Timath and frmath: experimental trace ensemble modules for ProMAX

David C. Henley

ABSTRACT

Much of the experimental software developed by CREWES is implemented within the MATLAB environment because of the sophistication of the MATLAB mathematical structure and the large library of functions and algorithms available. For testing programs in a production seismic environment, however, using many data sets of varying sizes and characteristics, it is often preferable to use an established standard seismic processing package such as ProMAX to apply a new algorithm. While several more mature CREWES algorithms have been embodied in their own ProMAX modules, it is time-consuming and often tedious to write a complete new module to test each new idea, particularly if the idea is relatively simple, or the concept being tested is speculative. Hence, we have created two new ProMAX modules intended to apply computations of virtually any kind to the data samples in a 2D array (trace ensemble). We introduce these modules here and demonstrate their currently available functions. We expect that they will find value as research tools as we continually add new test functions to their menus.

INTRODUCTION

ProMAX is a proven seismic industry standard package for processing seismic data, and hence has sophisticated I/O, data display, and database functions that usually ease the burden for those needing to process significant amounts of seismic data. Hence, we at CREWES currently release some of our more mature processing algorithms as ProMAX modules. The drawback of this approach for us, however, is that writing a complete ProMAX module requires significant time and background knowledge. Once a module is written, compiled, and actually running, however, incremental modifications and tests are quite easy and fast.

The difficulty of initial module construction and the comparative ease of subsequent modification suggested to us the idea of generic toolbox modules, which would be initially constructed with one useful function each, but could be subsequently expanded to include other menu-selected functions, as desired. Since there are already standard ProMAX functions which allow 1D mathematical operations on the samples of single traces, or simple operations between pairs of traces, we decided to create two modules, each of which would operate on a complete 2D ensemble of input traces, thus allowing multi-channel operations. One of the modules, Timath, is dedicated to array operations on samples in the time domain; while the other, Frmath, computes the 1D Fourier transforms of each trace of the input ensemble, to enable frequency domain array operations, before inverse transforming the data back to the original trace ensemble.

TIMATH

Timath is a ProMAX module which provides access to the samples of a complete ensemble of seismic traces so that any 1D or 2D mathematical algorithm can be applied to them in Fortran, using the full library of functions and subroutines available within the

ProMAX environment. All data I/O, including trace header storage, is handled by ProMAX itself and the main Timath routine; so the computation inside the Timath 'work' subroutine can be keyed to generic index variables such as I, J, K, and L. The ProMAX menu window for Timath contains five parameters, all defaulted to zero (in its default mode, Timath passes trace ensembles through with no alteration). The first Timath parameter is the integer 'function switch' that selects which internal algorithm is applied by Timath; parameters 2 and 3 are user-defined integer parameters; and 4 and 5 are user-defined real parameters.

In addition to the 'TRACES' array, which contains the samples of the input/output ensemble, Timath also makes available to the programmer the following real arrays, of the same dimension as TRACES: SCRATCH, VECT1, VECT2, R_TRIN, and R_TROUT. These arrays may be used as desired in computations, but final results, for output, must be placed back in TRACES. Additional information passed to the Timath 'work' subroutine is the trace header array OFFSET, which currently contains the source-receiver offset for each input trace (but could easily be altered to carry some other header instead); SAMPRAT, the sample rate in ms; NUMSMP, the number of samples per trace; and NT_READ, the number of traces in the current ensemble. The values of NUMSMP and NT_READ automatically set the dimensions of TRACES and the other internal real arrays, which are conceptually 2D arrays of dimension NUMSMP by NT_READ. Each array is actually stored as a 1D vector within Timath, however, with traces sequential within the vector.

In its current version, Timath contains two functions, a least-squares subtraction algorithm for trace pairs, and a single trace 'envelope' function, both of which are described below.

Least squares subtraction

This function is selected by setting the Timath function switch to '1'. It operates on consecutive pairs of traces within the input ensemble, traces 1 and 2, 3 and 4, and so forth. In each case the second of the two traces (usually a 'noise' estimate) is subtracted from the first after being multiplied by a 'gain' function; and the difference trace is placed into the first of the two traces. The gain function consists of the window-averaged product of the two traces, divided by the window-averaged square of the second (noise) trace. For detail on least-squares subtraction, see Henley (2009) (Appendix). The length in samples of the smoothing window for the gain function is specified by the first Timath menu integer parameter. Since the operation is applied to an entire ensemble at once, the gain function may be smoothed laterally before application, as well. The lateral smoothing window length in traces is specified by the second menu integer parameter. Defaulting either integer parameter to zero causes no smoothing to be applied to the gain function.

