Durham E-Theses

Inversion of seismic reflection data from the Gialo Field, Sirte Basin

BenGheit, Ali O.

Inversion of Seismic Reflection Data from the Gialo Field, Sirte Basin

Inversion of Seismic Reflection Data from the Gialo Field, Sirte Basin

by

Ali O. BenGheit

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Department of Geological Sciences

University of Durham

September 1996

10 MAR 1997

ACKNOWLEDGEMENTS

ABSTRACT

This project is concerned with the development of software to invert seismic reflection data for acoustic impedance, with application to the YY-reservoir area in Gialo Field, Sirte Basin. The problem was that of inverting post-stack seismic reflection data from two seismic lines into impedance profiles. The main input to the inversion process is an initial guess, or initial earth model, of the impedance profile defined in terms of parameters. These parameters describe the impedance and the geometry of the number of layers that constitute the earth model. Additionally, an initial guess is needed for the seismic wavelet, defined in the frequency domain using nine parameters.

The inversion is an optimisation problem subject to constraints. The optimisation problem is that of minimising the error energy function defined by the sum of squares of the residuals between the observed seismic trace and its prediction by the forward model for the given earth model parameters. To determine the solution we use the method of generalised linear inverses. The generalised inverse is possible only when the Hessian matrix, which describe the curvature of error energy surface, is positive definite. When the Hessian is not definite, it is necessary to modify it to obtain the nearest positive definite matrix. To modify the Hessian we used a method based on the Cholesky factorisation. Because the modified Hessian is positive definite, we need to find the generalised inverse only once. But we may need to restrict the step-length to obtain the minimum. Such a method is a step-length based method.

A step-length based method was implemented using linear equality and inequality constraints into a computer program to invert the observed seismic data for impedance. The linear equality and inequality constraints were used so that solutions that are geologically feasible and numerically stable are obtained.

The strategy for the real data inversion was to first estimate the seismic wavelet at the well, then optimise the wavelet parameters. Then use the optimum wavelet to invert for impedance and layer boundaries in the seismic traces.

In the three real data examples studied, this inversion scheme proved that the delineation of the Chadra sands in Gialo Field is possible. Better results could be obtained by using initial earth models that properly parameterise the subsurface, and linear constraints that are based on well data. Defining the wavelet parameters in the time domain may prove to be more stable and could lead to better inversion results.

CONTENTS

List of Figures

Figure 5.2a. The error energy surface showing the region of convergence for the global minimum (28, 54).

Figure 5.2b. The time domain wavelet used to generate the error energy surface of Figure 5.2a. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.3a The impedance profiles for the example of section 5.5.1.1. Note: for all the impedance profile figures in this chapter, the initial guess profile is a dashed line, the observed profile is a dotted line and the solution profile is a solid line. Note that t_1 and t_4 are misplaced by 8 ms. In this example convergence to the correct solution was obtained.

Figure 5.3b. The solution traces for the example of section 5.5.1.1. The first and last traces, i.e., traces numbered 1 and 8, are the observed seismic traces. Trace number 2 is the initial guess seismic response. Traces 3-7 represent the seismic response of the five iterations of inversion performed to obtain the final solution of trace 7. Observe that traces 7 and 8 are exactly the same. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.3c. The error traces for the example of section 5.5.1.1. These traces correspond to the solution traces of figure 5.3b. The first error trace corresponds to the solution trace numbered 2 and the last error trace corresponds to the solution trace numbered 7. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.4a. The impedance profiles for the example of section 5.5.1.2 showing the polarity reversal at boundary location 4. Only the first and last boundaries converged to their correct locations.

Figure 5.4b. The error traces for the example of section 5.5.1.2. They are the error traces for the two iterations performed by the inversion program, trace number 1 belongs to the first iteration. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.5a The solution traces for the boundary locations inversion problem when the observed trace (numbered 1 and 8) has a signal-to-noise ratio of 4. The solution traces for the five iterations are traces 2 to 7. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.5b The error traces corresponding to the solution traces of Figure 5.5a. Because inversion to the correct solution was obtained, the last error trace, number 6, represents the total random noise in the observed seismic trace. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.5c. The impedance profiles for the noise added boundary locations inversion example of section 5.5.1.3 when the signal-to-noise ratio of the observed

trace is 4. For such a signal-to-noise ratio convergence to the correct boundary locations was obtained.

Figure 5.6a The observed trace (numbered 1 and 8) has a signal-to-noise ratio of 2. The solution traces for the six iterations the inversion program performed are traces 2 to 7. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.6b. The impedance profiles for the boundary locations inversion data listed in Table 5.4. The signal-to-noise ratio of the observed trace in this example is 4. Note that, except for the first boundary, no boundary has convergence to its correct location.

Figure 5.7a. The observed trace (numbered 1 and 13) has a signal-to-noise ratio of 1. The solution traces for the 11 iterations the inversion program performed are traces 2 to 12. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.7b. The impedance profiles for the boundary locations inversion data listed in Table 5.5. The signal-to-noise ratio of the observed trace in this example is 1. Note that, except for the third boundary, no boundary has convergence to its correct location.

Figure 5.8a. The observed trace (numbered 1 and 12) has a signal-to-noise ratio of ½. The solution traces for the 10 iterations the inversion program performed are traces 2 to 11. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.8b. The impedance profiles for the boundary locations inversion data listed in Table 5.5. The signal-to-noise ratio of the observed trace in this example is ½. Although the first boundary, which has the highest reflection coefficient, and the second boundary converged to their correct values, the other boundary locations stayed well away from their correct locations.

Figure 5.9. The impedance profiles of the example of section 5.5.2.1 showing a non-unique convergence of the impedance solution.

Figure 5.10. The impedance profiles after equality constraining the first layer for the example of section 5.5.2.1.

Figure 5.11. The impedance profiles for the example shown in Figure 5.10 where the initial guess impedance is constant and equal to the first layer impedance.

Figure 5.12. The impedance profiles when three reversals were introduced in the initial guess impedance.

Figure 5.13. The impedance profiles when the initial guess has two reversals at boundaries 2 and 3, and a non-zero observed impedance gradient vector.

Figure 5.14. The impedance profiles for the data in Table 5.10. The signal-to-noise ratio of the observed trace is 4. In this inversion for impedance the solution

is closer is closer to the correct impedance values for the shallow layers where there is a high impedance contrast across their boundaries

Figure 5.15. The impedance profiles for the data in Table 5.11. The signal-to-noise ratio of the observed trace is 2. Comparing the impedance solution with that of Figure 5.14, where the signal-to-noise ratio is 4, we notice that the impedance trend of the solution profile has deteriorated for the deeper layers. But no polarity reversals resulted across any of the interfaces.

Figure 5.16. The impedance profiles for the data in Table 5.12. The signal-to-noise ratio of the observed trace is 1. In this inversion for impedance the solution profile differs greatly from the observed profile at layers 5 and 6 at which we have a small impedance contrast. At the boundary between layers 5 and 6 there is a polarity reversal in the solution profile.

Figure 5.17a. Impedance profiles for impedance inversion with slightly incorrect boundaries. Note that the impedance solution is in error only across the boundary that is incorrectly located.

Figure 5.17b. The solution traces after each iteration in solving the example of Figure 5.17a. Traces numbered 1 and 7 are the observed trace, while trace 2 is the seismic response of the initial guess and traces 3-6 are the seismic response after each iteration. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.17c. The error traces corresponding to the iterations of Figure 5.17b. Trace 1 is the error trace of the initial guess. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.18a. Solution traces when the boundaries are grossly incorrect. Traces numbered 1 and 6 are the observed trace, while trace 2 is the seismic response of the initial guess and traces 3-5 are the seismic response after each iteration. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.18b. The impedance profiles when the boundaries are grossly incorrect. The solution profile has a polarity reversal at the incorrect boundary.

Figure 5.19. The impedance profiles of the first impedance inversion run of Table 5.19. No polarity reversals were introduced, but the impedance trend of the solution profile have largely changed as compared to the observed impedance profile.

Figure 5.20. The impedance profiles of the first boundaries inversion run of Table 5.19. Note that in this run only boundaries 2 and 3 did not converge to their correct locations

Figure 5.21. The impedance profiles of the final impedance inversion run of Table 5.19. Note that only the impedance of layers 3 and 4 differ greatly from their correct impedance values, the other layers have their solution impedance close to their correct values.

Figure 5.22. The impedance profiles of the final impedance inversion run of Table 5.21. Even though the signal-to-noise for this test is much lower than the results of Figure 5.21, the final impedance solution obtained here is better.

Figure 5.23. The impedance profiles for the first impedance run (run number 1 in Table 5.25) when inverting for impedance, boundaries and wavelet parameters with a signal-to-noise ratio of 4 in the observed trace.

Figure 5.24. The impedance profiles for the first, and only, boundaries inversion run (run number 2 in Table 5.25) when inverting for impedance, boundaries and wavelet parameters with a signal-to-noise ratio of 4 in the observed trace. This is the only boundaries inversion needed because all the boundaries converged to their correct locations in this run.

Figure 5.25. Wavelet number 1 is the initial guess wavelet, and wavelets numbered 2-5 are the solution wavelets for the four iterations of inversion run number 4 in Table 5.25. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.26. The final impedance profiles corresponding to the solutions given in Table 5.24. A reasonable estimate of the observed impedance profile is obtained.

Figure 5.27. The final wavelet parameters inversion results in the three iterations. The final solution wavelet is w_{sol}= [4.2 44.3 54.8 79.1 115000. 115000. 0.3826 0.113 0]^T which is wavelet number 4.

Figure 5.28. The final impedance profiles corresponding to the solutions given in Table 5.26. The signal-to-noise ratio is 2. Boundaries 2, 4 and 5 did not converge to their correct locations, but the general impedance trend in the solution profile resembles that in the observed profile.

Figure 5.29. The final impedance profiles for the case when the signal-to-noise ratio in the observed trace is 1, and when inverting for impedance, boundaries and wavelet parameters. Notice that even for this low signal-to-noise ratio, all the boundaries converged to their correct locations.

Figure 6.1. The wavelet estimated as a Wiener shaping filter at well YY31. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.2. The amplitude spectrum of the Wiener wavelet estimated at well YY31 and shown in Figure 6.1.

Figure 6.3. The phase spectrum of the Wiener wavelet estimated at well YY31 and is shown in Figure 6.1.

Figure 6.4. Parameterising the amplitude spectrum of the Wiener wavelet estimated at well YY31.

Figure 6.20. The error traces corresponding to the solutions of Figure 6.19. The first error trace is that for initial guess and the last error trace is that for the seismic solution. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.21. The resulting wavelets in the first wavelet parameter inversion. The wavelet number 1 is the YY31 parameterised wavelet, and wavelet number 49 is the final wavelet. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.22. The solution traces that correspond to the wavelets of Figure 6.21. The first and last traces are the observed seismic trace, trace number 2 is the initial guess which corresponds to wavelet number 1 in Figure 6.21. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.23. The error trace that correspond to the solutions of Figure 6.22. Error traces number 1 corresponds to the initial guess wavelet of Figure 6.21, and error trace number 49 corresponds to the final wavelet. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.24. The optimum wavelet for well YY31 is wavelet number 4. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.25. The optimum impedance profile (solid line) for well YY31.

Figure 6.26. The optimum seismic solution for YY31 is trace number 5. Traces 1 and 6 are the observed seismic trace. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.27. The error traces corresponding to the seismic solutions of Figure 6.26. Error trace number 4 corresponds to the optimum seismic solution trace, which is number 5 in Figure 6.26. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.28. The acoustic impedance log of well YY04.

Figure 6.29. The parameterisation of the acoustic impedance log of YY04 into eight layers. Chadra A is the 10 ms thick sand layer between 670 ms and 680 ms two-way travel-time. The top of Augila Limestone is at 754 ms and extends to the bottom of the log.

Figure 6.30. The solution traces for the first inversion for impedance when optimising the wavelet at YY04. The first and last traces are the observed seismic trace. Trace number 2 is the initial guess seismic response, and trace number 190 is the final seismic solution. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.31. The impedance profiles for the inversion of Figure 6.30. Zero on the time axis corresponds to two-way travel-time of 610 ms.

Figure 6.32. The seismic solutions of the first boundaries inversion when optimising YY04 wavelet. The first and last traces are the observed seismic trace,

trace number 2 is the initial guess seismic response, and trace number 4 is the final seismic solution. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.33. The impedance profiles for the solution of Figure 6.32, i.e. for the first boundary location inversion when optimising the wavelet at well YY04. Zero on the time axis corresponds to 610 ms of two-way travel-time.

Figure 6.34. The resulting wavelets in the first inversion for the wavelet parameters in YY04. Wavelet number 1 is the initial parameterised wavelet. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.35. The optimum wavelet in YY04 is wavelet number 5 Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.36. The optimum impedance profile in well YY04. Zero on the time axis corresponds to 610 ms of two-way travel-time.

Figure 6.37. The observed seismic traces (CDP's 748-787) from Line 1973. The two sand bodies concerned are the two positive reflections (peaks) at about 710 ms and 730 ms on CDP 748. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.38. The initial earth model section for the observed seismic traces from Line 1973 (Figure 6.37). Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.39. The initial earth model section of Figure 6.38 displayed without the observed seismic traces. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.40. The initial synthetic seismograms for the earth model section in Figure 6.39. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.41. The initial error traces corresponding to the synthetic seismograms in Figure 6.40. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.42. The final impedance solution for the initial earth model of Figure 6.39. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.43. The final synthetic seismograms for the impedance solution of Figure 6.42. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.44. The final error traces corresponding to the synthetic seismograms of Figure 6.43. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.45. The observed seismic data of CDPs 1612-1661 in Line 1977. The seismic event (peak) at about 730 ms is interpreted as positive reflection from a Chadra sand body, and the broad positive reflection at 750-760 ms is a two step limestone bed. The two events are separated by a low impedance layer that has a

contact with the Chadra sand at 740 ms. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.46a. The initial earth model section for CDPs 1612-1636 and the corresponding observed. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.46b. The initial earth model section for CDPs 1637-1661 and the corresponding observed traces. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.47a. The initial earth model section for CDPs 1612-1636. These impedance profiles are also displayed in Figure 6.46a. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.47b. The initial earth model section for CDPs 1637-1661. These impedance profiles are also displayed in Figure 6.46b. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.48. The initial synthetic seismograms for the earth model section of Figures 6.47a and 6.47b. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.49. The error section corresponding to the initial earth model synthetic seismograms in Figure 6.48. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.50a. The final impedance section for CDPs 1612-1636. The sand layer that starts at about 730 ms appears to be a continuous sand body. The limestone layer, however, becomes more sandy to the left of the section. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.50b. The final impedance section for CDPs 1637-1661. The sand layer that starts at about 730 ms appears to be a continuous sand body. The limestone layer, however, becomes more sandy to the left of the section. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.51. The final seismic solution section for CDPs 1612-1661. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.52. The error traces section corresponding to the seismic solution in Figure 6.51. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.53a. The initial earth model section for example 2 in Line 1977 for CDPs 1536-1560. The upper sand at 705 ms is overlain by shale, and the two limestone layers below are underlain by shale. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.53b. The initial earth model section for example 2 in Line 1977, CDPs 1561-1585. The upper sand at 705 ms is overlain by shale, and the two

limestone layers below are underlain by shale. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.54a. The same initial earth model section in Figure 6.53a superimposed on the corresponding observed seismic traces of Figure 6.55. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.54b. The same initial earth model section in Figure 6.53b superimposed on the corresponding observed seismic traces of Figure 6.55. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.55. The observed seismic traces CDP 1557-1581 for example 2 from Line 1077. The positive seismic event at 705 ms is a sand body. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.56. The synthetic seismograms of the initial earth model traces in Figures 6.53a and 6.53b. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.57. The error traces of the synthetic seismograms of Figure 6.56. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.58a. The final impedance solution for example 2 on Line 1977, CDPs 1536-1560. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.58b. The final impedance solution for example 2 on Line 1977, CDPs 1561-1585. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.59. The final seismic solution section corresponding to the impedance solution in Figures 6.58a and 6.58b. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.60. The error section of Figure 6.59. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.61. The results of the eight iterations of the first wavelet parameters inversion at well YY31. Wavelet number 1 is the initial guess wavelet given by the parameters $\mathbf{w}_{ini} = [6 \ 38 \ 50 \ 80 \ 109000 \ 109000 \ -0.2 \ 0.115 \ 0]^T$, and wavelet number 9 is the final optimised wavelet for this inversion run. This wavelet has the parameters $\mathbf{w}_{sol} = [4 \ 24 \ 70 \ 74 \ 109000 \ 109000 \ -0.209 \ 0.115 \ 0]^T$. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.62. The impedance profiles for the only boundary locations inversion needed when optimising the wavelet at well YY31. The initial guess impedance profile is the dashed line with cross marks, and the solution impedance profile is the solid line. Note that all boundary locations are adjusted by 2 ms (one sample interval), except for the 10-th boundary location which was adjusted by 4 ms.

Figure 6.63. The results of the nine iterations of the final wavelet parameters inversion at well YY31. The optimum wavelet obtained is wavelet number 10 which has the parameters $w_{sol} = [3.9 \quad 24 \quad 70 \quad 74 \quad 109022 \quad 108993 \quad -0.206 \quad 0.115 \quad 0]^T$. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.64. The results of the 38 iterations of the first wavelet parameters inversion at well YY04. Wavelet number 1 is the initial guess wavelet given by the parameters $w_{ini} = [17 \quad 32 \quad 42 \quad 67 \quad 113330 \quad 113330 \quad 0.1 \quad 0.12 \quad 0.]^T$. Wavelet number 39 is the final optimised wavelet for this wavelet parameters inversion run, this wavelet has the parameters $w_{sol} = [21 \quad 36 \quad 60 \quad 66 \quad 113330 \quad 113330 \quad 0.7962 \quad 0.115 \quad 0]^T$. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.65. The impedance profiles for the only boundary locations inversion needed when optimising the wavelet at well YY04. The initial guess impedance profile is the dashed line with cross marks, and the solution impedance profile is the solid line. It can be observed that only boundary locations 3 and 4 are adjusted by 2 ms (one sample interval) each. This represents the minimum shift a single boundary can be adjusted.

Figure 6.66. The results of the nine iterations of the final wavelet parameters inversion at well YY04. The optimum wavelet obtained is wavelet number 44 which has the parameters $w_{sol} = [16.4 \quad 45.1 \quad 67.2 \quad 81.8 \quad 113330 \quad 113330 \quad 0.3128 \quad 0.118 \quad 0]^T$. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.67. The initial guess impedance section, or earth model section, that constitutes the input to the inversion of the observed section of Figure 6.68 around well YY31. The well is located at CDP 555, and measures the time window 580-804 ms two-way travel-time. The earth model data strictly follow the impedance profile of well YY31 given in Figure 6.14. The seismic time window was further extended, into the Augila Limestone, to 900 ms two-way travel-time. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.68. The observed seismic section used to invert for impedance around well YY31. This section is part of seismic Line 1973 between CDP 535 and CDP 575. The start of the well impedance measurements tie the seismic data at 580 ms and extends to 804 ms two-way travel-time. The 96 ms of section time between 804 ms and 900 ms is an extra section extended into the Augila Limestone. The initial guess earth model section for this observed seismic data, shown in Figure 6.67, has 18 layers. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.69. The initial guess seismic response section around well YY31. The well is located at CDP 555. This section is generated from the initial guess earth model of Figure 6.67 and the seismic wavelet optimised in section 6.9.1. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.70. The final impedance solution section around well YY31. The Augila Limestone starts at about 800 ms and continues to 900 ms two-way travel-time. The impedance contrast across the boundary represented by the top of Augila varies from one profile to the next. There is a large impedance contrast for the middle profiles that surround the well at CDP 555. The impedance contrast decreases to the left and right. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.71. The seismic solution section of the final impedance solution of Figure 6.70. The seismic events in this seismic section should be compared to the seismic events in the observed section in Figure 6.68. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.72. The observed seismic section used to invert for impedance around well YY04. This section is part of Line 1977 extending from CDP 1073 to CDP 1113. The well is located at the middle trace which is CDP 1093. The well impedance measurements start at 610 ms and extends for 234 ms to 844 ms two-way travel-time. The 56 ms of section time between 844 and 900 ms is an extra section extended into the Augila Limestone. The initial guess earth model section for this observed seismic section has 10 layers and is given in Figure 6.73. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.73. The initial guess impedance section, or earth model section, that constitutes the input to the inversion of the observed seismic section of Figure 6.72 around well YY04. The well is located at CDP 1093, and measures the time window 610-844 ms two-way travel-time. The earth model profiles are obtained from the impedance profile of well YY04 given in Figure 6.29. The seismic time window is further extended into the Augila Limestone to 900 ms. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.74. The initial guess seismic response section around well YY04. The well is located at CDP 1093. This section is generated from the initial guess earth model of Figure 6.73 and the seismic wavelet optimised in section 6.9.2. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.75. The final impedance solution section for inversion around YY04. The shallow layer is the Chadra A sand. It is thin at the middle traces and thicker to the left and right. The Augila Limestone starts at about 750 ms two-way travel-time. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.76. The seismic solution section corresponding to the impedance solution in Figure 6.75. The high amplitude associated with the middle traces for the shallow event is due to the thickness of the Chadra A sand being at, or near, the tuning thickness. The decrease in amplitude of the shallow event to the left and right of the middle traces is an indication of thickness increase of the sand body. The lower part

of the section, starting at about 750 ms, describes the Augila Limestone and closely resembles the corresponding section on the observed seismic section in Figure 6.72. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.77. The 11-CDP observed seismic section which is part of Line 1073 used to invert for impedance around the intersection with Line 1977. The middle trace, CDP 1042, is the trace located at the intersection. Correlation with Line 1977 produced only 7 seismic events that have good signal-to-noise ratio. The 7 interfaces are shown in the 8-layer initial guess impedance section of Figure 6.78. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.78. The initial guess impedance section on Line 1973 containing 11 impedance profiles around the intersection with Line 1977. The impedance profile located at the intersection is profile number 1042. this 8-layer impedance section was generated from the correlation of 7 seismic events on the two lines that have a good signal-to noise ratio, and the two impedance profiles of wells YY04 and YY31. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.79. The initial guess seismic response section on Line 1973 around the intersection with Line 1977. The shallow part of the section, above 700 ms two-way travel-time, is the response of the three interfaces in the Chadra sands, and the lower part, below 750 ms, is. the response of four interfaces in the Augila Limestone. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.80. The final impedance solution section corresponding to the observed seismic section on Line 1973 around the intersection with Line 1977. The impedance profile at the intersection is profile number 1042. This impedance section should be compared with the impedance solution section along Line 1977 given in Figure 6.85. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.81. The final seismic solution section of the impedance solution on Line 1973 shown in Figure 6.80. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.82. The 11-CDP observed seismic section which is part of Line 1077 used to invert for impedance around the intersection with Line 1973. The middle trace, CDP 1707, is the trace located at the intersection. Correlation with Line 1973 produced only 7 seismic events that have good signal-to-noise ratio. The 7 interfaces are shown in the 8-layer initial guess impedance section of Figure 6.83. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.83. The initial guess impedance section on Line 1977 containing 11 impedance profiles around the intersection with Line 1973. The impedance profile located at the intersection is profile number 1707. this 8-layer impedance section was generated from the correlation of 7 seismic events on the two lines that have a good

signal-to noise ratio, and the two impedance profiles of wells YY04 and YY31. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.84. The initial guess seismic response section on Line 1977 around the intersection with Line 1973. The shallow part of the section, above 700 ms two-way travel-time, is the response of the three interfaces in the Chadra sands, and the lower part, below 750 ms, is the response of four interfaces in the Augila Limestone. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.85. The final impedance solution section corresponding to the observed seismic section on Line 1977 around the intersection with Line 1973. The impedance profile at the intersection is profile number 1707. This impedance section should be compared with the impedance solution section along Line 1973 given in Figure 6.80. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.86. The final seismic solution section of the impedance solution on Line 1977 shown in Figure 6.85. Note that the vertical axis represents the two-way travel-time in ms.

List of Tables

Table 5.14. Showing that the boundary location 5 is perturbed by 10 ms, i.e. 5 samples, in the initial guess.

Table 5.15. Input and output data for impedance and boundaries inversion. The correct boundaries and impedance solutions were achieved in four impedance inversion runs and three boundary inversion runs.

Table 5.16. Impedance solution progression when inverting for boundaries and impedance. The maximum impedance values shown are those for layer 3. Imp means impedance inversion run, and Bnd is boundary inversion run.

Table 5.17. Impedance solution progression for the same problem in Table 5.16 after imposing a global upper bound of 12000 $gcm^{-3}ms^{-1}$ on all the layers.

Table 5.18. Input and output data for impedance and boundaries inversion when the signal-to-noise ratio is 10. The final boundaries and impedance solutions were achieved in four impedance inversion runs and three boundary inversion runs. The progression toward the final solution is summarised in Table 5.19.

Table 5.19. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 10. The solution results are given in Table 5.18. Imp means impedance inversion run, and Bnd is boundary inversion run.

Table 5.20. Input and output data for impedance and boundaries inversion when the observed trace has a signal-to-noise ratio of 2. The final boundaries and impedance solutions were achieved in two impedance inversion runs and one boundary inversion run. The progression toward the final solution is summarised in Table 5.21.

Table 5.21. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 2. The solution results are given in Table 5.20. Imp means impedance inversion run, and Bnd is boundary inversion run.

Table 5.22. Example of inversion for impedance, boundaries and wavelet parameters.

Table 5.23. The second example of inversion for impedance, boundaries and wavelet parameters.

Table 5.24. The input and output data for the example of inversion for impedance, boundaries and wavelet parameters when the signal-to-noise ratio in the observed trace is 4. The summary of the progression towards the solutions is given in Table 5.25.

Table 5.25. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 4. The solution results are given in Table 5.24. Imp means impedance inversion

run, Bnd is boundary inversion run, and Wvl means a wavelet parameters inversion run.

Table 5.26. The input and output data for the example of inversion for impedance, boundaries and wavelet parameters when the signal-to-noise ratio in the observed trace is 2. A summary of the progression toward the solutions is given in Table 5.27.

Table 5.27. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 2. The solution results are given in Table 5.26. Imp means impedance inversion run, Bnd is boundary inversion run, and Wvl means a wavelet parameters inversion run.

1

INTRODUCTION

To deduce information about the subsurface rock properties, we use seismic reflection data recorded at the surface of the earth. The most straightforward subsurface property which may be estimated from seismic reflection data is probably the acoustic impedance. Estimating subsurface parameters from the surface recorded seismic data is a seismic inverse problem. This thesis is concerned with inverting seismic reflection data to deduce the acoustic impedance distribution in the subsurface.

To solve the inverse problem, it is necessary to formulate the forward problem. The forward problem, or the forward model, is a mathematical relationship that predicts the observed data for a given set of model parameters. In the inverse problem we start with the observed data and a mathematical modelling procedure, and we estimate the model parameters.

The choice of the forward modelling procedure is crucial to solving the inverse problem at hand. In this research the objective is to determine the lateral distribution of the Oligocene Chadra sands of the Arida Formation in the Gialo Field, Sirte Basin. To achieve this, I have attempted to estimate the acoustic impedance profiles of observed seismic traces. The Chadra sands were deposited as sheet-like bodies, or bars, over a gently dipping surface. The bars are elongate, widespread sand bodies that interfinger and lens, and they extend for 4-6 km in width and 12-16 km in length, with thicknesses of up to 30 m. Such simple layer-cake geology suggests that the common depth point (CDP) stacking procedure is appropriate for this area, and the

application of normal-moveout correction followed by stacking approximates the response of normal incidence plane waves in a layered medium. Thus an appropriate trace model is assumed to be the convolutional model (Robinson, 1983)

$$s(t) = w(t) * r(t) \qquad (1.1)$$

where s(t) is the synthetic seismic trace, w(t) is the seismic wavelet, r(t) is the reflection coefficient function, t is a time variable, and * denotes convolution.

The advantage of using the convolutional model is that the continuous representation (1.1) is readily adaptable for the discrete case. The discrete representation of (1.1) is given by

$$s_i = \sum_{j=0}^{n} r_j w_{i-j} \qquad i=0, 1, 2, \ldots, m+n \qquad (1.2)$$

where s, r, w are digital wavelets of length $m+n+1$, $n+1$, and $m+1$, respectively.

The estimation of reflection coefficients in (1.2) is obtained by using the Goupillaud earth model which consists of a stratified system where all layers have equal two-way travel-time (Goupillaud, 1961). Thus for an earth model with $n+1$ interfaces the reflection coefficient for particle displacement (or velocity) at the k-th interface is

$$r_k = \frac{A_k - A_{k+1}}{A_k + A_{k+1}}, \qquad k=0, 1, 2, \ldots, n \qquad (1.3)$$

where A_k is the acoustic impedance of the layer above the k-th interface. Equation (1.3) shows how the reflection coefficients are related to the acoustic impedances for the two-way travel-time to each interface.

When solving the inverse problem, we make estimates of the model parameters. Thus it is important to know how many model parameters should be used and which parameters are significant. These model parameters define a geologic model whose seismic response agrees with the observed seismic data. The rock property that can be readily estimated from observed seismic data is the acoustic impedance, so that our aim becomes that of converting the seismic reflection data into acoustic impedance profiles as a function of two-way travel-time. Thus the geologic model parameters are those defining an acoustic impedance profile.

The acoustic impedance of the earth is a continuous function of time and to parameterise it we use a restricted number of layers. To each layer we assign three

parameters: (1) a parameter defining the two-way travel-time to the bottom of the layer; (2) an acoustic impedance parameter defining the starting value of the acoustic impedance in the layer; and (3) a parameter defining the linear gradient of the impedance within the layer. An example of acoustic impedance parameterisation is shown in Figure 1.1.

Equation (1.2) shows that knowledge of the seismic wavelet is essential to calculate the forward model response (synthetic seismogram). The seismic wavelet can be estimated as a Wiener shaping filter at a borehole located on the observed seismic data. This shaping filter converts the reflection coefficient sequence, obtained from the sonic and density logs at the borehole, into the seismic trace recorded at the well location. The seismic wavelet is parameterised in the frequency domain using nine parameters. They include four bounding frequency parameters defining a band-limited amplitude spectrum, two amplitude parameters defining the amplitudes of the two middle frequency parameters, and three phase parameters defining the phase spectrum in the following form

$$\phi(f) = \phi_0 + \phi_1 f + \phi_2 f^2 . \tag{1.4}$$

In this expression ϕ_0 is a constant phase parameter, and has the most effect on the wavelet (White; 1987). The term ϕ_1 only produces a time shift so it has no effect on the shape of the wavelet, and the quadratic term ϕ_2 describes the frequency dispersion in the wavelet. An example of amplitude spectrum parameterisation is shown in Figure 1.2.

Choosing the appropriate forward model and model parameters is only part of the inversion process. Indeed, the diversity of the inverse problem is not only due to the numerous forward models that can be adopted for a specific problem but also the variety of methods by which it can be solved.

A well studied impedance inversion method is recursive inversion. This method was described by Lavergne and Wills (1977), and Lindseth (1979). The recursive inversion is based on equation (1.3), where it can be rewritten to express A_{k+1} in terms of A_k and r_k:

$$A_{k+1} = A_k \frac{1 + r_k}{1 - r_k} . \tag{1.5}$$

That is, the impedance of a layer can be deduced from the reflection coefficients and the impedance of the layer above it. Although Lindseth (1979) demonstrated that the inherent band-limitedness of the seismic data could be largely overcome, mainly by

extracting the low frequency components from other types of data such as sonic logs, the recursive method still performed poorly in the presence of noise.

Describing the convolutional model for the seismic trace in terms of acoustic impedance parameters for layers and wavelet parameters makes it possible to solve the inversion problem using optimisation methods. Parker (1994) regarded the geophysical inverse problem as an optimisation problem subject to various constraints.
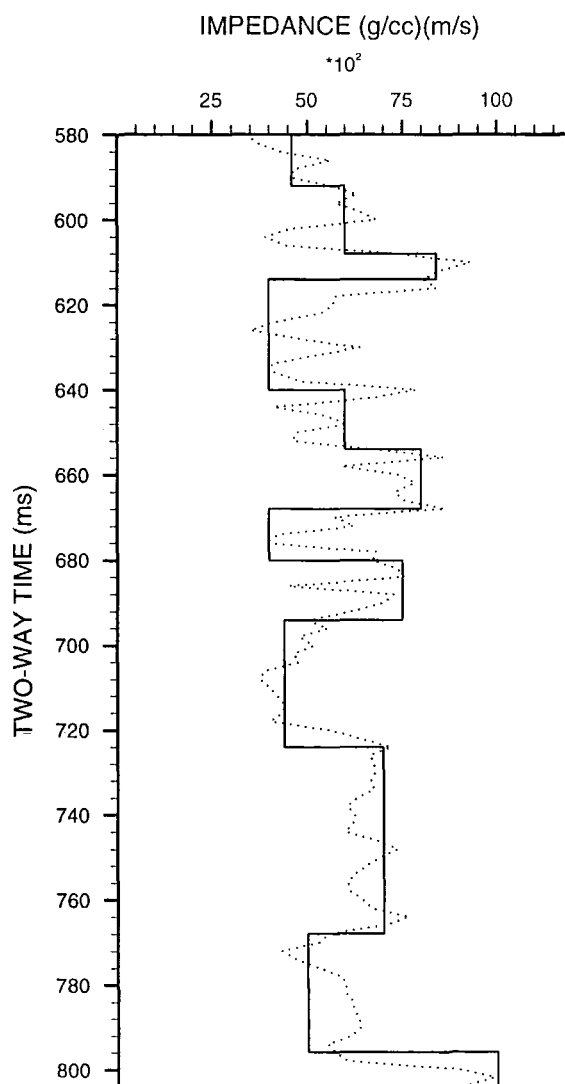
Figure 1.1 Parameterising a continuous acoustic impedance log (dotted line) into restricted number of layers (solid line).
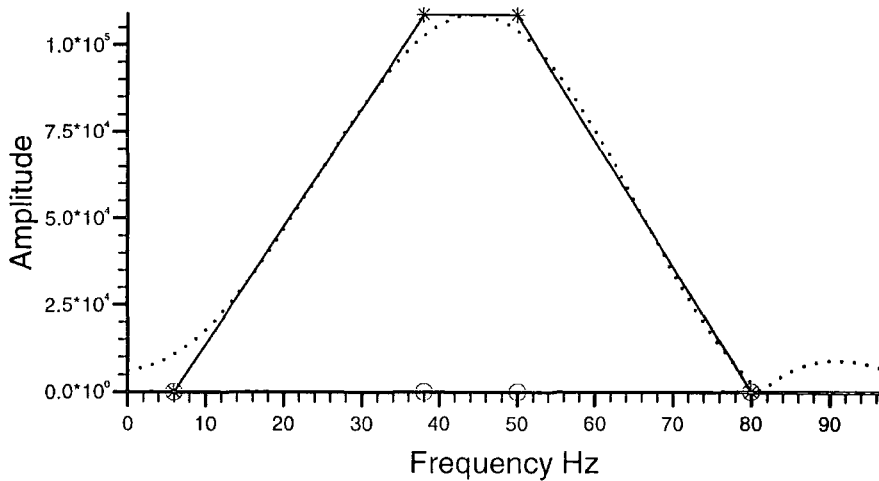
Figure 1.2 Parameterising an amplitude spectrum of a wavelet by four bounding frequency parameters and two amplitude parameters that describe the amplitude of the two middle frequencies. The four frequency parameters are marked by circles on the frequency axis and the corresponding amplitudes are marked by astrisks.

To use optimisation methods to solve the inverse problem, we use generalised linear inverses, which are linearised least-squares procedures derived by replacing the non-linear relationship between the observed data and the unknown parameters by a linear approximation. Unique and stable generalised inverses are possible only when the inverse problem is well-posed and noise-free. The important papers of Backus and Gilbert (1967, 1968, 1970) analysed the problems of uniqueness, resolution and stability of the inverse problem. In the Backus-Gilbert approach the unknown model parameters are obtained from linear combinations of the observed data. That is, observations are combined using an averaging kernel designed to optimise the trade-off between the resolution and the accuracy of the model. The Backus-Gilbert method was extensively applied to a variety of geophysical problems. Oldenburg (1981) and Treitel and Lines (1982) showed how the Backus-Gilbert theory is related to seismic wavelet deconvolution.

An elegant method for solving generalised linear inverse problems is based on the singular value decomposition, or SVD. Such an inverse is known as Lanczos inverse (e.g. Golub and Van Loan 1983). Jackson (1972) discussed using SVD to obtain stable solutions from ill-conditioned systems. Wiggins *et al.* (1976) analysed the residual statics problem and used SVD to obtain solutions for the linear systems of equations that are inherently non-unique for the very long spatial wavelength of the residual statics. van Riel and Berkhout (1985) used SVD to determine the resolving power of the linear inverse problem. Bilgeri and Carlini (1981) used SVD to solve the linear inverse problem for the reflection coefficients and also obtained wavelet

5

estimates, in the frequency domain. They then used the estimated reflection coefficients to compute acoustic impedance profiles using equation (1.5).

A popular generalised linear inversion method was first discussed by Levenberg (1944) and later refined by Marquardt (1963). To obtain a stable solution, the Marquardt-Levenberg method restricts the deviations of the estimated parameters from a reference parameter by using a damping factor. Draper and Smith (1981) and Fletcher (1987) describe the method in great detail. Lines and Treitel (1984) presented an excellent review of least-squares inversion and illustrated the use of Marquardt-Levenberg method to obtain stable generalised inverses. They also explained the damping factor in terms of the SVD method.

The Marquardt-Levenberg method has been widely used in geophysical inversion problems. Jupp and Vozoff (1975) used a modified Marquardt-Levenberg method to invert resistivity data. Sain and Kaila (1994) used the Marquardt-Levenberg method, where they called it damped least squares, in the inversion of wide-angle seismic reflection times to calculate interval velocities. More importantly, the Marquardt-Levenberg method was applied to calculate estimates of acoustic impedance profiles from stacked seismic data. Cooke and Schneider (1983) used the Marquardt-Levenberg method to invert for acoustic impedance from mainly synthetic data. Keys (1986) used the Marquardt-Levenberg method to show a relationship between generalised linear inversion methods and another class of linear inversion methods that are based on the Born approximation. Tian Gang and Goulty (1996) used the method to invert for the coal seam thickness, with special emphasis on thin layer inversion.

The Marquardt-Levenberg method is in a class of methods termed trust-region, because the restricted step length to the estimated parameters from the reference parameters is always taken to be unity (Gill et al., 1981). In practice it usually becomes necessary, when the problem is ill-posed and contains noise, to compute several trial steps before finding a satisfactory step. This means that algorithms based on trust-region methods will tend to be slow, especially when the problem involves a large number of parameters. A different class of methods are step-length-based methods (Gill et al., 1981). In a step-length-based method a step-length procedure must be included because a step length of unity does not necessarily satisfy the conditions of a solution. In this way only one step is computed, then scaled if necessary to obtain a satisfactory solution. This research uses an algorithm based on a step-length method that uses Cholesky factorisation to obtain a stable generalised linear inverse. Furthermore, this method lends itself to using linear equality and inequality constraints which are used here to steer the solution into a region that

satisfies geologic information known *a priori*, mainly from well log data, thus reducing the problem of non-uniqueness.

This thesis is organised as follows: chapter 2 gives a statement of the geologic problem and discusses the objectives of this research. It also describes the seismic and well log data available from two wells, how the analogue paper displays of the well data were digitised and used to compute a reflection coefficient sequence in each well, and the wavelet estimation. In chapter 3 the generalised linear inversion is discussed in terms of minimisation of the error energy function using the Gauss-Newton method. The conditions that must be satisfied for a solution to be a minimum are also discussed, and a Cholesky factorisation based method to obtain a stable solution, and a step-length procedure known as line search by back-tracking are described. Chapter 4 describes the theory of linear equality and inequality constraints, then restates the minimisation problem of chapter 3 subject to linear equality and inequality constraints. A computer program based on the active set strategy, which implements the methods of chapters 3 and 4, is discussed in chapter 5. Included in chapter 5 are tests on synthetic seismic data to assess the program performance and establish the validity of its results so that a strategy for the inversion of stacked seismic data could be developed. Inversion of the field data is discussed in chapter 6. Three examples taken from two seismic lines recorded in the Gialo area, Sirte Basin, are discussed. In each example, inversion for the Chadra sands was performed in an attempt to delineate these sand bodies by inversion for acoustic impedance. The conclusions of this research are summarised, along with recommendations for future work, in chapter 7.

2

GEOLOGIC PROBLEM, OBJECTIVES AND DATA

2.1. Introduction

To extract stratigraphic information about the subsurface geology from seismic reflection data that were recorded at the surface, we use the seismic inversion method. The inversion method used here inverts the seismic data into acoustic impedance, which is related to lithology. Section 2.2 starts by giving a brief description of the regional geology of the Sirte Basin. The main aim of that section, though, is to discuss the geologic problem to be solved by inversion, which concerns the potential reservoir rocks, the Oligocene Chadra sands A, B and C. These sands will be discussed in terms of their geologic aspects that are related to their inversion into acoustic impedance. The important parameters of the Chadra sands are their geometry and lithology; thus we describe their areal extent, orientation and thickness trends, and also describe their lithology type and contact relationships. We especially concentrate on the YY reservoir area where the seismic data available for this research were recorded. Indeed the objective of this research is to attempt to delineate the Chadra sands in the YY reservoir along the two seismic lines 1973 and 1977 using seismic inversion.

The seismic and well log data available for this research are described in section 2.3. Acquisition parameters for the seismic data were chosen to maximise the useful band-width, then later processed with relative amplitudes preserved and zero phase wavelet for best inversion results. The well data include mainly well log

measurements of sonic and density, but SP, caliper, gamma ray and resistivity measurements are also available. All the well log data are available in analogue paper display form, so they had to be manually edited first, then digitised to be used in a computer program. The steps of a computer program that produces an acoustic impedance log from the digitised sonic and density data are explained in section 2.4. Another output of this program is the reflection coefficient sequence at the well location. Section 2.5 discusses using the computed reflection coefficient sequence and the seismic trace at the well location to find an estimate of the effective seismic wavelet as a Wiener shaping filter.

Estimating seismic wavelets is always subject to errors. There are different possible sources of errors and section 2.6 discusses these error sources in some detail. The last section shows the results of estimating the effective seismic wavelets in the two wells YY04 and YY31 on seismic lines 1977 and 1973, respectively.

2.2. Geology of the area and objective of research

The Sirte Basin, in north-eastern Libya, was formed by large scale subsidence and block faulting which started in late Cretaceous time and continued, at least intermittently, to the Miocene and perhaps the present. Basement is formed by Precambrian to Cambro-Ordovician rocks, which were covered by early Palaeozoic, Mesozoic and Cenozoic sediments. Late Palaeozoic rocks were later removed by erosion. In Upper Cretaceous time, thick organic-rich shales, terrigenous clastics and evaporites were deposited on the down-faulted blocks or grabens. The horsts were probably sub-aerial at this time. In early Tertiary time (Palaeocene-Miocene) a marine transgression inundated the basin resulting in the deposition of carbonates (Augila Limestone). The carbonates grade into terrigenous clastics and evaporites to the south. This event was followed by a regression of the seas that culminated in Miocene time with relative emergence of the basin and retreat of the coastline to its present position, and development of the present geography by the end of Palaeocene time.

It was the Oligocene and Miocene regressions that resulted in the deposition of the thick Arida Formation. The Arida Formation is made up of the upper Arida shale and underlying Chadra sands. The Chadra sands were deposited on the Gialo structure, which is a horst and graben structure of mild relief defined by northwest-to-west trending faults. The Chadra sands may have a slightly unconformable contact with the underlying Augila limestone, but are conformably overlain by the thick Arida shale (Barr and Weegar, 1972).

The faults of the Gialo structure horst block break up the Gialo Field into separate pools. The amount of movement on the faults is uncertain, but it appears

that, at the Oligocene level, a throw of 3-5 m is most common. It is a characteristic of the Sirte Basin that deposition is contemporaneous with faulting. In this area fault-controlled movements formed a structurally high feature that was probably expressed as shoal during the deposition of the Chadra sands. Further, the orientation of the sand bodies in the Gialo Field area is largely controlled by the faults. Thus the Chadra sands are generally thinner over the high structure where it is approximately 60 m thick; to the east they are more than 90 m thick; in the west and northwest they are more than 120 m thick and more than 150 m thick in the south and southwest (Shelton, 1976).

The Chadra sands may be broadly divided into three sandstone units interbedded with shale, namely Chadra A, B and C. The sands are very fine to fine grained and are unconsolidated. Some discrete sand bodies in the Chadra units are recognised locally in Gialo Field. Units within A and B are more widely distributed as distinct bodies than units in Chadra C (Shelton, 1976). The interbedded Chadra shales that separate the sand bodies vary in thickness from less than a metre to approximately 15 m thick. Also, it should be noted that in spite of the structural growth, the thinner sections of the three sands are not vertically disposed but more nearly compensatory. That is, sand build-ups of Chadra C apparently contributed to a relatively thin section of Chadra B, and where Chadra B experienced build-up, Chadra A may be relatively thin. In general, the sand bodies are widespread and interfinger. They appear to have been deposited as sheet-like bodies or bars that pinch out up and down dip over the gently dipping surface of the Gialo structure. The bars are 4-6 km wide and 12-16 km long, and can be up to 30 m thick.

The contact relationships in the Chadra sands vary from massive sand bodies with sharp top and bottom contacts to a coarsening upward sequence that grades upwards from shales into sand, or a fining upward sequence that grades upwards from sand into shale. More commonly though, the sand bodies tend to have coarsening upward-fining upward cycles, with a hinge in the middle. In these cases, the hinge in the middle of the cycles is where the best reservoir rock (i.e. cleanest sand, highest porosity and permeability) and greatest hydrocarbon saturation occurs.

The Gialo Field is divided into three reservoirs, or pools, by the northwest trending faults: the E pool, the YY pool and the 4V pool. It is the objective of this research to define a seismic method to determine the distribution of the Chadra sands in the Gialo structure area by studying the possibility of delineating the sand bodies in the YY reservoir area.

In the YY area, the Chadra A and B sands are the dominant sand bodies (Shelton, 1976; Robinson, 1974). The Chadra B sand is older (stratigraphically lower) and is well developed over all the reservoir area. The thickest part of the Chadra B sand lies across the northwest portion of the YY reservoir forming a northwest trending elongated sand which thins off gently to the southwest. The Chadra B sand has a sharp lower contact and a thick shale bed which underlies the sand body. Figure 2.1 shows an isopack map of Chadra A, and Figure 2.2 shows an isopack map of Chadra B.

The Chadra A sand in the YY area consists of three sand bodies. The lower sand body is quite wide and covers most of the YY pool. The structurally high part is at the centre of the pool and coincides with the thickest lower sand body where it has about 30 m thickness. The sand pinches out rapidly in all directions. In most cases the lower sand has a sharp bottom contact with a thick shale layer which directly overlies the B sand.

The middle A sand body is found throughout the YY reservoir area. The thickest section of this sand body is found to the southwest of the structural high. This sand slopes off uniformly in all directions. A thick shale interbed lies between the middle and lower A sands, and the contacts are sharp.

The upper A sand body in the YY reservoir area occurs only in the southwest part of the area. It is thin and is separated from the underlying sand body by a reasonably thick shale barrier. The upper A sand is overlain by the thick Arida shale. For the purposes of this research, this sand is irrelevant since the seismic data available does not cover the area where it is present.

As previously mentioned, the objective of this research is to investigate the possibility of seismically delineating the Chadra sands in the YY reservoir area of Gialo Field. To do this, the seismic inversion method will be used to invert the seismic data into acoustic impedance to give an indication of lithology in the seismic traces. A computer program that performs the inversion of seismic data has been developed. The results from this research could later serve as a method to delineate the Chadra sands throughout the Gialo area, and then possibly attempt to detect the Chadra sands in Waha concession areas to the north and northeast of the Gialo area, where these sands are known to exist, with possible sand pinch outs which make good stratigraphic entrapment possibilities.
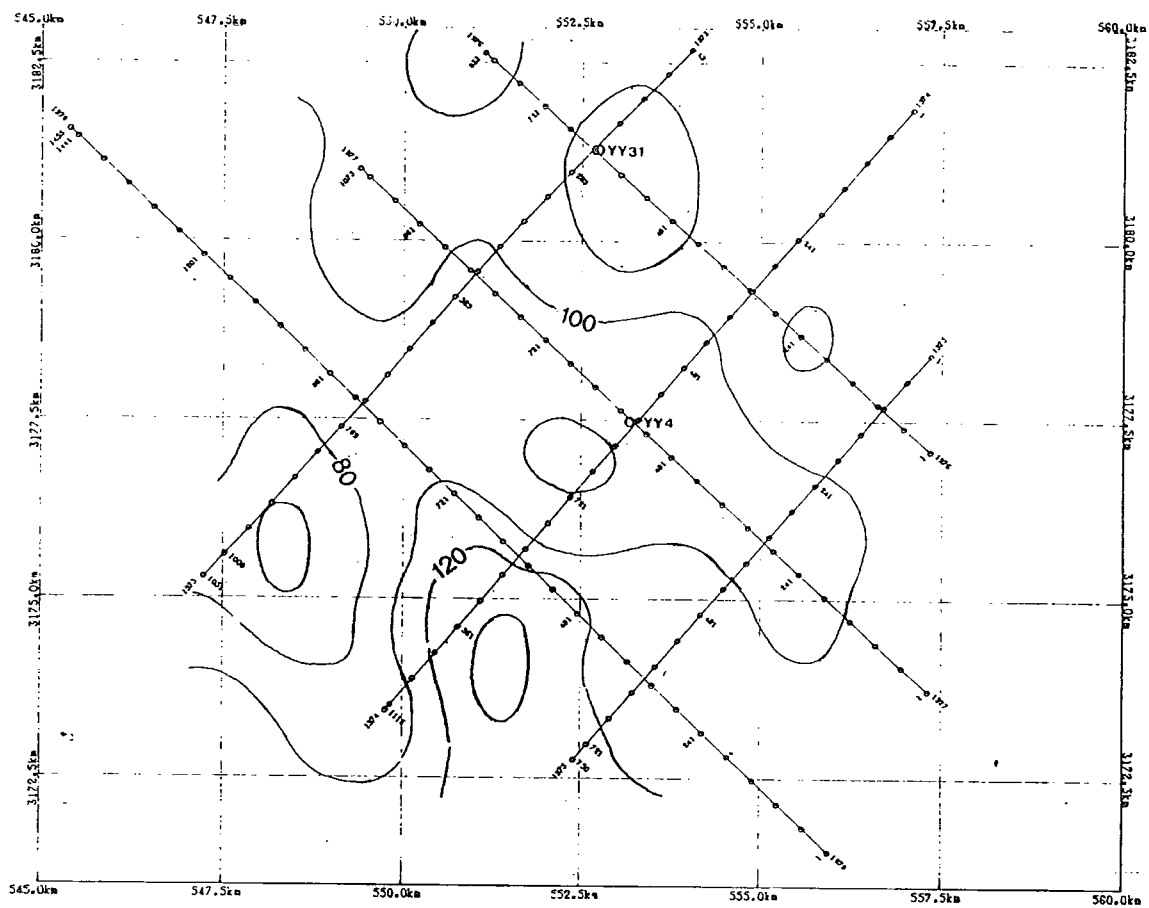
Figure 2.1. Approximate isopach map of Chadra A sand around the six seismic lines 1973-1978 in the Gialo Area. The contour interval is 20 ft. After Robinson (1974).

12
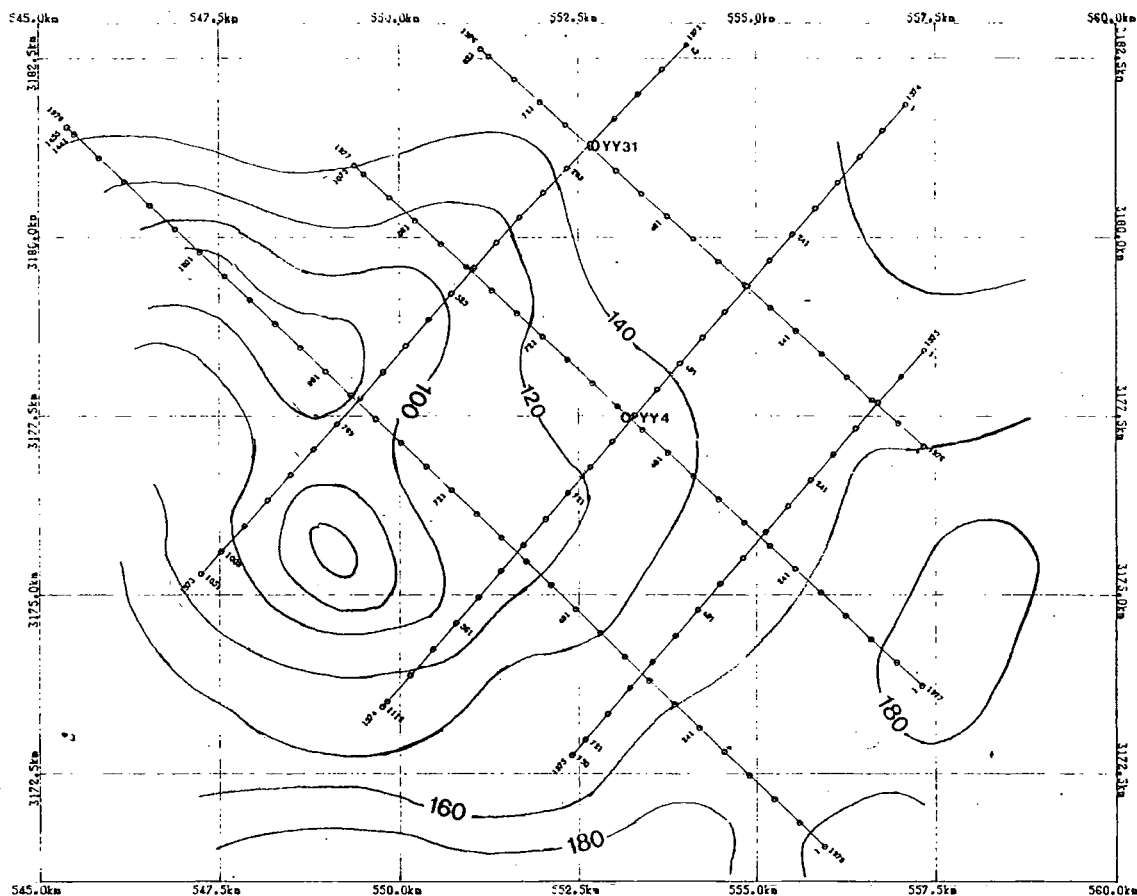
Figure 2.2. Approximate isopach map of Chadra B sand around the six seismic lines 1973-1978 in the Gialo Area. The contour interval is 20 ft. After Robinson (1974).

2.3. The well and seismic data available

2.3.1. Summary

The seismic data available for inversion are two intersecting seismic lines, namely lines 1973 and 1977. Located on line 1977 at shot point 548 (CDP 1093) is well YY04, and located on line 1973 at shot point 245 (CDP 555) and offset by 50 m is well YY31 (see Figure 2.1). In the sedimentary section of interest, sonic and density log data were recorded in both wells. None of the wells had check shots recorded in them, so problems with tying the well and seismic data are possible. The two seismic lines were recorded and later processed with the Oligocene Chadra sands in mind.

2.3.2. Field recording parameters and arrangements

The two seismic lines 1973 and 1977 are part of a survey comprising six lines totalling more than 60 km which were recorded in August 1989. The recording parameters were designed to allow the optimum possible frequency bandwidth for the Chadra sands section. In general, ground roll noise in the Sirte Basin tends to have long wavelength components and high amplitude, so that source and receiver arrays are always designed to be long enough to cancel-out the ground roll noise. In order to accommodate enough fold in the data, long receiver arrays translate into a long spread arrangements. Having a long spread is not a desirable recording pattern for the Chadra sands, because they are relatively shallow. Long off-set traces would have to be muted because of extensive NMO stretch, or even because of interference due to refracted arrivals. NMO stretch would add to the high frequency filtering effect of long receiver and sources arrays. The dilemma was that if one requires high frequency content in the data, then short source and receiver arrays have to be used implying that a dominant ground roll noise will also be recorded, thus resulting in seismic data with low signal-to-noise ratio. Conversely, if one uses long source and receiver arrays, which means also recording with long spread, then the resulting seismic data will not have the desirable high frequency content. The solution to this situation lies in the processing of the data: provided that the ground roll noise is recorded without spatial aliasing, then *fk*-filtering can be used to eliminate it and short receiver arrays will be satisfactory for recording the data.

From the analysis of two noise studies previously recorded in the area, it was decided to use 10 m receiver arrays and 20 m source arrays. A 120-channel symmetric split spread arrangement with the short 10 m length arrays were juxtaposed at 10 m spacing. The nearest offset (i.e. the nearest live geophone) was 65 m away

from the centre of the spread and the far offset was 655m. The energy source used was Vibroseis with a linear 10-82 Hz sweep, and the sweep length was 10 s. This arrangement gave a fold coverage of 30 fold. There was enough normal moveout at the Chadra sands two-way travel-time of approximately 600-750 ms to largely suppress multiple reflections. The sampling interval was 2 ms and the total two-way time recorded was 3 s.

2.3.3. Processing parameters and sequence

The two seismic lines 1973 and 1977 were actually processed twice, with Western Geophysical and CGG, both in London. Mainly the CGG processed lines will be discussed here because they were processed with preserved relative amplitudes and a zero phase wavelet; these data were used in the inversion analysis.

The processing sequence of the seismic data with CGG was as follows:

1. First, demultiplexing the field data, where the seismic traces for individual receivers are reassembled.
2. Sweep signal cross correlation with field recorded signals. This is done to compress the 13 s long frequency-sweep wave train into 3 s seismic traces.
3. Minimum phase conversion using the recorded sweep autocorrelation. This is necessary for seismic deconvolution.
4. Spherical divergence correction. This is applied to correct for geometrical spreading, that is, the signal amplitude decrease due to the energy distribution on expanding wave fronts.
5. Surface consistent amplitude correction to compensate for variations in coupling and attenuation by gathering traces according to their offsets, sources or receivers, and analysing their average amplitudes.
6. *fk*-filtering passing dips within the range – 5 ms/trace.
7. Gapped deconvolution where the operator length is 300 ms, the design window is from 0.2 to 2.7 sec, and pre-whitening of 1% was used. The gap was 16 ms long. This is a short gap in comparison to previous deconvolution gaps used on other vintages of data in the same area. This is due to the higher frequency content of this set of seismic data.
8. Application of field static corrections computed from the 49 upholes drilled on the six seismic lines, including the intersections. The seismic datum used is 100 m above sea level.
9. Long wavelength surface consistent residual static corrections. In CGG terminology a long wavelength anomaly means more than one-half of the maximum recording distance.
10. Short wavelength surface consistent residual static corrections. That is for less

than one-half of the maximum source-receiver offset distance.

11. Normal moveout and mute.

12. 3000% stack.

13. Zero phase conversion.

14. Bandpass filter of 6-10-80-90 Hz applied to the whole trace.

15. Application of a constant regional equalisation to all the data, since the relative amplitude of the data is preserved.

The dominant frequency obtained in these data was 50 Hz, which is better than any data recorded in this area before, where the dominant frequency was normally 35 Hz.

2.3.4. Well Data

There are two wells that can be used in this area; well YY04 and well YY31. Only YY04 is actually located on a seismic trace on line 1977, whereas YY31 is offset by 50 m to the nearest trace on seismic line 1973. Both of them have sonic and density measurements in the sedimentary section of interest, that is, the section between the Arida shale member, the three Chadra sand members (A, B, and C) of the Arida Formation, and the top part of the Augila Limestone. They also have resistivity, gamma ray, SP and caliper measurements. Well YY04 was recorded in 1970 and well YY31 in 1975, which meant that no digital recordings were available only paper displays. Furthermore, both wells have suffered from caving which sometimes is severe. When the caving is severe, the sonic log readings do not accurately represent the rock formation they measure, and this turned out to be a very important error to correct so that reasonably accurate acoustic impedance estimates at the two wells could be obtained.

2.4. Computing acoustic impedance from sonic and density logs

The acoustic impedance measurements at well locations are used to compute the reflection coefficient sequence, to compute an estimate of the seismic wavelet, and later in the inversion program to obtain an initial guess input for the seismic traces near the well, and also to give a general acoustic impedance trend for the whole area of study.

The impedance is computed from the velocity of the rock formations which is obtained from the sonic log, and the density which is obtained from the density log. In order to do this, we first need to digitise the analogue paper displays of these logs (by digitising on the break points, so that the values between any two adjacent points can be linearly approximated). The digitised data constitute the input to a FORTRAN program which has been specifically written to produce an acoustic impedance log as

16

an output, sampled at the seismic data sampling interval of 2 ms, and from which the reflection coefficient at the well is generated.

In this program, the first step is to transform the digitised data from the digitising table coordinates to the well log coordinates. This involves the translation of the point of origin of the digitised data to that of the well log, then rotating about this origin so that the two coordinate systems are fitted onto each other, and then scaling the data to the well log scale. This is done to both the sonic and density logs independently.

Because the sonic measurements, recorded as transit times, represent the time taken by an acoustic pulse to travel through a one foot thick section of the formation, and are hence given in units of microseconds per foot, it becomes logical to subdivide the section covered by the sonic log into one foot thick intervals and take their corresponding log readings as a measure of the travel time (one-way) through each interval. In doing so we have redigitised the sonic log into one foot thick intervals. Now, if we add as many of these one foot intervals (or their fractions) as needed to make the total of their corresponding transit time readings add up to 1 ms, the result will be a redigitised sonic log into 1 ms intervals, that is, we have sampled the sedimentary section measured by the sonic log into 1 ms sampling interval, or 2 ms sampling interval of two-way travel-time, which is the sampling interval of the seismic data.

To redigitise the sonic or density data into one foot intervals, the program takes the digitised depth values and simply rounds them to the nearest whole (integer) number of feet. This rounding will not result in great loss of accuracy in the depth values if we consider that normally any two successive digitised points cover at least a few feet of section; also an accuracy of − 0.5 ft at the ends of an interval of few feet should give an acceptable error. Having established these intervals along the digitised sonic or density log, the program then linearly interpolates within the end points, taken one interval at a time, to determine the transit time and density values at one foot spaced depth points. This is done for the complete log length, for both sonic and density.

In both wells in this area the sonic log covers most of the sedimentary section penetrated by the well while the density log covers only the zone of interest, including all the Chadra sand bodies. For this reason, when the sonic and density information are required to compute the acoustic impedance, the program selects only the zone of overlap between the sonic and density measurements and disregards the rest. The results of the program at this stage for the two wells are shown in Figures 2.3-2.6. Figure 2.3 shows the one foot sampled sonic log of well YY04 covering the interval of overlap with the density log. The depth interval covered here is approximately

17

600-951 m below the Kelly Bushing (KB). Only the one foot sampled data is shown in the Figure since they do not differ from the digitiser results. Figure 2.3 shows the one foot sampled density log for the same well in the same interval. In Figure 2.5 the sonic display for well YY31 is shown, where in this figure the depth interval covered is approximately 512-816 m, and Figure 2.6 shows the corresponding density interval. It is important to note that the difference in depth to the top of each interval in the wells does not reflect the structural dip in the area. The reason for this difference in depth is that recording of the density log in YY31 started at a shallower stratigraphic level and then only recorded a few metres in the underlying Augila Limestone, while in YY04 the density recording started at a deeper level and continued well into the underlying Augila. Also, note that all the results shown in Figures 2.3-2.6 are obtained after the analogue paper displays of the sonic and density measurements of the two wells were manually edited.

Next the selected sonic part is redigitised at 1 ms sampling interval. As previously mentioned, the way the program performs this is by adding as many of the one foot intervals of the sonic data, or their fractions, as needed to make the total of their transit times equal to 1 ms. It then determines their total thickness.

The thicknesses of the 1 ms intervals along with the selected density part (the part of the density data that overlaps the sonic data redigitised into one foot thick intervals) are used to define the thickness intervals in the density data that correspond to those 1 ms intervals of the sonic data. This is done by adding as many of the one foot thick intervals of the density log, or their fractions, as needed to make their total thickness equal to the thickness of the corresponding 1 ms intervals of the sonic, which were determined earlier, and then computes their average densities.

At this point we have 1 ms intervals with known thicknesses from which we can calculate their interval velocities. Also available are the corresponding average densities for these intervals, so we can readily compute their acoustic impedance and the corresponding reflection coefficient sequence for the well location. The acoustic impedance results for well YY04 are shown in Figures 2.5 and 2.6. In Fig. 2.6 the acoustic impedance is displayed against the two-way travel-time sampled at 2 ms, while Fig. 2.5 is a display of the same acoustic impedance sampled at the corresponding depth intervals. The difference, due to interval velocity, in time thickness and depth thickness of the limstone formations, starting at depth of about 790 m, and the shallower section of sandstone and shales is readily observed in the two Figures. Similar results for well YY31 are shown in Figures 2.9 and 2.10.

