University of Utah
Modeling and Tomography Development
Project

1993 Midyear Report

W. Cai, T. Nemeth, K. Olsen, F. Qin,
G. Schuster, Y. Sun, J. Xu, and C. Zhou

September 17, 1993
Contents

Preface iii

I Tomography Methods 1
1 Elastic WTW 3
2 Tomostatics 21
3 Wide Angle Crosswell Imaging 33

II Migration 45
4 Quasi-Monte Carlo Migration 47
5 Crosswell Seismic Migration 77
6 Conjugate Gradient Migration 107
7 Migration Velocity Analysis 121

III Data Processing 129
8 Data Conversion of 3-D to 2-D 131

IV FORTRAN Codes 141
9 Wavelet Recipe 143
10 WTW Recipe 167
Preface

This midyear report summarizes our research for the first half of 1993. Consortium members for 1993 include Advance Geophysical, Amerada Hess, Amoco, Chevron, Conoco, Exxon, Fujitsu, GRI, Japon, Noranda, Oyo (intended), Texaco, Marathon, Unocal, and Phillips.

This document contains brief descriptions of new Fortran packages and includes research summaries in some of the following areas:

- Elastic traveltime+waveform (WTW) inversion of crosshole data. Elastic WTW is applied to synthetic and real crosswell data. Results show that it is an improvement over the acoustic WTW algorithm because 2-component data can be inverted for both P- and S-velocity structure. However, single component data (i.e., pressure field data) appear to be inadequate for S-velocity inversion;

- 3-D prestack migration of synthetic CDP data. A new Kirchhoff algorithm reduces the computational cost of standard 3-D prestack migration by almost an order of magnitude;

- Least squares migration of crosshole data shows a significant improvement in reflector resolution compared to standard migration if there are just a few source gathers (e.g., VSP data). However, the improvement is not so great when the migrated section is a composite of many prestack migrated sections;

- Imaging shallow structures by inverting CDP data by a refraction seismic method;

- Wavelet filtering and acoustic WTW codes.

We are now devoting significant efforts to extracting lithologic information from reflection events.
Part I

Tomography Methods
Report 1

Elastic Wave Equation Travel Time And Wave Form Inversion Of Crosshole Seismic Data

Changxi Zhou

1.1 ABSTRACT

The acoustic Wave Equation Travel Time and Wave Form (acoustic WTW) Inversion of crosshole seismic data was presented at the 1993 annual tomography and modeling meeting (Zhou et al., 1993). The acoustic WTW tomograms showed about 6 times more spatial resolution than the corresponding travel time tomograms. In this report we present the elastic wave equation travel time and wave form (elastic WTW) inversion method of crosshole seismic data which inverts both the P-wave and S-wave velocity structure. Comparison of the elastic WTW tomograms with the acoustic WTW tomograms shows that both methods can invert for the high resolution P-wave velocity structure when the S-wave energy is very weak in the recorded seismograms. Although the real data we used is dominated by unconverted P-wave energy, the elastic wave form inversion can still invert z-component data for the S-wave velocity structure. This comparison also shows that elastic WTW inversion is superior to acoustic WTW inversion when there is strong converted S-waves information in the recorded seismograms. The disadvantage of elastic WTW inversion is that it requires about 3 times more CPU time than the acoustic WTW method.

1.2 INTRODUCTION

S-wave velocity structure is important information for oil exploration. Because of the difficulties in picking direct S-wave arrival times, the inversion of S-wave velocity structure is very difficult by the traditional travel time tomography method. To avoid
picking S-wave traveltimes and still be able to invert for S-velocities we suggest an elastic WTW method. In this method we not only invert for the high resolution P-wave velocity structure but also can invert for S-wave velocity structure.

The elastic WTW inversion method is a straightforward extension of the acoustic WTW method. We only need to replace the acoustic wave propagation finite-difference modeling with elastic wave propagation finite-difference modeling and to calculate the gradients to update both P-wave and S-wave velocities. For the real 3-D data, we use a filter in the frequency domain to transform the data to 2-D. The detailed analysis of this transform method shows that the 2-D approximation is justified (Xu, 1993).

Seismic exploration data contains two distinct types of information, information about smooth velocity structure is associated with the traveltime data while information concerning the detailed parts of the velocity structure is associated with the reflection amplitude (Claerbout, 1985; Jannane et al., 1989). Therefore, we use cross-hole travel time inversion to resolve the long wavelength P-wave velocity structure and then use elastic waveform inversion to invert for the short wavelength P-wave and S-wave velocity structure. The results show this to be a successful way to invert for the high resolution P- and S-wave velocity structure from arbitrary starting models. Successful inversion is achieved if two component data is used, while the success of single component data inversion for both P- and S-velocities is questionable.

1.3 THEORY

In this section we present an outline of the elastic WTW algorithm. We define our inversion as finding the P- and S-wave velocity model which predicts the observed seismograms \( p(x_r, t|x_s)_{obs} \) that minimize the following misfit function:

\[
E = \frac{1}{2} \sum_s \sum_r (1 - w)[\delta \tau_{rs}]^2 + \frac{1}{2} \sum_s \sum_r \int dt \delta p_{rs}(t) w \delta p_{rs}(t).
\]  

(1.1)

Here \( \delta p_{rs}(t) = p(x_r, t|x_s)_{obs} - p(x_r, t|x_s)_{cal} \) is the seismogram residual, and \( \delta \tau_{rs} = \tau_{obs}(x_r, x_s) - \tau_{cal}(x_r, x_s) \) is the travel time residual, or the difference between the observed and calculated first arrival times for a source at \( x_s \) and a receiver at \( x_r \). The \( w \) (discussed in Luo and Schuster, 1990) is a weighting factor used to balance out the strength from these two residuals. We assume the density can be obtained from either well log data or by a simple empirical relation between P-wave velocity and density (Gardner et al., 1974). For an explosion source and two component geophones we perform the following operations for one iteration of the elastic wave form inversion.

1. Solve the elastic wave equation with zero initial conditions for each shot point by a 4th order finite-difference algorithm:
1.3. THEORY

\[
\rho \frac{\partial}{\partial t} v_x^f = -\left( \frac{\partial}{\partial x} \sigma_{xx}^f + \frac{\partial}{\partial z} \sigma_{xz}^f \right) = 0, \tag{1.2}
\]

\[
\rho \frac{\partial}{\partial t} v_z^f = -\left( \frac{\partial}{\partial z} \sigma_{zz}^f + \frac{\partial}{\partial x} \sigma_{xz}^f \right) = 0, \tag{1.3}
\]

\[
\frac{\partial}{\partial t} \sigma_{xx}^f = (\lambda + 2\mu) \frac{\partial}{\partial x} v_x^f + \lambda \frac{\partial}{\partial z} v_z^f + \sum_j S_j, \tag{1.4}
\]

\[
\frac{\partial}{\partial t} \sigma_{zz}^f = (\lambda + 2\mu) \frac{\partial}{\partial z} v_z^f + \lambda \frac{\partial}{\partial x} v_x^f + \sum_j S_j, \tag{1.5}
\]

\[
\frac{\partial}{\partial t} \sigma_{xz}^f = \mu \left( \frac{\partial}{\partial x} v_z^f + \frac{\partial}{\partial z} v_x^f \right), \tag{1.6}
\]

where \( S_j \) denotes the j-th source and \((v_x^f, v_z^f, \sigma_{xx}^f, \sigma_{xz}^f, \sigma_{zz}^f)\) denote particle velocities and stresses. During the computation, we sample the particle velocities at appropriate receiver locations to give \( p(x_r, t|x_s)_{cal} \) from equations 1.2 and 1.3.

2. Calculate the weighted residuals as defined in equation 1.1. The criteria we use for choosing the weighting factor is to set \( w = 0 \) for \( \delta \tau > T/4 \), and \( w = 1 \) for \( \Delta \tau \leq T/4 \), where \( T \) is the period corresponding to the peak frequency of the first arrival wavelet in a seismogram.

3. Compute the elastic wave equation in reverse time with zero final conditions by using the same 4th-order finite-difference algorithm for each shot point. Here the seismogram residual is treated as the source time history at the receiver location.

\[
\rho \frac{\partial}{\partial t} v_x^b = -\left( \frac{\partial}{\partial x} \sigma_{xx}^b + \frac{\partial}{\partial z} \sigma_{xz}^b \right) = \sum_l \sum_k \tilde{\eta}_{lk}, \tag{1.7}
\]

\[
\rho \frac{\partial}{\partial t} v_z^b = -\left( \frac{\partial}{\partial z} \sigma_{zz}^b + \frac{\partial}{\partial x} \sigma_{xz}^b \right) = \sum_l \sum_k \tilde{\eta}_{lk}, \tag{1.8}
\]

\[
\frac{\partial}{\partial t} \sigma_{xx}^b = (\lambda + 2\mu) \frac{\partial}{\partial x} v_x^b + \lambda \frac{\partial}{\partial z} v_z^b, \tag{1.9}
\]

\[
\frac{\partial}{\partial t} \sigma_{zz}^b = (\lambda + 2\mu) \frac{\partial}{\partial z} v_z^b + \lambda \frac{\partial}{\partial x} v_x^b, \tag{1.10}
\]

\[
\frac{\partial}{\partial t} \sigma_{xz}^b = \mu \left( \frac{\partial}{\partial x} v_z^b + \frac{\partial}{\partial z} v_x^b \right), \tag{1.11}
\]

where \( \tilde{\eta}_{lk} \) and \( \tilde{\eta}_{lk} \) denote the appropriate source time histories, which are the residuals of the seismograms for the l-th receiver of the k-th shot array.

4. Compute the perturbation of Lame parameters (Mora, 1987).

\[
\delta \lambda = - \sum_j \int_0^T dt \left( \frac{\partial}{\partial x} v_x^f \frac{\partial}{\partial x} v_x^b + \frac{\partial}{\partial z} v_z^f \frac{\partial}{\partial z} v_z^b \right), \tag{1.12}
\]

\[
\delta \mu = - \sum_j \int_0^T dt \left( \frac{\partial}{\partial x} v_x^b \frac{\partial}{\partial x} v_x^f + \frac{\partial}{\partial z} v_z^b \frac{\partial}{\partial z} v_z^f \right).
\]
\begin{equation}
+ \left( \frac{\partial}{\partial z} v_x^b + \frac{\partial}{\partial x} v_z^b \right) \left( \frac{\partial}{\partial z} v_x^f + \frac{\partial}{\partial x} v_z^f \right). \tag{1.13}
\end{equation}

In this step, we use the boundary values which we saved in forward propagation modeling (step 1) to recover the forward propagation field in the reverse time array.

5. Calculate the perturbation of model parameters by using the weighted gradient of P- and S-wave velocities.

\begin{align}
\delta v_p &= 2v_p \rho \delta \lambda, \tag{1.14} \\
\delta v_s &= -4v_s \rho \delta \lambda + 2v_s \rho \delta \mu. \tag{1.15}
\end{align}

In every iteration we perform these five steps and use the subspace method (Kennett et al., 1988) to calculate the step length to update the P- and S-wave velocities. We assume density is known and choose P- and S-wave velocity values as model parameters in our inversion because they are much better resolved than Lame parameters, \( \lambda \) and \( \mu \) (Tarantola et al., 1985).

### 1.4 NUMERICAL EXAMPLES

We apply the elastic waveform inversion method to both synthetic and real crosshole seismic data. The real data is collected by Exxon (Chen et al., 1990) near Friendswood, Texas. In all cases we use a non-linear steepest descent method with pre-conditioning (Beydoun and Mendes, 1989). For each test we compare the elastic waveform inversion with acoustic waveform inversion to show the advantages of elastic waveform inversion.

1. Synthetic Crosshole Data

The elastic waveform inversion method is applied to the fault model. The P-wave velocity ranges from 2300 m/s to 3600 m/s (Figure 1.1a). We assume that \( V_z \) equals 0.5\( V_p \) to get the initial S-wave velocity distribution (Figure 1.1b). We use an empirical formula to assign the density distribution from the P-wave velocity distribution and use the same empirical formula to update the density in every iteration of the inversion. The "observed" seismograms in this case are generated by a 4th-order finite-difference solution to the 2-D elastic wave propagation. The fault model is discretized onto a mesh with 162x242 grid points with 18 explosion line sources and 36 two component receivers along the left side and right side of the model respectively; a 50 gridpoint wide absorbing sponge zone is extant along each boundary. The source function is a Ricker wavelet function added onto the x and z stress components. The receivers record particle velocities in the receiver locations. The source wavelet has a peak frequency of 60 Hz and the starting P-wave velocity model is homogeneous with a velocity value of 3000 m/s.
1.4. NUMERICAL EXAMPLES

Figure 1.2a shows the first arrival acoustic wave equation traveltime (WT) inversion for the P-wave velocity after the 10th iteration. We use this tomogram as the initial guess of the P-wave velocity distribution for elastic waveform inversion and use Poisson's relationship to assign the initial S-wave velocity model. After 6 iterations we get the tomograms for P- and S-wave velocities shown in Figure 1.2b. The tomograms show very good interface definition compared to the true model.

For comparison, Figure 1.3 shows the acoustic WTW tomogram for the P-wave velocity distribution after 6 iterations. The recorded seismogram we used for the inversion are the summation of z and x components of normal stress; these are generated by the 4th-order elastic wave propagation finite-difference modeling at appropriate receiver locations. There is converted S-wave energy in the recorded data so that the acoustic WTW tomogram is not as accurate as the elastic WTW tomogram.

The elastic WTW is robust with respect to the initial guess of the S-wave velocity distribution. This is demonstrated by elastic WTW inversion of the previous fault model shot gathers except the homogeneous S-wave velocity (1500 m/s) distribution is assigned when we begin the waveform inversion. Figure 1.4 shows the P-wave tomogram after 6 iterations. The accuracy of the tomogram is quite acceptable.

In the previous tests, elastic WTW was applied to 2-component data to invert for both P- and S-velocities. The next test attempts to invert for both P- and S-velocities from single component pressure field data. The results are shown in Figure 1.5 and demonstrate that the success of inverting single component data is questionable.

2. Real Crosshole Data

The elastic wave form inversion is now applied to a real crosshole seismic data set collected by Exxon near their Friendswood, Texas test site (Chen et al., 1990, Zhou et al., 1993). The offset of the two wells is 600 feet, the depth of the wells is 1000 feet, and the source and receiver intervals are 10 feet. There are 98 sources and 96 receivers in the source and receiver wells. The source is an explosion source which consists of a small amount of dynamite and the seismic data has a usable bandwidth of 80 to 600 Hz. The receivers used here are the hydrophones which record the pressure field. A typical unprocessed shot gather at intermediate depth is shown Figure 1.6a.

The processing steps applied to the shot gathers include (Cai and Schuster, 1993) 1). eliminating the tube waves by median filtering; 2). free-surface reflections were muted out; 3). an 80-600 Hz bandpass filter was applied to the data; 4). each seismogram was normalized to its maximum value; and 5). direct arrivals were muted after the waveform inversion was turned on. Each forward modeled shot gather used a source wavelet extracted from the corresponding observed shot gather and the wavelet is used as the time history of the x and z components of the normal stress when doing elastic wave propagation modeling; e.g., Figure 1.6b shows the first arrival source wavelet associated with a trace at intermediate depth. To accommodate the 80-600 Hz bandwidth of the data, a 2-D finite difference mesh of 303x501 gridpoints was used for the forward modeling and back-projection, with the same well geometry as in the Friendswood experiment. Well deviations in the source and receiver wells were
corrected by applying an appropriate time shift to the raw seismograms. The data were corrected to 2-D by multiplying the filter $\sqrt{i/\omega}$ by the spectrum of the observed seismograms. The final processed shot gather associated with Figure 1.6a is shown in Figure 1.6c.

Because the recorded data is dominated by unconverted P-waves, we apply the acoustic wave equation travelt ime and waveform inversion to the 98 shot gathers of the processed Friendswood data. After 46 iterations we get the P-wave tomogram in Figure 1.7a. Then we use this tomogram as the initial P-wave velocity model for the elastic WTW inversion and use the relationship $V_s = 0.5V_p$ to assign the S-wave velocity starting model from the P-wave velocity. After 6 iterations of elastic waveform inversion we get the P- and S-wave tomograms in Figure 1.7b and Figure 1.7c. The final tomograms provide fine layer and velocity resolution. Because the recorded data is dominated by unconverted P-waves, the elastic WTW tomogram (Figure 1.7b) is close to the acoustic WTW tomogram (Figure 1.7a). This structure and resolution are verified in Figure 1.8 which compares the smoothed sonic log (solid line) in the source hole to a vertical slice of the final elastic waveform inversion P-wave velocity tomogram (6b). The slices were taken from the tomogram along a vertical line 40 feet from the sonic log. Although we have no well logs of S-wave velocity, the S-wave tomogram shows a good correlation with the P-wave velocity structure.

Finally we compare the synthetic shot gathers computed from the velocity field in the, respectively, acoustic WTW 46th iteration tomogram and the final elastic WTW tomogram with the recorded shot gather (Figure 1.9). In the elastic waveform inversion synthetic shot gathers, the seismic events match the observed shot gathers better than those from acoustic synthetic shot gathers.

1.5 CONCLUSION

We presented the elastic WTW method for inverting 2-component crosshole seismic data. Inversion of both synthetic and real crosshole seismic data shows that this method can provide a significantly better model resolution than that given by travelt ime tomography. The synthetic test shows this method also can provide high resolution S-wave velocity tomograms for the data which contains significant S-wave information. Compared to acoustic WTW inversion, elastic WTW inversion method provides more information about the model from the same recorded seismic data. Results also show that inverting both P- and S-velocities from single component data is lightly to be unsuccessful.

Future work on elastic waveform inversion will study the possibility of invert the density of the model in the same time.
1.6 REFERENCES


Claerbout, J. F., 1985, Imaging the Earth’s interior, Blackwell Scientific, Boston, Mass..


Jannane, M., and Equipe de Tomographie Geophysique, 1989, The wavelengths of Earth structures that we can resolve from seismic reflection data, Geophysics, 54, 906-910.


1.7 ACKNOWLEDGEMENTS

We are very grateful to Fujitsu Computer Company for the use of their VPX-240 supercomputer. Acknowledgement is made to the Donors of The Petroleum Research Fund, administrated by the American Chemical Society for partial support of this research (contract PRF\$ 22807-AC2, PID 8909029). We are also grateful for the financial support provided by the 1992 University of Utah seismic tomography consortium members; Amerada Hess, Amoco, Arco, Chevron, Conoco, Exxon, Gas Research Institute, Japon, Marathon, Noranda, Oyo, Phillips, Texaco, and Unocal.
Figure 1.1: The P- and S-wave velocity structure of the Fault model
Figure 1.2: (a) The acoustic WT tomogram after 10 iterations. (b) The elastic WTW tomograms after 16 iterations (inverting P- and S-wave velocity structure in the same time).
Figure 1.3: Acoustic WTW tomogram after 16 iterations.
Figure 1.4: Elastic WTW tomogram after 16 iterations for a homogeneous S-wave velocity model as the starting model.
1.7. ACKNOWLEDGEMENTS

Figure 1.5: Elastic WTW tomogram after 16 iterations by using single component pressure field data.
Figure 1.6: A typical shot gather of Friendswood crosshole data collected by Exxon near their Friendswood, Texas test site. The source depth is 520 feet. (a) Raw shot gather. (b) First arrival wavelet extracted from the Figure 5a shot gather. (c) Figure 5a shot gather data after signal processing.
Figure 1.7: (a) Acoustic WTW tomogram after 46 iterations, (b) and (c) Elastic WTW P- and S-wave velocity tomograms after 6 iterations by using the Figure 1.7a tomogram as the initial model.
Figure 1.8: Elastic WTW P-wave tomogram velocity profiles (dashed lines) compared to the sonic log (solid line) in the source well. The profiles are extracted from the tomograms 40 feet from the source well.
Figure 1.9: Synthetic acoustic and elastic common shot gathers associated with the (a) acoustic waveform inversion tomogram in Figure 1.7a, and the (b) elastic waveform inversion tomogram in Figures 1.7b and 1.7c. Figure 1.9c depicts the corresponding observed shot gather. The source location for these common shot gathers is at the depth 520 feet.
REPORT 1. ELASTIC WTW
Report 2

Traveltime Inversion of Refraction Seismic Data for Statics Correction: Synthetic Data Test

Fuhao Qin

2.1 ABSTRACT

The refraction traveltime inversion method developed by Qin and Cai (1992) is improved to deal with an irregular earth surface and large models. A very low velocity value is given to the area above the free earth surface to ensure that this area will not affect the first arrival traveltime in any way. Large models are divided into several overlapping smaller segments in order to accommodate memory limitations in a workstation. The overlapping parts ensure that edge effects from the inversion will not affect the final tomogram. It is tested on a synthetic model that simulates a realistic seismic exploration site in South America. The inversion results using both eikonal equation traveltimes and finite-difference traveltimes as input data closely depict the shallow velocity distribution of the model. Therefore, it is very likely that the method will provide a good way to deal with the statics problem in seismic exploration.

2.2 INTRODUCTION

Zhu et al. (1992) showed that turning ray tomography (refraction traveltime inversion) can image near-surface velocities more accurately than ordinary refraction statics methods. However, the method of Zhu et al. is a linear inversion method. The raypaths are considered invariant with respect to the iteration number. Thus, it requires a good initial velocity model and the final result will rely on the estimation of the initial model.
Qin and Cai (1992) proposed a nonlinear inversion method which does not always require a good initial guess. The code updates raypaths as well as the velocity distribution iteration by iteration. Numerical results showed that this method is easy to use and effective for various problems ranging from engineering and exploration. However, the code used in Qin and Cai (1992) was not suitable for problems with an irregular free-surface and large models. In this report, I will discuss the treatment of an irregular surface and large models which are not uncommon in seismic exploration. The proposed scheme is tested on a synthetic model designed for a reflection study.

2.3 METHODOLOGY

In this section, I will first reiterate the general refraction traveltime inversion method of Qin and Cai (1992) and then discuss the irregular free-surface and large model treatments.

2.3.1 Refraction traveltime inversion scheme

To obtain the model that minimizes the first arrival traveltime residuals from any starting model, Qin and Cai (1992) proposed the following procedure:

1. Calculate all the refraction ray paths based on the eikonal equation solution (first arrival traveltime distribution).