Since the least squares function requires trace pairs, the input ensemble can be constructed using the two ProMAX functions 'Disk data input' for the raw input data traces and 'Disk data insert', in the 'merge' mode, for the 'noise' estimate traces. This will properly alternate the raw traces with their corresponding noise traces, and will create the trace header 'ds_seqno', which can be used to select the traces from the

ensemble after the least squares operation. To do this selection, we use an 'IF', 'Disk data output', and 'ENDIF', where the trace header for trace selection in the 'IF' statement is 'ds_seqno', and the required value is '1' (difference traces are placed by Timath into the first trace of each pair in the input/output TRACES array; and for these, 'ds_seqno' equals '1').

Max value envelope function

This function is selected by setting the Timath function switch to '2'. It is a single trace operation which finds the sample in each trace with the largest magnitude, then multiplies each sample by a user-chosen power of the reciprocal of its distance from the position of the maximum magnitude sample. The exponent for the reciprocal is specified by real parameter number 1 in the Timath menu and may be positive or negative. This function acts as a 'picking' function, which localizes the energy of a cross-correlation around its maximum, for example.

Other functions

The above functions are just two examples of the possibilities, which are limited only by imagination. Any number of different data traces can be combined into one ensemble for complex computations, if desired, using 'Disk data input' and 'Disk data insert', as described above. Sponsors who may have ideas for trace ensemble-based operations of some kind are welcome to contact the author with their ideas.

FRMATH

Like Timath, Frmath is a ProMAX module whose purpose is to give access to the traces of an input ensemble, but in the complex frequency domain, rather than the time domain. Also, like Timath, Frmath supports Fortran algorithms using any of the ProMAX functions and subroutines. It handles the data I/O, stores trace headers for the input/output ensemble, and presents the ensemble samples in a 'TRACES' array to the 'work' subroutine of Frmath. Like Timath, Frmath has five ProMAX menu parameters; a function switch, two user-defined integer parameters, and two user-defined real parameters (like Timath, the default of all parameters to zero causes input ensembles to pass through Frmath unaltered).

Frmath differs from Timath in that it immediately moves the TRACES array to the SCRATCH array, whose primary dimension has been padded to the next higher power of two. SCRATCH is then used to populate the complex C_TRIN array; and the FFT then performs the complex 1D Fourier transform of each input trace. In addition to C_TRIN, the following complex arrays of the same dimension are defined: C_TROUT, VECT1, and VECT2. All four complex arrays can be used in any subsequent computations within Frmath, but the final results for output should be placed in C_TROUT, upon which the inverse FFT will operate. The real part of C_TROUT is moved to SCRATCH, from which the original TRACES array is reconstituted, with the frequency-domain processed ensemble data. As in Timath, although the arrays used by Frmath are conceptually 2D, they are all represented internally as 1D vectors of dimension NUMSMP2 by NT_READ, where NUMSMP2 is the next power of two larger than NUMSMP. Within these arrays,

the Fourier-transformed complex traces are stored sequentially, with real values followed by imaginary values within each trace.

Sqrt (i/omega) function

At present, only one function has been implemented in Frmath; a version of the $\pm\sqrt{i/\omega}$ filter. To apply this operator, the real and imaginary filter amplitudes are computed, then moved to a complex vector. This filter vector is then multiplied with each input complex spectrum to yield the output. The sqrt(i/omega) function is selected by setting the Frmath function switch to 1. The only input parameter is the sign of the operator, which is provided by setting the first integer parameter to either plus one or minus one.

Other functions

As with Timath, Frmath operations are only limited by imagination. Spectral operations between traces, such as spectral division, can be used to test deconvolution schemes, for example; and 'Disk data input' and 'Disk data insert' can be used to construct an ensemble with traces from any desired sources. As with Timath, sponsor ideas for ensemble-based frequency domain operations are welcomed by the author.

