

Recent ProMAX module upgrades

David C. Henley and Kevin W. Hall

ABSTRACT

We describe here recent modifications and upgrades to several existing ProMAX modules, which are now available to sponsors through our website. Among the more important changes, which were made to all existing modules, were some clean-up of the Fortran, and substitution of newer memory allocation subroutines to enable the modules to compile and run successfully on 64 bit Linux systems. Specific modules which were given new capabilities are Gabor2 and Gabor_sc (both have been enabled to do colour correction), and Timath (a new function has been added to enable computation of Discrete Fourier Transforms).

INTRODUCTION

One of the ongoing mandates of CREWES is to introduce new processing algorithms corresponding to techniques developed by researchers in the course of their research for analyzing and interpreting seismic data. Most of these algorithms were originally developed in the MATLAB toolbox, in which form they can be used for limited testing on regular seismic data sets. To facilitate testing and use in routine processing by sponsor companies, however, we further implement some of the more promising or useful algorithms as ProMAX modules, since many of our sponsors either use ProMAX in their processing, have at least one ProMAX license on their premises, or have the capability to quickly modify a ProMAX module for use in their own processing system.

Each year prior to the Sponsors' Meeting, we update our ProMAX offerings by adding new modules and incorporating bug fixes, extensions and updates to existing modules. In 2009, we offered two new toolbox modules, TIMATH, and FRMATH (Henley, 2009a), updated GABOR_SC with a diagnostic mode (Henley, 2009b), and added a least squares subtraction mode to RADFILT (Henley 2009c). In addition, we re-compiled all ProMAX modules for compatibility with ProMAX version 2003.19 and released them through the website. In 2010, we offer no new modules, but have made significant extensions to GABOR_SC and GABOR2 and have added a new function to TIMATH; all of which we describe below. As well, we went back into the Fortran code for all modules, fixed some awkwardness in style, and made some necessary changes to allow the modules to compile and run on 64 bit Linux systems; most significantly by replacing the 32 bit memory allocation subroutines with their generic equivalents, which accommodate either 32 or 64 bit systems.

SPECIFIC MODULE UPDATES

Gabor2

The Gabor2 ProMAX module is our workhorse Gabor deconvolution module. It was first implemented for ProMAX in 2001 (Henley and Margrave, 2001) with Gaussian window functions (module name—Gabor), then in 2002 in its current form with Lamoureux window functions and other features to make it faster and more robust in a variety of processing scenarios (module name—Gabor2), (Margrave, et al, 2002). Gabor2

can be used to deconvolve traces one at a time, or in ensemble mode to deconvolve all the traces in an ensemble with an operator derived from the average Gabor spectrum of the ensemble.

The introduction of surface-consistency by Montana et al (2006) into Gabor deconvolution led to the development of an expanded version of Gabor2 (module name—Gabor_sc), specifically intended to apply the Gabor algorithm in a surface-consistent manner; but the first significant modification of the basic Gabor algorithm was introduced by Cheng and Margrave (2008, 2009), and involves modifying the “white” assumption used to approximate the earth’s reflectivity spectrum. In their work, Cheng and Margrave describe analyzing a reflectivity log from a well near a particular seismic line (and hence having the same gross spectral content as the actual earth function sampled by the seismic line). If Fourier magnitude spectra are computed for a series of time-shifted, overlapped windows applied to the reflectivity log, a 2D time/frequency spectrum can be obtained for the reflectivity log. Since the term “white” describes only the overall slope of the amplitude spectrum for any particular time window, the detailed spectra in each window can be least squares fitted to a low order polynomial curve, and the curve used to describe the “colour” or trend of the spectrum. The spectral shapes can thus be captured by only a few coefficients per spectrum rather than many amplitude values. When properly interpolated and extrapolated to the time/frequency grid used to apply Gabor deconvolution to a seismic trace, the 2D polynomial colour function derived from the well log spectra can be substituted for the “white” assumption used in the Gabor deconvolution spectral division. In effect, instead of dividing each element of the 2D Gabor deconvolution operator by 1 before applying it, we divide the operator, point by point, by the 2D colour function, thus steering the resulting deconvolution more nearly toward the true earth reflectivity function.