2. Find traveltime residual \( \Delta t_i = t_i^{\text{obs}} - t_i^{\text{cal}} \) for each source and receiver pair.

3. Calculate the negative gradient of the traveltime misfit function (the model updating direction) similar to the SIRT method (van der Sluis, A. and van der Vorst, H. A., 1987).

\[
    g_j = -\frac{\sum_{i=1}^{N_j} \Delta t_i}{N_j}
\]  

(2.1)

where \( g_j \) is the negative gradient in the \( j_{th} \) cell, \( N_j \) is the number of rays that visit the \( j_{th} \) cell and the summation \( i \) is over the indices associated with ray paths that visit the \( j_{th} \) cell.

4. If a cell has no ray passing through it, let it have the same gradient value as the cell just above it.

5. Smooth the gradient and update the model.

The key point in the above procedure is the fourth step. The physical explanation is to extend the gradient field downward from the deepest point where rays can reach. Or, in other words, since we do not have information beneath the depth of maximum ray penetration the best we can do is to assume that below this depth the velocity is
2.3. METHODOLOGY

Figure 2.1: Refraction ray paths and the downward extension of the gradient. The gradients in different shadings have similar values.

the same. Figure 2.1 shows a layered earth model with some refraction raypaths in it. The rays are concentrated near the interfaces and there is almost no information within each layer. However, the downward extrapolation of the gradient will provide each layer with the same updating direction.

In addition, the horizontal smoothing operator will help stabilize the refraction traveltime inversion and help avoid getting stuck in unreasonable local minima.

2.3.2 Irregular free-surface and large model

In seismic exploration, many surveys are carried out in areas where elevation changes with offset. It is therefore desirable to develop methods that can deal with this problem correctly. The criterion of correctness is simple; i.e., the region above the free-surface should not affect the first arrival traveltime calculation by any means. This, actually, is also very simple to achieve. Our solution is to give a low velocity value to the region above the free-surface. A low velocity is chosen so that no first arrival ray will go through it; it is not too low as to affect the stability of the finite-difference eikonal equation solver. We found that one third to one half of the minimum velocity of the model is good for the model we tested.

Another problem to deal with is the size of the model. A seismic line will extend for tens and even hundreds of kilometers. It will be too big to fit in a computer as a single model. However, we found that there is no problem to separate it into small segments. These segments should overlap one another. The overlap should be large enough to cover the edge effect of the inversion result. To invert a certain segment, only rays that start and end in this segment are used. After all segments are inverted, they are combined into one large model by throwing away the overlapped parts that are affected by edge artifacts.
2.4 MODEL TEST

The model used is the top part of a model designed by Kun Hua Chen (Chevron) for a seismic reflection study; it is based on topography in South America. The size of the earth model is 22 km by 12 km (Figure 2.2) and the model for the refraction study is reduced to 21.61 km by 1.2 km (Figure 2.3). The maximum surface elevation change along this line is about 700 meters and the velocity varies from 2000 m/s to 3800 m/s.

There are 261 sources evenly distributed within the offset range of 3000 m and 18600 m. The source interval is 60 m. For each source, 101 receivers are assumed to be located within a 6000 m offset range which is centered at the source location. The receiver interval is also 60 m.

The finite-difference eikonal equation solver was used to calculate the first arrival traveltimes. In the forward modeling process, the region above the irregular free-surface was given a velocity of 1000 m/s.

In the inversion process, the grid spacing was set to 10 m. We can not use a very large grid spacing, since: (1) The finite-difference eikonal equation solver requires a finer grid to obtain more accurate traveltimes; (2) The particular gradient calculation scheme we used also requires a fine grid; (3) The refractors are usually not very deep. The model size is then 2161 by 121. It was then divided into three parts as discussed in the previous section. Each part is 901 by 121 gridpoints and the inversion was carried out for each part. For the inversion, a 2000 m/s homogeneous velocity model is used as a starting model and the inversion is considered complete when traveltime residuals stop decreasing (i.e., after about 50 to 60 iterations). The final result was obtained by combining all three parts. Figure 2.4 shows the result.

Comparing Figure 2.4 and Figure 2.3, we can see that the velocity of the top weathered zone is well reconstructed. The depth and shape of the refractor is also reconstructed except near the left and right edges. The velocity trend of the refractor is depicted in the tomogram. The result will definitely be a good model to do statics correction. One thing that needs to be mentioned is the vertical strips caused by the downward gradient extrapolation. This is actually a good indication of the refraction ray penetration depth. Anything below this point is not trustworthy. Another problem with the tomogram is that the velocity of the refractor is not very smooth (i.e., horizontal oscillations). We are still working on this problem.

Although the inversion of eikonal equation traveltimes is successful, there is some doubt about whether high frequency eikonal traveltimes can simulate traveltimes for finite-frequency wave propagation. The following test was designed to make this point clear. In this test, the wave equation (PP4 in Recipe Book) was solved by a finite-difference method to calculate seismograms. First arrival traveltimes were then picked from the seismograms. Figure 2.5 shows a common shot gather (CSG) with the shot at an offset of 9000 m. The source wavelet is a 16 Hz Ricker wavelet. To satisfy the dispersion and stability criterions, the grid spacing was set at 10 m; and the time step
Figure 2.2: The South America model designed for the seismic reflection study. (Courtesy of Kun Hua Chen of Chevron Overseas Petr. Inc.)
Figure 2.3: Top part of the model shown in Figure 2.2 which is used in the following refraction study.
Figure 2.4: Inversion result using the eikonal equation traveltimes.
length was 1 ms. The CSG was plotted using a 1.4 second auto-gain-control (AGC) window.

To amplify the first arrival signal for correct picking, an AGC gain with a window of 0.1 s was applied to the seismogram before an automatic picker picked the traveltimes. Figure 2.6 shows the same seismogram as that in Figure 2.5 except for the AGC gain. The picker picks the first arrival time at the point where 5% of the largest amplitude first occurs in that trace. A time shift is applied which is obtained from the zero offset trace. It is then considered the first arrival traveltime.

Figure 2.7 shows the inversion result from the picked traveltime. There is no obvious difference between this result and that obtained from the eikonal traveltimes in Figure 2.4. So, we conclude that the first arrival traveltimes from a finite frequency seismic survey do include the refraction information needed for the inversion.

However, we still have a problem. Comparing Figures 2.5 and 2.6, we can see that the first arrivals for some of the refraction signals are very weak. It will be very difficult to pick them without the help of a small window AGC. In the real data case, the weak first arrival might be buried in noise and the AGC gain will not help much. Then, we will be in trouble. Suggested solutions include increasing the source energy and source frequency. If necessary, do two different surveys, one for seismic reflections whose aim is to penetrate as deep as possible, the other will be to emphasize the refraction first arrival.

2.5 DISCUSSION

The refraction traveltime (turning ray) tomography method has been further developed to deal with irregular free-surfaces and large model problems. It can easily be used for statics calculations in seismic data processing.

According to our numerical tests in this report and previous reports, this method is stable and converges to a model that minimizes the traveltime residual. However, it can not deal with the nonunique solution problem. It also requires an accurate recording of the first arrival traveltimes, since it is based on the eikonal equation solution.

Traveltimes picked from finite frequency seismogram can be used for refraction tomography. However, attention should be paid to the correct picking of the first arrival traveltimes since the first arrival signal may be very weak.

2.6 REFERENCES

Marsden, D., 1993, Static correction—a review, Part II: The Leading Edge, 12, 115-120.

Qin, Fuhao and Cai, Wenying, 1993, Inversion and imaging of refraction data: Univer-
Figure 2.5: A finite-difference common shot gather with an AGC window of 1.4 s.
Figure 2.6: The same seismogram as shown in Figure 2.5 except that the AGC window is 0.1 s instead of 1.4 s.


Figure 2.7: Inversion result using the picked finite-difference seismogram traveltimes.
3.1 ABSTRACT

I present a brief back of the envelope study that assesses the feasibility of imaging interwell lithology by seismic crosswell experiments with 1.0-1.5 mile well offsets. Preliminary results suggest that source energies of no more than about 250 Hz are detectable at well offsets of 1 mile or more. This assumes current source technology. If this is true then traveltome tomographic inversion and migration are limited to resolving structures no thinner than 400 feet and 40 feet, respectively.

3.2 INTRODUCTION

A major goal in reservoir engineering is to devise a means to accurately delineate formation lithology between wells. Towards this goal geophysicists have developed several crosswell imaging methods: traveltime crosswell tomography, migration of reflections and waveform inversion. Ideally, traveltime tomography can reconstruct the smooth part of the interwell velocities to a resolution of about two or three source wavelengths. Reflection migration can image the high wavenumber parts of the velocity interfaces, and waveform inversion can image both the high wavenumber and low wavenumber parts of the velocity field. These velocities can be used to assess the lithological distribution between adjacent wells.

3.2.1 Key problem

A key problem in crosswell imaging of gas reservoirs is that the distances between pre-existing gas wells are much greater than that for oil wells. Typically offsets of 1
to 1.5 miles or more are encountered. Such offsets coupled to short depth intervals present a formidable challenge to conventional imaging methods developed for oil well spacings of 500 to 1000 feet. Can the conventional imaging methods work well with well offsets of 1 to 1.5 miles? This is the subject of this short report.

3.3 BACK OF THE ENVELOPE CALCULATIONS

To answer the question "Can the conventional imaging methods work well with well offsets of 1 to 1.5 miles?", we first do some back of the envelope calculations to assess what is possible.

These calculations include estimating the amplitude attenuation, traveltime errors due to dispersion, Fresnel zone widths of the transmitted arrivals, and the wavelet stretch of the migrated reflections associated with 1-1.5 mile seismic crosswell imaging.

3.3.1 Amplitude attenuation estimate

We assume that the attenuation of a monochromatic seismic wave (in "Seismic Wave Attenuation" by Toksoz and Johnston, 1981, Geophysics Reprint Series) is roughly given by

$$ a(x) = a_0 e^{\frac{-\pi f x}{c Q}} \frac{1}{x} \tag{3.1} $$

where $f$ is the source frequency, $x$ is the source-receiver distance, $Q$ is the Quality factor, $a_0$ is a normalization factor, and $c$ is the medium velocity.

We will now use equation 3.1 to estimate the seismic attenuation associated with source-receiver offsets of 1 or more miles. The earth models are characterized by crosswell offsets of 500 feet to 7500 feet and we will use source frequencies of 250 Hz, 500 Hz, and 1000 Hz. Two types of rock lithologies are examined (see Figure 3.1), soft rocks ($Q \approx 10 \text{ to } 100$) and hard rocks ($Q \approx 200 \text{ to } 500$). Johnston (in "Seismic Wave Attenuation" by Toksoz and Johnston, 1981, Geophysics Reprint Series) shows graphs of compressional $Q$ vs confining pressure for kHz frequencies, suggesting that wet Wingate, Berea or Navajo sandstones have $Q$'s between 10 and 100. Even wet Bedford limestones have $Q$'s within this range. Smaller $Q$ values mean more attenuation.

Figure 3.2 plots $\frac{a(x)}{a(500)}$ versus well offset for various frequencies and $Q$ factors for a homogeneous earth model with velocity $v = 7000$ ft/s. I chose the normalization factor of $a(500)$ because we know that good signal can be obtained at that range (e.g., the Exxon Friendswood crosswell data with a well offset of 600 feet using a few grams of explosives. In this case the signal amplitude appeared to be no stronger than an order of magnitude of the background noise.). These plots show that:
Figure 3.1: Q values of sandstones, limestones and shales measured in the lab (from Toksoz and Johnson, 1981, Seismic Wave Attenuation, Geophysics Reprint Series)
Figure 3.2: Amplitude attenuation vs well offset for various values of Q.
• For soft rocks ($Q=100$) and high kHz source frequencies (1 kHz) there is almost a 10 order of magnitude decrease in amplitude (or 20 order magnitude decrease in energy) at 1 mile offset. Thus, 1 kHz energy at 1 mile well spacing appears to be nearly impossible for transmitted or reflected waves generated by a few grams of explosives. Larger quantities of explosives will destroy the well. Electromechanical sources might achieve greater energies than an explosive, but (I guess) not by more than 3 or 4 orders of magnitude.

• For soft rocks ($Q=100$) and moderate source frequencies (250 Hz) there are about 3 orders (for $Q = 100$) and 6 orders (for $Q = 50$) of magnitude decrease in amplitude (or 6 to 12 orders of magnitude decrease in energy) at 1 mile offset. Thus, 250 Hz energy at 1 mile spacing appears to be difficult, but perhaps attainable for longer sweeps and more power from an electromechanical source. A trapped wave such as a channel wave might make it through provided it is in relatively hard rock. Note that the attenuation estimate of a, say 500 Hz, source at $Q = 100$ is the same as the $Q = 50$ curve for a 250 Hz source (i.e., doubling the source frequency is equivalent to halving the Q value).

These rough calculations suggest that 250 Hz energy might be attainable at 1 mile offsets, but 1 kHz energy appears to me to be almost practically impossible with current technology.

### 3.3.2 Traveltime dispersion estimate

The arrival time of an event will be distorted due to attenuation of the amplitude as well as the inherent dispersion of the arrival (higher frequencies travel faster and lower frequencies travel slower). For example, an impulsive source will spread into a Gaussian shaped arrival as the faster higher frequencies are attenuated relative to the slower lower frequencies. This means that picking the peak of the Gaussian as the first arrival traveltime $t_{peak}$ will yield an erroneous traveltime (velocity) that is longer (slower) than the actual first arrival traveltime $t_{actual}$ (velocity). This error increases with an increase in well offset. Figure 3.3 depicts the spreading of an impulsive wavelet for a medium with a Q of less than 100.

To estimate $t_{peak} - t_{actual}$ with respect to offset I use Strick's model (equation 21 in Strick. 1970, Geophysics. v. 35, p. 387-403) of dispersion and attenuation for an almost constant-Q behavior of solids over a wide range of frequencies. Figure 3.4a plots $t_{peak} - t_{actual}$ values vs well offset for rocks with Q values of approximately Q=20, 70 and 100. This plot shows that at an offset of 3000 feet there can be a traveltime pick error of nearly 20 msec for a rock with Q=70. Figure 3.4b shows how this translates out to a velocity error if velocities are computed by the formula $v = d/t$. 
Figure 3.3: Plot of impulse response of an attenuative homogeneous medium for different source-receiver offsets, with a maximum offset of R=32,000 ft (from Strick, 1970, Geophysics, p. 387-403). The Q value is assumed to be less than 100.
Figure 3.4: a). $t_{\text{peak}} - t_{\text{actual}}$ vs well offset for homogeneous media with $Q=20$, 70, and 100. b). Velocity errors associated with picking the peak amplitude as the first arrival.
3.3.3 Traveltime tomographic resolution estimates

Traveltime tomography is a velocity imaging tool that can be applied to crosswell seismic data. Let's now estimate the image resolution associated with tomographic inversion of crosswell traveltime data at 1 mile offset.

Williamson and Worthington (1993, "Resolution limits in ray tomography due to wave behavior", p. 727-735 in Geophysics) showed that the maximum width $W$ of the first Fresnel zone of the transmitted arrival along a wavepath is approximated by

$$W = \left( \frac{dc}{f} \right)^{1/2}, \quad (3.2)$$

where $c$ is the velocity of the homogeneous medium, $d$ is the offset between source and receiver wells, and $f$ is the frequency of a monochromatic source. They claimed that the width of this Fresnel zone is roughly equal to the size of the smallest resolvable heterogeneity between the wells. Indeed, their numerical experiments seemed to verify this conclusion.

Figure 3.5 plots the Fresnel Zone width versus the well offset for source frequencies of 250 Hz, 500 Hz, and 1000 Hz. For a 250 Hz source, the Fresnel zone is almost 400 feet wide at a 1 mile well offset; this suggests a very poor resolution capability for traveltime tomographic methods. Resolving bodies no smaller than 400 feet is not much better resolution than provided by a good CDP imaging experiment.

3.3.4 Migration resolution estimates

Reflection migration is another imaging tool that can be applied to crosswell seismic data. Let's now estimate the image resolution associated with reflection migration of 1 mile offset crosswell seismic data.

If the traveltime $t_{rs}$ between the source and receiver of a reflected ray is given by

$$t_{rs} = \frac{2\sqrt{h^2 + d_{1/2}^2}}{c} \quad (3.3)$$

where $c$ is the velocity of the homogeneous medium, $d_{1/2}$ is the half-offset between source and receiver wells, and $h$ is the depth to the horizontal reflector as measured from the equi-level source and receiver (see Figure 3.6). Differentiating $t_{rs}$ with respect to $h$, and multiplying the result by $dh$ gives

$$dt \approx \frac{dt_{rs}}{dh} dh = \frac{2dh}{c} \frac{h}{\sqrt{h^2 + d_{1/2}^2}} \quad (3.4)$$

or rearranging terms we get

$$dh = \frac{.5c\sqrt{h^2 + d^{21/2}}}{h} dt \quad (3.5)$$
Figure 3.5: Fresnel zone width versus well offset for a homogeneous medium.

Figure 3.6: Reflection ray path for a source and receiver at the same depth level.
Figure 3.7: Wavelet stretch versus well offset for a migrated reflection.

where $dt$ can be considered as the wavelet period, and $dh$ is now interpreted as the wavelet stretch or thickness of the migrated wavelet. Equivalently, $dh$ can be interpreted roughly as an estimate of the thickness resolution of the migrated image.

Figure 3.7 plots $dh$ versus source well offset for the source frequencies of 250 Hz, 500 Hz and 1000 Hz, with reflector depths of 500 feet and 1000 feet; shallower reflector depths will result in worse resolution. Since 250 Hz may be a realistic source frequency, then the best depth resolution of the migrated image for offsets of 1 mile is about 40 (80) feet for a 1000 (500) feet deep reflector. Of course, deeper reflectors should provide better depth resolution but a well depth interval of larger than 1000 feet may be impractical for now. In any case a thickness resolution of 40 feet is not highly encouraging, and we are not even taking into account problems due to limited aspect ratios and migration velocity errors.

The previous analysis using Fresnel zones also suggests a minimum lateral resolution for reflection imaging. At 250 Hz, Figure 3.5 suggests a Fresnel zone width of
approximately 400 feet for transmitted arrivals, so that crosswell reflection wavepaths will have even larger Fresnel zones. Thus, the minimum lateral resolution of a migrated image will be no smaller than 400 feet for 250 Hz energy at 1 mile offset.

### 3.3.5 Aspect ratios

The slice projection theorem suggests that offset to depth ratios of 2:1, 3:1 and 6:1 will limit the dip resolution of dipping layers to be less than, respectively, 26 degrees, 18 degrees, and 9.6 degrees. This assumes imaging by travelttime tomographic methods. Therefore, if we have a well depth interval of 1000 feet then we can not resolve dips larger than about 10 degrees at a 1 mile offset.

### 3.4 CONCLUSION

Back of the envelope calculations suggest that it may be possible to propagate 250 Hz transmitted or reflected energy out to offsets of 1 mile. This assumes soft rocks with $Q$'s of about 100. However, imaging resolution by travelttime tomography or migration should be no better than about 400 feet or 40 feet, respectively. Travelttime pick errors due to dispersion are estimated to lead to velocity errors of at least 8 percent. For a 1000 foot well depth interval, dip resolution should be no better than about 10 degrees. Since Jerry Harris's source can provide good data out to offsets of 1300 feet, then the next step is to redo these calculations using the base well offset of 1300 feet rather than 500 feet.
Part II

Migration
Report 4

3-D Prestack Migration Using Quasi-Monte Carlo Methods

Yonghe Sun

4.1 ABSTRACT

We present the results of a preliminary study on the efficiency of 3-D prestack depth migration using Quasi-Monte Carlo methods. Two models are used for this study: one model has a flat horizontal reflector, and the other is the French 3-D reflector model consisting of two domes and a fault. The synthetic seismograms are migrated using both regularly spaced source-receiver grid points and quasi-randomly distributed source-receiver points. Results show that if the source-receiver positions are allowed to vary freely at the earth's surface, the Quasi-Monte Carlo migration can be an order of magnitude more efficient than the conventional migration using regularly spaced source-receiver points. When the source-receiver positions are severely limited to be a small subset of the regularly spaced grid points (i.e., typical field geometries), the efficiency improvement by a Quasi-Monte Carlo method is reduced to a factor of 5 or so for noise free data. The efficiency gain decreases as the level of random or coherent high frequency noise increases in the data.

4.2 BACKGROUND

4.2.1 Near-optimal quasi-Monte Carlo methods

Recently there has been a theoretical breakthrough in the understanding of the computational complexity of several Quasi-Monte Carlo methods for multivariate integration (Wozniakowski, 1991). See also, Cipra, "Breaking the Curse of Dimensionality", Science magazine, 1991. Wozniakowski showed that modified Hammersley points
lead to efficient numerical multivariate integration when dimensionality is high. Consider an \( n \)-dimensional integral of a multivariate function \( f(x_1, x_2, ..., x_n) \) over an \( n \)-dimensional unit cube,

\[
I = \int_0^1 \int_0^1 ... \int_0^1 f(x_1, x_2, ..., x_n) dx_1 dx_2 ... dx_n. \tag{4.1}
\]

The integral can be approximated by the summation,

\[
I \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i), \tag{4.2}
\]

where \( x_i (i = 1, 2, ..., N) \) is a set of \( N \) discrete points in the unit cube. The summation approaches the value of the integral for large \( N \) if the points are uniformly distributed in the cube. The following summarizes the asymptotic (for large \( N \)) average-case computational complexity \( N \) (or number of points required) for a given integral error tolerance \( \epsilon \) (Zwillinger, 1992) using various types of points:

- For regularly spaced grid points,
  \[
  N = C_1(1/\epsilon)^n.
  \]

- For random points,
  \[
  N = C_2(1/\epsilon)^2.
  \]

- For Halton points,
  \[
  N = C_3(1/\epsilon)[\log(1/\epsilon)]^n.
  \]

- For Hammersley points,
  \[
  N = C_4(1/\epsilon)[\log(1/\epsilon)]^{(n-1)}.
  \]

- For Wozniakowski points,
  \[
  N = C_5(1/\epsilon)[\log(1/\epsilon)]^{(n-1)/2}.
  \]

The Wozniakowski points are shifted Hammersley points. The proportionality constants \( C \)'s in the above expressions are all different and not analytically known. Clearly the regular grid points are very inefficient for large \( n \). The above expressions suggest that Wozniakowski points are "optimal" while Halton and Hammersley points are "nearly as good".
4.2. BACKGROUND

4.2.2 Quasi-Monte Carlo migration

The mathematical formulation of the diffraction stack 3-D prestack depth migration is very simple. It can be expressed by,

\[ M(x) = \sum_{r} \sum_{s} P_{sr}(t(x_s, x) + t(x_r, x)), \tag{4.3} \]

where \( M(x) \) is the migrated image intensity at the subsurface point \( x \), \( P_{sr}(t) \) is the gained seismogram trace at the receiver \( r \) for the source \( s \), \( t(x_s, x) \) is the travel time field for the source at \( x_s \), and the summations are performed over the 2-D areal distribution of the sources and the receivers. The 2-D summations are equivalent to a 4-D integration. The challenge is to perform the summations with practical turnaround time and limited computer resources. This is where we may benefit from optimal multi-dimensional integration algorithms.

