



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Seismic reflection imaging of underground cavities using open-source software

R. J. Mellors

January 6, 2012

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

# Seismic reflection imaging of underground cavities using open-source software

R. Mellors

## Table of Contents

<b>Seismic reflection imaging of underground cavities using open-source software .....</b>	<b>1</b>
<b>Introduction .....</b>	<b>2</b>
<b>Overview of the reflection seismic method .....</b>	<b>2</b>
<b>Seismic processing packages.....</b>	<b>4</b>
<b>Example of processing with Seismic Unix .....</b>	<b>6</b>
<b>Conclusions. ....</b>	<b>10</b>
<b>Acknowledgements .....</b>	<b>10</b>
<b>References .....</b>	<b>10</b>
<b>Appendix A. ....</b>	<b>12</b>

## Introduction

The Comprehensive Nuclear Test Ban Treaty (CTBT) includes provisions for an on-site inspection (OSI), which allows the use of specific techniques to detect underground anomalies including cavities and rubble zones. One permitted technique is active seismic surveys such as seismic refraction or reflection. The purpose of this report is to conduct some simple modeling to evaluate the potential use of seismic reflection in detecting cavities and to test the use of open-source software in modeling possible scenarios.

It should be noted that OSI inspections are conducted under specific constraints regarding duration and logistics. These constraints are likely to significantly impact active seismic surveying, as a seismic survey typically requires considerable equipment, effort, and expertise. For the purposes of this study, which is a first-order feasibility study, these issues will not be considered.

This report provides a brief description of the seismic reflection method along with some commonly used software packages. This is followed by an outline of a simple processing stream based on a synthetic model, along with results from a set of models representing underground cavities. A set of scripts used to generate the models are presented in an appendix. We do not consider detection of underground facilities in this work and the geologic setting used in these tests is an extremely simple one.

## Overview of the reflection seismic method

Reflection seismic surveys are an example of an active seismic study, in which seismic signals are deliberately produced by a source and then recorded by a set of sensors. In contrast to passive seismic monitoring, which attempts to measure signals from subsurface events of uncertain location and timing, the location and timing of the source used in active seismic are known exactly, which permits additional imaging capability. Reflection seismic surveys use reflected waves from underground interfaces to image the subsurface and are capable of imaging features as shallow as a few cm to 10's of km although the geometry of acquisition, source type, and instruments vary greatly with the target depths and settings. The method is the tool of choice for the hydrocarbon industry and hence an enormous amount of work has been conducted on seismic reflection acquisition and processing for that purpose.

Simple overviews of the technique are provided in many textbooks (e.g. *Kearey et al.*, [2002], *Sheriff and Geldart*, [1999]) while more advanced descriptions can be found in *Yilmaz* [1987; 2005] or *Ikelle and Amundsen* [2005]. During a survey, a series of sources (referred to as shots) are generated over a set of

locations. The signals produced by the shots are recorded at a set of sensors (geophones) arranged along a line (2D survey) or spaced over an area (3D survey). The geophones typically record only one component (vertical) of high-frequency (> 1 Hz) seismic energy. Each shot is recorded at the geophones and this is referred to as a shot gather with an individual geophone (or small group) recording a few seconds of data (a trace) after the shot time. These traces contain reflections from subsurface interfaces as well as seismic waves from refracted and direct waves. The goal of seismic processing is to isolate the reflections and add together (stack) all reflections from the same subsurface point. The result is an 'image' of the subsurface that represents changes in the seismic impedance below the surface. Typically, the vertical axis of these images are presented in two-way travel time (seconds or milliseconds) rather than units of distance due to uncertainties in the subsurface velocity structure.

Seismic processing, in the simplest form, selects traces from the shot gathers which represent the same reflection (a common midpoint gather, or CMP) and then applies a correction to adjust for the different path lengths (a normal moveout correction, or NMO). An NMO correction requires an estimate of the subsurface velocity structure, which is done by conducting a velocity analysis. After NMO correction, it is possible to stack the data, which improves the signal-to-noise of the signals. This represents the minimum processing. Alternate processing which uses inversion or tomographic approaches can be applied, but the procedure outlined above is most commonly used.

Common additional steps include statics corrections, which compensate for variations in sensor elevation and shallow velocity variations, and migration, which attempts to resolve non-planar subsurface structures and pronounced lateral velocity variations. A wide variety of gain adjustments, filtering and signal processing, such as deconvolution are usually applied as well. The range of operations depends on the type of data as well. Shallow seismic reflection, which can be roughly defined as imaging to depths of 10's of meters (or less) requires slightly different processing than deeper surveys, due to different source of noise such as air waves and high frequencies.

A key element controlling detection is the resolution of a survey, which defines the distance between two separable points. Vertical resolution is roughly  $\frac{1}{4}$  the wavelength while horizontal resolution depends on the wavelength and the depth [Yilmaz, 1987]. Wavelength is a function of frequency and sub-surface velocities. In general, frequency decreases with depth due to attenuation but also varies with the type of source. Imaging capability also depends strongly on the acquisition geometry, source, and instrumentation. In general, 3D seismic surveys will be much more effective than 2D surveys for a given target due to a more precise handling of the actual seismic raypaths, as 2D processing requires assumptions that are not valid in areas with laterally varying structure. However, 3D surveys require more effort to deploy and process.

