What is phase?

Steve Purves, Euclidity, Tenerife, Spain
steve@euclidity.com

Phase permeates seismic data processing and signal processing in general. But it can be awkward to understand, and manipulating it directly can lead to surprising results. It doesn't help that the word is used to mean a variety of things depending on whether we are referring to the propagating wavelet, the observed wavelet, post stack seismic attributes, or an entire seismic dataset. It's a subject that leads to a lot of discussion, so much so that we have publications dedicated to dealing with the ambiguities (e.g. Liner 2002, Roden 1999, Simm 2002).

In this tutorial, we'll focus on aspects of phase relevant to the interpreter. We'll look at how to manipulate the phase of a seismic trace by manipulating the phase of its Fourier transform. We'll use that functionality to generate the well-known *instantaneous phase* post-stack attribute. And we'll check how close to zero-phase our test data set is.

The plots included in this tutorial were created using GNU Octave (***http://www.gnu.org/software/octave/***) and the code to produce them, some of which is also included here, is available in the SEG tutorials GitHub repository (*http://github.com/seg*). The data we are using is the Penobscot 3D survey from the Open Seismic Repository (*https://opendtect.org/osr/*), which is openly licensed CC-BY-SA by the Canada Nova Scotia Offshore Petroleum Board, and dGB Earth Sciences.

The Hilbert transform

The Fourier transform is complex. Taking the transform of any real signal will result in a set of complex coefficients. Complex numbers are essentially 2D vectors, meaning they have 2 components: magnitude and phase angle. Most of the time when dealing with Fourier transforms we concentrate on the magnitude, which tells us about the distribution of signal energy through frequency. But every signal also has a phase spectrum and it is the phase that actually encodes the signal's structure — the distribution of the signal energy through time. We don't often examine phase spectrum as it is difficult to interpret, but we can manipulate Fourier phase to change the structure of our signal without affecting its amplitude spectrum.

The Hilbert transform is a linear operator that produces a 90° phase shift in a signal and it's a good first step in our exploration of phase. It is also commonly used in post-stack seismic analysis to generate the analytic signal from which we can compute the standard complex trace attributes such as envelope, instantaneous phase, and instantaneous frequency.

The definition of the Hilbert transform is rather cryptic; it is much easier to consider in terms of its Fourier transform definition. The Hilbert transform H of a signal u is related to the Fourier transform $\mathcal{F}$ like so:

$$\mathcal{F}\big(H(u)\big)(\omega) = \sigma_H(\omega) \cdot \mathcal{F}(u)(\omega) \tag{1}$$

$$\text{where } \sigma_H(\omega) = \begin{cases} i & \text{for } \omega < 0 \\ 0 & \text{for } \omega = 0 \\ -i & \text{for } \omega > 0 \end{cases} \tag{2}$$

So we apply a Hilbert transform by multiplying all negative frequencies by i and all positive frequencies by $-i$, leaving any DC component untouched. This is less than intuitive but we can gain some additional insight by thinking of this geometrically.

If we multiply any complex number a (such as a Fourier coefficient) by a second complex number b (such as σ above) this produces a complex number c which is a rotated by the angle of b. So to rotate any complex number by an angle, α, multiply it by the complex number $e^{i\alpha} = \cos(\alpha) + i.sin(\alpha)$.

Armed with that knowledge, and Euler's identity, $i = e^{i\frac{\pi}{2}}$, we see that in equation (2) above we are actually rotating the Fourier coefficients by $\frac{\pi}{2}$ or 90 degrees, and when we take the inverse Fourier transform of those modified coefficients we produce a 90 degree phase shifted version of our original signal.