4.2.3 Previous studies

Sikorski and Schuster (1991) used the Hammersley points for 3-D Kirchhoff migration of zero-offset synthetic data. Synthetic data are migrated for a horizontal reflector model and a salt dome model. For zero-offset or post-stack data, each source has only one receiver, which is coincident with the source. So in equation 4.3, the two-fold summations reduces to just the summation over the sources. Figure 4.1 shows the first 300 Hammersley points for 2-D integration (with \( n = 2 \)). The study showed that no substantial improvement of migration efficiency was gained relative to standard Kirchhoff migration. This suggests that the dimensionality \( n = 2 \) is probably too low to benefit from the Hammersley point integration intended for large \( n \). The fact that the Hammersley migration for \( n = 2 \) did not perform worse than conventional migration suggests that \( n = 2 \) may be just “barely” too low for the Hammersley points to be efficient. Prestack migration of 3-D data is a 4-D integration and \( n=4 \) may be large enough to show the efficiencies of Hammersley (or other Monte Carlo) migration. This is the subject of this report: 3-D prestack Monte Carlo migration.

4.2.4 The horizontal reflector model

Two models are used to study the efficiency of Quasi-Monte Carlo 3-D prestack migration. The first model consists of a horizontal flat reflector. The second model is the French model (French, 1974), consisting of two domes and a faulted horizontal reflector. In this section, we present the 3-D prestack migration results for the horizontal reflector model using the Quasi-Monte Carlo methods.

As a preliminary migration efficiency test, consider a unit 3-D cube of earth (Figure 4.2). The sources and receivers can be anywhere on the top surface of the cube. Assume that the there is a horizontal reflector at halfway depth (\( z=0.5 \) measured in
Figure 4.1: Locations of the first 300 Hammersley points.

(from Sikorski and Schuster, 1991)
4.2. **BACKGROUND**

![Diagram of a horizontal reflector model](image)

Figure 4.2: The horizontal reflector model.

whatever the unit is). Let the velocity above the reflector be constant \( v = 1 \) (choosing the length and time units so that \( v = 1 \)). Consider the following integral,

\[
M(x) = \int_{0}^{1} \int_{0}^{1} \int_{0}^{1} P d_{x} d_{y} d_{x_r},
\]  

(4.4)

where \( P \) is defined by the Ricker wavelet,

\[
P(t) = [1 - 2(\pi f_{m} t)^2] e^{- (\pi f_{m} t)^2},
\]  

(4.5)

\( f_{m} \) is the frequency parameter, and,

\[
t = |x_s - x| + |x_r - x| - |x_s^* - x_r|,
\]  

(4.6)

and \( x_s^* \) is an “image” point of \( x_s \) to be defined. This is to simulate the actual functional behavior of the migration integrand for a flat reflector with a uniform subsurface velocity.

Both \( x_s \) and \( x_r \) vary on the same unit square area at the top surface \( z = 0 \). For the reflector at \( z = 0.5 \), the image point \( x_s^* \) of \( x_s \) is at the depth \( z = 1 \) and has the same \( x \) and \( y \) coordinates as \( x_s \). Figure 4.3 shows the migration for a vertical line of subsurface points \( x = (x = 1/2, y = 1/2, 0 < z < 1) \) when a different number of \( N = \text{Kmax} \) of grid points (source-receiver pairs) are used for the integration. Here
The migrated image intensity should be zero except near $z = 0.5$ where the reflector position is. As expected, the integration becomes more accurate as the number grid points is increased. The migrated traces vanish (or free of migration noise) below the reflector because all the migration ellipses are above the reflector depth due to causality. Figure 4.4 shows the migrated traces as in Figure 4.3 except that the Hammersley points are used instead of the regular grid points. The same migrations are also performed using Halton points (Figure 4.5). Figures 4.4 and 4.5, show that the Quasi-Monte Carlo methods using Hammersley and Halton points are all much (one order of magnitude or) more efficient than the grid points for the same accuracy. For example, the maximum error (for $0 < z < 0.5$) in the grid point migration with $k_{\text{max}}=160000$ is similar in magnitude to that of the Halton point migration with $k_{\text{max}}=4096$ or $k_{\text{max}}=10000$. So Halton point migration is a factor of 16 to 40 times more efficient for this test. The performance using Hammersley points and shifted Hammersley points (or Wozniakowski points, not shown) are similar to that using Halton points.

4.3 THE FRENCH MODEL

This section presents the 3-D prestack migration results for the French model using the Halton points. The study above for the horizontal reflector model suggests that the above mentioned Quasi-Monte Carlo methods may perform similarly in terms of efficiency. So we use the Halton points for the French model study because Halton points are the most convenient to use.

4.3.1 General description

**Model Dimensions:** It is of more practical interest to see the efficiency of the Quasi-Monte Carlo methods for a model with more realistic source and receiver distribution geometries. For this purpose we use the French model (French, 1974). The top and side views (Figure 4.6(a)) of the model show the physical (in inches) and field-equivalent (in feet) sizes of the model. The model is immersed in a liquid with a field-equivalent velocity of 9160 ft/sec. The top of the model (i.e. the top horizontal reflector) is at a depth of 3430 ft. A Fortran function subroutine is written to compute the synthetic seismogram trace with the locations of a source-receiver pair as input. The seismogram waveform are obtained by time-shifting a Ricker wavelet of 45 Hz. The peak amplitude of the wavelet is set to unity regardless of the source-receiver positions.

**Coordinate System:** Figure 4.6(b) (Figure 10 in French’s paper) uses the same labeling as French who labeled the $x$-direction by “profile numbers” and $y$-direction by “shot numbers”. Although we do not use the same data grid as French did, we will use for convenience the same $x$ and $y$ labels. For example, a source at $x_s = (40, 70)$ is located at French’s 40-th profile location and at French’s 70-th shot location.
Figure 4.3: The migrated traces with various numbers of grid summation points for a centered vertical line of subsurface points in the horizontal reflector model.
Figure 4.4: The migrated traces with various numbers of Hammersley summation points for a vertical line of subsurface points in the horizontal reflector model.
Figure 4.5: The migrated traces with various numbers of Halton summation points for a vertical line of subsurface points in the horizontal reflector model.
Figure 4.6: (a) The French 3-D model with field equivalent dimensions. (b) The coordinate system for the French model. (Taken from French, 1974)
4.3. THE FRENCH MODEL

Fractional label values are allowed. The distance between the adjacent \( x \)-labels is
\[
h_x = (100.0 / 80.0) \times (8213.0 / 89.5) \text{ ft},
\]
and the distance between the adjacent \( y \)-labels is
\[
h_y = (100.0 / 80.0) \times (8212.0 / 93.0) \text{ ft}.
\]
The \( x \)-labels starts from 0 at the top of the liquid surface and increase downward by 1 at every depth interval of \( h_z = h_x / 8 \). So a subsurface point \( x = (40, 70, 250) \) is located at the depth of \( 250 \times h_x \) ft below the liquid surface. The dimensionless coordinate labels \((x, y, z)\) may be converted into \((X, Y, Z)\) measured in units of feet by
\[
X = (-15.5 + 100x/80) \times (8213/89.5) \quad (4.7)
\]
\[
Y = (-25.5 + 100y/80) \times (8212/93) \quad (4.8)
\]
\[
Z = (1/8) \times (100z/80) \times (8212/93). \quad (4.9)
\]

Source-Receiver Geometry:

The sources are distributed over a rectangular grid on the surface (Figure 4.7). \( NS = (ns_x, ns_y) \) specifies the number of source grid points in the \( x \)- and \( y \)-directions respectively. \( X1 = (x_1, y_1) \) specifies the \( x \)- and \( y \)-labels of the first source. \( DX = (dx, dy) \) specifies the source spacings (or increments in the coordinate labels between neighboring sources) in the \( x \)- and \( y \)-directions.

The receivers are distributed over the same rectangular grid points as the sources. For each source, only the receivers within a rectangular window around the source are active. The distances from the four edges of the rectangular window to the source is specified by four integers \( W = (nx_1, nx_2, ny_1, ny_2) \). The total number of active receivers for each source will be \( (nx_1 + nx_2 + 1) \times (ny_1 + ny_2 + 1) \).

When the Halton points are used for migration, the source and receiver coordinates predicted by a Halton point will be rounded to the nearest grid point to determine which source-receiver pair on the data grid should be used. If the receiver is out of the active-receiver-window for the source, the source-receiver pair will not be used. The total number of Halton points used should be much less than the number of data grid points (i.e., the number of sources times the number of active receivers around the source) to insure the quasi-randomness of the points used.

Image Points Under Consideration: For the sake of demonstration, throughout this report we choose not to present the migration results for the entire 3-D volume of the French model. Instead, we will only examine a 2-D slice of the 3-D volume. The 2-D slice is in the vertical plane with \( x = 48 \) (See e.g. Figure 4.10) and roughly coincides with French’s cross section number 7. It is discretized into 120 \( \times \) 150 grid points, with the \( y \)-labels of the grid points ranging between \( y = 1 \) and \( y = 120 \), and \( x \)-labels ranging between \( z = 200 \) and \( z = 349 \). The migrated image will be represented by \( M(i, j) \), with \( i = 1, 2, ..., 120 \) and \( j = 200, 201, ..., 349 \).

Measure of Migration Error: We do not know values of \( M(i, j) \) for an exact migration. But we do know that above and below the reflector, \( M(i, j) \) should be zero for an ideal migration. Away from the reflector, any value other than zero represents noise. Coherent noise is worse than random noise because coherent noise interferes
Figure 4.7: The source-receiver geometry parameters for the French model.
with interpretation. We use the following to define the coherent error of the migrated slice:

\[
B = \left(1/M_{\text{rms}}\right)^3 \sum_j \sum_{i=2}^{i=119} \max_{k=-8}^{k=+8} M(i-1, j-k)M(i, j)M(i+1, j+k) \tag{4.10}
\]

where \(M_{\text{rms}}\) is the rms value of the entire migrated image intensity, \(j\) runs from \(j = 200\) to \(j = 229\) (above the reflector) and then from \(j = 315\) to \(j = 349\) (below the reflector). The migration error \(B\) decreases as the migration noise decreases or as the noise coherency decreases. The \(B\) so defined may not produce a meaningful measure the number of points \(N\) used for migration is small. This is because the \(B\) values tend to fluctuate severely with \(N\) when \(N\) is small. The relative error, which ignores the coherency of the migration noise, will also be used to the measure the quality of migration,

\[
E = \left(1/M_{\text{rms}}\right)\left\{\sum_j \sum_{i=2}^{i=119} [M(i, j)]^2\right\}^{1/2}. \tag{4.11}
\]

Note that both \(B\) and \(E\) are dimensionless numbers do not change (neither will the interpretability) if the entire migrated image is scaled by an constant factor.

There are three types of noises in the migrated section: (1) Noise due to the presence of random noise in the data; (2) Noise (or view angle artifacts) due to the finite coverage by the source-receiver distribution; (3) Noise due to the statistical nature of the migration process. In this study, it is the third type of noise, or migration noise, that we are most concerned with. Migration relies on stacking and cancellation to produce near-zero image intensities in regions away from the scatterers and reflector. Incomplete cancellations with finite number of stackings are the origin of migration noise. The objective of this study is to see if Quasi-Monte Carlo methods are more efficient in minimizing the migration noise than the migration using regular grid points. In defining the migration error \(B\) and \(E\) for a migrated section, the three types of noises can not be separated. So they all contribute to the values of \(B\) and \(E\).

We will use \(B_0\), \(B_1\), \(B_2\), and \(B_3\) to denote the coherent error \(B\) when random noises (see equation 4.12 below) with peak-to-peak amplitudes of \(PP = 0\), 0.2, 0.5, and 1.0, respectively, are added to the synthetic traces. \(B_n\) will denote the coherent error \(B\) for the migrated image of pure noise seismograms with \(PP = 1.0\).

### 4.3.2 Examples of synthetic seismograms

Figure 4.8a shows the zero-offset seismogram traces for a line of 120 sources located at \(z = 0\), \(x = 48\), equi-spaced between \(y = 1\) and \(y = 120\), and directly above the imaging slice. Each source has a single coincident receiver. To verify the validity of the our Fortran subroutine for computing seismograms, the 2-D zero-offset data is migrated (Figure 4.8b) and compared with French’s result (French’s Figure 13, results for cross-section no. 7). The two migration results are very similar. Note that \(y\) increases from
Table 4.1: Data geometries and migration errors for the 3-D zero-offset migration using various numbers of regular grid points $N$. $X1 = (1,1)$ and $W = (0,0,0,0)$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>180</th>
<th>720</th>
<th>2880</th>
<th>11520</th>
<th>46080</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NS$</td>
<td>12.15</td>
<td>24.30</td>
<td>48.60</td>
<td>96.120</td>
<td>192.240</td>
</tr>
<tr>
<td>$DX$</td>
<td>8.8</td>
<td>4.4</td>
<td>2.2</td>
<td>1.1</td>
<td>0.5, 0.5</td>
</tr>
<tr>
<td>$B_0$</td>
<td>13648.8</td>
<td>10302.3</td>
<td>8307.33</td>
<td>4417.70</td>
<td>436.197</td>
</tr>
<tr>
<td>$E_0$</td>
<td>89.0589</td>
<td>84.219</td>
<td>78.2699</td>
<td>59.1968</td>
<td>25.0795</td>
</tr>
</tbody>
</table>

left to right in Figure 4.8 while $y$ increases from right to left in French’s figure. Also note that the ghost reflections in French’s figure for the tank-modeled data are not present in Figure 4.8a for the synthetic data set. Figure 4.9a shows a common-shot-point gather with the source located at $(48,60,0)$ and 120 receivers evenly spaced above the image slice as in Figure 4.8a. Figure 4.9b gives an expanded view of the center region in Figure 4.8a. Figure 4.9c is the same as Figure 4.9a except that random noise (with peak-to-peak amplitude equal to 1) has been added. To study the effects of noise on the migration quality and efficiency, the following form of noise are sometimes added to the synthetic traces:

$$n_i(x_s, x_r) = PP \times (\text{RAND}(i_{seed} + i) - 0.5),$$  \hspace{1cm} (4.12)

where $n_i$ is the noise value at the $i$-th data sample for the trace with source $x_s$ and receiver $x_r$, RAND($i$) is the $i$-th random number (between 0 and 1). The random number seed index $i_{seed}$ is given by the integer part of the number $0.004 \times (|X_s| + |Y_s| + |X_r| + |Y_r|) \times NT$, where the $X$'s and $Y$'s are the source and receiver coordinates in feet and $NT$ is number of samples per trace. The scaler multiplier $PP$ specifies the peak-to-peak level of the noise. Figure 4.9d gives an expanded view of the center region in Figure 4.8c, showing the coherency and high frequency nature of the noise (with $PP = 1$ and $NT = 2000$). In what follows, the synthetic data will be assumed noise free ($PP = 0.0$) unless explicitly stated otherwise.

### 4.3.3 3-D post-stack migration

Figure 4.10 shows the 3-D post-stack migration using various numbers of regular grid points. The data geometries and migration errors are listed in Table 4.1.

Figure 4.11 shows the 3-D post-stack migration using various numbers of Halton points. Note that number of Halton points used should be much less than the number of data points $384 \times 480$ available. The data geometries and migration errors are listed in Table 4.2.

As can be seen from both the migrated sections and the values of $B_0$ and $E_0$ in Tables 4.1 and 4.2, for a given quality of migration, the number of points required for the 3-D post-stack migration are comparable whether the regular grid points or the Halton points are used. Thus there is no efficiency gain using the Quasi-Monte Carlo
4.3. THE FRENCH MODEL

Figure 4.8: The 2-D zero-offset data migration. (a) Raw data; (b) After migration.

<table>
<thead>
<tr>
<th>$N$</th>
<th>720</th>
<th>1440</th>
<th>2880</th>
<th>5760</th>
<th>11520</th>
<th>23040</th>
<th>46080</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_0$</td>
<td>9784.34</td>
<td>10280.60</td>
<td>8037.70</td>
<td>6531.80</td>
<td>3751.08</td>
<td>1640.45</td>
<td>735.398</td>
</tr>
<tr>
<td>$E_0$</td>
<td>82.479</td>
<td>83.5608</td>
<td>76.7758</td>
<td>72.0500</td>
<td>59.8425</td>
<td>45.3999</td>
<td>33.4659</td>
</tr>
</tbody>
</table>

Table 4.2: Data geometries and migration errors for 3-D zero-offset migration using various numbers of Halton points $N$. $NS = (384, 480)$, $X1 = (1, 1)$, $DX = (0.25, 0.25)$, and $W = (0, 0, 0, 0)$. 
Figure 4.9: CSP gathers with and without noise. (a) A CSP gather with 120 receivers directly above the image slice. The source is coincident with 60-th receiver; (b) An expanded view of (a); (c) Same as (a) except that noise with peak-to-peak value PP = 1 has been added; (d) An expanded view of (c).
Figure 4.10: 3-D zero-offset migration using various numbers of regular grid points. (a) 720 points. (b) 2880 points. (c) 11520 points. (d) 46080 points.
Figure 4.11: 3-D zero-offset migration using various numbers of Halton points. (a) 720 points. (b) 2880 points. (c) 11520 points. (d) 46080 points.
4.3. THE FRENCH MODEL

<table>
<thead>
<tr>
<th>N</th>
<th>1620</th>
<th>6480</th>
<th>25920</th>
<th>103680</th>
</tr>
</thead>
<tbody>
<tr>
<td>NS</td>
<td>12,15</td>
<td>24,30</td>
<td>48,60</td>
<td>96,120</td>
</tr>
<tr>
<td>DX</td>
<td>8,8</td>
<td>4,4</td>
<td>2,2</td>
<td>1,1</td>
</tr>
<tr>
<td>B₀</td>
<td>10361.30</td>
<td>9032.04</td>
<td>6194.61</td>
<td>1837.09</td>
</tr>
<tr>
<td>B₁</td>
<td>11609.6</td>
<td>10069.07</td>
<td>7005.87</td>
<td>3017.48</td>
</tr>
<tr>
<td>B₂</td>
<td>12529.6</td>
<td>11851.9</td>
<td>9189.36</td>
<td>7086.97</td>
</tr>
<tr>
<td>B₃</td>
<td>12619.6</td>
<td>13007.9</td>
<td>11329.5</td>
<td>11678.7</td>
</tr>
<tr>
<td>B₄</td>
<td>12350.6</td>
<td>13473.3</td>
<td>13132.0</td>
<td>15398.5</td>
</tr>
<tr>
<td>E₀</td>
<td>83.3882</td>
<td>80.1096</td>
<td>68.0074</td>
<td>43.5962</td>
</tr>
<tr>
<td>E₁</td>
<td>84.7805</td>
<td>81.338</td>
<td>70.2098</td>
<td>51.9422</td>
</tr>
<tr>
<td>E₂</td>
<td>87.5523</td>
<td>84.3659</td>
<td>76.6260</td>
<td>70.1677</td>
</tr>
<tr>
<td>E₃</td>
<td>89.1986</td>
<td>86.7210</td>
<td>82.4603</td>
<td>83.1537</td>
</tr>
<tr>
<td>E₄</td>
<td>90.1379</td>
<td>88.1906</td>
<td>86.7751</td>
<td>91.3579</td>
</tr>
</tbody>
</table>

Table 4.3: Data geometries and migration errors for 3-D prestack migration with linear receiver arrangement using various numbers N of regular grid points. X₁ = (1,1) and W = (0,0,4,4).

method as was concluded by Sikorski and Schuster (1991) in the previous study of Hammersley migration.

4.3.4 3-D prestack, linear receiver arrangement

Figure 4.12 shows the 3-D prestack migration using regular grid points with a linear arrangement of 9 active receivers for each source. For the images in Figures 4.12(e) and (f), noise of PP = 0.5 was added to the seismogram traces. The data geometries and migration errors are listed in Table 4.3.

Figure 4.13 shows the 3-D prestack migration using various numbers of Halton points. For each source, the Halton points are used to randomly pick from a line of 41 actively receivers. The source grid spacing is DX = (0.25,0.25). For the images in Figures 4.13(e) and (f), noise of PP = 0.5 was added to the seismogram traces. The data geometries and migration errors are listed in Table 4.4. Here we allow more active receivers per source for the Halton migration than that for the grid point migration to insure the randomness of the Halton-selected points. This leads to somewhat different view angle coverage in the two types of migrations. The migration errors B and E will be dominated by the statistical migration noise over the view angle artifact noise unless the number of points used is very large, the efficiency gain analysis will not be affected much by the slight view angle effects.

As can been seen from both the migrated sections and the B₀ values in Tables 4.3 and 4.4, for a given quality of migration the number of points required for this 3-D prestack migration of noise-free data using regular grid points is about six times more than that required using the Halton points. For example, 25920 regular grid
Figure 4.12: 3-D prestack migration with linear receiver arrangement using various numbers of regular grid points. (a) 1620 points. (b) 6480 points. (c) 25920 points. (d) 103680 points. (e) same as (b) except that noise of PP=0.5 was added to the seismograms. (f) same as (c) except that noise of PP=0.5 was added to the seismograms.
Figure 4.13: 3-D prestack migration with linear receiver arrangement using various numbers of Halton points. (a) 1620 points. (b) 6480 points. (c) 25920 points. (d) 103680 points. (e) same as (b) except that noise of $PP = 0.5$ was added to the seismograms. (f) same as (c) except that noise of $PP = 0.5$ was added to the seismograms.
Table 4.4: Data geometries and migration errors for 3-D prestack migration with linear receiver arrangement using various numbers \( N \) of Halton points. \( NS = (384,480) \), \( X1 = (1,1) \), \( DX = (0.25,0.25) \), and \( W = (0,0,20,20) \).