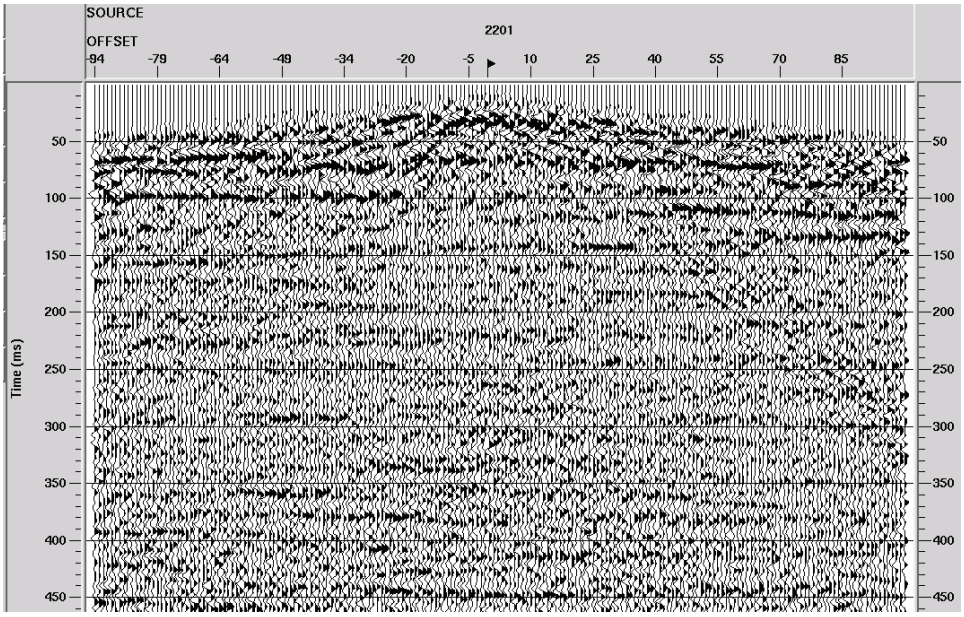
EXAMPLES

Timath examples

Least-squares subtraction

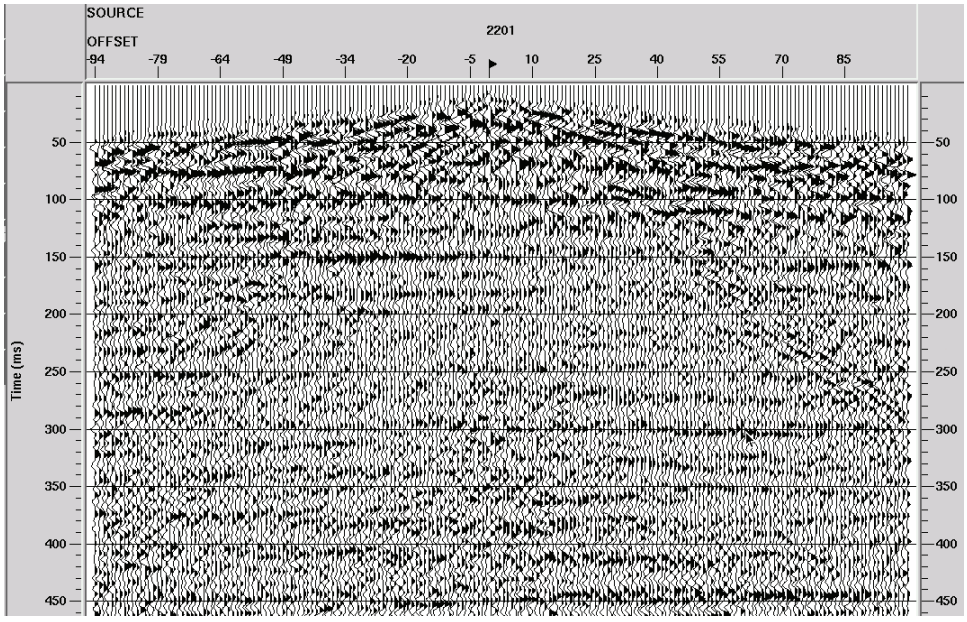
To demonstrate the least-squares subtraction function, we borrow data from the Priddis pulse/probe experiment (Margrave et al 2008), (Henley 2009). In this experiment, we compare data from the same geophone spread and source points for two different shots. During one of the shots, a monochromatic acoustic field is maintained in the earth by our mini-vibrator, operating at a constant frequency; while during the other shot, the vibrator is silent. The two shots are identical dynamite charges in a reloadable, cased borehole. By comparing the two shots using subtraction (and here, least-squares subtraction as well), we hope to detect any differences in earth response caused by the acoustic field.

Figure 1 shows a gather for one of the 'dynamite-only' shots, while Figure 2 shows the gather from another shot at the same shot point, but with the vibrator maintaining a 20 Hz acoustic field in the earth. Straight subtraction yields the difference shot in Figure 3, while least-squares subtraction, as implemented in Timath yields Figure 4. Without attempting an interpretation of these data, we can see that the least-squares subtraction results differ from the straight subtraction, especially in the shallow part of the record, which is where we expect to see differences, if our experiment is successful. In Figure 4, several shallow events are somewhat more continuous and higher in amplitude than in Figure 3. Figure 5 shows a side-by-side comparison, on which it is easier to see the differences.