18
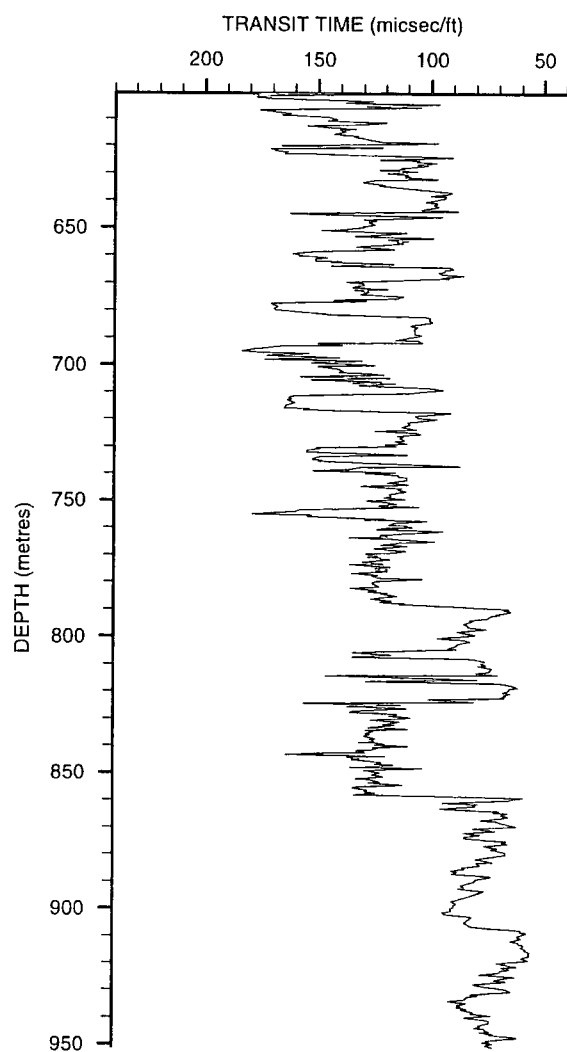
Figure 2.3 The sonic log of well YY04.
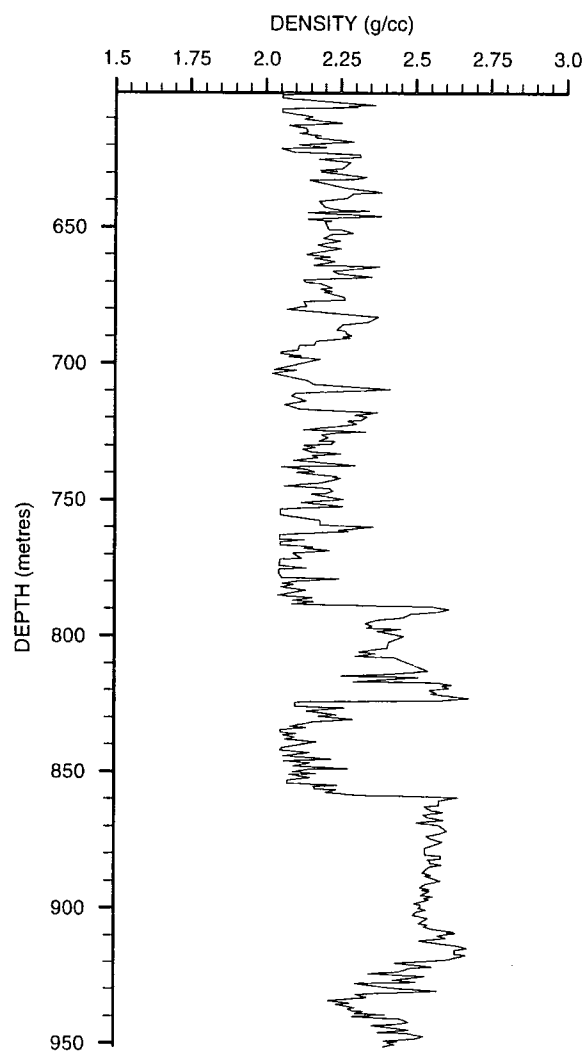
Figure 2.4 The density log of well YY04.
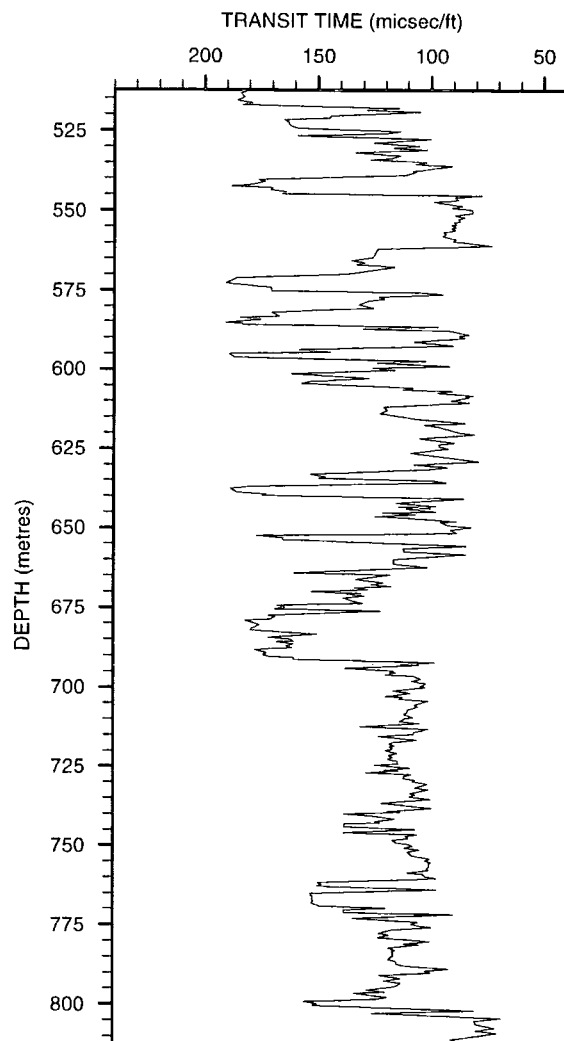
Figure 2.5 The sonic log of well YY31.
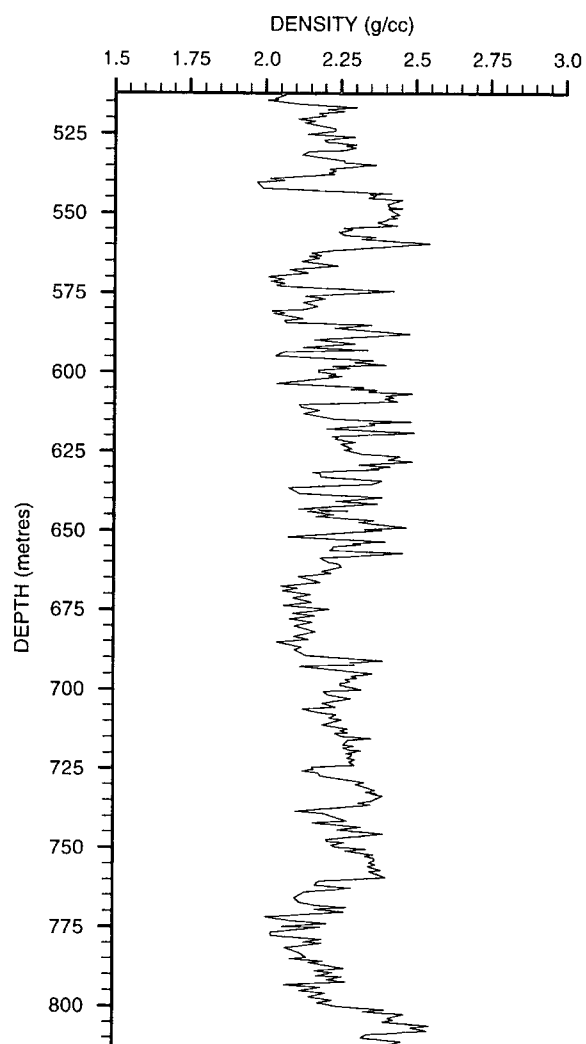
Figure 2.6 The density log of well YY31.
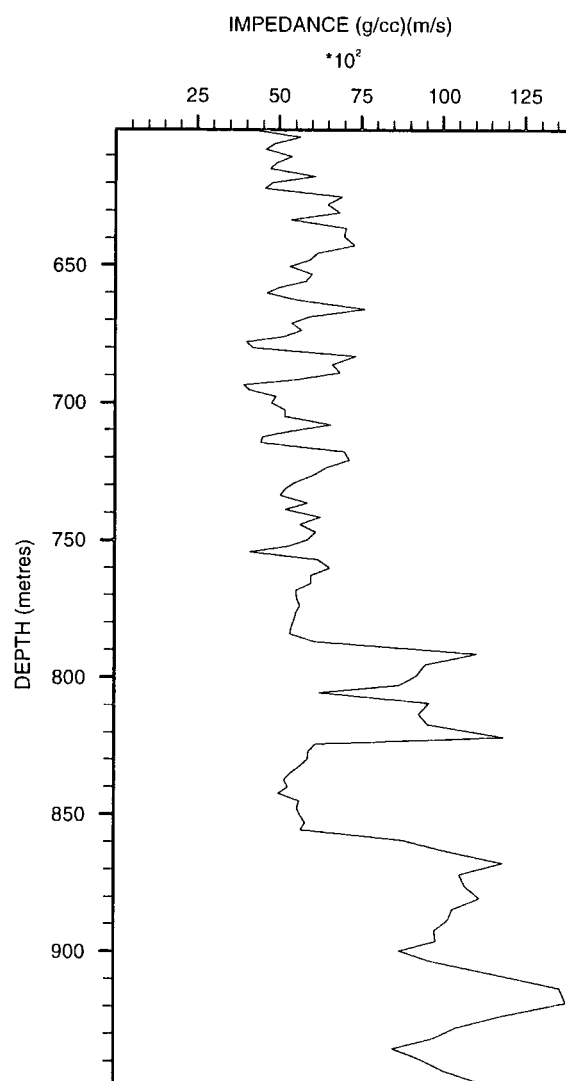
Figure 2.7 The acoustic impedance of well YY04 displayed against depth.
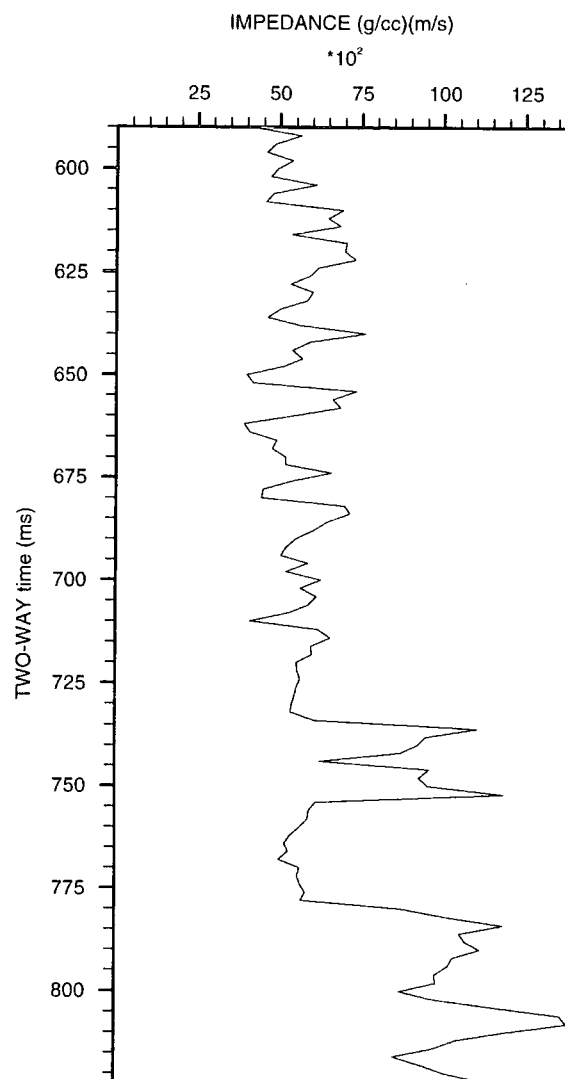
Figure 2.8 The acoustic impedance log of well YY04 displayed against two-way travel-time
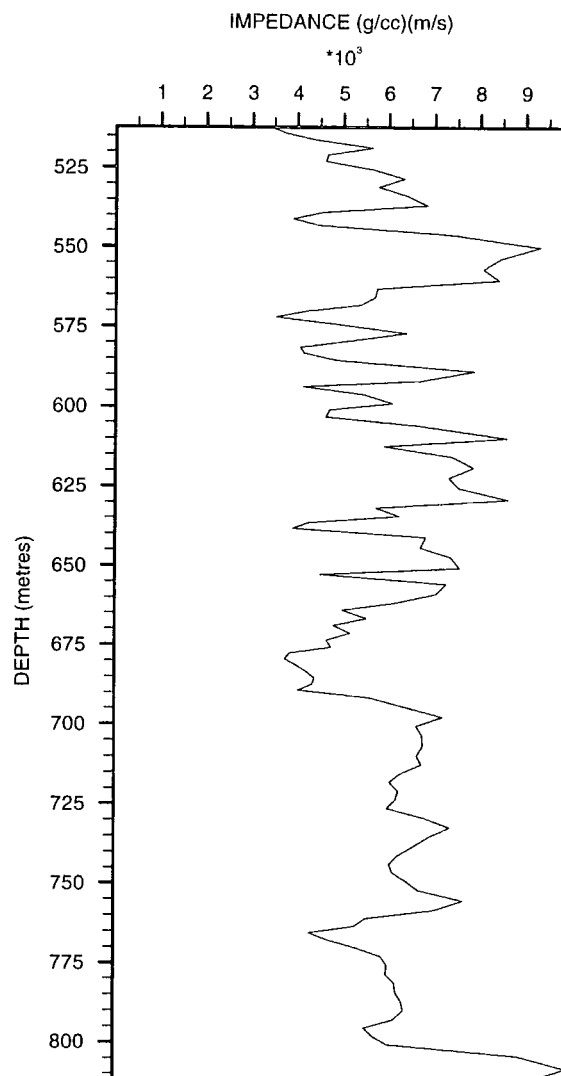
Figure 2.9 The acoustic impedance log of well YY31 displayed against depth.
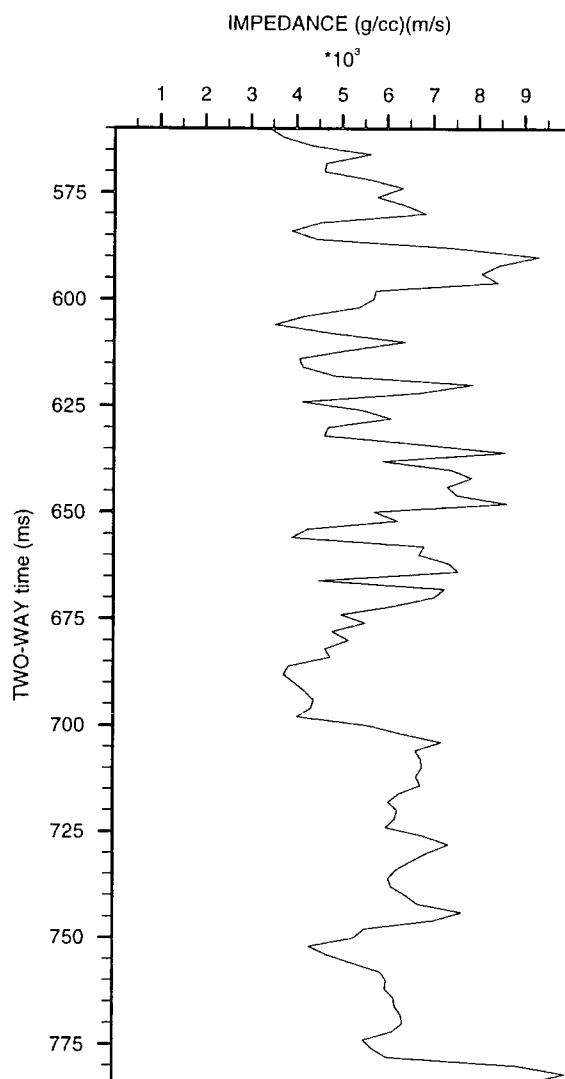
Figure 2.10 The acoustic impedance log of well YY31 displayed against two-way travel-time.

2.5. Estimating the wavelet from seismic and well data

The sonic and density information in a well can be used to obtain an acoustic impedance log, from which a reflection coefficient sequence at the well location can be computed. By assuming the noise-free convolution model, an estimate of the seismic wavelet can be made in a given seismic trace at, or close to, the well. That is, given

$$s(t) = w(t) * r(t) \qquad\qquad (2.1)$$

where $s(t)$ is the seismic trace at the well location with the computed reflection coefficient sequence $r(t)$, and $w(t)$ is the seismic wavelet to be estimated (Lines and Treitel, 1985), the problem becomes that of finding the equi-spaced coefficients, at the same sampling interval as the seismic traces, of a finite length Wiener shaping filter $w(t)$ that shapes $r(t)$ into $s(t)$. This is done by minimising the error energy between the desired output $s(t)$ and the actual output $c(t) = w(t) * r(t)$.

When minimising the error energy we obtain a system of linear simultaneous equations in the unknown wavelet coefficients. The solution of these normal equations comprises the wavelet coefficients. In solving the normal equations we use a recursive method developed by Levinson which exploits the symmetry about the main diagonal of the reflection coefficients autocorrelation matrix, or Toeplitz matrix. In order to stabilise the matrix division in the computation of the Wiener filter white noise may be added (Danielsen and Karlsson, 1984). This is achieved by adding a small positive constant to the diagonal in the Toeplitz matrix.

In practice we find that the reflection coefficient series determined at a well location is shorter than the seismic trace from which we need to compute the wavelet. Consequently we select a window on the seismic trace, the desired output, that corresponds to the reflection coefficient series. It is desirable to choose the seismic trace window to have the same length as the reflection coefficient series so that we can limit the contribution of those amplitude values outside the seismic trace window to the cross correlation function of the reflection coefficient series and the seismic trace window, hence limiting their effect on the wavelet estimate.

Estimates of the effective seismic wavelets in both wells YY31 and YY04 were made and the results are shown in Figures 2.11 and 2.12, respectively. In each well a set of seven wavelets was estimated. The central wavelet in each figure, wavelet number 4, is chosen as the effective seismic wavelet at the well location because it had the lowest error energy. For YY31 the effective wavelet was obtained when the tie between the acoustic impedance log and the seismic trace were at sample 280, which corresponds to two-way travel-time of 560 ms, while for YY04 the tie was

at sample 295, which corresponds to two-way travel-time of 590 ms. This large difference in the seismic tie for the two wells is largely due to the difference in depth at which each density log start. In both Figures 2.11 and 2.12, the wavelet estimate numbered 1 was obtained by shifting the acoustic impedance log 3 samples shallower than the tie sample position. In the wavelet numbered 2 the shift was 2 samples shallower, and in wavelet numbered 3 the shift was 1 sample shallower. For wavelet estimates numbered 5, 6 and 7 the time shift was deeper by 1, 2 and 3 samples with respect to the tie sample, respectively. This was done to optimise the wavelet estimate.
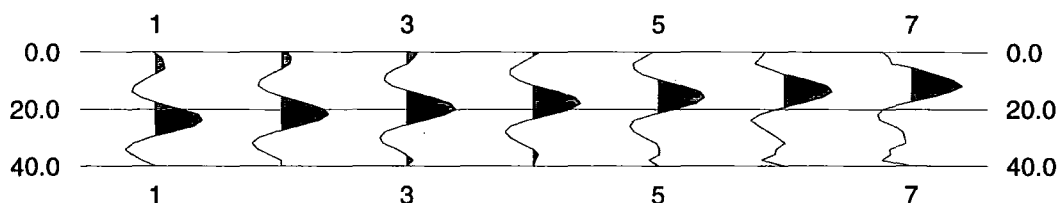
Figure 2.11 The wavelets numbered 1-7 are the effective seismic wavelets estimated at well YY31. The wavelet numbered 4 has the least error energy so that it is selected as the effective seismic wavelet at this well location. Note that the vertical axis represents the two-way travel-time in ms.
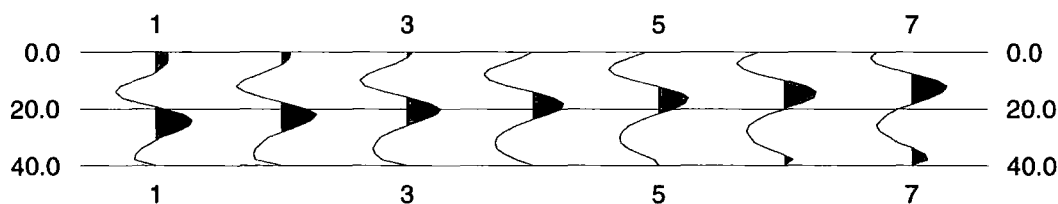
Figure 2.12 The wavelets numbered 1-7 are the effective seismic wavelets estimated at well YY04. The wavelet numbered 4 has the least error energy so that it is selected as the effective wavelet at this well location. Note that the vertical axis represents the two-way travel-time in ms.

2.6. The error in estimating the seismic wavelet in the area

It can be observed that there are two sources of error when estimating the seismic wavelet in any of the two wells. The first is due to inconsistencies in the seismic data and the second is due to inconsistencies in the well log data.

One obvious inconsistency in the seismic data applies to line 1973, where the nearest seismic trace on this line to well YY31-59 is about 50 m away. Such a distance could cause errors when estimating the wavelet, due to lateral variation in geology.

The other possible source of noise in the estimated seismic wavelet is due to the way in which the seismic amplitudes were processed. For example, if the seismic

data were processed toward structural interpretation, which is better achieved by minimising amplitude differences between events, automatic gain control is applied to the final display. This problem is minimised in this area by processing the two seismic lines concerned with preserved relative amplitudes. Relative amplitude preservation (Becquey *et al.*, 1979; Lavergne and Wills, 1977) means that a given reflecting horizon should show the same amplitude characteristic in the final processed seismic data as those measured by the sonic and density readings. This minimises the inconsistency in amplitudes between the computed reflection coefficient sequence and the corresponding seismic trace window, thus minimising errors in the cross correlation function and hence in the estimated seismic wavelet.

In this area, however, it is more likely that inconsistencies due to well log measurement errors will cause more instability in the estimated wavelet rather than inconsistencies of the seismic data. Errors in the log measurements are echoed as errors in the computed reflection coefficient sequence, and these errors will show up as high amplitude noise at the tail of the estimated wavelet.

To understand how errors in the computed reflection coefficient series are generated, we first need to understand how the sonic measurements are made with the sonic tool (Labo, 1986). In its simplest form a sonic tool consists of a pulse signal transmitter and two receivers. The receivers are placed one foot apart while the transmitter is positioned at least 5 feet away from the nearest receiver to separate the different wave modes that are generated when the pulse travels in the borehole, thus allowing the P-wave energy to arrive first at the receivers. The first arrival of each receiver wavetrain is then timed using a pre-assigned amplitude threshold. The arrival times at the two receivers are then subtracted and the difference Δt corresponds to the transit time, measured in μsft^{-1}, of the one foot section of the formation between the two receivers. The validity of the sonic measurements depends largely on changes in the hole diameter (Labo, 1986; Rider, 1986; Kokesh and Blizard, 1959). When the hole diameter is uniform and similar to the drill-bit size, the sonic readings are very reliable and no noise should be generated in the estimated seismic wavelet from the reflection coefficient series computed at such a well-behaved hole. When the hole diameter is larger than the drill-bit size, however, the hole is caved, due to fracture or erosion by the circulating mud, sonic readings may be unreliable. Caving is expected to occur in this area because the shales are young and unconsolidated, and also the Chadra sands are known to be fine-grained and loose. When the caving is serious, the travel path for the emitted pulse becomes considerably longer causing its amplitude to be significantly attenuated before it reaches the receivers. The far receiver signal will suffer more attenuation than the near receiver signal.

When the far receiver amplitude is attenuated enough and becomes near that of

the threshold amplitude, its detection time will come later than that of the near receiver causing the difference in detection time to be stretched. This is called Δt stretch and it can easily be up to $10\,\mu s$. The reading obtained by the sonic tool in this case will erroneously give a longer transit time, or lower velocity value, for the measured interval. In caved zones the far receiver attenuation may be so severe that the amplitude of the first cycle of the emitted signal drops below the threshold detection level, so this cycle is skipped and the second cycle is detected instead. This is called cycle skip and again gives an erroneous lower velocity value for the measured interval. If the cycle skip persists throughout the caved zone, the whole zone will be indicated as a low velocity zone by the sonic log. In some cases the caving is so severe that it becomes shorter for the transmitted pulse to travel through the mud than travelling through the formation, that is the P-wave energy lags the mud wave so the two receivers record the mud wave. In this case the sonic log gives a reading corresponding to the mud transit time, which is about $190\,\mu sft^{-1}$, instead of that of the formation. On the other hand, zones where the hole diameter is smaller than the drill-bit size are tight spots where the sonic tool might get obstructed while being pulled out during the recording of the sonic log. Such obstruction causes vibrations when the tool collides with the tight spots, and if they reach the receivers before the transmitted signal a higher velocity noise spike will be indicated by the sonic log. These can be readily seen on the sonic log as isolated spikes and thus edited out.

These unwanted logging effects will tend to increase (in the case of noise spikes), or decrease (in the case of Δt stretch, cycle skip or when recording the mud wave) the velocity values measured by the sonic log.

Enlargements in the hole diameter cause similar unwanted effects for the density log measurements because of the shallow depth of investigation of the density tool (about 10 cm) (Rider, 1986). If the hole diameter becomes large, the density tool loses contact with borehole wall and thus measures more of the drilling mud density which results in low density measurements.

The combined effect of enlarged holes is the measuring of low acoustic impedance in the zone where the hole was enlarged. When a reflection coefficient series is computed from this well, we obtain high amplitude reflection coefficients at the boundaries of the caved zone which do not represent the actual subsurface reflection coefficient series at the well location. When a seismic wavelet is estimated at the well, the computed reflection coefficient series does not compare to that in the seismic trace window; thus the cross correlation function will be in error which results in noise in the estimated seismic wavelet. Figure 2.13 shows an example of sonic log editing.
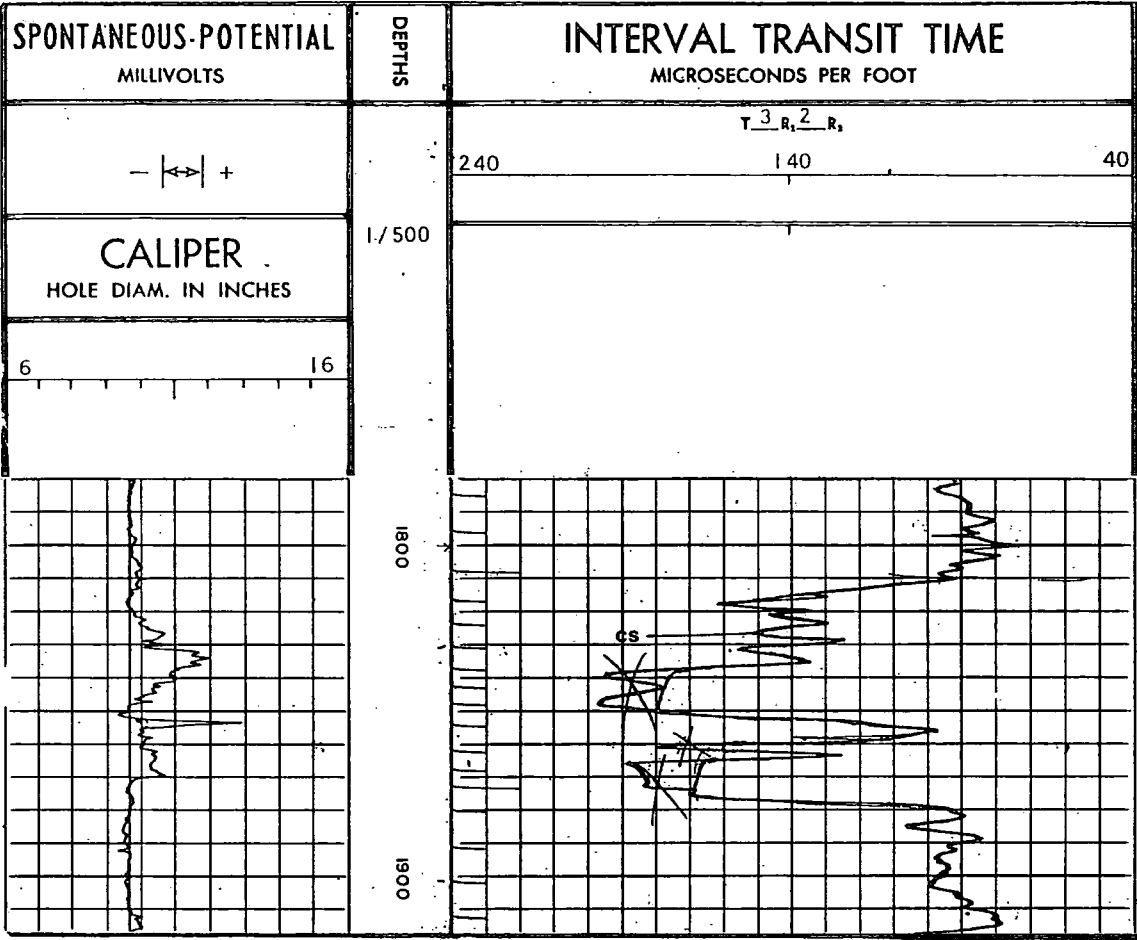
Figure 2.13. An example of sonic log editing. The spike denoted by cs, at depth 1828 feet, is an example of cycle skip. The part of the log marked by a cross is to be approximated by the solid line of lower transit time.

3

LEAST SQUARES INVERSION

3.1. Introduction

In this chapter the theory of least squares inversion is discussed. The least squares problem is presented as a minimisation of the error energy function which is the L_2-norm squared of the residual, or error, vector. The error vector is the difference between the observed seismic trace and the forward model, i.e. the synthetic seismogram. Thus the L_2-norm is a measure of the degree of fit between the observed and the synthetic trace. Since the synthetic seismogram is a function of certain subsurface parameters, the least squares inversion is an optimisation process in which the subsurface parameters are optimised to produce the best fit between the observed trace and the synthetic trace.

Where the synthetic seismogram is a linear function of the parameters, the optimisation problem becomes that of finding a generalised inverse of a rectangular matrix of derivatives, called the Jacobian matrix, which defines the variation of each of the error vector elements due to a change in each of the parameters. Where the synthetic seismogram is a non-linear function of the parameters, however, we linearise the problem and seek the solution iteratively, making use of the generalised inverse.

In section 3.2 the least squares problem is defined as the minimisation of the L_2-norm function. To find the minimum point, it is necessary to determine the gradient vector and a square matrix, called the Hessian matrix, that describes the curvature at the current point. For linear, or linearised, problems the Hessian can be approximated using the Jacobian matrix alone. Section 3.3 starts by defining a minimum point and discusses the sufficient and necessary conditions for its

existence. Since we are minimising problems with a linear forward model, this section also includes a brief discussion of the quadratic approximation of an error energy function. Section 3.4 discusses minimising a general error energy function using Gauss-Newton method. Since the error energy function to be minimised is not necessarily quadratic, the method is iterative. Section 3.5 shows that for the Gauss-Newton method to converge the Hessian matrix must be at least positive semi-definite. In practice, however, even if the Hessian is positive definite the Gauss-Newton solution might not give a reduction in error energy. The solution to this problem is to make linear search. Also, when the Hessian is positive definite but poorly conditioned, the Gauss-Newton solution may fail to produce a reduction in error energy; the obvious alternative in this case is to take the steepest descent direction. When the Hessian matrix is singular or indefinite, then the Hessian has to be modified so that it is positive definite for the Gauss-Newton method to converge.

Section 3.6 describes approximating the Jacobian matrix using finite differences. Section 3.7 gives a brief review of Cholesky factorisation and illustrates the important feature that the square root of any diagonal element of the positive definite symmetric Hessian constitutes an a *priori* bound on the elements of the corresponding row in its Cholesky factor. This feature is exploited in section 3.8 to modify an indefinite Hessian into a positive definite matrix in a minimum sense so that a sufficiently positive definite Hessian is not modified unnecessarily. Section 3.9 discusses a linear search using a backtracking strategy. In this method a local one dimensional quadratic model, in the Gauss-Newton descent direction, is minimised to obtain a step length that produces a decrease in error energy. In the final section 3.10, the problem of coping with a saddle point is discussed; at such a point the new descent direction must be defined in a direction of negative curvature.

3.2. Non-linear least squares inversion of seismic data

Seismic inversion may be viewed as a process of obtaining the best fit between the synthetic seismogram calculated from a subsurface geologic model and the finite set of field seismic observations. The subsurface geologic model is defined by two sets of parameters: the boundary location parameters, and the acoustic impedance parameters. The set of boundary location parameters describe the geometry of the geologic model and consist of the two-way travel-times to the base of each of the n lithologic units, or layers, that the model contains. Using vector notation they will be denoted by the n-vector $\mathbf{t} = [t_1 \ t_2 \ ... \ t_n]^T$. The set of acoustic impedance parameters describes the lithology of the geologic model. Each lithologic unit is described by two acoustic impedance parameters: the starting acoustic impedance and the linear acoustic impedance gradient within the unit. Thus for the n-layer model described by $\mathbf{t}$, we have an n-vector $\mathbf{x} = [x_1 \ x_2 \ ... \ x_n]^T$, where each element x_i describes the

starting acoustic impedance of the i-th layer in the model, and an n-vector $\mathbf{y} = [y_1 \, y_2 \, ... \, y_n]^T$ whose elements y_i describe the corresponding linear impedance gradients. Besides the boundary and acoustic impedance parameters, we also include in the inversion process the wavelet parameters that define the wavelet in the frequency domain. The wavelet parameters are four bounding frequency parameters, two amplitude parameters and three phase parameters, so that in vector form the wavelet is a 9-vector $\mathbf{w} = [w_1 \, w_2 \, ... \, w_9]^T$.

The synthetic seismogram is generated in the time domain using the noise-free convolutional model

$$s = w * r \qquad\qquad\qquad (3.1)$$

where s is the computed seismogram, or model response, w is the wavelet and r is the reflection coefficient sequence as defined by equation (1.1) in chapter 1. The actual computation of the synthetic seismogram is performed digitally, so that the resulting signal is a discrete time series denoted by s_i with a finite number of samples, say m. Similarly, the observed seismic data are also recorded in digital form and are denoted by s_i^{obs}.

The calculation of the time domain wavelet $w(t)$ from its parameters, and the calculation of the reflection coefficient sequence $r(t)$ from the boundary location and acoustic impedance parameters was the subject of a previous chapter. In this chapter we would like to think of the synthetic seismogram as a function of time represented by an m-vector $\mathbf{s}$ sampled at a sampling interval $\Delta\tau$ for time values $\tau_1, \tau_2, ..., \tau_m$ of the time variable τ. Thus $\mathbf{s}$ is a function of τ, and the boundary location, acoustic impedance and wavelet parameters. If we denote by an n-vector $\mathbf{x}$ the set, or in practice the subset, of those parameters of interest for a specific inversion problem, then $\mathbf{s}$ is a function of both τ and $\mathbf{x}$; i.e., $\mathbf{s}(\tau,\mathbf{x})$, with elements $s_i = s(\tau_i,\mathbf{x})$ for $i=1,2,....,m$. Corresponding to the synthetic seismogram are the field seismic observations s_i^{obs} recorded digitally at the same sampling interval. The purpose of inversion then becomes that of extracting model parameter estimates $\mathbf{x}$ that give the best fit of the synthetic seismogram $s_i = s(\tau_i,\mathbf{x})$ to the observed seismic data s_i^{obs}.

The best fit criterion used in this inversion is least squares, where the best fit is achieved when the sum of squares of the errors, or residuals, between the synthetic seismogram and the observed data is minimum. The m-vector of errors, or residuals, is defined as

34

$$\mathbf{r} = \mathbf{r(x)} = \left[r_1(\mathbf{x})\ r_2(\mathbf{x})\ \dots\ r_m(\mathbf{x}) \right]^T \tag{3.2}$$

where

$$r_i(\mathbf{x}) = s_i - s_i^{obs}\ , \quad \text{for } i=1,2,\ \dots\ ,m. \tag{3.3}$$

The total sum of the squares of the errors, the error energy, can be written as the dot product of the error vector, so that if we denote by $F(\mathbf{x})$ the function of this sum we have

$$F(\mathbf{x}) = \tfrac{1}{2} \sum_{i=1}^{m} \left[r_i(\mathbf{x}) \right]^2 = \tfrac{1}{2} \mathbf{r}^T(\mathbf{x})\mathbf{r(x)}\ , \tag{3.4}$$

where multiplication by $\tfrac{1}{2}$ is included to avoid the appearance of a factor of two in the derivatives. This is the non-linear least squares problem (Fletcher, 1987), where the synthetic seismogram is treated as a non-linear function of the parameters $\mathbf{x}$.

The contribution to the value of error energy $F(\mathbf{x})$ due to changes in the parameters x_1, x_2, $\dots$,x_n defines the gradient n-vector $\mathbf{g}$. Clearly the change in one parameter x_j will affect all elements of $\mathbf{r(x)}$ and each of these contributes to the total error energy $F(\mathbf{x})$ of (3.4). It is therefore convenient to define a new matrix $\mathbf{J(x)}$, called the Jacobian matrix of $F(\mathbf{x})$, which gives the variation of each $r_i(\mathbf{x})$, for $i=1,2,\ \dots\ ,m$, due to variation of each parameter x_j, that is

$$\mathbf{J(x)} = \begin{bmatrix} \dfrac{\partial r_1}{\partial x_1} & \dfrac{\partial r_1}{\partial x_2} & \cdot & \cdot & \dfrac{\partial r_1}{\partial x_n} \\[2mm] \dfrac{\partial r_2}{\partial x_1} & \dfrac{\partial r_2}{\partial x_2} & \cdot & \cdot & \dfrac{\partial r_2}{\partial x_n} \\[2mm] \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ \dfrac{\partial r_m}{\partial x_1} & \dfrac{\partial r_m}{\partial x_2} & \cdot & \cdot & \dfrac{\partial r_m}{\partial x_n} \end{bmatrix}. \tag{3.5}$$

Since we always have $m > n$ the matrix $\mathbf{J}$ is not a square matrix.

The elements of the gradient vector $\mathbf{g(x)}$ can now be derived by differentiating (3.4) with respect to each of the parameters x_j:

$$\frac{\partial F(\mathbf{x})}{\partial x_j} = \frac{\partial F}{\partial x_j} = \sum_{i=1}^{m} [r_i(\mathbf{x})] \frac{\partial r_i(\mathbf{x})}{\partial x_j} \quad , \tag{3.6}$$

so that the gradient vector

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} \dfrac{\partial F}{\partial x_1} \\ \dfrac{\partial F}{\partial x_2} \\ \cdot \\ \cdot \\ \dfrac{\partial F}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial r_1}{\partial x_1} & \dfrac{\partial r_2}{\partial x_1} & \cdot & \cdot & \dfrac{\partial r_m}{\partial x_1} \\ \dfrac{\partial r_1}{\partial x_2} & \dfrac{\partial r_2}{\partial x_2} & \cdot & \cdot & \dfrac{\partial r_m}{\partial x_2} \\ \cdot & \cdot & & & \cdot \\ \cdot & \cdot & & & \cdot \\ \dfrac{\partial r_1}{\partial x_n} & \dfrac{\partial r_2}{\partial x_n} & \cdot & \cdot & \dfrac{\partial r_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} r_1(\mathbf{x}) \\ r_2(\mathbf{x}) \\ \cdot \\ \cdot \\ r_m(\mathbf{x}) \end{bmatrix} \quad ,$$

or

$$\mathbf{g}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{r}(\mathbf{x}) \quad . \tag{3.7}$$

Assuming that $F(\mathbf{x})$ is twice continuously differentiable, so that the order of differentiation is interchangeable, a second differentiation of $F(\mathbf{x})$ gives

$$\frac{\partial^2 F(\mathbf{x})}{\partial x_k \partial x_j} = \sum_{i=1}^{m} \frac{\partial r_i(\mathbf{x})}{\partial x_k} \frac{\partial r_i(\mathbf{x})}{\partial x_j} + \sum_{i=1}^{m} r_i(\mathbf{x}) \frac{\partial^2 r_i(\mathbf{x})}{\partial x_k \partial x_j} \quad ,$$

or in matrix form, we obtain the Hessian matrix $\mathbf{H}(\mathbf{x})$ of $F(\mathbf{x})$,

$$\mathbf{H}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + \mathbf{S}(\mathbf{x}) \quad , \tag{3.8}$$

where $\quad \mathbf{S}(\mathbf{x}) = \displaystyle\sum_{i=1}^{m} r_i(\mathbf{x}) \frac{\partial^2 r_i(\mathbf{x})}{\partial x_k \partial x_j}.$

The problem with (3.8) is that the matrix $\mathbf{S}(\mathbf{x})$ is difficult to compute due to the second derivative term $\dfrac{\partial^2 r(\mathbf{x})}{\partial x_k \partial x_j}$, which will be expensive to approximate using finite differences.

At this point we would like to distinguish between linear or small residual problems, and non-linear or large residual problems. If we judge that the residuals $r_i(\mathbf{x})$ are small , then a good approximation to $\mathbf{H}(\mathbf{x})$ is still obtained after neglecting the last term $\mathbf{S}(\mathbf{x})$ in (3.8), (Fletcher, 1987), which gives

$$H(x) = J(x)^T J(x) \qquad\qquad (3.9)$$

This is equivalent to making linear approximation to the residuals $r_i(x)$. Thus using only the information from $r(x)$ and $J(x)$, which are required to determine the gradient (first derivative) vector $g(x)$, it is possible to approximate the Hessian (second derivative) matrix $H(x)$.

If, on the other hand, the residuals $r_i(x)$ are large, or the residuals are highly non-linear in the parameters vector x, then the contribution to $H(x)$ in (3.8) from $S(x)$ is significant and $S(x)$ should either be calculated or approximated. Any method that attempts to include $S(x)$ to define $H(x)$ is called the Newton method. If, however, $H(x)$ is approximated by (3.9) the resulting method is called the Gauss-Newton method. The Gauss-Newton method can be used to solve non-linear least squares inverse problems iteratively, that is, by taking a sequence of linear steps to the minimum (Lines and Treitel, 1984). This is the method used to minimise the error energy function $F(x)$ in this work.

3.3. Minimisation of the error energy function $F(x)$

3.3.1. Definitions of minimum points

Before proceeding to discuss the Gauss-Newton method used in the minimisation of the error energy function $F(x)$, local minimum definitions, the conditions by which it could be verified, and a brief discussion of quadratic functions are first given.

Definition 1:   A point x^* is said to be a strong local minimum of the function $F(x)$ if there exists a scalar δ defining a neighbourhood of x^* such that $F(x^*) < F(x^* + p)$, for all p satisfying $0 \le \| p \| \le \delta$, where $\| . \|$ denotes the L_2-norm.

Definition 2:   A point x^* is said to be a weak local minimum of the function $F(x)$ if there exists a scalar δ defining a neighbourhood of x^*, such that $F(x^*) \le F(x^* + p)$, for all p satisfying $0 \le \| p \| \le \delta$.

These two definitions imply that x^* is not a local minimum if every neighbourhood of x^* contains at least one point with a strictly lower function value (Gill *et al.*, 1981).

3.3.2. The first order condition

The first-order condition, by which we can verify that the function $F(x)$ has a minimum at the point x^*, can be deduced by approximating the function value $F(x^* + p)$ at a neighbouring point $x^* + p$ by using the first three terms of the Taylor series expansion

$$F(x^* + p) = F(x^*) + p^T g(x^*) + \tfrac{1}{2} p^T H(x^*) p + \dots \quad (3.10)$$

given that $g(x^*) = \nabla F(x^*)$ is the first derivative vector or gradient of $F(x)$ at x^*, and $H(x^*) = \nabla^2 F(x^*)$ is the matrix of second derivatives, or Hessian, of $F(x)$ at x^*. The higher order terms involving higher derivatives can be neglected for small enough $\| p \|$.

The first-order condition is proved by contradiction from the first two terms of equation (3.10) (Scales, 1985). If $p^T g(x^*) < 0$, then the first-order Taylor series expansion implies that $F(x^* + p) < F(x^*)$ which contradicts the definition of the minimum given above. If, however, $p^T g(x^*) > 0$, then $F(x^* - p) < F(x^*)$ which again contradicts the definition of the minimum. Hence, unless $p^T g(x^*) = 0$, which implies that $g(x^*) = 0$, every neighbourhood of x^* contains points with strictly lower function value than $F(x)$. This proves that at every local minimum

$$g(x^*) = 0 \quad (3.11)$$

which is the first-order condition for $F(x)$ to have a minimum. If condition (3.11) is satisfied, x^* is said to be a stationary point.

3.3.3. The second-order condition

The first-order condition is necessary but not sufficient for the point x^* to be a minimum. This is because a maximum or a saddle point at x^* can also satisfy the first-order condition (Gill *et al.*, 1981).

For x^* to be a local minimum we have to consider the second-order condition which can be derived from the first three terms of the Taylor series expansion (3.10) and the first-order condition (3.11)

$$F(x^* + p) = F(x^*) + \tfrac{1}{2} p^T H p \quad (3.12)$$

for a small enough $\| p \|$. If H is an indefinite matrix, then $p \neq 0$ can be chosen so that $p^T H(x^*) p < 0$. This would imply from (3.12) that every neighbourhood of x^* contains points of lower function value, which contradicts the definition of the

minimum. If, on the other hand, $\mathbf{p}^T\mathbf{H}(\mathbf{x}^*)\mathbf{p} > 0$ for all $\mathbf{p} \neq \mathbf{0}$, which implies that $\mathbf{H}$ is positive definite, then there is no point $\mathbf{x}^* + \mathbf{p}$ in the neighbourhood of $\mathbf{x}^*$ with lower function value. Therefore, for $\mathbf{x}^*$ to be a local minimum the Hessian matrix $\mathbf{H}$ has to be positive definite, which is the second-order condition.

For a general function $F(\mathbf{x})$, the second-order condition is not necessary for a strong minimum, because a minimum can still be strong if, for some $\mathbf{p}$, $\mathbf{p}^T\mathbf{H}(\mathbf{x}^*)\mathbf{p} = 0$, but the third-order term in Taylor series expansion (3.10) is positive. The second-order necessary condition for a strong minimum at $\mathbf{x}^*$ is that $\mathbf{H}$ is positive semi-definite (Scales, 1985). For a quadratic function $F(\mathbf{x})$, however, a positive definite $\mathbf{H}(\mathbf{x}^*)$ implies that $\mathbf{x}^*$ is a strong minimum, and a positive semi-definite $\mathbf{H}(\mathbf{x}^*)$ implies that the point is a weak minimum.

3.3.4. Function approximation using quadratic models

In the minimisation of a smooth error function $F(\mathbf{x})$, the approximation

$$F(\mathbf{x}+\mathbf{p}) = F(\mathbf{x}) + \mathbf{p}^T\mathbf{g}(\mathbf{x}) + \tfrac{1}{2}\mathbf{p}^T\mathbf{H}(\mathbf{x})\,\mathbf{p} \qquad (3.13)$$

is applied to give a quadratic approximation to $F(\mathbf{x}+\mathbf{p})$ and the process of finding the minimum is iterative. Iterative methods based on the quadratic approximation are simple and have rapid rates of local convergence when applied to general functions. Indeed if the function $F(\mathbf{x})$ is quadratic the minimisation is reduced to solving a linear system of equations, and so converges in only one iteration. Some of the reasons for using the quadratic approximation are the following:

1. For a quadratic function, any derivatives of order higher than the second are zero, and so the first three terms of its Taylor series expansion are exact regardless of the value of $\mathbf{p}$.

2. For a general function having a continuous second derivative, quadratic behaviour could be obtained over a sufficiently small neighbourhood of a local minimum $\mathbf{x}^*$.

3. Even away from the local minimum, quadratic information is more effective than linear information in predicting directions $\mathbf{p}$ in which a substantial decrease in the error function are made. This is because a Taylor series expansion of $F(\mathbf{x}+\mathbf{p})$ about the point $\mathbf{x}$ truncated after the quadratic terms will approximate $F(\mathbf{x}+\mathbf{p})$ to a given accuracy over a much larger neighbourhood of $\mathbf{x}$ than will the series expansion taken to linear terms only (Fletcher, 1987).

3.4. Minimisation of $F(x)$ using the Gauss-Newton method

The Gauss-Newton method when applied to the minimisation of a general function $F(x)$ is iterative. At the beginning of the k-th iteration let the current estimate of the minimum point be x_k. A descent vector p_k, called the Gauss-Newton vector, is determined and used to update x_k to obtain a new estimate x_{k+1} of the minimum.

The method is based on the quadratic approximation of $F(x)$, where derivatives of $F(x)$ that are higher than second-order are neglected. The quadratic approximation is obtained from the Taylor series expansion of $F(x)$ about x_k. That is, given the function value $F_k = F(x_k)$, the gradient $g_k = g(x_k)$ and the Hessian $H_k = H(x_k)$, at the point x_k, we have

$$F(x_k + p) = F_{k+1} = F_k + g_k^T p + \tfrac{1}{2} p^T H_k \, p \tag{3.14}$$

where $p = x - x_k$ and F_{k+1} is the quadratic approximation of $F(x)$ following the k-th iteration. Then the next estimate of the minimum is $x_{k+1} = x_k + p_k$, where p_k minimises the quadratic approximation F_{k+1}. The point x_{k+1} minimises F_{k+1} only if the Hessian H_k is positive definite, which is the second-order condition. Then at the minimum point x_{k+1} we have

$$\nabla F_{k+1} = g_{k+1} = g(x_k + p_k) = 0 \tag{3.15}$$

which is the first-order necessary condition. Thus from (3.14) and (3.15), we have

$$g_{k+1} = g_k + H_k \, p_k = 0 . \tag{3.16}$$

Then p_k is given by

$$H_k \, p_k = -g_k \; . \tag{3.17}$$

If $F(x)$ is quadratic, the global minimising point $x_{k+1} = x_k + p_k$ will be found in one iteration by solving the linear system (3.17) for p_k. For non-quadratic $F(x)$, $x_{k+1} = x_k + p_k$ will not be the minimum of $F(x)$ and the process has to be repeated iteratively.

3.5. Convergence of Gauss-Newton's Method

For the Gauss-Newton method to converge, it is of course required that $\mathbf{p}_k$ has to be a descent direction. This follows from rewriting equation (3.17) as

$$\mathbf{p}_k = -\mathbf{H}_k^{-1}\,\mathbf{g}_k \,, \tag{3.18}$$

hence

$$\mathbf{g}_k^T\mathbf{p}_k = -\mathbf{g}_k^T\mathbf{H}_k^{-1}\mathbf{g}_k < 0. \tag{3.19}$$

This is true for $\mathbf{g}_k \neq \mathbf{0}$ if $\mathbf{H}_k^{-1}$ is positive definite. This means that $\mathbf{H}_k$ is also positive definite. In the neighbourhood of a strong minimum, the Hessian matrix $\mathbf{H}$ is positive definite and the function $F(\mathbf{x})$ resembles the corresponding quadratic approximation. Therefore, Gauss-Newton's method converges to the minimum given that a sufficiently good quadratic approximation of $F(\mathbf{x})$ is possible.

In practice, however, difficulties with convergence to a minimum $\mathbf{x}^*$ may arise because $\mathbf{H}_k$ may not be everywhere positive definite. Even if $\mathbf{H}_k$ is positive definite at a point that is not a minimum, the point $\mathbf{x}_{k+1}$ may lie outside the region for which the quadratic approximation at $\mathbf{x}_k$ is valid; that is, the non-linearity of $F(\mathbf{x})$ between $\mathbf{x}_k$ and $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$ cannot be predicted by the Hessian alone. For example, if the curvature of the function $F(\mathbf{x})$ in part of the region between $\mathbf{x}_k$ and $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{p}_k$ is sharper than that predicted by the quadratic approximation, then the step $\mathbf{p}_k$ is too large and $\mathbf{p}_k$ has to be scaled to make a smaller step. This modification to the Gauss-Newton method can be achieved by doing a linear search along $\mathbf{p}_k$ to determine a scalar $\alpha > 0$ such that $F(\mathbf{x}_k + \alpha\mathbf{p}_k) < F(\mathbf{x}_k)$. The actual method of linear search used to find α will be discussed later in this chapter. However, it is important to emphasise that the natural value for α is unity, and we should expect that $\alpha \to 1$ as $\mathbf{x}_k \to \mathbf{x}^*$ (Gill *et al.*, 1981; Fletcher, 1987).

Other situations where Gauss-Newton's method might not converge are likely. One possibility is that the Hessian $\mathbf{H}_k$ is positive definite, but $\mathbf{g}_k^T \mathbf{p} = 0$ when $\mathbf{g}_k \neq \mathbf{0}$, which means that $\mathbf{x}_k$ is already a well determined minimum along $\mathbf{p}_k$, because $\mathbf{p}_k$ is orthogonal to $\mathbf{g}_k$, that is, moving along $\mathbf{p}_k$ is like moving along the contour line defined by $\mathbf{x}_k$ so that further progress in convergence would not be possible. One readily available solution in this case is that $\mathbf{p}_k$ takes the steepest descent direction. This behaviour can be explained if we consider equation (3.19) (Gill *et al.*, 1981), from which it can be noticed that if the condition number of $\mathbf{H}_k$ is not bounded by a

constant that is independent of x_k , then the solution of (3.17) will not be bounded away from orthogonality to the negative gradient.

Another possibility is that H_k is singular, in which case, as indicated by equation (3.17), there will be a solution p_k if g_k is a linear combination of the columns of $\mathbb{H}_k$, one of many possible solutions, so that no unique solution exists. If g_k is not a linear combination of the columns of $\mathbb{H}_k$, then no solution will exist at all.

One last possibility is that the Hessian H_k is indefinite. Then x_k is a saddle point if H_k is non-singular and $g_k = 0$; hence the only solution is the trivial solution $p_k = 0$. In this case the quadratic approximation is unbounded below (Gill *et al.*, 1981) and thus indicates that an infinite step could be taken from x_k. One possible direction for p_k that could be taken in this case is a direction of negative curvature.

Clearly, the Gauss-Newton method is not always satisfactory in minimising general functions, even when used in conjunction with linear search. Fortunately, it can be modified to provide a very reliable method. The general idea behind the modified Gauss-Newton method, as used here, is to replace H_k by a matrix $\overline{H}_k$ which is guaranteed always to be positive definite. The matrix $\overline{H}_k$ is chosen to be H_k whenever H_k is sufficiently positive definite.

3.6. Finite difference approximation of the Jacobian matrix

To compute the Hessian matrix or the gradient vector, we must compute the Jacobian matrix J. The partial derivatives of the Jacobian are approximated by finite difference derivative methods. Using the forward difference formula, the partial derivatives of the Jacobian (3.5) are approximated by

$$\frac{\partial r_i}{\partial x_j} = \frac{r_i(x_j + \delta x_j) - r_i(x_j)}{\delta x_j} \quad , \tag{3.20}$$

where δx_j is a small step size for the j-th column of the Jacobian. A crucial decision to be made here is the choice of the step size δx_j.

Dennis and Schnabel (1983) suggested using a step size $\delta x_j = \sqrt{\varepsilon_M}.x_j$, where ε_M is the machine precision. This will ensure that the rounding errors are acceptable. Notice that a constant step size is not used, since each δx_j depends on the value of x_j. This is highly recommended because in practice using a uniform step size could be disastrous if the components of x differ widely in magnitude. However, because x_j could become close to zero, the step size δx_j should be chosen in the following way:

$$\delta x_j = \sqrt{\varepsilon_M} \cdot \max\{|x_j|, \text{ typical } x_j\} \cdot \text{sign}(x_j) \,, \qquad (3.21)$$

where typical x_j is a typical size of x_j. A typical x_j chosen will depend on which set of parameters we are trying to invert for. For example, if we are inverting for the acoustic impedance, a typical starting impedance is x_j=8000 gcm^{-3}msec^{-1}, while for the impedance gradient a typical x_j=100 gcm^{-3}msec^{-1}/sample interval

Dennis and Schnabel (1983) suggested further that to improve the accuracy of the step δx_j as given by (3.21), we should make the following calculation

$$\delta x_j = (x_j + \delta x_j) - x_j \,\, .$$

This should improve the accuracy of any finite difference approximation in practice, since this tends to cancel out the rounding errors in machine representation of x_j.

3.7. The Cholesky factorisation

When, at the k-th iteration, the symmetric Hessian is the positive definite matrix $\mathbf{H}_k$, it can be factored using the $\mathbf{LDL}^T$ factorisation as

$$\mathbf{H}_k = \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T \qquad (3.22)$$

where $\mathbf{L}_k$ is a unit lower triangular matrix, and $\mathbf{D}_k$ is diagonal matrix with positive diagonal elements.

Since the diagonal elements of $\mathbf{D}_k$ are strictly positive, then (3.22) can be written as

$$\mathbf{H}_k = \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T = \mathbf{L}_k \mathbf{D}_k^{\frac{1}{2}} \mathbf{D}_k^{\frac{1}{2}} \mathbf{L}_k^T = \mathbf{R}_k^T \mathbf{R}_k \qquad (3.23)$$

where $\mathbf{R}_k$ is upper triangular matrix. Factorisation (3.23) is known as Cholesky factorisation. For the purpose of this work, however, either of the forms (3.22) or (3.23) will be referred to as the Cholesky factorisation (Gill *et al.*, 1981).

An important feature of the Cholesky factorisation could be revealed if we express the k-th diagonal element of $\mathbf{H}_k$ in equation (3.23) in terms of the elements of the s-th row of $\mathbf{R}_k$, that is

$$r_{1s}^2 + r_{2s}^2 + \,\ldots\, + r_{ss}^2 = h_{ss} \,, \qquad s=1,\,2,\,\ldots\,,n. \qquad (3.24)$$

Because the diagonal elements of $\mathbf{H}_k$ are strictly positive, expression (3.24) provides the following a *priori* bound on the elements of $\mathbf{R}_k$,

$$|r_{is}| \leq \sqrt{h_{ss}} \quad . \tag{3.25}$$

Thus the elements of $\mathbf{R}_k$ cannot grow without a bound, which makes this factorisation numerically stable even for a Hessian matrix dominated by zero, or very small, elements (Gill and Murray, 1974).

3.8. A Cholesky factorisation method for an indefinite Hessian

In this method we construct a positive definite matrix $\overline{\mathbf{H}}_k$ from a modified Cholesky factorisation of $\mathbf{H}_k$, that is

$$\overline{\mathbf{H}}_k = \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T \tag{3.26}$$

where $\mathbf{L}_k$ and $\mathbf{D}_k$ are the modified Cholesky factors of $\overline{\mathbf{H}}_k$ (Scales, 1985).

The Cholesky factorisation (3.22) can be performed column-wise, where at each step a column of $\mathbf{L}_k$ and $\mathbf{D}_k$ are determined, so that the j-th step is given by

$$d_j = h_{jj} - \sum_{r=1}^{j-1} l_{jr}^2 d_r \tag{3.27}$$

$$l_{ij} = \frac{h_{ij} - \sum_{r=1}^{j-1} l_{rj} l_{ir} d_r}{d_j} \quad , \quad i = j+1, \, j+2, \, \dots \, ,n. \tag{3.28}$$

We can make the analysis and computations more suitable if we rewrite (3.27) and (3.28) by making the substitution

$$c_{jr} = l_{jr} d_r \quad . \tag{3.29}$$

Thus we have,

$$d_j = h_{jj} - \sum_{r=1}^{j-1} l_{jr} c_{jr} \quad , \tag{3.30}$$

and

$$c_{ij} = h_{ij} - \sum_{r=1}^{j-1} l_{jr} c_{ir} \quad , \quad i=j+1, j+2, \dots ,n. \tag{3.31}$$

According to equation (3.28), if the off-diagonal elements are too large then their size can be decreased by increasing the elements of the diagonal matrix $\mathbf{D}_k$. This allows the Cholesky factors $\mathbf{L}_k$ and $\mathbf{D}_k$ to be computed subject to two requirements: (1) the elements of the diagonal matrix $\mathbf{D}_k$ are strictly positive, and (2) the elements of the factors $\mathbf{L}_k \mathbf{D}_k^{\frac{1}{2}}$ satisfy an a *priori* bound (Gill *et al.*, 1981).

The factorisation is computed in column order and at the j-th step the first j-1 columns are already computed. Let β be a constant that defines the bound on the elements of $\mathbf{L}_k \mathbf{D}_k^{\frac{1}{2}}$, and δ be a small positive number; then

$$\left| l_{rs} d_s^{\frac{1}{2}} \right| \leq \beta \quad , s=1, 2, \dots ,j\text{-}1, \quad r=1, 2, \dots ,n \tag{3.32}$$

and

$$d_j > \delta.$$

The value of β will be discussed later. δ is introduced to improve the condition of a positive definite, but very ill-conditioned $\mathbf{H}_k$. A suitable choice for δ is the relative machine precision ε_M (Gill *et al.*, 1981).

To compute the modified factors, we first compute

$$\phi_j = h_{jj} - \sum_{r=1}^{j-1} l_{jr} c_{jr} \tag{3.33}$$

and set an estimate for d_j as

$$\overline{d}_j = \max \left\{ \delta, \left| \phi_j \right| \right\} \quad , \tag{3.34}$$

and then compute

$$c_{ij} = h_{ij} - \sum_{r=1}^{j-1} l_{jr} c_{ir} \quad , \quad i=j+1, j+2, \dots ,n. \tag{3.35}$$

and find

$$l_{ij} = \frac{c_{ij}}{\overline{d}_j}.$$

When $\mathbb{H}_k$ is not sufficiently positive definite, then equation (3.33) will give $\phi_j < \delta$. In this case we could set $\overline{d} = \delta$. However the choice $\overline{d}_j = |\phi_j|$ has proved to be a better choice in practice (Gill and Murray, 1974).

At this point the initial estimate $\overline{d}_j$ of the diagonal element d_j could be modified if the off-diagonal column elements $l_{ij}d_j^{\frac{1}{2}}$, $i=j+1$, $j+2$, ... ,n are not bounded by β. Let

$$\theta_j = \max\left\{|c_{ij}|, i = j+1, j+2, ..., n\right\}. \qquad (3.35)$$

Since $c_{ij} = l_{ij}\overline{d}_j$ then clearly if $\theta_j^2 = \beta^2 \overline{d}_j$ the elements of $l_{ij}\overline{d}_j^{\frac{1}{2}} \leq \beta$, that is they are bounded by β. On the other hand if $\theta_j^2 > \beta^2 \overline{d}_j$, we choose d_j such that

$$\beta = \max\left\{|l_{ij}d_j^{\frac{1}{2}}|, i = j+1, j+2, ..., n\right\} \quad,$$

so that the largest in modulus of the elements $l_{ij}d_j^{\frac{1}{2}}$ is taken to be exactly equal to β. Thus d_j is set as follows:

$$d_j = \max\left\{\overline{d}_j, \theta_j^2/\beta^2\right\}.$$

Having determined d_j, the elements of the *j-th* row of $\mathbf{L}_k$ are computed as in the unmodified Cholesky factorisation method. Also, when the *j*-th column of $\mathbf{L}_k$ is ultimately computed it is bounded as follows,

$$\left|l_{ij}d_j^{\frac{1}{2}}\right| \leq \beta \quad, \quad i=j+1, \; j+2, \; ... \; ,n.$$

When the process is completed for all the columns of $\mathbf{H}_k$, the resulting matrices $\mathbf{L}_k$ and $\mathbf{D}_k$ are the factors of a positive definite matrix $\overline{\mathbf{H}}_k$ that is related to $\mathbf{H}_k$ in the following way

$$\overline{\mathbf{H}}_k = \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^{\mathrm{T}} = \mathbf{H}_k + \mathbf{E}_k \;, \qquad (3.37)$$

where $\mathbf{E}_k$ is a non-negative diagonal matrix whose *j-th* diagonal element is e_{jj}. Thus the positive definite matrix $\overline{\mathbf{H}}_k$ differs from $\mathbf{H}_k$ only in its diagonal elements.

To find the value of β, we observe that a lower bound should be imposed so that $\mathbf{H}_k$ is not modified unnecessarily, and an upper bound should also be imposed to preserve numerical stability and prevent excessively large elements in the factors. When $\mathbf{H}_k$ is positive definite, (3.27) shows that , for $j=1, 2, ..., n$ and each r $(r \leq j)$, it holds that $l_{jr}^2 d_r \leq h_{jj}$. Thus β should satisfy

$$\beta^2 \geq \gamma, \tag{3.38}$$

where γ is the largest in magnitude of the diagonal elements of $\mathbf{H}_k$, to ensure that $\mathbf{E}_k$ will be identically zero if $\mathbf{H}_k$ is sufficiently positive definite.

To impose an upper bound on β, Gill and Murray (1974) demonstrated that, for $n>1$,

$$\|\mathbf{E}_k\|_\infty \leq \left(\xi/\beta + (n-1)\beta\right)^2 + 2(\gamma + (n-1)\beta^2) + \delta = \zeta(\beta), \tag{3.39}$$

where the infinity norm of $\mathbf{E}_k$ is defined as the maximum absolute row sum, i.e.,

$$\|\mathbf{E}_k\|_\infty = \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^{n} |e_{ij}| \right\} ,$$

and ξ and γ are the largest in modulus of the off-diagonal and diagonal elements of $\mathbf{H}_k$. A reasonable value for β then is that which minimises $\zeta(\beta)$. This bound is minimised when

$$\beta^2 = \xi/\sqrt{n^2-1} \quad . \tag{3.40}$$

Thus from equations (3.39) and (3.40) the chosen value for β should be

$$\beta^2 = \max\left\{ \gamma, \xi/\sqrt{n^2-1}, \varepsilon_M \right\} \quad ,$$

where ε_M is the relative machine precision of the computer used and was included to allow for the case when $\|\mathbf{H}_k\|_\infty = 0$.

3.9. Line search by backtracking method

The aim of the line search is to find a step α_k which gives a significant reduction in function value along a descent, or search, direction $\mathbf{p}_k$. Because the search direction $\mathbf{p}_k$ and the starting point $\mathbf{x}_k$ are constant throughout the line search, the error function becomes a function of the single variable α_k; that is $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = F(\alpha_k)$. The function $F(\alpha_k)$ is assumed to possess a certain degree of smoothness, therefore we use a search method that exploits this smoothness (Luenberger, 1984). The search method should also make it possible to include the minimising value of α_k when $F(\alpha_k)$ is a quadratic with positive curvature.

One technique that satisfies the above requirements is based on curve fitting, in which we fit a smooth quadratic curve to the points $F(0)$ and $F(1)$ that are already available to us, in order to determine an estimate of the minimum point α_k. The only condition that should be imposed, in order to obtain a significant decrease in function value, is that

$$F(\mathbf{x}_k + \alpha_k) \le F(\mathbf{x}_k) + \rho \alpha \mathbf{g}_k^T \mathbf{p}_k \qquad (3.41)$$

for some value $\rho \in (0, \frac{1}{2})$ that, in practice, is taken to be very small so that a small decrease in function value is considered to be significant enough. Equation (3.41) states that the new point $\mathbf{x}_{k+1}$ should fall below the current point $\mathbf{x}_k$ and the line passing through $\mathbf{x}_k$ and having a slope that is a fraction ρ of the slope of the function at $\mathbf{x}_k$. If $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ has positive curvature at $\mathbf{x}_k$, then it can be proven that a step α_k that satisfies (2.1.1) exists (Dennis and Schnabel, 1983; Fletcher 1987).

The strategy for choosing α_k is to try the full Gauss-Newton step first, that is set $\alpha_k = 1$, then if $\mathbf{x}_k + \mathbf{p}_k$ is not acceptable, i.e., does not satisfy (3.41), we backtrack by reducing α_k until an acceptable $\alpha_k + \mathbf{p}_k$ is found (Dennis and Schnabel, 1983). Reducing α_k is restricted within upper and lower fractions of the previous unsatisfactory step. That is $\alpha_k \leftarrow \lambda \alpha_{prev}$ for some $\lambda \in [l, u]$, where $0 < l < u < 1$ so that we do not reduce the step either too little or too much (the values for l and u will be discussed later in this section). Defining

$$\hat{F}(\alpha) = F(\mathbf{x}_k + \alpha \mathbf{p}_k) \quad , \qquad (3.42)$$

as the one dimensional error energy function through $\mathbf{x}_k$ in the direction $\mathbf{p}_k$, if we need to backtrack, we use all the information available about $\hat{F}$ to make a quadratic approximation $\hat{F}(\alpha)$, then find $\hat{\alpha}$ that minimises $\hat{F}(\alpha)$ and take it as the next α_k to evaluate $\hat{F}(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$.

Initially, the information available to us about $\hat{F}(\alpha)$ is

$$\hat{F}(0) = F(\mathbf{x}_k) \quad , \tag{3.43}$$

and

$$\hat{F}^{`}(0) = \mathbf{g}_k^T \mathbf{p}_k \quad . \tag{3.44}$$

Then after taking the Gauss-Newton step we would have calculated

$$\hat{F}(1) = F(\mathbf{x}_k + \mathbf{p}_k) \quad . \tag{3.45}$$

Thus if $\hat{F}(1)$ does not satisfy (3.41); i.e., if $\hat{F}(1) > \hat{F}(0) + \rho \hat{F}^{`}(0)$, we make the one dimensional quadratic approximation

$$\hat{q}(\alpha) = \hat{F}(0) + \hat{F}^{`}(0)\alpha + \left[\hat{F}(1) - \hat{F}(0) - \hat{F}^{`}(0)\right]\alpha^2 \tag{3.46}$$

which satisfies (3.43), (3.44), and (34.5) and should have its minimum, for which $\hat{q}^{`}(\alpha) = 0$, at

$$\hat{\alpha} = \frac{-F^{`}(0)}{2\left[\hat{F}(1) - \hat{F}(0) - \hat{F}^{`}(0)\right]} \quad . \tag{3.47}$$

For $\hat{\alpha}$ to be a minimum we must have $\hat{q}^{``}(\alpha) > 0$, that is,

$$\hat{q}^{``}(\alpha) = 2\left[\hat{F}(1) - \hat{F}(0) - \hat{F}^{`}(1)\right] > 0.$$

This expression is satisfied at $\hat{\alpha}$, since $\hat{F}(1) > \hat{F}(0) + \rho \hat{F}^{`}(0) > \hat{F}(0) + \hat{F}^{`}(0)$. Also $\hat{\alpha} > 0$ because $\hat{F}^{`}(0)$ is negative. This minimum value $\hat{\alpha}$ is taken as the new value for α_k and then set $\alpha_k \leftarrow \lambda\alpha_{k-1}$ to evaluate $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ to see if we have obtained a reduction in the error energy function $F(\mathbf{x})$ satisfying (3.39); if not we need to backtrack again. To backtrack we set $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = \hat{F}(1)$ and calculate the new minimising $\hat{\alpha}$ according to (3.47), and continue the process until the required α_k is found.

There remains to find numeric values for the bounds l and u of $\lambda \in [l, u]$, where $0 < l < u < 1$. Note that since

$$\hat{F}(1) > \hat{F}(0) + \rho \hat{F}^{`}(0),$$

49

we have

$$\hat{\alpha} < \frac{1}{2(1-\lambda)} \ .$$

This shows that if $\hat{F}(1) \geq \hat{F}(0)$, then $\hat{\alpha} \leq \frac{1}{2}$, so that (3.47) gives the implicit upper bound of $u \approx \frac{1}{2}$ on the first value of λ. If, on the other hand, $\hat{F}(1)$ is much larger than $\hat{F}(0)$ then $\hat{\alpha}$ can be very small, in which case $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = F(x_k)$ and $\hat{F}(1) = \hat{F}(0)$. Thus we do not want to decrease α_k too much, since this is probably due to the poor approximation of $F(\alpha)$ by the quadratic model (3.46), so we impose a lower bound l, taken tentatively to be $\frac{1}{10}$. The bounds l and u mean that if $\hat{\alpha} \leq 0.1$ then we take $\alpha_k = 0.1$, and if $\hat{\alpha} \geq 0.5$ then we take $\alpha_k = 0.5$.

3.10. Descending in a direction of negative curvature

The point $\mathbf{x}_k$ where the gradient $\mathbf{g}_k$ is zero, or in practice $\|\mathbf{g}_k\|_2 \leq \varepsilon$, where ε is a small tolerance, has a descent direction $\mathbf{p}_k$ given by

$$\overline{\mathbf{H}}_k \mathbf{p}_k = -\mathbf{g}_k \ ,$$

that is obviously zero for a modified Hessian $\overline{\mathbf{H}}_k$. If the Hessian $\mathbf{H}_k$ is positive definite this indicates that the point $\mathbf{x}_k$ is a strong local minimum. If, however, $\mathbf{H}_k$ is indefinite then $\mathbf{x}_k$ is not a local minimum and an alternative descent direction should be defined.

The point $\mathbf{x}_k$ where $\mathbf{H}_k$ is indefinite and $\|\mathbf{g}_k\|_2 \leq \varepsilon$ is a saddle point, and the new descent direction to be taken is a direction of negative curvature. This direction $\mathbf{p}_k$ must exist if $\mathbf{H}_k$ is indefinite and is defined as

$$\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k < 0 \ . \tag{3.48}$$

To derive the negative curvature direction $\mathbf{p}_k$, we will use the Cholesky factors of $\overline{\mathbf{H}}_k$ and the diagonal matrix $\mathbf{E}_k$. The diagonal elements of $\mathbf{E}_k$ are non-zero if $\mathbf{H}_k$ is not sufficiently positive definite.

Gill and Murray (1974) showed that such a direction $\mathbf{p}_k$ is given by the system of linear equations

$$\mathbf{L}_k^T \mathbf{p}_k = \mathbf{e}_s \tag{3.49}$$

where $\mathbf{e}_s$ is a unit vector having its s-th element as unity, and s is the index at which the quantity $d_s - e_{ss}$ is least for $s=1, 2, ... ,n$.

If $\mathbf{H}_k$ is indefinite then from (3.27) we could have

$$d_s = h_{ss} + e_{ss} - \sum_{r=1}^{s-1} l_{sr}^2 d_r = \delta .$$ (3.50)

If in equation (3.50) $\delta=0$, then the quantity $d_s - e_{ss}$ is negative for an indefinite Hessian. Gill and Murray (1974) argued that it is safe to assume that this conclusion could still be obtained when δ is small.