The specific extensions made to Gabor2 to accommodate “colour correction” are as follows:

- 1) When instructed to do so by a menu parameter, Gabor2 now has the ability to read an external ASCII file, written in a particular format, to obtain the polynomial coefficients needed to reconstruct the polynomial spectra derived from the spectral analysis, in MATLAB, of a reflectivity log.
- 2) Gabor2 reads the external file and passes the polynomial coefficients for each analysis window to a new subroutine, COLOURCOR, which interpolates and extrapolates the polynomial coefficients from the centre times of the analysis windows to the centre times of the Gabor decon windows, evaluates the polynomials for each time/frequency grid point, normalizes the polynomial spectra in the time dimension, if requested, and returns the 2D gridded colour correction function.
- 3) Prior to applying the standard minimum phase computation to the Gabor deconvolution operator, the 2D colour correction function is divided into the Gabor magnitude spectrum of the operator.

The use of the updated Gabor2 is described in more detail in an accompanying report chapter (Henley et al, 2010). We show here three figures to illustrate the effect of colour correction as applied by Gabor2. Figure 1 is the colour correction function derived from an actual reflectivity log, Figure 2 shows a group of seismic traces deconvolved by Gabor2 using the normal “white” earth reflectivity assumption, and Figure 3 shows the same traces as Figure 2, but with colour correction applied during deconvolution, using the colour function in Figure 1. Differences in both frequency content and phase can be seen, especially in the vicinity of reflection events in the 800-1000 ms zone, where amplitudes are larger, high frequency content is increased, and different phase makes events slightly shallower in time.

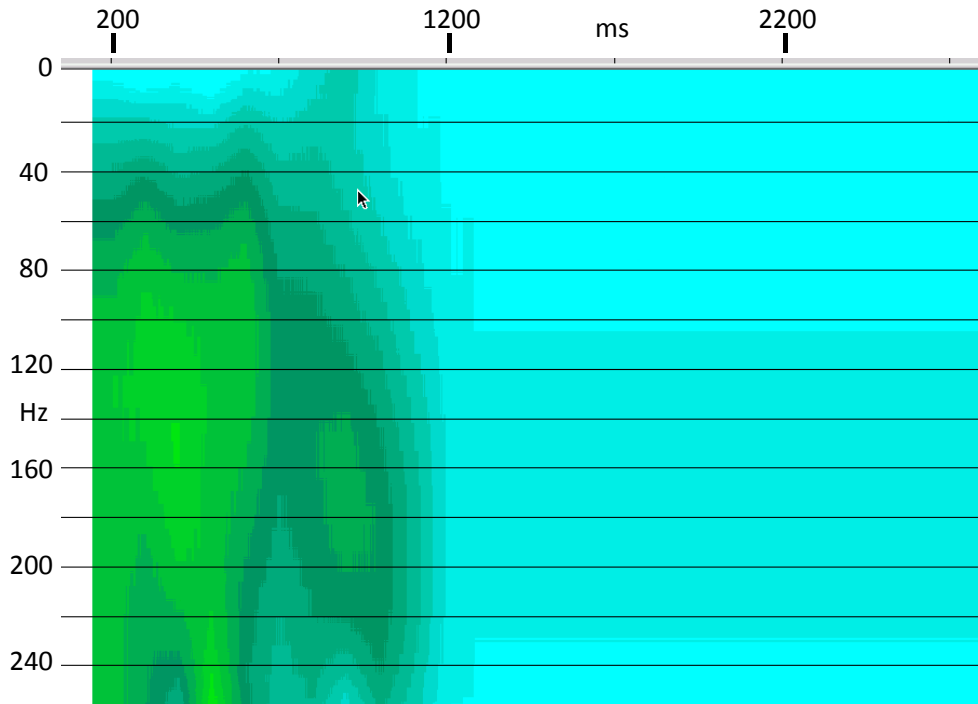


FIG. 1. 2D colour correction function derived from reflectivity spectra of a well log. Values deeper than 1200 ms are extrapolated.

