COMPUTING APPARENT DIP AND AMPLITUDE GRADIENTS – PROGRAM apparent_cmpt

Contents

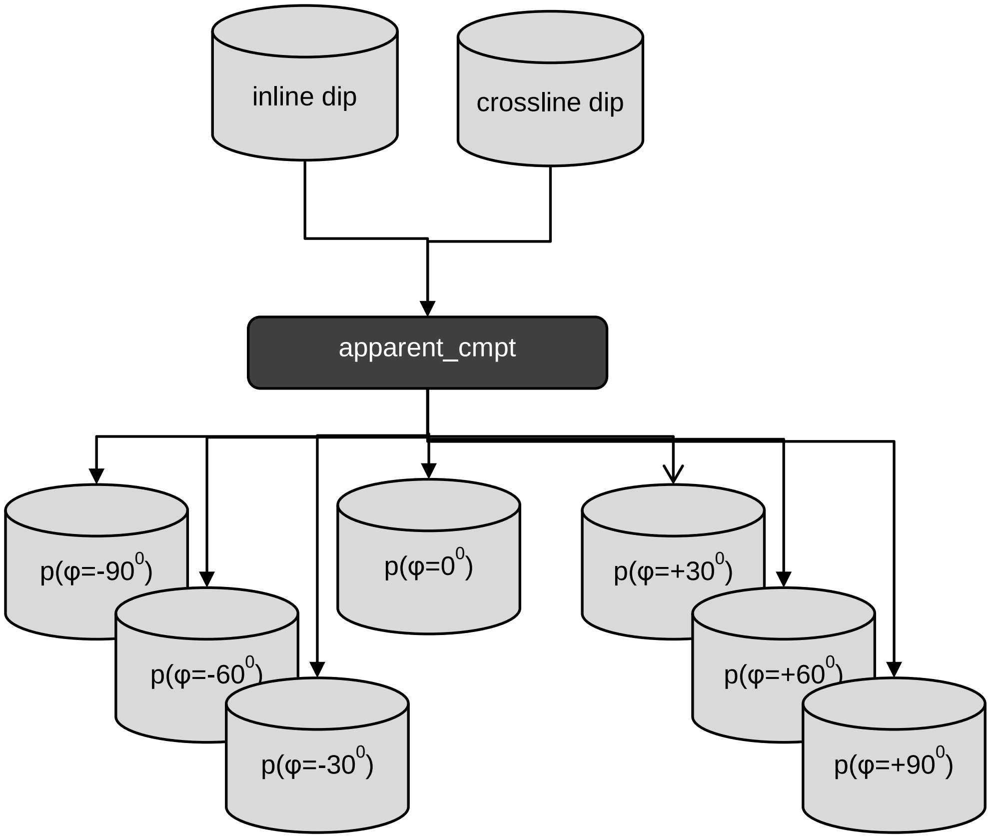
Overview

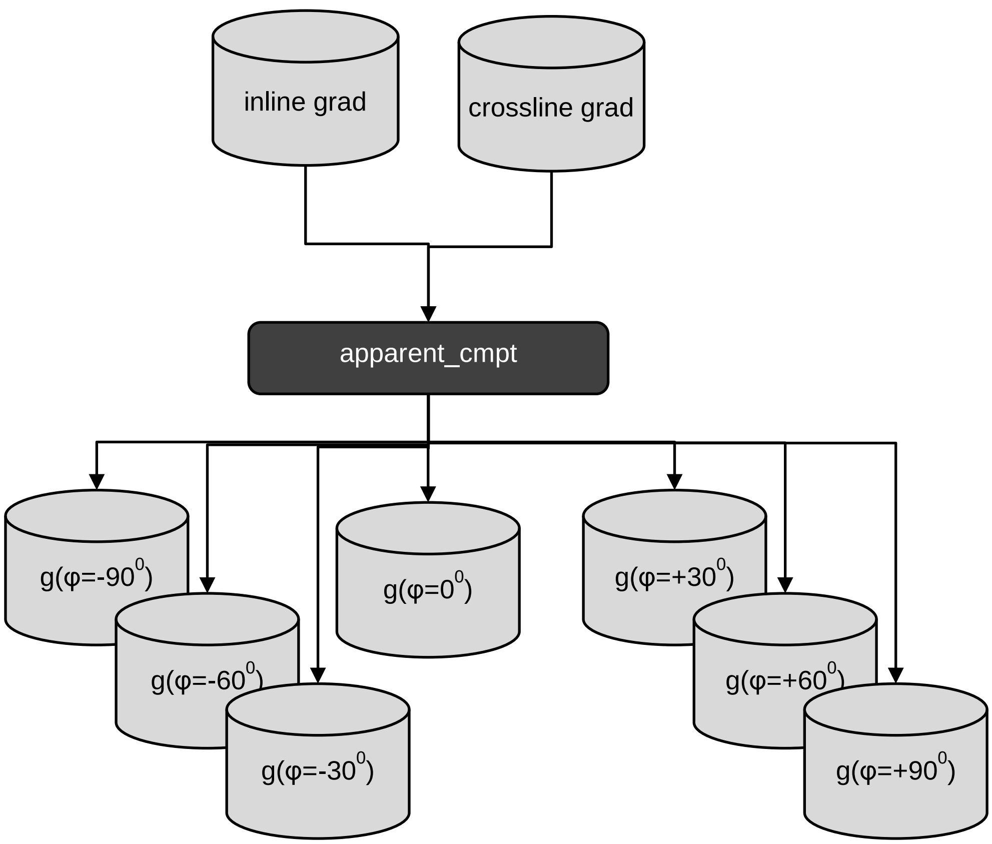
Vector attributes are represented by their vector components. For example, we can define the vector dip by either the dip magnitude – dip azimuth pair or by the inline dip – crossline dip pair, where the latter two values define apparent dips aligned with the survey orientation. In some cases, the interpreter may wish to examine the vector component at a specific orientation, say one perpendicular to the minimum horizontal stress. In other cases, the interpreter may wish to animate through a suite of apparent orientations, thereby highlighting structural or stratigraphic features perpendicular to that direction. Program **apparent_cmpt** provides a means of computing apparent components by any vector that can be represented by a magnitude and an azimuth. Because curvature is a not a first-order vector, but rather a second-order tensor, computing the apparent components of curvature is slightly more complicated and is performed by program **euler_curvature**.

Apparent component computation flow charts

The following two flow charts so the computation of apparent components from the vector dip and from the vector coherent energy gradient:

Geometric Attributes: Program **apparent_cmpt**





Definition of apparent component orientation

Like all other AASPI programs, the dip azimuth ranges between -180° and +180° and therefore the apparent dip components, are defined clockwise from North, where North is at 0°, East at 90°, West at -90°, and South at ±180°. Note that the apparent dip or amplitude gradient at azimuth φ is the negative of the apparent dip or amplitude gradient at azimuth φ+180° so that typically you should only compute components between ±90°

Output file naming convention