For the solution $\mathbf{p}_k$ of equation (3.49) to be a direction of negative curvature it has to satisfy (3.48) (More and Sorenson 1979). From equation (3.37) we have

$$\mathbf{H}_k = \overline{\mathbf{H}}_k - \mathbf{E}_k .$$

Thus

$$\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k = \mathbf{p}_k^T \overline{\mathbf{H}}_k \mathbf{p}_k - \mathbf{p}_k^T \mathbf{E}_k \mathbf{p}_k .$$

Given the factors $\mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T$ of $\overline{\mathbf{H}}_k$ and equation (3.49), we have

$$\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k = \mathbf{e}_s^T \mathbf{L}_k^{-1} (\mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T) \mathbf{L}_k^{-T} \mathbf{e}_s - \mathbf{p}_k^T \mathbf{E}_k \mathbf{p}_k ,$$

$$\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k = \mathbf{e}_s^T \mathbf{D}_k \mathbf{e}_s - \mathbf{p}_k^T \mathbf{E}_k \mathbf{p}_k ,$$

so that,

$$\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k = d_s - \sum_{i=1}^{n} p_i^2 e_{ii} .$$ (3.51)

From equation (3.49) it follows that $p_i = 0$ for $i=s+1, s+2, ... ,n$ and $p_s = 1$, so that

$$\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k = d_s - e_{ss} - \sum_{i=1}^{s-1} p_i^2 e_{ii} .$$

Since from (3.50) we have concluded that

$$d_s - e_{ss} < 0,$$

then it follows that

$$\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k = d_s - e_{ss} - \sum_{i=1}^{s-1} p_i^2 e_{ii} < 0.$$

(3.52)

Since $e_{ii} \geq 0$ for $i=1, 2, \dots ,n$, then $\sum_{i=1}^{s-1} p_i^2 e_{ii} > 0$, giving $\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k < 0$. Thus $\mathbf{p}_k$ is a direction of negative curvature when $\mathbf{H}_k$ is indefinite.

CHAPTER

4

LEAST SQUARES INVERSION WITH
CONSTRAINTS

4.1. Introduction

The least squares seismic inversion problem could be thought of as an optimisation problem where the error energy function is minimised subject to various constraints. These constraints may be in the form of geologic information known *a priori*, or are used to make the inversion process numerically stable. The constraints can be equality and/or inequality, but in either case they are a linear combination of the forward model parameters.

Section 4.2 discusses how the linear constraints arise in the three different sets of parameters used in the least squares seismic inversion. In section 4.3 the linear constraints are subdivided into five types. Then for numerical stability reasons, when no values are assigned to the constraints, default values are assigned within the computer program. Section 4.4 describes the equality and inequality constraints on the boundary location problem, and hence on the time thickness of the lithic units in the input earth model. Section 4.5 discusses the constraints on the wavelet parameters, and section 4.6 discusses the constraints that can be imposed on the acoustic impedance parameters.

In section 4.7 the Gauss-Newton method subject to linear equality and inequality constraints is introduced. In this section, a binding direction of movement is defined as a move along an active constraint, and a non-binding direction of movement is defined as a feasible move off an active constraint, thus making an active

constraint inactive. It is also stated that the constrained descent direction is determined in the null space of the subspace defined by the set of active constraints. In section 4.8 the necessary and sufficient conditions of a minimum point subject to linear equality constraints are described, while section 4.9 discusses these optimality conditions subject to linear inequality constraints.

Section 4.10 describes a method that uses the **QR** factorisation to determine a basis for the null space of the subspace spanned by the active set of constraints; such a basis defines the direction of the constrained descent vector. Section 4.11 illustrates using the **QR** factors of section 4.10 to obtain a least squares estimates of the Lagrange multipliers. The Lagrange multipliers are used to decide if a non-binding direction exists. If such a direction exists, the corresponding inequality constraint has a negative multiplier, and a move off the constraint could be made thus it could be deleted from the active set. When more than one constraint has a negative Lagrange multiplier, we delete one constraint at a time and choose the one with the most negative multiplier. Section 4.12 combines all the previous theory so far into a method called the active set method, which is used in the algorithm for a computer program to implement the inversion.

4.2. The constraints problem

In some cases, the straightforward minimisation of the error energy function $F(\mathbf{x})$ will not give results that are geologically possible, or numerically stable. In other cases there may be geologic information known *a priori* that should be included in the final solution of the inversion. For example, when determining the boundary location problem we sometimes find a descent vector $\mathbf{p}_k$ that gives an updated vector $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ with a lithic unit having a greater travel-time at its lower boundary than the layer below, for any value of the step length α_k, or we might obtain a negative value for one or more of the layer boundaries. This of course should not be allowed since it is physically impossible. A similar situation might arise for the bounding frequency parameters of the wavelet and its calculated amplitude spectrum.

To solve this problem, we should always keep the boundary location vector elements and the bounding frequency elements of the wavelet parameters vector in ascending order and they should differ by specified minimum values. For the boundary location vector the difference should be at least a sample interval Δt because, in this case, to compute the Jacobian matrix $\mathbf{J}$, we use the forward difference formula to find a finite-difference approximation to the derivatives of the synthetic seismogram with respect to the boundary location parameters. The sample interval difference is necessary as a minimum because we need to perturb each boundary location by one sample interval to obtain a reasonable approximation of the

derivatives. In the case of the bounding frequencies of the wavelet parameters, the bounding frequency parameters are perturbed forward by an amount δf_i corresponding to frequency f_i which is i-th bounding frequency where $i=1, 2, 3, 4$. The perturbations δf_i depend on f_i, but they always obey the relationship $\delta f_i \leq \Delta f$, where Δf is the frequency increment $1/T$ and T is wavelet period, so that the bounding frequency parameters are always kept apart by an amount Δf. This was found to be a reasonable choice for the 128-sample ($T=256$ ms) wavelet that we use in this inversion process.

Finally, the starting acoustic impedance vector elements should also be constrained to fall above a specific value for these impedance parameters to have any geological significance, and avoid any numerical problems. If we allow some of the starting acoustic impedance vector elements to become negative or zero, we might obtain reflection coefficients that are very large, or numerically undefined, and which are completely unrepresentative of the area's lithology. Indeed, unless the starting acoustic impedance of each layer is above a certain value, the acoustic impedance log would not be representative of the lithology of the area, so each starting acoustic impedance for each lithic unit is constrained to be greater than or equal to a pre-defined value depending on the area of study.

In addition to those constraints on the final solution that we must include in the inversion problem, there are other desirable constraints that we would like to include. Such constraints are usually used to confine the set of all iterative solutions { x_k } to be constrained within a region that is geologically feasible, and also to overcome, to some degree, the problem of non-uniqueness that is inherent in the inversion process. These constraints are generated from the geological information obtained from interpreting the well-log and the seismic data. They represent the limits to how far an initial geological model, represented as an acoustic impedance model, can be perturbed by the inversion process to reach the final solution. In other words, these constraints represent geological information known to us that the final solution should include. For example, it is probably known that a certain layer has a constant time thickness, or that it should not be less than a known time thickness. In other cases, we may require that the phase spectrum of the seismic wavelet be kept the same throughout the area of study. One important constraint that might often be used is to set some acoustic impedance gradient vector elements to a constant positive value where there is known to be a fining upward sequence within the input geologic model or, in some cases, we might want to set all the impedance gradient elements to zero throughout the inversion operation because all the layers in the geologic model are lithologically uniform. This will result in each of the lithic units having a constant acoustic impedance equal to its starting acoustic impedance.

4.3. Types of constraints and default bounds

All the constraints discussed above are examples of linear constraints. They are linear functions of the variables, say the elements of a vector $\mathbf{x}$, that have the general form $l(\mathbf{x}) = \mathbf{a}^T\mathbf{x} - \beta$, for some row vector $\mathbf{a}^T$ and a scalar β. The linear function $l(\mathbf{x})$ is specified to be exactly zero, non-negative, or non-positive. Thus we have two types of linear constraints to be considered (Gill *et al.*, 1981):

1. equality constraint having the form of $\mathbf{a}^T\mathbf{x} - \beta = 0$, and is written as $\mathbf{a}^T\mathbf{x} = \beta$;

2. inequality constraint having the form $\mathbf{a}^T\mathbf{x} - \beta \geq 0$, and it is written as $\mathbf{a}^T\mathbf{x} \geq \beta$.

Constraints of the form $\mathbf{a}^T\mathbf{x} - \beta \leq 0$ are equivalently stated as $-\mathbf{a}^T\mathbf{x} \geq -\beta$. A simple form of linear constraints occurs when the function $l(\mathbf{x})$ involves only one variable, that is one element of the vector $\mathbf{x}$, say x_i, then other possible constraint forms are:

3. x_i is fixed at β, so $x_i = \beta$;

4. x_i has a lower bound β, so $x_i \geq \beta$;

5. x_i has an upper bound β, so $x_i \leq \beta$.

The constraint forms 4 and 5 are called simple bounds on x_i.

No parameter in this inversion process is allowed to overflow. Thus a large enough default constant has been chosen to indicate that one, or more, of the parameters are becoming unacceptably large. The value of this constant is taken to be 10^{21}; therefore, any parameter that exceeds this constant will indicate a numerical error.

4.4. Constraints on the boundary location variables

The constraint forms 1 and 2 will be used in the boundary location problem to constrain the two-way travel-time thickness of any layer between boundary location variables t_i and t_{i+1}, where $i=1, 2, \ldots, n-1$ is the number of interfaces between the n lithic units. Thus, if we require an equality constraint on the i-th layer, then we have

$$-t_i + t_{i+1} = \beta_i, \tag{4.1}$$

or if we require an inequality constraint the we obviously have

$$-t_i + t_{i+1} \geq \beta_i. \tag{4.2}$$

Clearly the row vector here is $\mathbf{a}^T = (-1, 1)$.

It was mentioned in the previous section that for reasons of finding an approximation to the derivatives of the Jacobian matrix $\mathbf{J}_k$, each lithic unit should be at least one sample thick in terms of two-way travel time. This translates to having an

inequality constraint for each lithic unit where each $\beta_i = \Delta t$. Thus if we do not have any other equality or inequality constraint of the forms of equations (4.1) or (4.2), an inequality constraint of the form (4.2) with $\beta_i = \Delta t$ should always be imposed. For example, suppose we have a five layer model, then we have a boundary location vector with four variable elements, namely t_1, t_2, t_3, and t_4, and they represent the interfaces between the lithic units. The two way travel-time to the base of the lower unit, t_5, should always stay constant. Then if we do not have any equality or inequality constraint on the layers time thickness, we should honour the following constraints throughout the inversion process:

$$-t_1 + t_2 \geq \Delta t$$
$$-t_2 + t_3 \geq \Delta t$$
$$-t_3 + t_4 \geq \Delta t$$
$$-t_4 \geq \Delta t - t_5$$

Put into a matrix form, they become

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} \geq \begin{bmatrix} \Delta t \\ \Delta t \\ \Delta t \\ \Delta t - t_5 \end{bmatrix}$$

which is equivalent to

$$\mathbf{At} \geq \mathbf{b} \qquad\qquad (4.3)$$

If, on the other hand, the first two layers have to be 20 and 30 ms (two-way time) thick each, then we have the equality constraints

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} = \begin{bmatrix} 20 \\ 30 \end{bmatrix}$$

or

$$\mathbf{\hat{A}t} = \mathbf{\hat{b}} \ , \qquad\qquad (4.4)$$

and the inequality constraints

$$\begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} = \begin{bmatrix} \Delta t \\ \Delta t - t_5 \end{bmatrix}$$

or

$$\mathbf{A} \mathbf{t} \ge \mathbf{b} \,. \tag{4.5}$$

4.5. Constraints on the wavelet parameters

All the five forms of constraints discussed in section 4.2 are used for the wavelet parameters vector. The four bounding frequency parameters, $f_1, f_2, f_3,$ and f_4, are allowed to take any value within the interval $[\Delta f, f_N - 2\Delta f]$. The only requirement is that the closest two successive frequency parameters can be is Δf H_z. The frequency parameters of the wavelet can only be inequality constrained. The constraints for the amplitude and the phase parameters of the wavelet involve only one variable each; they are represented by forms 3, 4, and 5. Unless otherwise specified, the two amplitude parameters can vary in the interval $[0, 10^{21}]$ and the three phase parameters can vary only in the interval $[-10^{21}, 10^{21}]$. Thus the wavelet solution should at least satisfy the following set of inequality constraints:

$$\left. \begin{array}{rcl} w_1 & \ge & \Delta f \\ -w_1 + w_2 & \ge & \Delta f \\ -w_2 + w_3 & \ge & \Delta f \\ -w_3 + w_4 & \ge & \Delta f \\ -w_4 & \ge & -f_N + 2\Delta f \\ w_5 & \ge & 0 \\ -w_5 & \ge & -c \\ w_6 & \ge & 0 \\ -w_6 & \ge & -c \\ w_7 & \ge & -c \\ -w_7 & \ge & -c \\ w_8 & \ge & -c \\ -w_8 & \ge & -c \\ w_9 & \ge & -c \\ -w_9 & \ge & -c \end{array} \right\} , \tag{4.6}$$

where w_1 through w_9 are the nine parameters of the wavelet, f_N is the Nyquist frequency, Δf is the frequency increment, and the constant $c = 10^{21}$. Clearly, the first five constraints represent the inequality constraints on the frequency parameters of the

wavelet; then each successive two constraints represent the lower and upper bounds on the amplitude and phase parameters of the wavelet. Obviously, by changing the right-hand side of any constraints 6 through 15 of the set (4.6) we can change the lower or upper bounds of any of the amplitude and phase parameters.

4.6. Constraints on the acoustic impedance parameters

The acoustic impedance parameters include both the starting acoustic impedance parameters and the acoustic impedance gradient parameters. A positive lower bound that depends on the geology of the area should be imposed on all the starting acoustic impedance parameters. An upper bound could also be globally imposed but the default is 10^{21} gcm^{-3}ms^{-1}.

On the other hand, it is possible that the acoustic impedance gradient parameters would not be constrained so that it has an upper bound of 10^{21} gcm^{-3}ms^{-1}/sample interval, and a lower bound of -10^{21} gcm^{-3}ms^{-1}/sample interval. However, it is sometimes desirable to constrain all the acoustic impedance gradient parameters to be zero, and thus they are included in the set of equality constraints instead.

4.7. Linearly constrained Gauss-Newton method

Before proceeding to discuss using the Gauss-Newton method to minimise $F(\mathbf{x})$ subject to linear equality and inequality constraints on the variables vector $\mathbf{x}$, we first consider the optimality conditions for the minimisation of $F(\mathbf{x})$ subject only to (1) linear equality constraints, and (2) linear inequality constraints. For optimality of either minimisation problem, we need only to consider the points $\mathbf{x}$ which are *feasible*, that is which satisfy all the constraints $\hat{\mathbf{A}}\mathbf{x} = \hat{\mathbf{b}}$ in the equality case, and $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ in the inequality case.

When minimising subject to the equality constraints

$$\hat{\mathbf{A}}\mathbf{x} = \hat{\mathbf{b}} \quad , \qquad\qquad\qquad (4.7)$$

the matrix $\hat{\mathbf{A}}$ has m rows corresponding to the number of equality constraints, and n columns corresponding to the number of variables, or parameters, in the problem. The i-th row of $\hat{\mathbf{A}}$ is denoted by $\hat{\mathbf{a}}_i^T$, and its elements are the coefficients of the i-th constraint:

$$\hat{\mathbf{a}}_i^T\mathbf{x} = \hat{a}_{i1}x_1 + \hat{a}_{i2}x_2 + \ldots + \hat{a}_{in}x_n \quad .$$

59

In the equality constraints (4.7) a feasible point will exist only if $\hat{\mathbf{b}}$ is in the column space of $\hat{\mathbf{A}}$ (Strang, 1988), which means that the constraints have to be consistent.

When minimising subject to the inequality constraints

$$\mathbf{A}\mathbf{x} \geq \mathbf{b} \ , \qquad\qquad (4.8)$$

the i-th constraint $\mathbf{a}_i^T \mathbf{x} \geq \mathbf{b}_i$ is active, or binding, at the feasible point $\hat{\mathbf{x}}$ if $\mathbf{a}_i^T \hat{\mathbf{x}} = \hat{\mathbf{b}}_i$, and inactive if $\mathbf{a}_i^T \hat{\mathbf{x}} > \hat{\mathbf{b}}_i$. A constraint is violated when $\mathbf{a}_i^T \overline{\mathbf{x}} < \hat{\mathbf{b}}_i$ at the point $\overline{\mathbf{x}}$, which of course is not feasible. If the i-th constraint is active at the point $\hat{\mathbf{x}}$, then there are two possible feasible directions of movement $\mathbf{p}$. The first is when $\mathbf{p}$ satisfies

$$\mathbf{a}_i^T \mathbf{p} = 0 \quad ;$$

then $\mathbf{p}$ is a binding direction of movement with respect to the i-th constraint, because the constraint remains active at all points $\hat{\mathbf{x}} + \alpha \mathbf{p}$ for any α, and the new point $\hat{\mathbf{x}} + \alpha \mathbf{p}$ remains on the constraint. The second direction of movement is when $\mathbf{p}$ satisfies

$$\mathbf{a}_i^T \mathbf{p} > 0 \quad ;$$

then $\mathbf{p}$ is non-binding direction of movement with respect to the i-th constraint, because $\mathbf{a}_i^T (\hat{\mathbf{x}} + \alpha \mathbf{p}) = \mathbf{b}_i + \alpha \mathbf{a}_i^T \mathbf{p} > \mathbf{b}_i$ for any $\alpha > 0$. The i-th constraint becomes inactive at the new point $\hat{\mathbf{x}} + \alpha \mathbf{p}$ which now is moved off the constraint.

4.8. Conditions for a minimum subject to linear equality constraints

We now consider the optimality conditions for the problem

$$\left. \begin{array}{l} \text{minimise} \quad F(\mathbf{x}) \\ \text{subject to} \quad \hat{\mathbf{A}}\mathbf{x} = \hat{\mathbf{b}} \end{array} \right\} \qquad\qquad (4.9)$$

where $\hat{\mathbf{A}}$ is $m \times n$ matrix with the i-th row corresponding to the i-th equality constraint, and the rows of $\hat{\mathbf{A}}$ are linearly independent. Consider the step between any two feasible points $\overline{\mathbf{x}}$ and $\hat{\mathbf{x}}$. Since (4.7) applies, then by linearity

$$\hat{\mathbf{A}}\overline{\mathbf{x}} - \hat{\mathbf{A}}\hat{\mathbf{x}} = \hat{\mathbf{A}}(\overline{\mathbf{x}} - \hat{\mathbf{x}}) = \hat{\mathbf{b}} - \hat{\mathbf{b}} = \mathbf{0} \quad .$$

Thus any step $\mathbf{p}$ between two feasible points must satisfy

$$\hat{\mathbf{A}}\mathbf{p} = \mathbf{0} \ , \qquad\qquad (4.10)$$

which states that $\mathbf{p}$ is orthogonal to the columns of $\hat{\mathbf{A}}$. Such a vector $\mathbf{p}$ is a feasible direction with respect to the set of equality constraints (4.7).

The set of vectors that satisfy (4.7) lie in a subspace for which a basis must exist, and every feasible direction $\mathbf{p}$ can be written as a linear combination of the basis vectors that span this subspace (Gilbert, 1988). If we let the basis vectors be the columns of a matrix $\mathbf{Z}$, then

$$\hat{\mathbf{A}}\mathbf{Z} = \mathbf{0} \ ,$$

and every feasible vector $\mathbf{p}$ satisfying (4.10) can be written as

$$\mathbf{p} = \mathbf{Z}\mathbf{p}_z \ , \tag{4.11}$$

for some vector $\mathbf{p}_z$ called the projected feasible (descent) direction.

Now, if we write the Taylor series expansion of $F(\mathbf{x}^*)$, about the minimum point $\mathbf{x}^*$, as given by equation (3.14) in chapter 3, in terms of $\mathbf{p} = \mathbf{Z}\mathbf{p}_z$, we get

$$F(\mathbf{x}^* + \mathbf{Z}\mathbf{p}_z) = F(\mathbf{x}^*) + \mathbf{p}_z^T \mathbf{Z}\mathbf{g}(\mathbf{x}^*) + \tfrac{1}{2}\mathbf{p}_z^T \mathbf{Z}^T \mathbf{H}(\mathbf{x}^*)\mathbf{Z}\mathbf{p}_z \ . \tag{4.12}$$

Using a similar argument to that in the unconstrained case, in chapter 3, we can conclude that a necessary condition for $\mathbf{x}^*$ to be a local minimum of (4.9) is that

$$\mathbf{p}_z^T \mathbf{Z}^T \mathbf{g}(\mathbf{x}^*) = 0 \ ,$$

for every $\mathbf{p}_z$; thus

$$\mathbf{Z}^T \mathbf{g}(\mathbf{x}^*) = \mathbf{0} \ . \tag{4.13}$$

This is the necessary first order condition for $\mathbf{x}^*$ to be a local minimum. The vector $\mathbf{Z}^T \mathbf{g}(\mathbf{x}^*)$ is called the projected gradient at $\mathbf{x}^*$. The point $\mathbf{x}^*$ at which the projected gradient vanishes is called a constrained stationary point.

Equation (4.13) states that $\mathbf{g}(\mathbf{x}^*)$ is in the null space of the matrix $\mathbf{Z}^T$. This implies that $\mathbf{g}(\mathbf{x}^*)$ lies in the row space of $\hat{\mathbf{A}}$ (Gilbert, 1988), that is

$$\mathbf{g}(\mathbf{x}^*) = \sum_{i=1}^{m} \hat{\mathbf{a}}_i \lambda_i^* = \hat{\mathbf{A}}\lambda^* \ . \tag{4.14}$$

The vector λ^* is the vector of Lagrange multipliers (Gill *et al.*, 1981).

To derive the second order conditions for x^* to be a local minimum, we substitute (4.13) in the Taylor series expansion (4.12)

$$F(x^* + Zp_z) = F(x^*) + \tfrac{1}{2} p_z Z^T H(x^*) Zp_z \quad . \tag{4.15}$$

Similar to the unconstrained case, (4.15) indicates that the sufficient condition is that the matrix $Z^T H(x^*) Z$ be positive definite (not $H(x^*)$), but the necessary condition is that $Z^T H(x^*) Z$ be positive semi-definite (Scales, 1985).

4.9. Conditions for a minimum subject to linear inequality constraints

We now consider the optimality conditions for the problem:

$$\left. \begin{array}{l} \text{minimise} \;\; F(x) \\ \text{subject to} \;\; Ax \geq b \end{array} \right\} \tag{4.16}$$

where A is $m \times n$ matrix with the i-th row corresponds to the i-th inequality constraint, and the rows of A are linearly independent.

We derive the optimality conditions for the feasible point x^* in a manner similar to that for the equality constrained problem. Suppose that there are s rows of the matrix A contain the coefficients of the constraints that are active at x^*. If we let these s rows form the rows of a matrix $\hat{A}$, we then have $\hat{A} x^* = \hat{b}$. Since the rows of A are linearly independent, so are the rows of $\hat{A}$. Let Z be a matrix whose columns form a basis for the set of vectors orthogonal to the rows of $\hat{A}$. Then every vector p satisfying $\hat{A} p = 0$ can be written as a linear combination of the columns of Z.

The Taylor series expansion of F about x^* along a binding direction $p = Zp_z$ is given by

$$F(x^* + Zp_z) = F(x^*) + p_z^T Z g(x^*) + \tfrac{1}{2} p_z^T Z^T H(x^*) Zp_z \quad ,$$

which states that the first order necessary condition for x^* to be a local minimum is that $Z^T g(x^*) = 0$, which is equivalent to

$$g(x^*) = \hat{A}^T \lambda^* \quad . \tag{14.7}$$

Note that the Lagrange multipliers, λ_i, correspond only to the s active constraints.

The first order condition (14.7) guarantees that x^* is a stationary point along all binding directions p. However, in this case, where the active constraints are

inequality constraints, we have the added complication that other non-binding feasible directions could exist, and moving along such directions is a move off one, or more, constraints (Scales, 1985).

Now suppose that the point $\mathbf{x}^*$ satisfies the active constraints as equalities, then we have

$$\hat{\mathbf{A}}\mathbf{x}^* = \hat{\mathbf{b}} \cdot \qquad (4.18)$$

If $\mathbf{p}$ is a feasible non-binding descent direction then

$$\hat{\mathbf{A}}(\mathbf{x}^* + \mathbf{p}) \geq \hat{\mathbf{b}} \quad . \qquad (4.19)$$

Subtracting (4.18) we obtain

$$\hat{\mathbf{A}}\mathbf{p} \geq \mathbf{0} \quad , \qquad (4.19)$$

which shows that when the i-th constraint becomes inactive at the point $\mathbf{x}^* + \mathbf{p}$, we have

$$\mathbf{a}_i^T \mathbf{p} > 0 \quad . \qquad (4.21)$$

Also, because $\mathbf{p}$ is a descent direction then ,

$$\mathbf{g}^T(\mathbf{x}^*)\mathbf{p} < 0 \quad . \qquad (4.22)$$

From (5.17) we have,

$$\mathbf{g}^T(\mathbf{x}^*)\mathbf{p} = \lambda^{*T}\hat{\mathbf{A}}\mathbf{p} \quad ,$$

since $\mathbf{p}$ is non-binding for the i-th active constraint, then

$$\mathbf{g}(\mathbf{x}^*)^T \mathbf{p} = \lambda_i^* \mathbf{a}_i^T \mathbf{p} \quad . \qquad (4.23)$$

From (4.21) and (4.22) we conclude that if $\lambda_i < 0$ it is possible to make a descent move off the i-th constraint an consequently $\mathbf{x}^*$ is not a minimum. Clearly it is necessary to add the condition $\lambda_i \geq 0$ to the necessary condition (4.22) for $\mathbf{x}^*$ to be a local minimum.

The second order necessary condition for inequality constraints is similar to that for the equality constraints if all $\lambda_i \geq 0$, for $i = 1, 2, \ldots , s$ of the active inequality

constraints, since we can regard these constraints as equality. In this case the sufficient condition is also the same, that is $\mathbf{Z}^T \mathbf{H}(\mathbf{x}^*)\mathbf{Z}$ is positive definite (Gill *et al.*, 1981).

If , on the other hand, the Lagrange multipliers for the *i*-th active inequality constraint is zero, $\lambda_i = 0$, then equation (4.23) gives $\mathbf{g}^T(\mathbf{x}^*)\mathbf{p} = 0$. This is neither an ascent nor a descent direction. Here we consider only the active inequality constraints with $\lambda_i > 0$ (Scales, 1985). If there are q such constraints with a corresponding coefficients matrix $\hat{\mathbf{A}}_q$, and $\mathbf{Z}_q$ defined as $\hat{\mathbf{A}}_q \mathbf{Z}_q = \mathbf{0}$, then the second order necessary condition is $\mathbf{Z}_q^T \mathbf{H}(\mathbf{x}^*)\mathbf{Z}_q$ is positive semi-definite, and the second order sufficient condition is $\mathbf{Z}_q^T \mathbf{H}(\mathbf{x}^*)\mathbf{Z}_q$ is positive definite.

4.10. Determining the null space matrix Z

There are several methods that can be used to define the null space basis matrix $\mathbf{Z}$, but the method based on the **QR** factorisation of the equality constraints matrix $\hat{\mathbf{A}}$ has many advantages due to the orthogonal matrix $\mathbf{Q}$. For example, $\mathbf{Q}^T \mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ so that $\mathbf{Q}^T$ is the inverse of $\mathbf{Q}$, and that the orthogonal transformation $\mathbf{Q}\mathbf{x}$ preserves the Euclidean length of the vector $\mathbf{x}$, and the partitioning of $\mathbf{Q}$ into submatrices, say $\mathbf{Q}_1$ and $\mathbf{Q}_2$, results in the submatrices being orthogonal as well. Furthermore, the matrices $\mathbf{Q}$ and $\mathbf{R}$ obtained can readily be used to find estimates of the Lagrange multipliers at a possible minimum point $\mathbf{x}^*$.

We start with the full column rank matrix $\hat{\mathbf{A}}^T$ and find the factorisation

$$\mathbf{Q}^T \hat{\mathbf{A}}^T = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} , \qquad (4.24)$$

where $\mathbf{Q}$ is *nxn* orthogonal matrix, $\mathbf{R}$ is *mxm* non-singular upper-triangular matrix, and $\mathbf{0}$ is (*n-m*)x*m* null matrix. We then form, by taking the transpose of both sides of (4.24),

$$\hat{\mathbf{A}}\mathbf{Q} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \end{bmatrix} , \qquad (4.25)$$

where $\mathbf{L} = \mathbf{R}^T$, and the partition $\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix}$ such that $\mathbf{Q}_1$ is *nxm* and $\mathbf{Q}_2$ is *nx*(*n-m*) are two orthogonal matrices, thus from (4.25) we can obtain

$$\hat{\mathbf{A}}\begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \end{bmatrix} , \qquad (4.26)$$

so that

$$\hat{\mathbf{A}}\mathbf{Q}_1 = \mathbf{L} , \qquad (4.27)$$

and

$$\hat{A}Q_2 = 0 \ . \tag{4.28}$$

An obvious candidate for the null space basis matrix $\mathbb{Z}$ is Q_2. The columns of Q_2 will form the orthonormal basis of the null space of, the full row rank matrix, $\hat{A}$.

4.11. Estimating Lagrange Multipliers

It is of interest to compute the vector λ^* of Lagrange multipliers at the solution x^* to the linear equality constraints problem (4.9). This information is required by the active set method, which will be discussed in the next section, to make a decision about which of the active inequality constraints to be deleted from the matrix $\hat{A}$ (Fletcher, 1987). The vector λ^* is defined by (4.17):

$$g(x^*) = \hat{A}^T \lambda^*$$

where λ^* can be computed because the linear system (4.17) is consistent. However, Lagrange multipliers are not defined at a non-stationary point, and (4.17) is not generally consistent at such a point (Gill and Murray, 1979). Even if (4.17) were consistent, the computation of the multipliers using finite precision arithmetic causes computational error, so no exact values can be obtained. Nonetheless, it is essential to estimate λ^* at points for which (4.17) does not hold. Thus we compute a Lagrange multipliers vector estimate λ_k at the iterate x_k, such that λ_k has the property that (Gill *et al.*, 1981)

$$x_k \rightarrow x^* \quad \text{implies that} \quad \lambda_k \rightarrow \lambda^* \ .$$

The **QR** factors of $\hat{A}$ can be used to compute a least squares estimate of the Lagrange multipliers at any point x_k at which the gradient vector is g_k. Then we wish to find an estimate λ_k such that (Gill and Murray, 1979):

$$\left\| \hat{A}^T \lambda_k - g_k \right\|_2^2 \ , \tag{4.29}$$

is minimum, where the Lagrange multipliers vector λ_k is of length m corresponding to the m active constraints in $\hat{A}$. Because the Euclidean length is preserved by orthogonal transformations, the Euclidean length of the residual (4.29) transformed by Q^T stays the same; thus

$$\left\|\hat{A}^T\lambda_k - g_k\right\|_2^2 = \left\|Q^T(\hat{A}^T\lambda_k - g_k)\right\|_2^2 = \left\|Q^T\hat{A}^T\lambda_k - Q^Tg_k\right\|_2^2 .$$

Partitioning Q^T such that $Q^T = \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}$, where Q_1^T is mxn and Q_2^T is $(n\text{-}m)$xn, then substituting for Q^T and $Q^T\hat{A}^T$, then we have for (4.29)

$$= \left\|\begin{bmatrix} R \\ 0 \end{bmatrix}\lambda_k - \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}g_k\right\|_2^2 ,$$

thus we see that the residual vector will be minimised when

$$R\lambda_k - Q_1^Tg_k = 0$$

or

$$R\lambda_k = Q_1^Tg_k, \tag{4.30}$$

Because R is upper triangular, we use back-substitution to compute the least squares estimate of the Lagrange multipliers λ_k of (4.17).

CHAPTER

5

IMPLEMENTATION OF LEAST SQUARES
INVERSION

5.1. Introduction

The objectives of this chapter are to develop and test a strategy for solving the seismic inversion problem with linear equality and inequality constraints. A computer program has been written to carry out the inversion using an algorithm based on the active set method. The inversion program is tested on some synthetic seismic examples. Linear equality and inequality constraints are used to steer the inversion process into a region that contains geologically feasible solutions, thus reducing the problem of non-uniqueness, and reducing the risk of convergence to minima other than the global one.

Section 5.2 discusses the active set method. We start by determining the active set of constraints, then at each iteration we first determine if any of the inactive constraints could become active. The best candidate is the nearest inactive constraint in the direction of descent, so this section also describes how to calculate the step length to such a constraint. Section 5.3 describes in some detail the steps taken to implement, into a computer program, the least squares inversion of seismic data, with linear equality and inequality constraints using the active set method. In section 5.4 we discuss the region of convergence for the initial guess, and give as measure of the quality of convergence the error energy as a percentage of the energy in the observed trace. Synthetic examples showed that, when solving for the boundary location problem, the region of convergence is determined by the central lobe width of a zero

phase seismic wavelet. When solving for the acoustic impedance problem, convergence is almost always achieved, but not necessarily to the correct solution; i.e., we might obtain a non-unique solution, so this section also includes a discussion of the uniqueness of the inversion solution. In section 5.5 several synthetic examples are given to illustrate inversion for boundary locations, impedance values, boundaries-impedances combined, wavelet parameters, and boundaries-impedances-wavelet parameters combined. In section 5.6 some concluding remarks are given which should help in developing a strategy for the real seismic data inversion in the next chapter.

5.2. The active set method

This method uses the technique discussed in section 4.7 for solving an equality constrained problem to solve a general problem that contains both equality and inequality constraints. Thus the problem we need to solve is the following:

$$\left.\begin{array}{l} \text{minimise} \ \ F(\mathbf{x}) \\ \text{subject to} \ \overline{\mathbf{A}}\mathbf{x} = \overline{\mathbf{b}} \\ \qquad\qquad \mathbf{A}\mathbf{x} \geq \mathbf{b} \end{array}\right\} \qquad\qquad (5.1)$$

where the matrix $\overline{\mathbf{A}}$ contains the coefficients of the equality constraints, the vector $\overline{\mathbf{b}}$ contains the equality values, the matrix $\mathbf{A}$ contains the coefficients of the inequality constraints, and the vector $\mathbf{b}$ contains the inequality bounds. The optimality conditions for equality and inequality constraints have been discussed in sections (4.7) and (4.8), where we recall that only the constraints active at the current point $\mathbf{x}_k$ are significant. In this section we will assume that the projected Hessian $\mathbf{Z}_k^T\mathbf{H}_k\mathbf{Z}_k$ is always positive definite so that a descent direction $\mathbf{p}_k = \mathbf{Z}_k\mathbf{p}_z$ can be found using the Gauss-Newton step

$$\mathbf{Z}_k^T\mathbf{H}_k\mathbf{Z}_k\mathbf{p}_z = -\mathbf{Z}_k^T\mathbf{g}_z \qquad\qquad (5.2)$$

which is equation (3.17) of chapter 3 in the null space of $\hat{\mathbf{A}}_k$ defined by $\mathbf{Z}_k$ (when the projected Hessian is not positive definite then it should be modified as discussed in section 3.8).

The first step in the active set method is to construct the active set of constraints. An obvious candidate for this set is the set of equality constraints. This set is always included in the active set. Next we include those inequality constraints that are active at the current point $\mathbf{x}_k$. Suppose that we have s equality constraints and r inequality constraints, and suppose further that q of the inequality constraints are

active at $\mathbf{x}_k$. Let us denote by $\hat{\mathbf{A}}_k$ the $m{\times}n$ matrix, where $1 \leq m \leq s{+}q$, whose i-th row contains the coefficients of the i-th equality constraint for $1 \leq i \leq s$, and the coefficients of the $(i{-}s)$-th inequality constraint for $s{+}1 \leq i \leq s{+}q$. The rest of the inequality constraints that are not active are temporarily disregarded. The problem then becomes

$$\left.\begin{array}{l} \text{minimise } F(\mathbf{x}) \\ \text{subject to } \hat{\mathbf{A}}_k\mathbf{x}_k = \hat{\mathbf{b}}_k \end{array}\right\} \qquad (5.3)$$

which of course is the equality constrained problem of section 4.7, and for which a null space basis matrix could be computed as described in section 4.10.

When using the active set method to solve problem (5.1), there will always be some inequality constraints that are not active and so not included in the active set. The linear search step α_k obtained by the backtrack method (see section 3.9) must always take into account the possibility of violating one, or perhaps more, of the inequality constraints that are not active at the current point $\mathbf{x}_k$. Therefore, it is important to know beforehand the step length $\overline{\alpha}$ to the nearest inactive constraint. To find $\overline{\alpha}$ we compute all the step lengths σ_i from the current point $\mathbf{x}_k$ along the current search (descent) direction $\mathbf{p}_k$ to each of the inactive inequality constraint $\mathbf{a}_i^T$ and take the smallest as $\overline{\alpha}$.

To find σ_i, suppose for some $\mathbf{p}_k$ that $\mathbf{a}_i^T\mathbf{p}_k \geq 0$, then any positive move along $\mathbf{p}_k$ will not violate the constraint $\mathbf{a}_i^T$. That is, if $\mathbf{a}_i^T\mathbf{p}_k$ is non-negative for all such constraints then they impose no restriction on the step length σ_i. However, if $\mathbf{a}_i^T\mathbf{p}_k < 0$, then the constraint $\mathbf{a}_i^T$ becomes active at the critical step σ_i satisfying

$$\mathbf{a}_i^T(\mathbf{x}_k - \sigma_i\mathbf{p}_k) = \mathbf{b}_i \quad ,$$

which implies that

$$\sigma_i = \frac{\mathbf{b}_i - \mathbf{a}_i^T\mathbf{x}_k}{\mathbf{a}_i^T\mathbf{p}_k} \quad , \qquad (5.4)$$

where i is an index describing the inequality constraints not included in the active set. The step length $\overline{\alpha}$ is then taken as

$$\overline{\alpha} = \begin{cases} \min\{\sigma_i\}, & \text{if } \mathbf{a}_i^T\mathbf{p}_k < 0 \text{ for some } i\text{-th constraint not in the active set} \\ +\infty & , \text{ if } \mathbf{a}_i^T\mathbf{p}_k \geq 0 \text{ for all the constraint not in the active set} \end{cases} \quad , \qquad (5.5)$$

and it is the maximum non-negative feasible step that can be taken along $\mathbf{p}_k$. Thus it represents an upper bound on the line search step length α_k.

The following steps summarise the active set strategy:

1. Start at the current feasible point $\mathbf{x}_k$ with $\hat{\mathbf{A}}_k$, $\hat{\mathbf{b}}_k$, and $\mathbf{Z}_k$.

2. Test for convergence. If $\mathbf{x}_k$ is a minimum in the subspace defined by $\mathbf{Z}_k$, then we consider deleting a constraint from the active set. The decision to delete a constraint is based on the sign of the computed estimates of the Lagrange multipliers of the active set. The best constraint to delete is the active inequality constraint with the most negative multiplier. When all the multipliers are non-negative ($\lambda_i \geq 0$), or when none of the inequality constraints are included in the active set, then clearly convergence has been achieved and the inversion process terminates with the solution $\mathbf{x}^* = \mathbf{x}_k$. If a constraint is deleted from the active set, the matrices $\hat{\mathbf{A}}_k$, $\hat{\mathbf{b}}_k$, and $\mathbf{Z}_k$ are updated accordingly.

3. Solve (5.2) then find a descent direction $\mathbf{p}_k = \mathbf{Z}_k \mathbf{p}_z$.

4. Find the step length $\overline{\alpha}$ to the nearest inactive inequality constraint using (5.4) and (5.5).

5. Decide on the step length α_k, taken in the direction $\mathbf{p}_k$, that gives a reduction in error energy. If $\alpha_k = \overline{\alpha}$, then add the constraint that corresponds to $\overline{\alpha}$ to the active set, and update $\hat{\mathbf{A}}_k$, $\hat{\mathbf{b}}_k$, and $\mathbf{Z}_k$.

6. Compute $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$, set $k \leftarrow k+1$, and repeat the process at step 1.

5.3. The steps of the inversion algorithm

The inversion algorithm, using the active set strategy as implemented here, can be broadly divided into four basic steps: (1) handling of the active and inactive constraints, (2) determining the descent vector $\mathbf{p}_k$, (3) determining the step length α_k and (4) testing for convergence to the solution.

The set of all possible linear constraints that are applicable to the problems at hand, should always be determined first, before going into any other inversion step. This is equivalent to stating the region of feasible geologic solutions that we are seeking, so avoiding any other solutions that are mathematically possible but geologically not feasible. This, of course, requires a prior knowledge of the dominant lithological trends of the area in study, a requirement that is normally satisfied since we already have at least one well location at which a seismic wavelet was estimated, as discussed in chapter 2. For every problem to be solved, e.g. boundary locations, acoustic impedances or the wavelet at a well location, we always construct two matrices for the constraints coefficients, a matrix $\hat{\mathbf{A}}$ that contains coefficients of the equality constraints and the active inequality constraints, and a matrix $\mathbf{A}$ that contains

the coefficients of all possible inequality constraints; that is all active and inactive inequality constraints.

There is a lot of housekeeping when using constraints in an inversion program. Any inactive constraint that becomes active at any current point $\mathbf{x}_k$ should be added to the matrix $\hat{\mathbf{A}}$, and any active constraint that should become inactive should be deleted from $\hat{\mathbf{A}}$. Thus we need to keep track of which of the inequality constraints are active and which are inactive at the current point $\mathbf{x}_k$. To do this we observe that only the inequality constraints could be added to, or deleted from, the active set, so we construct two position index sets. The first set includes the position indices for all the inequality constraints of $\mathbf{A}$, either active or inactive. This set will be called the inequality constraints set. The second set includes the position indices for all the active inequality constraints of $\mathbf{A}$. This set will be called the working set. When an inactive inequality constraint becomes active at an updated point $\mathbf{x}_{k+1}$, its position index in the matrix $\mathbf{A}$, which is included in the inequality constraints set, is added to the end of the working set. If, on the other hand, an active constraint becomes inactive at $\mathbf{x}_{k+1}$, then its position index is deleted from the working set. The actual inequality constraints never change position in the matrix $\mathbf{A}$, only their respective position indices are added to, or deleted from, the working set. In this way, whenever we need to find the step to the nearest inactive constraint, we only need to consider those constraints whose position indices in $\mathbf{A}$, i.e., in the inequality constraints set, are not included in the working set.

The second step in the inversion algorithm is to determine the descent vector $\mathbf{p}_k$. This is the Gauss-Newton descent vector which can be computed in two steps. First, the modified Cholesky factors of the projected Hessian $\mathbf{Z}_k^T \mathbf{H}_k \mathbf{Z}$ are given by

$$\overline{\mathbf{H}}_z = \mathbf{Z}_k^T \mathbf{H}_k \mathbf{Z}_k + \mathbf{E}_k = \overline{\mathbf{L}}_k \overline{\mathbf{D}}_k \overline{\mathbf{L}}_k^T . \tag{5.6}$$

We solve

$$\overline{\mathbf{L}}_k \overline{\mathbf{D}}_k \overline{\mathbf{L}}_k^T \mathbf{p}_z = -\mathbf{Z}_k^T \mathbf{g}_k \tag{5.7}$$

for $\mathbf{p}_z$, which is in the null space of $\hat{\mathbf{A}}$ defined by $\mathbf{Z}$. Then we find $\mathbf{p}_k = \mathbf{Z}_k \mathbf{p}_z$. The resulting descent vector $\mathbf{p}_k$ satisfies all the active constraints defined by $\hat{\mathbf{A}}$ at the current point $\mathbf{x}_k$.

To solve (5.7), we set

$$\overline{\mathbf{D}}_k \overline{\mathbf{L}}_k^T \mathbf{p}_z = \mathbf{u}_k ,$$

so that

$$\overline{\mathbf{L}}_k^T \mathbf{p}_z = \overline{\mathbf{D}}_k^{-1} \mathbf{u}_k \ , \tag{5.8}$$

and

$$\overline{\mathbf{L}}_k \mathbf{u}_k = -\mathbf{Z}_k^T \mathbf{g}_k \ . \tag{5.9}$$

At this point the advantages of using the Cholesky factorisation become obvious. Because of the structures of the diagonal matrix $\overline{\mathbf{D}}_k$ and the lower triangular matrix $\overline{\mathbf{L}}_k$, no explicit matrix inverse computation is necessary to find $\mathbf{p}_z$. We first solve (5.9) for $\mathbf{u}_k$ by forward substitution, then solve (5.8) for $\mathbf{p}_z$ by back substitution. The descent vector $\mathbf{p}_k = \mathbf{Z}_k \mathbf{p}_z$ is then computed.

The third step in the inversion algorithm is to determine the step length used to update the current point $\mathbf{x}_k$ to obtain $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$. The natural value for α_k in the Gauss-Newton method is unity for quadratic error energy function $F(\mathbf{x})$. However, for a general error energy $F(\mathbf{x})$, quadratic approximation may be valid only at points $\mathbf{x}_k$ close to the minimum solution point $\mathbf{x}^*$. Nonetheless, we always first consider $\alpha_k = 1$ as the best choice for the step length whenever possible. In practice, however, it is useful to limit the maximum change that can be made in $\mathbf{x}_k$ at any one iteration, so that we attempt to force convergence to a solution nearest to the initial guess. This, of course, is also consistently achieved if reasonable upper and lower bounds are placed on the elements of $\mathbf{x}_k$ using the inequality constraints. But by simply limiting the step length we could possibly prevent all the extra housekeeping due to addition and deletion of constraints and the associated matrix updates. To limit the maximum change in $\mathbf{x}_k$ we compute, at each iteration, the step length $\tilde{\alpha}$ that satisfies $|\tilde{\alpha} \mathbf{p}_k| \le \Delta$, where D is a pre-defined constant parameter. Obviously, a different value for D has to be assigned depending on which set of variables we are considering. For example, if we are solving for the acoustic impedance problem, then we must have a constant Δ_x for the starting impedance variables $\mathbf{x}$, and a constant Δ_y for the impedance gradient variables $\mathbf{y}$. Thus we need to evaluate $\tilde{\alpha}_x$ and $\tilde{\alpha}_y$ and then set $\tilde{\alpha} = \min\{\tilde{\alpha}_x, \tilde{\alpha}_y\}$. When solving for the wavelet problem we evaluate $\tilde{\alpha} = \tilde{\alpha}_w$ for a predetermined Δ_w. For the boundary location problem, however, we also require that the minimum step length $\tilde{\alpha}$ results in at least one boundary location variable being perturbed by a minimum of one sample interval. This is because, if none of the boundary locations are perturbed, there is no change in the error energy.

Another upper limit imposed on α_k is due to the step length $\overline{\alpha}$, which is the step to the nearest inactive constraint, see section 5.2. The step length $\overline{\alpha}$ is the minimum step of all possible step lengths to the inactive inequality constraints. If $\overline{\alpha}$

72

is taken as α_k, then the inactive constraint that corresponds to $\overline{\alpha}$ should be added to the active set of constraints by adding a row to $\hat{\mathbf{A}}$ and then updating the position index set and the **QR**-factors of $\hat{\mathbf{A}}^T$, so that we obtain a new $\mathbf{Z}_k$ and $\mathbf{Q}_1$ by partitioning $\mathbf{Q}$.

The choice of the step length α_k is determined basically in three steps. First, if the $\min\{\overline{\alpha}, \tilde{\alpha}\} \leq 1$, then α_k is taken to be the $\min\{\overline{\alpha}, \tilde{\alpha}\}$, otherwise α_k is set to 1. Second, we test if $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < F(\mathbf{x}_k)$; if this is the case when $\alpha_k = \tilde{\alpha}$, then we accept α_k as the step length. However, when $\alpha_k = \overline{\alpha}$, we further investigate whether $\mathbf{p}_k \mathbf{g}_{k+1}$ is positive or negative ($\mathbf{g}_{k+1}$ is the gradient vector at $\mathbf{x}_{k+1}$). In the case when $\mathbf{p}_k \mathbf{g}_{k+1} < 0$, we accept $\alpha_k = \overline{\alpha}$ and add the corresponding constraint to the active set. In the case when $\mathbf{p}_k \mathbf{g}_{k+1} > 0$, then this would mean that there is a point between $\mathbf{x}_k$ and $\mathbf{x}_k + \alpha_k \mathbf{p}_k$ that should have a lower error energy value than $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$. The reason for doing this is to avoid adding a constraint to the active set whenever possible. Third, when $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) > F(\mathbf{x}_k)$, then we use the linear search method to find a new α_k for which we obtain a reduction in the error energy function as described in section 3.9.

The final step in the inversion program is to establish convergence to a solution $\mathbf{x}^*$. To do this we need to satisfy one of four convergence criteria. Two of these convergence tests are concerned with the error energy value, one with the projected gradient magnitude $\|\mathbf{Z}_k^T \mathbf{g}_k\|$ at $\mathbf{x}_k$, and one with the total number of iterations. When the error energy value at the current point is less than or equal to a small tolerance value *etol*, i.e. $F(\mathbf{x}_k) \leq etol$, then convergence is established and the current point becomes the solution $\mathbf{x}^*$. Also, when the decrease in error energy is within the tolerance, i.e. $F(\mathbf{x}_k) - F(\mathbf{x}_{k+1}) \leq etol$, then the iteration process should terminate and accept the current point as the solution $\mathbf{x}^*$.

The third criterion for convergence is applied to the projected gradient magnitude at the current point $\mathbf{x}_k$. When $\mathbf{x}_k$ is at, or close to, a minimum then, for a small tolerance *gtol*, $\|\mathbf{Z}_k^T \mathbf{g}_k\| \leq gtol$. This means that we are either at a weak minimum or at a saddle point. The decision that the point $\mathbf{x}_k$ is a weak minimum or a saddle point depends on whether the projected Hessian $\mathbf{Z}_k^T \mathbf{H}_k \mathbf{Z}_k$ has been modified by $\mathbf{E}_k$ to obtain the Cholesky factorisation $\overline{\mathbf{L}}_k \overline{\mathbf{D}}_k \overline{\mathbf{L}}_k^T$. When at least one of the elements of the diagonal matrix $\mathbf{E}_k$ is non zero, then $\mathbf{Z}_k^T \mathbf{H}_k \mathbf{Z}_k$ is indefinite which means that $\mathbf{x}_k$ is a saddle point, otherwise the projected Hessian is positive definite and $\mathbf{x}_k$ is a weak minimum.

A weak minimum at $\mathbf{x}_k$ indicates that convergence has been achieved at the subspace defined by $\mathbf{Z}_k$. However, there remains the possibility that further reductions in error energy could still be obtained if one, or more, of the active inequality constraints is deleted from the active set defined by $\hat{\mathbf{A}}$. Thus we compute the vector λ_k of the Lagrange multipliers. Reduction in error energy can be obtained

only if one, or more, of the constraints has a negative Lagrange multiplier (see section 4.9). We delete only one constraint at a time, and choose the one that is most negative first. When a constraint is deleted from the active set, the **QR** factors of $\hat{\mathbf{A}}^T$ are re-computed and a new null space matrix $\mathbf{Z}_k$ is determined. The case when all $\lambda_i \geq 0$ corresponds to a weak minimum, and the point $\mathbf{x}_k$ is accepted as the solution $\mathbf{x}^*$.

When $\mathbf{x}_k$ is a saddle point, we choose to take the projected descent direction $\mathbf{p}_z$ as a direction of negative curvature (see section 3.10) defined by

$$\overline{\mathbf{L}}_k^T \mathbf{p}_z = \mathbf{e}_s ,$$

where $\mathbf{e}_s$ is a unit vector having the value 1 at the coordinate s, and s is the direction that was modified the most to obtain the modified Cholesky factors $\overline{\mathbf{L}}_k$ and $\overline{\mathbf{D}}_k$. That is, we choose s such that $\overline{d}_{ss} - e_{ss}$ is least, where $\overline{d}_{ss}$ are the diagonal elements of $\overline{\mathbf{D}}_k$ and e_{ss} are the diagonal elements of $\mathbf{E}_k$ of (5.6). Having determined $\mathbf{p}_z$, it follows that $\mathbf{p}_k = \mathbf{Z}_k \mathbf{p}_z$, which is then taken as the descent direction of this iteration.

The last criterion for convergence is simply to assign a maximum number of iterations the inversion program can perform. Once this number of iterations is reached, the program terminates with $\mathbf{x}_k$ as the solution $\mathbf{x}^*$.

A schematic flowchart of the main steps of the algorithm is shown in Fig. 5.1. In the diagram we start at the k-th iteration with an initial guess $\mathbf{x}_k$, (i.e., boundary locations, acoustic impedances, or wavelet parameters at a well location.) and a Jacobian matrix $\mathbf{J}_k$, from which we can compute the gradient vector $\mathbf{g}_k$ and a Hessian matrix $\mathbf{H}_k$ using the residual vector $\mathbf{r}(\mathbf{x}_k) = \mathbf{s}(\mathbf{x}_k) - \mathbf{s}^{obs}$.

5.4. The initial guess and convergence to the correct solution

5.4.1. The region of convergence

The parameterised initial guess must be within a region of convergence around the solution for the inversion process to converge to the correct solution. In the case of the boundary location problem, this region is found to have a radius of one-half of the central lobe width for a zero phase seismic wavelet.

This can be illustrated by considering a three layer model, so that we have two boundary variables representing the middle interfaces; say t_1 and t_2. An error energy function was computed for a correct boundary locations solution $\mathbf{t} = \begin{bmatrix} 28 & 54 & 76 \end{bmatrix}^T$, with a starting impedance $\mathbf{x} = \begin{bmatrix} 5000 & 3000 & 4000 \end{bmatrix}^T$ and a zero impedance gradient at each layer, so that $\mathbf{y} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. The error energy function $F(\mathbf{x})$ was generated by scanning through all possible points $t_1 = 2, 4, \ldots, 74$ for each $t_2 = 4, 6, \ldots, 76$, so that the linear inequality constraint $t_2 - t_1 \geq 2$ was always satisfied. The wavelet used was near a zero phase having the parameters $\mathbf{w} = [10\ 20\ 60\ 75\ 110\ 110\ 0.418\ 0.113\ 0]^T$.
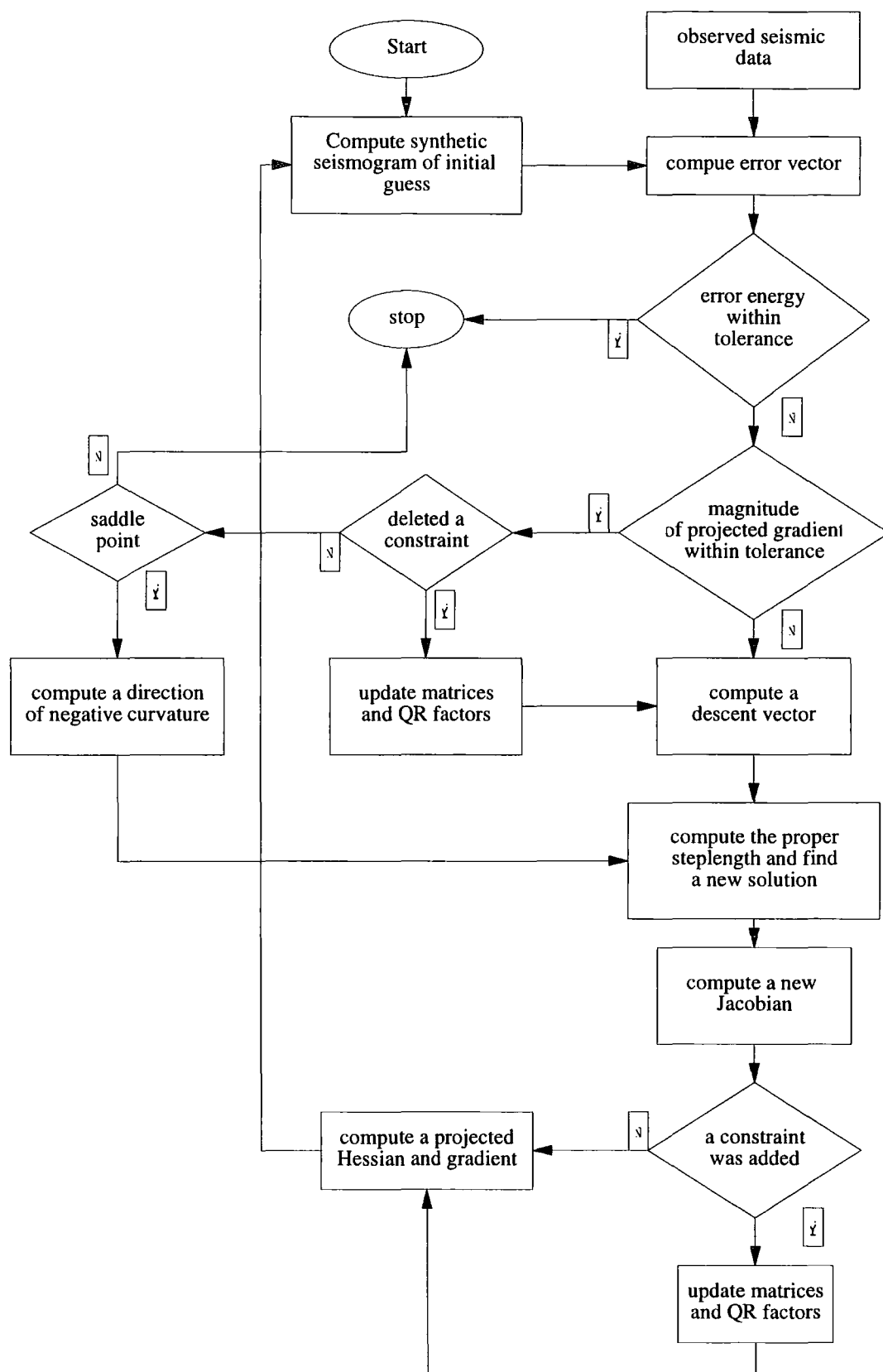
Figure 5.1. A schematic flowchart of the main steps of the inversion program using the active set strategy.

75

This wavelet has a central lobe width of approximately 18 ms. At each point (t_1, t_2), the initial guess was evaluated by convolving the reflection coefficient series generated from the two-way time at that point and the acoustic impedance information **x** and **y**, sampled at 2 ms with the 20 samples time domain wavelet obtained from the wavelet parameters **w**. The resulting 3-dimensional error energy surface is shown in Fig. 5.2a, where it can be seen that the point (28, 54) represents the global minimum error energy.
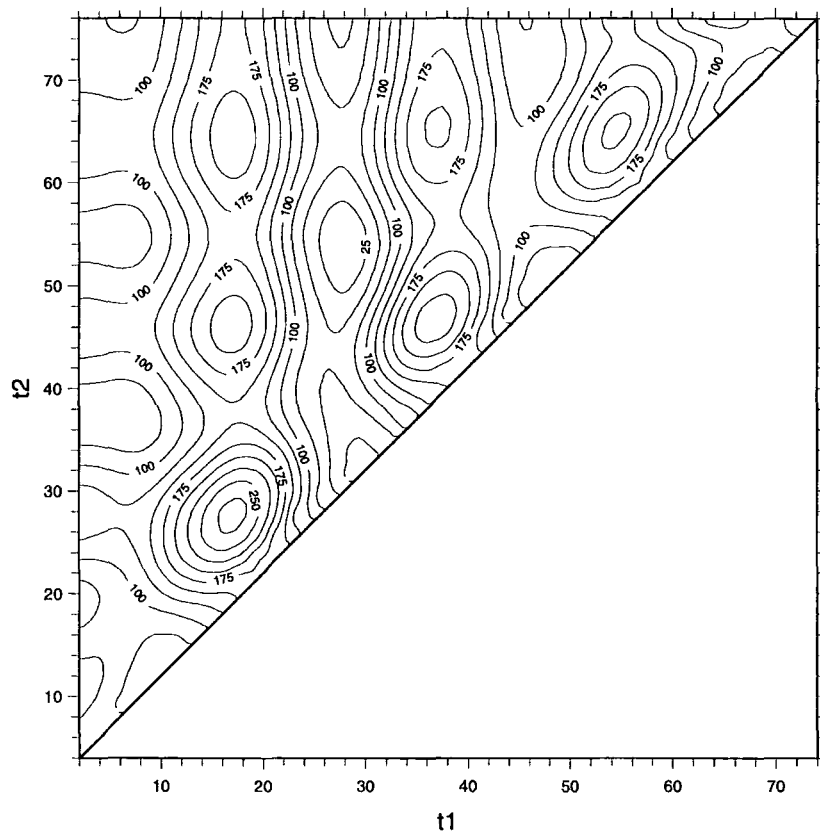
Figure 5.2a. The error energy surface showing the region of convergence for the global minimum (28, 54).

Figure 5.2a also illustrates that any point in the region surrounding the point (28, 54) at a radius of 8 ms will converge to this global minimum. The 20-sample time domain wavelet is shown in Figure 5.2b, where it can be seen that the central lobe width is approximately 18 ms, which also explains why other local minima tend to be separated by about 18 ms away from the global minimum along the lines $t_1 = 28$ and $t_2 = 54$. This is because the overlap of the secondary lobes of the wavelets from each of the two reflecting boundaries will result in lower error energy if they are separated by a multiple of the central lobe width.
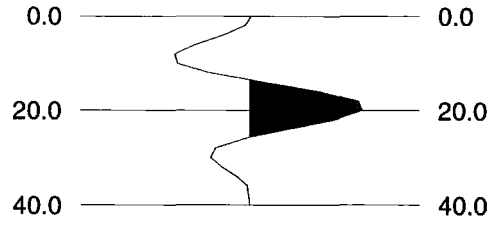
76

Figure 5.2b. The time domain wavelet used to generate the error energy surface of Figure 5.2a. Note that the vertical axis represents the two-way travel-time in ms.

5.4.2 The convergence criteria and solution quality

As previously mentioned in section 5.3, the convergence to a solution is established when one of four criteria is satisfied. When convergence is established because $F(\mathbf{x}_k) \leq etol$, then obviously the convergence is perfect, although it might not be unique. However, when convergence is reached due to $F(\mathbf{x}_k)\text{-}F(\mathbf{x}_{k+1}) \leq etol$, or $\left\| \mathbf{Z}_k^T \mathbf{g}_k \right\| \leq gtol$, or when the maximum number of iterations allowed is reached, then the solution obtained may still contain error energy that is probably much larger than *etol*, due to the high amplitude value of the samples in the problem at hand. In such cases we need to obtain a relative measure of the error energy with respect to the observed seismic trace. This relative error energy measure is obtained as a percentage:

$$E_{rel} \; (\%) = \frac{2(\text{error energy value at } \mathbf{x}^*)}{\text{observed trace energy}} \times 100$$

$$= \left(\frac{\displaystyle\sum_{i=1}^{m} \left(s_i(\mathbf{x}^*) - s_i^{obs} \right)^2}{\displaystyle\sum_{i=1}^{m} \left(s_i^{obs} \right)^2} \right) \times 100$$

where $s_i(\mathbf{x}^*)$ represents the synthetic seismic trace at the solution $\mathbf{x}^*$, and s_i^{obs} represents the observed seismic trace. Obviously when $F(\mathbf{x}_k) \leq etol$, then E_{rel} is very small, or possibly zero, for a small *etol*, while in the case of convergence with any of

the criteria E_{rel} could still be large, but the maximum is 200%, which is obtained when the traces $s(x^*)$ and s^{obs} are out of phase.

5.4.3. Uniqueness and resolution of inversion solutions

Obtaining a solution to the inversion problem with zero error energy does not mean that there is no other solution with zero error energy. This is the problem of non-uniqueness in inversion solutions. The uniqueness of the inversion solution is discussed by Backus and Gilbert (1970) and Parker (1977). The non-uniqueness, or ambiguity, in this inversion process is a product of the way we construct the earth model to obtain the initial guess and the inversion solution, and the criteria we use to establish the degree of fit between the inversion solution and the correct, or observed, seismic solution after each iteration.

The initial guess impedance profile, and also that of the solution, is constructed in such a way that we have discontinuities at the boundary locations. This would mean that the solution has infinite bandwidth. The synthetic seismogram and the observed trace are actually band-limited, thus when we compare them for their degree of fit, we can only make the comparison in this limited frequency band. The frequency components that are outside this band are filtered out by the wavelet filter when computing the synthetic seismogram. The result of this is that low frequency components can be added to the impedance profile and still obtain the same degree of fit with the observed data, so that the inversion process can add any low or high frequency components outside the frequency band defined by the wavelet and yet still give a perfect fit. This indicates that the more we require our solution to have a resolution higher than that defined by the band-limited seismic data, i.e. by making our impedance solution broad-band, the more non-uniqueness, or ambiguity, we introduce in the obtained impedance solution.

Another form of ambiguity occurs when, for example, we have a sand layer immersed in shale, and this layer is at or below its tuning thickness. At the tuning thickness of such a layer, we obtain a maximum reflection amplitude (Widess, 1973). This is due to the constructive interference between the primary and secondary lobes of the seismic wavelets that are reflected from the top and bottom boundaries of the layer, thus giving a strong trough and a strong peak amplitudes. For layer thickness less than the tuning thickness, the peak to trough time separation becomes invariant to the tuning thickness, and all thickness information becomes encoded in the peak to trough amplitude, which becomes progressively smaller as the layer becomes thinner. When inverting for a thin layer the ambiguity arises because the inversion process can either decrease the thickness of the layer or decrease the contrast in acoustic

impedance between the layer and its surrounding material, and still obtain the same seismic response.

5.5. Synthetic examples

Several examples will be given in this section to study the performance of the inversion program, and then to attempt to use the results to reach a conclusion on how to invert the field recorded seismic data. The synthetic examples are divided into inverting for (1) boundary locations, (2) acoustic impedance variables, (3) a combination of boundary locations and acoustic impedance variables, (4) wavelet parameters, and (5) a combination of boundary locations, acoustic impedance variables and wavelet parameters.

In almost every synthetic example discussed in the following sections, the observed trace is obtained by the time domain convolution of the earth model with the parameters shown in Table 5.1. In all cases the sampling interval is 2 ms. The boundary location vector t is in units of ms of two-way travel-time, the starting impedance vector x has units of $gcm^{-3}ms^{-1}$ and the impedance gradient y has units of $gcm^{-3}ms^{-1}$/sample interval.

layer number	observed boundary locations (ms)	observed starting impedances $(gcm^{-3}ms^{-1})$	observed impedance gradients $(gcm^{-3}ms^{-1}/sample)$
n	t_{obs}	x_{obs}	y_{obs}
1	60	11000	0
2	74	6000	0
3	82	8000	0
4	112	5000	0
5	126	7000	0
6	180	6000	0

Table 5.1 The synthetic observed earth model used to generate most of the observed seismic data for the synthetic examples studied in this chapter.

The wavelet used to generate the observed seismic trace has the parameters vector $w_{obs}=$ [24 28 55 84 115000 115000 0.418 0.113 0]^T. The initial guesses for all the synthetic examples are variants of the observed data of Table 5.1 and the wavelet parameters above.

5.5.1. Examples of boundary location inversion

Two representative examples are discussed here. First, we discuss an example where the boundaries are within the region of convergence and with correct polarities across the boundaries. In the second, we add the complication of having polarity

reversals across some of the boundaries. We then discuss the effect of random noise on the inversion for boundary locations. We examine the inversion results when the signal-to-noise ratio in the observed trace is 4, 2, 1 and ½.

5.5.1.1. A simple boundary location problem

In this example all the boundary locations are misplaced, with respect to observed earth model boundaries, by up to 8 ms. The observed boundary locations and the initial guess data are shown in Table 5.2.

layer *number* n	observed boundary locations (ms) t_{obs}	solution boundary locations (ms) t_{sol}	initial boundary locations (ms) t_{ini}	observed starting impedances $(gcm^{-3}ms^{-1})$ x_{obs}	observed impedance gradients $(gcm^{-3}ms^{-1}/sample)$ y_{obs}
1	60	60	52	11000	0
2	74	74	70	6000	0
3	82	82	88	8000	0
4	112	112	104	5000	0
5	126	126	132	7000	0
6	180	180	180	6000	0

Table 5.2. Data for the boundary location inversion problem of section 5.5.1.1. Since this is a boundary locations inversion problem, no impedance variables are present, and only the constants x_{obs} and y_{obs} are shown.

The 8 ms difference in boundary locations for layers t_1 and t_4 represents the maximum value for any of the boundaries to converge, that is the radius of the convergence zone. For this example convergence to the correct solution, i.e. t_{obs}, was obtained in five iterations. The initial relative error energy, with respect to t_{obs}, was $E_{rel}= 184\%$, and the final solution had zero error energy. The impedance profiles for the observed data, initial guess and inversion solution for this example are shown in Figure 5.3a. The solutions after each iteration are shown in Figure 5.3b. In Figure 5.3b, the first and last traces are the observed seismic trace, and it is shown twice so that it can be compared to the initial guess and the final solution. The second trace is the seismic response of the initial guess impedance profile, and the trace before the last, i.e. trace number 7, is the seismic response to the final impedance solution. Corresponding to each iteration is an error trace. The set of all error traces is shown in Figure 5.3c. The first error trace represents the initial E_{rel} of 184% the last error trace has all its elements being zero and represents the final error energy; and the middle error traces show the progression of the error trace towards zero at each iteration. This example illustrates that the boundary location solution can converge to the global minimum when the initial guess is within the region of convergence.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.3a The impedance profiles for the example of section 5.5.1.1. Note: for all the impedance profile figures in this chapter, the initial guess profile is a dashed line, the observed profile is a dotted line and the solution profile is a solid line. Note that t_1 and t_4 are misplaced by 8 ms. In this example convergence to the correct solution was obtained.

Figure 5.3b. The solution traces for the example of section 5.5.1.1. The first and last traces, i.e., traces numbered 1 and 8, are the observed seismic traces. Trace number 2 is the initial guess seismic response. Traces 3-7 represent the seismic response of the five iterations of inversion performed to obtain the final solution of trace 7. Observe that traces 7 and 8 are exactly the same. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.3c. The error traces for the example of section 5.5.1.1. These traces correspond to the solution traces of figure 5.3b. The first error trace corresponds to the solution trace numbered 2 and the last error trace corresponds to the solution trace numbered 7. Note that the vertical axis represents the two-way travel-time in ms.

5.5.1.2. Boundary locations problem with a polarity reversal

In this example the observed earth model was altered so that a polarity reversal was introduced in layer 4 in the initial guess. The observed and initial guess earth models are listed in Table 5.3.

layer number	observed boundary locations (ms)	solution boundary locations (ms)	initial boundary locations (ms)	observed starting impedances ($gcm^{-3}ms^{-1}$)	initial starting impedances ($gcm^{-3}ms^{-1}$)	observed impedance gradients ($gcm^{-3}ms^{-1}$/sample)
n	t_{obs}	t_{sol}	t_{ini}	x_{obs}	x_{init}	y_{obs}
1	60	60	56	11000	11000	0
2	74	72	72	6000	6000	0
3	82	86	86	8000	8000	0
4	112	108	108	5000	7500	0
5	126	126	130	7000	7000	0
6	180	180	180	6000	6000	0

Table 5.3. The observed, solution and initial guess data for the example of section 5.5.1.2, the initial guess is showing incorrect boundaries and a reversed polarity in layer 4.

