Processing the Teapot Dome Land 3D Survey with Madagascar

Karl L. Schleicher[1]

ABSTRACT

This paper explains scripts that process a small 3D land seismic survey with the open software system, Madagascar. The processing sequence includes data loading, geometry plotting, spreading correction, surface consistent decon, scaling, refraction statics application, stacking velocity analysis, and post stack migration. The processing scripts also illustrate how seismic processors display and select processing parameters. The scripts mostly use the trace and header (or tah) programs recently added to Madagascar that are useful to handle data that is not regularly sampled.

The workflow is divided into stages that correspond to a typical processing sequence. A computer processing stage produces printed information and displays that are required to create the processing scripts for the next processing stage. The seismic processor selects which data to process, parameters to use. The displays are also used to verify data is correctly processed before moving to the next stage.

The paper assumes the reader has installed the Madagascar software. The reader may recreate the processing or build on this paper by selecting your own parameters, using different programs, updating programs, or writing new programs. The scripts can also be used as a template to process other 3D surveys.

INTRODUCTION

This paper explains scripts that process a small 3D land seismic survey with the open software system Madagascar. The processing sequence includes data loading, geometry plotting, spreading correction, surface consistent decon, scaling, refraction statics application, stacking velocity analysis, and post stack migration. The processing scripts are also illustrate how seismic processors display and select processing parameters. The scripts mostly use the trace and header or tah programs recently added to Madagascar that are useful to handle data that is not regularly sampled.

The Teapot Dome 3D survey is a land 3D dataset from Wyoming provided by the U.S. Department of Energy and the Rocky Mountain Oilfield Test Center (RMOTC). Data (including unprocessed prestack seismic data, preprocessed seismic data, final migrated image, processing reports, well logs, production history, and GIS data) may be downloaded from the Internet. The intended use of the data is for scientific research, testing and demonstrating software, training end users, and as and exploration/production analog. If you use any of this data in a presentation or publication, acknowledge RMOTC and the U.S. Department of Energy as the data source.

[1] **e-mail:** k_schleicher@hotmail.com

The workflow is divided into stages that correspond to a typical processing sequence. A computer processing stage produces printed information and displays that are required to create the processing scripts for the next processing stage. The seismic processor selects which data to process, parameters to use. The displays are also used to verify data is correctly processed before moving to the next stage.

Following sections describe SConstruct files in separate directories that apply each processing stage. These sections describe the processing stage and include some of the displays and the conclusions drawn for the processing. I encourage you to recreate this processing sequence and to refer to the SConstruct files for the details of the commands to use. You may recreate the processing or build on this paper by selecting your own parameters, using different programs, updating programs, or writing new programs.

To reproduce these results you must first install Madagascar. I processed this data on a Linux laptop purchased in 2010 and a MacBook Pro purchased in 2013. It required about 40 Gbytes for free disk space. I recomment you avoid Virtual machine or Cygwin, because these systems may encounter runtime and incompatibility problems.

The processing follows the basic sequence applied by Excell adapted to the Madagascar software. The Scripts build on the accomplishments of the "SEG 3D Seismic Processing Working Workshop in Houston 2015 Land 3D". See `http://www.ahay.org/wiki/Houston_2015` for more information.

DATA LOADING

The SConsruct file to download the Teapot Dome data from the Internet and convert to Madagascar format is in a separate directory to avoid unnecessary reruns. If you are working in a group it is best to get a copy of the data on a thumb drive to limit internet traffic. After getting a copy of the data on your computer you can comment the Internet download commands in the SConstruct file by adding a hashtag to the front of the lines stating with Fetch (do not miss commenting the continuation lines on the pdf files).

To view the data loading SConstruct script starting from your Teapot Dome directory, (`$RSFSRC/book/data/teapotdome` in the Madagascar download directory) type into a terminal window:

```
cd Fetchpaper
gedit SConstruct
```

Use an editor to look at the script. Make and updates (eg comment the Fetch commands) and update the file. To run the script, type:

```
scons
```

This will download about 11.4 Gbytes of data, mostly the two prestack traces segy format trace files. The file npr3_field.segy contains the unprocessed or raw field data with geometry loaded in the trace headers and npr3_gathers.segy contains the preprocessed data. On a good day downloading takes about 40 minutes using my home internet service and much longer if I am sharing a busy public network. The segy data is converted to Madagascar format (i.e. .rsf files) using sfsegyread. Each segy file creates a trace file and a trace header file. A summary list of a trace header files is created using the sfheaderattr program. It shows the trace headers contain fldr, tracf, iline,xline, offset, sx, sy, gx, and gy. These header keys are used throughout processing. The summary listing is used so often when building later SConstruct files that it is stored in a text file, fieldheaderattr.txt and included in some of the later SConstruct files. A detailed listing of some of the important headers can be created by running:

```
sftahread input=npr3_gathers.rsf  \
| sftahgethw  key=fldr,tracf,iline,xline,offset,sx,sy,gx,gy >/dev/null
```

followed with a quick control-c.

If you look carefully at sx,sy,gx,gy and offset you will notice the source and receiver (x,y) coordinates are scaled by 1000 to avoid rounding when stored as integers in the segy headers. Most seismic processing is done using (iline,xline,offset), but it may be necessary to scale the x,y coordinates before programs that use the source-receiver azimuth (e.g. azimuth dependent nmo, 3D dnmo, prestack Kirchoff migration).

The pdf file, 3dload_Teapot_Dome_3D.pdf, is downloaded and displayed. It describes the processing grid or the four corners, the relationship between the real world (x,y) coordinates and the (inline,xline) bin numbers. This is a critical parameter for binning, the process that computes and loads the (inline,xline) coordinates into the trace headers. These attributes are already in the Teapot Dome trace headers, but this information is almost always included with the seismic data because it is critical to connect the data to the physical world. Some program may require the four corners, and it is much easier to have them supplied than inferring them from trace headers.

The pdf file teapot_processing.pdf provides some basic information about the field parameters and the processing sequence. The field parameters may help you understand the geometry plots described in the next section. The processing sequence is typical for land seismic processing and include:

1. Refraction statics, which are often computed by the field crew.
2. Surface consistent amplitude and decon
3. Two passes of velocity analysis and residual statics

This paper recreates part of this sequence using Madagascar.

The PDF files and the headerattr print indicates processing units are feet. The PDFs lists the bin size as 110 feet by 110 feet, group interval is 220 feet by 880 foot line spacing, source interval 220 feet by 2200 foot line spacing, and velocities. The dmo velocity field is downloaded and printed by SConstruct. It is good starting information about the stacking velocity and will be used later in the processing.

The final piece of information created in this directory is the segy text headers. The files are created by sfsegyread and can be listed to the screen with

```
cat npr3\_field.thdr
or
cat npr3\_gathers.thdr.
```

GEOMETRY DISPLAY

A stacking diagram is a good way to become a familiar with the data you have downloaded. Assuming you are in the `$RSFSRC/book/data/teapotdome/Fetchpaper` directory, you can look at the SConstruct file to make a stacking diagram or fold plot with the commands:

```
cd ../geompaper
gedit SConstruct
```

The sffold command uses the trace header file created by sfsegyread to make a 3D histogram that shows the trace distribution in offset, xline, and iline. Sffold required the trace headers to be in floating point format and sfsegyread outputs the data in integer format, so sfdd is used to convert the headers to float. The program requires the header names and the first, number, and increment for offset, xline, and iline. The sfheaderattr run in the `Fetchpaper` directory provides the minimum and maximum of these headers. The increments (d1, d2, d3) are small enough to get good resolution on the plot, usually the group interval, xline interval, and iline interval. When you run:

```
scons
```

a fold plot or stacking diagram movie will be displayed on the screen. The movie shows the (offset,midpoint) trace distribution, one iline at a time. The first thing you will notice is some (xline,iline) locations do not have traces. 3D surveys are not usually perfect rectangles. They are collected over an exploration objective and there may be surface access restrictions. The standard approach in processing is to expand the data with zeros to a make a rectangular volume.

Stop the movie and position it to iline 144 and the plot should look like Figure 1. I selected iline 144 because it looks like this midpoint line very near to both a receiver line and a shot line. The relationship between the surface geometry and midpoint stacking plot is easier to observe. On this xline there are traces with almost zero offset and some traces form linear (xline,offset) patterns. These are traces from shots on the receiver line. The distance between zero offset traces is 20 xlines or 2200 ft (since xlines are 110 ft). 2200 feet is the shot line interval described in teapot_processing.pdf. There are also traces that form hyperbolic patterns in (xline,offset). The hyperbola apexes approximately line up with an offset interval of 1660 ft, twice the receiver line spacing.

Type q in the movie window to quit from the iline foldplot movie and an offset fold plot movie will plot. If you stop the movie at offset 400 or 5000, it should look like Figure 2. These plots show the (xline,iline) trace distribution for offsets 200 +- 100 or 5000 +- 100 feet. This is called a 'fold map for a small offset subset'. The survey shape was designed to just cover the Teapot Dome anticline.

Run:

```
scons view
```

to recreate some foldplots and Figure 3, a map of the source and receiver coordinates will appear. A few observations can be made:

1. The shot lines are at a 45 degree angle to the receiver lines.
2. Shot lines are further apart than receiver lines (as described in teapot_processing.pdf). This is a common survey design deploys more receivers than the more expensive shots.
3. Shot and receiver lines are nearly uniform. The variation is likely to be due to surface access problems. There is more irregularity in the shots, which are harder to place then geophones.

The take away from this section is that traces in a 3D land surveys have a complex distribution of source and receiver coordinates. There are books on the topic (eg 3D Seismic Survey Design, G. J. O. Vermeer)

