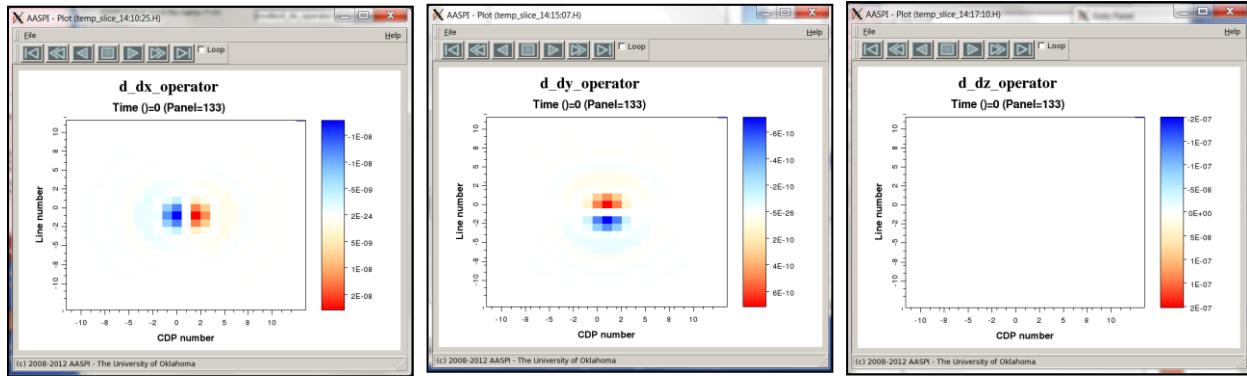


Figure 22.

Note that the slice through d/dz operator is zero along $t=0.0$ s, since it defines a plane of symmetry through an antisymmetric operator. The d/dx operator is rotationally symmetric about the x-axis, the d/dy operator rotationally symmetric about the y-axis, and the d/dz operator rotationally symmetric about the z-axis. A vertical slice along line 0 through the same operators gives:

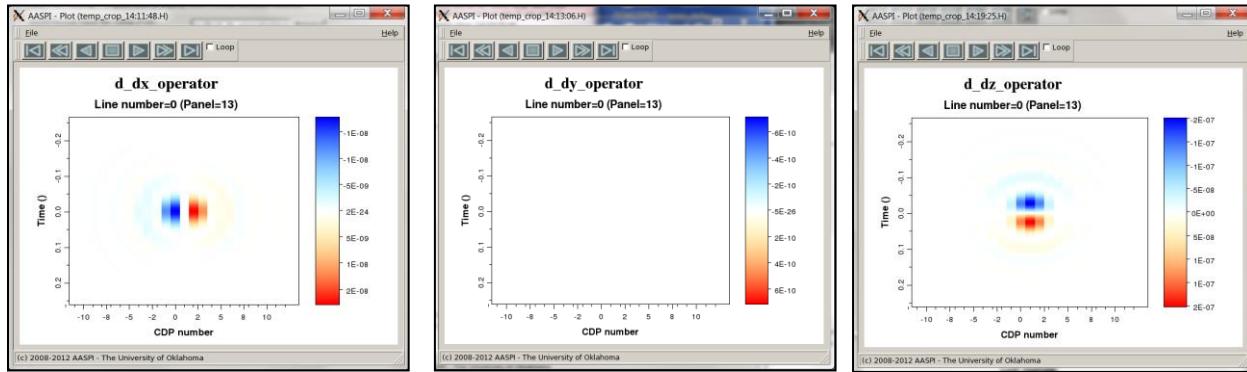


Figure 23.

where, now the d/dy operator is zero along its plane of symmetry, $y=0.0$.

The short-wavelength derivative operator with amplitude corner points of 1.0, 1.0, 1.0, 0.0 shown earlier appears as follows on a vertical slice:

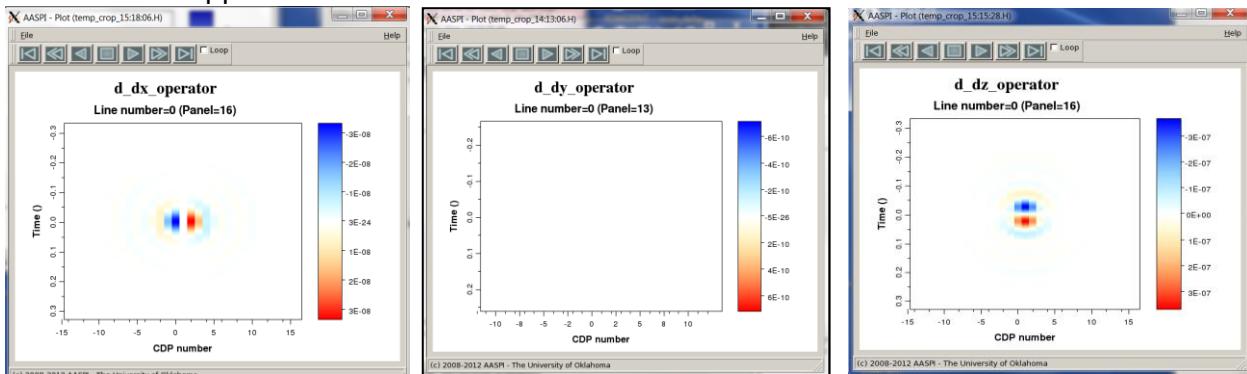


Figure 24.

Geometric Attributes: Program **curvature3d**

Careful inspection will show that this latter operator is a little shorter in all three dimensions.

Note that the extent of the short-wavelength and long-wavelength operators is about the same, but that the long-wavelength operator is more monotonic, while the short wavelength operator is slightly more oscillatory. Also note that the operators are 3D, and are not restricted to the inline, crossline, and vertical axes. This 3D design makes the application quite robust with respect to the axis of acquisition. In early testing, we have rotated the operators by 45° and obtained identical curvature images.

Compressing the vertical operator to reduce vertical mixing

For fairly flat geology with dips less than 15° we can modify the derivative operators to be more compressed in the vertical direction. If we set the value of $vcompress=0.25$ we obtain the following vertical slices through the operators along the $y=0.0$ plane:

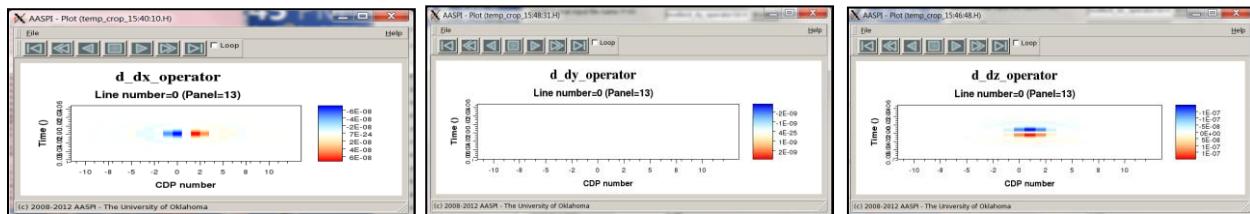


Figure 25.

Geometric Attributes: Program **curvature3d**

Let us examine the impact of changing the vertical size of the operator on the Boonsville data volume, beginning with the isotropic (*vcompress*=1.0) operator:

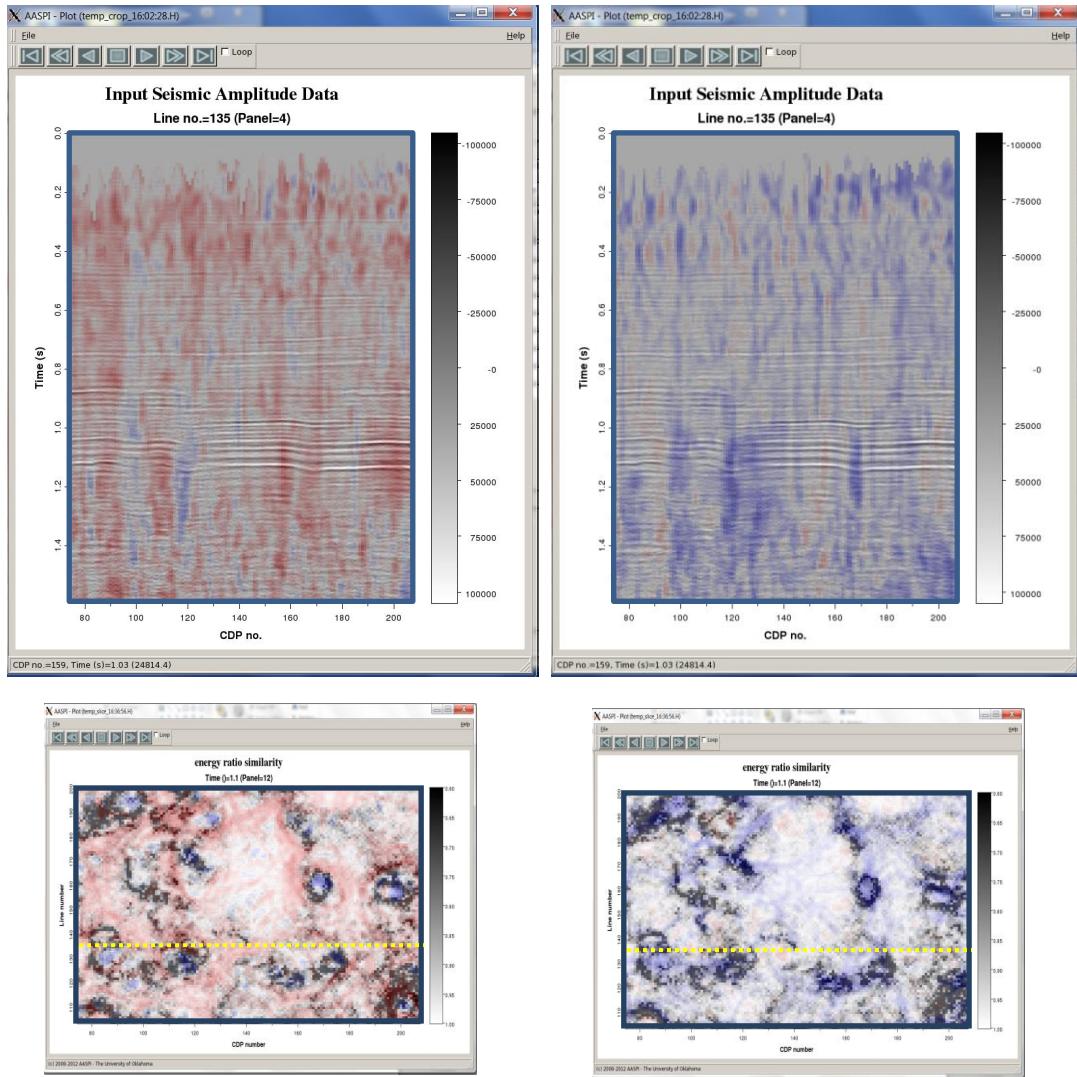


Figure 26.

Geometric Attributes: Program **curvature3d**

Repeating the computation with $vcompress=0.50$ gives:

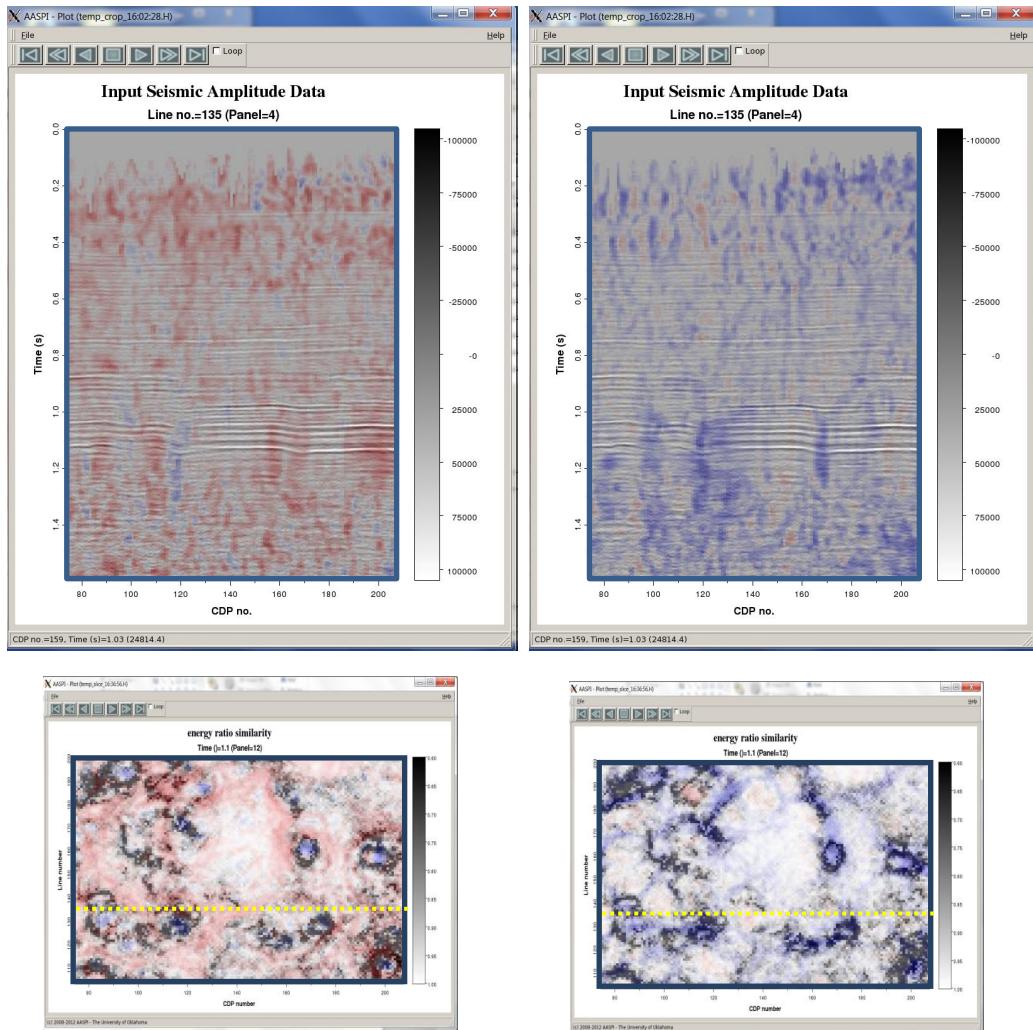


Figure 27.

Geometric Attributes: Program **curvature3d**

Finally, setting a value of `vcompress=0.25` gives:

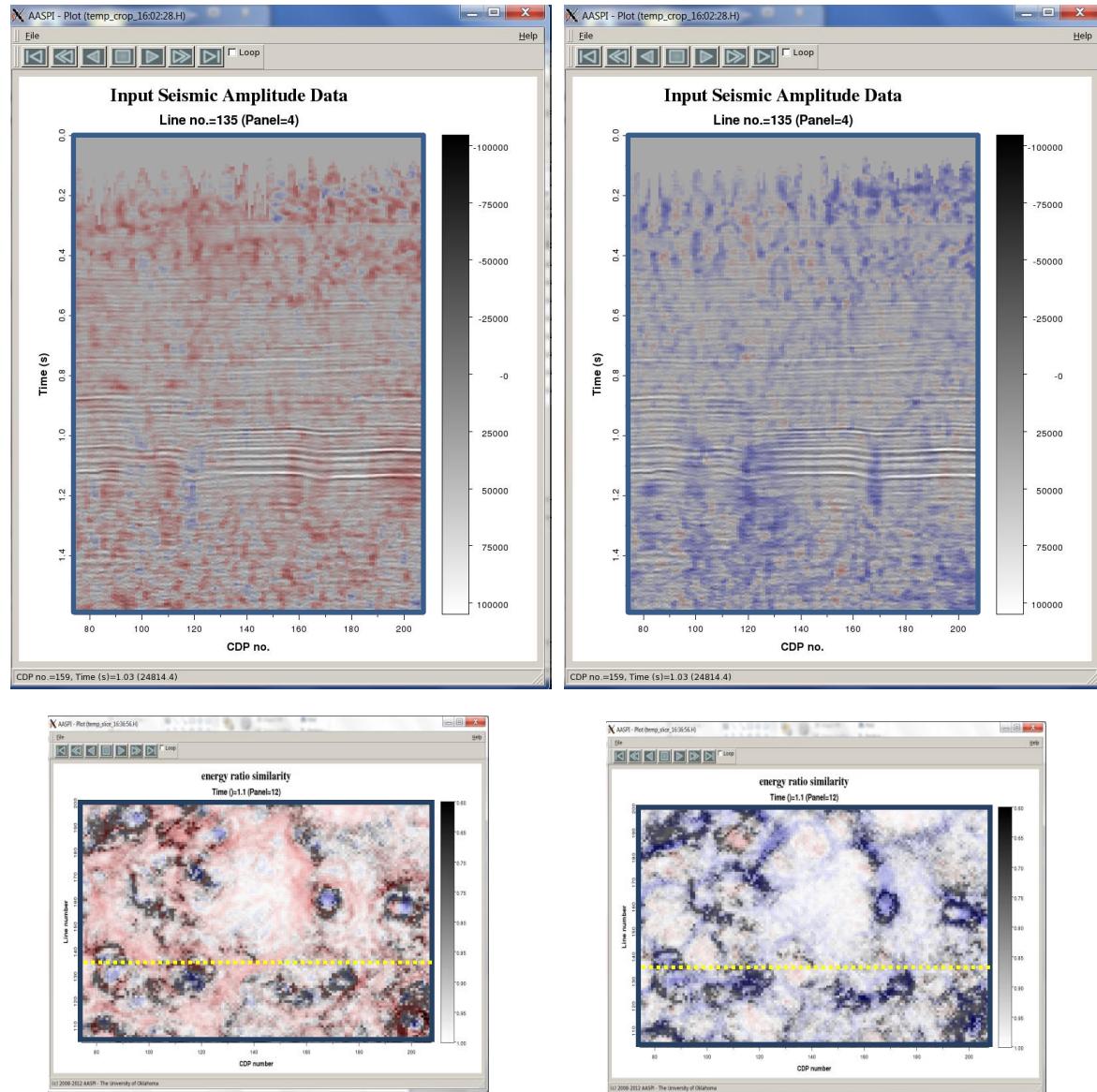


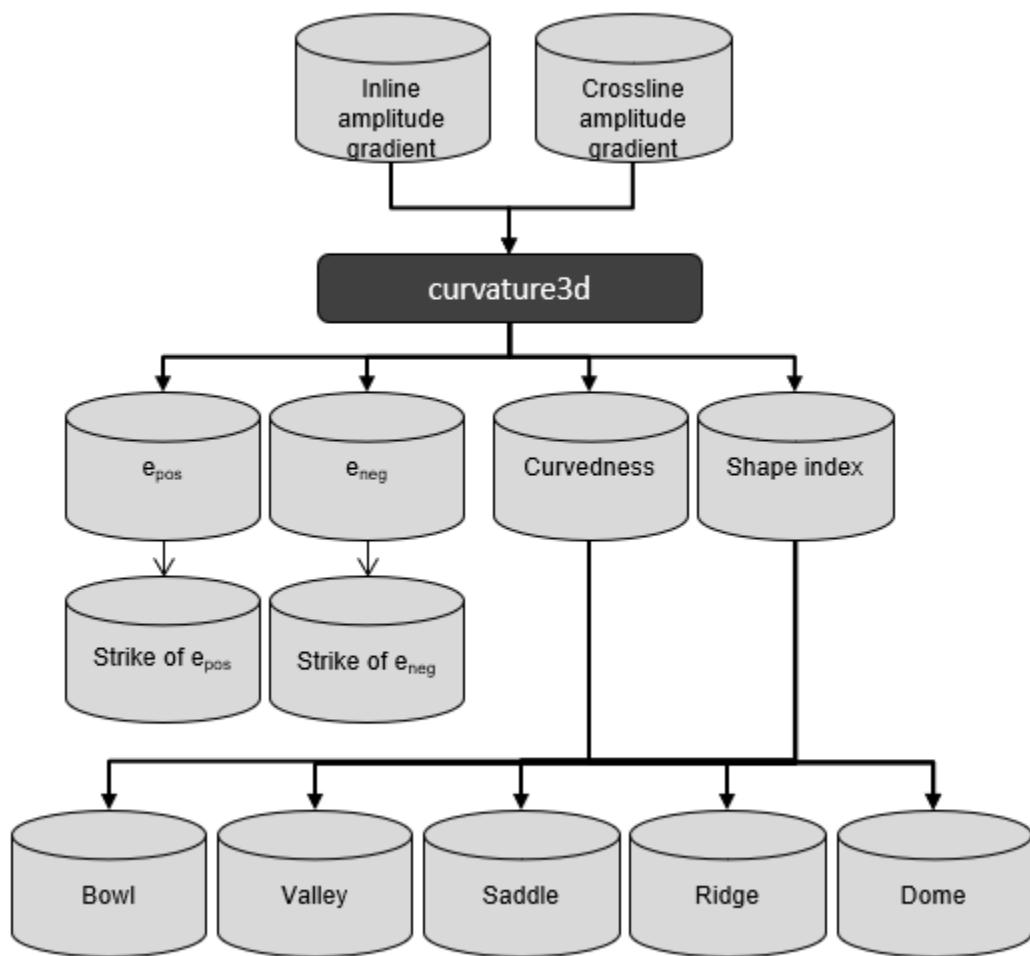
Figure 28.

Comparing these three results we note that compressing the vertical operator limits the leakage of strongly deformed reflectors from overprinting those above and below. Philosophically, one might think of using different wavelengths to compute curvature – with long-wavelength operators in the more gently variable lateral dimensions and short-wavelength operators in the more rapidly variable vertical dimension. Please send us suggestions or insight.

Geometric Attributes: Program **curvature3d**

Computation flow chart for *amplitude* curvature

The input to program **curvature3d** will be the inline and crossline components of the inline and crossline coherent energy gradients computed using program **similarity3d**. The basic curvature outputs include the value and strike (eigenvalue and eigenvector) of the most-positive and most-negative principal curvatures, e_{pos} and e_{neg} . Internal to the program, these principal curvatures are combined to construct the curvedness and shape index, which in turn are used to generate the dome, ridge, saddle, valley, and bowl shape components. Unlike structural curvature which measures the dip in m/m or ft/ft (through the conversion of $\tan\theta$) the amplitude gradients use measures of mV^2/m or mV^2/ft (assuming the amplitude is measured in mV). Due to this difference in units, it makes no sense to fit a circle tangent to the surface in amplitude curvature such that we revert to the simpler concept of most-positive and most-negative curvature that will highlight crests and troughs of amplitude and not include the effects of structural dip. Furthermore, concepts of reflector rotation and reflector convergence lose their meaning for amplitude curvature.



Geometric Attributes: Program **curvature3d**

Output file naming convention for amplitude curvature

By default, program **curvature3d** will generate the following output files:

Output file description	File name syntax
Program log information	curvature3d_ <i>unique_project_name_suffix</i> .log
Program error/completion information	curvature3d_ <i>unique_project_name_suffix</i> .err
Filtered curvature derivative	d_dr_curvature_operator_ <i>unique_project_name_suffix</i> .H
Inline curvature derivative operator	d_dx_curvature_operator3d_ <i>unique_project_name_suffix</i> .H
Crossline curvature derivative operator	d_dy_curvature_operator3d_ <i>unique_project_name_suffix</i> .H
Positive curvature	e_pos_ <i>unique_project_name_suffix</i> .H
Strike of positive curvature	e_pos_strike_ <i>unique_project_name_suffix</i> .H
Negative curvature	e_neg_ <i>unique_project_name_suffix</i> .H
Strike of negative curvature	e_neg_strike_ <i>unique_project_name_suffix</i> .H

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.

If selected, you can also output the following volumes:

Output file description	File name syntax
Curvedness	e_curvedness_ <i>unique_project_name_suffix</i> .H
Shape index	e_shape_index_ <i>unique_project_name_suffix</i> .H
Dome shape component	e_dome_ <i>unique_project_name_suffix</i> .H
Ridge shape component	e_ridge_ <i>unique_project_name_suffix</i> .H
Saddle shape component	e_saddle_ <i>unique_project_name_suffix</i> .H
Valley shape component	e_valley_ <i>unique_project_name_suffix</i> .H
Bowl shape component	e_bowl_ <i>unique_project_name_suffix</i> .H

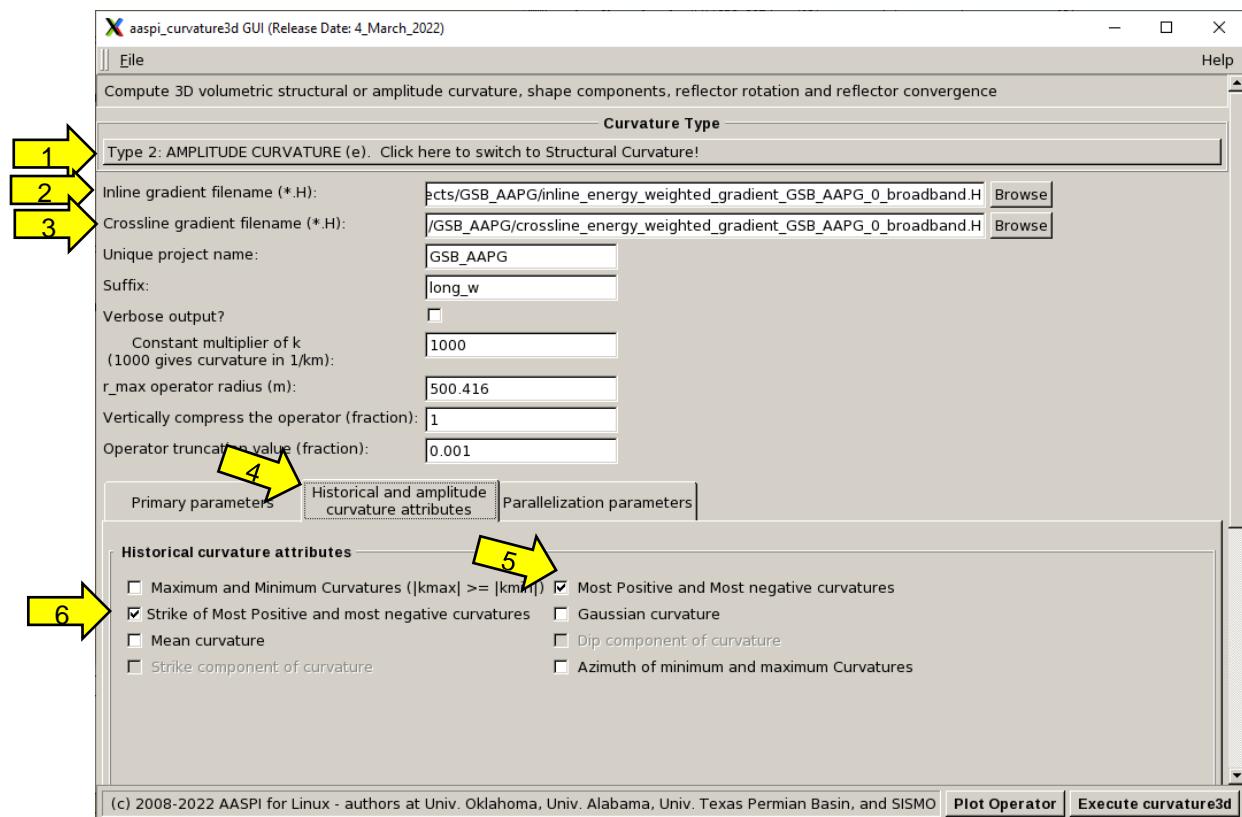
Finally, you also have the option to generate some (less useful) attributes similar to the historical structural curvature attributes:

Output file description	File name syntax
Maximum curvature value	e_max_ <i>unique_project_name_suffix</i> .H
Maximum curvature azimuth	e_max_azim_ <i>unique_project_name_suffix</i> .H
Minimum curvature value	e_min_ <i>unique_project_name_suffix</i> .H
Minimum curvature azimuth	e_min_azim_ <i>unique_project_name_suffix</i> .H

Geometric Attributes: Program **curvature3d**

Invoking the **curvature3d** GUI and computing amplitude curvature

The previous examples computed structural curvature, or the derivatives of the structural dip components. We may also wish to compute the amplitude curvature, or the derivatives of the amplitude gradient files *inline_energy_gradient_boonsville_0.H* and *crossline_energy_gradient_boonsville_0.H*.



To compute amplitude curvature, (1) click the curvature type button and bring up *Type 2: AMPLITUDE CURVATURE (e)*. While the structural curvature attributes begin with the letter 'k', the amplitude (or energy) curvature attributes will begin with the letter 'e'. When you (2) Browse to select the inline component you will only see attributes beginning with names beginning with the characters *inline_energy_weighted_gradient* and *inline_RMS_amplitude_weighted_gradient* in them.

The (3) corresponding crossline component should automatically appear. The design of the derivative operators is exactly the same as for structural curvature. However, certain attributes

Geometric Attributes: Program **curvature3d**

are not well defined. Because of the seismic wavelet, amplitude gradients make little sense in 3D Euclidean space (they might for impedance or Poisson's ratio). The horizontal dimensions are still spatial, measured in feet or meters, but the 'vertical' dimension is now in amplitude (measured in mV or some other unit). For this reason, you are not allowed to compute principal curvatures. In contrast, (5) the most-negative and most-positive curvatures e_{neg} , and e_{pos} are still well defined, so under the (4) *Historical and amplitude curvature attributes* tab place a checkmark in front of them. Likewise, the strike of these amplitude lineaments is also well defined. However this is no amplitude curvature computed in the structural dip and strike directions, which is disabled and appears as gray. Finally, click *Execute curvature3d* and wait for the job to complete.