The initial E_{rel}= 105% and the inversion program terminated after two iterations because the solution converged with E_{rel}= 35% after both the first and second iteration. The impedance profiles of this example are shown in Figure 5.4a, where it can be observed that the inversion solution has converged to the two correct boundary locations for interfaces 1 and 5, while for the interfaces in between the inversion has failed to converge to the correct boundary locations. This, of course, is due to the polarity reversal across interface 4, at 108 ms. The error traces shown in Figure 5.4b give a better insight into the optimum solution obtained. In the second trace much of the error energy is concentrated around the interface where the polarity reversal is found. There is also strong error energy at around 84 ms, but this is mainly due to the difference in reflection strength, i.e. impedance contrast, around that boundary in the initial guess impedance profile as compared to that in the observed profile. Thus the concentration of error energy should give us an idea on where adjustments in the earth model should be made to obtain an improved initial guess.

5.5.1.3. The effect of noise on boundary locations inversion

In the previous two synthetic examples of boundary locations inversion, the signal-to-noise ratio was infinite because no noise was present in the observed trace. In real seismic data, however, we should not expect the observed trace to be noise-free, so that in this section we consider the performance of the inversion process when the observed trace is contaminated with different levels of noise energy.

The type of noise that will be added to the observed trace is random noise with zero mean energy value. The random noise is generated from a flat band-limited amplitude spectrum and random phase spectrum which are then transformed into the time domain by inverse Fourier transformation (Fox, 1987). By making use of Parceval's theorem, (Brigham, 1988), we can determine the appropriate amplitude value of the band-limited amplitude spectrum to generate random noise of specified energy.

We re-consider the boundary locations inversion problem of section 5.5.1.1, see Table 5.2, with different added random noise energies so that the observed trace has a signal-to-noise ratio of 4, 2, 1 and ½. In all cases the random noise is band limited in to the range 10-85 Hz.
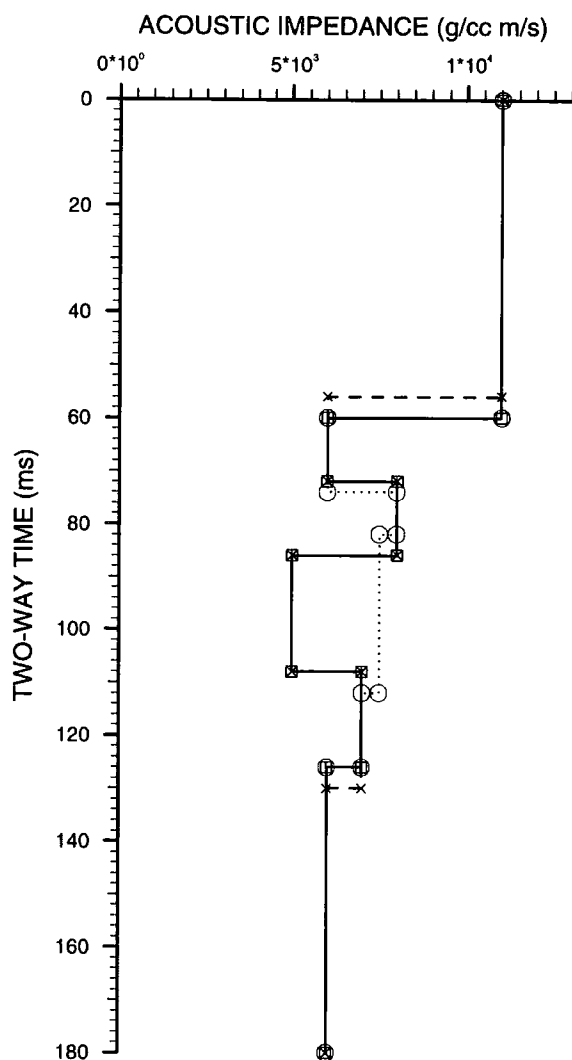
ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.4a. The impedance profiles for the example of section 5.5.1.2 showing the polarity reversal at boundary location 4. Only the first and last boundaries converged to their correct locations.
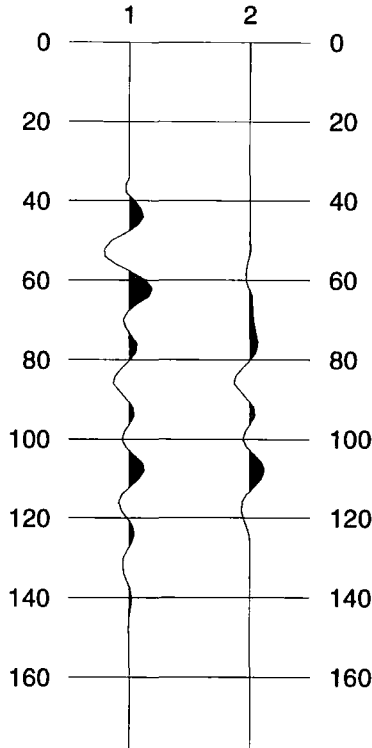
Figure 5.4b. The error traces for the example of section 5.5.1.2. They are the error traces for the two iterations performed by the inversion program, trace number 1 belongs to the first iteration. Note that the vertical axis represents the two-way travel-time in ms.

The solution traces when the signal-to-noise ratio is 4 are shown in Figure 5.5a. The initial E_{rel}= 220%, and in this case convergence to the correct solution was obtained in five iterations with E_{rel}= 25%, which could be expected since signal-to-noise ratio is 4. In Figure 5.5a the first and last traces are the noise contaminated observed traces, and traces 2-7 are the seismic responses for each of the five iterations. The corresponding error traces are shown in Figure 5.5b where, because convergence to the correct solution was achieved, the last error trace represents the total random noise that is contained in the noisy observed trace. The impedance profiles for this example are shown in Figure 5.5c, where convergence to the correct solution is shown.

Figure 5.5a The solution traces for the boundary locations inversion problem when the observed trace (numbered 1 and 8) has a signal-to-noise ratio of 4. The solution traces for the five iterations are traces 2 to 7. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.5b The error traces corresponding to the solution traces of Figure 5.5a. Because inversion to the correct solution was obtained, the last error trace, number 6, represents the total random noise in the observed seismic trace. Note that the vertical axis represents the two-way travel-time in ms.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.5c. The impedance profiles for the noise added boundary locations inversion example of section 5.5.1.3 when the signal-to-noise ratio of the observed trace is 4. For such a signal-to-noise ratio convergence to the correct boundary locations was obtained.

The previous example was repeated with a lower signal-to-noise ratio of 2. In this case the initial E_{rel}= 237%, and that, except for the first boundary, all boundaries failed to converge to their correct locations, so that the final E_{rel}= 42%. The inversion input and results are shown in Table 5.4.

layer number	observed boundary locations (ms)	solution boundary locations (ms)	initial boundary locations (ms)	observed starting impedances $(gcm^{-3}ms^{-1})$	observed impedance gradients $(gcm^{-3}ms^{-1}/sample)$
n	t_{obs}	t_{sol}	t_{ini}	x_{obs}	y_{obs}
1	60	60	52	11000	0
2	74	72	70	6000	0
3	82	84	88	8000	0
4	112	114	104	5000	0
5	126	134	132	7000	0
6	256	256	256	6000	0

Table 5.4. The data for the boundary locations inversion when the signal-to-noise ratio of the observed trace is 2.

Figure 5.6a shows the solution traces, which are numbered 2-7, for the six iterations the program performed. The impedance profiles for this example are shown in Figure 5.6b where it can be seen that the solution of the fifth boundary was in error by as much as 8 ms; i.e. the solution has diverged from the correct boundary. It is more likely for this boundary to have a large error in the final solution because of the low impedance contrast across it.
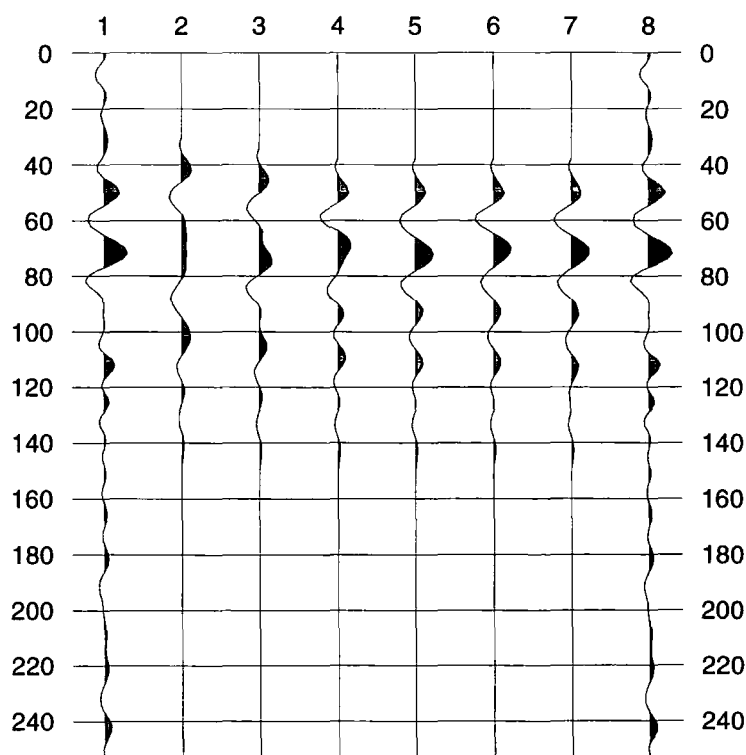
Figure 5.6a The observed trace (numbered 1 and 8) has a signal-to-noise ratio of 2. The solution traces for the six iterations the inversion program performed are traces 2 to 7. Note that the vertical axis represents the two-way travel-time in ms.

ACOUSTIC IMPEDANCE (g/cc m/s)
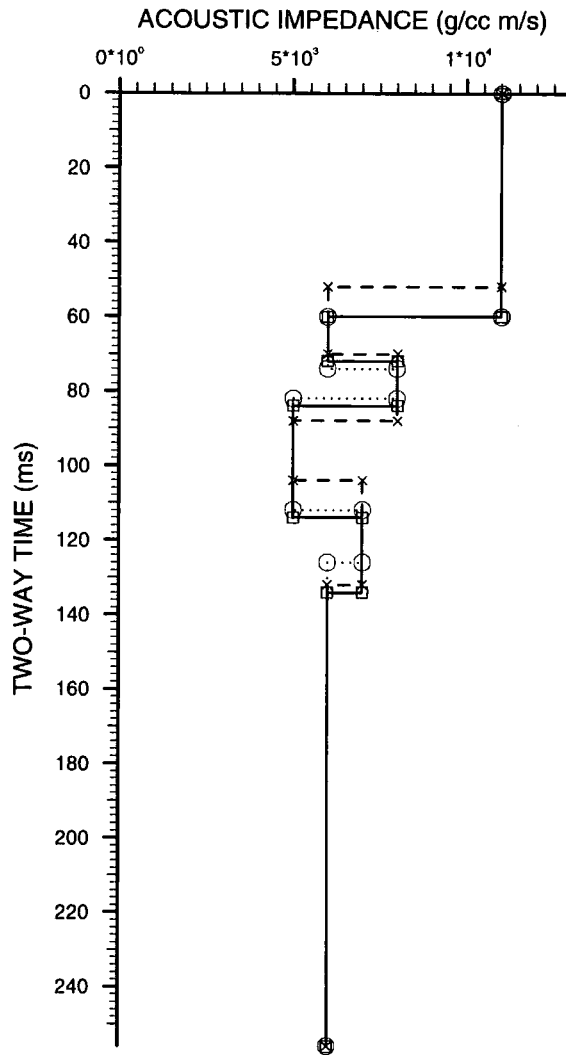
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.6b. The impedance profiles for the boundary locations inversion data listed in Table 5.4. The signal-to-noise ratio of the observed trace in this example is 4. Note that, except for the first boundary, no boundary has convergence to its correct location.

To examine the effect of noise even further, the previous example was repeated with a signal-to-noise ratio of 1. The data for this test are listed in Table 5.5.

layer number	observed boundary locations (ms)	solution boundary locations (ms)	initial boundary locations (ms)	observed starting impedances $(\text{gcm}^{-3}\text{ms}^{-1})$	observed impedance gradients $(\text{gcm}^{-3}\text{ms}^{-1}/\text{sample})$
n	t_{obs}	t_{sol}	t_{ini}	x_{obs}	y_{obs}
1	60	62	52	11000	0
2	74	72	70	6000	0
3	82	82	88	8000	0
4	112	114	104	5000	0
5	126	142	132	7000	0
6	256	256	256	6000	0

Table 5.5. The data for the boundary locations inversion when the signal-to-noise ratio of the observed trace is 1. The observed traces are shown in Figure 5.7a numbered 1 and 13.

The observed and solution traces are shown in Figure 5.7a. The initial $E_{rel}=$ 413%, then after 11 iterations the program terminated with $E_{rel}=78\%$. The impedance profiles are shown in Figure 5.7b, where it can be seen that only the third boundary converged to its correct solution, and that the fifth boundary, with the low reflection coefficient, actually diverged even further from its correct location as compared to the previous case when the signal-to-noise ratio was 2 that is shown in Figure 5.6b.
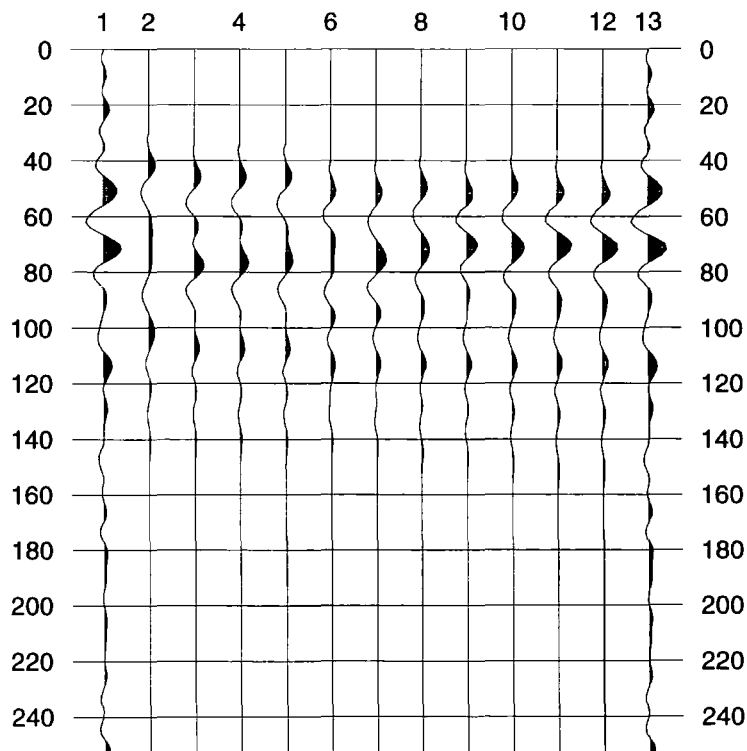
Figure 5.7a The observed trace (numbered 1 and 13) has a signal-to-noise ratio of 1. The solution traces for the 11 iterations the inversion program performed are traces 2 to 12. Note that the vertical axis represents the two-way travel-time in ms.

ACOUSTIC IMPEDANCE (g/cc m/s)
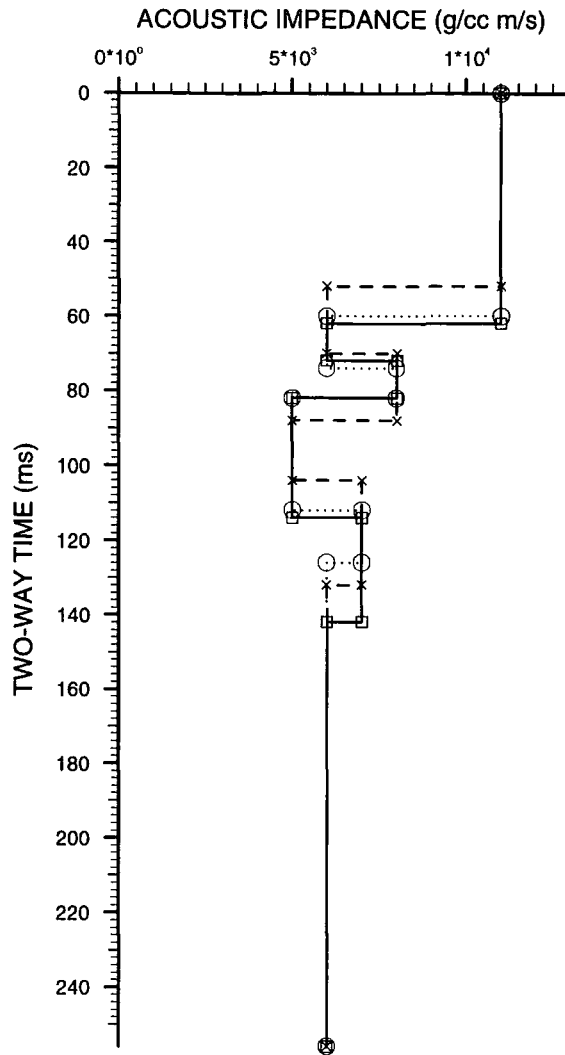
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.7b. The impedance profiles for the boundary locations inversion data listed in Table 5.5. The signal-to-noise ratio of the observed trace in this example is 1. Note that, except for the third boundary, no boundary has convergence to its correct location.

To take noise investigation one last step further, Figure 5.8a shows the same example with the observed trace having signal-to-noise ratio of ½. Figure 5.8a shows the solution traces of the 10 iterations the inversion program took to reduce E_{rel} from 287% to 183%. Although the first boundary, which has the highest reflection coefficient, and the second boundary converged to their correct values, the other boundary locations stayed well away from their correct locations. Table 5.6. shows the inversion results, and Figure 5.8b shows the impedance profiles for this example.

layer number n	observed boundary locations (ms) t_{obs}	solution boundary locations (ms) t_{sol}	initial boundary locations (ms) t_{ini}	observed starting impedances $(gcm^{-3}ms^{-1})$ x_{obs}	observed impedance gradients $(gcm^{-3}ms^{-1}/sample)$ y_{obs}
1	60	60	52	11000	0
2	74	74	70	6000	0
3	82	96	88	8000	0
4	112	116	104	5000	0
5	126	142	132	7000	0
6	256	256	256	6000	0

Table 5.6. The data for the boundary locations inversion when the signal-to-noise ratio of the observed trace is ½. The observed traces are shown in Figure 5.8a numbered 1 and 12.
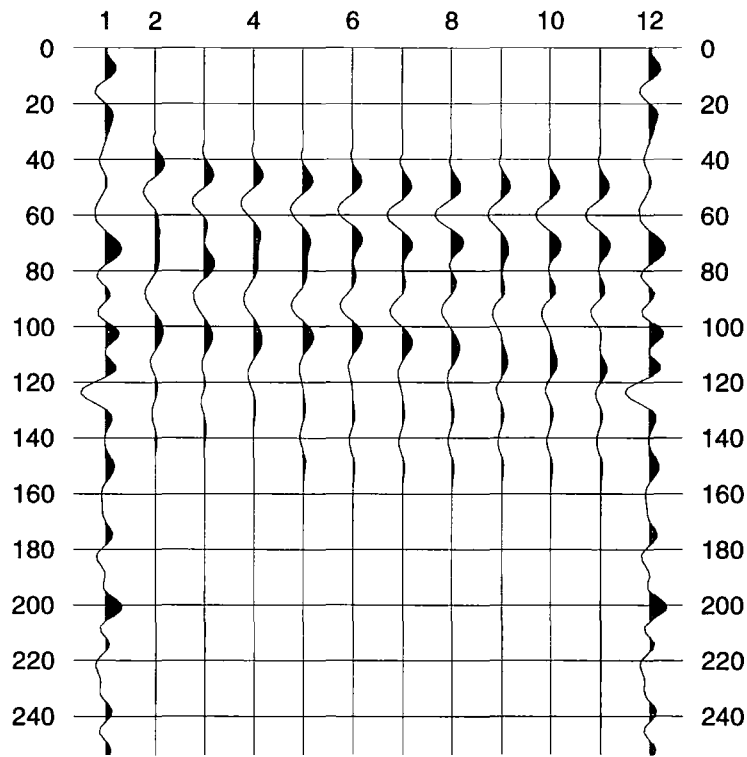
Figure 5.8a The observed trace (numbered 1 and 12) has a signal-to-noise ratio of ½. The solution traces for the 10 iterations the inversion program performed are traces 2 to 11. Note that the vertical axis represents the two-way travel-time in ms.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.
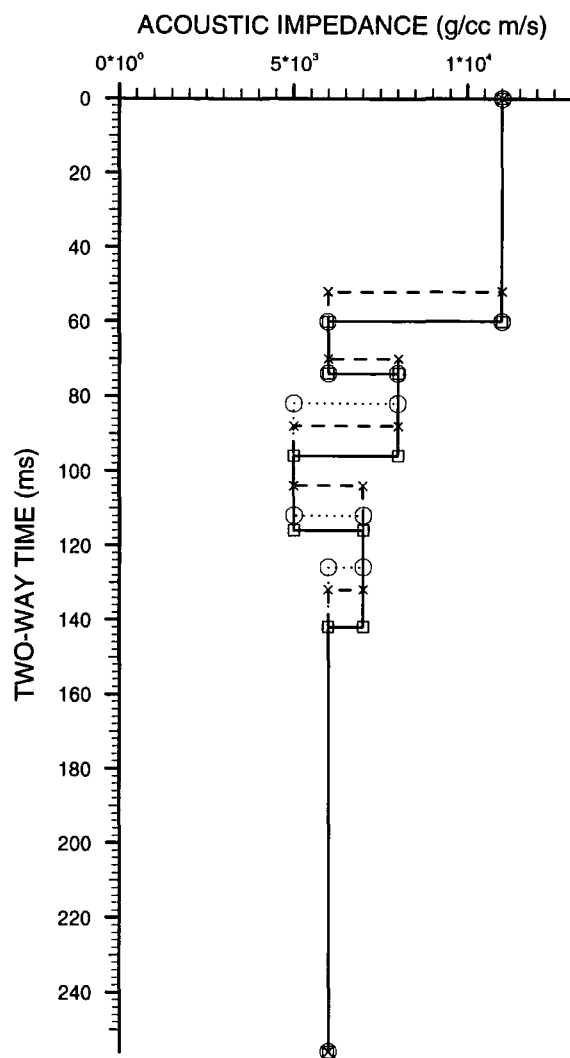
Figure 5.8b. The impedance profiles for the boundary locations inversion data listed in Table 5.5. The signal-to-noise ratio of the observed trace in this example is ½. Although the first boundary, which has the highest reflection coefficient, and the second boundary converged to their correct values, the other boundary locations stayed well away from their correct locations.

5.5.2. Examples of impedance inversion

A number of examples of impedance inversion are given here. The first is a straightforward impedance inversion problem where the boundary locations and wavelet parameters have their correct values. We then examine the same example with the impedance of the first layer fixed. We also examine an interesting case where the initial guess impedance profile has a constant value. The second example has polarity reversals across some of the interfaces. We then repeat the same example with the added complication of having impedance gradients in the observed

impedance profile. Finally, we consider the effect of random noise on impedance inversion.

5.5.2.1. A simple impedance inversion problem

The input and output data for this problem are shown in Table 5.7. Here we are only solving for the starting impedance vector x, so that each element in the vector y is equality constrained to zero.

The initial $E_{rel}= 11\%$, and the final error energy for the inversion solution x_{sol} is zero. The impedance profiles for this example are shown in Figure 5.9. Obviously this is a case where convergence to a non-unique solution is obtained. It is important to notice that in this solution the ratio of any of x_{obs} elements to the corresponding element in x_{sol} is always 1.07, that is $x_{obs}= 1.07x_{sol}$, so that they are equal except for a scale factor of 1.07. Convergence to x_{sol} occurred after four iterations.

layer number	observed boundary locations (ms)	observed starting impedances (gcm^{-3}ms^{-1})	solution starting impedances (gcm^{-3}ms^{-1})	initial starting impedances (gcm^{-3}ms^{-1})	observed impedance gradients (gcm^{-3}ms^{-1}/sample)
n	t_{obs}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	11000	10220	10000	0
2	74	6000	5603	5500	0
3	82	8000	7471	9000	0
4	112	5000	4669	7000	0
5	126	7000	6536	7500	0
6	180	6000	5603	5000	0

Table 5.7. The input and output data for the simple impedance problem of section 5.5.2.1.

To solve the non-uniqueness problem in this example it is necessary to equality constrain only one of the starting impedance values. This is because by fixing the impedance in one layer we are forcing the impedance profile of the solution to have the low frequency trend that is present in the observed impedance profile, thus the inversion impedance solution will converge to the same values as those in the observed impedance profile. Figure 5.10 shows the impedance results obtained when the first layer of the problem of Figure 5.9 is fixed, i.e. equality constrained, to its correct value. The correct solution was achieved, i.e. zero error energy, in four iterations.

An interesting example is shown in Figure 5.11 were the same impedance solution was obtained, with zero error energy, when the initial guess vector, x_{ini}, had all its elements equal to that of the first layer impedance value. This solution was

obtained in four iterations, and the impedance of the first layer was fixed. In this case, the initial E_{rel} = 100%.
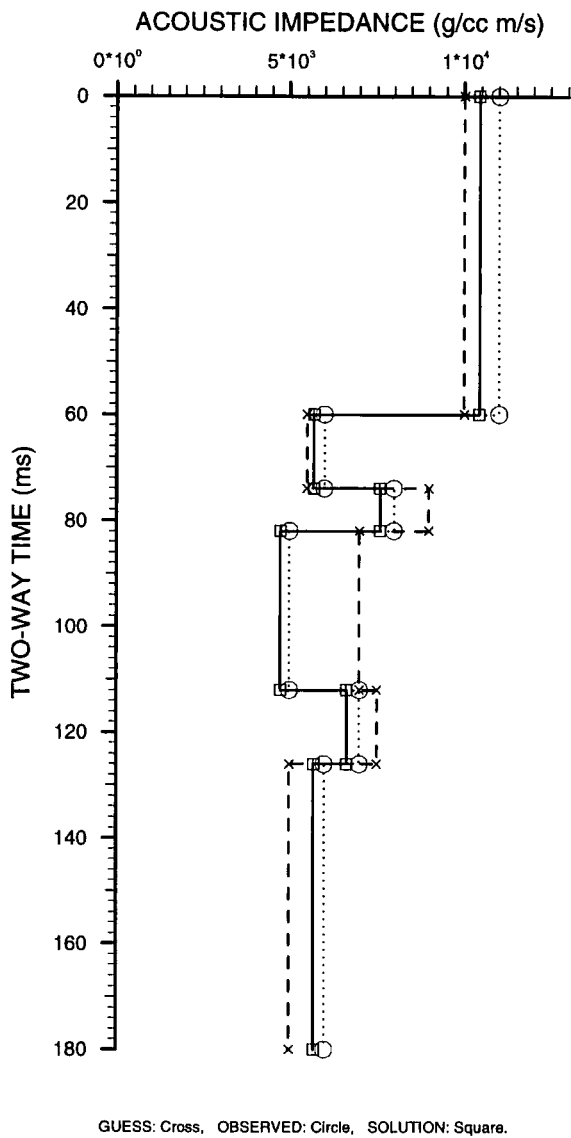
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.9. The impedance profiles of the example of section 5.5.2.1 showing a non-unique convergence of the impedance solution.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.10. The impedance profiles after equality constraining the first layer for the example of section 5.5.2.1.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross,   OBSERVED: Circle,   SOLUTION: Square.
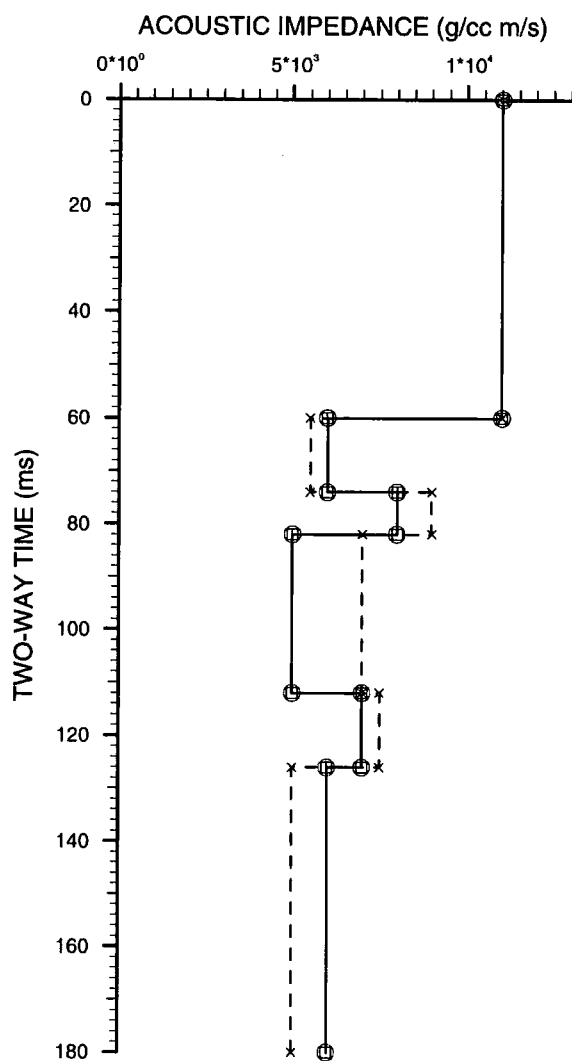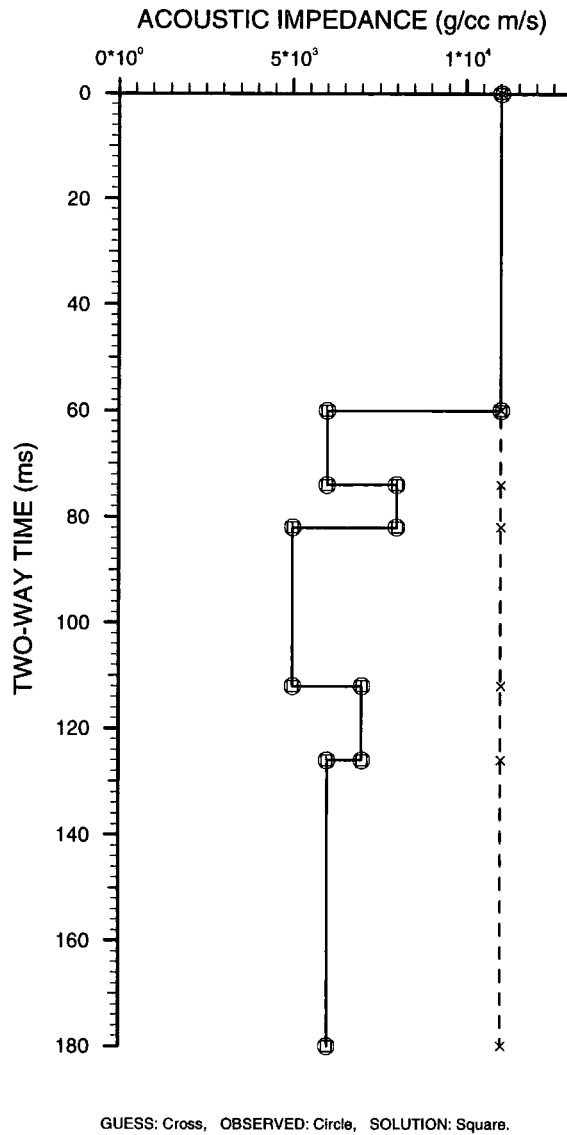
Figure 5.11. The impedance profiles for the example shown in Figure 5.10 where the initial guess impedance is constant and equal to the first layer impedance.

5.5.2.2. Impedance inversion with polarity reversals

The input and output data for an example where three polarity reversals were introduced in the initial guess are shown in Table 5.8.   When solving for the impedance problem with these reversals, convergence to the correct solution was achieved in only four iterations with zero error energy.

layer number	observed boundary locations (ms)	observed starting impedances (gcm⁻³ms⁻¹)	solution starting impedances (gcm⁻³ms⁻¹)	initial starting impedances (gcm⁻³ms⁻¹)	observed impedance gradients (gcm⁻³ms⁻¹/sample)
n	t_{obs}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	11000	11000	11000	0
2	74	6000	6000	5500	0
3	82	8000	8000	4000	0
4	112	5000	5000	7000	0
5	126	7000	7000	6000	0
6	180	6000	6000	5000	0

Table 5.8. Input data and solution for the impedance inversion with three polarity reversals example.

In this example the initial error energy is 194%, and the first layer impedance is fixed to its correct value of 11000 gcm⁻³ms⁻¹. The impedance results of this example are shown in Figure 5.12.

Inverting for impedance with polarity reversals in the initial guess in the above example was done with the impedance gradient at each layer fixed to be zero. To see the effect on the inversion result of having impedance gradients in the observed data, when the initial guess has polarity reversals, a test was performed by using the data in Table 5.9. The reason for this example is to investigate the possibility that polarity reversals could be removed simply by changing the gradients in the layers at the bottom of which we have the polarity reversals. Table 5.9 shows the input and inversion results when the two reversals are present in the initial guess, and the observed impedance profile layers have both positive and negative gradients. The initial E_{rel}= 194%.

layer number	observed boundary location (ms)	observed starting impedance (gcm⁻³ms⁻¹)	solution starting impedances (gcm⁻³ms⁻¹)	initial starting impedances (gcm⁻³ms⁻¹)	observed impedance gradients (gcm⁻³ms⁻¹/sample)	solution impedance gradients (gcm⁻³ms⁻¹/sample)	initial impedance gradients (gcm⁻³ms⁻¹/sample)
n	t_{obs}	x_{obs}	x_{sol}	x_{ini}	y_{obs}	y_{sol}	y_{ini}
1	60	11000	11000	11000	-50	-50	0
2	74	6000	6000	5500	50	50	0
3	82	8000	8000	4000	-100	-100	0
4	112	5000	5000	7000	100	100	0
5	126	7000	7000	6000	-20	-20	0
6	180	6000	6000	5000	0	0	0

Table 5.9. The initial guess has two reversals introduced at boundaries 2 and 3, and a non-zero observed impedance gradient vector.

The inversion converged to the correct solution in four iterations with zero error energy. At each iteration the starting impedance of the first layer was fixed at 11000 gcm^{-3}ms^{-1}. The impedance results of the problem of Table 5.9 are shown in Figure 5.13.
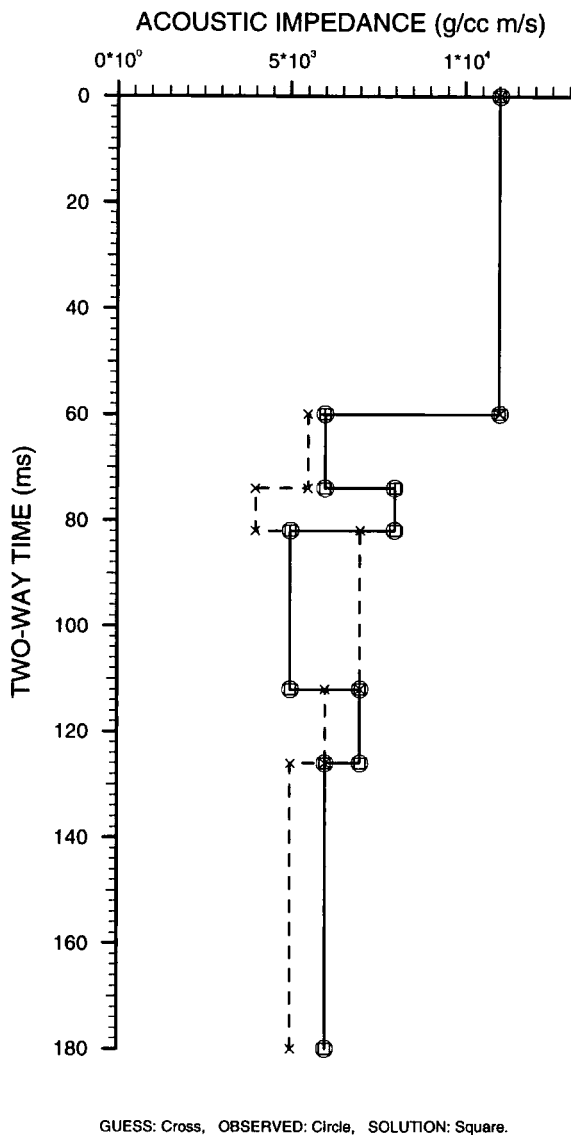
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.12. The impedance profiles when three reversals were introduced in the initial guess impedance.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross,   OBSERVED: Circle,   SOLUTION: Square.
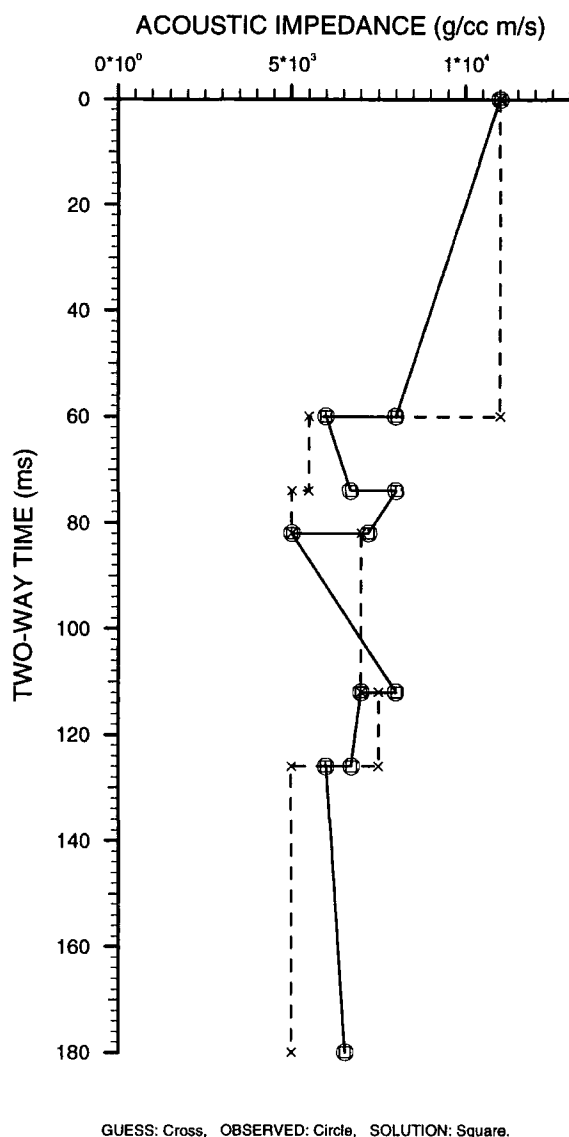
Figure 5.13. The impedance profiles when the initial guess has two reversals at boundaries 2 and 3, and a non-zero observed impedance gradient vector.

5.5.2.3. The effect of noise on impedance inversion

Quantitatively inverting for impedance depends largely on the magnitude of the reflection across boundaries. The addition of random noise to the observed trace will significantly effect the amplitude of the reflection. Thus, for any amount of noise contained in the observed trace, we should not expect the value of the inverted impedance to converge to its correct value. But it is important to investigate for what signal-to-noise ratios could we obtain reasonable estimates of the observed impedance profile, so in this section we consider the impedance inversion example shown in Figure 5.10 for signal-to-noise ratios of 4, 2, and 1. In all cases the impedance of the first layer is equality constrained to its correct value of $11000 \ \mathrm{gcm^{-3}ms^{-1}}$.

102

The input and output data for the case when the signal-to-noise ratio is 4 are shown in Table 5.10.

layer number	observed boundary locations (ms)	observed starting impedances ($gcm^{-3}ms^{-1}$)	solution starting impedances ($gcm^{-3}ms^{-1}$)	initial starting impedances ($gcm^{-3}ms^{-1}$)	observed impedance gradients ($gcm^{-3}ms^{-1}$/sample)
n	t_{obs}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	11000	11000	11000	0
2	74	6000	6168	5500	0
3	82	8000	7993	9000	0
4	112	5000	4182	7000	0
5	126	7000	6685	7500	0
6	256	6000	6321	5000	0

Table 5.10. The data for the impedance parameters inversion when the observed trace has a signal-to-noise ratio of 4. The impedance solutions for this example are shown in Figure 5.14.

The impedance profiles for this example are shown in Figure 5.14. The relative error energy for the initial impedance guess E_{rel}= 47%, and for the final solution E_{rel}= 22%, which was reached in four iterations. It can be observed from Figure 5.14 that the impedance solution for the shallow layers resembles the observed impedance more closely than the deeper layers impedance solution. This can be attributed to the high impedance contrast between the shallow layers than between the deeper ones. That is, the added random noise effects deeper layers, which have weaker reflections, more than the shallower reflections which have stronger reflections. However, no polarity reversals were introduced in the final solution, and that the impedance trends are still reasonably preserved, i.e. the second and last layers have close impedance values and the fourth layer has the lowest impedance.

The same example was repeated with the signal-to-noise ratio in the observed trace being 2. the data for this example are given in Table 5.11, and the impedance profiles are given in Figure 5.15. The initial E_{rel}= 54% and the inversion program performed four iterations to reduce E_{rel} to 48%.

103

ACOUSTIC IMPEDANCE (g/cc m/s)
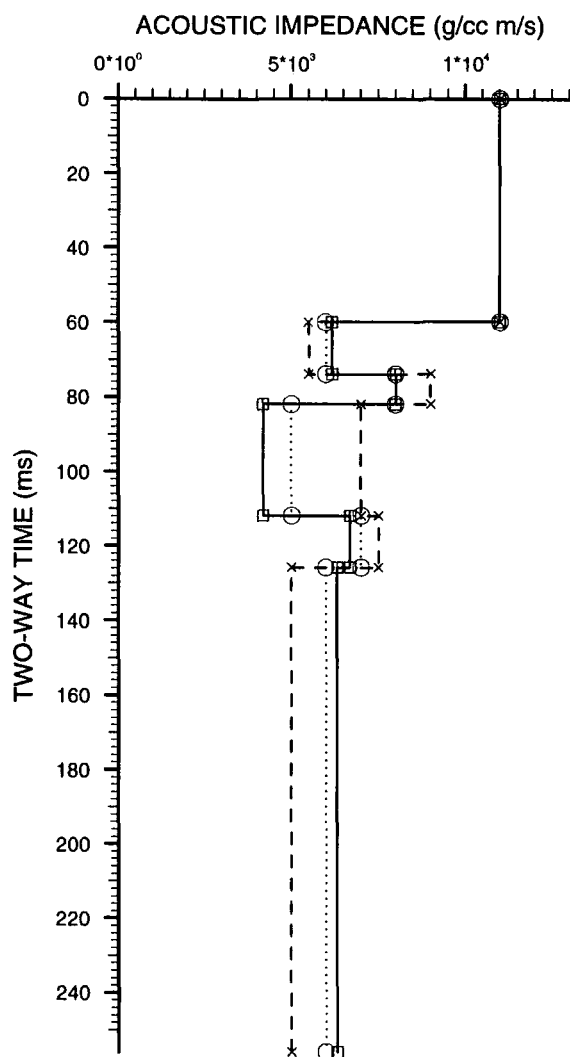
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.14. The impedance profiles for the data in Table 5.10. The signal-to-noise ratio of the observed trace is 4. In this inversion for impedance the solution is closer is closer to the correct impedance values for the shallow layers where there is a high impedance contrast across their boundaries

layer number	observed boundary locations (ms)	observed starting impedances (gcm^{-3}ms^{-1})	solution starting impedances (gcm^{-3}ms^{-1})	initial starting impedances (gcm^{-3}ms^{-1})	observed impedance gradients (gcm^{-3}ms^{-1}/sample)
n	t_{obs}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	11000	11000	11000	0
2	74	6000	5194	5500	0
3	82	8000	7010	9000	0
4	112	5000	4358	7000	0
5	126	7000	5462	7500	0
6	256	6000	4452	5000	0

Table 5.11. The data for the impedance parameters inversion when the observed trace has a signal-to-noise ratio of 2. The impedance solutions for this example are shown in Figure 5.15.

104

From the impedance profiles of Figure 5.15 we notice that no polarity reversals occurred, the best impedance estimates are for the shallow layers with high reflection coefficients, and that the solution profile impedance trends are deteriorating as compared to the correct, observed, impedance profile. For example, the impedance value for the fourth layer is comparable to that of the sixth layer, which is the trend in the observed profile, even though the fourth layer has less impedance than its correct value. Also, notice that the impedance of the fifth layer has become considerably smaller.
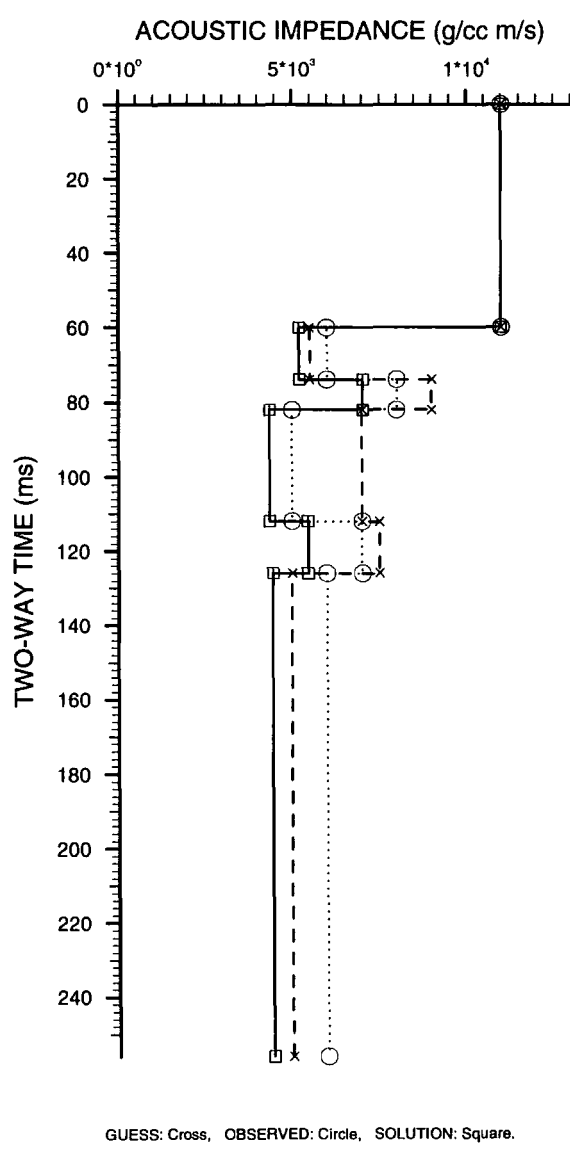
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.15. The impedance profiles for the data in Table 5.11. The signal-to-noise ratio of the observed trace is 2. Comparing the impedance solution with that of Figure 5.14, where the signal-to-noise ratio is 4, we notice that the impedance trend of the solution profile has deteriorated for the deeper layers. But no polarity reversals resulted across any of the interfaces.

Finally, Table 5.12 shows the input and output data when the signal-to-noise ratio of the observed trace is 1. Figure 5.16 shows the corresponding impedance profiles.

layer number	observed boundary locations (ms)	observed starting impedances (gcm^{-3}ms^{-1})	solution starting impedances (gcm^{-3}ms^{-1})	initial starting impedances (gcm^{-3}ms^{-1})	observed impedance gradients (gcm^{-3}ms^{-1}/sample)
n	t_{obs}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	11000	11000	11000	0
2	74	6000	5734	5500	0
3	82	8000	7724	9000	0
4	112	5000	5607	7000	0
5	126	7000	8779	7500	0
6	256	6000	11279	5000	0

Table 5.12. The data for the impedance parameters inversion when the observed trace has a signal-to-noise ratio of 1. The impedance solutions for this example are shown in Figure 5.16.

The inversion program performed seven iterations to reduce E_{rel} from 124% to 86%. The impedance solution profile in Figure 5.16 shows the extent of deterioration to the impedance trend that can occur when the signal-to-noise ratio is low. Layer 6, which is of low impedance that is comparable to shale type lithology, has its impedance greatly increased so that now it could be interpreted as limestone. There is a polarity reversal between layers 6 and 5. Layers 2 and 5 almost have the same impedance value. Obviously, it is the deeper layers with low impedance contrast that had their impedance values differ the most from their correct impedance.

5.5.2.4 Inverting for impedance with slightly incorrect boundaries

The examples of sub-section 5.5.2.2 have illustrated that if no noise is present in the observed trace then convergence to the correct solution is almost always possible, even in the presence of polarity reversals. But, up to now, when solving for impedance, all the boundary locations and wavelet parameters have the correct values. In the next two examples we investigate impedance inversion with incorrect boundary locations, while having correct wavelet parameters, and the observed trace is noise-free.
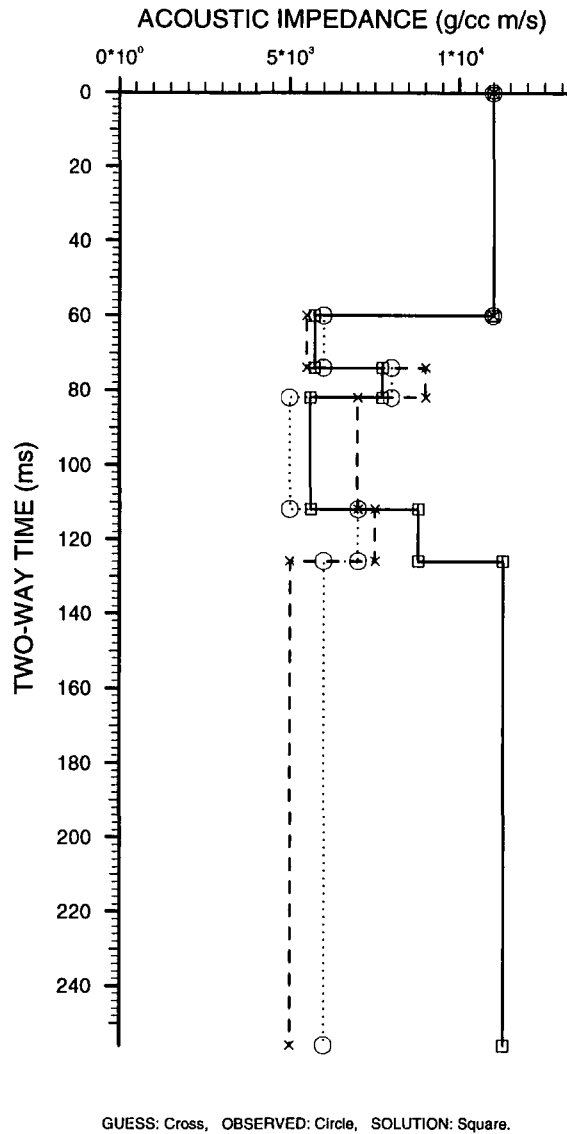
ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.16. The impedance profiles for the data in Table 5.12. The signal-to-noise ratio of the observed trace is 1. In this inversion for impedance the solution profile differs greatly from the observed profile at layers 5 and 6 at which we have a small impedance contrast. At the boundary between layers 5 and 6 there is a polarity reversal in the solution profile.

With incorrect boundary locations one should not expect to converge to the correct impedance solution, but we would like to examine the optimum solution obtained. A simple approach to this problem is to use the correct solution as the initial guess except that only one of the boundary locations is perturbed by a small value, say 4 ms, and observe the resulting optimum impedance solutions.

An example where the fifth boundary location is perturbed by 4 ms is shown in Table 5.13. The first layer impedance is fixed at its correct value of 11000 gcm^{-3}ms^{-1}. In this example the initial E_{rel}= 2.6%.

107

layer number	observed boundary locations (ms)	initial boundary locations (ms)	observed starting impedances (gcm^{-3}ms^{-1})	solution starting impedances (gcm^{-3}ms^{-1})	initial starting impedances (gcm^{-3}ms^{-1})	observed impedance gradients (gcm^{-3}ms^{-1}/sample)
n	t_{obs}	t_{ini}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	60	11000	11000	11000	0
2	74	74	6000	5999	6000	0
3	82	82	8000	8011	8000	0
4	112	112	5000	5021	5000	0
5	126	130	7000	7483	7000	0
6	180	180	6000	6989	6000	0

Table 5.13. Inversion for impedance when the initial guess has boundary location 5 perturbed by 4 ms, i.e. 2 samples.

The resulting optimum impedance solution is shown in Figure 5.17a, which was reached in three iterations with $E_{rel}= 1.5\%$. It can be seen that for the inversion process to accommodate the 4 ms difference in the fifth boundary, it had to increase the reflection coefficient at the boundary at 112 ms and decrease the reflection coefficient at the boundary that is wrongly located at 130m. This it did by increasing the impedance contrast at the 112 ms boundary and decreasing the impedance contrast at the 130 ms boundary. To understand the reason for this, we need to look at Figure 5.17b, where it can be seen that by shifting the boundary location from 126 ms to 130 ms, we have shifted the reflection to a lower location so that it is not in alignment with the corresponding reflection energy in the correct solution. To correct this, in order to reduce error energy, the inversion process reduced the reflection energy from the initial guess in the lower boundary. The only way for the inversion process to do this was by reducing the reflection coefficient, thus reducing the impedance contrast across the boundary. But by reducing the reflection coefficient across the boundary at 130 ms, the constructive interference zone between the boundaries at 112 ms and 130 ms also required the reflection coefficient at 112 ms to be increased. Of course this has to be done in an optimum way and there will be a limit to how much the error can be reduced, so a minimum error energy was still present as shown by the error traces of Figure 5.17c, which is a display of the error traces for each of the iterations (the first error trace belongs to the initial guess). It is important to note that the error is concentrated around the 126-130 ms interval.

ACOUSTIC IMPEDANCE (g/cc m/s)
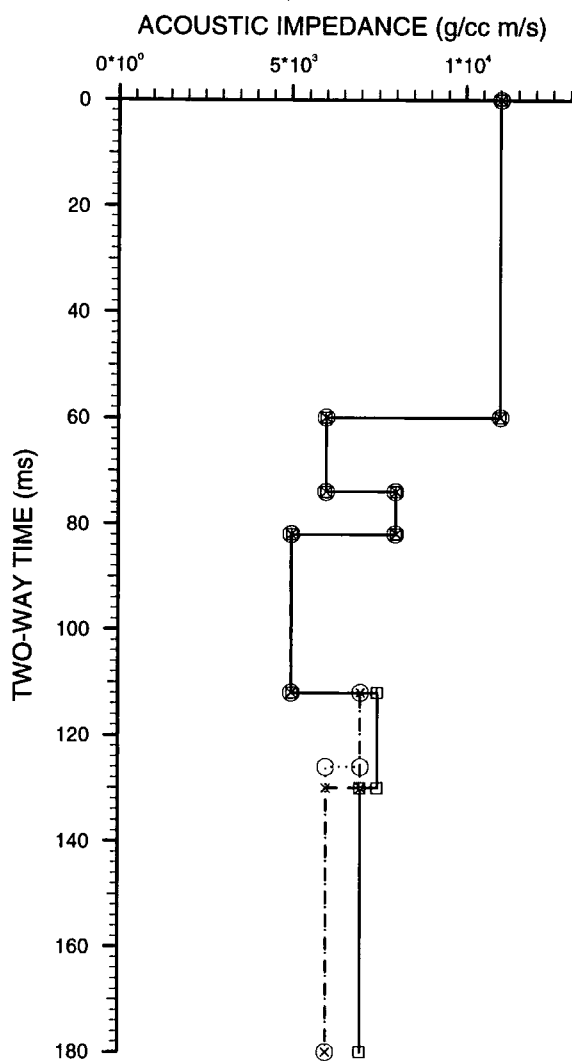
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.17a. Impedance profiles for impedance inversion with slightly incorrect boundaries. Note that the impedance solution is in error only across the boundary that is incorrectly located.

Figure 5.17b. The solution traces after each iteration in solving the example of Figure 5.17a. Traces numbered 1 and 7 are the observed trace, while trace 2 is the seismic response of the initial guess and traces 3-6 are the seismic response after each iteration. Note that the vertical axis represents the two-way travel-time in ms.

Figure 5.17c. The error traces corresponding to the iterations of Figure 5.17b. Trace 1 is the error trace of the initial guess. Note that the vertical axis represents the two-way travel-time in ms.

5.5.2.5. Impedance inversion with grossly incorrect boundaries

In this case the boundary locations could be incorrect by as much as the radius of convergence. Similar to the previous example, the fifth boundary was perturbed by 10 ms, which is about equal to the radius of the convergence region. The input data and impedance inversion solution are given in Table 5.14.

layer number	observed boundary locations (ms)	initial boundary locations (ms)	observed starting impedances (gcm^{-3}ms^{-1})	solution starting impedances (gcm^{-3}ms^{-1})	initial starting impedances (gcm^{-3}ms^{-1})	observed impedance gradients (gcm^{-3}ms^{-1}/sample)
n	t_{obs}	t_{ini}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	60	11000	11000	11000	0
2	74	74	6000	6000	6000	0
3	82	82	8000	8006	8000	0
4	112	112	5000	5011	5000	0
5	126	136	7000	7255	7000	0
6	180	180	6000	8094	6000	0

Table 5.14. Showing that the boundary location 5 is perturbed by 10 ms, i.e. 5 samples, in the initial guess.

Similar to the previous example, the input impedance is the same as that of the observed impedance, and the impedance of the first layer is fixed at 11000 gcm^{-3}ms^{-1}.

111

The initial E_{rel}= 7%, and after three iterations the program reached an optimum solution with E_{rel}= 0.8%. Figure 5.18a shows that by perturbing the interface by 10 ms, which is equivalent to about one half cycle of the seismic wavelet, means that the reflection energy at the synthetic seismogram is out of phase as compared to the corresponding correct solution. To accommodate this the inversion program had to reverse the polarity of the reflection coefficient, thus reversing the impedance contrast trend across the perturbed boundary. Figure 5.18b shows that the inversion process has achieved this by increasing the impedance value of the lower layer. Note that a small increase in impedance contrast across the boundary above the perturbed one was also necessary in this example to accommodate the polarity reversal at the fifth boundary.
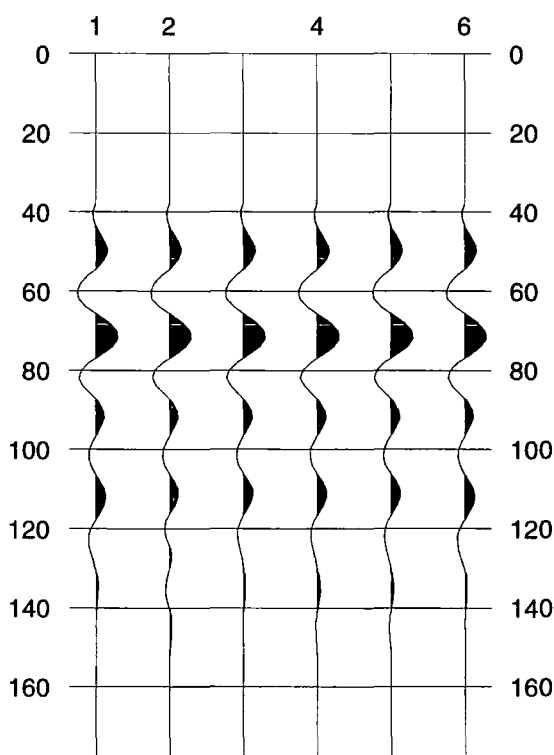
Figure 5.18a. Solution traces when the boundaries are grossly incorrect. Traces numbered 1 and 6 are the observed trace, while trace 2 is the seismic response of the initial guess and traces 3-5 are the seismic response after each iteration. Note that the vertical axis represents the two-way travel-time in ms.

5.5.3. Inversion of both impedance and boundaries

The synthetic examples of section 5.5.2 are more representative of inversion of field data since it is more likely that the initial guess will be in error in both impedance and boundary locations. However, to invert only for impedance when the boundary locations are incorrect will not give the correct impedance results. One way to solve this inversion problem is to invert first for impedance and then boundary

locations; then, if convergence is not achieved, repeat the sequence as many times as necessary until the program terminates, with any of the convergence criteria, for both problems.
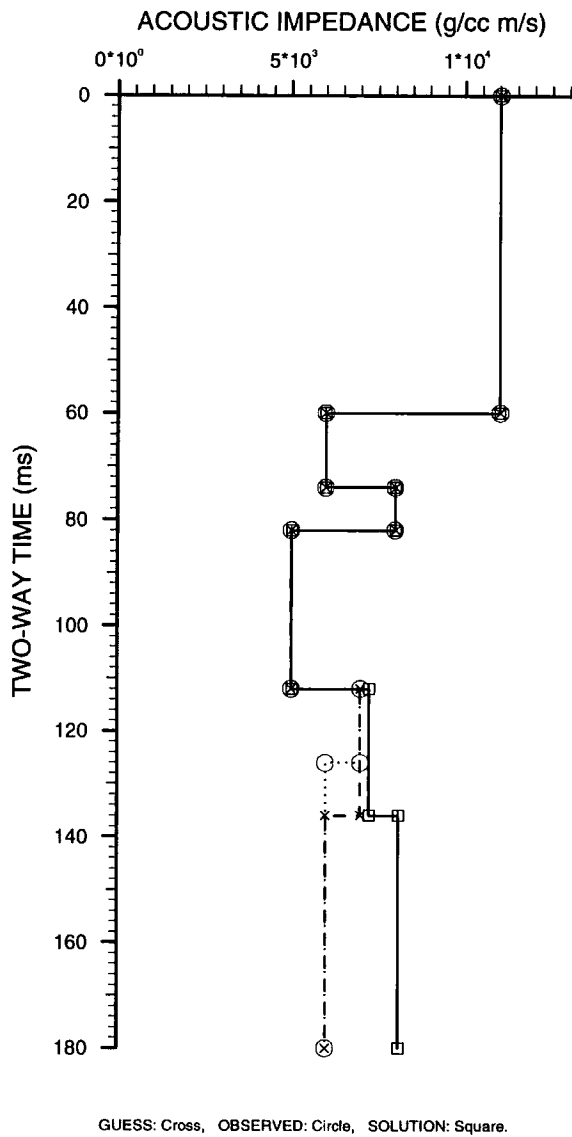
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.18b. The impedance profiles when the boundaries are grossly incorrect. The solution profile has a polarity reversal at the incorrect boundary.

5.5.3.1 Example of impedance and boundary inversion

In this example four of the five boundary interfaces of our observed earth model have been perturbed. Two interfaces were perturbed by up to 4 ms. The first layer impedance is fixed at 11000 gcm^{-3}ms^{-1}, and all the other layers had their impedance perturbed. The input data and output results are shown in Table 5.15.

layer number n	observed boundary locations (ms) t_{obs}	solution boundary locations (ms) t_{sol}	initial boundary locations (ms) t_{ini}	observed starting impedances (gcm^{-3}ms^{-1}) x_{obs}	solution starting impedances (gcm^{-3}ms^{-1}) x_{sol}	initial starting impedances (gcm^{-3}ms^{-1}) x_{ini}	observed impedance gradients (gcm^{-3}ms^{-1}/sample) y_{obs}
1	60	60	56	11000	11000	11000	0
2	74	74	72	6000	6000	7000	0
3	82	82	82	8000	8000	9500	0
4	112	112	108	5000	5000	4000	0
5	126	126	128	7000	7000	6000	0
6	180	180	180	5000	5000	4500	0

Table 5.15. Input and output data for impedance and boundaries inversion. The correct boundaries and impedance solutions were achieved in four impedance inversion runs and three boundary inversion runs.

The initial E_{rel} was 50%. It took four impedance inversions and three boundary inversions to reach the zero error energy final solution given in Table 5.15. The progression to this final solution is summarised in Table 5.16.

Run number	1	2	3	4	5	6	7
Inversion type	Imp	Bnd	Imp	Bnd	Imp	Bnd	Imp
Error energy (%)	22	10	5.2	4.95	1.8	1	0
Max. impedance	21246	21246	16278	16278	9372	9372	8000

Table 5.16. Impedance solution progression when inverting for boundaries and impedance. The maximum impedance values shown are those for layer 3. Imp means impedance inversion run, and Bnd is boundary inversion run.

One important observation from Table 5.16 is the value of maximum impedance for one of the impedance variables, namely for layer 3, has reached 21246 gcm^{-3}ms^{-1}. Such an unrealistically high impedance value occurred because the first impedance inversion run had to accommodate the fourth layer which was initially guessed to be too thin. This was corrected after the right layer thickness was retained in the boundary inversion at run 6.

5.5.3.2. Impedance and boundary inversion with constraints

The inversion example in the last sub-section showed that unrealistically high impedance values can sometimes arise during repeated inversion runs for boundary locations and impedance. One way to avoid such high impedance values is to impose an upper bound on the impedance allowed in each layer, that is we make use of inequality constraints.

The previous example was rerun, but this time an upper bound of 12000 $gcm^{-3}ms^{-1}$ was globally imposed on all the layers. Convergence to the zero error energy solution was reached in two impedance inversion runs and only one boundary inversion run. The progression to this solution is summarised in Table 5.17.

Run number	1	2	3
Inversion type	Imp	Bnd	Imp
Error energy (%)	30	9	0
Max. impedance	12000	12000	8000

Table 5.17. Impedance solution progression for the same problem in Table 5.16 after imposing a global upper bound of 12000 $gcm^{-3}ms^{-1}$ on all the layers.

As expected, the maximum impedance value imposed was not exceeded which made it possible for the boundary locations to converge in one inversion run. Obviously, imposing upper bounds on the impedance values contributed significantly to the stability of the problem, thus reducing the number of iterations required for convergence.

5.5.3.3. The effect of noise on impedance and boundaries inversion

The first test of impedance and boundaries inversion when the observed trace is contaminated with noise is summarised in Table 5.18. The signal-to-noise ratio for these data is 10.

layer number n	observed boundary locations (ms) t_{obs}	solution boundary locations (ms) t_{sol}	initial boundary locations (ms) t_{ini}	observed starting impedances ($gcm^{-3}ms^{-1}$) x_{obs}	solution starting impedances ($gcm^{-3}ms^{-1}$) x_{sol}	initial starting impedances ($gcm^{-3}ms^{-1}$) x_{ini}	observed impedance gradients ($gcm^{-3}ms^{-1}$/sample) y_{obs}
1	60	60	56	11000	11000	11000	0
2	74	72	72	6000	5824	7000	0
3	82	84	82	8000	8915	9500	0
4	112	112	108	5000	6000	4000	0
5	126	126	128	7000	7456	6000	0
6	256	256	256	5000	4930	4500	0

Table 5.18. Input and output data for impedance and boundaries inversion when the signal-to-noise ratio is 10. The final boundaries and impedance solutions were achieved in four impedance inversion runs and three boundary inversion runs. The progression toward the final solution is summarised in Table 5.19.

For this test the impedance of the first layer was equality constrained to its correct value of 11000 $gcm^{-3}ms^{-1}$, and an upper bound of 11000 $gcm^{-3}ms^{-1}$ and a lower bound of 4000 $gcm^{-3}ms^{-1}$ were imposed globally on all the other layers. The initial E_{rel}= 68%, then after four impedance inversions and three boundary inversions E_{rel} was reduced to 8%. The progression toward the solutions t_{sol} and x_{sol} is summarised in Table 5.19. One important observation in Table 5.19 is the high number of iterations for the impedance inversion runs as compared to the boundaries inversion runs. This is mainly due to additions and deletions of bounding constraints during each run of impedance inversion.

Run number	0	1	2	3	4	5	6	7
Inversion type	-	Imp	Bnd	Imp	Bnd	Imp	Bnd	Imp
Error energy (%)	68	56	24	13	12	9	8.8	7.7
Number of iterations	-	9	4	8	2	6	2	5

Table 5.19. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 10. The solution results are given in Table 5.18. Imp means impedance inversion run, and Bnd is boundary inversion run.

The impedance profiles of the first impedance run are shown in Figure 5.19 which illustrates how the impedances of the deepest two layers have decreased while the impedance of the shallower layers have generally increased, thus changing the impedance trend as compared to the observed profile.

The impedance profiles of the first impedance run are shown in Figure 5.20 where it can be noticed that all boundaries have converged to their correct locations except for boundaries 2 and 3. One possible reason for this is that the enclosed layer, i.e. layer 3, has a thickness that is very close to the tuning thickness for the given seismic wavelet, where non-unique solutions could exist.

The impedance profiles for the final impedance run are shown in Figure 5.21, where it can be observed that despite of the impedance of layer 3 being higher than its correct value (which was accommodated by the inversion program by increasing the impedance of layer 4 and slightly increasing the impedance of layer 5), the general trend of the solution impedance profile reasonably represents that of the observed profile.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.19. The impedance profiles of the first impedance inversion run of Table 5.19. No polarity reversals were introduced, but the impedance trend of the solution profile have largely changed as compared to the observed impedance profile.

117

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.20. The impedance profiles of the first boundaries inversion run of Table 5.19. Note that in this run only boundaries 2 and 3 did not converge to their correct locations

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.21. The impedance profiles of the final impedance inversion run of Table 5.19. Note that only the impedance of layers 3 and 4 differ greatly from their correct impedance values, the other layers have their solution impedance close to their correct values.

The previous test was re-run with a signal-to-noise ratio of 2 in the observed trace. The input and output data are given in Table 5.20, and a summary of the progression toward the final impedance profile solution is given in Table 5.21.

layer number	observed boundary locations (ms)	solution boundary locations (ms)	initial boundary locations (ms)	observed starting impedances ($gcm^{-3}ms^{-1}$)	solution starting impedances ($gcm^{-3}ms^{-1}$)	initial starting impedances ($gcm^{-3}ms^{-1}$)	observed impedance gradients ($gcm^{-3}ms^{-1}$/sample)
n	t_{obs}	t_{sol}	t_{ini}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	58	56	11000	11000	11000	0
2	74	74	72	6000	5447	7000	0
3	82	82	82	8000	8433	9500	0
4	112	112	108	5000	4908	4000	0
5	126	126	128	7000	6837	6000	0
6	256	256	256	5000	5125	4500	0

Table 5.20. Input and output data for impedance and boundaries inversion when the observed trace has a signal-to-noise ratio of 2. The final boundaries and impedance solutions were achieved in two impedance inversion runs and one boundary inversion run. The progression toward the final solution is summarised in Table 5.21.