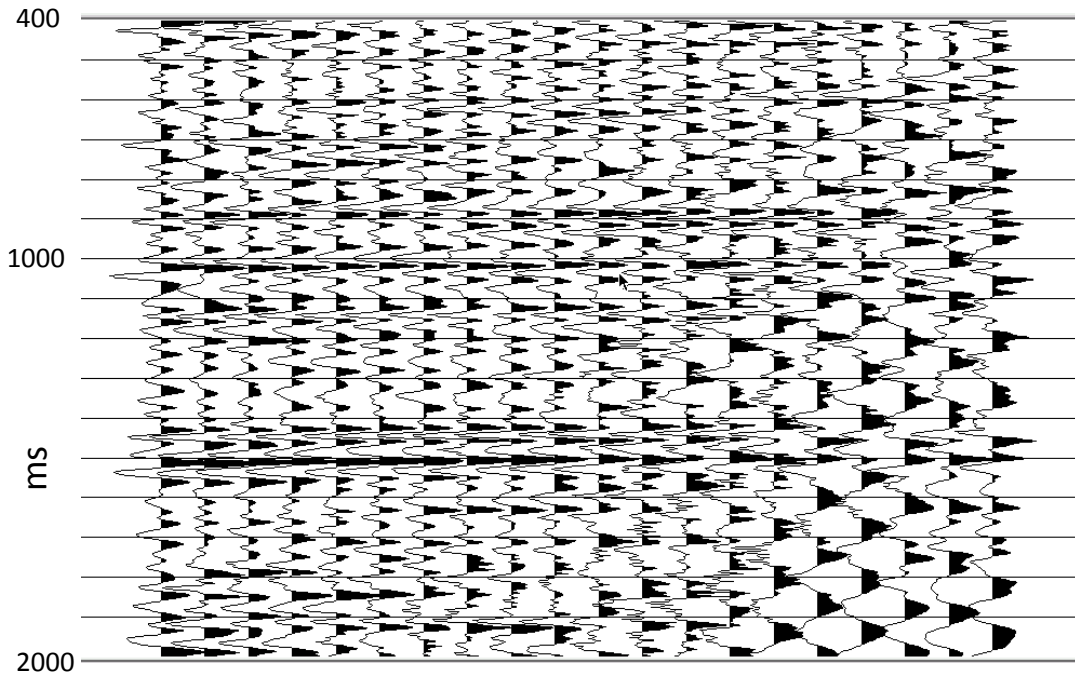


FIG. 2. Seismic traces deconvolved with Gabor2, using no colour correction.

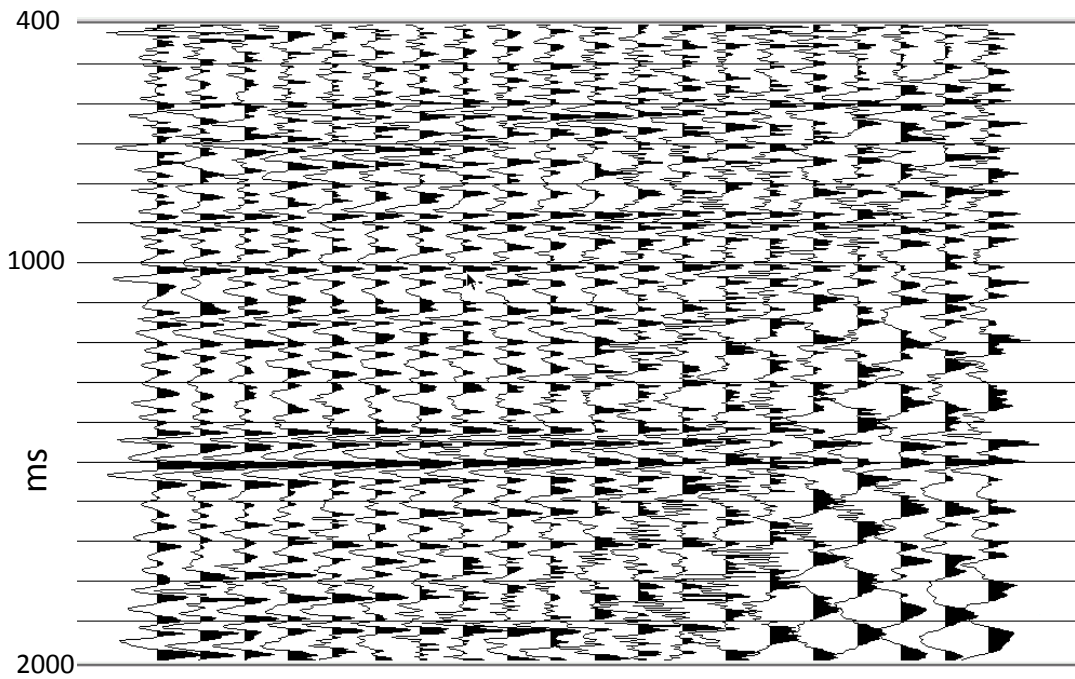


FIG. 3. Seismic traces from Figure 2 deconvolved with Gabor2 using colour correction function from Figure 1.