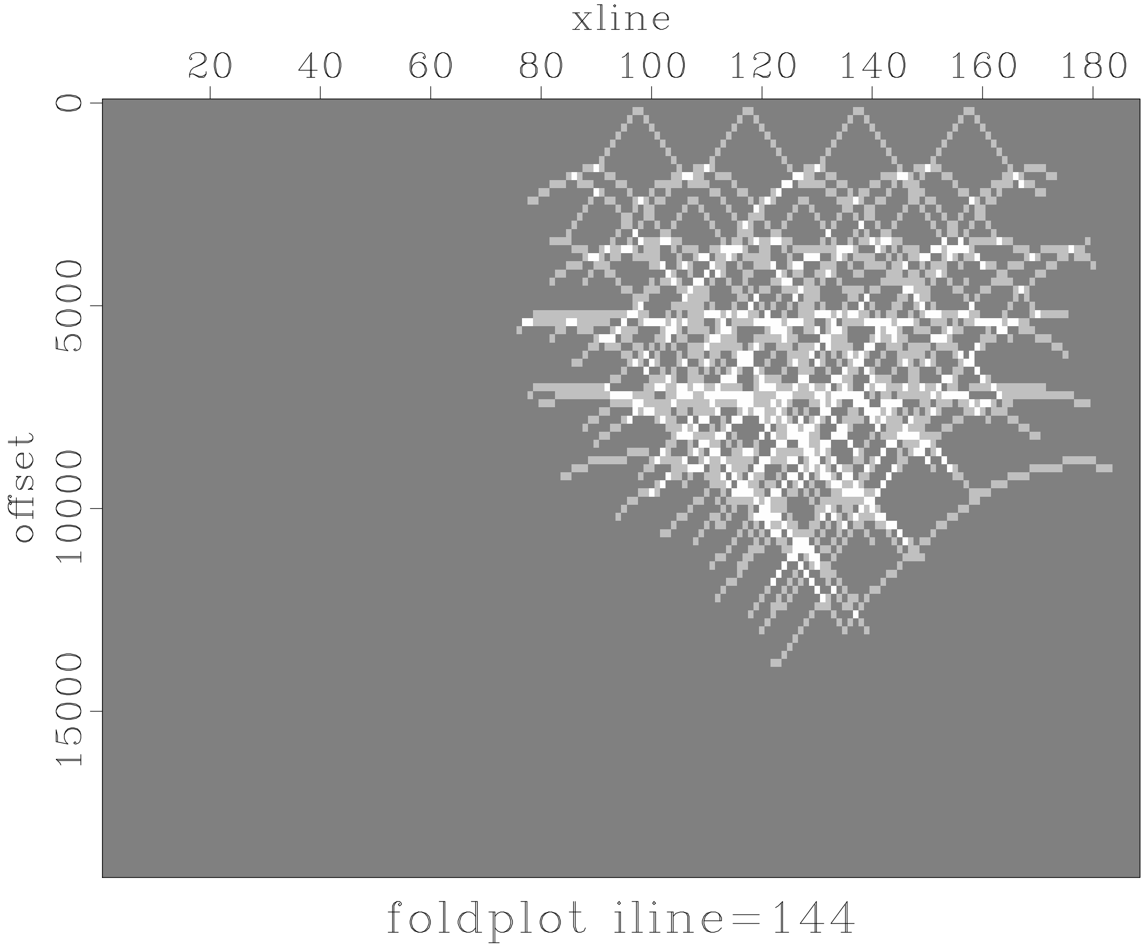
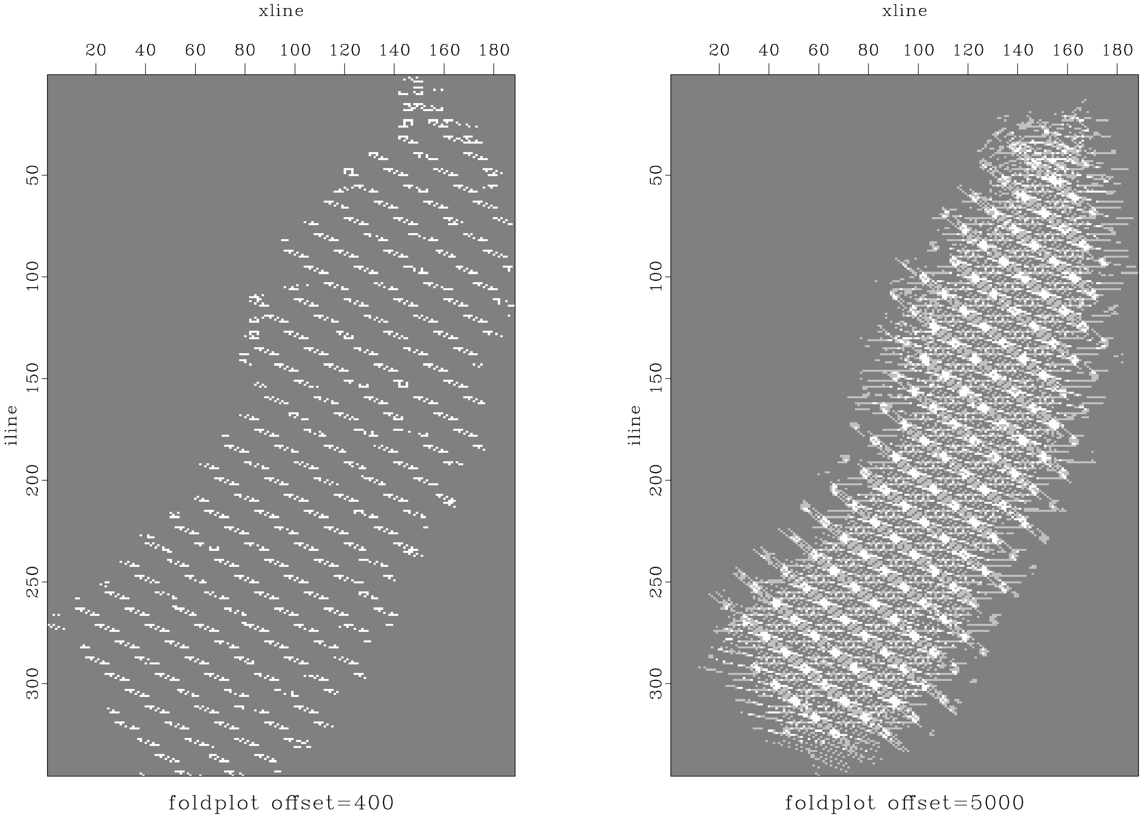


Figure 1: Fold plot for iline 144 shows the trace (xline,offset) distribution. geompaper/ foldplot144



foldplot offset=400

foldplot offset=5000

Figure 2: Fold plots for offsets 400 and 5000 meter shows the trace (xline,iline) distribution
geompaper/ foldmap-400-5000

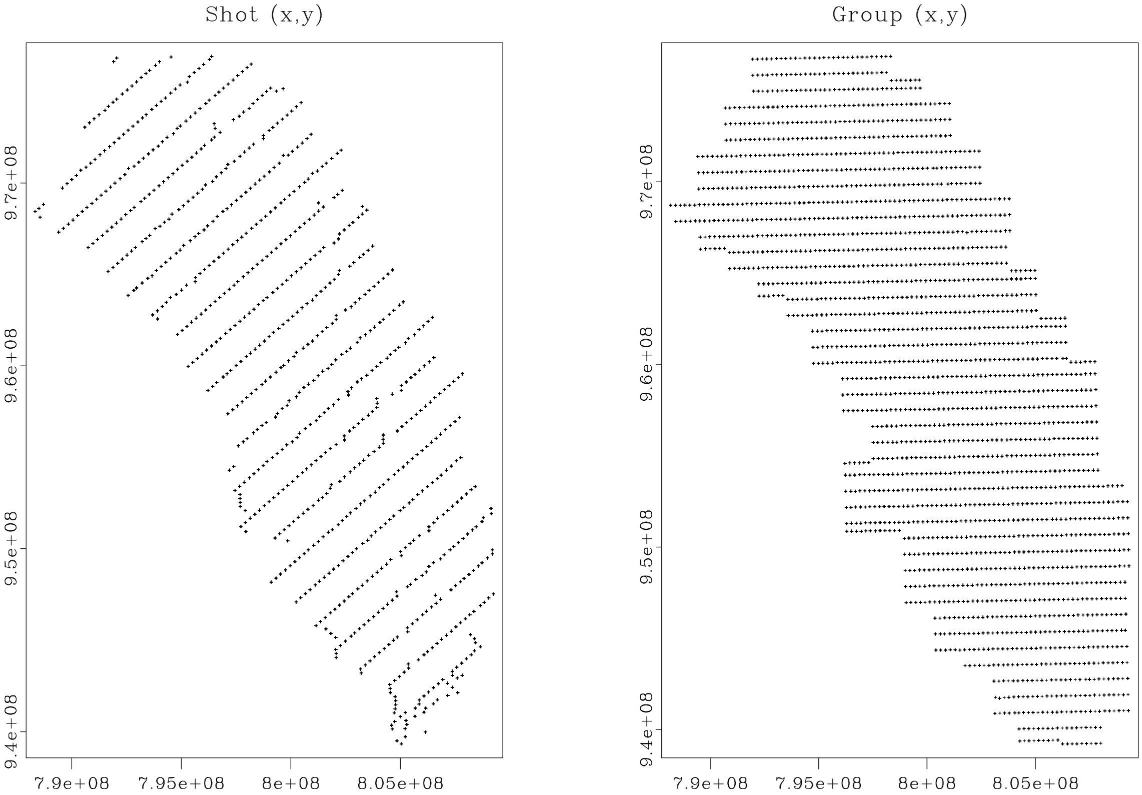


Figure 3: source and receiver coordinate plots. geompaper/ shots-receivers

FIRST LOOK

It's important to make a few plots of seismic traces to get an idea of the signal quality and the problems that will need to be addressed during processing. This section also applies:

1. some basic single trace processing (tpow, mute, ags, decon)
2. picks a predecon mute
3. single velocity normal moveout
4. picks a stack mute
5. brute stack a single line

Starting from the `$RSFSRC/book/data/teapotdome/geompaper` directory, you can look at the SConstruct file for the 'first look' processing with the commands:

```
cd ../firstlook
gedit SConstruct
```

The `firstlook/SConstuct` file makes use of the 'trace with header' (tah) programs. These program names all start with sftah, for example sftahnmo, sftahagc and sftahmute. The tah programs are similar to Seismic Unix programs and are especially useful for handling irregularly sampled data and single channel processing. It took some experiments with sfheaderattr and sftahgethw to determine good header keys and values to use to select and display a few shots. Sftahsort was used to select a few shots and order them by (fldr,tracf) and sfwrite put the data to an output file ordered by (time,tracf, fldr). After some experiments I learned the last two samples on most traces were unusually large and I omitted these time samples with sftahwindow. There are a different number of traces on each shot, so I used sftahwrite to select a few shots and create a file organized by the (time,trace,shot) header keys.

After reading the SConstruct file, you can run it from the command line with:

```
scons
```

The first plot you will see is a few shotpoints from the input file. One of the shotpoints looks like Figure 4. This plot shows the shot 'ffid 214' was recorded by 17 receiver cables. The shotpoint is located very near one of the cables and the first arrivals on that cable are approximately linear because the trace offsets are nearly regularly sampled. First breaks on cables further from the shot are hyperbolic. Inline offsets are approximately linearly sampled and there is a significant crossline offset. This results in hyperbolic offset distribution, ie:

```
totaloffset=sqrt(inlineoffset**2 + crosslineoffset**2)
```

To get a better look at the data, amplitudes are scaled by t**2, a correction of the amplitude lost due to geometric wavefront spreading. An interactive plot of ffid 214 is created using the sfimage command. You can rubber band a zoom area and create a plot similar to Figure 5. The plot look a little different because the plot in this paper is created using sfgrey while the interactive plot is created using sfimage. One big difference is sfimage laterally interpolates trace amplitudes while sfgrey just uses the amplitude from the nearest trace.

Figure 5 shows linear first breaks, noise on near offsets, and a large linear event that passes 4 seconds near trace 520. You can measure the velocity of this event using the sfimage plot by clicking the right mouse button (on a Mac trackpad place TWO fingers on the trackpad and while the two fingers are touching the trackpad, click the trackpad button). The mouse coordinate will appear on the upper left of the display. I picked the event time and used the group interval (220 feet from the pdf in the `Fetchpaper` directory) to estimate the event velocity:

```
Time           trace  offset
minimum  540    0
4.066s        519    (540-519)*220 feet=4620 ft
1.314s        534    (540-534)*220 feet=1320 ft
v=delta-offset / delta-time = 3300/2.752=1199 ft/s.
```

This is close to the speed of sound in air, 1115 ft/s (I looked this up on the Internet using Google), confirming my suspicion that this is an airwave.

You can identify ground roll on Figure 5. It is the faint energy with time increasing linearly to about 2.1 s. on the trace 500. It is very weak and it looks like the field arrays have successfully attenuated this noise. The final events you can identify are the signal, the reflection events which have hyperbolic time with apexes from .8 to 1.1 s. It is easier to see on the zoomed display in Figure 6.

