Q tools: Summary of CREWES software for Q modelling and analysis

Gary F. Margrave

ABSTRACT

CREWES has a number of tools in our Matlab toolbox for modelling constant Q attenuation and analysis of the results. The analysis tools can also be applied to real data. This paper presents an overview of those tools and illustrates their use in a few simple cases. We have tools to generate the theoretical impulse response of 1D anelastic propagation, to build a Q matrix that applies first-order Q transmission effects to a stationary signal, to create an inverse Q matrix that removes Q effects, to calculate the drift time given a Q model, to drift correct a synthetic seismogram, to estimate Q from a seismogram, to build an empirical Q model from standard P-sonic and density logs, and to create a theoretical zero-offset VSP given models of Q, velocity, and density.

INTRODUCTION

The effects of anelastic attenuation are present in all seismic data and range from subtle to dramatic. Despite the unavoidability of these effects, they are often neglected in seismic modelling, perhaps because the theoretical details can be challenging. Also, underlying all models of visco-elasticity, there is an ill-defined and non-unique empirical model of rock behaviour that describes the actual loss mechanism. However, the tools discussed here are based on the constant-Q model (Kjartansson, 1979) which is a widely accepted approximation to observed attenuation behaviour. The constant-Q model refers to a Q that is independent of frequency but may still be a function of position. This model is mathematically simple and provides a good description for the first-order effects of attenuations.

Among the first-order effects are (1) time and frequency dependent attenuation, (2) minimum-phase dispersion, (3) measureable traveltime differences between sonic logging frequencies and seismic frequencies (e.g. drift), and (4) frequency-dependent reflectivity. The various functions described here provide a relatively simple access to these effects and allow them to be studied in relation to other seismic processes such as deconvolution.

These tools are found in the CREWES Matlab library in a subdirectory call Qtools. This library can be downloaded from the CREWES website via the link on the main page named “For Our Sponsors\CREWES Software”. Download will require a CREWES userid which is freely available to all employees of Sponsor companies. After download, the toolbox must be installed in Matlab by clicking on the “Set Path” on the main command ribbon. Once installed, then typing the command “help Qtools” will give the following response:

```
  Contents of Qtools

    These are a collection of tools intended to illustrate and explore the
    constant Q model of attenuation.

    Functions:
    EINAR ... creates the constant Q impulse response (wavelet) based on
            Kjartansson's 1979 paper
    QMATRIX ... creates a matrix which can be applied to a reflectivity or to a
            stationary trace to install minimum phase time-frequency decay.
            This is the transmission effect of Q. Options exist to model
            drift delay relative to Nyquist or, more realistically,
            relative to sonic logging frequencies.
    INVQ ... design an inverse Q matrix. This is done by finding a pseudo
            inverse to the forward problem (qmatrix).  The the pseudo
            inverse is used to allow thresholding of the singular values.
    FAKEQ ... given velocity and density logs, invent a Q log based on
            empirical rules.
    TDRIFT ... calculate the drift time. This is the traveltime difference at
            seismic frequencies minus that at logging frequencies.
    DRIFT_CORR ... given a seismogram computed with logging velocities, apply
            the drift delay to simulate having done the computation with
            check-shot corrected velocities.
    QESTIMATOR ... estimate Q by various methods.
    QZ2QINT ... given a finely layered Q model as a function of z, compute
            the effective interval Q over a large interval. This is the
            expected result of a Q measurement.
    VSPMODELQ ... compute a 1D VSP given a Q model and velocity and density
            logs. Based on Ganley 1981.

    Demonstration scripts
    DEMO_Q_WAVELETS ... show the basic nature of Q wavelets using einar.
    DEMO_CONVMTX_QMTX ... illustrate the creation of stationary and
            nonstationary synthetic seismograms using Toeplitz convolution
            matrices and nonstationary Q matrices.
    DEMO_INVQ ... shows the use of the invq command to render a nonstationary
            seismogram stationary.
    TEST_QESTIMATOR ... demo the use of qestimator.
    TEST_VSPMODELQ ... demo the creation of VSP synthetics using vspmodelq.
```

Note that the function and script names appear in "all caps" in the listing above but their actual names are all lowercase. This is simply a formatting convention employed in Matlab to make the command names stand out from their descriptions.

This paper gives explicit tutorial-level descriptions of the functions: *einar, qmatrix, invq, fakeq, tdrift, qestimator,* and *vspmodelq*. Functions *drift_corr* and *qz2qint* are not discussed here but their use is straightforward to someone who has mastered the other tools.

DESCRIPTION OF THE TOOLS

einar

The function *einar* is named for Einar Kjartansson of Kjartansson (1979). It computes the impulse response of constant-Q wave propagation assuming a 1D homogeneous medium. Kjartansson gives an exact expression for this and various approximations. The function *einar* implements equation 45 of Kjartansson (1979). Script *demo_q_wavelets* is provided to illustrate the use of *einar* and produces Figures 1 and 2.

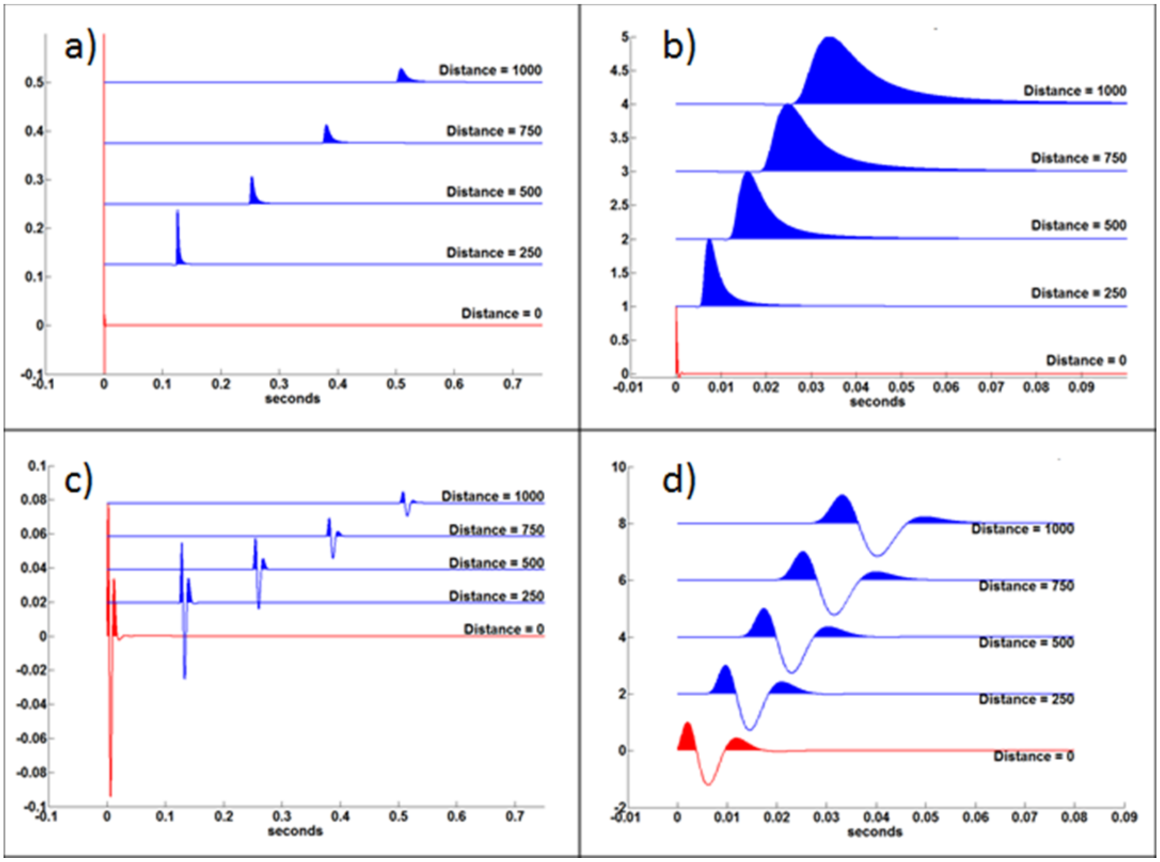

Figure 1: A set of figures created by the script *demo_q_wavelets* that illustrates the output of function *einar*. Panels a) and b) show the impulse response for the Q process assuming a Q of 50 and an initial unit spike. Panel a) shows the actual propagated impulse responses for various propagation distances. In panel b) the impulses are all normalized to a maximum of 1 and most of the propagation delay is removed to ease comparison. Panels c) and d) are similar except that the initial signal is a minimum phase wavelet with a dominant frequency of 100 Hz. In all panels the initial wavelet is in red and the propagated wavelets are in blue.