Different target depths require different equipment. Deeper surveys, such as land-based petroleum industry surveys, which typically require imaging at depths greater than 1 km in layered sedimentary basins, require energetic sources (truck-mounted vibrators or explosives) to provide sufficient seismic energy at depth. Shallow reflection surveys may use a sledgehammer, small explosive source (e.g. shotgun shell), weight drop, or small vibration sources. In addition to the energy of the source, frequency content is important. Higher frequencies are capable of resolving smaller features but require appropriate sources (typically explosive) and suitable geophones. The geometry of the survey also varies with target depth. Shallow surveys may have a spacing of 1 or 2 meters between geophones while a typical petroleum survey may use 20 or 30 meters.

The geologic setting is critical, as some areas are more amenable to seismic imaging than others. Extreme topography or water (e.g. rivers, lakes and swamps) are obvious impediments, but other conditions such as glacial till over irregular bedrock or layered basaltic lava flows can be difficult to image through, as these conditions lead to high attenuation, especially of the higher frequencies needed to resolve small features at depth. Often, difficulties due to geology cannot be predicted before actually recording data.

Detection of underground cavities with seismic reflection has proven to be an extremely challenging problem. Most work has been done to detect underground mines (e.g. *Branham and Steeples, 1988*), which are typically at depths of less than 100 m and a few meters at most across. In general, these relatively small voids were 'visible' as sections with less coherent reflections and anomalous amplitudes. These voids, which were smaller than the seismic wavelength, did produce diffractions and in some cases (*Peterie et al., 2009*), these diffractions were useful in imaging voids.

## **Seismic processing packages**

Seismic processing packages are available at various levels, from simple packages designed for small shallow surveys to large petroleum industry scale packages. Here a brief overview of some commonly used ones, with emphasis on open-source packages with a long history.

### Open Source

#### *Seismic Unix*

<http://www.cwp.mines.edu/cwpcodes/>

Well supported and has been for ~ 20 years. Continued support seems likely. Standard package for many. Includes forward modeling capability. Disadvantages: complex, requires linux (mostly – windows ports are available but may not be up to date). Also has several GUI developed, notably GEBR (supported by Petrobras)

<http://www.gebrproject.com/>

[http://reproducibility.org/wiki/Other\\_open-source\\_geophysical\\_packages](http://reproducibility.org/wiki/Other_open-source_geophysical_packages)

*Madagascar*

Similar to seismic unix but written in modular form.

*SIOSEIS*

Command-driven seismic processing with display capabilities. Easier to use than seismic unix but designed for marine reflection seismology.

<http://sioseis.ucsd.edu/>

*JRG*

Resource Geology Seismic Processing System for Java (JRG)

Limited (no decon, migration, or muting) but user friendly. Java -based so runs on any platform.

<http://crack.seismo.unr.edu/jrg/>

*Proprietary packages*

*SPW*

PC based seismic reflection. \$5000 for IRIS members

<http://www.passcal.nmt.edu/content/seismic-processing-workshop-spw-iris-members>

Lab manual available. Easier to use than Seismic Unix and widely used for classroom teaching.

*WinSeis, WinSeisLite*           \$1300

Kansas Geological Survey

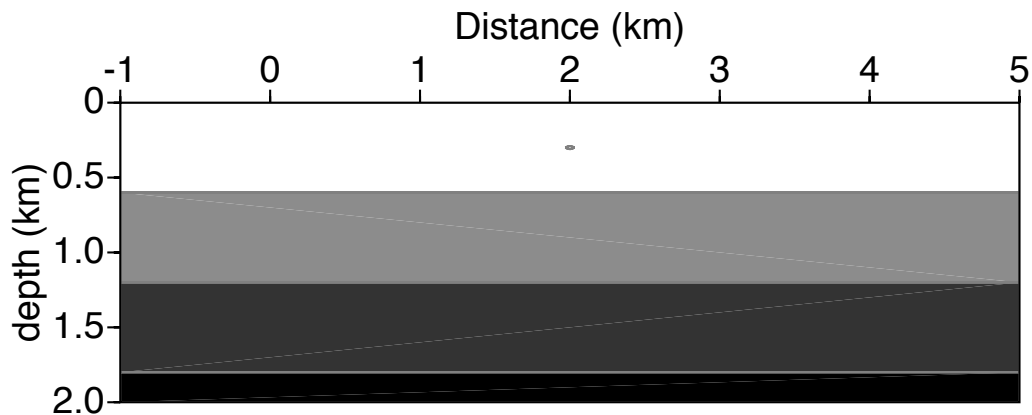
Designed for shallow (< 100 m) seismic

*Promax*

Processing package capable of handling large datasets and advanced processing typical of petroleum seismic surveys.

## Example of processing with Seismic Unix

In this section a simple example processing flow is shown using a synthetic data set constructed in the first two steps. Processing and scripts are adapted from example presented in *Forel et al (2005)*. The primary goal is to demonstrate the feasibility and scope of effort using a simplified model and processing scheme. It is possible to enhance the complexity of both the model and the processing by including more complex geology, attenuation and techniques such as migration. Various types of synthetics (e.g. finite-difference) capable of handling complex settings can also be tested.



### NTS Model - 60 m (diameter) cavity z=290-310

**Figure 1.** Initial model used to investigate detection of an underground anomaly. Shading is proportional to velocity. Rubble-filled cavity is 300 m below the 2 km mark.

The model is 6 km long, extends to a depth of 2.0 km, and contains four flat-lying layers (Table 1; Figure 1). The anomaly, which consists of a low-density zone intended to mimic a rubble-filled cavity, is situated in the center of the model at a depth of 300 m and a diameter of 60 m. The anomaly lies completely within layer 1 (low-velocity alluvium). The model is constructed using triangular elements and the response simulated using Gaussian beam synthetics, which are fast and well-suited for moderately complex geometries.