If the *inline and crossline dip* components are selected program **apparent_cmpt** will generate the following output files:

Output file description	File name syntax
Program log information	apparent_dip_*unique_project_name_suffix*.log
Program error/completion information	apparent_dip_*unique_project_name_suffix*.err
Apparent dip component at phi degrees	apparent_dip_*unique_project_name_suffix_phi*.H

where the values in red are defined by the program GUI. There will be multiple apparent dip files generated where the values of phi are determined by *First output azimuth, Last output azimuth*, and *Output azimuth increment*. 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 the *inline and crossline energy-weighted gradient* components are selected program **apparent_cmpt** will generate the following output files:

Output file description	File name syntax
Program log information	apparent_energy_weighted_gradient_*unique_project_name_suffix*.log
Program error/completion information	apparent_energy_weighted_gradient_*unique_project_name_suffix*.err
Apparent dip component at phi degrees	apparent_energy_weighted_gradient_*unique_project_name_suffix_phi*.H

Whereas for the *inline and crossline RMS-amplitude-weighted gradient* components are selected program **apparent_cmpt** will generate the following output files:

Output file description	File name syntax
Program log information	apparent_RMS_amplitude_weighted_gradient_*unique_project_name_suffix*.log
Program error/completion information	apparent_RMS_amplitude_weighted_gradient_*unique_project_name_suffix*.err

Apparent dip component at phi degrees	apparent_RMS_amplitude_weighted_gradient_*unique_project_name_suffix_phi*.H

As with the apparent dip option, there will be multiple apparent gradient files generated where the values of phi are determined by *First output azimuth, Last output azimuth*, and *Output azimuth increment*.