<table>
<thead>
<tr>
<th>( N )</th>
<th>1620</th>
<th>3240</th>
<th>6480</th>
<th>12960</th>
<th>25920</th>
<th>103680</th>
</tr>
</thead>
<tbody>
<tr>
<td>( B_0 )</td>
<td>8689.58</td>
<td>7797.44</td>
<td>5486.40</td>
<td>3721.46</td>
<td>2364.86</td>
<td>808.250</td>
</tr>
<tr>
<td>( B_1 )</td>
<td>10385.98</td>
<td>9280.90</td>
<td>7205.91</td>
<td>5174.41</td>
<td>3543.73</td>
<td>1148.80</td>
</tr>
<tr>
<td>( B_2 )</td>
<td>12221.6</td>
<td>11078.4</td>
<td>10196.0</td>
<td>8495.97</td>
<td>6913.58</td>
<td>2601.74</td>
</tr>
<tr>
<td>( B_3 )</td>
<td>13008.4</td>
<td>11789.4</td>
<td>11916.6</td>
<td>11103.6</td>
<td>10551.26</td>
<td>5573.90</td>
</tr>
<tr>
<td>( B_4 )</td>
<td>13372.6</td>
<td>12117.4</td>
<td>12773.1</td>
<td>12861.0</td>
<td>13610.0</td>
<td>10586.7</td>
</tr>
<tr>
<td>( E_0 )</td>
<td>77.4893</td>
<td>75.1303</td>
<td>66.8978</td>
<td>58.9988</td>
<td>50.2430</td>
<td>34.0690</td>
</tr>
<tr>
<td>( E_1 )</td>
<td>80.4019</td>
<td>78.3535</td>
<td>71.7986</td>
<td>64.6303</td>
<td>56.3478</td>
<td>38.6581</td>
</tr>
<tr>
<td>( E_2 )</td>
<td>85.0510</td>
<td>83.8450</td>
<td>80.6904</td>
<td>76.6840</td>
<td>70.8577</td>
<td>52.9479</td>
</tr>
<tr>
<td>( E_3 )</td>
<td>87.2681</td>
<td>86.5911</td>
<td>85.5649</td>
<td>84.3955</td>
<td>82.0860</td>
<td>70.0268</td>
</tr>
<tr>
<td>( E_n )</td>
<td>88.3340</td>
<td>87.9304</td>
<td>88.0907</td>
<td>88.9327</td>
<td>89.4556</td>
<td>87.852</td>
</tr>
</tbody>
</table>

points are required (Figure 4.12) to reduce the migration error to \( B_0 = 6194.61 \). This migration quality can be achieved by using about 4500 Halton points, with an efficiency gain factor of \( 25920/4500 \approx 6 \). If the \( E_0 \) measure is used (ignoring the coherency of the migration noise), the efficiency gain is somewhat smaller, approximately \( 25920/6000 \approx 4.5 \).

In the above efficiency analysis, we did not use cases with very small number of migration points, say 1620, because the error estimates \( B \) and \( E \) tend to fluctuate severely with the number of points and may not be good error measures. Besides, the migration image qualities are so poor for such cases, it is unimportant how efficiently they can be obtained anyway.

As the noise level increases, the migration efficiency gain using Halton points decreases. This is because there is no efficiency gain when pure noise is migrated (see \( E_n \) and \( B_n \) values in Tables 4.3 and 4.4). The \( E_n \) and \( B_n \) values do not vary much with either the number or the type of points used for migration. Examination of the \( E_1 \) and \( B_1 \) values shows that the efficiency gain drops to about \( 25920/6500 \approx 4 \) when the noise peak-to-peak amplitude \( PP = 0.20 \) is 20% of the signal amplitude. The efficiency gain drops to about 2 when \( PP = 0.50 \).

The efficiency gain factor decreases as the number of points \( N \) for migration increases and varies with the data noise level depending which type of noise in the migrated section dominates. As the number of migration points further increases, the migration noise will become small relative to migrated data noise and the view angle artifacts. So a further increase of points in the same data region will not improve the image and is therefore a computation waste. In \( N = 518400 \) cases in Tables 4.14, the \( B \) and \( E \) values will be very poor measure of the statistical noise due to migration process, and are no longer valid for efficiency analysis.
### 4.3. THE FRENCH MODEL

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>8100</td>
<td>32400</td>
<td>129600</td>
<td>518400</td>
</tr>
<tr>
<td>$NS$</td>
<td>12.15</td>
<td>24.30</td>
<td>48.60</td>
<td>96.120</td>
</tr>
<tr>
<td>$DX$</td>
<td>8.8</td>
<td>4.4</td>
<td>2.2</td>
<td>1.1</td>
</tr>
<tr>
<td>$B_0$</td>
<td>6768.24</td>
<td>6541.64</td>
<td>4079.98</td>
<td>495.479</td>
</tr>
<tr>
<td>$B_1$</td>
<td>8642.29</td>
<td>7308.30</td>
<td>4908.67</td>
<td>1050.16</td>
</tr>
<tr>
<td>$B_2$</td>
<td>12629.0</td>
<td>9463.00</td>
<td>7919.22</td>
<td>3936.85</td>
</tr>
<tr>
<td>$B_3$</td>
<td>15385.3</td>
<td>11836.6</td>
<td>11978.2</td>
<td>9340.95</td>
</tr>
<tr>
<td>$B_n$</td>
<td>16805.2</td>
<td>14085.1</td>
<td>16167.5</td>
<td>16229.6</td>
</tr>
<tr>
<td>$E_0$</td>
<td>72.0266</td>
<td>72.0419</td>
<td>58.5572</td>
<td>26.8319</td>
</tr>
<tr>
<td>$E_1$</td>
<td>76.3500</td>
<td>73.4497</td>
<td>62.1162</td>
<td>36.1747</td>
</tr>
<tr>
<td>$E_2$</td>
<td>83.6012</td>
<td>78.2071</td>
<td>73.1852</td>
<td>57.9756</td>
</tr>
<tr>
<td>$E_3$</td>
<td>87.7818</td>
<td>83.5562</td>
<td>84.6616</td>
<td>77.6280</td>
</tr>
<tr>
<td>$E_n$</td>
<td>89.9036</td>
<td>88.2877</td>
<td>94.2051</td>
<td>93.3682</td>
</tr>
</tbody>
</table>

Table 4.5: Data geometries and migration errors for 3-D prestack migrated image using various numbers $N$ of regular grid points, with 5 lines of receivers for each source. $X1 = (1,1)$ and $W = (2,2,4,4)$.

### 4.3.5 3-D prestack, several receiver lines

Figure 4.14 shows the 3-D prestack migrated image using various numbers of regular grid points with 5 lines of (each line with 9 active) receivers for each source. For the images in Figures 4.14(e) and (f), noise of $PP = 0.5$ was added to the seismogram traces. The data geometries and migration errors are listed in Table 4.5.

Figure 4.15 shows the 3-D prestack migration using various numbers of Halton points. For each source, the Halton points are used to randomly pick from $5 \times 41$ actively receivers. The source grid spacing is $DX = (1.0,1.0)$. For the images in Figures 4.15(e) and (f), noise of $PP = 0.5$ was added to the seismogram traces. The data geometries and migration errors are listed in Table 4.6.

As can be seen from both the migrated sections and the $B_0$ and $E_0$ values, for a given quality of migration, the number of points required for this 3-D prestack migration of noise-free data using regular grid points is about an order of magnitude more points than required by Halton migration. For example, 129600 regular grid points are required (Figure 4.14) to reduce the migration error to $B_0 = 4079.98$. This migration quality can be achieved by using about 14000 Halton points, with an efficiency factor of $129600/14000 \approx 9$. If the $E_0$ measure is used, the efficiency factor is about $129600/15000 \approx 8$. Again we notice that as the data noise level increases, the Halton migration efficiency gain factor decreases. For the data noise level $PP = 0.5$, the factor drops to about $129600/19000 \approx 7$. 
Figure 4.14: 3-D prestack migration with 5 lines of active receivers per source, using various numbers of regular grid points. (a) 8100 points. (b) 32400 points. (c) 129600 points. (d) 518400 points. (e) same as (b) except that noise of $PP = 0.5$ was added to the seismograms. (f) same as (c) except that noise of $PP = 0.5$ was added to the seismograms.
Figure 4.15: 3-D prestack migration using various numbers of Halton points, with 5 lines of active receivers per source. (a) 8100 points. (b) 32400 points. (c) 129600 points. (d) 518400 points. (e) same as (b) except that noise of $PP = 0.5$ was added to the seismograms. (f) same as (c) except that noise of $PP = 0.5$ was added to the seismograms.
Table 4.6: Data geometries and migration errors for 3-D prestack migrated image using various numbers $N$ of Halton points, with 5 lines of receivers for each source. $NS = (96, 120)$, $X1 = (1, 1)$, $DX = (1, 1)$, and $W = (2, 2, 20, 20)$.

<table>
<thead>
<tr>
<th>N</th>
<th>8100</th>
<th>15000</th>
<th>32400</th>
<th>60000</th>
<th>129600</th>
<th>518400</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_0$</td>
<td>5299.94</td>
<td>3802.90</td>
<td>1945.98</td>
<td>1028.18</td>
<td>600.679</td>
<td>394.674</td>
</tr>
<tr>
<td>$B_1$</td>
<td>6796.39</td>
<td>5218.94</td>
<td>2782.62</td>
<td>1532.00</td>
<td>837.237</td>
<td>447.709</td>
</tr>
<tr>
<td>$B_2$</td>
<td>9675.40</td>
<td>8549.21</td>
<td>5446.39</td>
<td>3692.47</td>
<td>1968.76</td>
<td>719.284</td>
</tr>
<tr>
<td>$B_3$</td>
<td>11682.7</td>
<td>11505.8</td>
<td>8900.66</td>
<td>7642.56</td>
<td>5037.69</td>
<td>1781.81</td>
</tr>
<tr>
<td>$B_n$</td>
<td>12832.3</td>
<td>13644.2</td>
<td>12316.1</td>
<td>13014.4</td>
<td>12989.6</td>
<td>13268.1</td>
</tr>
<tr>
<td>$E_0$</td>
<td>66.9235</td>
<td>59.4045</td>
<td>47.9673</td>
<td>38.1634</td>
<td>30.6923</td>
<td>24.6930</td>
</tr>
<tr>
<td>$E_1$</td>
<td>70.4788</td>
<td>64.1045</td>
<td>52.9492</td>
<td>42.9658</td>
<td>34.0735</td>
<td>26.0996</td>
</tr>
<tr>
<td>$E_2$</td>
<td>78.6583</td>
<td>75.3400</td>
<td>66.5640</td>
<td>57.6973</td>
<td>45.9628</td>
<td>31.8322</td>
</tr>
<tr>
<td>$E_3$</td>
<td>84.1717</td>
<td>83.6041</td>
<td>78.9801</td>
<td>73.7683</td>
<td>63.3902</td>
<td>44.6969</td>
</tr>
<tr>
<td>$E_n$</td>
<td>87.4497</td>
<td>88.8434</td>
<td>88.2309</td>
<td>88.3834</td>
<td>87.5659</td>
<td>88.3329</td>
</tr>
</tbody>
</table>

Table 4.7: Data geometries and migration errors for 3-D prestack migrated image using various numbers $N$ of regular grid points, with an area of $7 \times 7$ active receivers per source. $X1 = (1, 1)$ and $W = (3, 3, 3, 3)$.

<table>
<thead>
<tr>
<th>N</th>
<th>3675</th>
<th>14700</th>
<th>58800</th>
<th>235200</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NS$</td>
<td>12,15</td>
<td>24,30</td>
<td>48,60</td>
<td>96,120</td>
</tr>
<tr>
<td>$DX$</td>
<td>8.8</td>
<td>4.4</td>
<td>2.2</td>
<td>1.1</td>
</tr>
<tr>
<td>$B_0$</td>
<td>3615.91</td>
<td>6535.61</td>
<td>2002.96</td>
<td>430.964</td>
</tr>
<tr>
<td>$E_0$</td>
<td>58.6483</td>
<td>70.8535</td>
<td>47.2922</td>
<td>25.1940</td>
</tr>
</tbody>
</table>

4.3.6 3-D prestack, areal receivers

Figure 4.16 shows the 3-D prestack migration using regular grid points with an areal distribution ($7 \times 7$ active) active receivers for each source. The data geometries and migration errors are listed in Table 4.7.

Figure 4.17 shows the 3-D prestack migration using various numbers of Halton points. For each source, the Halton points is used to pick randomly from $21 \times 61$ active receivers. The source grid spacing is $DX = (1.0, 1.0)$. The data geometries and migration errors are listed in Table 4.8.

As can be seen from both the migrated sections and the $B_0$ and $E_0$ values, the number of points required for this 3-D prestack migration using regular grid points for a given quality of migration is about six times more than that required using the Halton points. For example, 58800 regular grid points are required (Figure 4.16) to reduce the migration error to $B = 2002.96$. This migration quality can be achieved by using about 10000 Halton points, with an efficiency factor of 6. Note that the source coverage $NS = (40, 120)$ for the migrations in Figure 4.17 is smaller than those in Figure 4.16. The view angle artifacts will be correspondingly larger.
3. THE FRENCH MODEL

Figure 4.16: 3-D prestack migration using various numbers $N$ of regular grid point, with an area of $7 \times 7$ active receivers per source. (a) 3675 points. (b) 14700 points. (c) 58800 points. (d) 235200 points.

<table>
<thead>
<tr>
<th>$N$</th>
<th>4000</th>
<th>8000</th>
<th>16000</th>
<th>40000</th>
<th>64000</th>
<th>100000</th>
<th>235200</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_0$</td>
<td>3635.32</td>
<td>2337.69</td>
<td>1276.09</td>
<td>561.471</td>
<td>406.60</td>
<td>341.861</td>
<td>286.917</td>
</tr>
<tr>
<td>$E_0$</td>
<td>59.082</td>
<td>50.2510</td>
<td>40.4572</td>
<td>30.4122</td>
<td>26.6120</td>
<td>24.563</td>
<td>21.6038</td>
</tr>
</tbody>
</table>

Table 4.8: Data geometries and migration errors for 3-D prestack migrated image using various numbers $N$ of Halton points, with an area of $21 \times 61$ active receivers per source. $NS = (40,120)$, $X_1 = (1,1)$, $DX = (1,1)$, and $W = (10,10,30,30)$. 
Figure 4.17: 3-D prestack migration using various numbers of Halton points, with an area of $21 \times 61$ active receivers per source. (a) 4000 points. (b) 15000 points. (c) 64000 points. (d) 235200 points.
4.4 CONCLUSION AND DISCUSSIONS

Results from the synthetic model migration for the French model suggest that the Quasi-Monte Carlo methods may improve the efficiency of 3-D prestack migration by a factor 5 or so for noise-free or data with low level random or high frequency coherent noise. For a given 3-D data set with good S/N ratio, if regular grid 3-D prestack migration produces good quality images, synthetic tests suggest that the same quality migration or better (with less coherent noise or ghost images) can be obtained with 1/5 of the computation by using a quasi-random subset of the data. For 3-D post-stack migration, the Quasi-Monte Carlo methods are not advantageous. The Halton migration efficiency gain decreases as the (random or high frequency) data noise level increases. The Halton migration efficiency may not suffer from coherent noises (such as reflection multiples) with frequencies comparable to the dominant frequency of the signal. This is because the frequency content is expected to dictate the computational complexity of the seismogram functions to be migrated. However further study is needed for coherent noise of various frequency spectra. Further tests are being conducted to investigate (1) the relative advantages of the Quasi-Monte Carlo methods. Preliminary study show the various near-optimal points produce similar efficiency gain. Halton points are the one of the easier sets to use. (2) Trace interpolation to better honor the predicted quasi-random points. (2) optimal 3-D data design. (3) Parallel computation design. Future work will also include testing the migration algorithm on a tank-model or a field data set.

4.5 ACKNOWLEDGEMENT

We would like to thank Gerard Schuster, Kris Sikorski, Sia Hassanzadeh, and Bee Bednar for helpful suggestions. We would also like to thank the Reservoir Imaging and Characterization Group at the Amoco Research Center in Tulsa for computer time.

4.6 REFERENCES


French, W. S., 19974, Two-Dimensional and Three-Dimensional Migration of Model-Experiment Reflection Profiles, Geophysics, 39, p265-277.

algorithm to 3-D migration, University of Utah Tomography Development Project, 1991 final report, p234-259.


Report 5

A Constrained Kirchhoff Migration of Crosswell Seismic Data

Fuhao Qin

5.1 ABSTRACT

Crosswell migration imaging is affected by various migration artifacts due to limited aperture coverage and large incidence angle rays. This paper shows that some well-posed constraints in the migration process can significantly alleviate these artifacts. A reflector dip constraint significantly decreases the migration artifacts, and elimination of large incidence angle rays alleviates the resolution loss caused by wavelet stretching and postcritical phase change. The migrated results suggest that crosswell migration reveals much more lithologic detail relative to the surface seismic image.

5.2 INTRODUCTION

Many studies (Mason, 1981; Wong et al., 1983; Bregman et al., 1989; Phillips and Fehler, 1991; Luo and Schuster, 1989) have been carried out to image the earth structure via a crosswell seismic survey. The most widely studied and used technique is traveltime tomography. However, Williamson and Worthington (1993) showed that the resolution of traveltime tomography is $\sqrt{\lambda D}$ for noise free data, where $\lambda$ is the wavelength and $D$ is the source receiver distance. If the traveltime error and limited ray coverage are taken into account then the velocity resolution from traveltime tomography is even lower. It is almost certain that the first arrival traveltime tomogram alone can not provide the maximum lithological resolution we expect from a crosswell experiment. More information must be used to obtain the fine structure of the earth.
In this regards, two techniques are very promising. One is full waveform inversion and the other is reflection migration. Waveform inversion, unlike traveltime inversion, finds the model that best fits the entire seismic record. This enables the inversion to reconstruct not only the low-frequency features of the model that affect the traveltimes, but also the high-frequency information that determines the reflectivity. Tarantola (1987) provided the basis of waveform inversion, while the efforts of Luo and Schuster (1989) and Zhou et al. (1993) applied it to real crosswell data. However, full waveform inversion does have some drawbacks. First it is sensitive to the initial guess model and, more importantly, it requires a great deal of computer memory and CPU time. Thus much work is being devoted to make it more practical.

On the other hand, reflection migration with its effectiveness and efficiency offers the most promise for practical imaging. Besides these general features for migration, there are several reasons for which crosswell reflection migration may produce higher resolution images of the earth than the surface migration:

- In crosswell migration, the migration velocity is taken from the first arrival travelt ime tomogram or the sonic log data. These velocities are more accurate than those obtainable for surface seismic data.

- High frequency sources can be used in a crosswell experiment, which is sometimes difficult to implement in surface seismic experiments.

- The rays travel a relatively shorter distance compared to a surface seismic survey. This, combined with higher frequency sources and an accurate background velocity model, largely decreases the uncertainty of the image position.

- No statics corrections are needed since most of the sources and receivers are placed well beneath the weathered zone. This compares favorably to surface seismic data processing where statics problems are always troublesome.

Several authors (Hu et al., 1988; Findlay et al., 1991 and Lazaratos et al., 1992) have discussed crosswell reflection imaging methods. Hu et al. (1988) applied prestack reverse-time migration to synthetic and scale-model crosswell data. The disadvantages of reverse-time migration are: First, it is time consuming since it solves the wave equation by a finite-difference method. When the frequency is very high, the stability and dispersion criterion of the finite-difference scheme requires a tremendous computer time and memory. Second, it needs extensive preprocessing to make the data spacing equal to the finite-difference grid. Findlay et al. (1991) successfully applied a generalized Kirchhoff integral migration method to crosswell seismic for opencast coal exploration. Although successful, their tests were limited to only the near offset, shallow depth coal exploration problems. Lazaratos et al. (1992) used a variation of CDP stacking for crosswell reflection imaging. While the method is very efficient, it may need human interaction with the computer when there are structures with various dipping angles.
5.3. **CONSTRAINED KIRCHHOFF INTEGRAL MIGRATION METHOD**

Despite the advantages of crosswell reflection migration, several problems are also involved in crosswell reflection imaging:

- Crosswell reflections can be obscured by a mixture of interfering wave types. such as direct waves, S-waves, converted waves, multiples and tube waves from both wells.
- Crosswell reflection surveys have only limited source/receiver aperture.
- The ray coverage is very uneven in terms of both the number of visiting rays and the ray incidence angles.
- Rays with extremely large incidence angles exist, which will stretch the source wavelet and cause loss of resolution in the migrated section.

None of the existing migration methods can deal with those problems automatically. However, this paper will show that some well designed constraints to the migration method will alleviate the above problems in crosswell data migration. Here, we use a constrained Kirchoff integral migration method since the Kirchoff integral migration (Schneider, 1978) is the most flexible migration method and can be modified in various ways to adapt different situations.

In the following text, we first show the methodology of a constrained Kirchoff integral migration and then discuss the effects of the constraints introduced by applying the method to an Exxon real data set from a test site near Friendswood, Texas (Chen et al., 1990).

### 5.3 CONSTRAINED KIRCHHOFF INTEGRAL MIGRATION METHOD

Our migration method is based on the Kirchoff integral method (Schneider, 1978) and we use a finite-difference solution to the eikonal equation (Qin, et al., 1992) for the travelt ime calculations. The migration formula we used for a single source is an approximation of equation (7) in Timoshin (1970):

\[
IM(r) = \sum_{n=0}^{N} \cos(\theta) \frac{1}{r_{eff}} U(z_n, t_r + t_s)
\]  

(5.1)

where IM(r) is the image at r, \( r_{eff} \) is the sum of the distance from source to the image point at r and from the image point to the receiver. N is the total number of receivers and \( \theta \) is the angle between the horizontal direction and the ray direction at the receiver. \( z_n \) is the depth of the \( n^{th} \) receiver, \( t_r \) is the travelt ime from the image point to the receiver, and \( t_s \) is the travelt ime from the image point to the source.

