COMPUTING REFLECTOR CURVATURE, ROTATION, AND CONVERGENCE - PROGRAM curvature3d

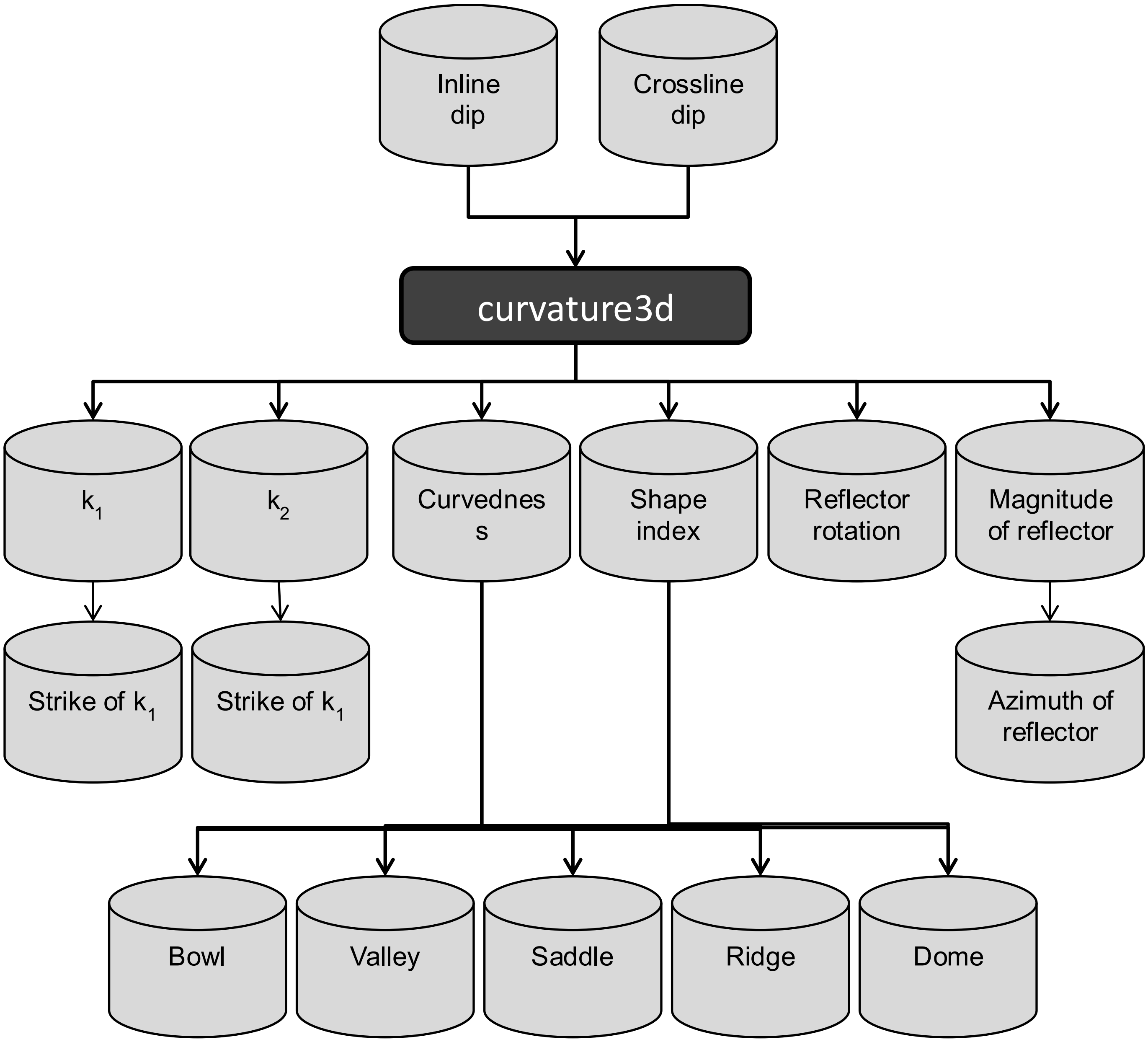
Program **curvature3d** has two modes – computing second derivatives of the phase of seismic waveforms (1^{st} derivatives of structural dip components) or structural curvature, and computing 2^{nd} derivatives of amplitude, (1^{st} derivatives of amplitude gradients) or *amplitude* curvature. In addition 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). At present, Marfurt favors applying an explicitly-defined band-pass filter to the derivative operators, rather than the somewhat heuristic fractional derivative operator.

In 2010, major update was made to compute derivatives using true 3D vs. 2D (time slice, depth slice, or local horizon slice) operators. 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 facilitated the introduction of several new attributes that required 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.

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.

Computation flow chart for *structural* curvature

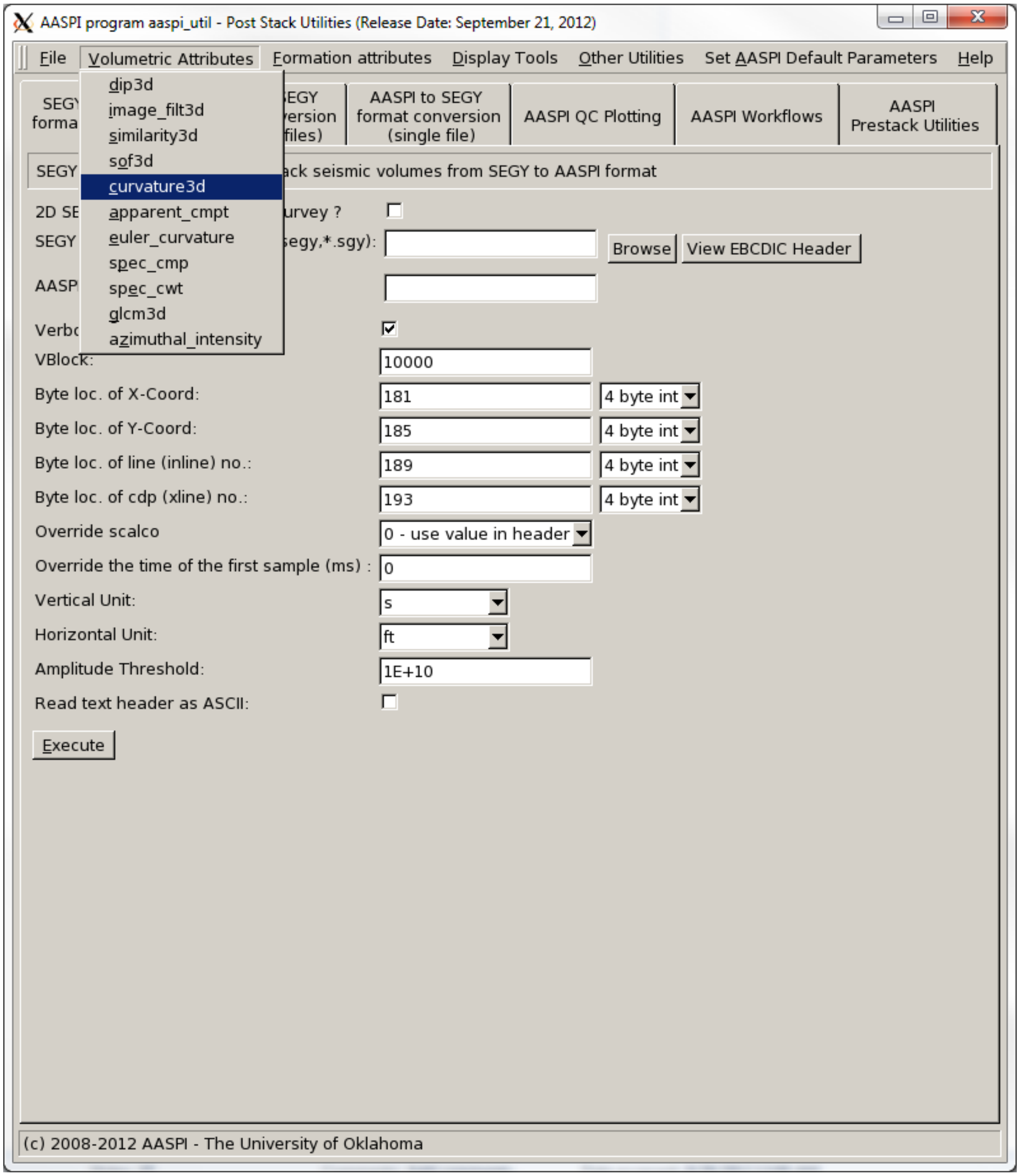
The input to program **curvature3d** will be the inline and crossline components of dip computed using programs **dip3d** or **image_filt3d**. The basic curvature outputs include the value and strike (eigenvalue and eigenvector) of the most-positive and most-negative principal curvatures, $k_1$and k_2. 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 convergences which can be used to map clinoforms, fans, levees, and erosional unconformities. A suite of optional volumes of historical interest are 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 ones discussed above.



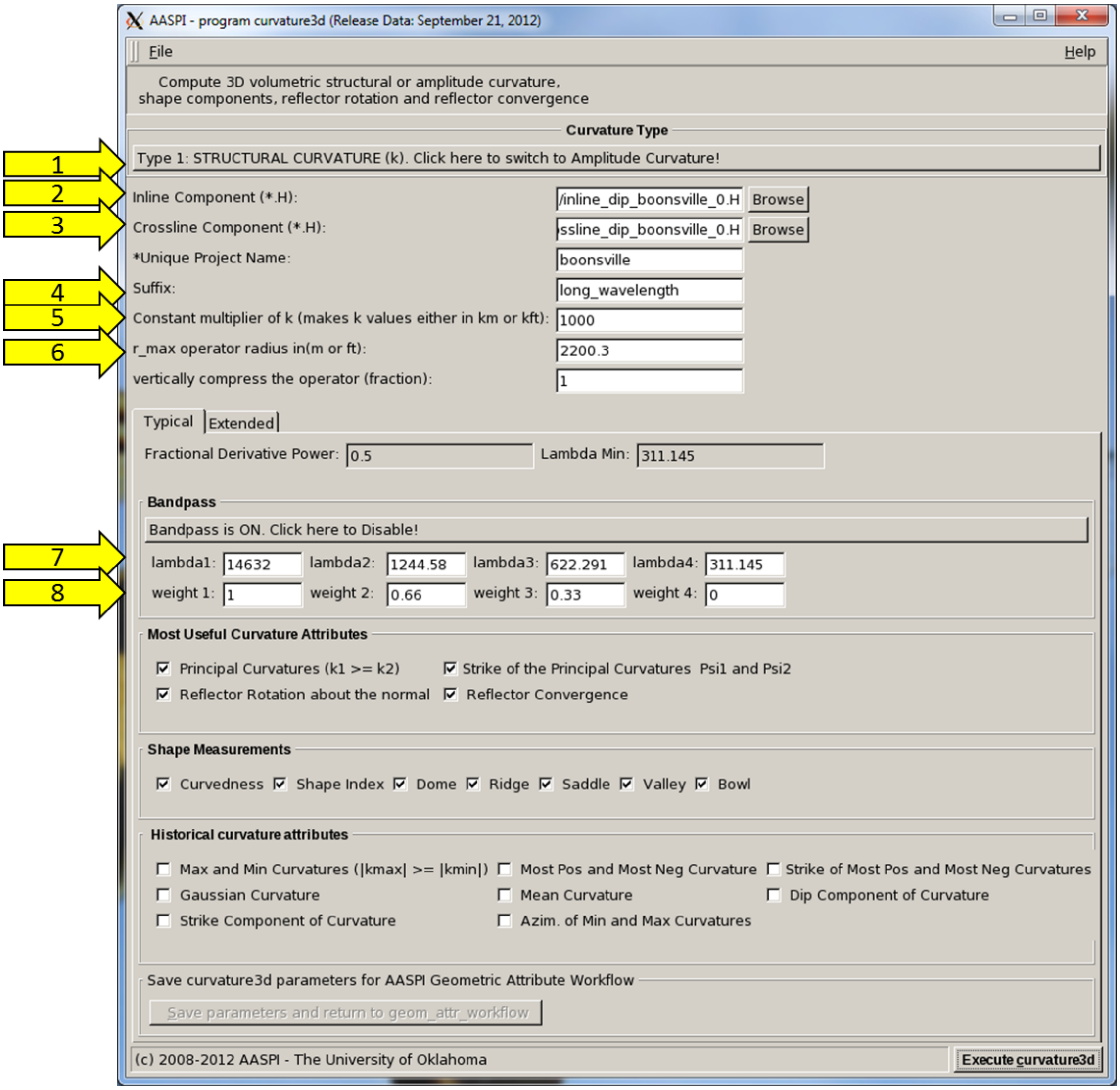
Computing *structural* curvature

Volumetric Attributes- Curvature3d

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



The following GUI should appear:



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 section.

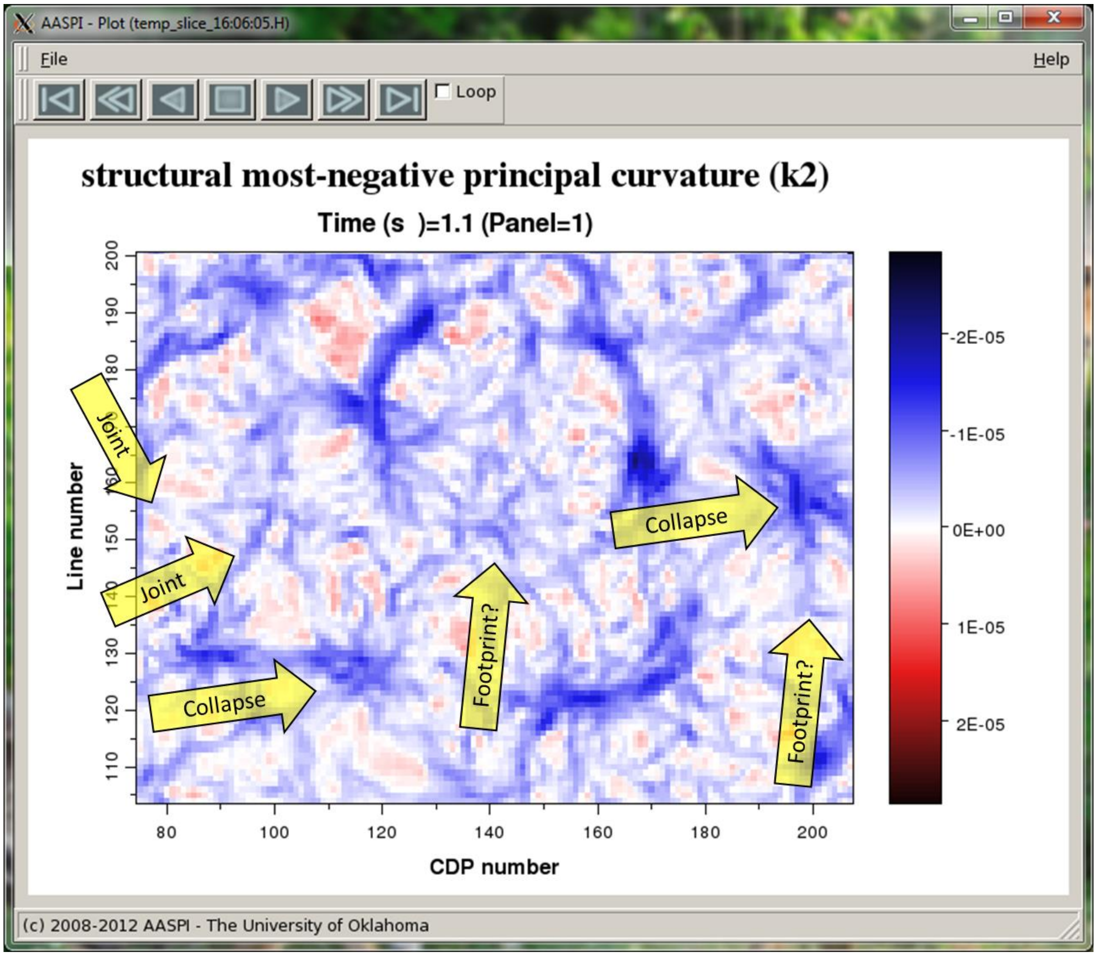
Our input files are (2) inline_dip_boonsville_0.H and (3) crossline_dip_boonsville_0.H computed from program **dip3d**. Set (4) the *Suffix* to read 'long_wavelength' and (5) the *Constant multiplier of k* to be 1000, giving us curvature measurements in kft or km. Normally, you should not need to change (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.

The **curvature3d** GUI allows for definition of a filter defined by four points. The actual filter implementation is done convolutionally in the space domain and the filter display in the wavenumber domain. However, for better understanding we designed the GUI to define the filter in the wavelength domain. The shortest (and default) wavelength *lambda4* is defined by Nyquist and is 331 ft for the Boonsville data volume and assigned a weight of 0.0 . The longest (and default) wavelength *lambda1* is defined by the shortest lateral dimension of the survey and is 14,632 ft for the Boonsville data and assigned a weight of 1. The default for *lambda2=2*lambda4* with a weight of 0.33, and *lambda3=4*lambda4* and a weight of 0.66. We will plot the operators and their spectra later in this section.

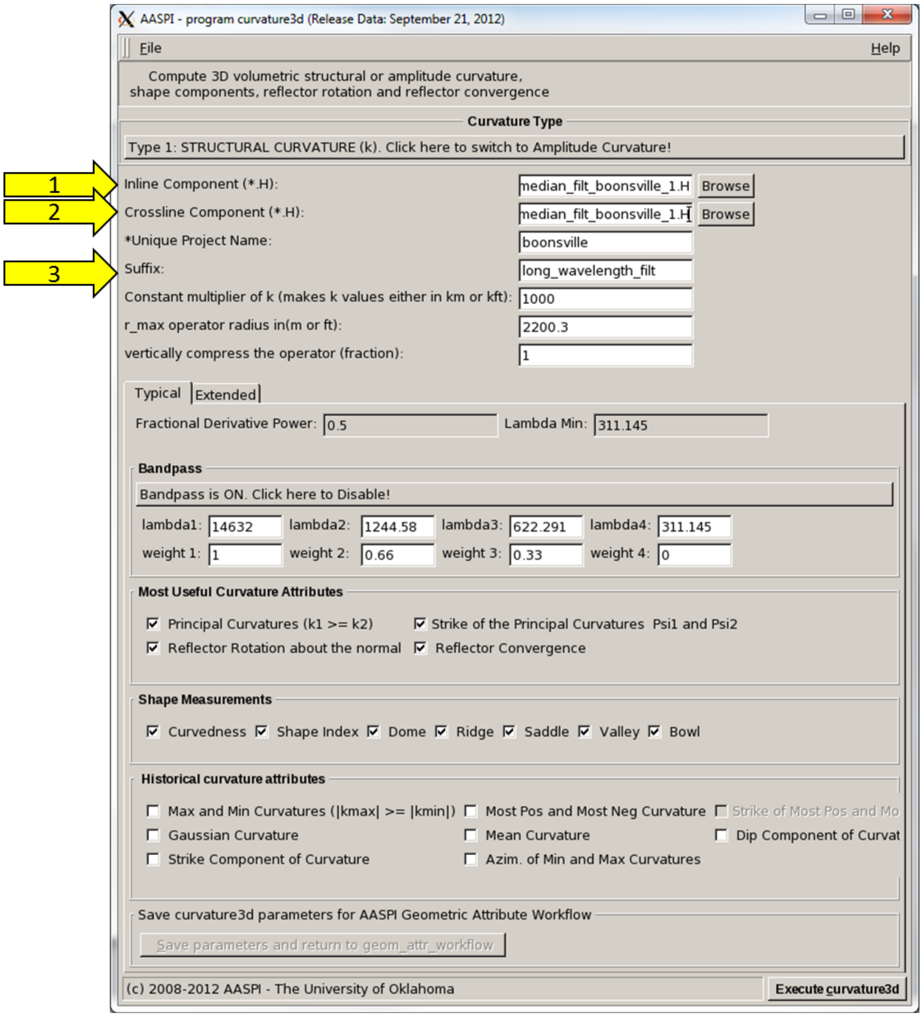
Don't change (7) the default value of *lambda4* of 311 ft. Since the Boonsville data were sampled using 110 ft by 110 ft bins, Nyquist sampling criteria requires two samples per wavelength ($2 \times [110^2+110^2]^{1/2} = 311\ ft$ for a structural feature aligned diagonal to the inline and crossline axes). Keep the weight parameters (8) as default for this case. If you have two or more surveys with mixed sampling (say 82.5 ft, 55 ft, and *110 ft* you may wish to apply the same value of *lambda4* to obtain consistent wavenumbers in the output curvature images. Check the full suite of curvature attributes indicated above (checked boxes). As you might suspect, the output looks like this:

```
  process                            task       time (hr)  time/trace (s)
  11:                  read and scale data         0.000          0.000
  11:                    send data via MPI         0.000          0.000
  11:                 receive data via MPI         0.000          0.000
  11:                 send results via MPI         0.000          0.000
  11:              receive results via MPI         0.000          0.000
  11:                    Hilbert transform         0.000          0.001
  11:                 calculate cov matrix         0.000          0.000
  11:               calculate eigenvectors         0.000          0.001
  11:                       calculate edge         0.000          0.000
  11:                 project data onto v1         0.000          0.000
  11:                write results to disk         0.000          0.000
  11:                           total time         0.009          0.030
        12 :end loop over lines
        12 number of traces processed:        1067
  process                            task       time (hr)  time/trace (s)
  12:                  read and scale data         0.000          0.000
  12:                    send data via MPI         0.000          0.000
  12:                 receive data via MPI         0.002          0.008
  12:                 send results via MPI         0.000          0.000
  12:              receive results via MPI         0.000          0.000
  12:                    Hilbert transform         0.000          0.000
  12:                 calculate cov matrix         0.000          0.000
  12:               calculate eigenvectors         0.000          0.001
  12:                       calculate edge         0.000          0.000
  12:                 project data onto v1         0.000          0.000
  12:                write results to disk         0.000          0.000
  12:                           total time         0.009          0.030
        11 : memory residing only on slaves deallocated
        11 : shared arrays residing on both master and slave deallocated
        12 : memory residing only on slaves deallocated
        12 : shared arrays residing on both master and slave deallocated
         0 :end loop over lines
  process                            task       time (hr)  time/trace (s)
   0:                  read and scale data         0.000          0.000
   0:                    send data via MPI         0.000          0.000
   0:                 receive data via MPI         0.000          0.000
   0:                 send results via MPI         0.000          0.000
   0:              receive results via MPI         0.000          0.000
   0:                    Hilbert transform         0.000          0.000
   0:                 calculate cov matrix         0.000          0.000
   0:               calculate eigenvectors         0.000          0.000
   0:                       calculate edge         0.000          0.000
   0:                 project data onto v1         0.000          0.000
   0:                write results to disk         0.001          0.000
   0:                           total time         0.009          0.000
total data written to disk:       90307 traces        800 samples
total data written to disk:     288.982 Mbytes
transfer rate            :      67.315 Mbytes/s
         0 : memory deallocated residing only on master deallocated
         0 : shared arrays residing on both master and slave deallocated
         1 :normal completion. routine similarity3d
         3 :normal completion. routine similarity3d
         4 :normal completion. routine similarity3d
         2 :normal completion. routine similarity3d
        10 :normal completion. routine similarity3d
         5 :normal completion. routine similarity3d
         7 :normal completion. routine similarity3d
         8 :normal completion. routine similarity3d
        12 :normal completion. routine similarity3d
         9 :normal completion. routine similarity3d
         6 :normal completion. routine similarity3d
         0 :normal completion. routine similarity3d
Closing file: /home/kmarfurt/projects/boonsville/coherent_energy_boonsville_0.H
        11 :normal completion. routine similarity3d
Closing file: /home/kmarfurt/projects/boonsville/crossline_energy_gradient_boonsville_0.H
Closing file: /home/kmarfurt/projects/boonsville/energy_ratio_similarity_boonsville_0.H
Closing file: /home/kmarfurt/projects/boonsville/inline_energy_gradient_boonsville_0.H
Closing file: /home/kmarfurt/projects/boonsville/outer_product_similarity_boonsville_0.H
Closing file: /home/kmarfurt/projects/boonsville/sobel_filter_similarity_boonsville_0.H
Closing file: /home/kmarfurt/projects/boonsville/total_energy_boonsville_0.H
Closing file: /nfs/raid1/home/kmarfurt/projects/boonsville/crossline_dip_median_filt_boonsville_1.H
Closing file: /nfs/raid1/home/kmarfurt/projects/boonsville/d_mig_boonsville.H
Closing file: /nfs/raid1/home/kmarfurt/projects/boonsville/inline_dip_median_filt_boonsville_1.H
[kmarfurt@tripolite boonsville]$ aaspi_util &:ls  -ltr *o
```

Let's plot *k2_boonsville_long_wavelength.H* (we will call this the most-negative *principal* curvature) using the *AASPI QC Plottting tab* and capture the following image delineating some of the karst features and connecting faults or diagenetically-altered joints:



Note that in addition to the joints and collapse features that there are some short-wavelength NS and EW lineaments corresponding to acquisition footprint. Previously, we used program **image_filt3d** to filter our volumetric inline and crossline dip volumes. Let's examine the effect of filtering the median-filtered version of the inline and crossline_dip volumes.



Enter (1) the *Inline component* to be *inline_dip_median_filt_boonsville.H* and (2) the *Crossline component* to be *crossline_dip_median_filt_boonsville.H* as the input dip volumes. Enter (3) the *Suffix* to be long_wavelength_filt to differentiate from curvature volumes that have not been filtered. Click *Execute.* After completion type

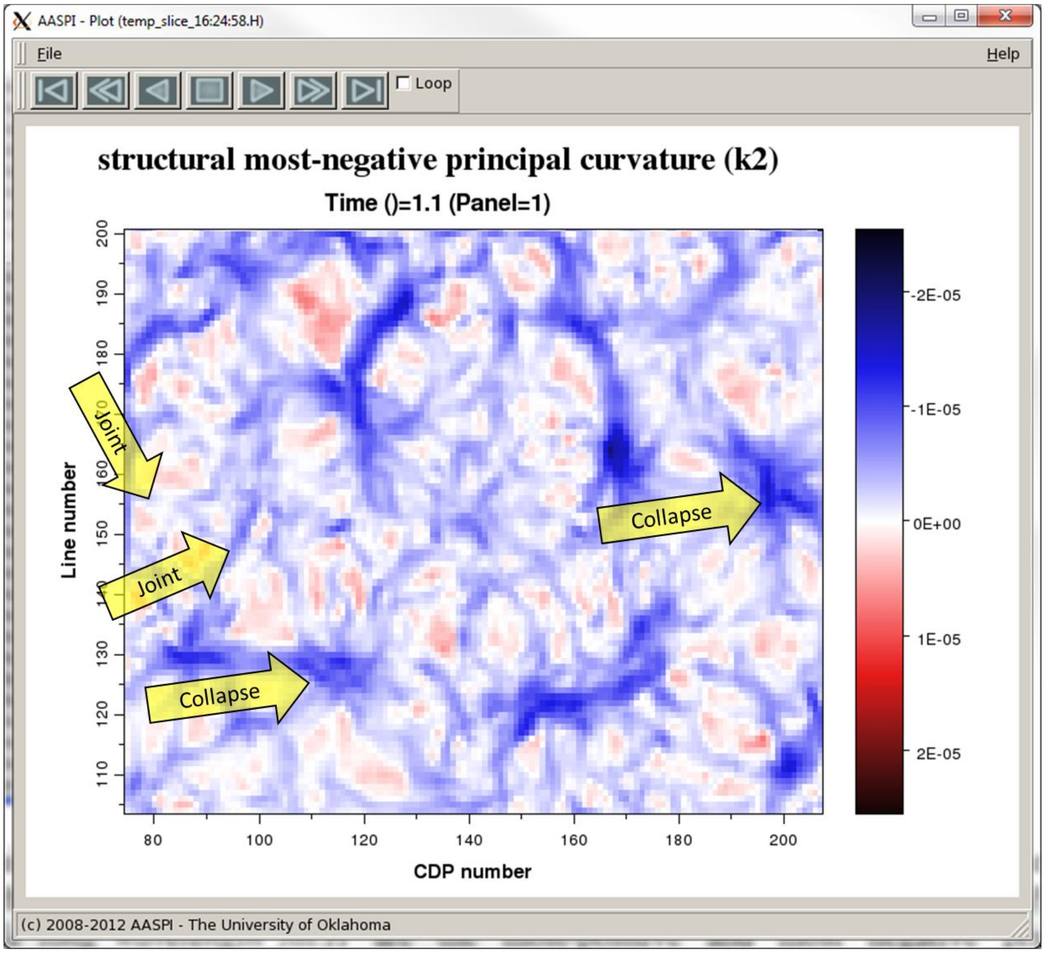
*ls –ltr *long_wavelength*.H*

in your xterm to see which files that have been generated:

```
[kmarfurt@tripolite boonsville]$ ls -ltr *long_wavelength*H
-rw-r--r-- 1 kmarfurt aaspi  296 Nov 27 20:29 d_dr_spectrum_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi  294 Nov 27 20:29 d_dr_operator_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3234 Nov 27 20:30 k2_strike_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3234 Nov 27 20:30 k1_strike_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3251 Nov 27 20:30 k_valley_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3249 Nov 27 20:30 k_s_index_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3251 Nov 27 20:30 k_saddle_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3299 Nov 27 20:30 k_rot_normal_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3237 Nov 27 20:30 k_ridge_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3245 Nov 27 20:30 k_dome_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3263 Nov 27 20:30 k_curvedness_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3291 Nov 27 20:30 k_converge_mag_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3286 Nov 27 20:30 k_converge_azim_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3245 Nov 27 20:30 k_bowl_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3247 Nov 27 20:30 k2_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi 3247 Nov 27 20:30 k1_boonsville_long_wavelength.H
-rw-r--r-- 1 kmarfurt aaspi  301 Nov 27 20:43 d_dr_spectrum_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi  299 Nov 27 20:43 d_dr_operator_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3766 Nov 27 20:44 k_converge_azim_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3725 Nov 27 20:44 k_bowl_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3714 Nov 27 20:44 k2_strike_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3714 Nov 27 20:44 k1_strike_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3731 Nov 27 20:44 k_valley_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3729 Nov 27 20:44 k_s_index_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3730 Nov 27 20:44 k_saddle_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3779 Nov 27 20:44 k_rot_normal_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3716 Nov 27 20:44 k_ridge_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3725 Nov 27 20:44 k_dome_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3743 Nov 27 20:44 k_curvedness_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3771 Nov 27 20:44 k_converge_mag_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3727 Nov 27 20:44 k2_boonsville_long_wavelength_filt.H
-rw-r--r-- 1 kmarfurt aaspi 3727 Nov 27 20:44 k1_boonsville_long_wavelength_filt.H
[kmarfurt@tripolite boonsville]$
```

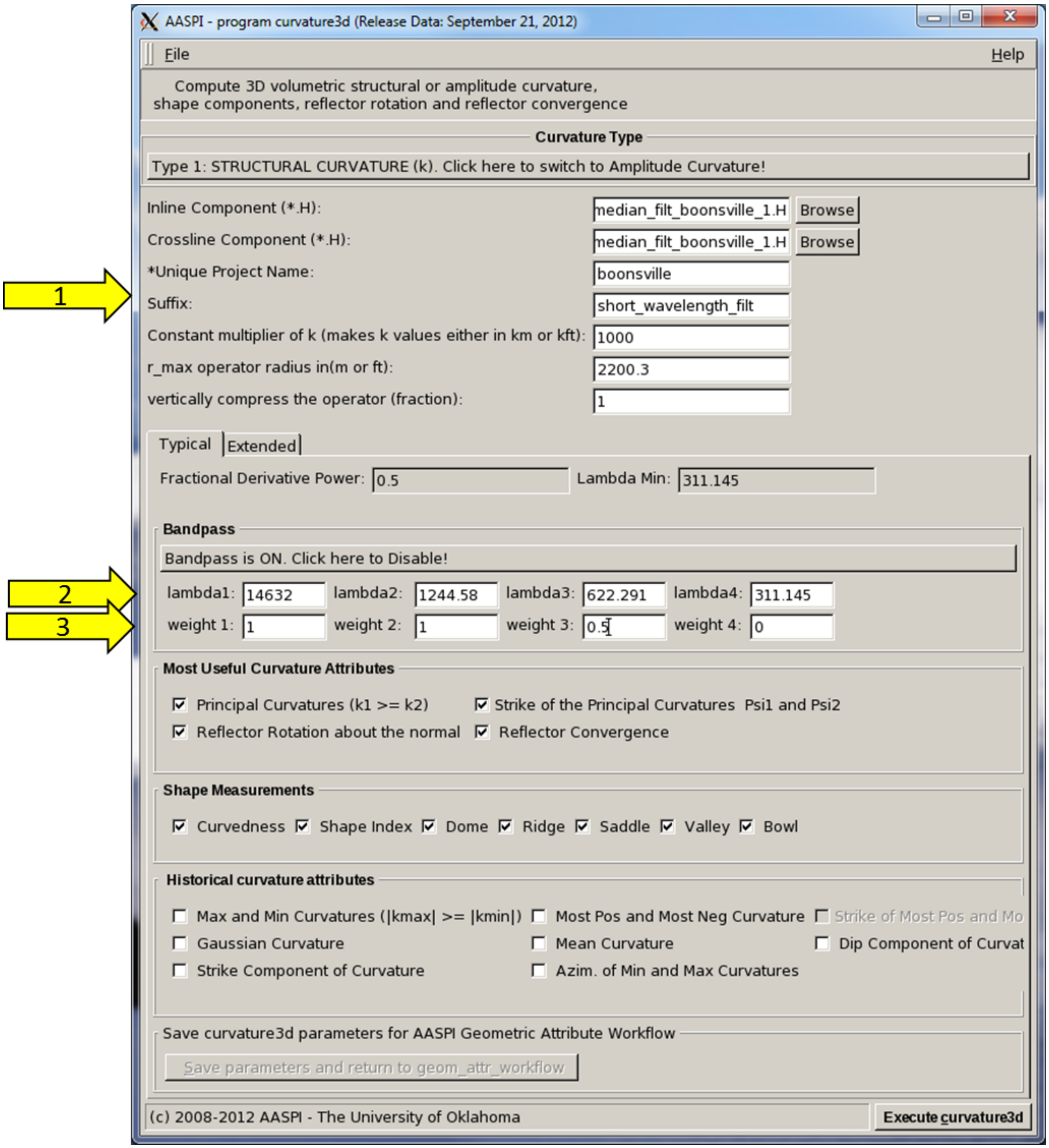
Files beginning with the letter ‘k’ correspond to structural curvature (vs. amplitude or energy curvature which will start with the letter ‘e’). *k1_boonsville_long_wavelength_filt.H* and *k2_boonsville_long_wavelength_filt.H* are the most-positive and most negative principal curvatures computed for the Boonsville survey using ‘long wavelength’ parameters (the four values of lambda1, lambda2, lambda3, and lambda4) from the filtered data.

