Computing Structural Dip – Program dip3d

Geometric Attributes and dip3d

Overview

There are currently five popular ways of computing estimates of volumetric dip based on discrete semblance scans, the gradient structure tensor (GST), the ratio of smoothed instantaneous wavenumber to smoothed instantaneous frequency, trace-to-trace cross-correlation, and the prediction error filter (sometimes called “predictive painting”). Program dip3d provides solutions based on two of these algorithms: on the GST and on complex semblance scans.

In their original paper on semblance-based coherence, Marfurt et al. (1998) described the need to first search for the local dip, resulting in what today would be called a structure-oriented coherence computation. Later, Marfurt (2006) modified the search to provide a more accurate estimate about faults and under discontinuities by introducing a Kuwahara window construct. For years, the AASPI developers felt that although this method was computationally more intensive than the more popular GST algorithm, that its improved lateral resolution, at least over the implementation originally proposed by Bakker (2002) and Randen et al. (2000). As part of a computational geophysics course held at OU during the Spring semester, 2017, we programmed up the GST algorithm based on its original description, then modified it to use the analytic trace rather than the simple “real” seismic trace, evaluated the improvements identified by Luo et al. (2006), added a Kuwahara (Kuwahara et al., 1976) filter, preconditioned the data by applying a short window ACG filter, and reduced the size of the analysis window size. The result of this effort is a GST-based dip estimation that provides comparable results to the previous semblance-search algorithm, but for volumes with dips exceeding 45°, at significantly less computation cost. In general, the semblance-based algorithm provides slightly better lateral resolution than the GST-based algorithm, while the GST-based algorithm provides slightly better angular resolution than the semblance-based algorithm. Because the two algorithms are quite different in the objective function they measure, their results are slightly different in areas of rapid lateral and vertical changes in dip. In this documentation, we will describe both algorithms in detail and provide insight as to where and why one might encounter differences.

Program dip3d uses a seismic amplitude volume as its input and generates estimates of inline dip, crossline dip, dip magnitude, dip azimuth, and a confidence measure of these estimates. The inline and crossline dip components are critical for almost all subsequent AASPI computations. The dip magnitude and dip azimuth are useful if you wish to display dip azimuth modulated by dip magnitude using the multiattribute display programs hplot, hsplot, hlsplot, or corender. The confidence volume is used to remove artifacts in the dip volumes using program...
**filter_dip_components.** Initially, these input seismic amplitude volume would be either your time- or depth-migrated amplitude or acoustic impedance. However, as we progress through the AASPI software, we shall wish to recompute the dip components from the data that have been subjected to structure-oriented filtering using the program sof3d, or have been spectrally balanced using the programs spec_cmp or spec_cwt.

**Invoking the dip3d GUI**

To begin, at the top of the aaspi_util GUI, click the Geometric Attributes tab, located to the right of the ‘File’ tab. A down drop menu will appear containing the AASPI geometric attribute programs – dip3d, filter_dip_components, similarity3d, sof3d, curvature3d, apparent_cmpt, euler_curvature, glcm3d, and disorder. In general, we will proceed from top to bottom, with the output of dip3d being required input for similarity3d, sof3d, curvature3d, and glcm3d. Note to the right of the aaspi_util GUI there is an AASPI Workflows tab that provides a means to run a set of programs in sequence as a single job, allowing the ability to run a computationally intensive suite of programs overnight, or simply allow time to focus on other tasks.

Clicking dip3d results in the following GUI:
Click (1) Browse, and select \textit{d\_mig\_GSB\_small.H} as your input seismic data set. You also need to (2) choose a unique project name, which will be tacked on to an attribute descriptor to organize the output. We purposely used the project name previously used in the conversion from segy to AASPI format, but this is not necessary. Type in ‘GSB\_small’. Then (3) select a \textit{Suffix}. If this is the original migrated data and you wish to distinguish between it and a data volume that has gone through two passes of structure-oriented filtering, you might give this run a \textit{Suffix} = 0 or \textit{Suffix} = \textit{orig}, and the 2\textsuperscript{nd} run a \textit{Suffix} = \textit{pc\_2}. In this example, I will want to compare the results of using the gradient structure tensor algorithm and give it the \textit{Suffix} = \textit{GST}. Later, I will generate a similar result using the semblance algorithm and give it the \textit{Suffix} = \textit{semblance}.

The toggle bar (4) defines the algorithm to use. The default will be to use the more efficient GST algorithm. (If you click this toggle the algorithm will switch to the semblance search algorithm).
**Simple Data Conditioning**

The input seismic data are slightly conditioned prior to dip computation using either the GST or semblance search algorithms in order to generate more accurate dip estimates.

*Bandpass filtering*

For most interpretation tasks, such as seismic impedance inversion, we wish to maintain as broad a bandwidth as possible. However, some frequency components may have a lower signal-to-noise ratio. Common examples may include high-frequency noise associated with undersampled shallow scatterers that are aliased and therefore overprint lower frequency reflectors of interest, and low “apparent” frequency steeply dipping migration artifacts that are often caused by migration operator aliasing. In both examples, a simple bandpass filter using corner points $f_1, f_2, f_3$, and $f_4$, described in this documentation result in improved volumetric dip estimates.