Figure 1 shows the impulse response of the 1D constant Q process for various propagation distances. In panels a) and b) the initial signal was a unit impulse while in panels c) and d) it was a minimum phase wavelet with a dominant frequency of 100 Hz. The progressive attenuation of high frequencies is clearly indicated by the progressive widening of the wavelet and its diminishing overall amplitude. The minimum phase nature of the Q process is indicated (although not proven) by the causal nature of each wavelet. Panels c) and d) are created from panels a) and b) by stationary convolution of the minimum phase wavelet with each trace in each panel.

Figure 2 shows the amplitude spectra of the wavelets in Figure 1c. The blue curve is the amplitude spectrum of the initial minimum phase wavelet, $|\widehat{w}(f)|$ and the remaining curves are related by

$$|\widehat{w}(t_k, f)| = |\widehat{w}(f)| e^{-\frac{\pi f t_k}{Q}} \tag{1}$$

where t_k is the traveltime for the kth propagation distance. The abrupt flattening of the various spectra at roughly -180 db is indicative of the limit of numerical precision.

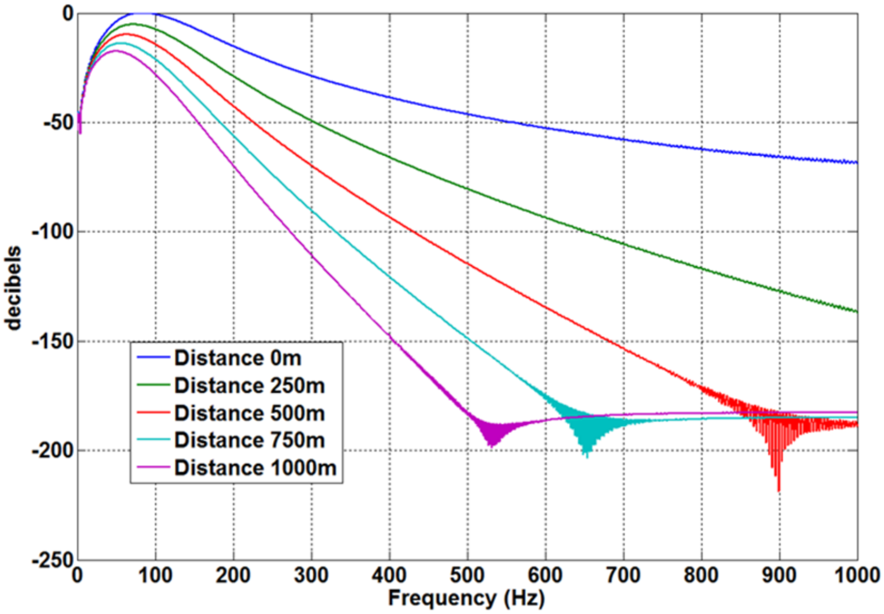

Figure 2: The amplitude spectra of the wavelets in panel c) of Figure 1 are shown. The limit of numerical precision is reached at about -180 db.

QMATRIX

The function *qmatrix* uses *einar* to create a Q matrix which, when applied to a column vector representing either reflectivity or a stationary seismogram, applies the forward Q effect. This is a very good model for Q effects on transmission but does not model the Q reflectivity effect (see *vspmodelq* for this). A Q matrix is a generalization of the familiar Toeplitz matrix which accomplishes a stationary convolution. Figure 3 compares convolution and Q matrices and is created using the script *demo_convmtx_qmtx*. The wavelets in each column of the convolution matrix are identical except for a delay corresponding to propagation. In contrast, the Q matrix shows a progressively decaying wavelet with a greater delay as described by *einar* and illustrated in Figure 1. Figure 4 shows the amplitude spectra of each column of the convolution matrix and the Q matrix. This illustrates the basic stationarity of the former and the nonstationarity of the latter. *qmatrix* allows the Q value to be either a single scalar number or a vector specifying Q for each time. In the latter case, the specified Q should be the average Q. The average Q and local Q are related by

$$Q_{ave}(t_n) = \left[\frac{1}{t_n}\sum_{k=1}^{n}\frac{\Delta t_k}{Q_k}\right]^{-1} \qquad (1)$$

where Q_k and Δt_k and the interval Q and time thickness of the k$^{\text{th}}$ layer and $t_n = \sum_{k=1}^{n}\Delta t_k$.

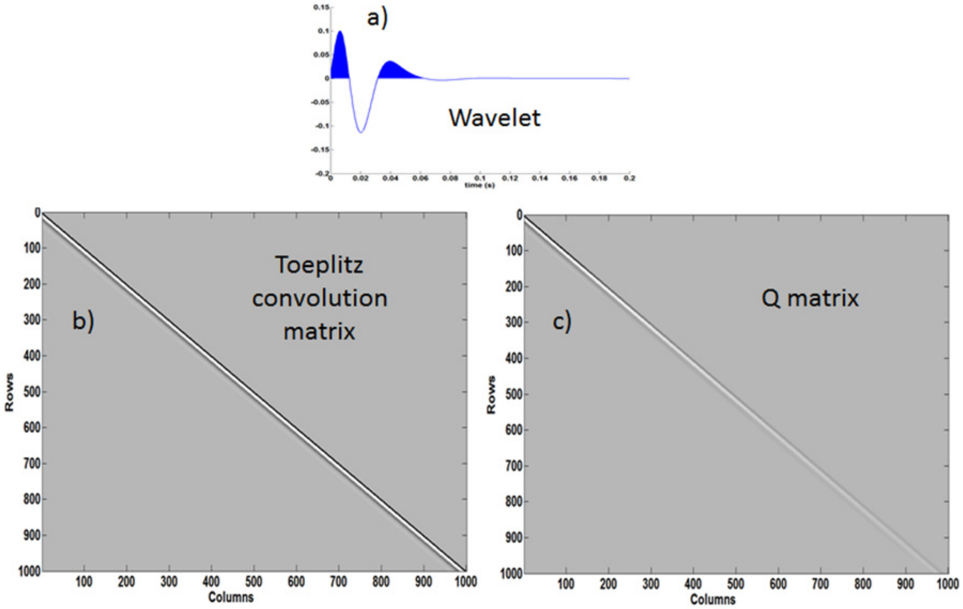

Figure 3: Panels b) and c) show a Toeplitz convolution matrix and a Q matrix (Q=50). Each column of the Toeplitz matrix contains the wavelet shown in panel a) with a progressive delay as required by the convolution process. The values in the Toeplitz matrix are constant along any diagonal. The Q matrix contains wavelets with a delay similar the convolution matrix but there is progressive attenuation of each wavelet and a progressively greater delay.

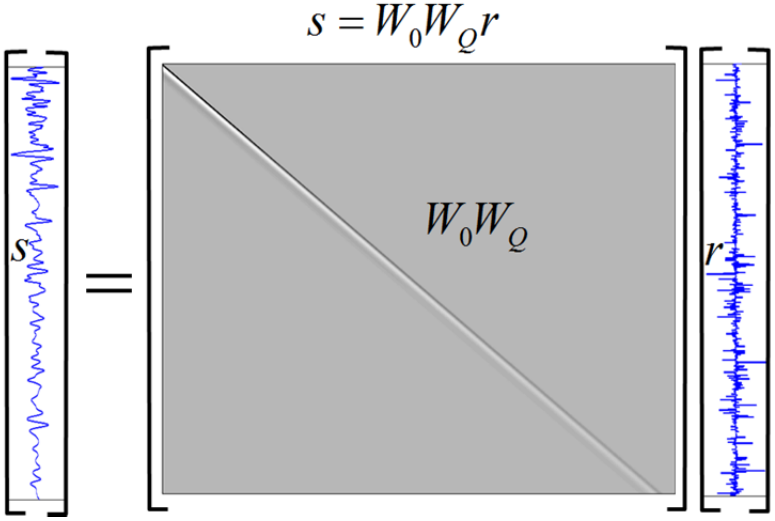

Figure 4: The nonstationary convolutional model of a seismic trace is illustrated. The Q matrix (gray) is shown as the matrix product $\boldsymbol{W_0 W_Q}$ where $\boldsymbol{W_0}$ is a stationary convolution matrix and $\boldsymbol{W_Q}$ is a nonstationary Q matrix containing the Q impulse response. The Q matrix multiplies a column vector containing a reflectivity series (in time) to produce the nonstationary seismogram.