Run number	0	1	2	3	4	5	6	7
Inversion type	-	Imp	Bnd	Imp	Bnd	Imp	Bnd	Imp
Error energy (%)	68	56	24	13	12	9	8.8	7.7
Number of iterations	-	9	4	8	2	6	2	5

Table 5.21. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 2. The solution results are given in Table 5.20. Imp means impedance inversion run, and Bnd is boundary inversion run.

Despite the considerably lower signal-to-noise ratio in this test as compared to the pervious one, the final impedance profiles given in Figure 5.22 show that all the boundaries have converged to their correct locations, except the first boundary which remained away from its correct location by one sample interval. This resulted in an impedance solution profile that closely resembles the observed one.

One conclusion that could be drawn from this test is that the distribution of the random noise in the observed trace is probably more important than the amount of noise energy present. That is, if there is more noise energy present at the intervals where there is a small reflection energy, or a layer that is close to its tuning thickness, then this noise energy will have a large effect on the final solution.
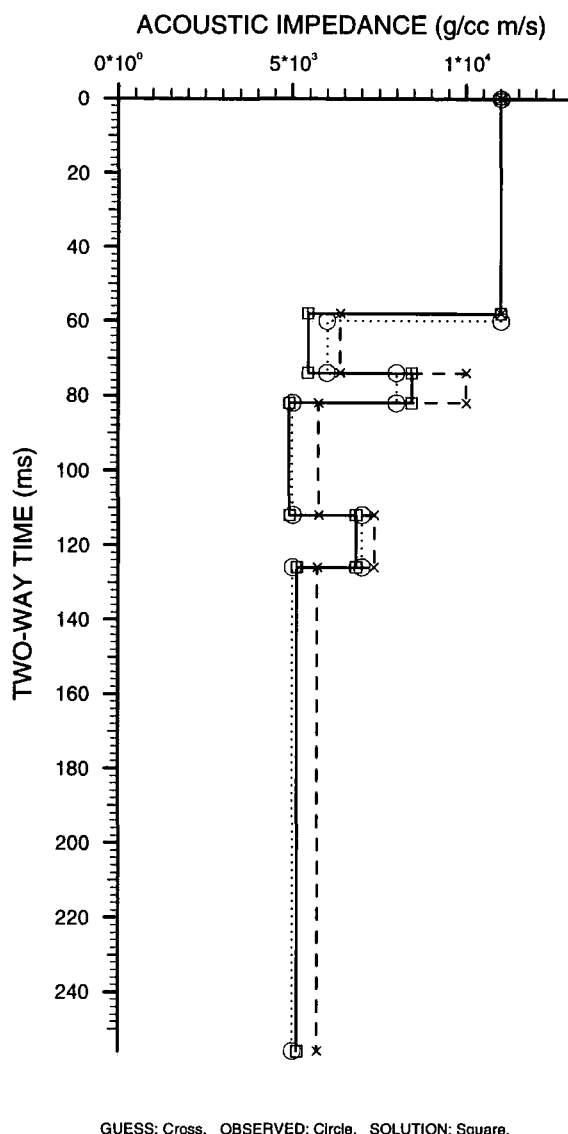
ACOUSTIC IMPEDANCE (g/cc m/s)

0*10⁰   5*10³   1*10⁴

TWO-WAY TIME (ms)

0 20 40 60 80 100 120 140 160 180 200 220 240

GUESS: Cross,   OBSERVED: Circle,   SOLUTION: Square.

Figure 5.22. The impedance profiles of the final impedance inversion run of Table 5.21. Even though the signal-to-noise for this test is much lower than the results of Figure 5.21, the final impedance solution obtained here is better.

5.5.4. Wavelet parameters inversion

Inverting for wavelet parameters in the following synthetic examples will be divided into three sets, namely the four frequency parameters, the two amplitude parameters, and the first two phase parameters, which are the constant phase parameter and the linear phase parameter. The third, or quadratic phase parameter, will always be fixed at zero; i.e. we are making the assumption that no wavelet dispersion is taking place. When inverting for any one set of parameters, the other parameters are kept fixed, i.e. equality constrained, except for the frequency parameters, which cannot be constrained neither equality nor inequality (see section 4.5). The reason for this is that the wavelet parameters inversion problem is very poorly conditioned, which results in a poor rate of convergence. The observed earth

model given in Table 5.1 and the observed wavelet w_{obs}= [24 28 55 84 115000 115000 0.418 0.113 0]^T in section 5.5 will be used to generate the observed seismic trace.

To invert for the frequency parameters an initial guess wavelet was generated by perturbing the observed wavelet to w_{ini}= [10 33 60 100 115000 115000 0.418 0.113 0]^T. This wavelet gave initial E_{rel} of 6%. After six iterations convergence to the correct (observed) wavelet with zero error energy was obtained.

When inverting for the amplitude parameters the initial guess wavelet was w_{ini}= [24 28 55 84 120000 110000 0.418 0.113 0]^T. The initial E_{rel}= 0.52%, and after 36 iterations convergence to the wavelet solution w_{sol}= [23.9 27.9 56 84.1 113197 110013 0.418 0.113 0]^T was obtained with error energy value of 33, which was equivalent to E_{rel}= 0.000012%.

The constant phase parameter ϕ_0 and the linear phase parameter ϕ_l were inverted for separately, that is, when inverting for one the other parameter was fixed at its correct value. When inverting for ϕ_0 the initial wavelet was w_{ini}= [24 28 55 84 115000 115000 0.0 0.113 0]^T which gave an initial E_{rel}= 18%. Convergence to the correct solution, with zero error energy, occurred after 143 iterations. When inverting for the linear phase parameter ϕ_l the initial guess wavelet used was w_{ini}= [24 28 55 84 115000 115000 0.4138 0.12 0]^T. This wavelet had E_{rel}= 11%, and convergence to the correct solution with E_{rel}= 0% was obtained after 55 iterations.

5.5.5. Inversion for impedances, boundaries and wavelet parameters

In this case all the variable types, namely impedance, boundary locations and wavelet, are considered to be incorrect. It was found that the best approach for this problem is to first invert for the impedance parameters then the boundary location parameters, and then repeat this sequence until convergence is achieved. Next, we invert for the wavelet parameters, then repeat the impedance-boundaries inversion sequence until again convergence is achieved. We then keep repeating the inversion for wavelet parameters and seek convergence of the impedance-boundaries inversion until we judge that the improvement in the obtained solution does not justify any extra computer time.

In the above approach, when inverting for the wavelet parameters the linear phase parameter ϕ_l should not be allowed to vary significantly, indeed it should only be allowed to vary so that the time shift it produces is within the interval ±½ sample interval. The privilege of producing a time shift that is close to a sample interval should be reserved to the boundary locations inversion. This tends to numerically stabilise the inversion process by limiting the non-uniqueness of the solution. Furthermore, the constant phase term ϕ_0, should also be restricted by using inequality

constraints to impose upper and lower bounds to a limited interval about $\phi_0=0$. In this way we prevent the solution from catapulting away from the nearly zero phase wavelet that the observed trace contains. It was found that when the constant phase term changes substantially from its near zero value, the boundary locations will tend not to converge to their correct solution. This implies that one should have as good an estimate of the wavelet as possible, and only allow the wavelet phase parameters to vary as little as possible.

5.5.5.1. Example of inversion for impedances, boundaries and wavelet parameters

For this example we use the observed and initial guess impedance and boundary locations data given in Table 5.22, which were used previously in section 5.5.3.1. For the observed wavelet we use the wavelet of section 5.5.4, which is $w_{obs}=$ [24 28 55 84 115000 115000 0.418 0.113 0]^T, and from which we generate an initial guess wavelet $w_{ini}=$ [22 35 60 90 115000 115000 0.3 0.115 0]^T.

layer number	observed boundary locations (ms)	solution boundary locations (ms)	initial boundary locations (ms)	observed starting impedances (gcm^{-3}ms^{-1})	solution starting impedances (gcm^{-3}ms^{-1})	initial starting impedances (gcm^{-3}ms^{-1})	observed impedance gradients (gcm^{-3}ms^{-1}/sample)
n	t_{obs}	t_{sol}	t_{ini}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	60	56	11000	11000	11000	0
2	74	72	72	6000	6880	7000	0
3	82	84	82	8000	10746	9500	0
4	112	112	108	5000	7254	4000	0
5	126	126	128	7000	10011	6000	0
6	180	180	180	5000	6957	4500	0

Table 5.22. Example of inversion for impedance, boundaries and wavelet parameters.

Except for the impedance of the first layer, which was equality constrained at 11000 gcm^{-3}ms^{-1}, all impedance parameters had imposed on them lower and upper bounds of 3500 and 12000 gcm^{-3}ms^{-1}, respectively. The constant phase parameter of the wavelet was restricted to be in the interval [0, 0.5] radians, and the linear phase parameter was restricted to vary between 0.111 and 0.117 radians/Hz.

The initial guess gave a relative error energy of 63%. The first impedance inversion reduced E_{rel} to 36%, and during which the third impedance reached the 12000 gcm^{-3}ms^{-1} bound, so that the corresponding inequality constraint became active and was added to the active set. The first inversion for the boundaries reduced E_{rel} to 11%, and only two boundaries, the second and the third, did not converge to their correct values by a sample interval. In the second impedance inversion the active constraint was deleted from the active set, and the E_{rel} was reduced to 1.5%. The

second inversion for the boundaries failed to reduce the error energy, so E_{rel} remained at 1.5%.

The next inversion was for the wavelet parameters which reduced E_{rel} to 1.14%. Then inversion for the boundary locations failed again to reduce E_{rel}. The next impedance inversion reduced E_{rel} to 1.11%.

A further inversion for the boundaries, after a second inversion for the wavelet parameters, did not reduce the error energy. It seemed that the boundary locations have converged, but unfortunately not to the correct locations.

Several alternating impedance and wavelet parameter inversions were performed until E_{rel} was 1.096%. The final boundary locations and impedance values are given in Table 5.22 , and the final wavelet parameter solution was $w_{sol}=$ [18.1 36.6 60.9 82.9 115000.3 14999.9 0.4497 0.111 0]^T.

5.5.5.2. Second example of inversion for impedances, boundaries and wavelet parameters

In the last example, convergence to the correct solution was not achieved. The reason was that layer 3 became thinner than its correct thickness, and the boundary inversion was not able to correctly adjust this thickness. An obvious reason for this is that the impedance of the same layer has increased to 10746 $gcm^{-3}ms^{-1}$, which resulted in convergence to a local minimum other than the global one. The remedy for this is to prevent the inversion process from moving into a region that contains a local minimum. To do this, we notice that the initial guess impedance of layer 3 is 9500 $gcm^{-3}ms^{-1}$, and we could judge, possibly from information known *a priori*, that such a value is already high enough, so it should constitute an upper bound. Thus for this example we solve the same problem as the last example, only this time we impose an upper bound of 9500 $gcm^{-3}ms^{-1}$ on all layers, except the first. The data for this example are given in Table 5.23.

layer number	observed boundary locations (ms)	solution boundary locations (ms)	initial boundary locations (ms)	observed starting impedances $(gcm^{-3}ms^{-1})$	solution starting impedances $(gcm^{-3}ms^{-1})$	initial starting impedances $(gcm^{-3}ms^{-1})$	observed impedance gradients $(gcm^{-3}ms^{-1}/sample)$
n	t_{obs}	t_{sol}	t_{ini}	x_{obs}	x_{sol}	x_{ini}	y_{obs}
1	60	60	56	11000	11000	11000	0
2	74	74	72	6000	6007	7000	0
3	82	82	82	8000	8093	9500	0
4	112	112	108	5000	5013	4000	0
5	126	126	128	7000	7027	6000	0
6	180	180	180	5000	4975	4500	0

Table 5.23. The second example of inversion for impedance, boundaries and wavelet parameters.

We started by inverting for impedance. This time the initial E_{rel} was reduced from 63% to 45%, layer 3 stayed at the upper impedance value of 9500 $gcm^{-3}ms^{-1}$, and every other layer, except the first, had less impedance value. A first inversion for the boundary locations reduced E_{rel} to 12%, and not all the boundaries converged to their correct locations. Then a first inversion for impedances reduced E_{rel} to 5%, and layer 3 still had the highest impedance value of 9500 $gcm^{-3}ms^{-1}$. A second inversion for boundaries reduced E_{rel} to 2.85%, and all but two boundaries (the top and bottom of layer 3) did not converge to their correct locations. A second inversion for impedances reduced E_{rel} to 1.75%, and layer 3 was still at 9500 $gcm^{-3}ms^{-1}$. The third inversion for boundaries failed to reduce E_{rel}.

The next step was to invert for the wavelet parameters. This wavelet inversion reduced E_{rel} to 1.31%. Then inversion for boundaries reduced E_{rel} to 0.58%, and this time the boundaries converged to their correct locations. Next inversion for impedances reduced E_{rel} to 0.3%, during which the constraint defining the upper bound on layer 3 was deleted from the active set, and the impedance for layer 3 became 8690 $gcm^{-3}ms^{-1}$.

Several wavelet-impedance inversions later reduced E_{rel} to a mere 0.01%. By then it was possible to stop the inversion and the final impedance solution is given in Table 5.23. The wavelet parameters converged to $w_{sol}=$ [18.1   36.6   60.9   82.9 115000.3 14999.9 0.4497 0.111 0]^T.

5.5.5.3. The effect of noise on inversion for impedance, boundaries and wavelet parameters

The same example in the previous section is repeated in this section with signals-to-noise ratios of 4, 2 and 1 added to the observed trace. Another change in this example is the linear phase term in the initial guess wavelet is equality constrained to its correct value of 0.1 radians/Hz, so that the initial guess wavelet

$\mathbf{w}_{ini}$= [22 35 60 90 115000 115000 0.3 0.1 0]^T, while the observed wavelet $\mathbf{w}_{obs}$= [24 28 55 84 115000 115000 0.4138 0.113 0]^T. The input and output impedance and boundary locations data are given in Table 5.24, and a summary of the progression towards the final solution is given in Table 5.25.

layer number n	observed boundary locations (ms) $\mathbf{t}_{obs}$	solution boundary locations (ms) $\mathbf{t}_{sol}$	initial boundary locations (ms) $\mathbf{t}_{ini}$	observed starting impedances (gcm^{-3}ms^{-1}) $\mathbf{x}_{obs}$	solution starting impedances (gcm^{-3}ms^{-1}) $\mathbf{x}_{sol}$	initial starting impedances (gcm^{-3}ms^{-1}) $\mathbf{x}_{ini}$	observed impedance gradients (gcm^{-3}ms^{-1}/sample) $\mathbf{y}_{obs}$
1	60	60	56	11000	11000	11000	0
2	74	74	72	6000	5343	7000	0
3	82	82	82	8000	7941	9500	0
4	112	112	108	5000	5352	4000	0
5	126	126	128	7000	7560	6000	0
6	256	256	256	5000	5500	4500	0

Table 5.24. The input and output data for the example of inversion for impedance, boundaries and wavelet parameters when the signal-to-noise ratio in the observed trace is 4. The summary of the progression towards the solutions is given in Table 5.25.

Run number	0	1	2	3	4	5	6	7	8	9
Inversion type	-	Imp	Bnd	Imp	Wvl	Imp	Wvl	Imp	Wvl	Imp
Error energy (%)	99	80	40	26	23.08	23	22.98	22.975	22.967	22.691
Number of iterations	-	4	4	10	4	4	4	2	3	2

Table 5.25. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 4. The solution results are given in Table 5.24. Imp means impedance inversion run, Bnd is boundary inversion run, and Wvl means a wavelet parameters inversion run.

As Table 5.25 illustrates, the initial E_{rel}= 99%, the first impedance run reduced E_{rel} to 80% in four iterations. The impedance profiles for this run are shown in Figure 5.23. The next inversion run was for boundary locations which reduced E_{rel} to 40% in four iterations. The impedance profiles for this boundaries inversion run are shown in Figure 5.24 where it can be observed that all the boundaries have converged to their correct locations. Another impedance inversion run reduced E_{rel} to 26%.

Because the boundary locations are already at their correct locations, the next inversion runs are wavelet parameters and impedances. The first wavelet parameters inversion reduced E_{rel} to 23.08% in four iterations. The resulting wavelets are shown in Figure 5.25. Then after three more impedance inversion runs and two wavelet parameters inversion runs E_{rel} levelled at 22.96%. The final impedance profiles are shown in Figure 5.26 where it could be noticed that a reasonable estimate of the observed impedance profile is obtained. The final wavelet parameters inversion

results are shown in Figure 5.27 where the final solution is obtained in three iterations. The final wavelet solution parameters are $\mathbf{w}_{sol}=$ [4.2 44.3 54.8 79.1 115000. 115000. 0.3826 0.113 0]^T.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.23. The impedance profiles for the first impedance run (run number 1 in Table 5.25) when inverting for impedance, boundaries and wavelet parameters with a signal-to-noise ratio of 4 in the observed trace.

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.24. The impedance profiles for the first, and only, boundaries inversion run (run number 2 in Table 5.25) when inverting for impedance, boundaries and wavelet parameters with a signal-to-noise ratio of 4 in the observed trace. This is the only boundaries inversion needed because all the boundaries converged to their correct locations in this run.

Figure 5.25. Wavelet number 1 is the initial guess wavelet, and wavelets numbered 2-5 are the solution wavelets for the four iterations of inversion run number 4 in Table 5.25. Note that the vertical axis represents the two-way travel-time in ms.

128

ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.26. The final impedance profiles corresponding to the solutions given in Table 5.24. A reasonable estimate of the observed impedance profile is obtained.
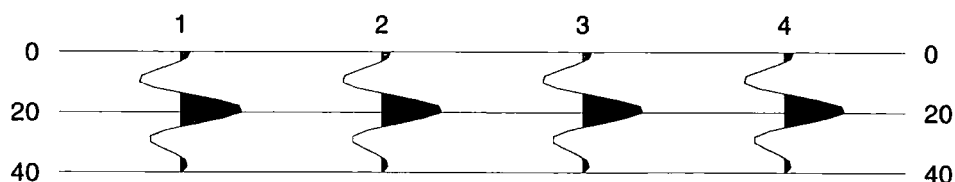
Figure 5.27. The final wavelet parameters inversion results in the three iterations. The final solution wavelet is $w_{sol} = [4.2 \quad 44.3 \quad 54.8 \quad 79.1 \quad 115000. \quad 115000. \quad 0.3826 \quad 0.113 \quad 0]^T$ which is wavelet number 4.

This example was repeated for a signal-to-noise ratio of 2 in the observed trace. The input and output data for this test are listed in Table 5.26, and the progression toward the solutions is given in Table 5.27.

layer number n	observed boundary locations (ms) t_{obs}	solution boundary locations (ms) t_{sol}	initial boundary locations (ms) t_{ini}	observed starting impedances $(gcm^{-3}ms^{-1})$ x_{obs}	solution starting impedances $(gcm^{-3}ms^{-1})$ x_{sol}	initial starting impedances $(gcm^{-3}ms^{-1})$ x_{ini}	observed impedance gradients $(gcm^{-3}ms^{-1}/sample)$ y_{obs}
1	60	60	56	11000	11000	11000	0
2	74	72	72	6000	70438	7000	0
3	82	82	82	8000	8962	9500	0
4	112	114	108	5000	4120	4000	0
5	126	122	128	7000	6282	6000	0
6	256	256	256	5000	5241	4500	0

Table 5.26. The input and output data for the example of inversion for impedance, boundaries and wavelet parameters when the signal-to-noise ratio in the observed trace is 2. A summary of the progression toward the solutions is given in Table 5.27.

Run number	0	1	2	3	4	5	6	7	8	9
Inversion type	-	Imp	Bnd	Imp	Wvl	Imp	Wvl	Imp	Wvl	Imp
Error energy (%)	99	80	40	26	23.08	23	22.98	22.975	22.967	22.691
Number of iterations	-	4	4	10	4	4	4	2	3	2

Table 5.27. A summary of the progression towards the solution when inverting for boundaries and impedance when the observed trace signal-to-noise ratio is 2. The solution results are given in Table 5.26. Imp means impedance inversion run, Bnd is boundary inversion run, and Wvl means a wavelet parameters inversion run.

The final impedance profiles in this test are shown in Figure 5.28. In the figure, boundaries 2, 4 and 5 did not converge to their correct locations, but the general impedance trend in the solution profile resembles that in the observed profile. The final solution wavelet parameters are $w_{sol}=$ [25.8 29.7 57 81 115000. 115000. 0.3981 0.113 0]^T, which closely resemble the observed wavelet parameters $w_{obs}=$ [24 28 55 84 115000. 115000. 0.4138 0.113 0]^T.
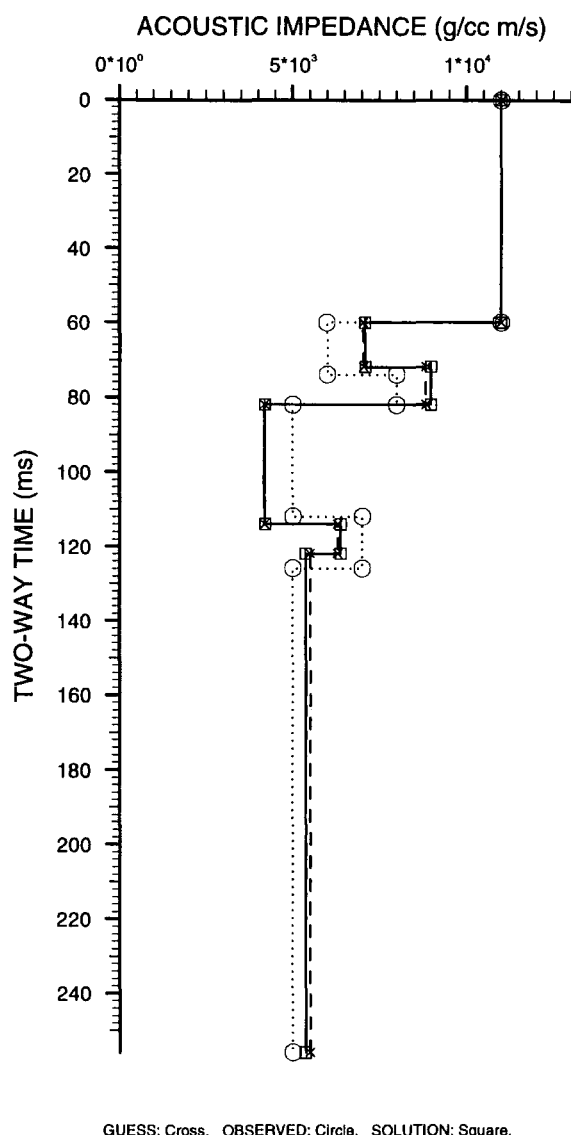
ACOUSTIC IMPEDANCE (g/cc m/s)

GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.28. The final impedance profiles corresponding to the solutions given in Table 5.26. The signal-to-noise ratio is 2. Boundaries 2, 4 and 5 did not converge to their correct locations, but the general impedance trend in the solution profile resembles that in the observed profile.

Finally, the same test was repeated for a signal-to-noise ratio of 1 in the observed trace. Only the final impedance profiles are shown in Figure 5.29, where we notice that the impedance values for layers 3 and 5 are much higher than their correct values, and for layer 6 it is much lower than its correct value. The general impedance trend in the shallow part of the profile is generally preserved, but in the lower part the impedance trend has largely changed. Notice that in this case all the boundaries converged to their correct locations. The final solution wavelet parameters converged to $\mathbf{w}_{sol}$= [25.7 29.7 58 78.6 115000. 115000. 0.307 0.113 0]^T, which is a good approximation to the observed wavelet parameters.

131

ACOUSTIC IMPEDANCE (g/cc m/s)
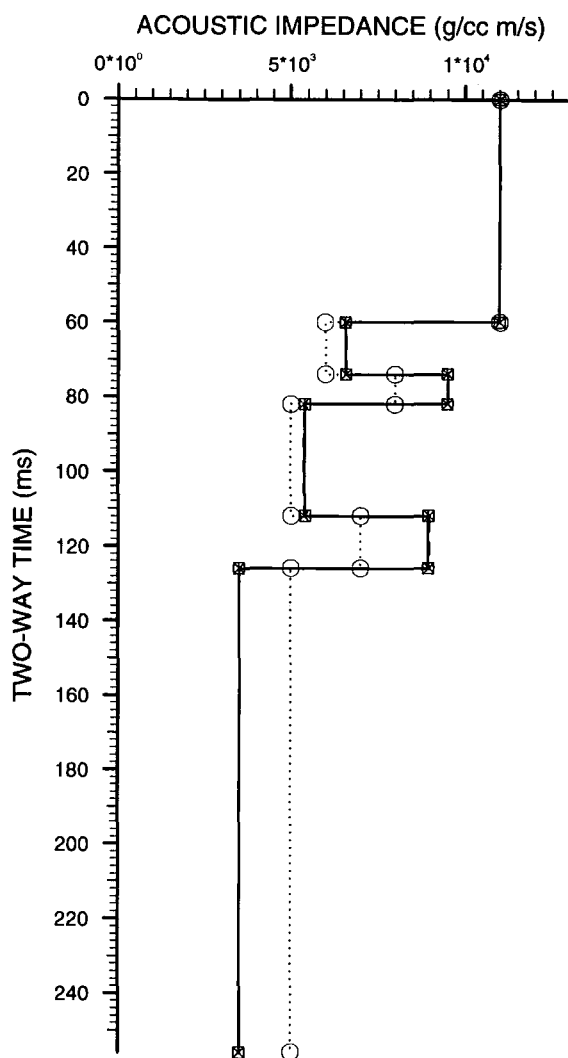
GUESS: Cross, OBSERVED: Circle, SOLUTION: Square.

Figure 5.29. The final impedance profiles for the case when the signal-to-noise ratio in the observed trace is 1, and when inverting for impedance, boundaries and wavelet parameters. Notice that even for this low signal-to-noise ratio, all the boundaries converged to their correct locations.

5.6. Conclusions

The synthetic examples of the previous sections illustrate the following:

1. For the impedance inversion to converge to the correct solution, it is extremely important to have the boundaries as close as possible to their correct locations.

2. The boundaries that are located incorrectly will lead to polarity reversals when the error in their locations approaches the radius of convergence.

3. When the wavelet is also incorrect, we should expect the tolerance in the boundary locations error to be less than the radius of convergence.

4. To avoid the impedance solution from catapulting away from the correct values, it is necessary to use inequality constraints.
5. We start the inversion by inverting for impedances.

CHAPTER

6

REAL DATA INVERSION

6.1. Introduction

The different synthetic inversion examples of chapter 5 suggest the following procedure for the inversion of field seismic data. First, we start at a well location where a Wiener estimate of the seismic wavelet was made (see section 2.5). We parameterise the amplitude and the phase spectra of the wavelet and obtain the nine parameters describing it. We also parameterise the acoustic impedance log of the well to obtain the earth model describing the subsurface geology there. When parameterising the acoustic impedance log we always keep the number of the earth model parameters to a minimum, and only use the layers that contribute significantly to the synthetic seismogram energy, or use only earth model parameters to which we can associate the reflection energy on the observed seismic trace. We then invert for the wavelet parameters to obtain an optimum wavelet using the earth model generated by parameterising the acoustic impedance log at the well. Finally, we make the assumption that the seismic wavelet does not change throughout the seismic line. This is because the specific shape of the wavelet tends to remain fairly consistent from one shot point to the next for the same seismic survey using the same source, geophones and recording instruments, and this wavelet propagates in the earth under similar circumstances thus remaining largely unchanged in character. The recorded seismic data are then processed using the same processing sequence; thus the effective seismic wavelet will also remain fairly consistent from one seismic trace to the next in the final seismic data. This would imply that we can invert for the boundary locations

134

and impedance parameters away from the well using the same wavelet optimised at the well. Furthermore, we try as far as possible to make a good initial guess for the boundary locations by making a structural interpretation of the seismic data, so that only small adjustments to the boundary locations will be necessary; thus the inversion problem becomes mostly that of inverting for the acoustic impedance. Also, using the proper constraints on the impedance in each layer we can prevent the final solution from converging to an incorrect one (non-unique problem), and prevent polarity reversals across boundaries which might occur due to the presence of noise, thus minimising the effect of the noise on the inversion results.

In section 6.2 we discuss the parameterisation of the wavelet estimated at well YY31, and section 6.3 discusses the parameterisation of the wavelet estimated at well YY04. Section 6.4 discusses the parameterisation of the impedance log at YY31 to obtain an earth model which will be used in optimising the wavelet estimated there, and section 6.5 discusses the optimisation of the wavelet at YY31. Section 6.6 discusses the parameterisation of the acoustic impedance of well YY04, which is used to optimise the wavelet estimated at that well in section 6.7. Having obtained the optimum wavelet at YY31 and YY04, section 6.8 discusses the acoustic impedance inversion of part of Line 1973 and of two parts of Line 1977. In section 6.9 we consider a different method of optimising the wavelet at well YY31 and YY04. In this method we only invert for the wavelet parameters and boundary location, and assume that the impedance of the layers is the same as that in the well impedance profile. Section 6.10 discusses inversion for impedance around well YY31 using the wavelet optimised at the well in section 6.9, and section 6.10 discusses inversion for impedance around well YY04 also using the wavelet optimised at the well in section 6.9. In section 6.12 we discuss the inversion at the intersection of Lines 1973 and 1977. In section 6.13 we draw some conclusions on the inversion of the real seismic data.

6.2. Parameterising the wavelet estimated at well YY31

The method of estimating the effective seismic wavelet as a Wiener shaping filter was discussed in section 2.5. The time domain wavelet estimate in YY31 is shown in Figure 6.1 To parameterise the wavelet we use the Fourier transform to obtain its amplitude and phase spectra which we parameterise by nine parameters that describe the wavelet in the frequency domain. It is these nine parameters that we optimise in order to obtain an optimum wavelet at a well location.

Figures 6.2 and 6.3 show the amplitude and the phase spectra of the wavelet of Figure 6.1. In these two figures, and for the other amplitude and phase spectra figures, only the amplitude and the phase values for frequencies up to 100 Hz are

shown since the seismic signal in the observed data is below 90 Hz due to the band pass filter of 6-10-80-90 Hz applied during processing the data.
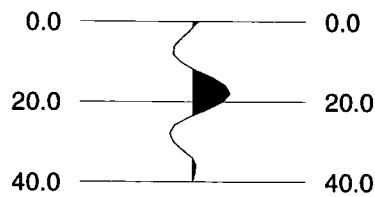
Figure 6.1. The wavelet estimated as a Wiener shaping filter at well YY31. Note that the vertical axis represents the two-way travel-time in ms.
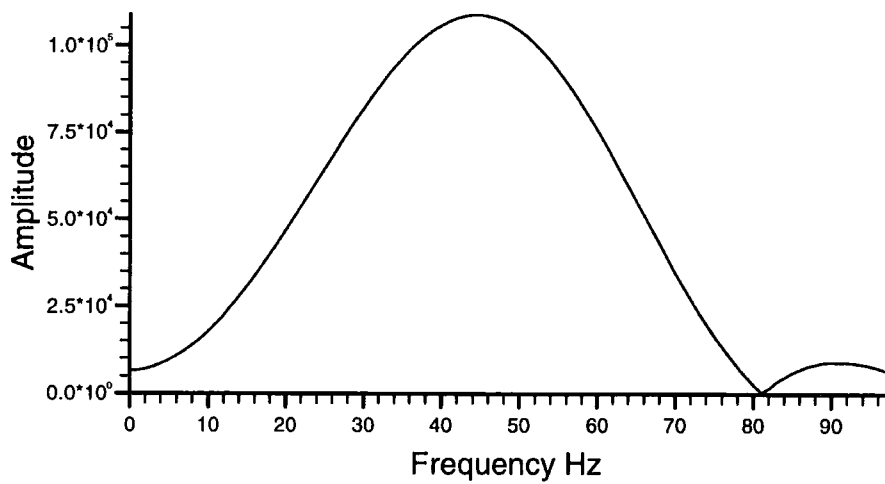
Figure 6.2. The amplitude spectrum of the Wiener wavelet estimated at well YY31 and shown in Figure 6.1.
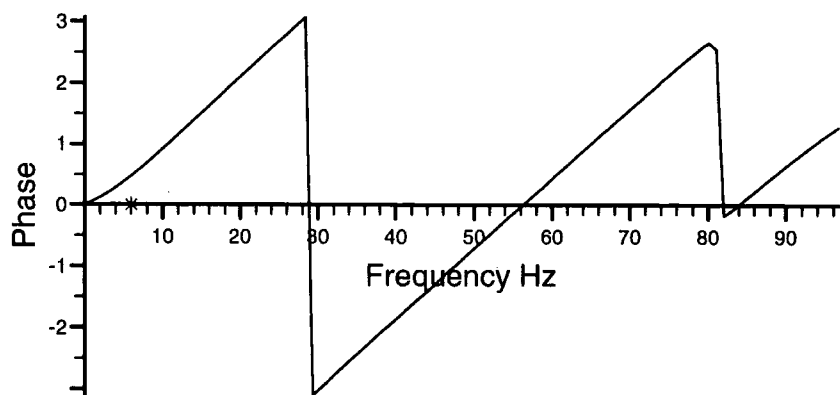
Figure 6.3. The phase spectrum of the Wiener wavelet estimated at well YY31 and is shown in Figure 6.1.

To parameterise the amplitude spectrum, we use four bounding frequency parameters, f_1, f_2, f_3 and f_4, and two amplitude parameters a_1 and a_2. The amplitude value of f_1 and f_4 are always kept at zero, while the amplitude parameters a_1 and a_2 describe the amplitude values for f_2 and f_3. To obtain the amplitudes for those frequencies between the four bounding frequencies we use linear interpolation. Figure 6.4 shows how the amplitude spectrum of the seismic wavelet estimated at YY31 was parameterised. The dotted line is the amplitude spectrum shown in Figure 6.2, and the solid line is the parameterised amplitude spectrum defined by the four bounding frequencies which are marked by circles on the frequency axis. Also, the four corresponding amplitude values are marked by asterisks. In Figure 6.4 the four bounding frequencies, f_1, f_2, f_3 and f_4 are 6, 38, 50 and 80, respectively. The amplitude values for both f_2, and f_3 are 108821.
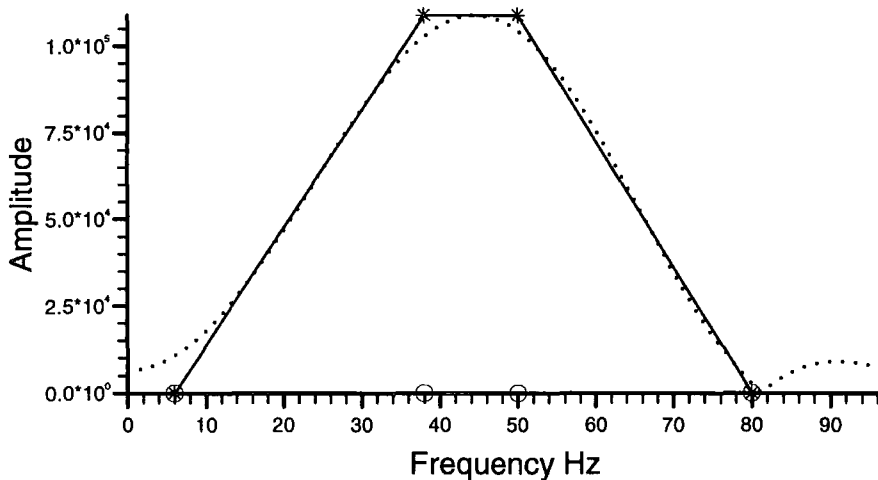
Figure 6.4. Parameterising the amplitude spectrum of the Wiener wavelet estimated at well YY31.

Figure 6.5 shows the parameterisation of the phase spectrum of the wavelet estimated at YY31, which was shown in Figure 6.3. The relevant frequency range is determined by f_1, f_2, f_3 and f_4 which are obtained by parameterising the amplitude spectrum. Figure 6.5 shows that the phase spectrum can be reasonably approximated by the line $\phi(f) = \phi_0 + \phi_1 f$, that is the quadratic phase term ϕ_2 of equation (1.4) is set to zero. The dotted curve in Figure 6.5 is the phase spectrum of Figure 6.3, and the solid line is the linear approximation to the phase spectrum in the interval $[f_1, f_4]$, and is drawn after wrapping it around $-\pi$ and π. The lower frequency limit $f_1 = 6$ Hz is marked on the frequency axis by an asterisk. The solid line is extrapolated to $f = 0$ Hz to obtain the value of ϕ_0. The solid line has a slope of ϕ_1. From Figure 6.5 $\phi_0 = -0.2$ radians, and $\phi_1 = 0.115$ radian/Hz.
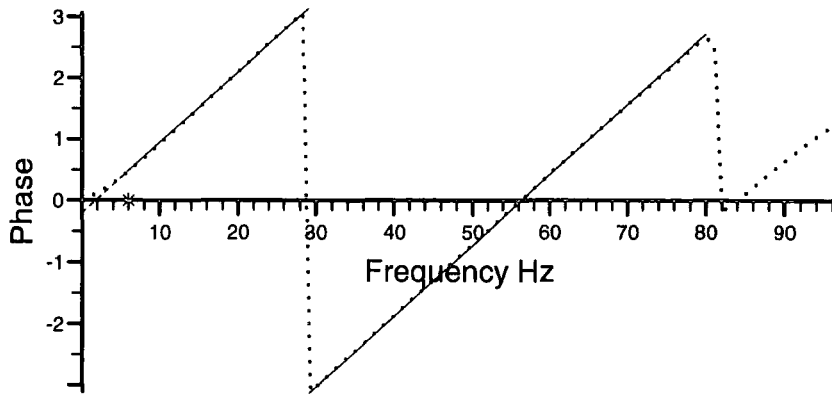
Figure 6.5. Parameterising the phase spectrum of the Wiener wavelet estimated at well YY31.

The parameterisation results of Figures 6.4 and 6.5 define the wavelet of well YY31 by the 9-vector w=[6 38 50 80 108821 108821 -0.2 0.115 0]^T. The time domain representation of the of the nine parameter wavelet, obtained by doing the inverse Fourier transformation, is shown in Figure 6.6, along with the Wiener wavelet so that the two wavelets can be compared to assess the degree of similarity. Clearly the parameterised wavelet is a good approximation to the Wiener wavelet.
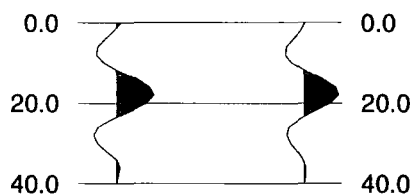
Figure 6.6. The estimated wavelet at YY31 (left) as compared to its parameterised equivalent. Note that the vertical axis represents the two-way travel-time in ms.

6.3. Parameterising the wavelet estimated at YY04

Following the same procedure as in section 6.2, we determine the nine frequency domain parameters that define the wavelet at the well YY04. The time-domain wavelet estimated as a Wiener shaping wavelet is shown in Figure 6.7.

The amplitude spectrum of the Wiener wavelet at YY04 is shown in Figure 6.8. The amplitude notch present at about 17 Hz can be explained by the presence of a prominent d.c. component in the wavelet (Bath, 1974). The parameterisation of the amplitude spectrum is shown in Figure 6.9, where it can be observed that the four

bounding frequencies are 17, 32, 42 and 67 Hz, and the amplitude for each of the two middle frequencies is 113330.
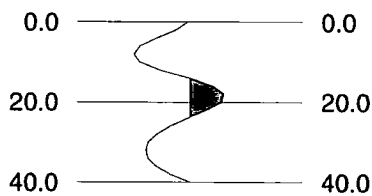
Figure 6.7. The Wiener wavelet estimated at well YY04. Note that the vertical axis represents the two-way travel-time in ms.
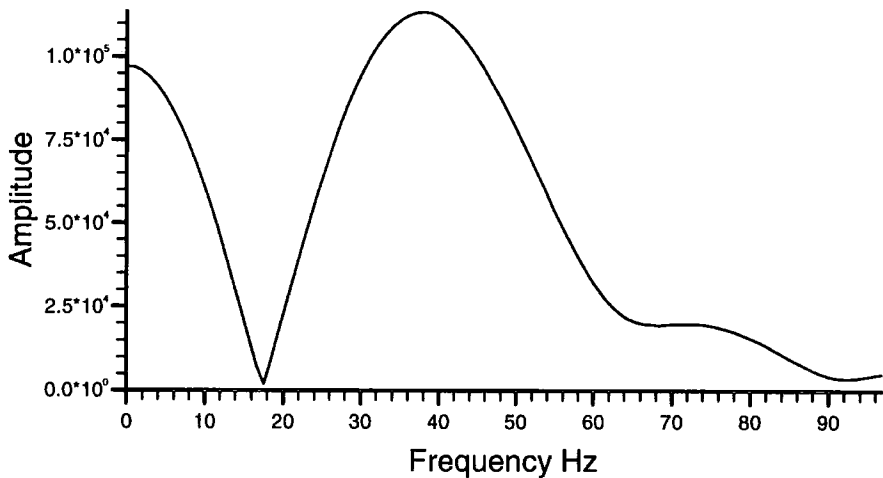
Figure 6.8. The amplitude spectrum of the Wiener wavelet estimated at well YY04 and is shown in Figure 6.7.
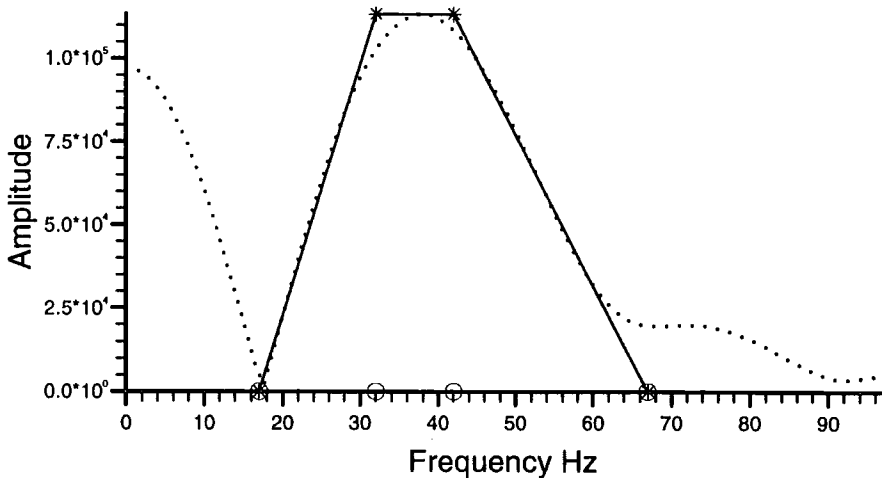
Figure 6.9. The parameterisation of the amplitude spectrum of the Wiener wavelet estimated at well YY04.

The phase spectrum for the wavelet estimated at YY04 is shown in Figure 6.10 and its parameterisation is shown in Figure 6.11, where the linear approximation gives $\phi_0 = 0.1$ radians, and $\phi_1 = 0.12$ radian/Hz. The resulting parameterised wavelet is w=[17 32 42 67 113330 113330 0.1 0.12 0] which is plotted along with the Wiener wavelet in Figure 6.12.
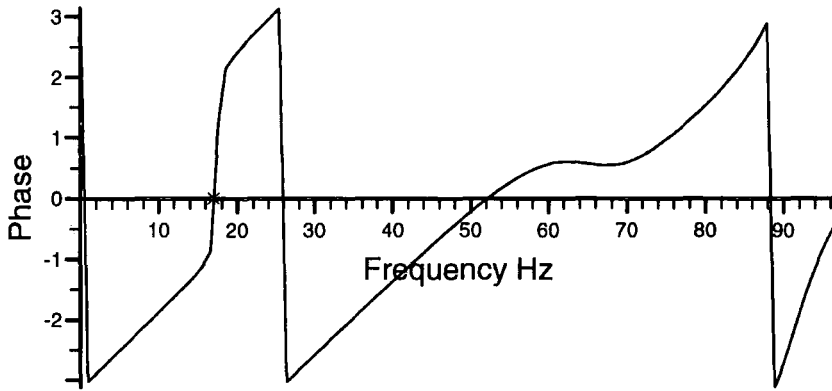
Figure 6.10. The phase spectrum of the Wiener wavelet estimated at well YY04 and is shown in Figure 6.7.
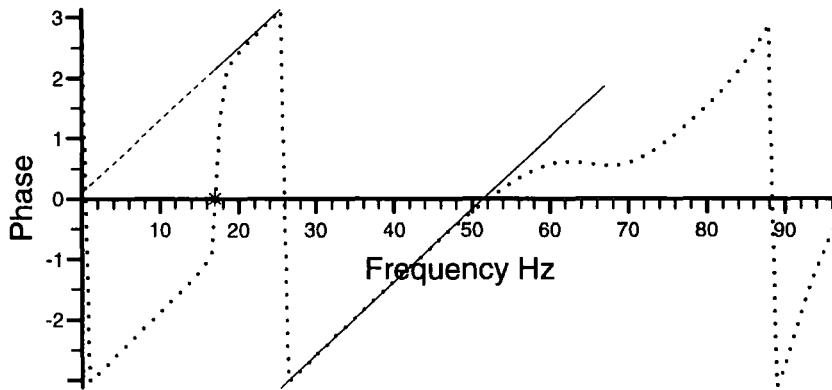
Figure 6.11. Parameterisation of the phase spectrum of the Wiener wavelet estimated at well YY04.
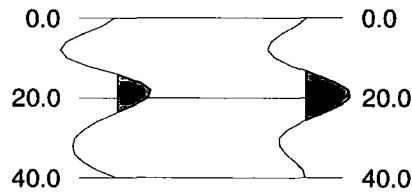
Figure 6.12. The estimated wavelet at YY04 (left) as compared to its parameterised equivalent. Note that the vertical axis represents the TWO-WAY TRAVEL-TIME in ms.

6.4. Parameterising the acoustic impedance log of YY31

The acoustic impedance log of YY31 is shown in Figure 6.13. The high impedance layer at the bottom of the log is the thin part of the Augila limestone that was measured by the log. The Augila limestone is overlain by the Chadra sands which is present from the top of Augila to about 720 ms of two-way travel-time. Then the Arida shale overlies the Chadra sands and extends to about 690 ms two-way travel-time. The more recent material overlying the Arida shale is made up of interbedded sandstones and shales with some limestone as well, which are not of interest in this work but which will also be included in the impedance log parameterisation.

Figure 6.14 shows the parameterised impedance log of YY31. The dotted line is the impedance log of Figure 6.13 and the solid line is its parameterisation into 12 layers of constant acoustic impedance, that is $y_i = 0$ for $i = 1, 2, \ldots, 12$. This parameterised acoustic impedance will be used to optimise the wavelet parameters at YY31 obtained in section 6.2.

6.5. Optimising the wavelet estimate at YY31

In order to optimise the estimated wavelet parameters we use a similar approach to that in section 5.5.5. We first invert for impedance, then for boundary locations, then we repeat the impedance and boundary locations inversion sequence as many times as needed until convergence is achieved for both problems. Then we invert for the wavelet parameters, and then again repeat inverting for impedance and boundaries in succession until convergence. The whole sequence is then repeated until convergence is achieved in the three problems, or until the improvement in error energy reduction is very small. The resulting wavelet parameters obtained are then the optimum parameters we are seeking, which are used to invert for the impedance and boundary locations across the seismic section.
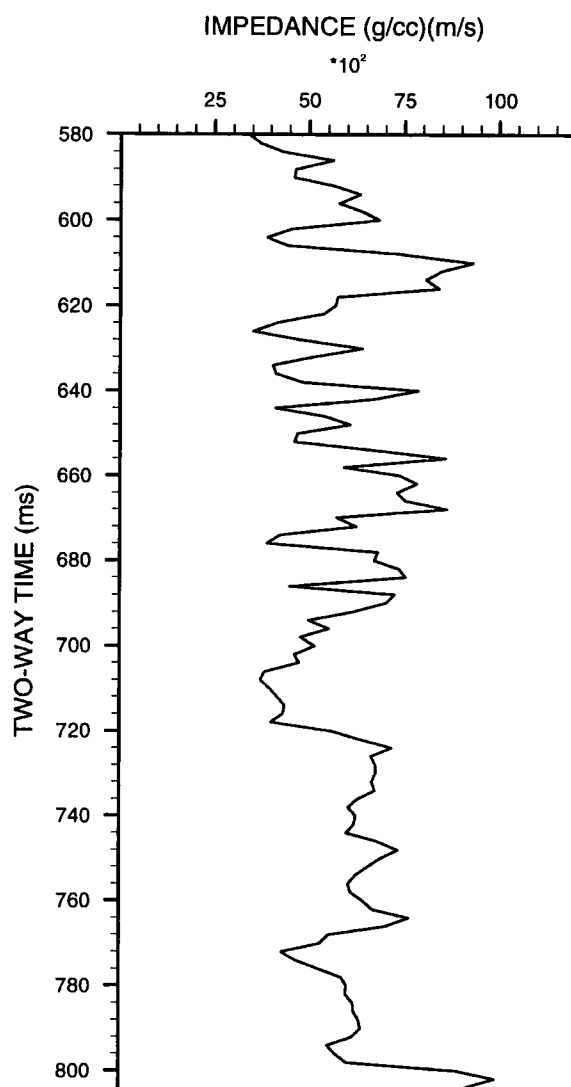
Figure 6.13. The acoustic impedance log of YY31.
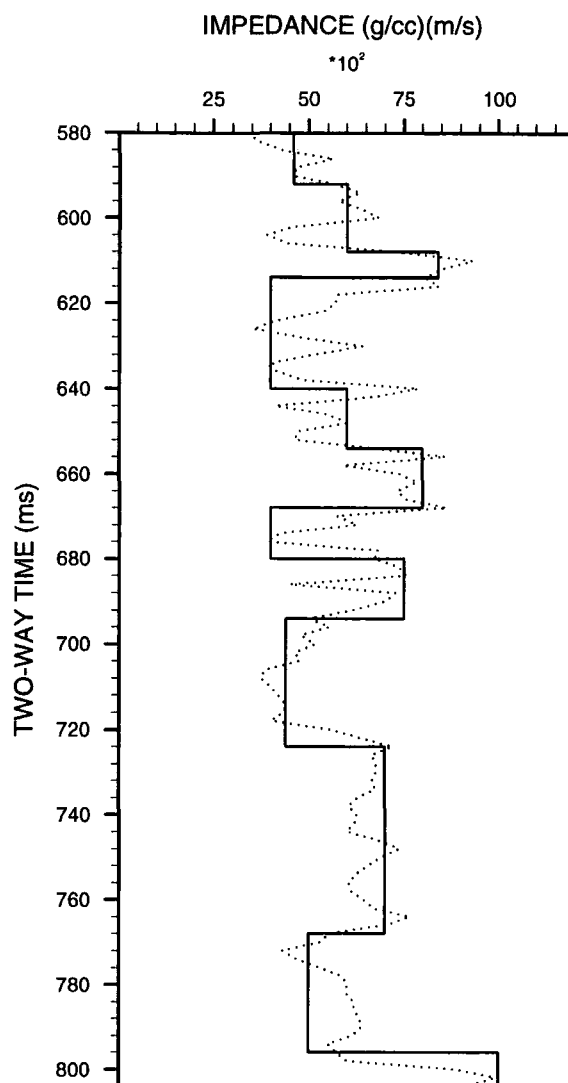
Figure 6.14. Parameterising the acoustic impedance log of YY31 into 12 layers.

Optimising the wavelet at YY31 gave the following results: the first impedance inversion started with the earth model with $E_{rel} = 47\%$, then after 18 iterations the relative error energy was reduced to $E_{rel} = 32\%$. The initial earth model is shown in Figure 6.15 as a dashed line with cross marks, and the final solution is shown in the same figure as a solid line. No constraints on the boundary locations were applied, but the impedance was constrained to vary between 3500 and 9000 gcm⁻³ms⁻¹ in all layers except the first and last, which were allowed to vary in broader the range of 1500 to 20000 gcm⁻³ms⁻¹. This is because the reflection energy at he first boundary could be contaminated by reflection energy from the layer(s) above it, and we should try to minimise its effects on optimisation results. Similarly, the last layer reflection energy could contain reflection energy from the layers below it.
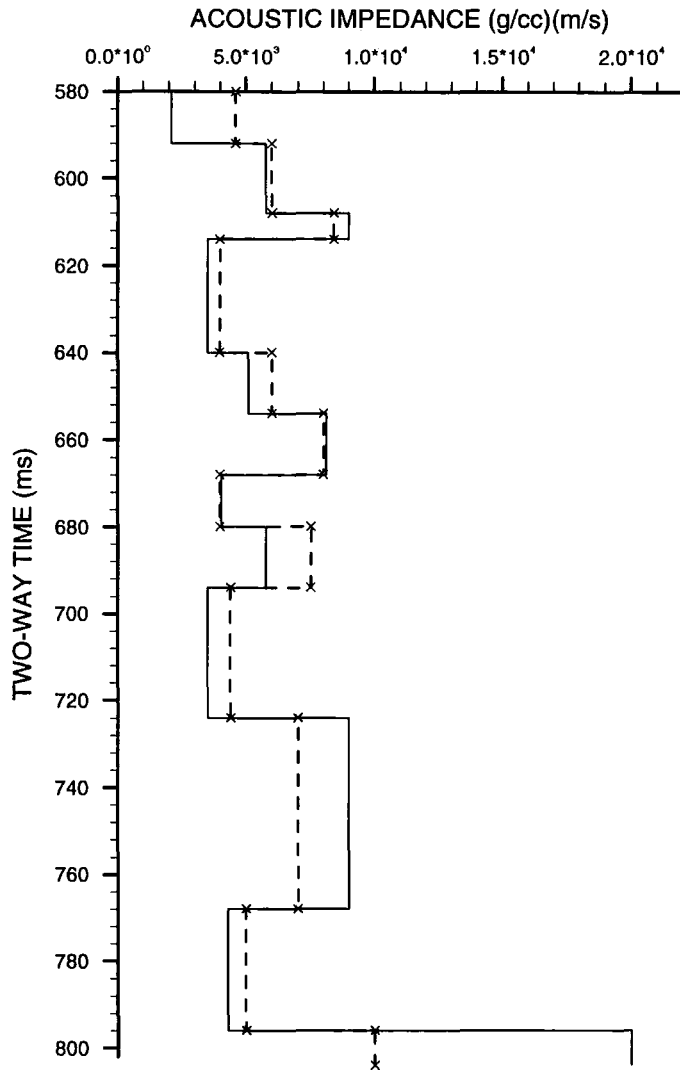
Figure 6.15. The first impedance inversion results when optimising the wavelet at YY31. The dashed line with crosses is the initial earth model, and the solid line is the inversion solution.

The progression of the impedance inversion solutions in the 18 iterations is shown in Figure 6.16, where the first and last traces are the observed seismic trace at the well and is displayed twice only for comparison purposes. The first trace after the observed trace, trace number 2, is the initial guess synthetic seismogram. Figure 6.17 shows the corresponding error traces. The first error trace in Figure 6.17 illustrates how the initial error energy is distributed in the initial guess synthetic seismogram, and the last trace should reveal where most of the reduction in error energy has taken place.
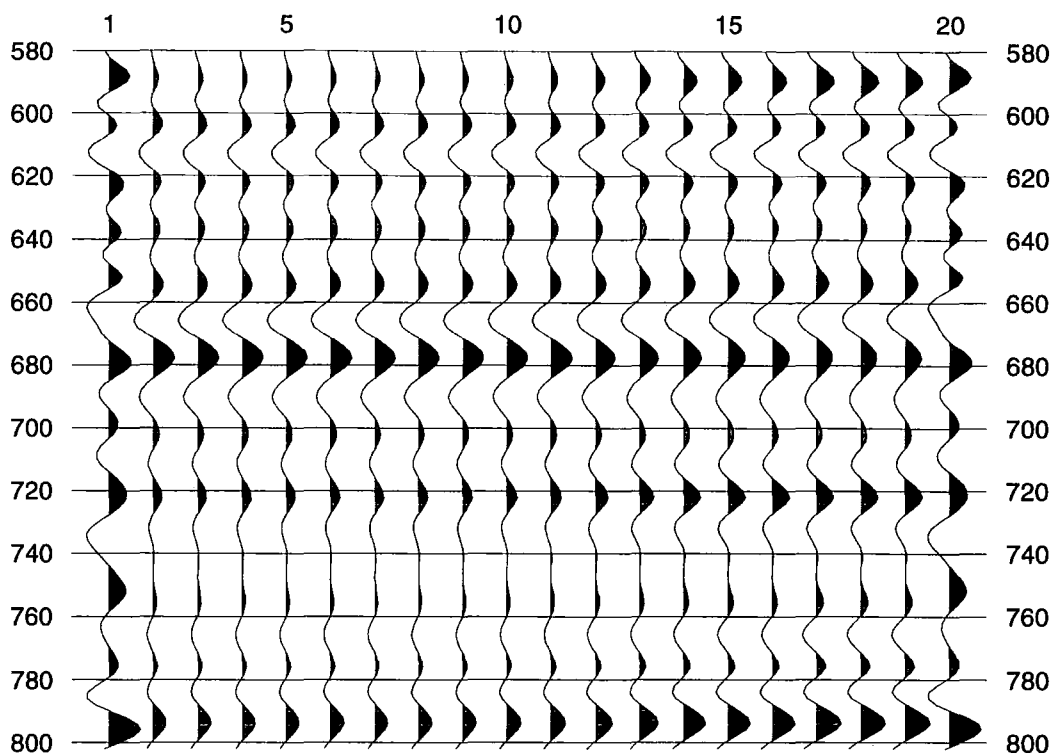
Figure 6.16. The first impedance inversion solutions when optimising the wavelet of well YY31. Traces number 1 and 20 are the observed traces, trace number 2 is the initial guess seismic response, and trace number 19 is the final solution seismic trace. Note that the vertical axis represents the two-way travel-time in ms.
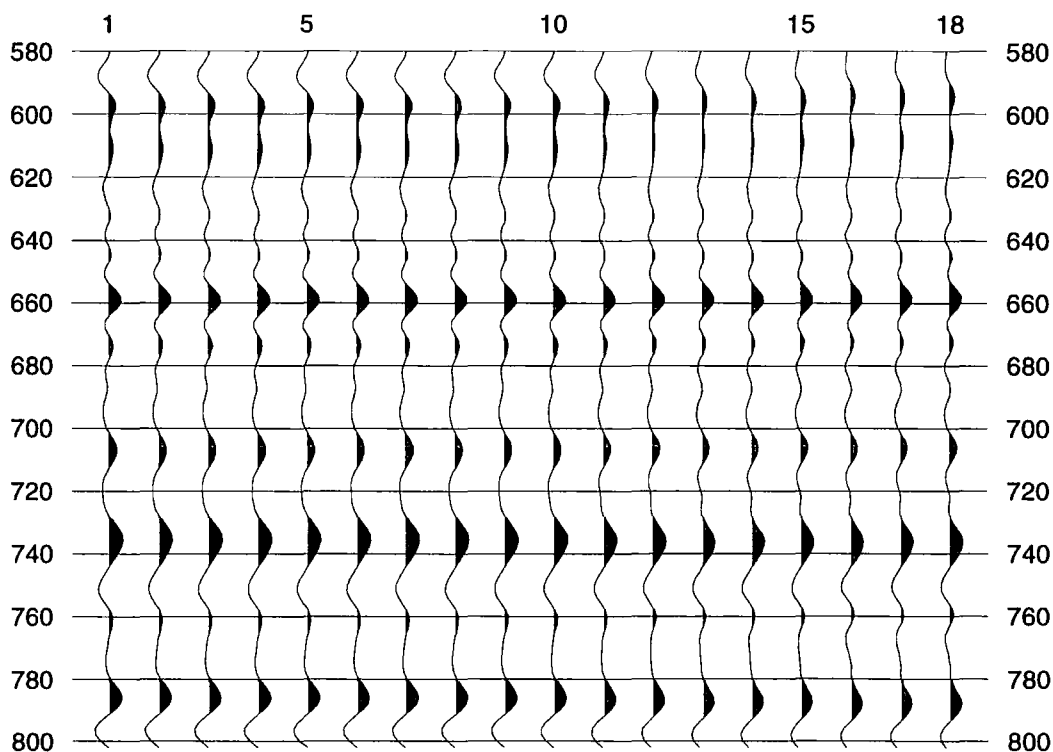
Figure 6.17. The error traces that correspond to the solutions shown in Figure 6.16. Trace number 1 is the initial guess error trace, and trace number 18 is the final solution error trace.

The first boundary locations inversion results are shown in Figure 6.18, where the dashed line represents the initial guess and the solid line is the solution. After one iteration the resulting $E_{rel} = 21\%$. The solution trace for the single iteration is shown in Figure 6.19 and its error trace is shown in Figure 6.20. This boundary location inversion was the only boundaries inversion required to converge to the boundaries solution. Thus, one impedance inversion was required before inverting for the wavelet parameters. This impedance inversion reduced the error energy to $E_{rel} = 17\%$.
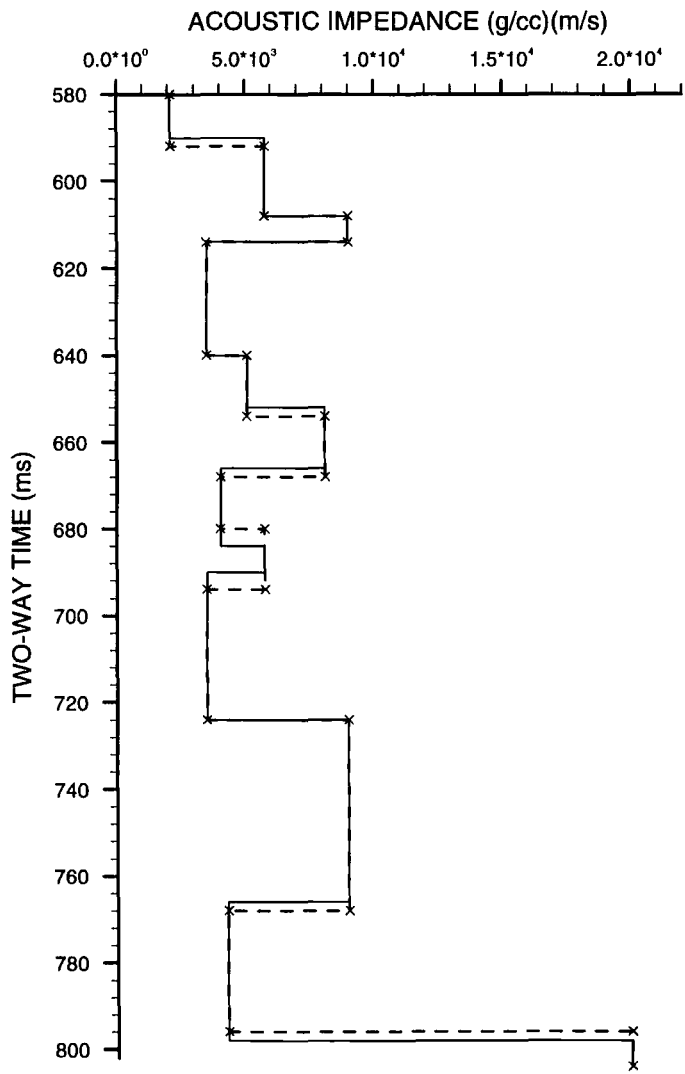
Figure 6.18. The impedance profiles for the first boundary locations inversion when optimising the wavelet at well YY31.
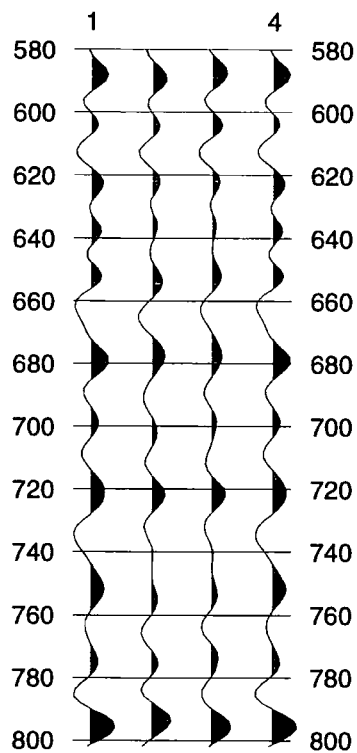
Figure 6.19. The first boundary locations inversion solutions when optimising the wavelet at well YY31. Traces number 1 and 4 are the observed seismic traces, trace number 2 is the initial guess seismic response, and trace number 3 is the seismic response solution of the single iteration performed. Note that the vertical axis represents the two-way travel-time in ms.
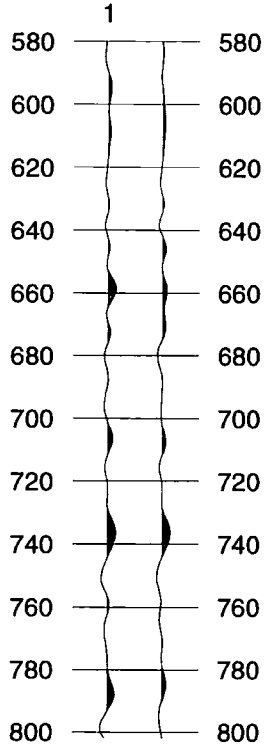
1

580		580
600		600
620		620
640		640
660		660
680		680
700		700
720		720
740		740
760		760
780		780
800		800

Figure 6.20. The error traces corresponding to the solutions of Figure 6.19. The first error trace is that for initial guess and the last error trace is that for the seismic solution. Note that the vertical axis represents the two-way travel-time in ms.

The results of the first wavelet parameter inversion are shown in Figure 6.21, where it can be observed that 49 iterations were performed by the program to reduce the error energy to $E_{rel} = 14.8\%$. Notice that in Figure 6.21, wavelet number 1 is the parameterised wavelet obtained in Section 6.2. The solution traces for the 49 iterations in this wavelet inversion are shown in Figure 6.22 and their corresponding error traces are shown in Figure 6.23.

Figure 6.21. The resulting wavelets in the first wavelet parameter inversion. The wavelet number 1 is the YY31 parameterised wavelet, and wavelet number 49 is the final wavelet. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.22. The solution traces that correspond to the wavelets of Figure 6.21. The first and last traces are the observed seismic trace, trace number 2 is the initial guess which corresponds to wavelet number 1 in Figure 6.21. Note that the vertical axis represents the two-way travel-time in ms.

149

Figure 6.23. The error trace that correspond to the solutions of Figure 6.22. Error traces number 1 corresponds to the initial guess wavelet of Figure 6.21, and error trace number 49 corresponds to the final wavelet. Note that the vertical axis represents the two-way travel-time in ms.

The optimum wavelet parameters were obtained after three more alternating inversions for impedance and wavelet parameters. The optimum wavelet is shown in Figure 6.24 as wavelet number 4 which was obtained with E_{rel} = 14.375%. The optimum wavelet parameters vector is w=[15.8 21 62.5 66.5 110000 110000 -.229 .116 0]^T
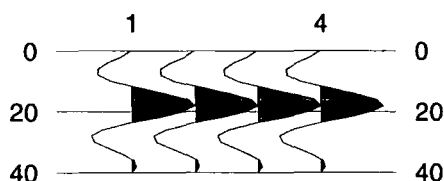
Figure 6.24. The optimum wavelet for well YY31 is wavelet number 4. Note that the vertical axis represents the two-way travel-time in ms.

A final inversion for impedance gave the impedance solution shown in Figure 6.25 with $E_{rel} = 14.3$ in four iterations. The final solution traces are shown in Figure 6.26 for this impedance inversion, and the corresponding error traces are shown in Figure 6.27.
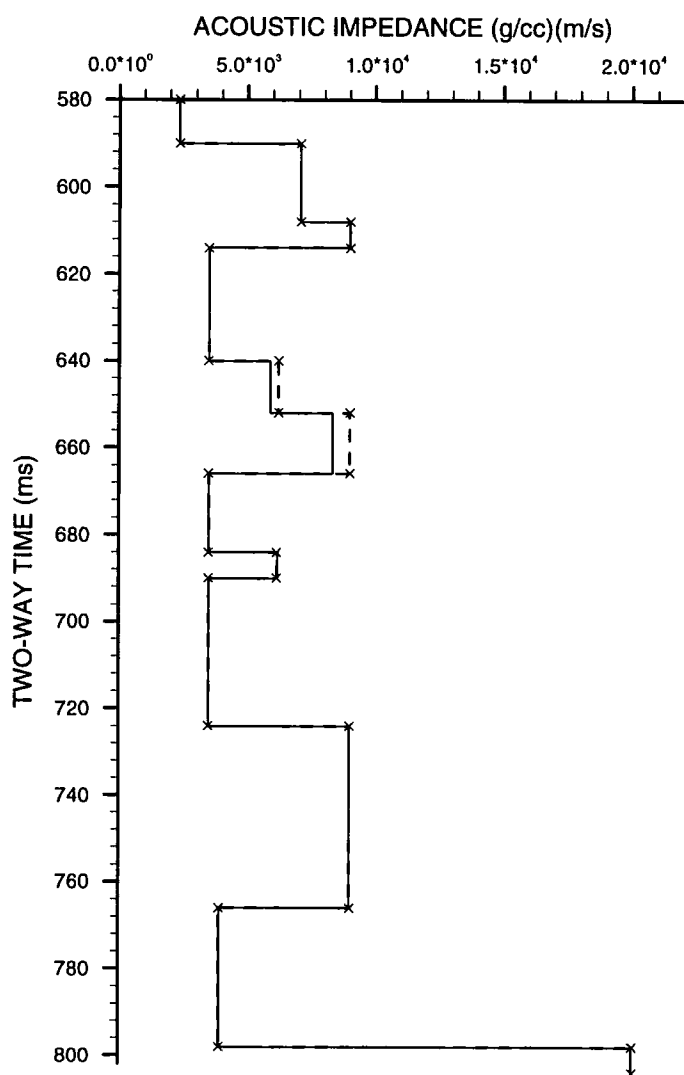
Figure 6.25. The optimum impedance profile (solid line) for well YY31.
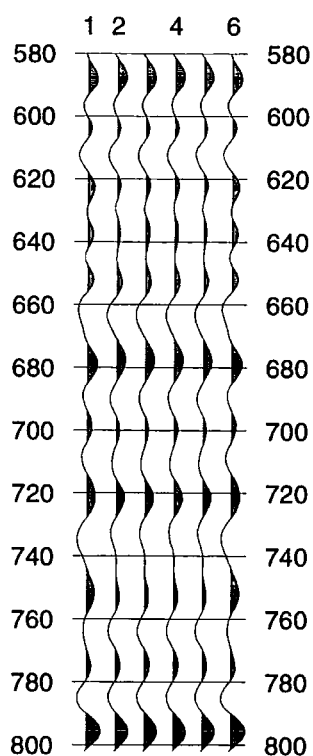
Figure 6.26. The optimum seismic solution for YY31 is trace number 5. Traces 1 and 6 are the observed seismic trace. Note that the vertical axis represents the two-way travel-time in ms.
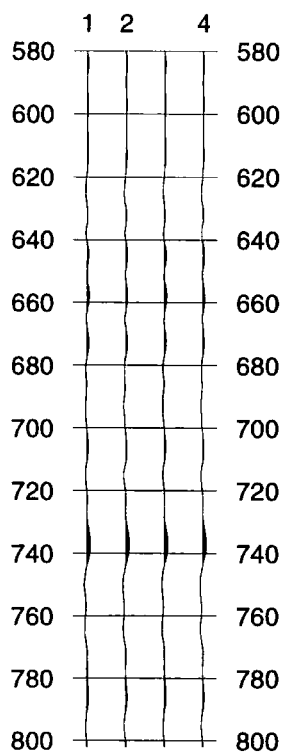
Figure 6.27. The error traces corresponding to the seismic solutions of Figure 6.26. Error trace number 4 corresponds to the optimum seismic solution trace, which is number 5 in Figure 6.26. Note that the vertical axis represents the two-way travel-time in ms.