Although not all problems in crosswell migration can be avoided, some of them can be alleviated prior to and during the migration process. The following techniques are used to achieve this goal.
5.3.1 Raw data processing

Reflections in crosswell seismograms are usually concealed by some other wave modes or noise. Data preprocessing is therefore needed to extract the reflection signals. The usual procedure is first to remove the tube waves and then to remove the direct waves. If necessary, up- and down-going wave separation and/or P- and S-wave separation are also performed. The usual processing procedure is shown on the flow chart in Figure 5.1. Please refer to Cai and Schuster (1993) for detailed processing procedure for the data used in this paper.

5.3.2 Incidence angle constraint

Due to the geometry of the crosswell experiment, reflection rays can have a very large incidence angle. These rays can be harmful for two reasons. One is that postcritical angle incidence rays will have a phase change (Aki and Richards, 1980, p. 155-163) which can blur the stacked image. The other is that larger incidence rays will stretch the wavelet in the seismogram. This will smear the migration image and cause resolution loss. Figure 5.2 shows the wavelet stretch effect where S and R indicate the source receiver positions, respectively; t is the traveltime of the reflection arrival; D is half of the well separation; H is the distance between the reflector and the line that connects the source and the receiver. It is clear that

\[ \frac{\partial t}{\partial H} = \frac{2}{v} \frac{H}{\sqrt{D^2 + H^2}} \]  \hspace{1cm} (5.2)

where v is the velocity of the medium. Equation 5.2 indicates that for a reflection ray path shown in Figure 5.2, the traveltime does not change much with H when H is small. This causes the wavelet stretching in migration (small H corresponds to a large incidence angle).

To see the effect of the above two factors, we did the following test. Figure 5.3a shows a simple two layer model. The depth of the model is 300 meters; the width of the model is 120 meters and the layer interface is located at 150 meters depth. Figure 5.3b shows a common shot gather (CSG) with the shot at a depth of 75 meters in the left edge and all receivers are along the right edge. The CSG is calculated by a fourth order finite-difference solution to the acoustic wave equation. Since the velocities for the upper and lower layers are 2000 m/s and 3200 m/s, respectively, the receiver at a depth of 75 meters records the critical reflection. All reflections received by receivers above/below 75 meters are pre-critical/postcritical reflection. We do see a slight change in the wavelet shape. However, the change is not dramatic. It will not affect the migration too much. Figure 5.3d which is part of the migrated section of the post-critical reflections supports this observation. The reflections are basically in phase. However, Figure 5.3d does have a fat image due to the wavelet stretching for large incidence angle reflections.
Figure 5.1: Data processing flow chart (Courtesy of Cai and Schuster, 1993).
Figure 5.2: Wavelet stretch effect by large incidence angle ray.

To alleviate this problem, rays with an incidence angle larger than 75 degrees are simply eliminated. This will be referred to as the incidence angle constraint.

5.3.3 Dip angle constraint

Because of the limited aperture coverage for the crosswell experiment, the reverse-time extrapolation of the recorded wave fields can not be sharply focused on the reflector as it is supposed to be. Instead artifacts will appear in the migration image as migration smiles in the surface seismic migration. It can have any appearance in a crosswell situation. To decrease the migration artifacts, we eliminate image points through which the presumed reflector has a dip angle larger than a predetermined value. This value is based on well log and other geologic information and can be updated according to the migrated image. This will be referred to as the dip angle constraint.

The presumed reflector dip angle ($\alpha$) and the incidence angle ($\beta$) can be calculated in different ways. The most straightforward method is to use the geometrical relations shown in Figure 5.4 after the raypaths are found. In this case, the rays that arrive on and leave from the reflector are considered as first arrival rays and can be found either by raytracing or by finite-difference solutions to the eikonal equation.
5.3. CONSTRAINED KIRCHHOFF INTEGRAL MIGRATION METHOD

\( V_1 = 2000 \text{ m/s} \)
\( \rho = 2.2 \text{ g/cm}^3 \)

\( V_2 = 3200 \text{ m/s} \)
\( \rho = 2.2 \text{ g/cm}^3 \)
Figure 5.3: (a) A two layer model. (b) A single shot gather with the shot at a depth of 75 meters. (c) Migrated section of pre-critical reflections. (d) Migrated sections of post-critical reflection.
Figure 5.4: Presumed reflector dip angle and incidence angle can be found after both the rays that come to and leave from the assumed reflector are calculated.

5.4 FRIENDSWOOD CROSSWELL DATA MIGRATION

In this section, the crosswell seismic data from an Exxon test site near Friendswood, Texas (Chen et al., 1990) is migrated by the constrained Kirchhoff integral method described above. The effects of different constraints on the Kirchhoff migration are evaluated and the crosswell migration results are compared to a surface seismic section to show the differences in resolution.

5.4.1 Friendswood crosswell data

The Friendswood crosswell data were collected from two wells that are separated by 600ft. The sources are small explosives located at a depth range from 30 ft to 1000 ft with an interval of 10 ft. The receivers are hydrophones within a depth range between 10 ft and 960 ft and the receiver interval is also 10 ft.

Figure 5.5 shows a common shot gather (CSG) of the recorded data where the source is located at a depth of 550 ft. The data are displayed after normalizing the traces. There is no significant shear wave energy in the recorded data set. However, the P-wave primary reflections are concealed by the other wave modes, such as tube waves and free surface reflections. Therefore, preprocessing is needed to enhance the reflections and to separate the up- and down-going waves.

Figure 5.6 is the same common shot gather as Figure 5.5 after processing and Figure 5.7 shows the separated up- and down-going waves. Please refer to Cai and Schuster (1993) for the detailed data processing.

First arrival traveltine tomography was carried out by Nemeth, Normark and Qin (1993). Figure 5.8 shows the velocity tomogram. The resolution of the tomogram is
Figure 5.5: A common shot gather (CSG) of the recorded data set after normalizing the traces. The shot depth is 550 ft.
Figure 5.6: Same CSG as Figure 5.5 after processing.
Figure 5.7: (a) Up-going wave seismogram extracted from Figure 5.6. (b) Down-going wave seismogram extracted from Figure 5.6.
relatively low even though the tomography result gives very small traveltime residuals. There are no fine structures in the tomogram. However, this tomogram provides a good velocity distribution for migration.

5.4.2 Results of constrained Kirchhoff integral migration

The constrained Kirchhoff migration is applied to the crosswell reflection data first. The reflector dip angle is constrained to be less than 15° and the incidence angle limit is set at 75°. Figure 5.9 shows the result. It can be seen that the migration section has a much higher interface resolution than that of the traveltime tomogram. We even get a clear image just below the well. Subwell images can be very important to the interpreter.
Figure 5.9: Migrated section of the constrained Kirchhoff integral migration.
5.4. FRIENDSWOOD CROSSWELL DATA MIGRATION

To show how constraints affect the image, Figure 5.10 shows the migrated section without the incidence angle constraints. It is clear that it has a lower resolution than the image in Figure 5.9. The wavelet stretch associated with large incidence angle rays widens the events and causes the resolution loss while the incidence angle constraint in Figure 5.9 avoids this problem.

Figure 5.11 shows the migrated section without the reflector dip angle constraint. This result is much noisier due to the migration artifacts especially at the near well regions. Although the reflector dip constraint cannot enhance the reflection energy, it largely decreases the migration artifacts.

5.4.3 Up- and down-going waves

For the crosswell seismograms, the up-going and down-going waves can be dealt with separately. The separation of these two wave types can be done either by f-k filtering or by median filtering (Cai and Schuster, 1993).

Figure 5.12 and 5.13 show the migrated sections by the constrained Kirchhoff migration method for the up-going wave and down going wave, respectively. The image at the bottom of the up-going section (Figure 5.12) is better while the image at the top is better for the down-going section (Figure 5.13).

If the up-going and down-going migrated sections are to be summed together, attention should be paid to the migration formulae for up- and down-going waves. This is because the reflection coefficient for down-going waves has a sign opposite to that for the up-going waves. Hence, there is half a cycle phase change between the up- and down-going waves (Figure 5.14). Figure 5.14a shows a CSG gather for a shot at depth 35 meters in the model shown in Figure 5.3a. The reflections are up-going waves. Figure 5.14b shows a CSG for the same model with the shot depth of 215 meters. The reflections are all down-going waves and they obviously have an opposite sign relative to the up-going waves. The Kirchhoff integral formula does not automatically take this into account. Figure 5.15a shows part of the migrated section for the up-going reflections (Figure 5.14a) and Figure 5.15b shows part of the migrated section for the down-going reflections (Figure 5.14b). The phases of the two sections are obviously different. Therefore, a negative sign is usually given to the down-going wave migration formula. However, this may not always be the right thing to do. When the wavelet in the seismograms is similar to a sinusoidal shape, the negative sign is automatically incorporated since the migration process will turn the wavelet upside down for down-going waves (Figure 5.16).

5.4.4 Comparison to surface seismic

Figure 5.17 shows the summation of the up-going and down-going sections compared with the sonic log and the synthetic seismogram for the source well. The correlation between the synthetic and migrated sections is obvious. Most interfaces in the top
Figure 5.10: Migrated section without the incidence angle constraint.
Figure 5.11: Migrated section without the dip angle constraint.
Figure 5.12: Migrated section for up-going waves.
Figure 5.13: Migrated section for down-going waves.
Figure 5.14: (a) CSG for the model shown in Figure 5.3a with a shot depth of 35 m; (b) CSG for the model shown in Figure 5.3a with a shot depth of 215 m.
Figure 5.15: (a) Migrated up-going wave section for the model shown in Figure 5.3a; (b) Migrated down-going wave section for the model shown in Figure 5.3a.
Figure 5.16: (a) Up-going and down-going reflections; (b) Migrated trace for up-going and down-going reflections.
and bottom of the model are well imaged in the migrated section. However, some structures in the middle of the model are not very clear or have a shifted depth compared with the synthetics. One reason is that there are no strong reflection interfaces in the middle part of the model (This can be seen from the synthetic section). Another reason is that the up-going and down-going sections are not perfectly stacked due to the inaccuracy of the migration velocity and/or the wavelet variation. As a comparison, Figure 5.18 show the surface seismic section for the same region as that in Figure 5.17 (courtesy of Chen et al., 1990). This suggests that crosswell data migration can provide a much higher resolution image of the earth structure than surface seismic imaging.

5.5 DISCUSSION AND CONCLUSION

A good migration image depends on a good migration velocity. The crosswell migration velocity can always be found by tomographic methods. Sonic log information can be used as a constraint or as an initial model in the tomography process. If sonic log data are directly used as a migration velocity, the sharp contrasts in the sonic log should be smoothed to decrease traveltimes in ray tracing or head wave traveltimes by an eikonal travelt ime calculation.

It is very difficult to coherently sum the migrated up-going waves with the down-going waves. For this reason, it may be desirable to shape the wavelet in the seismogram into a simpler minimum phase signal (Findlay and Goulty, 1991). However, for many complicated real data sets, it is not easy to find a good shaping filter. On the other hand, the difference between the up-going and down-going migrated sections provides a very good means to further analysis the accuracy of the migration velocity.

Dip angle constraints are very helpful due to the source and receiver distribution geometry in a crosswell geometry. Incidence angle constraints are desired mainly due to the wavelet stretching. The postcritical reflection phase change may not be a big problem in some cases, since the wavelet change is not abrupt. Other constraints may be designed for some particular geological and geometrical situations.

Finally, crosswell migration has a much higher resolution than that from the surface seismic data. However, the data processing before migration is critical to the crosswell migration. The processing includes elimination of tube wave and direct wave, separation of P- and S-waves and up- and down-going waves, etc.

5.6 ACKNOWLEDGEMENT

We thank Dr. Sen Chen, Dr. Linda Zimmerman, Dr. Sam Blakeslee and Exxon Production Research for providing the Friendswood data and Wenyong Cai and Gerard Schuster for providing the processed the data set. We also thank the sponsors of 1992
Figure 5.17: Summation of up-going wave migration section and down-going wave section, compared with sonic log data and synthetic seismograms.
Figure 5.18: The surface seismic section of the same region as that of the crosswell section (after Chen et al.).
University of Utah Modeling and Tomography Development Project for their financial support.

5.7 REFERENCES


Report 6

Conjugate Gradient Migration of Crosswell Seismic Data. Some Preliminary Results

Tamas Nemeth

6.1 ABSTRACT

A conjugate gradient migration method is applied to crosswell seismic data in order to decrease the migration artifacts and to enhance the detectability of reflections. The migrated sections for both synthetic and real data show that the conjugate gradient migration method provides a better estimate of the earth structure at the expense of more CPU time.

6.2 THEORY

The theory of conjugate gradient migration is shown in Schuster (1993). The conjugate gradient scheme (p. 73 in Nolet, 1987) applied to the equations $L_0 \Delta s = p_{ref}$ is given by
\[ \Delta s^{(0)} = 0; \; d^{(0)} = p^{ref}; \; r^{(0)} = L_0^T p^{ref}; \; m^{(0)} = r^{(0)} \]
and for \( i = 0, 1, 2, \ldots \)
\[ d^{(i)} = L_0 \; m^{(i)} \]
\[ \alpha_i = (r^{(i)}, r^{(i)})/(d^{(i)} , d^{(i)}) \]
\[ \Delta s^{(i+1)} = \Delta s^{(i)} + \alpha_i \; m^{(i)} ; \; d^{(i+1)} = d^{(i)} - \alpha_i \; d^{(i)} \]
\[ r^{(i+1)} = L_0^T d^{(i+1)} \]
if \( r^{(i+1)} = 0 \) then quit
\[ \beta_i = (r^{(i+1)}, r^{(i+1)})/(r^{(i)}, r^{(i)}) \]
\[ m^{(i+1)} = r^{(i+1)} + \beta_i \; m^{(i)} \]

where \( d^{(i)} \) is interpreted as the remodeled data, \( r^{(i)} \) is interpreted as the migrated data residual and \( m^{(i)} \) is interpreted as a weighted sum of conjugate gradient migrated data and data residuals.

### 6.3 METHODOLOGY

We avoid constructing the matrix \( L \) or its transpose by recognizing that \( L_0^T d \) means migration of the data and \( L_0 m \) means forward modeling of the slowness perturbation \( m \). To execute both the forward modeling and migration algorithms efficiently, we use asymptotic Green's functions (Schuster, 1993). The asymptotic Green's functions are calculated by using a finite-difference solution to the eikonal equation (Qin et al., 1992) and using geometrical spreading estimates for the transport terms.

The asymptotic solution for both forward modeling and migration is very efficient computationally, since we compute the traveltime field for a particular source or receiver just once. Reuse of the traveltime field for that source or receiver requires a read from hard disk.

In spite of its simplicity, the asymptotic forward modeling appears to adequately simulate the real data (compare the measured seismograms in Figure 6.1 and Figure 6.3 to the calculated seismograms on Figure 6.2 and Figure 6.4, respectively. The measured seismograms are part of the Friendswood data.). Apparently, given the correct velocity model, the eikonal equation solver computes the phase information efficiently. Using the geometrical spreading estimates to approximate the amplitude information (transport equation) also seems to be sufficient at this stage. The bandlimited frequency spectrum of the measured seismograms is also preserved in the calculated seismograms, since both the asymptotic forward and inverse modeling algorithms have flat bandpass characteristics, so subsequent convolutions with these operators do not change the frequency spectrum.
There are several problems to be overcome in migration, such as the limited source-receiver aperture, uneven ray coverage and source wavelet stretching for large incidence angles. Therefore constraints are applied to the asymptotic inverse modeling (migration) in order to suppress some regular artifacts. This time I used the following constraints:

- Incidence angle constraint. Rays with an incidence angle larger than a predetermined value are severely suppressed or simply eliminated.
- Dip angle constraint. Image points, through which the reflector has a dip angle larger than a predetermined value, are eliminated.

Although some of the above mentioned problems in migration can be overcome by applying constraints to the migration procedure, other problems can be overcome by applying an iterative scheme, which is here the conjugate gradient scheme. The conjugate gradient scheme is computationally efficient, since it only involves asymptotic forward and inverse modeling, and computation of the $l_2$ norm of vectors.

However, in practical applications we face several problems. When we iterate several times, the average amplitude level of subsequent seismograms and migrated sections usually increases. Theoretically, in the conjugate gradient scheme scalars $\alpha_i$ and $\beta_i$ take care of the proper magnitude level of subsequent seismograms and migrated sections. Since we do not apply $L$ or $L^T$ matrices directly, but instead we do forward and inverse modeling with constraints, in some cases the calculated $\alpha_i$ and $\beta_i$ values are not optimal. For example, if $\alpha_i$ is such that $\alpha_i d^{(i)} > d^{(i)}$, then in the new iteration $d^{(i)}$ will be dominated by the $\alpha_i d^{(i)}$ term with a minus sign. And the sign change represents a polarity change of the seismograms, so we weaken the solution vector by adding the new migrated section to it, instead of improving it. The proper calculation of optimal $\alpha_i$ and $\beta_i$ values for practical applications is a problem to be solved.

### 6.4 Numerical Results

In this section a synthetic bandlimited data set and the Friendswood data set are used to test the conjugate gradient migration.

**Synthetic data test.** This synthetic data set previously was used by Qin (1993). The model size is 120 m in width and 300 m in depth. There are 51 sources evenly distributed along the left well and 101 receivers on the right edge.

Figure 6.5 shows a prestack migrated section (shot depth is 10 m) after the first iteration. There are very strong migration artifacts in the upper half of the model. Figure 6.6 depicts the prestack migrated section of the same shot depth after the fifth iteration. The migration artifacts are significantly suppressed after the fifth iteration. Figure 6.7 shows the migrated section after the first iteration for the combined prestack migrated sections. The strong migration artifacts of prestack migrated sections are mostly filtered out. Figure 6.8 depicts the migrated section after the fifth iteration. The artifacts remaining after the first iteration mostly disappear.
Friendswood data. The real data test was carried out for a data set collected near Friendswood, TX by Exxon. The description of the experiment is given in Chen et al. (1990).

Figure 6.9 shows the migrated section after the first iteration. Figure 6.10 depicts the migrated section after the third iteration. Notice that after the third iteration the horizontal reflectors became much pronounced and also the elliptic migration artifacts are weaker.

6.5 CONCLUSION AND DISCUSSION

As the numerical results indicate, conjugate gradient migration can help to enhance the reflectors and to suppress the migration artifacts. A surprising result is that the asymptotic modeling provides a very acceptable means for modeling the data. However, much research is necessary to test the conjugate gradient scheme in numerical implementations. Future research will also explore asymptotic waveform inversion.

6.6 REFERENCES


Figure 6.1: A common source gather of the Friendswood data. Source depth is 720 ft.
Figure 6.2: A calculated common source gather to the Friendswood data. Source depth is 720 ft.
Figure 6.3: A common source gather of the Friendswood data. Source depth is 310 ft.
Figure 6.4: A calculated common source gather to the Friendswood data. Source depth is 310 ft.
Figure 6.5: A prestack migrated section of the synthetic data after the first iteration. The source depth is 10 m.
Figure 6.6: A prestack migrated section of the synthetic data after the fifth iteration. The source depth is 10 m.
Figure 6.7: The migrated section of the synthetic data after the first iteration.
Figure 6.8: The migrated section of the synthetic data after the fifth iteration.
Figure 6.9: The migrated section of the Friendswood data after the first iteration.
Figure 6.10: The migrated section of the Friendswood data after the third iteration.
Figure 6.11: The undamped migrated section of a single shot gather (source depth is 720 ft) of the Friendswood data after the first iteration.
Figure 6.12: The undamped migrated section of a single shot gather (source depth is 720 ft) of the Friendswood data after the fifth iteration. Note that the fifth iterate contains less migration artifacts than iterate one in the previous page.
Figure 6.13: The damped migrated section of a single shot gather (source depth = 720 ft) of the Friendswood data after the first iteration. The damping factor was implemented by incorporating the constraints $\|u^e\|$ into the cost function.
Figure 6.14: A calculated common source gather (source depth is 720 ft) to the Friendswood data after the first undamped migration iteration.
Figure 6.15: A calculated common source gather (source depth is 720 ft) to the Friendswood data after the fifth undamped migration iteration.
Figure 6.16: A calculated common source gather (source depth is 720 ft) to the Friendswood data after the first damped migration iteration.
Report 7

Generalized Migration Velocity Analysis: Optimization in the Model Domain

Tamas Nemeth

7.1 ABSTRACT

In this paper a general theory of migration velocity analysis is presented. Also some migration velocity analysis formulas are shown to be special cases of this general expression. Expressing the migration velocity analysis in general terms gives the opportunity to extend the existing techniques to more complex situations. Finally, in the Appendix the migration velocity analysis technique is linked to the optimization theory.

7.2 THEORY

Migration velocity analysis techniques have recently become common in seismic data processing. Many algorithms were derived for simplified reflector geometries. In this chapter I derive an equation, named the general migration velocity analysis equation, which is appropriate for any kind of reflector geometry and velocity field. I also discuss the application of the obtained equation.

Assume that the traveltimes associated with the acoustic reflections can be described by

\[ t = L s, \]  

(7.1)

where \( t \) is the travelt ime vector, \( L \) is the wavepath matrix and \( s \) is the slowness vector. A small perturbation with respect to the slowness parameter to this equation yields:

\[ \delta t = L \delta s + \delta L s. \]  

(7.2)
Equation 7.2 shows that two perturbations, the slowness perturbation and the wavepath perturbation, contribute to the traveltime perturbation. Rigorously, the wavepath perturbation is a function of the slowness perturbation, but if we do not know the exact relation between these two perturbations we can treat them independently. One of the results of the independent treatments is that physically non-feasible models can be constructed. For example, we can perturb the wavepaths in such a way that they do not follow Snell's law determined by the actual velocity field. However, this price is bearable, since in seismic data processing we go beyond the physically possible processes in many cases.

Since on the right hand side of equation 7.2 there are two unknowns, and on the left hand side is just one unknown, we can choose $\delta t$ to be any value. Choose $\delta t = 0$, then equation 7.2 can be rewritten as:

$$0 = L \delta s + \delta L s,$$

which states that, if we perturb a slowness vector, we can always perturb the wavepath matrix in such a way that the gross traveltime change will be zero. The wavepath perturbation, which occurs in equation 7.3, can be in general a physically non-feasible process since it does not necessarily obey Snell's law.