*Tapered windows*

The vertical seismic axis is typically sampled finer than the horizontal axes (say $\Delta z=5$ m, versus $\Delta x=\Delta y=25$ m for depth-migrated data). For average quality seismic data, a good rule-of-thumb is to choose the full vertical analysis window to approximate the wavelength of the dominant spectral component. For higher quality seismic data that have been spectrally balanced, one may choose a shorter vertical window that corresponds to the spectral component at the high end of the flattened spectrum, thereby minimizing stratigraphic mixing. Internal to these algorithms, the amplitudes used in the vertical analysis window are subjected to a 20% Tukey taper at both the top and bottom, thereby reducing stratigraphic mixing and slightly improving the vertical resolution of the result.

*Short-window AGC*

The goal of both the semblance search and GST algorithms is to determine which dip best aligns the seismic waveforms in a 3D analysis window. The semblance is maximum when wavelengths of the same amplitude are aligned along the candidate dip. In the GST algorithm, the first eigenvector represents the direction in which the amplitude change is maximum. For a constant amplitude reflector, this direction will be perpendicular to the reflector dip. In contrast, the second and third eigenvectors will be orthogonal to each other and parallel to reflector dip, with their corresponding eigenvalues equal to zero. For this reason, a short window AGC will minimize the lateral variation in reflectivity along a reflector, allowing its dip to be more accurately estimated. The length of the AGC window is two times that of the dip analysis window.

**Algorithm Choice 1: Computing Volumetric Dips using the Gradient Structure Tensor**

Because the program structure of semblance search and the GST algorithm are radically different, it is easier to develop and maintain two separate programs that share many of the same subroutines, python scripts, and graphical user interface. The input to program `dip3d_gst` is either a time- or depth-migrated seismic amplitude volume. For 3D data volumes, the output always includes the inline and crossline component of vector dip and the confidence of those estimates. For a suite of 2D data lines the crossline dip is undefined and so will not be computed.
Returning to the GUI above, the Conversion velocity (7) is read in from the file `aaspi_default_parameters`, which is currently set to be 4,000 m/s for data measured in m. For the `GSB_small` data, we may wish to use a slower velocity of about 3,000 m/s. You will always wish to (8) compute the inline and crossline dip components, so this box is disabled. In contrast, you may or may not wish to compute in (9) dip magnitude or (10) dip azimuth. Let’s check this to see what it provides. ‘Want dip confidence result?’ is chosen by default since it will be required if you run program `filter_dip_components`. Asking for any of these files does not influence the run time, just the disk space used. The original GST publications by Bakker (2002) and Randen et al. (2000) define an attribute called (11) `chaos`, which is provided by many of the commercial software packages. At the name implies, low values (i.e. chaos=-1) of chaos correspond to planar events, moderate values (chaos=0) correspond to edges, while high values (chaos=+1) correspond to random data. The final option, (12) `gst similarity`, is a measure of planarity and is used internally in the Kuwahara filtering (and thus for the output confidence result). Details on the construction of the GST, the improvements made in our implementation, and the definition of the output attributes are described in the following four gray theory boxes.
The amplitude gradient vector and the simplest gradient structure tensor (GST)

The vector gradient, \( \mathbf{g} \), is the derivative of the seismic amplitude, \( u \), in each of the three Cartesian directions, which we will approximate using central differences:

\[
\mathbf{g}_{klm} = \begin{pmatrix}
    \frac{\partial u_{k+1,l,m}}{\partial x} \\
    \frac{\partial u_{k,l+1,m}}{\partial y} \\
    \frac{\partial u_{k,l,m+1}}{\partial z}
\end{pmatrix} \approx \begin{pmatrix}
    \frac{u_{k+1,l,m} - u_{k-1,l,m}}{2\Delta x} \\
    \frac{u_{k,l+1,m} - u_{k,l-1,m}}{2\Delta y} \\
    \frac{u_{k,l,m+1} - u_{k,l,m-1}}{2\Delta z}
\end{pmatrix}.
\]

(1)

The gradient structure tensor at grid point \((k,l,m)\), \( \mathbf{C}_{klm} \), is computed by cross-correlating and autocorrelating these three components:

\[
\mathbf{C}_{klm} = \begin{pmatrix}
    g_{x,klm} & g_{y,klm} & g_{z,klm} \\
    g_{x,klm} & g_{y,klm} & g_{z,klm} \\
    g_{x,klm} & g_{y,klm} & g_{z,klm}
\end{pmatrix}.
\]

(2)

within an analysis window.