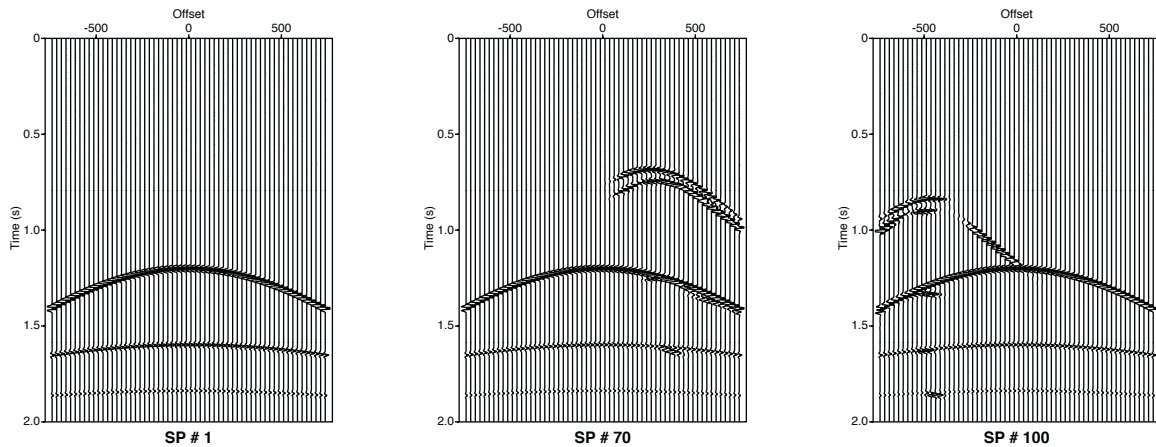
*Table 1. Model 1*

Layer	Vp (m/s)	density	Depth (top)
Alluvium	1000	2.4	0
Tuff (Tertiary)	3000	2.8	600
Paleozoic	4000	2.8	1200
Granite	5000	2.8	1800

The initial acquisition uses a split-spread geometry with 60 geophones and 40 shot points along a line. Spacing between geophones is 25 m. Once the model is constructed, acquisition is simulated and a set of synthetic shot gathers generated.



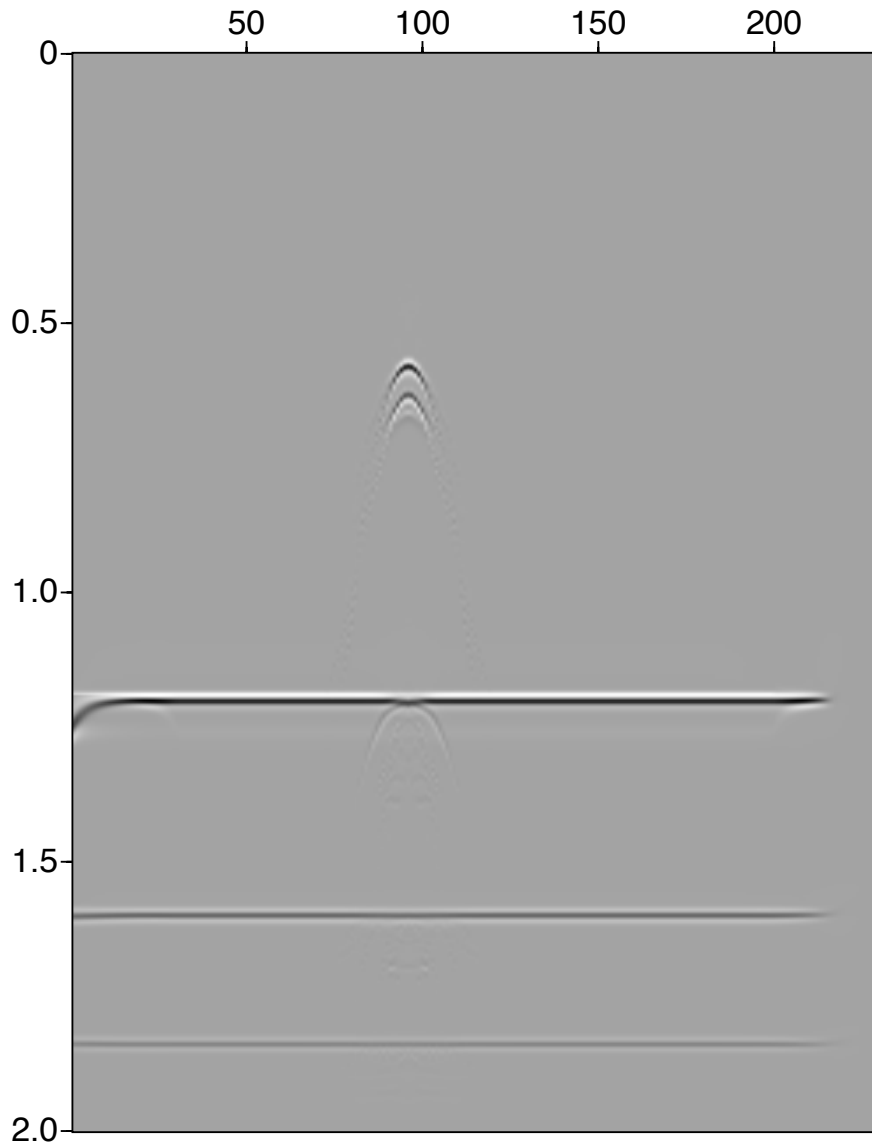
Figure 2 shows example of synthetic shot gathers from the model with the acquisition parameters. This mimics what might be done during an actual survey, although actual data would contain more noise and would be degraded at the higher frequencies relative to these synthetics due to source characteristics and attenuation.