Theory: Computing apparent components

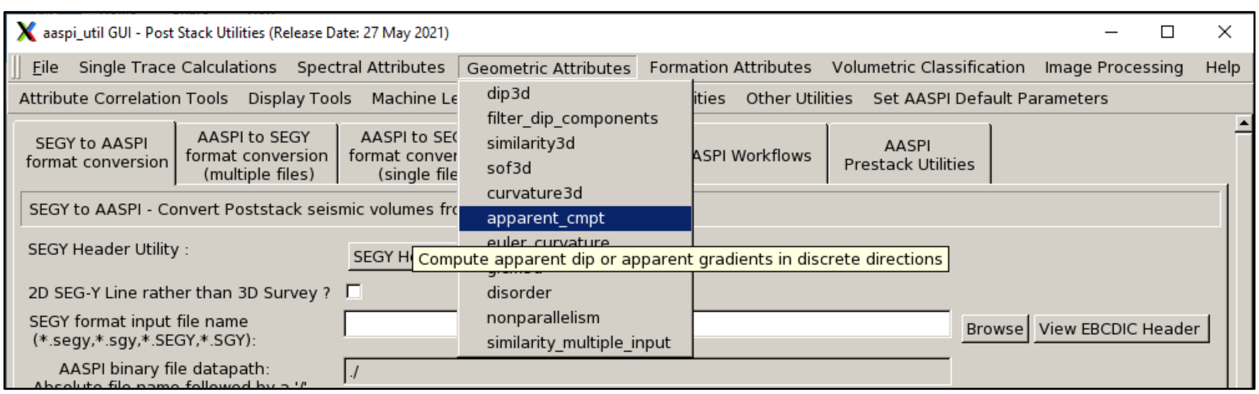
Most modern workstations provide a "calculator" function that circumvents the need to use program apparent dip. If this is the case, the apparent dip at $\varphi=30^0$ is computed using the formula

$$p(\varphi=30^0)=p\cos(30^0)+q\sin(30^0). \quad (1)$$

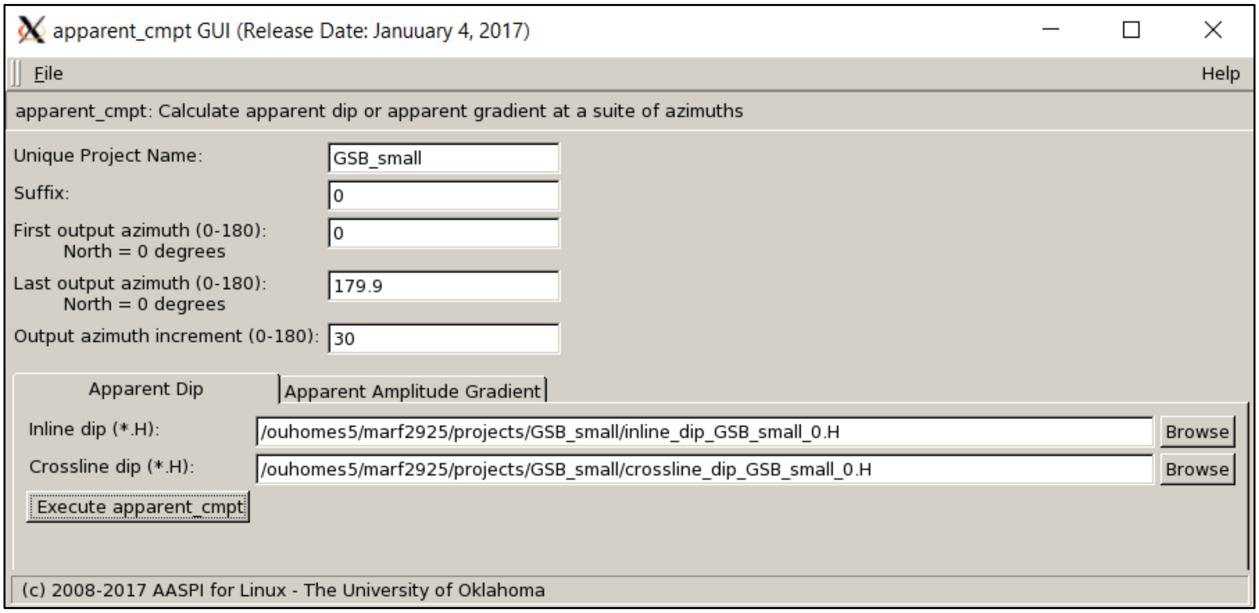
where *p* is the inline dip component and *q* is the crossline dip component computed using program **dip3d**. Program **apparent_cmpt** will compensate for rotated and counterclockwise survey axes.