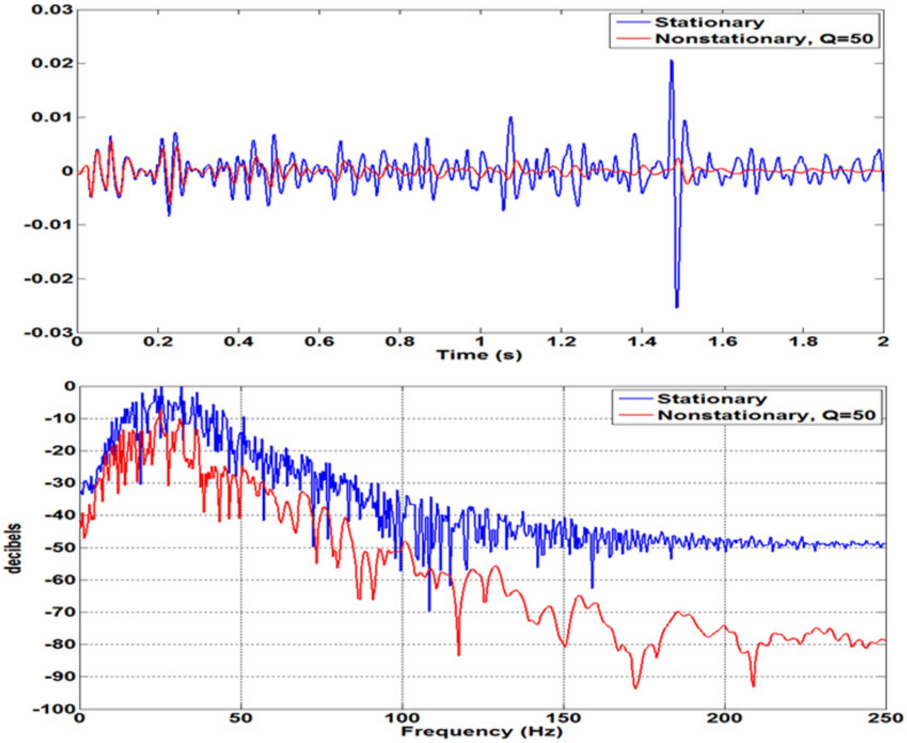

Figure 5: Stationary and nonstationary synthetic seismograms created from the same reflectivity are compared in the time domain (top) and the frequency domain (bottom).

The Q matrix can be constructed mathematically from the matrix product of a Toeplitz matrix containing the source wavelet multiplying from the left into the Q impulse response matrix. It facilitates the nonstationary convolutional model of a seismic trace which gives a much more realistic seismogram than the stationary model. The nonstationary convolutional model is illustrated in Figure 4.

Figure 5 compares the stationary and nonstationary synthetic seismograms as created by the corresponding convolutional models. In the stationary case we have

$$s_{stat} = \boldsymbol{W_0} r \tag{3}$$

where r is a column vector of reflectivity, $\boldsymbol{W_0}$ is a Toeplitz convolution matrix, and s_{stat} is the stationary seismogram as a column vector. Similarly, the nonstationary trace is created by

$$s_{non} = \boldsymbol{W_0 W_Q} r \tag{4}$$

where $\boldsymbol{W_0 W_Q}$ is the Q matrix. As can be seen, the two trace models produce very different results. The differences become more apparent when local spectra are examined as shown in Figure 6.

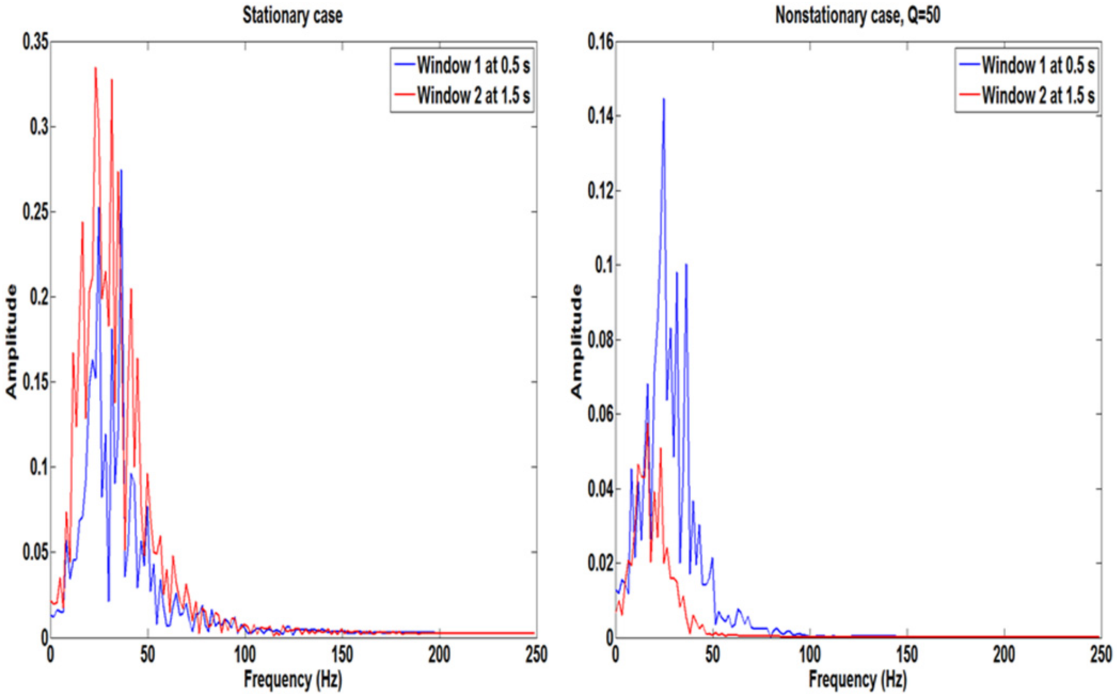

Figure 6: Local Fourier spectra are shown for two 0.6 second windows centered at .5 s and 1.5 s for the seismograms of Figure 5. In the stationary case (left) the spectra are essentially similar with all differences being due to the different reflectivity sequences in each window. In the nonstationary case, the spectrum in the deeper window is dramatically attenuated due to the Q effect.

INVQ

Function *invq* computes the pseudo inverse of a Q matrix assuming a perfect unit-impulse source. This can then be applied to any trace as an inverse Q filter. The pseudo inverse is used (*pinv* in Matlab) to control the behavior of the inverse for small singular values. Very small singular values in the Q matrix correspond to very high attenuation and, in a seismogram, are associated with pure noise or noisy signal. The *pinv* command, and hence *qinv* as well, has a threshold parameter such that singular values smaller than the threshold are not inverted but are simply zeroed.

In data processing, an inverse Q matrix is usually applied before deconvolution. The intent is to remove the Q effect and allow the deconvolution algorithm to estimate the stationary wavelet. However, in equation 3, it can be seen that formally $\boldsymbol{W_0}$ should be inverted first and then $\boldsymbol{W_Q}$ is inverted. However, this is not practical because the stationary wavelet is not known beforehand. Figure 7 was created by script *demo_invq* and illustrates the use of *invq* to render a nonstationary seismogram stationary. That $\boldsymbol{W_Q^{-1}} s_{non} \sim s_{stat}$ means that $\boldsymbol{W_0}$ and $\boldsymbol{W_Q}$ almost commute.

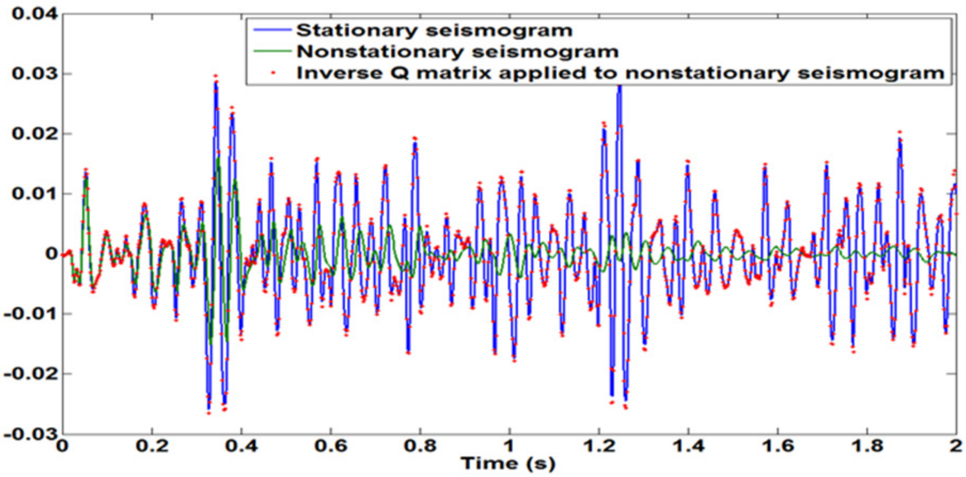

Figure 7: The result of applying an inverse Q matrix to the nonstationary seismogram of equation (4). The result is very nearly equal to the stationary seismogram.