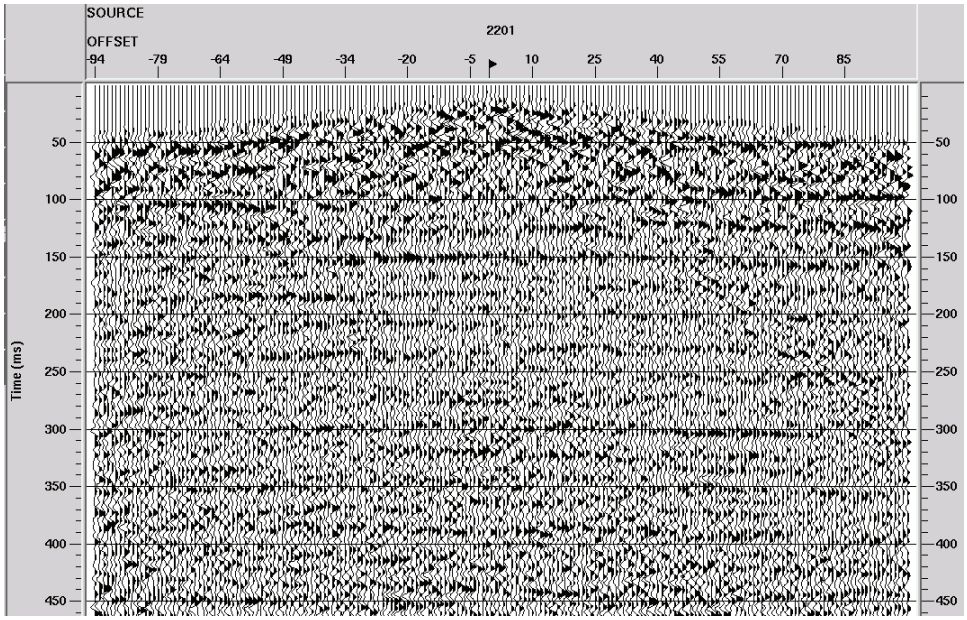
Priddis pump/probe experiment—shot 2201—dynamite only

FIG. 1. Typical shot gather for 'dynamite only' survey at Priddis



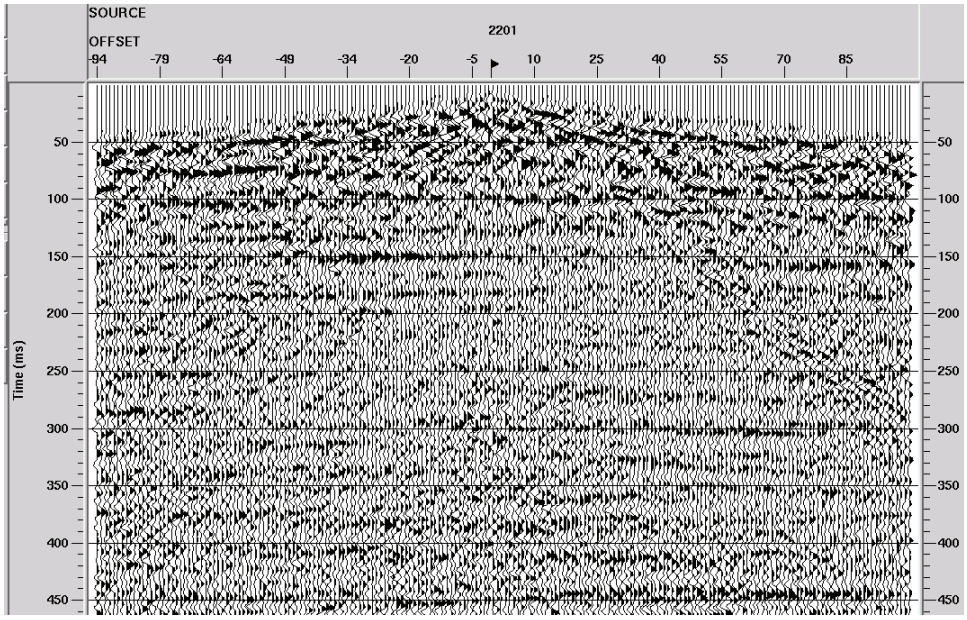
Priddis pump/probe experiment—shot 2201—dynamite plus 20 Hz

FIG. 2. Shot gather for 'dynamite plus 20 Hz' corresponding to shot in Figure 1



'Dynamite plus 20 Hz' minus 'Dynamite only'—straight subtraction

FIG. 3. Straight arithmetic difference between gathers in Figure 1 and Figure 2.