The output files will look like:

```
|-rw-r---- 1 kmarfurt aaspi 46314 Nov 27 23:35 curvature3d_boonsville_v.v_10000ft.out
|-rw-r---- 1 kmarfurt aaspi 37 Nov 28 00:11 live_processor_list
|-rw-r---- 1 kmarfurt aaspi 2399 Nov 28 00:12 e_valley_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2400 Nov 28 00:12 e_s_index_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2399 Nov 28 00:12 e_saddle_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2398 Nov 28 00:12 e_ridge_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2403 Nov 28 00:12 e_pos_strike_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2396 Nov 28 00:12 e_pos_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2403 Nov 28 00:12 e_neg_strike_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2396 Nov 28 00:12 e_neg_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2397 Nov 28 00:12 e_dome_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2403 Nov 28 00:12 e_curvedness_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2397 Nov 28 00:12 e_bowl_boonsville_long_wavelength.H@0
|-rw-r---- 1 kmarfurt aaspi 2378 Nov 28 00:12 d_dz_operator3d.H@0
|-rw-r---- 1 kmarfurt aaspi 5682 Nov 28 00:12 d_dz_operator3d.H
|-rw-r---- 1 kmarfurt aaspi 2378 Nov 28 00:12 d_dy_operator3d.H@0
|-rw-r---- 1 kmarfurt aaspi 5682 Nov 28 00:12 d_dy_operator3d.H
|-rw-r---- 1 kmarfurt aaspi 2378 Nov 28 00:12 d_dx_operator3d.H@0
|-rw-r---- 1 kmarfurt aaspi 5682 Nov 28 00:12 d_dx_operator3d.H
|-rw-r---- 1 kmarfurt aaspi 300 Nov 28 00:12 d_dr_spectrum_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 294 Nov 28 00:12 d_dr_operator_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5792 Nov 28 00:13 e_neg_strike_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5792 Nov 28 00:13 e_dome_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5810 Nov 28 00:13 e_curvedness_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5792 Nov 28 00:13 e_bowl_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5798 Nov 28 00:13 e_valley_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5800 Nov 28 00:13 e_s_index_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5798 Nov 28 00:13 e_saddle_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5785 Nov 28 00:13 e_ridge_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5792 Nov 28 00:13 e_pos_strike_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5789 Nov 28 00:13 e_pos_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 5787 Nov 28 00:13 e_neg_boonsville_long_wavelength.H
|-rw-r---- 1 kmarfurt aaspi 49370 Nov 28 00:13 curvature3d_boonsville_long_wavelength.out
[kmarfurt@tripolite boonsville]$
```

Note the prefix 'e' before the attribute names. Let us use the *AASPI QC Plotting tab* to plot *e_neg_boonsville_long_wavelength.H* and obtain:

Geometric Attributes: Program **curvature3d**

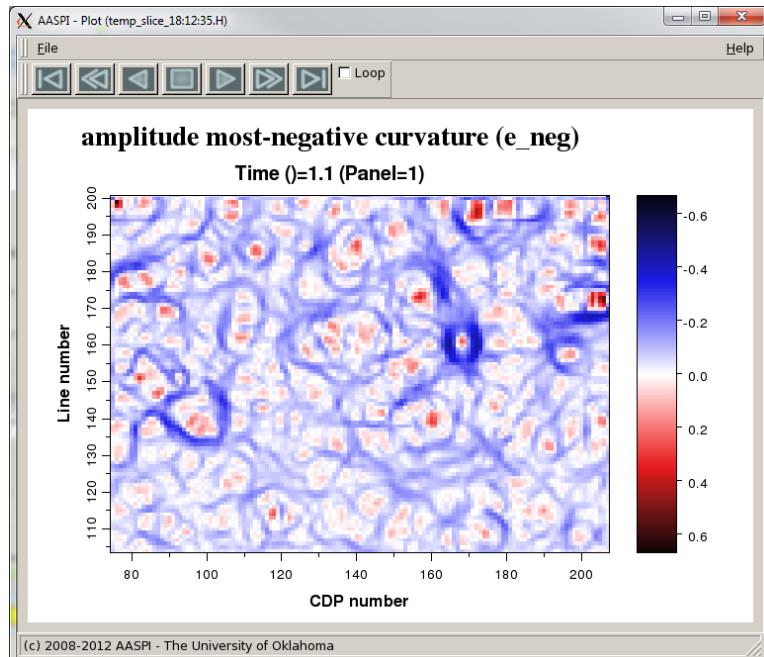


Figure 29.

Note the difference between the e_{neg} image and the k_{neg} image. e_{neg} will have strong negative (in this color scheme) black values where the coherent energy is minimum. To demonstrate, let us use PowerPoint to blend e_{neg} and the coherent energy images:

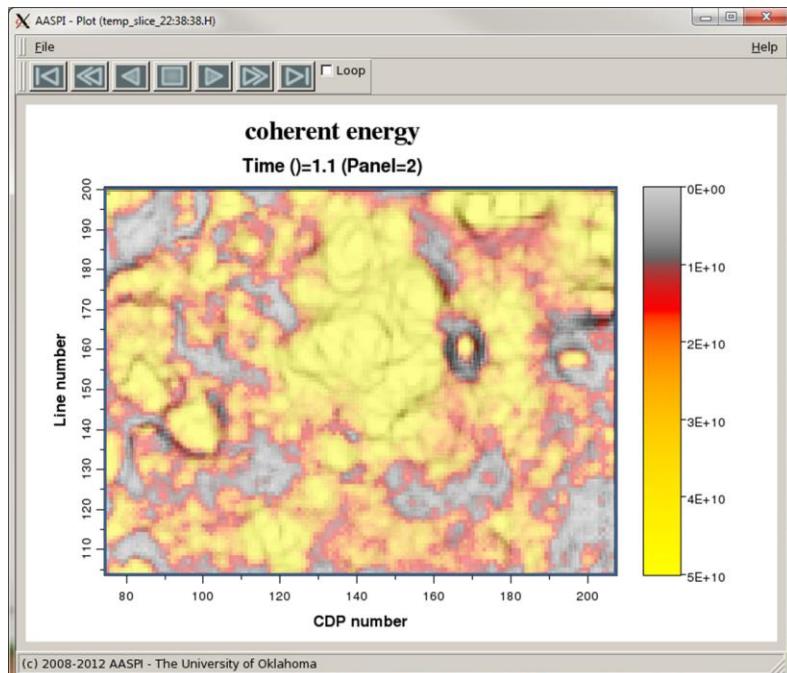


Figure 30.

Not surprisingly, the low values of e_{neg} track the low (dark blue) values of the coherent energy. However, these lineaments continue across the higher energy areas of the time slice, forming a

Geometric Attributes: Program **curvature3d**

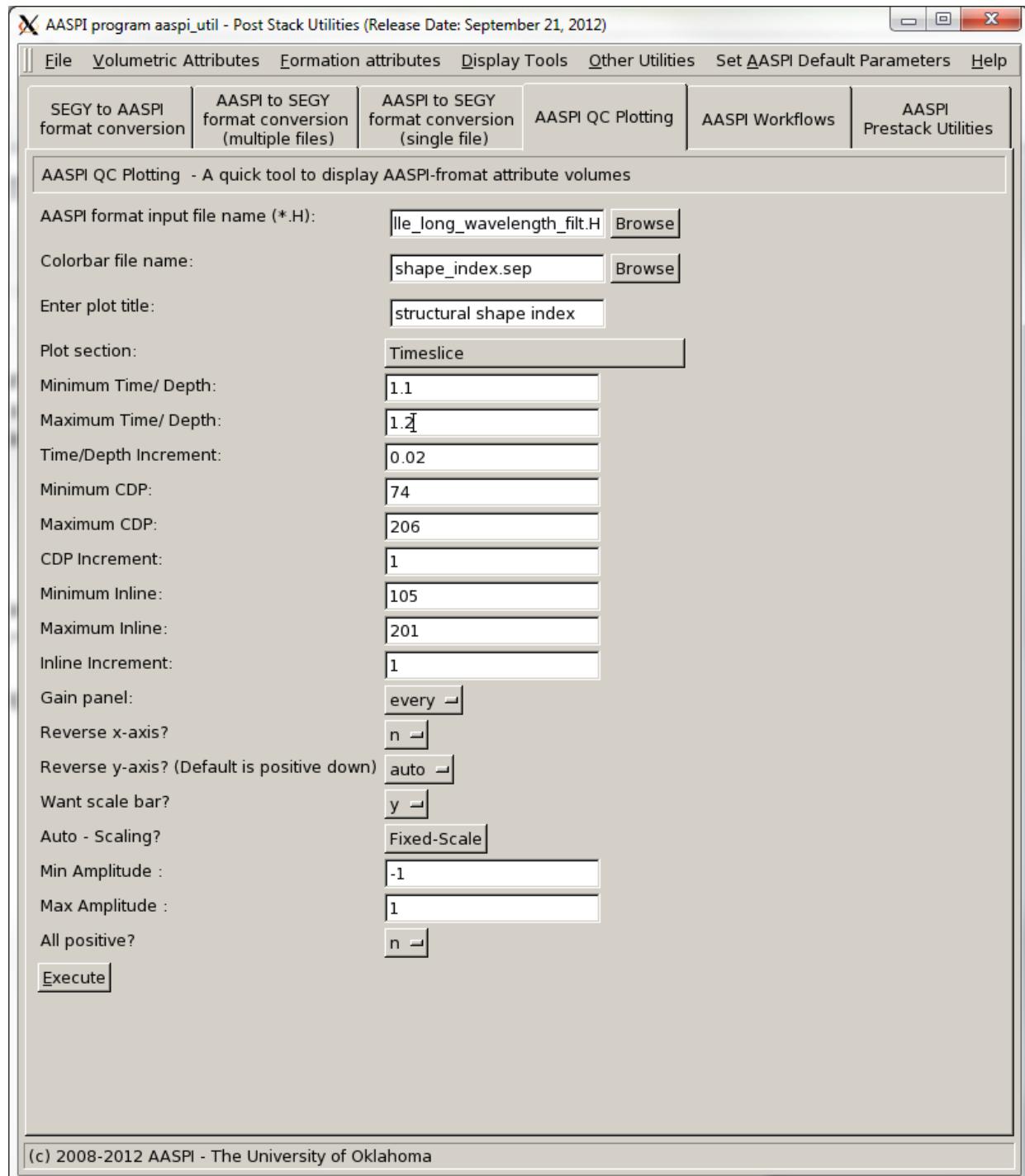
network. We have found e_{neg} and e_{pos} attributes to be very effective in mapping fractures (or very large cleats) in coal seams, as well as fractures in carbonates.

We will return to several of the other curvature attributes when we address the multiattribute display and rose diagram GUIs found under *Other Tools* later.

Plotting shape components

The shape of structural deformation can often be associated with a particular play. Carbonate buildups and injectites may appear as domes and karst collapse often appear as bowls. However, much like the azimuth of vector dip gains value by modulation by the magnitude of vector dip to differentiate strongly dipping from nearly flat features, the shape index needs to be modulated by the curvedness to differentiate strongly deformed features from nearly planar features. Plot the shape index file *k_shape_index_boonsville_long_w_filt.H* using the following AASPI QC Plotting tab parameters:

Geometric Attributes: Program curvature3d



Geometric Attributes: Program **curvature3d**

Curvedness and the Shape Index

Roberts (2001), Bergbauer et al. (2003), and al-Dossary and Marfurt (2006) show how principal components of curvature, k_1 and k_2 , can be combined to generate a shape index. Shape indices were developed for terrain analysis and are commonly used in meteorological and ecological studies.

Curvedness, C , is a measure of total deformation and is defined as:

$$C = (k_1^2 + k_2^2)^{1/2}.$$

where k_1 and k_2 are the most-positive and most-negative principal curvatures, with

$$k_1 \geq k_2.$$

The shape index, s , is defined as

$$s = -\frac{2}{\pi} \text{ATAN}\left(\frac{k_2 + k_1}{k_2 - k_1}\right).$$

The values of the shape index range between -1.0 and +1.0 with $s=-1.0$ indicating a bowl, $s=-0.5$ a valley, $s=0.0$ a saddle, $s=+0.5$ a ridge, and $s=+1.0$ a dome. If the curvedness $C=0.0$, the shape index is undefined, and we have a perfect plane:

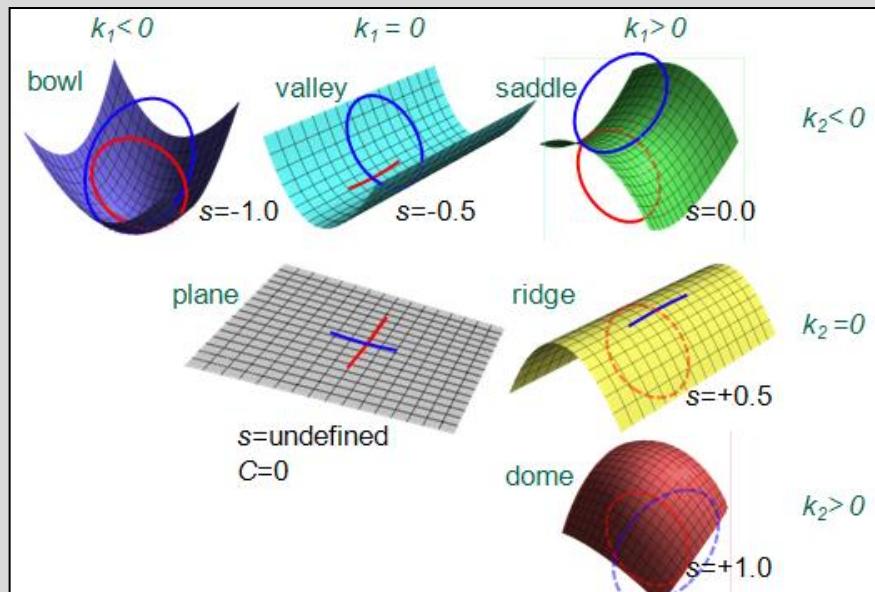


Figure A1. (Figure after Mai, 1999)

Use the *color bar file shape_index.sep*, set *Clip* to 1.0 (since the data will range from -1.0 to +1.0) and *All positive* to *n*. The image at *t=1.1* s will look like the following (see next page):

Geometric Attributes: Program **curvature3d**

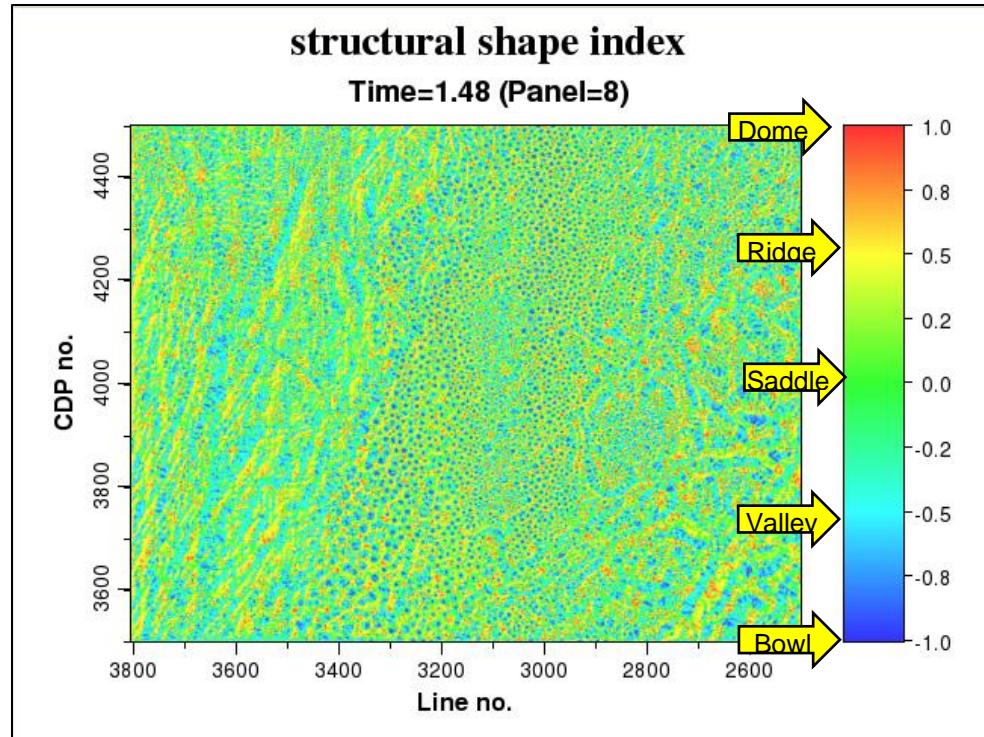


Figure 22.

where the arrows indicate the shape on the 1D color bar to the right. Plotting the curvedness results in the following image:

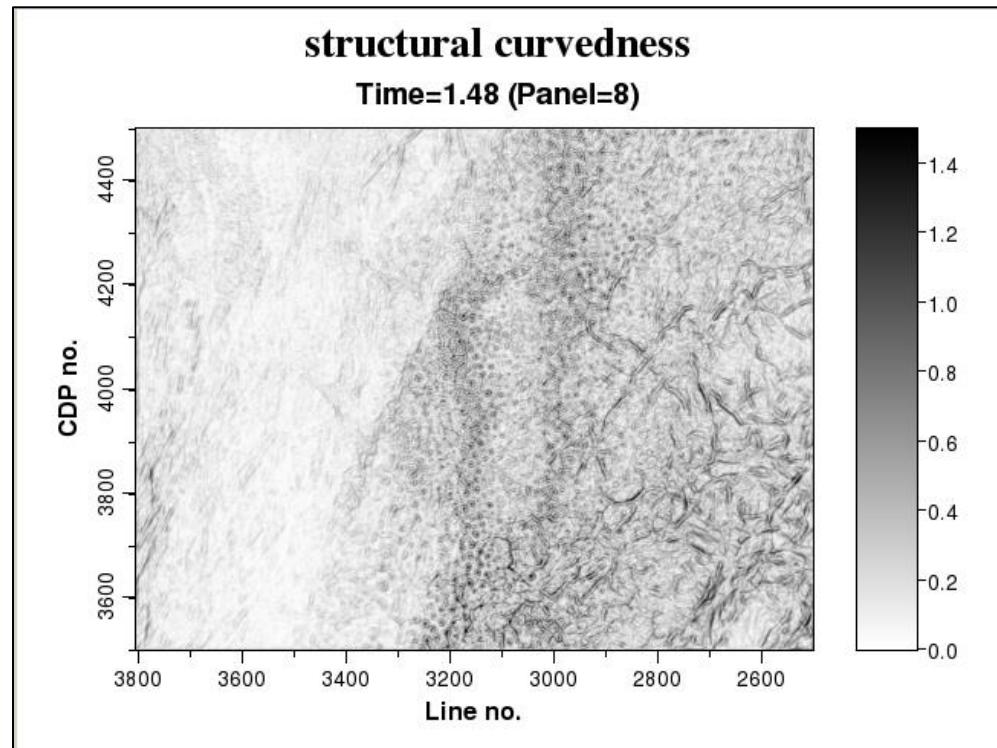
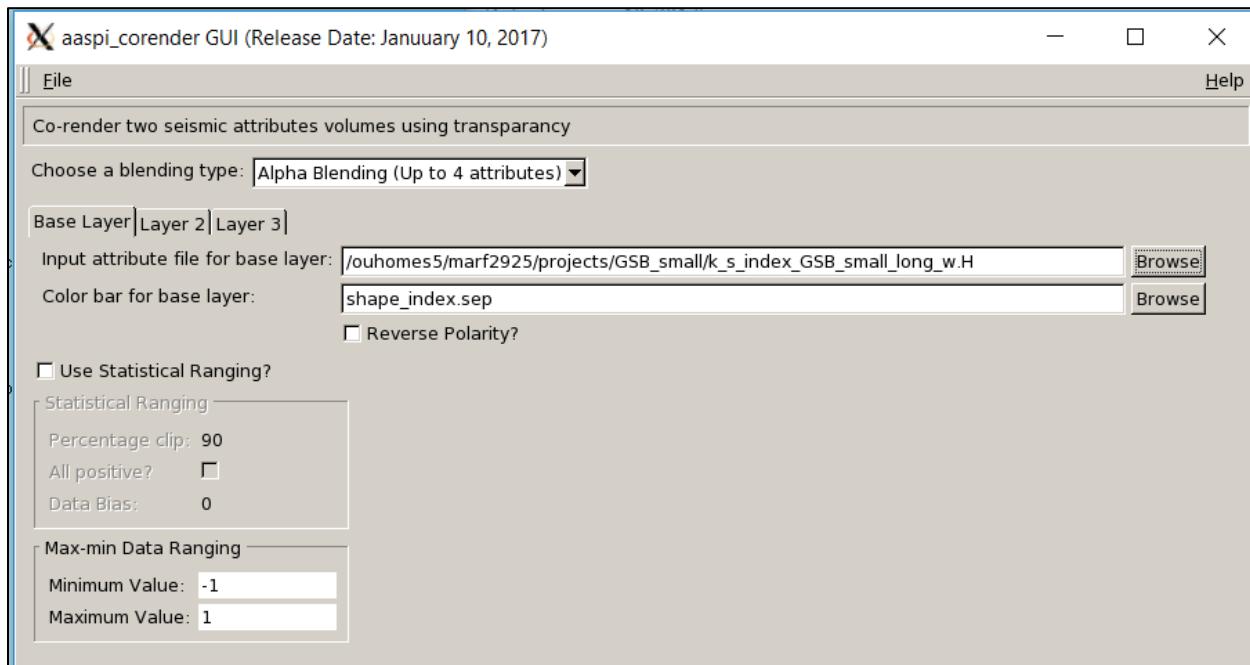


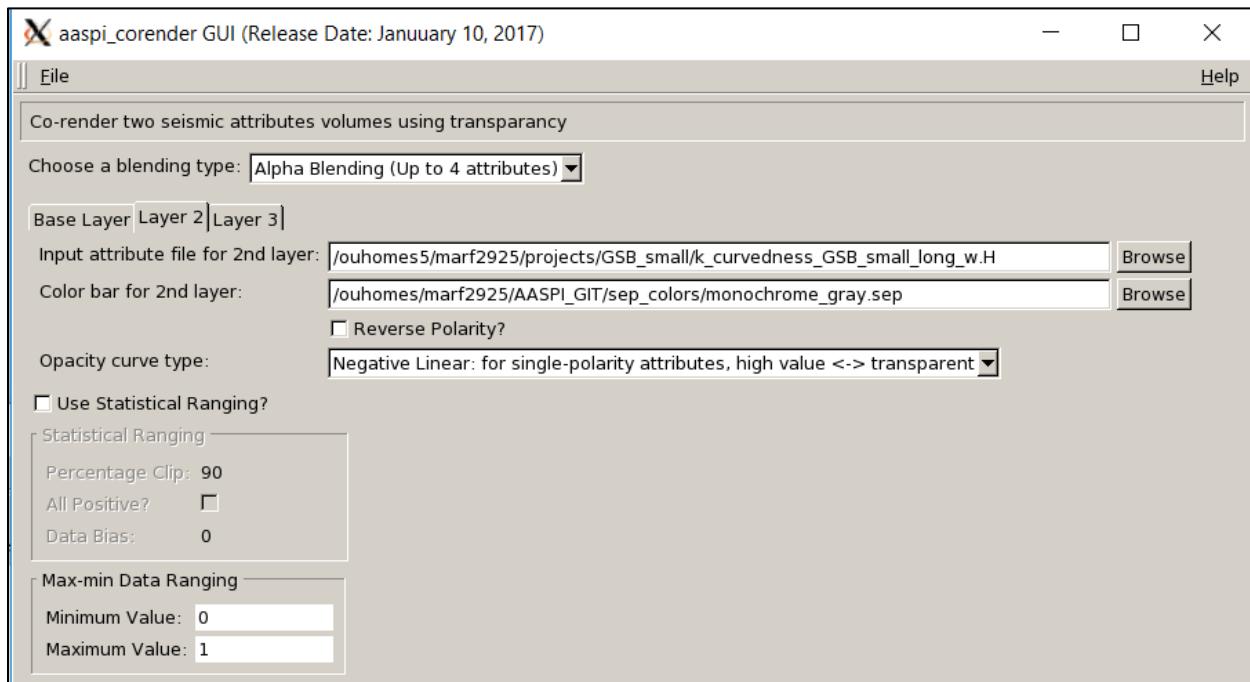
Figure 23.

Geometric Attributes: Program **curvature3d**

To blend the two images use the program **corender** and set the background layer to be the shape index *k_s_index_GSB_small_long_w*:



and layer two to be the curvedness *k_curvedness_GSB_small_long_w*:



To obtain the following image (see next page):

Geometric Attributes: Program **curvature3d**

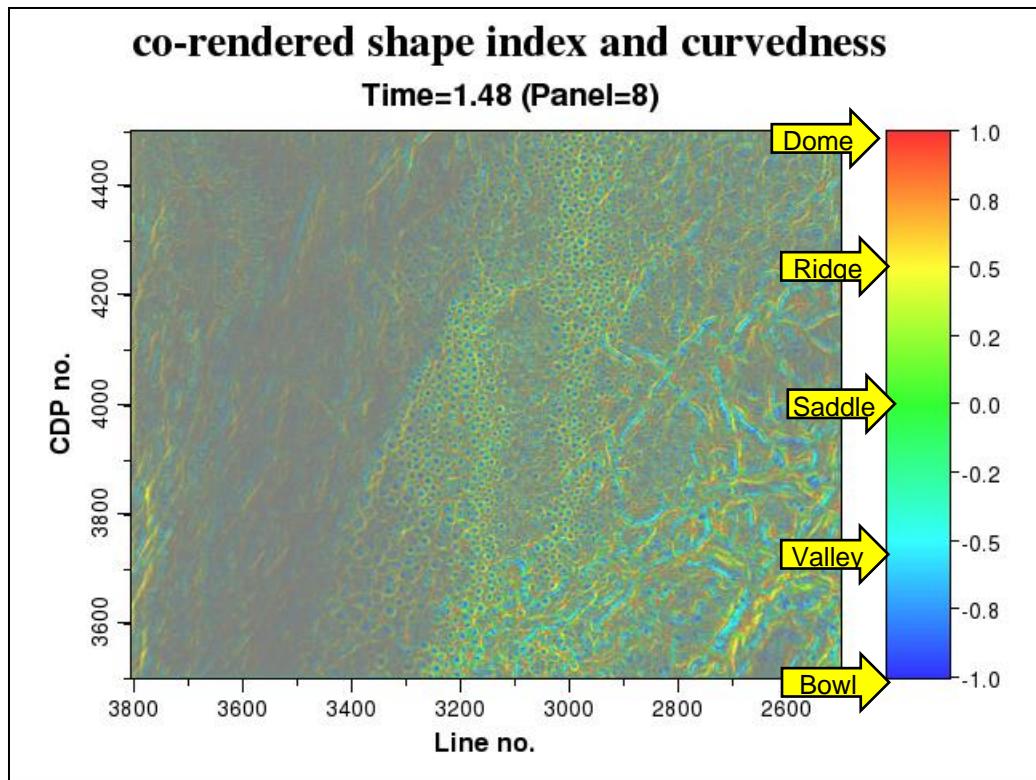
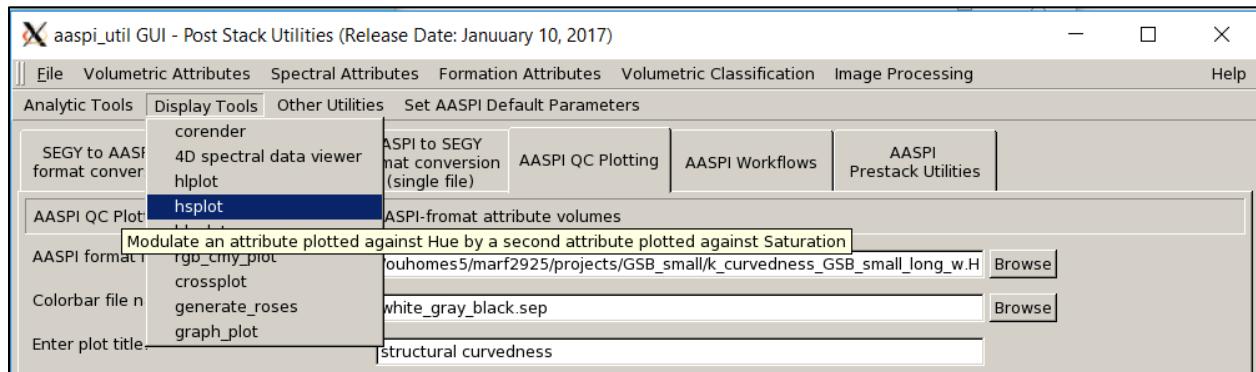


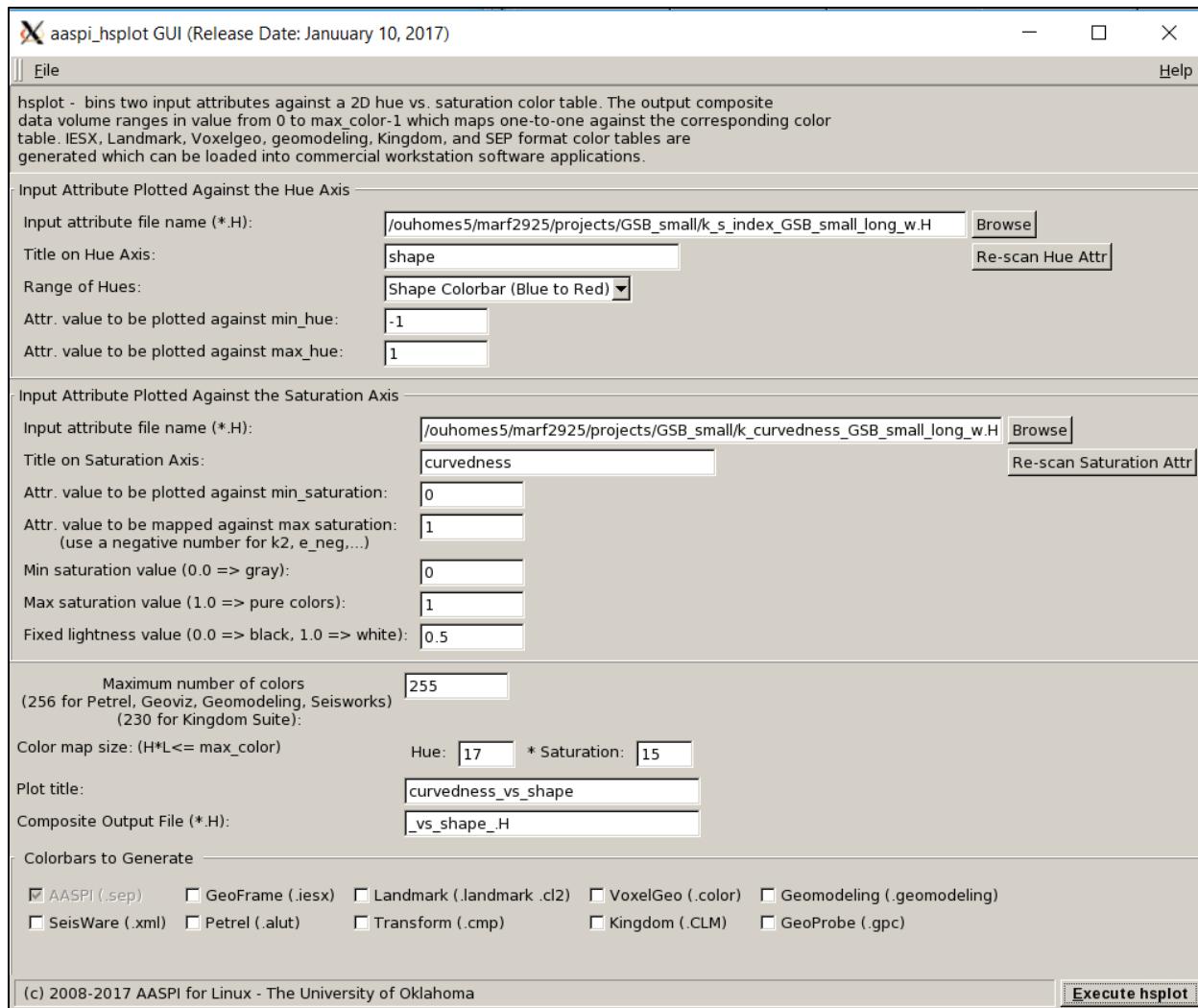
Figure 24.