Plotting *k2_boonsville_long_wavelength_filt.H* we obtain:

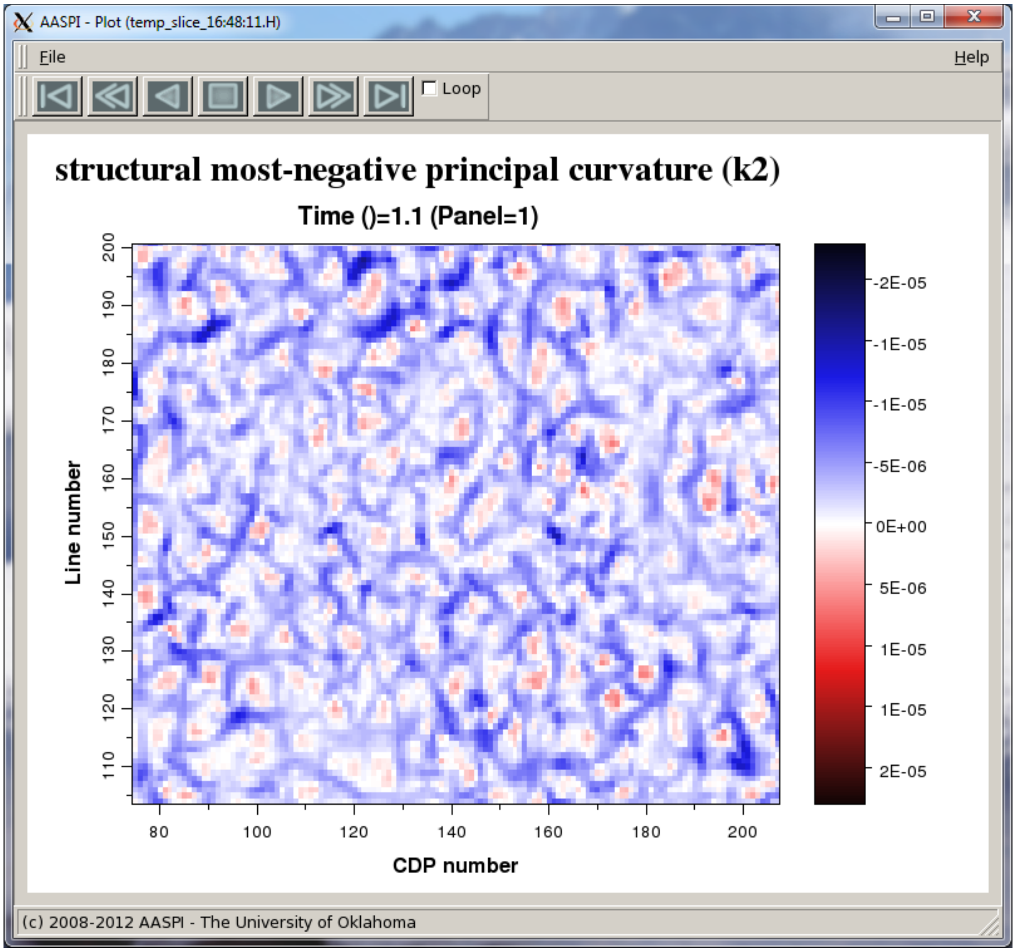


Note that the image is quite similar to the k_2 principal curvature computed from the unfiltered data, except that the footprint has been suppressed.

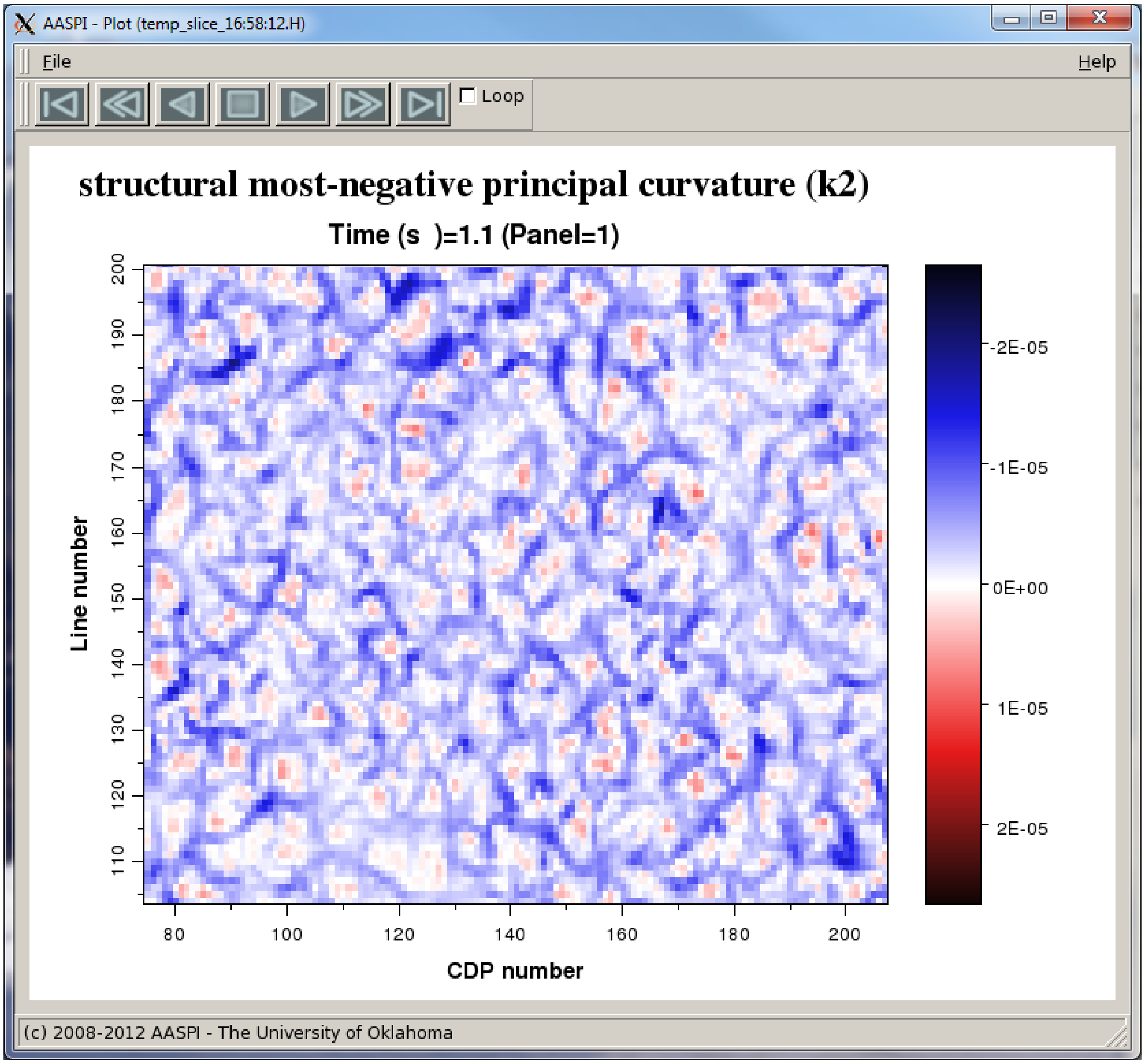
To obtain a shorter-wavelength version of the most negative principal curvature (1) set the *Suffix* to read short_wavelength_filt, (2) keep the values of *lambda1* through *lambda4* the same for now, and (3) modify *weight 1* through *weight 4* to be 1, 1, 0.5, and 0 as shown below:



Click *Execute*, and bring up the AASPI QC Plottting tab to plot the result *k2_boonsville_short_wavelength.H:*



Much of the NS and EW lineaments are due to acquisition footprint. The footprint is exacerbated if we plot the k_2 most-negative principal curvature computed from the unfiltered dip estimates (figure below):

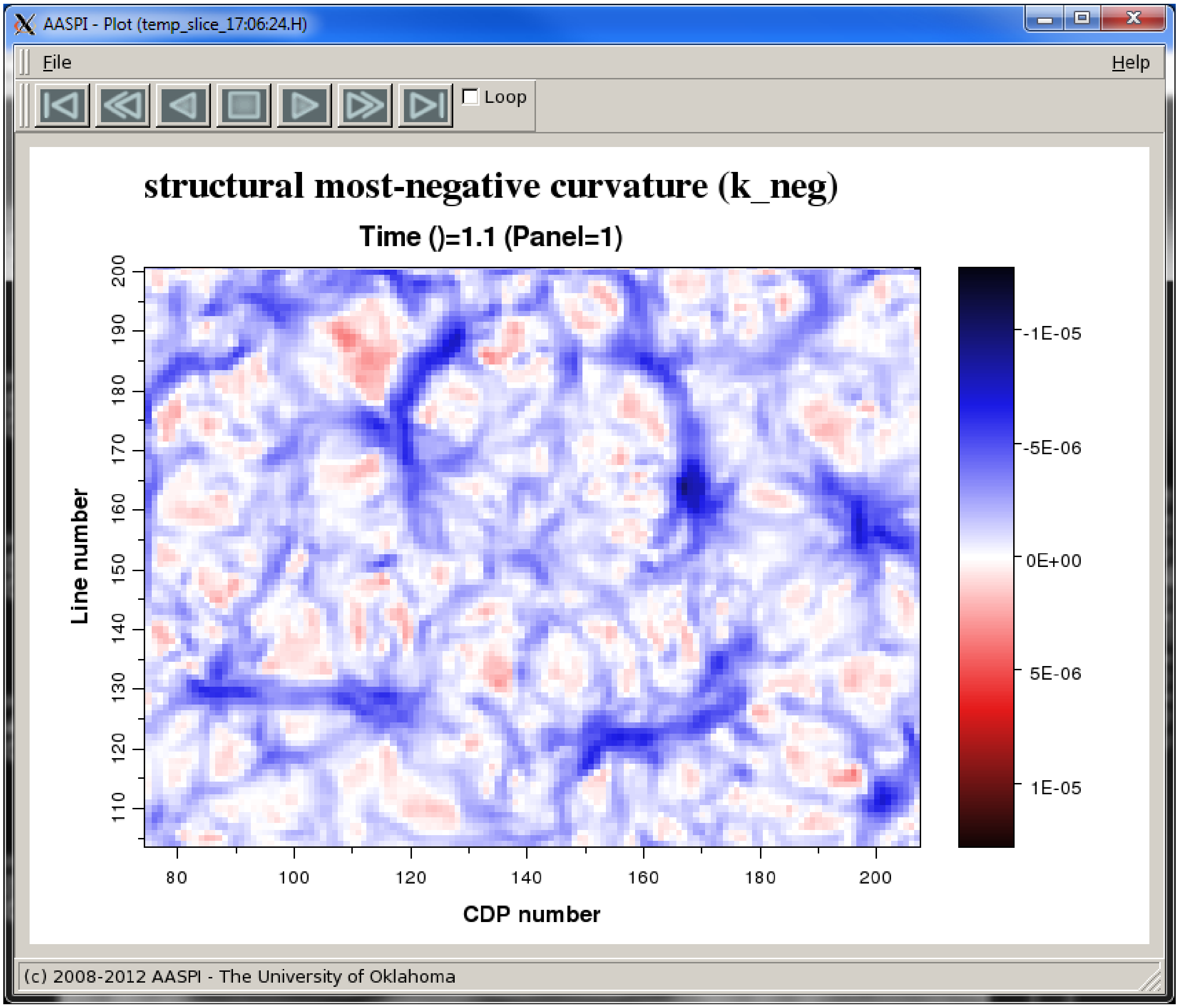


When volumetric curvature was first introduced, there was (and still remains) some confusion on the definition of maximum and minimum curvature. The AASPI software will use the convention used by the majority of geophysical workers (e.g. Sigismundi and Soldo, 2003) and almost all of the differntial geometry 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), however, 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) used the most-negative and most-positive curvatures. 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:

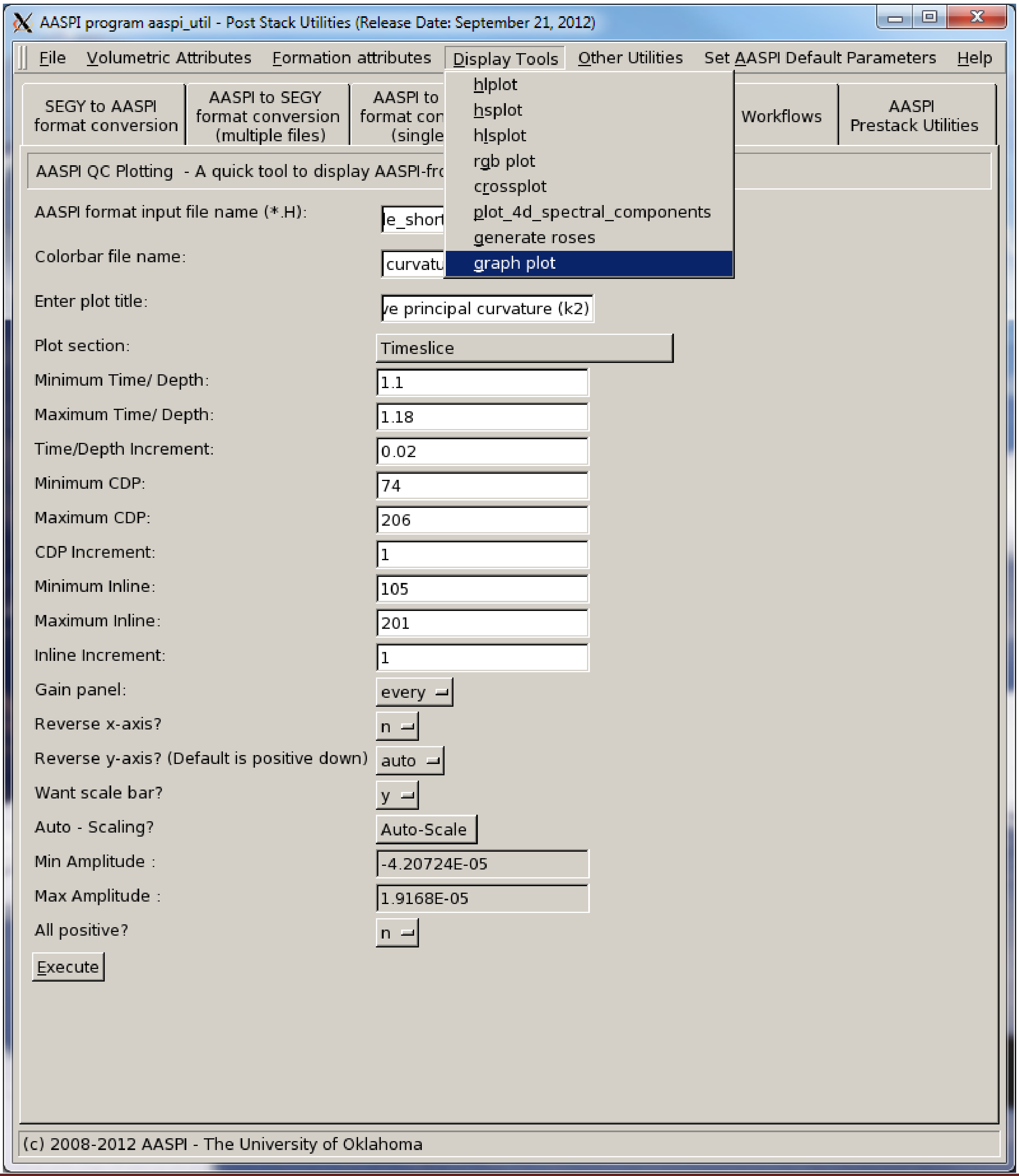


Indeed, k_{neg} and k_2 will produce nearly identical results if the overall dip is relatively flat. In the Boonsville survey, the average dip is less than 5^0. For more tectonically deformed terrains, the images can be significantly different, as described by Mai et al. (2009).

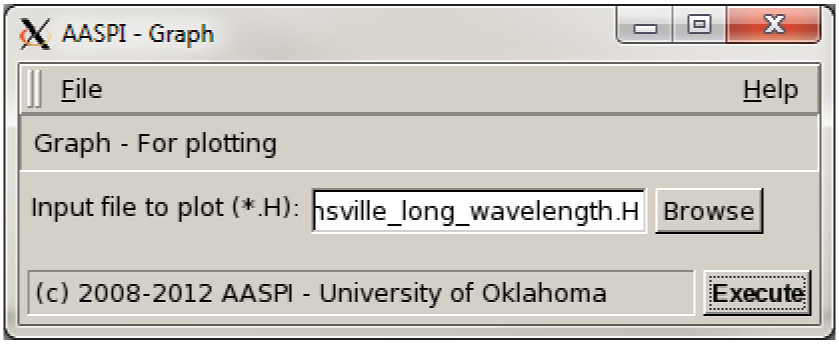
Defining and displaying operators and operator spectra used in curvature3d

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

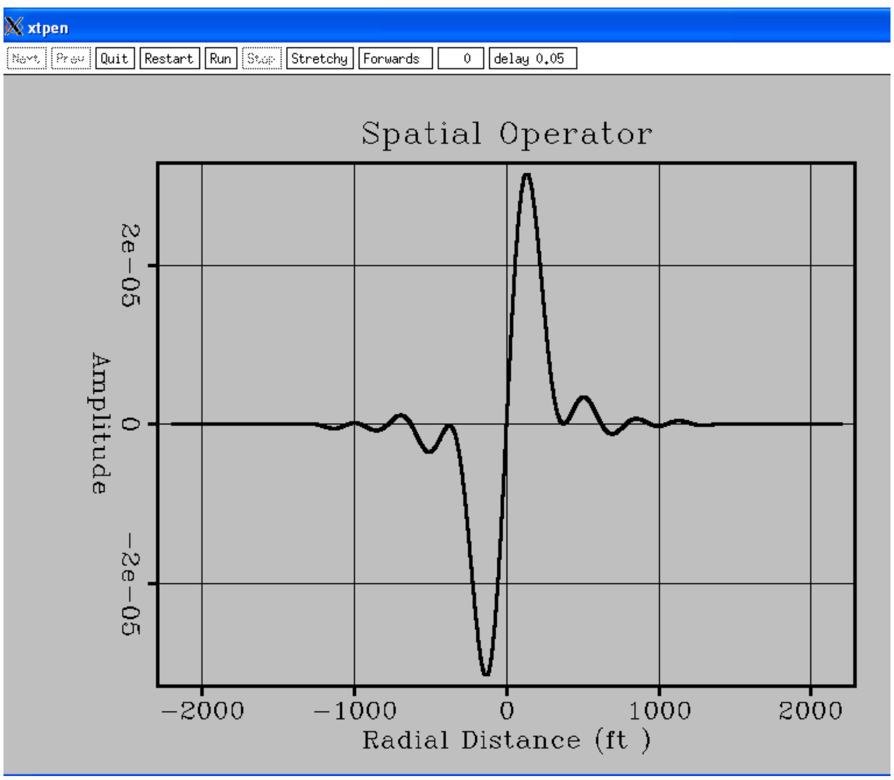
To facilitate graphing these operators, we have run a very simple GUI to display them, found under the Display tools tab. Let's go under *Display Tools* in **aaspi_util** and invoke **graph plot:**



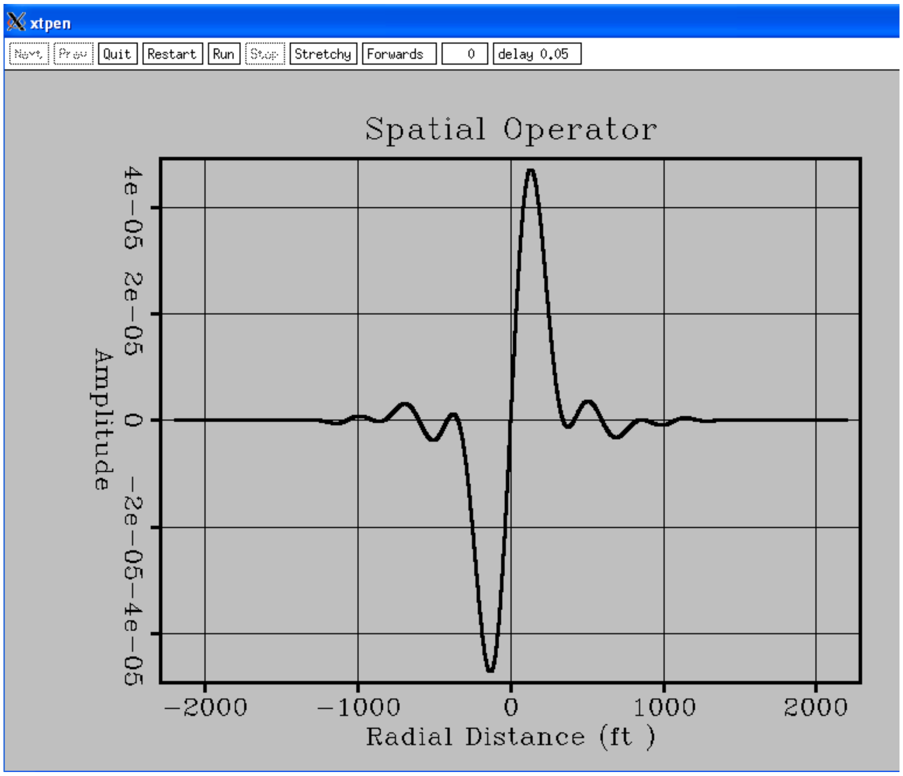
The following GUI appears:



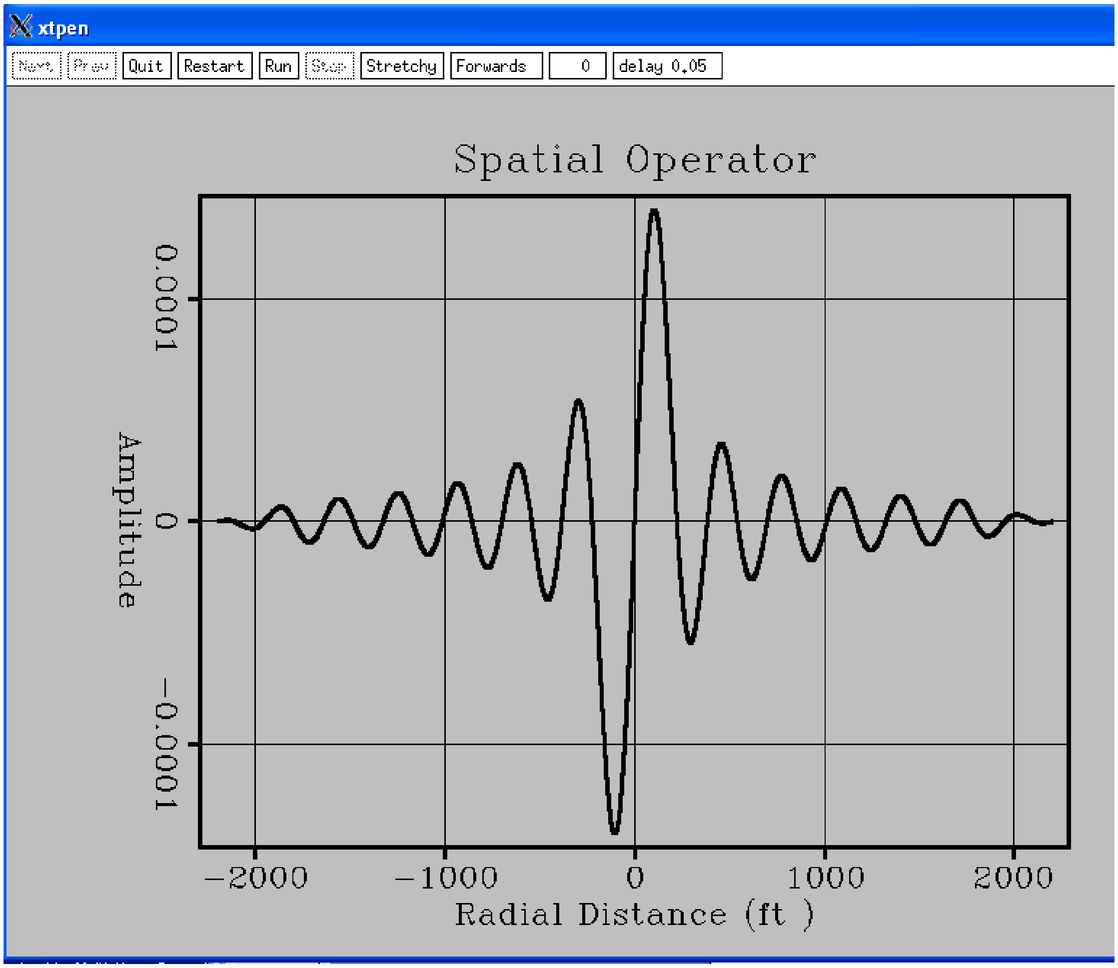
Select *d_dr_operator_boonsville_long_wavelength_filt.H* and obtain the following image:



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



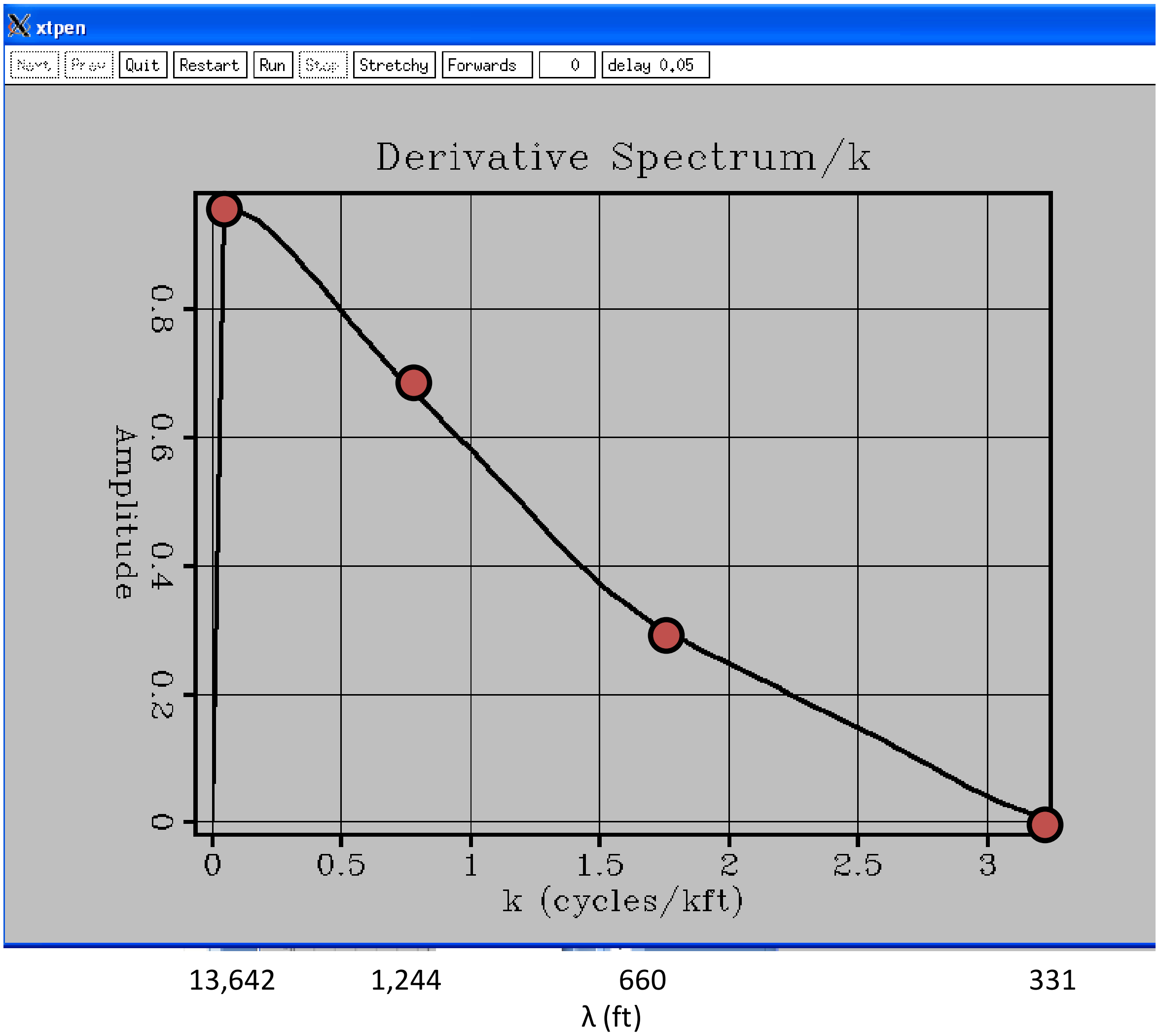
The short wavelength operator was computed using values of *lambda1* through *lambda4* of 1.0, 1.0, 0.5, and 0.0. Let's rerun **curvature3d** without 'any' filter using values of *lambda1* through *lambda4* of 1.0, 1.0, 1.0, and 1.0.. Recall that this would run the filter right up against the Nyquist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