Gabor_sc

While the basic principles behind Gabor_sc are much the same as those behind Gabor2, the former differs in its factorization of the Gabor spectra of input traces, and incorporates structures which allow not only for averaging spectral factors over source and receiver surface locations, but over subsurface location (midpoint) and absolute offset. The averaged spectral factors are then used in a second pass of Gabor_sc to construct and apply Gabor deconvolution operators specific to each input trace. If desired, new estimates of spectral factors can be made during the second pass, and the updated factors used to deconvolve the traces during a third pass, and so forth. As can be imagined, the memory allocation required to accommodate this averaging and updating of Gabor spectral operators can be quite large. The module was introduced in 2007 (Henley et al, 2007) as the first ProMAX implementation of the work by Montana et al (2006).

Another feature unique to Gabor_sc is a diagnostic mode, in which each trace selected by the input parameters is analyzed with the Gabor transform (Henley, 2009b). The Gabor spectrum of the input trace can be subsequently output as a set of seismic traces, one trace per Gabor window, whose sample values are Gabor magnitudes for the particular window. These traces, each annotated with a new trace header carrying the time of the Gabor window centre for that window, can be plotted in gray-shade or colour contour mode to provide a 2D plot of the Gabor spectrum. Not only can the Gabor spectrum of each input trace be plotted, but its subsequent factors, as well, including the Q-function, wavelet spectrum, and residual spectrum (deconvolved trace spectrum).

Gabor_sc was updated to accommodate colour correction in the following ways:

- 1) Triggered by a menu parameter, Gabor_sc, like Gabor2, has the ability to read an ASCII file, in a particular format, to obtain the coefficients for the polynomial spectra of each of the windows used to analyze the input well log.
- 2) The polynomial coefficients are passed to the new subroutine, COLOURCOR, which interpolates and extrapolates the coefficients to the time grid associated with the Gabor decon parameters, evaluates the polynomials at each point of the time/frequency grid, and optionally normalizes the spectra in time.
- 3) If Gabor_sc is run in deconvolution mode, the colour correction function is used to modify the Gabor magnitudes of each deconvolution operator after it is assembled from the four pertinent components (one each for source location, receiver location, midpoint, and absolute offset) stored in the Gabor magnitude arrays accumulated by the first pass. This colour application is just prior to the minimum phase computation, so that the phase of the operator will accurately reflect that of the well log earth function.
- 4) If Gabor_sc is run in diagnostic mode, then the colour correction function itself can be output as a set of seismic traces, with time window centres in the trace headers, intended to be plotted in Gray-shade or colour contour. Figure

1 is an example of such a plot. Since the colour function is interpolated/extrapolated to the same time/frequency grid as the Q-function and the other Gabor spectra available as output from Gabor_sc, its plots are easily compared with plots of these other spectra.

For more detail, including documentation, on the new version of Gabor_sc, see the accompanying report chapter (Henley et al, 2010).

Timath

As described by Henley (2009a), the Timath module was created to allow rapid implementation of relatively simple algorithms to be applied to one or more traces in an ensemble. The module is written in such a way that the infrastructure (ensemble reading and writing, parameter input, and the like) need not be changed when a new algorithm is implemented. All one needs to do is to choose a new value for the selection switch variable, to be tested for selecting the new function, then to write the function itself, inside a logical test loop for this switch variable value, as a few lines of Fortran utilizing the “standard” array names for the input traces, output traces, any intermediate scratch arrays, and any input parameters. Since a complete ensemble of traces is presented each time, the Fortran operation can operate on any single trace, or any combination of traces within the ensemble, as desired. As an example, one of the more useful functions placed in the initial version of Timath is the least squares subtraction algorithm, which computes 2D gain functions which minimize the least squares difference between two groups of input traces.