Instead of simply blending the two attributes using program **corender**, one can directly map them to a 2D color bar using program **hsplot**, found under *Display Tools* on the **aaspi_util** GUI:



The input files and ranges are the same as for **corender**:

Geometric Attributes: Program curvature3d



This program creates an explicit 2D color bar that can be loaded into most interpretation software packages, as observed on the following page:

Geometric Attributes: Program **curvature3d**

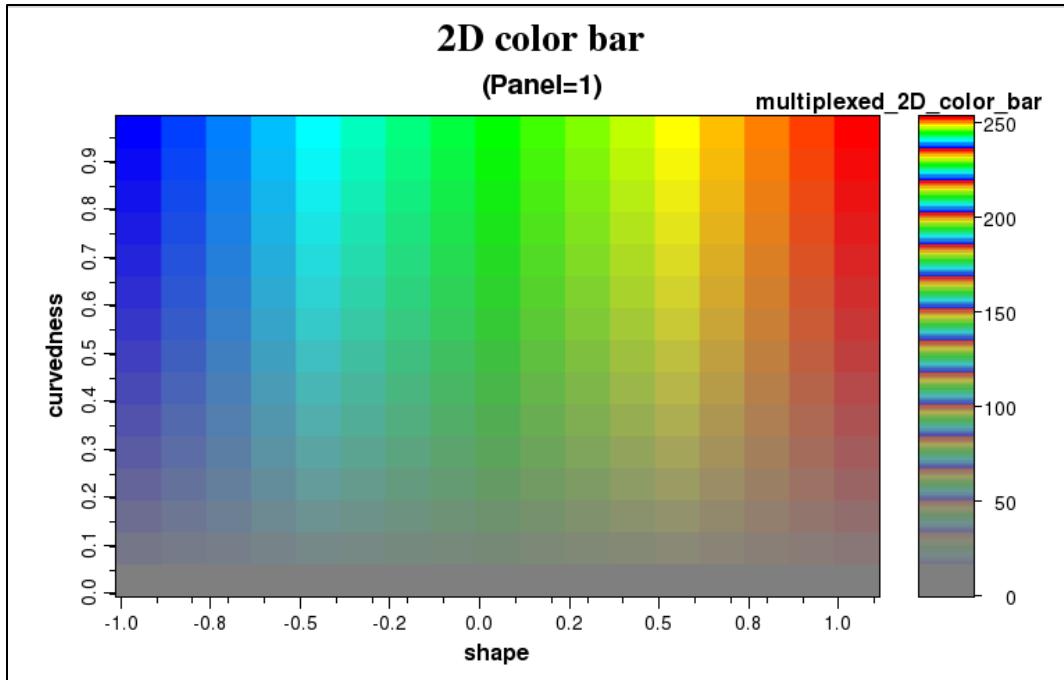


Figure 25.

resulting in the following image:

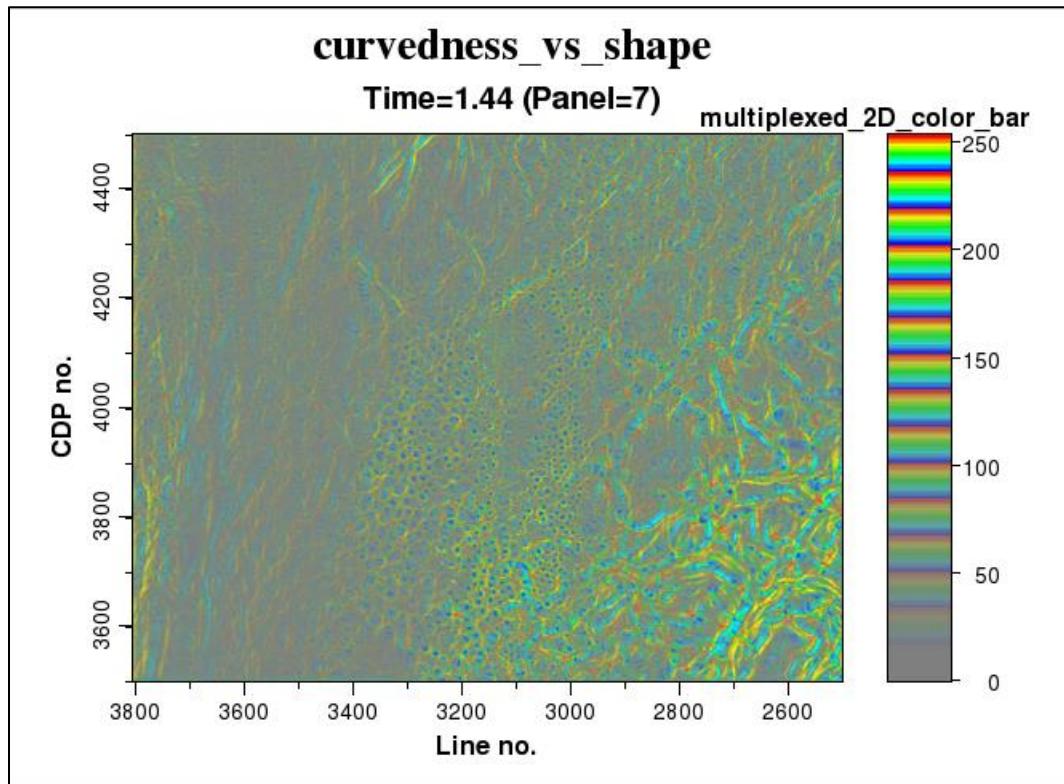


Figure 26.

Geometric Attributes: Program **curvature3d**

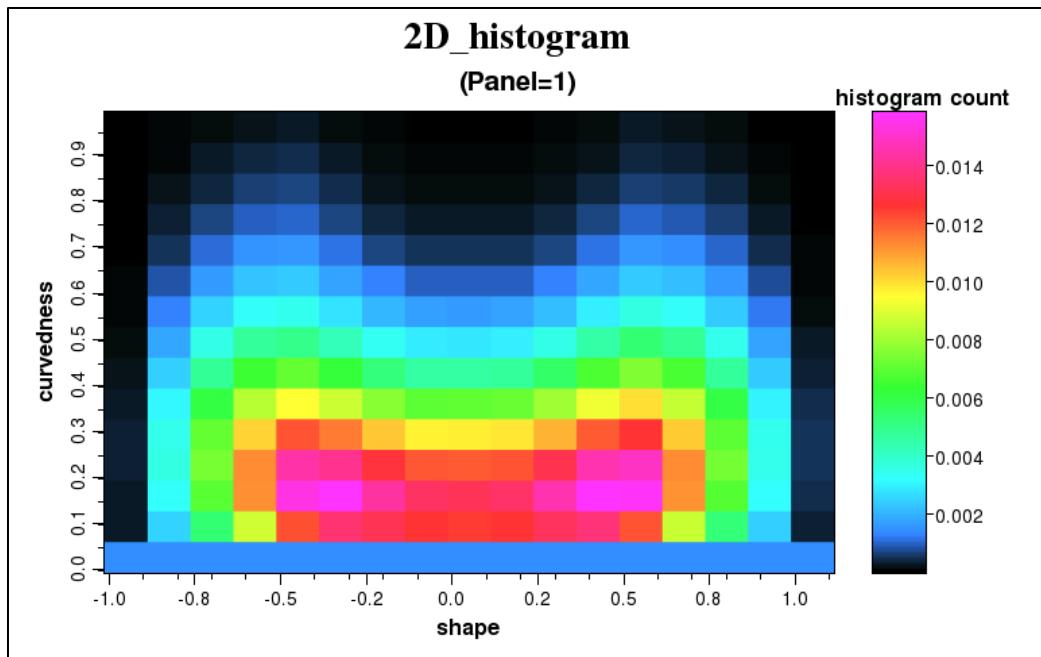


Figure 27.

Most of the data appear to be moderately deformed, with valleys, ridges, and saddles the more common shape.

These composite attribute images can be converted back to SEG-Y format and coupled with their corresponding color bars (see documentation on program **hsplot**) and can be displayed in most interpretation workstations. Remember that the data needs to be scaled with user-defined parameters set to a minimum value of 0 and maximum value of 255 for 256 colors when converting to 8-bit data. In this case discussed here, I am using 4096 colors, so I need to store the data as 16-bit integers and set the values to range between 0 and 4095 to obtain a one-to-one data value to color value mapping.

Plotting shape components

While composite curvedness vs. shape index image above is amenable to visual interpretation it is less amenable to statistical analysis. To allow statistical analysis program **curvature3d** outputs each of the shape components as individual volumes. Basically, a bowl shape is assigned the intensity of the curvedness if the shape index falls between values of -1.0 and -0.75. Similarly, the valley-shape attribute is assigned the intensity of the curvedness if the shape index falls between -0.75 and -0.25. Your plot of the bowl component *k_bowl_boonville_long_w_filt_0.H* should look like the following images:

Geometric Attributes: Program **curvature3d**

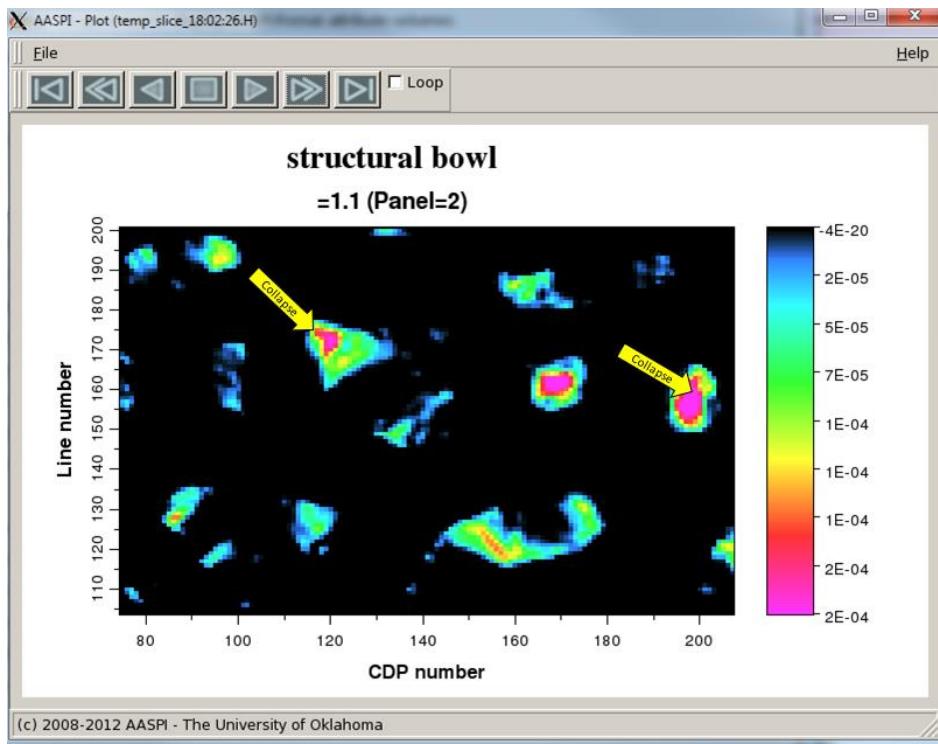


Figure 28.

Note the collapse features appear as strong amplitude (pink) bowls. Similarly, you can plot the valley-shape component, *k_valley_boonville_long_w_filt.H*, which looks like:

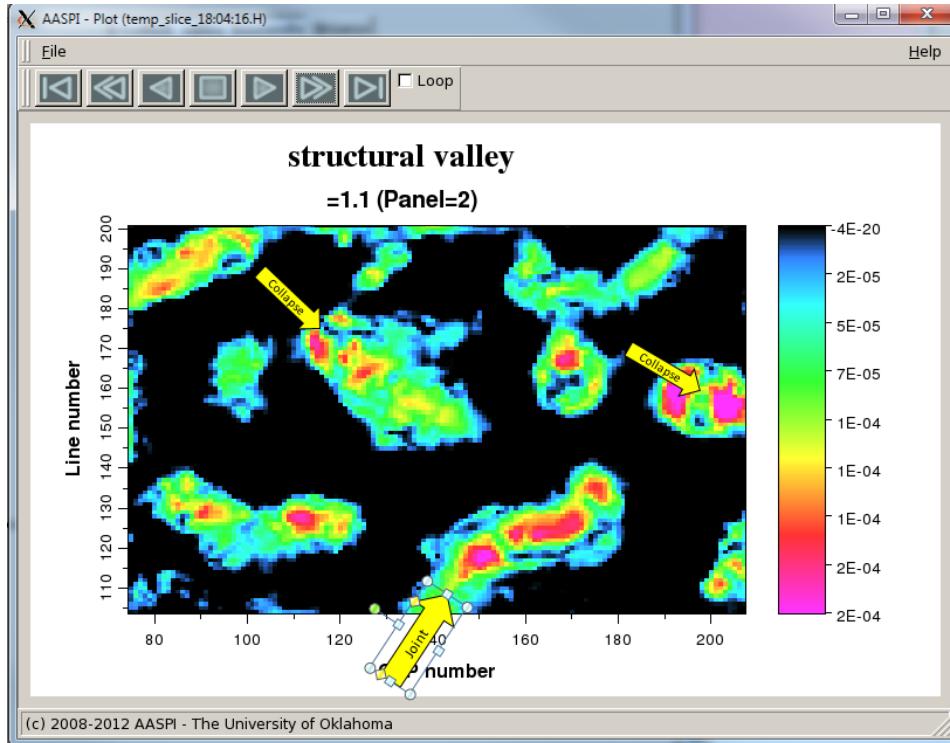


Figure 29.

Geometric Attributes: Program **curvature3d**

Note that the collapse features are lower amplitude (their valley component approaches 0) while circular zones around them are classified as valleys – star-shaped lows radiating out from the collapse features. We also see two prominent joints that appear as valleys.

Plotting lineament intensity vs. lineament strike

The strike is undefined for plane and perfectly formed domes, bowls, and saddles. In contrast, it is well defined for valley and ridge shapes. You will therefore want to modulate the lineament strike by the intensity of either the valley or ridge component. Since the Ellenburger formation in the Boonsville survey is characterized by diagenetically altered (karstified) fractures that appear as valleys, the valley component is the appropriate attribute to choose. For reverse and thrust faulting and folding, the ridge component may provide greater information as discussed in a paper by Mai et al. (2009):

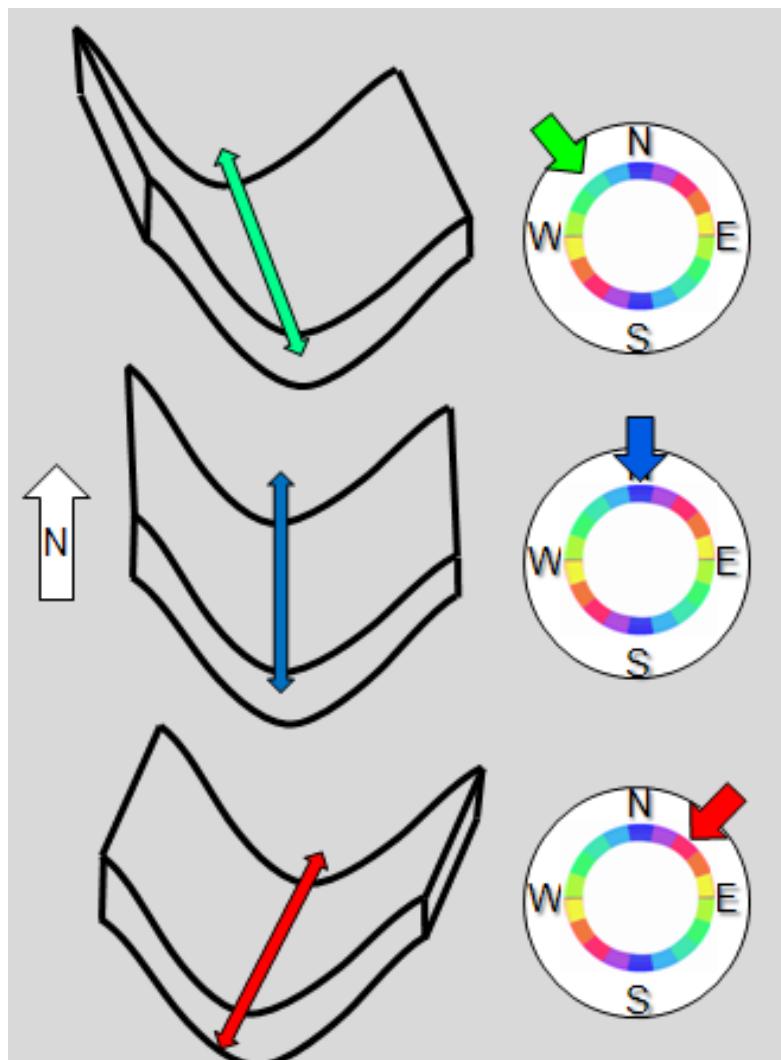
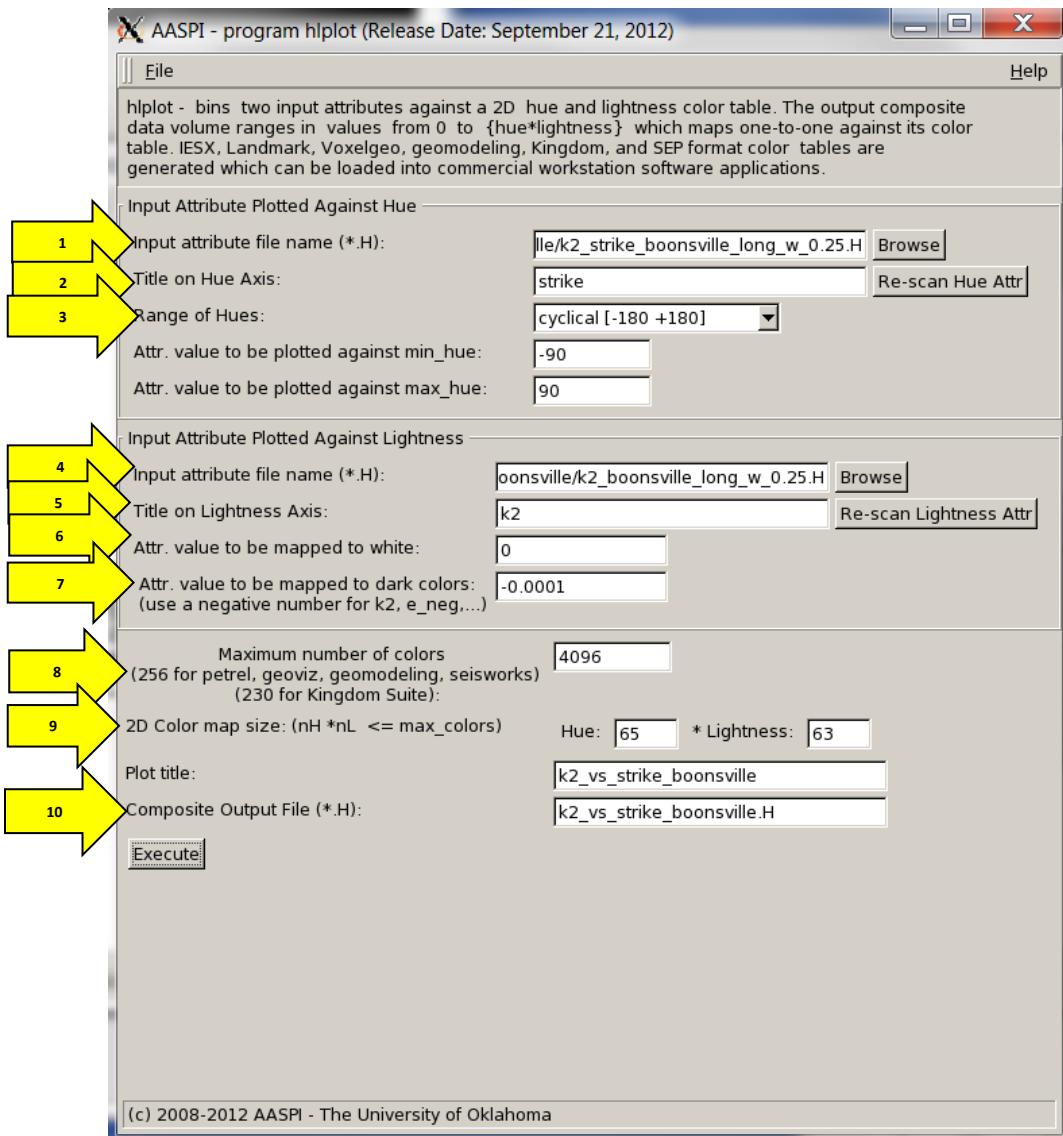


Figure 30.

Geometric Attributes: Program **curvature3d**

To modulate the strike of the Boonsville valleys, re-invoke program **hlplot** and fill in the parameters as shown below.



In this example we want to plot (1) k2_strike_boonsville_long_w_0.25.H against hue and (4) most negative curvature k2_boonsville_long_w_0.25.H against lightness. Label your axes (2) as *strike* and (5) *k2* to annotate properly your color bar file name and output 2D color bar axes. The Range of Hues goes from -180 (yellow for east-west) through 0 (blue for north-south) to 180 (yellow for west-east). The range of the azimuth of strike should always be set to be -90 to +90. The range of the k2 curvature should be a (6) value of 0.0 to be plotted against white and (7) a value of -0.0001 to be plotted against darker colors. I enter (8) the number of colors as 4096, which is the maximum number I can load 4096 colors into Petrel using our Ocean application. The (9) 65 hue by 63 lightness bins gives 4095 colors with the last 4096th color being used for white. Provide (10) an output file name and click *Execute*:

Geometric Attributes: Program **curvature3d**

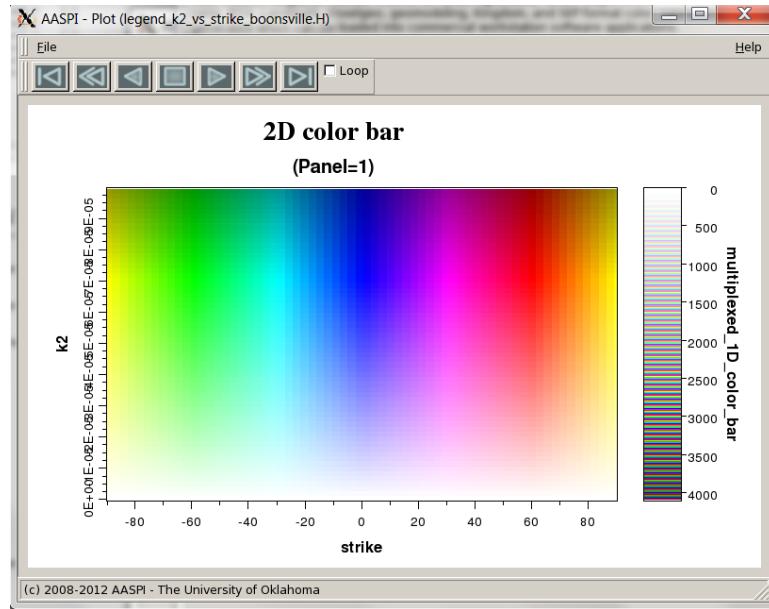


Figure 31.

and note the 2D color bar is as we designed it where values at $+90^\circ$ have the same color as those at -90° . We will also obtain an image of the color bar plotted as a wheel:

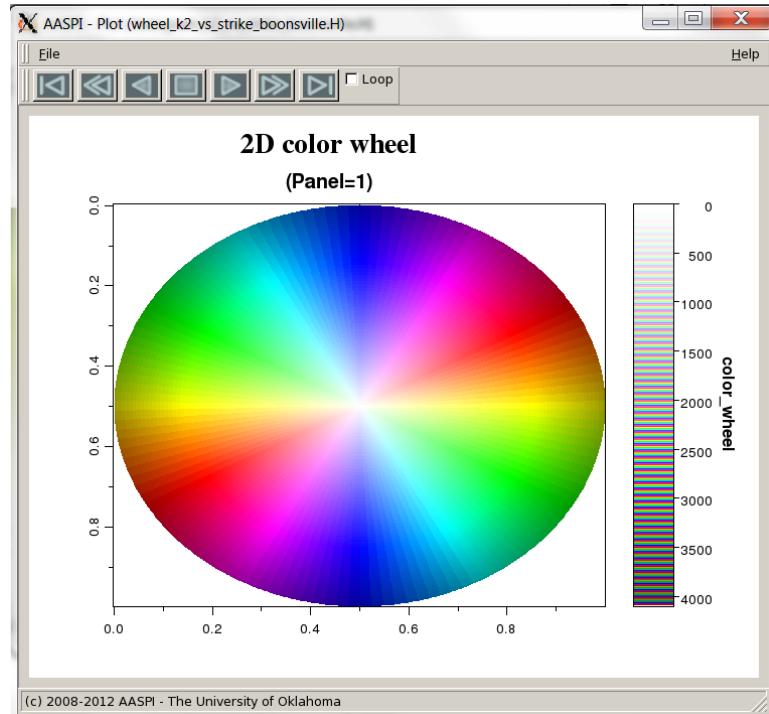


Figure 32.

Geometric Attributes: Program **curvature3d**

The histogram below indicates that there is a relatively even distribution of lineaments along all strike directions:

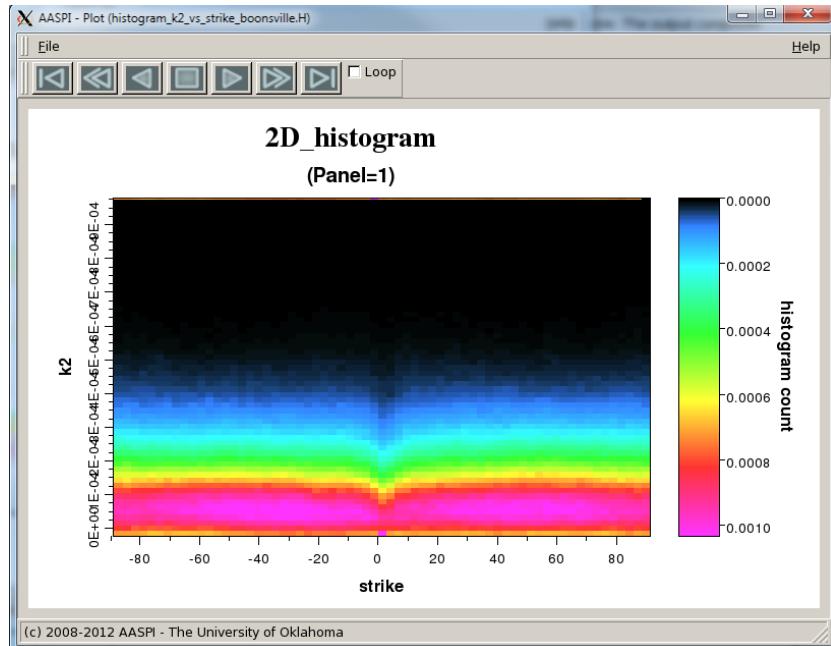


Figure 33.

