Book I

Using the Digitized Data

By
Robert W. Lankston
April 29, 2023

I began the work described in the Log segment of this narrative on January 11, 2008. The Log describes steps to convert the 8000 samples per second data (0.125 ms sampling) in the Hess .wav files into discrete seismic traces in SU format. Initially, when this book was posted in 2012, this book contained a single, rather large SU file. Now, this book contains the SEG-Y format files for the respective six lines.

I selected the data with the 0.125 ms sampling for this process anticipating that I could more precisely define time zero on each trace.

The conversion process as described in the Log should be considered a proof of concept. It uses a synthesis of tools and reflects my programming skills of that time. Today, I might consider building the entire conversion process in Python. Python provides modules for reading .wav files, can handle the tasks that awk and C were used for, and provides for system calls to Seismic Unix (SU) modules.

The C codes referenced in the Log have filenames that relate to Track 4 of the United States Geological Survey (USGS) archive tape. The same code, with changes to filenames and threshold settings, was used to process the Track 2 data. This book contains the data from Track 2 only. One SU format file from Track 2, approximately 5000 traces, was the initial content of this book.

The Epilogue section gives the highlights of modifications to the results of the 2008 conversion effort. As a result of the 2012-13 work described in the Epilogue, this book now contains an SEG-Y format trace file for each of the six survey lines that were preserved on the USGS archive tape described in Book H and contained in the file of Hess' 8000 Hz sampling of the Track 2 data, i.e., the single SU file described in the Log was removed from this book.

In Step 4 below, a reference is made to reading ahead 3 s on the input data file. This was done because the shots were nominally 4 s apart. The shot interval in time can be seen by displaying a down-sampled .wav file from Book H with an audio recording/editing application such as Audacity® (Figure 1). Also, Otis et al. (1977) indicate a 4 s shot interval when they used the Wold seismic system.

I have made some changes to the original Log and Epilogue texts. Specifically, literature references and hyperlinks were not in the original texts, and some have now been added. I have added footnotes to help tie the Log text to the C codes better than when the C codes and the Log were first loaded on the digital archive. I changed some filenames in the Log or Epilogue texts to match files in this book or in Book H of this collection. I added Figure 2 in the Step 8 section of the Log to illustrate the value of a static correction step after converting the traces from .wav format.

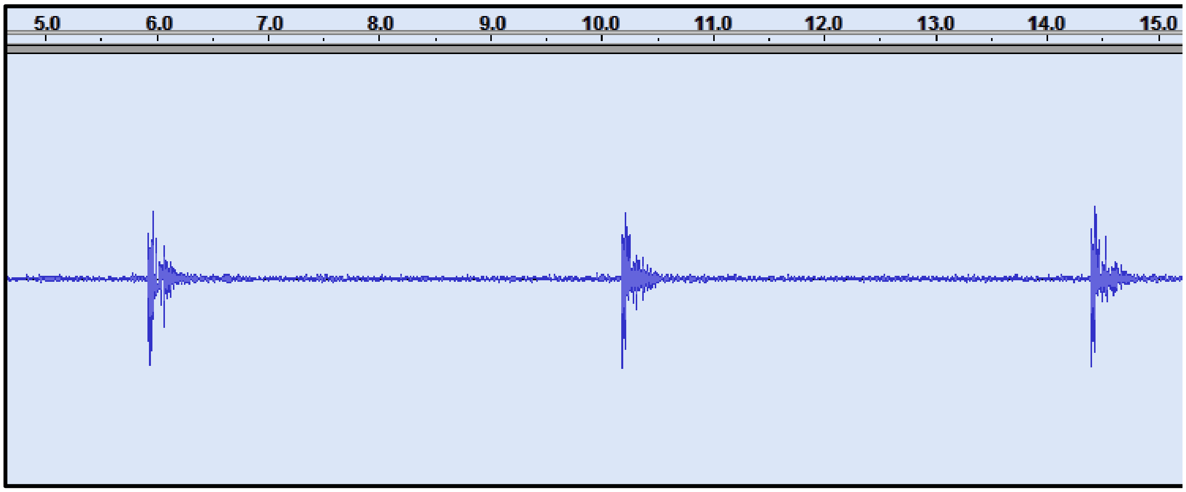

Figure 1. Waveforms for three shots. Each burst of energy is the energy received at the hydrophone for one shot. The scale at the top of the image is in seconds. The shots are approximately 4 s apart, but the intervals vary. This variability, of course, introduces uncertainty in any positioning scheme that assumes a constant, nominally 10 m spacing between traces. Any reflected seismic energy is obscured by noise after approximately 1 s.