I used the sfimage interactive plot to pick a pre decon mute. I turned the mouse coordinates on (see the instruction in the description of Figure 5 above) and used it to list the mouse (trace number,time) pairs as I pointed at the location where I wanted the mute to start. I converted the trace numbers to offsets (remembering the inline receiver interval is 220 feet). I wanted to remove most of the first breaks on longer offsets. On traces with offset less than 1760 feet, I selected a less severe mute. This preserves some near offset, small offset data to image as shallow as possible. There is an oppotunity later in the processing sequence for a more severe stack mute after NMO. Figure 7 shows the data before and after the mute.

Figure 8 shows the data at four processing stages with incremental improvement from each of the processes. The processes are amplitude correction (sftahgain tpow=2), agc (sftahagc) , decon (sttahpef), and statics (sftahstatics).

The stack mute is applied after NMO, so we need a velocity function. I selected a velocity function from the center of the project from the file $RSFSRC/book/data/teapotdome/Fetchpaper/npr3_dmo.vel. This file contains velocity functions for several locations. To identify a velocity from the center of the project I converted CDP to iline,xline. The information in the file $RSFSRC/book/data/teapotdome/Fetchpaper/3dload_Teapot_Dome_3D.pdf defined the relationship between (iline,xline) and CDP:

```
iline=(CDP-1)/188+1
```

```
xline=CDP-(iline-1)*188
```

I converted several velocity CDPs and looked at the fold map in Figure 2 to select the a velocity location near the center of the survey (CDP 31705, iline169, xline 121).

Figure 9 shows part of a shotpoint after nmo with and without the stack mute applied. This plot was created to check the stack mute. I used an offset=depth mute estimated from one of the dmo velocity functions estimating depth = vdmo*t/2. I checked the mute using sfimage (like I checked the pre decon mute) and using this display.

Figure 10 is a brute stack of line 141 from the 3D survey. The brute stack is the name of the first stack on a land survey made usually using a single velocity function and no residual statics.

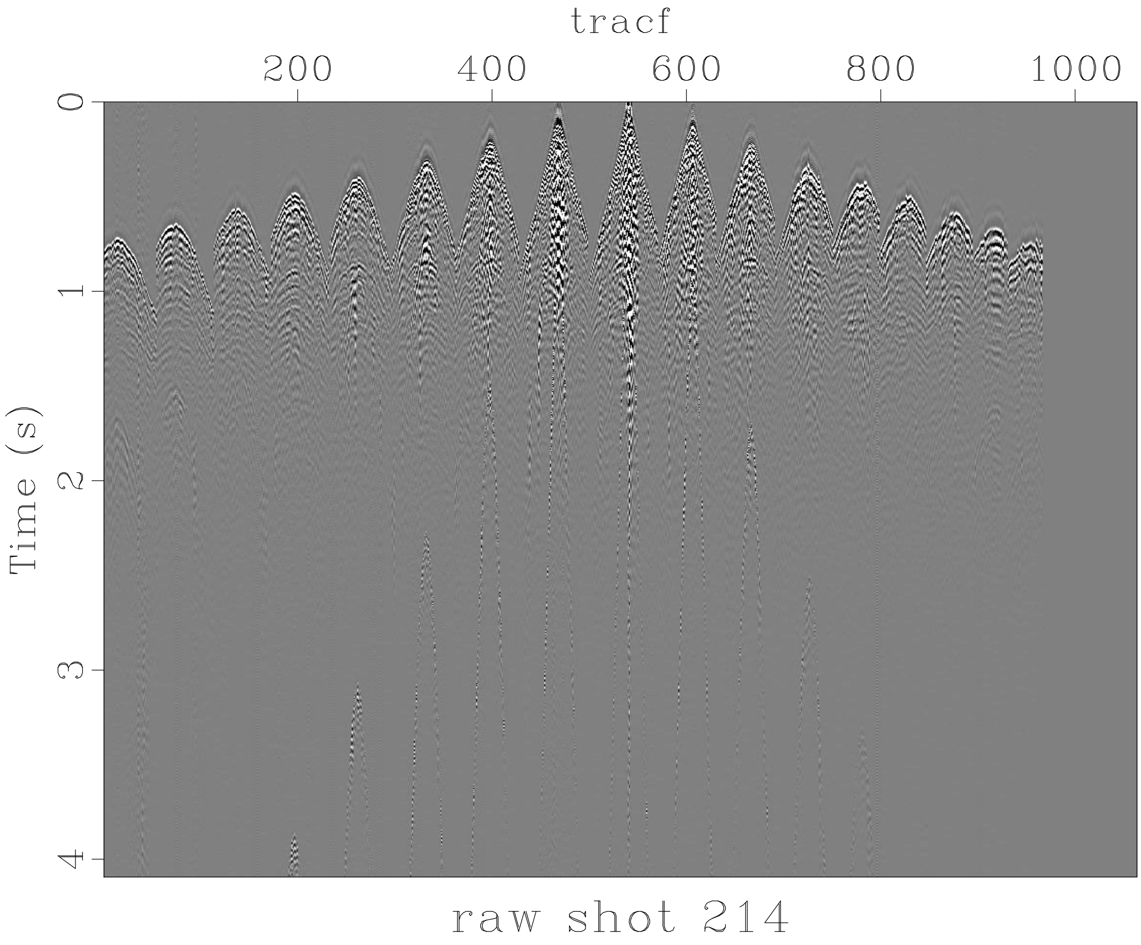


Figure 4: Field traces for field file identifier (FFID) 214. Shows 17 receiver cables recorded data on this shotpoint. firstlook/ rawshot1

SURFACE CONSISTENT DECON

The previous section, first look, created a few displays for the trace data, picked a predecon mute, a stack mute, and showed improvement from some basic processing. The 'first look'

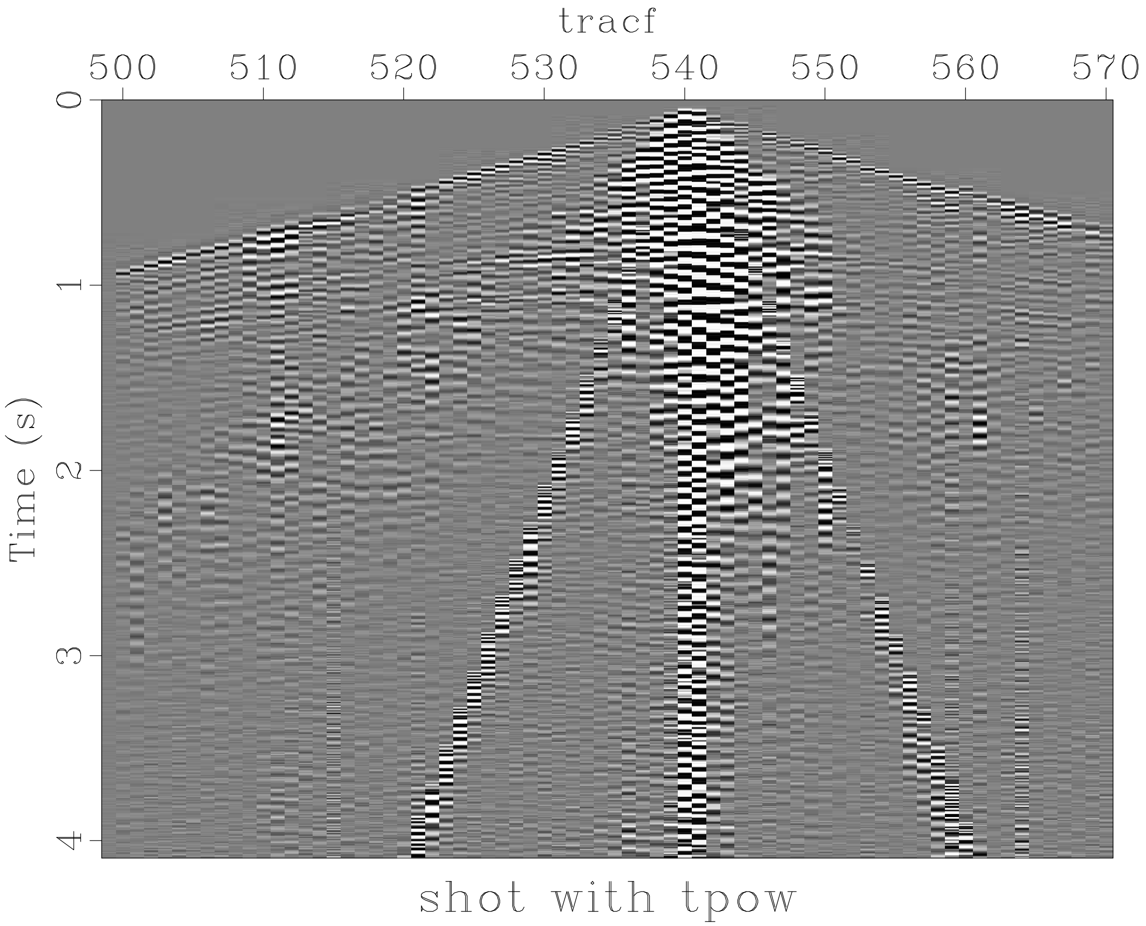


Figure 5: Zoom of the central cable in Figure 4. This plot shows linear first breaks, noise on near offsets, and a large amplitude, linear event that passes 4 seconds near trace 520. The large amplitude, linear event is an airwave. firstlook/ tpowshotzoom1

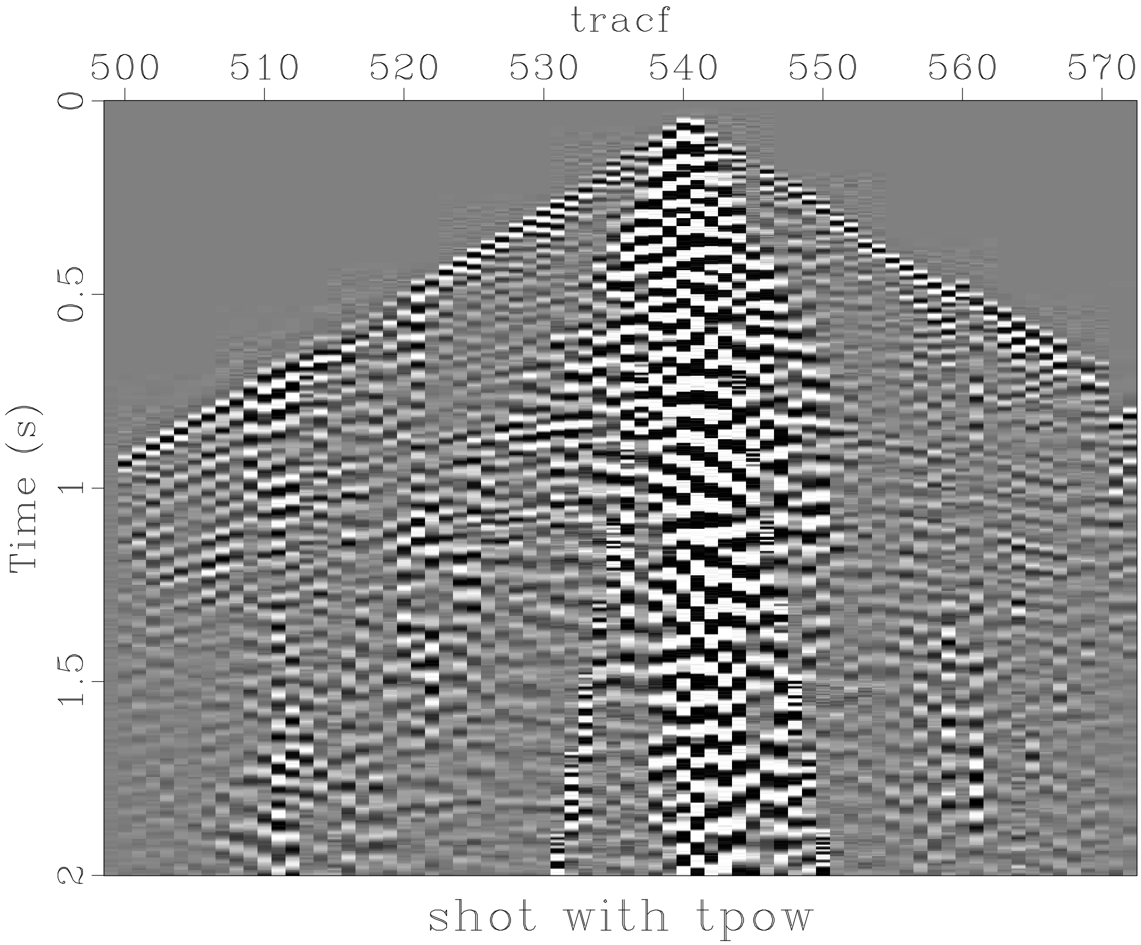


Figure 6: Zoom of the first Figure 5 that created to show the signal. The reflections are the events with hyperbolic time with apexes from .8 to 1.1 s. firstlook/ tpowshotzoom