Examining the time slice at $t=1.1$ s, we note the strong lineaments previously seen on the $k2$ curvature image, now are color coded according to their strike direction with colors ranging from yellow (-90° or west-east) through teal (-45° or northwest-southeast), blue (0° or north-south), salmon ($+45^\circ$ or northeast-southwest) to yellow ($+90^\circ$ or east-west):

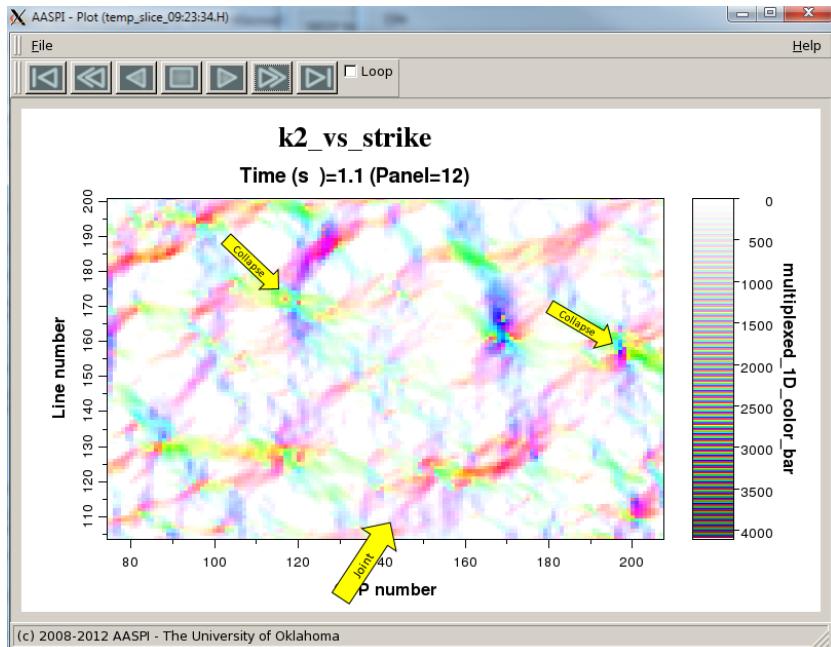


Figure 34.

Geometric Attributes: Program **curvature3d**

An advantage of using white as a background color is that it allows simply, co-rendering with coherence using opacity:

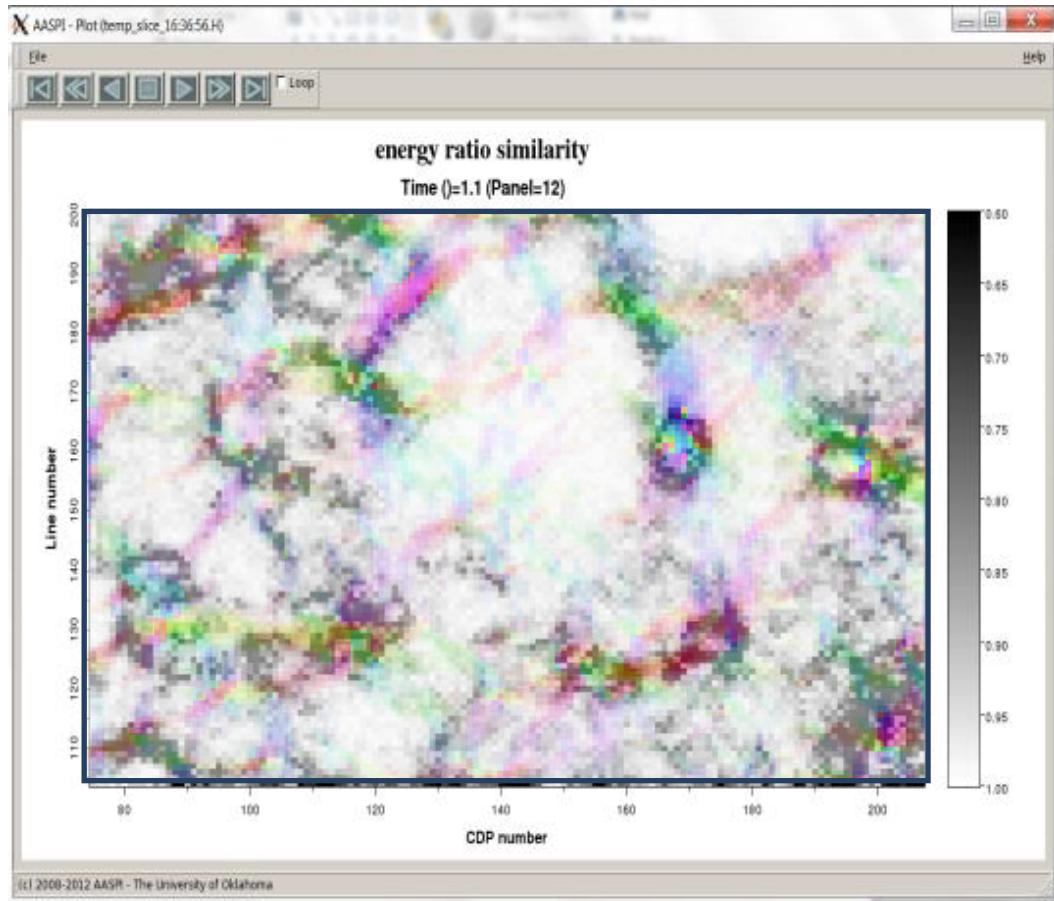
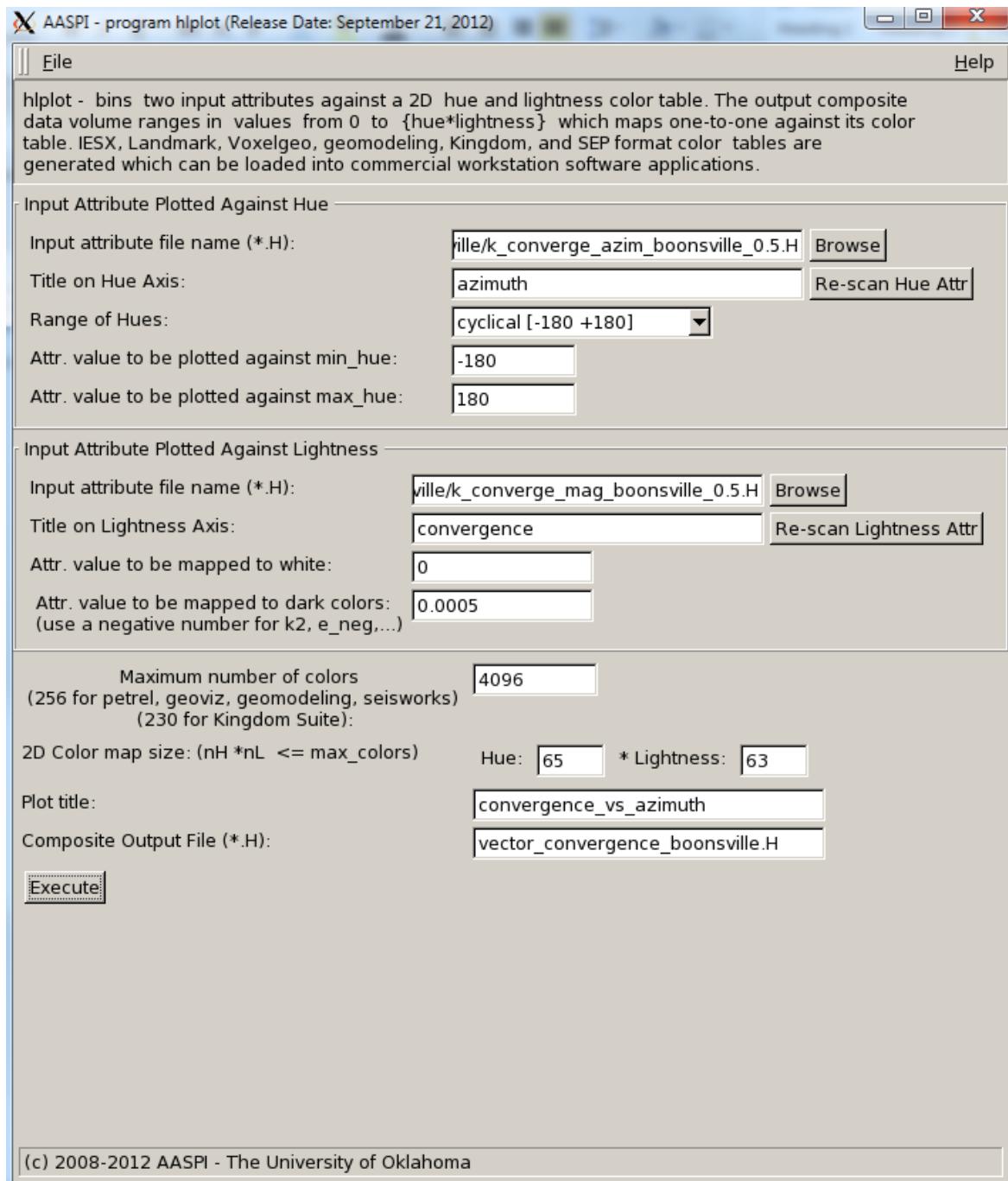


Figure 35.

Reflector Convergence

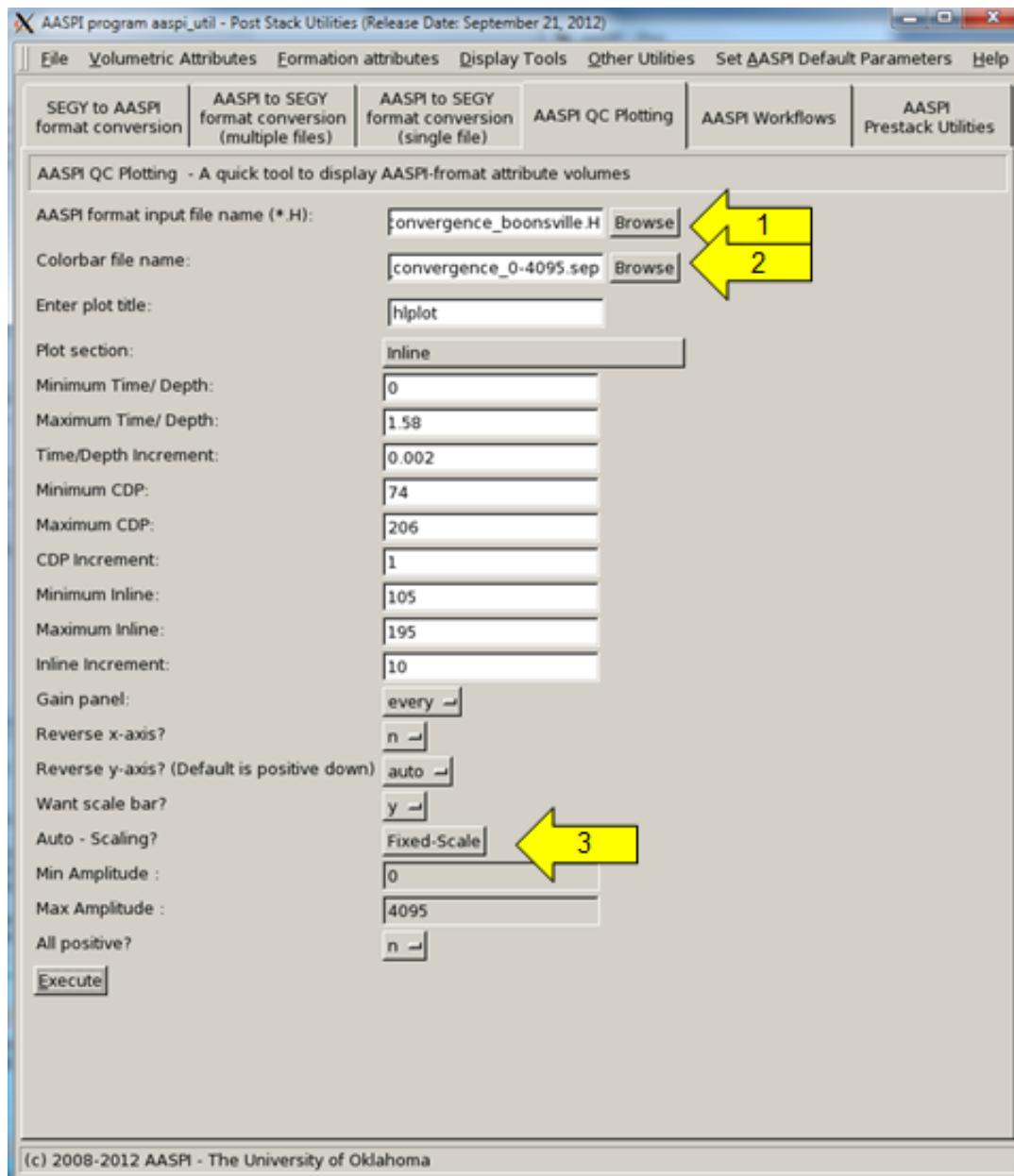
Reflector convergence is a vector that measures both the azimuth and magnitude to which reflectors converge, providing a means of mapping angular unconformities, the degree and direction of progradation, overbank deposits, and simple thickening and thinning of beds. Since we are treating this attribute as a vector, divergence to the east will be displayed as convergence to the west. We modulate the azimuth of convergence by its magnitude using program **hplot** (see next page):

Geometric Attributes: Program **curvature3d**



Several QC time slices will be generated, along with a histogram and color wheel. Remembering the output file name we assigned to the plot at the bottom of the panel, we can type that into the *AASPI QC Plotting* tab and obtain.

Geometric Attributes: Program curvature3d



Where (1) indicates the file name that needs to be typed. Note that with the September 2012 release, the (2) color bar, and (3) range of data scaling are carried along with most of the attribute files. The following two images appear, where the I have co-rendered them with seismic amplitude on the vertical slice and coherence on the time slice. I appear to have some edge effects on the left and right edges of the computation for this data volume.

Geometric Attributes: Program **curvature3d**

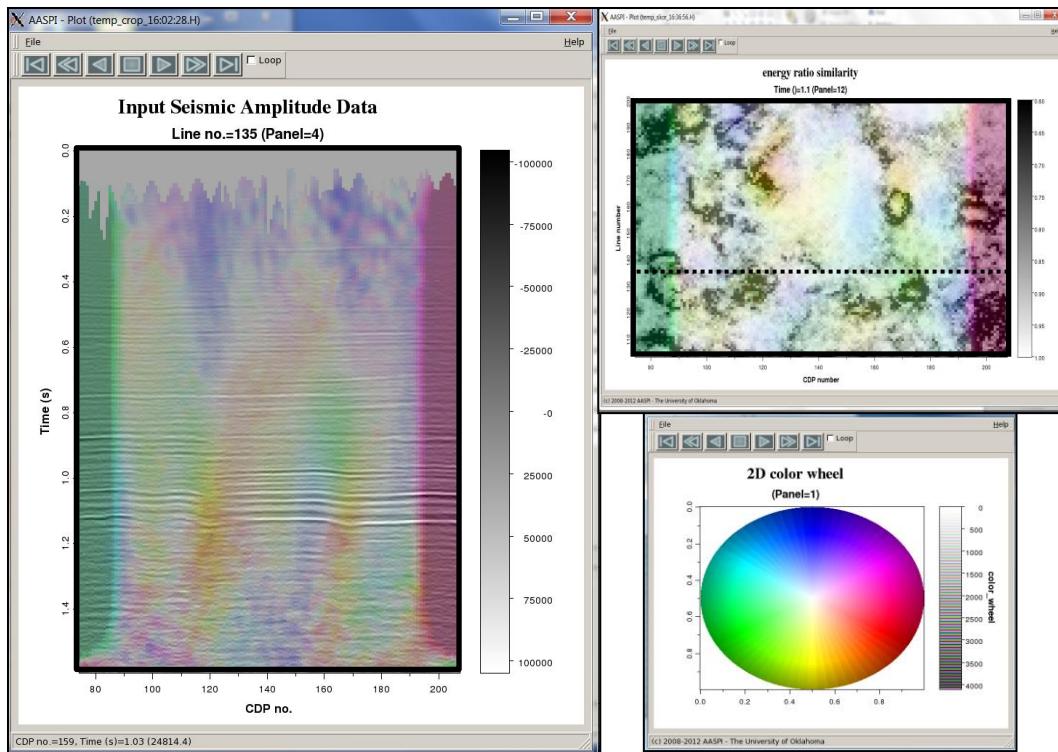
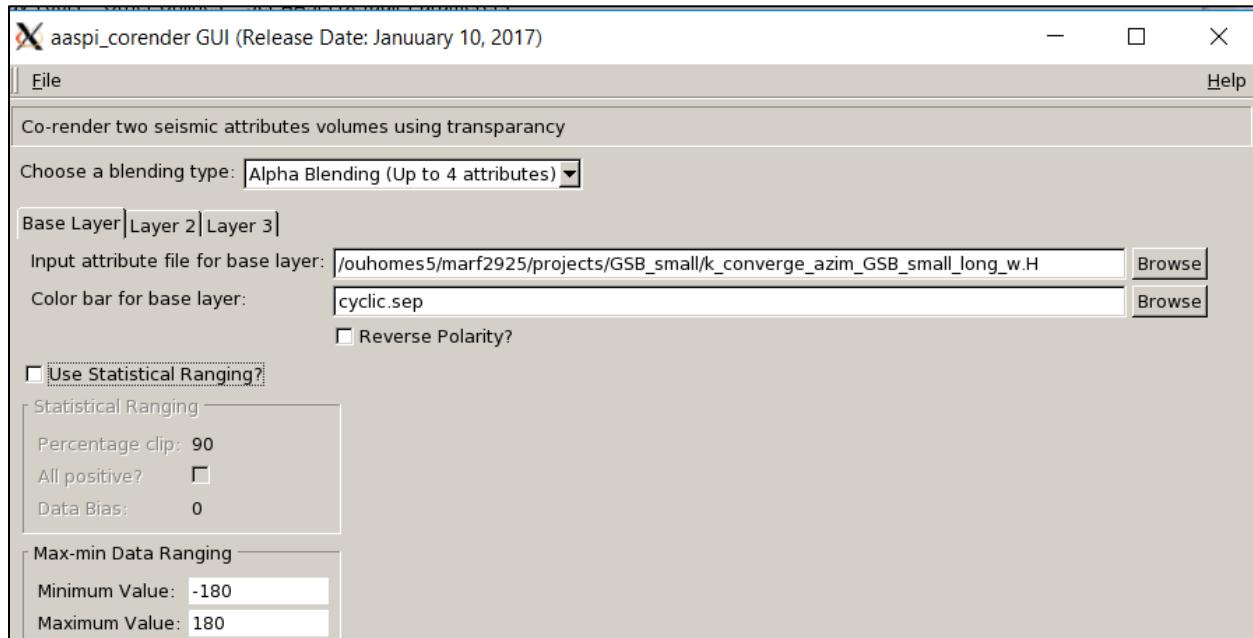


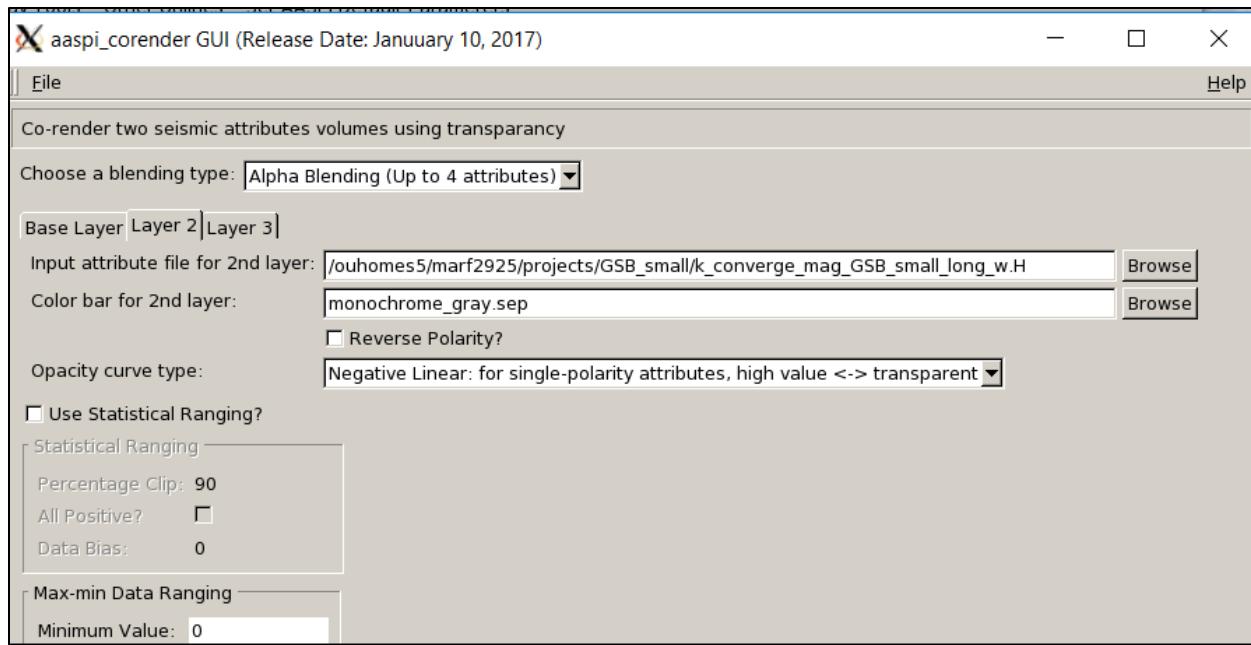
Figure 36.

Using program **corender**, the base layer will be *k_converge_azim_GSB_small_long_w.H* plotted against a cyclic color bar.



Layer 2 will be *k_converge_mag_GSB_small_long_w.H* plotted against a monochrome gray color bar with high magnitude values being transparent so that the azimuth can show through:

Geometric Attributes: Program **curvature3d**



The corresponding time slice is then:

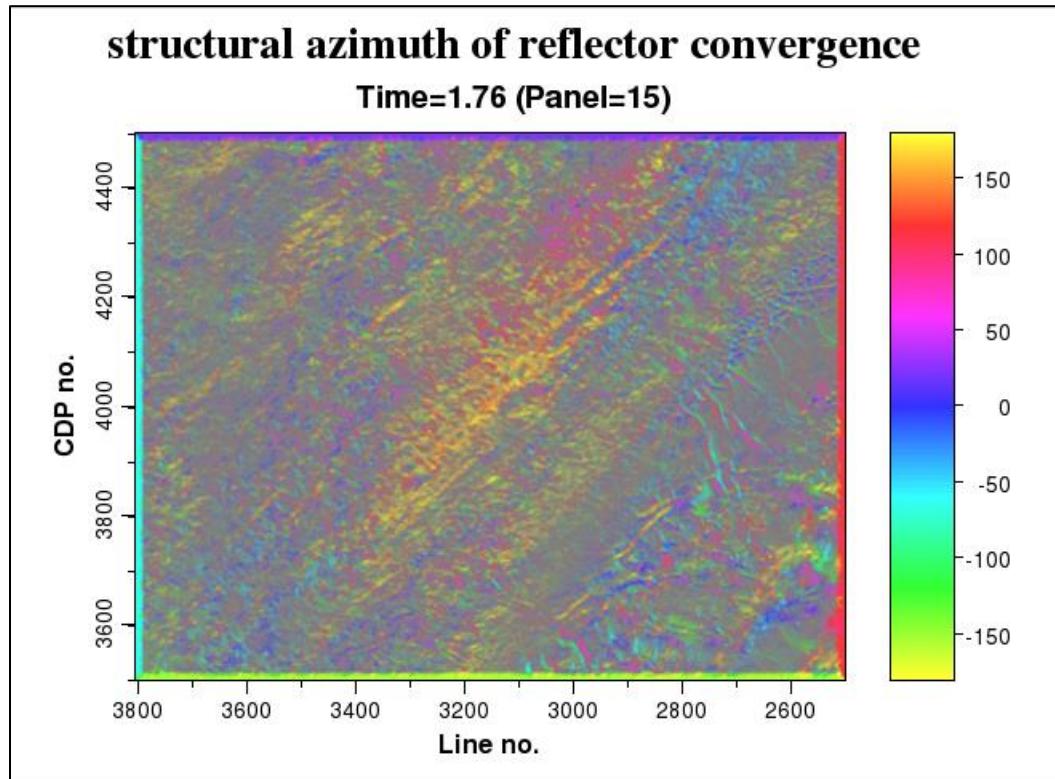


Figure 37.

Geometric Attributes: Program **curvature3d**

where the orange anomaly in the middle corresponds to prograding, downlapping onto the deeper water to the southeast. Shallower, at $t=1.68$ s, one sees the corresponding toplap of the sequence, in this case pinching out (converging) to the northwest, appearing as cyan:

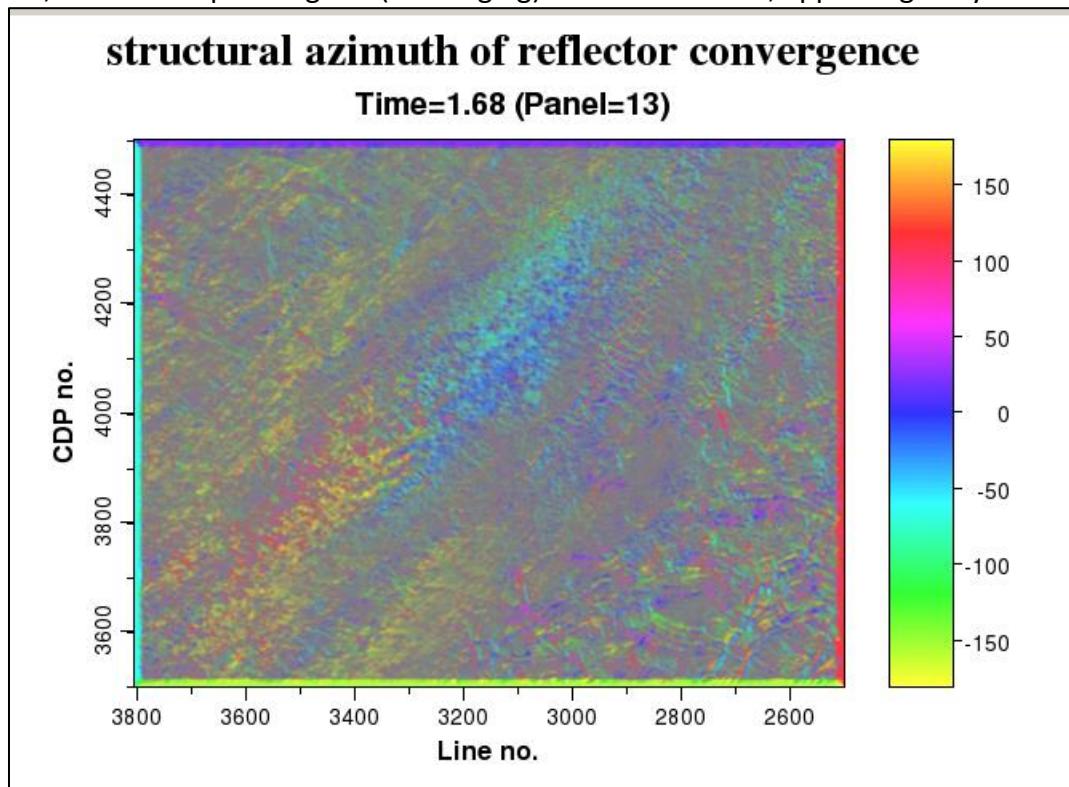
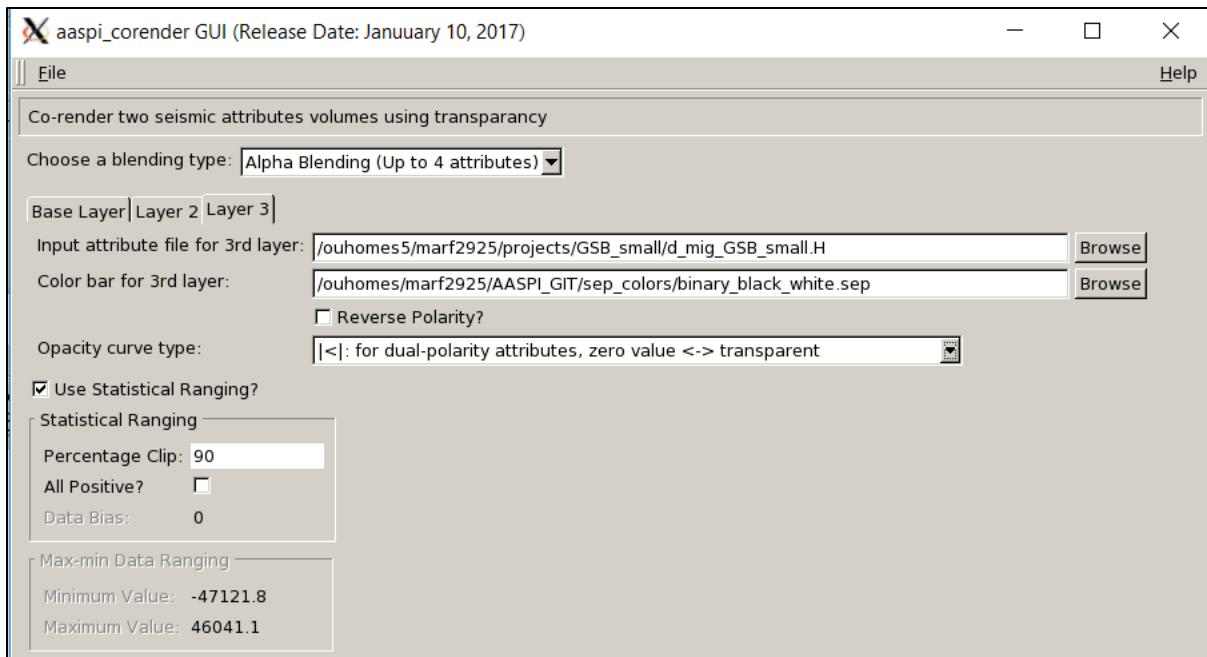


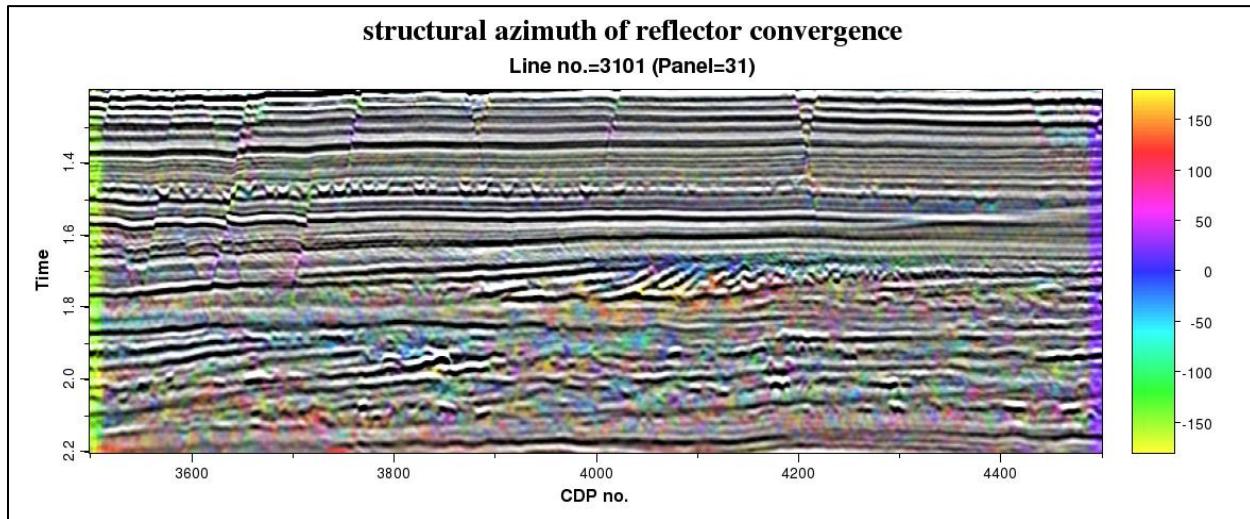
Figure 38.