6.6. Parameterising the acoustic impedance log of YY04

The acoustic impedance log of well YY04 is shown in Figure 6.28. The top of the Augila limestone is at 750 ms two-way travel-time and it extends to the bottom of the log, so that this well covers a thicker section of the limestone. The parameterisation of the impedance log is shown in Figure 6.29 as a solid line. The dotted line is the impedance log of Figure 6.28. In this parameterisation, more emphasis was given to the limestone section since the limestone beds produce most of the reflection energy on the seismic trace. The only other layer that was parameterised is the Chadra A sand. Thus, this well was parameterised using only 8 layers.
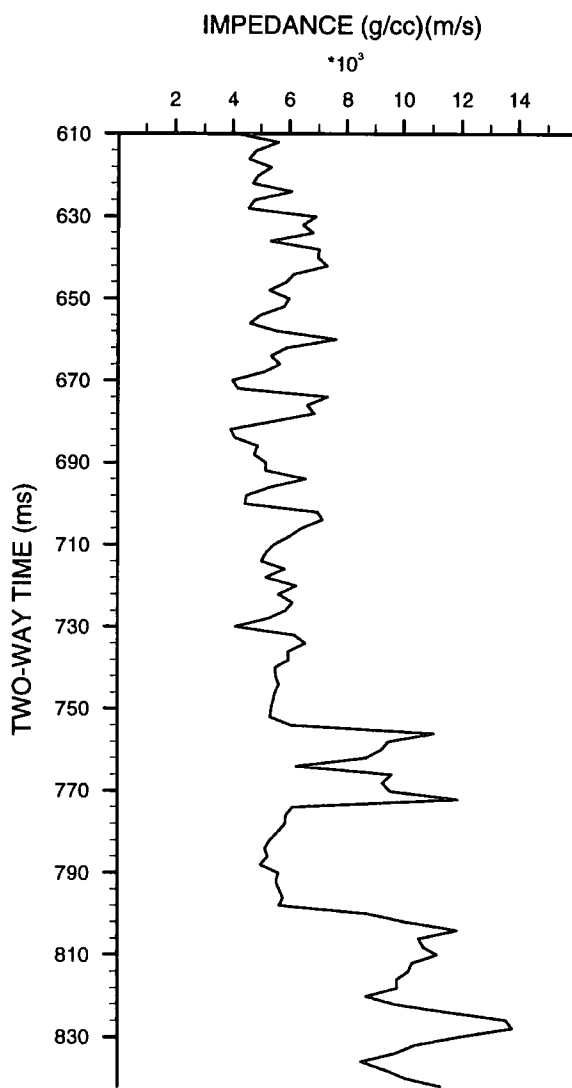
Figure 6.28. The acoustic impedance log of well YY04.
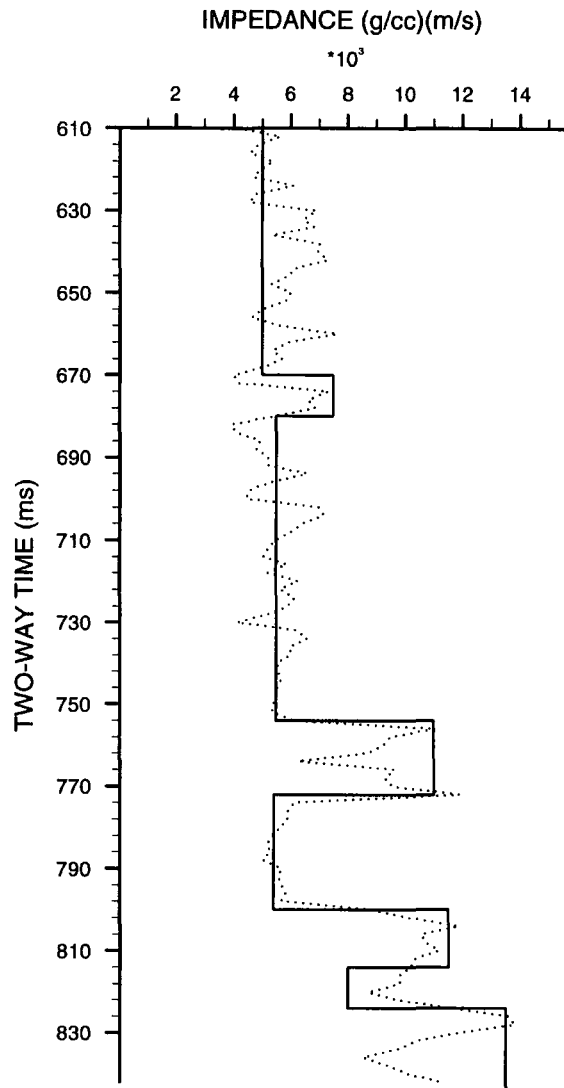
153

IMPEDANCE (g/cc)(m/s)

Figure 6.29. The parameterisation of the acoustic impedance log of YY04 into eight layers. Chadra A is the 10 ms thick sand layer between 670 ms and 680 ms two-way travel-time. The top of Augila Limestone is at 754 ms and extends to the bottom of the log.

6.7. Optimising the estimated wavelet at YY04

The wavelet parameters we want to optimise here are those obtained in section 6.3 which are $\mathbf{w} = [17 \quad 32 \quad 42 \quad 67 \quad 113330 \quad 113330 \quad 0.1 \quad 0.12 \quad 0]^T$. To optimise this wavelet we use the earth model obtained in section 6.6. The impedance of the first and last layers was allowed to vary in the range 500-95000 $\text{gcm}^{-3}\text{ms}^{-1}$, but the other layers had their impedance constrained within the range 450-15000 $\text{gcm}^{-3}\text{ms}^{-1}$.

The initial guess wavelet and earth model produced a relative error energy of $E_{rel} = 58\%$. The first impedance inversion reduced the error energy to $E_{rel} = 53\%$ after

154

nine iterations. The solutions for the nine iterations are shown in Figure 6.30 and the impedance solution for this impedance inversion is shown in Figure 6.31.
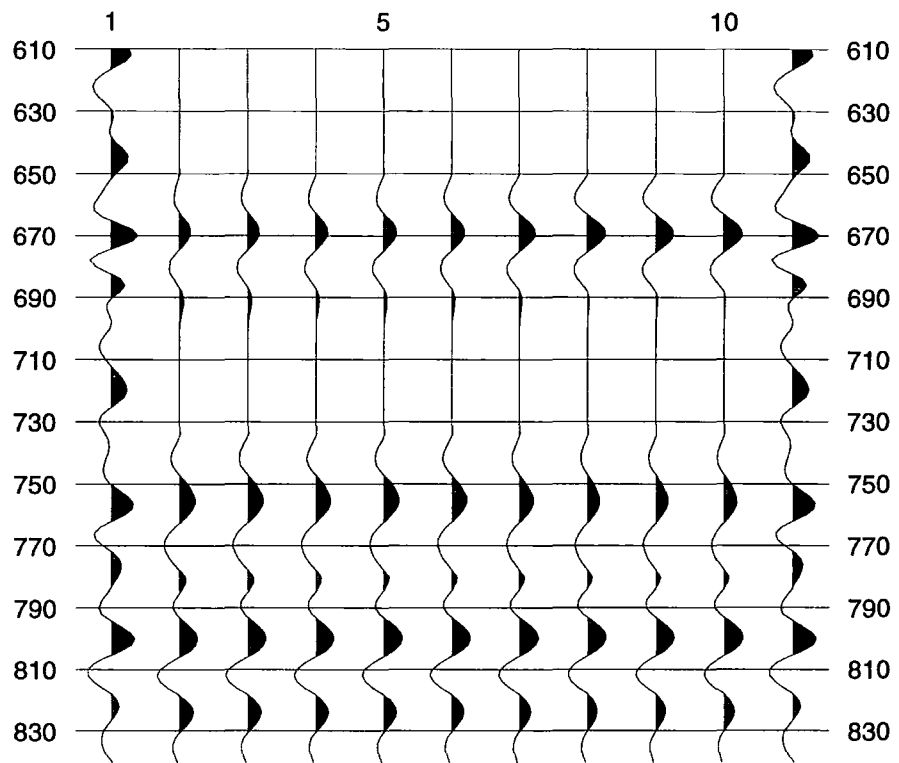
Figure 6.30. The solution traces for the first inversion for impedance when optimising the wavelet at YY04. The first and last traces are the observed seismic trace. Trace number 2 is the initial guess seismic response, and trace number 190 is the final seismic solution. Note that the vertical axis represents the two-way travel-time in ms.
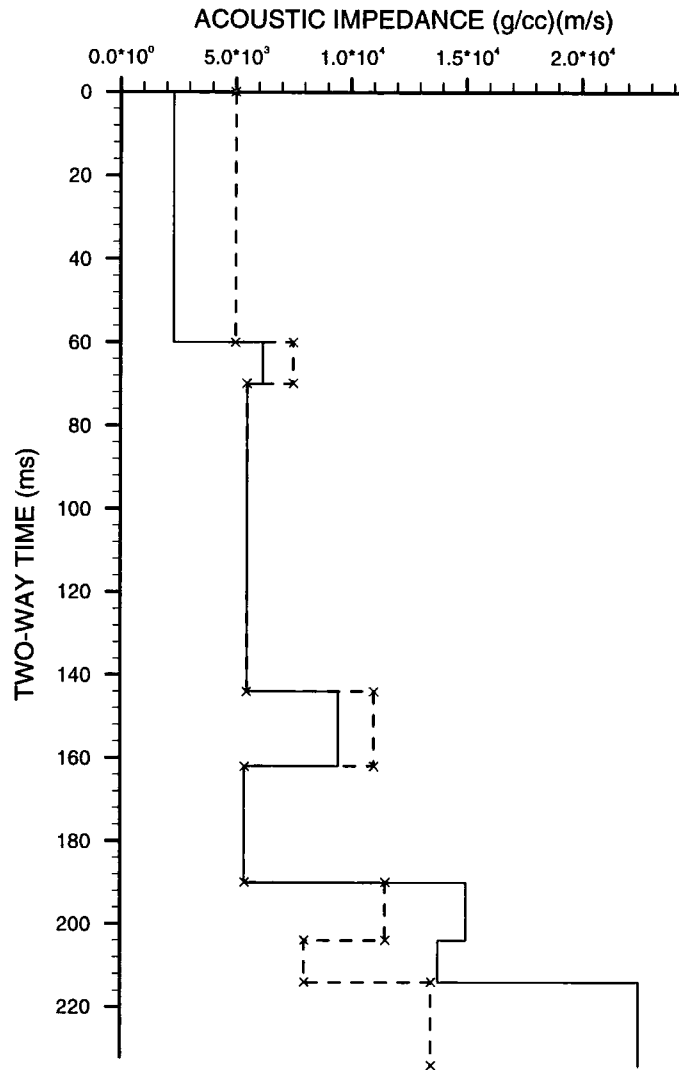
Figure 6.31. The impedance profiles for the inversion of Figure 6.30. Zero on the time axis corresponds to two-way travel-time of 610 ms.

The first boundary locations inversion reduced the error energy to $E_{rel} = 46\%$ in three iterations. The solutions for this boundaries inversion are shown in Figure 6.32. The impedance profile for these boundaries inversion is given in Figure 6.33.

The first wavelet parameters inversion was done after two more impedance inversions runs and one boundary locations inversion run. This has reduced the error energy so that $E_{rel} = 41\%$ after 43 iterations. The resulting wavelets for all the iterations are shown in Figure 6.34.

Figure 6.32. The seismic solutions of the first boundaries inversion when optimising YY04 wavelet. The first and last traces are the observed seismic trace, trace number 2 is the initial guess seismic response, and trace number 4 is the final seismic solution. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.33. The impedance profiles for the solution of Figure 6.32, i.e. for the first boundary location inversion when optimising the wavelet at well YY04. Zero on the time axis corresponds to 610 ms of two-way travel-time.

158

Figure 6.34. The resulting wavelets in the first inversion for the wavelet parameters in YY04. Wavelet number 1 is the initial parameterised wavelet. Note that the vertical axis represents the two-way travel-time in ms.

The final optimum wavelet obtained is shown in Figure 6.35, and the final impedance solution obtained after one more impedance inversion run is given in Figure 6.36. The optimum wavelet has parameters $\mathbf{w}$ = [23 30 34 83 113330 113330 0.018 0.128 0.]^T. This optimum wavelet will be used to invert for acoustic impedance in parts of Line 1977.
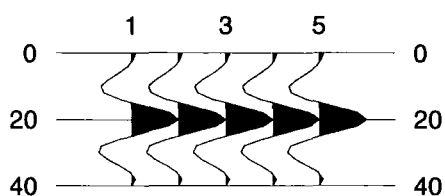
Figure 6.35. The optimum wavelet in YY04 is wavelet number 5 Note that the vertical axis represents the two-way travel-time in ms.

159

ACOUSTIC IMPEDANCE (g/cc)(m/s)
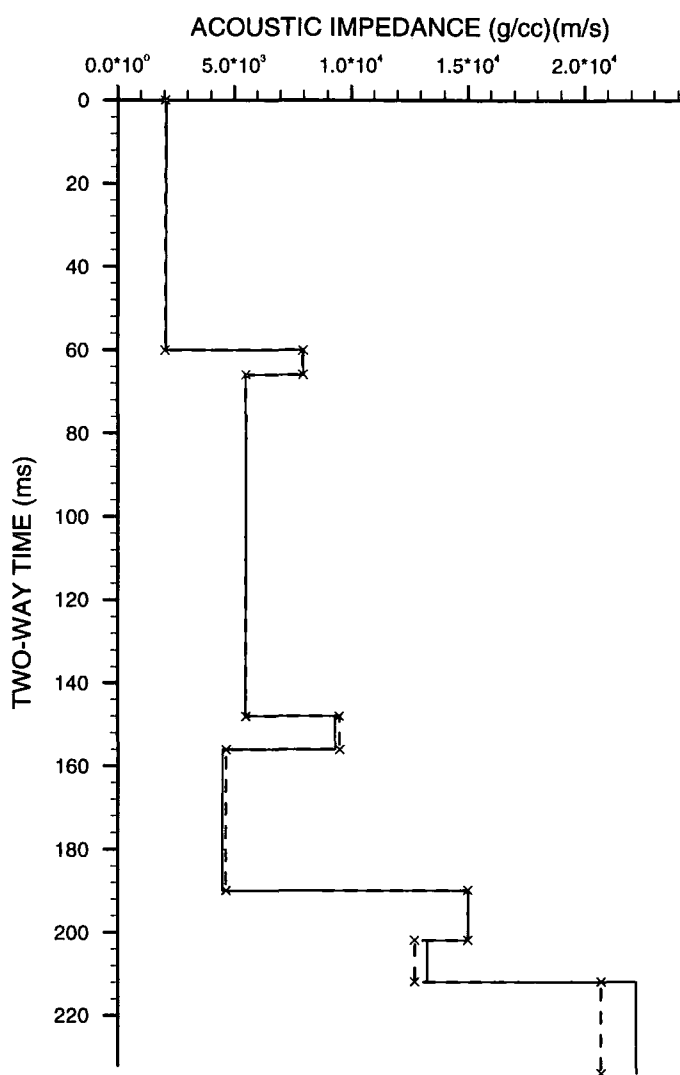
Figure 6.36. The optimum impedance profile in well YY04. Zero on the time axis corresponds to 610 ms of two-way travel-time.

6.8. Impedance inversion examples from Lines 1973 and 1977

In the next examples of inversion the objective is to obtain an impedance section of a number of traces from Lines 1973 and 1977 in a limited time window. In each example, the resulting impedance section represents an interpretation of the lithology within this limited number of traces and time window as determined by the inversion program. In all the examples, each observed seismic trace has its own initial guess, and we use the seismic wavelet optimised for each seismic line as described in sections 6.6 and 6.7. The process is to first solve for impedance then boundary locations then repeat this sequence as many times as is needed to obtain convergence for both impedance and boundary location inversion. The result will represent the final solution required. This process is repeated for every trace selected

160

for inversion. Then all the impedance solutions obtained are displayed together to represent the lithologic interpretation of the geology.

Before the data are inverted, we first structurally interpret the seismic data so that the number of layers is defined for every seismic trace, which need not necessarily be the same for every trace. Also, by picking the boundary locations as accurately as possible, we make certain that we are within the convergence zone defined by one-half of the central lobe width of the optimised seismic wavelet. Thus we satisfy most of the requirements reached in section 5.6 of chapter 5. In this way the inversion problem becomes mostly that of impedance inversion with small boundary locations adjustments.

6.8.1. Inversion for impedance example from Line 1973

This example represents the inversion of two seismic events originally interpreted as two Chadra sand bodies, and which extend for 40 seismic traces. The observed seismic data for this example are shown in Figure 6.37, and they are CDP numbers 748-787 in line 1973.
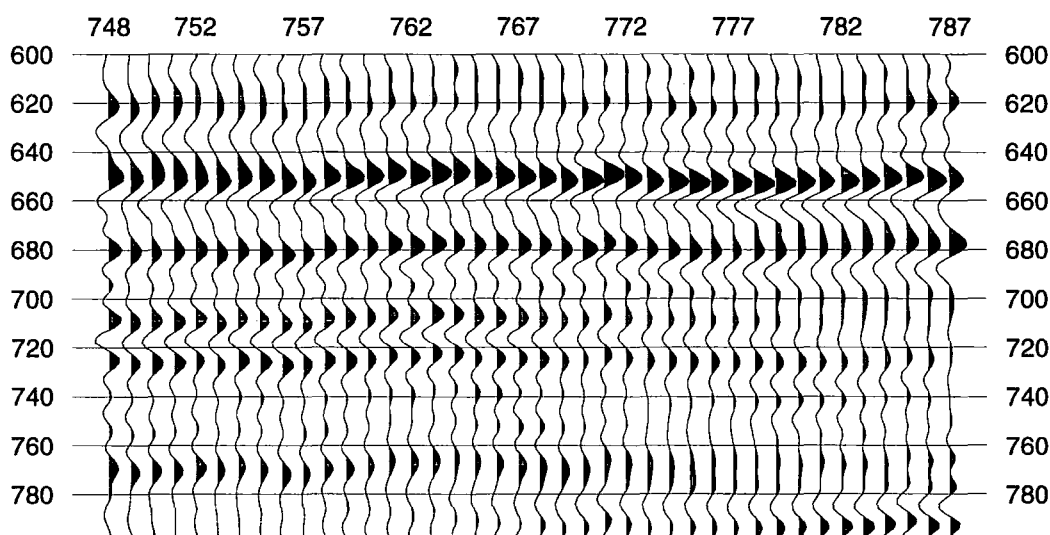
Figure 6.37. The observed seismic traces (CDP's 748-787) from Line 1973. The two sand bodies concerned are the two positive reflections (peaks) at about 710 ms and 730 ms on CDP 748. Note that the vertical axis represents the two-way travel-time in ms.

The two sand bodies concerned show up as two positive reflections (peaks) at about 710 ms and 730 ms two-way travel-time. The lower event has better continuity and seems to extend across the whole section in Figure 6.37, while the shallower event seems to terminate, possibly due to the sand body pinching out at about CDP number 779. The trough above the shallower event is interpreted as a negative reflection from the top of a shale layer that overlies the sand body and extends across

161

the section. The weak trough below the deeper event is interpreted as a possible low velocity sand layer.

To help solve the seismic boundaries (structural) problem, we display in Figure 6.38 the initial guess for each seismic trace on to the seismic trace plotted only in variable area display, i.e. not in a wiggle trace, so that we can visually confirm that the structural problem is properly solved. The display in Figure 6.38 also enables us to detect where a relative increase or decrease in acoustic impedance contrast across any boundary is present, so that they can be adjusted accordingly in the initial earth model. This would imply that we start with initial guesses that are closer to the solution.
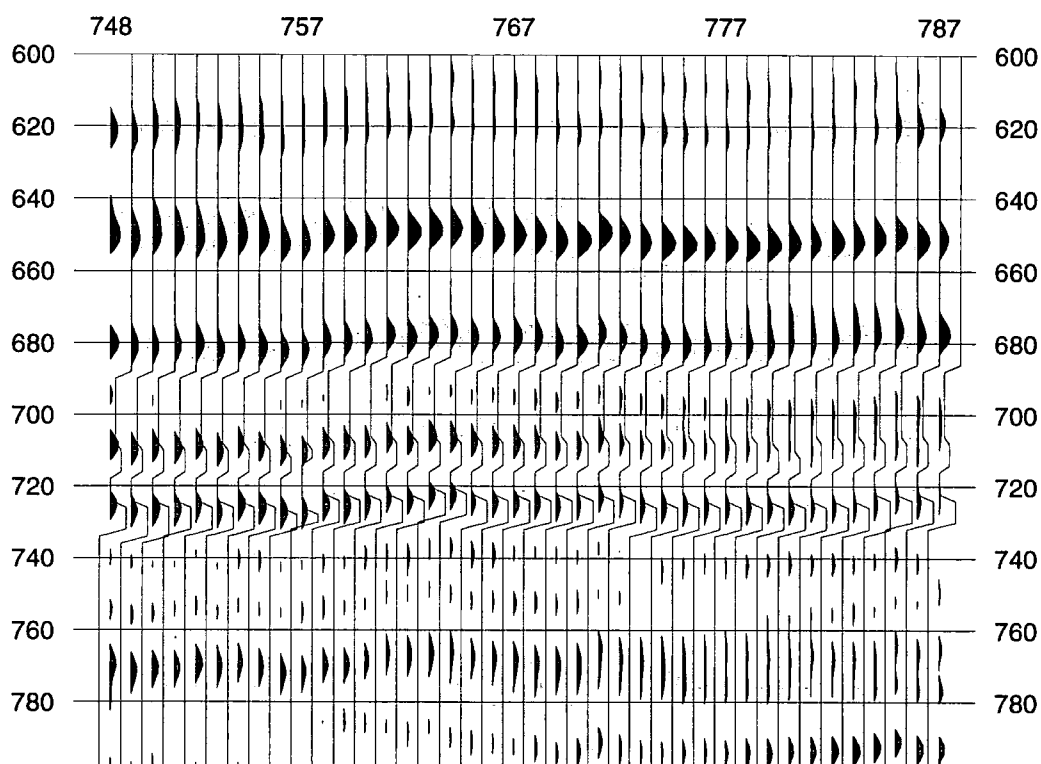
Figure 6.38. The initial earth model section for the observed seismic traces from Line 1973 (Figure 6.37). Note that the vertical axis represents the two-way travel-time in ms.

The initial guess section comprising the 40 earth models is shown in Figure 6.39, their synthetic seismograms section is shown in Figure 6.40, and the corresponding initial error traces are shown in Figure 6.41. The error traces section will give an indication to the degree of improvement of the final impedance solution obtained after the inversion of all 40 seismic traces. During the inversion of these data the top and bottom layers were allowed to vary over a wide range, but the other layers were constrained to vary within 3500-9000 $gcm^{-3}ms^{-1}$.
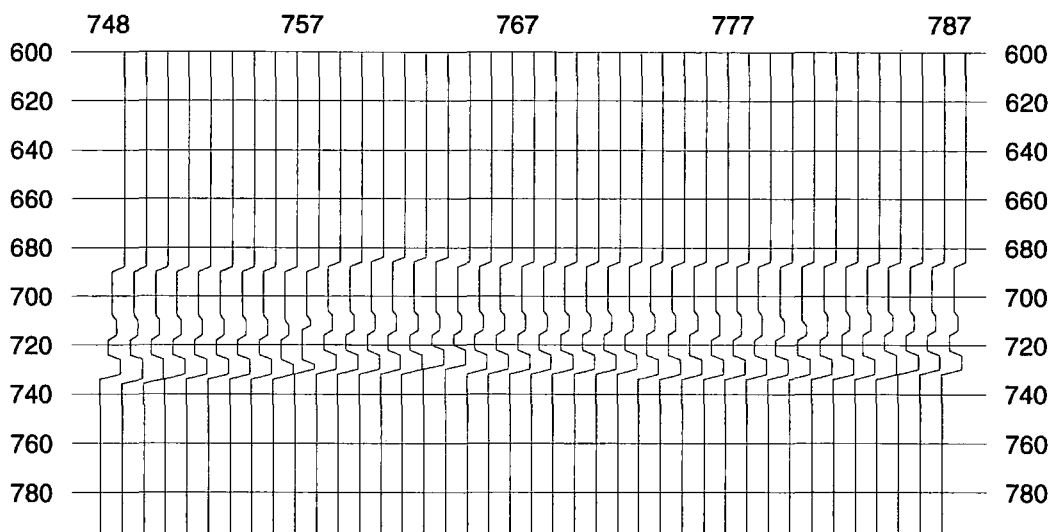
162

Figure 6.39. The initial earth model section of Figure 6.38 displayed without the observed seismic traces. Note that the vertical axis represents the two-way travel-time in ms.
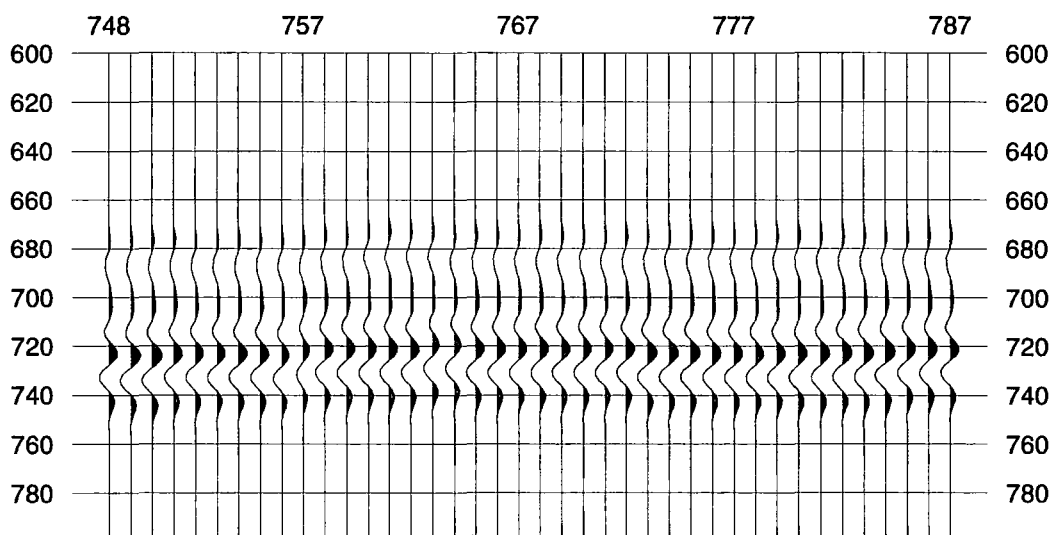
Figure 6.40. The initial synthetic seismograms for the earth model section in Figure 6.39. Note that the vertical axis represents the two-way travel-time in ms.
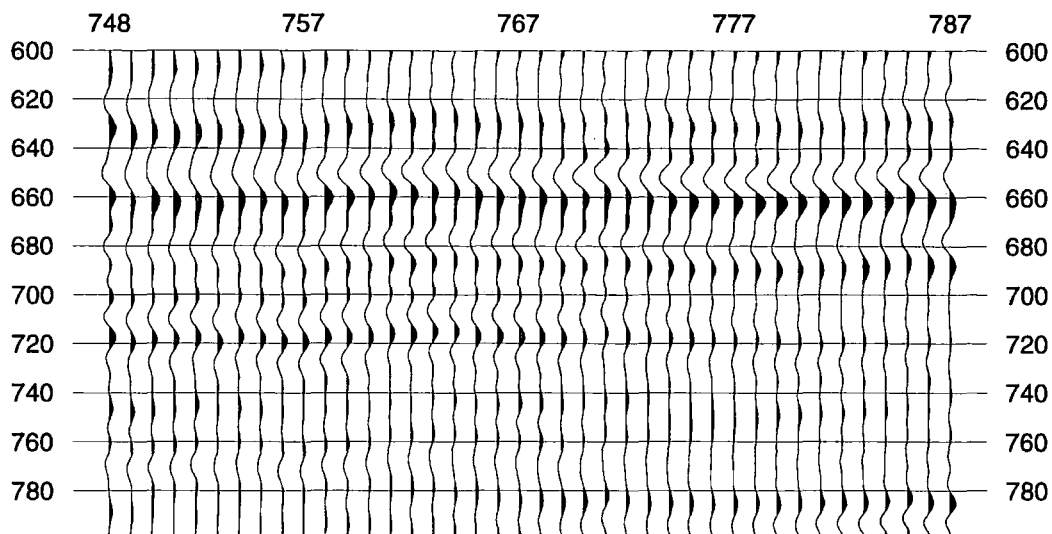
Figure 6.41. The initial error traces corresponding to the synthetic seismograms in Figure 6.40. Note that the vertical axis represents the two-way travel-time in ms.

The final impedance section is shown in Figures 6.42, their synthetic seismograms are shown in Figure 6.43 and the corresponding error section is shown in Figure 6.44.
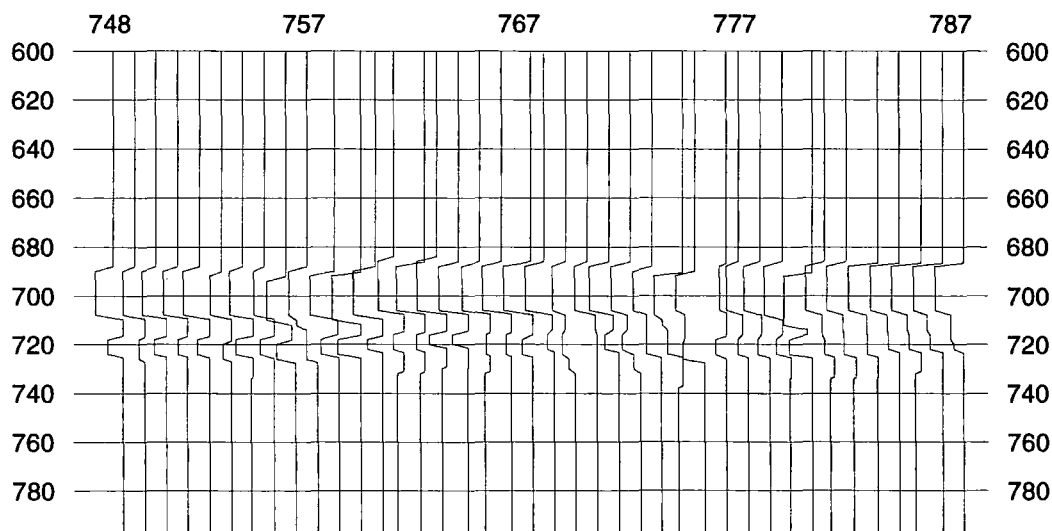
Figure 6.42. The final impedance solution for the initial earth model of Figure 6.39. Note that the vertical axis represents the two-way travel-time in ms.

164
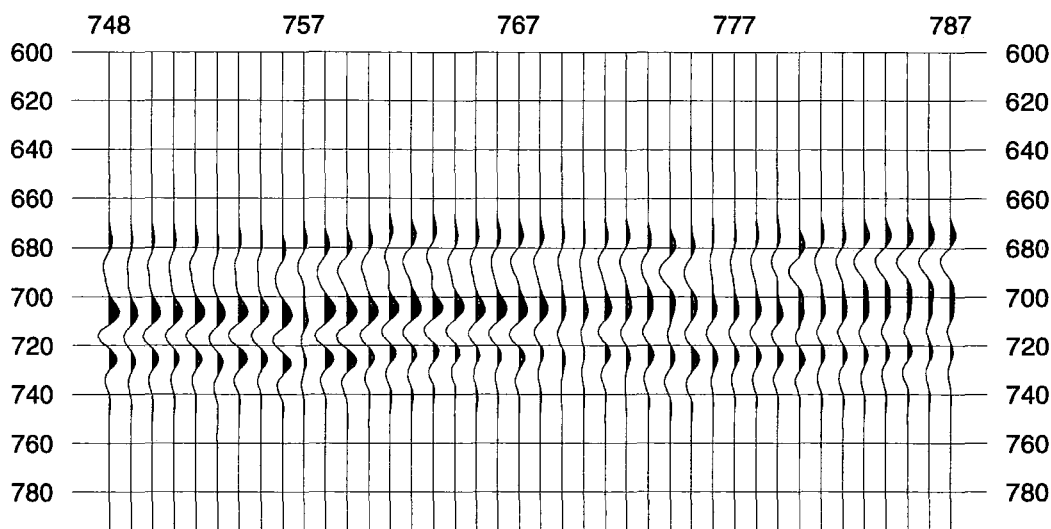
Figure 6.43. The final synthetic seismograms for the impedance solution of Figure 6.42. Note that the vertical axis represents the two-way travel-time in ms.
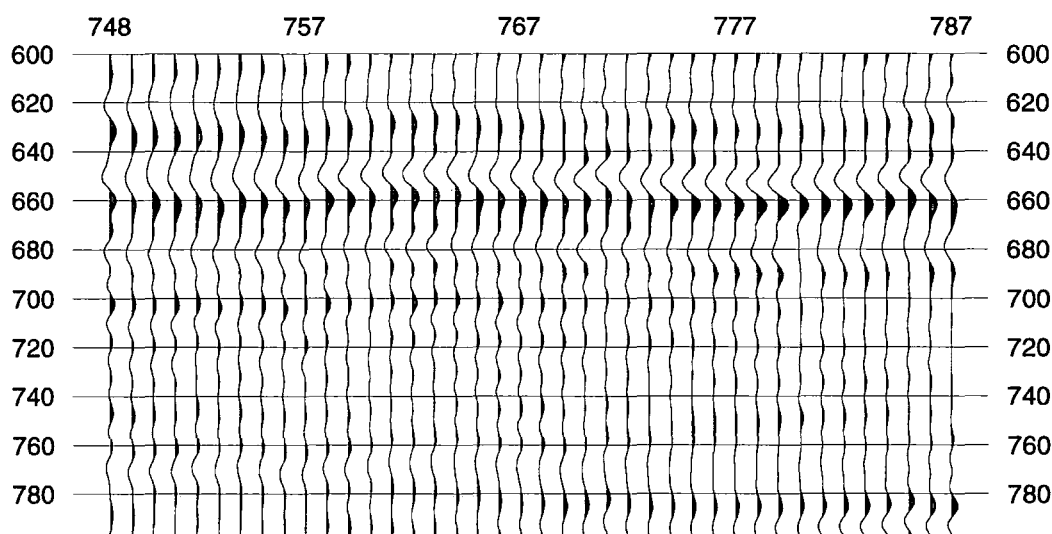
Figure 6.44. The final error traces corresponding to the synthetic seismograms of Figure 6.43. Note that the vertical axis represents the two-way travel-time in ms.

The inversion results shown in Figure 6.42 could be interpreted as follows. The upper shale cover is persistent across the section. The sand body at 710 ms is present at those traces where it was originally thought to exist. However, the shale layer that underlies the upper sand body seems to terminate at CDP 779, and this sand body merges with the lower sand at CDP traces 780-787. There is no shale layer that underlies the lower sand, so that the lower sand body at 730 ms continues to lower depths.

6.8.2. Example 1 of inversion for impedance from Line 1977

The observed seismic data for this example are shown in Figure 6.45 which consists of 61 seismic traces representing CDPs 1612-1661. The seismic event at about 730 ms two-way travel-time is interpreted as positive reflection from a Chadra sand body that is overlain by a shale cover. The top of the shale gives the negative reflection at about 720 ms. The Chadra sand body overlies a lower impedance layer which in turn overlies a possible limestone bed that is composed of two layers with the lower one having higher impedance, so that there is a two step increase for the limestone in the acoustic impedance profile in this model.
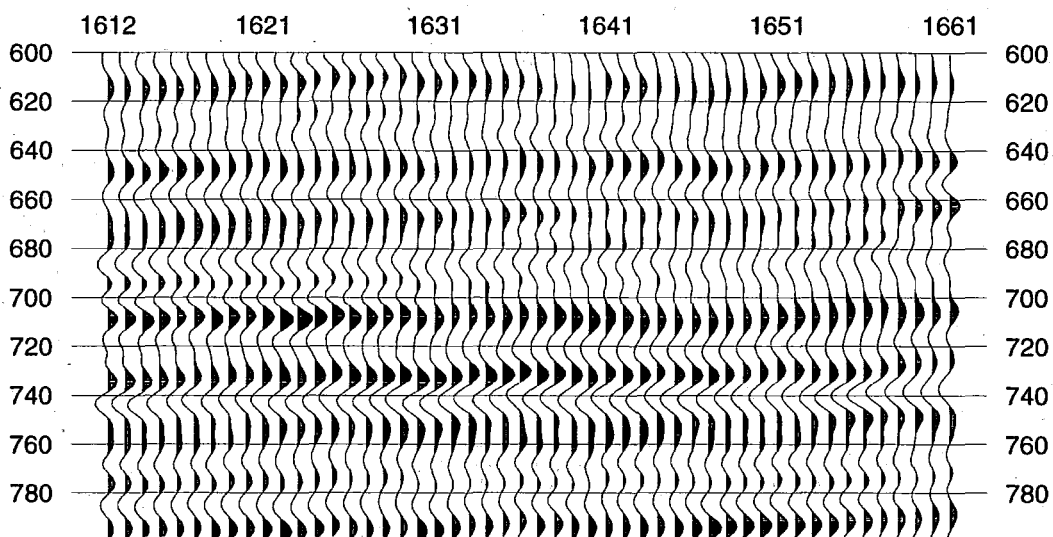
Figure 6.45. The observed seismic data of CDPs 1612-1661 in Line 1977. The seismic event (peak) at about 730 ms is interpreted as positive reflection from a Chadra sand body, and the broad positive reflection at 750-760 ms is a two step limestone bed. The two events are separated by a low impedance layer that has a contact with the Chadra sand at 740 ms. Note that the vertical axis represents the two-way travel-time in ms.

The initial earth model section is shown in Figures 6.46a and 6.46b super-imposed on the observed seismic traces. The initial earth model section is also shown separately in Figures 6.47a and 6.47b. Although the two successive positive events between 750 and 760 ms are possible limestone layers, they are actually assigned impedance values that are comparable with sandstone impedance. This is also to investigate how the program would deal with such situations. The wavelet used in the inversion of this example is the optimum seismic wavelet obtained in section 6.7.

Figure 6.46a. The initial earth model section for CDPs 1612-1636 and the corresponding observed. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.46b. The initial earth model section for CDPs 1637-1661 and the corresponding observed traces. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.47a. The initial earth model section for CDPs 1612-1636. These impedance profiles are also displayed in Figure 6.46a. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.47b. The initial earth model section for CDPs 1637-1661. These impedance profiles are also displayed in Figure 6.46b. Note that the vertical axis represents the two-way travel-time in ms.

The initial guess synthetic seismogram section is shown in Figure 6.48, and the corresponding initial error traces are given in Figure 6.49. The error traces section shows a consistent error at about 750 ms in all the error traces.
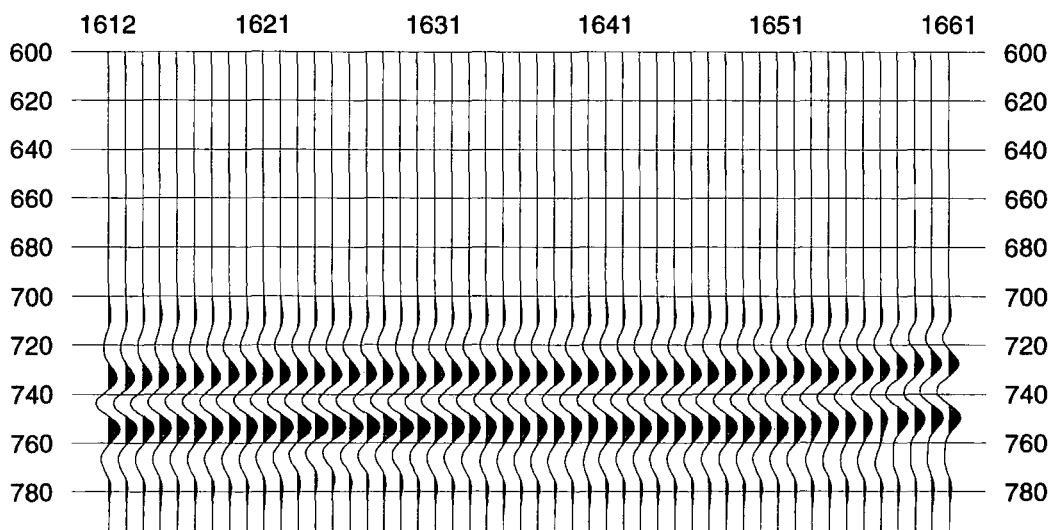
Figure 6.48. The initial synthetic seismograms for the earth model section of Figures 6.47a and 6.47b. Note that the vertical axis represents the two-way travel-time in ms.
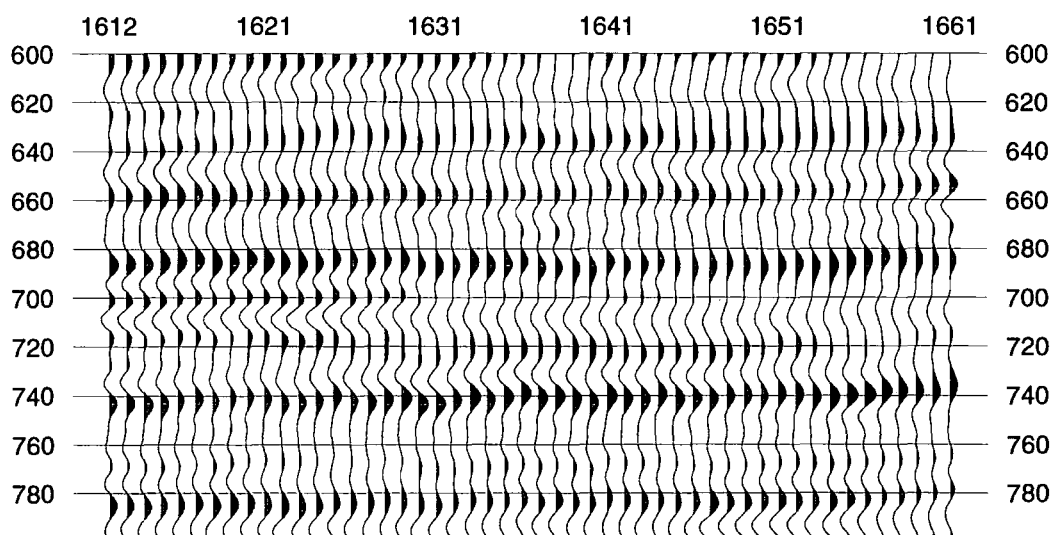
Figure 6.49. The error section corresponding to the initial earth model synthetic seismograms in Figure 6.48. Note that the vertical axis represents the two-way travel-time in ms.

The final impedance solution section is shown in Figures 6.50a and 6.50b, and its seismic response section is given in Figure 6.51 while its final error traces section is given in Figure 6.52. Clearly the seismic event at 730 ms which was interpreted as a sand shows as a prominent sand body in the final impedance section. The underlying shale is also present across the section. The limestone layer, however, seems to become more sandy in some places. This actually agrees with the known distribution of this limestone in this area. That is, the continuous limestone layer is somewhat below this level, and at this level the limestone is more discontinuous.

Figure 6.50a. The final impedance section for CDPs 1612-1636. The sand layer that starts at about 730 ms appears to be a continuous sand body. The limestone layer, however, becomes more sandy to the left of the section. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.50b. The final impedance section for CDPs 1637-1661. The sand layer that starts at about 730 ms appears to be a continuous sand body. The limestone layer, however, becomes more sandy to the left of the section. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.51. The final seismic solution section for CDPs 1612-1661. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.52. The error traces section corresponding to the seismic solution in Figure 6.51. Note that the vertical axis represents the two-way travel-time in ms.

6.8.3. Example 2 of inversion for impedance from line 1977

In this example we are interested in a sand body and two limestone layers. All three layers are interbedded with shale. The sand body is overlain by shale, and the lower limestone layer is underlain by shale. The initial earth model section is shown in Figures 6.53a and 6.53b, and also in Figures 6.54a and 6.54b where the earth model traces are superimposed on the observed seismic traces. The observed seismic traces are shown in Figure 6.55, where the sand body is represented by the positive reflection at about 705 ms two-way travel-time, the first limestone layer is the positive seismic event at about 735 ms and extends from CDP 1536 to CDP 1581, and it

seems to thin out towards the left and right of the section. The second (lower) limestone is interpreted to be positive seismic reflection at about 750 ms.

Figure 6.53a. The initial earth model section for example 2 in Line 1977 for CDPs 1536-1560. The upper sand at 705 ms is overlain by shale, and the two limestone layers below are underlain by shale. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.53b. The initial earth model section for example 2 in Line 1977, CDPs 1561-1585. The upper sand at 705 ms is overlain by shale, and the two limestone layers below are underlain by shale. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.54a. The same initial earth model section in Figure 6.53a superimposed on the corresponding observed seismic traces of Figure 6.55. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.54b. The same initial earth model section in Figure 6.53b superimposed on the corresponding observed seismic traces of Figure 6.55. Note that the vertical axis represents the two-way travel-time in ms.

173

Figure 6.55. The observed seismic traces CDP 1557-1581 for example 2 from Line 1077. The positive seismic event at 705 ms is a sand body. Note that the vertical axis represents the two-way travel-time in ms.

The synthetic seismograms of the initial earth model section are given in Figure 6.56 and the corresponding error traces in Figure 6.57. We observe that much of the error in the time window of interest is consistently concentrated at the level of the shale between the lower sand and the limestone.
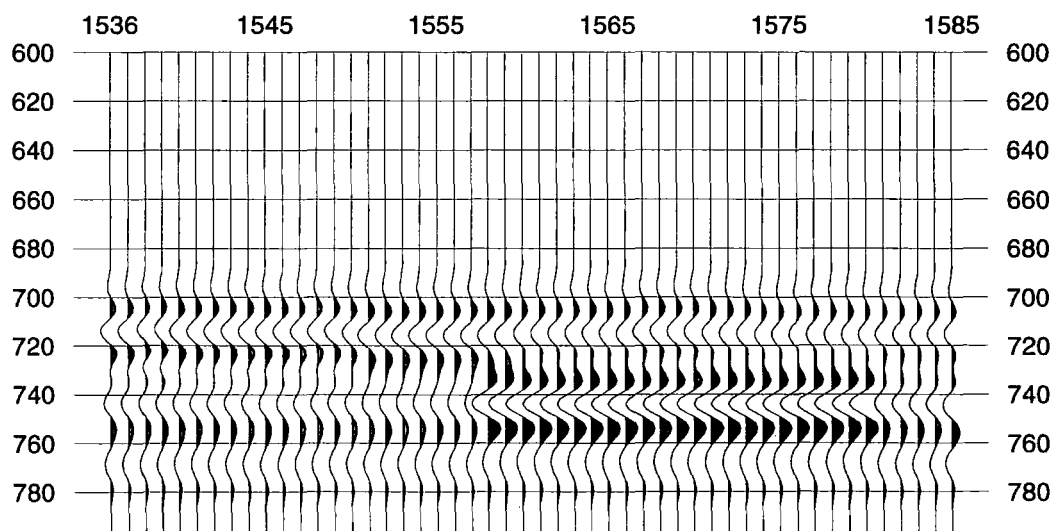
Figure 6.56. The synthetic seismograms of the initial earth model traces in Figures 6.53a and 6.53b. Note that the vertical axis represents the two-way travel-time in ms.
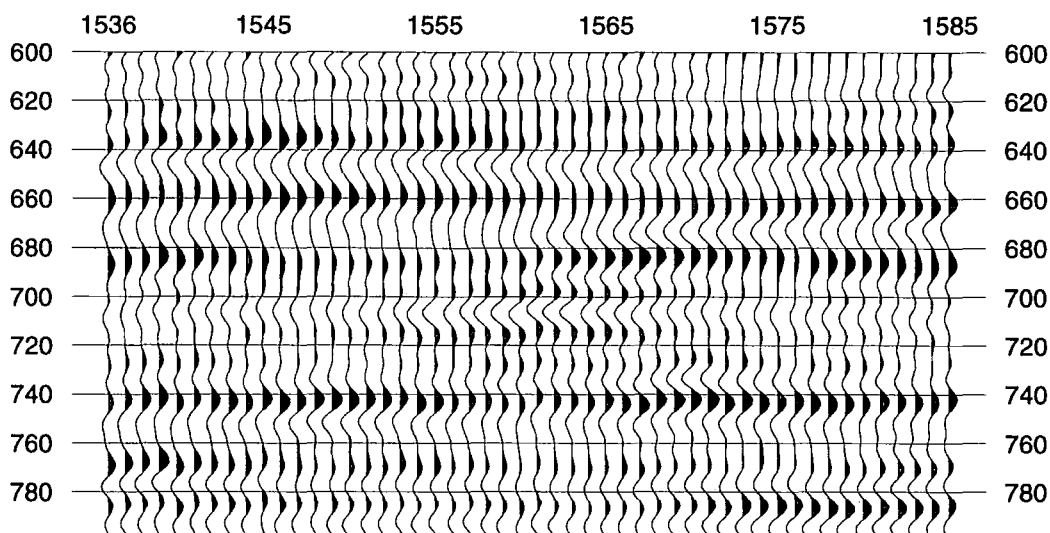
Figure 6.57. The error traces of the synthetic seismograms of Figure 6.56. Note that the vertical axis represents the two-way travel-time in ms.

The impedance inversion results are given in Figures 6.58a and 6.58b, the final synthetic seismograms are given in Figure 6.59, and the corresponding error traces are given in Figure 6.60.
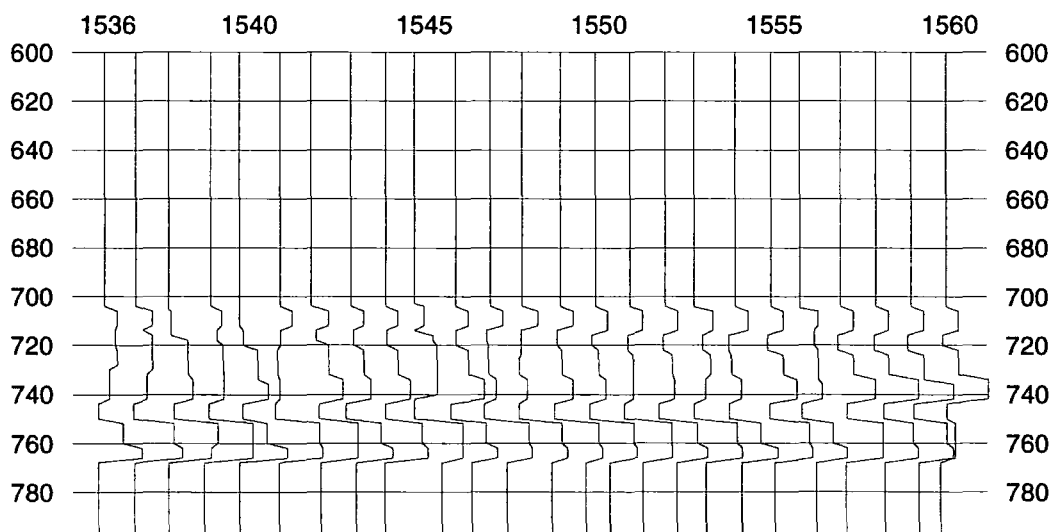
Figure 6.58a. The final impedance solution for example 2 on Line 1977, CDPs 1536-1560. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.58b. The final impedance solution for example 2 on Line 1977, CDPs 1561-1585. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.59. The final seismic solution section corresponding to the impedance solution in Figures 6.58a and 6.58b. Note that the vertical axis represents the two-way travel-time in ms.

176
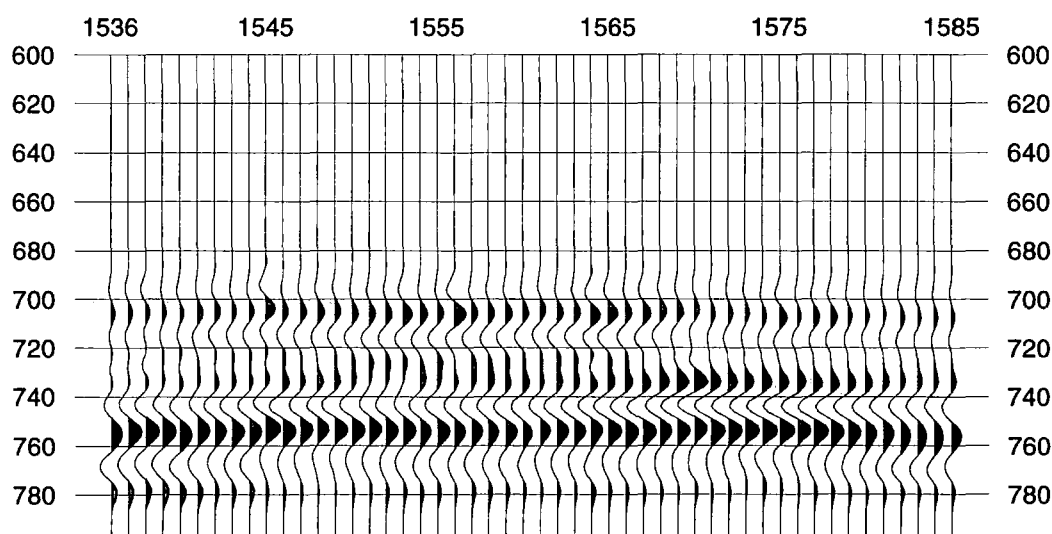
Figure 6.60. The error section of Figure 6.59. Note that the vertical axis represents the two-way travel-time in ms.

Similar to the results of the first example on this line, there is strong evidence for the existence of the sand on top of the limestone, but the shale below it seems to disappear in some places. Also, the upper limestone layer changes to sand on some of the traces, but the lower limestone is more extensive, and is present in almost all the traces.

6.9. Another approach to optimising the wavelet parameters at the wells YY31 and YY04

The various synthetic examples that contain noise in the previous chapter suggest that when random noise is present in the observed seismic trace the impedance values resulting from the inversion process should not be taken as the correct values representing the impedance of the subsurface layers. Since at a well location it can be safely assumed that the impedance values of the subsurface layers are already known, it becomes logical, when optimising the wavelet parameters at a well location, to omit the step of inversion for impedance and only invert for wavelet parameters and then boundary locations. We then repeat this sequence until convergence is achieved. Furthermore, to minimise the influence of noise we include in the earth model only those layers to which we can attach a strong reflection (Brown et al.; 1989).

6.9.1. A new optimum wavelet at well YY31

Well YY31 ties the observed seismic trace at 580 ms two-way travel-time, and measures an impedance section having 224 ms thickness. The initial guess earth

177

model used is the same 12-layer earth model shown in Figure 6.14. The initial wavelet parameters are those obtained in section 6.2 which has the parameters w_{ini} = [6 38 50 80 109000 109000 -0.2 0.115 0]^T. To obtain an optimum wavelet, we first inverted for the wavelet parameters, then inverted for boundary locations, and finally inverted for the wavelet parameters.

In the first wavelet parameters inversion the initial E_{rel} = 47%, then after eight iterations E_{rel} = 45%. Figure 6.61 shows the results of the first wavelet parameters inversion, where wavelet number 1 is the initial guess wavelet of section 6.2, and wavelet number 9 is the final solution wavelet for this inversion run. This final solution wavelet has the parameters w_{sol} = [4 24 70 74 109000 109000 -0.209 0.115 0]^T.
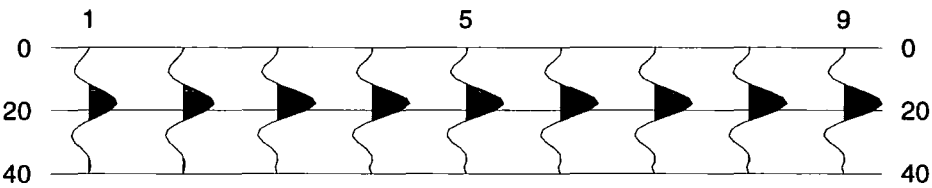
Figure 6.61. The results of the eight iterations of the first wavelet parameters inversion at well YY31. Wavelet number 1 is the initial guess wavelet given by the parameters w_{ini} = [6 38 50 80 109000 109000 -0.2 0.115 0]^T, and wavelet number 9 is the final optimised wavelet for this inversion run. This wavelet has the parameters w_{sol} = [4 24 70 74 109000 109000 -0.209 0.115 0]^T. Note that the vertical axis represents the two-way travel-time in ms.

The next inversion run was for boundary locations. This reduced E_{rel} to 34.47% in three iterations. The final impedance profiles for this inversion run are given in Figure 6.62. This was the only boundary locations inversion needed and, as illustrated by Figure 6.62, all boundary locations were adjusted by 2 ms, except for boundary 10 which was adjusted by 4 ms.

The final inversion run needed to obtain an optimum wavelet was wavelet parameters inversion. This inversion run reduced E_{rel} to 34.37% in nine iterations. The resulting wavelets are shown in Figure 6.63 where wavelet number 10 is the optimum wavelet obtained at well YY31 and has the parameters w_{sol} = [3.9 24 70 74 109022 108993 -0.206 0.115 0]^T.

Figure 6.62. The impedance profiles for the only boundary locations inversion needed when optimising the wavelet at well YY31. The initial guess impedance profile is the dashed line with cross marks, and the solution impedance profile is the solid line. Note that all boundary locations are adjusted by 2 ms (one sample interval), except for the 10-th boundary location which was adjusted by 4 ms.

Figure 6.63. The results of the nine iterations of the final wavelet parameters inversion at well YY31. The optimum wavelet obtained is wavelet number 10 which has the parameters $w_{sol} = [3.9\ 24\ 70\ 74\ 109022\ 108993\ -0.206\ 0.115\ 0]^T$. Note that the vertical axis represents the two-way travel-time in ms.

6.9.2. A new optimum wavelet at well YY04

Well YY04 ties the observed seismic trace at 610 ms two-way travel-time, and measures an impedance section having 234 ms thickness. The initial wavelet parameters were obtained from the parameterisation of the Wiener wavelet estimated at this well. The parameterised wavelet was determined in section 6.3 to be $\mathbf{w}_{ini}$ = [17 32 42 67 113330 113330 0.1 0.12 0.]^T. The initial guess earth model was determined in section 6.6 and is shown in Figure 6.29. In this earth model the impedance profile in well YY04 is represented by 8 layers.

Similar to section 6.9.1, we only needed to invert for the wavelet parameters, then boundary locations, and finally for wavelet parameters to obtain a new optimum wavelet in well YY04.

The first wavelet parameters inversion results are shown in Figure 6.64. The initial E_{rel} = 58% which was reduced to 51% in 38 iterations. The final wavelet for this inversion run had the parameters $\mathbf{w}_{sol}$ = [21 36 60 66 113330 113330 0.7962 0.115 0]^T.
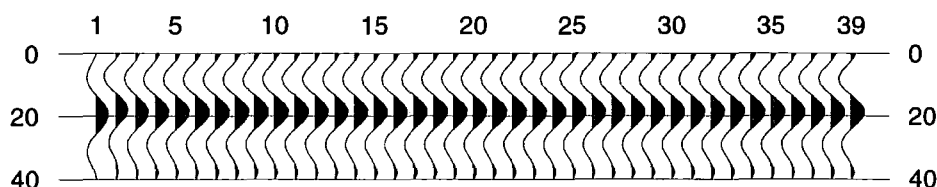
Figure 6.64. The results of the 38 iterations of the first wavelet parameters inversion at well YY04. Wavelet number 1 is the initial guess wavelet given by the parameters $\mathbf{w}_{ini}$ = [17 32 42 67 113330 113330 0.1 0.12 0.]^T. Wavelet number 39 is the final optimised wavelet for this wavelet parameters inversion run, this wavelet has the parameters $\mathbf{w}_{sol}$ = [21 36 60 66 113330 113330 0.7962 0.115 0]^T. Note that the vertical axis represents the two-way travel-time in ms.

We next invert for the boundary locations. This was the only boundary locations inversion needed, and it reduced E_{rel} to 45.74% in three iterations. The impedance profiles for this inversion run are shown in Figure 6.65, where it can be noticed that only two boundary locations, namely boundary locations 3 and 4, were each adjusted by 2 ms, or one sample interval, which represents the minimum shift a single boundary could be adjusted.
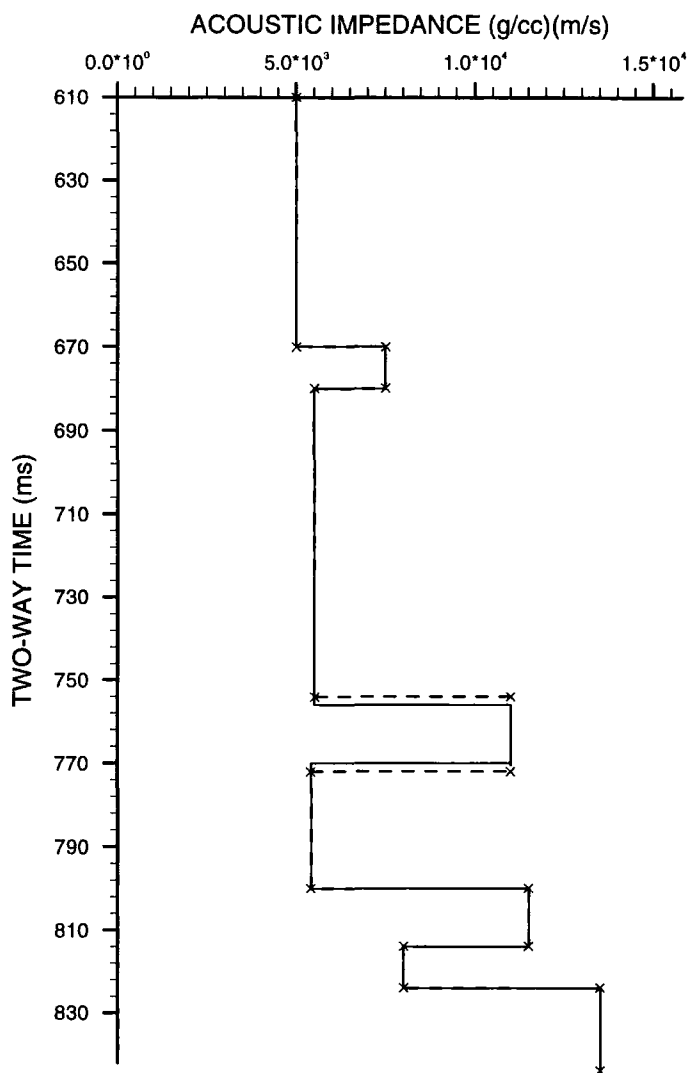
Figure 6.65. The impedance profiles for the only boundary locations inversion needed when optimising the wavelet at well YY04. The initial guess impedance profile is the dashed line with cross marks, and the solution impedance profile is the solid line. It can be observed that only boundary locations 3 and 4 are adjusted by 2 ms (one sample interval) each. This represents the minimum shift a single boundary can be adjusted.

The last inversion needed to obtain an optimum wavelet in YY04 was for wavelet parameters. This inversion run reduced E_{rel} to 40.79% in 43 iterations. The resulting wavelets are given in Figure 6.66, where wavelet number 44 is the optimum wavelet obtained at well YY04, and has the parameters w_{sol} = [16.4 45.1 67.2 81.8 113330 113330 0.3128 0.118 0]^T.
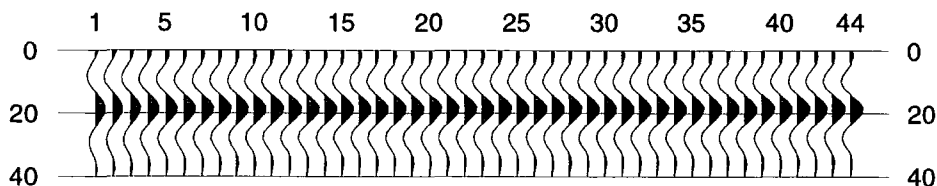
181

Figure 6.66. The results of the nine iterations of the final wavelet parameters inversion at well YY04. The optimum wavelet obtained is wavelet number 44 which has the parameters w_{sol} =[16.4 45.1 67.2 81.8 113330 113330 0.3128 0.118 0]^T. Note that the vertical axis represents the two-way travel-time in ms.

6.10. Inversion for impedance around well YY31 on Line 1973

Well YY31 is located on CDP 555 on Line 1973. A seismic section comprising of 41 seismic traces was chosen with the well location in the middle, i.e. the 21-st trace. To invert for acoustic impedance we use the initial guess model data of well YY31. The time window for the 41 seismic traces is 320 ms two-way travel-time. The seismic window starts at 580 ms and ends at 900 ms, so that it extends to a lower seismic time than what well YY31 measured. This means that inversion for impedance covers 96 ms more of the Augila Limestone section than the well has measured, so that the initial guess impedance section has 18 layers instead of the 12 layers in well YY31 impedance profile of Figure 6.14.

The initial guess impedance section is shown in Figure 6.67, where for the time window 580-804 ms we strictly follow the initial guess of well YY31, but for the lower part we attach to every strong positive reflection (peak) a boundary location across which there is an increase in impedance, and attach to every strong negative reflection (trough) a boundary location across which there is a decrease in impedance. The resulting initial guess impedance section has 18 layers and is shown in Figure 6.67.

The observed seismic traces are shown in Figure 6.68 where well YY31 is located at the middle trace, which is CDP 555. The initial guess seismic response section is shown in Figure 6.69. This seismic response section is generated using the initial guess earth model of Figure 6.67 and the seismic wavelet which was optimised in section 6.9.1.

182

Figure 6.67. The initial guess impedance section, or earth model section, that constitutes the input to the inversion of the observed section of Figure 6.68 around well YY31. The well is located at CDP 555, and measures the time window 580-804 ms two-way travel-time. The earth model data strictly follow the impedance profile of well YY31 given in Figure 6.14. The seismic time window was further extended, into the Augila Limestone, to 900 ms two-way travel-time. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.68. The observed seismic section used to invert for impedance around well YY31. This section is part of seismic Line 1973 between CDP 535 and CDP 575. The start of the well impedance measurements tie the seismic data at 580 ms and extends to 804 ms two-way travel-time. The 96 ms of section time between 804 ms and 900 ms is an extra section extended into the Augila Limestone. The initial guess earth model section for this observed seismic data, shown in Figure 6.67, has 18 layers. Note that the vertical axis represents the two-way travel-time in ms.

183

Figure 6.69. The initial guess seismic response section around well YY31. The well is located at CDP 555. This section is generated from the initial guess earth model of Figure 6.67 and the seismic wavelet optimised in section 6.9.1. Note that the vertical axis represents the two-way travel-time in ms.

The final impedance section is given in Figure 6.70 and its corresponding seismic solution section is given in Figure 6.71. A comparison between the seismic solution section in Figure 6.71 and the observed seismic section in Figure 6.68 reveals that the positive reflection from the top of Augila Limestone across the section, at about 800 ms, has been properly reproduced in Figure 6.71. That is, the changes in the reflection amplitude in the seismic solution section from the top of the limestone resembles that in the observed seismic section. This is echoed in the final impedance solution section of Figure 6.70, at 800 ms, as an increase in the impedance contrast across the boundary representing the Augila Limestone for the seismic traces with large reflection amplitude. Indeed the seismic response of all the limestone section below 800 ms in Figure 6.71 closely resembles the corresponding observed seismic section of Figure 6.68.

Figure 6.70. The final impedance solution section around well YY31. The Augila Limestone starts at about 800 ms and continues to 900 ms two-way travel-time. The impedance contrast across the boundary represented by the top of Augila varies from one profile to the next. There is a large impedance contrast for the middle profiles that surround the well at CDP 555. The impedance contrast decreases to the left and right. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.71. The seismic solution section of the final impedance solution of Figure 6.70. The seismic events in this seismic section should be compared to the seismic events in the observed section in Figure 6.68. Note that the vertical axis represents the two-way travel-time in ms.

Similar comparisons could be made for the shallower seismic events. For example, the positive-negative reflection character around 660 ms, which is most prominent at CDP's 560-574. However, it should be mentioned that the lack of exact fit between the seismic solution section of Figure 6.71 and the observed seismic section of Figure 6.68 could be attributed to the generally poor signal-to-noise ratio of the observed seismic data, so that only the strong reflection events, such as the

reflection from the top of Augila Limestone, could be inverted for with reasonable results.

6.11. Inversion for impedance around well YY04 on Line 1977

Well YY04 is located at CDP 1093 on Line 1977. To invert for impedance around well YY04, a 41 CDP traces seismic section was chosen with the well located at the middle trace. This observed seismic section extends from CDP 1073 to CDP 1113 and ties well YY04 at 610 ms two-way travel-time. The time window chosen starts from 610 ms to 900 ms, i.e. 290 ms. This implies that the Augila Limestone is represented by a thicker time section than what has been measured by YY04. The observed seismic section is shown in Figure 6.72.

the initial guess impedance section is shown in Figure 6.73. The initial guess impedance profiles of Figure 6.73 were obtained from the parameterised impedance profile of YY04 given in Figure 6.29. But because the observed seismic section covers more limestone section, the initial guess impedance profiles have 10 layers instead of 8 layers as given in Figure 6.29. The first 8 layers of each of the initial guess impedance profiles were strictly obtained from the parameterised impedance profile of well YY04.