**Figure 2.** Example shot gathers using the model shown in Figure 1. Shot gather #1 (left) shows reflections from the three main interfaces but is sufficiently far from the anomaly that no reflections or diffractions are recorded. Shot #70, which is located left of the anomaly, shows a clear reflection/diffraction with the apex at the anomaly location. Shot #100 also shows the anomaly. Signal anomalies below the expected position may be multiples, which are signals that reflect from more than one layer.

The next processing step re-organizes the shot gathers into common midpoint (CMP) gathers (sometimes referred to as common depth point [CDP] gathers). With known geometry this step is straightforward, both for synthetics and actual data. For real data, in which the instrument and source location deviate from ideal locations, a bin size is chosen and all reflections within that bin are summed. Actual data would also require gain adjustment and filtering (or deconvolution) at this stage to improve reflection signal quality. Once the CMP gathers are created, velocity analysis to estimate stacking velocities is conducted. In this case, velocities are selected manually using constant velocity stacks. Alternate approaches use measures of the similarities between traces (e.g. semblance) are also possible. For low-noise synthetics with a flat-layer model, most approaches will work well. Velocity analysis with actual data is usually more difficult. Once a set of interval velocities are determined, normal moveout correction is applied along with a mute, which removed parts of each seismogram that has been excessively affected by the NMO operation. NMO (and mute) is followed by stacking (Figure 3). The final result shows a clear detection of the anomaly. Horizontal resolution is good and on the order of 10's of meters. Estimation of depth depends on the velocity model, which will have errors.

A static correction for sensor elevation or shallow velocity variations is not needed in this case but would be necessary in actual data (and might require several iterations in specific settings). Along with the reflection processing, it may be useful to conduct refraction processing to estimate near-surface velocities as these are useful in estimating static corrections as well as defining absolute velocities (rather than the stacking velocities estimated during the reflection processing, which differ from absolute velocities).



**Figure 3.** Stacked section after NMO correction. Section is unmigrated and position of anomaly is marked by diffraction hyperbola with apex at location. Note primary multiple at twice the two-way travel time.

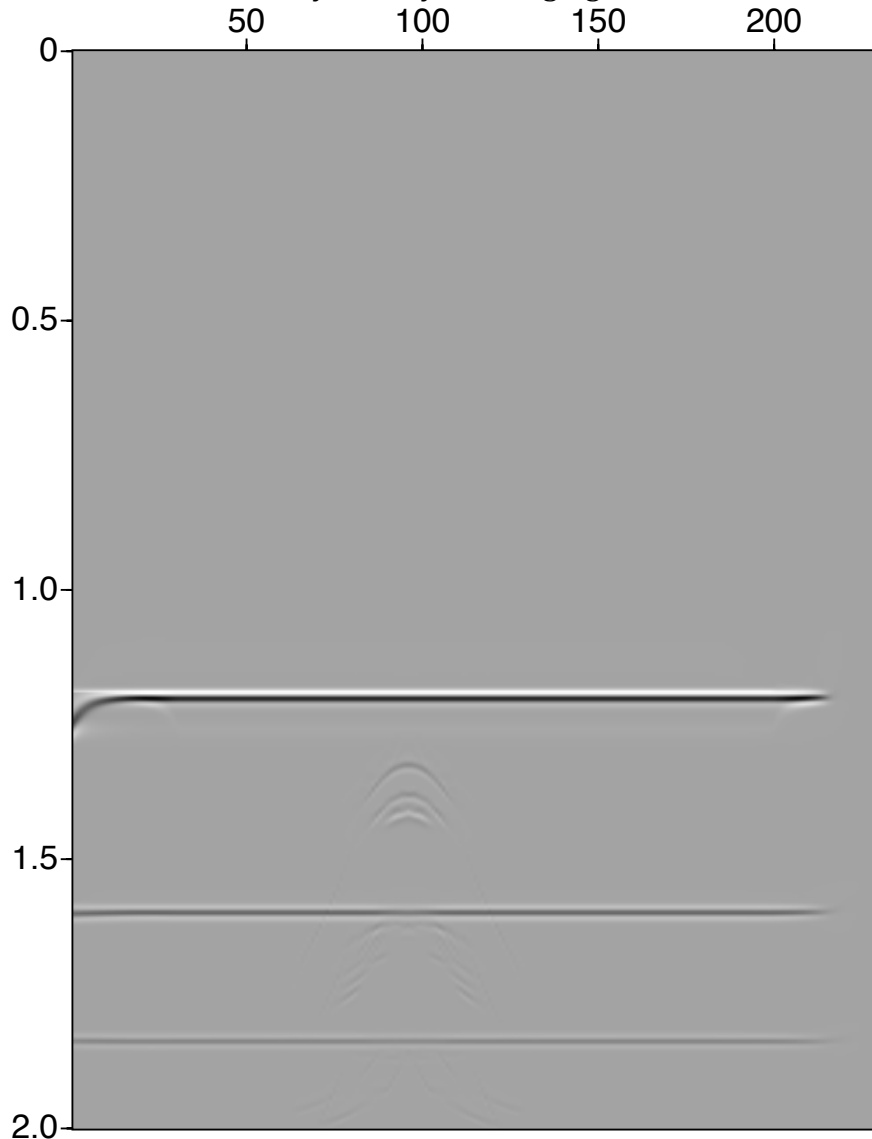
It is important to note that these results are much better than would be expected to occur with real data. Variations in geology along with differences in the source as well as other sources of noise (ground roll, air wave, refractions, etc) will

greatly degrade the image. A shallow, low-velocity layer would likely have significant impact on the image.