'Dynamite plus 20 Hz' minus 'Dynamite only'—least-squares subtraction

FIG. 4. Least-squared difference between gathers in Figure 1 and Figure 2

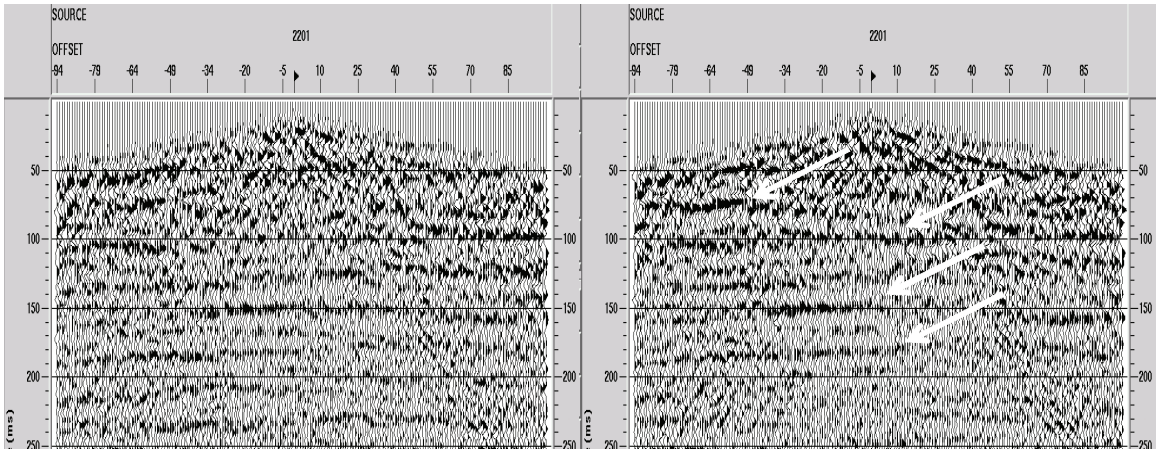


FIG. 5. Comparison between regular arithmetic subtraction (left) and least-squares subtraction (right) for the Priddis pulse/probe experiment. “Dynamite only” shot subtracted from “dynamite plus 20 Hz” shot.

Envelope/picking function

We use an ensemble of cross-correlation functions to illustrate the actions of the envelope/picking function in Timath. Figure 6 shows a set of correlation functions as computed in the ProMAX cross-correlation module, while Figure 7 shows the same data after application of the envelope/picking function, with a relatively mild exponent of 0.25. The action of the function can be readily seen. Side lobes of the correlation are greatly suppressed, in favour of the main lobe. Increasing the exponent to 2.0 yields the results in Figure 8, in which the correlations have assumed the appearance of spikes, due to the great suppression of any energy away from the absolute maximum value of the function.

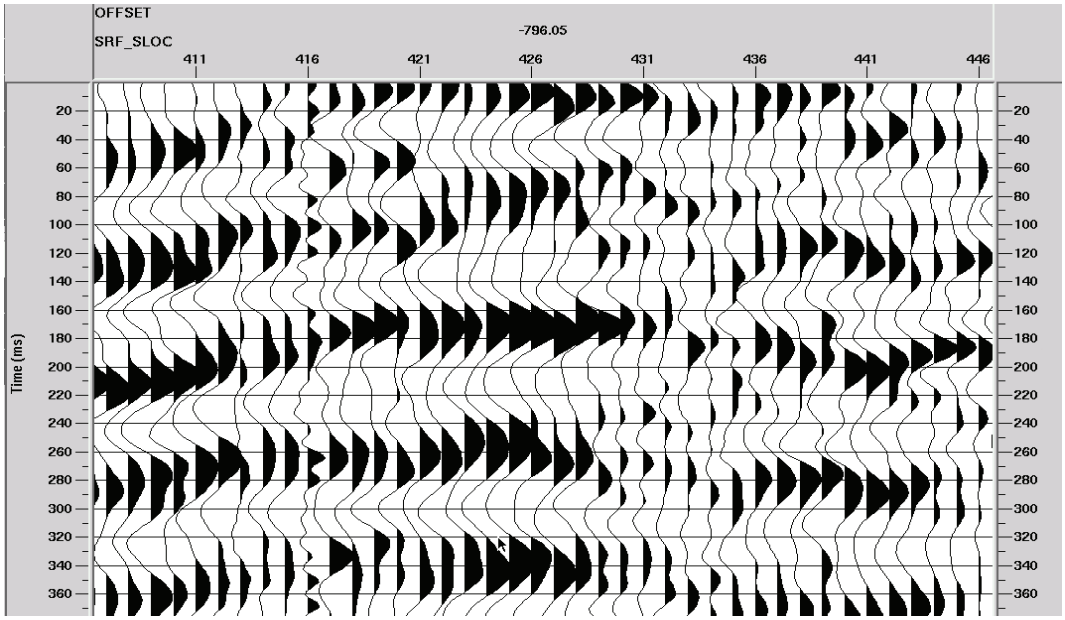


FIG. 6. Example of cross-correlation functions used to derive statics functions in the 'statics deconvolution' technique.