To better understand these relationships, add the seismic amplitude data volume *d_mig_GSB_small.H* as a third layer to program **corender**:

Geometric Attributes: Program **curvature3d**



and obtain an image for inline 3101 like this:



and a corresponding image for CDP (crossline) 4000 (plotted with x-reverse=y) like:

Geometric Attributes: Program **curvature3d**

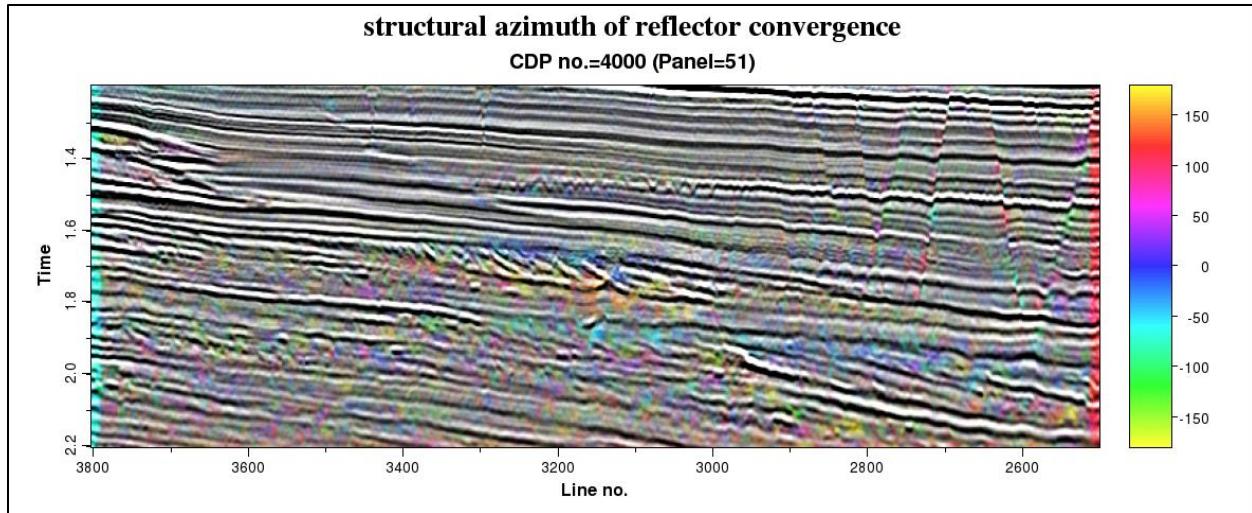


Figure 40.

Note the cyan color at the top of the prograding package and the yellow color at the bottom.

Geometric Attributes: Program **curvature3d**

Reflector Convergence

To compute reflector convergence, we need to convert our inline and crossline components of dip from degrees to the reciprocal of the slope p and q , measured in m horizontal per m vertical. To compute the normal, we note that the reciprocal of the slope in the vertical direction is 1 m horizontal per m vertical. Normalizing these three components, Marfurt and Rich (2010) showed how to compute the unit normal:

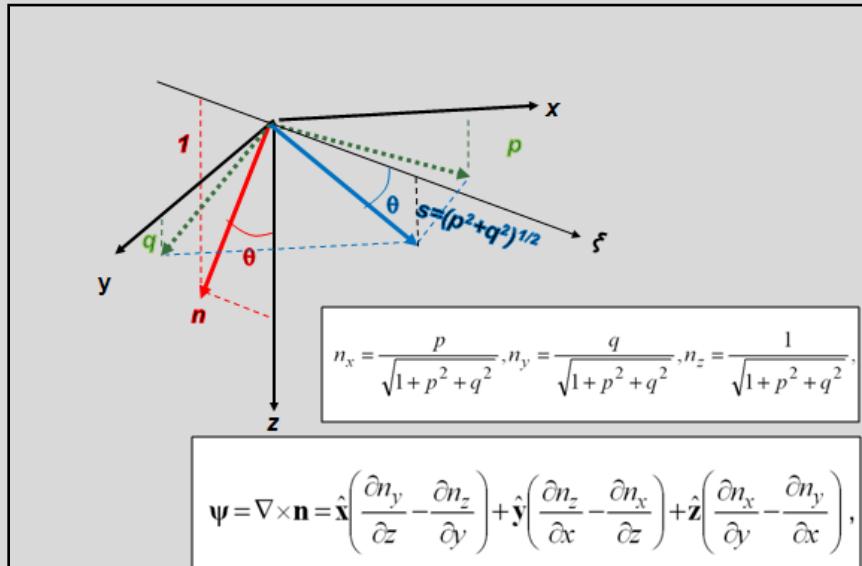


Figure B1.

Such unit normals fall out automatically from dip estimation techniques based on the Gradient Structure Tensor. To measure rotation, we compute Ψ , the curl of the unit normal, \mathbf{n} . For interpretation purposes, we break this 3-component curl into two parts – the part that rotates about the average unit normal, $k_{rot} = \bar{\mathbf{n}} \cdot \Psi$, which we will associate with syntectonic deposition and structural rotation about faults and $\mathbf{k}_{conv} = \bar{\mathbf{n}} \times \Psi$, which is a vector that we will associate with angular unconformities, downlap, toplap, and other stratigraphic phenomena:

$$\begin{aligned} \mathbf{k}_{conv} = \bar{\mathbf{n}} \times \Psi &= \hat{\mathbf{x}} \left[\bar{n}_y \left(\frac{\partial n_x}{\partial y} - \frac{\partial n_y}{\partial x} \right) - \bar{n}_z \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) \right] \\ &\quad + \hat{\mathbf{y}} \left[\bar{n}_z \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) - \bar{n}_x \left(\frac{\partial n_x}{\partial z} - \frac{\partial n_y}{\partial x} \right) \right] \\ &\quad + \hat{\mathbf{z}} \left[\bar{n}_x \left(\frac{\partial n_z}{\partial x} - \frac{\partial n_x}{\partial z} \right) - \bar{n}_y \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) \right] \end{aligned}$$

Geometric Attributes: Program **curvature3d**

Reflector Rotation about the Normal

Reflector convergence is a measure of reflector rotation within the analysis window about the two (often near horizontal) axes that define the average structural dip and measures vertical (more exactly, parallel to \mathbf{n}) changes in dip. In contrast, reflector rotation about the average normal axis measures lateral changes in dip, perpendicular to \mathbf{n} . Such lateral changes occur when there is rotation about a fault, or when the dip of the infill of a graben (or incised channel) differs laterally from that of the neighboring fault block (or flood plain).

Reflector Rotation

Recall that in order to compute reflector convergence, one first computes normal unit vector, \mathbf{n} , from the apparent dip components, p and q , and then computes the curl of \mathbf{n} , $\Psi = \nabla \times \mathbf{n}$. The components of Ψ perpendicular to the average normal, $\bar{\mathbf{n}}$ defined the reflector convergence, or reflector rotations within the analysis window about the two axes parallel to the average reflector orientation. The component of Ψ parallel to the average normal will define the reflector rotation about the normal:

$$k_{rot} = \bar{\mathbf{n}} \cdot \Psi = \bar{n}_x \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) + \bar{n}_y \left(\frac{\partial n_z}{\partial x} - \frac{\partial n_x}{\partial z} \right) + \bar{n}_z \left(\frac{\partial n_x}{\partial y} - \frac{\partial n_y}{\partial x} \right).$$

k_{rot} is associated with structural rotation about faults and laterally variable syntectonic deposition parallel to faults or lateral changes in the dip of channel fill with respect that of the neighboring flood plain. Rotation down to the right will be considered positive (shown as red) and reflector up to the right negative (shown as blue) in the cartoon below.

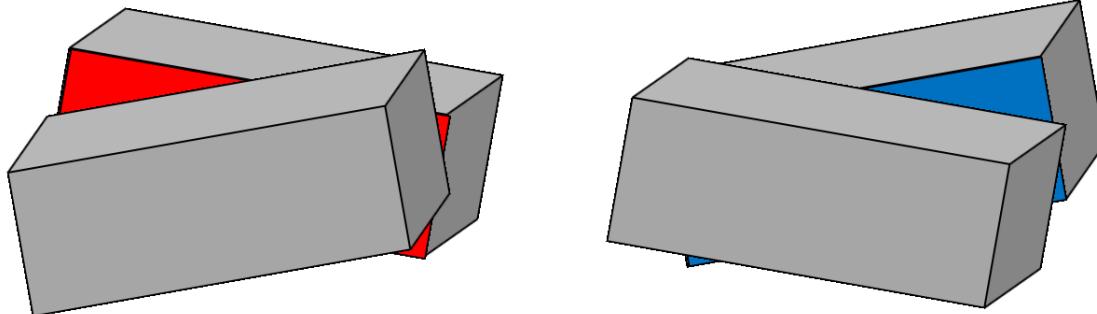


Figure C1.

Geometric Attributes: Program **curvature3d**

The following image shows co-rendered time slices at $t=1.280$ s through the GSB survey most-positive and most-negative curvature volumes plotted against a gray scale color bar.

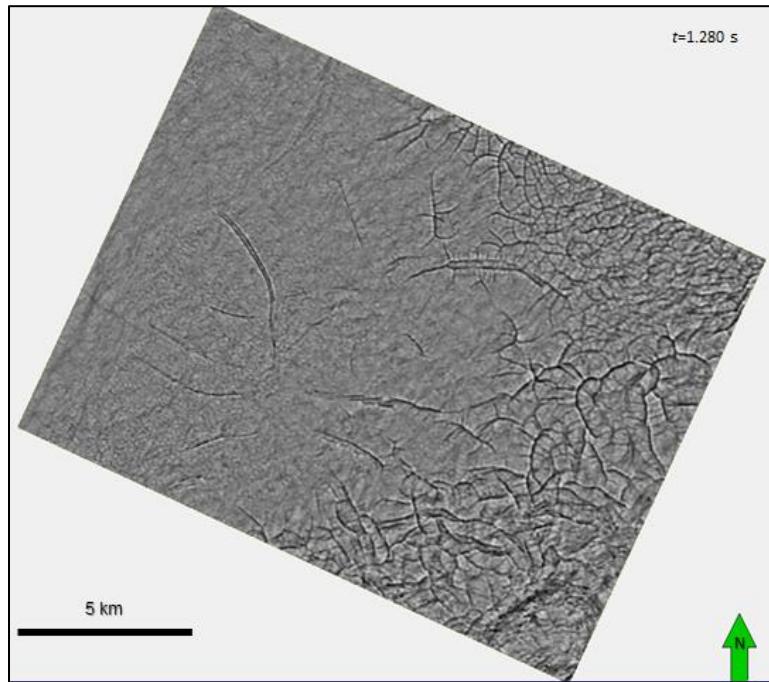


Figure 41.

The next image shows the reflector rotation about the normal, k_{rot} , plotted against a red-white-blue color bar:

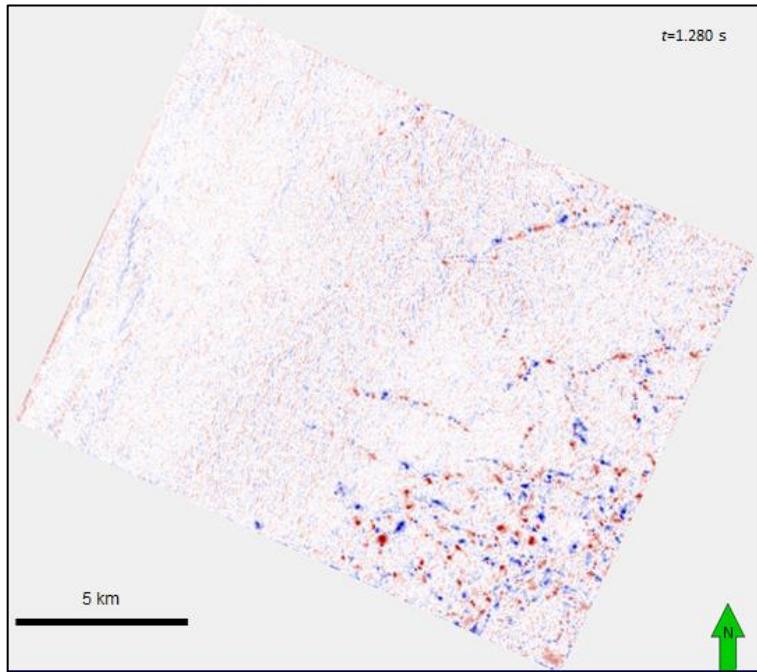


Figure 42.

Geometric Attributes: Program **curvature3d**

Finally, co-rendering the two previous images gives:

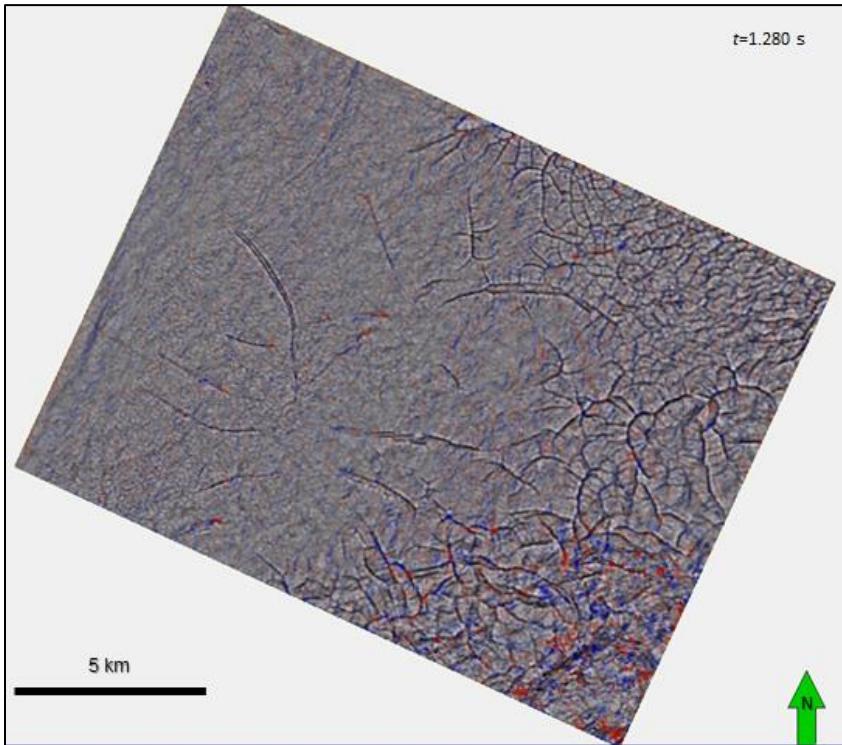


Figure 43.

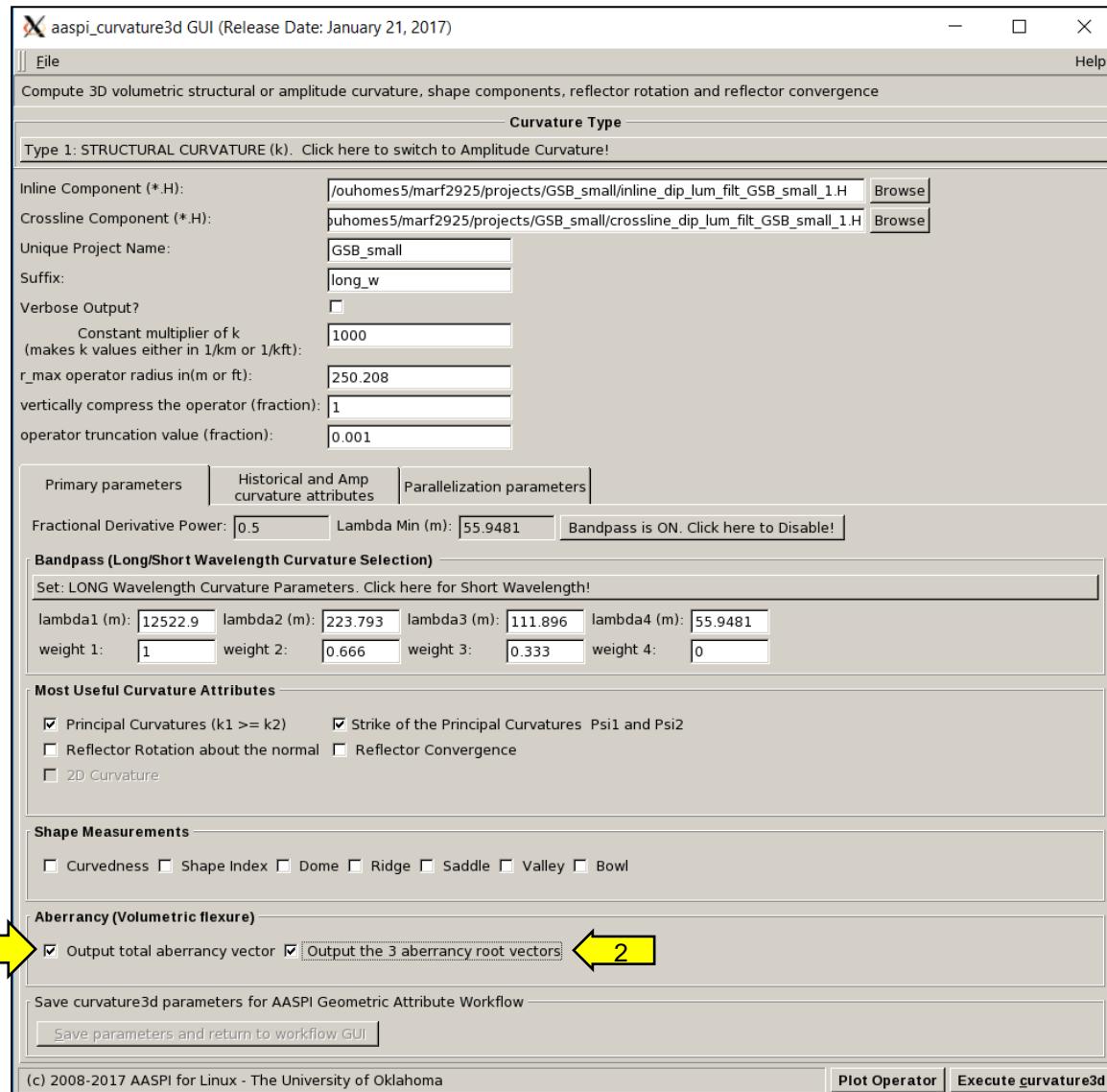
In this image, the red and blue areas appear strongest near the intersections of the normal faults, suggesting that there may be significant shearing favorable to fracture generation.

Aberrancy

While vector dip is the first derivative of structure and the principal curvatures are second derivatives of structure, aberrancy, often called flexure, is the third derivative of structure. Aberrancy was first added to the AASPI software in August, 2016. Further work on developing stable and accurate measures of the azimuth of aberrancy was completed in January 2017. While there is only principal vector dip (and many apparent dips), two principal curvatures, k_1 and k_2 , (and many apparent or Euler curvatures), there are three principal aberrancies (and many apparent aberrancies or flexures at arbitrary azimuth). Unlike k_1 and k_2 , which are signed values associated with a strike, the aberrancies are represented by vectors with azimuths ranging from -180° to $+180^\circ$ and magnitudes (or “values”) strictly positive. Sorting the three roots into maximum, intermediate, and minimum aberrancy attribute volumes by their magnitude is problematic, in that the geologic feature (say a small fault) exhibiting maximum aberrancy value in one location may be overwhelmed by a stronger fault cutting across it at a different location, with the resulting maximum aberrancy image showing two disjoint faults, with the fault of smaller magnitude being mapped by the intermediate aberrancy. For this reason, we provide a simple vector sum of the three aberrancy vectors, denoted as the total

Geometric Attributes: Program **curvature3d**

aberrancy. Further research needs to be conducted to determine whether anomalously large values in the two smaller aberrancy roots provide value in highlighting areas of strong shearing that may be an indicator of natural fractures. Returning to the curvature GUI:



Select (1) *Output total aberrancy vector*, and if you are interested, (2) *Output the 3 aberrancy root vectors*, and click *Execute*. The 27 second derivatives needed by aberrancy are computed by taking first derivatives of the nine first derivatives used in computing curvature, reflector convergence, and reflector rotation. Thus, aberrancy runs about four times longer than curvature by itself. For this same reason, if computing aberrancy, outputting the (intermediate) curvature volumes requires further disc space, but only minimal additional compute time.

Geometric Attributes: Program **curvature3d**

Aberrancy and the 2nd derivatives of the dip vector, **p**

Since it is a third derivative of structure, aberrancy is not only more mathematically complicated than curvature, but also significantly more difficult to program. Unlike curvature, mathematically sophisticated but computationally simple eigenvector and eigenvalue analysis has not yet been found, requiring a search for the three extrema of the apparent aberrancy (apparent flexure for a given azimuth) within a rotated coordinate system. The breakthrough came through by Di and Gao (2016) whose work in horizon-based aberrancy greatly simplified the search for extrema by first rotating the coordinate system to be aligned with local reflector dip and azimuth.

While showing a promising solution, volumetric aberrancy based on Di and Gao's (2013) work computationally intractable, requiring different rotations of large data windows about each voxel prior to computing the derivatives. For this reason, we first (efficiently) compute all the derivatives in the original (unrotated) coordinate system as done for volumetric curvature. Recall that the un-normalized dip vector **p** has elements,

$$p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} p \\ q \\ 1 \end{pmatrix}$$

representing an azimuth $\varphi = \text{ATAN2}(p_1, p_2)$, and dip $\theta = \text{ATAN}\left[\left(p_1^2 + p_2^2\right)^{1/2}\right]$.

Rotation of this vector by the rotation matrix, **R**, gives

$$p'_k = \sum_{n=1}^3 R_{kn} p_n,$$

where

$$\mathbf{R} = \begin{bmatrix} \cos \varphi \cos \theta & -\sin \varphi \cos \theta & \sin \theta \\ \sin \varphi & \cos \varphi & 0 \\ -\cos \varphi \sin \theta & \sin \varphi \sin \theta & \cos \theta \end{bmatrix}.$$

Similarly, one needs to rotate the derivative operators

$$\frac{\partial}{\partial x'_i} = \sum_{m=1}^3 R_{im} \frac{\partial}{\partial x_m},$$

and explicitly repeating a second first derivative:

$$\frac{\partial}{\partial x'_j} = \sum_{l=1}^3 R_{il} \frac{\partial}{\partial x_l}.$$

Combining these terms gives

$$\frac{\partial^2 p'_k}{\partial x'_i \partial x'_j} = \sum_{l=1}^3 \sum_{m=1}^3 \sum_{n=1}^3 R_{im} R_{jl} R_{kn} \frac{\partial^2 p_n}{\partial x_m \partial x_l}.$$

In this manner, one can efficiently compute the 3x3x3 2nd derivative matrix $\frac{\partial^2 p_n}{\partial x_m \partial x_l}$ by convolution in the

unrotated world coordinate system and obtain the corresponding rotated operator $\frac{\partial^2 p'_k}{\partial x'_i \partial x'_j}$ by a suite of three 3D rotations.

Geometric Attributes: Program **curvature3d**

Such unit normals fall out automatically from dip estimation techniques based on the Gradient Structure Tensor. To measure rotation, we compute ψ , the curl of the unit normal, \mathbf{n} . For interpretation purposes, we break this 3-component curl into two parts – the part that rotates about the average unit normal, $k_{rot} = \bar{\mathbf{n}} \cdot \psi$, which we will associate with syntectonic deposition and structural rotation about faults and $\mathbf{k}_{conv} = \bar{\mathbf{n}} \times \psi$, which is a vector that we will associate with angular unconformities, downlap, toplap, and other stratigraphic phenomena:

$$\begin{aligned}\mathbf{k}_{conv} = \bar{\mathbf{n}} \times \psi &= \hat{\mathbf{x}} \left[\bar{n}_y \left(\frac{\partial n_x}{\partial y} - \frac{\partial n_y}{\partial x} \right) - \bar{n}_z \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) \right] \\ &\quad + \hat{\mathbf{y}} \left[\bar{n}_z \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) - \bar{n}_x \left(\frac{\partial n_x}{\partial y} - \frac{\partial n_y}{\partial x} \right) \right] \\ &\quad + \hat{\mathbf{z}} \left[\bar{n}_x \left(\frac{\partial n_z}{\partial x} - \frac{\partial n_x}{\partial z} \right) - \bar{n}_y \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) \right].\end{aligned}$$

Aberrancy computation

Once the 2nd derivatives of the dip vector \mathbf{p}' have been computed in the rotated (primed) coordinate system, Di and Gao (2016) find the apparent flexure, $f(\psi)$, at azimuth ψ to be:

$$f(\psi) = \frac{\partial^3 z'}{\partial x'^3} \cos^3 \psi + 3 \frac{\partial^3 z'}{\partial x'^2 \partial y'} \cos^2 \psi \sin \psi + 3 \frac{\partial^3 z'}{\partial x' \partial y'^2} \cos \psi \sin^2 \psi + \frac{\partial^3 z'}{\partial y'^3} \sin^3 \psi.$$

The extrema are found by determining the roots of

$$\begin{aligned}\frac{df(\psi)}{d\psi} &= \\ 3 \cos^3 \phi_1 \cdot [-\frac{\partial^3 z'}{\partial x' \partial y'^2} \tan^3 \psi - (2 \frac{\partial^3 z'}{\partial x'^2 \partial y'} - \frac{\partial^3 z'}{\partial y'^3}) \tan^2 \psi + (2 \frac{\partial^3 z'}{\partial x' \partial y'^2} - \frac{\partial^3 z'}{\partial x'^3}) \tan \psi + \frac{\partial^3 z'}{\partial x'^2 \partial y'}] &= 0\end{aligned}$$

which is a cubic equation in $\tan \psi$. Once found, the three aberrancy magnitudes are simply $f(\psi_1)$, $f(\psi_2)$, and $f(\psi_3)$, using the equation above. The value of $\tan \psi$ ranges between -90° and $+90^\circ$. If the extreme value $f(\psi_i)$ is less than 0, one flips the direction of ψ_i by 180° , thereby obtaining an azimuth of the flexure and sets the value of f to be a non-negative magnitude of the flexure. The final step is to unrotate the azimuth of the aberrancy vector back to the world coordinate system.