Furthermore, we can generalise the definition of the Hilbert above to produce a phase shift to any angle, α.

$$\sigma_H(\omega) = \begin{cases} e^{i\alpha} & \text{for } \omega < 0 \\ 0 & \text{for } \omega = 0 \\ e^{-i\alpha} & \text{for } \omega > 0 \end{cases} \tag{3}$$

Phase shifting in GNU Octave

So let's use this to phase shift a seismic trace. We'll take the fast Fourier transform (FFT) of our signal, manipulate the coefficients, and apply the inverse FFT. The following code implements this for an array `x` (see `fftshifter.m` in the repo):

```
% Create an array to apply to the coefficients
% positive and negative frequencies appropriately
R0 = exp(i*phase_shift_in_radians);
N = length(x);
R = ones([1 N]);
M = ceil((N+1)/2);
R(1:M) = R0;
R(M+1;N) = conj(R0);

% Apply the phase shift in the frequency domain
Xshifted = R.*fft(x);
% Recover the shifted time domain signal
y = real(ifft(Xshifted));
```

Complex trace attributes

So now we have this ability to phase shift what can we do with it? Well we can compute some seismic attributes; there's a whole set of commonly used attributes that depend on the Hilbert transform:

```
% Load a seismic trace
% This trace was exported from OpendTect as a Simple File
filename = 'data/trace_il1190_xl1155.trace'
[s, t] = load_simple_trace(filename);
% Compute the analytic (complex) trace
z = s + i*fftshifter(s,-pi/2);
envelope = abs(z);
phase = angle(z);
```

The code above generates the Hilbert transform, envelope, and phase traces for a trace from the Penobscot dataset. A section of this trace is shown in Figure 1 (see *plot_complex_attributes_on_a_trace.m* for full plotting code).

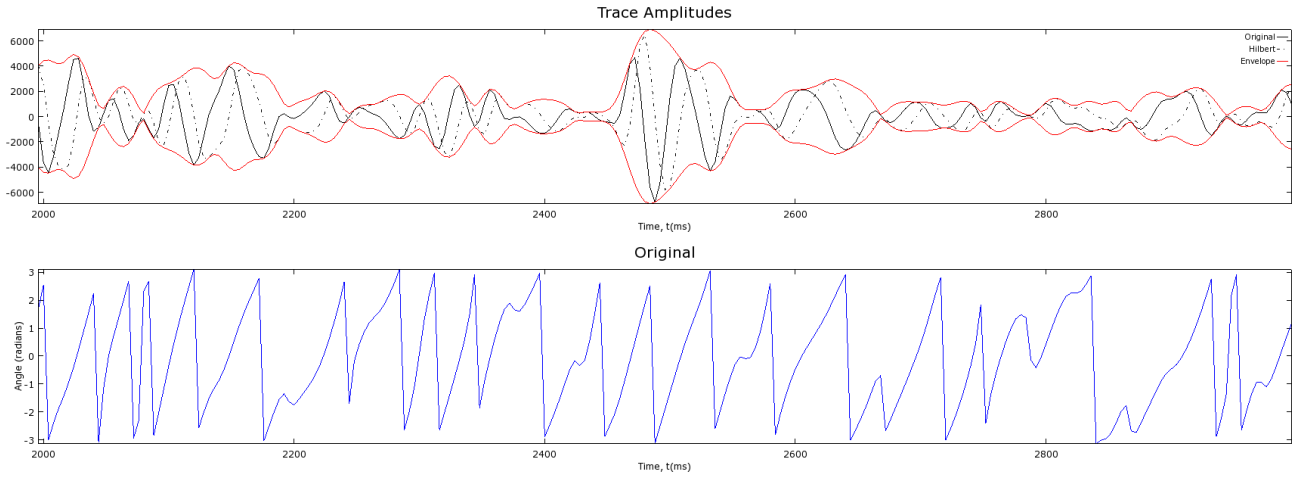

Figure 1 – The complex trace attributes envelope and instantaneous phase calculated on a trace from the Penobscot 3D seismic dataset (il:1190, xl:1155). (top) envelope is shown in red bounding the different phase shifted versions of the trace. The large central package is the Abenaki unconformity. (bottom) Instantaneous phase, a generally well behaved monotonically increasing function where dominant reflectors are well resolved.