The apparent_component GUI

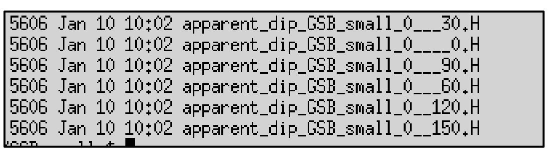
To invoke program **apparent_cmpt**, click the *Geometric Attributes* tab and select **apparent_cmpt**:



The following GUI appears:



The parameters are fairly obvious. Enter the inline dip and crossline dip component file names at bottom of the panel. In this example, I have entered values of 0 for the first azimuth, 150 for the last azimuth, and 30 for the azimuth increment, which will result in apparent dip components at 0^0, 30^0, 60^0, 90^0, 120^0, and 150^0. Note that components at 180^0, 210^0, 240^0, 270^0, 300^0 and 330^0 would be identical but with opposite sign. Program **apparent_cmpt** is not computationally intensive; within a few moments the following output files are generated:

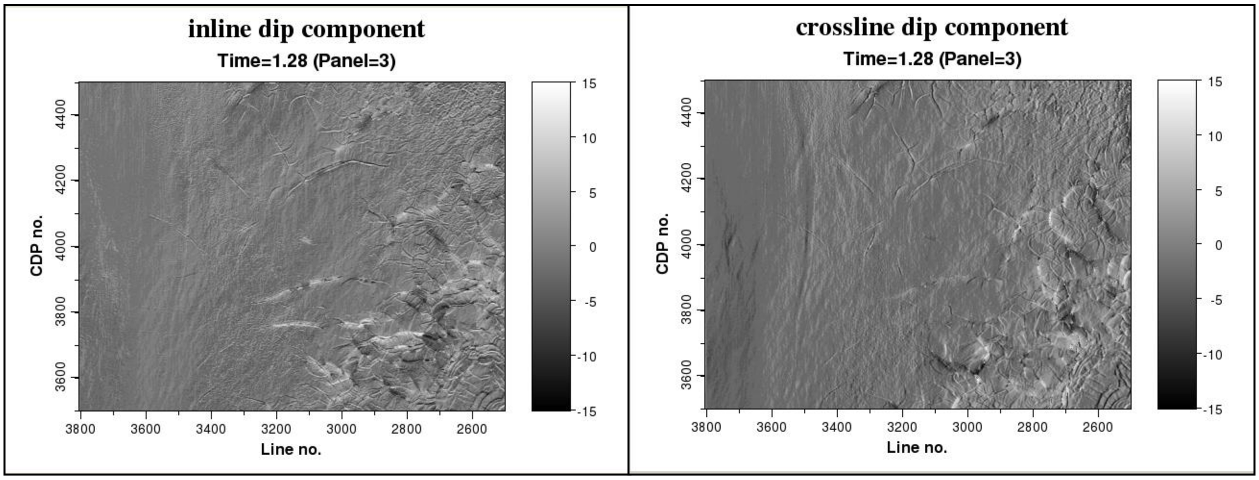


```
5606 Jan 10 10:02 apparent_dip_GSB_small_0___30.H
5606 Jan 10 10:02 apparent_dip_GSB_small_0____0.H
5606 Jan 10 10:02 apparent_dip_GSB_small_0___90.H
5606 Jan 10 10:02 apparent_dip_GSB_small_0___60.H
5606 Jan 10 10:02 apparent_dip_GSB_small_0__120.H
5606 Jan 10 10:02 apparent_dip_GSB_small_0__150.H
```