From equation 7.2 it is straightforward to see that

$$\delta s = -L^{-1} \delta L s.$$  \hspace{1cm} (7.4)

This type of equation is the general equation for the migration velocity analysis, as we will see later in the next chapter where special cases of the generalized migration velocity analysis equation will be given.

For different shot gathers the slowness perturbation is:

$$\delta s = -(L^i)^{-1} \delta L^i s,$$  \hspace{1cm} (7.5)

where the $i$-th index stands for the $i$-th shot gather. Since different shots illuminate different parts of the earth, in general $L^i \neq L^j$ and $\delta L^i \neq \delta L^j$, where $j$ is an index other than that of the $i$-th shot. If $\delta L^i$ can be measured, as it is a case in seismics, the slowness perturbation $\delta s$ can be computed. For detailed discussion see the Appendix.

Now we apply the above algorithm to seismic migration. In prestack migration, if the correct velocity model was applied all the individual migrated sections will be identical. If the migration velocity differs from the correct one, the individual migrated sections also will be different (if nonuniqueness is not present). Rearranging the migrated sections into common-image gathers, the image points of each individual event can be traced. We can measure for each trace the vertical component $\delta z$ of the wavepath perturbation $\delta L$ (Figure 7.1), which can be recast into $\delta L$ as:

$$\delta L = 2 \delta z \cos(\phi) \cos(\gamma),$$  \hspace{1cm} (7.6)

where $\gamma$ is the angle of incidence and $\phi$ is the angle of dip (Stork, 1992).
7.3 \textit{SOME SPECIAL CASES OF MIGRATION VELOCITY ANALYSIS}

Figure 7.1: a) Common-image gather after a prestack depth migration, where \( z_0 \) is the correct depth, \( z_m \) is the depth of the image points, \( \delta z \) is the vertical wavepath perturbation. b) Common-image gather after a prestack time migration, where \( t_0 \) is the zero-offset time, \( t_m \) is the zero-offset time of the image points, \( \delta \tau \) is the time perturbation.

There is an alternative to this procedure. We apply prestack time migration instead of prestack depth migration. In this case we measure directly \( \delta \tau = \delta L s \) on the common image gather, without any approximation. According to this statement, we have a simple procedure for migration velocity analysis:

a) Measure the time residuals \( \delta \tau \) on the prestack time migrated common-image gathers.

b) Backpropagate these residuals by the computed old raypaths.

c) Update the velocity model.

In practical applications we might solve the normal equation to equation 7.4, that is

\[
\delta s = - \left( L^T L + \epsilon I \right)^{-1} L^T \delta \tau. \tag{7.7}
\]

\[\text{7.3 \ SOME SPECIAL CASES OF MIGRATION VELOCITY ANALYSIS}\]

There were several migration velocity analysis formulas published recently. In this section I show that these formulas are special cases for the general formula, described in equation 7.4.

First investigate the classical migration velocity analysis formula, given in Al-Yahya (1989):

\[
z_m^2 = \gamma^2 z^2 + \left( \gamma^2 - 1 \right) h^2 \tag{7.8}
\]
where \( z_m \) is the migrated depth, \( \gamma = \frac{V_p}{V} \) is the ratio between the migration and the correct velocities, \( z \) is the correct depth and \( h \) is the half offset. This equation can be rewritten as

\[
(z + dz)^2 = \frac{s^2}{(s + ds)^2} z^2 + \left( \frac{s^2}{(s + ds)^2} - 1 \right) h^2.
\]  

(7.9)

using expressions \( s = \frac{1}{V} \) and \( s + ds \approx \frac{1}{V + dV} \), as \( dV \to 0 \). Neglecting the second order terms, equation 7.9 yields:

\[
ds = -\frac{z}{z^2 + h^2} dz s.
\]  

(7.10)

It is straightforward to see that this expression is a special case of equation 7.4 for the particular choice of \( L = 2\sqrt{z^2 + h^2} \).

Similarly, the formula for a dipping layer, given in Nemeth (1992)

\[
z_m^2 = \gamma^2 z^2 + \left( \gamma^2 - 1 \right) h^2 \left( 1 - \sin^2(\varphi) \right)
\]  

(7.11)

can be derived from the particular choice of \( L = 2\sqrt{\frac{z^2}{\cos^2(\varphi)}} + h^2 \left( 1 + \sin^2(\varphi) \right) \), which is, in fact, a hyperbolic approximation of the NMO-equation of the common-depth point gathers (Nemeth, 1993).

A different algorithm is published in Lafond and Levander (1993). Their equation 4 can be represented in the matrix form as

\[
dV = t^{-1} dL
\]  

(7.12)

where \( V \) is the velocity vector. Replacing \( V \) with \( s^{-1} \) and after some algebraic manipulations we get equation 7.4. They used an approximation in the estimation of \( dL \) in equation 7.4, namely

\[
dL = (\cos(\alpha^s) + \cos(\alpha^r)) \, dz,
\]  

(7.13)

where \( \alpha \) is the angle of the ray at the imaging point. (Here \( s \) stands for the source and \( r \) for the receiver.)

Finally, it should be mentioned that Stork (1992) used an approximation to equation 7.4 by choosing

\[
dL = 2 \, dz \, \cos(\alpha) \, \cos(\varphi),
\]  

(7.14)

where \( \alpha \) is the angle of incidence and \( \varphi \) is the angle of dip. His formula differs from equation 7.4 in the sign.

### 7.4 FUTURE WORK

Future work in this area will include the coding up of this technique and testing it with traveltime and waveform type optimization methods.
7.5. REFERENCES


7.6 APPENDIX

In this appendix a more general description of migration velocity analysis is given. It is shown that mathematically the migration velocity analysis corresponds to an optimization in the model domain, contrary to the conventional tomography which is an optimization in the data domain. In data domain optimization we try to match the calculated data (i.e. seismograms in seismics) with the measured data, while in the model domain optimization we match the independent calculated models (i.e. prestack migrated sections in seismics) with each other.

For a comparison, both the data and model domain optimizations are briefly described in this chapter. It is also shown that, while data domain optimization (conventional tomography) usually exploits a linearized version of the nonlinear system \( Ls = t \), the model domain optimization (migration velocity analysis) is made possible by the nonlinearity of the system and the independence of the individual experiments (for example, independence of each shot gather from each other in seismics in the sense, that they illuminate different parts of the earth).

Assume a nonlinear relation between the model and the data:

\[
t^i = L(s_o) s_o,
\]

where \( s_o \) is the true model, \( t \) is the data and \( L(s_o) \) is an operator, connecting the data with the model. \( L(s_o) \) is a nonlinear operator, since it is a function of \( s_o \). The \( i \) superscript stands for the \( i \)-th independent individual experiment. Solving \( s_o \) from \( t^i \) is a nonlinear problem, since we have to invert the model-dependent operator \( L(s_o) \).
The strategy to solve equation 7.15 is to set a trial model \( s_t \), form operators \( L(s_t) \) and adjust the model with respect of these trial operators. There are two ways to carry out this procedure, matching the calculated data with the measured data and matching the calculated models with each other.

**Optimization in the data domain.** In this case we have the original data vector and a calculated data vector:

\[
\begin{align*}
t^i &= L^i(s_o) s_o, \\
t'^i &= L^i(s_t) s_t,
\end{align*}
\]  

(7.16)  

(7.17)

where \( t^i \) is the calculated data vector for the \( i \)-th experiment. We want to find the data residual \( \Delta t^i \):

\[
\Delta t^i = t^i - t'^i = L^i(s_o) s_o - L^i(s_t) s_t.
\]  

(7.18)

This is a rather general expression. We can simplify it, specifying the type of nonlinearity of operator \( L^i(s) \). Assuming a linear operator \( L^i = L^i(s_o) = L^i(s_t) \), equation 7.18 simplifies to

\[
\Delta t^i = L^i(s_o - s_t) = L^i \Delta s.
\]  

(7.19)

On the other hand, assuming a weak nonlinearity \( L^i(s_o) = L^i(s_t) + dL^i(s_t) \), equation 7.18 results in

\[
\Delta t^i = L^i(s_t) \Delta s + dL^i(s_t) s_o.
\]  

(7.20)

Equations 7.18, 7.19, 7.20 are the basic equations for the tomography type problems. In seismic tomography (both in the traveltime and in the waveform) we use equation 7.19 (Tarantola, 1987), attacking the nonlinearity by resetting the operators \( L^i(s_t) \) after each iteration.

**Optimization in the model domain.** In this case we use the inverse operators \( L^i(s_t)^{-1} \), rather than \( L^i(s_t) \). Applying these operators results in

\[
\begin{align*}
L^i(s_o)^{-1} t^i &= s_o, \\
L^i(s_t)^{-1} t^i &= s^i,
\end{align*}
\]  

(7.21)  

(7.22)

where \( s^i \) are the solution vectors for the \( i \)-th independent experiment. We mention that if \( s_t \neq s_o \), then in general \( s^j \neq s^i \neq s_o \), where \( j \) is an index other than that of the \( i \)-th experiment. This inequality is the consequence of the nonlinearity of the system. In some special cases it might happen that \( s^j = s^i \) even in the case of \( s_t \neq s_o \). This situation is the manifest of the nonuniqueness of the problem. (In seismic data processing nonuniqueness can be present in the case when several slowness models can give the same migrated section, i.e. a unique correspondence between the long and the short wavelength terms of the slowness field cannot be established.)

Similar to the data residual, here we want to find the model residual \( \Delta s^i \):

\[
\Delta s^i = s_o - s^i = L^i(s_o)^{-1} t^i - s^i = \left[ L^i(s_o)^{-1} L^i(s_t) - I \right] s^i.
\]  

(7.23)
Assuming a linear operator $L^i = L^i(s_0) = L_i(s_t)$, equation 7.23 simplifies to

$$\Delta s^i = \left[ L^i(s_t)^{-1} L_i(s_t) - I \right] s_t = 0. \tag{7.24}$$

Equation 7.24 expresses, that in case of linear $L^i$, model domain optimization is impossible since all the calculated models should agree with the original one.

Assuming the weak nonlinearity $L^i(s_0) = L_i(s_t) + dL^i(s_t)$ and put it back into equation 7.23:

$$\Delta s^i = \left[ \left( L^i(s_t) + dL^i(s_t) \right)^{-1} L_i(s_t) - I \right] s_t. \tag{7.25}$$

Equation 7.25 can be simplified as

$$\Delta s^i = \left[ \left( I + L^i(s_t)^{-1} dL^i(s_t) \right)^{-1} - I \right] s_t 
\approx \left( I - L^i(s_t)^{-1} dL^i(s_t) - I \right) s_t = -L^i(s_t)^{-1} dL^i(s_t) s_t, \tag{7.26}$$

assuming that $L^i(s_t) \gg dL^i(s_t)$.

If we set up a trial model $s_t$, than $s^i = s^i = s_t$ and equation 7.23 can be rewritten as:

$$s_0 = s_t + \Delta s^i = s_t + \Delta s^i \quad \text{or} \quad \Delta s^i = \Delta s^i. \tag{7.27}$$

Since each individual experiment involves different parts of the model domain (for example, each shot illuminates different partsof the earth in seismics), in general $L^i(s_t) \neq L_i(s_t)$ (independence condition). In this case $dL^i(s_t)$ should not be equal to $dL^i(s_t)$ to make equation 7.27 true. If $dL^i(s_t)$ or $\Delta dL(s_t) = dL^i(s_t) - dL^i(s_t)$ are measurable, than $\Delta s$ can be computed. If we have a unique solution to equation 7.15, than $dL^i(s_t) = dL^i(s_t)$ is true just in case of $dL^i(s_t) = dL^i(s_t) = 0$, or - which is the same - when $s_t = s_0$. In the opposite case, when $dL^i(s_t) = dL^i(s_t)$ even in case of $dL^i(s_t) \neq dL^i(s_t) \neq 0$ or $s_t \neq s_0$, a nonuniqueness is present. In this case the optimization might get stuck in local minima.

In summary, model domain optimization is possible in the case, when the system is nonlinear and the individual experiments are independent from each other in the sense, that they illuminate different parts of the model domain. In seismics the individual shots are independent in this sense and migration velocity analysis is a common working procedure.
Part III

Data Processing
Report 8

Numerical Verification of Converting 3-D Data to 2-D Data

Jinlong Xu

8.1 ABSTRACT

A filtering operation is applied to 3-D data to convert them to 2-D data. The 3-D data are calculated from the Exxon Friendswood model and compared to synthetic 2-D data. The results show that the conversion is almost error free.

8.2 INTRODUCTION

2-D waveform tomography methods require that the 3-D data be converted to 2-D data. Therefore, the 3-D to 2-D filtering operation is needed. How accurate is this filtering operation? This paper demonstrate that it is very accurate for Exxon Friendswood model.

8.3 THEORY

Assume that seismic waves honor the acoustic wave equation.

\[ (\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2})p = -f, \]  \hspace{1cm} (8.1)

where the \( p \) is pressure and the \( f \) is a source function.

Taken the Fourier transform of both sides, equation (8.1) becomes the Helmholtz equation below

\[ (\nabla^2 + k^2)P = -F, \]  \hspace{1cm} (8.2)
\[ P = \int_{-\infty}^{+\infty} pe^{-i\omega t} dt, \quad (8.3) \]
\[ F = \int_{-\infty}^{+\infty} fe^{-i\omega t} dt, \quad (8.4) \]

where \( k^2 = \omega^2/c^2 \).

The Green's function in the frequency domain is the solution of the following equation

\[ (\nabla^2 + k^2)G = -\delta(x - x_0), \quad (8.5) \]

where the solution of equation (8.2) can be expressed as

\[ P = GF \quad (8.6) \]

The \( P^{(2)} \) and \( G^{(2)} \) respectively denote solutions of equations (8.2) and (8.5) in two dimensions; and \( P^{(3)} \) and \( G^{(3)} \) respectively denote solutions of equations (8.2) and (8.5) in three dimensions. Thus we have

\[ P^{(2)} = G^{(2)} F, \quad (8.7) \]
\[ P^{(3)} = G^{(3)} F. \quad (8.8) \]

In a homogeneous medium, we know that

\[ G^{(2)} = \frac{i}{4} H_0^{(1)}(kr), \quad (8.9) \]
\[ G^{(3)} = \frac{1}{4\pi r} e^{i kr}, \quad (8.10) \]

where \( k = \omega/c, \ r = \sqrt{||x||_2} \), and \( H_0^{(1)} \) the Hankel function of the first-kind and zero-order.

When \( kr \gg 1 \), we have

\[ H_0^{(1)} \approx \left( \frac{2}{\pi kr} \right)^{1/2} e^{i (kr - \pi/4)}. \quad (8.11) \]

Plugging (8.11) into (8.9), we have

\[ G^{(2)} = \left( \frac{2\pi rc}{\omega} \right)^{1/2} e^{i \frac{kr}{4\pi r}} e^{i kr} = \left( \frac{2\pi rc}{\omega} \right)^{1/2} e^{i \frac{k}{4} G^{(3)}. \quad (8.12) \]

From equations (8.7) and (8.8), we have

\[ P^{(2)} = G^{(2)} F = \left( \frac{2\pi rc}{\omega} \right)^{1/2} e^{i \frac{k}{4} G^{(3)} F = \left( \frac{2\pi rc}{\omega} \right)^{1/2} e^{i \frac{k}{4} P^{(3)}}. \quad (8.13) \]

The solution of equation (8.1) is

\[ p^{(2)} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left( \frac{2\pi rc}{\omega} \right)^{1/2} e^{i \frac{k}{4} P^{(3)}} d\omega. \quad (8.14) \]

This is the formula which we use to convert 3-D data to 2-D data.
8.4 NUMERICAL RESULTS

We use the 2.5-D finite-difference code (Xu, 1993) to generate 3-D data. The 2-D data are generated by a 2-D finite-difference code (Schuster and Xu, 1993).

First we have to verify our algorithm for a homogeneous model. Figure 8.1 shows a homogeneous crosswell model. Table 8.1 lists input parameters for the 2.5-D and 2-D finite-difference codes. Figure 8.2 shows that the 3-D and 2-D seismic traces are different from one another. After conversion of 3-D data to 2-D data using equation 8.14, the differences between them are very small (Figure 8.4). This result is expected from the theory in the above section. But the question is how well does it work for in heterogeneous media?

To answer this question and to verify the results of real data inversion by the WTW (Zhou, 1993), we choose the Exxon Friendswood velocity model (Figure 8.4). For the 52nd shot and the 40th receiver, the 3-D data and 2-D data are different as expected (Figure 8.5). After conversion of the 3-D data to 2-D data, most of the wave trains are matched. Figures 8.7, 8.8, 8.9, 8.10, and 8.11 show the seismograms associated with the 52nd shot. It is hard to see the difference between the 3-D and 2-D seismograms (Figures 8.7 and 8.8), but the differences between them are obvious with the subtraction of the 2-D seismogram from the 3-D seismogram (Figure 8.9). Figure 8.10 shows the seismogram which is the 3-D to 2-D conversion of the 3-D seismogram, and the differences between them are small (Figure 8.11). In this case, we chose r and c in equation 8.14 to be equal to the shot-receiver distance and 6500 ft/s respectively.

8.5 CONCLUSION

The 3-D to 2-D filtering operation is shown to be valid for the Exxon crosswell model. Therefore, 2-D waveform inversion can be applied to 3-D data sets after application of the 2-D conversion method; this assumes an approximate 2-D earth model.
8.6 REFERENCES


Figure 8.1: 2-D crosswell model. The 3-D crosswell model is the rotation of the 2-D model about the vertical source axis well. The source in the source well is located at the depth of 520 ft. There are 98 receivers evenly spaced in the receiver well with 10 ft intervals.

Figure 8.2: Comparison between 3-D and 2-D data for a homogeneous model
Figure 8.3: Comparison between 2-D and converted 3-D to 2-D data for a homogeneous model.

Figure 8.4: The Exxon model. The distribution of the source and the receivers are the same as in Figure 8.1.
8.6. REFERENCES

Figure 8.5: Comparison between 3-D data and 2-D data for the Exxon Friendswood model.

Figure 8.6: Comparison between 2-D and converted 2-D data. The source is at the depth of 520 ft. The receiver is at the depth of 400 ft.
Figure 8.7: 3-D synthetic seismogram with the shot at the depth of 520 ft.

Figure 8.8: 2-D synthetic seismogram with the shot at the depth of 520 ft.
Figure 8.9: 3-D synthetic seismogram data minus 2-D synthetic seismogram with the shots at the depth of 520 ft.

Figure 8.10: Converted 2-D seismogram from the 3-D synthetic seismogram.
Figure 8.11: 2-D synthetic seismogram minus converted 2-D seismogram with the shot at the depth of 520 ft.
Part IV

FORTRAN Codes
Report 9

Filter Coherent Events With A Wavelet Transform

Yonghe Sun

FORTRAN CODE: WVLTF.F, WVFILT.F, SEGY2FUF.F, and FUF2SEGY.F

PURPOSE: To remove or extract coherent events in a CSP gather or other types of gathers. The coherent events can be tube waves, ground roll, first arrivals, or other coherent type of noise or signals.

ALGORITHM DESCRIPTION:
If the coherent arrivals at different seismic traces are time-shifted so that they are all aligned at a constant time, the event will vary slowly from trace to trace. The event can therefore be eliminated by performing a wavelet transform (or Fourier transforms) for constant time slices and discarding the long wavelength (or low frequency) components of the transforms. The event can be windowed so that only a small band of the data around the event will be affected. The wavelet filter has been found more effective than the median filter (Schuster and Sun, 1992) for removing coherent events.

The wavelet transform is a procedure for a multiresolution decomposition of the data. For \( k \leq 0 \), let \( U_\phi(n, k) \) be a representation of a signal at the \( k \)-th level resolution. The signal at the highest level (\( k = 0 \)) of resolution is the input data itself. That is, \( U_\phi(n, 0) \) will be set to the \( n \)-th data sample. The signal at the lower (\( k < 0 \)) levels of resolution are computed by

\[
U_\phi(n, k) = \sum_{i=-\infty}^{\infty} \beta(i - 2n)U_\phi(i, k + 1) \tag{9.1}
\]

\[
U_\psi(n, k) = \sum_{i=-\infty}^{\infty} (-1)^i\beta(1 - i + 2n)U_\phi(i, k + 1) \tag{9.2}
\]
where the $\beta_i$'s are the coefficients (Lai, 1992) related to the wavelet used, $U_\phi(n,k)$ are the $k$-th level details of the signal. This decomposition process is performed for $k=-1,-2,\ldots$, and so on until the desired level (say with $k=\text{min}1v1$) of resolution is reached or until $k$ is somewhat smaller than $-\log_2(N)$ where $N$ is the number of points in the input data.

The signal can be reconstructed by starting from some level of resolution (say with $k=\text{min}1v1$) and adding the same and higher level details of the signal,

$$U_\phi(n,k) = \sum_{i=-\infty}^{+\infty} 2\beta(i - 2n)U_\phi(i, k-1) + \sum_{i=-\infty}^{+\infty} 2(-1)^n\beta(1 - n + 2i)U_\phi(i, k+1). \quad (9.3)$$

This reconstruction process is performed until $k = 0$ is reached and $U_\phi(n,0)$ contains the $n$-th data sample of the reconstructed signal. If for $k=\text{min}1v1$, the values of $U_\phi(n,k)$ are set to zero to start the reconstruction, then the reconstructed signal will lack the low resolution (or frequency) components that are present in the input signal. This is how the low frequency filter is built from using the wavelet transform. The wavelet decomposition uses bases that are of (effectively) local support which is desirable for reducing filter artifacts if the components to be filtered out can not be represented by basis functions of nonlocal support.

**Algorithm Implementation:** The algorithm is implemented in single precision Fortran as the subroutine “WVFILT” contained in the file “WVFILT.F”. The input to the subroutine includes the picked travel times of the coherent event, the window size around the event in the CSP gather, and the filter level parameter. The output data are the filtered CSP gather and an error flag.