Note that at a peak or a trough, the value of \( u \) is an extremum and changes slowly. For a flat reflector, the value of \( g_z \approx 0 \). In contrast, the Hilbert transform, \( g_z^H \), at this location is an extremum. We can either augment the GST matrix in equation 2 by adding its Hilbert transform, or follow Luo et al. (2006) and compute the GST as weighted derivatives of the phase.
The gradient structure tensor based on lateral changes in phase

Luo et al. (2006) address the gradient instability problem encountered at peaks and troughs of the seismic amplitude by computing a weighted gradient of the phase, \( \phi \), defined as

\[
\phi_{klm} = \text{ATAN} 2 \left[ \frac{u_{klm}^H}{g_x} \right],
\]

(3)

where the superscript H denotes the Hilbert transform. The derivatives of the phase are poorly defined at ±180°. Taner et al. (1979) computed the derivatives of the arctangent, giving

\[
\left( \frac{\partial \phi_{klm}}{\partial x} \right)^2 = \frac{u_{klm}^H g_x}{u_{klm}^H g_x},
\]

(4)

where the envelope squared

\[
e_{klm}^2 \equiv u_{klm}^H g_x^2 + u_{klm}^H g_x^2.
\]

(5)

Luo et al. (2006) then weight equation 4 by \( e_{klm}^2 \) and obtain a weighted phase gradient vector, \( \Gamma \):

\[
\Gamma_{klm} = \left( \frac{\partial \phi_{klm}}{\partial x} \right)^2 = \frac{u_{klm}^H g_x}{u_{klm}^H g_x} - u_{klm}^H g_x.
\]

(6)

Using this measure, their GST becomes

\[
C_{klm} = \begin{pmatrix}
\Gamma_{x,klm} & \Gamma_{x,klm} & \Gamma_{x,klm} \\
\Gamma_{y,klm} & \Gamma_{y,klm} & \Gamma_{y,klm} \\
\Gamma_{z,klm} & \Gamma_{z,klm} & \Gamma_{z,klm}
\end{pmatrix}.
\]

(7)

Averaging values over \( N \) traces and ±\( K \) samples gives:

\[
\overline{C} = \frac{1}{N(2K + 1)} \sum_{k=-K}^{K} \sum_{n=1}^{N} C_{k,n,m(n)},
\]

(8)

where \( k(n) \) and \( m(n) \) indicate the indices in the x and y directions for trace \( n \).

Numerical experiments show that subtracting the mean gradient vector \( \Gamma_{mean} \) from the \( \Gamma \) before computing equation 7 results in inferior results.
The gradient structure tensor based on complex amplitude gradients

An alternative improvement to the GST that is less elegant but somewhat simpler and provides slightly better results than that proposed by Luo et al. (2007) is to compute the GST using the gradients of both the original data and of its Hilbert transform:

\[
C_{klm} = (e^{ilm})^2 \left[ \begin{array}{cccc}
(g_{x,k}^{klm} - \mu_x)(g_{x,k}^{klm} - \mu_x) & (g_{y,k}^{klm} - \mu_y)(g_{y,k}^{klm} - \mu_y) & (g_{z,k}^{klm} - \mu_z)(g_{z,k}^{klm} - \mu_z) \\
(g_{y,k}^{klm} - \mu_y)(g_{x,k}^{klm} - \mu_x) & (g_{y,k}^{klm} - \mu_y)(g_{y,k}^{klm} - \mu_y) & (g_{z,k}^{klm} - \mu_z)(g_{z,k}^{klm} - \mu_z) \\
(g_{z,k}^{klm} - \mu_z)(g_{x,k}^{klm} - \mu_x) & (g_{z,k}^{klm} - \mu_z)(g_{y,k}^{klm} - \mu_y) & (g_{z,k}^{klm} - \mu_z)(g_{z,k}^{klm} - \mu_z)
\end{array} \right] + \left[ \begin{array}{cccc}
(g_{x,H,k}^{klm} - \mu_x^H)(g_{x,H,k}^{klm} - \mu_x^H) & (g_{y,H,k}^{klm} - \mu_y^H)(g_{y,H,k}^{klm} - \mu_y^H) & (g_{z,H,k}^{klm} - \mu_z^H)(g_{z,H,k}^{klm} - \mu_z^H) \\
(g_{y,H,k}^{klm} - \mu_y^H)(g_{x,H,k}^{klm} - \mu_x^H) & (g_{y,H,k}^{klm} - \mu_y^H)(g_{y,H,k}^{klm} - \mu_y^H) & (g_{z,H,k}^{klm} - \mu_z^H)(g_{z,H,k}^{klm} - \mu_z^H) \\
(g_{z,H,k}^{klm} - \mu_z^H)(g_{x,H,k}^{klm} - \mu_x^H) & (g_{z,H,k}^{klm} - \mu_z^H)(g_{y,H,k}^{klm} - \mu_y^H) & (g_{z,H,k}^{klm} - \mu_z^H)(g_{z,H,k}^{klm} - \mu_z^H)
\end{array} \right],
\]

(9)

where

\[
\mu_j = \frac{1}{N(2K + 1)} \sum_{k=K}^{K} \sum_{n=1}^{N} g_{j,k}^{n,m(n)}
\]

(10a)

and

\[
\mu_j^H = \frac{1}{N(2K + 1)} \sum_{k=K}^{K} \sum_{n=1}^{N} g_{H,j,k}^{n,m(n)}
\]

(10b)

are the average gradient and its Hilbert transform in the analysis window, and where \(e^{ilm}\) is the envelope. Averaging values over \(N\) traces and \(\pm K\) samples gives:

\[
\bar{C} = \frac{1}{N(2K + 1)} \sum_{k=-K}^{K} \sum_{n=1}^{N} C_{j,k}^{n,m(n)}
\]