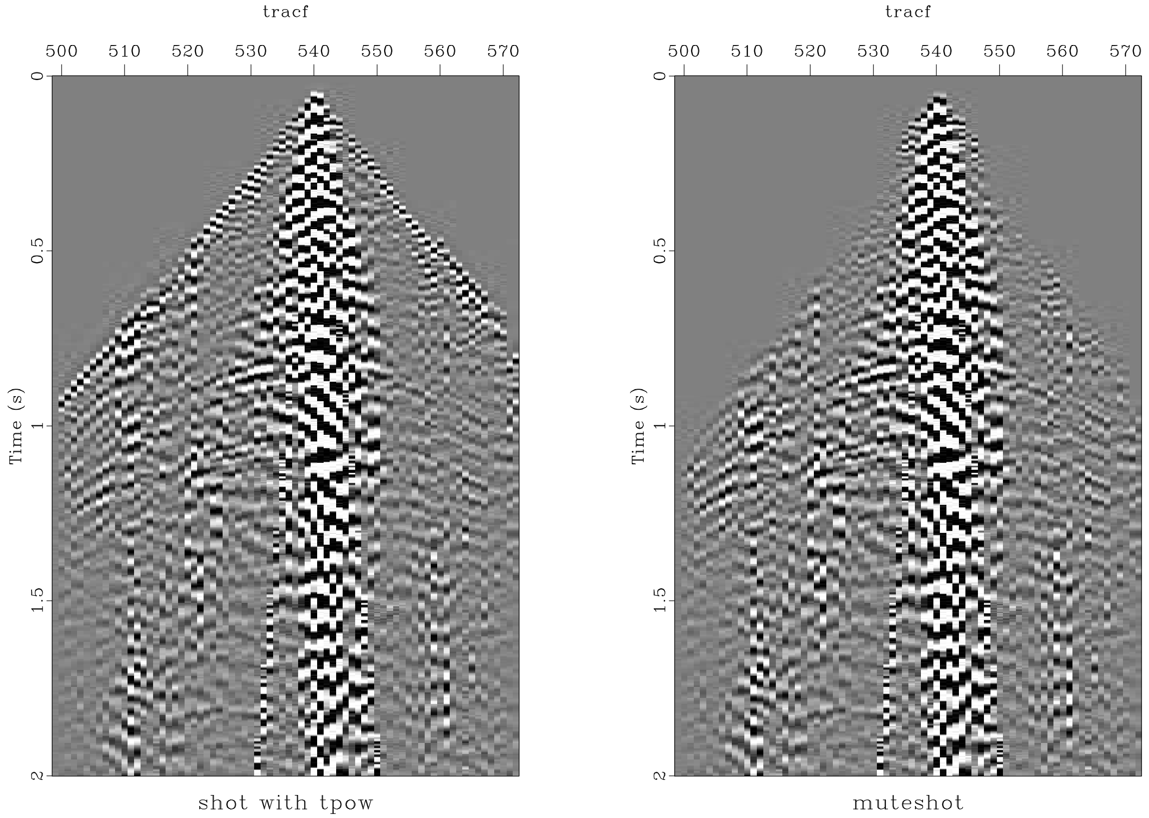


Figure 7: Data before and after the predecon mute. firstlook/ mutecheck

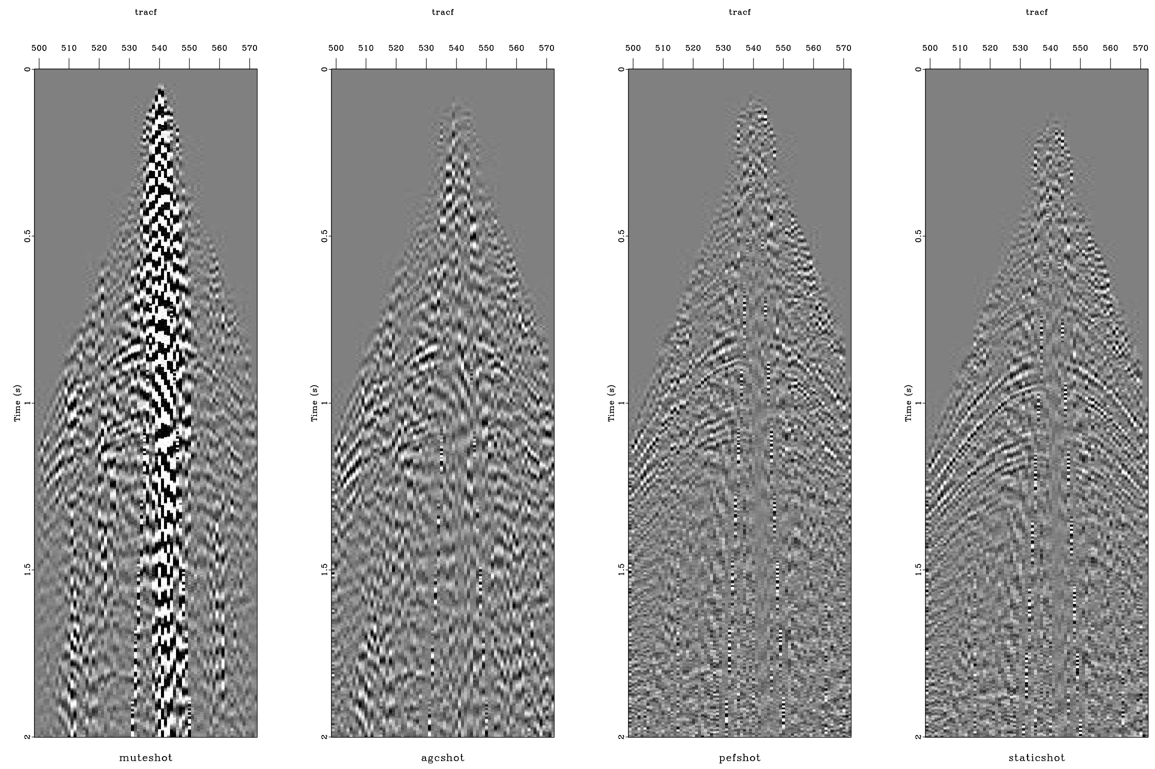


Figure 8: Data at 4 processing stages showing incremental improvement for each process. Displays are after amplitude correction (sftahgain tpow=2), after agc (sftahagc), after decon (sttahpef), and statics (sftahstatics). firstlook/ shots

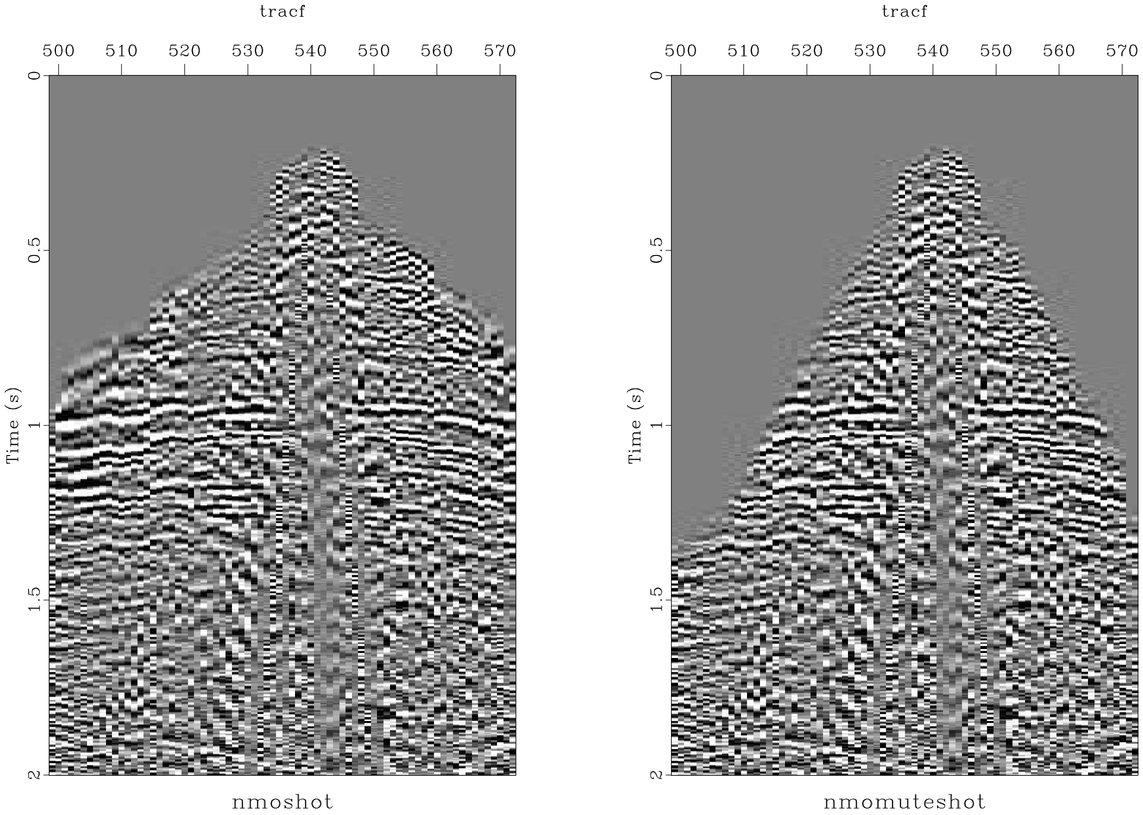


Figure 9: Data after nmo with and without the stack mute applied. This plot checks the stack mute. I used an offset=depth mute estimated from one of the dmo velocity funcions estimating depth = vdmo*t/2. firstlook/ nmomutecheck