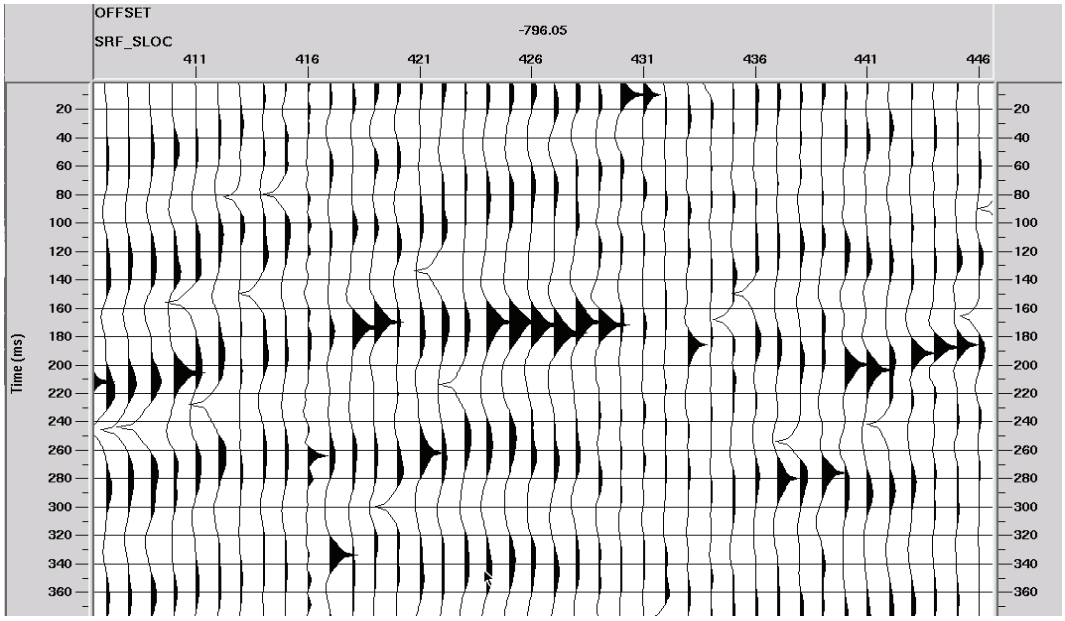


FIG. 7. Cross-correlations of Figure 5 after application of the Timath 'envelope/picking' function with an exponent of 0.25

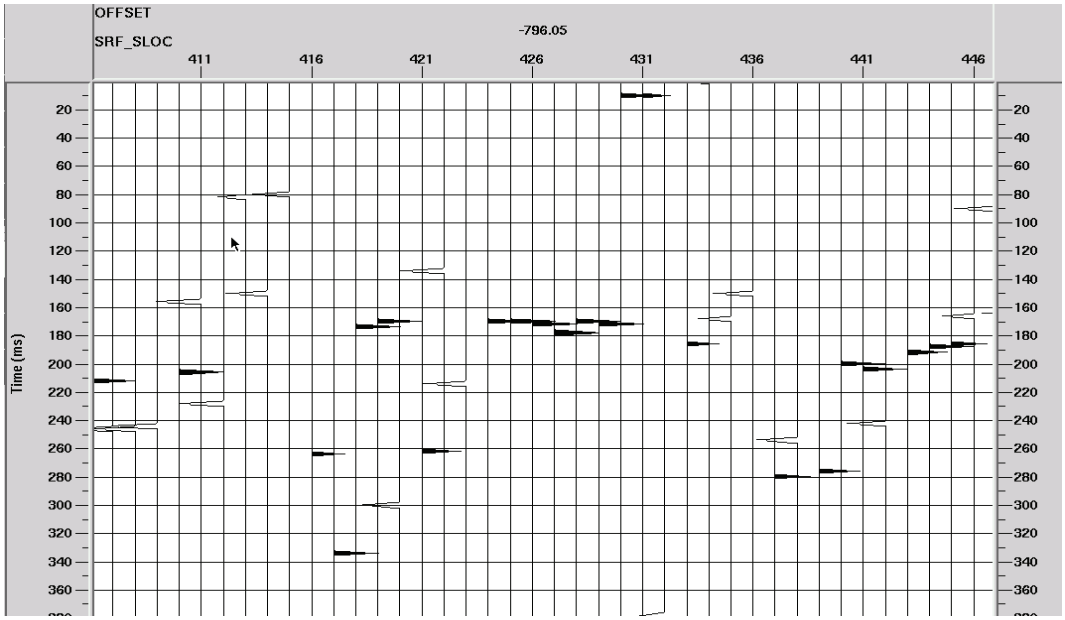


FIG. 8. Cross-correlations of Figure 5 after application of the Timath 'envelope/picking' function with an exponent of 2.0

Frmath example

Since there is only one function currently included in Frmath, the Sqrt(i/omega) filter, we illustrate it with two figures. Figure 9 shows a trace ensemble with no filter applied, while Figure 10 shows the same ensemble after application of the filter with positive polarity.