Corendering the aberrancy vector azimuth and magnitude at the same time slices shown in previous images through complex faulting, syneresis and the shelf edge produces the following images:

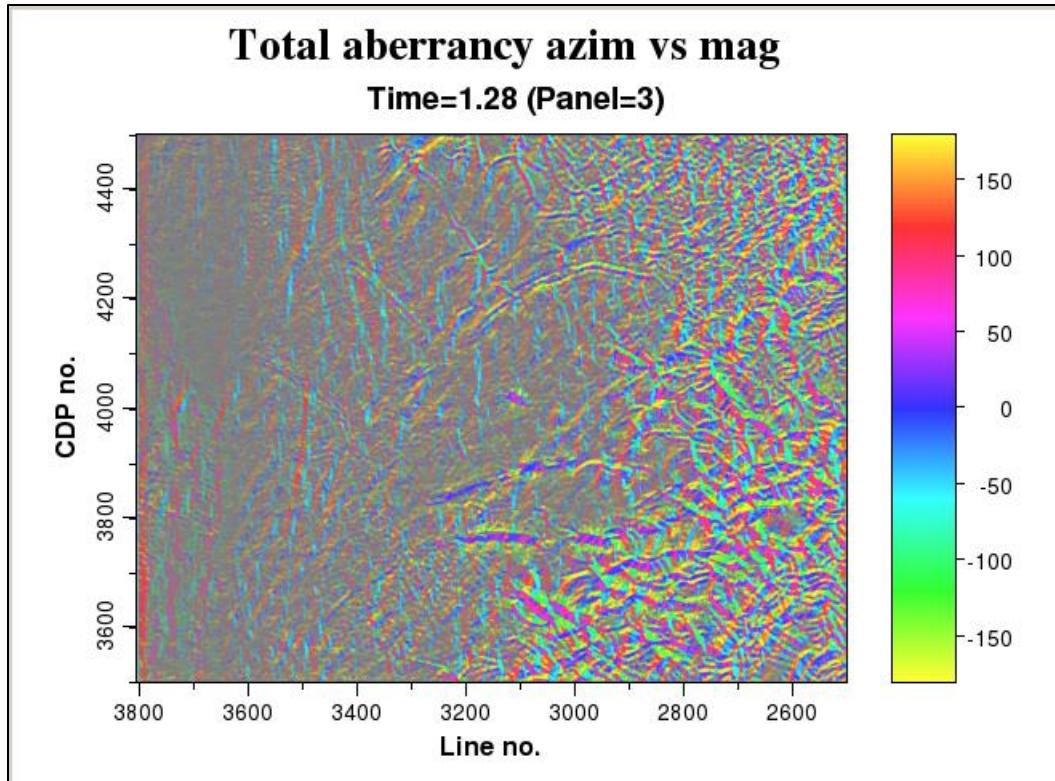


Figure 44.

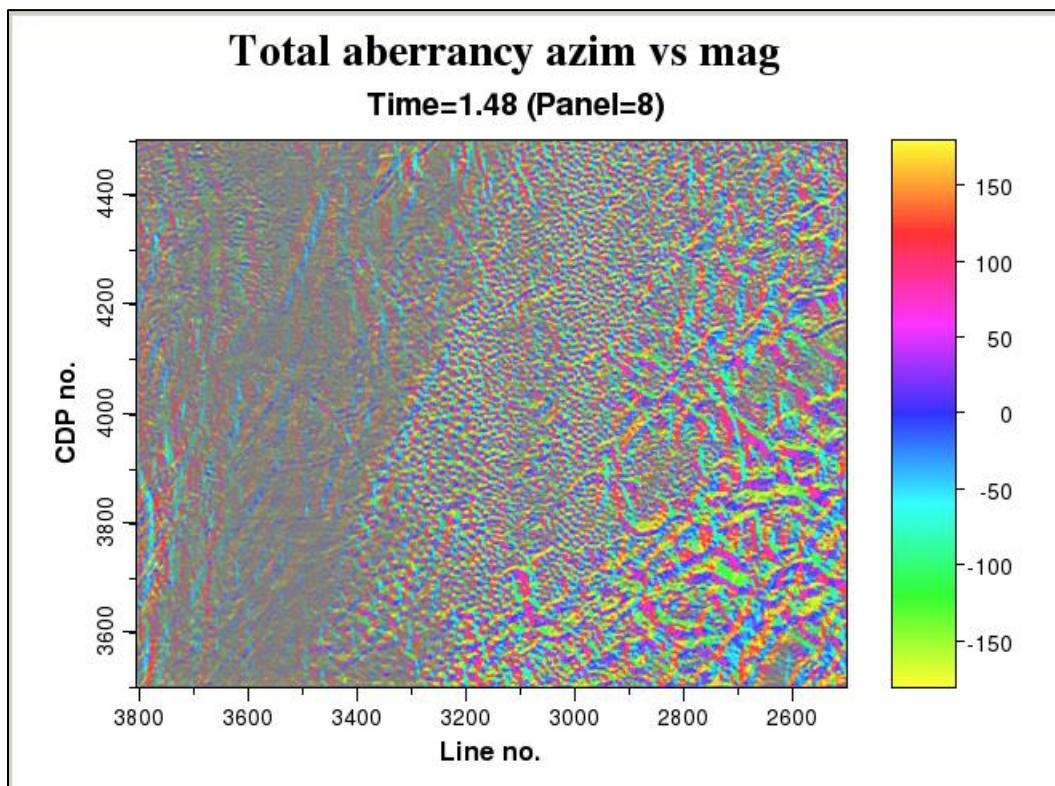


Figure 45.

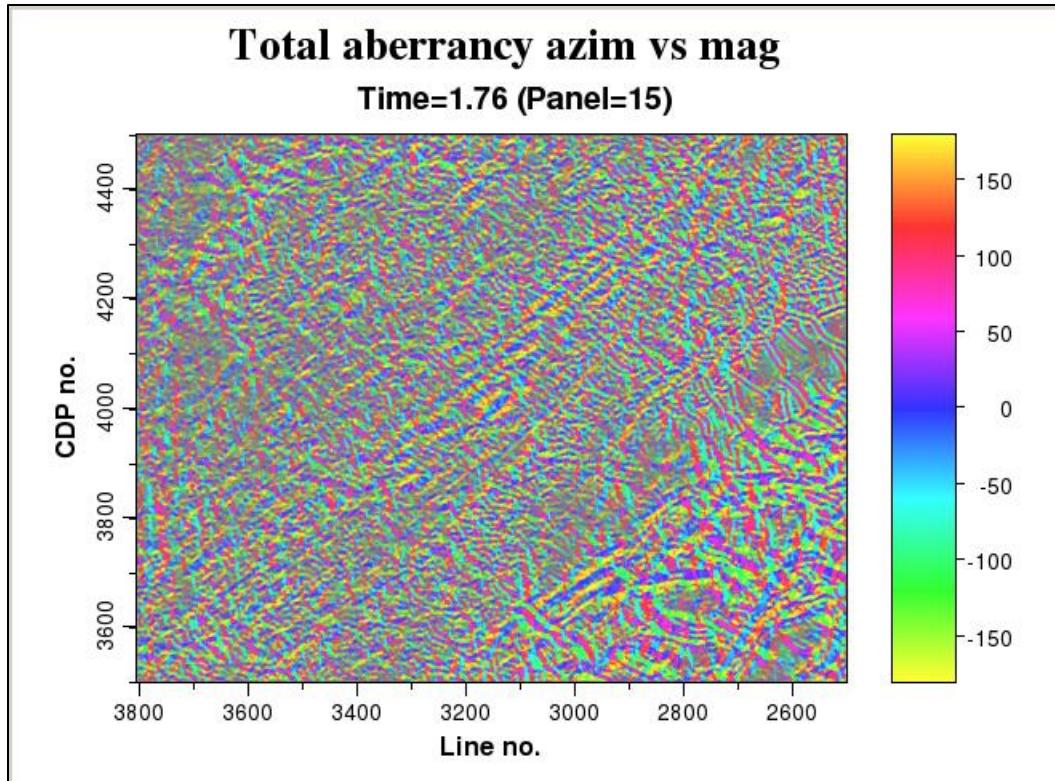


Figure 46.

Note the progradation features appearing as yellow-blue cutting from the center to the upper right of this image.

Since aberrancy is complementary to coherence in mapping faults, let us corender it as well (in this case multispectral coherence scaled to range between 0.8 and 1.0):

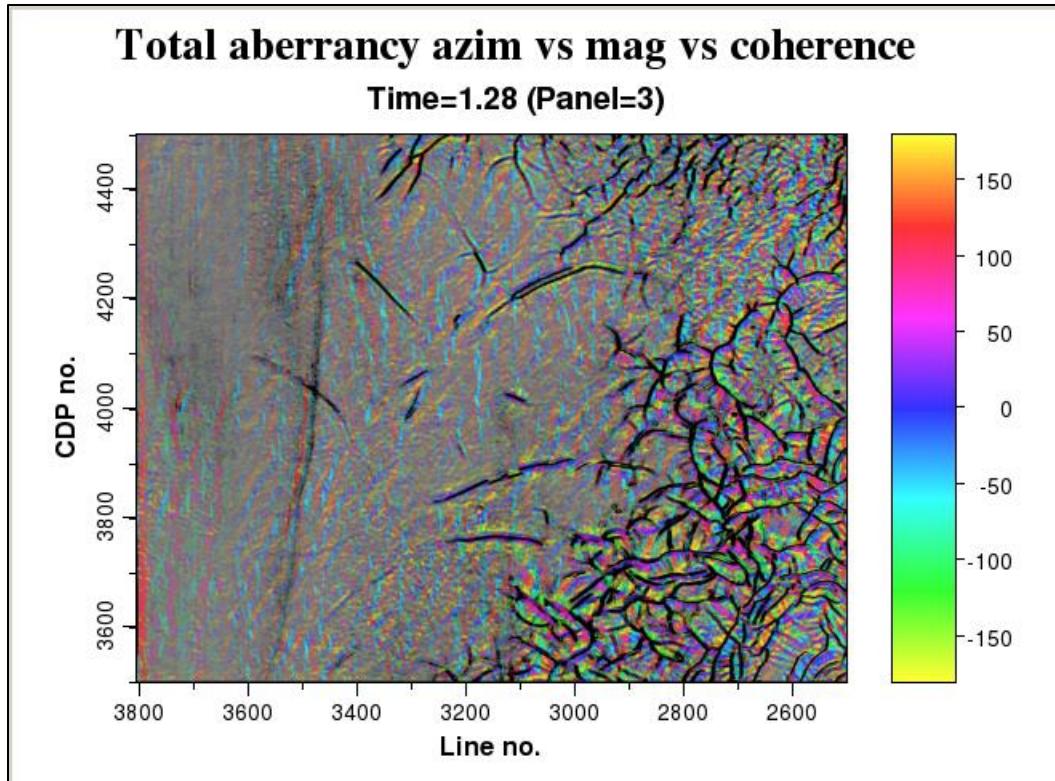


Figure 47.

In most areas, coherence overlays the faults. However, note in the NE corner of the image that more faults (with minimal offset) are imaged by aberrancy. There is also the suggestion of additional faulting (or more specifically, flexing) within the larger fault blocks towards the SE corner of the survey.

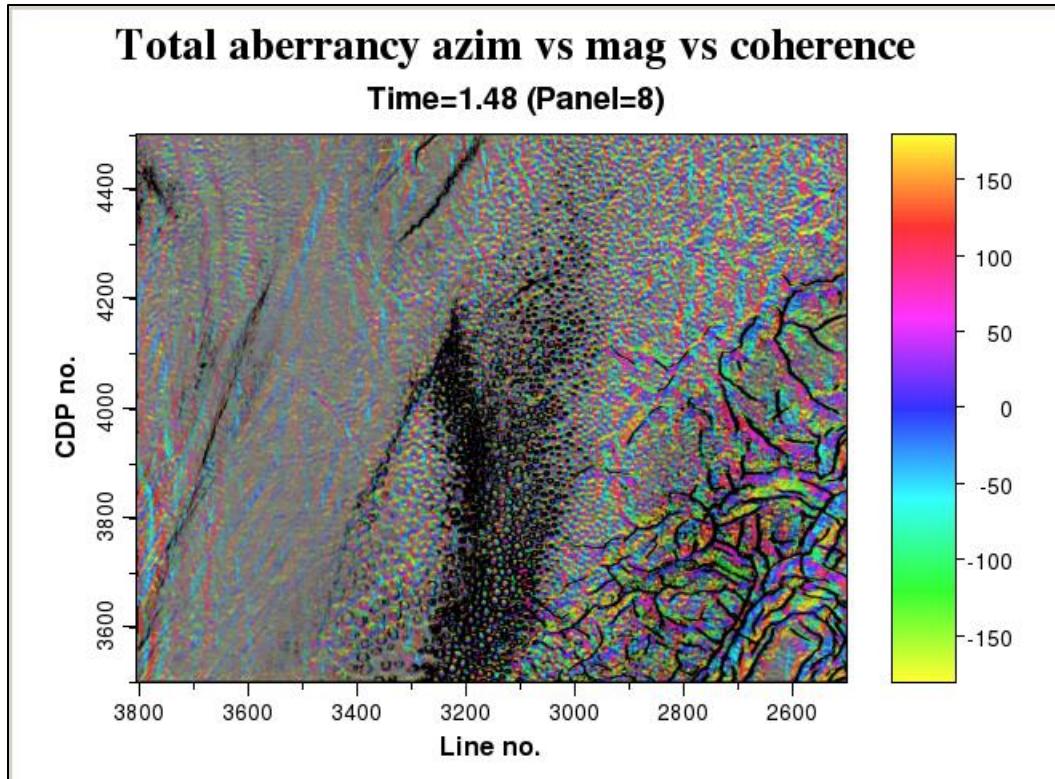


Figure 48.

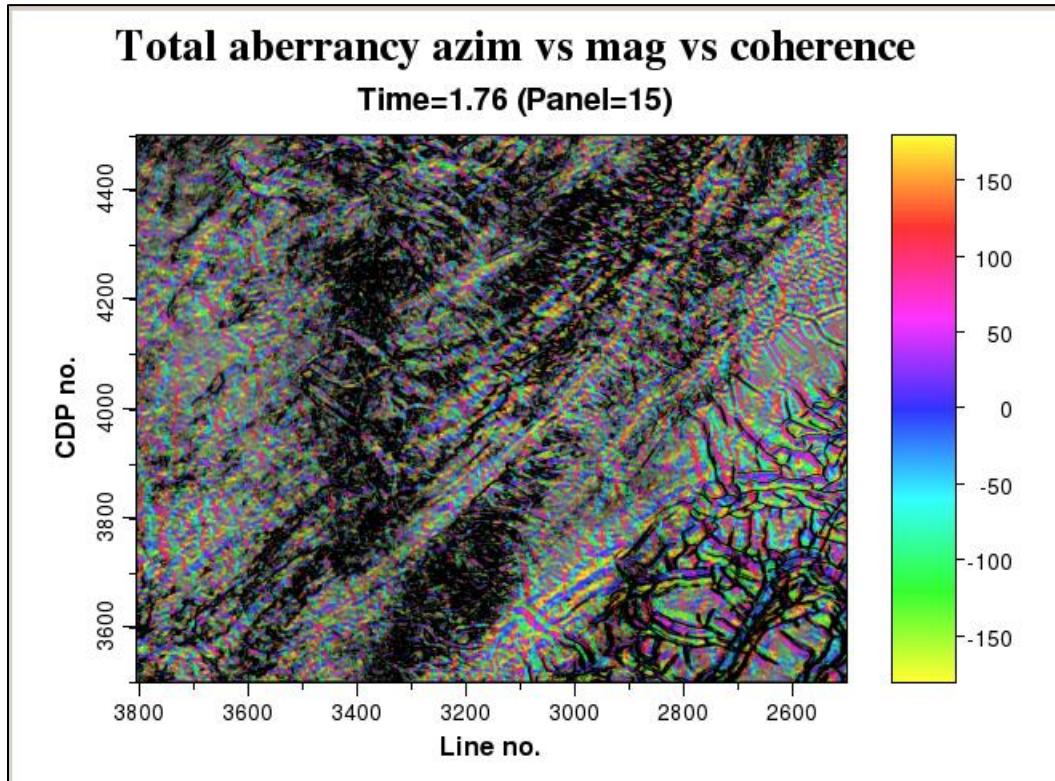


Figure 49.

Geometric Attributes: Program **curvature3d**

More Examples of Curvature

While the GSB data volume is useful for testing, other examples using licensed data better demonstrate the value of this family of attributes. Most of these examples come from either publications or from Marfurt's SEG not-so-short course found under www.mcee.ou.edu/aaspi/upload.

Examples showing most-positive and most-negative principal curvatures

Many workers associate curvature with faults. While this may sometimes be the case, they are usually juxtaposed about a fault, or simply measure folding. An example of the later features is shown in images presented by Mai et al. (2009) over the Chicontepec Basin of Mexico for the most-positive curvature, k_1 :

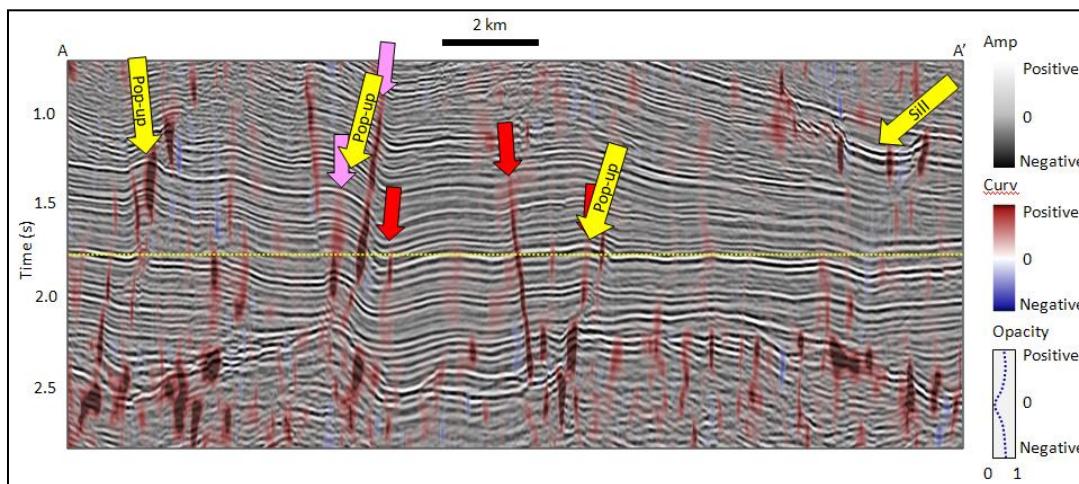


Figure 50.

and the most-negative principal curvature, k_2 :

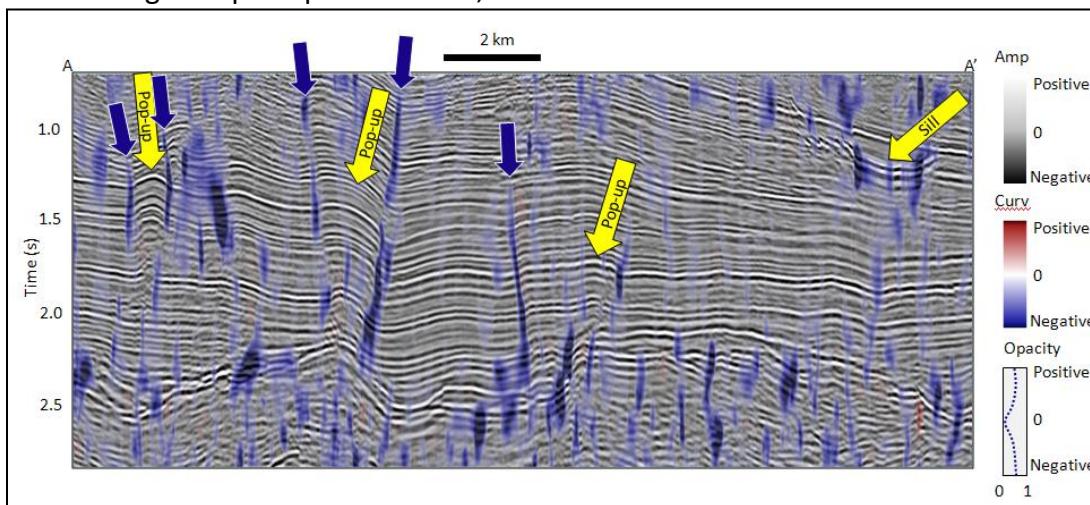


Figure 51.

Geometric Attributes: Program **curvature3d**

To display a common spatial correlation of coherence and most-positive and most-negative principal curvatures, Chopra and Marfurt (2007) co-rendered the three attributes to produce the following image. The curvature anomalies in this image do not show ‘faults’ although they may show zones of conjugate faults giving rise to a curved surface at resolution of surface seismic data.

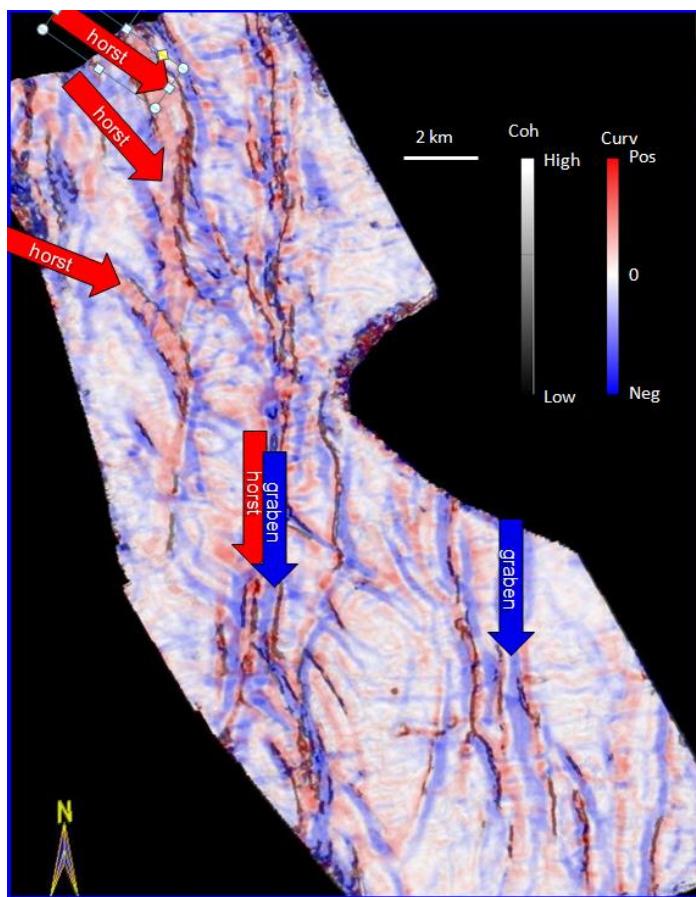


Figure 52.

Examples showing Curvedness and Shape Index and Shape Components

The two principal curvatures, k_1 and k_2 , define five quadratic shapes, and a degenerate planar ‘shape’ when $k_1 = k_2 = 0$. Typically, we will plot these shapes using the end members of a 2D color bar (modified by Mai et al., 2009, from an earlier paper by Bergbauer et al., 2003):

Note the definition of *curvedness* in the figure above. The *shape index* is a simple function of the two principal curvatures, k_1 and k_2 :

Geometric Attributes: Program **curvature3d**

Using the 2D color bar modulating the shape index by the curvedness using program **hplot**, we show the same Chicontepec, Mexico image displayed earlier:

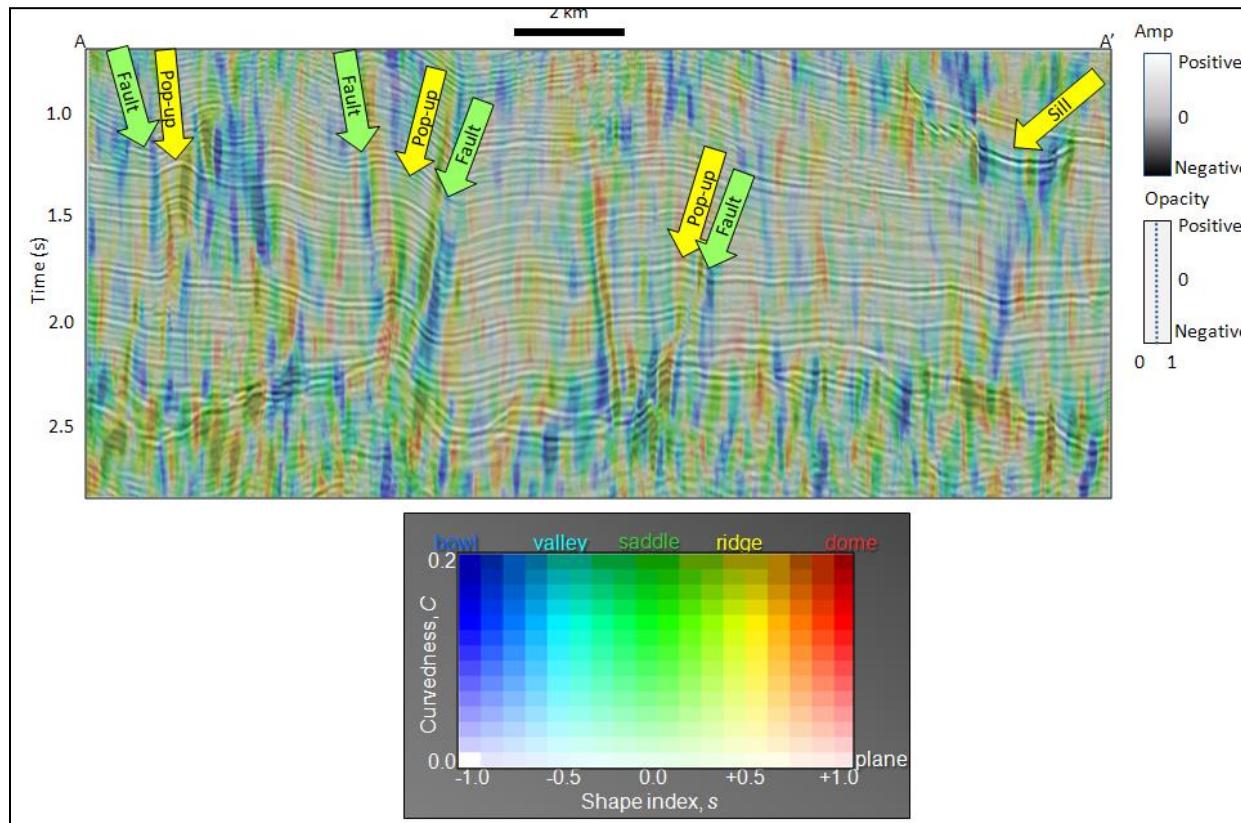


Figure 53.

Marfurt (2010) imported the result of a data volume courtesy of Devon Energy and co-render it using Petrel with the seismic amplitude on the vertical slice and energy ratio coherence on the time slice where the blue corresponds to collapse features (yellow arrow) seen in the Ellenburger Limestone underlying the Barnett Shale. Note the valleys (in cyan) and ridges (in yellow) bracket the strike slip fault (magenta arrow) and normal fault (green arrow).

Geometric Attributes: Program **curvature3d**

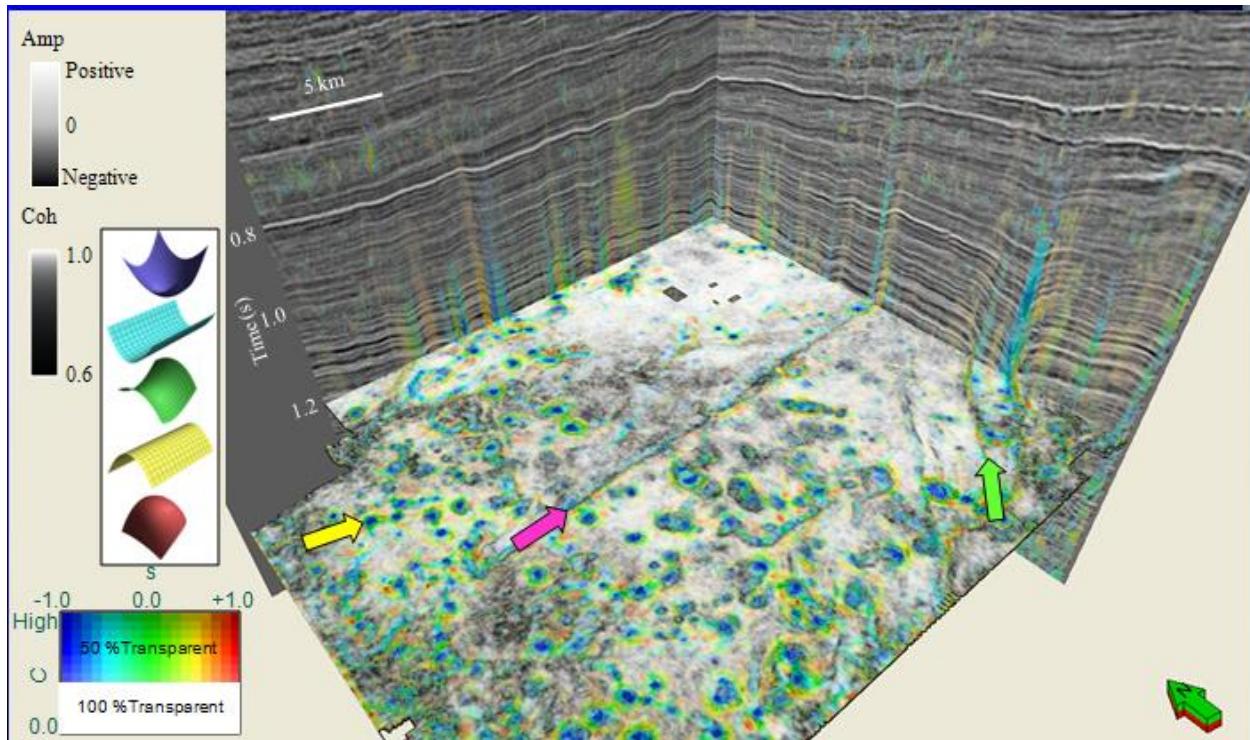


Figure 54.