Once you have the complex trace representation, `z`, there are many additional attributes that you can compute such as instantaneous frequency, phase acceleration, envelope weighted frequency just by playing around with the complex trace.

Furthermore, we can easily extend our analysis from a single trace to compute attributes on 2D, 3D, or 4D data by just iterating over traces in the dataset we load. Figure 2 shows the result of the same code operating on a 2D section (see *plot_complex_attributes_on_a_slice.m*).

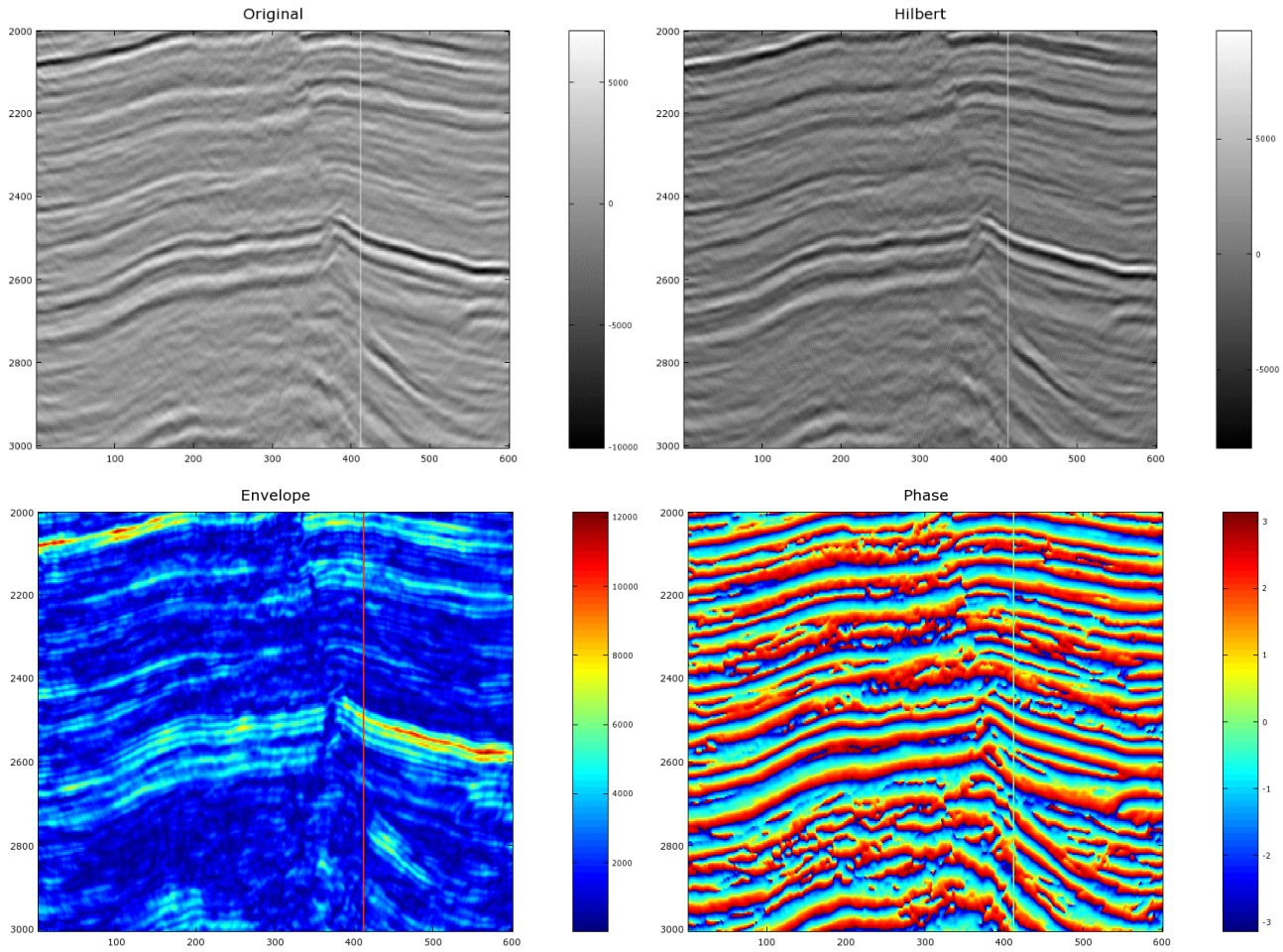

Figure 2 – Complex trace attributes computed on crossline 1155 from Penobscot 3D. The vertical line shows the position of the trace form Figure 1. (bottom-left) the Abenaki unconformity runs through the bright reflector package at around 2500-2600ms.

Checking the phase of your seismic

The complex attributes generated from a seismic trace and its Hilbert transform are useful in seismic interpretation. However, the quality of our seismic interpretation depends on another aspect of phase and that is the phase of our seismic dataset, and whether the observed wavelet have been corrected to a zero-phase response during processing.

Both Liner (2002) and Roden & Sepulveda (1999) highlighted this importance and Roden & Sepulveda in particular gave a practical way to assess the phase of your data as a quality check for your seismic interpretation. We'll look at this more closely here.