The Fortran main program “WVLTF.F” is provided to demonstrate the usage of the subroutine “WVFILT.F”. Two other Fortran programs are also provided: “SEGY2FUF.F” for converting a SEGY formatted CSP gather into Fortran unformatted data required by the main program “WVLTF.F”, and “FUF2SEG.Y.F” for converting a Fortran unformatted gather into a SEGY formatted gather. The “README” file accompanying the programs contains more detailed information on how to use these codes and some sample inputs. The following outlines the procedure for using the main program “WVLTF.F”:

- Make sure the file “params.inc” exists. It contains the parameters for dimensioning arrays.
- Compile and link the programs to create the executable “WVLTF”:

  ```
  f77 wvltf.f wvfilt.f -o wvltf
  ```
- Prepare the namelist input file “WVFILTER.IN” containing the information about the size of the CSP gather, the filter parameters, and input and output file names (see below for format).
• Prepare the input file containing the picked traveltimes for the coherent event. The file name is specified by the variable “timefil” in the namelist input file “WVFILTER.IN”. The picked traveltimes are arranged as follows,

1,  0.1142
2,  0.1137
3,  0.1123
4,  0.1115
...

where the first entry in each record is the trace number, and the second entry in the record is the picked arrival time of the coherent event at the trace. If the arrival times for some traces are not picked, the arrival times for these traces will be interpolated linearly from those of the picked traces.

• Prepare the Fortran unformatted data file containing the CSP gather. The file name is specified by the variable “infil” in the namelist input file “WVFILTER.IN”. The file should be readable by the Fortran statements:

```fortran
open(unit=2,file=infil,form='unformatted')
do ntr = 1,ntrtot
   read(1)(data(i,ntr),i=1,KSAMP)
end do
close(unit=1)
```

If the CSP gather is in SEGY format, it can be converted into Fortran unformatted by running the program “SEGY2FUF.F”. The executable “SEGY2FUF” can be created by

```
f77 segy2fuf.f -o segy2fuf
```

• Execute the program “WVLTF”. The output file contains the filtered CSP gather. The output file name is specified by the variable “outfil” in the namelist input file “WVFILTER.IN”. The output filtered CSP data file can be read in the same way as the input CSP data file. The output file is Fortran unformatted, and can be converted back to SEGY formatted by running the program "FUF2SEGY.F". The executable “FUF2SEGY” can be created by

```
f77 fuf2seg.f -o fuf2seg
```
- The above steps can be repeated if more than one coherent event is to be filtered.

**EXAMPLES:** Figure 9.1 shows a synthetic crosswell CSP gather containing first arrivals, upgoing and downgoing tubes waves, and reflections. Figure 9.2, 9.3, and 9.4 shows respectively the same CSP gather with the first arrivals, upgoing tube waves, and downgoing tube waves filtered out.

**ALGORITHM CAVEATS:** The following are caveats about possible pitfalls in using "WVLTF.F" and "WVFILT".

- The algorithm will not be effective if the shape or amplitude of the wavelets associated with the coherent arrivals varies too rapidly from trace to trace. An extreme example of rapid wavelet amplitude variation is to have a dead trace in between every other trace.

- Removal of one event will affect other events with similar moveouts. If several events are to be removed, the more energetic events should be removed first. This may reduce the overall filter artifacts.

- The computation time increases linearly with the window size.

- To reduce the artifacts at the edges of the filter window, new data points are added to the beginning and end of the input data array for the wavelet transform. If the input data array is \( u(i) \), for \( i = 0, 1, 2, \ldots, N \), then the new data points for \( i < 0 \) is added by "reflection": \( u(i) = u(N - i) \). Similarly, for \( i > N \), \( u(i) = u(N - (i - N)) \).

- The wavelet transform used here does not require the length of the input data array to be an integer power of 2. For an introduction of other wavelet transforms, see Press et al., 1992.

**REFERENCE:**


**INPUT FORMATS**

"WVFILTER.IN" File:

```plaintext
$wvfilter
ntrot = 96, KNSAMP = 1500, t0 = 0.0, dt = 2.5e-4,
ntrmin = 1, ntrmax = 96, dtlow = 0.02, dtup = 0.02,
minlvl = -3, infil = 'unfiltuf.dat',
timefil = 'tpick.dat', outfil = 'filtuf.dat',
$end
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntrtot</td>
<td>The number of traces in the CSP gather.</td>
</tr>
<tr>
<td>KNSAMP</td>
<td>The number of samples per trace in the CSP gather.</td>
</tr>
<tr>
<td>t0</td>
<td>The time at the first data sample.</td>
</tr>
<tr>
<td>dt</td>
<td>The sample rate in the same time unit as &quot;t0&quot;.</td>
</tr>
<tr>
<td>ntrmin</td>
<td>The lowest trace number in the filter window.</td>
</tr>
<tr>
<td>ntrmax</td>
<td>The highest trace number in the filter window.</td>
</tr>
<tr>
<td>dtlow</td>
<td>The time width of the filter window to the smaller time side of the picked event.</td>
</tr>
<tr>
<td>dtup</td>
<td>The time width of the filter window to the larger time side of the picked event.</td>
</tr>
<tr>
<td>minlvl</td>
<td>The low cut filter level of resolution. Typical values of minlvl are -2 or -3.</td>
</tr>
<tr>
<td>infil</td>
<td>Name of the file containing the Fortran unformatted un-filtered input CSP gather.</td>
</tr>
<tr>
<td>timefil</td>
<td>Name of the file containing the picked arrival times of the coherent event.</td>
</tr>
<tr>
<td>outfil</td>
<td>Name of the file containing the Fortran unformatted filtered output CSP gather.</td>
</tr>
</tbody>
</table>

Table 9.1:
Figure 9.1: The unfiltered synthetic crosswell CSP gather containing first arrivals, upgoing and downgoing tubes waves, and reflections.
Figure 9.2: The same CSP gather as in Figure 9.1 after the first arrivals are removed.
Figure 9.3: The same CSP gather as in Figure 9.1 after the upgoing tube waves are removed.
Figure 9.4: The same CSP gather as in Figure 9.1 after the downgoing tube waves are removed.
APPENDIX: Subroutine file "WVFILT.F"

c This file contains a Fortran subroutine "wvfilt" and subroutines it calls.
c The subroutine "wvfilt" uses a wavelet transform to filter out a picked
c coherent event within a specified window in a gather of seismic traces.
c
c The filtering action consists of three steps: (1) Line up the picked event
c in time; (2) Decompose the windowed seismogram along the lines parallel to
c the picked event using the wavelet transform; (3) Reconstruct the seismogram
c by the inverse wavelet transform using only the high resolution (or short
c wavelength) components of the transform. A detailed description of the
c algorithm by Schuster and Sun can be found in the 1992 University of Utah
c Modeling and Tomography Report.
c
c The gather can be a common shot point gather (CSP gather), a common receiver
c point gather, or any other collection of traces. To be specific we will
c refer the gather as a CSP gather regardless of the nature of the collection.
c A wavelet associated with the cubic spline is used for the transform. The
c related coefficients for the wavelet were computed by Minjun Lai.
c
c Note: Within the window, any other event parallel to (or a constant time
shift away from the picked event are also filtered out along with
the picked event. Events nearly parallel to the picked events will
be partially filtered out.
c
Created by Yonghe Sun, Nov, 1991
Last modified by Yonghe Sun, April 10, 1993

c**************************************************************************USAGE**************************************************************************

c The subroutine can be called by:
c
   call wvfilt(data,nrtot,KNSAMP,tarriv,t0,dt,
c               $ dtlow,dtup,ntrmin,ntrmax,window,minlvl,info)

c For a sample program calling this subroutine, see files "README" and
c "wvltf.f".

c The Fortran INCLUDE file "params.inc" must exist to be included as
c part of this subroutine at compilation time. The file "params.inc"
c contains parameter statements for dimensioning arrays used in this
c subroutine and other related subroutines. The "params.inc" must
c have at least the following parameter statements:
c    parameter (nt0=2001,ntr0=601)
c    parameter (nt4win=3*nt0+1)
c    parameter (nsmplmn=-50,nsmplmx=nt0+126,nlvlmn=-20,nlvlmx=0)
c
*******DESCRIPTION OF VARIABLES*************
c dt           KNSAMP-th sample
c t0            -->|                  |<--
t
-------------|------------------------------- 1st trace
c -------------|-------------------------------- 2

--------------------------------------------- ntrmin
-----------------------------------------------
c-----------------------------------------------
*: marks window
^: picked event
*: parallel event

--------------------------------------------- ntrmax
---------------------------------------------
c---------------------------------------------
c--------------------------------------------- ntrtot

---|dtlow| dtup|---

Input and output variables defining the CSP gather:

DATA-------Real input and output array dimensioned DATA(NT0,NTR0).
    The element (DATA(I,J),I=1,KNSAMP),J=:,NTRTOT) initially
    contains the I-th input data sample in the J-th trace.
    DATA is also used for the filtered gather as the output of
    the subroutine.
DT--------Real input. The sample rate.
KNSAMP-----Integer input. The number of data samples per trace. KNSAMP>0.
    The time at the KNSAMP-th sample is T0+(KNSAMP-1)*DT.
NTRTOT-----Integer input. The number of traces in the gather. NTRTOT>1.
    T0--------Real input. The time at the first sample of the traces.

Input variables defining the windowed region containing the picked event:
(Part of the windowed region may be outside the time span of CSP gather)
TARRIV-----Real input array dimensioned TARRIV(NTR0).
(TARRIV(I),I=1,NTRTOT) contains the arrival time of the coherent event at the I-th trace. TARRIV may contain values less than T0 or larger than the maximum sample time T0+(KNSAMP-1)*DT. Only values for NTRMIN.le.I.le.NTRMAX will be used in the subroutine.

NTRMAX------Integer input. The largest trace number in the windowed traces. NTRMIN.lt.NTRMAX.le.NTRTOT.

NTRMIN------Integer input. The smallest trace number in the windowed traces. 1.le.NTRMIN.lt.NTRMAX.

DTLOW------Real input. The lower time bound of the window is at TARRIV(I)-DTLOW for the I-th trace. DTLOW.ge.0. The width of the window in time is DTLOW+DTUP.

DTUP-------Real input. The upper time bound of the window is at TARRIV(I)+DTUP for the I-th trace. DTUP.ge.0. The width of the window in time is DTLOW+DTUP.

Input variable defining the filter low cut:

MINLVL------Integer input. The low cut filter level of resolution.

NLVMLM.1e.MINLVL.lt.NLVMLM=0. Typical values are -2, or -3. The smaller MINLVL, the more effective will be the removal the coherent event at the expense of more filter artifacts.

The wavelet transform is a process of multiresolution decomposition. The input signal is at the highest level of resolution. Only lower and lower frequency components of the signal are present at the lower and lower levels of resolution. High pass or low cut filter is achieved by discarding the a low resolution representation of the signal when reconstructing the signal by the inverse wavelet transform.

Working variables:

WINDOW------Real working array dimensioned WINDOW(NT4WIN,NTR0).

It contains the windowed data from the CSP gather in DATA.

Array declaration parameters: (all set in file "params.inc")

NT0--------Integer. NT0.gt.KNSAMP.

NTR0--------Integer. NTR0.ge.NTRTOT.

NT4WIN------Integer. Used for the declaring the leading dimension of the working variable WINDOW. It specifies the limit on the
number of time samples in the window. The time span of the
windowed region may be larger than the time span of the CSP
gather, depending on the moveout TARRIV of the event and
desired time widths DTLOW and DTUP. The parameter NT4WIN
should be several time larger than NTR0.

NLVLMN----Integer. The limit on lowest level of resolution during the
wavelet transform. Should be smaller that NLVLMX-LOG2(NTR0).

NLVLMX----Integer. The highest resolution level number. For
convenience it will be set to zero: NLVLMX=0. The input
signal to the wavelet transform is assumed to be the signal
at NLVLMX-th (the highest) resolution. The user should not
need to change this parameter.

NSMPLMN----Negative integer. The number (-NSMPLMN) should somewhat
larger than the number of coefficients (currently 35) used
for the wavelet. The user should not need to change this

NSMPLMX----A number larger than NTR0 plus the number of
coefficients (currently 35) used for the wavelet. The user
should not need to change this parameter. Example:
NSMPLMX=NTR0+126.

Error indicator parameter:

INFO------Integer output. A positive value of INFO indicates error.
0 Normal execution of the subroutine "wvfiln";
1 Error: NTRTOT is less than 1;
2 Error: NTRTOT is greater than NTR0;
3 Error: KNSAMP is less than 1;
4 Error: KNSAMP is greater than NTO;
5 Error: DT is less than or equal to 0;
6 Error: DTLOW is less than 0;
7 Error: DTUP is less than 0;
8 Error: NTRMIN is less than 1;
9 Error: NTRMIN is greater than NTRTOT;
10 Error: NTRMAX is less than or equal to NTRMIN;
11 Error: NTRMAX is greater than NTRTOT;
12 Error: Parameter NT4WIN is too small;
13 Error: Parameter NSMPLMN is too large;
14 Error: Parameter NSMPLMX is too small;
15 Error: MINLVL is less than the parameter NLVLMN;
16 Error: MINLVL is greater than NLVLMX.
subroutine wvfilt(data,ntrtot,KNSAMP,tarriv,t0,dt,
$     dtlow,dtup,ntmin,ntmax,window,minlvl,info)
include 'params.inc'
real*4 window(nt4win,ntr0),tarriv(ntr0)
real*4 data(nt0,ntr0)
integer ntmin(ntr0),ntmax(ntr0)

info=0

c Check input information:
if(ntrtot.lt.1)info=1
if(ntrtot.gt.ntr0)info=2
if(KNSAMP.lt.1)info=3
if(KNSAMP.gt.ntr0)info=4
if(dt.le.0.0)info=5
if(dtlow.lt.0.0)info=6
if(dtup.lt.0.0)info=7
if(ntmin.lt.1)info=8
if(ntmin.gt.ntrtot)info=9
if(ntmax.le.ntmin)info=10
if(ntmax.gt.ntrtot)info=11
if(info.gt.0)return

c Transfer the windowed region of DATA into the working variable WINDOW:
call windowing(data,ntrtot,KNSAMP,tarriv,t0,dt,
$     dtlow,dtup,ntwid,ntmin,ntmax,ntrmin,ntrmax,window,ierr)
if(ierr.eq.1)then
    info = 12
    return
endif

c Remove the windowed coherent arrival in WINDOW by wavelet transforms:
call wvltransf(ntwid,ntrmin,ntrmax,window,minlvl,ierr)
if(ierr.gt.0)then
    info=12+ierr
    return
end if

c Transfer the filtered WINDOW back to the windowed region of DATA:
call unwindow(data,ntrtot,KNSAMP,tarriv,t0,dt,
$     ntwid,ntmin,ntmax,ntrmin,ntrmax,window)

return
end

c-------------------------------------------------------------
c subroutine doing wavelet decomposition and reconstruction
c by discarding the low resolution of the windowed data.
c ntwid------integer. Width of the window in time samples; ntwid>0.
c uphi(i,n)--real array. contains the i-th data sample at the n-th level
c of resolution. For the highest level, n=nlvlmx, and the
c uphi(:,nlvlmax) contains the input signal. The uphi(i,n)
c has nonzero elements only for Nmin(n).le.i.le.Nmax(n).
c upsi(i,n)--real array. contains the i-th data detail orthogonal to n-th
c level of resolution. For the highest level, n=nlvlmx, and the
c uphi(:,nlvlmax) contains values of zeros. For n<nlvlmx, the
c uphi(i,n) has nonzero elements only for Mmin(n).le.i.le.Mmax(n).
c------

subroutine wvltfilt( ntwid, ntrmin, ntrmax, window, minlvl, ierr)
include 'params.inc'
real*4 window(nt4win,ntr0)
integer Nmin(nlvlmn:nlvlmx),Nmax(nlvlmn:nlvlmx),
$ Mmin(nlvlmn:nlvlmx),Mmax(nlvlmn:nlvlmx)
real*4 uphi(nsmlmn:nsmlmx,nlvlmn:nlvlmx),
$ upsi(nsmlmn:nsmlmx,nlvlmn:nlvlmx)

Nmin(nlvlmx) = 0
Nmax(nlvlmx) = ntrmax-ntrmin

ierr = 0
if(Nmin(nlvlmx).lt.nsmlmx)ierr=1
if(Nmax(nlvlmx).gt.nsmlmx)ierr=2
maxlvl = nlvlmx
if(minlvl.lt.nlvlmn)ierr=3
if(minlvl.gt.nlvlmx)ierr=4
if(ierr.gt.0)return

c Loop through each time slice in the window:
do 20 nt = 1, ntwid
   if(minlvl.eq.nlvlmx)goto 50

c Take a time slice from WINDOW as input signal:
i = 0
do 10 ntr=ntrmin,ntrmax
   uphi(i,nlvlmx) = window(nt,ntr)
i = i + 1
10    continue

c Decompose the signal:
call decom(uphi, upsi, Nmin, Nmax, Mmin, Mmax, minlvl, ierr)

c Discard the lowest resolution specified:
50     do n=Mmin(minlvl), Mmax(minlvl)
          uphi(n, minlvl) = 0.0
     end do

     if(minlvl.eq.nlvlmax)goto 60

c Reconstruct the signal:
call reccn(uphi, upsi, Nmin, Nmax, Mmin, Mmax, minlvl, maxlvl)

c Transfer the filtered time slice back to WINDOW:
60     i = 0
      do 40 ntr=ntrmin, ntrmax
          window(nt, ntr) = uphi(i, nlvlmax)
          i = i + 1
40         continue

20     continue

return
end

c-----------------------------
c subroutine transfer the filtered data in WINDOW back to DATA:
    subroutine unwindow(data, ntrtot, KNSAMP, tarriv, t0, dt,
    $    ntwid, ntrmin, ntrmax, ntrmin, ntrmax, window)
    include 'params.inc'
    real*4 window(nt4win, ntr0), tarriv(ntr0)
    real*4 data(nt0, ntr0)
    integer ntrmin(ntr0), ntrmax(ntr0)

    do 10 ntr = ntrmin, ntrmax
        if(ntrmin(ntr).gt.KNSAMP.or.ntrmax(ntr).lt.1)goto 10
        ntmin=max(ntmin(ntr), 1)
        ntmax=min(ntrmax(ntr), KNSAMP)
        t = (ntmin-ntmin(ntr))*dt
        do 20 nt = ntmin, ntmax
call interpol(window(i,ntr),t,0.0,dt,ntwid,data(nt,ntr))
    t = t + dt
20  continue
10  continue

return
end

c-----------------------------------------------
c subroutine windowing DATA for filtering and place it in WINDOW.
c ntwid------output, width of the WINDOW in time samples; ntwid>0.
c ntmin(n)----output, the window starts near the ntmin(n) data sample
  in the n-th trace.
c ntmax(n)----output, the window ends near the ntmax(n) data sample
  in the n-th trace.
c WINDOW(i,n)--contains the windowed data. For the n-th trace, the
  (WINDOW(i,n),i=1,ntwid) windows the time interval defined
  by the samples (DATA(j,n),j=ntmin(n),ntmax(n)). The
  picked coherent event has a constant time index i in the
  variable WINDOW(i,n) for (ntmin.1e.n.1e.ntrmax). Since
  the picked traveltimes TARRIV are in general not T0 plus
  integer multiples of DT, interpolation is required to
  obtain WINDOW from DATA for a given trace.
subroutine windowing(data,ntrtot,KNSAMP,tarriv,t0,dt,
$    dtlow,dtup,ntwid,ntmin,ntmax,ntrmin,ntrmax,window,ierr)
include'params.inc'
real*4 window(nt4win,ntr0),tarriv(ntr0)
real*4 data(nt0,ntr0)
integer ntmin(ntr0),ntmax(ntr0)

ierr = 0

c Computing the window bounds:
dtinv=1.0/dt
ndtlow = nint(dtlow*dtinv)
if(ndtlow.lt.0)ndtlow = 0
ndtup = nint(dtup*dtinv)
if(ndtup.lt.0)ndtup = 0
ntwid = ndtlow + ndtup + 1
if(ntwid.gt.nt4win)then
    ierr = 1
return
end if

c Compute WINDOW from DATA:
do 10 ntr = ntrmin,ntrmax
    ntmin(ntr) = 1+nint( (tarriv(ntr)-t0)*dtinv ) - ndtlow
    ntmax(ntr) = ntmin(ntr) + ntwid-1
    t = tarriv(ntr) - ndtlow*dt
do 20 nt = 1,ntwid
    call interpl(data(1,ntr),t,t0,dt,KNSAMP,window(nt,ntr))
    t = t + dt
20   continue
10   continue
$return$
$end$

-------------------------------------------------------------------
c subroutine doing interpolation
c a---------input, data vector containing data.
c   a(1) contains data at time t=t0,
c n---------input, total number of samples.
c dt---------input, sampling rate
c aout------output, the interpolated a(t) at t.

subroutine interpl(a,t,t0,dt,n,aout)
integer n
real*4  a(n),t,dt,dtinv
dtinv=1.0/dt
iwk = nint((t-t0)*dtinv)
p   = (t-t0 - iwk*dt)*dtinv
iwk = iwk+1
i1 = min( max(iwk-1,1), n )
i2 = min( max(iwk ,1), n )
i3 = min( max(iwk+1,1), n )
aout = ( (a(i3)-a(i2)-a(i2)+a(i1))*p+(a(i3)-a(i1)) )*p*0.5 + a(i2)
$return$
$end$

-------------------------------------------------------------------
c Decomposition
c
containing the function u(t) sampled at Ndata points.

c subroutine decomp(ups, upsi, Nmin, Nmax, Mmin, Mmax, minlv, ierr)
include 'params.inc'
parameter (nmx=35)
real*4 beta(0:nmx), h(-nmx:nmx)
integer Nmin(nlvlnm:nlvlmx), Nmax(nlvlnm:nlvlmx),
$ Mmin(nlvlnm:nlvlmx), Mmax(nlvlnm:nlvlmx)
real*4 uphi(nsmplnm:nsmplmx, nvlmn:nlvlmx),
$ upsi(nsmplnm:nsmplmx, nvlmn:nlvlmx)
data beta/0.54173575, 0.30682963, -0.03549797, -0.07780792,
$ 0.02268462, 0.02974681, -0.01214548, -0.01271542,
$ 0.00614143, 0.00579932, -0.00307862, -0.00274528,
$ 0.00154623, 0.00133086, -0.00078046, -0.00065562,
$ 0.00039593, 0.00032676, -0.00020179, -0.00016429,
$ 0.00010326, 0.00008317, -0.00005303, -0.00004234,
$ 0.00002731, 0.00002165, -0.00001410, -0.00001111,
$ 0.00000730, 0.00000572, -0.00000378, -0.00000295,
$ 0.00000196, 0.00000152, -0.00000102, -0.00000079/
ierr=0

c Compute the unspecified coefficients from the specified ones by symmetry:
jmin = -nmx
jmax = +nmx
do 10 j=jmin,jmax
   iw = iabs(j)
   h(j) = beta(iw)
10 continue

c Compute the nonzero element index bounds for various levels of resolution:
k=nlvlnm-1
100 continue
   iw = Nmax(k+1) - jmin
   if(mod(iw,2).ne.0)iw = iw - 1
   Nmax(k) = iw/2
   iw = Nmin(k+1) - jmax
   if(mod(iw,2).ne.0)iw = iw + 1
   Nmin(k) = iw/2

   iw = Nmax(k+1) + jmax - 1
   if(mod(iw,2).ne.0)iw = iw - 1
Mmax(k) = iwk/2
iwk = Nmin(k+1) + jmin - 1
if(mod(iwk,2).ne.0) iwk = iwk + 1
Mmin(k) = iwk/2

if(Nmin(k).lt.nsmlmn .or. Mmin(k).lt.nsmlmn) ierr=1
if(Nmax(k).gt.nsmlmx .or. Mmax(k).gt.nsmlmx) ierr=2
if(ierr.gt.0) return

c Decomposition:
   k = k - 1
   if(k.lt.minlvl) goto 110
   goto 100
110 continue

   do 200 k=nlvlmx-1,minlvl,-1
       do 220 n=Nmin(k),Nmax(k)
           uphi(n,k) = 0.0
           do 240 i=Nmin(k+1),Nmax(k+1)
               j = i-2*n
               if(j.gt.jmax .or. j.lt.jmin) goto 240
               uphi(n,k) = uphi(n,k) + h(j)*uphi(i,k+1)
240     continue
220     continue

   do 230 n=Mmin(k),Mmax(k)
       upsi(n,k) = 0.0
       do 250 i=Nmin(k+1),Nmax(k+1)
           j = 1-i+2*n
           if(j.gt.jmax .or. j.lt.jmin) goto 250
           upsi(n,k) = upsi(n,k) + ((-1)**i)*h(j)*uphi(i,k+1)
250     continue
230     continue

   return
end

c---------------------------------------------------------------
c Reconstruction:
   subroutine recon(uphi,upsi,Nmin,Nmax,Mmin,Mmax,minlvl,maxlvl)
   include'params.inc'
parameter (nmx=35)
real*4 beta(0:nmx),h(-nmx:nmx)
integer Nmin(nlvlnm:nlvlmx),Nmax(nlvlnm:nlvlmx),
$ Mmin(nlvlnm:nlvlmx),Mmax(nlvlnm:nlvlmx)
real*4 uphi(nsmplmx:nsmplmx,nlvlnm:nlvlmx),
$ upsi(nsmplmx:nsmplmx,nlvlnm:nlvlmx)
data beta/0.54173575, 0.30682963,-0.03549797,-0.07780792,
$ 0.02268462, 0.02974681,-0.01214548,-0.01271542,
$ 0.00614143, 0.00579932,-0.00307862,-0.00274528,
$ 0.00154623, 0.00133086,-0.00078046,-0.00065562,
$ 0.00039593, 0.00032676,-0.00020179,-0.00016429,
$ 0.00010326, 0.00008317,-0.00005303,-0.00004234,
$ 0.00002731, 0.00002165,-0.00001410,-0.00001111,
$ 0.00000730, 0.00000572,-0.00000378,-0.00000295,
$ 0.00000196, 0.00000152,-0.00000102,-0.00000079/

c Compute the unspecifie coefficients from the specified ones by symmetry:
        jmin = -nmx
        jmax = +nmx
       do 10 j=jmin,jmax
           iwk = iabs(j)
           h(j) = beta(iwk)*2.0
10 continue

c Reconstruction:
       do 200 k=minlv1+1,nlvlmx,1
      do 220 n=Nmin(k),Nmax(k)
        uphi(n,k) = 0.0
       do 240 i=Nmin(k-1),Nmax(k-1)
           j = n - 2*i
            if(j.lt.jmin .or. j.gt.jmax)go to 240
            uphi(n,k) = uphi(n,k) + h(j)*uphi(i,k-1)
240 continue
        if(k.gt.maxlvl1)go to 220
       do 250 i=Mmin(k-1),Mmax(k-1)
           j = 1-n+2*i
            if(j.lt.jmin .or. j.gt.jmax)go to 250
           uphi(n,k) = uphi(n,k) + ((-1)**n)*h(j)*upsi(i,k-1)
250 continue
220 continue

200 continue
return
end
Acoustic Wave Equation
Waveform Inversion Code (v 1.0)

Changxi Zhou

FORTRAN CODE: wtw-gen.f

PURPOSE:
Invert crosshole data for high resolution velocity structure. (about 6 times greater spatial resolution than the corresponding traveltime tomogram)

ALGORITHM DESCRIPTION: The inversion algorithm minimizes the misfit function:

\[ E = \frac{1}{2} \sum_s \sum_r \int dt \delta p_{rs}(t) w \delta p_{rs}(t). \]  \hspace{1cm} (10.1)

Here \( \delta p_{rs}(t) = p(x_r, t|x_s)_{obs} - p(x_r, t|x_s)_{cal} \) is the seismogram residual. We perform the following operations for one iteration of the acoustic waveform inversion.

1. Solve the acoustic wave equation with zero initial conditions for each shot point by a 4th-order finite-difference algorithm (Schuster, et al. 1989 (PP4.f))
2. Calculate the residual as defined in equation 10.1.
3. Compute the acoustic wave equation in reverse time with zero final condition by using the same 4th-order finite-difference algorithm (PP4.f) for each shot point. Here the seismogram residual is treated as the source time history at the receiver location.
4. Calculate the gradient of the velocity:

\[ \gamma(x) = \frac{1}{c^2(x)} \sum_s \int dt \hat{p}(x, t|x_s)_{cal} \hat{p}'(x, t|x_s), \]  \hspace{1cm} (10.2)

where \( \hat{p}(x, t|x_s)_{cal} \) is calculated in step 1, \( \hat{p}'(x, t|x_s) \) is calculated in step 3.

CODES IN THE PACKAGE: There are 10 files in the FORTRAN code package, including the main waveform inversion code, and the input files.
- \textit{wtw-gen.f}: Waveform inversion code based on the 2-D acoustic wave equation.
- \textit{select.f}: Select the tomogram from the output file velou.wtw
- \textit{wtw.pa}: The parameter file of the common arrays used in the code wtw-gen.f
- \textit{wtw.in}: The input file for wtw-gen.f. When doing forward modeling, wtw.in contains the geometry of the model (see wtw.ina for forward modeling and wtw.inc for inversion)
- \textit{velin}: The velocity distribution of the fault model. (used for forward modeling)
- \textit{coord.peak}: Data file of the coordinates of source and receivers. Generated by wtw-gen.f when doing forward modeling or built by person.
- \textit{hold.wtw}: Traveltime tomogram of the fault model used as initial guess for wtw-gen.f when doing waveform inversion.

\textbf{ALGORITHM IMPLEMENTATION:} User can use wtw-gen.f for synthetic test or for real data inversion. When doing synthetic test, user can use wtw-gen.f to do forward modeling first and generate the data files for waveform inversion. When doing real data inversion, user must construct the input data file in the required format.

\textbf{Synthetic test (fault model):}
- First, copy wtw.ina to wtw.in and run wtw-gen. The code generates coord.peak (coordinates of sources and receives), preal (seismograms), etc.
- Using wt.f to do traveltime inversion to get tomogram hold.wtw for waveform inversion.
- Copy wtw.inc to wtw.in and run wtw-gen to do waveform inversion.

\textbf{Real data inversion:}
- First construct wtw.in, coord.peak, preal, wave.new(wavelet of the sources), density.in (density of the model if assigned by data file), and mut.dat (if the free surface reflections need to be mute out).
- Run wt.f to do traveltime inversion and get hold.wtw for wtw-gen.f
- Run wtw-gen to do waveform inversion.

\textbf{ALGORITHM CAVEATS:} The following are caveats about possible pitfalls in using wtw-gen.f.
- Make sure you choose dx and dt so that stability and dispersion conditions are honored (Levander, 1988). We honor the dispersion criteria by setting dx to be 1/5 the minimum wavelength \( \lambda_{\text{min}} = c_{\text{min}}/(2 \times \nu_m) \) where \( \nu_m \) is the peak wavelet frequency, and set \( c_{\text{max}} dt/dx < 0.56 \) to honor stability, where \( c_{\text{min}} \) and \( c_{\text{max}} \) are the minimum and maximum model velocities, respectively.

- The array dimensions in wtw_pa need to be manually adjusted for large models.

- Make sure the wavelet is the time integral of the seismogram when doing real data inversion.

**CODE INSTALLATION:** Compile code by typing:

```bash
f77 -O3 wtw-gen.f -o wtw-gen
f77 -O3 select.f -o select
```

**REFERENCE**


**APPENDIX**

---

This code is used to inverse velocity using waveform information.

This code is based on the theory:

WAVE EQUATION TRAVELTIME + WAVEFORM INVERSION (Yi Luo and

The code is written by Changxi Zhou, partly adapted from the code wt and
PP4 which were written by Yi Luo and Gerard T. Schuster, respectively.

This code has two functions:
1. forward modeling using finite-difference scheme to solve
   acoustic wave equation with second
   order accuracy in time and fourth order in space.
2. velocity inversion using cross-well seismic data using
   2-4 finite difference scheme.

Depending on your option, this code may invoke these input files:
1. 'wtw.in' is the input parameter
2. 'velin' is the input velocity file which is needed when you set
   kvelin=1 (doing forward modeling, also see wtw.in)
3. 'hold.wtw' contains the result of the last iteration in order to let you
   continue the iteration if the result is not satisfactory. This
   file will be required when you set kvelin=2
4. 'coord.peak' contains the picked travel times, source coordinates and
   geophone coordinates, the weight for the picked travel time or
   traces.
   this file is neccessary for inversion (key=2). The
   traveltimes in this file won't be used when you do waveform inversion,
   so you can put any value in that
   position.
5. 'preal' stores the seismograms for inversion.
6. 'wave.new' store the wavelet of different common shot gathers
   when you invert the real data (nredsou=1)
7. 'mut.dat' stores the position of free surface reflections in
   different gathers, etc, when you want to mute some information
   in the real data to help the inversion.

This code has output files:
1. 'wtw.out' general output file
2. 'hold.wtw' keeps the last iteration's results in order to let do
   inversion processing later
3. 'trares' contains the travel time and waveform residuals
   for each iteration.
4. 'velou.wtw' which contains the initial velocity model, and the
   reconstructed velocity models for each iteration.
5. 'coord.peak' was generated during forward modeling (key=0), which
   will be used in the inversion. As mentioned above, coord.peak contains
   the picked travel time information, and the coordinates of sources
   and geophones.
6. 'preal' stores the synthetic seismograms computed by WTW code
   when key=0, which means do forward modeling.
**include 'wtw.pa'**

**open files**

```
c41=9./8.
c42=-1./24.
```

**read in data the area data where the residual should be equal to zero**

```
call openfiles
```

**Section 2 ---- input parameters**

```
read(k18,*) key,nit,nfree,mufree,kvelin
write(k19,*)'key,nit,nfree,mufree,kvelin'
write(k19,*) key,nit,nfree,mufree,kvelin
```

```
read(k18,*) musou,mugeo
read(k18,*) ndens,denco,cp00,denp
read(k18,*) nrecordt,ntinterp,ntpsou,ndecreas,mudir
na=40
mskipx=1
mskipz=1
k0=na
nopx=na+10
noz1=na+7
noz2=na+10
nzorig=na+7
if(nfree.ge.1) then
  k0=0
  noz1=3
  nzorig=3
end if
write(k19,*)'k0,na,nopx,nox1,nox2,nzorig'
write(k19,*) k0,na,nopx,nox1,nox2,nzorig
```

```
read(k18,*)dvmin,dvmax,vmin,vmax,tmax
dkmin=dvmin*2.
dkmax=dvmax*2.
```
write(k19,*)'dvmin,dvmax,vmin,vmax,tmax'
write(k19,*) dvmin,dvmax,vmin,vmax,tmax

read(k18,*)nxm,nzm,nt,dx,dt
dtx=dt/dx
dxm=dx
if(kvelin.eq.2)read(k14,*)nxm,nzm,dxm
if(kvelin.eq.2)read(k14)nxm,nzm,dxm
if(kvelin.eq.1)read(k15,*)nxm,nzm,dxm
nn=nint(dxm/dx)
nxm1=nopx
nxm2=nzxorig
nxm2=nxm1+(nxm-1)*nn
nxm=nxm2-nxm1+1
nzm=nzm2-nzm1+1
nx=nxm2+(nopx-1)
nz=nzm2+(noz2-1)
nbord=6*(nx-na-na)+6*(nz-na-k0)
write(k19,*)'nx,nz,nt,nbord'
write(k19,*) nx,nz,nt,nbord
write(k19,*)'nxm1,nxm2,nzm1,nzm2,nxm,nzm'
write(k19,*) nxm1,nxm2,nzm1,nzm2,nxm,nzm
if( nbord.gt.nbord0.or.nx.gt.nx0.or.nz.gt.nz0.or.nt.gt.nt0 )then
write(*,*)'nx0,nz0,nt0,nbord0 should be >= ',nx,nz,nt,nbord
stop 0001
end if

read(k18,*) nposou,vm
nts=nint(1./vm/dt)
write(k19,*)'nts,vm'
write(k19,*) nts,vm

call config
write(k19,*)'nxs,nzs'
write(k19,*) (nxss(is),nzss(is),is=1,ns)
write(k19,*)'nxg,nzg for is=1, and ngmax=',ngmax
write(k19,*) (nxg(ig,1),nzg(ig,1),ig=1,ngis(1))
if( key.le.0 ) then
write(k20,*)ns1,xmin1,xmax1,dx1,zmin1,zmax1,dz1,ngmax,tmin1,tmax1
do 8091 is=1,ns1
xs=( nxss(is)-nopx )*dx
zs=( nzss(is)-nzorig)*dx
8091 write(k20,*) is,xs,zs,ngis(is)
   end if
   do 8093 is=1,ns1
       write(k17)( g1(ig,is),ig=1,ngmax)
8093 continue
   do 8094 is=1,ns1
       write(k17)( d2(ig,is),ig=1,ngmax)
8094 continue
  c  End  of  input  (  end  of  section  2  )
  c
  c
  c  Section  3  ----  assign  velocity  and  density  model
    call velmod(kvelin,nxm1,nxm2,nzm1,nzm2)
  c
  c
cmax=0.
cmin=1.e10
cmean=0.
do 1811 k=1,nz
do 1811 j=1,nx
   if(cmax.lt.c1(j,k)) cmax=c1(j,k)
   if(cmin.gt.c1(j,k)) cmin=c1(j,k)
cmean=c1(j,k)+cmean
1811 continue
  cmean=cmean/(nx*nz)
  write(k19,*)'cmin,cmean,cmax',cmin,cmean,cmax
  write(*,*)'cmin,cmean,cmax',cmin,cmean,cmax
  c
  c
  nx11=nxm/mskipx
  nz11=nzm/mskipz
  if(mod(nxm,mskipx).ne.0)nx11=nx11+1
  if(mod(nzm,mskipz).ne.0)nz11=nz11+1
  write(k19,*)'nx11,nz11',nx11,nz11
  c
  do 4231 k=nzm1,nzm2,mskipz
      write(k13,*)(c1(j,k),j=nxm1,nxm2,mskipx)
4231 continue
  c
j=nx/2+1
write(*,2071)(c1(j,k),k=1,nz)
write(k19,2071)(c1(j,k),k=1,nz)
close(k18)
c
End of assigning velocity (end of section 3)
c
assigning source function
call source(nopsou,vm)
approximation of inverse hessan
if( key.ge.1 ) call precod
c
data=0
do 5102 k=nsbin,nssend
ndata=ngis(k)*nt+ndata
5102 continue
write(*,*)' total data number=',ndata
write(k19,*)' total data number=',ndata
mute free surface reflections?
if(mufree.eq.1) then
open(78,file='mut.dat')
do 28 i=1,ns
read(78,*) ms,tmut1(i),mutg1(i),tmut2(i),mutg2(i),mut(i)
tgk(i)=(tmut1(i)-tmut2(i))/float(mutg1(i)-mutg2(i))
28 continue
do 38 i=1,ns
read(78,*) (tmut0(i,j),j=1,ngis(i))
38 continue
close(78)
endif
c
Section 4 ---- loop for iteration
do 800 iter=1,nit
c
if(ndens.eq.1) then
do 4227 iz=1,nz
do 4227 ix=1,nx
c1(ix,iz)=1.e-3*dtx/2.1*sqrt(6500./c1(ix,iz))
c41(ix,iz)=c1(ix,iz)*c41
c42(ix,iz)=c1(ix,iz)*c42
4227 continue
endif

c  adelt=0.
nts=(tmax/dt)*2
tmax=0.
resid=0.
rewind(k11)
if( key.ge.1.and.nsbin.gt.1) then
do 5101 is=1,nsbin-1
do 5101 it=1,nrecordt
read(k11) (pr(it,k),k=1,ng)
5101 continue
end if

c  do 4225 iz=1,nz
do 4225 ix=1,nx
g1(ix,iz)=0.
d2(ix,iz)=0.
4225 continue
pur=0.
do 4224 iz=1,nz
do 4224 ix=1,nx
c2(ix,iz)= c1(ix,iz)**2/c1(ix,iz) *dtx *dtx
4224 continue

c  loop for sources
resddd=0.
do 8001 is=nsbin,nsend
nxs=nxss(is)
nzs=nzss(is)
ng=ngis(is)
write(*,*) 'iter=',iter,' ishot=',is,' nxs,nzs ',nxs,nzs
write(k19,*) 'iter=',iter,' ishot=',is,' nxs,nzs ',nxs,nzs

c  if( key.ge.1 ) then
do 51 it=1,nrecordt
read(k11) (pr(it,k),k=1,ng)
51 continue
if(ninterp.eq.1) then
  do 52 k=1,ng
  do 52 it=nrecordt-1,1,-1
  if(it*2+1.le.nt0) then
\begin{verbatim}
pr(it*2,k)=(pr(it,k)+pr(it+1,k))/2.
pr(it*2-1,k)=pr(it,k)
endif
52   continue
endif
end if
c
call forw(nfree,c41,c42)
call norm
c
if( key.le.0 ) then
do 809 it=1,nt
   write(k11) (pc(it,k),k=1,ngis(is) )
809   continue
call tpicmax(tmax,resdeltp)
do 8092 ig=1,ng
   xg=(nxg(ig,is)-npx)*dx
   zg=(nzg(ig,is)-nzorig)*dx
   write(k20,108)xg,zg,tri(ig,is),iweig(ig,is),ig,ng,is
8092  continue
go to 8001
end if
c
call residual(tmax,ntes,resid,adelt,resdeltp)
call back(nfree,c41,c42)
resddd=resddd+resdeltp
8001  continue
write(*,*) 'resddd=',resddd
if( key.le.0 ) go to 991
adelt=sqrt( adeltndata )
c
cmax=0.
cmean=0.
cmin=1.e10
do 1011 k=1,nz
   do 1011 j=1,nx
      if(cmax.lt.c1(j,k)) cmax=c1(j,k)
      if(cmin.gt.c1(j,k)) cmin=c1(j,k)
      cmean=c1(j,k)+cmean
1011 continue
cmean=cmean/(nx*nz)
do 42211 iz=1,nz
\end{verbatim}
do 4221 i=1, nx
    g1(i, iz) = 2.*g1(i, iz)/cmean**3
4221 continue

continue

! muting c1 in the noinversion area

! preconditioning
    do 4222 i=1, nz
    do 4222 j=1, nx
        d2(i, iz) = g1(i, iz)*wafic(i, iz)
    enddo
4222 continue

! zmaxd2=0.
    do 221 k=1, nz
    do 221 j=1, nx
        if(abs(d2(j, k)).gt.zmaxd2) zmaxd2=abs(d2(j, k))
    enddo
221 continue

stleng=2.*(vmax+vmin)/100.
    if(iter.lt.6) stleng=stleng*2.
    do 202 k=1, nz
    do 20 j=1, nx
        c1(j, k) = c1(j, k) + d2(j, k)/zmaxd2*stleng
            if( c1(j, k).lt.vmin ) c1(j, k)=vmin
            if( c1(j, k).gt.vmax ) c1(j, k)=vmax
    enddo
202 continue
20 continue

! do 4411 i=1, nz
    ! do 4412 i=1, npx
4412 c1(i, iz) = c1(npx+1, iz)
    ! do 4413 i=nx-npx+1, nx
4413 c1(i, iz) = c1(nx-npx, iz)
4411 continue

    ! do 4414 i=1, nx
    ! do 4415 i=1, noz1
4415 c1(i, iz) = c1(i, noz1+1)
    ! do 4416 i=nz-noz2+1, nz
4416 c1(i, iz) = c1(i, nz-noz2)
4414 continue

! cmax=0.
! cmin=1.e10
cmean=0.
do 1711 k=1,nz
do 1711 j=1,nx
if(cmax.lt.c1(j,k)) cmax=c1(j,k)
if(cmin.gt.c1(j,k)) cmin=c1(j,k)
cmean=c1(j,k)+cmean
1711 continue

cmean=cmean/(nx*nz)
c

j=nx/2+1
write(*,2071)(c1(j,k),k=1,nz)
write(k19,2071)(c1(j,k),k=1,nz)
write(*,*)'iter=',iter,' resid=',resid,'  tmax=',tmax
write(k19,*)'iter=',iter,' resid=',resid,'  tmax=',tmax
write(*,*)'rms time res.=',adelt
write(k19,*)'rms time res.=',adelt
write(k19,*)'cmin,cmean,cmax',cmin,cmean,cmax
write(*,*)'cmin,cmean,cmax',cmin,cmean,cmax

4232 do 4232 k=nzm1,nzm2,mskipz
  write(k13,*) (c1(j,k),j=nxm1,nxm2,mskipx)
4232 continue

c
rewind(k14)
write(k14,*)nxm,nzm,dx
do 1401 iz=nzm1,nzm2
write(k14,*)(c1(iz,i),i=nxm1,nxm2)
1401 continue

c
800 continue
991 continue
2071 format(12f6.0)
108 format(3f15.7,5i5)

stop
eend