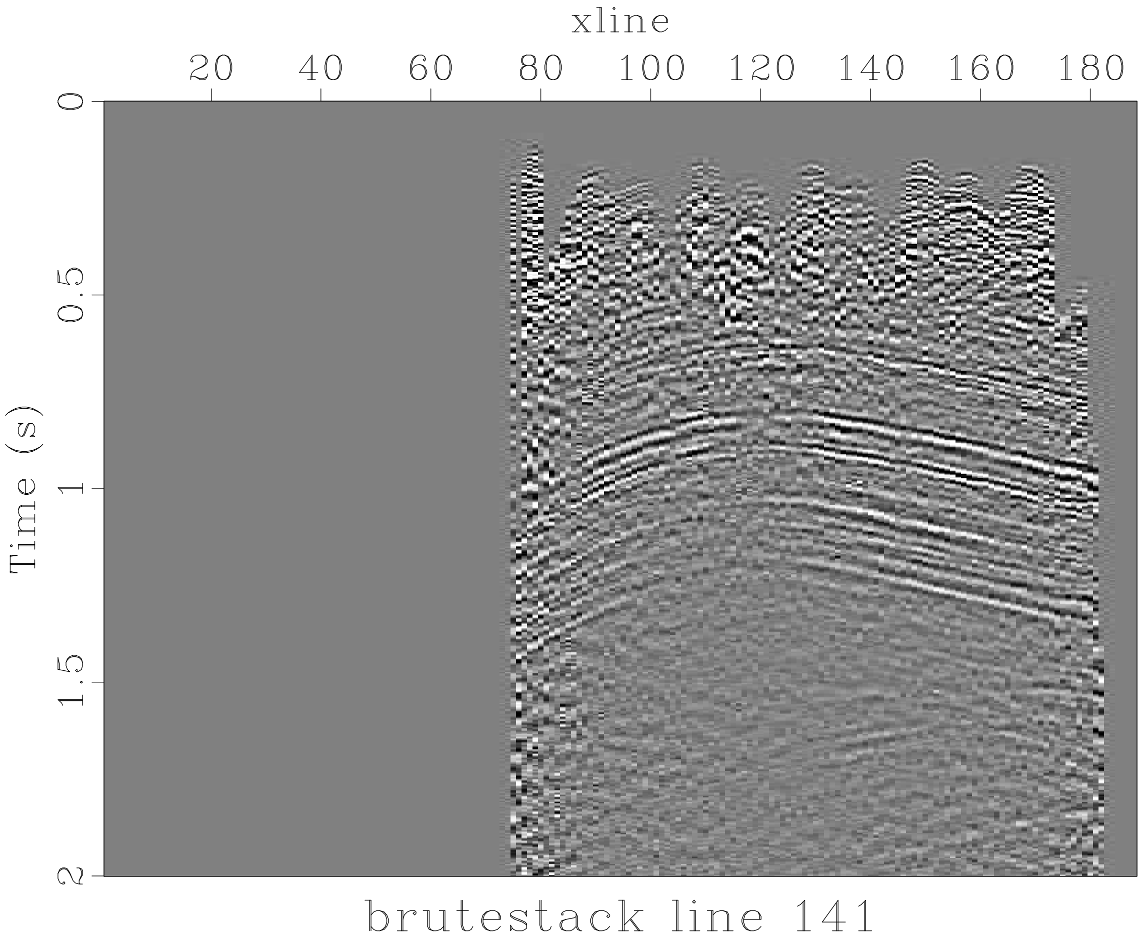


Figure 10: A brute stack of line 141. firstlook/ brutestack141

section used single trace deconvolution. Surface consistent decon (Hutchinson and Link, 1984; Cary and Lorentz, 1993) is more stable on data with poor signal and is widely used for land processing. Surface consistent decon computes a constant operator for each surface location.

Starting from the `$RSFSRC/book/data/teapotdome/firstlook` directory, you can look at the SConstruct file for the surface consistent decon with the commands:

```
cd ../scdeconpaper
gedit SConstruct
```

The key program for this processing stage is sftahscdecon. It was derived from sftahpef and is very similar. The Madagascar program allows a shot decon or a receiver decon to be applied. This is simpler than the multi component decomposition described in the references (Hutchinson and Link, 1984; Cary and Lorentz, 1993) and is an algorithm widely used in industry. The program computes a single operator for each ensemble of traces. To apply shot consistent decon, to data must be sorted to shot order and key='sx,sy' is input to sftahscdecon. The program averages trace autocorrelations and uses the header keys sx and sy to identify the end of an ensemble. At the end of the ensemble the averaged autocorrelation is used to design a prediction error filter (decon filter) and applies the filter to all traces in the gather. After shot decon the data is written to a file so the data can be sorted into receiver gathers and sftahscdecon used to apply receiver consistent decon. Other than the key parameter, the parameters in sftahscdecon are the same as the parameters in sftahpef. This SConstruct file used the same parameter values for the length of the decon operator and the decon design gate (minlag, maxlag, minfor, and maxcorr.) as used in sftahpef in the `firstlook` directory. Since decon will be applied twice in the surface consistent processing, pnoise was increased from .01 to .1 to prevent over whitening the data.

Figure 11 shows a few CDP gathers with shot and receiver consistent decon applied. Figure 12 shows the stack for line 141 with surface consistent decon.

STACK WITH V(T,X,Y)

Previous sections “first look” and “surface consistant decon”, applied normal moveout with a single velocity function from the file: `$RSFSRC/book/data/teapotdome/Fetchpaper/npr3_dmo.vel`.

This section interpolates the velocities so NMO can be applied with a space variant velocity field, V(t,x,y), and the stackes the data.

Starting from the `$RSFSRC/book/data/teapotdome/scdeconpaper` directory, you can look at the SConstruct file for stack with V(t,x,y) with the commands:

```
cd ../velspaper
gedit SConstruct
```

There are two stages for this process. First the velocity is interpolated with a python program, interpvel.py to every to every (t,xline,iline) location. The interpolation is linear in time and uses radial basis functions in space. The interpolated velocity is saved in an rsf

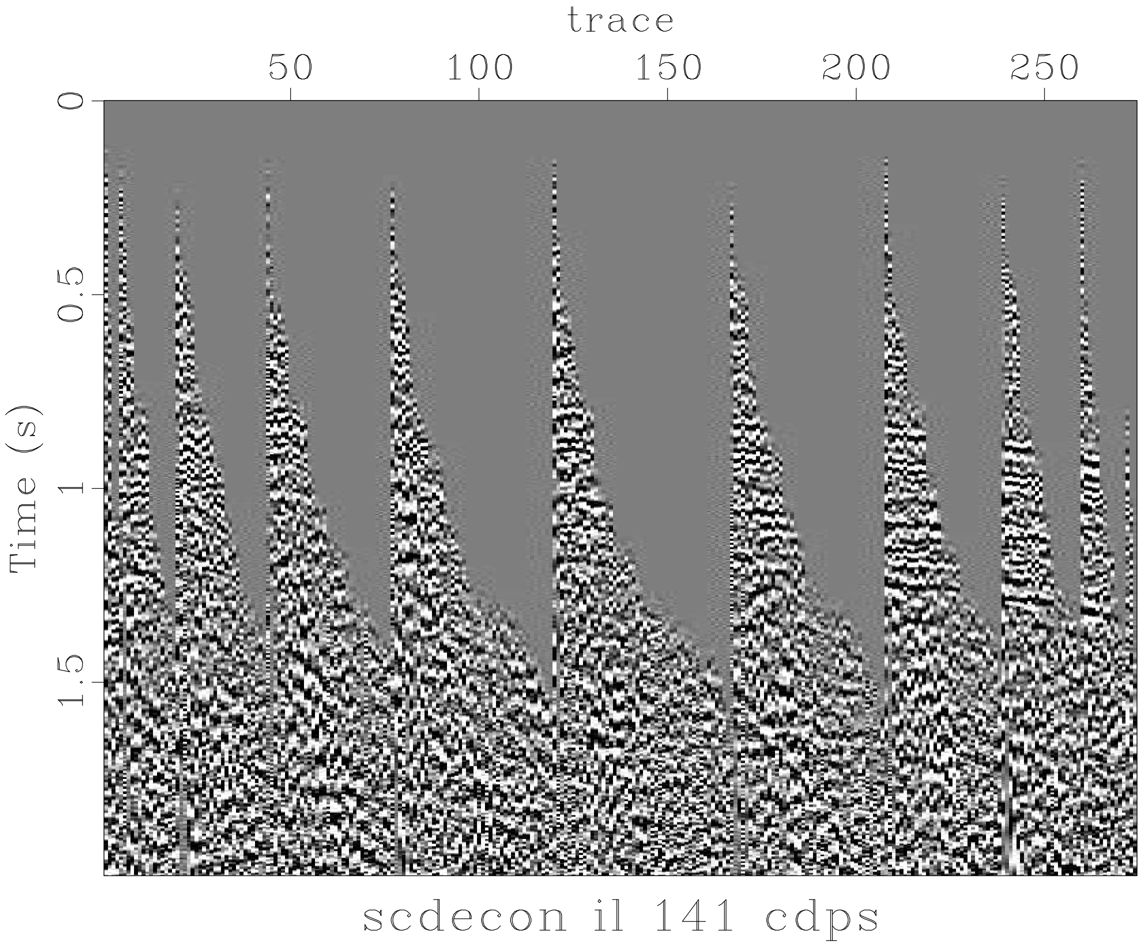


Figure 11: A few CDP gathers with shot and receiver consistent decon applied. scdeconpaper/ scdeconcdps

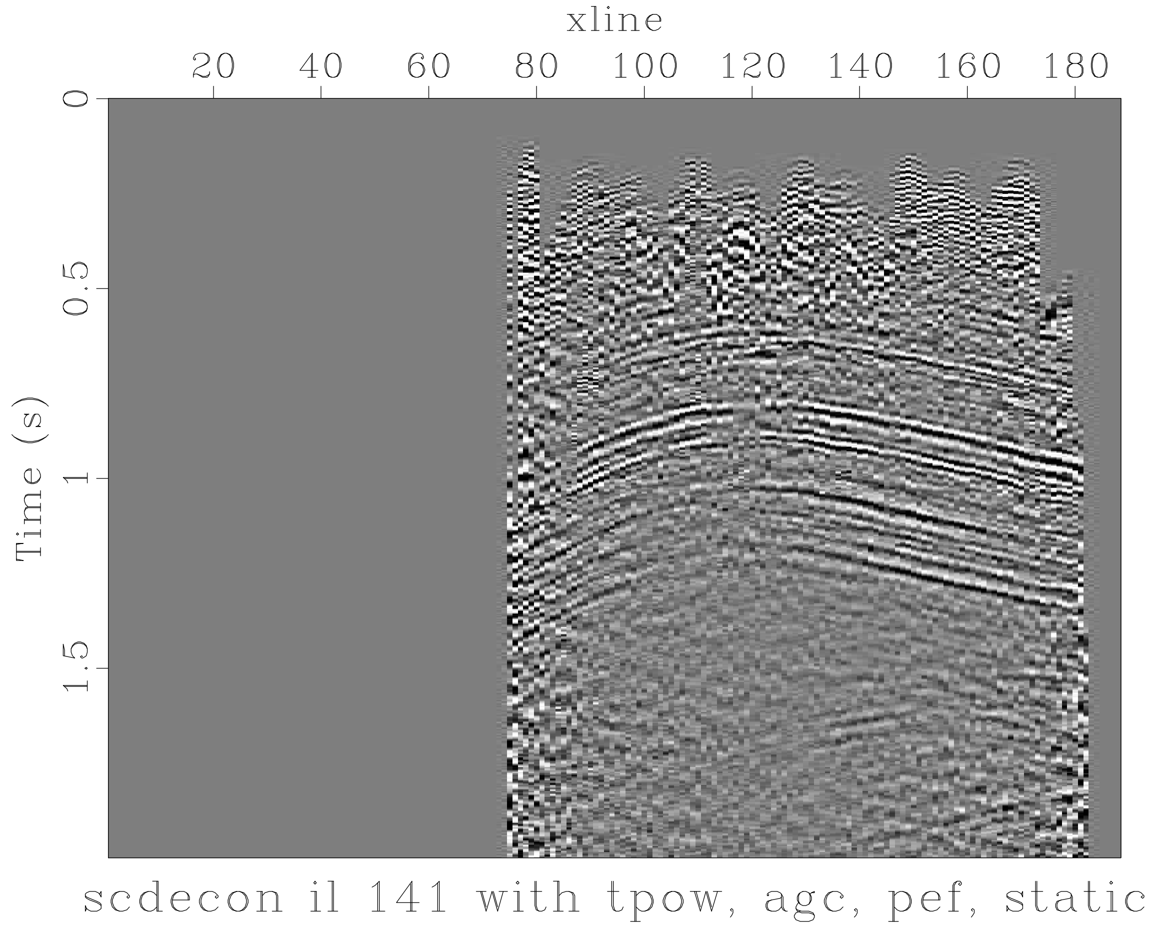


Figure 12: The stack for line 141 with shot and receiver consistent decon applied. scdeconpaper/ scdeconstack141

file and displayed (Figure 13). The second stage is to apply NMO using this velocity field using sftahnmo and create gathers with NMO and stacks. As each trace is processed by sftahnmo, the (t,xline,iline) is read from the trace header and the velocity function for that location is read from file created by interpvel.py. A few CDP gathers and a stack of iline 141 are created and shown in figures 14 and 15. These can be compared to processing with a single velocity function in figures 11 and 12.