FAKEQ

Function *fakeq* generates a plausible Q structure from assumed linear relationships between p-wave velocity, v_p, and density, ρ, of the form

$$Q_v(z) = Q_0 \frac{v_p(z)-v_1}{v_0-v_1} + Q_1 \frac{v_p(z)-v_0}{v_1-v_0} \tag{5}$$

and

$$Q_\rho(z) = Q_0 \frac{\rho(z)-\rho_1}{\rho_0-\rho_1} + Q_1 \frac{\rho(z)-\rho_0}{\rho_1-\rho_0} \tag{6}$$

where Q_0, Q_1, v_0, v_1, ρ_0, ρ_1 are all specified constants that determine the linear relationships. A final Q is then determined from

$$\frac{1}{Q} = \frac{1}{Q_v} + \frac{1}{Q_\rho}. \tag{7}$$

Figure 8 shows the results for a particular instance derived from a blocked well. Note that the fluctuations in Q come from those in both velocity and density logs.

TDRIFT

Function *tdrift* computes the so-called drift time given a Q model. The theory of constant Q predicts that velocity must be frequency dependent. That is, two monochromatic waves having frequencies f_1 and f_2 respectively will propagate at different wavespeeds $v(f_1)$ and $v(f_2)$. Furthermore, the frequency dependence of wavespeed is given by

$$v(f) = v_0 \left[1 + \frac{1}{\pi Q} ln\left(\frac{f}{f_0}\right)\right] \tag{8}$$

(Aki and Richards, 2002, equation 5.81). Here v_0 is the velocity at frequency f_0 and is called the reference velocity and frequency. Usually we are concerned with the velocity variation from the dominant frequency of well logging, about 12500 Hz, to the dominant frequency of seismic exploration, about 40 Hz. This is a frequency span of about 2.5 orders of magnitude. The essential behaviour of equation 8 is shown in Figure 9 for a range of Q values. It follows that a synthetic seismogram computed from well logs that are uncorrected for this effect will predict reflection event times that are systematically too small (early) compared to those seen in seismic data. The difference between the event time at seismic frequencies and at sonic logging frequencies is call the drift time. Given a layered medium, the vertical traveltime is given by

$$t(z_n, f) = \sum_{k=2}^{n} \frac{\Delta z_k}{v_k(f)} \tag{9}$$

where $\Delta z_k = z_k - z_{k-1}$ is the layer thickness, and $v_k(f)$ is the frequency dependent phase velocity of the kth layer. Thus the drift time is

$$t_{drift}(z_n) = t(z_n, f_s) - t(z_n, f_0) \tag{10}$$

where $f_s = 40$ Hz is the dominant seismic frequency and $f_0 = 12500$ Hz is the dominant well logging frequency. Figure 10 shows the drift time as calculated by *tdrift* for the well logs of Figure 8. Note that *tdrift* calculates a one-way time so the results must be doubled for surface reflection data. Script *demo_drift* creates these figures.

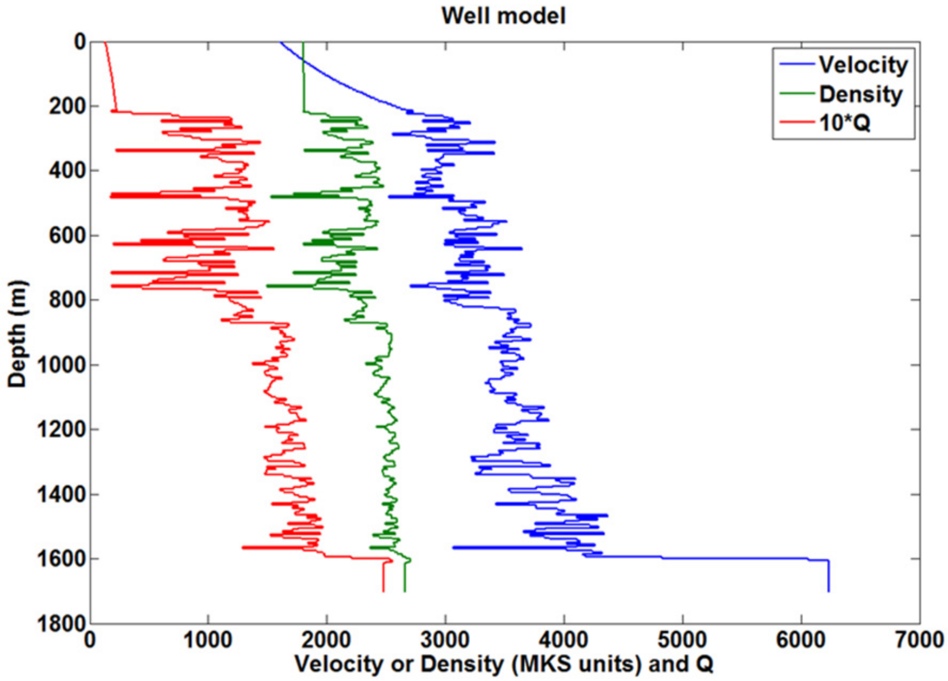

Figure 8: Velocity and density models derived from blocked well logs and the Q model that results from equations 5,6, and 7, given the values $Q_0 = 20$, $Q_1 = 500$, $v_0 = 1500$, $v_1 = 4500$, $\rho_0 = 1800$, $\rho_1 = 3000$.

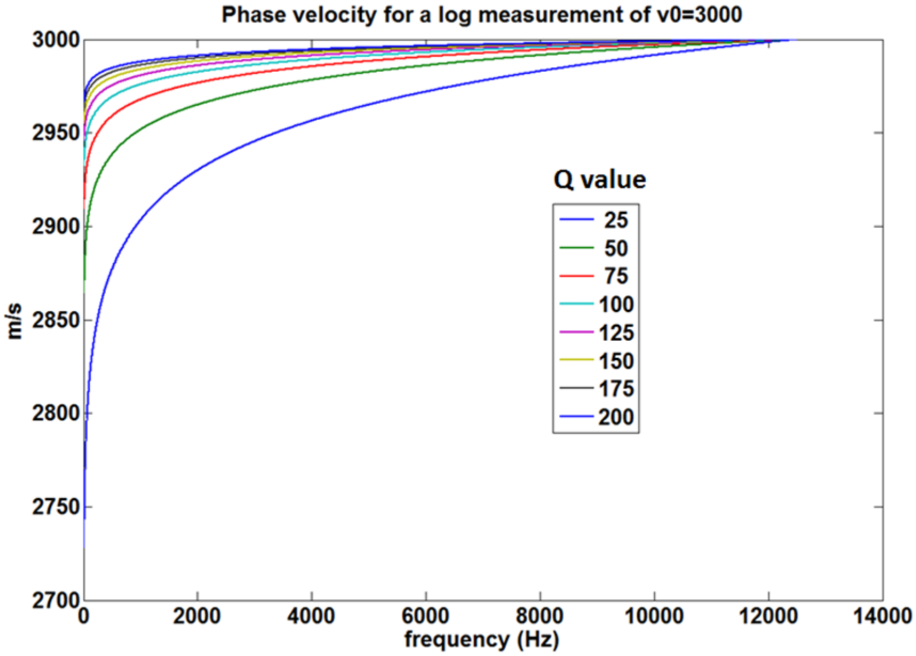

Figure 9: The frequency variation of a reference velocity $v_0 = 3000$ m/s at reference frequency $f_0 = 12500$ Hz. The variation is that predicted by equation 8.