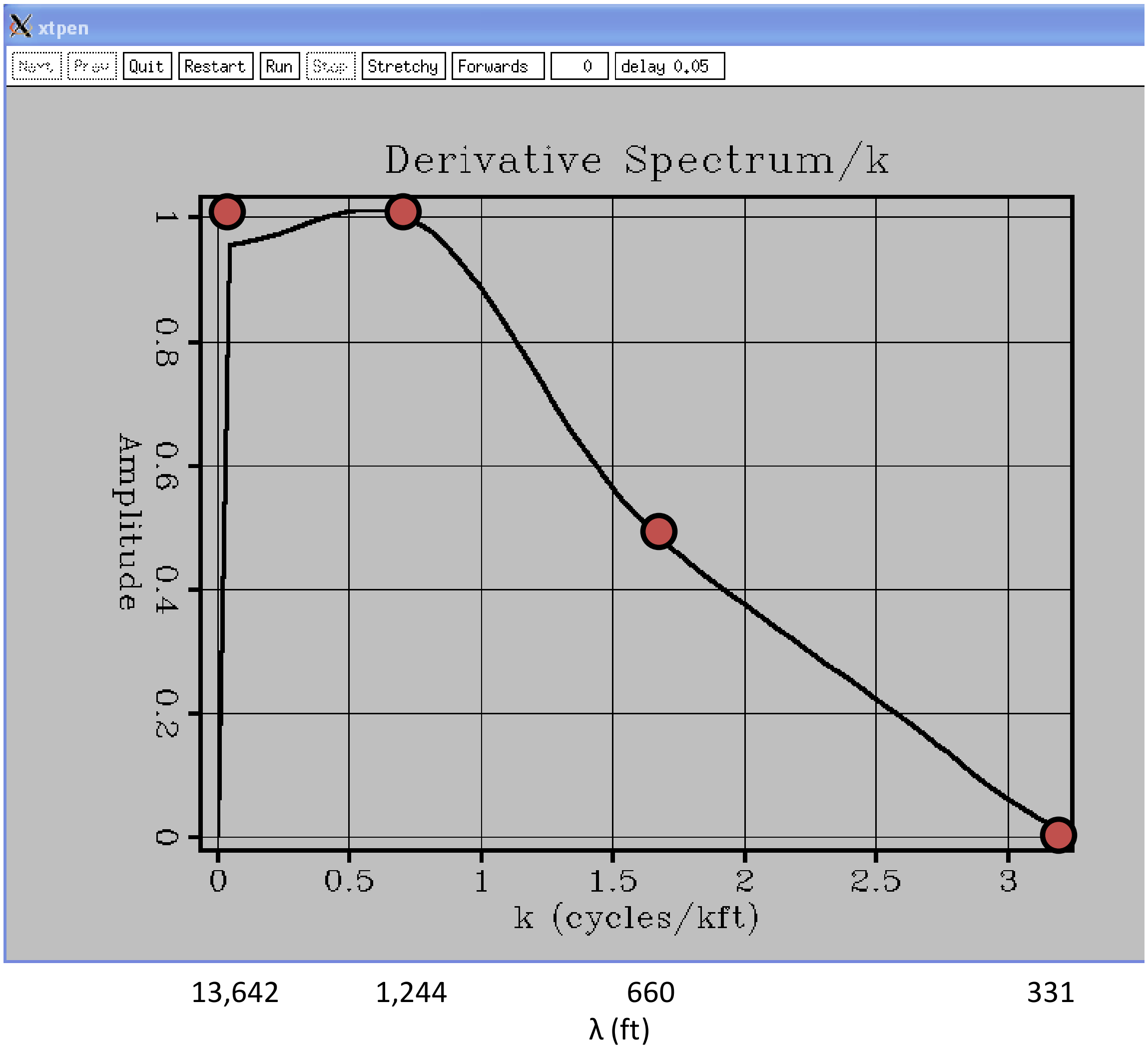


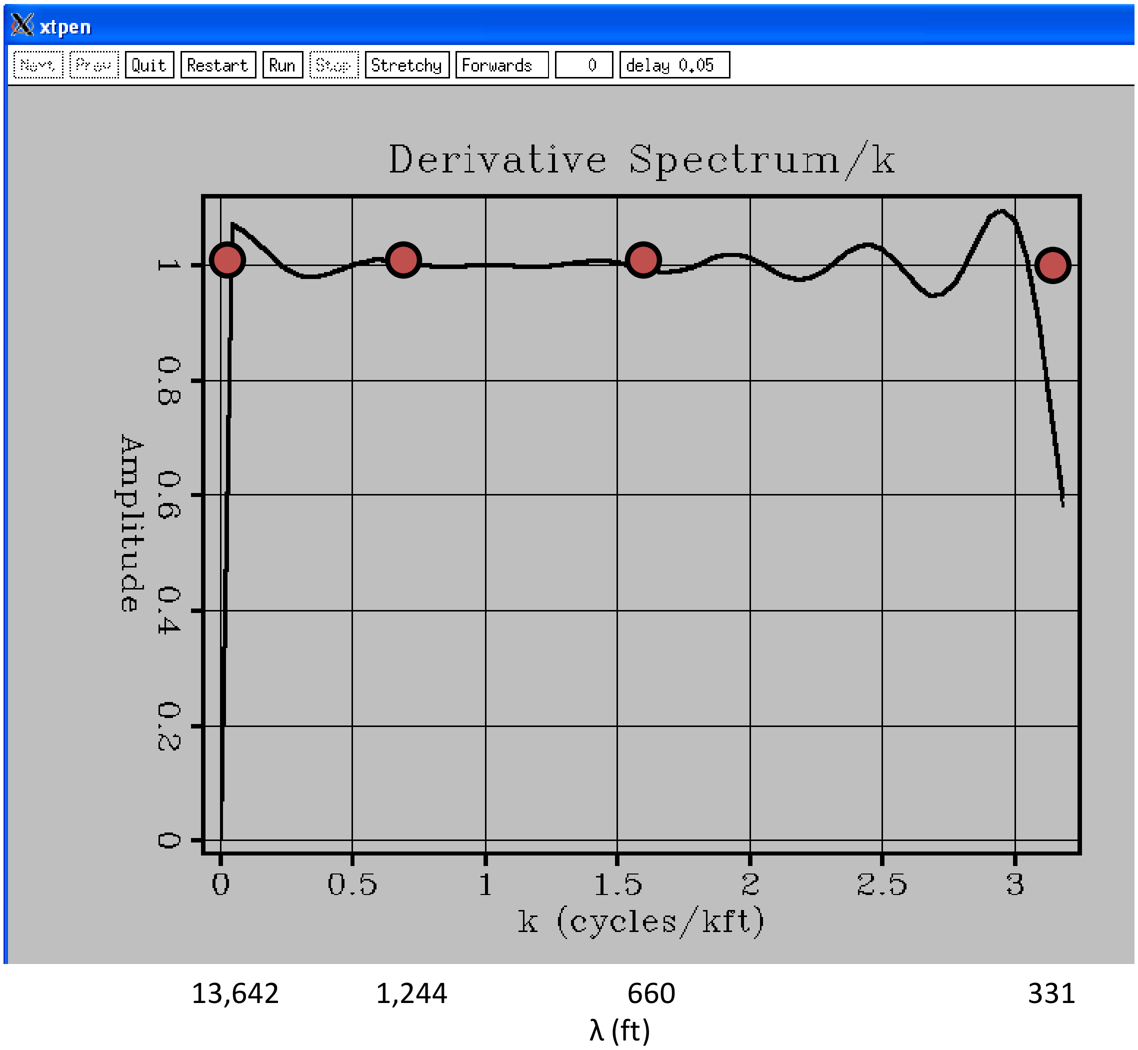
Part of the oscillation is associated with the software attempting to avoid going beyond Nyquist by applying a 'boxcar' filter in the wavenumber domain that transforms to a sinc function in the space domain. Even so, recall that the 10th order finite difference operator (five points on either side of 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:

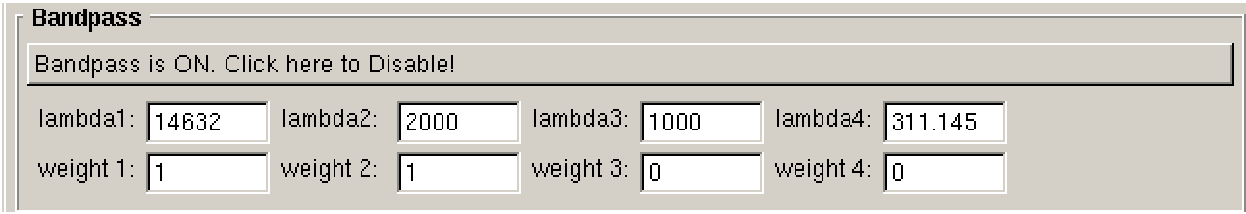

xtpen
Next
Prev
Quit
Restart
Run
Stop
Stretchy
Forwards
0
delay 0.05
Derivative Spectrum/k
Amplitude
0
0.2
0.4
0.6
0.8
0
0.5
1
1.5
2
2.5
3
k (cycles/kft)
13,642
1,244
660
331
λ (ft)


xpen
Next
Prev
Quit
Restart
Run
Stop
Stretchy
Forwards
0
delay 0.05
Derivative Spectrum/k
Amplitude
0
0.2
0.4
0.6
0.8
1
0
0.5
1
1.5
2
2.5
3
k (cycles/kft)
13,642
1,244
660
331
λ (ft)



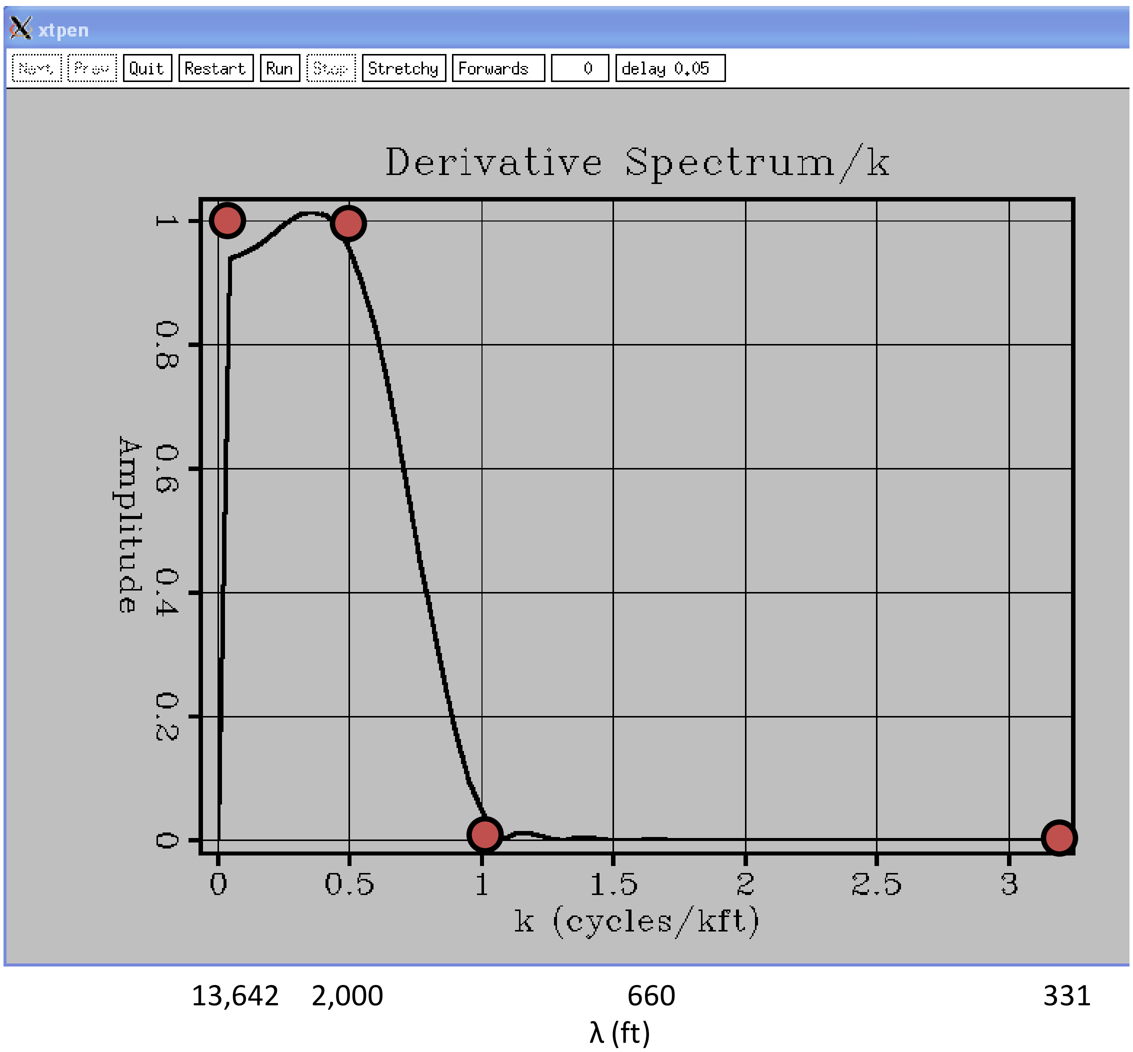
In all three of these images I have added the corresponding wavelength λ along the horizontal axis and the weights entered at *lambda1* through *lambda4* as red circles. Each spectrum has been normalized by 1/k. Since the first derivative of a continuous media can be perfectly approximated in the frequency 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:

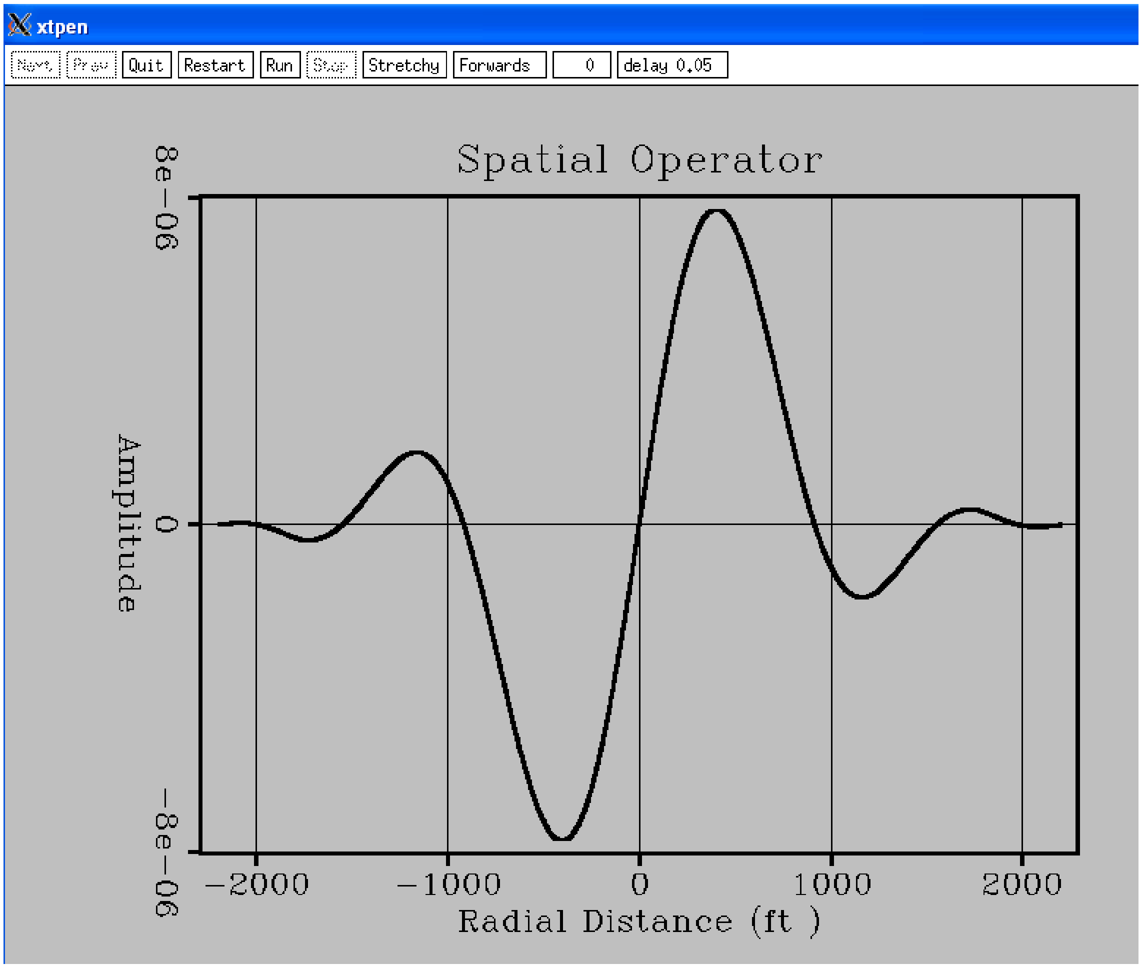


where now, no wavelengths are allowed smaller than 1000 ft. (the *weight3*=0.0 and *lambda3*=1000).

The spectrum looks like this:



while the operator looks like this:

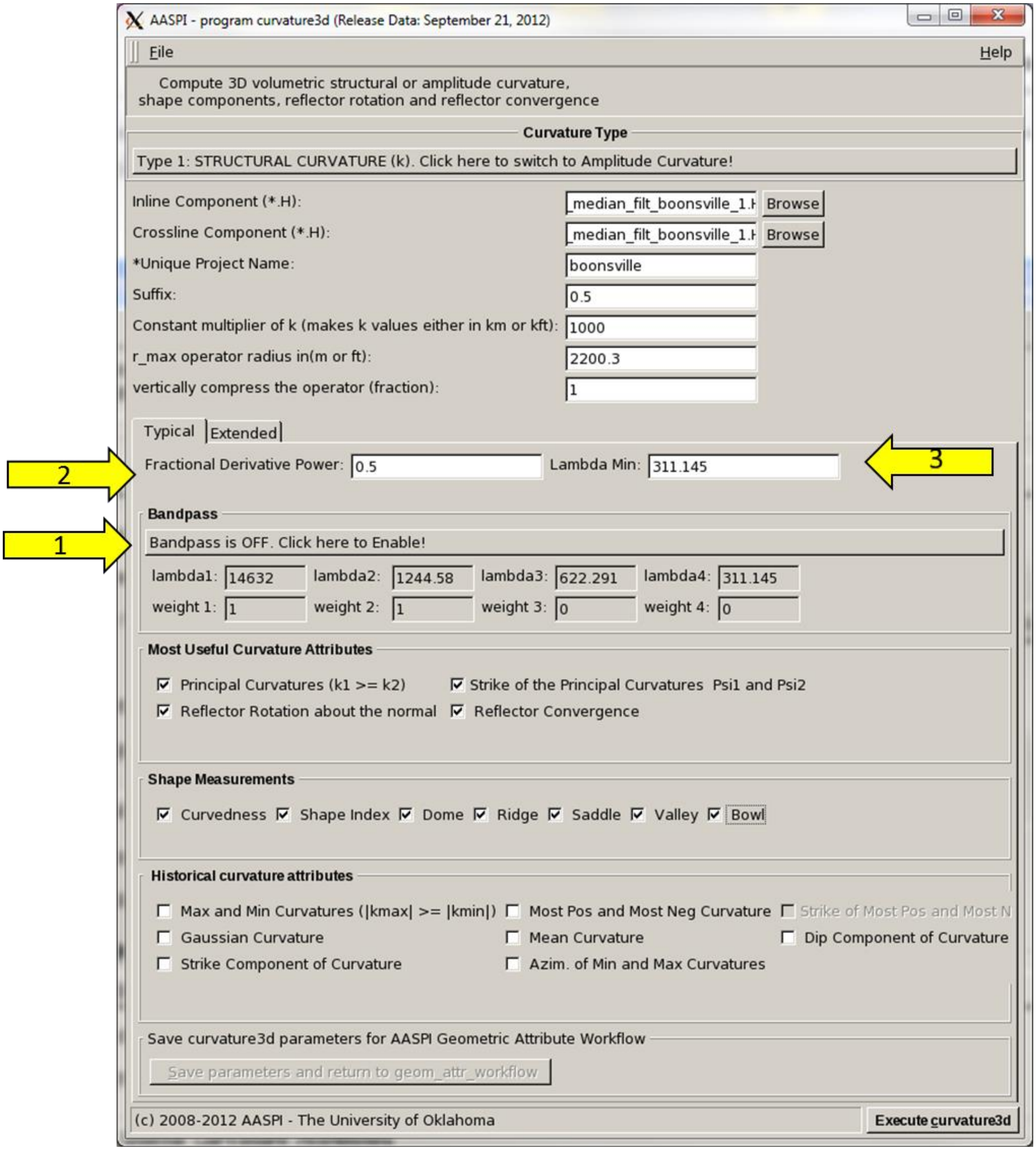


giving the following image at t=1.1 s:

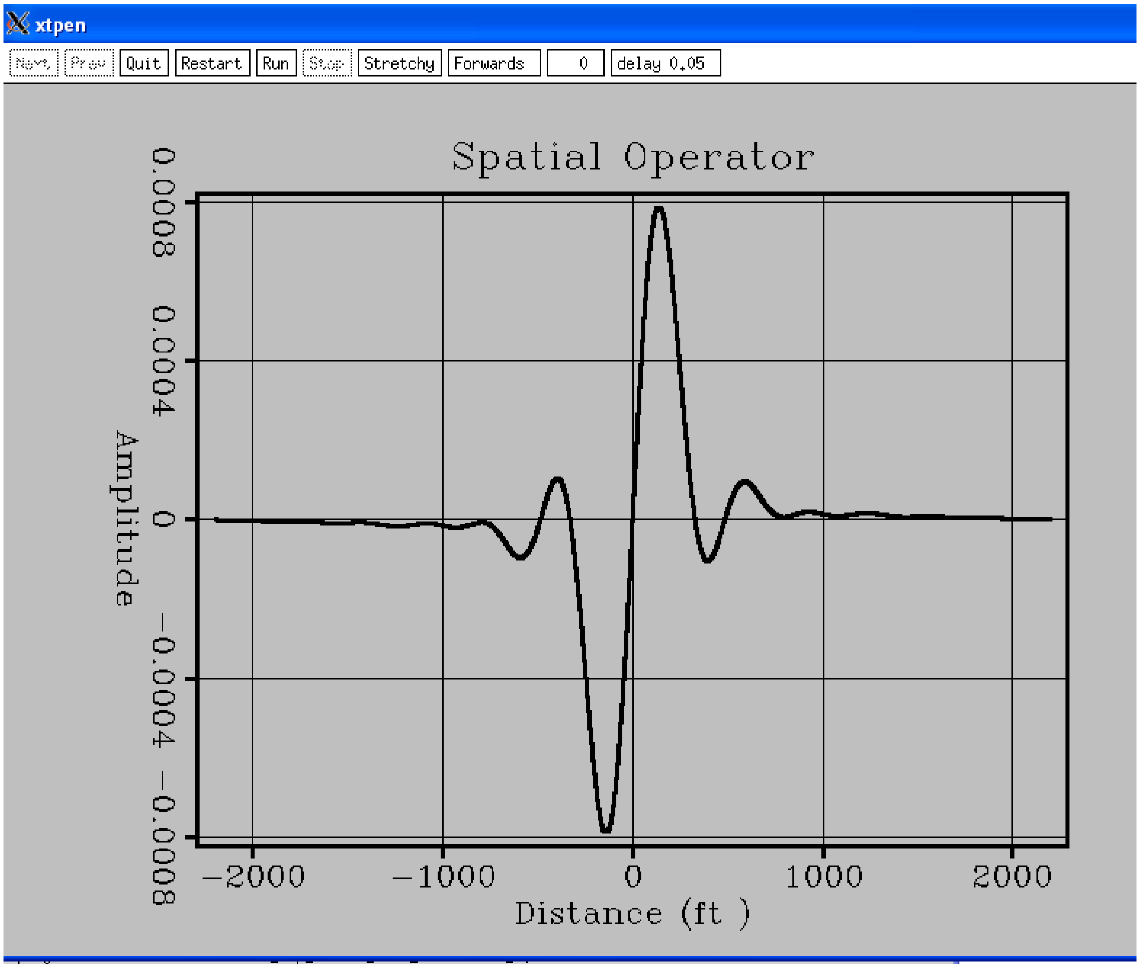
While there may be some value of such low-pass filtered data for statistical analysis, it is clearly inferior to the previous relatively broad band 'long wavelength' version of most-negative principal curvature for purposes of structural or stratigraphic interpretation.

Fractional Derivative Operators

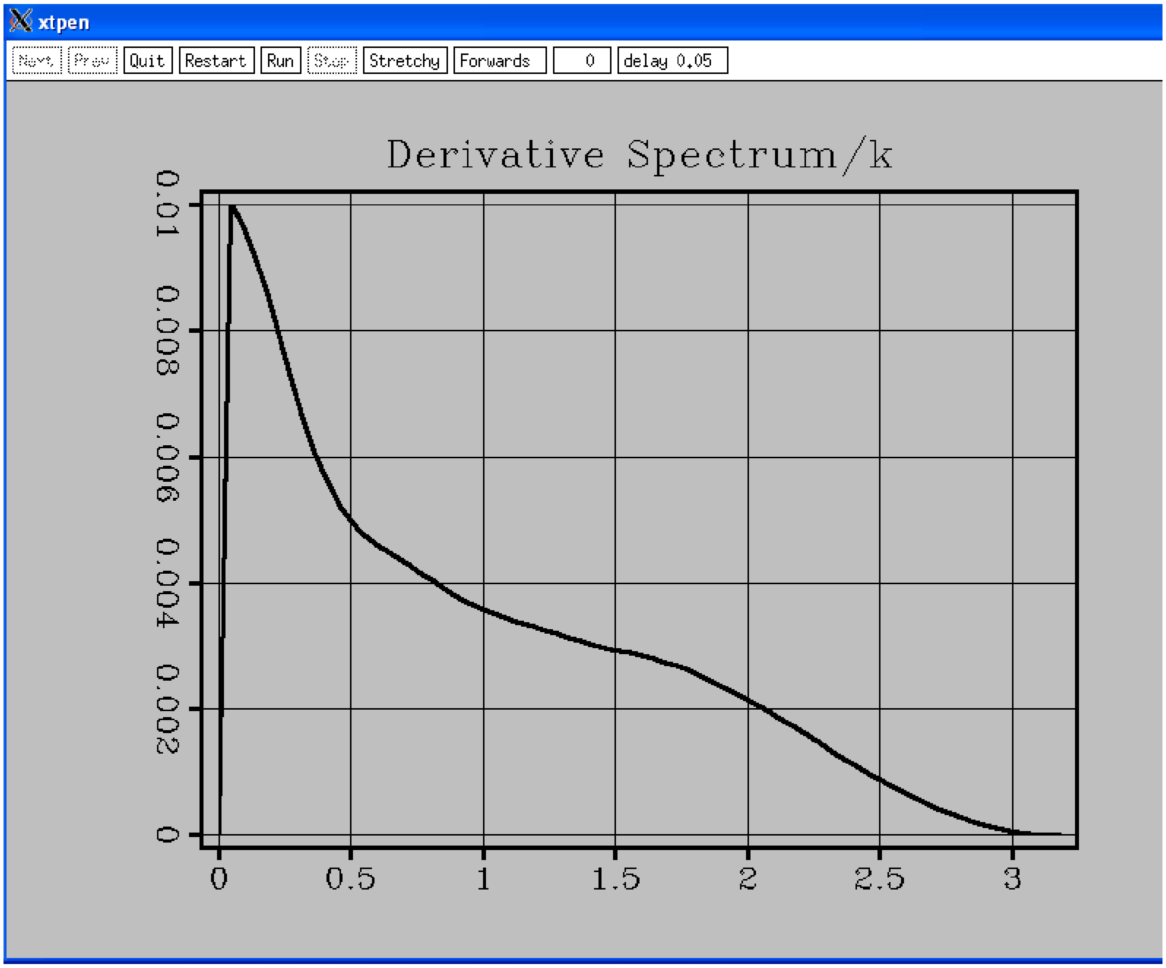
The above discussion, while perhaps somewhat tedious, is very straightforward for a processing geophysicists. Al-Dossary and Marfurt (2006) are perhaps responsible for introducing the concept of fractional derivatives to the curvature literature. Their work was inspired by high quality images generated 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 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 α=0.5. Do not set (3) the value of Lambda Min to be smaller than Nyquist. The resulting spatial operator looks as follows:

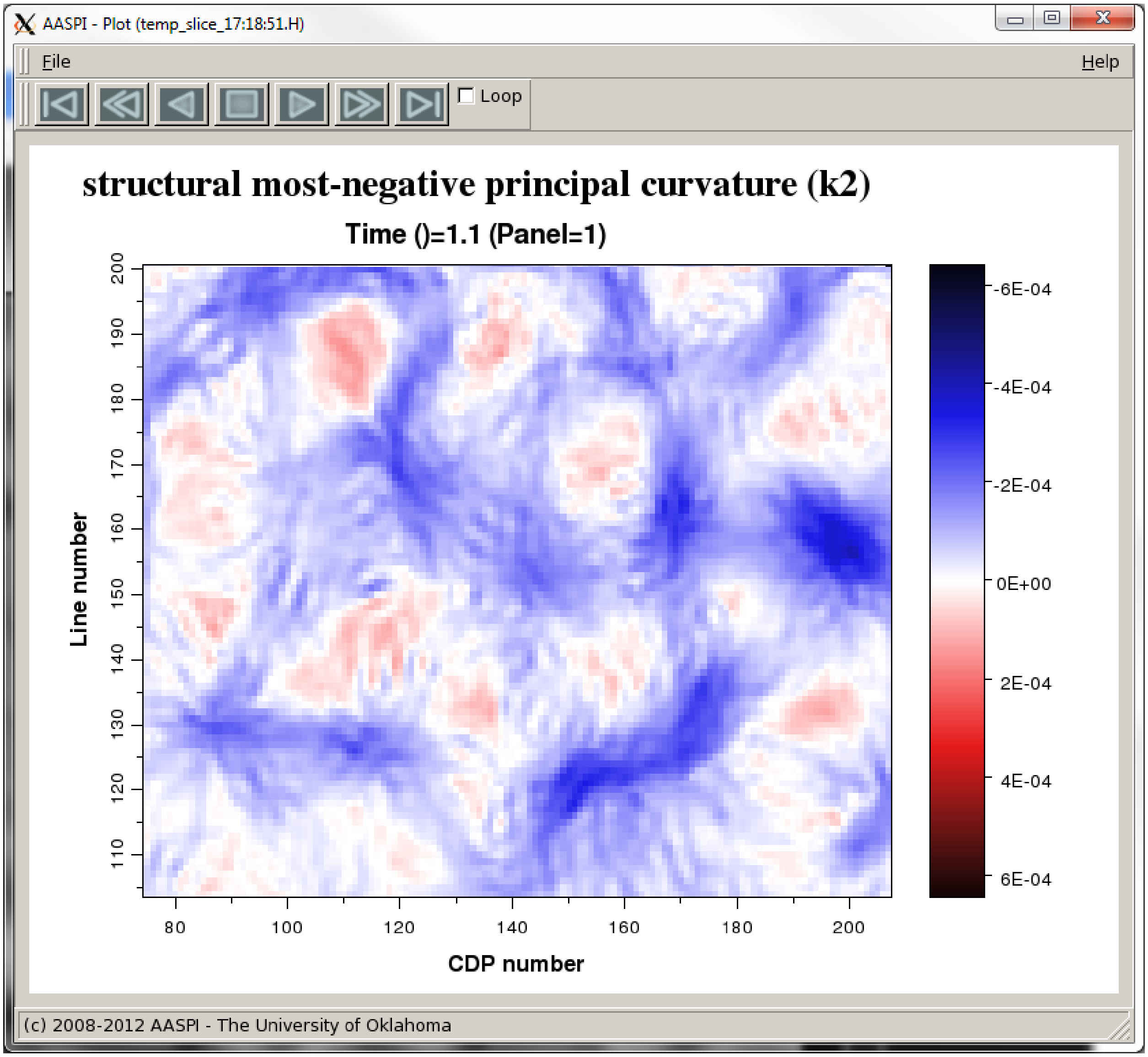


while the corresponding spectrum appears like this:

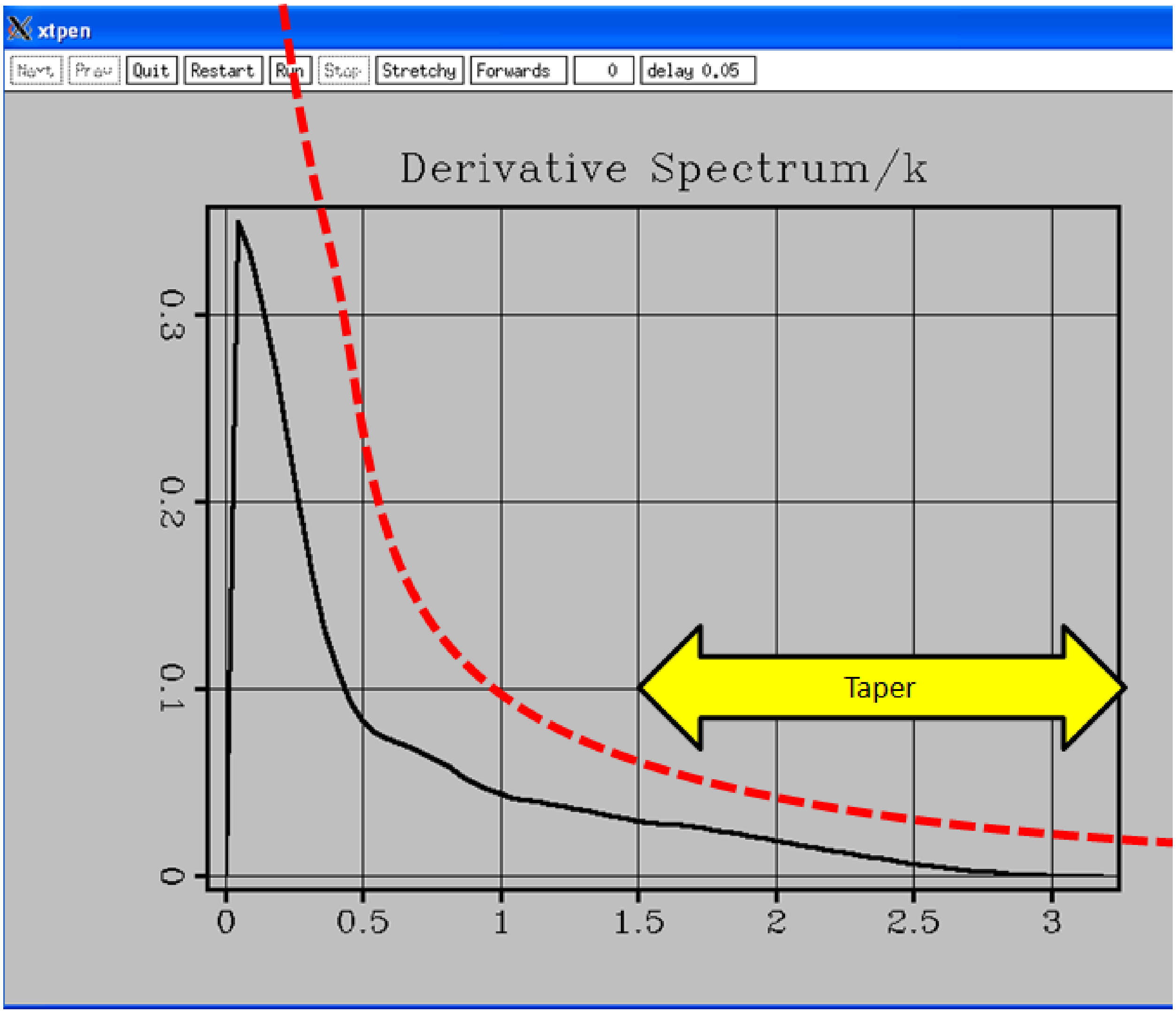


Note that with the exception of the vertical scale, the shape of both the spectrum and the operator look very similar to the long wavelength filter discussed earlier.