(11)

where \(k(n)\) and \(m(n)\) indicate the indices in the x and y directions for trace \(n\). In contrast the use of the weighted phase gradient, subtracting the mean values of \(g\) and \(g^H\) in equation 9 provides superior images.
Eigenvectors of the GST and dip components

The GST measures the change of seismic amplitude (or alternatively, the weighted change in the phase) in each of the three Cartesian directions. To determine the direction of maximum change, we decompose $\mathbf{C}$ into its eigenvectors and eigenvalues:

$$\mathbf{C} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \mathbf{v}_3^T \end{pmatrix}, \quad (12)$$

where $(\lambda_j, \mathbf{v}_j)$ is the $j$th eigenvalue-eigenvector pair. Since the GST quantifies the 3D change in amplitude, its first eigenvector

$$\hat{\mathbf{n}} = \mathbf{v}_1 = \begin{pmatrix} v_{1x} \\ v_{1y} \\ v_{1z} \end{pmatrix} \quad (13)$$

provides an estimate of the normal to a hypothesized reflector. The apparent dips, $p$ and $q$, measured in m/m (or ft/ft) are then (e.g. Marfurt and Rich, 2010)

$$p = \frac{v_{1x}}{v_{1z}}, \quad \text{and} \quad (14)$$

$$q = \frac{v_{1y}}{v_{1z}}. \quad (15)$$
Chaos, planes, edges, and GST similarity

Chaos

Randen et al. (2000) and Bakker (2002) define chaos to be
\[ c = \frac{2\lambda_2}{\lambda_1 + \lambda_4} - 1. \]  
(16)

End members include the case of a planar reflector where all the variation is defined by the first eigenvector, such that \( \lambda_1 >> \lambda_2 = \lambda_3 = 0 \), resulting in a value of chaos \( c = -1 \). Another end member is when there is no organization, such that the three eigenvectors are nearly equal, \( \lambda_1 \approx \lambda_2 \approx \lambda_3 \), resulting in a value of chaos \( c = +1 \). Finally, if there is an edge, \( \lambda_1 \approx \lambda_2 >> \lambda_3 \approx 0 \), resulting in a value of chaos \( c = 0 \).

Eigenvalue estimates of a plane

In Kuwahara filter dip estimation we wish to determine which of a suite of candidate overlapping windows is most planar. The following measure
\[ r = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \]  
(17)
ranges between \( r=1/3 \) for a random reflector and \( r=1 \) for a planar reflector. Some simple scaling provides the measure
\[ \tilde{r} = \frac{\lambda_1 - \lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \]  
(18)
which shifts the value for a random reflector to be \( \tilde{r} = 0 \).

Eigenvalue estimates of an edge

A measure of the existence of an edge can be written as
\[ f = 2 \frac{\lambda_2 - \lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}, \]  
(19)
which ranges between \( f=1 \) for an edge \( \lambda_1 \approx \lambda_2 >> \lambda_3 \approx 0 \), \( f=0 \) for a plane \( \lambda_1 >> \lambda_2 = \lambda_3 \approx 0 \), and \( f=0 \) for a random pattern \( \lambda_1 \approx \lambda_2 \approx \lambda_3 \).

Planarity, or GST similarity

In order to penalize geometries that better represent an edge, and favor geometries that are more planar, we combine the values \( \tilde{r} \) and \( f \) to define the GST similarity (or planarity)
\[ s = \tilde{r} (1 - f). \]  
(20)

Comparison of Phase-Gradient vs. Complex Amplitude Gradient Estimates of the GST

Initial testing of several examples shows slightly different images provided when using the GST computed by equation 7 based on the phase gradient and equation 9 based on the complex amplitude gradient. The examples below use the same input data and window size of 5 traces by
±5 samples. Unlike the semblance scan algorithm described layer in this documentation, the analysis window does not adapt to a trial dip, but rather is always use a rectangular or elliptical prism with a flat top and bottom. Most parts of the dip field away from the faults are identical. However, green rectangles indicate areas where a given algorithm provides fault edges that are either more focused or more continuous than those indicated by the red rectangles compute using the alternative GST matrix. Comparing the output of several data volumes shows that the GST computed using the complex amplitude gradient is slightly better than that using the weighted phase gradient, such that the former choice is set to be the default setting for the dip3d GUI. Our analysis is by no means exhaustive; for this reason, we retain this alternative estimation of the GST as a choice for those interested in evaluating its effectiveness on their data volume.

Algorithm Choice 2: Computing Volumetric Dips using a Discrete Semblance Search Algorithm

The semblance search is perhaps the first algorithm used to compute volumetric dip (e.g. Marfurt et al., 1998) and is easier to understand, but more difficult to program than the gradients structure tensor described as algorithm choice 1 above. Basically, the user defines a discrete range and increment of candidate dips. The analytic semblance is then computed along each vector dip over a window of traces. That vector dip exhibiting the maximum semblance is the “winner” and serves as the first approximation to the true 3D dip. The algorithm then attempts to improve upon this dip estimate by passing a 2D paraboloid through the semblance values of neighboring dips. If the peak of the paraboloid falls within the window of neighboring dips, the interpolated dip value is used as an improved estimate. If the peak of the paraboloid falls outside the window of neighboring dips, the “extrapolated” dip value is rejected, and the original discrete estimate is used as the output value of dip. The corresponding value of the semblance serves as an output measure of confidence in the search.
Volumetric Attributes: Program dip3d