Internal to program **curvature3d**, one can apply filters to these two attributes to generate separate shape ‘components’, such as the filters used to ‘extract’ bowl shapes shown below:

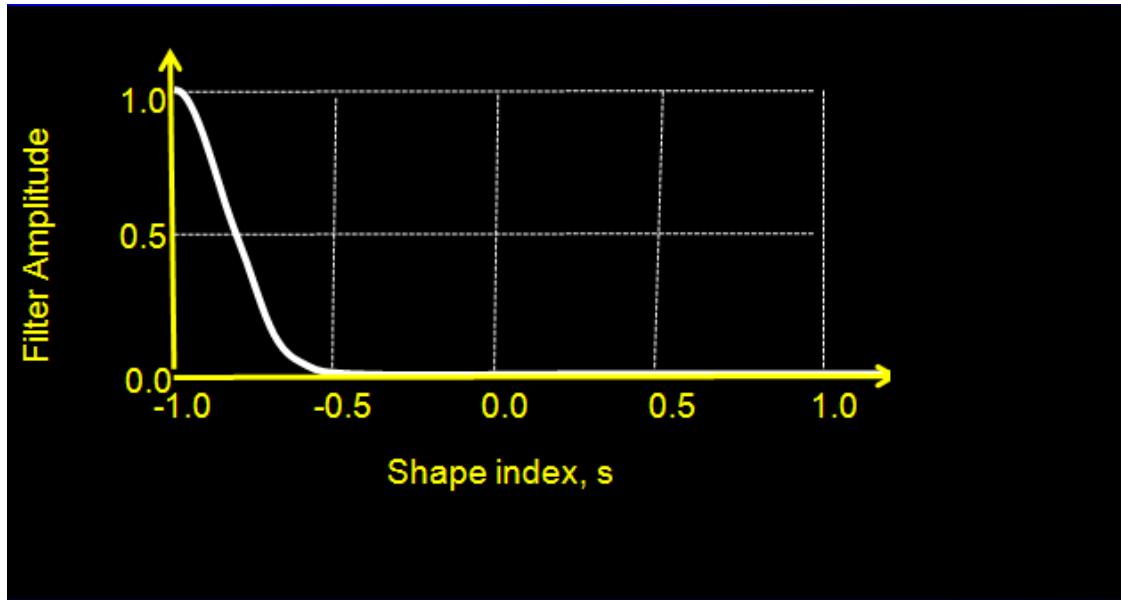


Figure 55.

Geometric Attributes: Program **curvature3d**

The result from Marfurt (2010), is the following image corresponding to the vertical and time slices shown above where he has used a monochrome white to blue color bar to plot the intensity of the bowl component (again, note the correlation with the collapse features):

Using the opacity settings in the figure above, we can turn on volume rendering and display the ‘collapse chimneys’ in 3D:

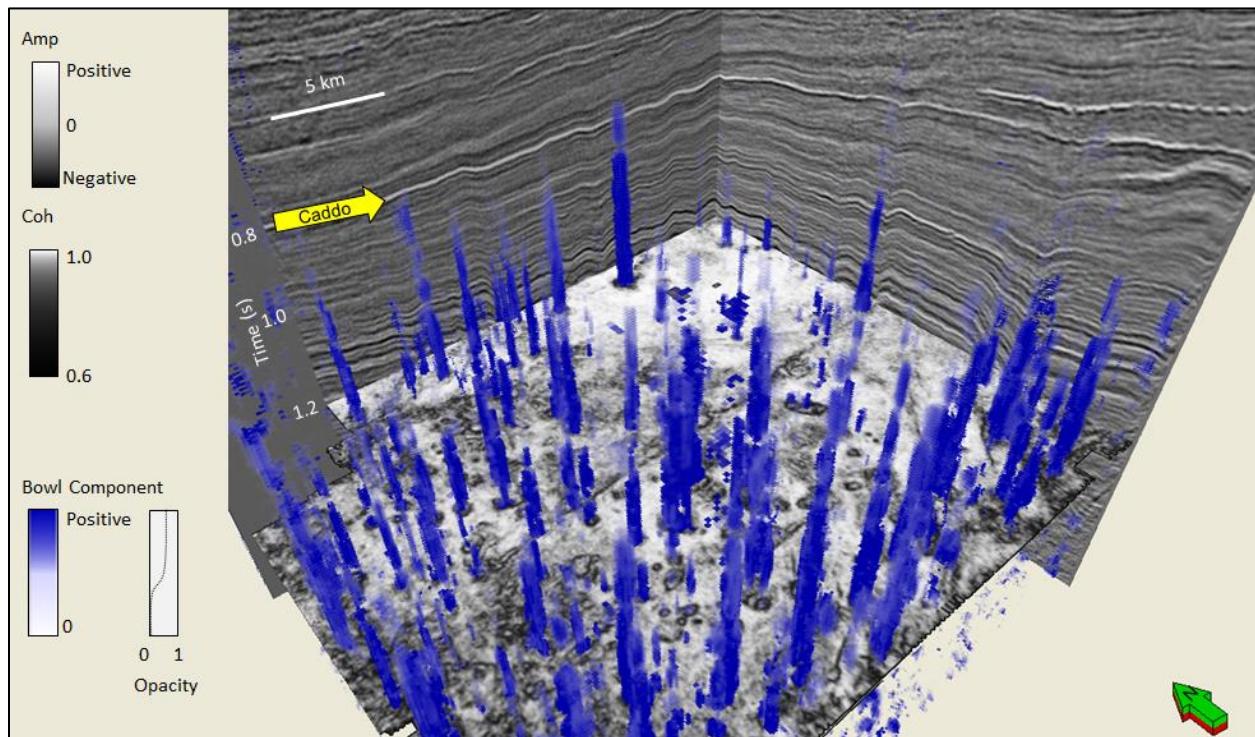


Figure 56.

Note the collapse features seen in the coherence time slice at the Ellenburger level continue as bowl-shape features up into the Atoka formation, which lies just below the strong Pennsylvanian-age Caddo horizon in this image. These bowls form geohazards that need to be avoided in hydraulic fracturing of the Barnett Shale, but form ‘sweet spots’ having greater accommodation space for deposition of sands and gravels during Atoka time. (Data courtesy of Devon Energy).

Chopra and Marfurt (2011) use a similar workflow and display technique to show the correlation of shapes to energy ratio coherence for a survey acquired in the Western Sedimentary Basin of Canada:

Geometric Attributes: Program **curvature3d**

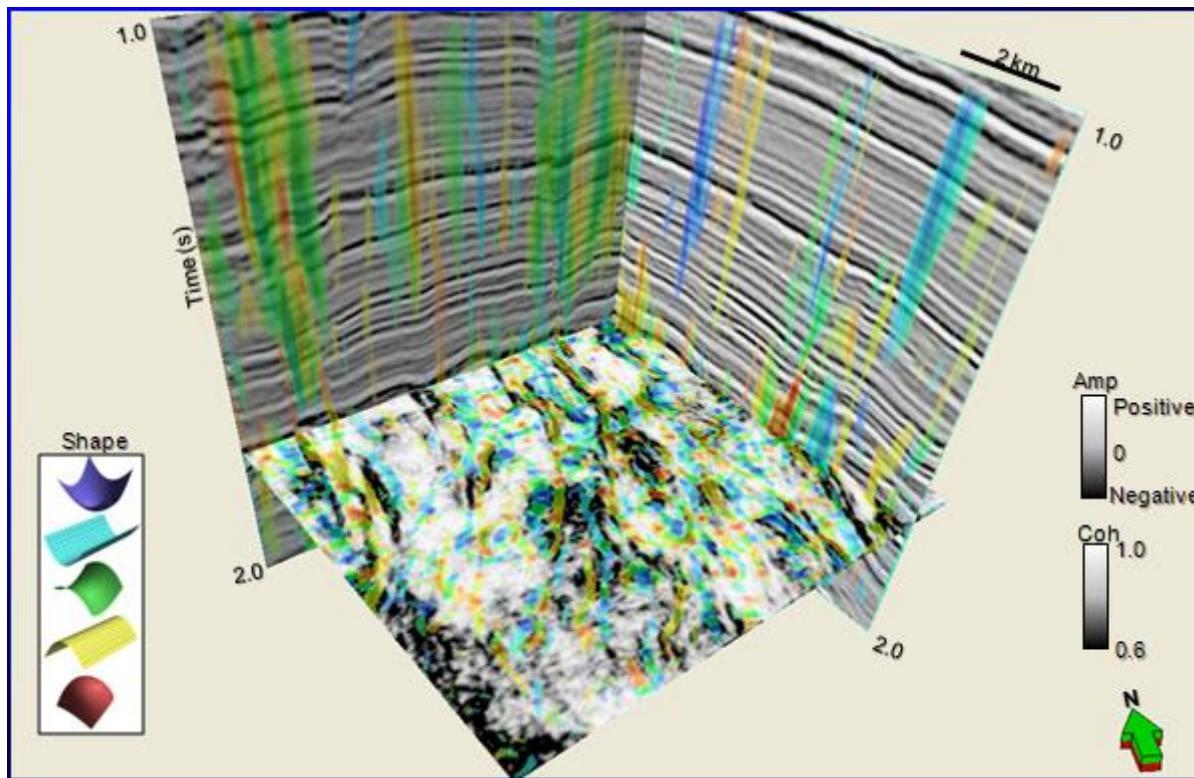


Figure 57.

In this next image, Roderick Perez picked the top of the carbonate reef system in the Horseshoe Atoll area of west Texas and co-rendered a time slice through the shape index to validate the correlation:

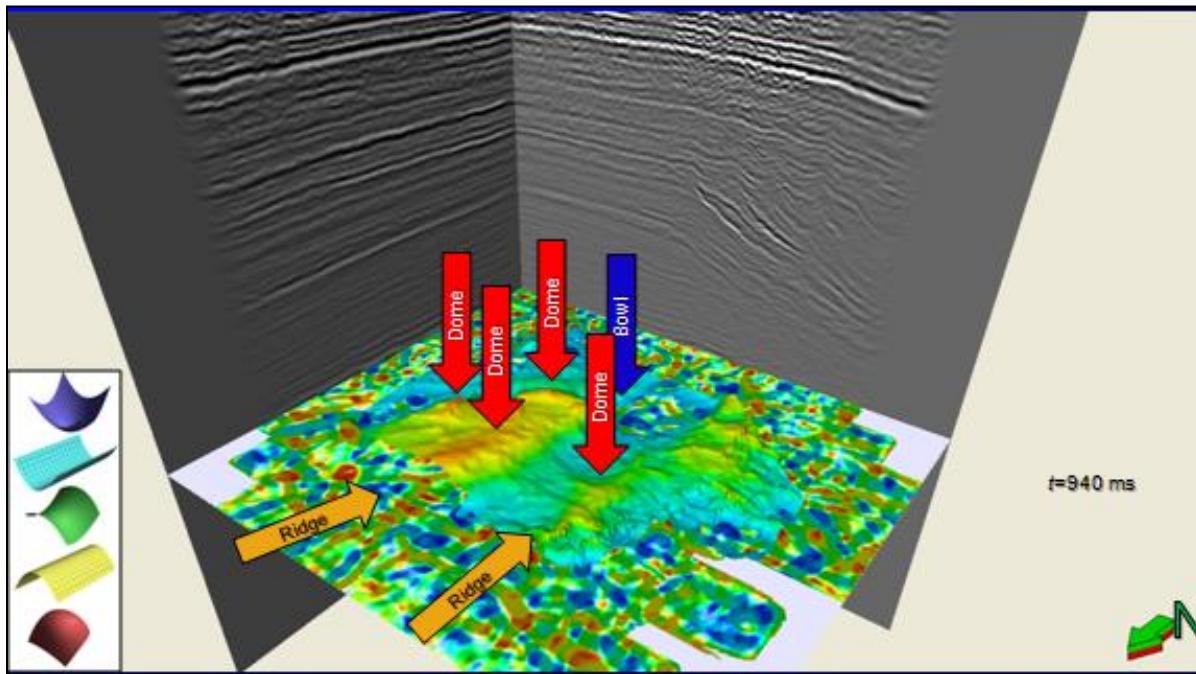


Figure 58.

Geometric Attributes: Program **curvature3d**

Next, we compute the dome components and use Petrel's box probe (their implementation of the geoprobe concept) to highlight some of the pinnacle reefs:

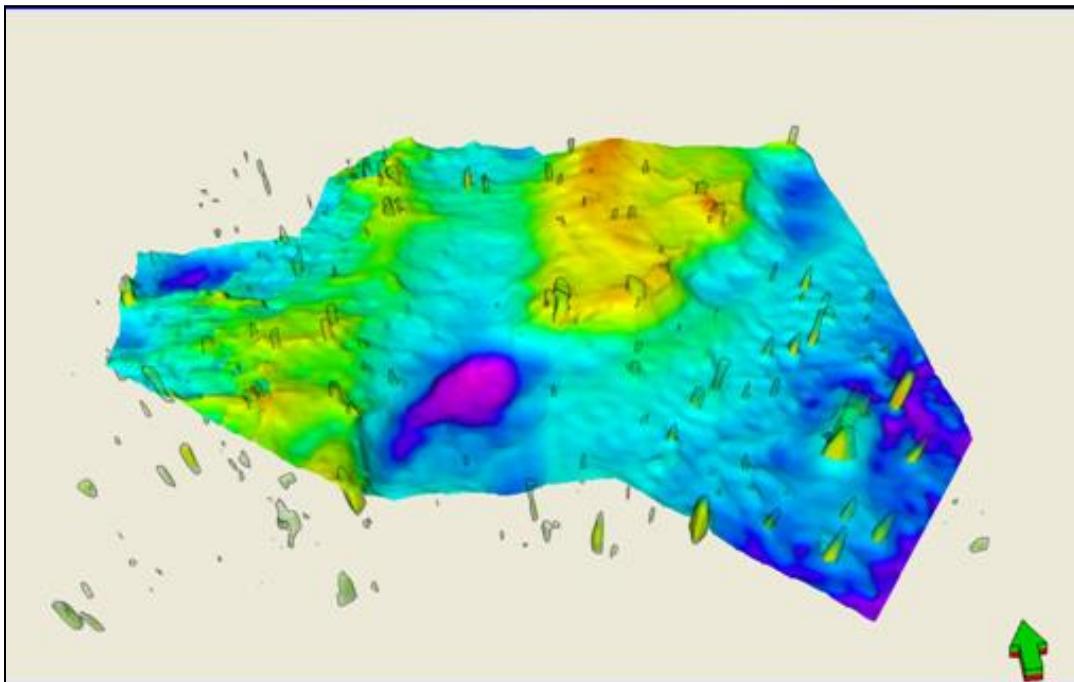


Figure 59.

Since the pinnacle reefs have coalesced, we use Petrel's attribute calculator to simply add both dome and ridge shapes in the following figure:

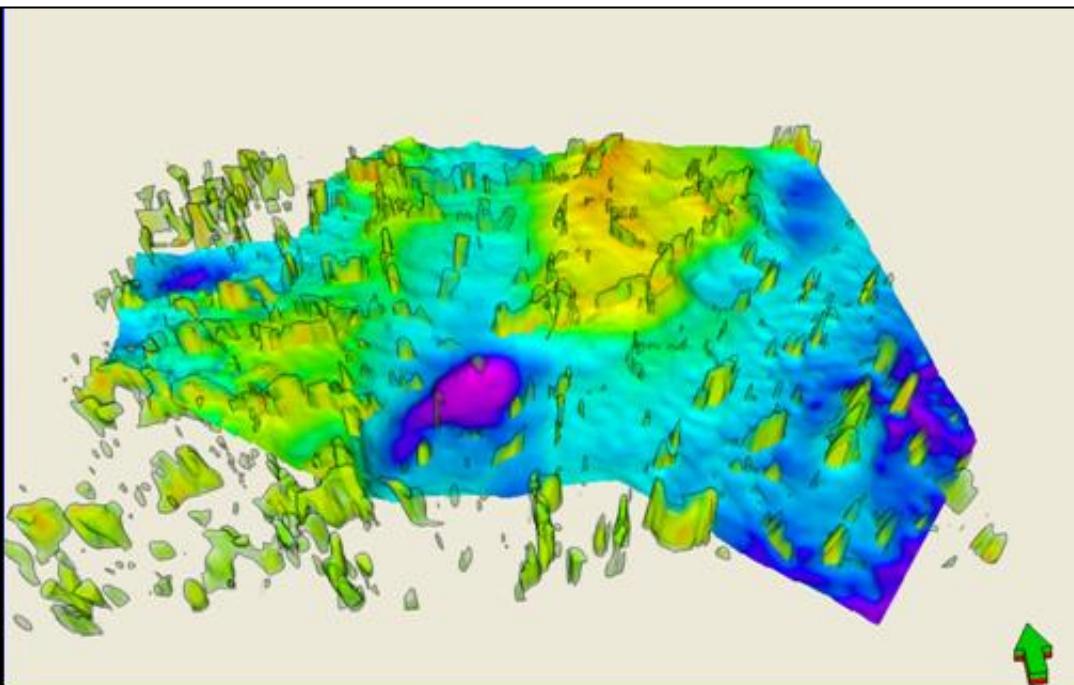


Figure 60.

Geometric Attributes: Program **curvature3d**

Finally, one can use the concept of geobodies to quantify the volumes of the previously delineated shapes:

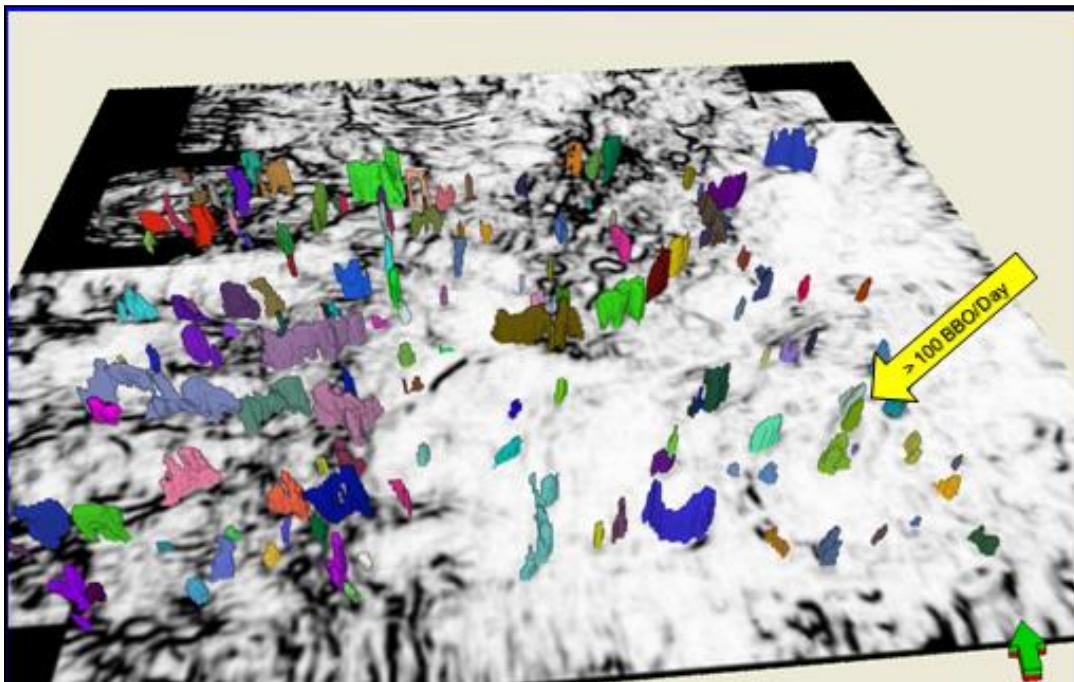


Figure 61.

The yellow arrow indicates one of the geobodies that was successfully drilled in the Summer of 2010.

Examples of Reflector Convergence

In this image from Marfurt and Rich (2010) analyzing features seen near the Atoka unconformity of the Central Basin Platform of west Texas, reflector convergence is plotted as a vector using a 2D color bar, where the hue indicates the direction of convergence (pinchouts or bed thinning) and the lightness, the magnitude of convergence, with parallel layers appearing as white. When co-rendering using transparency in Petrel with the seismic amplitude on the vertical slice and with coherence on the time slice, note how the pinch out onto the erosional unconformity appears as purple to the NE and green to the SW.

Geometric Attributes: Program **curvature3d**

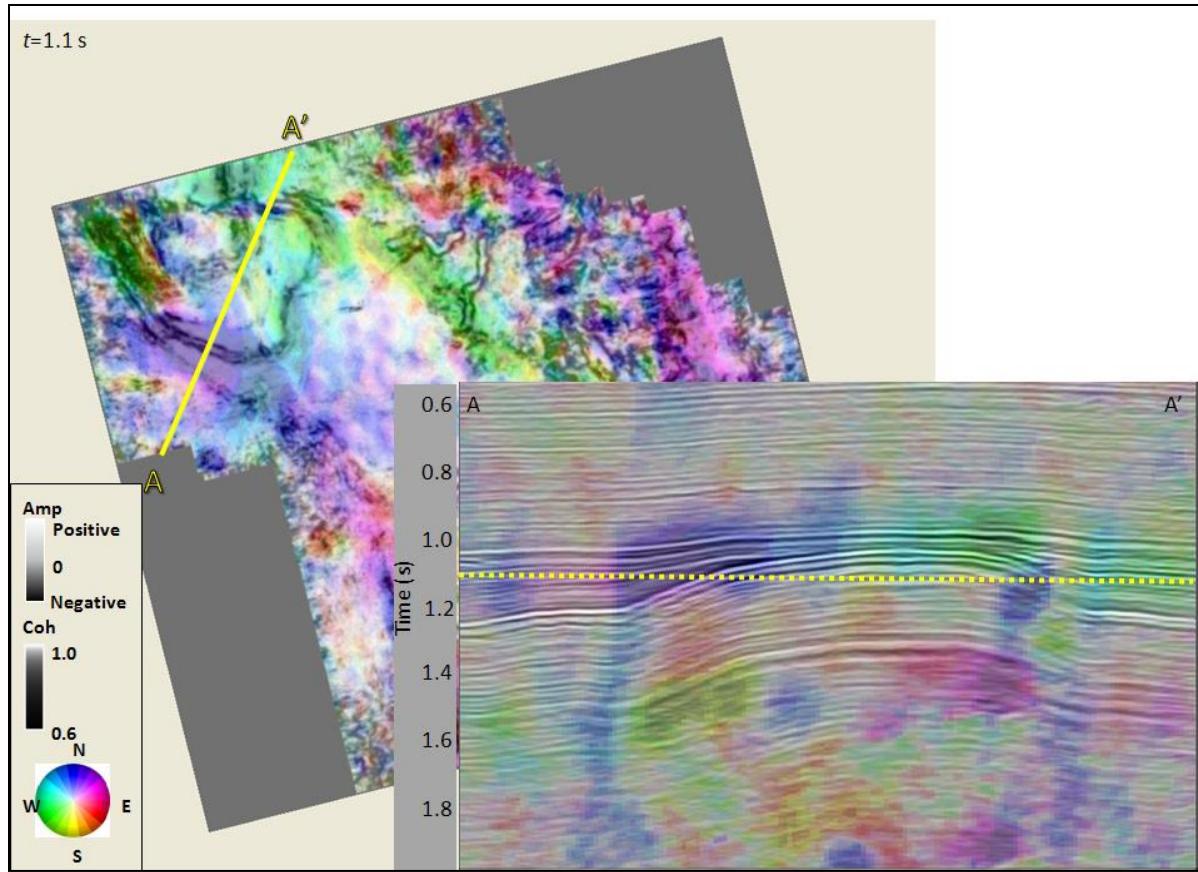


Figure 62.

We draw a second example from Gupta et al. (2011) analyzing unconformable deposition of the Woodford Shale onto the underlying eroded Hunton Limestone of north-central Oklahoma:

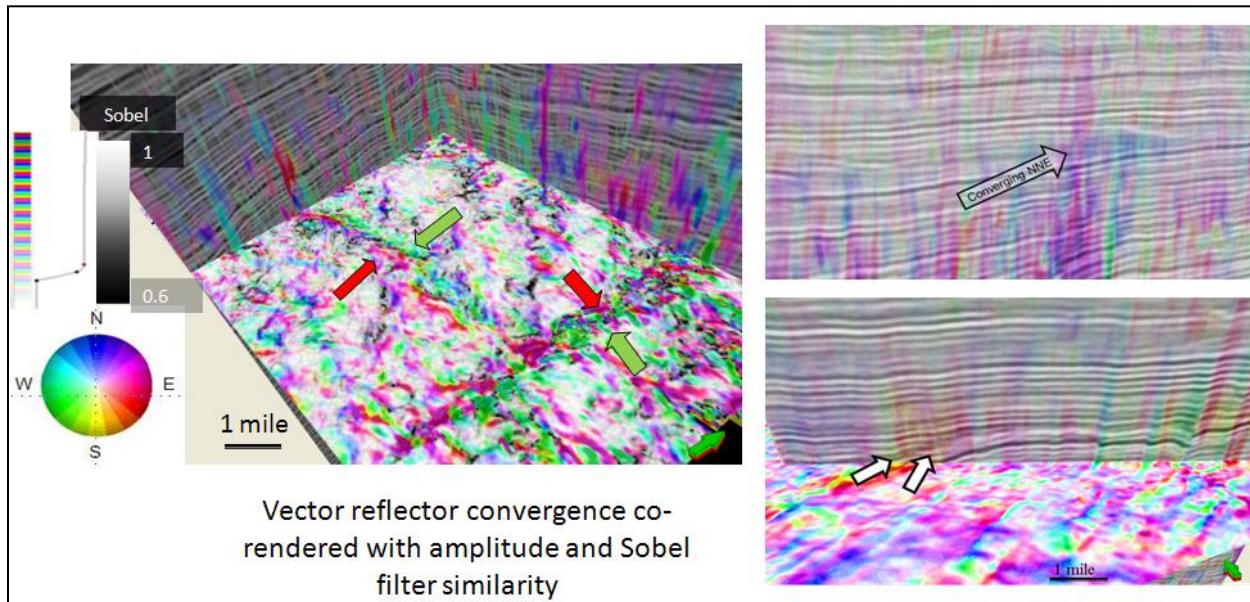


Figure 63.

Geometric Attributes: Program **curvature3d**

The next example comes from Chopra and Marfurt (2013) and shows the convergence within and above an incised channel in Canada. The first pair of figures show time slices at $t=1.700$ s through the incision. The top figure shows coherence, while the bottom shows coherence co-rendered with reflector convergence. Note the reflectors converge *towards* the channel cut (red on the left and green on the right).