Log of Steps to Read and Reformat the Track 2, demodulated, 8000 Hz .wav File

By
Robert W. Lankston
1-11-2008

The primary purpose of this Log is to record the steps that were followed in generating an SU format file of seismic traces. The Log references specific files that were developed in the course of generating the SU format file from the original Track 2, demodulated, 8000 Hz .wav file. However, not all of the referenced files are contained in this collection relating to the 1970 Wold-Crosby seismic survey of Flathead Lake. The processes described in this Log were subsequently applied to the Track 4, 8000 Hz file, but the SU file resulting from that conversion is not in this collection.

###
###

The conversion from .wav to SU has eight conceptual steps. Some of the steps might be combined into one script or program.

1. The original data file[1] was copied from the Richard L. Hess deliverable to the Linux workspace. The file was put in a separate folder to eliminate confusion with previous Track 2 work that was done on the 1000 Hz file in early 2007. Blanks that were in the Hess filename were replaced with _ as a matter of file naming style.

2. The open-source application *sox* (or SoX) was used to change the .wav format to a flat ASCII format. The following *sox* command was used.

```
sox  -S Track2_8000.wav Track2_8_all.dat
```

The -S caused a progress monitor to be displayed. The input file (Track2_8000.wav) was the original Hess .wav file. The output file extension, .dat, indicated to *sox* that the output format was to be data or flat ASCII. The output file name was Track2_8_all.dat.

3. Step 2 generated a very large file, over 6 Gb. The size was reduced a little by using awk to reformat the file. The following awk script was used. [2]

```
if (NR>2)
     {
      printf ("%13.6f %14.7e\n", (NR-3)/8000,$2)
     }
```

The first two lines of the *sox* output contain the sample rate (8000) and a comment. These two lines are skipped by the NR>2 test above. The time field (column 1 in the sox output) is recalculated because the output of *sox* has a fixed field size that results in decimal places being dropped as the time gets larger. The e format is used for the signal amplitude in order to maintain precision regardless of the magnitude of the number. This may not be critical.

The file that resulted from the awk script was approximately 5/6 the size of the sox output, i.e., approximately 5 Gb. In this reformatting effort, the reduced file was named Track2_8_all2.dat. The first few lines from the file are printed below.

```
0.000000  0.0000000e+00
0.000125  0.0000000e+00
0.000250  0.0000000e+00
0.000375  3.0517578e-05
0.000500  0.0000000e+00
0.000625  0.0000000e+00
0.000750  0.0000000e+00
0.000875  0.0000000e+00
0.001000 -3.0517578e-05
0.001125  0.0000000e+00
```

The first column is the calculated time in 1/8000 s intervals. The second column is the signal amplitude returned by sox.

---

[1] The original Hess file name is "Track2 Demodulated Downsampled 8000__01.wav". The file name in Book H is Track2_8000_Hz.wav.

[2] Additional code is needed to indicate input and output filenames.

Steps 2 and 3 could be handled in C by writing code to directly read the signal amplitudes as binary numbers from the .wav file. I was more comfortable generating ASCII files that I could visually read and assess.

4. The next step was to separate out traces. This required more custom programming because the trace lengths vary from trace to trace (Figure 1), and no trace initiation marks were on any of the tracks on the archive tape. The path through this is

a) read the new ASCII file (the output of the awk script, i.e., Track2_8_all2.dat, looking for a single point amplitude or the energy in a time window that exceeds a set threshold. Do this for the entire input data file to generate a table of pointers to the starting times of the respective traces. This will be discussed as Step 4 in what follows b).

b) read in one second of data from the main data file (Track2_8_all2.dat) after each of the pointer positions from a. This will be discussed as Step 5 below.

Various schemes can be designed to recognize single point amplitudes or window energies, and all will work to one degree or another. An automatic scheme to recover every trace may be elusive. All algorithms may need to be followed with some manual editing.

Using code in a compiled language like C is preferable to using awk because of the execution speed advantage.