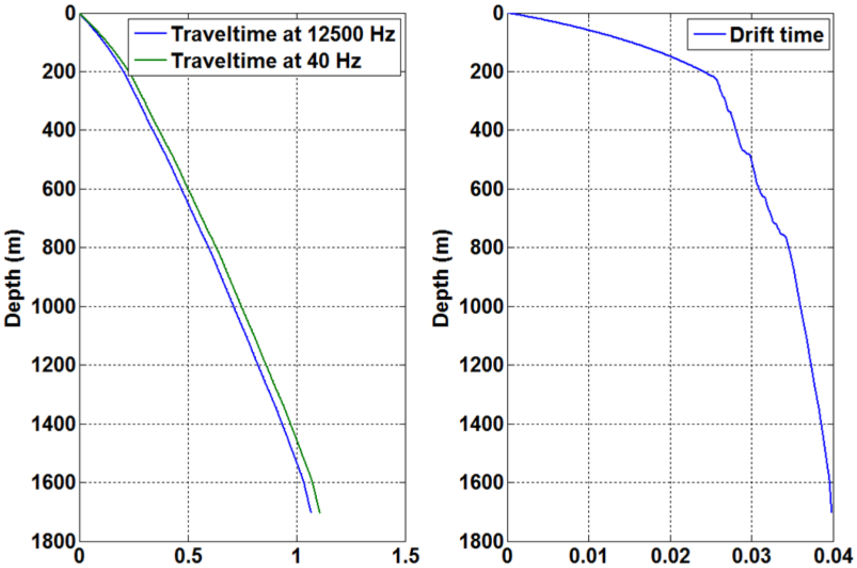

Figure 10: The traveltimes at two different frequencies are shown for the logs in figure 8. The difference in traveltimes, or drift time, is also shown.

QESTIMATOR

The function *qestimator* is designed to provide easy access to several different methods of Q estimation. Presently there are two methods available: spectral ratioing, and spectral matching, with more methods anticipated soon. To understand these methods, consider equation (1) for two times $t_1 < t_2$. We have

$$|\hat{w}(t_1,f)| = |\hat{w}(f)|e^{-\frac{\pi f t_1}{Q}} \tag{11}$$

and

$$|\hat{w}(t_2,f)| = |\hat{w}(f)|e^{-\frac{\pi f t_2}{Q}} \tag{12}$$

Take the ratio of these equations and then the natural logarithm to form the *log spectral ratio* or lsr

$$lsr(Q,\Delta t,f) = ln\frac{|\hat{w}(t_2,f)|}{|\hat{w}(t_1,f)|} = -\frac{\pi f \Delta t}{Q} \tag{13}$$

where $\Delta t = t_2 - t_1$. Equation 13 predicts that the lsr should be a linear function of frequency with zero intercept and slope $-\pi\Delta t/Q$. Thus a least squares fit of a first order polynomial to the lsr will predict the interval Q between the times t_1 and t_2.

The spectral division in equation 13 can be a problem because of noise and also because of notches in the spectra caused by reflectivity. A first fix for this is to somehow smooth the amplitude spectra involved before taking the ratio. Also, since these are local spectra, the Fourier transform may not be the best method of spectral estimation. Function *estimator* offers the option to smooth the Fourier spectra, or to compute Burg or multitaper spectra which are naturally smooth.

The second estimation method, called spectral matching finds Q by minimizing the objective function

$$\vartheta_Q = \left\| |\hat{w}(t_2,f)| - |\hat{w}(t_1,f)|e^{-\frac{\pi f \Delta t}{Q}} \right\| \tag{14}$$

so Q is estimated by

$$Q_{est} = min_Q(\vartheta_Q). \tag{15}$$

This just means that we scan over the range of possible Q's (using integer values) to find the one which makes equation 14 minimal.

Figures 11 and 12 are generated by script *test_qestimator* and show the performance of both methods on columns extracted from a Q matrix. This is the easiest possible case for Q estimation and is analogous to testing on a very high quality VSP wavefield. Whenever reflections are involved, the situation becomes much more challenging as reflectivity causes notches and other spectral features that make the estimation more challenging.

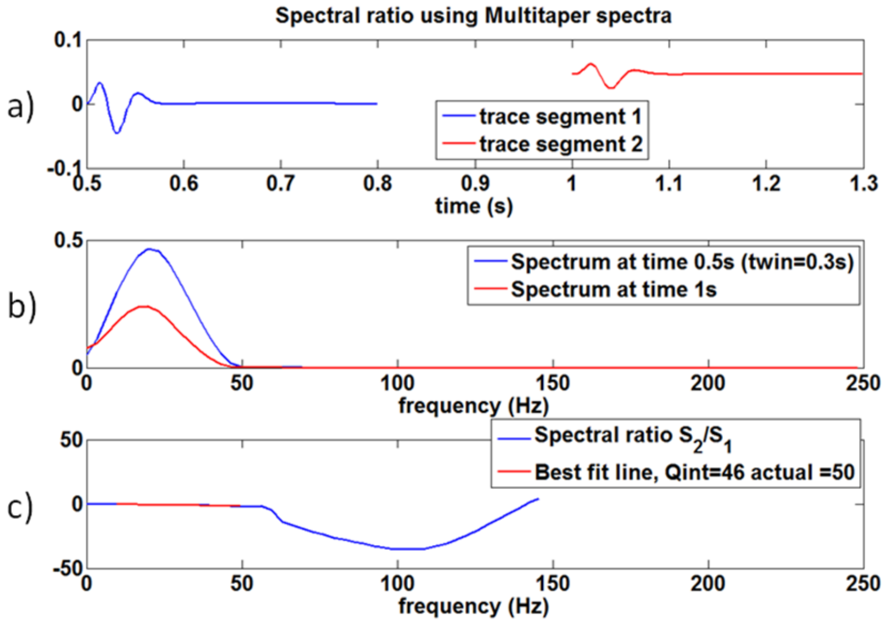

Figure 11: Performance of the spectral ratio method using columns of a Q matrix where the Q value was 50. a) Shows the two wavelets used and their relative times. b) Shows the multitaper spectra of the two wavelets. c) Shows the log spectral ratio in blue and in red is the portion of the frequency band over which a linear fit was performed. The estimated Q was 46.

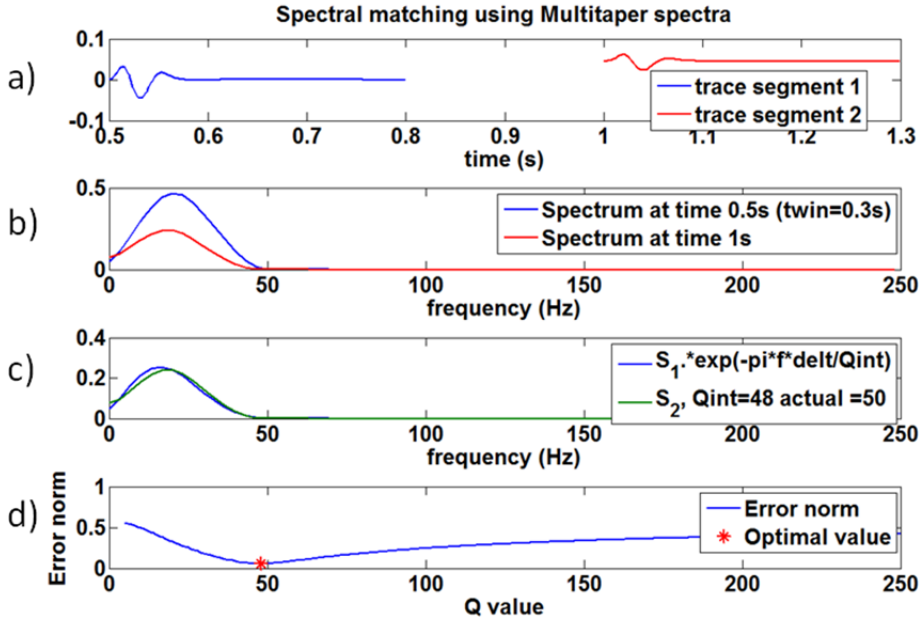

Figure 12: Similar to Figure 11 except that the Q estimation method is spectral matching. Panels a) and b) show the two wavelets in the time and frequency domain. Panel c) shows the match between the two spectra using the estimated Q and panel d) shows the objective function (equation 14) with at marker at the minimum. The estimated Q is 48.