*Table 2. Models*

Model	Cavity diameter	Depth	Chimney diameter, height	density	Velocity	Q
1	60	290-310	NA			No
2	60	900-920	NA			No

Another model was run (Table 2) to evaluate the effect of depth on the imaging capability. An increase of depth to 900 m (and below the first major interface) significantly degraded the reflections (Figure 4). While still faintly visible in the synthetic model, this would likely be very challenging in real data.



**Figure 4.** Stacked section for a cavity at depth of 900 m.

## Conclusions.

Freely-available open-source software is capable of generating synthetic models and processing reflection seismic images with relatively little effort. In this set of tests we used Seismic Unix. While simple models were straightforward, more complex models were difficult to build and synthetics sometimes failed, often for unclear reasons. In general, this is likely to difficulties resulting from sparse documentation.

An option would be to test other methods (such as finite-difference wave propagation) to generate the models. A disadvantage is the greatly increased computational time required to generate the models.

A significant disadvantage is that the synthetic models are much simpler than real data and hence likely greatly over-estimate detection capabilities. Future work would include more realistic models, inclusion of noise, and possibly 3D models and processing.

## Acknowledgements

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

## References

Branham K.L., and Steeples, D.W., 1988, Cavity detection using high-resolution seismic reflection methods, *Min. Engin.*, 40, 115-119.

Forel, D. T. Benz, and W. D. Pennington, *Seismic data processing with Seismic Unix*, 2005, Course Notes Series No. 12, L. M. Gochioco, editor, Soc. Exploration Geophysics, Tulsa, OK, USA.

Hatton, L., M. H. Worthington, and J. Makin, *Seismic data processing: Theory and practice*, 1986, Blackwell Publishing, pp. 177.

Kearey, P., Brooks, M., and I. Hill, 2002, *An introduction to geophysical exploration*, Blackwell Publishing, pp. 262.

Miller, R.D., and Steeples, D.W., 1991, Detecting voids in a 0.6-m coal seam, 7 m deep, using seismic reflection, *Geoexploration*, 28, 109-119.

Peterie, S. L., and R. Miller, 2009, Diffraction versus reflection processing for shallow void detection, *SEG annual meeting*, p. 1421.

Sheriff, R. E., and L. P. Gerhart, exploration seismology, second edition, 1995, Cambridge University Press, pp. 592.

Steeple, D. and R. Miller, 1998, Avoiding pitfalls in shallow seismic reflection surveys, *Geophysics*, 63, (4) , 1213-1224.

Yilmaz. O. Seismic data processing, 1987, Soc. Exploration Geophysics, pp. 526.

Ikelle, L. and L. Amundsen, Introduction to petroleum seismology, Society of Exploration Geophysics, pp. 679.

## Appendix A.

### Scripts

Adapted from templates in *Forel et al.*, (2007)

A) STEP1\_make\_nts\_model\_V1\_cavity.csh

*Creates a 2D 4 layer model with a cavity using triangular elements.*

B) STEP2\_acquire\_nts.sh

*Simulates a set of shot gathers over model using Gaussian beam synthetics*

C) STEP3\_sort2cmp.sh

*Sorts shot gathers into CMP gathers*

D) STEP4\_iva.sh

*Interactive velocity analysis*

E) STEP4\_ALT\_do\_sembalance.sh

*Alternate velocity analysis*

F) STEP5\_nmo\_nts.sh

*(requires manual editing step from velocity analysis)*

*Apply normal moveout correction derived from velocity analysis*

G) STEP6\_do\_stack.sh

*Stack NMO*

Scripts to display subsets of the data

showcmp.sh Create figure of 1 CMP

shownmo.sh Create figure of NMO correction

showshot.sh Create figure of shot gather

## STEP1\_make\_nts\_model\_V1\_cavity.csh

```
#
# NTS
#
set modfile = model_nts.dat
set psfile = model_nts.eps
set XMIN = "0"
set XMAX = "6"
\rm -r $psfile
#
# no spaces between commas
# 1 alluvium          1000 m/s sloth = 1.00 top = 0
# 2 top cavity       700 m/s sloth = 2.00 top = 2900 60 m wide
# 3 bottom cavity 700 m/s sloth = 2.00 bottom = 310 60 m wide
# 4 tuff (Tertiary) 3000 m/s sloth = 0.11 top = 600
# 5 Paleozoic       4000 m/s sloth = 0.06 top = 1200
# 6 granite         5000 m/s sloth = 0.04 top = 1800
#
# 1.000000
# 0.111111
# 0.062500
# 0.040000
#
trimodel xmin=$XMIN xmax=$XMAX zmin=0 zmax=2 \
1 xedge=$XMIN,$XMAX \
  zedge=0,0 \
  sedge=0,0 \
2 xedge=1.97,2.00,2.03 \
  zedge=0.30,0.29,0.30 \
  sedge=0,0,0 \
3 xedge=1.97,2.00,2.03 \
  zedge=0.30,0.31,0.30 \
  sedge=0,0,0 \
4 xedge=$XMIN,$XMAX \
  zedge=0.6,0.6 \
  sedge=0,0 \
5 xedge=$XMIN,$XMAX \
  zedge=1.2,1.2 \
  sedge=0,0 \
6 xedge=$XMIN,$XMAX \
  zedge=1.8,1.8 \
  sedge=0,0 \
7 xedge=$XMIN,$XMAX \
  zedge=2.0,2.0 \
  sedge=0,0 \
kedge=1,2,3,4,5,6,7 \
maxangle=10 \
densfill=1.0,0.1,2.4 \
densfill=2.0,0.3,1.4 \
densfill=1.0,0.8,2.4 \
densfill=1.0,1.4,2.8 \
densfill=1.0,1.7,2.8 \
densfill=1.0,2.1,2.8 \
sfill=1.0,0.1,0,0,1.00,0,0 \
sfill=2.0,0.3,0,0,2.00,0,0 \
```

```

sfill=1.0,0.8,0,0,0.11,0,0 \
sfill=1.0,1.4,0,0,0.06,0,0 \
sfill=1.0,1.7,0,0,0.04,0,0 > $modfile
#
spsplot < $modfile > $psfile \
    gedge=0.5 gtri=2.0 gmin=0 gmax=1 sgmin=0.0 sgmax=0.2 \
    title="NTS Model - 60 m (diameter) cavity z=290-310 " \
    labelz="depth (km)" labelx="Distance (km)" \
    dxnum=1.0 dznum=0.5 wbox=6 hbox=2