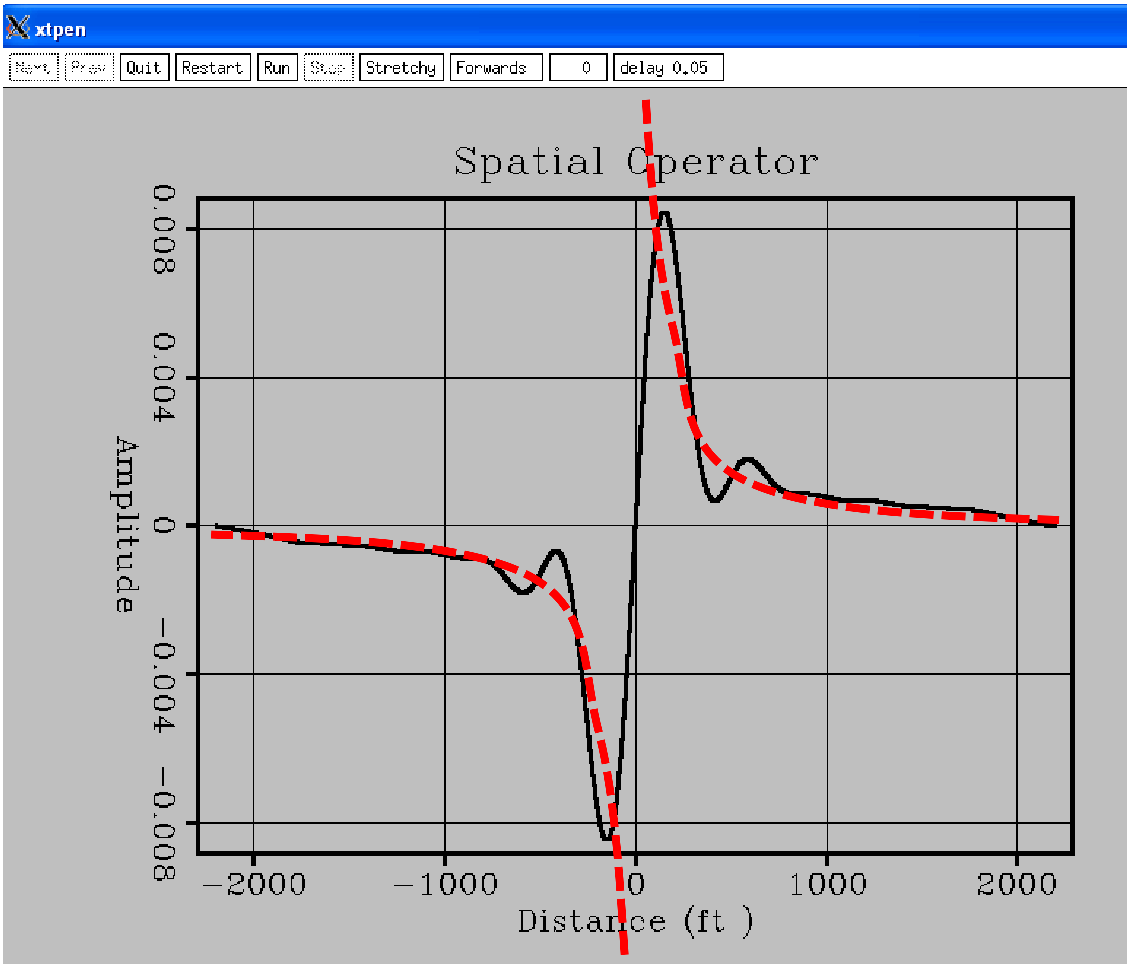
The time slice through the most-negative principal curvature at t=1.1 s looks like this:



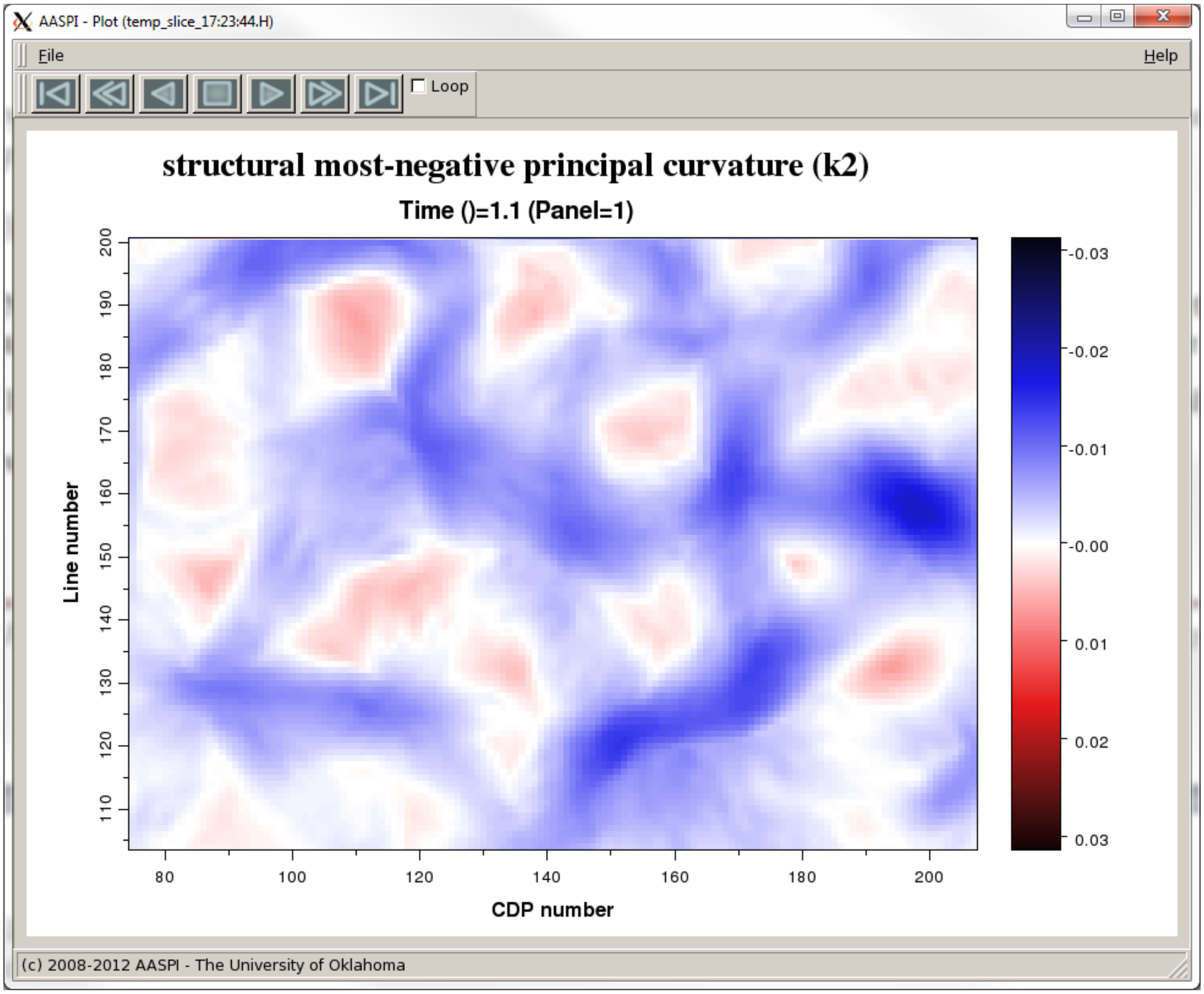
If we push the fractional derivative concept to its endpoint, with the *Fractional derivative power* = 0.0, we should get an operator that looks like $ik^{0.0} = i$ which is equivalent to a Hilbert transform. Normalizing by *1/k*, we obtain the following spectrum



There are two limitations the AASPI software applies to any desired spectrum. First, all data are band-limited, with the minimum wavelength for these data, λ_{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 λ_{taper} =2λ_{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 ±*20*Δx* by ±*20*Δy* (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. The corresponding operator looks like this



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



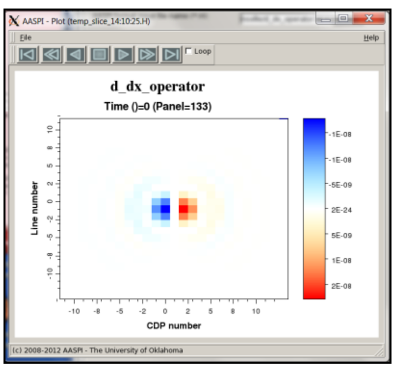
which is somewhat longer wavelength than image generated using a *Fractional derivative power*=0.50 .

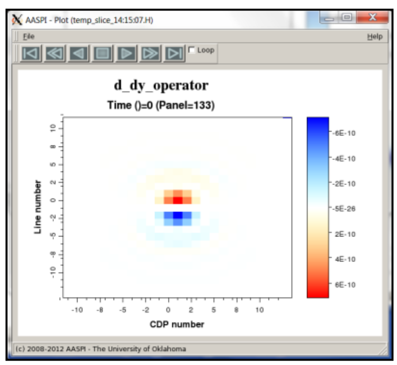
Implementation of Operators in 3D

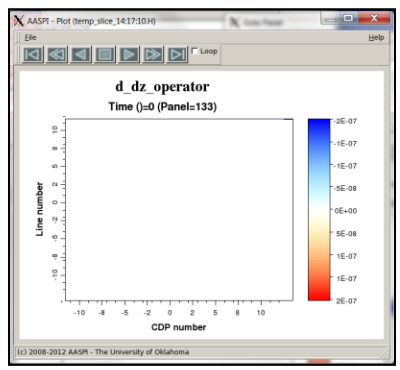
Now that we know how to generate and quality control the derivative operators and their normalized spectra, we can now examine how they are applied in a 3D sense. If we define our 1D operator above as $\frac{\partial}{\partial r}$, where $r=[x^2+y^2+z^2]^{1/2}$ is the distance of any point in the analysis window to the center point, we define the three spatial operators as

$$\frac{\partial}{\partial x} = \frac{x}{r}\frac{\partial}{\partial r},$$
$$\frac{\partial}{\partial y} = \frac{y}{r}\frac{\partial}{\partial r}, \text{ and}$$
$$\frac{\partial}{\partial z} = \frac{z}{r}\frac{\partial}{\partial r}$$

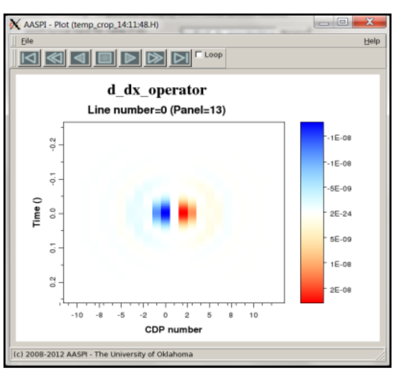
To examine these operators, use the *AASPI QC* Plotting tab in **aaspi_ut**il to display the file *d_dx_operator3d_boonsville_long_wavelength.H, d_dx_operator3d_boonsville_long_wavelength.,* and *d_dx_operator3d_boonsville_long_wavelength.H* . Time slices at time t=0 through the operator produces the following images:

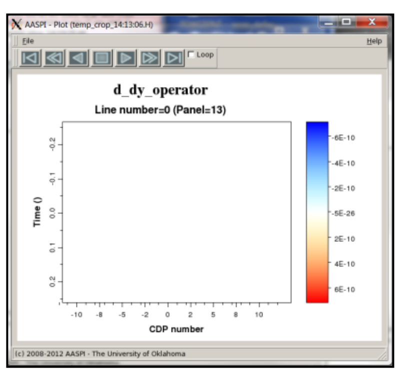


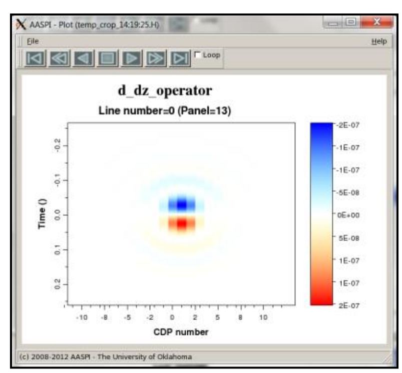




Note that the slice through d/dz operator is zero along *t*=0.0 s, since it defines a plane of symmetry through an antisymmetric operator. The d/dx operator is rotationally symmetric about the x-axis, the d/dy operator rotationally symmetric about the y-axis, and the d/dz operator rotationally symmetric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

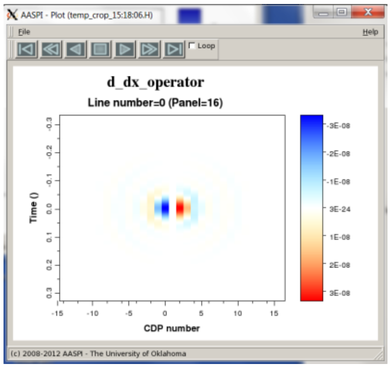


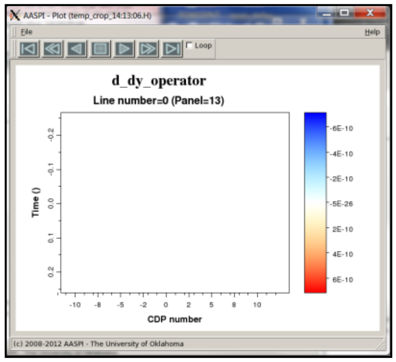


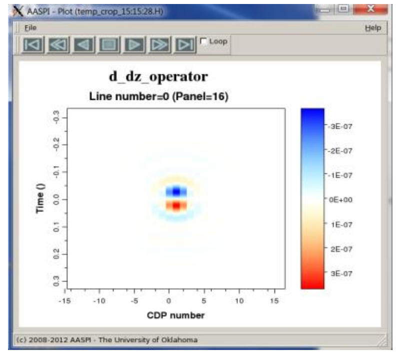


where now the d/dy operator is zero along its plane of symmetry, *y*=0.0 .

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





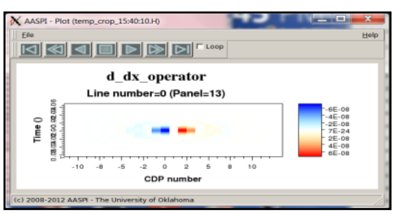


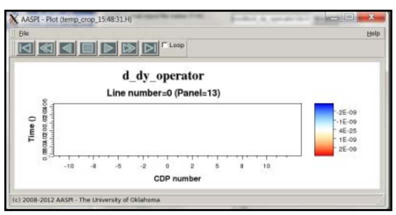
Careful inspection will show that this latter operator is a little shorter in all three dimensions.

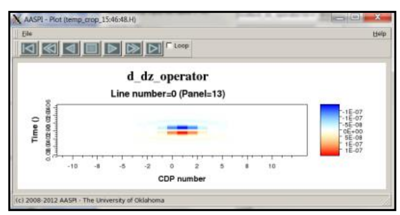
Note that the extent of the short-wavelength and long-wavelength operators is about the same, but that the long-wavelength operator is more monotonic, while the short wavelength operator is slightly more oscillatory. Also note that the operators are 3D, and are not restricted to the inline, crossline, 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 45^0 and obtained identical curvature images.

Compressing the vertical operator to reduce vertical mixing

For fairly flat geology with dips less than 15^0 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:

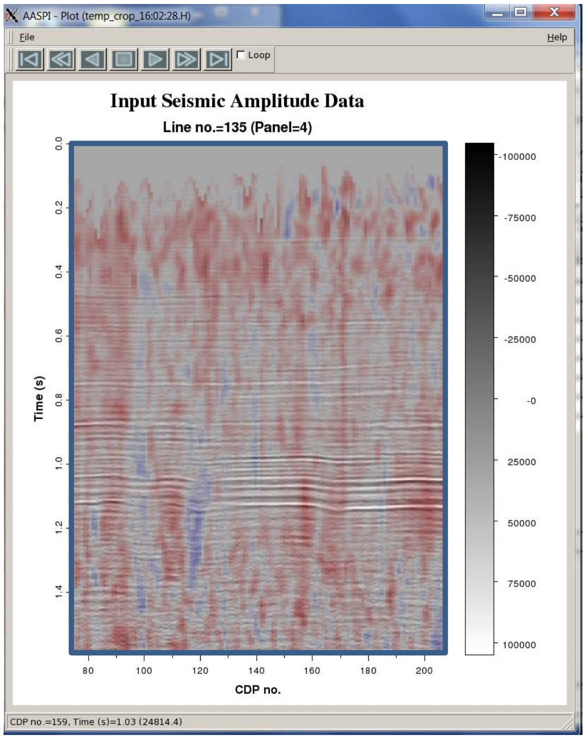


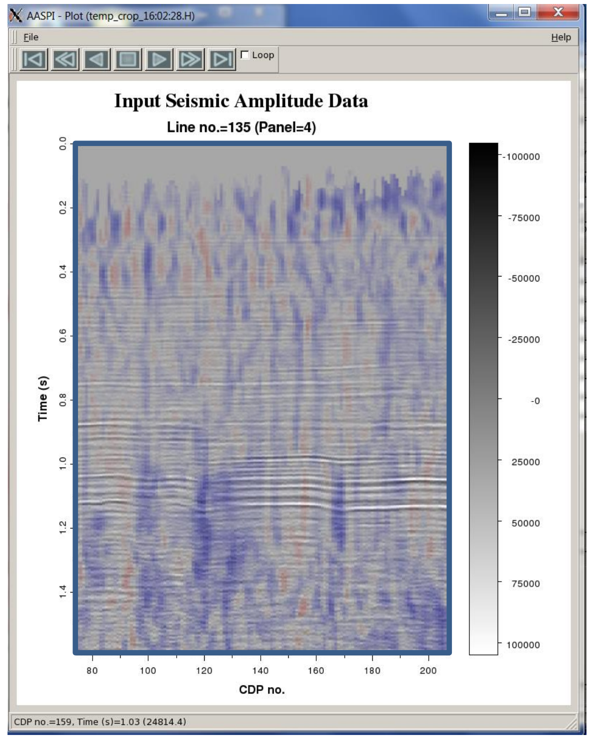


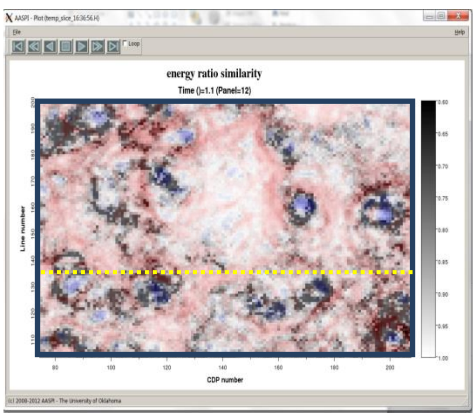


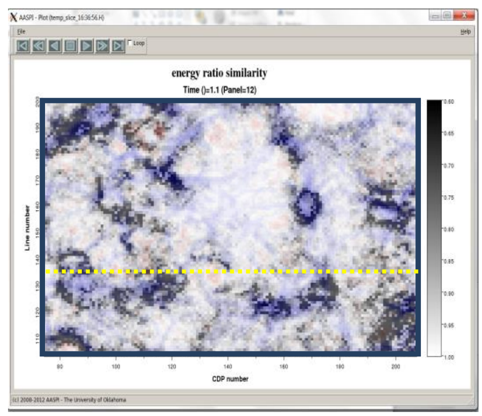
Let's examine the impact of changing the vertical size of the operator on the Boonsville data volume, beginning with the isotropic (*vcompress*=1.0) operator:

Volumetric Attributes- Curvature3d

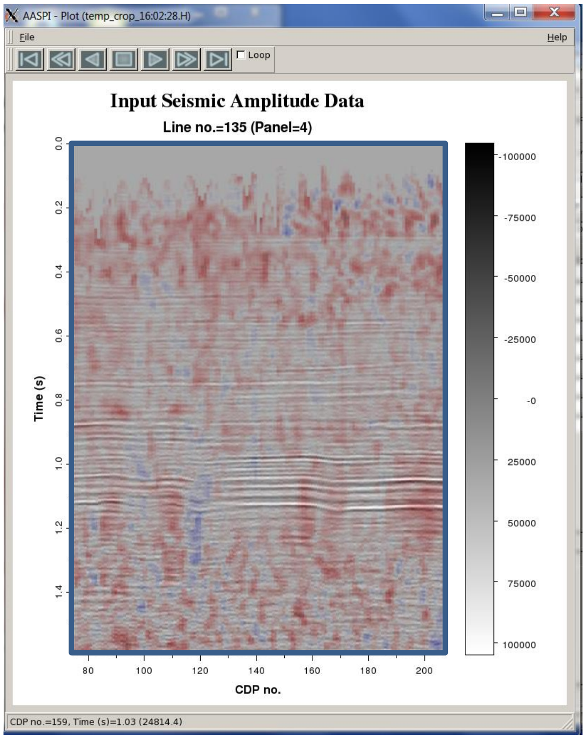


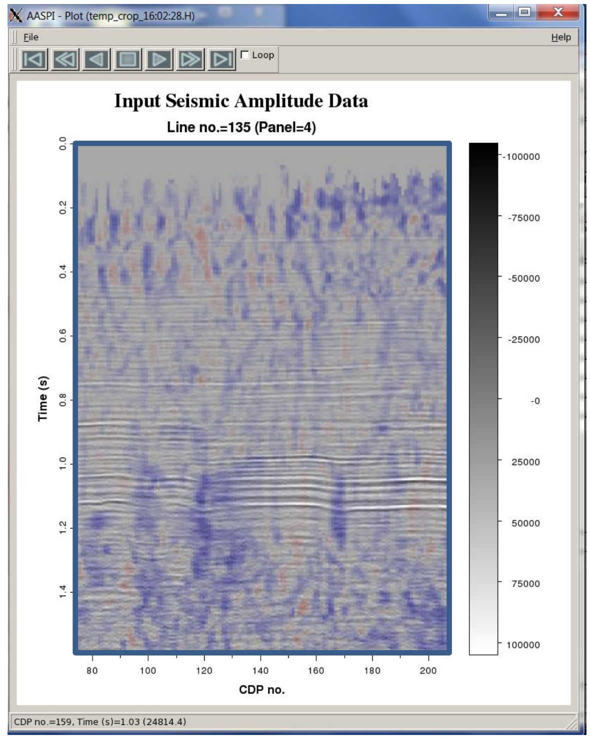


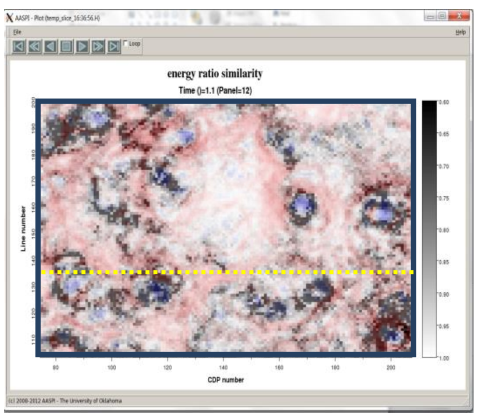


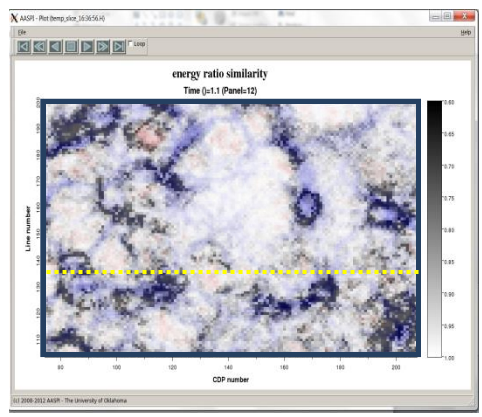


Repeating the computation with *vcompress*=0.50 gives

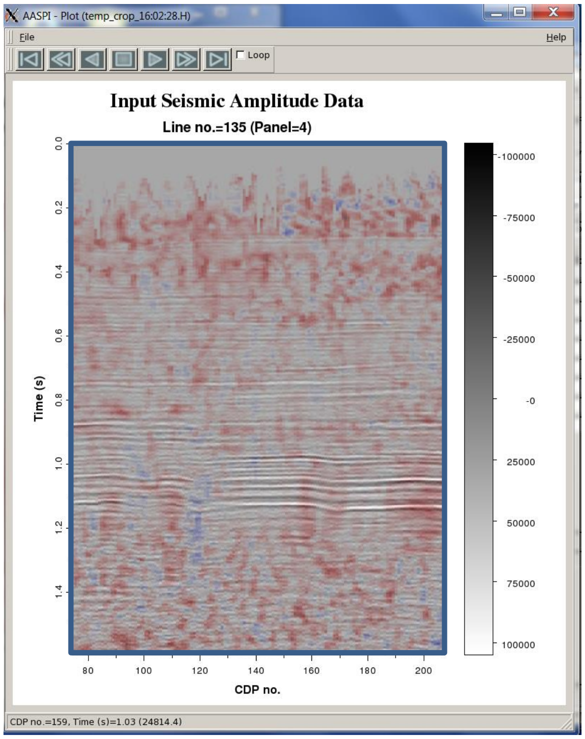


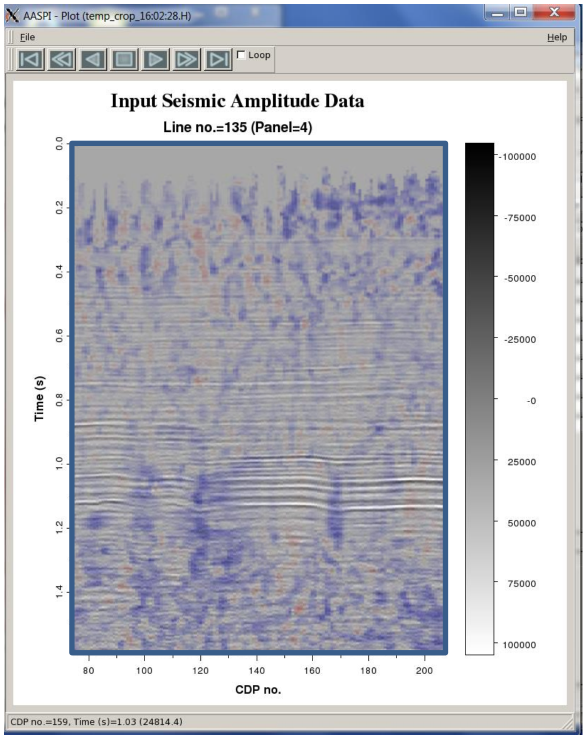


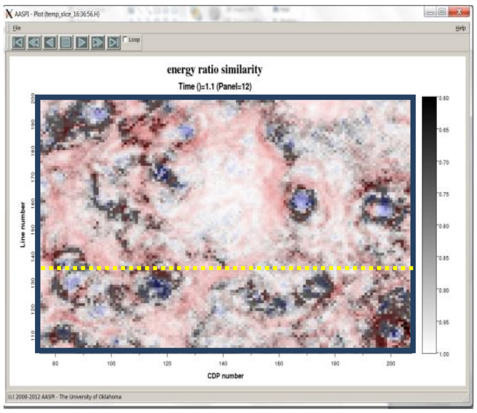


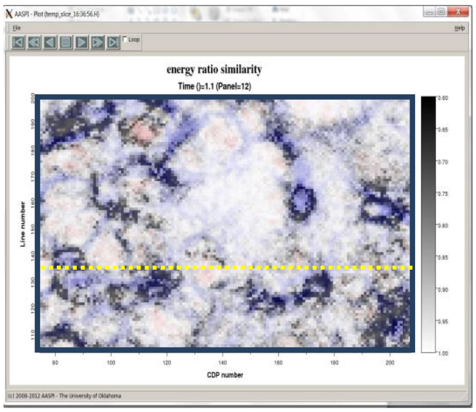


Finally, setting a value of *vcompress*=0.25 gives





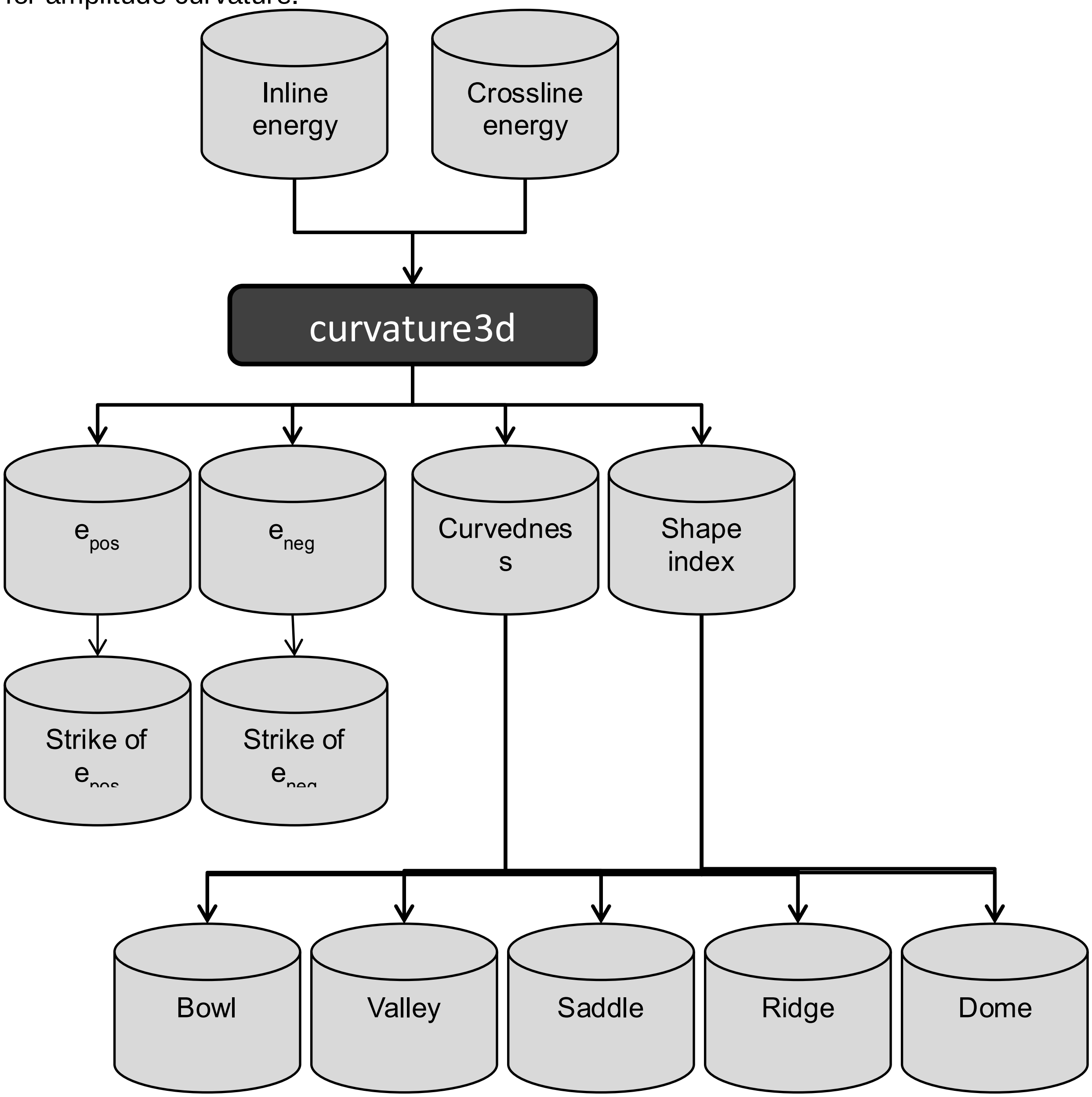




Comparing these three results we note that compressing the vertical operator limits the leakage of strongly deformed reflectors from overprinting those above and below. Philosophically, one might think of using different wavelengths to compute curvature – with long-wavelength operators in the more gently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

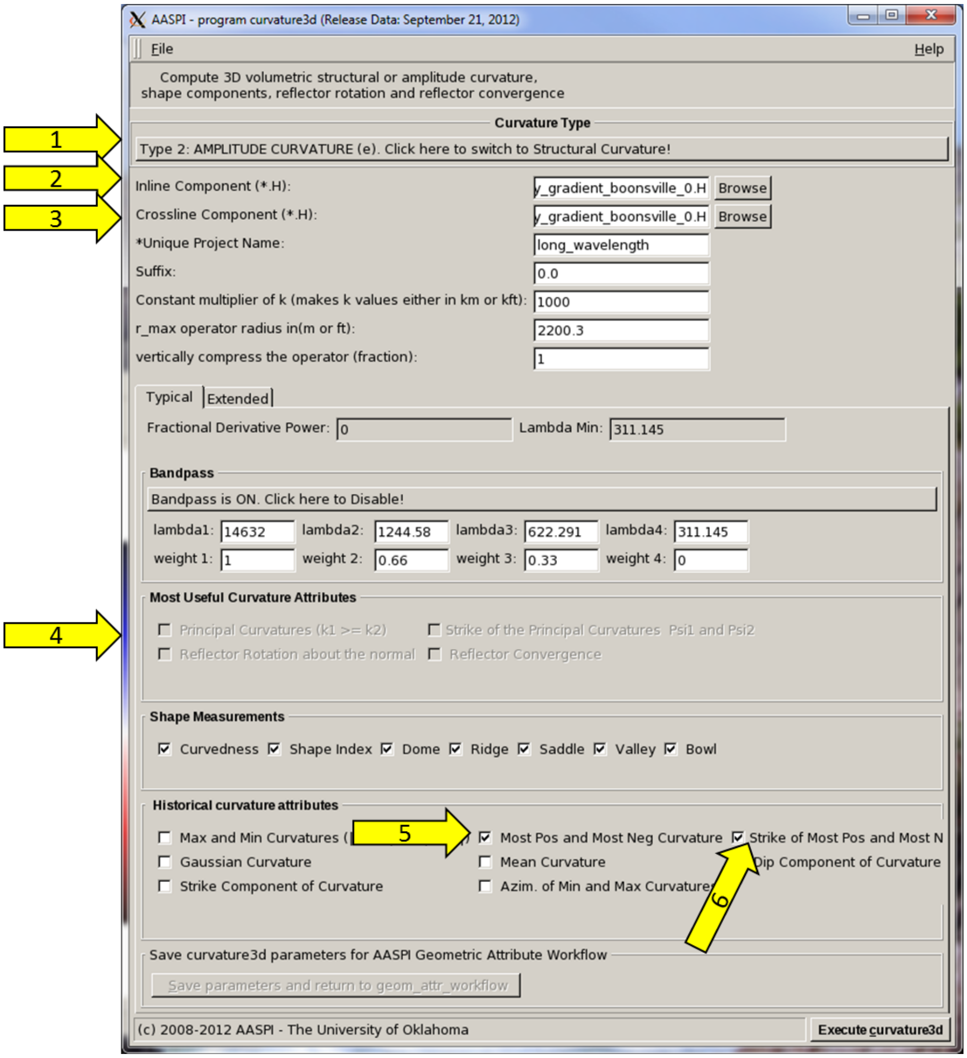
Computation flow chart for *amplitude* curvature

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 uses measures the dip in m/m or ft/ft (through the conversion of tanθ) 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.