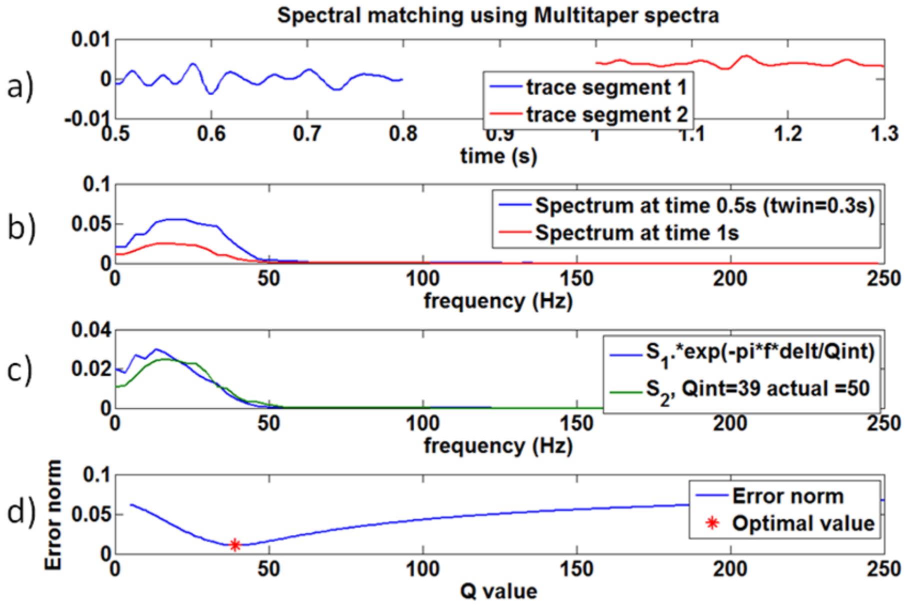

Figure 13: Similar to Figure 12 except that trace segments from a nonstationary reflection seismogram constructed from the same Q matrix are used. The Q estimate is much worse in this case.

VSPMODELQ

The final tool discussed here is function *vspmodelq*. This is an implementation of the exact 1D theory described in Ganley (1981) and then again in Dong and Margrave (2003). An outline of the theory is also contained in this paper's Appendix. The function calculates a zero offset VSP assuming a 1D medium specified by velocity, density, and Q models as in Figure 8. Unlike the Q matrix approach described earlier, this method handles both the transmission and reflection effects for primaries and all multiples. (Q matrix is a transmission model for primaries only). This is of special interest to study the theoretically predicted frequency dependence of reflection coefficients.

The essential idea is suggested by Figure 14. Here D_k and U_k are the down-going and up-going fields at the top of layer k. The A_k are 2x2 layer propagator matrices such that

$$\begin{bmatrix} D_k \\ U_k \end{bmatrix} = A_k \begin{bmatrix} D_{k+1} \\ U_{k+1} \end{bmatrix}. \qquad (16)$$

The A_k include the effects of:

- Frequency dependent reflection and transmission
- Frequency dependent phase velocity propagation
- Frequency dependent attenuation
- Reverberations

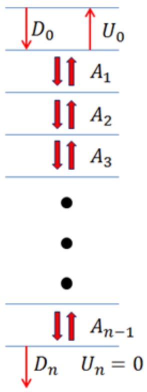

Figure 14: Illustration of the algorithm for vspmodelq.

Concatenating the A_k, $A = A_1 A_2 A_3 \cdots A_{n-1}$, connects the solution in the upper layer to the lower half-space:

$$\begin{bmatrix} D_0 \\ U_0 \end{bmatrix} = A \begin{bmatrix} D_n \\ 0 \end{bmatrix}. \tag{17}$$

Assuming D_0 is known, we have two equations with two unknowns U_0 and D_n. Once these are known, the solution can be constructed at any receiver depth

$$\begin{bmatrix} D_r \\ U_r \end{bmatrix} = A_r \cdots A_{n-2} A_{n-1} \begin{bmatrix} D_n \\ 0 \end{bmatrix} \tag{18}$$

Features of this method include

- Normal incidence, full waveform solution.
- Plane waves (no geometrical spreading).
- Free-surface at top and half-space at bottom.
- Model parameters $v_p, \rho, Q,$ can be specified well-log sampling.
- v_p obeys Aki-Richards dispersion law $v_p(f) = v_p(f_0)\left[1 - \frac{1}{\pi Q} ln \frac{f}{f_0}\right]^{-1}$. Here $f_0 \approx 12500\text{Hz}$ is the sonic logging frequency.
- Q is taken to be frequency independent (constant-Q model).
- All multiples are included.
- Frequency dependent reflectivity is included.
- Up-going and down-going fields are calculated separately.
- Drift effect is automatically simulated.

Results from the script *test_vspmodelq* are shown in Figures 15-19. These are based on the well-log model shown in Figure 8. Figure 15 shows the total field (downgoing + upgoing) while Figures 15 and 16 show the downgoing and upgoing fields respectively. The separation of these fields is exact as they are computed separately in the algorithm. All three of these figures are in true relative amplitude to each other and displayed without clipping. This means that the upgoing field appears quite weak. Figure 17 is provided to show more detail in the upgoing field by showing it scaled independently but still without clipping. Also shown on Figure 15 are first arrival traveltimes at each receiver depth as predicted with logging velocities and also as predicted with the velocities adjusted to the dominant frequency of the source wavelet. It is apparent that the latter match the data much better. An enlarged version of this is shown in Figure 18. This illustrates that the modelling code faithfully reproduces the drift effect.

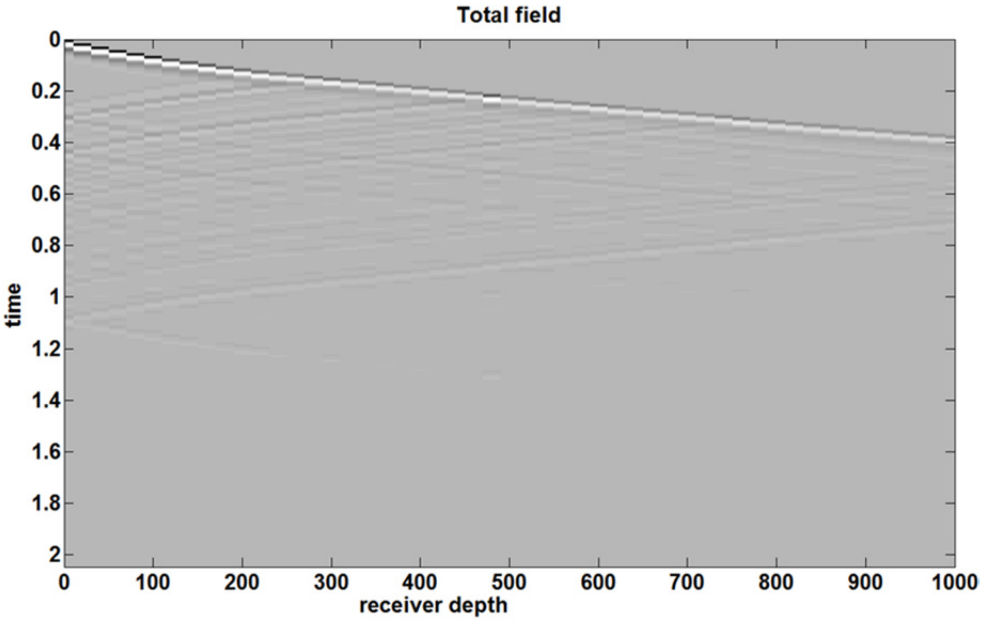

Figure 15: The result of VSP modelling for the well-log model of Figure 8. The source wavelet is a minimum phase wavelet with a dominant frequency of 30 Hz asin Figure 3. This image shows the total wavefield which is the sum of upgoing and downgoing fields.

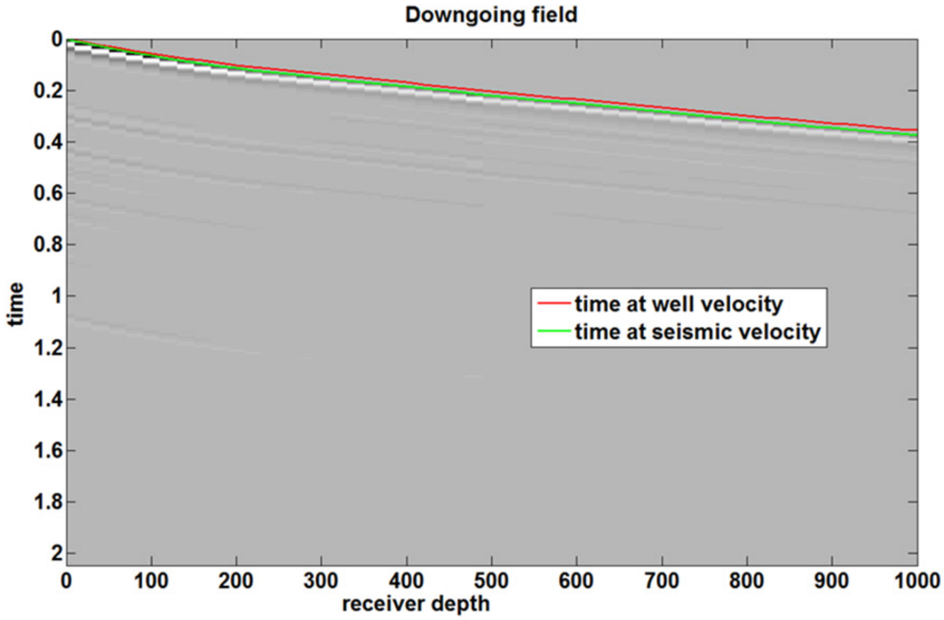

Figure 16: The downgoing field for the wavefield of Figure 15. In addition to the primary downgoing wave, several surface multiples are visible. Colored lines indicate predicted traveltimes at well and seismic velocities. See also Figure 19.