In this attempt to recover traces, I applied the same process that I applied in early 2007 to the 1000 Hz Track 2 file. However, I completed the earlier work in awk. I ported the awk scripts for processing the 8000 Hz file to C. The only potential problem in doing this is that the C programs must be compiled with options to handle large files.

I implemented tasks a and b above in two programs, one to generate pointers to the start of each trace (the a task) and one to read the data file and output the traces with a uniform length (the b task). The two tasks could be combined into one application.

The first program[3] is named

TraceStarts.c

and involves an iterative process of reading in 30 ms of data and calculating the sum of the absolute values of the amplitudes. If the sum does not exceed the threshold, the window is moved ahead 1 sample (1/8000 s), and the sum is recalculated. If the sum exceeds the threshold, the time of the start of the 30 ms window is output to a file, i.e., Starts.dat in TraceStarts.c) for use in the next task, i.e., the b task. I chose the threshold by trial and error, and I set the window length arbitrarily. Longer or shorter windows may need higher or lower thresholds, respectively. The Track 2 and Track 4 data files may require different threshold values.

[3] The C files should be opened in a C development environment or with an editor that can maintain C language formatting. Also, some editing of input and output filenames in the posted version of the code may be needed. The file names in the C codes in Book I may refer to Track 4 files instead of Track 2 files as described in this Log.

After detecting the threshold value for one trace, the program reads ahead 3 s but does not use any of the amplitude values that it reads. The program then restarts the process of calculating the amplitude sums and checking for a threshold crossing. This process recovers approximately 5000 trace starting times from the file generated in Step 3, i.e., Track2_8_all2.dat.

5. The file output in Step 4 (Starts.dat) is the set of trace start times. However, you may want to subtract 10 ms or more from the set of threshold crossing times and make a decision later as to just where time 0 is on the traces.

The second of the two programs mentioned in Step 4, i.e., the b task, is named

TraceOutputs.c

This program reads a threshold-exceeding time from the output in Step 4 (Starts.dat) and reads the trace data file, i.e., the output of Step 3 (Track2_8_all2.dat), forward until that threshold-exceeding time is encountered. From this point forward, sample amplitudes are copied from the trace data file to the output file. I output 1 s of amplitude values though this is probably more samples than actually contain seismic information (Figure 1). After 1 s (8000 samples) of amplitude values for a trace is output, the next threshold-exceeding time is read in, and the data file is read forward until that time is encountered.

The output of this step is a flat ASCII file named T2-8kHz.dat.

Most seismic data processing systems will accept input from a flat ASCII file, but each trace is expected to have the same length, which TraceOutputs.c provides. The output of this step (Step 5) could be a binary file, and the following step, i.e., 6, could be skipped.

6. The TraceOutputs.c program, i.e., Step 5, concludes with an ASCII file of signal amplitudes organized in traces of 1 s duration (8000 samples). This can be read with the SU program a2b (ASCII to binary):

```
a2b < T2-8kHz.dat >traces.bin n1=1
```

The input and output files are indicated by the redirection arrows. The input file (T2-8kHz.dat) is the output of Step 5[4]. The output of a2b is a binary file. The n1=1 indicates that 1 ASCII value occurs on each line in the input ASCII file. The use of a2b is explained in the SU documentation.

7. Before the traces can be used in SU, the binary file must have trace headers applied. This is done with the program suaddhead. The command is:

```
suaddhead <traces.bin >traces.su ns=8000
```

The variable ns indicates the number of time samples in each trace, i.e., 1 s of record at 1/8000 s sampling equals 8000. The output extension, su, indicates an SU format trace file. As with a2b, the use of suaddhead is explained in the SU documentation.

---

[4] Be sure to check the input and output file names in the C codes so that they match your file naming convention.