Figure 6.72. The observed seismic section used to invert for impedance around well YY04. This section is part of Line 1977 extending from CDP 1073 to CDP 1113. The well is located at the middle trace which is CDP 1093. The well impedance measurements start at 610 ms and extends for 234 ms to 844 ms two-way travel-time. The 56 ms of section time between 844 and 900 ms is an extra section extended into the Augila Limestone. The initial guess earth model section for this observed seismic section has 10 layers and is given in Figure 6.73. Note that the vertical axis represents the two-way travel-time in ms.

186

Figure 6.73. The initial guess impedance section, or earth model section, that constitutes the input to the inversion of the observed seismic section of Figure 6.72 around well YY04. The well is located at CDP 1093, and measures the time window 610-844 ms two-way travel-time. The earth model profiles are obtained from the impedance profile of well YY04 given in Figure 6.29. The seismic time window is further extended into the Augila Limestone to 900 ms. Note that the vertical axis represents the two-way travel-time in ms.

The initial guess seismic response section is shown in Figure 6.74. This section was generated from the initial guess impedance profiles of Figure 6.73 and the seismic wavelet that was optimised at YY04 in section 6.9.2.

The final impedance solution section is given in Figure 6.75, and the corresponding seismic solution section is given in Figure 6.76. The trace to trace amplitude variation of the shallow seismic event at 670 ms two-way travel-time,
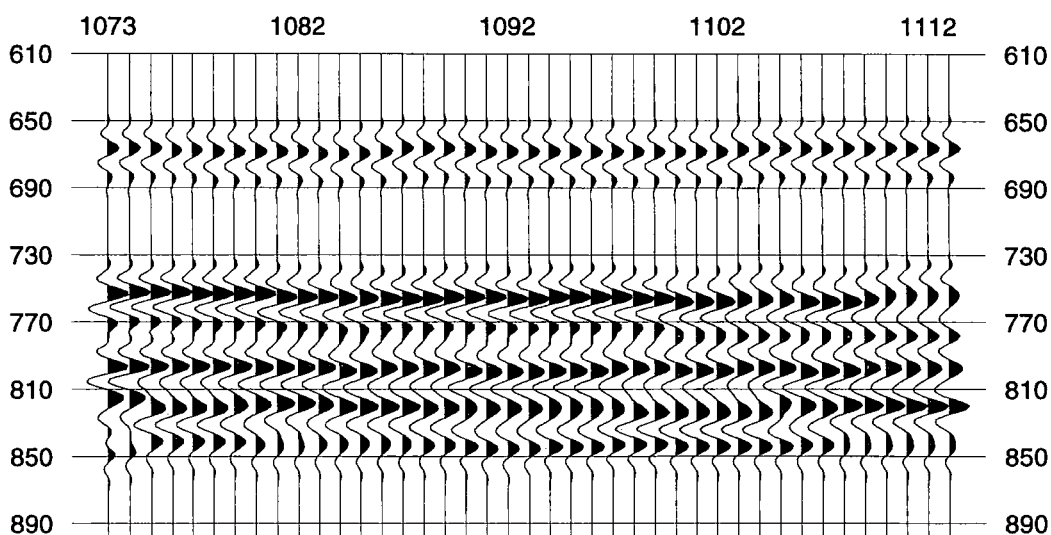
Figure 6.74. The initial guess seismic response section around well YY04. The well is located at CDP 1093. This section is generated from the initial guess earth model of Figure 6.73 and the seismic wavelet optimised in section 6.9.2. Note that the vertical axis represents the two-way travel-time in ms.

187

which is the positive reflection from the top of Chadra A sand, in the seismic solution section in Figure 6.76, is almost the same as the corresponding seismic event in the observed seismic section in Figure 6.72. The final impedance solution section shows that this sand body is thinnest at CDP 1088 and it thickens to the left and right. Indeed the high amplitude of this seismic event is due to the sand body thickness at the traces around CDP 1088 being at the tuning thickness. Then as the thickness of the sand increases in the traces on the left and right of CDP 1088, the resulting amplitude of the seismic reflection decreases. Note that the solution impedance profiles 1100 to 1112 in Figure 6.75 show a two-step increase of impedance in the sand body. This is an indication of the thickness increase in the sand body, and that the base of the sand body in these impedance profiles is at a somewhat lower level, possibly at 690 ms, which was not parameterised in the initial guess impedance profiles. Thus the base of the sand body is not present in the solution impedance profiles 1100 to 1112.

The lower part of the initial guess impedance section that starts at about 750 ms two-way travel-time represents the Augila Limestone. The seismic solution for this part is given in Figure 6.76, and resembles the corresponding observed seismic section given in Figure 6.72. This indicates that the inversion process has determined a final impedance solution, shown in Figure 6.75, that closely represents the limestone section impedance. This is possible because we have included in the initial guess impedance section only the layers that have a high signal-to-noise ratio.

Figure 6.75. The final impedance solution section for inversion around YY04. The shallow layer is the Chadra A sand. It is thin at the middle traces and thicker to the left and right. The Augila Limestone starts at about 750 ms two-way travel-time. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.76. The seismic solution section corresponding to the impedance solution in Figure 6.75. The high amplitude associated with the middle traces for the shallow event is due to the thickness of the Chadra A sand being at, or near, the tuning thickness. The decrease in amplitude of the shallow event to the left and right of the middle traces is an indication of thickness increase of the sand body. The lower part of the section, starting at about 750 ms, describes the Augila Limestone and closely resembles the corresponding section on the observed seismic section in Figure 6.72. Note that the vertical axis represents the two-way travel-time in ms.

6.12. Inversion for impedance around the intersection of Lines 1973 and 1977

Inverting for impedance at the intersection of Lines 1973 and 1977 would test the reliability of the inversion process in a location where the seismic data were recorded in two different directions, and the seismic wavelet used in the inversion of each line was estimated, and later optimised, at a different well location so that each wavelet could be slightly different from the other.

The inversion for impedance at the intersection of the two lines is performed on two sections each having 11 CDP traces. Each section is chosen so that the CDP trace at the intersection is located in the middle of the section, i.e. the 6-th seismic trace. For Line 1973 the seismic trace at the intersection is CDP 1042, and for Line 1977 it is CDP 1707. The seismic section from Line 1973 starts at CDP 1037 and ends at CDP 1047, and the seismic section from Line 1977 starts at CDP 1702 and ends at CDP 1712.

When generating the initial guess impedance sections we only considered the seismic events that could be correlated across the two observed seismic sections, so that we only consider those seismic events with good signal-to-noise ratio. This was necessary so that the comparison of the inversion results around the intersection is not effected by the noise in the observed seismic data. Such a seismic correlation produced two seismic events from two Chadra sands and three seismic events from the Augila Limestone layers. Thus the initial guess impedance section comprises of 8

189

layers, i.e. 7 interfaces. The initial impedance values for both sections are estimated using the impedance profiles of wells YY04 and YY31.

The observed seismic section used to invert for impedance on Line 1973 around the intersection with Line 1977 is shown in Figure 6.77. The middle trace, CDP 1042, is located at the intersection. The 8-layer initial guess impedance section used in the inversion is given in Figure 6.78, and its initial guess seismic response section is shown in Figure 6.79. The final impedance solution section is shown in Figure 6.80, and its final seismic solution section is given in Figure 6.81.

The observed seismic section used for inversion on Line 1977 around the intersection with Line 1973 is shown in Figure 6.82. The middle trace, CDP 1707, is located at the intersection. Similar to the intersecting seismic section on Line 1973, the initial guess impedance section comprises of 8 layers and is shown in Figure 6.83, and its initial guess seismic response section is shown in Figure 6.84. The final impedance solution section is shown in Figure 6.85, and its final solution section is given in Figure 6.86.

Figure 6.77. The 11-CDP observed seismic section which is part of Line 1073 used to invert for impedance around the intersection with Line 1977. The middle trace, CDP 1042, is the trace located at the intersection. Correlation with Line 1977 produced only 7 seismic events that have good signal-to-noise ratio. The 7 interfaces are shown in the 8-layer initial guess impedance section of Figure 6.78. Note that the vertical axis represents the two-way travel-time in ms.

190

Figure 6.78. The initial guess impedance section on Line 1973 containing 11 impedance profiles around the intersection with Line 1977. The impedance profile located at the intersection is profile number 1042. this 8-layer impedance section was generated from the correlation of 7 seismic events on the two lines that have a good signal-to noise ratio, and the two impedance profiles of wells YY04 and YY31. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.79. The initial guess seismic response section on Line 1973 around the intersection with Line 1977. The shallow part of the section, above 700 ms two-way travel-time, is the response of the three interfaces in the Chadra sands, and the lower part, below 750 ms, is the response of four interfaces in the Augila Limestone. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.80. The final impedance solution section corresponding to the observed seismic section on Line 1973 around the intersection with Line 1977. The impedance profile at the intersection is profile number 1042. This impedance section should be compared with the impedance solution section along Line 1977 given in Figure 6.85. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.81. The final seismic solution section of the impedance solution on Line 1973 shown in Figure 6.80. Note that the vertical axis represents the two-way travel-time in ms.

192

Figure 6.82. The 11-CDP observed seismic section which is part of Line 1077 used to invert for impedance around the intersection with Line 1973. The middle trace, CDP 1707, is the trace located at the intersection. Correlation with Line 1973 produced only 7 seismic events that have good signal-to-noise ratio. The 7 interfaces are shown in the 8-layer initial guess impedance section of Figure 6.83. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.83. The initial guess impedance section on Line 1977 containing 11 impedance profiles around the intersection with Line 1973. The impedance profile located at the intersection is profile number 1707. this 8-layer impedance section was generated from the correlation of 7 seismic events on the two lines that have a good signal-to noise ratio, and the two impedance profiles of wells YY04 and YY31. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.84. The initial guess seismic response section on Line 1977 around the intersection with Line 1973. The shallow part of the section, above 700 ms two-way travel-time, is the response of the three interfaces in the Chadra sands, and the lower part, below 750 ms, is the response of four interfaces in the Augila Limestone. Note that the vertical axis represents the two-way travel-time in ms.

Figure 6.85. The final impedance solution section corresponding to the observed seismic section on Line 1977 around the intersection with Line 1973. The impedance profile at the intersection is profile number 1707. This impedance section should be compared with the impedance solution section along Line 1973 given in Figure 6.80. Note that the vertical axis represents the two-way travel-time in ms.

194

Figure 6.86. The final seismic solution section of the impedance solution on Line 1977 shown in Figure 6.85. Note that the vertical axis represents the two-way travel-time in ms.

Comparing the two solution impedance profiles at the intersection, we observe that the thickness of the two shallow Chadra sand layers are noticeably different, but their corresponding observed traces are also different at the time level of these sands. This is due to the random noise contained in each of the observed traces. The impedance trends, however, are still largely preserved in each impedance profile, especially at Augila Limestone level. Indeed both the layer thicknesses and impedance trends of the limestone layers are similar in the two impedance profile. This indicates that even though the two observed seismic traces at the intersection of Lines 1973 and 1977 are so contaminated with noise that we can only correlate the seismic events in short intervals, their inversion results showed that we could obtain reasonable impedance results but not layer thickness results.

6.13. Conclusions

The following points could be deduced from the previous examples:

1. The observed seismic data used in this work has a poor signal-to-noise ratio, which could be severe at some interval.

2. Observing the differences in the error traces sections before and after the inversion would reveal the degree of fit of the final seismic solution section to the observed data section. The seismic solution section generally agrees with the observed data, but some error still remains.

195

3. The remaining error could be attributed to error in the wavelet since this wavelet was optimised at the well, which is some distance away in all the examples.

4. The final impedance solution always differs from the initial earth model. This implies that a better initial earth model is needed, which in turn requires a detailed knowledge about the subsurface geology of the area. This can only be achieved if a geologist who knows the area is available to participate during the development of the earth model.

5. Despite all the above, the inversion strategy and the computer program developed in this research produced encouraging results about Chadra sands delineation. But it is important to analyse more data, with more wells, and with the participation of a geologist.

7

CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

7.1. Conclusions

1. Using the Cholesky factorisation is both simple and numerically efficient. It also uses less computer time since no function values are evaluated unnecessarily.

2. The inversion of seismic data into impedance can be a very useful tool. However, it must be applied with care, since a poor initial guess or incorrectly chosen constraints could lead to misleading results.

3. A good initial guess depends largely on the availability of well data. In each of the two seismic lines available for this research, only one well was available, and they were 4 km apart. This has limited our ability to detect any changes in the shape of the wavelet away from the well. Although the two estimated Wiener wavelets were closely similar, there still remains the possibility that each wavelet could change in shape away from the well.

4. It is essential to involve a geologist that is familiar with the subsurface geology in the area. This will make it possible to arrive at initial earth models that better model the lateral variations in geology away from the well.

5. Using linear equality and inequality constraints proved to be a very useful and practical way to obtain solutions that accommodate desirable geologic information into the inversion process. This reduces the problem of non-uniqueness and increases the chance of converging to the correct impedance solutions.

6. Inversion at the two well locations showed that we can obtain a good match, i.e. with low error energy, between the well impedance profile and the impedance profile of the inversion solution. This is largely because the wavelet was estimated there, and a good initial guess could always be determined from the well data.

7. The real data inversion examples of chapter 6 showed that the Chadra sands could be delineated away from the two wells with reasonable results.

8. It is always more stable to invert those parts of the seismic trace that contain strong reflection energy. This is due to the low signal-to-noise ratio in the parts where not enough contrast is present across the interfaces that produce the reflection energy.

9. The impedance inversion results should not be considered as accurate measurements of the acoustic impedance profile at the observed seismic location; instead they should be interpreted in terms of their geologic feasibility, initial guess model limitations and observed seismic data reliability.

7.2. Suggestions for future work

Probably the most important suggestion that one could make is to reconsider the way we parameterise the wavelet. Parameterising the wavelet in the frequency domain resulted in a poorly conditioned inverse problem. This is due to the fact that the error energy function is highly sensitive to changes in the phase parameters while it is considerably less sensitive to the amplitude parameters. One remedy is to invert for the phase parameters separately from the frequency and amplitude parameters. That is, we optimise the phase spectrum independently from the amplitude spectrum.

Another possibility is to parameterise the seismic wavelet in the time domain. That is, we consider the sample amplitudes of the estimated Wiener wavelet as the wavelet parameters. This may produce a better conditioned inverse problem. This would be at the expense of determining more parameters, since in our case we will need to determine twenty parameters instead of nine. Also, increasing the number of

parameters increases the problem of non-uniqueness, but with a proper choice of constraints this could be minimised.

Furthermore, the assumption that the wavelet stays the same from one trace to the next, might not be a sufficiently correct one. I believe that much of the error energy in the final seismic solution could be attributed to changes in the wavelet from trace to trace. This would suggest that we should also consider inverting for the wavelet parameters in each trace along with boundaries and impedances.

References

Aki, K. and Richard, P. G., 1980, Quantitative Seismology. Vols. **1** and **2**. W. H.
    Freeman and Co.

Al-Sadi, H. N., 1980, Seismic Exploration. Berkhauser Verlag, Basel.

Backus, G. E. and Gilbert, J. F., 1967, Numerical application of a formalism for
    geophysical inverse problems. Geophys. J. R. astr. Soc., **13**, 247-276.

Backus, G. E. and Gilbert, J. F., 1968, The resolving power of gross earth data.
    Geophys. J. R. astr. Soc., **16**, 169-205.

Backus, G. E. and Gilbert, J. F., 1970, Uniqueness in the inversion of inaccurate gross
    Earth data. Phil. Trans. R. Soc. London, Ser. A266, 123-192.

Barr, F. T. and Weegar, A. A., 1972, Stratigraphic nomenclature of the Sirte Basin,
    Libya. Published by the Petroleum Exploration Society of Libya, Tripoli,
    Libya.

Bates, D. M. and Watts, D. G., 1988, Nonlinear Regression Analysis. Wiley.

Bath, M., 1974, Spectral Analysis in Geophysics. Elsevier Scientific Publishing Co.

Becquey, M., Lavergne, M. and Willm, C., 1979, Acoustic impedance logs computed
    from seismic traces, Geophy., **44**, 1485-1501.

Bilgeri, D. and Carlini, A., 1981, Non-linear estimation of reflection coefficients from
    seismic data, Geophy. Pros., **29**, 672-686.

Bjorck, A., 1990, Least squares methods. In Handbook of Numerical Analysis, vol. **1**.
    North-Holland.

Bracewell, R. N., 1986, The Fourier transform and its applications. McGraw-Hill
    International Editions.

Brigham, E. O., 1988, The fast Fourier transform and its applications. Printice-Hall,
    Englewood Cliffs.

Brown, R. L., McElhattan, W., Santiago, D. L., 1989, Wavelet estimation: An
    interpretive approach. Society of Exploration Geophysicists, Tulsa,
    Oklahoma.

Cooke, D. A. and Schneider, W. A., 1983, Generalized linear inversion of reflection
    seismic data. Geophy., **48**, 665-676.

Danielsen, V. and Karlsson, T. V., 1984, Extraction of signatures from seismic and
    well data, First Break 2(4), 14-21.

Dennis, J. E. and Schnabel, R. B., 1983, Numerical methods for unconstrained
    optimization   and nonlinear equations. Prentice-Hall Inc.

Dimri, V., 1992, Deconvolution and inverse theory. Application to geophysical problems. Elsevier.

Draper, N. R. and Smith, H. 1981, Applied Regression Analysis, second edition, McGraw-Hill, New York.

Fletcher, R., 1987, Practical methods of optimization. John Wiley and Sons.

Fox, C. G., 1987, An inverse Fourier transform algorithm for generating random signal of specified spectral form, Computer and Geosciences, **13**, 369-374.

Gill, P. E. and Murray, W., 1974, Newton-type method for unconstrained and linearly constrained optimization. Math. prog. **7**, 311-350.

Gill, P. E. and Murray, W., 1979, The computation of Lagrange multiplier estimates for constrained minimization. Math. prog., **17**, 32-60.

Gill, P. E., Murray, W., and Wright, M. H., 1981, Practical optimization. Academic Press.

Gill, P. E., Murray, W., and Wright, M. H., 1991, Numerical linear algebra and optimization, vol. **1**, Addison-Wesley.

Golub, G. H. and Van Loan, C. F., 1989, Matrix computations. The John Hopkins University Press.

Jackson, D. D., 1972, Interpretation of inaccurate, insufficient and inconsistent data. Geophy. J., **28**, 97-109.

Jupp, D. L. B. and Vozoff, K., 1975, Stable iterative methods for the inversion of geophysical data, Geophy. J. R. astr. soc. , **42**, 957-976.

Kanasewich, E. R., 1981, Time sequence analysis in geophysics. The University of Alberta Press.

Keys, R. G., 1986, An application of Marquardt's procedure to the seismic inverse problem. Proc. of the IEEE, **74**(3), 476-486.

Kokesh, F. P. and Blizard, R. B., 1959, Geometrical factors in sonic logging, Geophy. **24**, 64-76.

Labo, J., 1986, A practical introduction to borehole geophysics. SEG Publication.

Lavenberg, K., 1944, A method for the solution of certain nonlinear problems in least squares, Quarterly of applied mathematics, **2**, 164-168.

Lavergne, M. and Wills, C., 1977, Inversion of seismograms and pseudo velocity logs, Geophy. Pros., **25**, 231-250.

Lindseth, R. O., 1979, Synthetic sonic logs - a process for stratigraphic interpretation. Geophy., **44**, 3-26.

Lines, L. R. and Treitel, S., 1984, Tutorial: A review of least squares inversion and its application to geophysical problems. Geophy. Prosp., **32**, 159-186.

Lines, L. R. and Treitel, S., 1985, Wavelets, well logs and Wiener filters. First Break, **3**(8), 9-14.

Luenberger, D. G., 1984, Linear and nonlinear programming. Addison-Wesley Publishing Co.

Marquardt, D. W., 1963, An algorithm for least squares estimation of nonlinear parameters. J. Soc. Indust. Appl. Math., **11**, 431-441.

Menke, W., 1989, Geophysical data analysis: Discrete inverse theory. International Geophysics series, vol. **45**, Academic Press Inc.

More, J. J. and Sorensen, D. C., 1979, On the use of directions of negative curvature in modified Newton methods. Math. prog., **15**, 1-20.

Oldenburg, D. W., 1981, A comprehensive solution to the linear deconvolution problem. Gophy. J., **65**, 331-358.

Parker, R. L., 1977, Understanding inverse theory. Ann. Rev. Earth Planet. Sci., **5**, 35-64.

Parker, R. L., 1994, Geophysical inverse theory. Princeton University Press, Princeton, New Jersey.

Rider, M. H., 1986, The geological interpretation of well logs. Blackie, Glasgow, London.

Robinson, E. A., 1983, Seismic velocity analysis and the convolutional model. International Human Resources Development Corp.

Robinson, E. A. and Treitel, S., 198, Geophysical signal analysis. Prentice-Hall, Englewood Cliffs, New Jersey.

Robinson, R. M., 1974, Notes on the geology of Arida 'A' sandstones, Gialo Field, Libya. Internal geological report, Exploration Dept., Waha Oil Co.

Sain, K., and Kaila, K. L., 1994, Inversion of wide-angle seismic reflection times with damped least squares. Geophy., **59**, 1735-1744.

Scales, L. E., 1985, Introduction to non-linear optimization., MacMillan Publishers Ltd.

Sebateir, P. C., 1977, On generalized inverse problems and constraints. J. Geophy., **43**, 115-137.

Shelton, J. W., 1976, The Oligocene Chadra sands, Gialo Field, Libya. Internal geological report, Exploration Dept., Waha Oil Co.

Strang, G., 1988, Linear Algebra and its applications. Harcourt Brace Jovanovich Publishers.

Tarantula, A., 1987. Inverse Problem theory. Elsevier.

Tian Gang, and N. R. Goulty, 1997, Seismic inversion for coal seam thickness: trials from the Belvoir coalfield, England. In press. Geophy. pros.

Treitel, S., and Lines, L., R., 1982, Linear inverse theory and deconvolution. Geophy., **47**, 1153-1159.

van Reil, P., and Berkhout, A. J., 1985, Resolution in seismic trace inversion by parameter estimation. Geophy., **50**, 1440-1455.

Widess, M. B., 1973, How thin is a thin bed?. Geophy. **38**, 1176-1180.

Wiggins, R. A., Larner, K. L. and Wisecup, R., 1976, Residual statics analysis as a general linear inverse problem. Geophy., **41**, 922-938.

White, R. E., 1977, Estimation problems in seismic deconvolution. In: Deconvolution and Inversion (eds. M. Bernabini, P. Carrion, G. Jacovitti, F. Rocca, S. Treitel and M. H. Worthington), 5-37. Blackwell Scientific Publications, Oxford.

Appendix: Computer program listing.