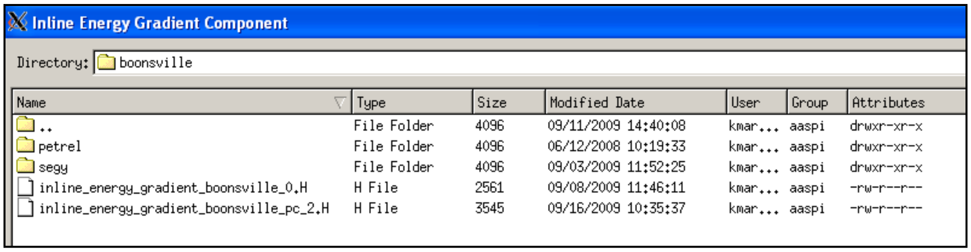


Computing amplitude curvature

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* .

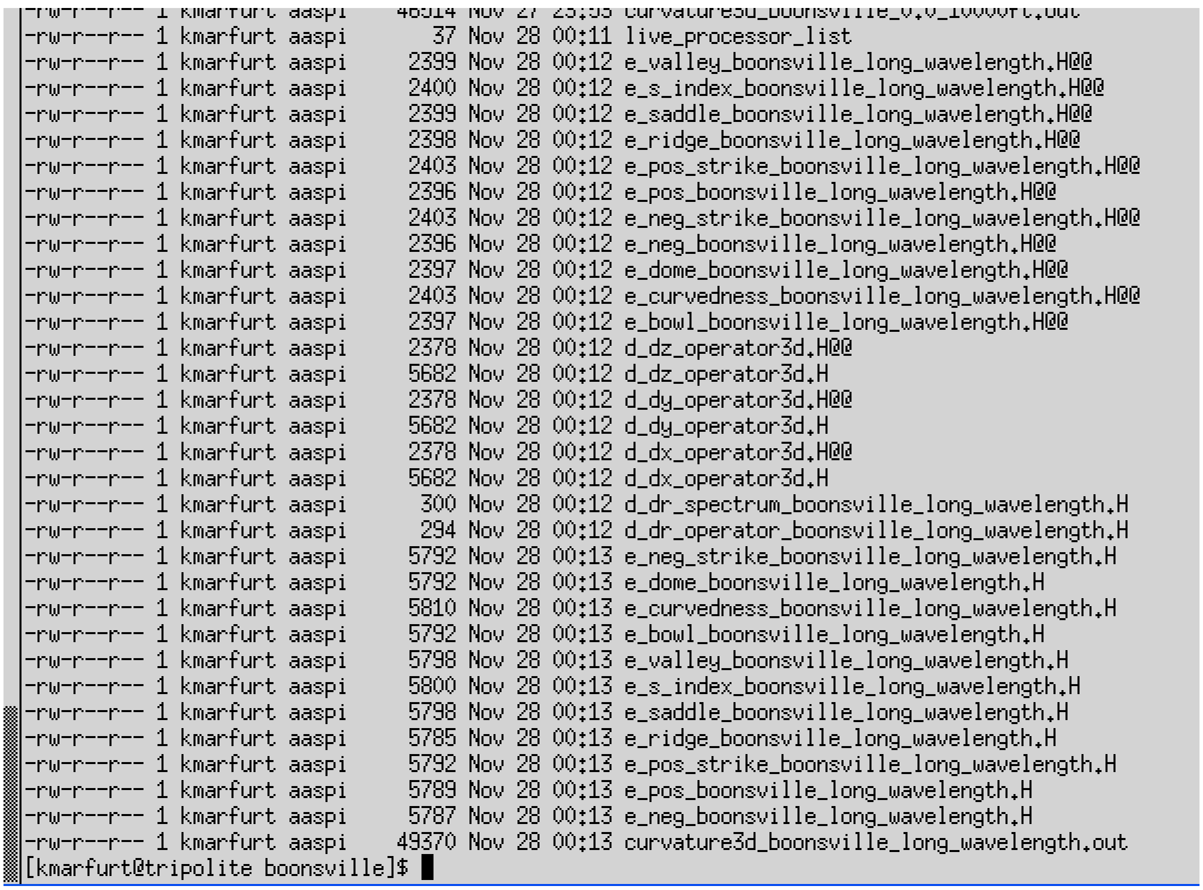


To compute amplitude curvature, (1) click the curvature type button and bring up *Type 2: AMPLITUDE CURVATURE (e)* .While the structural curvature attributes begin with the letter ‘k’, the amplitude (or energy) curvature attributes will begin with the letter ‘e’. When you (2) Browse to select the inline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

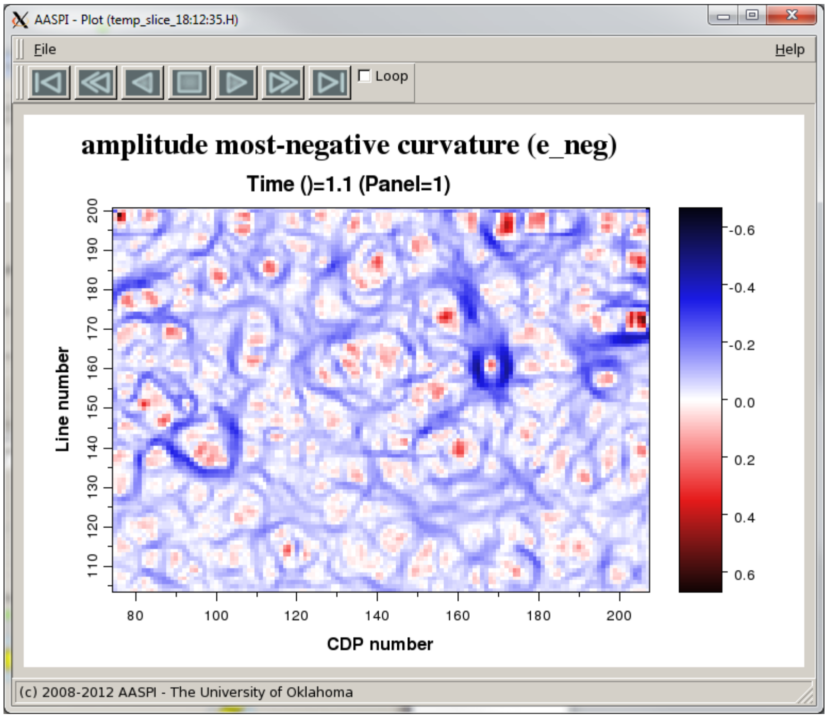


Select *inline_energy_gradient_boonsville_0.H.* Then (3) repeat the process for the crossline component. The design of the derivative operators is exactly the same as for structural curvature. However, certain attributes are not well-defined. Amplitude gradients do not exist in 3D Euclidean space. The horizontal dimensions are still spatial, measured in feet or meter, but the 'vertical' dimension is now in amplitude (measured in mV or some other unit). For this reason, you are not allowed to compute (4) the k_1 and k_2 principal curvatures. In contrast, (5) the most-negative and most-positive curvatures e_{neg}, and e_{pos} are still well-defined, so place a checkmark in front of them. Likewise, the strike of these amplitude lineaments is also well defined. Finally, click *Execute* and wait for the job to complete.

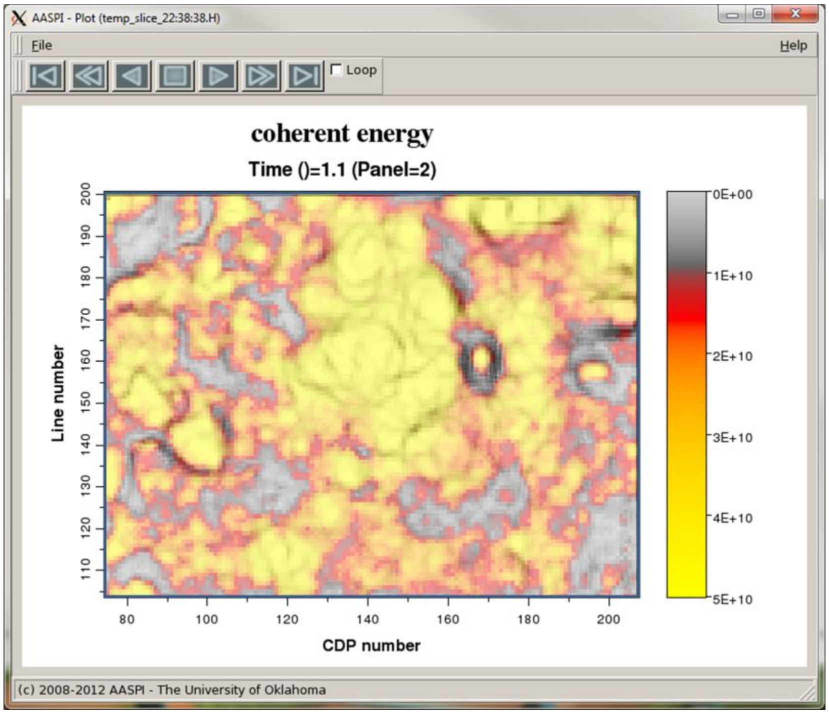
The output files will look like



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



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's use PowerPoint to blend e_{neg} and the coherent energy images:

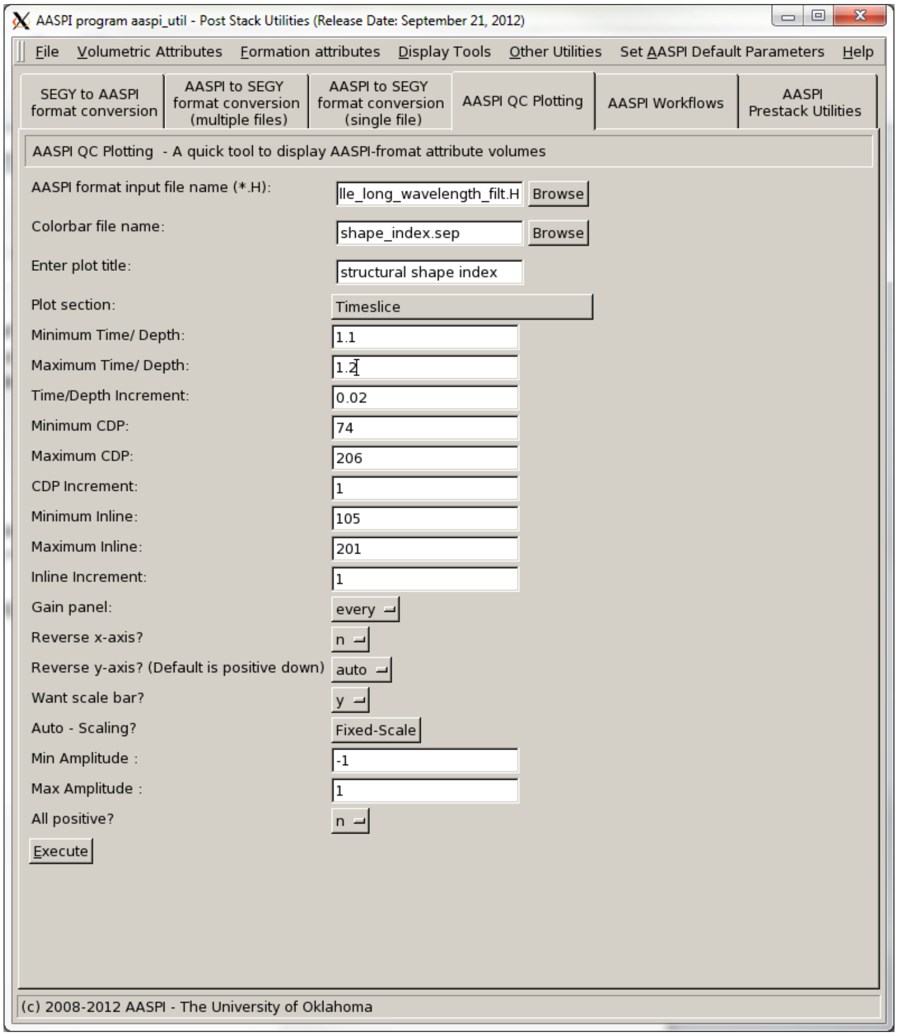


Not surprisingly, the low values of e_{neg} track the low (dark blue) values of the coherent energy. However, these lineaments continue across the higher energy areas of the time slice, forming a network. We have found e_{neg} and e_{pos} attributes to be very effective in mapping fractures (or very 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_s_index_boonsville_long_w_filt.H* using the following AASPI QC Plottting tab parameters:


AASPI program aaspi_util - Post Stack Utilities (Release Date: September 21, 2012)
File Volumetric Attributes Formation attributes Display Tools Other Utilities Set AASPI Default Parameters Help
SEGY to AASPI format conversion
AASPI to SEGY format conversion (multiple files)
AASPI to SEGY format conversion (single file)
AASPI QC Plotting
AASPI Workflows
AASPI Prestack Utilities
AASPI QC Plotting - A quick tool to display AASPI-fromat attribute volumes
AASPI format input file name (*.H):
lle_long_wavelength_filt.H
Browse
Colorbar file name:
shape_index.sep
Browse
Enter plot title:
structural shape index
Plot section:
Timeslice
Minimum Time/ Depth:
1.1
Maximum Time/ Depth:
1.2
Time/Depth Increment:
0.02
Minimum CDP:
74
Maximum CDP:
206
CDP Increment:
1
Minimum Inline:
105
Maximum Inline:
201
Inline Increment:
1
Gain panel:
every
Reverse x-axis?
n
Reverse y-axis? (Default is positive down)
auto
Want scale bar?
y
Auto - Scaling?
Fixed-Scale
Min Amplitude :
-1
Max Amplitude :
1
All positive?
n
Execute
(c) 2008-2012 AASPI - The University of Oklahoma

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

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

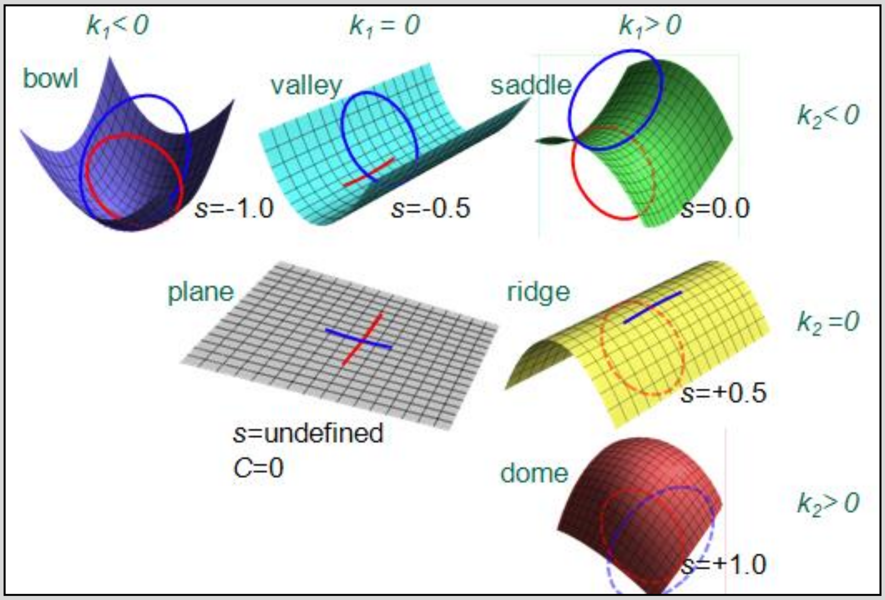
where k_1 and k_2 are the most-positive and most-negative principal curvatures, with

$$k_1 \geq k_2 .$$

The shape index, s, is defined as

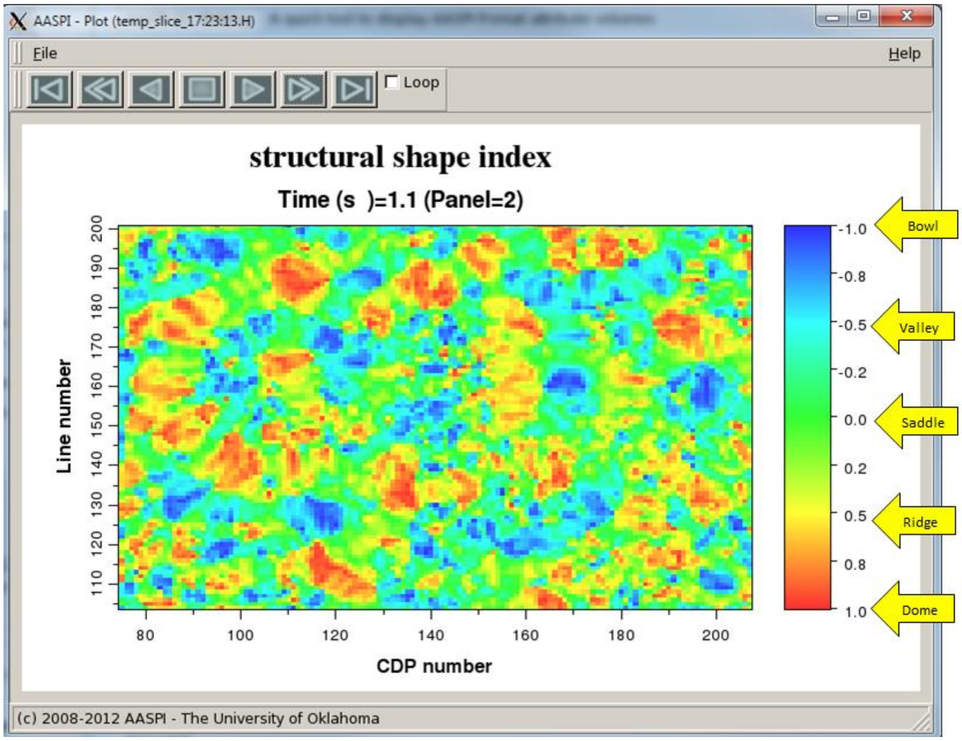
$$s = -\frac{2}{\pi} ATAN\left(\frac{k_2 + k_1}{k_2 - k_1}\right).$$

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:



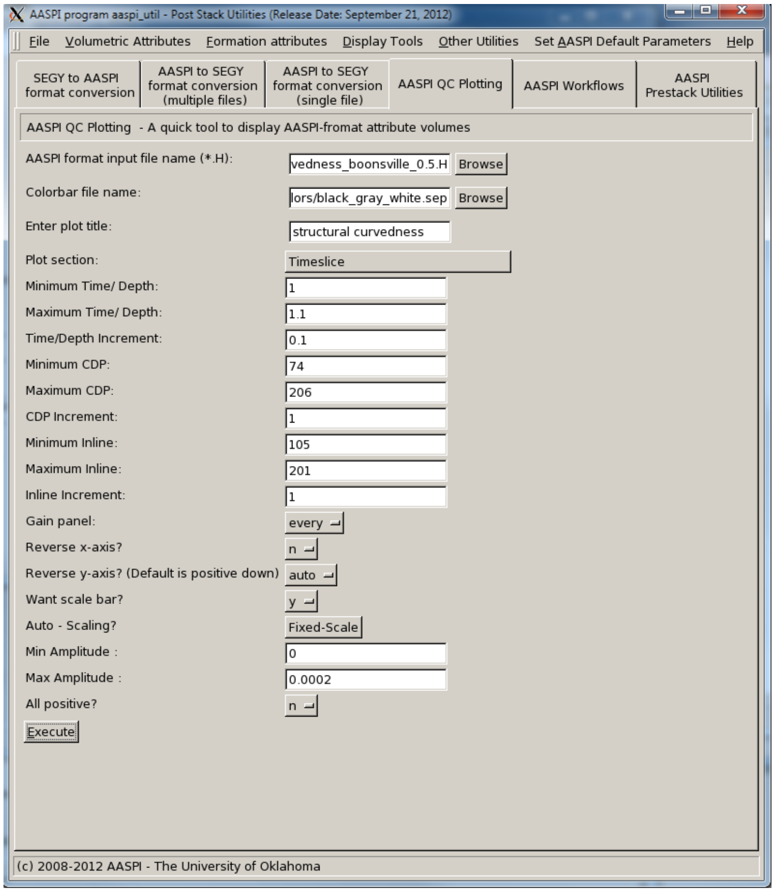
(Figure after Mai, 1999)

Use the *Colorbar 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

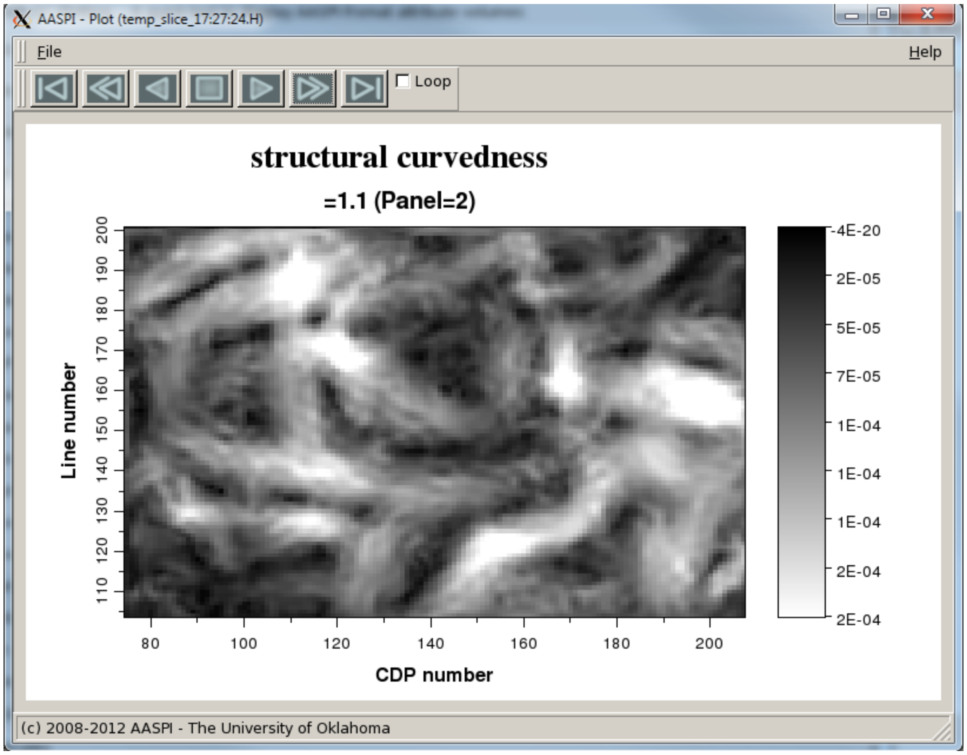


where the arrows indicate the shape on the 1D color bar to the right.

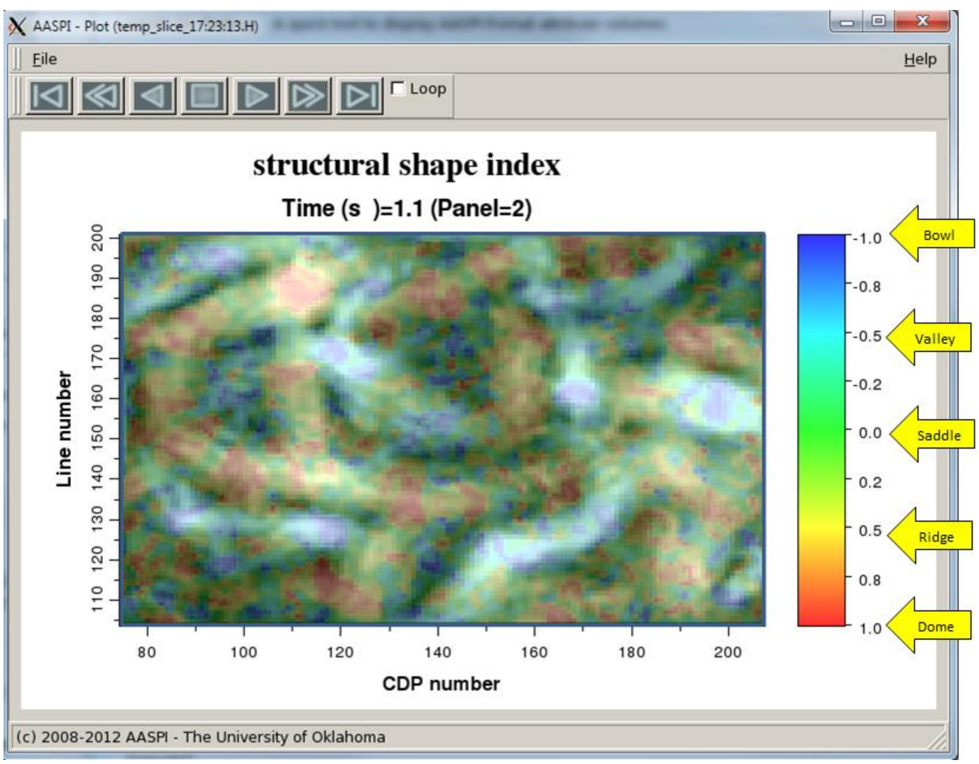
Plot the curvedness file by setting the following parameters on the *AASPI QC Plottting tab:*



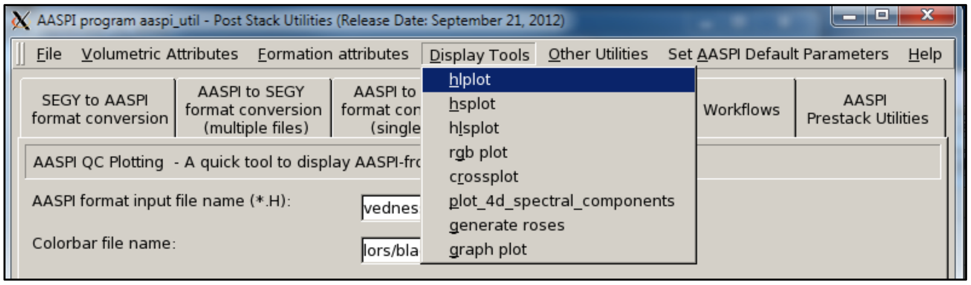
where in addition to selecting the name, *k_curvedness_boonsville_long_w_filt.H*, we have chosen the *Colorbar file* to be *black_gray_white.sep*, and a Min Amplitude value of 0.0 and a Max Amplitude value of 0.0002, since the curvedness is a strictly positive attribute. The resulting image appears like this:



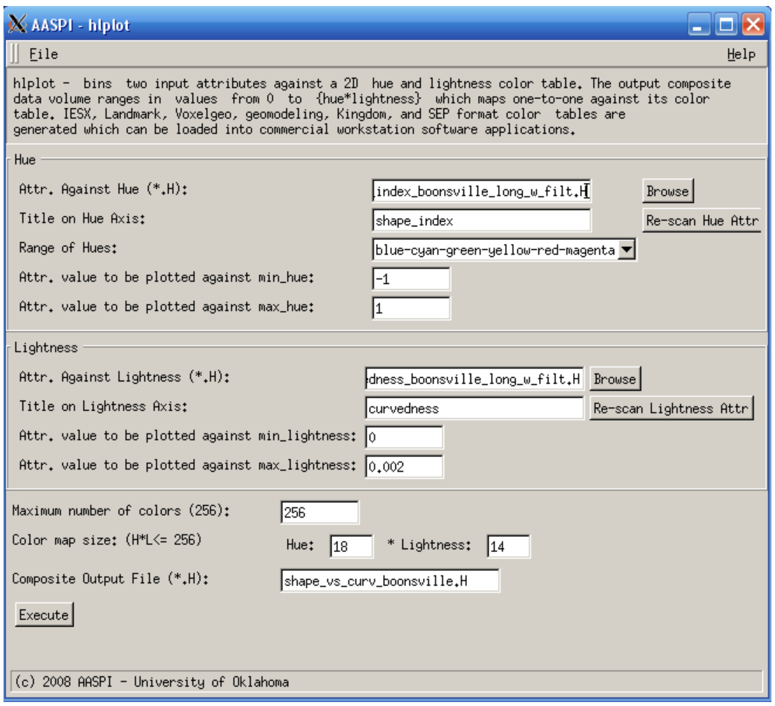
Blending the two previous images in PowerPoint, we obtain:



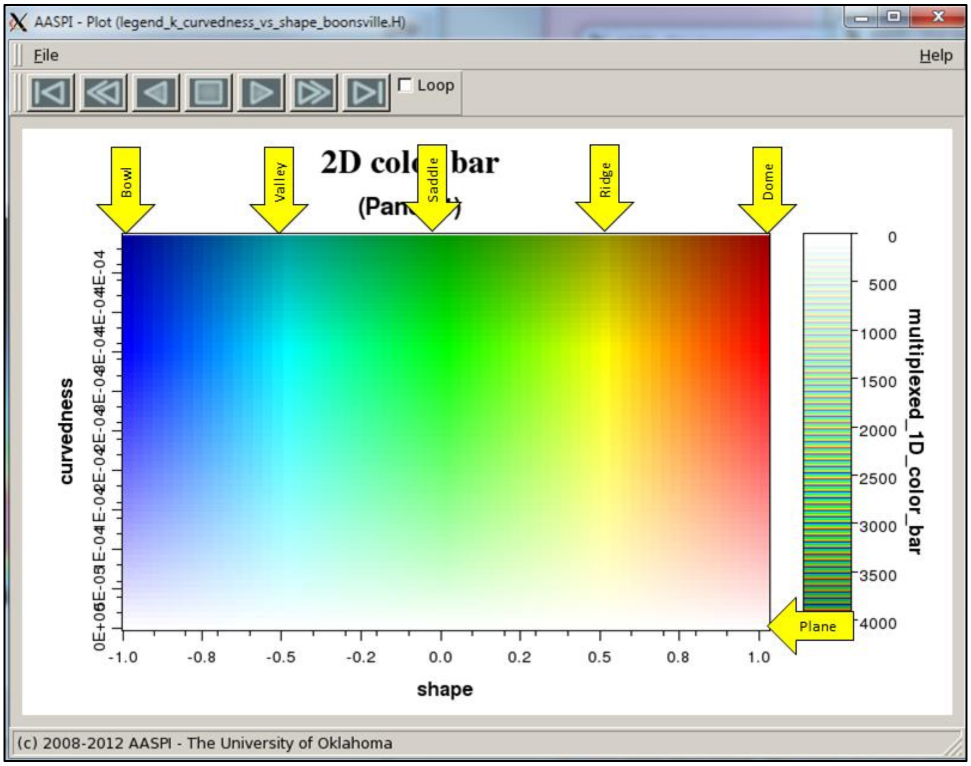
Instead of simply blending the two attributes, we can modulate one by the other using program **hlplot,** found under *Other Tools* on the **aaspi_util** GUI:



The **hlplot** GUI has a layout identical to that of hsplot and looks like this:

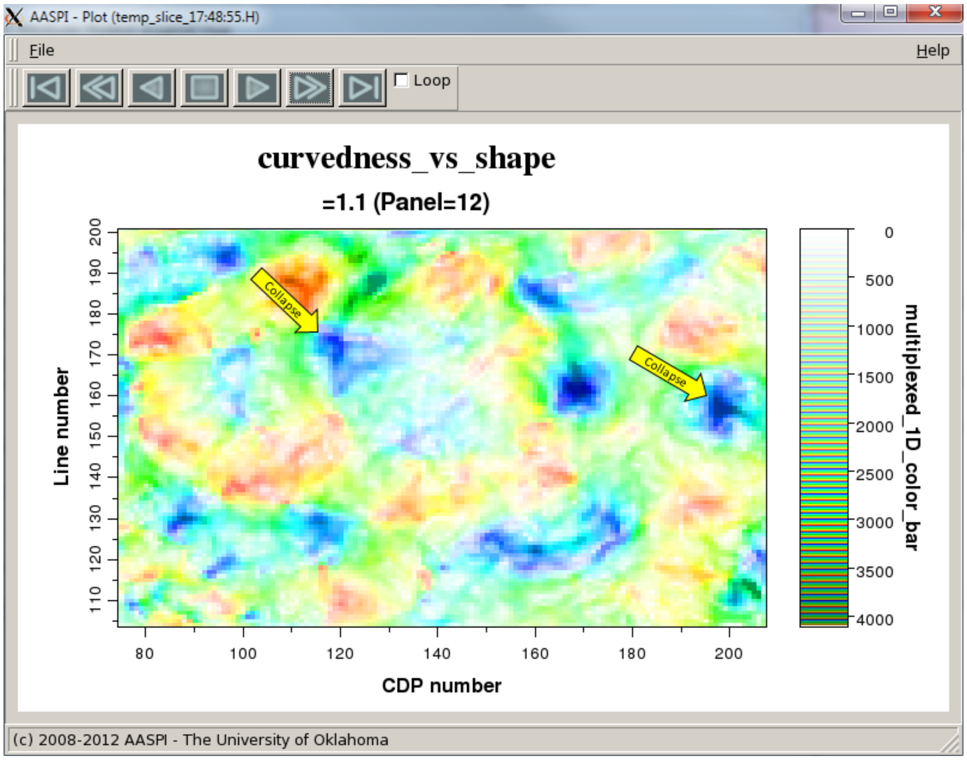


The only difference is that we will plot one attribute (in this case the shape index) against hue, and the other (curvedness) against lightness such that highly deformed features will appear with the appropriate color while planar features appear as white. Select the two input files that we previously plotted using the AASPI QC Plottting tab. Set the *Title on Hue Axis* to be *shape_index* and the *Title on Lightness Axis* to be *curvedness*. The range of the shape index file will be read from the history *.H file and entered as -1 and +1. The ranges of the curvedness file will be larger than we want, with a value of 0.00841995 . Change that value of the *Attr. value to be plotted against max_lightness* to be the maximum value displayed earlier in the curvedness plot using the *AASPI QC Plottting tab* – a value of about 0.0002. Provide an output file name and click *Execute* to obtain the following 2D color bar and the co-rendered image:

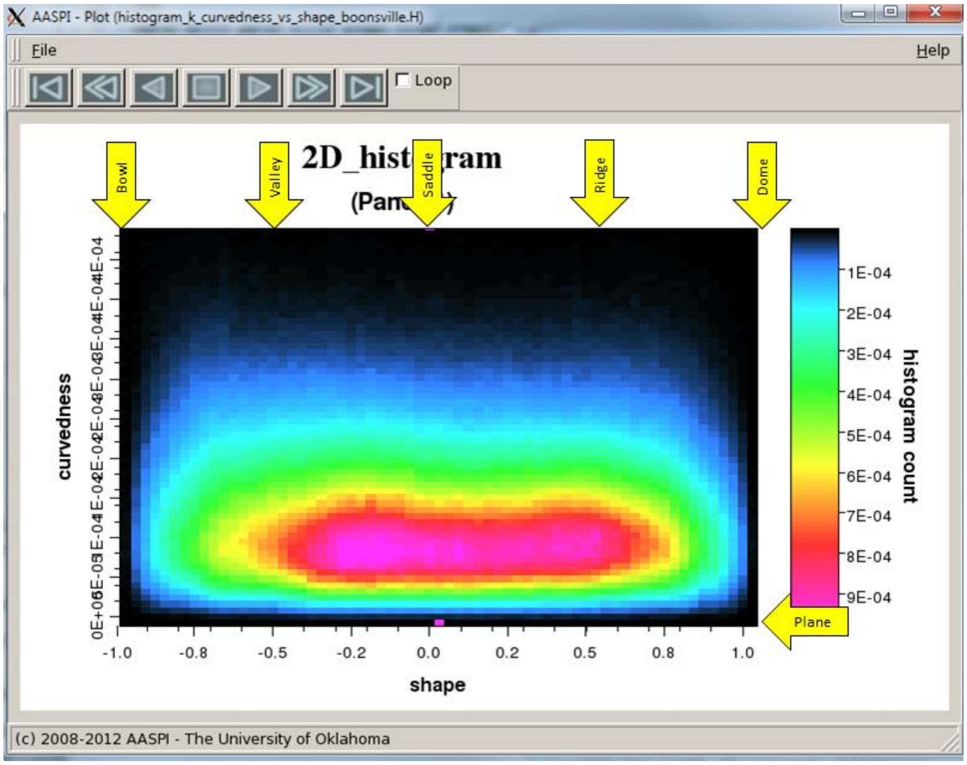


Note that values of curvedness that are close to 0.0 will appear as white, indicating a planar (undeformed) feature. The strongly deformed features (*c*>0.0002) will be displayed as we did earlier in the AASPI QC Plottting tab plot of the shape index. Intermediate values of curvedness will result in progressively darker colors

The resulting image for the Boonsville survey looks like this:



Note the previously-identified collapse features appear as bowls (in blue). These bowls are connected in some case by valleys (in cyan) and in others separated by saddles (in green). Remnant, unkarstified highs appear as ridges (in yellow) and domes (in red) as well as locally planar features (in white). A histogram of the shape distribution also is generated.

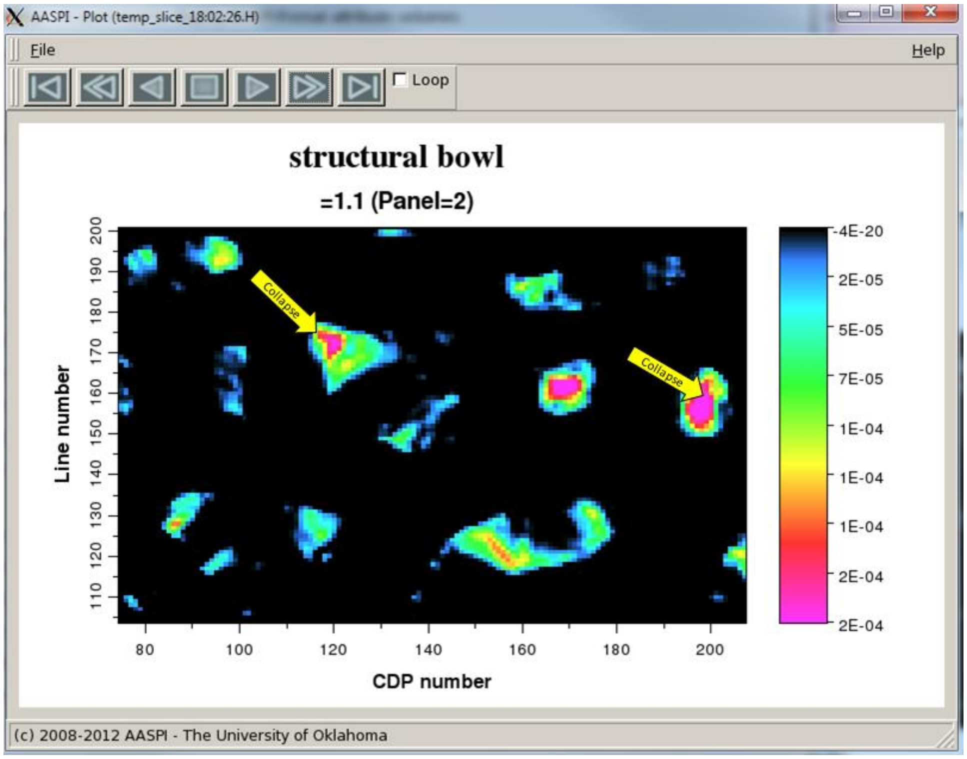


Most of the data appear to be moderately deformed, with valleys, ridges, and saddles the more common shape.

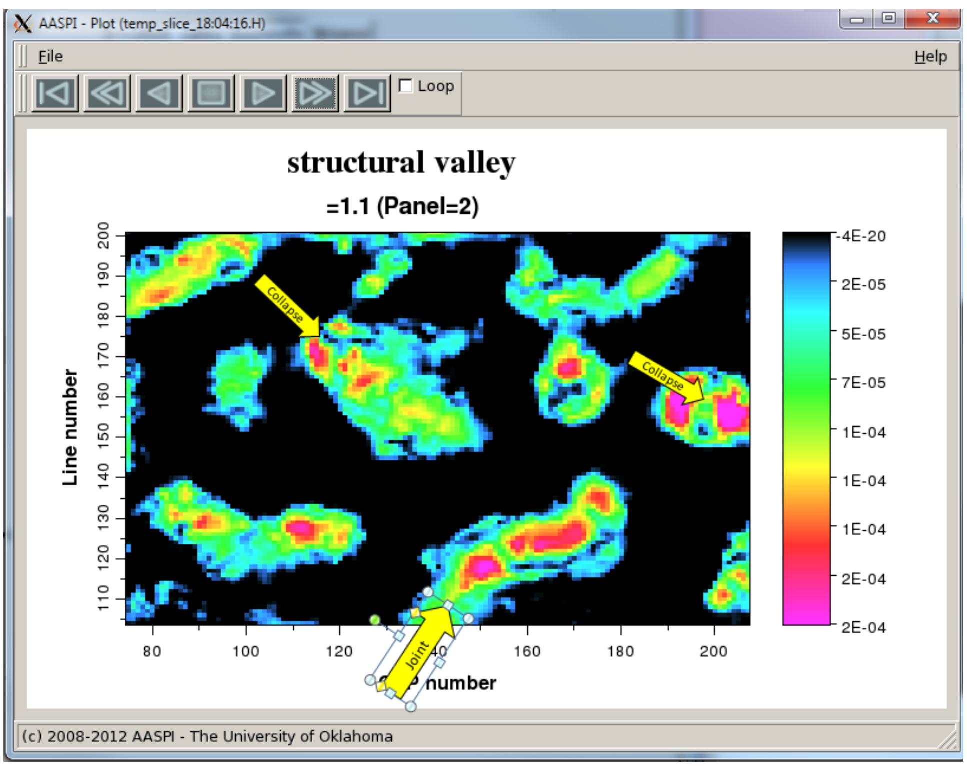
These composite attribute images can be converted back to SEG-Y format, and coupled with their corresponding colorbars (see Section 8 on **hsplot**) can be displayed in most interpretation workstations. Remember that the data needs to be scaled with user-defined parameters set to a minimum value of 0 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 of 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_boonsville_long_w_filt_0.H* should look like this:



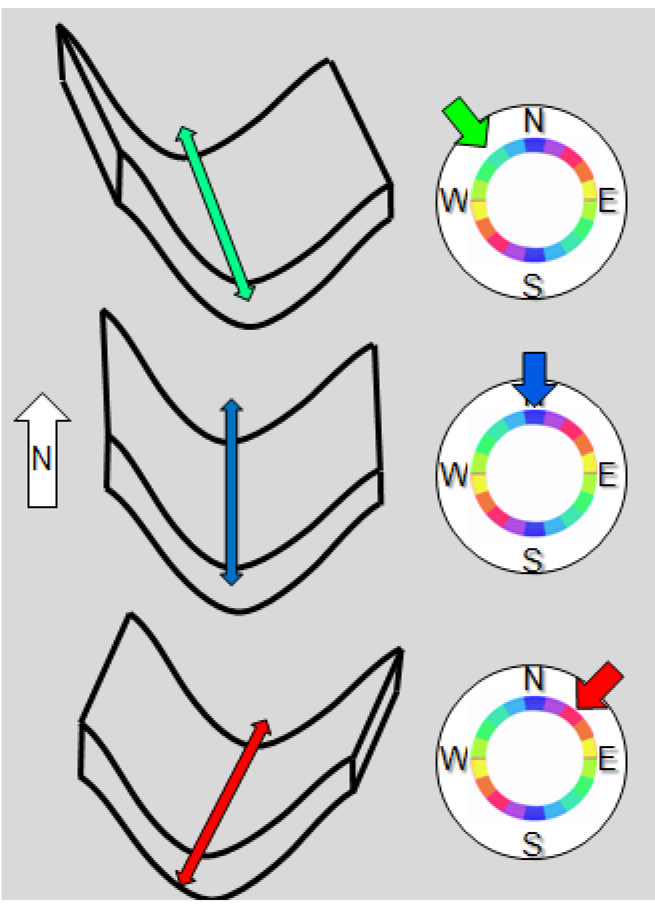
Note the collapse features appear as strong amplitude (pink) bowls. Similarly, you can plot the valley-shape component, *k_valley_boonsville_long_w_filt.H,* which looks like



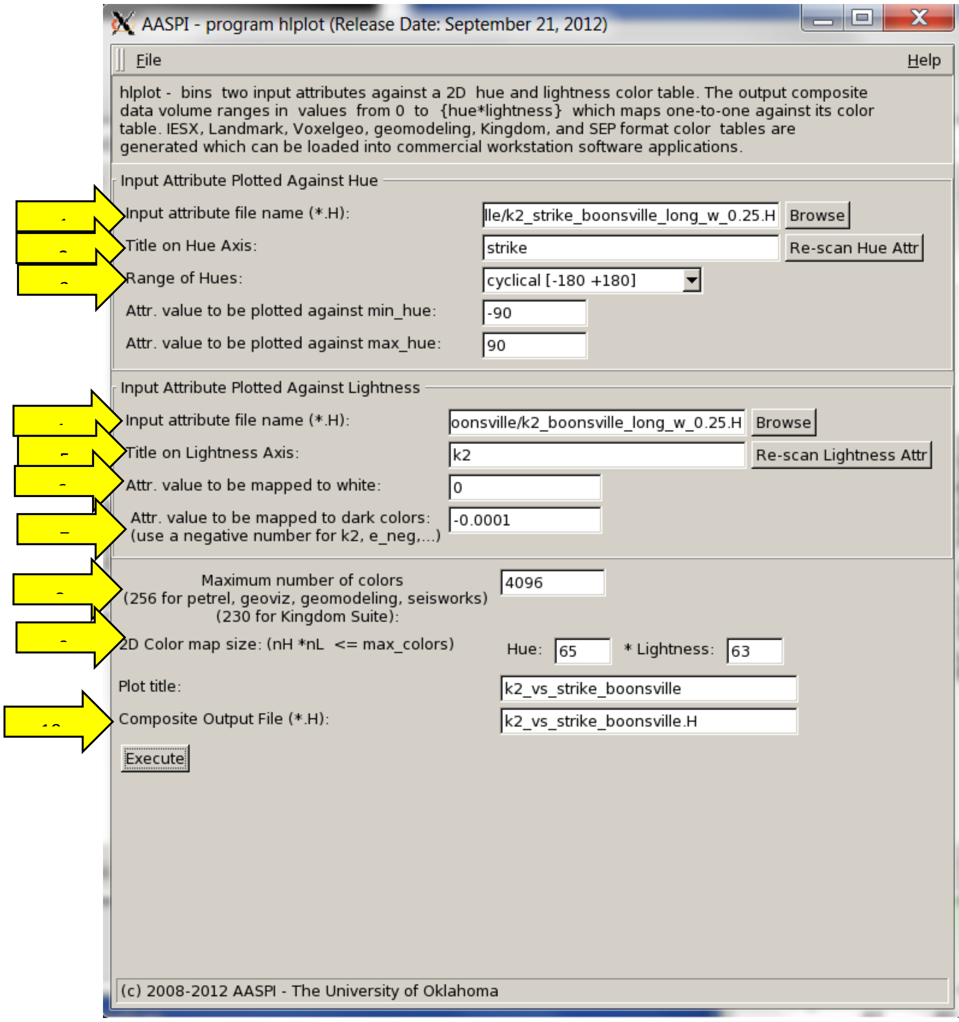
Note that the collapse features are lower amplitude (their valley component approaches 0) while circular zones around them are classified as valleys – star-shaped lows radiating out from the collapse features. We also see two prominent joints that appear as valleys.

Plotting lineament intensity vs. lineament strike

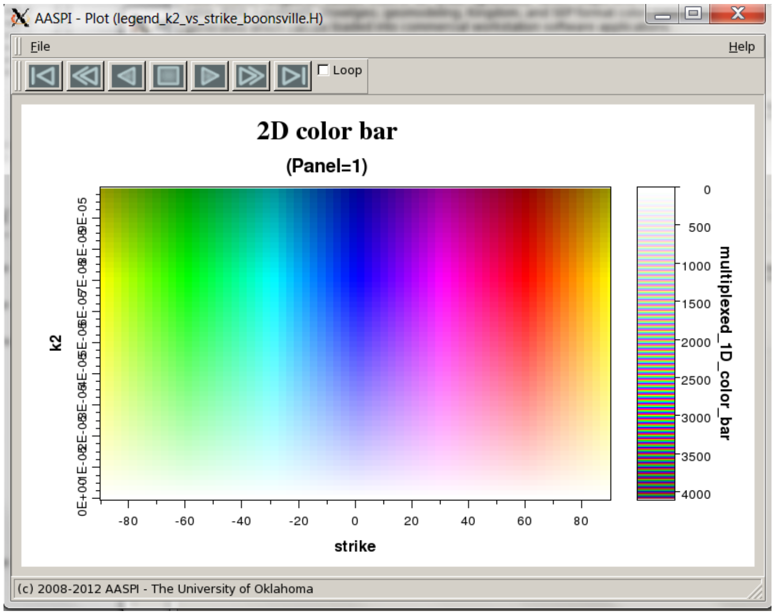
The strike is undefined for plane and perfectly-formed domes, bowls, and saddles. In contrast, it is well-defined for valley and ridge shapes. You will therefore want to modulate the lineament strike by the intensity of either the valley or ridge component. Since the Ellenburger formation in the Boonsville 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):



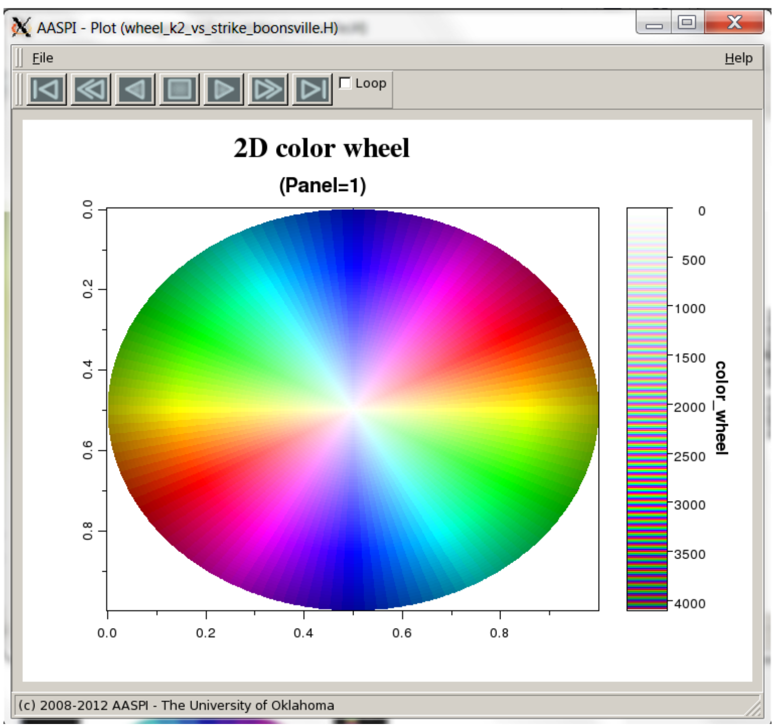
To modulate the strike of the Boonsville valleys, re-invoke program **hlplot** and fill in the parameters as shown below



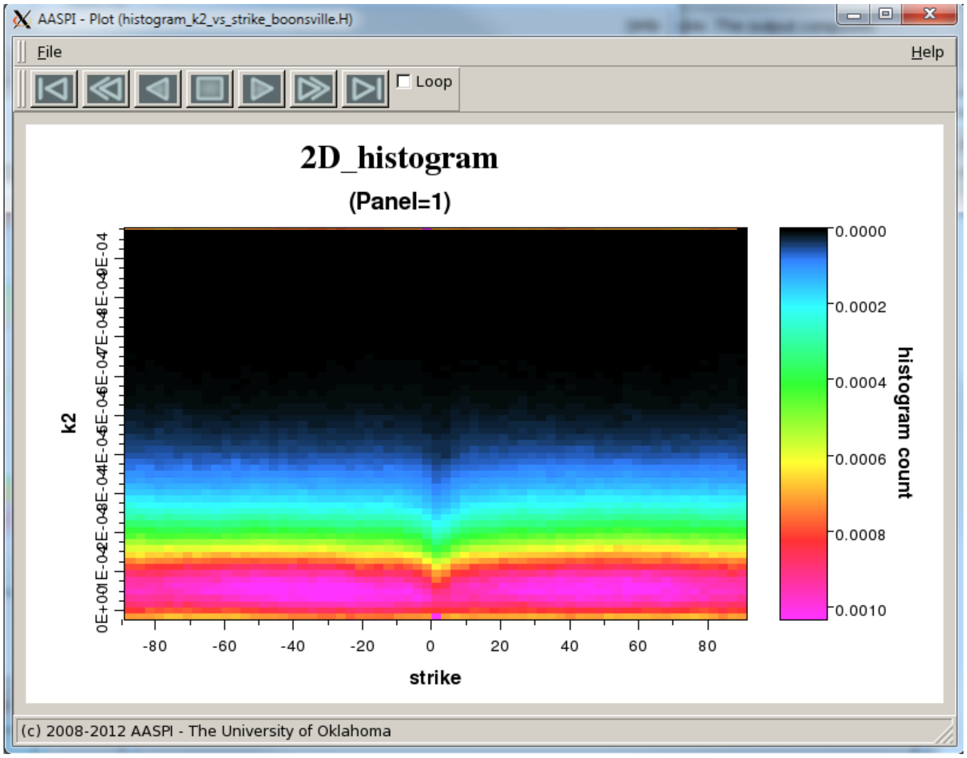
In this example we want to plot (1) k2_strike_boonsville_long_w_0..25.H against hue and (4) most negative curvature k2_boonsville_long_w_0.25.H against lightness . Label your axes (2) as *strike* and (5) *k2* to annotate properly your color bar file name and output 2D color bar axes. The Range of Hues 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:*



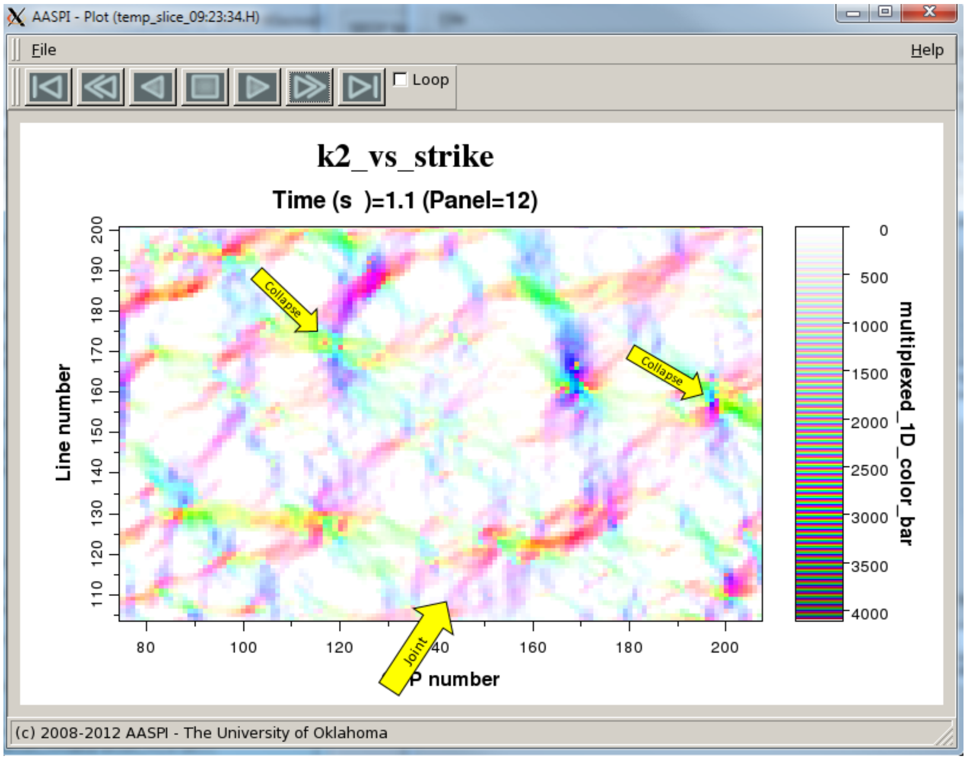
and note the 2D color bar is as we designed it where values at +90^0 have the same color as those at -90^0. We will also obtain an image of the color bar plotted as a wheel:



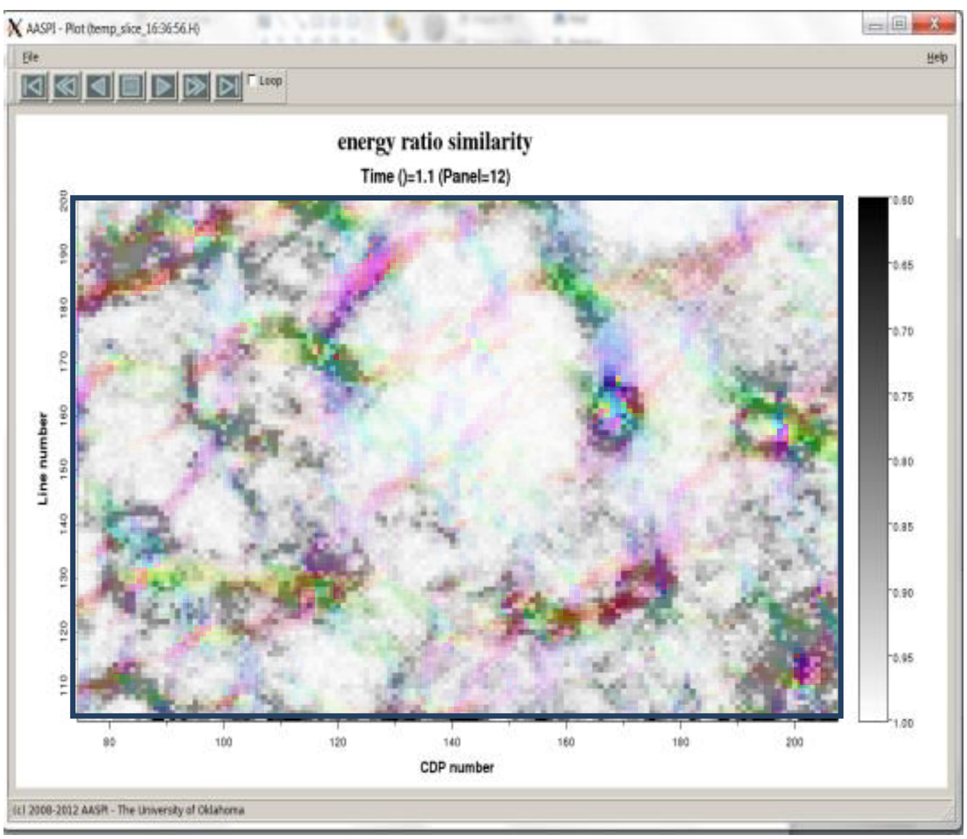
The histogram below indicates that there is a relatively even distribution of lineaments along all strike directions:



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 (-90^0 or west-east) through teal (-45^0 or northwest-southeast), blue (0^0 or north-south), salmon (+45^0 or northeast-southwest) to yellow (+90^0 or east-west).:

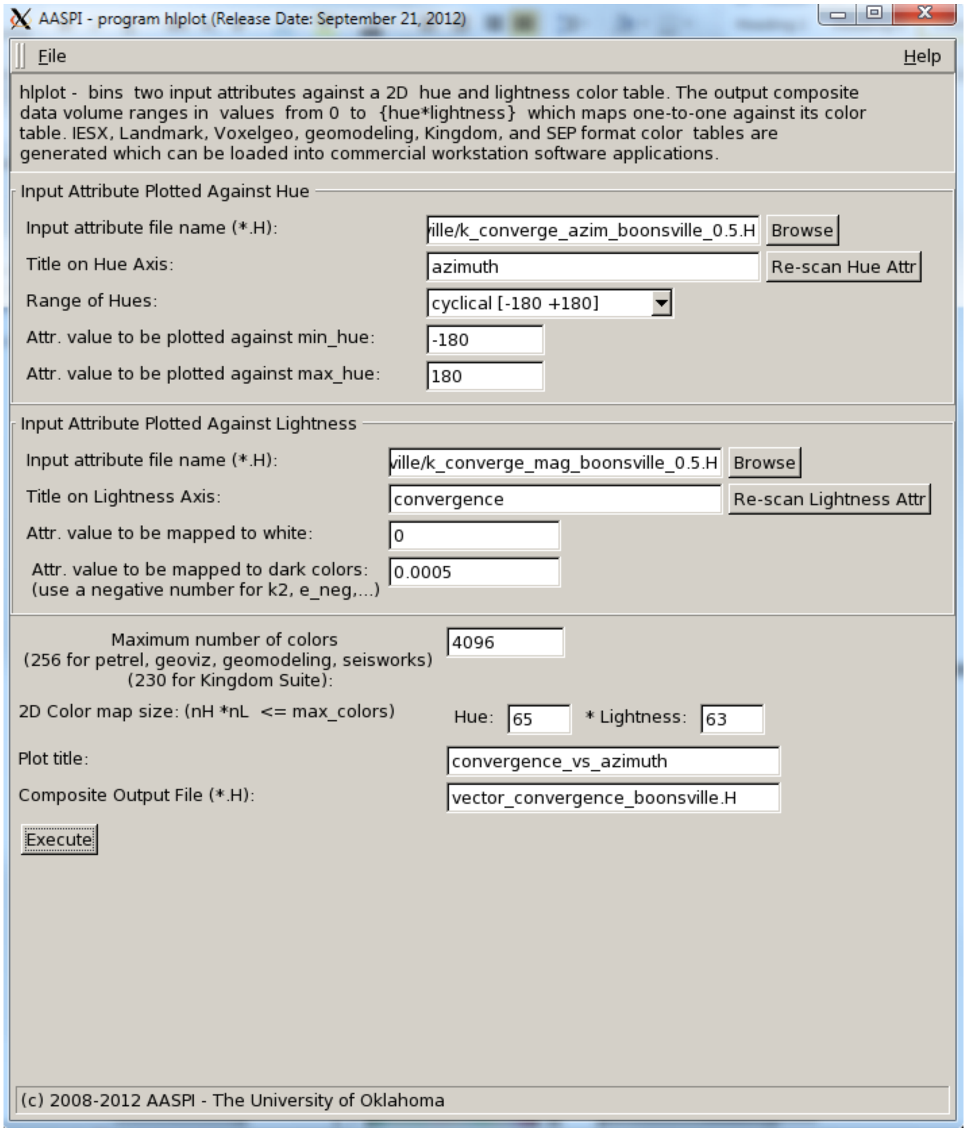


An advantage of using white as a background color is that it allows simply co-rendering with coherence using opacity:

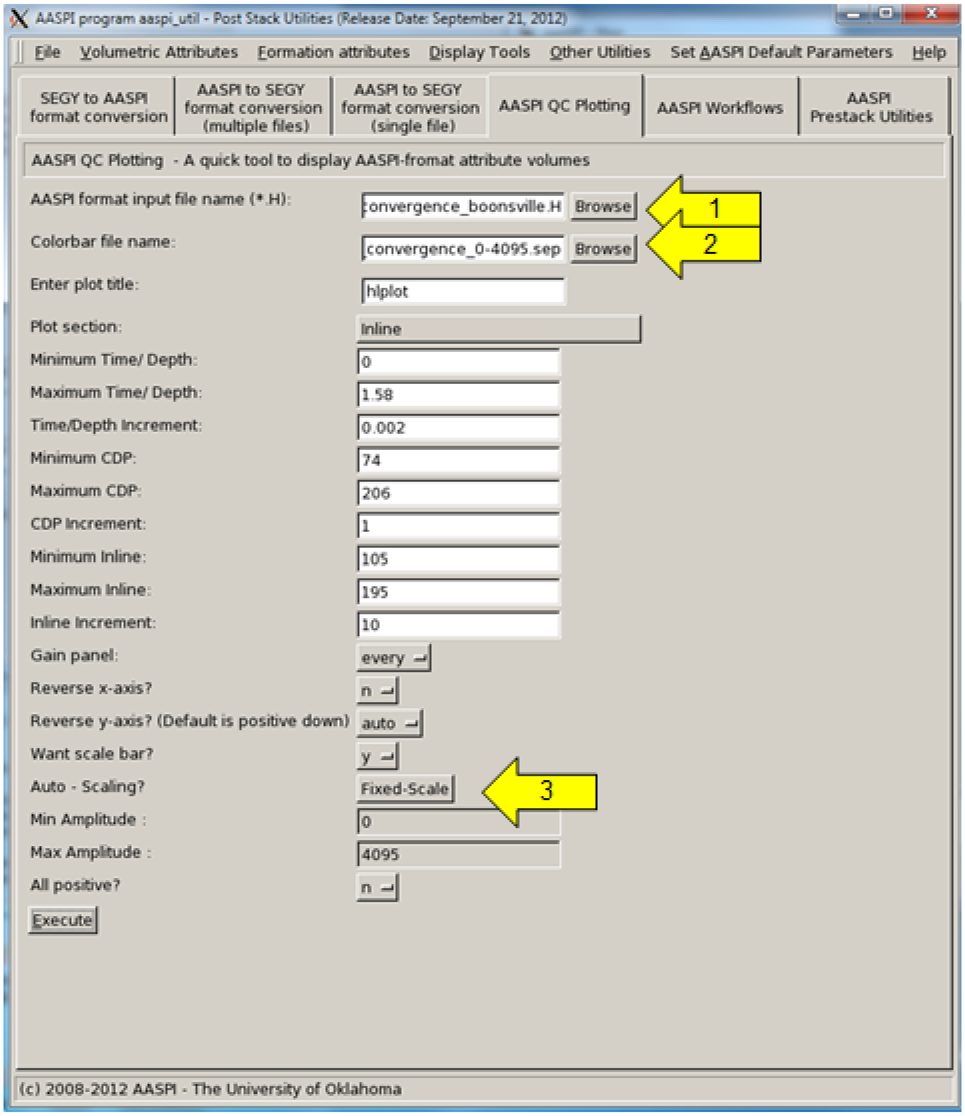


Reflector Convergence

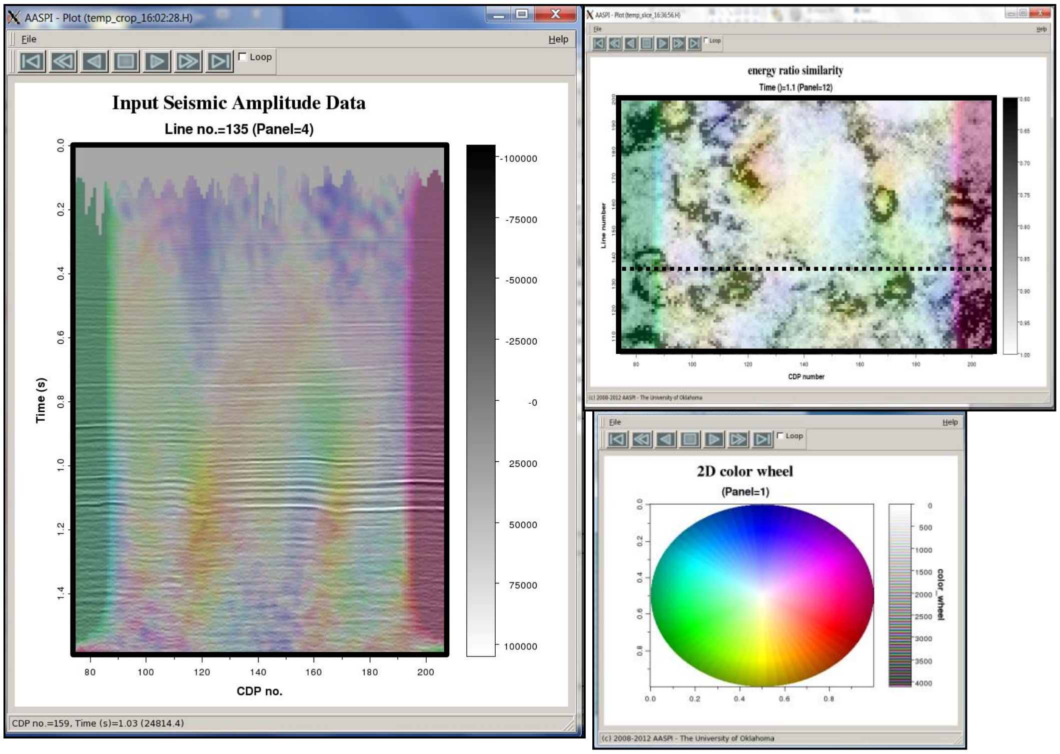
Reflector convergence is a vector that measures both the azimuth and magnitude to which reflectors converge, providing a means of mapping angular unconformities, the degree and direction of progradation, overbank deposits, and simple thickening and thinning of beds. Since we are treating this attribute 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 hlplot:



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

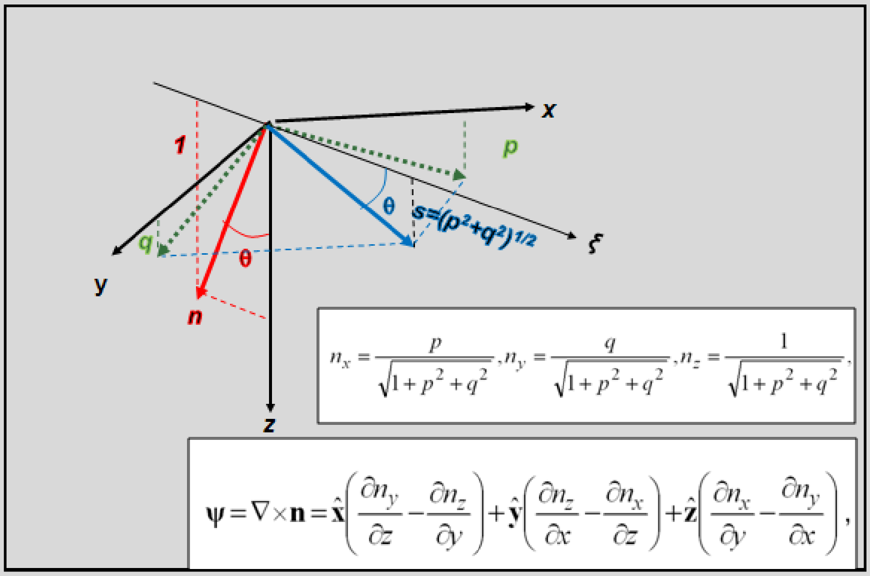


Where (1) indicates the file name that needs to be typed. Note that with the September 2012 release, the (2) color bar, and (3) range of data scaling are carried along with most of the attribute files. The following two images appear, where the I have co-rendered them with seismic amplitude on the vertical 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.


AASPI - Plot (temp_crop_16:02:28.H)
File
Help
Loop
Input Seismic Amplitude Data
Line no.=135 (Panel=4)
Time (s)
CDP no.
-100000
-75000
-50000
-25000
-0
25000
50000
75000
100000
CDP no.=159, Time (s)=1.03 (24814.4)
energy ratio similarity
Line number
CDP number
2D color wheel
(Panel=1)
color_wheel
(c) 2008-2012 AASPI - The University of Oklahoma

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:



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

$$\begin{aligned}
\mathbf{c} = \mathbf{n} \times \boldsymbol{\psi} = \hat{\mathbf{x}} & \left[n_y \left(\frac{\partial n_x}{\partial y} - \frac{\partial n_y}{\partial x} \right) - n_z \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) \right] \\
+ \hat{\mathbf{y}} & \left[n_z \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) - n_x \left(\frac{\partial n_x}{\partial y} - \frac{\partial n_y}{\partial x} \right) \right] . \\
+ \hat{\mathbf{z}} & \left[n_x \left(\frac{\partial n_z}{\partial x} - \frac{\partial n_x}{\partial z} \right) - n_y \left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y} \right) \right]
\end{aligned}$$

Reflector Rotation about the Normal

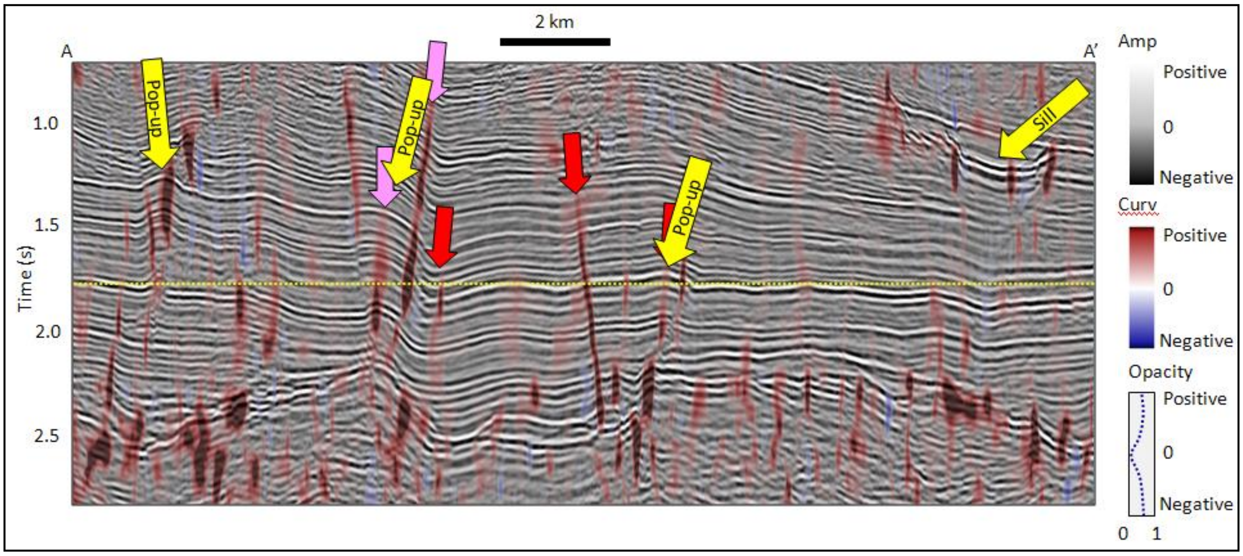
$$r = \mathbf{n} \bullet \boldsymbol{\psi} = n_x\left(\frac{\partial n_y}{\partial z} - \frac{\partial n_z}{\partial y}\right) + n_y\left(\frac{\partial n_z}{\partial x} - \frac{\partial n_x}{\partial z}\right) + n_z\left(\frac{\partial n_x}{\partial y} - \frac{\partial n_y}{\partial x}\right)$$

Examples

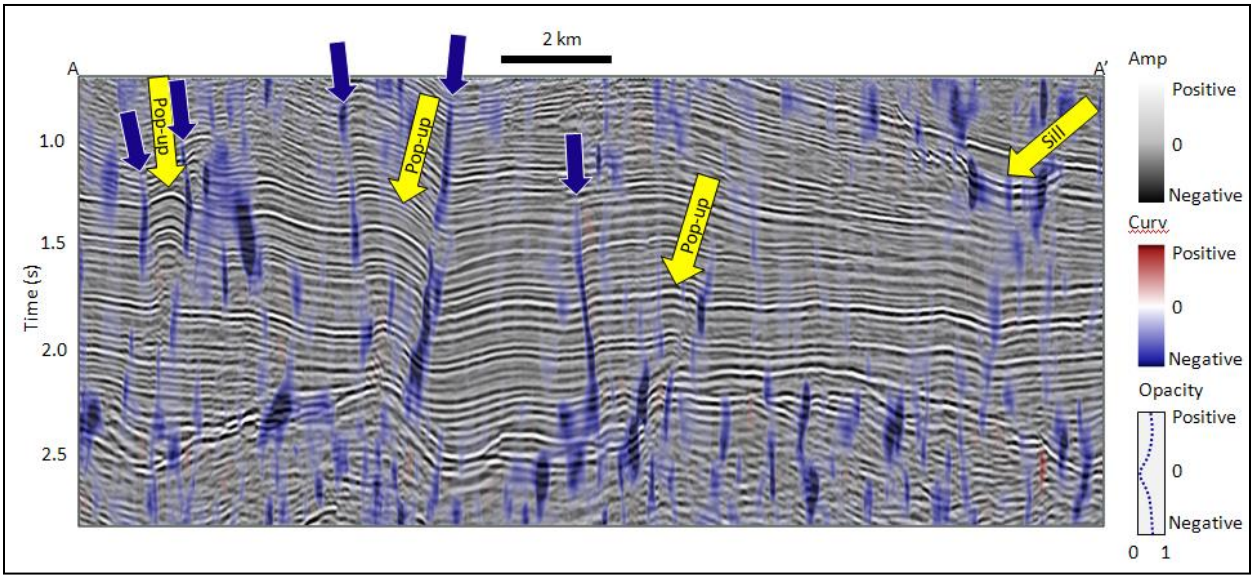
While the Boonsville data volume is useful for testing, other examples using licensed data better demonstrate the value of this family of attributes. Most of these examples come from either publications or from Marfurt's SEG not-so-short course found under www.geology.ou.edu/aaspi/upload .

Examples showing most-positive and most-negative principal curvatures

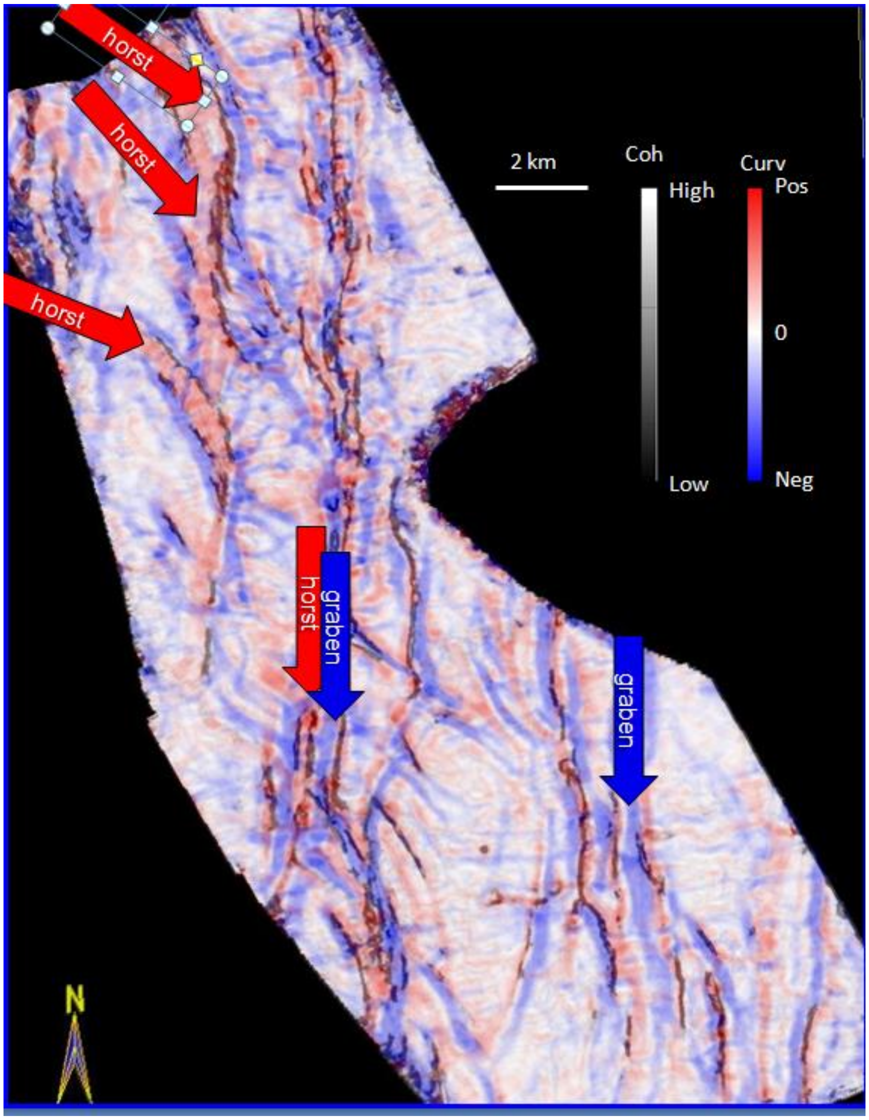
Many workers associate curvature with faults. While this may sometimes be the case, they are usually juxtaposed about a fault, or simply measure folding. An example of the later features are shown in images presented by Mai et al. (2009) over the Chicontepec Basin of Mexico for the most-positive curvature,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



and the most-negative principal curvature, k_2:



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. Clearly, 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.

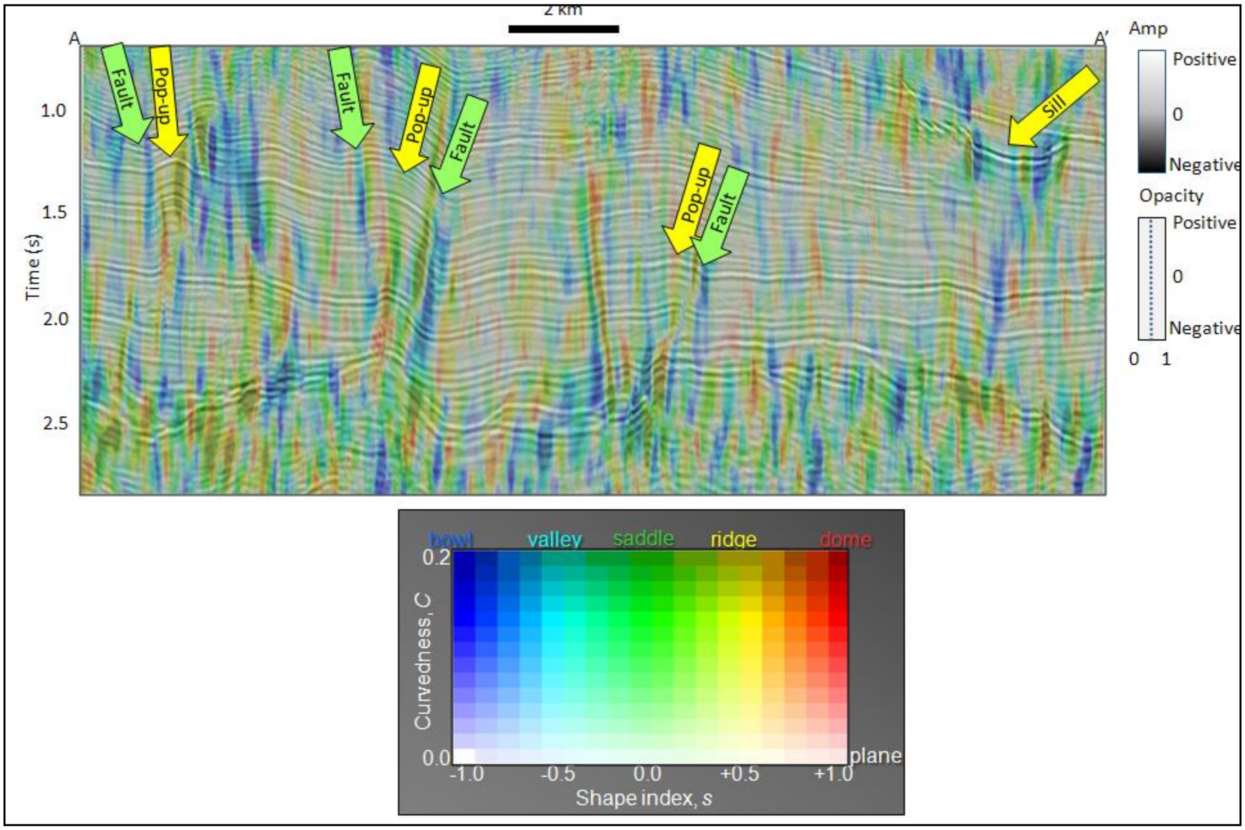


Examples Showing Curvedness and Shape Index and Shape Components

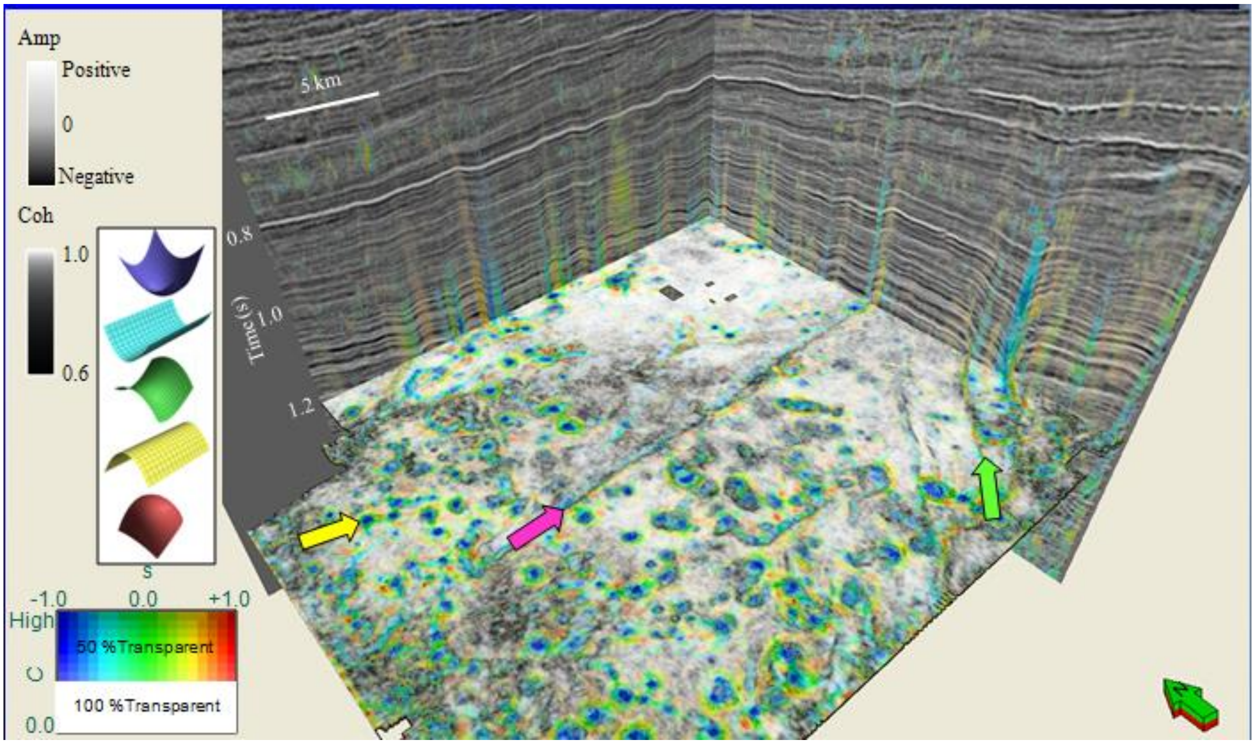
The two principal curvatures, k_1 and k_2, define five quadratic shapes, and a degenerate planar 'shape' when $k_1 = k_2 = 0$. Typically, will plot these shapes using the end members of a 2D color bar (modified by Mai et al., 2009, from an earlier paper by Bergbauer et al.,2003):

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:

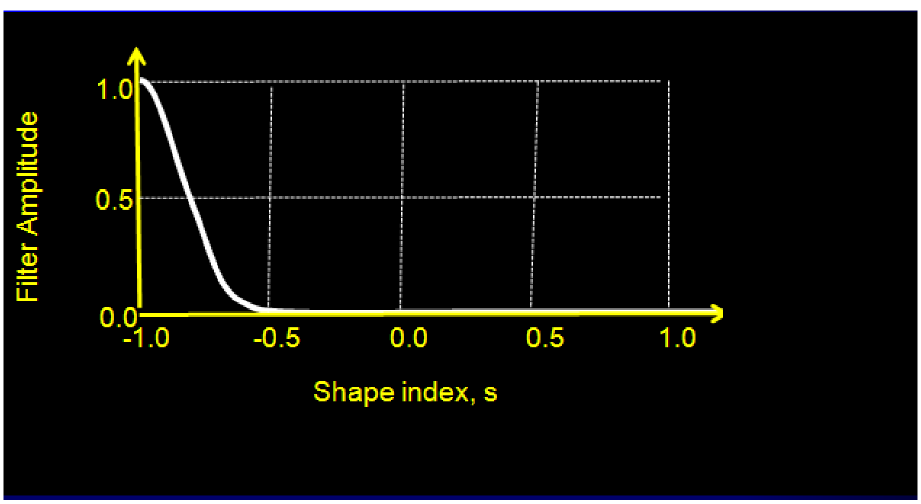
Using the 2D color bar modulating the shape index by the curvedness using program hlplot, we show the same Chincontepec, Mexico image displayed earlier:



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).

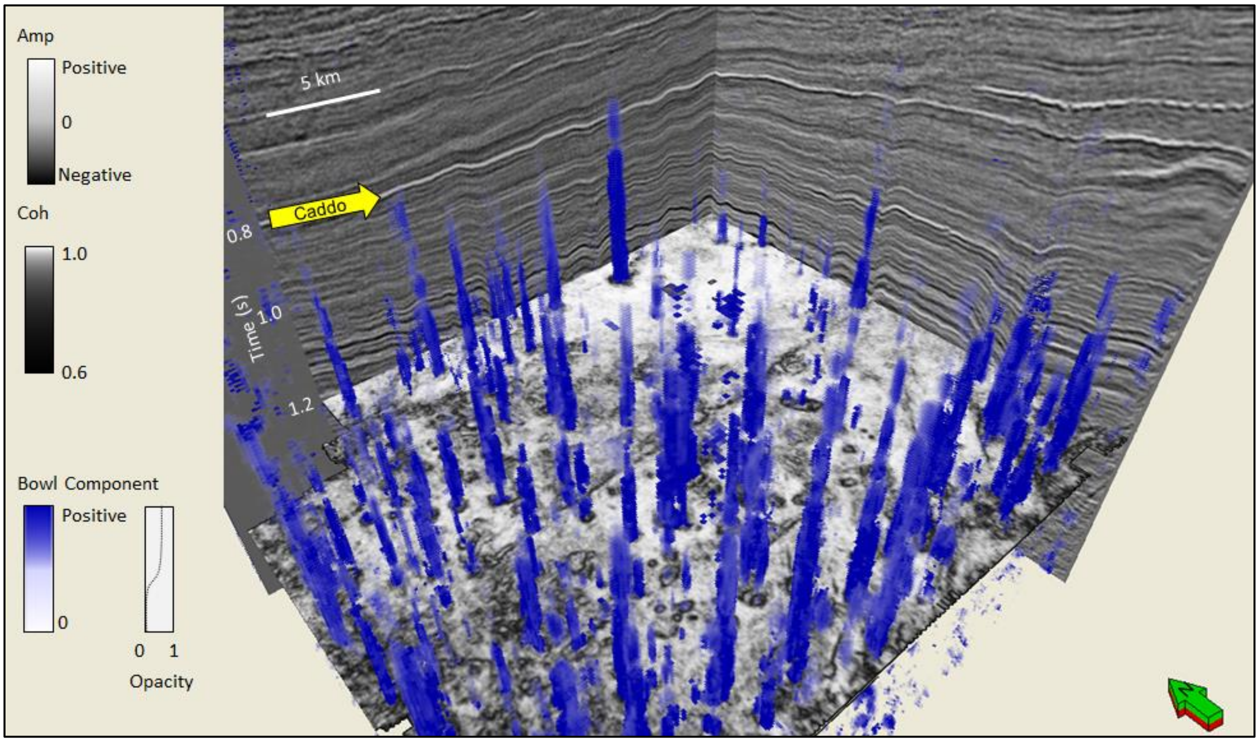


Internal to program curvature3d, one can apply filters to these two attributes to generate separate shape 'components', such as the filter used to 'extract' bowl shapes shown below:



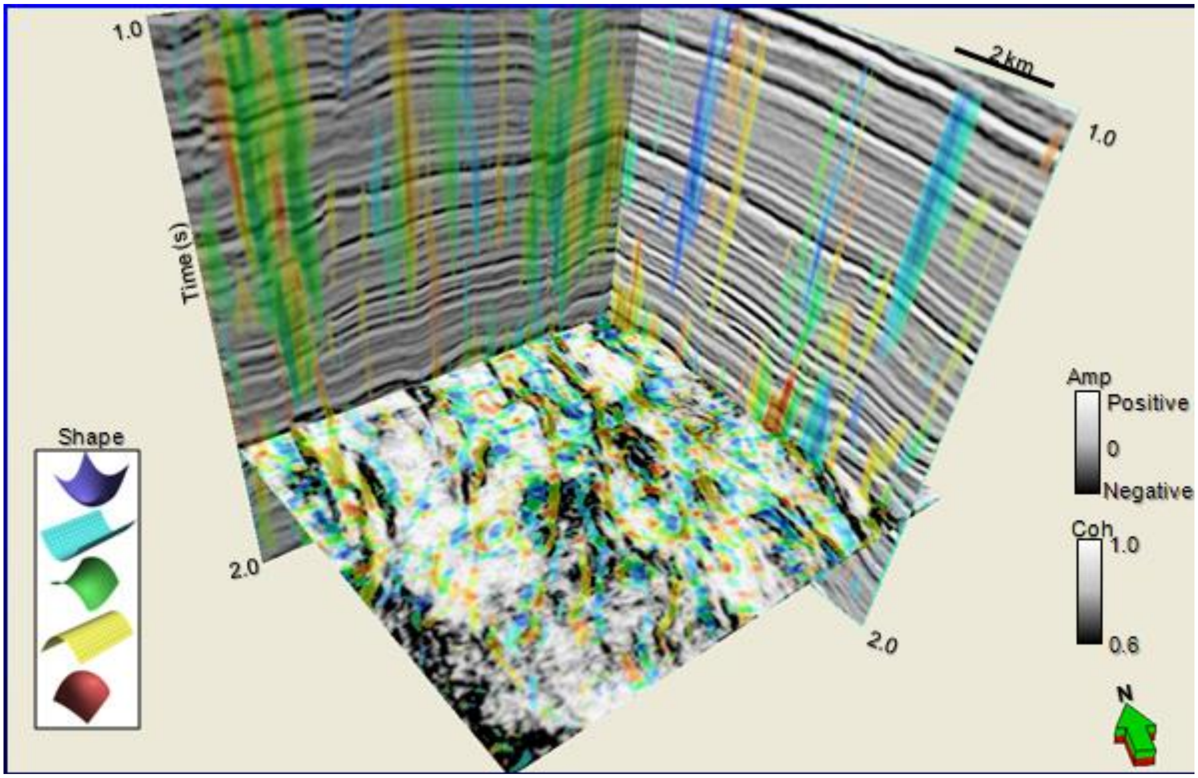
The result from Marfurt (2010), is the following image corresponding to the vertical and time slices shown above where he has used a monochrome white to blue color bar to plot the intensity of the bowl component (again, note the correlation with the collapse features):

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

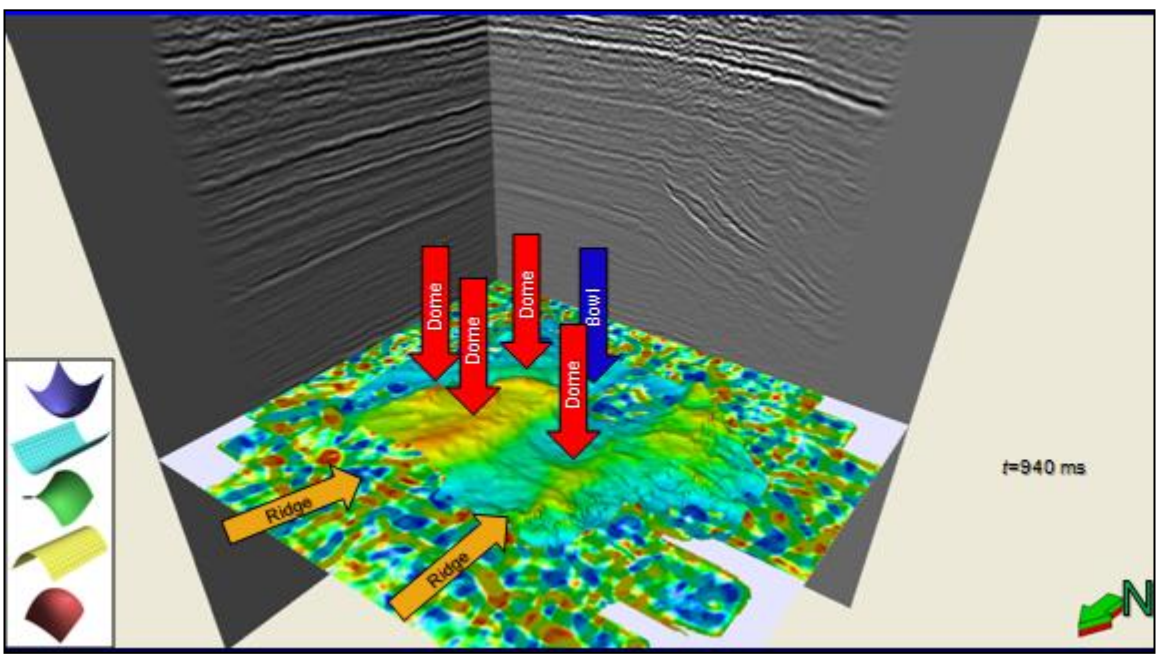


Note the collapse features seen in the coherence time slice at the Ellenburger level continue as bowl-shape features up into the Atoka formation which lies just below the strong Pennsylvanian-age Caddo horizon in this image. These bowls form geohazards that need to be avoided in hydraulic fracturing 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).