```

## **STEP2\_acquire\_nts.sh**

```

#!/bin/sh
num=4
nangle=201
fangle=-65
langle=65
nt=501
dt=0.004

inmodel=model_nts.dat
outseis=seis_nts.su

rm -f survey_nts.txt
survey=survey_nts.txt
#
# i loop 40 source
# j 60 geophone (split-spread)
# k loop layers 2 through 4
# do not do layers 1 and 5
#
#
echo "--Begin looping over triseis."

i=0

\rm temp*
# number of shots
while [ "$i" -ne "200" ]
do
    fs=`bc -l <<-END
    $i * 0.025
    END`
    sx=`bc -l <<-END
    $i * 25
    END`
    fldr=`bc -l <<-END
    $i + 1
    END`

    j=0

    while [ "$j" -ne "60" ]
    do
        fg=`bc -l <<-END

```



```

    $i * 0.025 + $j * 0.025
    END`
    gx=`bc -l <<-END
    $i * 25 + $j * 25 - 738.5
    END`
    offset=`bc -l <<-END
    $j * 25 - 738
    END`
    tracl=`bc -l <<-END
    $i * 60 + $j + 1
    END`
    tracf=`bc -l <<-END
    $j + 1
    END`

    echo "$x = $sx GX=$gx fldr=$fldr Offet=$offset
    tracl=$tracl fs=$fs fg=$fg"
    echo "$x = $sx GX=$gx fldr=$fldr Offet=$offset
    tracl=$tracl fs=$fs fg=$fg" >> $survey

    k=2
    while [ "$k" -ne "8" ]
    do
        triseis < $inmodel xs=0,4.95 xg=-0.7385,5.688
    zs=0,0 zg=0,0 \
        nangle=$nangle fangle=$fangle langle=$langle \
        kreflect=$k krecord=1 fpeak=40 lscale=0.5 \
        ns=1 fs=$fs ng=1 fg=$fg nt=$nt dt=$dt |
        suaddhead nt=$nt |
        sushw
    key=dt,tracl,tracl,fldr,tracl,tracl,offset,sx,gx \

        a=4000,$tracl,$tracl,$fldr,$tracl,1,$offset,$sx,$gx >> temp$k

        k=`expr $k + 1`

    done
    j=`expr $j + 1`

done
i=`expr $i + 1`

done
echo "--End looping over triseis."

# change this for different number of layers!!!!!!
# sum contents of temp files
susum temp2 temp3 > tempa
susum tempa temp4 > tempb
susum tempb temp5 > tempc
susum tempc temp6 > tempd
susum tempd temp7 > $outseis

# remove temp files

```

```
#rm -r temp*

\mv $outseis ./data
echo " --Output file ** $outseis **"
```

### STEP3\_sort2cmp.sh

```
#!/bin/sh
suchw < ./data/seis_nts.su key1=cdp key2=gx key3=sx a=1525 b=1 c=1
d=50 | susort > cmp_nts.su cdp offset
\mv cmp_nts.su ./data
exit
```