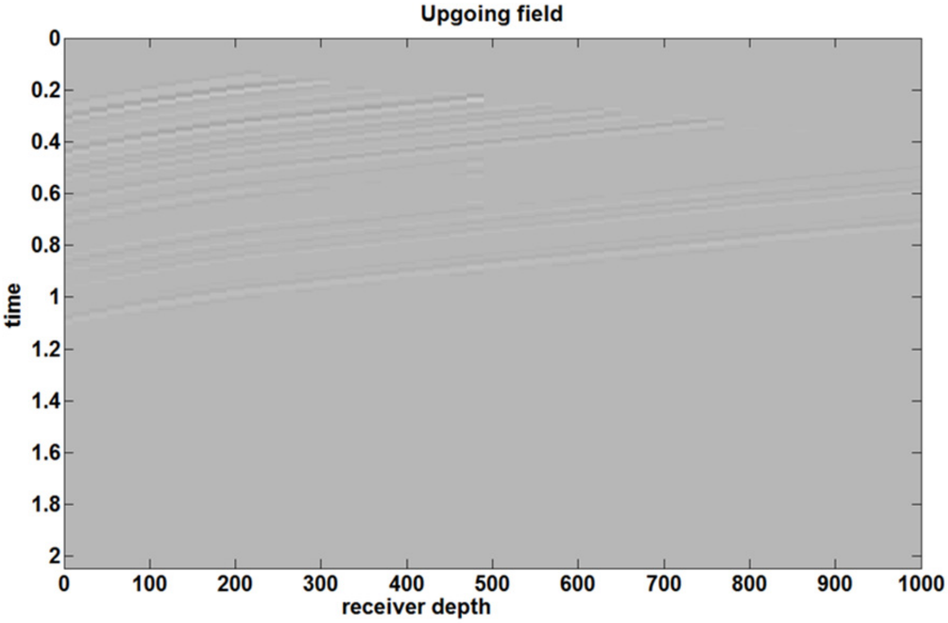

Figure 17: The upgoing field corresponding to Figures 15 and 16. This is displayed in true relative amplitude, relative to those other figures and so appears weak. Figure 18 shows the upgoing field independently normalized where more detail is visible.

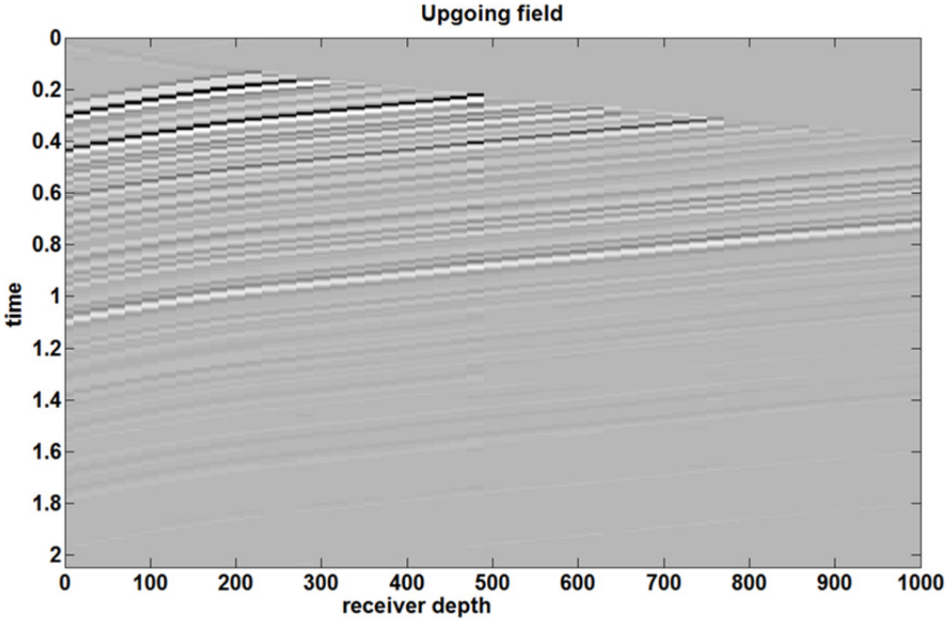

Figure 18: Similar to Figure 17 except that the data have been independently normalized so that the details of the upgoing field can be observed.

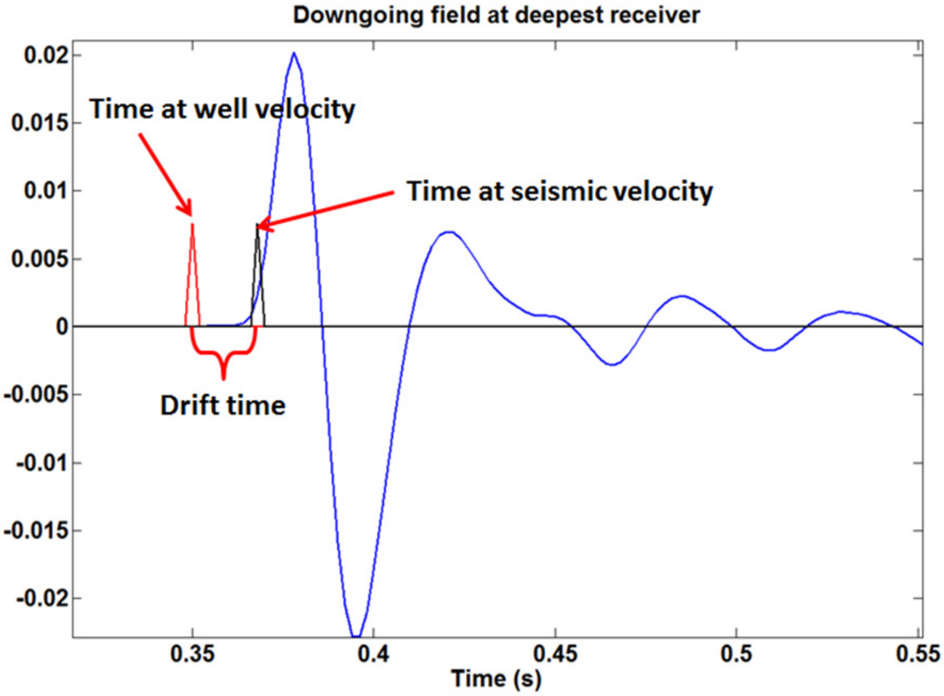

Figure 19: The deepest receiver trace for the downgoing field of Figure 16 is shown enlarged. Spikes are superimposed at the traveltime predicted from the well velocities and at that predicted at seismic frequencies. The difference between these times is the drift time (see Figures 9 and 10).

CONCLUSIONS

CREWES has developed a useful library of tools to study attenuation in seismic wave propagation. A number of these tools have been collected into a single subdirectory called Qtools and were described here. These tools are all based on the constant-Q model of attenuation, which assumes that Q is independent of frequency. The impulse response of the model was examined and the fundamental time-frequency decay with minimum phase dispersion was illustrated. A simple tool, the Q matrix, was examined and shown to be a generalization of the convolution matrix to nonstationary attenuation. A stationary seismogram can be converted to a nonstationary one with transmission Q effects by application of the appropriate Q matrix. The Q effect can be inverted by generating the inverse of the Q matrix.

Tools to calculate an empirical Q log from velocity and density logs were also shown. In addition, the drift effect was examined and a tool to estimate Q from data was shown. Finally, a 1D VSP modelling code that includes the effects of attenuation was examined. This code computes the complete multiply scattered field with attenuation, frequency dependent phase velocity, and frequency dependent reflectivity. It is hoped that these codes can serve as a nucleus from which to study attenuation in real data and to create more complex modelling codes.

ACKNOWLEDGEMENTS

I am grateful to the Sponsors of CREWES and NSERC for supporting this work.

A number of students have helped with these codes over the years. In particular, I wish to recognize the contributions of Peng Cheng, Linping Dong, Carlos Montana, and Heather Lloyd.

REFERENCES

Aki K. and P. G. Richards, 2002, Quantitative Seismology 2nd Edition, University Science Books.