Roden & Sepulveda pointed out that where we have a strong reflection from a sharp geological interface that we can confidently expect to produce a zero phase response, we can measure the actual response in the seismic dataset on that horizon to determine the amount of phase error present. We can then use that information to at least understand the quality of the picks we make but also possibly to apply a simple correction to a dataset for a given set of reflections.

Examining local phase

The principle Roden applies is that at strong reflections in zero-phase data we would expect peaks in the trace to align with peaks in the envelope. So any horizon that we pick on zero phase data could

also be picked on the envelope of that data. Figure 3 shows a segment of our trace from Figure 1 with envelope peaks marked as spikes (see *find_peaks.m*).

In the bottom half of the plot we've isolated those events and plotted the actual signal phase against the phase we'd expect for a zero-phase dataset, the difference between the two is the phase error present in the trace.

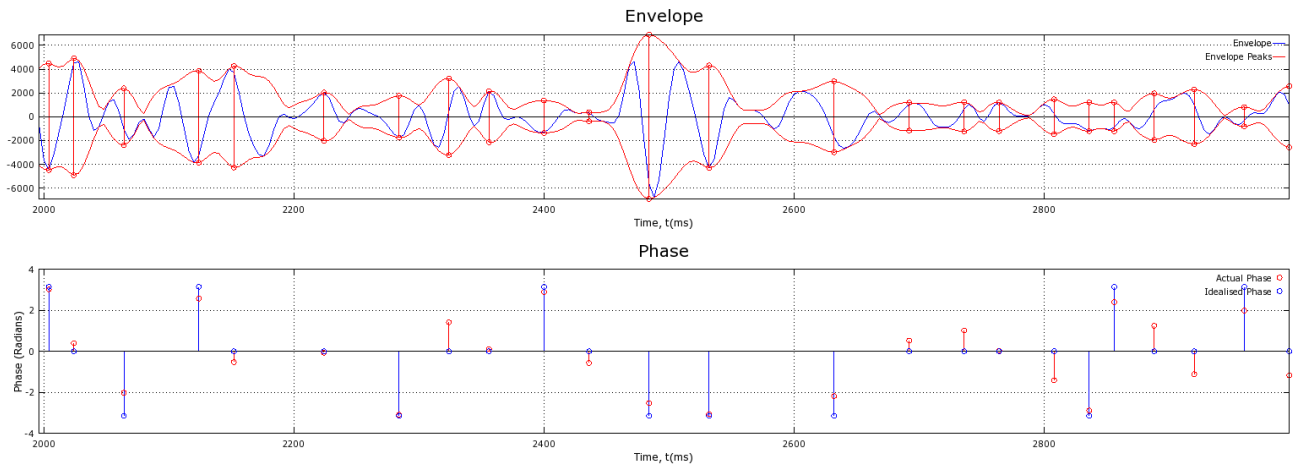

Figure 3 – (top) complex trace with envelope peaks marked. (bottom) phase at envelope peaks plotted alongside equivalent phase response for a zero-phased dataset. Difference between the two shows local phase error as described by Roden & Sepulveda. If we look at the phase error across the trace is clear that we need to take into account the nature of the reflector when interpreting this.