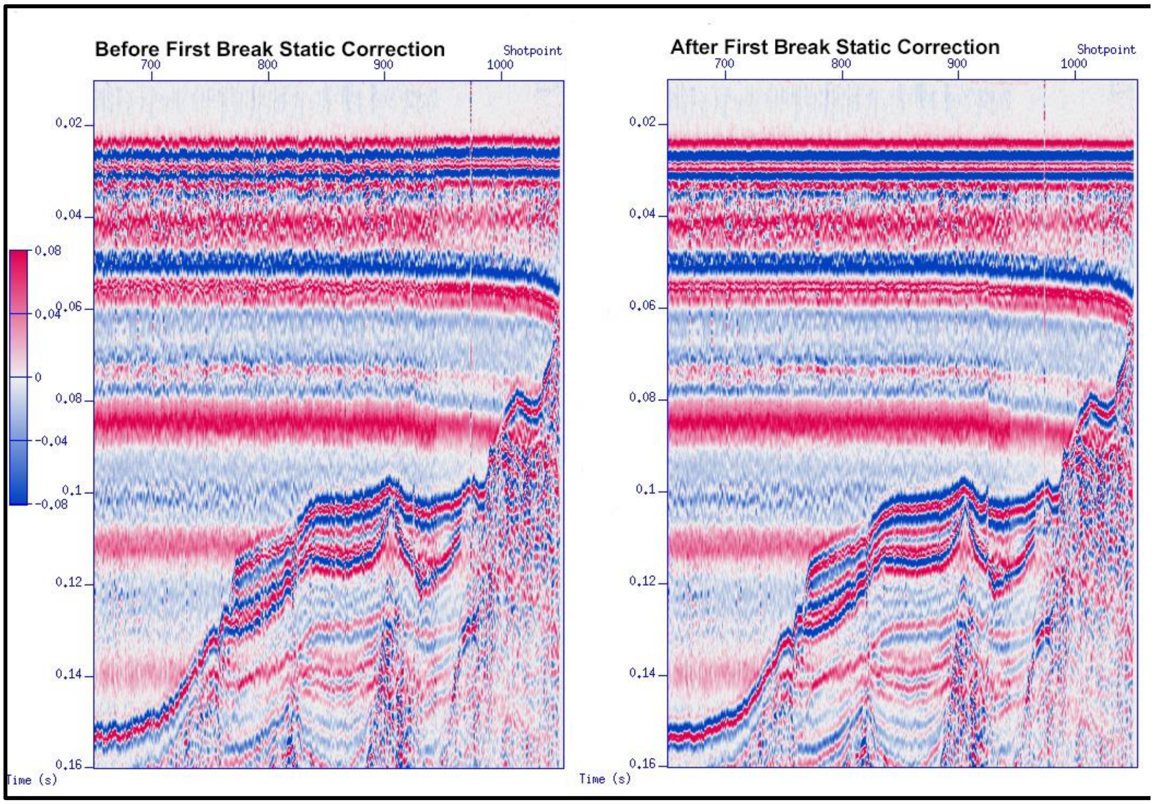

Figure 2. Traces before and after static corrections applied to first arriving energy pulse. The vertical axis is time in seconds. The horizontal axis is shotpoint number. Shotpoints are nominally 10 m apart. These sections show the same 3+ km of the eastern part of Line E. This is the same segment of Line E that is in the Wold (1976) report. The color bar at the left indicates signal amplitude. Fully saturated red is +0.08 on a relative scale while fully saturated blue is -0.08. In the uncorrected panel, the first arriving energy shows a significant amount of jitter or jaggedness from trace to trace. I assert that the peak of the first energy burst should be at the same time on every trace (*maybe this leads to the same result that I would have obtained by picking a peak amplitude instead of an energy level in a window*). Adjusting the traces to accommodate this assertion yields the panel on the right. Reflection events in the corrected panel have greater continuity than corresponding events in the uncorrected data.

8. From here, you are pretty much on your own to complete adding trace header values, specifically trace geometry. You will probably also want to apply static corrections to the individual traces to account for the trace to trace amplitude differences that caused the direct wave arrival times recognized in the energy summing step (Step 4) to be somewhat different from trace to trace (Figure 2).

You may want to separate the traces for individual lines into separate SU files. SU has a windowing module for this. The group and source locations in the SU[5] file in this archive

[5] The SU file referenced in the Log was replaced in 2013 with SEG-Y files as described in the Epilogue.

assumed that the times in the .wav file could be linearly transformed to distance, i.e., 10 m intervals[6] . You may want to tweak this depending on your own analysis of the survey line lengths, e.g., after studying the Silverman et al. (1971) base map in this archive. The line coordinates can also be recovered after georeferencing the Silverman et al. (1971) base map to a standard base in a geographical information system (GIS).