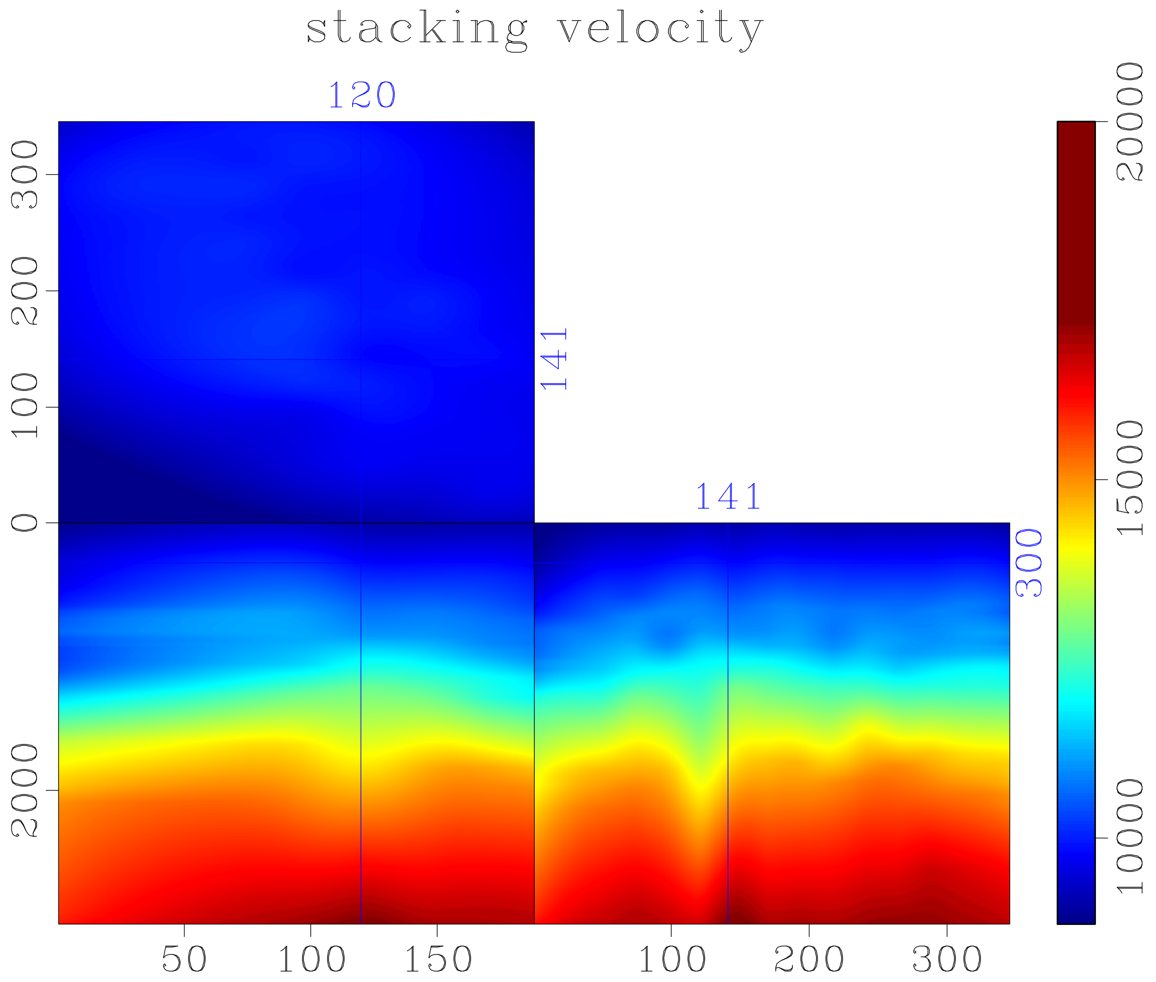


Figure 13: The downloaded velocity field velocity field, npr3_dmo.vel, interpolated using radial basis functions. velspaper/ vel3

MIGRATION

This section migrates the stacked volume using 3-D zero-offset extended split step Fourier migration. It should be possible to significantly improve these results with a faster migration program, higher frequency results, and zeroing portions of the migrated volume that are zero on the input stack volume.

A simple V(z) velocity field is used for the migration. The SConstruct file takes about 8 hours to run migration on my MacBook Pro, even though migration has been limited to

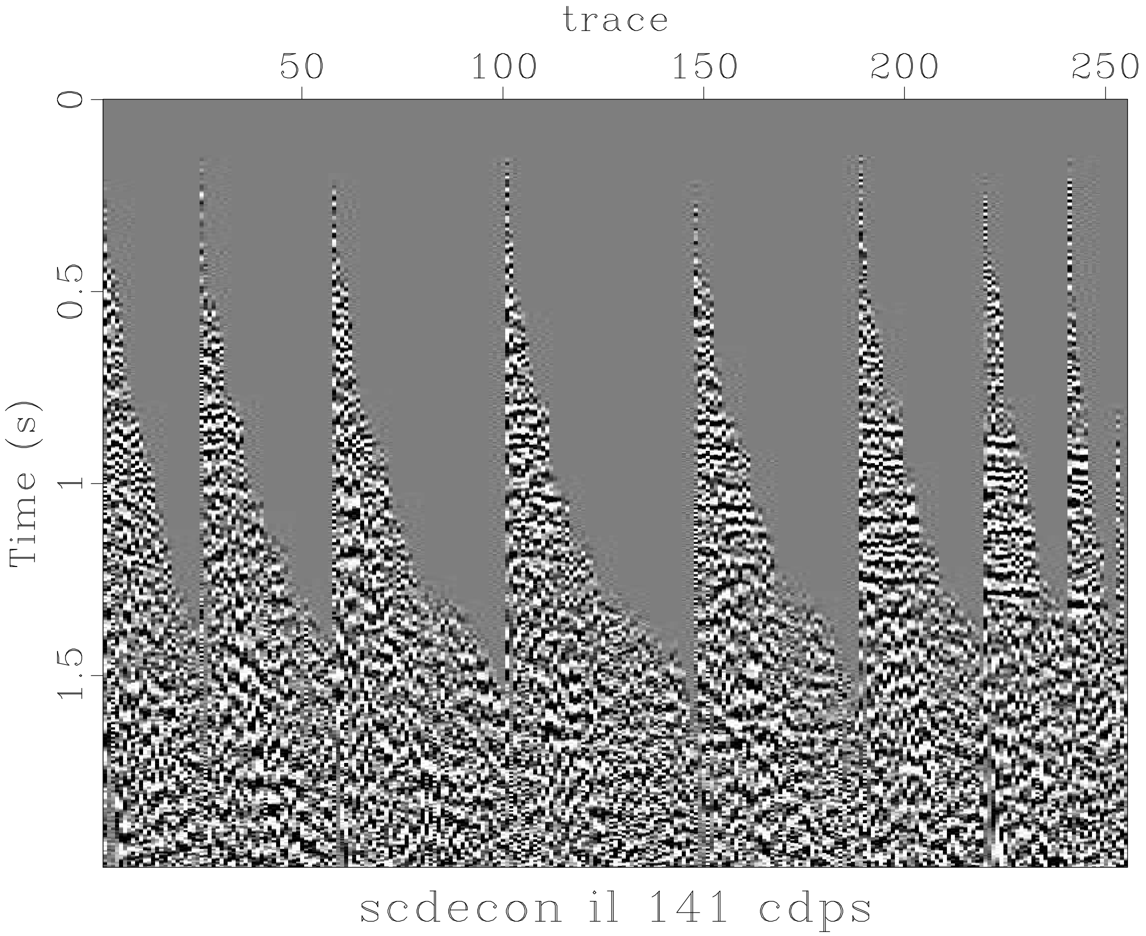


Figure 14: A few CDP gathers from iline 141 with NMO using V(t,x,y) applied. velspaper/ vtxycdps

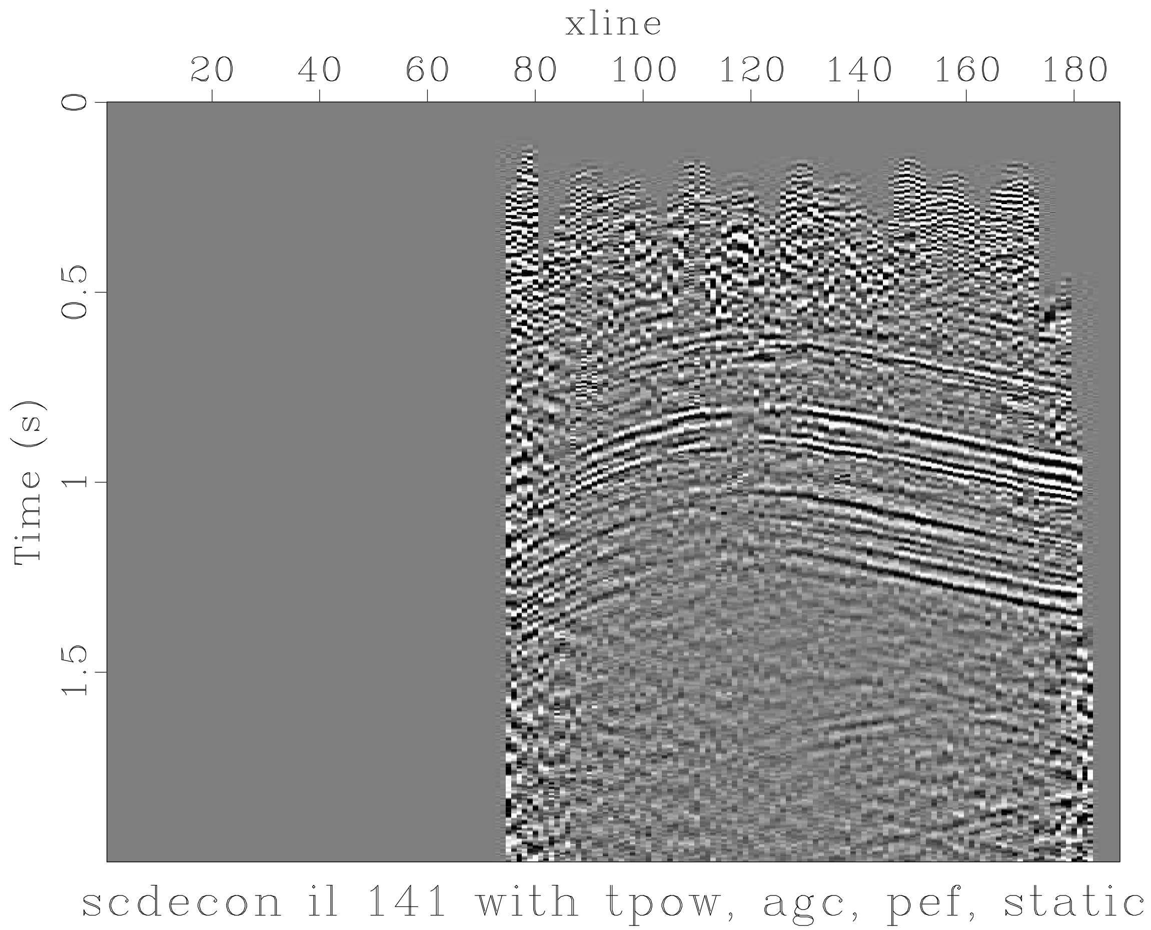


Figure 15: Stack of iline 141 with NMO using v(t,x,y). velspaper/ vtxystack141

20 Hz. This algorithm allows the velocity field to vary laterally and it should be possible to produce equivalent results much more efficiently using phase shift migration.