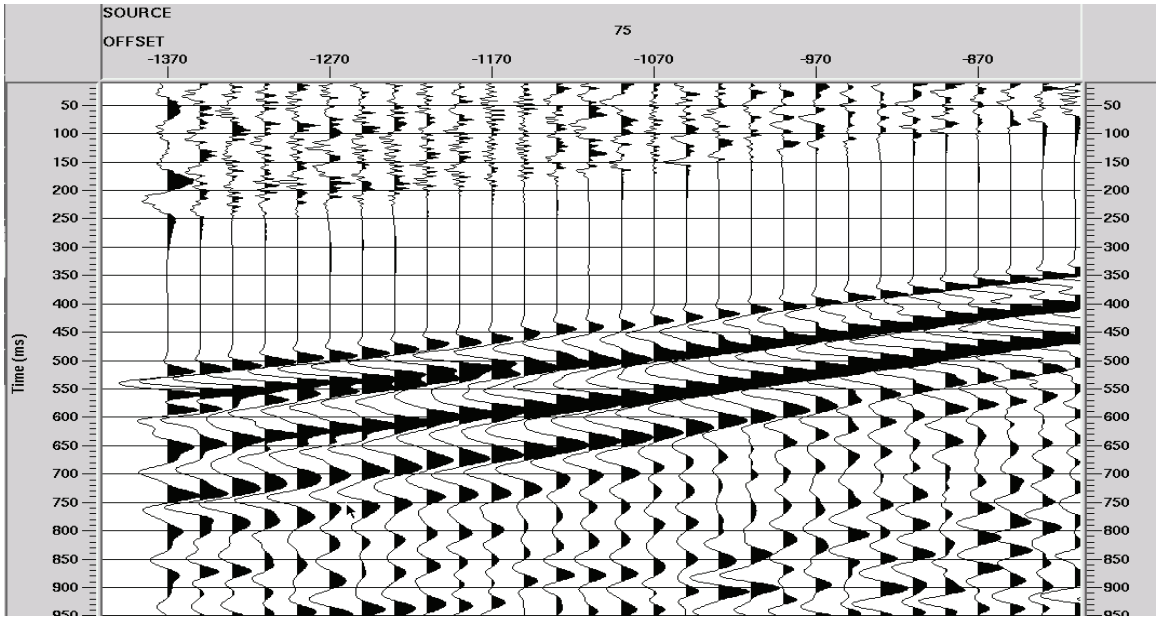


FIG. 9. A small ensemble of seismic traces

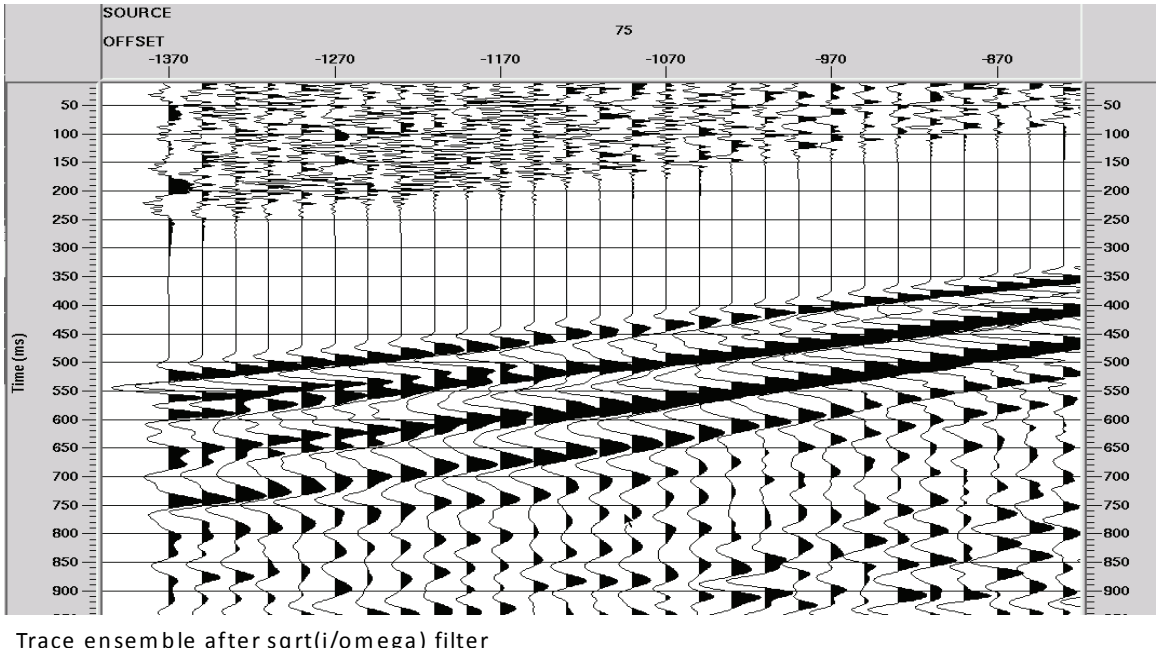


FIG. 10. Small ensemble of seismic traces after application of sqrt(i/omega) filter in frmath.