Seismic amplitude

**dip3d_similarity**

- Inline dip
- Crossline dip
- Confidence

- Dip magnitude
- Dip azimuth
Under the *Primary parameters* tab, note that (5) the default dip search will be from -20° to +20° and (6) the search increment will be 4°. The reflectors in the GSB_small survey are relatively flat, rarely exceeding 10°. The discrete searches will occur over inline dip components, \( p \), and crossline dip components, \( q \), forming a radial grid of discrete dips as described in the algorithm implementation box. Thus, for the values of 20° and 5° above, there will be 91 discrete semblance evaluations along candidate dips. Doubling the dip search range (\( \Theta_{\text{Max}} \)) but retaining the same search increment (\( \Delta \Theta \)) would quadruple the computational effort. Note that this search is more exhaustive than that provided in commercial algorithms which limit the search to inline and crossline \((5+1+5)+(5+5)=21\) discrete semblance evaluations. This more exhaustive search provides superior results for aliased data and very steep dips.
Output files include the inline and crossline components of vector dip for 3D data volumes, a measure of confidence in the dip estimate, and optional estimates of dip magnitude and dip azimuth. Note that chaos and gst similarity volumes are disabled, and can only be computed when using the GST (mode 1 of ) dip computation.
**Semblance computation**

Almost all AASPI algorithms use the analytic trace (the original trace and its Hilbert transform) rather than simply the original trace, thereby avoiding inaccurate measures about zero crossings about an otherwise a strong event, where “strong” can be estimated by the value of the envelope. Taner and Koehler (1969) defined semblance, $s$, as the ratio of the energy of the average trace over that average energy of the individual traces. Generalizing this to definition to be the energy of the analytic trace along dip ($p,q$), we obtain for a window defined by $J$ traces and $\pm K$ time or depth samples:

$$s = \frac{\sum_{k=-K}^{+K} \left[ \frac{1}{J} \sum_{j=1}^{J} d(t - px_j - qy_j, x_j, y_j) \right]^2 + \frac{1}{J} \sum_{j=1}^{J} d^H(t - px_j - qy_j, x_j, y_j) \right]^2}}{\sum_{k=-K}^{+K} \left[ \frac{1}{J} \sum_{j=1}^{J} d(t - px_j - qy_j, x_j, y_j) \right]^2 + \frac{1}{J} \sum_{j=1}^{J} d^H(t - px_j - qy_j, x_j, y_j) \right]^2},$$

(21)

where the superscript $H$ denotes the Hilbert transform of the measured seismic trace, $d$.

An alternative estimate can be made by using the L1 norm (i.e. absolute values) rather than the L2 norm (i.e. squared values):

$$s = \frac{\sum_{k=-K}^{+K} \left[ \frac{1}{J} \sum_{j=1}^{J} d(t - px_j - qy_j, x_j, y_j) \right] + \frac{1}{J} \sum_{j=1}^{J} d^H(t - px_j - qy_j, x_j, y_j) \right]}{\sum_{k=-K}^{+K} \left[ \frac{1}{J} \sum_{j=1}^{J} d(t - px_j - qy_j, x_j, y_j) \right] + \frac{1}{J} \sum_{j=1}^{J} d^H(t - px_j - qy_j, x_j, y_j) \right]},$$

(22)

This latter estimate provides slightly improved results for three reasons. First, since we are interested in estimating whether an event is planar, rather than planar with a constant value, the L1 norm is less sensitive to lateral changes in amplitude along the reflector. Second, the L1-norm is less sensitive to spikes or other high amplitude events in the data, such as ground roll or migration aliasing artifacts. Finally, depending on the dynamic range of the seismic data, the L2 norm may be subject to round off problems when computing the numerators and denominators using an efficient add-drop algorithm. We have observed this latter problem in megamerge surveys where the input data have not been properly balanced.
**Angle interpolation**

The first step in dip estimation is to search over discrete angles. Note in the image below if the angle with the largest semblance is the red angle, it has eight neighbors. In contrast, if the angle with the largest semblance is the green angle, it has six neighbors.

In either case, we define a parabola that intersects each of the neighboring points of the form

\[ s(p, q) = \alpha_1 p^2 + \alpha_2 pq + \alpha_3 q^2 + \alpha_4 p + \alpha_5 q + \alpha_6. \]  

(23)

This equation is evaluated at each of neighboring (8 or 6) discrete angles as well as the center angle, numbered 1, 2, ..., \( n \), to obtain

\[
\begin{pmatrix}
    p_1^2 & p_1 q_1 & q_1^2 & p_1 & q_1 & 1 \\
    p_2^2 & p_2 q_2 & q_2^2 & p_2 & q_2 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    p_n^2 & p_n q_n & q_n^2 & p_n & q_n & 1 \\
\end{pmatrix}
\begin{pmatrix}
    \alpha_1 \\
    \alpha_2 \\
    \alpha_3 \\
    \alpha_4 \\
    \alpha_5 \\
    \alpha_6 \\
\end{pmatrix}
= \begin{pmatrix}
    s_1 \\
    s_2 \\
    \vdots \\
    s_n \\
\end{pmatrix},
\]