### STEP4\_iva.sh

```
#!/bin/sh
#cmp1=60
cmp1=60 cmp2=80 cmp3=100
cmp4=120 cmp5=140 cmp6=160
cmp7=180 cmp8=200 cmp9=220
cmp10=240 cmp11=260 cmp12=280
cmp13=300 cmp14=320 cmp15=340
cmp16=360 cmp17=380 cmp18=400

numCMPs=18

indata=./data/cmp_nts.su
outpicks=vpick.txt

myperc=98 # percentage value for plot
plottype=0 # = = wiggle plot 1 = image plot

nvs=100 # number of velocities
dvs=50 #velocity intervals
fvs=100 # first velocity

fc=100 # first CVS velocity
lc=5100 # last CVS velocity
nc=10 # number of CVS velocities
XX=11 # ODD number of CMPS to stack into central CVS

# semblance velocities
# Vel = fvs + (( nvs-1) * dvs ) = lvs
# 5000 = 500 + (( 99-1) * 45 )
# 3900 = 1200 + (( 100-1) * 27 )

lvs=`bc -l <<-END
$fvs + (( $nvs - 1 ) * $dvs )
END`

echo ""
echo "interactive velocity analysis"
echo ""
```

```

rm -f panel.* picks.* par.* tmp*

# make empty files
> $outpicks
> par.cmp

nt=`sugethw ns < $indata | sed 1q | sed 's/.*ns=//'\`
dt=`sugethw dt < $indata | sed 1q | sed 's/.*dt=//'\`
ft=`sugethw delrt < $indata | sed 1q | sed 's/.*delrt=//'\`

dt=`bc -l <<-END
scale=6
$dt / 1000000
END`
#
if [ $ft -ne 0 ]; then
    tstart=`bc -l <<-END
    scale=6
    $ft / 1000
    END`
else
    tstart=0.0
fi

repick=1

echo $nt $dt $ft

# iva loop
i=1
while [ $i -le $numCMPs ]
do
    eval picknow=\$cmp$i
    #
    if [ $repick -eq 1 ] ; then
        echo ""
        echo "preparing CMP $i of $numCMPs for picking"
        echo "location is CMP $picknow"
    fi

# plot CMP

    suwind < $indata key=cdp min=$picknow max=$picknow >
panel.$picknow
    if [ $repick -eq 1 ] ; then
        if [ $plottype -eq 0 ] ; then
            suxwigb < panel.$picknow xbox=634 ybox=10 wbox=300
hbox=450 \
                title="CMP gather $picknow" \
                labell1=" Time (s) " label2="offset (m) "
key=offset \
                perc=$myperc verbose=0 &
        else
            suximage < panel.$picknow xbox=634 ybox=10 wbox=300
hbox=450 \

```

```

                                title="CMP gather $picknow" \
                                label1=" Time (s) " label2="offset (m) "
key=offset \
                                perc=$myperc verbose=0 &
                                fi
                                else
                                if [ $plotype -eq 0 ] ; then
                                suxwigb < panel.$picknow xbox=946 ybox=10 wbox=300
hbox=450 \
                                title="CMP gather $picknow" \
                                label1=" Time (s) " label2="offset (m) "
key=offset \
                                perc=$myperc verbose=0 &
                                else
                                suximage < panel.$picknow xbox=946 ybox=10 wbox=300
hbox=450 \
                                title="CMP gather $picknow" \
                                label1=" Time (s) " label2="offset (m) "
key=offset \
                                perc=$myperc verbose=0 &
                                fi
                                fi

# plot constant velocity stack

if [ $repick -eq 1 ] ; then
X=`expr $XX - 1`

k1=`expr $picknow - $X / 2`
k2=`expr $picknow + $X / 2`

suwind < $indata key=cdp min=$k1 max=$k2 > tmp0

dc=`bc -l <<-END
( $lc - $fc ) / ( $nc - 1 )
END`

m=`bc -l <<-END
( $lc - $fc ) / ( ( $nc - 1 ) * $XX )
END`

j=1
while [ $j -le $nc ]
do
vel=`bc -l <<-END
$fc + $dc * ( $j - 1 )
END`
#
echo " vel = $vel"

sunmo < tmp0 vnmo=$vel verbose=1 | sustack >> tmp1

j=`expr $j + 1`
done

loV=`bc -l <<-END

```

```

    $fc - ( $X / 2 ) * $m
    END`

    echo $loV
    suximage < tmp1 xbox=322 ybox=10 wbox=300 hbox=450 \
        title="CMP $picknow constant velocity stacks" \
        label1="Time (s) " label2="Velocity (m/s)" \
        f2=$loV d2=$m verbose=0 \
        perc=$myperc n2tic=5 cmap=rgb0 &
    fi
# instructions
    echo "start CVS CMP = $k1 End CVS CMP = $k2"
    echo "type S to pick (when mouse pointer at proper place)"
    echo "type Q to stop"

## repick: 1=false, 0=true
    if [ $repick -eq 0 ] ; then

        wc -l tmp7 | sed 's/^ *\(.*\)/\1/' > tmp4
        sed 's/tmp7//' < tmp4 > tmp5
        npair=`sort < tmp5`

# plot semblance
        suvelan < panel.$picknow nv=$nvs dv=$dvs fv=$fvs |
        suximage xbox=10 ybox=10 wbox=300 hbox=450 perc=99 \
            units="semblance" f2=$fvs d2=$dvs n2tic=5 \
            title="Semblance plot CMP $picknow" cmap=hsv2 \
            label1=" Time (s) " label2="Velocity (m/s) "\
            legend=1 units=Semblance verbose=0 gridcolor=black
        \
            grid1=solid grid2=solid mpicks=picks.$picknow \
            curve=tmp7 npair=$npair curvecolor=white

    else
        suvelan < panel.$picknow nv=$nvs dv=$dvs fv=$fvs |
        suximage xbox=10 ybox=10 wbox=300 hbox=450 perc=99 \
            units="semblance" f2=$fvs d2=$dvs n2tic=5 \
            title="Semblance plot CMP $picknow" cmap=hsv2 \
            label1=" Time (s) " label2="Velocity (m/s) "\
            legend=1 units=Semblance verbose=0 gridcolor=black
        \
            grid1=solid grid2=solid mpicks=picks.$picknow

    fi

# end first pair of plots
    sort < picks.$picknow -n |
        mkparfile string1=tnmo string2=vnmo > par.$i

    echo "cdp=$picknow" >> tmp2
    cat par.$i >> tmp2

    sunmo < panel.$picknow par=tmp2 verbose=0 > tmp8

    if [ $plottype -eq 0 ] ; then