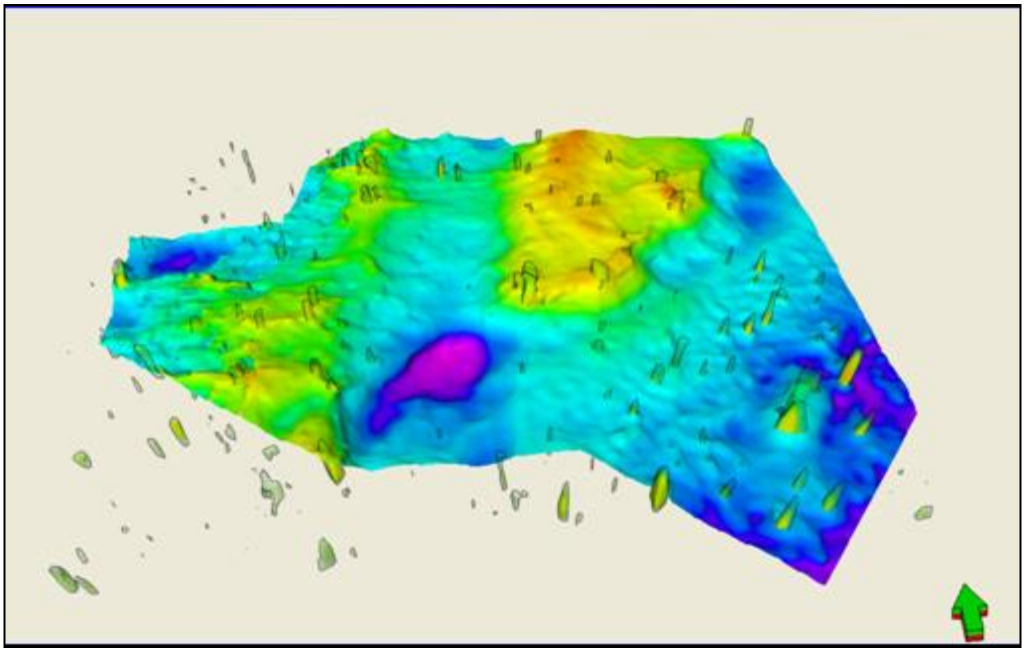
Chopra and Marfurt (2011) use a similar workflow and display technique to show the correlation of shapes to energy ratio coherence for a survey acquired in the Western Sedimentary Basin of Canada:



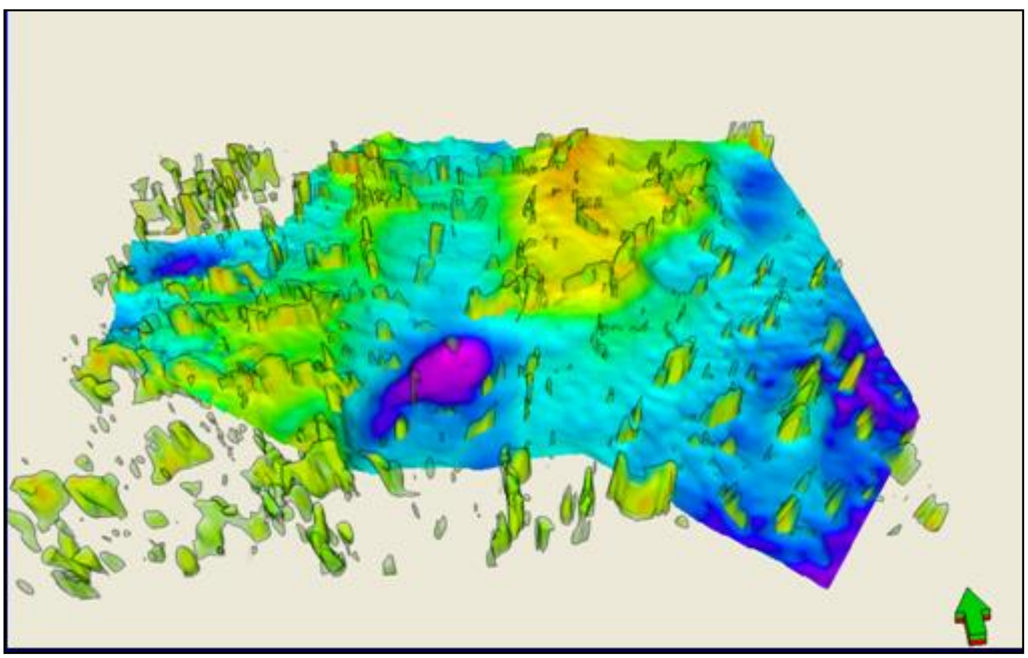
In this next image, Roderick Perez picked the top of the carbonate reef system in the Horseshoe Atoll area of west Texas and co-rendered a time slice through the shape index to validate the correlation:



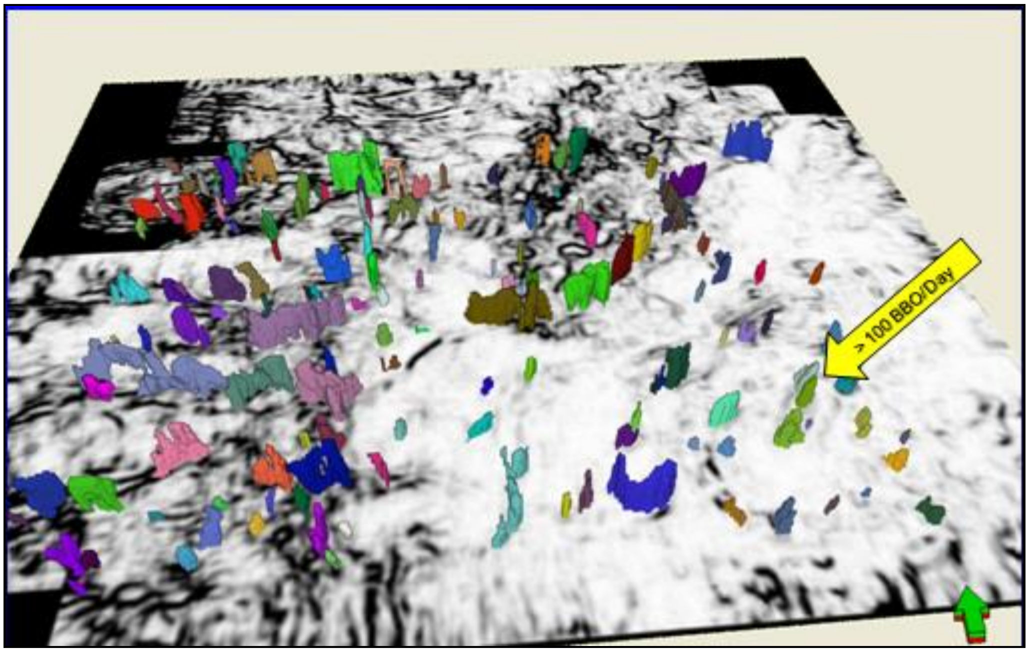
Next, we compute the dome components and use Petrel's boxprobe (their implementation of the geoprobe concept) to highlight some of the pinnacle reefs:



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



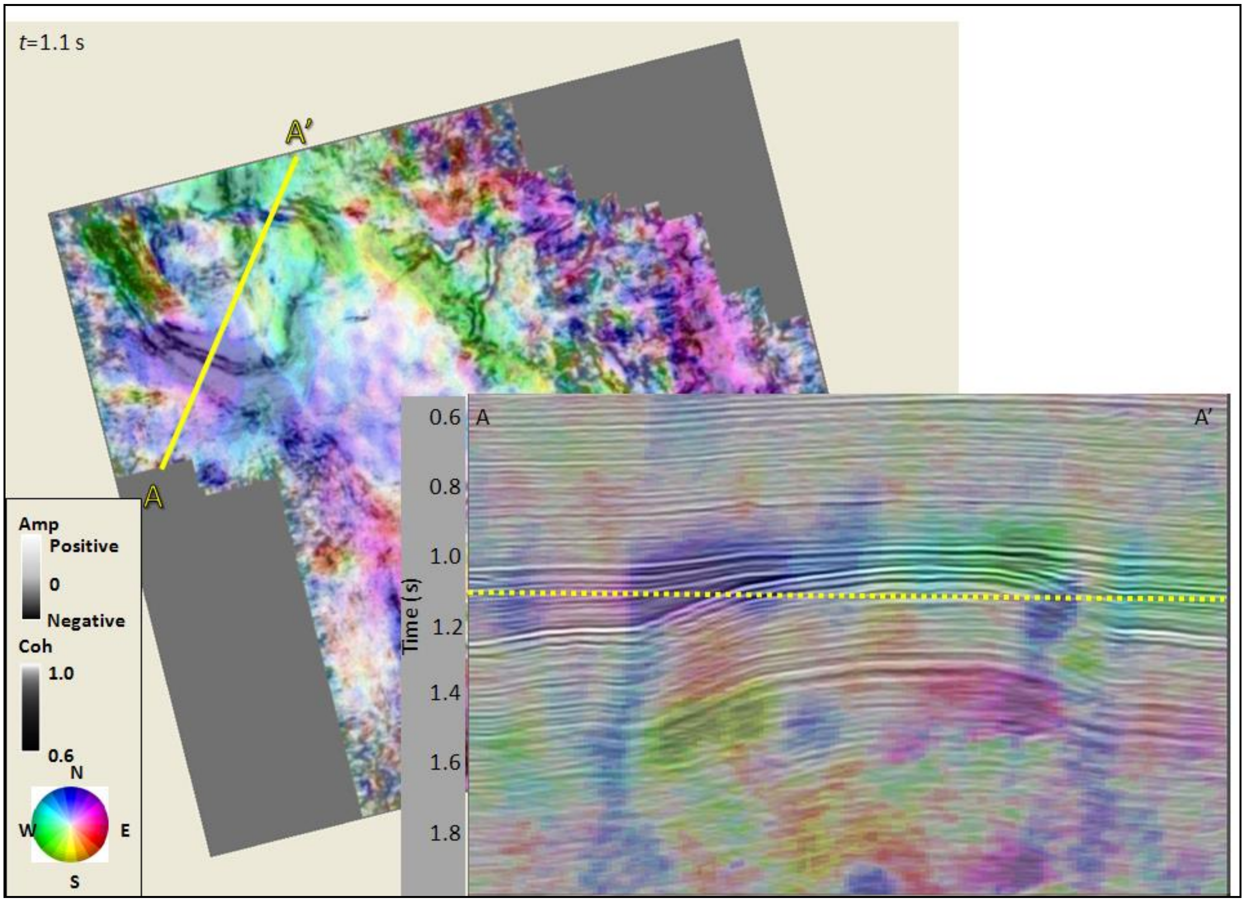
Finally, one can use the concept of geobodies to quantify the volumes of the previously-delineated shapes:



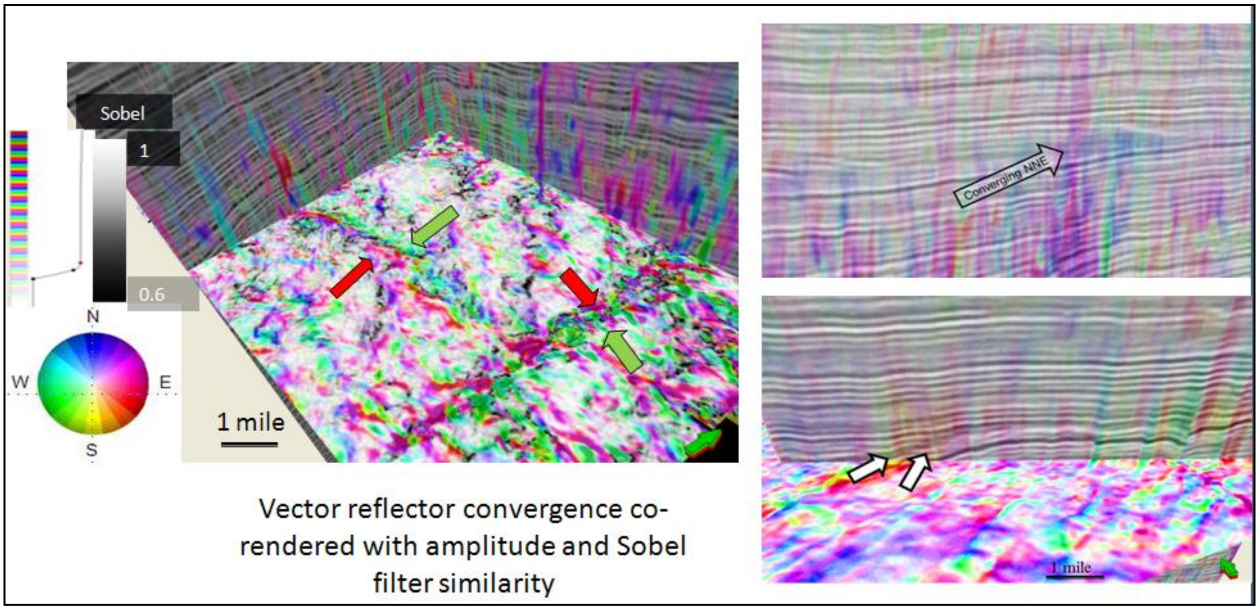
The yellow arrow indicates one of the geobodies that was successfully drilled in the Summer of 2010.

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. 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.

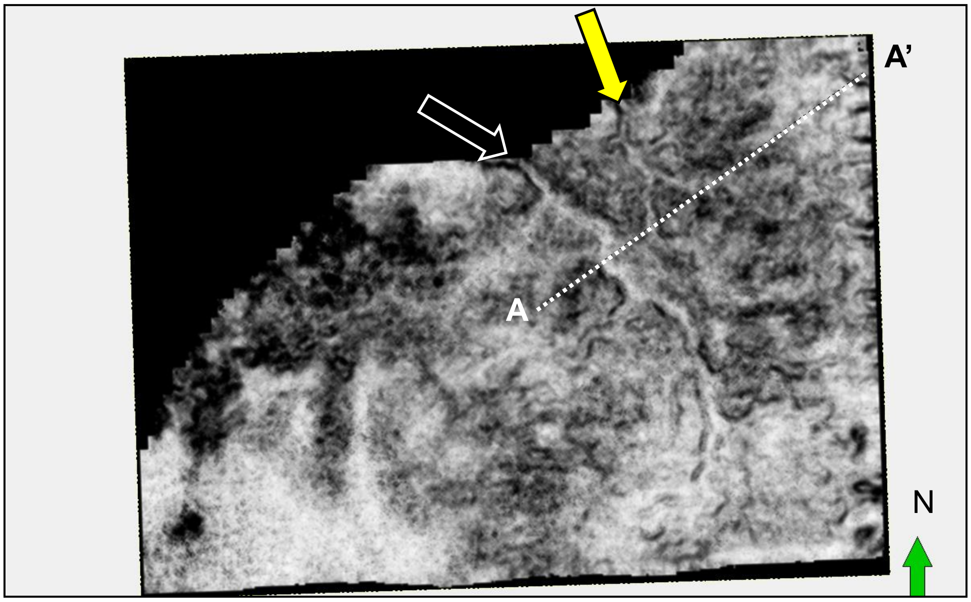


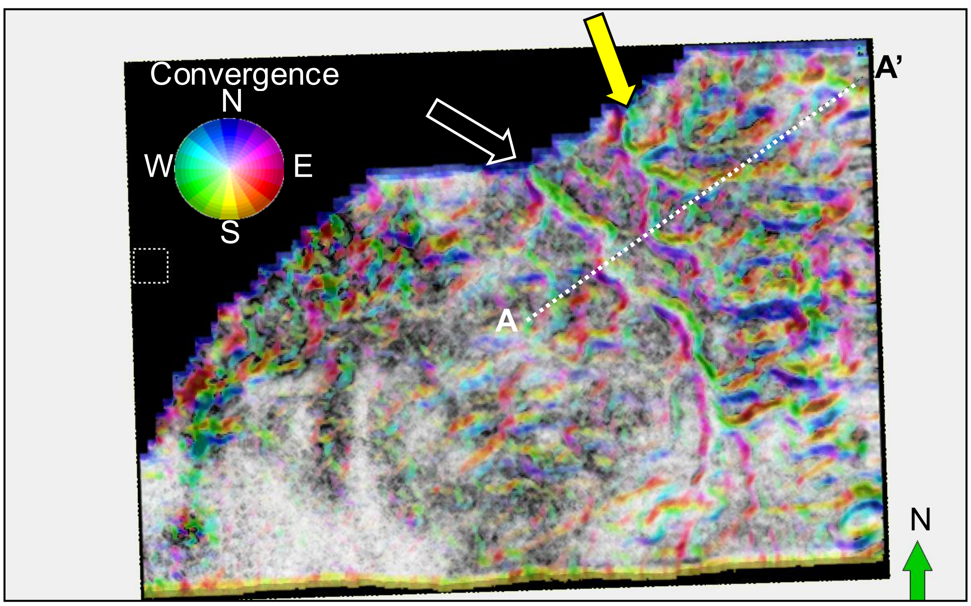
We draw a second example from Gupta et al. (2011) analyzing unconformable deposition of the Woodford Shale onto the underlying eroded Hunton Limestone of north-central Oklahoma:



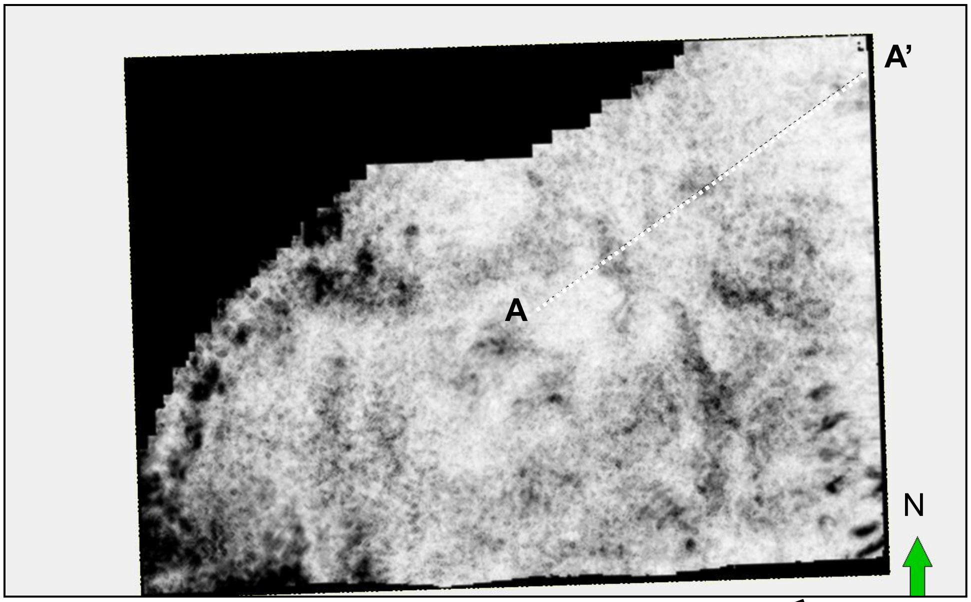
Vector reflector convergence co-rendered with amplitude and Sobel filter similarity

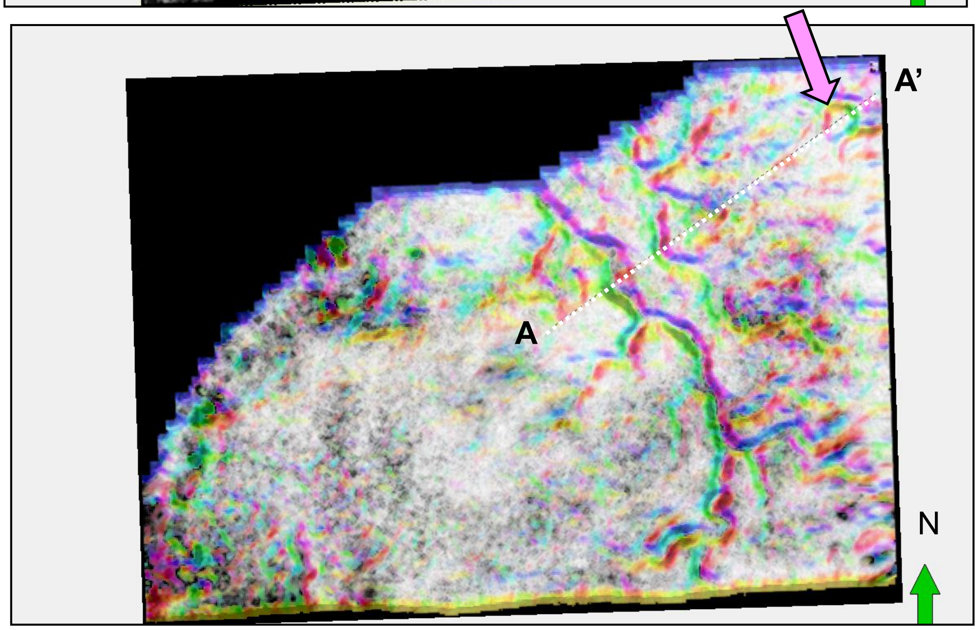
The next example comes from Chopra and Marfurt (2013) and shows the convergence within and above and incised channel in Canada. The first pair of figures show time slices at t=1.700 s through the incisement. The top figure shows coherence, while the bottom shows coherence co-rendered with reflector convergence.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



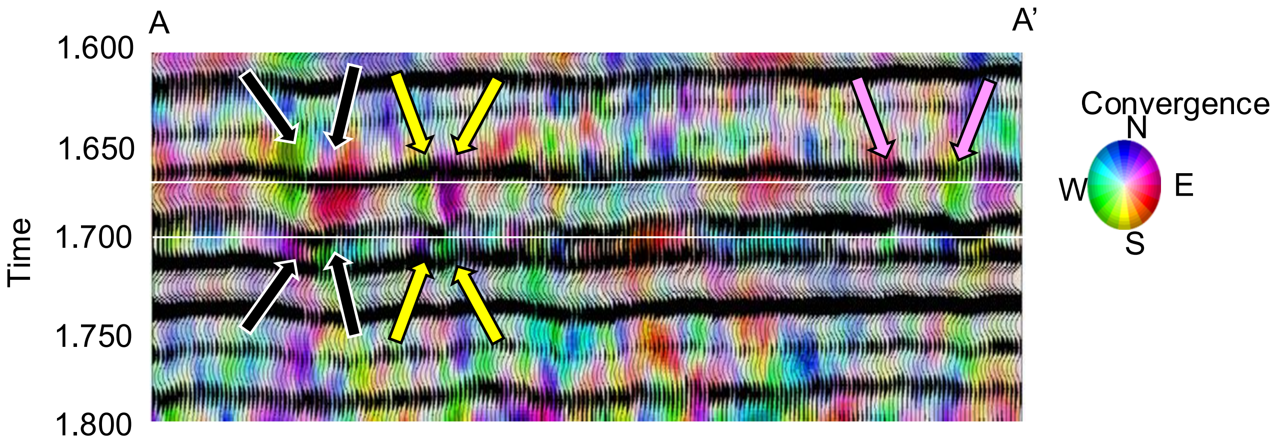


The next two images show a shallower time slice above the incisement at t=1.670 s. In the coherence image there is no indication of the deeper channel. However, in the reflector convergence image we see convergence *away* from the channel center, indicating differential compaction into the channel. We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

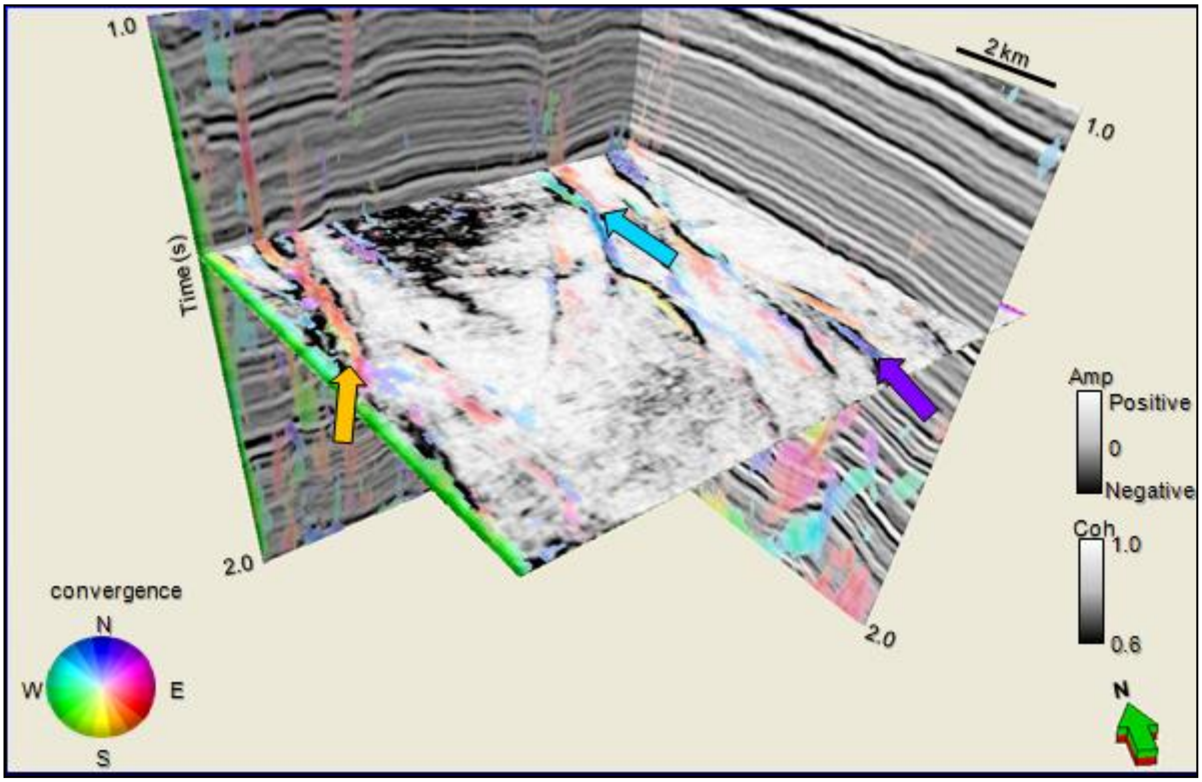




We show line AA' through the seismic amplitude co-rendered with reflector convergence. Location of line is shown in the previous two figures while white time lines in this vertical slice indicate the time slices shown in the two previous figures. Black and yellow arrows indicate convergence anomalies associated with the more western and more eastern channel. Note the flip in the direction of the convergence anomalies between the incisement 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.

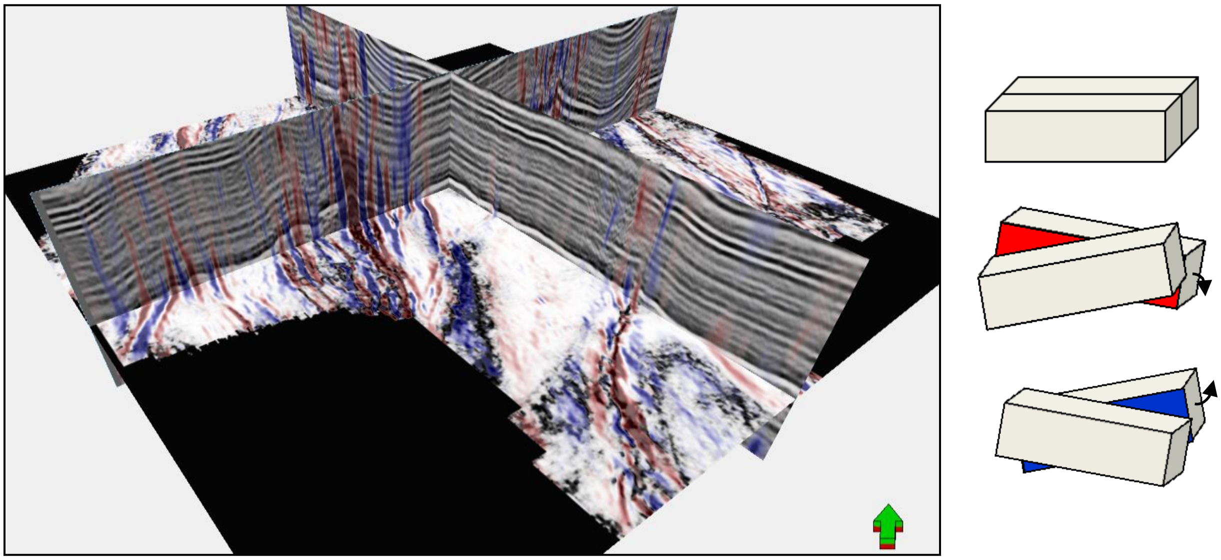


Finally, we draw an image from Chopra and Marfurt (2011) from the Western Sedimentary Basin that shows the vector convergence anomalies within fault blocks, which we interpret to be rotation of the graben provided increased accommodation space in the direction indicated by the hue. Note that most of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

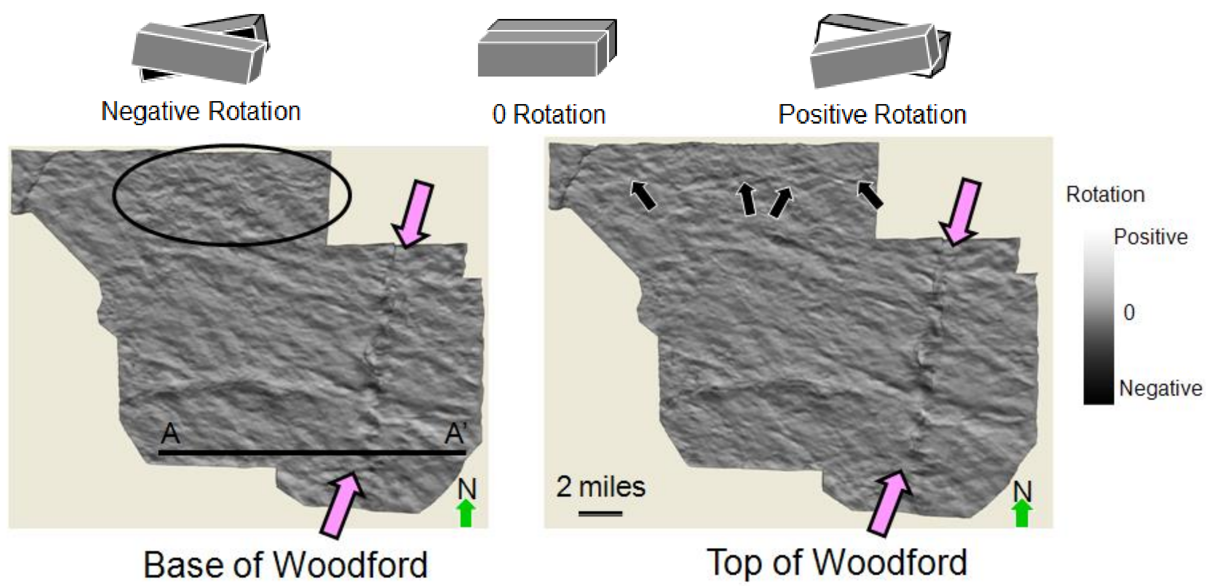


EXAMPLES SHOWING ROTATION ABOUT THE NORMAL

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:

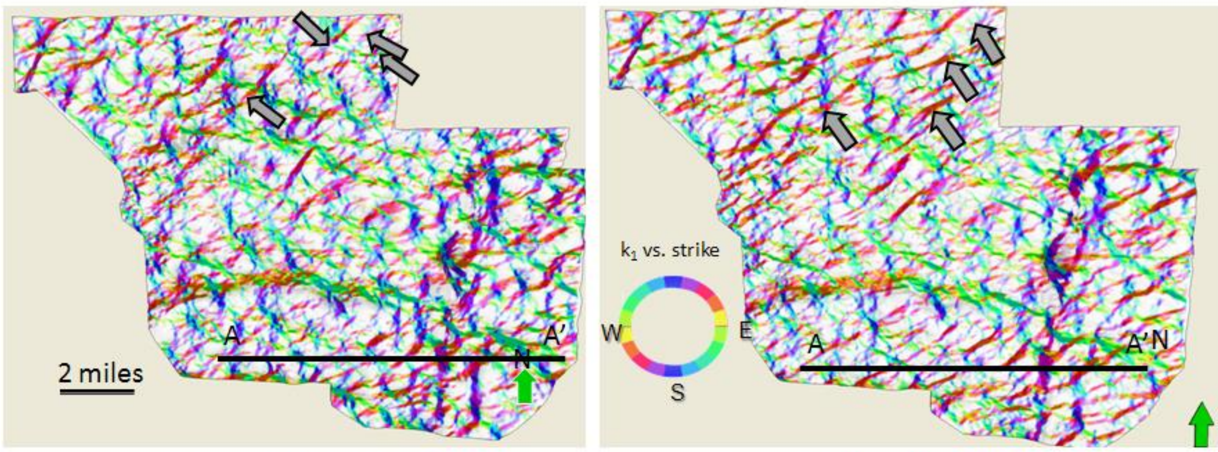


The Woodford Shale in the Anadarko Basin has variable thickness. Using core data, Gupta et al. (2011) explain the lateral variability in SiO_2 (primarily chert), $CaCO_3$, and TOC (total organic carbon) as being control by irregular water depths over the previously eroded Hunton Limestone. The differences between the top and bottom the Woodford show up well in the following image of reflector rotation:



Examples Showing Structural Lineaments

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



Base of Woodford   Top of Woodford

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

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.