Starting from the `$RSFSRC/book/data/teapotdome/velspaper` directory, you can look at the SConstruct file for migration with the commands:

```
cd ../zomigpaper
gedit SConstruct
```

This migration algorithm downward continues each frequency through each depth step. The program requires the velocity to be converted from Vrms(t) to interval slowness (1/Vint) in depth and spread into a 3D volume with depth as the slowest axis. The resulting file is slo(xline,iline,depth). The stack data must be converted the frequency and transposed to make frequency the slowest axis. The resulting data in fft(xline,iline,frequency).

After migration using the sfzomig3 program, the data is in depth slices in the file mig(xline,iline,depth). To compare with the input stack volumes, the data must be transposed to make depth the fastest axis and converted from depth to two way vertical travel time. The migrated data converted to time is in the file mig_t.

After reading the SConstruct file and you are ready to compute for a few hours, run the script with:

```
scons view
```

Figure 16 is the final stack with a 20 Hz high cut filter. The maximium frequency in migration was limited to reduce runtime. You can compare Figure 16 to Figure 15 and observe how much signal is lost in order to get a faster runtime.

Figure 17 shows the interval velocity slowness in a 3D cube with depth as the slowest axis. This is the file required to define the velocity to sfzomig3.

Figure 18 shows the real part of the stack data after Fourier transform and transpose to make frequency the slowest axis. This is the file required to define the surface data to sfzomig3.

Figure 19 is the migrated data converted back to time. It can be compared to the commercial migration in Figure 20. One large difference in these results is the commercial migration has zeroed the areas zero on the input stack, a common practice. Another difference is 20 Hz high cut filter applied to limit compute time when creating Figure 19.

CONCLUSIONS

This paper describes basic processing of the Open 3D seismic survey Teapot Dome using the open software package Madagascar. The scripts are distributed with the Madagascar software and the results in this paper can be reproduced on a Linux or Mac laptop computer.

This processing framework has four intended uses:

1. A 3D land seismic processing tutorial

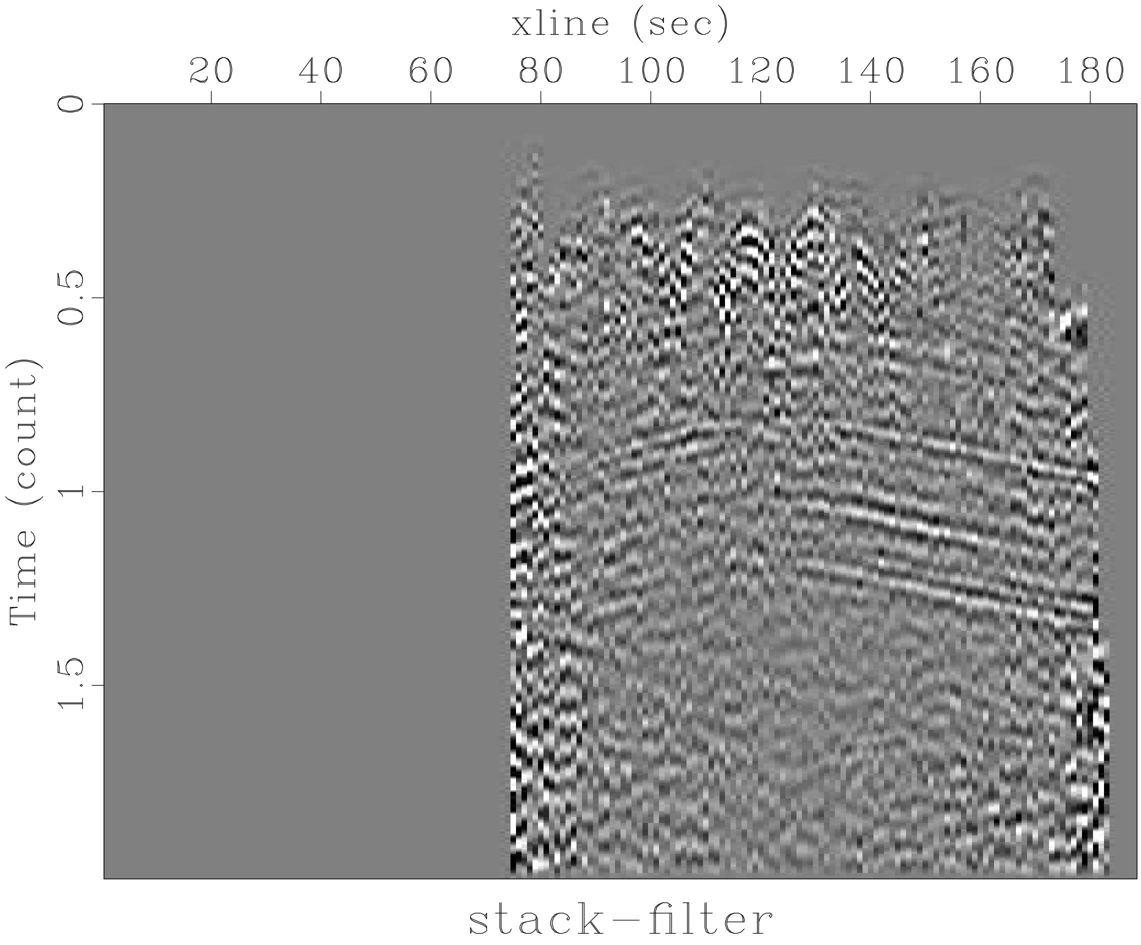


Figure 16: Final stack with a 20 Hz high cut filter. zomigpaper/ stack-filter

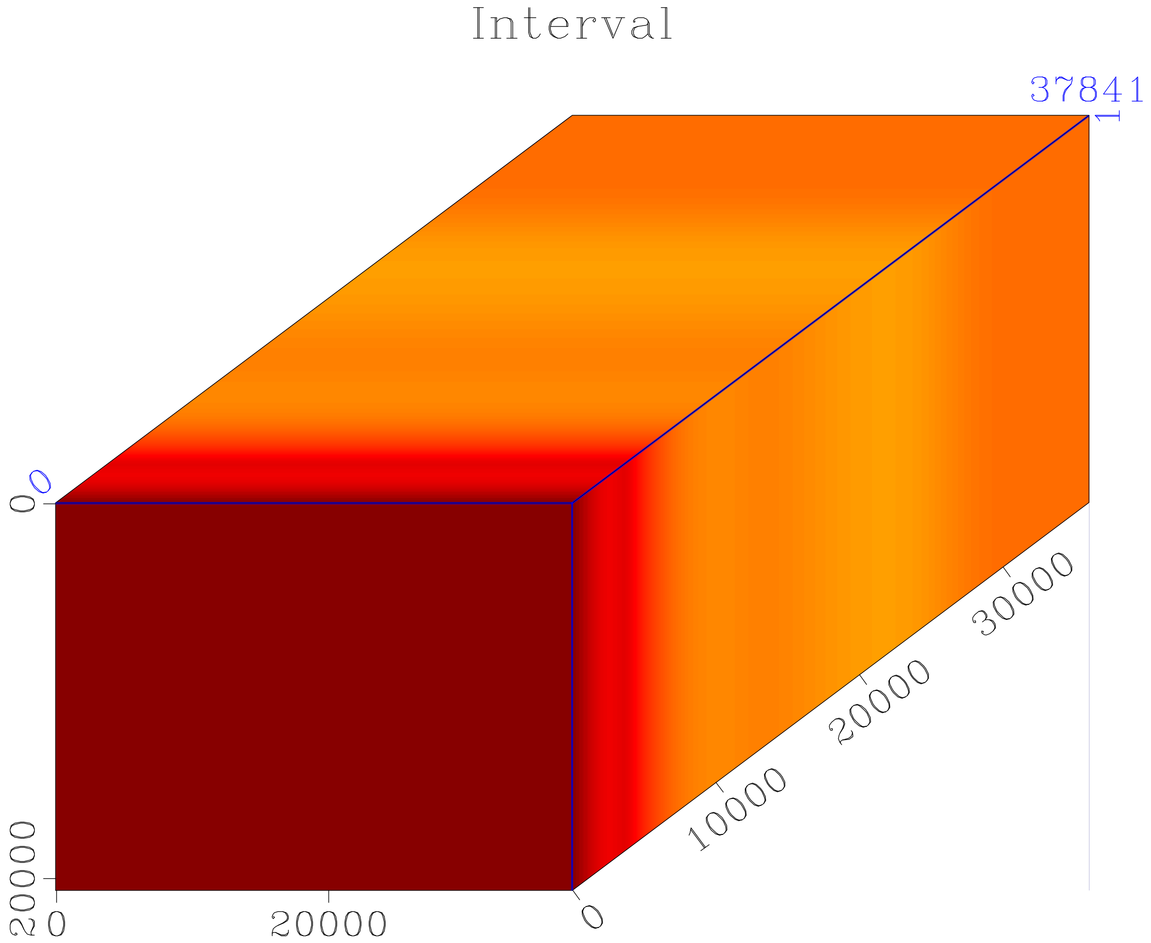


Figure 17: The interval velocity slowness for input to sfzomig3. zomigpaper/ slo

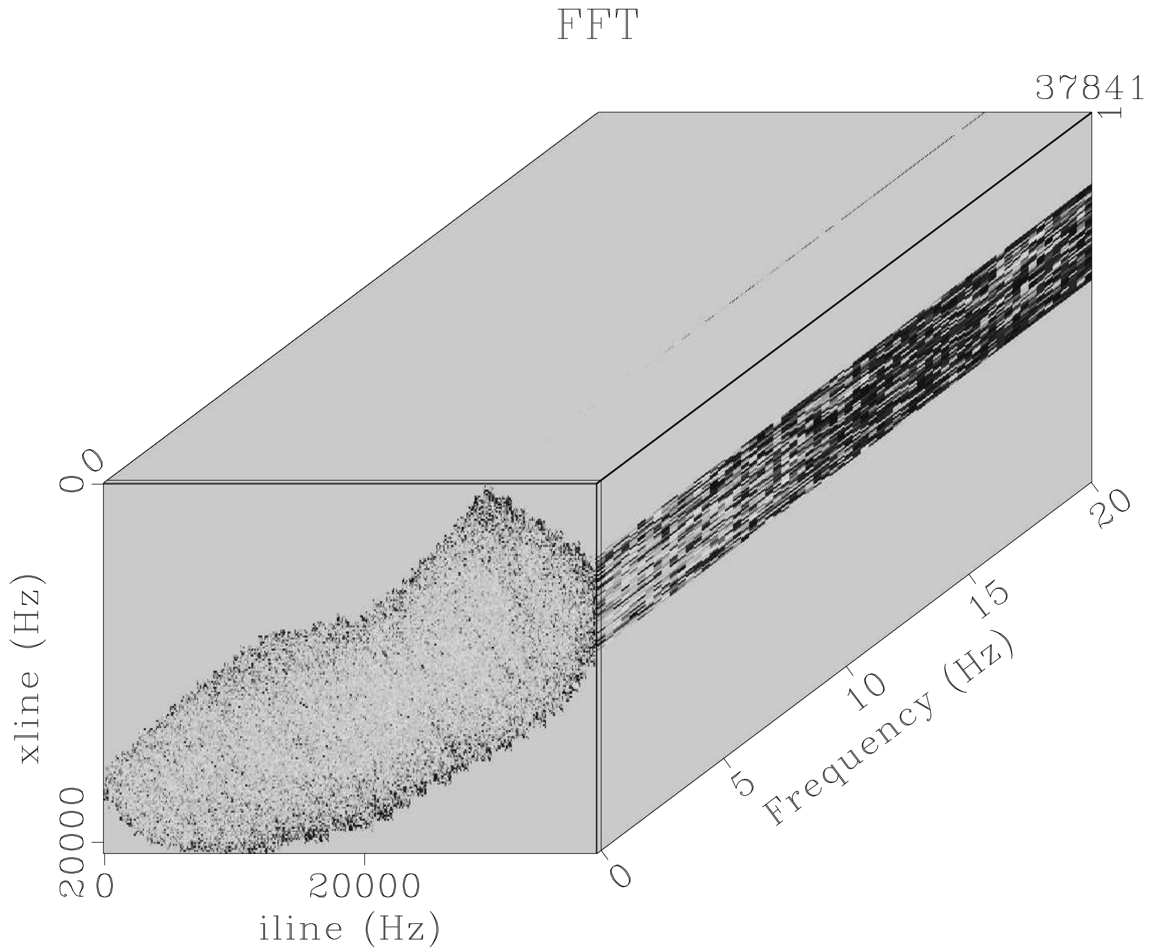


Figure 18: The real part of the frequency domain seismic data for input to sfzomig3. zomigpaper/ real

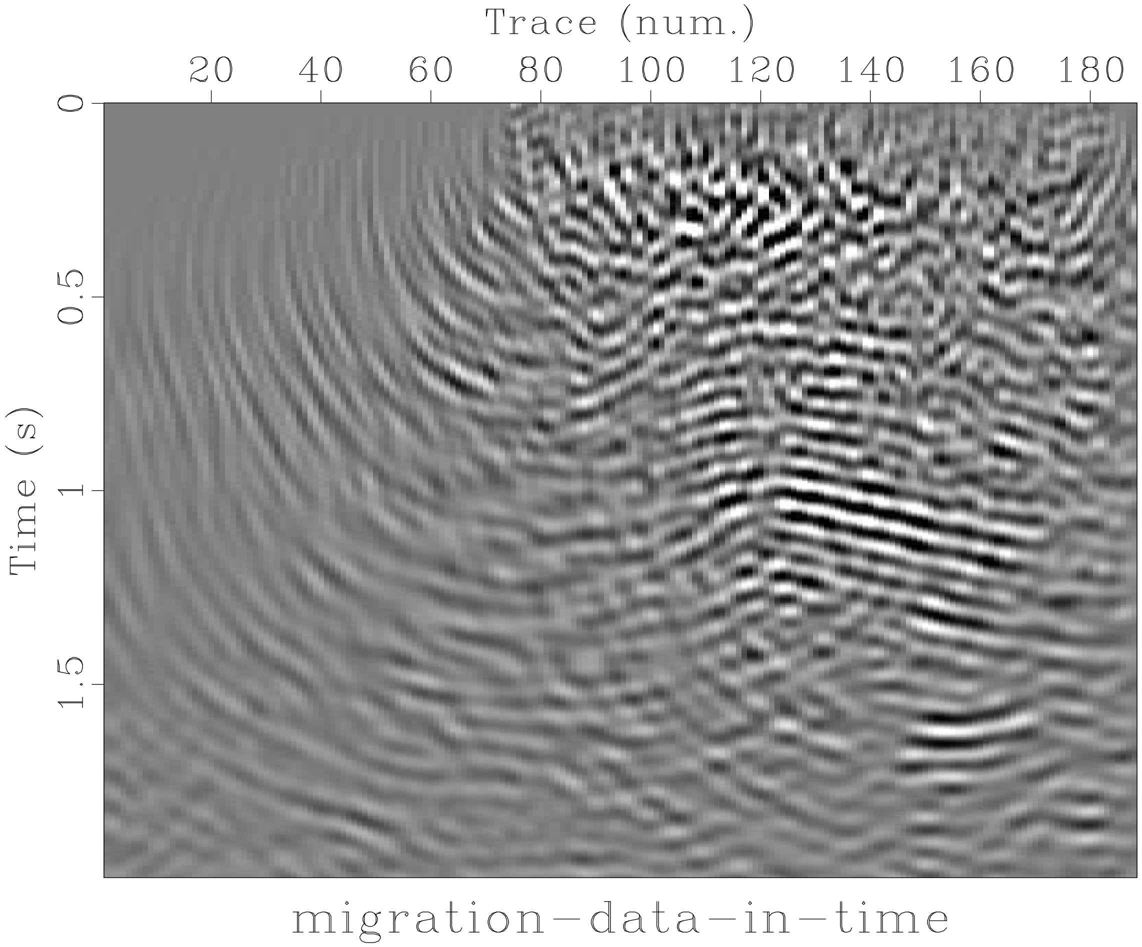


Figure 19: The migrated data. Wavefronts have not been muted from the zero portions of the stack volume. Migration was severely frequency limited to reduce computer time. zomigpaper/ mig-t

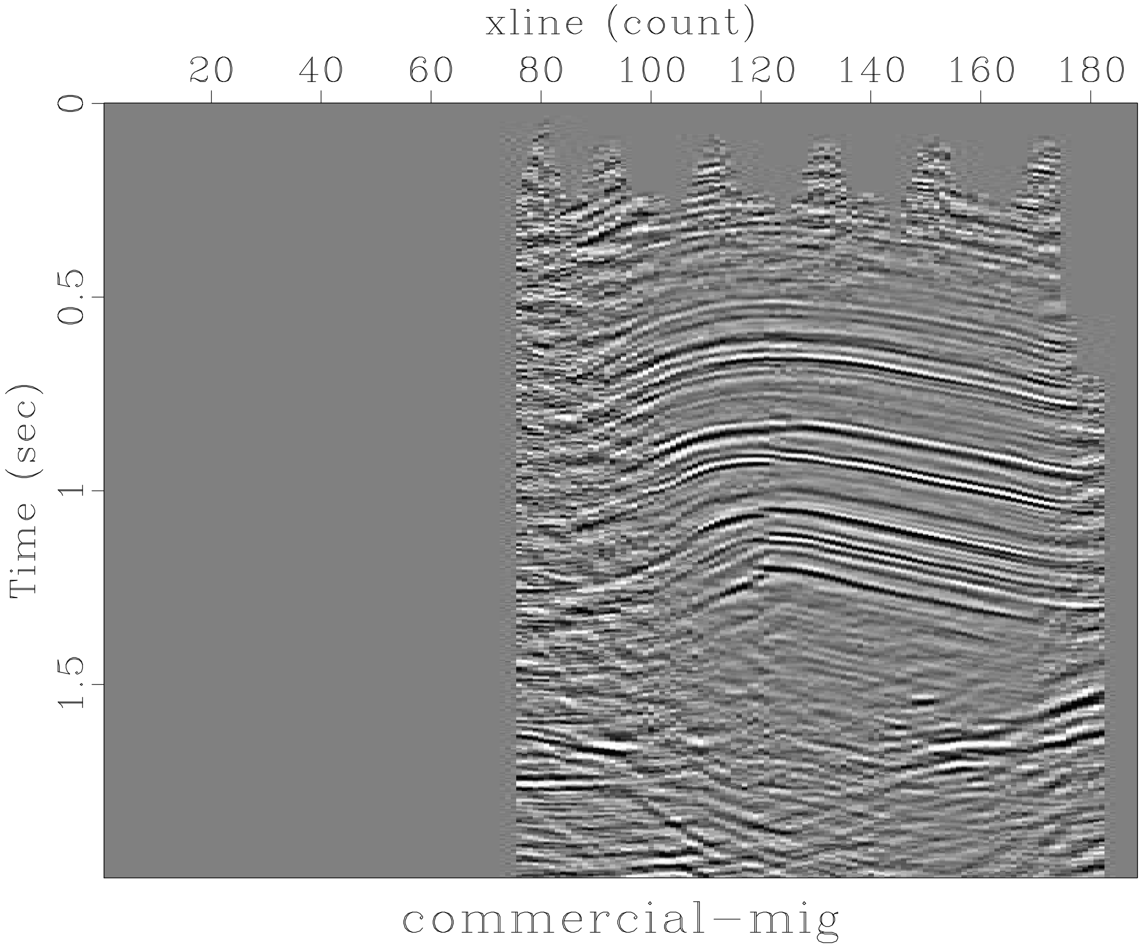


Figure 20: Commercial migration. zomigpaper/ filt-mig

2. A Madagascar tutorial
3. Prepare 3D data to test and validate new seismic algorithms or programs
4. Encourage others to write new programs, improve existing programs, and select better parameters to improve on the results show in this paper.

The processing applied in the scripts includes data loading, geometry plotting, spreading correction, surface consistent decon, scaling, refraction statics application, stacking velocity analysis, and post stack migration. The processing scripts also illustrate how seismic processors display and select processing parameters.

There are some missing processing stages, notably velocity estimation (interactive velocity picking, automatic velocity picking, tomographic velocity inversion, FWI), statics estimation (refraction statics, residual statics), noise attenuation, and prestack migration (Kirchhoff time or depth migration, shot migration, reverse time migration). Results from some processing stages should be improved, notable the post stack migration.

ACKNOWLEDGEMENTS

The Teapot Dome 3D survey is a land 3D dataset from Wyoming provided by the U.S. Department of Energy and the Rocky Mountain Oilfield Test Center (RMOTC). Data (including unprocessed prestack seismic data, preprocessed seismic data, final migrated image, processing reports, well logs, production history, and GIS data) may be downloaded from the Internet

Excell provided the original commercial processing. Excell's processing sequence and results are important references.

The scripts build on the accomplishments of the "SEG 3D Seismic Processing Working Workshop in Houston 2015 Land 3D". See `http://www.ahay.org/wiki/Houston_2015` for more information. I thank the participants for their contributions.

I thank the developers of the Madagascar open-source software package (Fomel et al., 2013).

REFERENCES

Dave Hutchinson and Brian Link, 1984, Surface consistency: A solution to the problem of deconvolving noisy seismic data, SEG Technical Program Expanded Abstracts 1984, 515-518

Peter W. Cary and Gary A. Lorentz, 1993, Four - component surface - consistent deconvolution, GEOPHYSICS 1993 58, 383-392

Fomel, S., P. Sava, I. Vlad, Y. Liu, and V. Bashkardin, 2013, Madagascar open-source software project: Journal of Open Research Software, 1, e8.