So we see at the Abenaki envelope peak we have a phase error of 0.64 radians on this trace, something that we may want to correct or at least take into account. So what can we do with this information? We can create sections or cubes that highlight sparse reflection boundaries and mark these with our phase error, or we can generate maps to understand how close to zero-phase our data are on the event we are interpreting.

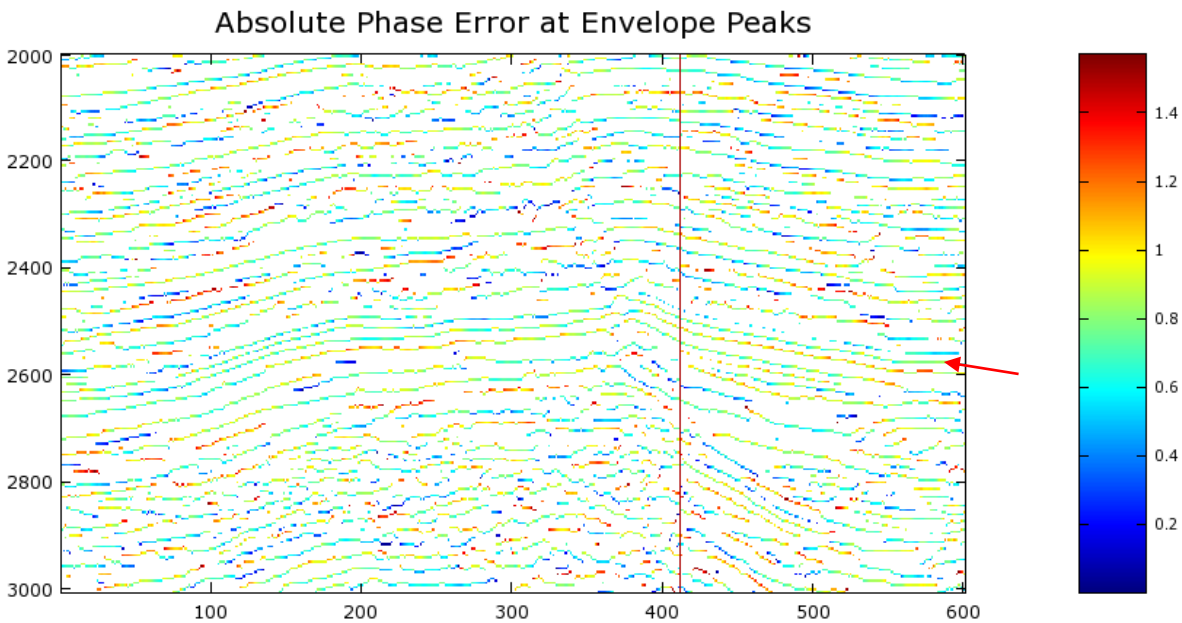

Figure 4 – absolute phase error plotted at envelope peaks. Red arrow marks the Abenaki reflector.

References

Liner, C (2002). Phase, phase, phase. *The Leading Edge* **21**, p 456–7.

Roden, R and H Sepulveda (1999). The significance of phase to the interpreter: practical guidelines for phase analysis. *The Leading Edge* **18**, July 1999, p 774–777.

Simm, R and R White (2002), Tutorial: Phase, polarity and the interpreter's wavelet. *First Break* **20** (5), p 277–281