(24)

or in matrix form,

\[ \mathbf{P}\alpha = \mathbf{s}, \]  

(25)

with the solution

\[ \alpha = (\mathbf{P}^T \mathbf{P} + \varepsilon \mathbf{I})^{-1} \mathbf{P}^T \mathbf{s}. \]  

(26)

The maximum of the parabola is found by setting

\[ \frac{\partial s}{\partial p} = 2\alpha_1 p + \alpha_2 q + \alpha_4 = 0, \]

(27)

and solving the resulting 2x2 equation using QR factorization.
The Analysis window parameters tab:

Analysis window parameters are found under the Analysis window parameters tab, indicated by the red box on the image below:

By default, the (12) Dip window height will be ±5 samples (for the GSB survey with ∆t=0.004 s, ±0.020 s), and both the (13) Inline window radius the (14) Crossline window radius will be one trace (in this case, 12.5 m by 25 m) or 5 traces total. If you select (15) the Use rectangular vs. elliptical window? option, you will define a rectangular window, which for this example contains nine traces. Larger windows result in smoother dip estimates with a computational cost...
proportional to the number of traces that fall within the window. For this reason, computations using rectangular windows run 1.8 times longer than those using elliptical windows with the same radii. The following gray box provides more detail on the window definition:

**Analysis window description in the AASPI software**

Almost all of the AASPI geometric attributes are computed within a running analysis window. Typically, these windows are defined by their length and width in m (or ft) and height is s (or km, kft, m, ft), and are oriented along structural dip and azimuth. The figure below shows a typical rectangular analysis window. Let’s assume the trace separation is $d_{cdp}=12.5\ m$ in the inline direction and $d_{line}=25\ m$ in the crossline direction. If you wanted to have the same dip resolution in both directions you may choose an inline_window_radius=$2*d_{cdp}=25\ m$ and crossline_window_radius=$d_{line}=25\ m$. The program default is to set the window radii to be the inline and crossline trace spacing (bin size) resulting in a 5-trace analysis circular window. Larger analysis windows result in longer run times, increased angular resolution, and decreased spatial resolution (smearing). If the data are noisy, a circular window radius equal to 2 bins is a good place to start, resulting in a 13-trace analysis window if $d_{cdp}=d_{line}$.

The window on the upper right corresponds to a survey with $d_{cdp}=12.5\ m$ and $d_{line}=25\ m$. A circular window with inline and crossline radii of 25 m therefore contains 7 traces. If we place a check mark in the Use rectangular window? option, we will obtain the window on the right which contains 15 traces.

If the analysis window straddles a fault, the result may be an apparent dip joining two unrelated reflectors on either side of the fault. To address this issue, program dip3d uses a Kuwahara (Kuwahara et al., 1976) filter that examines the value of reflector confidence (planarity for the GST algorithm, and semblance in the discrete dip semblance search algorithm) in a suite of overlapping windows. The window with the largest value of confidence is the winner, with its value of vector dip set to be the output.
**Implementation details: Kuwahara windows**

Programs dip3d, sof3d, and sof_prestack all use a modification of overlapping window parameter estimates introduced by Kuwahara et al. (1976) in medical imaging. The original idea is simple. If an analysis window contains five traces, then there are total of five windows (a centered window and four adjacent, offset windows) that contain the analysis point. In Kuwahara et al.’s (1976) original work and Luo et al.’s (2002) edge-preserving smoothing algorithm, one calculates the mean and standard deviation of each window. That window which has the smallest standard deviation is hypothesized to be less noise-contaminated. The mean of this window is then used as the output for the analysis point. Marfurt (2006) modified this approach for volumetric dip calculations where he used 3D rather than 2D overlapping windows. In addition, the semblance of the analytic signal is used rather than the standard deviation to determine which window is least contaminated by noise. The inline and crossline components of reflector dip of this ‘best’ window are then assigned to be the output at the analysis point. The figure in the lower left represents a 13-trace circular analysis window centered about the analysis point indicated by the red solid dot. Each of the traces represented by the green dots in the figure in the lower right form the center of their own 13-trace analysis windows, yet contains the trace represented by the red dot. The window having the highest analytic semblance best represents the signal; its dip components are therefore as the dip at the red analysis trace.

![Diagram of Kuwahara windows](image)

For reasonably good quality seismic data, place a checkmark in front of (16) **Search lateral windows**. If the data are very noisy, using Kuwahara windows can give rise to a ‘patchy’ appearance on the dip components. If such blockiness is unacceptable, remove the checkmark in front of **Search lateral windows**. For good quality seismic data, place a checkmark in front of (17) **Search vertical windows**. This option will also introduce some blockiness, but avoids smearing angular unconformities, onlap, toplap, and other configurations important to seismic stratigraphy interpretation.