```

```

        suxwigb < tmp8 xbox=634 ybox=10 wbox=300 hbox=450 \
            title="CMP $picknow after NMO" \
            label1=" Time (s) " label2="offset (m) "
key=offset \
    perc=$myperc verbose=0 &
    else
        suximage < tmp8 xbox=634 ybox=10 wbox=300 hbox=450 \
            title="CMP $picknow" after NMO \
            label1=" Time (s) " label2="offset (m) "
key=offset \
    perc=$myperc verbose=0 &
    fi

    j=1
    while [ $j -le 8 ]
    do
        sustack < tmp8 >> tmp3
        j=`expr $j + 1`
    done

    suxwigb < tmp3 xbox=946 ybox=10 wbox=300 hbox=450 \
        title="CMP gather $picknow repeat stack trace" \
        label1=" Time (s) " d2num=50 key=cdp \
        perc=$myperc verbose=0 &

    sed < par.$i '
    s/tnmo/xin/
    s/vnmo/yin/
#          ' > par.uni.$i

##velocity profile
#
    unisam nout=$nt fxout=$tstart dxout=$dt \
        par=par.uni.$i method=mono |
    xgraph n=$nt nplot=1 d1=$dt f1=$tstart x2beg=$fvs x2end=$lvs \
        title="CMP $picknow Stacking Velocity Function" \
        -geometry 300x450+10+10 -bg white style=seismic \
        grid1=solid grid2=solid linecolor=2 marksize=1 mark=0 \
        titleColor=black axesColor=blue &
#
    echo " "
    cat picks.$picknow
    echo "Picks OK? (y/n) " > /dev/tty
#
    rm tmp*

#n mean re-loop otherwise next CMP

    case $response in
        n*)
            i=$i
            echo ""
            echo "repick CMP $picknow. Overlay previous picks."
            repick=0
            cp picks.$picknow

```

```

                ;;
*)
    echo "$picknow $i" >> par.cmp
    i=`expr $i + 1`
    repick=1
    echo "closing"
    zap xwigb > tmp6
    zap ximage > tmp6
    zap xgraph > tmp6
                ;;
    esac
done

mkparfile < par.cmp string1=cdp string2=# > par.0

i=0
while [ $i -le $numCMPs ]
do
    sed < par.$i 's/$/ \\ /g' >> $outpicks
    i=`expr $i + 1`
done
##
echo ""
echo "The output file of t-v pairs is "$outpicks
exit

```

#### **STEP4 ALT do semblance.sh**

```

#!/bin/sh
indata=cmp_nts_200.su

nv=100
dv=25
fv=500

suvelan < $indata nv=$nv dv=$dv fv=$fv smute=1.5 |
suxcontour bclip=0.5 wclip=0.0 f2=$fv d2=$dv \
    label1=" Time (s)" label2="Velocity" \
    title = "Semblance: $indata" \
    windowtitle= "Semblance: $indata" \
    grid1=solid grid2=solid cmap=default |
    nc=12 nlabelc=0 &

exit

```

#### **STEP5 nmo\_nts.sh**

```

#!/bin/sh

set -x
indata=./data/cmp_nts.su

```

```

outdata=./data/nmo_nts.su
sunmo < $indata \
cdp=1,458 \
tnmo=1.18734,1.58311,1.84697 \
vnmo=1007.5,1750,2410 \
tnmo=1.18734,1.58311,1.84697 \
vnmo=1007.5,1750,2410 > $outdata

exit

```

### **STEP6\_do\_stack.sh**

```
sustack < ./data/nmo_nts.su > ./data/stack_nts.su
```

### **showshot.sh**

```

#!/bin/sh
#showshot model shot
#
suwind < ./data/seis_nts.su key=fldr min=$1 max=$1 > shot$1.su

suxwigb < shot$1.su title="SP # $1 " key=offset \
    label1=" Time (s)" label2="Offset" \
    x2beg=-750 x2end=750 perc=99 &

supswigp < shot$1.su title="SP # $1" key=offset \
    label1=" Time (s)" label2="Offset" \
    x2beg=-750 x2end=750 perc=99 > shot$1.eps &
exit

```

### **showcmp.sh**

```

#!/bin/sh
#showcmp model cmp
#
echo $1
if [ $1 -le 458 ]; then
    if [ $1 -ge 1 ]; then
        suwind < ./data/cmp_nts.su key=cdp min=$1 max=$1 >
        cmp_nts_$1.su

        suxwigb < cmp_nts_$1.su title="SP # $1 " key=offset \
            label1=" Time (s)" label2="Offset" \
            x2beg=-1500 x2end=1500 perc=99 &

        supswigp < cmp_nts_$1.su title="SP # $1" key=offset \
            label1=" Time (s)" label2="Offset" \
            x2beg=-1500 x2end=1500 perc=99 > cmp_nts$1.eps &
        exit
    fi
fi

```



```
        fi
fi

echo usage: showcmp.sh [model number] [CMO between 1 and 458]
exit
```

### **shownmo.sh**

```
#!/bin/sh
#showcmp model nmo
#
echo $1
if [ $1 -le 458 ]; then
    if [ $1 -ge 1 ]; then
        suwind < ./data/nmo_nts.su key=cdp min=$1 max=$1 >
nmo_nts_$1.su

        suxwigb < nmo_nts_$1.su title="SP # $1 " key=offset \
            labell1=" Time (s)" label2="Offset" \
            x2beg=-1500 x2end=1500 perc=99 &

        supswigp < nmo_nts_$1.su title="SP # $1" key=offset \
            labell1=" Time (s)" label2="Offset" \
            x2beg=-1500 x2end=1500 perc=99 > nmo_nts$1.eps &
        exit
    fi
fi

echo usage: shownmo.sh [nmo between 1 and 458]
exit
```