Early in 2010, we received a request from a sponsor to implement a Discrete Fourier Transform for use as a high-resolution diagnostic on seismic traces with many samples and relatively discrete spectral peaks. Since the Discrete Fourier Transform is based on the exact calculation of the Fourier Transform from its sine/cosine definition over an arbitrary number of time samples (not constrained to a power of two), its computation consumes computer resources; so the DFT is not usually implemented in a processing package, including ProMAX. Once we began, it took less than an hour to introduce and test a simple DFT function within Timath. Figure 4 shows an example of the kind of detail seen in a spectrum obtained from the DFT on a very long seismic trace. As an example of the kind of phenomena that can be explored with the DFT as a diagnostic, Figure 5 shows the DFT of a very long 10 Hz ‘mono-frequency’ trace generated by a vibroseis unit. Its fundamental frequency and harmonics are readily apparent. Note that the analysis window is an integral number of seconds. When the window length is changed by only a single sample, the harmonic structure and Gibbs’ phenomena change dramatically, as can be seen in Figure 6. An example like this should raise questions about the faith we place in the details of spectral analysis, since results can be so profoundly influenced by small variations in the analysis parameters.

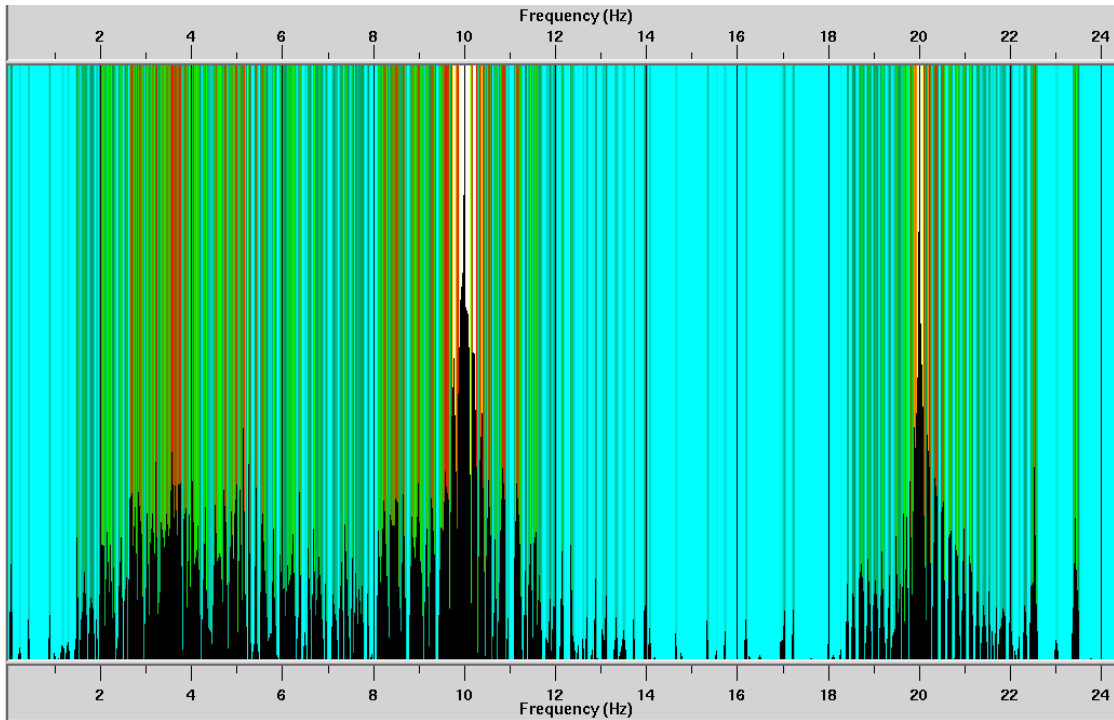


FIG. 4. Example of a DFT magnitude spectrum from a very long 10 Hz mono-frequency Vibroseis sweep.