DISCUSSION

The new ProMAX functions Timath and Frmath are described here, not because they constitute new research, but because they will provide considerable time savings for testing new 2D ensemble-oriented functions. We intend them primarily for our internal use, but will continue to re-release them as we include new functions. As suggested above, we welcome any ideas for new functions to be tested.

ACKNOWLEDGEMENTS

The author acknowledges the sponsors of CREWES for continuing support.

REFERENCES

Henley, D.C., 2009, Priddis pulse/probe experiment:still ambiguous: CREWES 2009 Research Report, **21**.
Henley, D.C., 2009, Radial filtering on steroids: the latest algorithm: CREWES 2009 Research Report, **21.**
Margrave, Gary F., Bertram, M., W. Hall, K.W., Han-Xing Lu, Bonham, K., Wong, J., Gallant, E., and Henley, D. C., 2008, Priddis pulse probe experiment: CREWES 2008 Research Report, **20**.

APPENDIX

Below is the documentation for Timath and Frmath, respectively.

Timath

```
Time domain math

This module is intended to provide a platform for computations in
the time domain. It reads in panels of traces, then hands each trace
to a subroutine in which operations can be performed on any trace,
or pair of traces, optionally using any of the user-defined
functions and parameters read in from the menu.


Theory

While some operations in ProMAX lend themselves to time domain trace
computation, these are limited in scope. The intent of this module
is to provide an easily-modified platform for testing various time
domain computations.


Usage

TIMATH may be applied to any ensemble of seismic traces, including
radial trace gathers.  The required parameters are as follows:

function switch

This parameter is used to select which function to apply to the time
domain samples of the input traces. New functions will be added as
they are conceived and tested. If this parameter is defaulted to
zero, the data are passed through the algorithm unchanged. The
current functions are:

      1 = perform least-squares subtraction of trace pairs in the
          input panel. The convention is that the odd-numbered
          traces in the input panel are the "raw" traces, and the
          even-numbered traces are the "noise" traces. For this
          operation, the first integer parameter is the smoothing
          length for a boxcar filter applied to the least-squares
          gain function in time, while the second integer parameter
          is the smoothing length for a boxcar filter applied to the
```

```
          least-squares gain function laterally. Defaulting either
          parameter causes no smoothing to be applied.

      2 = find the absolute maximum of each trace; compute an
          envelope function defined as a real power of the
          reciprocal difference between each sample position and
          the position of the detected maximum; multiply the trace
          by its envelope function. The first real parameter in the
          menu is the exponent to which the reciprocal difference is
          raised.

          This function serves to localize a trace around its
          Largest sample value--a kind of "picking" function.

      3 = undefined

      4 = undefined
          ...................

parameter 1

This is an integer parameter, default 0, user-defined

parameter 2

This is an integer parameter, default 0, user-defined

parameter 3

This is a floating point parameter, default 0.0, user-defined

parameter 4

This is a floating point parameter, default 0.0, user-defined

General

This is an experimental module, subject to modification
and change as testing proceeds.


References

D.C. Henley dhenley@ucalgary.ca
```

Frmath

```
Frequency domain math

This module is intended to provide a platform for computations in
the frequency domain. It reads in panels of traces, then hands each
trace to a subroutine which transforms the trace to the frequency
domain.
```

Inside this subroutine, any frequency domain math can then be performed, optionally using any of the user-defined functions and parameters read in from the menu.

Theory

While some operations in ProMAX lend themselves to frequency domain computation, these are limited in scope. The intent of this module is to provide an easily-modified platform for testing various frequency domain computations.

Usage

FRMATH may be applied to any ensemble of seismic traces, including radial trace gathers. The required parameters are as follows:

function

This parameter selects which test function in this routine to apply to the input data traces. The default for this parameter is zero, which causes the routine to pass the unaltered data. The list is as follows:

1 = multiply the trace in the complex frequency domain by plus or minus the square root of i/omega (omega = angular frequency) For this function, parameter 1 is either plus one or minus one to determine the sign of the operator.

2 = undefined
3 = undefined
.....

parameter 1

This is an integer parameter, default 0, user-defined

parameter 2

This is an integer parameter, default 0, user-defined

parameter 3

This is a floating point parameter, default 0.0, user-defined

parameter 4

This is a floating point parameter, default 0.0, user-defined

General

This is an experimental module, subject to modification and change as testing proceeds.

References

D.C. Henley dhenley@ucalgary.ca