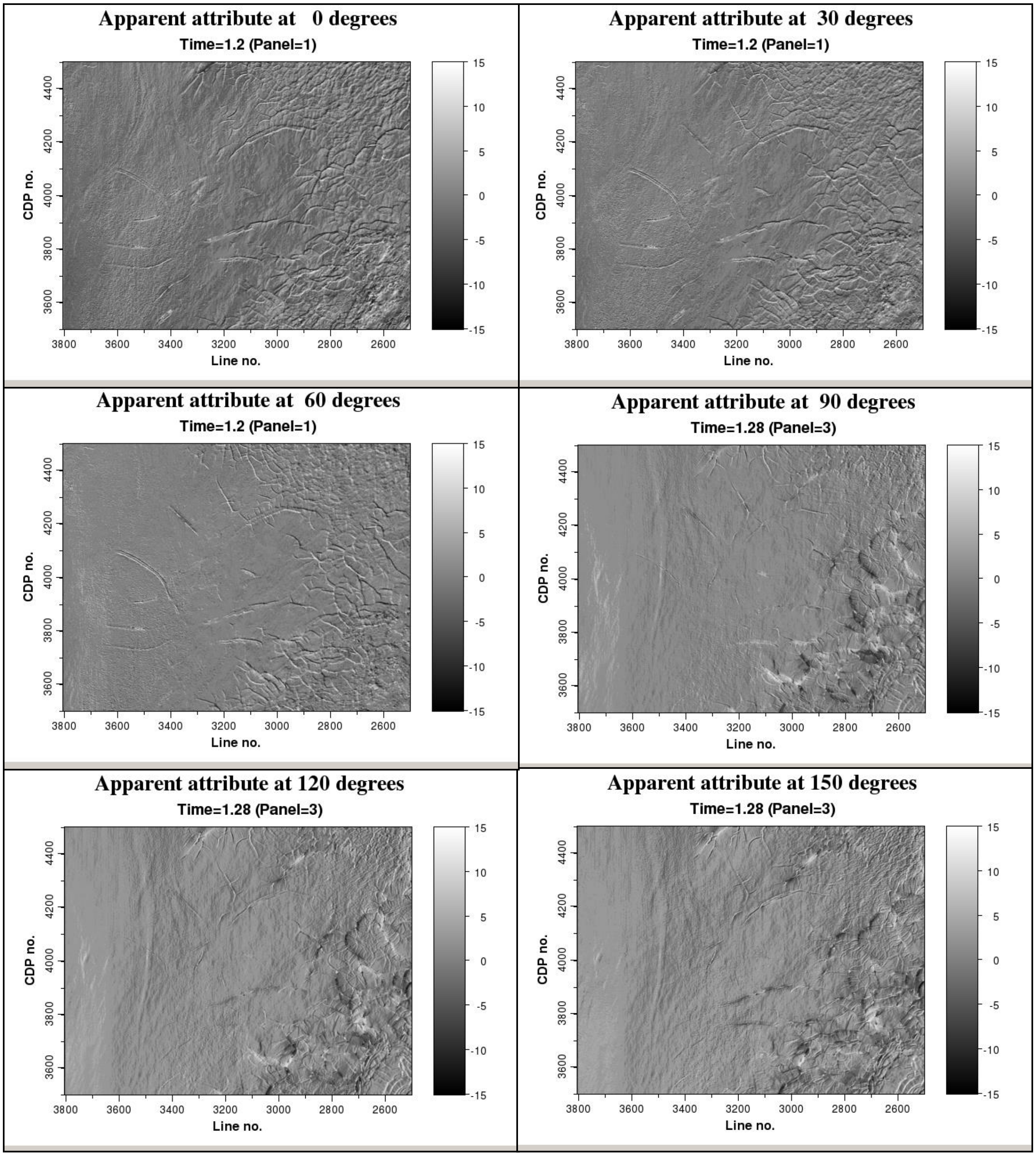
Note how the azimuth values of 0, 30, 60, 90, 120, and 150 have been added to the end of the file name. If the inline axis is oriented North and the crossline axis East, **apparent_cmpt** will reproduce the inline and crossline components of dip at 0^0 and 90^0.

Example 1: Computing apparent dip

For the GSB survey, the inline axis is at $+25^0$ and the crossline axis is at -65^0 (counter-clockwise survey axes). These latter two input files look like the images below:

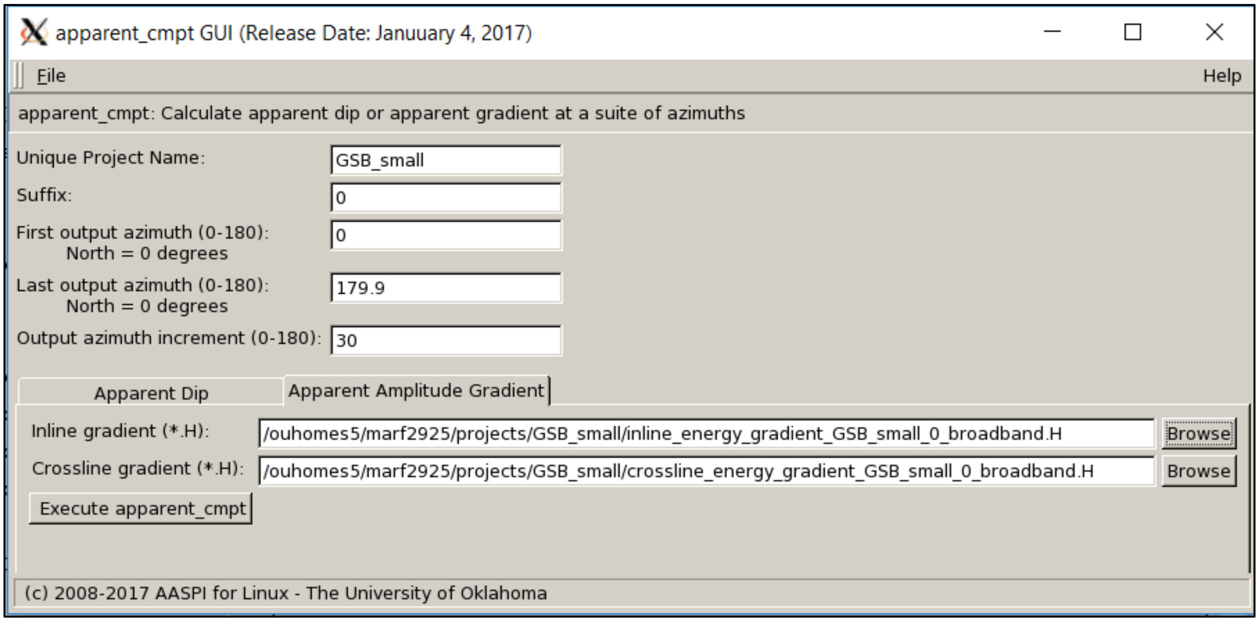


Plotting the results of **apparent_cmpt** at the same time slice, we observe the inline and crossline dip components correctly plotted at 0^0 and 90^0, along with the other components.