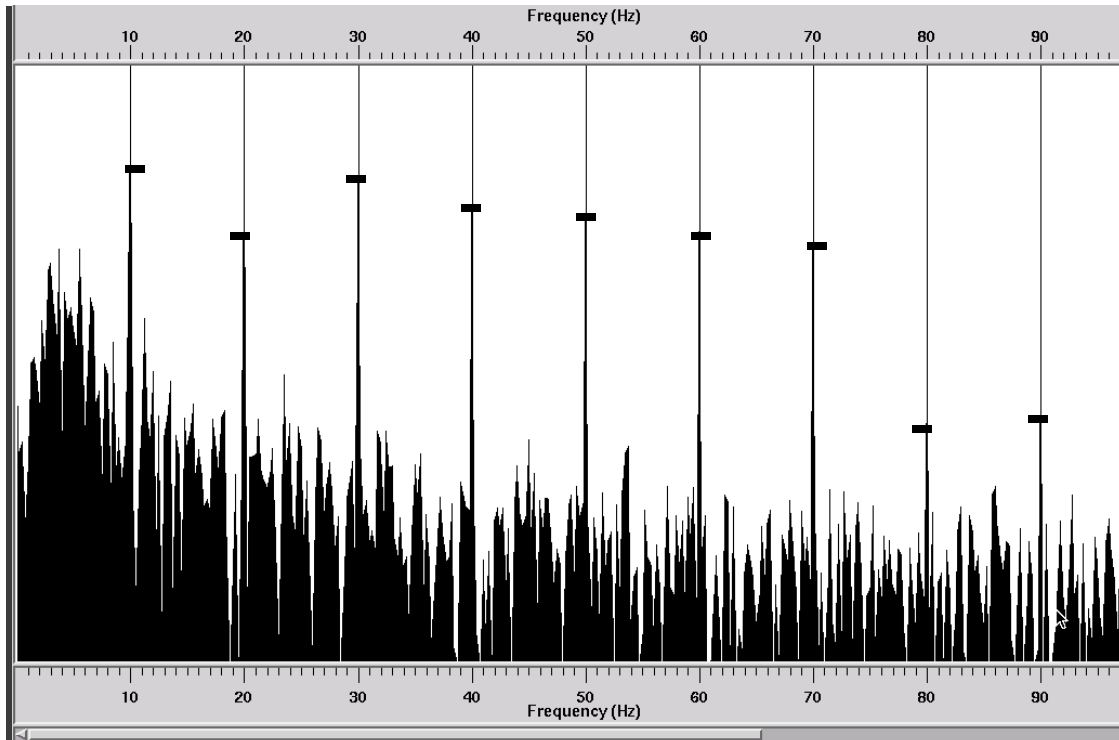


FIG. 5. DFT magnitude spectrum of very long 10 Hz mono-frequency Vibroseis sweep. Start time for DFT window 10 seconds, time gate 4000 ms