To minimize the ‘patchy’ appearance that can occur when using Kuwahara windows, (18) set a threshold value of the analytic semblance, $s_{\text{upper}}=0.95$. If the semblance of the centered window is greater than the value $s_{\text{upper}}$, its value of dip and azimuth will be used. If the semblance is less
than this value of \( s_{\text{upper}} \), the Kuwahara window concept will be implemented, with the dip components of the (centered, laterally offset, or vertically offset) window having the highest value of semblance assigned to the analysis point. A deeper discussion of Kuwahara windows can be found in the documentation for program sof3d.

As described by equations 18 and 19, we can compute semblance using either an L1-norm or an L2-norm. Referring back to the window definition tab of the dip3d GUI, (20) selecting the L1-norm is the default operation, since it avoids potential numerical roundoff when using a more efficient add-drop algorithm described in the gray box below.

**Efficient semblance computation using an add-drop algorithm**

Examining equations 18 and 19 will show a suite of sums in the numerator and denominator. To understand how to do this computation more efficiently, let’s rewrite it as

\[
s(m\Delta t) = \frac{N(m\Delta t)}{D(m\Delta t)} = \frac{\sum_{k=-K}^{K} u[(m+k)\Delta t]}{\sum_{k=-K}^{K} v[(m+k)\Delta t]},
\]

where for equation 18

\[
u[(m+k)\Delta t] = \left[ \frac{1}{J} \sum_{j=1}^{J} d[(m+k)\Delta t - px_j - qy_j, x_j, y_j] \right]^2 + \left[ \frac{1}{J} \sum_{j=1}^{J} d^H[(m+k)\Delta t - px_j - qy_j, x_j, y_j] \right]^2,
\]

and

\[
v[(m+k)\Delta t] = \frac{1}{J} \sum_{j=1}^{J} [d[(m+k)\Delta t - px_j - qy_j, x_j, y_j]]^2 + \frac{1}{J} \sum_{j=1}^{J} [d^H[(m+k)\Delta t - px_j - qy_j, x_j, y_j]]^2.
\]

At the next time sample, one can compute

\[
N[(m+1)\Delta t] = N(m\Delta t) + \{u[(m+K)\Delta t] - u[(m+K-1)\Delta t]\},
\]

\[
D[(m+1)\Delta t] = D(m\Delta t) + \{v[(m+K)\Delta t] - v[(m+K-1)\Delta t]\},
\]

and

\[
s[(m+1)\Delta t] = \frac{N[(m+1)\Delta t]}{D[(m+1)\Delta t]},
\]

where the computation requires only 2 rather than 2K+1 summations over \( u \) and \( v \). The add-drop algorithm is routinely used in AGC scaling and time-variant filtering. In general, this computational construct works very well for semblance as well. However, care must be taken to first scale the data to avoid round-off errors as we accumulate the sums. Such potential problems are avoided by using the default algorithm for semblance search using the L1-norm (equation 19) rather than the L2-norm in computing semblance described by equations 18.

Returning to the window parameters GUI, in the GST mode 1, we will always (20) Remove mean from window, such that this option will be disabled. In the semblance search mode 2, we will in general not need to remove the mean prior to semblance calculation, since the seismic amplitude data have zero mean. However, if we wish to compute dip from an impedance inversion or Poisson’s ratio volume, we will need to select this option to provide an accurate estimate.
The (21) four frequencies \( f_1, f_2, f_3, \) and \( f_4 \), allow a means to apply a simple Ormsby filter to the data. Default frequency values (or wavenumbers for depth-migrated data) pass all frequencies, from zero to Nyquist. However, if there are problematic steeply dipping low apparent frequency migration artifacts or ground roll in the data, it may help to set the values of \( f_1 \) and \( f_2 \) above the noise level.

The final parameters (22) First Line Out, Last Line Out, First CDP Out, and Last CDP Out, provide a means to compute dip estimates on a smaller piece of the data for parameter testing.

**Parallelization Parameters**

The parallelization parameters are the same for all parallel AASPI algorithms and has its own documentation page.

**Algorithm Comparison and Parameter Sensitivity**

The results from the alternative algorithms are in general comparable, with results dependent on the particular data set being analyzed.

**Execution**

After selecting all your parameters, type ‘Execute dip3d’. Intermediate output comes to the xterm from which you launched aaspi_util as well as in the file dip3d_GSB_small.out. Some of the output appears below:
The computation is computed using a stencil, with the amount of work divided equally across (in this example 24 processors or core). Thus, the 1001 CDPs of the survey are spread equally across the 24 processors, most working on 42 CDPS, the others on 41 CDPS. As each seismic line is completed, the results are sent to master processor (processor 0), saved in an array, and written to disk. Progress of the program is shown in the bottom of the file above where the master node (indicated by the prefix “0:”) indicating the first_line_out, current_line, and last_line_out and an estimated time of arrival (completion) ETA measured in hours. Near the bottom of the *.out file you will see information like this:
which give some computational statistics for each process (in this case process 20, with a wall clock time of 0.169 hrs. Memory is deallocated, and each process echoes out that it has completed normally:

We’ve already discussed the *dip3d_GSB_small_0.out* file that is a copy of the results that came to the screen. *dip3d* also has created the following files:

```
20 send loop over lines
20 number of traces processed: 25891

<table>
<thead>
<tr>
<th>process</th>
<th>task</th>
<th>time (hr)</th>
<th>time/trace (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20:</td>
<td>read and scale data</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>send data via MPI</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>receive data via MPI</td>
<td>0.024</td>
<td>0.003</td>
</tr>
<tr>
<td>20:</td>
<td>send results via MPI</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>receive results via MPI</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>hilbert transform</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>scan discrete dip</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>interpolate dip</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>write results to disk</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>20:</td>
<td>total time</td>
<td>0.188</td>
<td>0.023</td>
</tr>
</tbody>
</table>
```

```
As described in the documentation on the AASPI software environment, each of the AASPI-format files `inline_dip_GSB_small.H`, `crossline_dip_GSB_small.H`, `dip_azimuth_GSB_small.H`, `dip_magnitude_GSB_small.H`, and `conf_GSB_small_0.H`, (the history files) will have a corresponding *.H@@ (header file format) in the local directory, and corresponding *.H@ (binary results), and *.H@@@ (binary trace headers) that reside in the directory you defined earlier in the .datapath file.

We can plot our dip components to QC the results by clicking (1) the AASPI QC Plotting tab in the aaspi_util GUI:

![AASPI utility GUI](image)

where (1) the file `inline_dip_GSB_small_0.H` is selected and (2) will be plotted as time slices:
We have also computed volumetric estimates of dip magnitude and dip azimuth. Using AASPI program corender (under the Display tab) where (1) the base layer is set to be (2) the dip azimuth plotted against (3) a cyclical color bar with values ranging between (4) -180° and +180°. (5)
Change the default title to represent the images co-rendered. Then (6) click the tab for layer 2, and then (3) chose the dip magnitude file which will be plotted against (8) a monochrome gray color bar with (9) transparency set such that high values of dip magnitude will be transparent, allowing the underlying dip magnitude value to show through. Finally, (10) set the default dip magnitude color range to be $0^\circ-15^\circ$. Smaller ranges will result in “brighter” colors.

You should obtain the following image:
To co-render with energy-ratio similarity (computed in program similarity3d), return to program corender and (11) choose the Layer 3 tab, (12) select the energy-ratio similarity volume, (13) plot it against a monochrome black color bar, and (14) set the high coherence values to be transparent, allowing the underlying co-rendered dip azimuth vs dip magnitude images to show through. Finally, (15) adjust the range of the similarity and (16) modify your plot title:
One can also plot the results in the vertical slice. As before, use dip azimuth as the base layer, plotted against a cyclical color bar, dip magnitude as the second layer plotted against a monochrome gray color bar, with high dip magnitude values transparent, but now in program **corender**, under the **Layer 3** tab, (12) select the energy-ratio similarity volume, (13) plot it against a **binary black white** color bar, and (14) set the low values about amplitude zero crossings to be transparent, allowing the underlying co-rendered dip azimuth vs. dip magnitude images to show through. Next, (15) choose **Statistical Ranging** of the color bar, and (16) set the **Percentage Clip** =90, so that the color bar ranges between the 5 and 95 percentile ranges of the amplitude data. Finally, (17) modify your plot title and plot vertical slices:
Volumetric Attributes: Program dip3d

Here are two representative images:

![Corendered dip azimuth, dip magnitude, amplitude](image1)

![Corendered dip azimuth, dip magnitude, amplitude](image2)
at an approximate 1:1 scale, using the value of velocity used in the calculation:

Using program **apparent_cmpt**, we can compute apparent dip at any arbitrary azimuth, \( \phi \), using the simple formula

\[
p_{\text{app}}(\phi) = p \cos(\phi) + q \sin(\phi).
\]

The **apparent_cmpt** GUI is simple:

![apparent_cmpt GUI](image)

where here, the GUI asks to compute apparent dips between 0\(^0\) and 179.9\(^0\) at 30\(^0\) increments, resulting in six files:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Absolute Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSB_small</td>
<td>/ohomes5/mar2925/projects/GBS_small/inline_dip_GSB_small_0.H</td>
</tr>
<tr>
<td>GSB_small</td>
<td>/ohomes5/mar2925/projects/GBS_small/crossline_dip_GSB_small_0.H</td>
</tr>
</tbody>
</table>

Note that if one computed to 359.9\(^0\) that the next six values will be numerical inverses of those in the opposite direction. The inline azimuth of the GSB survey is +25\(^0\) while the crossline azimuth is oriented clockwise at -65\(^0\). Plotting the results of **apparent_cmpt** at the same time slice, we see the inline and crossline dip components correctly plotted at 0\(^0\) and 90\(^0\), along with the other components.
Volumetric Attributes: Program dip3d

1.1

Attribute-Assisted Seismic Processing and Interpretation - 30 January 2018
Volumetric Attributes: Program dip3d

Notice, there are some small glitches in dip estimation. We can compute the confidence that we have at each voxel of the dip estimate. Since program dip3d uses a Kuwahara multiple overlapping window dip search, the confidence is simply the coherence (in this case semblance of the analytic data) of the window used.

Overall, this is a high quality seismic data volume. Not surprisingly, the lowest confidence is adjacent to the major discontinuities. Deeper in the survey, we some areas (seen as black) where our dip estimates are less accurate:
We will try to further improve our dip estimates with some simple median filtering using program `filter_dip_components`.

**References**