Archive Files Referenced in the Log

Track 2_8000.wav
TraceStarts.c
TraceOutputs.c

##
##

Epilogue

By
Robert W. Lankston
11-19-2013

Since the work described in the January 2008 Log above was undertaken, I have conducted a variety of experiments with the data that were recovered from the USGS archive tape. During the image-to-data effort that is described in Book K, I determined the x-y coordinates for the line end points and turning points in a geographical information system (GIS) as suggested in Step 8 of the Log. I added those x-y coordinates to the headers of the traces generated through the image-to-data process. I assumed that the respective sources and receivers were coincident, i.e., zero offset. I applied the same line end and turning point coordinates to the traces recovered from the digitized archive tape. In general, one should not expect traces to be coincident between the image-to-data traces and the traces generated from the Hess .wav files for any given line.

I replaced the single SU file from the 2008 work described in the Log with individual SEG-Y files for the respective six lines that I recovered from Track 2 on the USGS archive tape. I applied only minimal processing to the traces.

The traces recovered from the archive tape show a significant amount of ringing, i.e., the generally horizontal red and blue stripes visible in Figure 2. To reduce the ringing, every trace in a given line was summed or stacked, time sample by time sample. The logic is that the geologic features in the traces would be random and would add out of the sum while the ringing would add in-phase. The sum trace, normalized by the number of traces in the summation, was then subtracted from each trace in the line. This process is described more fully in the Lankston (2011) article that is in Book J.

Then, the traces were filtered with a zero phase 35, 70, 250, 350 Hz trapezoidal filter with notches at 60, 120, 180, and 240 Hz. I set time zero arbitrarily, and that should be checked against the redisplayed images. The final step was to down-sample the traces from 8000 Hz

---

[6] As indicated in the Otis et al. (1977) paper.

sampling to 1000 Hz sampling and convert from SU format to SEG-Y format. Six 1 ms sampling trace files, i.e., Lines A, B, C, D, E, and F, are in this book. These files should be readable and displayable in a commercial or open-source 3-D seismic workstation.

References Cited

Lankston, R. W., 2011, New display of the 1970 Flathead Lake seismic data: Northwest Geology, v. 40, p. 55-62.

Otis, R. M., Smith, R. B, and Wold, R.J, 1977, Geophysical survey of Yellowstone Lake: Journal of Geophysical Research, v. 82, p. 3705.

Silverman, A. J., Pevear, D. R., and Prahl, S. R., 1971, Bathymetry of Flathead Lake, Montana: unpublished. (URL: http://scholarworks.umt.edu/cgi/viewcontent.cgi?filename=2&article=1015&context=flathead&type=additional)

Wold, R. J., 1976, Marine geophysical instrumentation: Office of Naval Research, NSTL Station, MS.

Citing this Narrative

For “books”, i.e., the ScholarWorks term for a folder or a directory, ScholarWorks will display a Recommended Citation for the entire book. Some of the books (directories) in this collection have what ScholarWorks calls “supplemental material” (files). ScholarWorks does not suggest a citation for the individual files, and you may have occasions when you want a reader to be able to find exactly what you are talking about in your own work.

Therefore, I am suggesting a citation for this narrative. My suggested citation is in a slightly different form than the ScholarWorks form, but that is not critical. Every medium has its preferred format. The important components of a digital citation are the author(s), the title of the work, the year of creation, the name of the collection in which the work appears, and the URL of the work. The following meets those specifications.

Lankston, R. W., 2023, “I. Using the digitized data – narrative by Robert W. Lankston” in *1970 Flathead Lake Seismic Survey*. URL: https://scholarworks.umt.edu/cgi/viewcontent.cgi?filename=0&article=1007&context=flathead&type=additional

You can find the URL for my narrative or any of the other files in this book by moving your mouse to the Download button next to the file title in the list of files for the book. Right click the mouse and, in the pop-up options box, choose to copy the link address. Then, the address is in your clipboard, and you can drop the text string of the address into your work.

Feedback
The standard ScholarWorks format limits how/when an author's email address is displayed. With the author convention that I use, my email address is not displayed for books such as Book D. Actually, my email address is not even in that book's metadata. If you need to contact me regarding the content, a broken URL, and so forth, you can use:

rwlankston@gmail.com

Last update: 5/29/2023 7:30 PM