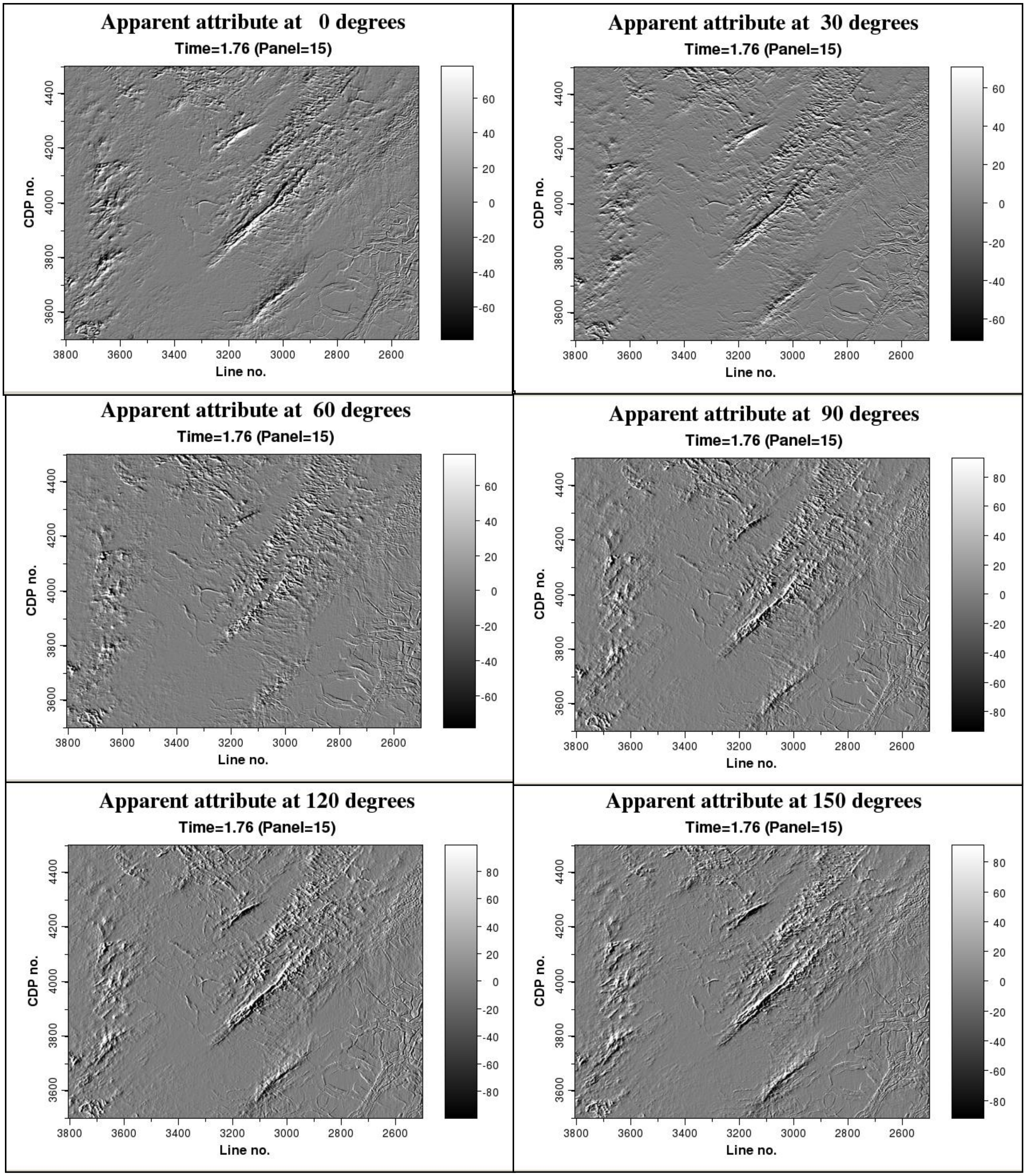


Example 2: Computing apparent energy-weighted amplitude gradients

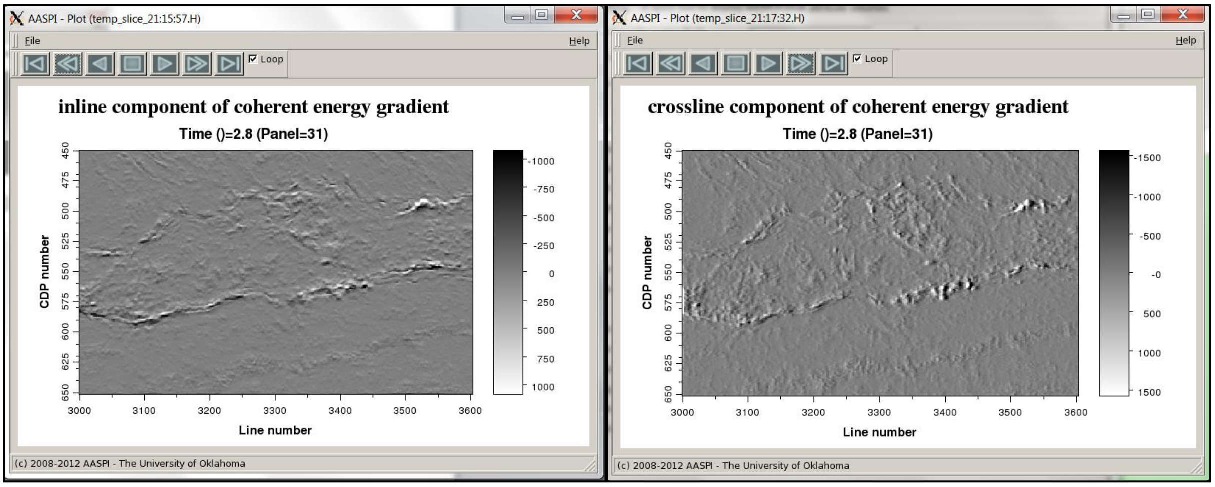
To compute apparent amplitude gradients, simply click the *Apparent Amplitude Gradient* tab and enter the inline and crossline energy gradients:



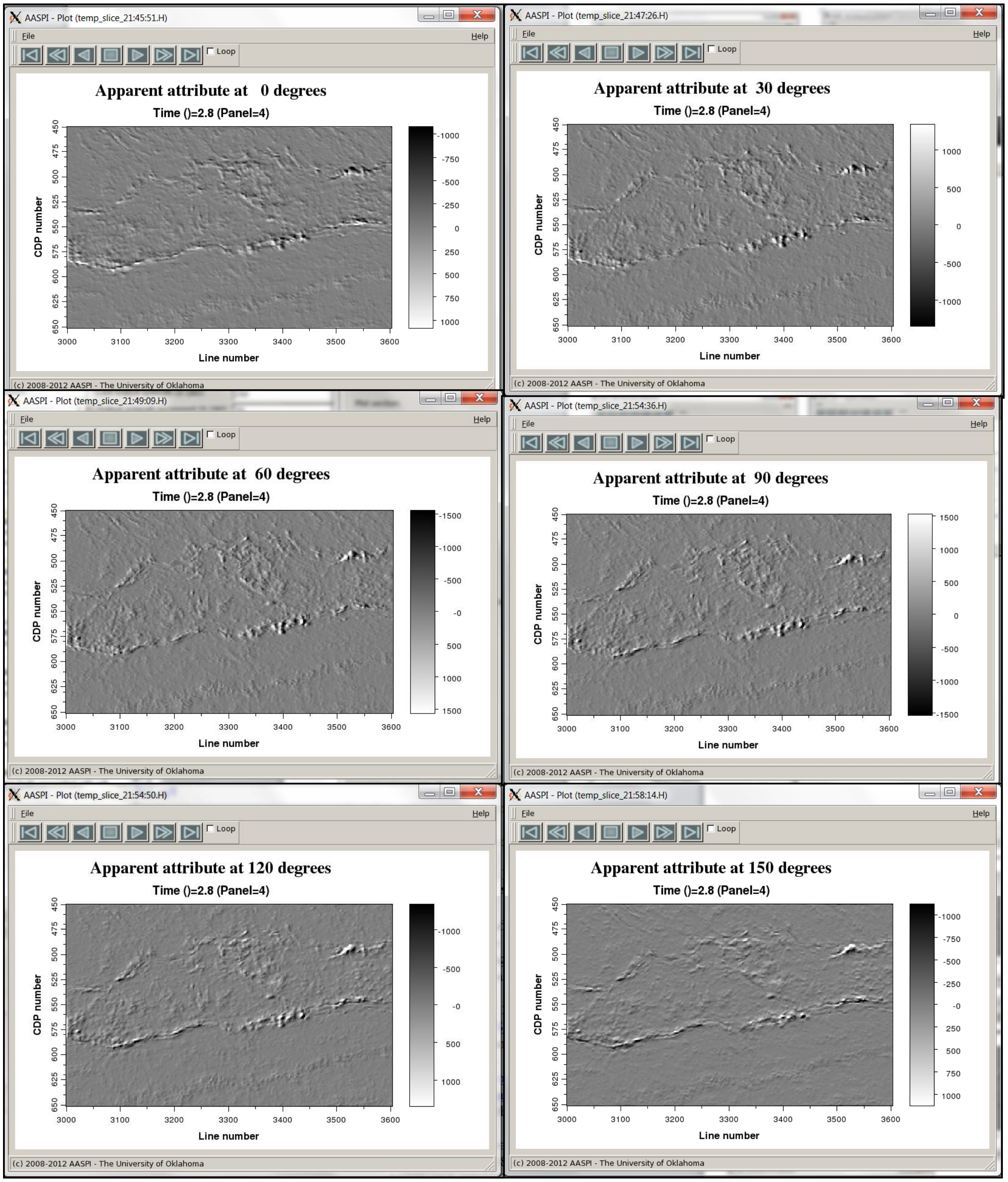
The amplitude gradients for the Great South Basin *GSB_small* survey look like the following images on the next page:


Apparent attribute at 0 degrees
Time=1.76 (Panel=15)
CDP no.
Line no.
Apparent attribute at 30 degrees
Time=1.76 (Panel=15)
CDP no.
Line no.
Apparent attribute at 60 degrees
Time=1.76 (Panel=15)
CDP no.
Line no.
Apparent attribute at 90 degrees
Time=1.76 (Panel=15)
CDP no.
Line no.
Apparent attribute at 120 degrees
Time=1.76 (Panel=15)
CDP no.
Line no.
Apparent attribute at 150 degrees
Time=1.76 (Panel=15)
CDP no.
Line no.

Mai et al. (2008) developed this algorithm to analyze the fractured Cuu Long Basin, Vietnam. As before, we plot the inline gradient (at 160^0) and crossline gradient (at 70^0) through the basement high at t=2.8 s as reference images:



The results of **apparent_cmpt** look the following images on the next page.



References

Mai, H.T., K. J. Marfurt, and T. T. Mai, 2009, Multi-attributes display and rose diagrams for interpretation of seismic fracture lineaments, example from Cuu Long basin, Vietnam: The 9th SEGJ International Symposium Imaging and Interpretation - Science and Technology for Sustainable Development, paper 1D93.