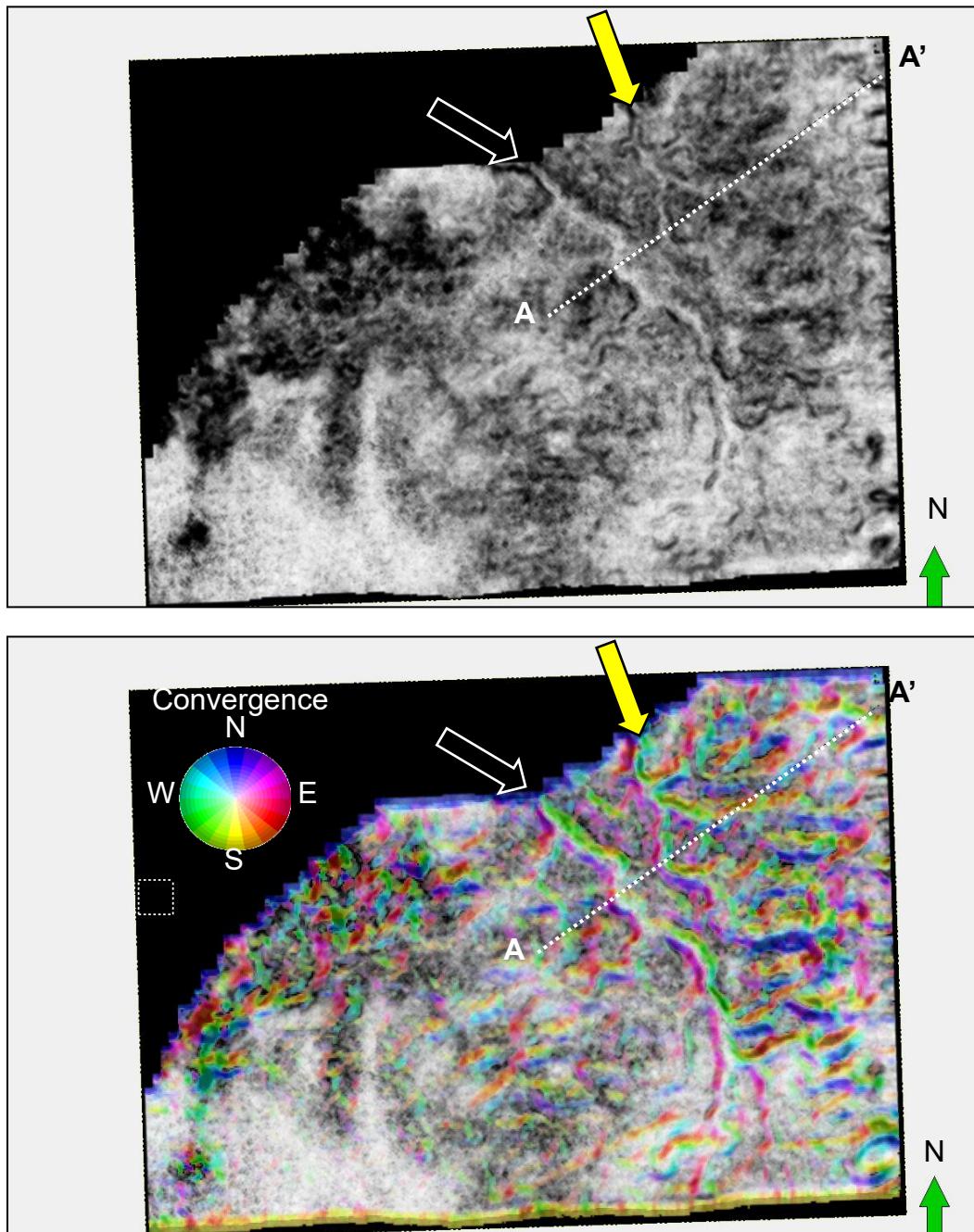


Figure 64.

Geometric Attributes: Program **curvature3d**

The next two images show a shallower time slice above the incisement at $t=1.670$ s. In the coherence image, there is no indication of the deeper channel. However, in the reflector convergence image we see convergence *away* from the channel center, indicating differential compaction into the channel. We also note a circular buildup in the upper right corner indicated by the magenta arrow.

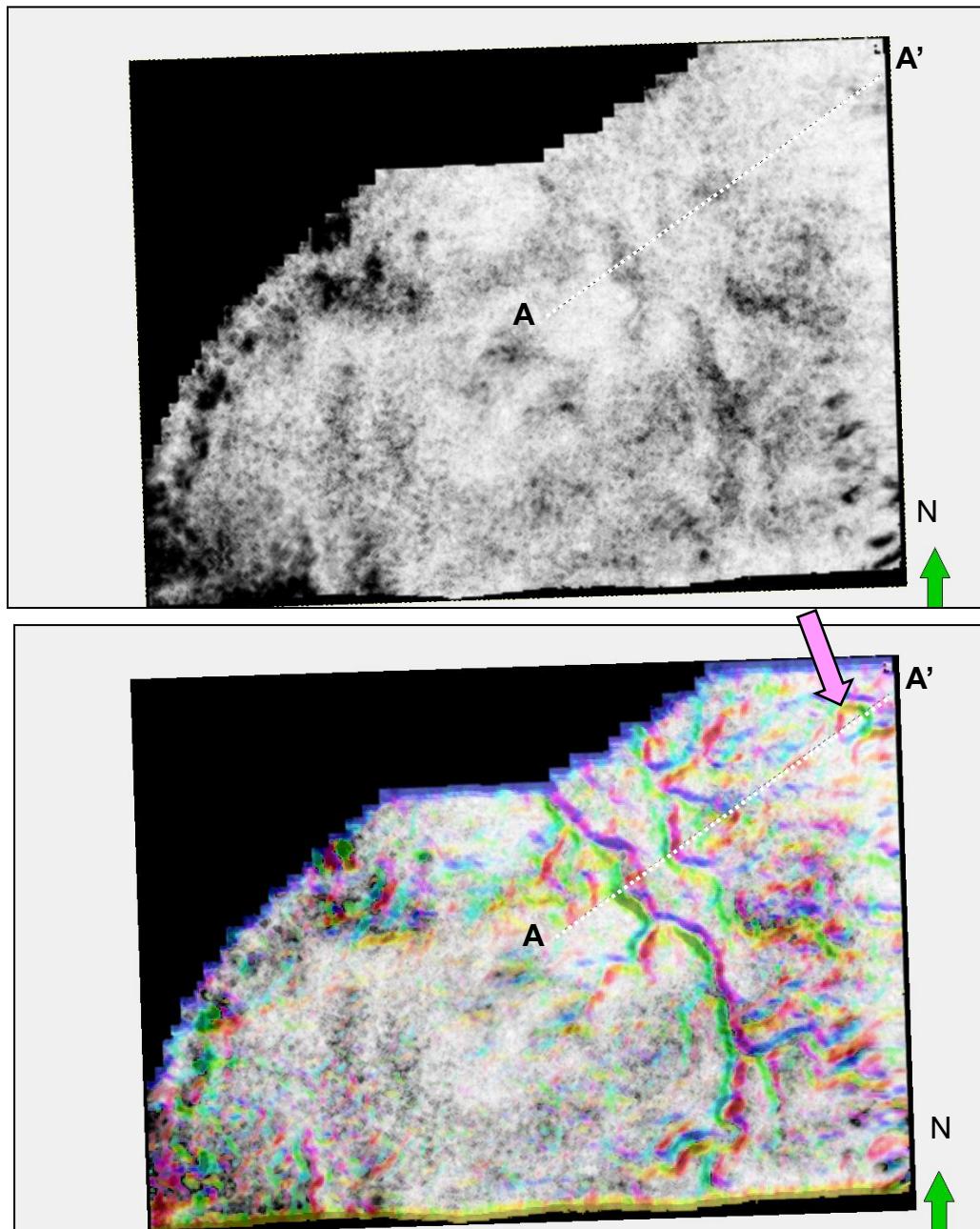


Figure 65.

We show line AA' through the seismic amplitude co-rendered with reflector convergence. Location of line is shown in the previous two figures while white timelines in this vertical slice indicate the time slices shown in the two previous figures. Black and yellow arrows indicate

Geometric Attributes: Program **curvature3d**

convergence anomalies associated with the more western and more eastern channel. Note the flip in the direction of the convergence anomalies between the incision seen at t=1.700 s and the differential compaction seen at t=1.670 s. Pink arrows indicate the near-circular buildup seen in the shallower time slice at t=1.670 s.

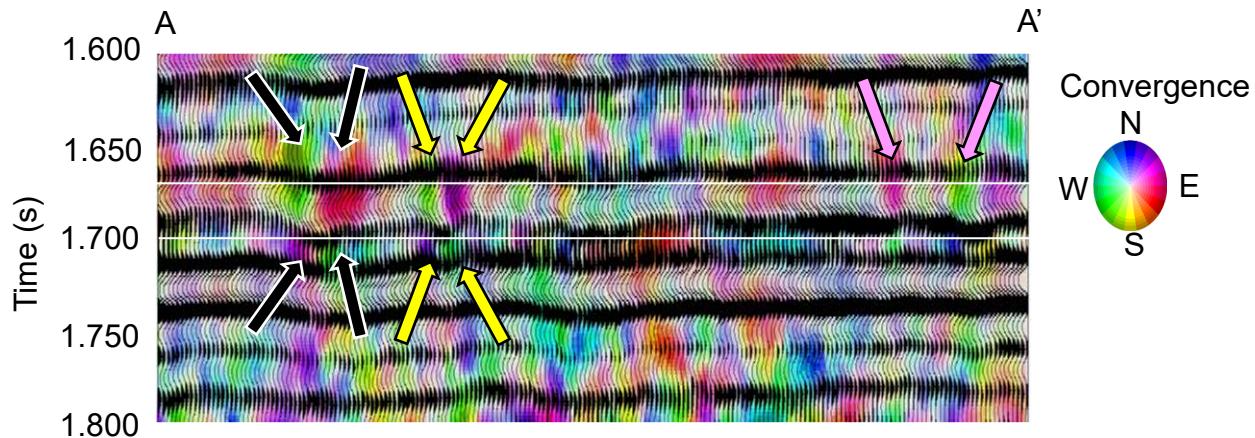


Figure 66.

Finally, we draw an image from Chopra and Marfurt (2011) from the Western Sedimentary Basin that shows the vector convergence anomalies within fault blocks, which we interpret to be rotation of the graben providing increased accommodation space in the direction indicated by the hue. Note that most of the features are displayed as white, implying the features were laid down parallel before structural deformation:

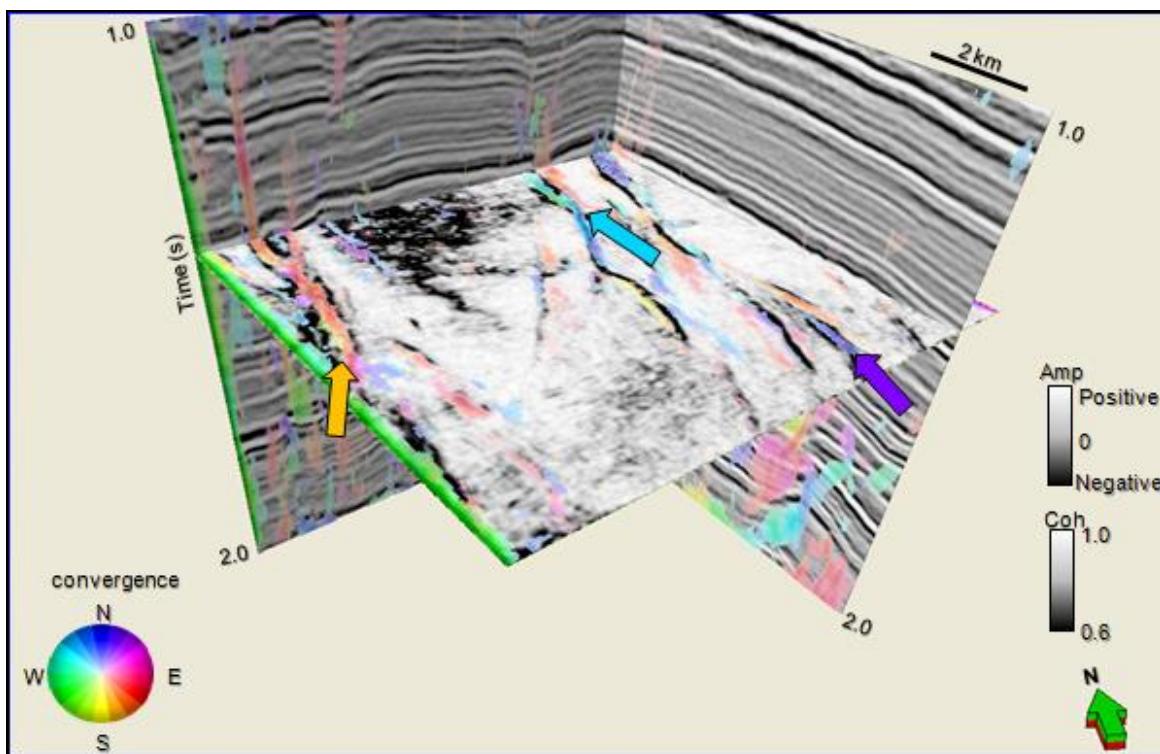


Figure 67.

Geometric Attributes: Program **curvature3d**

Examples showing rotation about the normal

An example of rotation comes from Chopra and Marfurt (2013). Here we define the polarity of the rotation with red indicating down the right and blue indicating down to the left:

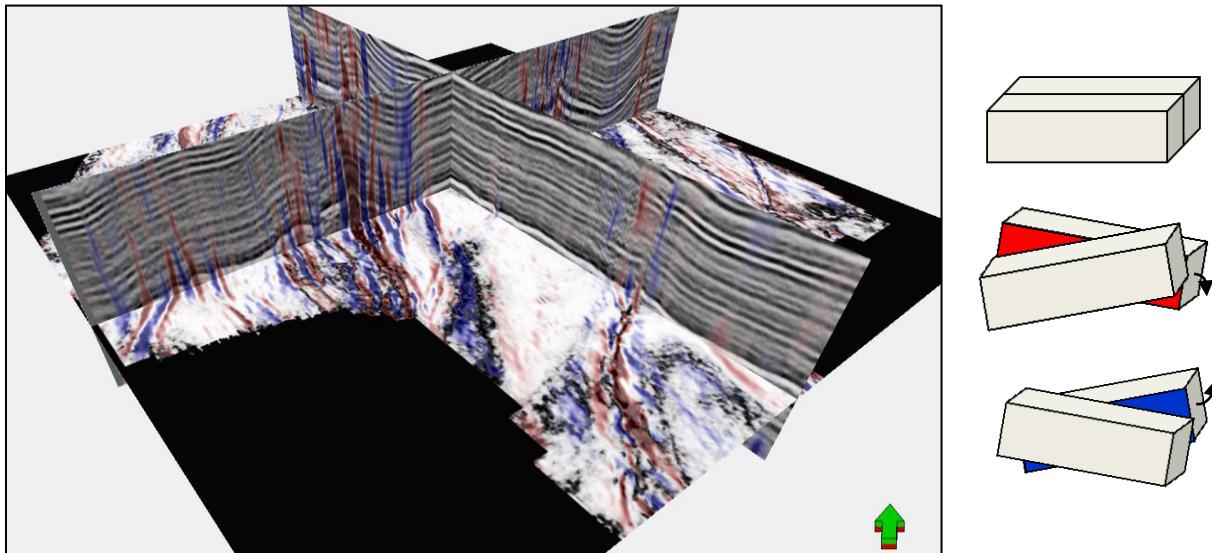


Figure 68.

The Woodford Shale in the Anadarko Basin has variable thickness. Using core data, Gupta et al. (2011) explain the lateral variability in SiO₂ (primarily chert), CaCO₃, and TOC (total organic carbon) as being controlled by irregular water depths over the previously eroded Hunton Limestone. The differences between the top and bottom of the Woodford show up well in the following image of reflector rotation:

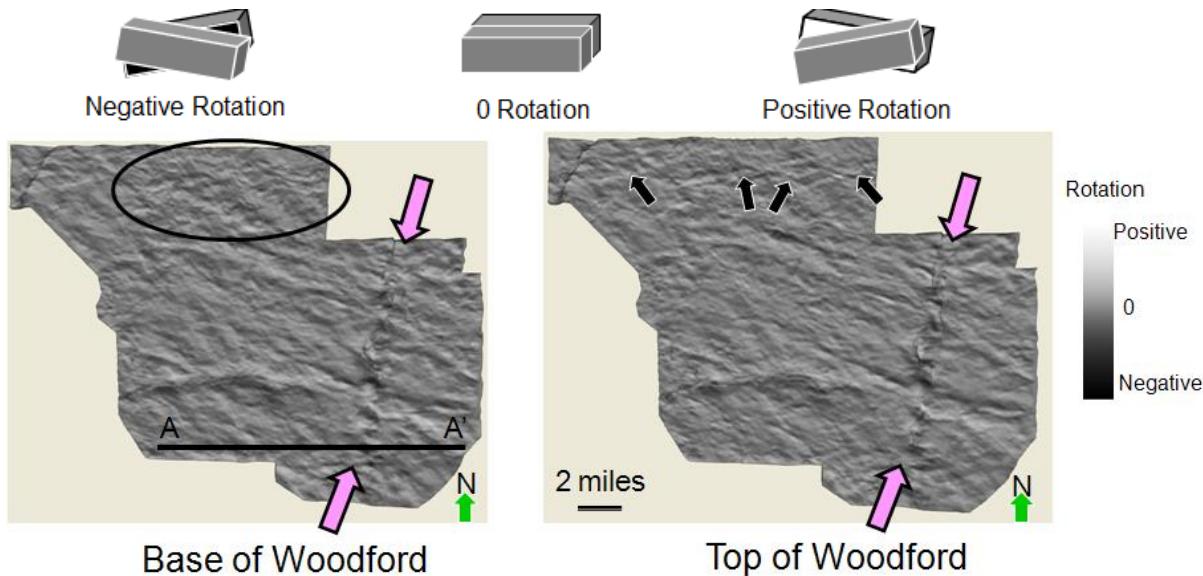


Figure 69.

Geometric Attributes: Program **curvature3d**

Examples showing structural lineaments.

For comparison, we display the corresponding image of most-positive principal curvature lineaments color-coded by their strike:

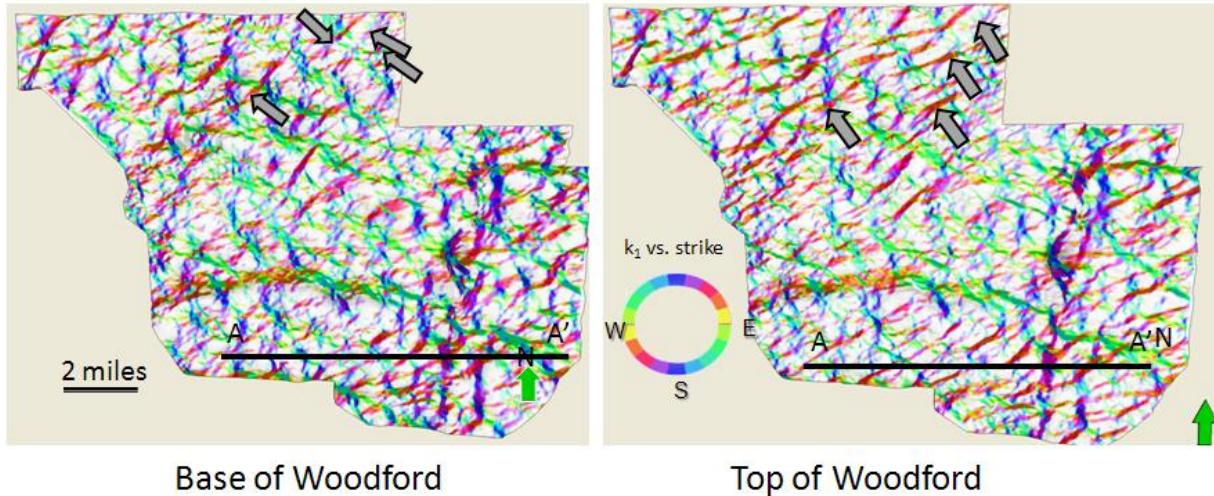


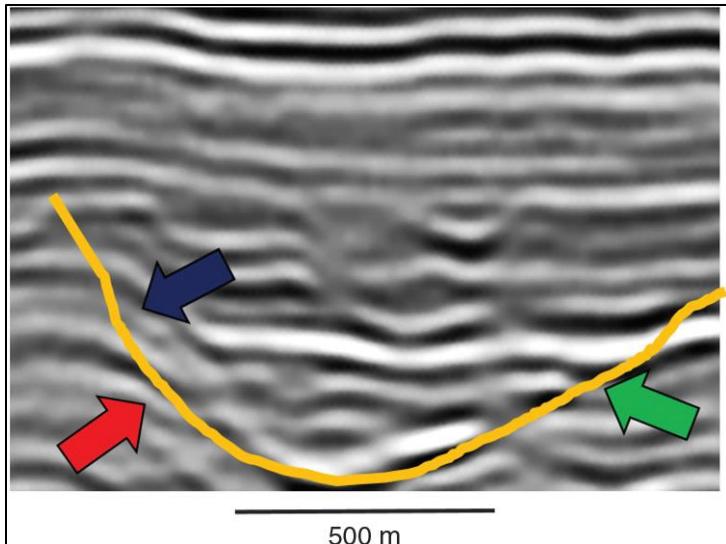
Figure 70.

Other examples of lineaments can be found in following sections in generating rose diagrams and computing azimuthal intensity.

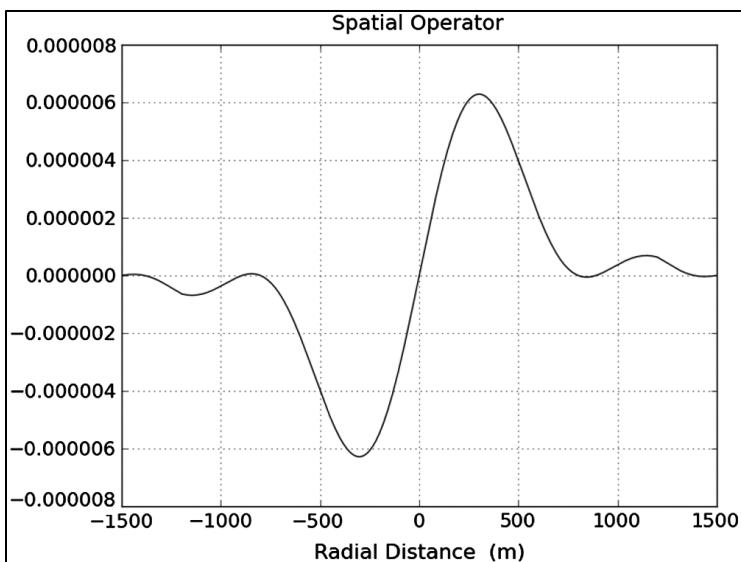
Example showing very long curvature anomalies

In general the default values work very nicely in mapping fault and stratigraphic edges which are dominated by relatively short wavelength lateral changes in dip. However, if the goal is to enhance the appearance of very long wavelength valleys and ridges, the interpreter will need to enlarge the analysis window values of *rmax* from its defaults. The following image from Wallet (2016) shows the size of channel features of interest to be about 1 km:

Geometric Attributes: Program **curvature3d**



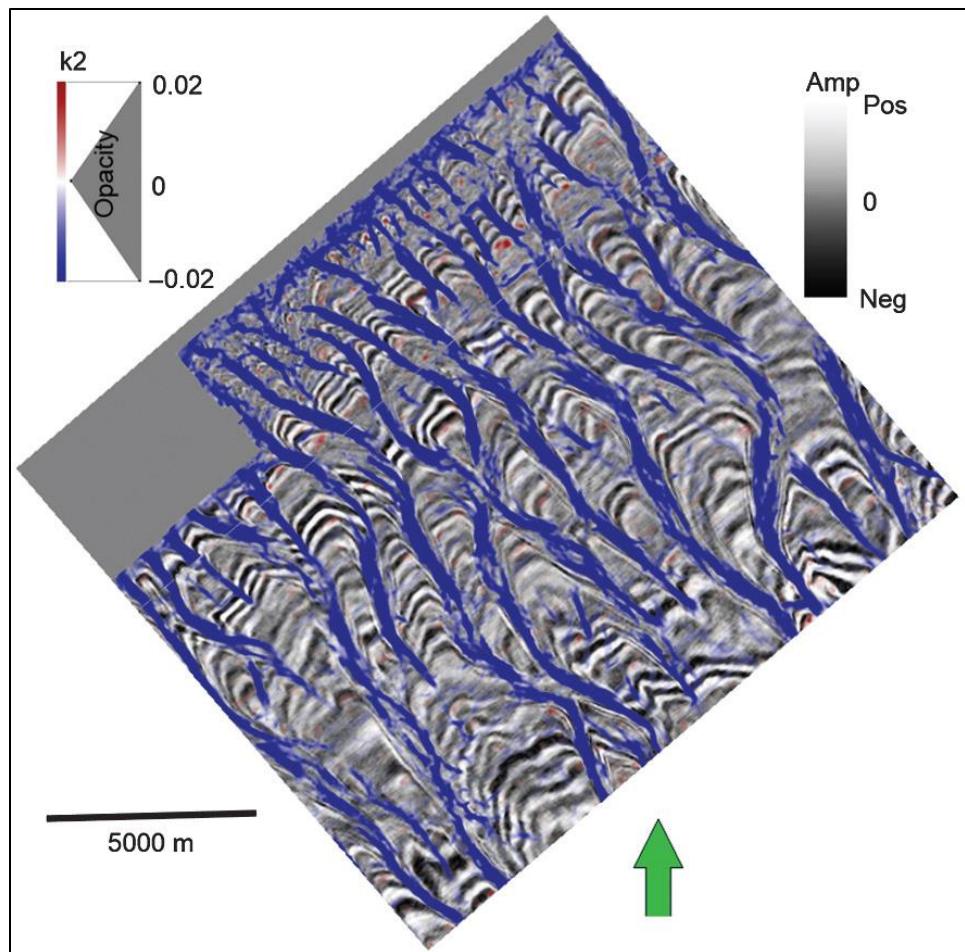
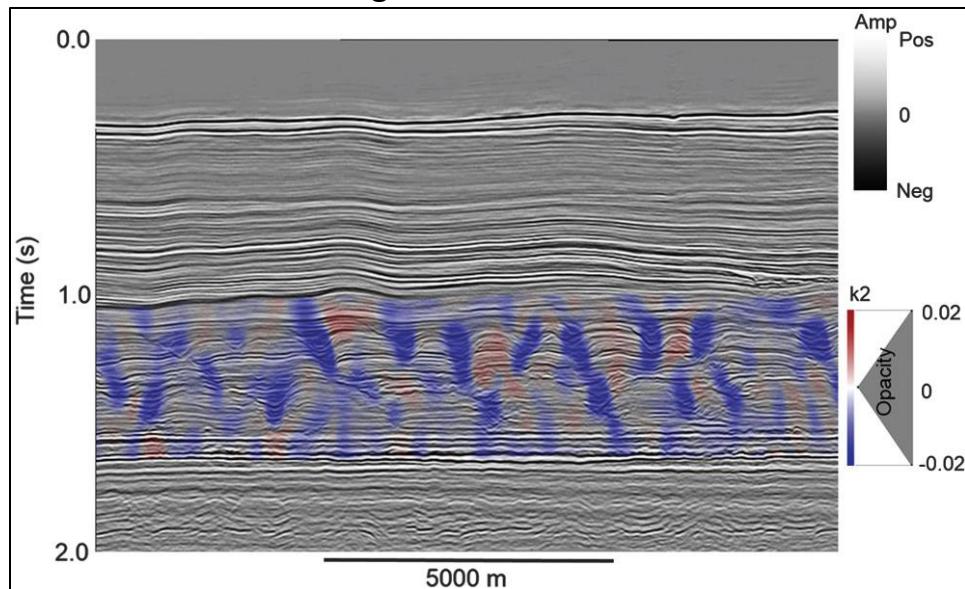
In order to capture this size feature, Wallet (2016) used a longer wavelength operator where the value of rmax was set to be 1500 rather than the default 500 m. Because the computation is convolutional in the inline and crossline directions, the computation time increases as the



square of the effective length of the spatial operator.

The resulting vertical and phantom horizon slices through the most negative curvature k_2 volume look like this:

Geometric Attributes: Program curvature3d



Geometric Attributes: Program **curvature3d**

References

- Al-Dossary, S., and K. J. Marfurt, 2006, Multispectral estimates of reflector curvature and rotation: *Geophysics*, **71**, P41-P51.
- Chopra, S., and K. J. Marfurt, 2007, Volumetric curvature attributes add value to 3D seismic data interpretation: *The Leading Edge*, **26**, 856-867.
- Cooper, G. R. J., and D. R. Cowans, 2003, Sunshading geophysical data using fractional order horizontal gradients, *The Leading Edge*, **22**, 204-205.
- Chopra, S., and K. J. Marfurt, 2011: Which Curvature do you want to use?: GCSSEPM 31st Annual Bob. F. Perkins Research Conference.
- Chopra, S., and K. J. Marfurt, 2013, Volumetric estimates of seismic reflector rotation and convergence – tools for mapping rotation about faults and seismic stratigraphy: Submitted to TLE.
- Bergbauer, S., T. Mukerji, and P. Hennings, 2003, Improving curvature analyses of deformed horizons using scale-dependent filtering techniques: *AAPG Bulletin*, **87**, 1255-1272.
- Gupta, N., S. Sarkar, and K. J. Marfurt, 2011, Seismic characterization of the Woodford shale in the Anadarko basin: *SEG Expanded Abstracts* **30**, 1083-1087.
- Mai, H. T., K. J. Marfurt, and A. S. Chávez-Pérez, 2009, Coherence and volumetric curvatures and their spatial relationship to faults and folds, an example from Chicontepec Basin, Mexico: 79th Annual International Meeting of the SEG, *Expanded Abstracts*, 1063-1067.
- Marfurt, K. J., 2006: Robust estimates of reflector dip and azimuth: *Geophysics*, **71**, P29-P40.
- Marfurt, K. J., and J. Rich, 2010, Beyond curvature – Volumetric estimation of reflector rotation and convergence: 80th Annual International Meeting of the SEG, *Expanded Abstracts*, 1467-1472.
- Marfurt, K. J. , 2010, The shape of seismic interpretation: 2010 Bob F. Perkins Research Conference on Seismic Geomorphology, 241-294.
- Rich, J., 2008, Expanding the applicability of curvature attributes through clarification of ambiguities in derivation and terminology: 78th Annual International Meeting of the SEG, *Expanded Abstracts*, 884-888.
- Roberts, A., 2001, Curvature attributes and their application to 3D interpreted horizons. *First Break*, **19**, 85-99.
- Sigismondi, E.M., and C. J. Soldo, 2003, Curvature attributes and seismic interpretation: Case studies from Argentina basins: *The Leading Edge*, **22**, 1122-1126.
- Wallet, B. C., 2016, Attribute expression of channel forms in a hybrid carbonate turbidite formation: *Interpretation*, **4**, SE75-SE86.