Dong L. and G.F. Margrave, 2003, Synthetic seismograms with attenuation, in the 15th annual CREWES report.

Ganley, D. C., 1981, A method for calculating synthetic seismograms which include the effects of absorption and dispersion, Geophysics, **46**, 1100-1107.

Kjartansson, E, 1979, Constant Q-wave Propagation and Attenuation, Journal of Geophysical Research, **84**, 4737-4748.

APPENDIX: TECHNICAL DESCRIPTION OF VSPMODELQ

Dong and Margrave (2003) describes the basic idea but there are a number of mistakes in the equations so Ganley (1981) is the better reference. The description here is for 1D propagation without attenuation whereas Ganley includes attenuation. That just means the description here is relatively simple while the code *vspmodelq* does include attenuation. We will work in the frequency domain and develop expressions for the up and downgoing wavefields for any frequency. These can then be inverse Fourier transformed to give the time-domain fields.

Consider layers j and j+1 in a layered medium. Let D_j and U_j denote the downgoing and upgoing fields for a single frequency at the top of layer j and similarly for layer j+1. Then let D_j' and U_j' denote the downgoing and upgoing fields at the bottom of layer j and similarly for layer j+1. We wish to develop a transformation that takes D_j and U_j into D_{j+1} and U_{j+1}. We will call this transformation a single step *wavefield extrapolation* and the final result is equation A13.

$D_j \downarrow$	$U_j \uparrow$
$D_j' \downarrow$	$U_j' \uparrow$
$D_{j+1} \downarrow$	$U_{j+1} \uparrow$
$D_{j+1}' \downarrow$	$U_{j+1}' \uparrow$

First, we will convert D_j and U_j into D_j' and U_j' which is purely a propagation effect in a constant velocity layer. In 1D, this propagation is simply a time delay and, since we are working in the frequency domain, this is a linear phase shift (including Q as in *vspmodelq* means that the propagation velocity becomes frequency dependent and there is also a real-valued exponential giving the spectral decay, see Ganley (1981)). Thus

$$D_j' = e^{-2\pi i f h_j / v_j} D_j \text{ and } U_j' = e^{2\pi i f h_j / v_j} U_j \qquad \text{A1}$$

where f is frequency, h_j is the layer thickness, and v_j is the layer velocity. Note the sign difference in the exponentials in these two equations. This is because they have opposite time relations meaning that D_j' is later in time than D_j while U_j' is earlier in time than U_j. For this reason, the two exponentials must have different signs but the choice of which is positive and which is negative is dependent upon the Fourier transform sign convention in use and is therefore software dependent. The choice made here is consistent with *fftrl* and *ifftrl* in the CREWES Matlab library. If we define $P_j = e^{-2\pi i f h_j / v_j}$, meaning the propagator for layer j, we can re-write A1 as

$$D_j' = P_j D_j \text{ and } U_j' = P_j^* U_j \qquad \text{A2}$$

where * indicates complex conjugation. (As mentioned above, the propagator is modified in *vspmodelq* to use frequency dependent phase velocities for propagation and also includes exponential decay of the amplitude spectrum with time and frequency.)

Now we need to relate D_j' and U_j' to D_{j+1} and U_{j+1}. This is done using reflection and transmission coefficients. We define the reflection coefficient for incidence from above as

$$R_j = \frac{v_{j+1}\rho_{j+1} - v_j\rho_j}{v_{j+1}\rho_{j+1} + v_j\rho_j}. \qquad \text{A3}$$

where ρ_j is the layer density. Note that the definition here is opposite in sign to that of Ganley (1981) and this will result in sign differences in many subsequent equations. Note also, that the extension to Q required that equation A3 be modified to use complex-valued and frequency dependent phase velocities. This makes the reflectivity frequency dependent. The reflection coefficient for incidence from below is denoted $R_j' = -R_j$. The transmission coefficients for incidence from above and below are $T_j = 1 - R_j$ and $T_j' = 1 - R_j'$. Now, consider the upgoing field at the bottom of layer j (U_j'). This will consist of a reflection from D_j' and a transmission from U_{j+1}, this is

$$U_j' = -R_j D_j' + T_j' U_{j+1} \qquad \text{A4}$$

similarly

$$D_{j+1} = T_j D_j' - R_j' U_{j+1}. \qquad \text{A5}$$

These are equations (30) and (31) of Ganley (1981). From A5 we have

$$D_j' = \frac{1}{T_j}\left(D_{j+1} - R_j U_{j+1}\right). \qquad \text{A6}$$

where $R_j' = -R_j$ has been used. Substituting A6 into A4 gives

$$U_j' = -\frac{R_j}{T_j} D_{j+1} + \left(T_j' - \frac{R_j R_j'}{T_j}\right) U_{j+1} = -\frac{R_j}{T_j} D_{j+1} + \frac{1}{T_j} U_{j+1} \qquad \text{A7}$$

where the last step follows from the relations between the reflection and transmission coefficients. Now, use equations A2 in A6 and A7 to get

$$P_j D_j = \frac{1}{T_j}\left(D_{j+1} - R_j U_{j+1}\right) \qquad \text{A8}$$

and

$$P_j^* U_j = -\frac{R_j}{T_j} D_{j+1} + \frac{1}{T_j} U_{j+1}. \qquad \text{A9}$$

Noting that $P_j P_j^* = P_j^* P_j = 1$, we have

$$D_j = \frac{P_j^*}{T_j} D_{j+1} - \frac{R_j P_j^*}{T_j} U_{j+1} \tag{A10}$$

and

$$U_j = -\frac{P_j R_j}{T_j} D_{j+1} + \frac{P_j}{T_j} U_{j+1}. \tag{A11}$$

We can write equations A10 and A11 as the matrix equation

$$\begin{bmatrix} D_j \\ U_j \end{bmatrix} = \begin{bmatrix} \frac{P_j^*}{T_j} & -\frac{R_j P_j^*}{T_j} \\ -\frac{P_j R_j}{T_j} & \frac{P_j}{T_j} \end{bmatrix} \begin{bmatrix} D_{j+1} \\ U_{j+1} \end{bmatrix} \tag{A12}$$

or

$$\begin{bmatrix} D_j \\ U_j \end{bmatrix} = A_j \begin{bmatrix} D_{j+1} \\ U_{j+1} \end{bmatrix} \tag{A13}$$

where

$$A_j = \begin{bmatrix} \frac{P_j^*}{T_j} & -\frac{R_j P_j^*}{T_j} \\ -\frac{P_j R_j}{T_j} & \frac{P_j}{T_j} \end{bmatrix} \tag{A14}$$

is the layer extrapolation matrix. Note that A_j takes the solution at the top of the deeper layer j+1 to the solution at the top of the shallower layer j, and is therefore an upward extrapolation matrix.

Consider now a system of n layers above a half space. We can relate the solution in layer 1 to that in layer n+1 (the half space) by repeated application of equation A13. That is

$$\begin{bmatrix} D_1 \\ U_1 \end{bmatrix} = A_1 A_2 \cdots A_n \begin{bmatrix} D_{n+1} \\ U_{n+1} \end{bmatrix} = A \begin{bmatrix} D_{n+1} \\ U_{n+1} \end{bmatrix} \tag{A15}$$

where we have defined $A_1 A_2 \cdots A_n = A$. We now need to specify the wavefields in equation A15 in a way that solves our VSP problem. First, if our only source is at the top of the first layer, then we can set $U_{n+1} = 0$ because there can be no upcoming wave from the half space. Second, let our source be specified by a wavelet W and then we have $D_1 = W + R_0 U_1$. Thus we can write equation A15 as

$$\begin{bmatrix} W + R_0 U_1 \\ U_1 \end{bmatrix} = A \begin{bmatrix} D_{n+1} \\ 0 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} D_{n+1} \\ 0 \end{bmatrix}. \qquad \text{A16}$$

Equation 16 is two equations in two unknowns with the solution

$$D_{n+1} = \frac{W}{a_{11} - R_0 a_{21}} \qquad \text{A17}$$

and

$$U_1 = \frac{a_{21} W}{a_{11} - R_0 a_{21}}. \qquad \text{A18}$$

Algorithm:

1. Given a model of n layers over a half-space, compute the layer extrapolation matrices (equation A14) and the total extrapolation matrix (equation A15).
2. Given a wavelet, solve for the upgoing field in layer 1 and the downgoing field in the half space.
3. Compute the field at any receiver depth by extrapolating the downgoing field in the half space up to the receiver depth using the appropriate extrapolation matrices.
4. Repeat for all wavelet frequencies and then inverse Fourier transform.