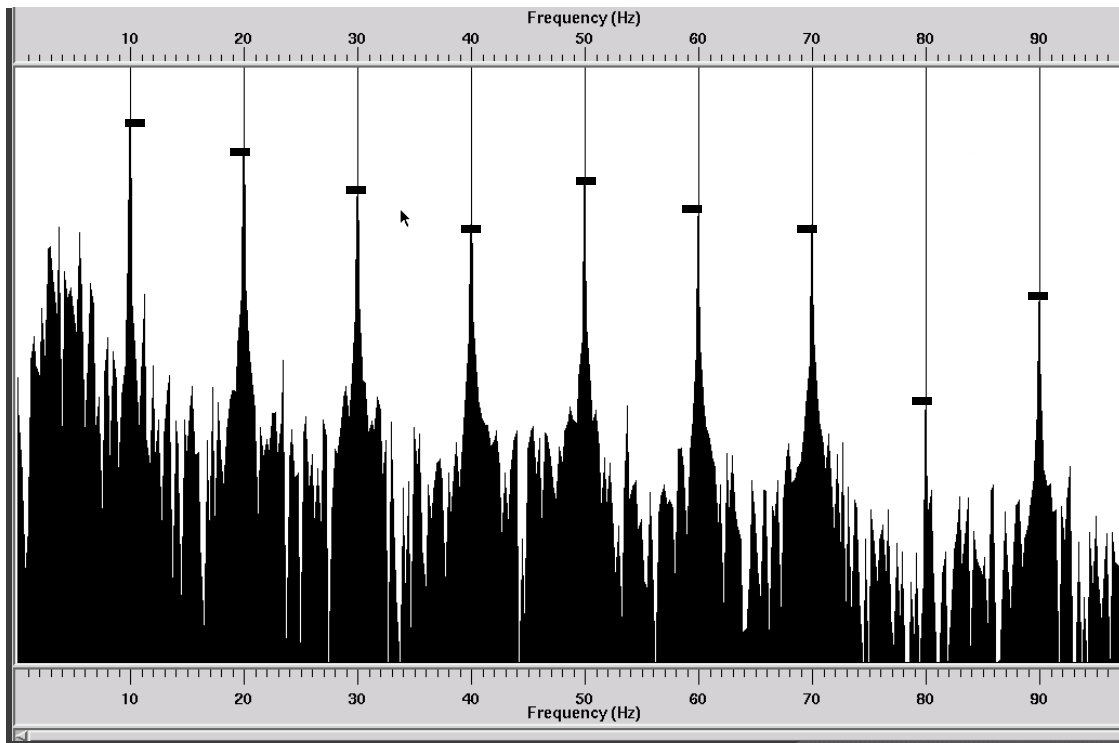


FIG.6. DFT magnitude spectrum of very long 10 Hz mono-frequency Vibroseis sweep. Start time for DFT window 10 seconds, time gate 4002 ms. One sample change in the length of the window makes a very large difference in the details of the spectrum!

INTO THE 64 BIT WORLD

When CREWES first began to acquire 64 bit Linux computers for seismic processing, we noted that existing ProMAX modules written by us would compile but not execute on these new machines, yielding error messages that strongly suggested that memory allocation was not being properly done. Since CREWES is gradually migrating to 64 bit machines, as are some of our sponsors, we decided to invest the necessary effort to make our software usable on the newer machines.

Thanks to a lot of work on the part of Kevin Hall, we discovered the main issues preventing our current ProMAX modules from executing properly—not surprisingly, mostly to do with memory allocation and storage of contiguous variables:

- 1) The biggest and most obvious problem was the use in our modules of an obsolete subroutine for memory allocation, designed to address memory space with only 32 bits. This subroutine, `mem_resbuff`, was replaced with its generic equivalent, `memory_resbuff`, which determines from context whether to reserve memory in 32 or 64 bit address space.
- 2) The second major problem was in specifying the index variables used to partition and access the main memory buffer, `RSPACEz`. For earlier versions of ProMAX, these variables could be defined in the COMMON block in the INCLUDE file for each module, and they were always designated as

INTEGER. In current use, these must be separate from the COMMON block and must be specified with their own variable type designation, PTRDIFF_T.

- 3) Some small problems with contiguous variable type designations were located and corrected to clean up the Fortran code.
- 4) None of our modules freed memory when no longer needed, so this oversight was corrected for all modules, as well.

All of the ProMAX modules currently available on the CREWES website have had the described modifications applied; and all have subsequently been compiled on both 32 and 64 bit machines and tested briefly to ensure that they do, in fact, execute and return reasonable results.

SUMMARY

We have described the extensions made to three existing ProMAX modules, and have described the updates made to CREWES ProMAX modules in order to compile and run them on 64 bit Linux systems. As of Sponsors' Meeting time, all these changes will be reflected in the software release available to sponsors on the CREWES website, and on the USB memory stick received by attendees.

ACKNOWLEDGEMENTS

Thanks to CREWES sponsors for continuing support.

REFERENCES

- Cheng, P., and Margrave, G.F., 2008, Color correction for Gabor deconvolution, 2008 CREWES research report, **20**.
- Cheng, P., and Margrave, G.F., 2009, The influence of reflectivity color on Gabor deconvolution, 2009 CREWES research report, **21**.
- Henley, D.C., 2009, Timath and frmath: experimental trace ensemble modules for ProMAX, 2009 CREWES research report, **21**.
- Henley, D.C. 2009, Intelligent design: the diagnostic mode of Gabor deconvolution, 2009 CREWES research report, **21**.
- Henley, D.C., 2009, Radial filtering on steroids: the latest algorithm, 2009 CREWES research report, **21**.
- Henley, D.C., and Margrave, G.F., 2001, A ProMAX implementation of nonstationary deconvolution, 2001 CREWES research report, **13**.
- Henley, D.C., Cheng, P., and Margrave, G.F., 2010, Earthtones: nonstationary colour correction in ProMAX, 2010 CREWES research report, **22**.
- Henley, D.C., Margrave, G.F., and Montana, Carlos, 2007, Gabor deconvolution: surface and subsurface consistent, 2007 CREWES research report, **19**.
- Margrave, G.F., Henley, D.C., Lamoureux, M.P., Iliescu, V., and Grossman, J.P., An update on Gabor deconvolution, 2002 CREWES research report, **14**.
- Montana, C.A., Margrave, G.F., and Henley, D.C., 2006, Surface-consistent Gabor deconvolution, 2006 CREWES research report, **18**.