This appendix gives a listing of the computer program used to invert for impedance and boundary locations of the seismic traces in this research. The program is written in Fortran 77 and runs successfully on HP or Sun4 workstations.

```
24          CONTINUE
            BC(MAC)=VAL
          ENDIF
        ELSE
          KY=KY+1
          IF(.NOT.LOGYE(KY)) THEN
            DO 26 J=1,NALL
              IF(J.EQ.I) THEN
                SAVE(J)=1.
              ELSE
                SAVE(J)=0.
              ENDIF
26          CONTINUE
            IF(LOGYL(KY)) THEN
              VAL=YL(KY)
            ELSE
              VAL=-1.E+21
            ENDIF
            XCASE=.FALSE.
            CALL VIOXY(SAVE,NP,VAL,Y,N,DY,CVIOL,ACTV,XCASE)
            IF(CVIOL) THEN
              WRITE(*,*)'Fatal error: YLB CONSTRAINT ',KY,' VIOLATED.'
              VIO=.TRUE.
              RETURN
            ENDIF
            MAC=MAC+1
            IF(ACTV) THEN
              INDA=INDA+1
              DO 27 J=1,NALL
                AE(INDA,J)=SAVE(J)
27            CONTINUE
              BE(INDA)=VAL
              INDXA(INDA)=MAC
            ENDIF
            INDB=INDB+1
            INDXB(INDB)=MAC
            DO 28 J=1,NALL
              AC(MAC,J)=SAVE(J)
28          CONTINUE
            BC(MAC)=VAL
C
            SAVE(I)=-1.
            IF(LOGYU(KY)) THEN
              VAL=-YU(KY)
            ELSE
              VAL=-1.E+21
            ENDIF
            XCASE=.FALSE.
            CALL VIOXY(SAVE,NP,VAL,Y,N,DY,CVIOL,ACTV,XCASE)
            IF(CVIOL) THEN
              WRITE(*,*)'Fatal error: YUB CONSTRAINT ',KY,' VIOLATED.'
              VIO=.TRUE.
              RETURN
            ENDIF
            MAC=MAC+1
            IF(ACTV) THEN
              INDA=INDA+1
              DO 29 J=1,NALL
                AE(INDA,J)=SAVE(J)
29            CONTINUE
              BE(INDA)=VAL
              INDXA(INDA)=MAC
            ENDIF
            INDB=INDB+1
            INDXB(INDB)=MAC
            DO 30 J=1,NALL
              AC(MAC,J)=SAVE(J)
30          CONTINUE
            BC(MAC)=VAL
          ENDIF
        ENDIF
31    CONTINUE
C
      RETURN
      END
C--------------------------------------------------------------
C
      SUBROUTINE REDCHR(inbuff,number,retry,I)
C
      CHARACTER BLK*1
      PARAMETER(BLK=' ')
      CHARACTER inbuff*(*),cobuff*32
      INTEGER   number,ic
C
      J=0
      DO 10 I=1,32
        ic=ichar(inbuff(I:I))
        IF(ic.EQ.32) THEN
          CONTINUE
        ELSEIF((ic.GE.48).AND.(ic.LE.57)) THEN
          J=J+1
          cobuff(J:J)=inbuff(I:I)
        ELSEIF((ic.EQ.43).OR.(ic.EQ.45)) THEN
          J=J+1
          cobuff(J:J)=inbuff(I:I)
        ELSE
          retry=.TRUE.
          RETURN
        ENDIF
10    CONTINUE
      IF(J.LT.32) THEN
        DO 20 I=J+1,32
          cobuff(I:I)=BLK
20      CONTINUE
      ENDIF
C
      READ(cobuff,'(I8)') number
      RETURN
      END
C--------------------------------------------------------------
C
      SUBROUTINE PRMATX(iunit,A,MPP,M,N,NIK,FORM)
C
      INTEGER MPP,M,N,JK,IK,iunit
      REAL    A(MPP,N)
      CHARACTER FORM*(*)
C
      JK=0
      DO 20 IK=1,N,NIK
        JK=JK+1
        IF(JK*NIK.LE.N) THEN
          DO 10 I=1,M
            WRITE(iunit,FORM) (A(I,J),J=IK,IK+NIK-1)
10        CONTINUE
          IF(JK*NIK.LT.N) WRITE(iunit,*) 'cont. .....'
        ELSE
          DO 15 I=1,M
            WRITE(iunit,FORM) (A(I,J),J=IK,N)
15        CONTINUE
        ENDIF
20    CONTINUE
C
      RETURN
      END
C--------------------------------------------------------------
C Home.
*     integer function common_handler(isig,icode,isigcontext,iaddr)
C
*     integer isig,icode,isigcontext,iaddr
```

```
C
*     write(*,10) loc(icode),loc(iaddr)
* 10  FORMAT('ieee exception ',Z3,'  occurred at address ',Z8)
C
*     return
*     end
C--------------------------------------------------------------
C Durham.
*     integer function common_handler(sig,sip,uap)
C
*     integer sig
*     structure /fault/
*       integer address
*     end structure
*     structure /siginfo/
*       integer si_signo
*       integer si_code
*       integer si_errno
*       record /fault/ fault
*     end structure
*     record /siginfo/ sip
C
*     write(*,10) sip.si_code,sip.fault.address
* 10  FORMAT("ieee exception ",i4,"  occurred at address ",z8)
C
*     end
C--------------------------------------------------------------
```

```
      WRITE(6,523)
523   FORMAT('  ivob> FINALLY: Plot ALL results AGAIN? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) GO TO 510
      ynans=BLK
      WRITE(6,524)
524   FORMAT('  ivob> FINALLY: WRITE XYT in/out-put to file? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) GO TO 520

      STOP
      END
```

```fortran
 10        CONTINUE
           ELSEIF(I.EQ.NI) THEN
              DO 11 J=1,N
                 IF(J.EQ.I) THEN
                    SAVE(J)=-1.
                 ELSE
                    SAVE(J)=0.
                 ENDIF
 11           CONTINUE
           ELSE
              DO 12 J=1,N
                 IF(J.EQ.(I-1)) THEN
                    SAVE(J)=-1.
                 ELSEIF(J.EQ.I) THEN
                    SAVE(J)=1.
                 ELSE
                    SAVE(J)=0.
                 ENDIF
 12           CONTINUE
           ENDIF
c
           CALL VIOT(SAVE,TE(I),T,NI,DT,SI,CVIOL,ACTV)
           IF(CVIOL.OR.(.NOT.ACTV)) THEN
              WRITE(*,*)'Fatal error: TE CONSTRAINT ',I,' VIOLATED.'
              VIO=.TRUE.
              RETURN
           ENDIF
           MAE=MAE+1
           INDA=INDA+1
           INDXA(INDA)=-1
           DO 13 J=1,N
              AE(MAE,J)=SAVE(J)
 13        CONTINUE
           BE(MAE)=TE(I)
       ENDIF
 14    CONTINUE
c
       DO 25 I=1,NI
          IF(.NOT.LOGTE(I)) THEN
             IF(I.EQ.1) THEN
                DO 16 J=1,N
                   IF(J.EQ.I) THEN
                      SAVE(J)=1.
                   ELSE
                      SAVE(J)=0.
                   ENDIF
 16             CONTINUE
             ELSEIF(I.EQ.NI) THEN
                DO 17 J=1,N
                   IF(J.EQ.N) THEN
                      SAVE(J)=-1.
                   ELSE
                      SAVE(J)=0.
                   ENDIF
 17             CONTINUE
             ELSE
                DO 18 J=1,N
                   IF(J.EQ.I) THEN
                      SAVE(J-1)=-1.
                      SAVE(J)  = 1.
                   ELSE
                      SAVE(J) =0.
                   ENDIF
 18             CONTINUE
             ENDIF
c
             IF(LOGTL(I)) THEN
                VAL=TL(I)
             ELSEIF(I.EQ.NI) THEN
                VAL=SI-TMAX
             ELSE
                VAL=SI
             ENDIF
c
             CALL VIOT(SAVE,VAL,T,NI,DT,SI,CVIOL,ACTV)
             IF(CVIOL) THEN
                WRITE(*,*)'Fatal error: TLB CONSTRAINT ',I,' VIOLATED.'
                VIO=.TRUE.
                RETURN
             ENDIF
             MAC=MAC+1
             IF(ACTV) THEN
                INDA=INDA+1
                DO 19 J=1,N
                   AE(INDA,J)=SAVE(J)
 19             CONTINUE
                BE(INDA)=VAL
                INDXA(INDA)=MAC
             ENDIF
             INDB=INDB+1
             INDXB(INDB)=MAC
             DO 20 J=1,N
                AC(MAC,J)=SAVE(J)
 20          CONTINUE
             BC(MAC)=VAL
c
             DO 21 J=1,N
                SAVE(J)=-SAVE(J)
 21          CONTINUE
             IF(LOGTU(I)) THEN
                VAL=-TU(I)
             ELSE
                VAL=-UBMAX
             ENDIF
             CALL VIOT(SAVE,VAL,T,NI,DT,SI,CVIOL,ACTV)
             IF(CVIOL) THEN
                WRITE(*,*)'Fatal error: TUB CONSTRAINT ',I,' VIOLATED.'
                VIO=.TRUE.
                RETURN
             ENDIF
             MAC=MAC+1
             IF(ACTV) THEN
                INDA=INDA+1
                DO 22 J=1,N
                   AE(INDA,J)=SAVE(J)
 22             CONTINUE
                BE(INDA)=VAL
                INDXA(INDA)=MAC
             ENDIF
             INDB=INDB+1
             INDXB(INDB)=MAC
             DO 23 J=1,N
                AC(MAC,J)=SAVE(J)
 23          CONTINUE
             BC(MAC)=VAL
          ENDIF
 25    CONTINUE
c
       RETURN
       END
c--------------------------------------------------------
       SUBROUTINE VLXY(X,LOGXE,XE,LOGXL,XL,LOGXU,XU,Y,LOGYE,YE,LOGYL,YL,
      +            LOGYU,YU,AE,BE,AC,BC,INDXA,INDXB,SAVE,MAE,MAC,
      +            INDA,INDB,MC2P,NP,N,NECXY,VIO)
c
       INTEGER MAE,MAC,INDA,INDB,INDXA(NP),INDXB(MC2P),N,KX,KY,NECXY
       REAL    AE(NP,2*N),BE(NP),AC(MC2P,2*N),BC(MC2P),SAVE(NP),X(N),
      +        XE(N),XL(N),XU(N),Y(N),YE(N),YL(N),YU(N),DX,DY
       LOGICAL LOGXE(N),LOGXL(N),LOGXU(N),LOGYE(N),LOGYL(N),
      +        LOGYU(N),VIO,CVIOL,ACTV,XCASE
c
       DX =.01
       DY =.0001
       NALL=2*N
       INDA=0
       INDB=0
       MAE =0
       MAC =0
       NECXY=0
       VIO =.FALSE.
c
       KX=0
       KY=0
       DO 12 I=1,2*N
          IF(MOD(I,2).GT.0) THEN
             KX=KX+1
             IF(LOGXE(KX)) THEN
                NECXY=NECXY+1
                DO 10 J=1,NALL
                   IF(J.EQ.I) THEN
                      SAVE(J)=1.
                   ELSE
                      SAVE(J)=0.
                   ENDIF
 10             CONTINUE
                XCASE=.TRUE.
                CALL VIOXY(SAVE,NP,XE(KX),X,N,DX,CVIOL,ACTV,XCASE)
                IF(CVIOL.OR.(.NOT.ACTV)) THEN
                   WRITE(*,*)'Fatal error: XEQ CONSTRAINT ',KX,' VIOLATED.'
                   VIO=.TRUE.
                   RETURN
                ELSE
                   INDA=INDA+1
                   INDXA(INDA)=-1
                   MAE=MAE+1
                   DO 11 J=1,NALL
                      AE(MAE,J)=SAVE(J)
 11                CONTINUE
                   BE(MAE)=XE(KX)
                ENDIF
             ENDIF
          ELSE
             KY=KY+1
             IF(LOGYE(KY)) THEN
                NECXY=NECXY+1
                DO 13 J=1,NALL
                   IF(J.EQ.I) THEN
                      SAVE(J)=1.
                   ELSE
                      SAVE(J)=0.
                   ENDIF
 13             CONTINUE
                XCASE=.FALSE.
                CALL VIOXY(SAVE,NP,YE(KY),Y,N,DY,CVIOL,ACTV,XCASE)
                IF(CVIOL.OR.(.NOT.ACTV)) THEN
                   WRITE(*,*)'Fatal error: YEQ CONSTRAINT ',KY,' VIOLATED.'
                   VIO=.TRUE.
                   RETURN
                ELSE
                   MAE=MAE+1
                   INDA=INDA+1
                   INDXA(INDA)=-1
                   DO 14 J=1,NALL
                      AE(MAE,J)=SAVE(J)
 14                CONTINUE
                   BE(MAE)=YE(KY)
                ENDIF
             ENDIF
          ENDIF
 12    CONTINUE
c
       KX=0
       KY=0
       DO 31 I=1,2*N
          IF(MOD(I,2).GT.0) THEN
             KX=KX+1
             IF(.NOT.LOGXE(KX)) THEN
                DO 20 J=1,NALL
                   IF(J.EQ.I) THEN
                      SAVE(J)=1.
                   ELSE
                      SAVE(J)=0.
                   ENDIF
 20             CONTINUE
                IF(LOGXL(KX)) THEN
                   VAL=XL(KX)
                ELSE
                   VAL=1.E-21
                ENDIF
                XCASE=.TRUE.
                CALL VIOXY(SAVE,NP,VAL,X,N,DX,CVIOL,ACTV,XCASE)
                IF(CVIOL) THEN
                   WRITE(*,*)'Fatal error: XLB CONSTRAINT ',KX,' VIOLATED.'
                   VIO=.TRUE.
                   RETURN
                ENDIF
                MAC=MAC+1
                IF(ACTV) THEN
                   INDA=INDA+1
                   DO 21 J=1,NALL
                      AE(INDA,J)=SAVE(J)
 21                CONTINUE
                   BE(INDA)=VAL
                   INDXA(INDA)=MAC
                ENDIF
                INDB=INDB+1
                INDXB(INDB)=MAC
                DO 22 J=1,NALL
                   AC(MAC,J)=SAVE(J)
 22             CONTINUE
                BC(MAC)=VAL
c
                SAVE(I)=-1.
                IF(LOGXU(KX)) THEN
                   VAL=-XU(KX)
                ELSE
                   VAL=-1.E+21
                ENDIF
                XCASE=.TRUE.
                CALL VIOXY(SAVE,NP,VAL,X,N,DX,CVIOL,ACTV,XCASE)
                IF(CVIOL) THEN
                   WRITE(*,*)'Fatal error: XUB CONSTRAINT ',KX,' VIOLATED.'
                   VIO=.TRUE.
                   RETURN
                ENDIF
                MAC=MAC+1
                IF(ACTV) THEN
                   INDA=INDA+1
                   DO 23 J=1,NALL
                      AE(INDA,J)=SAVE(J)
 23                CONTINUE
                   BE(INDA)=VAL
                   INDXA(INDA)=MAC
                ENDIF
                INDB=INDB+1
                INDXB(INDB)=MAC
                DO 24 J=1,NALL
                   AC(MAC,J)=SAVE(J)
```

```fortran
          IF(perform.EQ.'sc') WRITE(6,*) 'Exit status: 0'
          IF(perform.EQ.'fl') WRITE(15,*) 'Exit status: 0'
        ENDIF
        IF(prblm.EQ.'bn') THEN
          problm(1)=.TRUE.
          problm(2)=.FALSE.
        ENDIF
        IF(prblm.EQ.'ai') THEN
          problm(1)=.FALSE.
          problm(2)=.TRUE.
        ENDIF
        IF(perform.EQ.'sc') WRITE(6,*) ' '
        IF(perform.EQ.'fl') WRITE(15,*) ' '
        DO 452 I=1,K
          ER(I)=0.
452     CONTINUE
        DO 712 I=1,M
          FSIMP(I,JK)=ACOU(I)
          FSRSP(I,JK)=RSPI2(I)
          FSERR(I,JK)=DSC2(I)
          IF(I.LE.NI) THEN
            XFS(I,JK)=XN(I)
            YFS(I,JK)=YN(I)
            TFS(I,JK)=TN(I)
          ENDIF
712     CONTINUE
        GO TO 999
      ELSE
        IF(problm(1)) THEN
          problm(1)=.FALSE.
          problm(2)=.TRUE.
          IF(perform.EQ.'sc') WRITE(6,'('' * ai:'',$)')
          IF(perform.EQ.'fl') WRITE(15,'('' * ai:'',$)')
          DO 450 I=1,N
            T(I)=TN(I)
450       CONTINUE
          GO TO 467
        ENDIF
        IF(problm(2)) THEN
          problm(2)=.FALSE.
          problm(1)=.TRUE.
          IF(perform.EQ.'sc') WRITE(6,'('' * bn:'',$)')
          IF(perform.EQ.'fl') WRITE(15,'('' * bn:'',$)')
          DO 451 I=1,NI
            X(I)=XN(I)
            Y(I)=YN(I)
451       CONTINUE
          GO TO 467
        ENDIF
      ENDIF
      ELSE
        IF(.NOT.KEEPJG) THEN
          CALL PRDMAT(DJM2T,DJM2,H,NP,MP,NP,N,M,N)
          DO 369 I=1,M
            DSC1(I)=DSC2(I)
            IF(I.LE.N) G(I)=GP1(I)
            DO 369 J=1,N
              DJM(I,J)=DJM2(I,J)
              DJMT(J,I)=DJM2T(J,I)
369       CONTINUE
          KX=0
          KY=0
          DO 370 I=1,N
            IF(problm(1)) T(I)=TN(I)
            IF(problm(2)) THEN
              IF(MOD(I,2).GT.0) THEN
                KX=KX+1
                X(KX)=XN(KX)
              ELSE
                KY=KY+1
                Y(KY)=YN(KY)
              ENDIF
            ENDIF
370       CONTINUE
        ENDIF
        IF(DELION) DELION=.FALSE.
        IF(ADDION) ADDION=.FALSE.
        IF(KEEPJG) KEEPJG=.FALSE.
        GO TO 35
      ENDIF
99    CONTINUE
C
C Write an exit status to the screen.
      PRINT*
      IF(VIO.OR.FVIOL.OR.LWSHRT.OR.SINGUL.OR.
     +  (problm(1).AND.(MA.GT.MAC1)).OR.
     +  (problm(2).AND.(MA.GT.MAC2)).OR.
     +  (problm(1).AND.(INDB.LT.0)).OR.
     +  (problm(2).AND.(INDB.LT.0))) THEN
        IF(VIO) PRINT*,' Exit status: VIO=1'
        IF(FVIOL) PRINT*,' Exit status: FVIOL=1'
        IF(LWSHRT) PRINT*,' Exit status: LWSHRT=1'
        IF(SINGUL) PRINT*,' Exit status: SINGUL=1'
        IF(problm(1).AND.(MA.GT.MAC1)) PRINT*,' Exit status: MA > MAC1'
        IF(problm(2).AND.(MA.GT.MAC2)) PRINT*,' Exit status: MA > MAC2'
        IF((problm(1).OR.problm(2)).AND.(INDB.LT.0))
     +                PRINT*,' Exit status: INDB < 0'
      ELSE
        PRINT*,' Exit status: 0'
      ENDIF
      PRINT*
      PRINT*,' Number of traces : ',JK
      PRINT*,' Number of samples: ',M
      IF(perform.EQ.'fl') WRITE(15,*) ' Number of traces : ',JK
      IF(perform.EQ.'fl') WRITE(15,*) ' Number of samples: ',M
395   CONTINUE
C
      CLOSE(15)
      ynans=BLK
      WRITE(6,530)
530   FORMAT(' ivob> SCALE IMPEDENCE solution to max.? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'N').OR.(ynans.EQ.'n')) THEN
        IMPNORM=.FALSE.
        GO TO 510
      ELSEIF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) THEN
        IMPNORM=.TRUE.
        VIMAX=0.
        VALMAX=0.
        DO 531 J=1,JK
          CALL MAXVAL(FSIMP(1,J),VALMAX,IPOS,MP,M)
          FSIMAX(J)=VALMAX
          IF(VALMAX.GT.VIMAX) THEN
            JPOS=J
            VIMAX=VALMAX
          ENDIF
531     CONTINUE
        DO 533 J=1,JK
          IF(J.EQ.JPOS) GO TO 533
          SCALIMP=VALMAX/FSIMAX(J)
          DO 532 I=1,M
            FSIMPSC(I,J)=FSIMP(I,J)*SCALIMP
532       CONTINUE
533     CONTINUE
      ELSE
        PRINT*,' Warninig: Answer y/n.'
        GO TO 395
      ENDIF
```

```fortran
C Apply highcut filter on impedence data.
*
*     OPEN(18,FILE='ivo_filtout')
*     WRITE(18,500)
* 500 FORMAT(//,'  GIIMP and FSIMP: before Filtering:',/)
*     DO 501 I=1,NSWIN
*       WRITE(18,502) I,(GIIMP(I,J),J=1,JK),(FSIMP(I,J),J=1,JK)
* 501 CONTINUE
* 502 FORMAT(3X,I3,2X,3(1X,F8.2),5X,3(1X,F8.2))
*
*     FRQHC=100.
*     DO 396 J=1,JK
*       CALL LPFIL(GIIMP(1,J),M,MP,FRQHC,SAVEQ,SINTB,SI)
*       CALL LPFIL(FSIMP(1,J),M,MP,FRQHC,SAVEQ,SINTB,SI)
*       IF(IMPNORM) CALL LPFIL(FSIMPSC(1,J),M,MP,FRQHC,SAVEQ,SINTB,SI)
* 396 CONTINUE
*
*     WRITE(18,503)
* 503 FORMAT(//,'  GIIMP and FSIMP: after Filtering:',/)
*     DO 504 I=1,NSWIN
*       WRITE(18,502) I,(GIIMP(I,J),J=1,JK),(FSIMP(I,J),J=1,JK)
* 504 CONTINUE
C
C Go into graphics.
510   CONTINUE
      ynans=BLK
      PTYPE='Observed Seismic data:'
      AIORSEI='SEIDAT'
      WRITE(6,511)
511   FORMAT('  ivob> Plot observed SEISMIC Data? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,SEIDAT,
     +                MP,NTP,NSWIN,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
      ynans=BLK
      PTYPE='Final impedence solution:'
      AIORSEI='AIMDAT'
      WRITE(6,512)
512   FORMAT('  ivob> Plot FINAL IMPEDENCE Solution? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,FSIMP,
     +                MP,NTP,M,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
      IF(IMPNORM) THEN
        ynans=BLK
        PTYPE='Final normalised impedence solution:'
        AIORSEI='AIMDAT'
        WRITE(6,534)
534     FORMAT('  ivob> Plot FINAL NORMAL. IMPEDENCE Solution? (y/n): ',
     +         $)
        READ(5,'(A)') ynans
        IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,
     +          FSIMPSC,MP,NTP,M,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
      ENDIF
      ynans=BLK
      PTYPE='Final seismic solution:'
      AIORSEI='SEIDAT'
      WRITE(6,513)
513   FORMAT('  ivob> Plot FINAL SEISMIC Solution? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,FSRSP,
     +                MP,NTP,M,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
      ynans=BLK
      PTYPE='Final error traces:'
      AIORSEI='SEIDAT'
      WRITE(6,514)
514   FORMAT('  ivob> Plot Final ERROR traces? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,FSERR,
     +                MP,NTP,M,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
      ynans=BLK
      PTYPE='Initial guess impedence:'
      AIORSEI='AIMDAT'
      WRITE(6,515)
515   FORMAT('  ivob> Plot INITIAL GUESS IMPEDENCE? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,GIIMP,
     +                MP,NTP,M,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
      ynans=BLK
      PTYPE='Initial guess seismic response:'
      AIORSEI='SEIDAT'
      WRITE(6,516)
516   FORMAT('  ivob> Plot INITIAL GUESS SEISMIC response? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,GIRSP,
     +                MP,NTP,M,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
      ynans=BLK
      PTYPE='Initial Guess error traces:'
      AIORSEI='SEIDAT'
      WRITE(6,517)
517   FORMAT('  ivob> Plot INITIAL GUESS ERROR traces? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) CALL PRESUL(AIORSEI,GIERR,
     +                MP,NTP,M,JK,SI,PTYPE,FNGDAT,FNGCON,nsss)
518   CONTINUE
      ynans=BLK
      WRITE(6,519)
519   FORMAT('  ivob> Plot ALL results AGAIN? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) THEN
        GO TO 510
      ELSEIF((ynans.EQ.'N').OR.(ynans.EQ.'n')) THEN
        GO TO 520
      ELSE
        PRINT*,'Warning: Answer y or n.'
        GO TO 518
      ENDIF
520   CONTINUE
C
      FORMX='(20(1X,F7.0))'
      FORMY='(20(1X,F7.3))'
      FORMT='(20(1X,F5.0))'
      ynans=BLK
      WRITE(6,521)
521   FORMAT('  ivob> Want to write INPUT XYT to file? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) THEN
*       OPEN(16,FILE='ivob.'//FNGDAT(:INDEX(FNGDAT,BLK)-1)//'.input')
        OPEN(16,FILE='ivob.input')
        WRITE(16,*)
        WRITE(16,*) '** Input boundaries T:'
        CALL PRMATX(16,TIG,50,NIMAX,JK,13,FORMT)
        WRITE(16,*)
        WRITE(16,*) '** Input impedence X:'
        CALL PRMATX(16,XIG,50,NIMAX,JK,10,FORMX)
      ENDIF
      ynans=BLK
      WRITE(6,522)
522   FORMAT('  ivob> Want to write OUTPUT XYT to file? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) THEN
*       OPEN(17,FILE='ivob.'//FNGDAT(:INDEX(FNGDAT,BLK)-1)//'.output')
        OPEN(17,FILE='ivob.output')
        WRITE(17,*)
        WRITE(17,*) '** Output boundaries T:'
        CALL PRMATX(17,TFS,50,NIMAX,JK,13,FORMT)
        WRITE(17,*)
        WRITE(17,*) '** Output impedence X:'
        CALL PRMATX(17,XFS,50,NIMAX,JK,10,FORMX)
      ENDIF
      ynans=BLK
```

```
                KX=KX+1
                S1=S1+AC(INDX2(I),J)*X(KX)
              ELSE
                KY=KY+1
                S1=S1+AC(INDX2(I),J)*Y(KY)
              ENDIF
            ENDIF
   10     CONTINUE
C
C Find step length D(I) to the I-th constraint.
          D(I)=S1-BC(INDX2(I))
          IF(D(I).EQ.0.) THEN
            IF(S2.EQ.0.) THEN
              D(I)=0.
            ELSEIF(S2.GT.0.) THEN
              D(I)=1.E20
            ELSE
              D(I)=0.
            ENDIF
          ELSEIF(D(I).GT.0.) THEN
            IF(S2.EQ.0.) THEN
              D(I)=1.E20
            ELSEIF(S2.GT.0.) THEN
              D(I)=1.E20
            ELSE
              D(I)=-D(I)/S2
            ENDIF
          ELSE
            D(I)=-1.E20
            print*,'a constraint is violated.'
          ENDIF
   20   CONTINUE
C
C Test if any (or some) of the constraints could become active, if so
C then find the minimum step length DMX of the positive steps and its
C position IA in the constraints matrix AC so that if we need to move
C the full step length we know which passive constraint we add to the
C set of active constraints.
        DO 30 I=1,IND2
          IF(D(I).LT.0.) GO TO 30
          IF(.NOT.CD) THEN
            TEMP=D(I)
            IA  =I
            CD  =.TRUE.
          ELSE
            IF(D(I).LT.TEMP) THEN
              TEMP=D(I)
              IA  =INDX2(I)
            ENDIF
          ENDIF
   30   CONTINUE
        IF(CD) DMX=TEMP
C
        RETURN
        END
C--------------------------------------------------------------------
C
        SUBROUTINE LSQ(X,XN,Y,YN,T,TN,NP,NI,N,FNO,FNT,FNN,G,P,DS,RSPN,
      +              RSPO,MP,M,WV,LWP,LWV,SI,RSPT,MWP,ACOU,REF,ALAMDA,
      +              STP,IMPROV,problm,NSWIN)
C
C Performs quadratic line search along P, using backtrack method.
C Parameter ALF ensures sufficient decrease in function value.
C
        INTEGER MP,NP,LWP,LWV,MWP,NI,N,M,KX,KY,NSWIN
        LOGICAL IMPROV,problm(2)
      *             ,FNOGT
        REAL X(NP),XN(NP),Y(NP),YN(NP),T(NP),TN(NP),G(NP),P(NP),RSPO(MP),
      +     WV(LWV),RSPT(MWP),RSPN(MP),DS(MP),ACOU(MP),REF(MP),FNN,FNT,
      +     FNO,STP,TOLX,ALAM1,TMPLAM,ALAMDA,ALAMN,SLOPE,TEMP,TEST,
      +     STPLAM
        PARAMETER (ALFA=1.E-5,TOLX=1.E-7)
C
        TEST = 0.
        SLOPE= 0.
        ALAM1= 1.
C
C Find slope.
        IF(problm(1)) STP=1.
        DO 10 I=1,N
          SLOPE=SLOPE+STP*G(I)*P(I)
   10   CONTINUE
C
C Compute minimum step ALAMN.
        KX=0
        KY=0
        DO 12 I=1,N
          IF(problm(1)) TEMP=ABS(P(I))/AMAX1(ABS(T(I)),1.)
      *               TEMP=ABS(STP*P(I))/AMAX1(ABS(T(I)),1.)
          IF(problm(2)) THEN
            IF(MOD(I,2).GT.0) THEN
              KX=KX+1
              TEMP=ABS(STP*P(I))/AMAX1(ABS(X(KX)),1.)
            ELSE
              KY=KY+1
              TEMP=ABS(STP*P(I))/AMAX1(ABS(Y(KY)),1.)
            ENDIF
          ENDIF
          IF(TEMP.GT.TEST) TEST=TEMP
   12   CONTINUE
        ALAMN=TOLX/TEST
C
C Use quadratic model.
   13   CONTINUE
        TMPLAM=-SLOPE/(2.*(FNT-FNO-SLOPE))
        IF(TMPLAM.LT.0.) THEN
          PRINT*,'Warning: TMPLAM_LSQ negative. Set to: TMPLAM=.5'
          TMPLAM=.5
        ENDIF
        IF(TMPLAM.EQ.0.) THEN
          PRINT*,'Warning: TMPLAM in LSQ is zero.'
          PRINT*,'          set to: TMPLAM=.5
          TMPLAM=.5
        ENDIF
C
C Set TMPLAM within [0.1*ALAM1,0.5*ALAM1].
        IF(TMPLAM.LT.0.1*ALAM1) TMPLAM=0.1*ALAM1
        IF(TMPLAM.GT.0.5*ALAM1) TMPLAM=0.5*ALAM1
C
C Compute a new point and function value FNEW.
        STPLAM=STP*TMPLAM
        KX=0
        KY=0
        DO 15 I=1,N
          IF(problm(1)) TN(I)=T(I)+SI*REAL(ANINT(STPLAM*P(I)/SI))
          IF(problm(2)) THEN
            IF(MOD(I,2).GT.0) THEN
              KX=KX+1
              XN(KX)=X(KX)+STPLAM*P(I)
            ELSE
              KY=KY+1
              YN(KY)=Y(KY)+STPLAM*P(I)
            ENDIF
          ENDIF
   15   CONTINUE
        IF(problm(1)) CALL RESPON(X,Y,TN,NP,NI,WV,LWP,LWV,SI,RSPN,MP,M,
      +              RSPT,MWP,ACOU,REF,NSWIN)
        IF(problm(2)) CALL RESPON(XN,YN,T,NP,NI,WV,LWP,LWV,SI,RSPN,MP,M,
      +              RSPT,MWP,ACOU,REF,NSWIN)
```

```
        CALL SUBVEC(RSPN,RSPO,DS,MP,M)
        FNN=VNORM(DS,MP,M)/2.
C
C Test for convergence.
        IF(FNN.LT.FNO) THEN
          ALAMDA=STPLAM
          IMPROV=.TRUE.
          RETURN
        ELSEIF((TMPLAM.LT.ALAMN).OR.(ABS(FNN-FNO).LE.ALFA)) THEN
          KX=0
          KY=0
          DO 20 I=1,N
            IF(problm(1)) TN(I)=T(I)
            IF(problm(2)) THEN
              IF(MOD(I,2).GT.0) THEN
                KX=KX+1
                XN(KX)=X(KX)
              ELSE
                KY=KY+1
                YN(KY)=Y(KY)
              ENDIF
            ENDIF
   20     CONTINUE
          FNN  =FNO
          ALAMDA=STP
          IMPROV=.FALSE.
          RETURN
        ELSE
C
C Interchange FNN and FNT, ALAM1 and TMPLAM, and repeat.
          ALAM1=TMPLAM
          FNT  =FNN
          GOTO 13
        ENDIF
C
        END
C
C-------------------------------------------------------------------
        SUBROUTINE VIOT(C,B,T,N,D,SI,CVIOL,ACTV)
C
        INTEGER N
        REAL C(N),T(N),B,D,S,SI
        LOGICAL CVIOL,ACTV
C
        CVIOL=.FALSE.
        ACTV =.FALSE.
        S    =0.
        DO 10 I=1,N-1
          S=S+C(I)*T(I)
   10   CONTINUE
        S=S-B
        IF(S.LT.0.) CVIOL=.TRUE.
        IF((S.LE.D).AND.(S.GE.0.)) ACTV=.TRUE.
C
        RETURN
        END
C-------------------------------------------------------------------
        SUBROUTINE VIOXY(C,NP,B,X,N,D,CVIOL,ACTV,XCASE)
C
        INTEGER N
        REAL    C(NP),X(N),B,D,S
        LOGICAL CVIOL,ACTV,XCASE
C
        CVIOL=.FALSE.
        ACTV =.FALSE.
        S=0.
        K=0
        J=2
        IF(XCASE) J=1
        DO 10 I=J,2*N,2
          K=K+1
          S=S+C(I)*X(K)
   10   CONTINUE
        S=S-B
        IF(ABS(S).LE.D) THEN
          ACTV=.TRUE.
        ELSEIF((S.LT.0.).AND.(ABS(S).GE.D)) THEN
          CVIOL=.TRUE.
        ELSE
          CONTINUE
        ENDIF
**
*       IF(S.LT.0.) CVIOL=.TRUE.
*       IF((S.LE.D).AND.(S.GE.0.)) ACTV=.TRUE.
**
*       IF(ABS(S).LE.D) THEN
*         ACTV=.TRUE.
*       ELSEIF(S.LT.0.) THEN
*         CVIOL=.TRUE.
*       ELSE
*         CONTINUE
*       ENDIF
C
        RETURN
        END
C-------------------------------------------------------------------
        SUBROUTINE VLT(T,LOGTE,TE,LOGTL,TL,LOGTU,TU,AE,BE,AC,BC,INDXA,
      +              INDXB,SAVE,MAE,MAC,INDA,INDB,MC1P,NP,N,NI,SI,
      +              NECT,VIO)
C
        INTEGER MAE,MAC,INDXA(NP),INDXB(MC1P),INDA,INDB,NECT,IEQ,ILB
        REAL    T(NI),TE(NI),TL(NI),TU(NI),AE(NP,NI),BE(NP),
      +        AC(MC1P,NI),BC(MC1P),SAVE(NI),TMAX,TOTEQ,TOTLB
        LOGICAL LOGTE(NI),LOGTL(NI),LOGTU(NI),CVIOL,ACTV,VIO
C
        DT  =.02
        HSI =SI/2.
        MAE =0
        MAC =0
        INDA=0
        INDB=0
        NECT=0
        VIO=.FALSE.
        TMAX=T(NI)
        TOTEQ=0.
        TOTLB=0.
        IEQ =0
        ILB =0
        DO 30 I=1,NI
          IF(LOGTE(I)) THEN
            TOTEQ=TOTEQ+TE(I)
            IEQ=IEQ+1
          ENDIF
          IF(LOGTL(I)) THEN
            TOTLB=TOTLB+TL(I)
            ILB=ILB+1
          ENDIF
   30   CONTINUE
        UBMAX=TMAX-TOTEQ-TOTLB-((NI-IEQ-ILB-1)*SI)
        DO 14 I=1,NI
          IF(LOGTE(I)) THEN
            NECT=NECT+1
            IF(I.EQ.1) THEN
              DO 10 J=1,N
                IF(J.EQ.I) THEN
                  SAVE(J)=1.
                ELSE
                  SAVE(J)=0.
                ENDIF
```

```fortran
      IF(problm(2)) CALL ADDROW(A,B,AC2,BC2,INDXA,INDXB(IADD),
     +                          MC2P,NP,MA,NA,MAC2)
C
C If add nonexistent constraint to A, or delete from empty INDXB exit.
      IF(problm(1).AND.(MA.GT.MAC1)) GO TO 99
      IF(problm(2).AND.(MA.GT.MAC2)) GO TO 99
      IF(INDB.LT.0) GO TO 99
      INDA=INDA+1
      CALL TRANSP(A,AT,NP,NP,MA,NA)
      CALL QR(AT,QT,R,SAVEQ,NP,NP,NA,MA)
      CALL TRANSP(QT,Q,NP,NP,NA,NA)
      CALL PARTQ(Q,Q1,Z,NP,NA,INDA,NZ)
      CALL TRANSP(Q1,Q1T,NP,NA,NA,MA)
      CALL TRANSP(Z,ZT,NP,NP,NA,NZ)
      ADDION=.TRUE.
      ELSE
      IMPROV=.FALSE.
      IF(problm(1)) CALL LSQ(X,XN,Y,YN,T,TN,NP,NI,N,ER(K),ERT,
     +   ER(K+1),G,P,DSCN,RSPP,RSPO,MP,M,WV,LWP,LWV,SI,RSPT,MWP,
     +   ACOU,REF,ALFA,ALF,IMPROV,problm,NSWIN)
      IF(problm(2)) CALL LSQ(X,XN,Y,YN,T,TN,NP,NI,N,ER(K),ERT,
     +   ER(K+1),G,P,DSCN,RSPP,RSPO,MP,M,WV,LWP,LWV,SI,RSPT,MWP,
     +   ACOU,REF,ALFA,ALF,IMPROV,problm,NSWIN)
      K=K+1
      IF(IMPROV) THEN
      ALF=ALFA
      DO 364 I=1,M
         RSPI2(I)=RSPP(I)
         DSC2(I) =DSCN(I)
364   CONTINUE
C
      IF(problm(1)) CALL JCB(X,Y,TN,NI,WV,LWP,LWV,SI,RSPI2,
     +   RSPP,RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,
     +   M,N,CENTRL,problm,NSWIN)
      IF(problm(2)) CALL JCB(XN,YN,T,NI,WV,LWP,LWV,SI,RSPI2,
     +   RSPP,RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,
     +   M,N,CENTRL,problm,NSWIN)
      CALL TRANSP(DJM2,DJM2T,MP,NP,M,N)
      CALL PMTVEC(DJM2T,DSC2,GP1,NP,MP,N,M)
      ELSE
C
C No LS improvement add IADD constraint to A andupdate INDXA and INDXB.
C Also put TN,XN, and YN back to what they were before the call to LSQ.
C Because LSQ will make TN=T, XN=X, and YN=Y.
      IF(INDA.EQ.0) MA=INDA
      IF(problm(1)) CALL ADDROW(A,B,AC1,BC1,INDXA,INDXB(IADD),
     +                          MC1P,NP,MA,NA,MAC1)
      IF(problm(2)) CALL ADDROW(A,B,AC2,BC2,INDXA,INDXB(IADD),
     +                          MC2P,NP,MA,NA,MAC2)
C
C If add nonexistent constraint to A, or delete from empty INDXB exit.
      IF(problm(1).AND.(MA.GT.MAC1)) GO TO 99
      IF(problm(2).AND.(MA.GT.MAC2)) GO TO 99
      IF(INDB.LT.0) GO TO 99
      INDA=INDA+1
      CALL TRANSP(A,AT,NP,NP,MA,NA)
      CALL QR(AT,QT,R,SAVEQ,NP,NP,NA,MA)
      CALL TRANSP(QT,Q,NP,NP,NA,NA)
      CALL PARTQ(Q,Q1,Z,NP,NA,INDA,NZ)
      CALL TRANSP(Q1,Q1T,NP,NA,NA,MA)
      CALL TRANSP(Z,ZT,NP,NP,NA,NZ)
      ADDION=.TRUE.
      IF(problm(1)) THEN
         DO 665 I=1,N
            TN(I)=T(I)+P(I)
665      CONTINUE
      ENDIF
      IF(problm(2)) THEN
      XX=0
      KY=0
      DO 656 I=1,N
         IF(MOD(I,2).GT.0) THEN
            KX=KX+1
            XN(KX)=X(KX)+ALF*P(I)
         ELSE
            KY=KY+1
            YN(KY)=Y(KY)+ALF*P(I)
         ENDIF
656   CONTINUE
      ENDIF
      ENDIF
      ENDIF
C
      ELSE
      IMPROV=.FALSE.
      IF(problm(1)) CALL LSQ(X,XN,Y,YN,T,TN,NP,NI,N,ER(K),ERT,
     +   ER(K+1),G,P,DSCN,RSPP,RSPO,MP,M,WV,LWP,LWV,SI,RSPT,MWP,
     +   ACOU,REF,ALFA,ALF,IMPROV,problm,NSWIN)
      IF(problm(2)) CALL LSQ(X,XN,Y,YN,T,TN,NP,NI,N,ER(K),ERT,
     +   ER(K+1),G,P,DSCN,RSPP,RSPO,MP,M,WV,LWP,LWV,SI,RSPT,MWP,
     +   ACOU,REF,ALFA,ALF,IMPROV,problm,NSWIN)
      K=K+1
      IF(IMPROV) THEN
      ALF=ALFA
      DO 365 I=1,M
         RSPI2(I)=RSPP(I)
         DSC2(I) =DSCN(I)
365   CONTINUE
      IF(problm(1)) CALL JCB(X,Y,TN,NI,WV,LWP,LWV,SI,RSPI2,
     +   RSPP,RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,
     +   M,N,CENTRL,problm,NSWIN)
      IF(problm(2)) CALL JCB(XN,YN,T,NI,WV,LWP,LWV,SI,RSPI2,
     +   RSPP,RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,
     +   M,N,CENTRL,problm,NSWIN)
      CALL TRANSP(DJM2,DJM2T,MP,NP,M,N)
      CALL PMTVEC(DJM2T,DSC2,GP1,NP,MP,N,M)
      ELSE
      KEEPJG=.TRUE.
      ENDIF
      ENDIF
C
      ELSE
C DMAX>1, move unit step. Find ERT. If ERT<ER(K) update, else do LS.
      ALF=1.
      IF(problm(1)) THEN
      IF(DMAX.LT.1.) THEN
         IF(DMAX.LT.ALFMAX) THEN
            IF(PCHNGED) ALF=DMAX
         ELSE
            ALF=ALFMAX
         ENDIF
         DO 781 I=1,N
            P(I)=ALF*P(I)
781      CONTINUE
         CALL BNP(P,N,SI,-1)
      ENDIF
      DO 165 I=1,N
         TN(I)=T(I)+P(I)
165   CONTINUE
      ENDIF
      IF(problm(2)) THEN
      IF(ALFMAX.LT.ALF) ALF=ALFMAX
      KX=0
      KY=0
      DO 166 I=1,N
         IF(MOD(I,2).GT.0) THEN
            KX=KX+1
            XN(KX)=X(KX)+ALF*P(I)
         ELSE
            KY=KY+1
            YN(KY)=Y(KY)+ALF*P(I)
         ENDIF
166   CONTINUE
      ENDIF
      IF(problm(1)) CALL RESPON(X,Y,TN,NP,NI,WV,LWP,LWV,SI,RSPI2,
     +   MP,M,RSPT,MWP,ACOU,REF,NSWIN)
      IF(problm(2)) CALL RESPON(XN,YN,T,NP,NI,WV,LWP,LWV,SI,RSPI2,
     +   MP,M,RSPT,MWP,ACOU,REF,NSWIN)
      CALL SUBVEC(RSPI2,RSPO,DSC2,MP,M)
      ERT=VNORM(DSC2,MP,M)/2.
      PMAG=SQRT(VNORM(P,NP,N))
      IF((ABS(ERT-ER(K)).LE.ERRLMT).AND.
     +   ((PMAG).LE.ERRLMT)) THEN
      K=K+1
      ER(K)=ERT
      KEEPJG=.TRUE.
      IMPROV=.FALSE.
      ELSEIF(ERT.LT.ER(K)) THEN
      K=K+1
      ER(K)=ERT
      IF(problm(1)) CALL JCB(X,Y,TN,NI,WV,LWP,LWV,SI,RSPI2,RSPP,
     +   RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,M,N,CENTRL,problm,NSWIN)
      IF(problm(2)) CALL JCB(XN,YN,T,NI,WV,LWP,LWV,SI,RSPI2,RSPP,
     +   RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,M,N,CENTRL,problm,NSWIN)
      CALL TRANSP(DJM2,DJM2T,MP,NP,M,N)
      CALL SUBVEC(RSPI2,RSPO,DSC2,MP,M)
      CALL PMTVEC(DJM2T,DSC2,GP1,NP,MP,N,M)
      ELSE
      IMPROV=.FALSE.
      CALL LSQ(X,XN,Y,YN,T,TN,NP,NI,N,ER(K),ERT,ER(K+1),G,P,DSCN,
     +   RSPP,RSPO,MP,M,WV,LWP,LWV,SI,RSPT,MWP,ACOU,REF,ALFA,ALF,
     +   IMPROV,problm,NSWIN)
      K=K+1
      IF(IMPROV) THEN
      ALF=ALFA
      DO 367 I=1,M
         RSPI2(I)=RSPP(I)
         DSC2(I) =DSCN(I)
367   CONTINUE
      IF(problm(1)) CALL JCB(X,Y,TN,NI,WV,LWP,LWV,SI,RSPI2,RSPP,
     +   RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,M,N,CENTRL,problm,NSWIN)
      IF(problm(2))CALL JCB(XN,YN,T,NI,WV,LWP,LWV,SI,RSPI2,RSPP,
     +   RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,M,N,CENTRL,problm,NSWIN)
      CALL TRANSP(DJM2,DJM2T,MP,NP,M,N)
      CALL PMTVEC(DJM2T,DSC2,GP1,NP,MP,N,M)
      ELSE
      KEEPJG=.TRUE.
      ENDIF
      ENDIF
      ENDIF
      ENDIF
C
C Test for convergence or termination.
199   CONTINUE
      EPCENT=100.*ER(K)/ERSPO
      IF(perform.EQ.'sc') THEN
      IF(EPCENT.GE.10000.) THEN
         WRITE(6,'(3X,''EE='',E12.7,3X,''EE%='',F7.1)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.1000.) THEN
         WRITE(6,'(3X,''EE='',E12.7,3X,''EE%='',F7.2)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.100.) THEN
         WRITE(6,'(3X,''EE='',E12.7,3X,''EE%='',F7.3)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.10.) THEN
         WRITE(6,'(3X,''EE='',E12.7,3X,''EE%='',F7.4)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.1.) THEN
         WRITE(6,'(3X,''EE='',E12.7,3X,''EE%='',F7.5)') ER(K),EPCENT
      ELSE
         WRITE(6,'(3X,''EE='',E12.7,3X,''EE%='',F7.6)') ER(K),EPCENT
      ENDIF
      ENDIF
      IF(perform.EQ.'fl') THEN
      IF(EPCENT.GE.10000.) THEN
         WRITE(15,'(3X,''EE='',E12.7,3X,''EE%='',F7.1)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.1000.) THEN
         WRITE(15,'(3X,''EE='',E12.7,3X,''EE%='',F7.2)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.100.) THEN
         WRITE(15,'(3X,''EE='',E12.7,3X,''EE%='',F7.3)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.10.) THEN
         WRITE(15,'(3X,''EE='',E12.7,3X,''EE%='',F7.4)') ER(K),EPCENT
      ELSEIF(EPCENT.GE.1.) THEN
         WRITE(15,'(3X,''EE='',E12.7,3X,''EE%='',F7.5)') ER(K),EPCENT
      ELSE
         WRITE(15,'(3X,''EE='',E12.7,3X,''EE%='',F7.6)') ER(K),EPCENT
      ENDIF
      ENDIF
      IF(((.NOT.(DELION.OR.ADDION)).AND.(K.GE.1)).AND.
     +   (TERMNT.OR.(ABS(ER(K)-ER(K-1)).LE.ERRLMT).OR.(ER(K).LT.ERRLMT)
     +   .OR.(K.GE.KLIMIT))) THEN
C
C TEST if we need to TAKE ANOTHER RECORD, or reiterate on the same one.
      IF((K.LE.2).AND.problm(1)) bnconv=.TRUE.
      IF((K.LE.2).AND.problm(2)) aiconv=.TRUE.
      IF(problm(1).AND.aiconv) THEN
      IF(k.GT.2) THEN
         aiconv=.FALSE.
      ELSE
         bnconv=.TRUE.
      ENDIF
      ENDIF
      IF(problm(2).AND.bnconv) THEN
      IF(K.GT.2) THEN
         bnconv=.FALSE.
      ELSE
         aiconv=.TRUE.
      ENDIF
      ENDIF
      IF(bnconv.AND.aiconv) THEN
C                        Write an exit status on screen.
      IF(VIO.OR.FVIOL.OR.LWSHRT.OR.SINGUL.OR.
     +   (problm(1).AND.(MA.GT.MAC1)).OR.
     +   (problm(2).AND.(MA.GT.MAC2)).OR.
     +   (problm(1).AND.(INDB.LT.0)).OR.
     +   (problm(2).AND.(INDB.LT.0))) THEN
      IF(perform.EQ.'sc') THEN
      IF(VIO) PRINT*,'          Exit status: VIO=1'
      IF(FVIOL) PRINT*,'          Exit status: FVIOL=1'
      IF(LWSHRT) PRINT*,'         Exit status: LWSHRT=1'
      IF(SINGUL) PRINT*,'         Exit status: SINGUL=1'
      IF(problm(1).AND.(MA.GT.MAC1))
     +      PRINT*,' Exit status: MA > MAC1'
      IF(problm(2).AND.(MA.GT.MAC2))
     +      PRINT*,'          Exit status: MA > MAC2'
      IF((problm(1).OR.problm(2)).AND.(INDB.LT.0))
     +      PRINT*,'          Exit status: INDB < 0'
      ENDIF
      IF(perform.EQ.'fl') THEN
      IF(VIO) WRITE(15,*)'          Exit status: VIO=1'
      IF(FVIOL) WRITE(15,*)'          Exit status: FVIOL=1'
      IF(LWSHRT) WRITE(15,*)'         Exit status: LWSHRT=1'
      IF(SINGUL) WRITE(15,*)'         Exit status: SINGUL=1'
      IF(problm(1).AND.(MA.GT.MAC1))
     +      WRITE(15,*)' Exit status: MA > MAC1'
      IF(problm(2).AND.(MA.GT.MAC2))
     +      WRITE(15,*)'          Exit status: MA > MAC2'
      IF((problm(1).OR.problm(2)).AND.(INDB.LT.0))
     +      WRITE(15,*)'          Exit status: INDB < 0'
      ENDIF
      ELSE
```

```fortran
C an upper or lower triangular matrix with unit diagonal elements.
C When A is upper triangular the value of FBFLAG is set to +1.0 to solve
C for X by forward-substitution. However, when A is lower triangular the
C value of FBFLAG is set to -1.0 to solve for X by back-substitution.
C
      REAL      A(NP,NP),X(NP),B(NP),FBFLAG
      INTEGER N
C
      DO 50 I=1,N
         X(I)=0.
50    CONTINUE
      IF(FBFLAG.LT.0.) THEN
         DO 20 I=N,1,-1
            SUM=0.
            DO 10 J=N,I,-1
               SUM=SUM+A(I,J)*X(J)
10          CONTINUE
            X(I)=B(I)-SUM
20       CONTINUE
      ELSE
         DO 40 I=1,N
            SUM=0.
            DO 30 J=1,I
               SUM=SUM+A(I,J)*X(J)
30          CONTINUE
            X(I)=B(I)-SUM
40       CONTINUE
      ENDIF
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE FBSUBB(A,X,B,FBFLAG,SINGUL,NP,N)
C
C This subroutine performs both back- and forward-substitution depending
C on the value of FBFLAG:
C                        FBFLAG=-1.0 then performs back-substitution.
C                        FBFLAG=+1.0 then performs forward-substitution.
C
C Basically this subroutine solves the linear system AX=B, where A is
C an upper or lower triangular matrix. When A is upper triangular the
C value of FBFLAG is set to +1.0 so we can solve for X by forward-
C substitution. However, when A is lower triangular the value of FBFLAG
C is set to -1.0 to solve for X by back substitution.
C
      REAL      A(NP,NP),X(NP),B(NP),FBFLAG,TINY
      LOGICAL SINGUL
      INTEGER N
C
C Set the singularity test value TINY.
      TINY=10.E-20
      DO 50 I=1,N
         X(I)=0.
50    CONTINUE
      IF(FBFLAG.LT.0.) THEN
         DO 20 I=N,1,-1
            SUM=0.
            DO 10 J=N,I,-1
               SUM=SUM+A(I,J)*X(J)
10          CONTINUE
            IF(ABS(A(I,I)).LE.TINY) THEN
               WRITE(*,*)'*The matrix A is singular at subroutine FBSUBB'
               SINGUL=.TRUE.
               RETURN
            ENDIF
            X(I)=(B(I)-SUM)/A(I,I)
20       CONTINUE
      ELSE
         DO 40 I=1,N
            SUM=0.
            DO 30 J=1,I
               SUM=SUM+A(I,J)*X(J)
30          CONTINUE
            IF(ABS(A(I,I)).LE.TINY) THEN
               WRITE(*,*)'***The matrix A is singular at the subroutine
     *FBSUBB'
               SINGUL=.TRUE.
               RETURN
            ENDIF
            X(I)=(B(I)-SUM)/A(I,I)
40       CONTINUE
      ENDIF
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE MINVAL(X,VAL,IPOS,NP,N)
C
C This subroutine finds the first minimum value VAL of the array X,
C and returns its position in the array IPOS. If there is more than
C one equal minimum value, IPOS will contain the position of the
C first minimum.
C
      REAL      X(NP),VAL
      INTEGER IPOS,N
C
      VAL =X(1)
      IPOS=1
      DO 10 I=2,N
         IF(X(I).LT.VAL) THEN
            VAL =X(I)
            IPOS=I
         ENDIF
10    CONTINUE
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE ABSMNV(X,VAL,IPOS,NP,N)
C
C Finds the first ABSOLUTE minimum value VAL of the array X,
C and returns its position in the array IPOS. If there is more than
C one equal minimum value, IPOS will contain the position of the
C first minimum.
C
      REAL      X(NP),VAL
      INTEGER IPOS,N
C
      VAL =ABS(X(1))
      IPOS=1
      DO 10 I=2,N
         IF(ABS(X(I)).LT.VAL) THEN
            VAL =ABS(X(I))
            IPOS=I
         ENDIF
10    CONTINUE
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE MAXVAL(X,VAL,IPOS,NP,N)
C
C This subroutine finds the first maximum value VAL of the array X,
C and returns its position in the array IPOS. If there is more than
C one equal minimum value, IPOS will contain the position of the
```

```fortran
C first maximum.
C
      REAL      X(NP),VAL
      INTEGER IPOS,N
C
      VAL =X(1)
      IPOS=1
      DO 10 I=2,N
         IF(X(I).GT.VAL) THEN
            VAL =X(I)
            IPOS=I
         ENDIF
10    CONTINUE
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE ABSMXV(X,VAL,IPOS,NP,N)
C
C Finds the first ABSOLUTE maximum value VAL of the array X,
C and returns its position in the array IPOS. If there is more than
C one equal minimum value, IPOS will contain the position of the
C first maximum.
C
      REAL      X(NP),VAL
      INTEGER IPOS,N
C
      VAL =ABS(X(1))
      IPOS=1
      DO 10 I=2,N
         IF(ABS(X(I)).GT.VAL) THEN
            VAL =ABS(X(I))
            IPOS=I
         ENDIF
10    CONTINUE
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE UNIVEC(UVEC,IVEC,MP,M)
C
C This subroutine finds a unit vector UVEC having its only unit value
C at the position IVEC. UNIVEC has physical length MP, and actual
C length M.
C
      REAL      UVEC(MP)
      INTEGER IVEC
C
      DO 10 I=1,M
         UVEC(I)=0.
10    CONTINUE
      UVEC(IVEC)=1.0
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE DLINDX(INDEX,ID,NP,N)
C
C This subroutine removes the ID-th element of the integer
C array INDEX. It is similar to DELROW.
C
      INTEGER INDEX(NP)
C
C Test if the set INDEX is empty, if so reduce the dimension of
C INDEX by unity, write a message  and return.
      IF(N.EQ.0) THEN
         WRITE(*,*)' '
         WRITE(*,*)'You are trying to delete an index from an empty'
         WRITE(*,*)'set INDXB. This error message came from DLINDX.'
         GO to 20
      ENDIF
      DO 10 I=1,N
         IF(I.EQ.N) THEN
            INDEX(I)=0
         ELSEIF(I.LT.ID) THEN
            GO TO 10
         ELSE
            INDEX(I)=INDEX(I+1)
         ENDIF
10    CONTINUE
20    CONTINUE
      N=N-1
C
      RETURN
      END
C-------------------------------------------------------
C
      SUBROUTINE FINDMX(AC,BC,INDX1,IND1,INDX2,IND2,NE,X,Y,T,SI,P,D,
     +            CD,DMX,IA,MCP,NP,N,NI,problm)
C
C Finds which constraints that are not in the active set closest to the
C current point in the direction P, and finds the step length DMX to it.
C It will also return the position IA of this closest constraint in the
C constraints matrix AC (and constraints vector BC). If the full step
C length in the direction P is taken, the the constraint defined by AI
C will become active and is added to the active set A.
C The way this subroutine finds DMX is by determining the step length
C D(I) to each I-th passive constraint and choose the minimum of the
C positive steps as DMX (the reason we consider the positive step
C lengths only because if the step length is negative then we are
C moving away from the constraint if we are moving in the direction of
C P, so that constraint will never become active.
C
      REAL      AC(MCP,NP),BC(MCP),D(MCP),X(NP),Y(NP),T(NP),
     +          P(NP),DMX,TEMP,S1,S2
      INTEGER INDX1(NP),INDX2(MCP),NE,IA,N,KX,KY,III,II
      LOGICAL CD,problm(2)
C
C Set constant variables.
      TEMP=0.
      DMX=0.
      IA =0
      CD =.FALSE.
      III=0
C
C Multiply the constraints (rows of AC) with Z and P.
      DO 20 I=1,IND2
         S1=0.
         S2=0.
         II=0
         KX=0
         KY=0
         DO 5 JJ=NE+1,IND1
C                             next statement is only an extra caution.
C                             the one after is the real test needed.
            IF(INDX1(JJ).LT.0) GO TO 5
            IF(INDX1(JJ).EQ.INDX2(I)) THEN
               D(I)=-10000.
               GO TO 20
            ENDIF
5        CONTINUE
         DO 10 J=1,N
            S2=S2+AC(INDX2(I),J)*P(J)
            IF(problm(1)) S1=S1+AC(INDX2(I),J)*T(J)
            IF(problm(2)) THEN
               IF(MOD(J,2).GT.0) THEN
```

```
C
      RETURN
      END
C-------------------------------------------------------------
C
      SUBROUTINE UNIMAT(UNIT,NPP,N)
C
      REAL UNIT(NPP,NPP)
C
      DO 20 I=1,N
         DO 10 J=1,N
            UNIT(I,J)=0.
10       CONTINUE
         UNIT(I,I)=1.
20    CONTINUE
      RETURN
      END
C  -------------------------------------
C
      SUBROUTINE PRDMAT(A,B,C,MPP,IRP,NPP,M,IR,N)
C
      REAL A(MPP,IRP),B(IRP,NPP),C(MPP,NPP)
C
      DO 20 I=1,M
         DO 20 J=1,N
            SUM=0.
            DO 10 K=1,IR
               SUM=SUM+A(I,K)*B(K,J)
10.          CONTINUE
            C(I,J)=SUM
20    CONTINUE
      RETURN
      END
C  -------------------------------------
C
      SUBROUTINE TRANSP(ONE,TWO,MPP,NPP,M,N)
C
      REAL ONE(MPP,NPP),TWO(NPP,MPP)
C
      DO 10 I=1,M
         DO 10 J=1,N
            TWO(J,I)=ONE(I,J)
10    CONTINUE
      RETURN
      END
C  -------------------------------------
C
      SUBROUTINE PMTVEC(A,X,Y,MPP,NPP,M,N)
C
      REAL A(MPP,NPP),X(NPP),Y(MPP)
C
      DO 20 I=1,M
         SUM=0.
         DO 10 K=1,N
            SUM=SUM+A(I,K)*X(K)
10       CONTINUE
         Y(I)=SUM
20    CONTINUE
      RETURN
      END
C  -------------------------------------
C
      SUBROUTINE DELROW(A,B,INDEX1,IND1,IDEL,MP,NP,M,N)
C
C This subroutine deletes row number IDL from the matrix A, and an
C element at position IDEL from both B and INDEX1. Then adds the
C element that was deleted from INDEX1 to the end of INDEX2.
C
      REAL    A(MP,NP),B(MP)
      INTEGER INDEX1(MP),IDEL,IND1,M,N
C
      IND1=IND1-1
      DO 30 I=1,M
         IF(I.EQ.M) THEN
            DO 10 J=1,N
               A(I,J)=0.0
10          CONTINUE
            B(I)    =0.
            INDEX1(I)=0
         ELSEIF(I.LT.IDEL) THEN
            GO TO 30
         ELSE
            DO 20 J=1,N
               A(I,J)=A(I+1,J)
20          CONTINUE
            B(I)    =B(I+1)
            INDEX1(I)=INDEX1(I+1)
         ENDIF
30    CONTINUE
      M=M-1
C
      RETURN
      END
C
C-----------------------------------------------------------------
C
      SUBROUTINE ADDROW(A,B,AC,BC,INDEX,IAD,MCP,NPP,M,N,MC)
C
C This subroutine adds a row to the working set.
C
      REAL    A(NPP,NPP),B(NPP),AC(MCP,NPP),BC(MCP)
      INTEGER INDEX(NPP),IAD
C
C Increment the row dimension and test if we are trying to add a
C non-existent constraint, if so write a message and return to exit.
      M=M+1
      IF(M.GT.MC) THEN
         WRITE(*,*)' '
         WRITE(*,*)'You are trying to add a constraint numbered ',MC+1
         WRITE(*,*)'You do not have more than ',MC,' constraints.'
         WRITE(*,*)'This error message came from ADDROW.'
         GO TO 20
      ENDIF
C
C Put the IAD-th row of AC into the N-th row of A, and the IAD-th
C element of BC into the N-th element of B.
      DO 10 J=1,N
         A(M,J)=AC(IAD,J)
10    CONTINUE
      B(M)    =BC(IAD)
C
C Update INDEX.
      INDEX(M)=IAD
20    CONTINUE
C
      RETURN
      END
C-----------------------------------------------------------------
C
      SUBROUTINE CHOLSK(G,L,D,E,C,NP,M,perform)
C
      REAL G(NP,NP),L(NP,NP),D(NP),E(NP),C(NP,NP),
     +     GAMA,XI,XNU,EM,DELTA,COLSUM
      REAL DS,DR,COND
      INTEGER IPOS
      CHARACTER perform*(*)
C
C Set constants, and compute the norm of G and then DELTA.
      GAMA=0.
```

```
      XI=0.
      EM=1.92092896E-7
      COLSUM=0.
      DO 12 J=1,M
         SUM=0.
         DO 11 I=1,M
            SUM=SUM+ABS(G(I,J))
11       CONTINUE
         IF(SUM.GT.COLSUM) COLSUM=SUM
12    CONTINUE
      DELTA=AMAX1(EM*COLSUM,EM)
C
C Find the maximum magnitude diagonal element GAMA and the maximum
C magnitude off-diagonal element XI of the matrix G.
      DO 15 I=1,M
         DO 15 J=1,M
            IF(I.EQ.J) THEN
               IF(ABS(G(I,J)).GT.GAMA) GAMA=ABS(G(I,J))
            ELSE
               IF(ABS(G(I,J)).GT.XI) XI=ABS(G(I,J))
            ENDIF
15    CONTINUE
C
C Finding BS.
      XNU =AMAX1(1.,SQRT(FLOAT(M)*FLOAT(M)-1.))
      XNU=XI/XNU
      BS  =AMAX1(GAMA,XNU,EM)
C
C Put diagonal of G into diagonal of C.
      DO 20 I=1,M
         C(I,I)=G(I,I)
20    CONTINUE
C
C Start Looping.
      J=0
25    J=J+1
C
C Step 4.
      DO 45 K=1,J-1
         L(J,K)=C(J,K)/D(K)
45    CONTINUE
      DO 55 I=J+1,M
         SUM=0.
         DO 50 K=1,J-1
            SUM=SUM+L(J,K)*C(I,K)
50       CONTINUE
         C(I,J)=G(I,J)-SUM
55    CONTINUE
      THETA=0.
      IF(J.EQ.M) GO TO 65
      DO 60 I=J+1,M
         IF(THETA.LE.ABS(C(I,J))) THETA=ABS(C(I,J))
60    CONTINUE
65    CONTINUE
      D(J)=AMAX1(DELTA,ABS(C(J,J)),THETA*THETA/BS)
      E(J)=D(J)-C(J,J)
      IF(J.EQ.M) GO TO 80
      DO 70 I=J+1,M
         C(I,I)=C(I,I)-C(I,J)*C(I,J)/D(J)
70    CONTINUE
C Return to loop.
      GO TO 25
80    CONTINUE
C
C Set diagonal elements of L to unity, and restore DELTA.
      DO 85 I=1,M
         L(I,I)=1.
85    CONTINUE
C
      CALL MINVAL(D,DR,IPOS,NP,M)
      CALL MAXVAL(D,DS,IPOS,NP,M)
      COND=DS/DR
      IF(perform.EQ.'sc') THEN
         IF(COND.LT.10.) WRITE(6,112) COND
         IF((COND.GE.10.).AND.(COND.LT.100.)) WRITE(6,113) COND
         IF((COND.GE.100.).AND.(COND.LT.1000.)) WRITE(6,114) COND
         IF((COND.GE.1000.).AND.(COND.LT.10000.)) WRITE(6,115) COND
         IF((COND.GE.10000.).AND.(COND.LT.100000.)) WRITE(6,116) COND
         IF((COND.GE.100000.).AND.(COND.LT.1000000.)) WRITE(6,117) COND
         IF((COND.GE.1000000.)) WRITE(6,118) COND
      ENDIF
      IF(perform.EQ.'fl') THEN
         IF(COND.LT.10.) WRITE(15,112) COND
         IF((COND.GE.10.).AND.(COND.LT.100.)) WRITE(15,113) COND
         IF((COND.GE.100.).AND.(COND.LT.1000.)) WRITE(15,114) COND
         IF((COND.GE.1000.).AND.(COND.LT.10000.)) WRITE(15,115) COND
         IF((COND.GE.10000.).AND.(COND.LT.100000.)) WRITE(15,116) COND
         IF((COND.GE.100000.).AND.(COND.LT.1000000.)) WRITE(15,117) COND
         IF((COND.GE.1000000.)) WRITE(15,118) COND
      ENDIF
112   FORMAT(3X,'Hc=',F7.5,$)
113   FORMAT(3X,'Hc=',F7.4,$)
114   FORMAT(3X,'Hc=',F7.3,$)
115   FORMAT(3X,'Hc=',F7.2,$)
116   FORMAT(3X,'Hc=',F7.1,$)
117   FORMAT(3X,'Hc=',F7.0,$)
118   FORMAT(3X,'Hc=',E7.2,$)
C
      RETURN
      END
C-----------------------------------------------------------------
C
      SUBROUTINE PARTQ(Q,Q1,Q2,NP,N,INDX,NQ2)
C
C Partitions the matrix Q into two matrices Q1 and Q2.
C
      REAL    Q(NP,NP),Q1(NP,NP),Q2(NP,NP)
      INTEGER N,INDX,NQ2
C
C Start at column 1 and loop to column N of Q.
      DO 30 J=1,N
         IF(J.LE.INDX) THEN
            DO 10 I=1,N
               Q1(I,J)=Q(I,J)
10          CONTINUE
         ELSE
            DO 20 I=1,N
               Q2(I,J-INDX)=Q(I,J)
20          CONTINUE
         ENDIF
30    CONTINUE
      NQ2=N-INDX
C
      RETURN
      END
C-----------------------------------------------------------------
C
      SUBROUTINE FBSUBA(A,X,B,FBFLAG,NP,N)
C
C
C This subroutine performs both back- and forward-substitution depending
C on the value of FBFLAG:
C                   FBFLAG=-1.0 then performs back-substitution.
C                   FBFLAG=+1.0 then performs forward-substitution.
C
C IMPORTANT:
C          The diagonal elements of the matrix A SHOULD equal unity;
C          i.e., each diagonal element should equal to 1.0.
C Basically this subroutine solves the linear system AX=B, where A is
```

```
       READ(11,*) GARBG
       READ(11,*) NCREC
       IF(NREC.NE.NCREC) THEN
         PRINT*,'** Fatal Error: RECORD number from GDATA: ',NREC
         PRINT*,'                RECORD number from CDATA: ',NCREC
         PRINT*,'                they are NOT the same. Only previous'
         PRINT*,'                records are used.'
C
C Go to plot previous records results.
         GO TO 395
       ENDIF
       READ(11,*) GARBG
       DO 766 I=1,NI
*        IF(I.EQ.NI) THEN
*          READ(11,*) NTXYD,
*             LOGXEB(I),XEB(I),LOGXLB(I),XLB(I),LOGXUB(I),XUB(I),
*             LOGYEB(I),YEB(I),LOGYLB(I),YLB(I),LOGYUB(I),YUB(I)
*        ELSE
           READ(11,*) NTXYD,
     +          LOGTEB(I),TEB(I),LOGTLB(I),TLB(I),LOGTUB(I),TUB(I),
     +          LOGXEB(I),XEB(I),LOGXLB(I),XLB(I),LOGXUB(I),XUB(I),
     +          LOGYEB(I),YEB(I),LOGYLB(I),YLB(I),LOGYUB(I),YUB(I)
*        ENDIF
 766   CONTINUE
*      ELSE
*        DO 767 I=1,NI
*          LOGTEB(I)=.FALSE.
*             TEB(I)=0.
*          LOGTLB(I)=.TRUE.
*             TLB(I)=SI
*          LOGTUB(I)=.FALSE.
*             TUB(I)=0.
*          LOGXEB(I)=.FALSE.
*             XEB(I)=0.
*          LOGXLB(I)=.TRUE.
*             XLB(I)=2000.
*          LOGXUB(I)=.FALSE.
*             XUB(I)=0.
*          LOGYEB(I)=.TRUE.
*             YEB(I)=0.
*          LOGYLB(I)=.FALSE.
*             YLB(I)=0.
*          LOGYUB(I)=.FALSE.
*             YUB(I)=0.
* 767     CONTINUE
       ENDIF
       READ(12,REC=NREC) (trace(I),I=1,NSAMPT)
       DO 9 I=1,NSWIN
         RSPO(I)=trace(NSSS-1+I)
         SEIDAT(I,JK)=trace(NSSS-1+I)
 9     CONTINUE
       ERSPO=VNORM(RSPO,MP,NSWIN)/2.
C
       IF(perform.EQ.'sc') THEN
         IF(prblm.EQ.'ai') WRITE(6,'(''* ai:'',$)')
         IF(prblm.EQ.'bn') WRITE(6,'(''* bn:'',$)')
       ENDIF
       IF(perform.EQ.'fl') THEN
         IF(prblm.EQ.'ai') WRITE(15,'(''* ai:'',$)')
         IF(prblm.EQ.'bn') WRITE(15,'(''* bn:'',$)')
       ENDIF
 467   CONTINUE
       CHDIST=.FALSE.
       TERMNT=.FALSE.
       DELION=.FALSE.
       ADDION=.FALSE.
       SINGUL=.FALSE.
       NEGLAG=.FALSE.
       IMPROV=.FALSE.
       FVIOL =.FALSE.
       VIO   =.FALSE.
       KEEPJG=.FALSE.
       DONEA =.FALSE.
       IF(problm(1)) N=NI-1
       IF(problm(2)) N=2*NI
       CALL ZERO2D(A,NP,NP,NP,NP)
       CALL ZERO2D(R,NP,NP,NP,NP)
       CALL ZERO2D(Q1,NP,NP,NP,NP)
       CALL ZERO2D(Q1T,NP,NP,NP,NP)
       CALL ZERO2D(H,NP,NP,NP,NP)
       CALL ZERO2D(HP,NP,NP,NP,NP)
       CALL ZERO2D(HTE,NP,NP,NP,NP)
       CALL ZERO1D(INDXA,NP,NP)
       CALL ZERO1D(INDXB,MC2P,MC2P)
       CALL ZERO1D(INDXA1,NP,NP)
       CALL ZERO1D(INDXB1,MC1P,MC1P)
       CALL ZERO1D(INDXA2,NP,NP)
       CALL ZERO1D(INDXB2,MC2P,MC2P)
       INDA1=0
       INDB1=0
       INDA2=0
       INDB2=0
       INDA=0
       INDB=0
       NECT =0
       NECXY=0
C
C Set up the constraints matrices.
       DONEA=.FALSE.
       IF(problm(1)) CALL VLT(T,LOGTEB,TEB,LOGTLB,TLB,LOGTUB,TUB,
     +   A1,B1,AC1,BC1,INDXA1,INDXB1,SAVEQ,MAE1,MAC1,INDA1,INDB1,
     +   MC1P,NP,N,NI,SI,NECT,VIO)
       IF(problm(2)) CALL VLXY(X,LOGXEB,XEB,LOGXLB,XLB,LOGXUB,XUB,
     +   Y,LOGYEB,YEB,LOGYLB,YLB,LOGYUB,YUB,A2,B2,AC2,BC2,INDXA2,
     +   INDXB2,SAVEQ,MAE2,MAC2,INDA2,INDB2,MC2P,NP,NI,NECXY,VIO)
       IF(VIO) GO TO 99
C
C Find wavelet, compute synthetic seismogram, error vector and energy.
       IF(.NOT.(WWEINR.OR.READWV)) THEN
         CALL GWV(WP,SI,TDUR,WV,LWP,LWV,LWEST,
     +            PH,CWV,AWV,FWV,PVIOL,LWSHRT)
         IF(FVIOL.OR.LWSHRT) GO TO 99
         READWV=.TRUE.
       ENDIF
       IF(WWEINR) LWV=LWWNR
       M=0
       CALL RESPON(X,Y,T,NP,NI,WV,LWP,LWV,SI,RSPI,MP,M,RSPT,MWP,ACOU,REF,
     +NSWIN)
       CALL SUBVEC(RSPI,RSPO,DSC1,MP,M)
       IF(M.NE.NSWIN) THEN
         PRINT*,'            M= ',M
         PRINT*,'        NSWIN= ',NSWIN
         PRINT*,'       RECNUM= ',NREC
         PRINT*,' Traces used= ',JK
         PRINT*
         PRINT*,' Warning: ONLY the above number of traces was used.'
         PRINT*,'          Since NSWIN not equal to RECNUM.'
         PRINT*
C
C Plot results so far
         GO TO 395
       ENDIF
       K=1
       ER(K)=VNORM(DSC1,MP,M)/2.
       EPCENT=100.*ER(K)/ERSPO
       IF(perform.EQ.'sc') THEN
         WRITE(6,'(5X,''**ER(1)='',E12.7)') ER(1)
         IF(EPCENT.GE.10000.) THEN
           WRITE(6,'(''          EE(%)='',F7.1)') EPCENT
```

```
         ELSEIF(EPCENT.GE.1000.) THEN
           WRITE(6,'(''          EE(%)='',F7.2)') EPCENT
         ELSEIF(EPCENT.GE.100.) THEN
           WRITE(6,'(''          EE(%)='',F7.3)') EPCENT
         ELSEIF(EPCENT.GE.10.) THEN
           WRITE(6,'(''          EE(%)='',F7.4)') EPCENT
         ELSEIF(EPCENT.GE.1.) THEN
           WRITE(6,'(''          EE(%)='',F7.5)') EPCENT
         ELSE
           WRITE(6,'(''          EE(%)='',F7.6)') EPCENT
         ENDIF
       ENDIF
       IF(perform.EQ.'fl') THEN
         WRITE(15,'(5X,''**ER(1)='',E12.7)') ER(1)
         IF(EPCENT.GE.10000.) THEN
           WRITE(15,'(''          EE(%)='',F7.1)') EPCENT
         ELSEIF(EPCENT.GE.1000.) THEN
           WRITE(15,'(''          EE(%)='',F7.2)') EPCENT
         ELSEIF(EPCENT.GE.100.) THEN
           WRITE(15,'(''          EE(%)='',F7.3)') EPCENT
         ELSEIF(EPCENT.GE.10.) THEN
           WRITE(15,'(''          EE(%)='',F7.4)') EPCENT
         ELSEIF(EPCENT.GE.1.) THEN
           WRITE(15,'(''          EE(%)='',F7.5)') EPCENT
         ELSE
           WRITE(15,'(''          EE(%)='',F7.6)') EPCENT
         ENDIF
       ENDIF
C
C If have not done yet, set TN=T, XN=X, YN=Y, GIIMP,GIRSP,GIERR.
       IF(.NOT.(iginfo)) THEN
         DO 17 I=1,NI
           TN(I)=T(I)
           XN(I)=X(I)
           YN(I)=Y(I)
           TIG(I,JK)=T(I)
           XIG(I,JK)=X(I)
           YIG(I,JK)=Y(I)
 17      CONTINUE
         DO 11 I=1,M
           GIIMP(I,JK)=ACOU(I)
           GIRSP(I,JK)=RSPI(I)
           GIERR(I,JK)=DSC1(I)
 11      CONTINUE
         iginfo=.TRUE.
       ENDIF
C
       IF(ER(K).LE.ERRLMT) THEN
         WRITE(*,*)' '
         WRITE(*,*)'*IGUESS: ',JK,' of Record: ',nrec,' is perfect.'
         DO 12 I=1,M
           FSIMP(I,JK)=ACOU(I)
           FSRSP(I,JK)=RSPI(I)
           FSERR(I,JK)=DSC1(I)
           IF(I.LE.NI) THEN
             XFS(JK,I)=X(I)
             YFS(JK,I)=Y(I)
             TFS(JK,I)=T(I)
           ENDIF
 12      CONTINUE
         GO TO 999
       ENDIF
C                 THIS IS THE RE-ITERATION POINT.
 35    CONTINUE
C
       IF(perform.EQ.'sc') THEN
         IF(K.LT.10) WRITE(*,'(4X,''K='',I1,$)') K
         IF((K.GE.10).AND.(K.LT.100)) WRITE(*,'(3X,''K='',I2,$)') K
         IF((K.GE.100).AND.(K.LT.1000)) WRITE(*,'(2X,''K='',I3,$)') K
       ENDIF
       IF(perform.EQ.'fl') THEN
         IF(K.LT.10) WRITE(15,'(4X,''K='',I1,$)') K
         IF((K.GE.10).AND.(K.LT.100)) WRITE(15,'(3X,''K='',I2,$)') K
         IF((K.GE.100).AND.(K.LT.1000)) WRITE(15,'(2X,''K='',I3,$)') K
       ENDIF
C
C Set the matrix A and its dimensions.
       IF(.NOT.DONEA) THEN
         IF(problm(1)) THEN
           INDA=INDA1
           INDB=INDB1
           NA  =N
           IF(INDA.EQ.0) THEN
             MA=N
             DO 36 I=1,INDB
               INDXB(I)=INDXB1(I)
 36          CONTINUE
           ELSE
             MA=INDA
             DO 37 I=1,MA
               B(I)=B1(I)
               DO 37 J=1,NA
                 A(I,J)=A1(I,J)
 37          CONTINUE
             DO 38 I=1,INDA
               INDXA(I)=INDXA1(I)
 38          CONTINUE
             DO 338 I=1,INDB
               INDXB(I)=INDXB1(I)
 338         CONTINUE
           ENDIF
         ENDIF
C
         IF(problm(2)) THEN
           INDA=INDA2
           INDB=INDB2
           NA  =N
           IF(INDA.EQ.0) THEN
             MA=N
             DO 39 I=1,INDB
               INDXB(I)=INDXB2(I)
 39          CONTINUE
           ELSE
             MA=INDA
             DO 40 I=1,MA
               B(I)=B2(I)
               DO 40 J=1,NA
                 A(I,J)=A2(I,J)
 40          CONTINUE
             DO 41 I=1,INDA
               INDXA(I)=INDXA2(I)
 41          CONTINUE
             DO 441 I=1,INDB
               INDXB(I)=INDXB2(I)
 441         CONTINUE
           ENDIF
         ENDIF
C
C Find QR-decomposition of AT.
         CALL TRANSP(A,AT,NP,NP,MA,NA)
         CALL QR(AT,QT,R,SAVEQ,NP,NP,NA,MA)
         CALL TRANSP(QT,Q,NP,NP,NA,NA)
         CALL PARTQ(Q,Q1,Z,NP,NA,INDA,NZ)
         CALL TRANSP(Q1,Q1T,NP,NP,NA,NA)
         CALL TRANSP(Z,ZT,NP,NP,NA,NZ)
C Compute the initial Jacobian DJM and DJMT.
         CALL JCB(X,Y,T,NI,WV,LWP,LWV,SI,RSPI,RSPP,RSPM,ACOU,REF,RSPT,
     +     MWP,DJM,MP,NP,M,N,CENTRL,problm,NSWIN)
```

```
C
      COMPLEX CWV(LWP)
      LOGICAL FVIOL,LWSHRT
C
C This subroutine creates a wavelet defined by a time duration T, four
C bounding frequencies WP(1), WP(2), WP(3) and WP(4), two amplitude
C parameters WP(5) and WP(6) on WP(2) and WP(3), respectively, a
C constant phase component WP(7), a linear phase component WP(8), a
C quadratic phase componenet WP(9) and a sampling interval SI.
C
      FVIOL=.FALSE.
      LWSHRT=.FALSE.
C
C Determine the the number of frquency components and samples in FWV
C (and AWV) and WV, respectively. Also determine the Nyquist frequency.
      LW=INT(T/SI)
      FN =500./SI
C
C Determine the discrete frequencies and put in FWV.
      DO 10 I=1,LW
        FWV(I)=(FLOAT(I-1)*1000.)/T
 10   CONTINUE
C
C If any of WP(1:4) is not in FWV, set FVIOL=.TRUE., and return.
      DO 15 I=1,4
        IF(WP(I).GT.FWV(LW/2+1)) THEN
          WRITE(*,*)'A bounding frequency > or = the Nyquist.'
          FVIOL=.TRUE.
          RETURN
        ENDIF
 15   CONTINUE
C
C Find slopes of amplitude spectrum envelope.
      SLPL=WP(5)/(WP(2)-WP(1))
      SLPR=WP(6)/(WP(4)-WP(3))
      SLPM=(WP(5)-WP(6))/(WP(3)-WP(2))
C
C Determine the amplitudes of AWV.
      J=2
      DO 20 I=1,LW
        IF(FWV(I).LT.FN) THEN
          IF(FWV(I).LE.WP(1)) THEN
            AWV(I)=0.
          ELSEIF((FWV(I).GT.WP(1)).AND.(FWV(I).LE.WP(2))) THEN
            AWV(I)=SLPL*(FWV(I)-WP(1))
          ELSEIF((FWV(I).GT.WP(2)).AND.(FWV(I).LE.WP(3))) THEN
            AWV(I)=WP(5)+SLPM*(FWV(I)-WP(2))
          ELSEIF((FWV(I).GT.WP(3)).AND.(FWV(I).LE.WP(4))) THEN
            AWV(I)=SLPR*(WP(4)-FWV(I))
          ELSE
            AWV(I)=0.
          ENDIF
          PH(I)=-(WP(7)+WP(8)*FWV(I)+WP(9)*FWV(I)*FWV(I))
        ELSEIF(FWV(I).EQ.FN) THEN
          AWV(I)=0.
          PH(I) =0.
        ELSE
          AWV(I)=AWV(I-J)
          PH(I) =-PH(I-J)
          J=J+2
        ENDIF
        CWV(I)=CMPLX(AWV(I)*COS(PH(I)),AWV(I)*SIN(PH(I)))
 20   CONTINUE
C
C  Transforming the complex wavelet transform into the time domain.
      CALL FORK(LWP,LW,CWV,1.)
C
C Take real part of CWV as wavelet WV.
      DO 30 I=1,LWV
        WV(I)=REAL(CWV(I))
 30   CONTINUE
C
      RETURN
      END
C---------------------------------------------------------------
C
      SUBROUTINE SUBVEC(X,Y,G,MPP,M)
C
C Subtracts Y from X put result in G.
C
      REAL X(MPP),Y(MPP),G(MPP)
C
      DO 10 I=1,M
        G(I)=X(I)-Y(I)
 10   CONTINUE
      RETURN
      END
C ----------------------------------------------------------
C
      REAL FUNCTION VNORM(X,NPP,N)
C
C Finds norm of X, which is inner product of X with itself.
C
      REAL X(NPP)
C
      SUM = 0.
      DO 10 I=1,N
        SUM = SUM+X(I)*X(I)
 10   CONTINUE
      VNORM = SUM
      RETURN
      END
C -----------------------------------------------------------
C
      REAL FUNCTION PRODIN (X,Y,NPP,N)
C
C This function finds the inner product of two vectors
C X and Y, each of length N.
C
      REAL X(NPP),Y(NPP)
C
      PRODIN = 0.
      DO 10 I=1,N
        PRODIN = PRODIN+X(I)*Y(I)
 10   CONTINUE
C
      RETURN
      END
C -----------------------------------------------------------
C
      SUBROUTINE ZERO2D(A,MPP,NPP,M,N)
C
      REAL A(MPP,NPP)
      DO 10 I=1,M
        DO 10 J=1,N
          A(I,J)=0.
 10   CONTINUE
      RETURN
      END
C -----------------------------------------------------------
C
      SUBROUTINE ZERO1D(A,MPP,M)
C
      REAL A(MPP)
      DO 10 I=1,M
        A(I)=0.
 10   CONTINUE
      RETURN
      END
C-----------------------------------------------------------
```

```
C
      SUBROUTINE QR(A,Q,R,SAVEQ,MPP,NPP,M,N)
C
C This subroutine finds the QR-decomposition of the matrix A which
C have the physical dimensions (MP,NP), and the actual dimensions
C (M,N). The output will be the (M,M) matrix Q, and the (M,N) matrix
C R. The physical dimensions of the matrices Q and R are (MP,MP) and
C (MP,NP), respectively.
C
      REAL A(MPP,NPP),R(MPP,NPP),Q(MPP,MPP),SAVEQ(MPP),XM,SIGMA,GAMA
      INTEGER M,N
C
C Zero out R and SAVEQ.
      DO 2 I=1,MPP
        SAVEQ(I)=0.
        DO 2 J=1,NPP
          R(I,J)=0.
 2    CONTINUE
C
C Put A in R, because we actually compute the decomposition in R.
C On output, however, R will contain the upper-triangular matrix R
C of the QR-decomposition.
      DO 3 I=1,M
        DO 3 J=1,M
          R(I,J)=A(I,J)
 3    CONTINUE
C
C Set the matrix Q to be the identity.
      CALL UNIMAT(Q,MPP,M)
C
C Start decomposing the input matrix R into Q and output R. We star
C by the first column in the input matrix R and compute the rotating
C matrix Q that will rotate the first column to one axis then take the
C second column and compute the matrix Q that will rotate it to two-
C axis plane and so on until we rotate all the columns if we have less
C columns than rows or as many columns as rows if we have more columns
C than rows.
      DO 99 J=1,N
        SIGMA=0.
        XM=0.
        GAMA=0.
C
C Find the scaling factor XM, which is the element of largest magnitude
C of the current j-TH column of the matrix R, and use it to normalize
C the column vector of R.
        DO 5 I=J,M
          IF(ABS(R(I,J)).GT.XM) XM=ABS(R(I,J))
 5      CONTINUE
C
C If the j-TH column of A is zero, then leave GAMA=0. and compute Q=I.
C Otherwise continue to compute Q. Normalize the current j_TH column
C of R first.
        IF(XM.EQ.0.) GO TO 25
        DO 10 I=J,M
          R(I,J)=R(I,J)/XM
 10     CONTINUE
C Compute SIGMA.
        SUM=0.
        DO 15 I=J,M
          SUM=SUM+R(I,J)*R(I,J)
 15     CONTINUE
        SIGMA=XM*SQRT(SUM)
C
C Let SIGMA take the sign of the first element of the current vector.
        IF(R(J,J).LT.0.) SIGMA=-SIGMA
C
C Let the j-TH column of R contain U (after upscaling the vector by XM).
        DO 20 I=J,M
          IF(I.EQ.J) THEN
            R(I,J)=SIGMA+XM*R(I,J)
          ELSE
            R(I,J)=XM*R(I,J)
          ENDIF
 20     CONTINUE
C
C Compute GAMA.
        GAMA=1./(SIGMA*R(J,J))
        IF(GAMA.LT.1.0E-20) WRITE(*,300) J
 300    FORMAT(15X,'The matrix is singular at the ',I2,'_TH iteration')
C
C We jump to the next statement if GAMA=0, i.e., when the vector is zero.
C That is, when the matrix is singular.
C
 25     CONTINUE
C Compute Q.
        DO 45 JJ=1,M
          DO 30 I=1,M
            SAVEQ(I)=Q(I,JJ)
 30       CONTINUE
          DO 45 I=1,M
            SUM=0.
            IF(I.LT.J) THEN
              SUM=SUM+SAVEQ(I)
            ELSE
              DO 40 II=J,M
                SAVE=R(II,J)*R(I,J)
                IF(II.EQ.I) THEN
                  SAVE=1.-GAMA*SAVE
                ELSE
                  SAVE=-GAMA*SAVE
                ENDIF
                SUM=SUM+SAVEQ(II)*SAVE
 40           CONTINUE
            ENDIF
            Q(I,JJ)=SUM
 45       CONTINUE
C
C Form R by applying Q to all the column vectors of R. When we are
C applying Q to a vector we do not actually multiply Q by that vector;
C i.e., Qv where v is the vector, what we do is this: Q is not evaluated
C but is represented as Q=I-GAMA*uu'. So that, Qv=(I-GAMA*uu')v, thus
C Qv=v-cu, where the scalar c=GAMA*u'v. Here ' denotes Transpose.
C Note that if we are at the last column we do not need to apply Q to
C any column other than the last so we GO TO 56.
        IF(J.EQ.N) GO TO 56
        DO 55 JJ=J+1,N
          C=0.
          DO 50 II=J,M
            C=C+R(II,J)*R(II,JJ)
 50       CONTINUE
          C=C*GAMA
          DO 55 II=J,M
            R(II,JJ)=R(II,JJ)-C*R(II,J)
 55     CONTINUE
 56     CONTINUE
        DO 82 II=1,M
          SAVEQ(II)=R(II,J)
 82     CONTINUE
        R(J,J)=R(J,J)-SIGMA
        C=0.
        DO 80 II=J,M
          C=C+SAVEQ(II)*R(II,J)
 80     CONTINUE
        C=C*GAMA
        DO 81 II=J,M
          R(II,J)=R(II,J)-C*SAVEQ(II)
 81     CONTINUE
 99   CONTINUE
```

```
      PROGRAM IVO
c
c This program performs nonlinear least-squares inversion with
c linear equality and inequality constraints.
c
      CHARACTER BLK*1
      PARAMETER(MP=1024,NP=100,NTP=720,LWP=256,MWP=MP+LWP-1,
     +          MC1P=2*NP,MC2P=4*NP,MC3P=18,BLK=' ')
c
      REAL X(NP),XN(NP),Y(NP),YN(NP),T(NP),TN(NP),WP(9),
     +     WV(LWP),PH(LWP),TEB(NP),TLB(NP),TUB(NP),
     +     XEB(NP),XLB(NP),XUB(NP),YEB(NP),YLB(NP),YUB(NP),
     +     ER(500),ES(NP),RSPI(NP),RSPI2(MP),RSPO(MP),
     +     RSPT(MWP),RSPP(MP),RSPM(MP),DJM(MP,MP),DJMT(NP,MP),
     +     DJM2(MP,NP),DJM2T(NP,MP),Q1(NP,NP),Q1T(NP,NP),Q1TG(NP,NP),
     +     VECLAG(NP),Z(NP,NP),ZT(NP,NP),H(NP,NP),HTE(NP,NP),HP(NP,NP),
     +     G(NP),GP(NP),GP1(NP),DSC1(MP),DSC2(MP),TK(NP),P(NP),PZ(NP),
     +     A1(NP,NP),B1(NP,NP),A2(NP,NP),B2(NP),A(NP,NP),AT(NP,NP),
     +     B(NP),AC1(MC1P,NP),BC1(MC1P),AC2(MC2P,NP),BC2(MC2P),
     +     Q(NP,NP),R(NP,NP),QT(NP,NP),SAVEQ(MP),CL(NP,NP),CLT(NP,NP),
     +     CD(NP),CDRCP(NP),CE(NP),SIGS(NP),DIST(MC2P),ACOU(MP),REF(MP),
     +     DSCN(MP),DMAX,GPC(NP),trace(2000),AWV(LWP),FWV(LWP),HPK(NP),
     +     HPKG(NP),ALF,PGP1,ERT,PSTPMN,ERRLMT,FMAG,ALFLMTT,ALFLMTX,
     +     ALFLMTY,PNORM1,PNORM2,ALFMAX,TDUR,SI,CC(NP,NP),ERSPO,EPCENT,
     +     GIIMP(MP,NTP),GIRSP(MP,NTP),GIERR(MP,NTP),FSIMPSC(MP,NTP),
     +     FSIMP(MP,NTP),FSRSP(MP,NTP),FSERR(MP,NTP),VIMAX,VALMAX,
     +     SEIDAT(MP,NTP),XIG(50,NTP),YIG(50,NTP),TIG(50,NTP),
     +     XFS(50,NTP),YFS(50,NTP),TFS(50,NTP)
     +     ,SINTB(MP),PRQHC
      INTEGER INDXA1(NP),INDXA2(NP),INDXA(NP),INDXB1(MC1P),INDXB2(MC2P),
     +        INDXB(MC2P),INDA1,INDA2,INDB1,INDB2,INDA,INDB,IADD,
     +        KX,KY,NECT,NECXY,LWEST,LWV,irp,NI,M,LWWNR,KLIMIT,
     +        NSSS,NSWIN,NSAMPT,iinput,ipos,NREC,NCREC,LENREC,NIMAX,
     +        NTOTREC(NTP),FSIMAX(NTP)
      COMPLEX CWV(LWP)
      LOGICAL LOGTEB(NP),LOGTLB(NP),LOGTUB(NP),LOGXEB(NP),LOGXLB(NP),
     +        LOGXUB(NP),LOGYEB(NP),LOGYLB(NP),CHDIST,TERMNT,
     +        DELION,ADDION,SINGUL,NEGLAG,IMPROV,FVIOL,DONEA,retry,
     +        LNSHRT,CENTRL,VIO,KEEPJG,problm(2),READWV,CSTATE,
     +        PRESNT,WWEINR,bnconv,aiconv,iginfo,PCHNGED,IMPNORM
      CHARACTER GARBG*15/BLK/,FNWWNR*60/BLK/,FNGDAT*60/BLK/,
     +          FNGCON*60/BLK/,FNSEIS*60/BLK/,ibuff*32/BLK/,
     +          ynans*1/BLK/,prblm*2/BLK/,perform*2/BLK/,
     +          FORMX*80/BLK/,FORMY*80/BLK/,FORMT*80/BLK/
      CHARACTER PTYPE*40/BLK/,AIORSEI*6/BLK/
$OPTION RANGE ON
$OPTION LIST ON
*      external common_handler
*
c
      WRITE(*,'(/)')
      PRINT*,'ivob V.26.8.96'
      WRITE(*,'(/)')
*
*      iee=ieee_handler("set","common",common_handler)
*      if(iee.ne.0) PRINT*,'Could not establish fp signal handler.'
*
c
      LENREC=6000
      irp=0
725   CONTINUE
*      PRINT*
      WRITE(*,726)
726   FORMAT(' ivob> Which PROBLEM to always solve FIRST? (bn/ai): ',$)
      READ(5,'(A)') prblm
      IF(prblm.EQ.'bn') THEN
        problm(1)=.TRUE.
        problm(2)=.FALSE.
      ELSEIF(prblm.EQ.'ai') THEN
        problm(2)=.TRUE.
        problm(1)=.FALSE.
      ELSE
        irp=irp+1
        IF(irp.GE.4) STOP': program terminated.'
        PRINT*,' Warning: Please answer bn/ai.'
        GO TO 725
      ENDIF
      irp=0
750   CONTINUE
      PRINT*
      WRITE(6,751)
751   FORMAT(' ivob> Use WEINER wavelet estimate? (y/n): ',$)
      READ(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) THEN
        WWEINR=.TRUE.
752     CONTINUE
*        PRINT*
        WRITE(6,753)
753     FORMAT(' qnrly> Enter WEINER wavelet FILE name: ',$)
        READ(5,'(A)') FNWWNR
        INQUIRE(FILE=FNWWNR(:INDEX(FNWWNR,' ')-1),EXIST=PRESNT)
        IF(.NOT.PRESNT) THEN
          irp=irp+1
          IF(irp.GT.4) STOP': program terminated.'
          PRINT*,' Warning: Enter another WEINER wavelet file.'
          GO TO 752
        ENDIF
        OPEN(UNIT=9,FILE=FNWWNR(:INDEX(FNWWNR,' ')-1))
        LWWNR=0
754     CONTINUE
        LWWNR=LWWNR+1
        READ(9,FMT=*,IOSTAT=ierr) WV(LWWNR)
        IF(ierr.EQ.0) GO TO 754
        IF(ierr.GT.0)STOP' :ERROR reading unit attatched to WEINER WV'
        LWWNR=LWWNR-1
      ELSEIF((ynans.EQ.'N').OR.(ynans.EQ.'n')) THEN
        WWEINR=.FALSE.
      ELSE
        irp=irp+1
        IF(irp.GT.4) STOP': program terminated.'
        PRINT*,' Warninig: Answer y/n.'
        GO TO 750
      ENDIF
c
      irp=0
755   CONTINUE
      PRINT*
      WRITE(6,756)
756   FORMAT(' ivob> Enter GUESS DATA FILE name: ',$)
      READ(5,'(A)') FNGDAT
      INQUIRE(FILE=FNGDAT(:INDEX(FNGDAT,' ')-1),EXIST=PRESNT)
      IF(.NOT.PRESNT) THEN
        irp=irp+1
        IF(irp.GT.4) STOP': program terminated.'
        PRINT*,' Warninig: the File entered does not exist.'
        PRINT*,'          Enter another GUESS DATA file.'
        GO TO 755
      ENDIF
      irp=0
757   CONTINUE
*      PRINT*
      WRITE(6,758)
758   FORMAT(' ivob> Enter CONSTRAINT FILE name: ',$)
      READ(5,'(A)') FNGCON
      INQUIRE(FILE=FNGCON(:INDEX(FNGCON,' ')-1),EXIST=PRESNT)
      IF(.NOT.PRESNT) THEN
        irp=irp+1
        IF(irp.GT.4) STOP': program terminated.'
        PRINT*,' Warninig: the File entered does not exist.'
```

```
        PRINT*,'          Enter another GUESS CONSTRAINTS file.'
        GO TO 757
      ENDIF
      irp=0
759   CONTINUE
*      PRINT*
      WRITE(6,760)
760   FORMAT(' ivob> Enter SEISMICDAT FILE name: ',$)
      READ(5,'(A)') FNSEIS
      INQUIRE(FILE=FNSEIS(:INDEX(FNSEIS,' ')-1),EXIST=PRESNT)
      IF(.NOT.PRESNT) THEN
        irp=irp+1
        IF(irp.GT.4) STOP': program terminated.'
        PRINT*,' Warninig: the File entered does not exist.'
        PRINT*,'          Enter another SEISMIC DATA file.'
        GO TO 759
      ENDIF
      OPEN(UNIT=10,FILE=FNGDAT(:INDEX(FNGDAT,' ')-1))
      OPEN(UNIT=11,FILE=FNGCON(:INDEX(FNGCON,' ')-1))
      OPEN(UNIT=12,FILE=FNSEIS(:INDEX(FNSEIS,' ')-1),ACCESS='DIRECT',
     +     RECL=LENREC)
c
      PRINT*
      irp=0
761   continue
      write(6,762)
762   format(' ivob> Enter the START seismic sample : ',$)
      read(5,'(A)') ibuff
      retry=.false.
      call redchr(ibuff,iinput,retry,ipos)
      if(retry) then
        print*,' Warning: Non-numeric character at index:',ipos
        irp=irp+1
        IF(irp.gt.3) stop' Fatal Error: too many trials.'
        print*,'          Enter the START seismic SAMPLE again.'
        irp=irp+1
        go to 761
      endif
      nsss=iinput
      irp=0
763   continue
      write(6,764)
764   format(' ivob> Seismic WINDOW length (samples): ',$)
      read(5,'(A)') ibuff
      retry=.false.
      call redchr(ibuff,iinput,retry,ipos)
      if(retry) then
        print*,' Warning: Non-numeric character at index:',ipos
        irp=irp+1
        IF(irp.gt.3) stop' fatal error: too many trials.'
        print*,'          Enter Seismic WINDOW length (samples) again.'
        irp=irp+1
        go to 763
      endif
      nswin=iinput
c
769   continue
      ynans=BLK
      write(6,768)
768   format(' ivob> Want PERFORMANCE information (y/n): ',$)
      read(5,'(A)') ynans
      IF((ynans.EQ.'Y').OR.(ynans.EQ.'y')) THEN
        write(6,770)
770     format(' ivob> Information on SCREEN or in FILE (sc/fl): ',$)
        read(5,'(A)') perform
        IF(perform.EQ.'sc') THEN
          CONTINUE
        ELSEIF(perform.EQ.'fl') THEN
          OPEN(15,FILE='ivob.per')
          PRINT*,'** Performance file name is called: ivob.per'
        ELSE
          PRINT*,' Waxning: Expexted answer sc or fl.'
          PRINT*,'          Start again.'
          GO TO 769
        ENDIF
      ELSEIF((ynans.EQ.'N').OR.(ynans.EQ.'n')) THEN
        perform=blk
      ELSE
        PRINT*,' Warning: Expexted answer y or n.'
        PRINT*,'          Start again.'
        GO TO 769
      ENDIF
c
      CENTRL=.FALSE.
      readwv=.FALSE.
      PSTPMN=2.
      ALFLMTT=6.
      ALFLMTX=5000.
      ALFLMTY=500.
      KLIMIT =INT(MP/3)-1
      ERRLMT =1.E+1
      TDUR   =256.
      GTOL=2.
      SI=2.
      HSI=SI/2.
      NSAMPT=1500
      NIMAX=0
      JK=0
      PRINT*
      PRINT*
999   CONTINUE
      JK=JK+1
      CHDIST=.FALSE.
      TERMNT=.FALSE.
      DELION=.FALSE.
      ADDION=.FALSE.
      SINGUL=.FALSE.
      NEGLAG=.FALSE.
      IMPROV=.FALSE.
      FVIOL =.FALSE.
      VIO   =.FALSE.
      KEEPJG=.FALSE.
      bnconv=.FALSE.
      aiconv=.FALSE.
      iginfo=.FALSE.
c
      IF(.NOT.(WWEINR.OR.READWV)) THEN
        READ(10,*) GARBG
        READ(10,*) (WP(I),I=1,9),LWV,LWEST
        READ(10,*) GARBG
      ENDIF
      READ(10,*) GARBG
      READ(10,*) NREC,NI,CSTATE
      IF(NREC.EQ.99999) THEN
c                              ** go to plot results
        WRITE(6,*) ' Trace: Last   Record:',NREC
        IF(perform.EQ.'fl') WRITE(15,*) ' Trace: Last   Record:',NREC
        JK=JK-1
        GO TO 99
      ENDIF
      print*,' Trace:',JK,' Record:',NREC
      IF(perform.EQ.'fl') WRITE(15,*) ' Trace:',JK,' Record:',NREC
c
      NTOTREC(JK)=NREC
      IF(NI.GT.NIMAX) NIMAX=NI
      DO 765 I=1,NI
        READ(10,*) NTXYD,T(I),X(I),Y(I)
765   CONTINUE
      IF(CSTATE) THEN
```

```fortran
C
      SUBROUTINE RESPON(X,Y,Z,NP,NI,W,LWP,LW,SI,RSP,MP,M,RSPT,MWP,
     +          ACOU,REF,NSWIN)
C
C Finds the model response, i.e. synthetic seismogram.
C
      REAL     X(NP),Y(NP),Z(NP),W(LWP),RSP(MP),RSPT(MWP),ACOU(MP),
     +         REF(MP),SI
      INTEGER NI,MI,LW,M,MT,NSWIN
C
C Find impedance log, AC, reflection coefficient series, synthetic.
C All at sampling rate SI, and select valid part.
      CALL IMPLOG(X,Y,Z,NI,NP,ACOU,MI,MP,SI,NSWIN)
      CALL GENRFS(ACOU,MP,MI,REF,LR)
      MT=LR+LW-1
      CALL FOLD(LR,REF,LW,W,MT,RSPT,MP,LWP,MWP)
      M=MI
      DO 10 I=1,M
         RSP(I)=RSPT(LW/2-1+I)
  10  CONTINUE
C
      RETURN
      END
C-----+-------------------------------------------------------------
      SUBROUTINE BNP(P,N,SI,IBFAPP)
C
      INTEGER N,IBFAPP
      REAL    P(N),SI
C
      DO 20 I=1,N
         IF(IBFAPP.GT.0) P(I)=SI*ANINT(P(I)/SI)
         IF(IBFAPP.LT.0) P(I)=SI*AINT(P(I)/SI)
  20  CONTINUE
C
      RETURN
      END
C-----+----------------------------------------------------------
      SUBROUTINE IMPLOG(X,Y,T,NI,NP,AC,IA,MP,SI,NSWIN)
C
      INTEGER NI,MP,IA
      REAL X(NP),Y(NP),T(NP),AC(MP),SI,TA,TP
C
C This subroutine takes X, Y and T (with dimensions N), which describe
C the starting impedance, the impedance gradient and the accumulative
C time thickness, respectively, and produces a new impedance log AC,
C sampled at intervals SI. IA is the initial size of AC taken taken
C to be large enough. On exit AC have a size of IA.
C
      TA=0.
      IA=0
      DO 15 I=1,NI
         TP=0.
         IF((I.EQ.1).AND.(T(1).LT.SI)) GO TO 15
         IF((I.GT.1).AND.((T(I)-T(I-1)).LT.SI)) GO TO 15
  10     IF(TA.GE.T(I)) GO TO 15
         IA=IA+1
         AC(IA)=X(I)+Y(I)*TP
         TA=TA+SI
         TP=TP+SI
         GO TO 10
  15  CONTINUE
      IF(IA.NE.NSWIN) THEN
         PRINT*,' Fatal Error: M and NSWIN not equal.'
         PRINT*,'                           M= ',M
         PRINT*,'                       NSWIN= ',NSWIN
      ENDIF
C
      RETURN
      END
C
C-------------------------------------------------------------
C
      SUBROUTINE JCB(X,Y,T,NI,W,LWP,LWV,SI,RSP,RSPP,RSPM,AC,R,RSPT,MWP,
     +          DJ,MP,NP,M,N,CENTRL,problm,NSWIN)
C
C This subroutine finds the Jacobian matrix.
      REAL X(NP),Y(NP),T(NP),W(LWP),RSP(MP),RSPP(MP),RSPM(MP),RSPT(MWP),
     + DJ(MP,NP),AC(MP),R(MP),SI,TT,XX,YY,DX,DY,DT,EPSLN/3.45266983E-4/
      INTEGER NI,LWV,LWP,M,N,MWP,KX,KY,NSWIN
      LOGICAL problm(2),CENTRL
C
C Loop through columns of DJ, by perturbing a variable in each loop.
      KX=0
      KY=0
      DO 40 J=1,N
         IF(problm(1)) THEN
            IF(CENTRL) THEN
               DT  =SI/2.
               TT  =T(J)
               XX  =X(J+1)
               T(J) =T(J)+DT
               DP  =T(J)-TT
               X(J+1)=X(J+1)+Y(J+1)*DT
               IF((SI*ANINT(T(J)/SI)).EQ.(SI*ANINT(TT/SI))) THEN
                  DO 10 I=1,M
                     RSPP(I)=RSP(I)
  10              CONTINUE
               ELSE
                  CALL RESPON(X,Y,T,NP,NI,W,LWP,LWV,SI,RSPP,MP,M,RSPT,
     +                 MWP,AC,R,NSWIN)
               ENDIF
               T(J) =TT
               X(J+1)=XX
               T(J) =T(J)-DT
               DM  =TT-T(J)
               X(J+1)=X(J+1)-Y(J+1)*DT
               IF((SI*ANINT(T(J)/SI)).EQ.(SI*ANINT(TT/SI))) THEN
                  DO 20 I=1,M
                     RSPM(I)=RSP(I)
  20              CONTINUE
               ELSE
                  CALL RESPON(X,Y,T,NP,NI,W,LWP,LWV,SI,RSPM,MP,M,RSPT,
     +                 MWP,AC,R,NSWIN)
               ENDIF
            ELSE
               DT  =SI
               TT  =T(J)
               XX  =X(J+1)
               T(J) =T(J)+DT
               DP  =T(J)-TT
               X(J+1)=X(J+1)+Y(J+1)*DT
               CALL RESPON(X,Y,T,NP,NI,W,LWP,LWV,SI,RSPP,MP,M,RSPT,MWP,
     +              AC,R,NSWIN)
            ENDIF
         ENDIF
         IF(problm(2)) THEN
            IF(MOD(J,2).GT.0) THEN
               KX=KX+1
               XX=X(KX)
               DX=EPSLN*AMAX1(X(KX),8000.)
**             IF((KX.LT.NI).AND.(XX.LT.X(KX+1))) DX=-DX
               X(KX)=X(KX)+DX
            ELSE
               KY=KY+1
               YY=Y(KY)
               DY=EPSLN*AMAX1(ABS(Y(KY)),1000.)
               Y(KY)=Y(KY)+DY
            ENDIF
            CALL RESPON(X,Y,T,NP,NI,W,LWP,LWV,SI,RSPP,MP,M,RSPT,MWP,AC,R,
     +           NSWIN)
         ENDIF
C
C differentiate and put results in DJ(I,J).
         DO 30 I=1,M
            IF(problm(1)) THEN
               IF(CENTRL) THEN
                  DJ(I,J)=(RSPP(I)-RSPM(I))/(DP+DM)
               ELSE
                  DJ(I,J)=(RSPP(I)-RSP(I))/DP
               ENDIF
            ENDIF
            IF(problm(2)) THEN
               IF(MOD(J,2).GT.0) THEN
                  DJ(I,J)=(RSPP(I)-RSP(I))/(X(KX)-XX)
               ELSE
                  DJ(I,J)=(RSPP(I)-RSP(I))/(Y(KY)-YY)
               ENDIF
            ENDIF
  30     CONTINUE
         IF(problm(1)) THEN
            T(J) =TT
            X(J+1)=XX
         ENDIF
         IF(problm(2)) THEN
            IF(MOD(J,2).GT.0) THEN
               X(KX)=XX
            ELSE
               Y(KY)=YY
            ENDIF
         ENDIF
  40  CONTINUE
C
      RETURN
      END
C----------------------------------------------------------------
C
      SUBROUTINE GENRFS (AC,MP,L,RF,LRF)
C
      REAL  AC(MP),RF(MP)
C
      LRF = L-1
      DO 10 I=1,LRF
  10     RF(I) = (AC(I+1)-AC(I))/(AC(I+1)+AC(I))
C
      RETURN
      END
C----------------------------------------------------------------
C
      SUBROUTINE FORK(LP,LX,CX,SIGNI)
C
      COMPLEX CX(LP),CARG,CW,CTEMP
C
      J=1
      IF(SIGNI.LT.0.) THEN
         SC=1.
      ELSE
         SC=1./FLOAT(LX)
      ENDIF
*     SC=SQRT(1./FLOAT(LX))
C
      DO 30 I=1,LX
         IF(I.GT.J)GO TO 10
         CTEMP=CX(J)*SC
         CX(J)=CX(I)*SC
         CX(I)=CTEMP
  10     M=LX/2
  20     IF(J.LE.M)GO TO 30
         J=J-M
         M=M/2
         IF(M.GE.1)GO TO 20
  30     J=J+M
C
      L=1
  40  ISTEP=2*L
C
      DO 50 M=1,L
         CARG=(0.,0.)*(3.14159265*SIGNI*(M-1))/FLOAT(L)
         CW=CEXP(CARG)
C
         DO 50 I=M,LX,ISTEP
            CTEMP=CW*CX(I+L)
            CX(I+L)=CX(I)-CTEMP
  50        CX(I)=CX(I)+CTEMP
C
      L=ISTEP
      IF(L.LT.LX)GO TO 40
C
      RETURN
      END
C----------------------------------------------------------------
C
      SUBROUTINE FOLD(LA,A,LB,B,LC,C,MPP,LWP,MWP)
C
C Description :
C           This subroutine performs polynomial multiplication,
C           or equivalently convolution, of two input time series.
C           The operation is conducted in the time domain.
C
C           N.B. At the beginning and end of the output trace the
C           number of products per coefficient is adjusted so
C           that no implicit padding of the input time series
C           takes place. The output series is consequently not
C           valid within MAX0(LA,LB) samples from either end.
C Arguments :
C           LA - Length in samples of time series A.
C           LB - Length in samples of time series B.
C           A,B - Arrays containing input time series.
C           LC - Length of output time series, needed for
C                dimensioning purposes. (Equal to LA+LB-1)
C           C - Array containing output time series.
C
C Modification :
C           This subroutine has been modified to include the physical
C           dimensions of the arrays A,B and C. This modification
C           was necessary to use it in the constrained case.
C           The modification date August 9, 1993.
C
      REAL A(MPP),B(LWP),C(MWP)
C
      DO 10 I=1,MWP
         C(I)=0.
  10  CONTINUE
      DO 20 I=1,LA
         DO 20 J=1,LB
            K  =I+J-1
            C(K)=C(K)+A(I)*B(J)
  20  CONTINUE
C
      RETURN
      END
C----------------------------------------------------------------
C
      SUBROUTINE GWV(WP,SI,T,WV,LWP,LWV,LWEST,PH,CWV,AWV,FWV,FVIOL,
     +          LWSHRT)
C
      INTEGER LWP,LWV,LW,LWEST
      REAL AWV(LWP),FWV(LWP),PH(LWP),WP(9),WV(LWP),SI,T,SLPL,SLPM,SLPR
```

```fortran
            CALL TRANSP(DJM,DJMT,MP,NP,M,N)
            CALL PMTVEC(DJMT,DSC1,G,MP,MP,N,M)
            CALL PRDMAT(DJMT,DJM,H,NP,MP,NP,N,M,N)
            DONEA=.TRUE.
         ENDIF
C
C Find HP and GP.
         CALL PRDMAT(ZT,H,HTE,NP,NP,NP,NZ,N,N)
         CALL PRDMAT(HTE,Z,HP,NP,NP,NP,NZ,N,NZ)
         CALL PMTVEC(ZT,G,GP,NP,NP,NZ,N)
C
C Find Cholesky factors CL, CD and CE of HP, magnitudes CEM of CE,
C and GRM of GP.
         CALL CHOLSK(HP,CL,CD,CE,CC,NP,NZ,perform)
*        CALL ZERO2D(CLT,NP,NP,NP,NP)
         CALL TRANSP(CL,CLT,NP,NP,NZ,NZ)
         CEM=SQRT(VNORM(CE,NP,NZ))
         GRM=SQRT(VNORM(GP,NP,NZ))
C
C Test if GRM is large.
         IF(GRM.GT.GTOL) THEN
C
C                            if so find P using PZ, CL and CD.
            DO 52 I=1,NZ
               GPC(I)= -GP(I)
52          CONTINUE
            CALL FBSUBA(CL,TK,GPC,1.,NP,NZ)
            DO 53 I=1,NZ
               CDRCP(I)=TK(I)/CD(I)
53          CONTINUE
            CALL FBSUBA(CLT,PZ,CDRCP,-1.,NP,NZ)
            CALL PMTVEC(Z,PZ,P,NP,NP,N,NZ)
C
         ELSEIF(CEM.GT.0.) THEN
C
C                            Or else modified HP, find negative curvature P.
            CALL SUBVEC(CD,CE,SIGS,NP,NZ)
            CALL MINVAL(SIGS,VAL,ISIG,NP,NZ)
            CALL UNIVEC(ES,ISIG,NP,NZ)
            CALL ZERO1D(PZ,NP,NP)
            CALL FBSUBA(CLT,PZ,ES,-1.,NP,NZ)
C
C Test if the small GRM is positive, so saddle point. Or GRM is zero,
C so a weak minimum.
            IF(GRM.GT.0.) THEN
               SSIGN=PRODIN(PZ,GP,NP,NZ)
               CALL PMTVEC(Z,PZ,P,NP,NP,N,NZ)
               IF(SSIGN.GT.0.) THEN
                  DO 54 I=1,N
                     P(I)=-P(I)
54                CONTINUE
               ENDIF
            ELSE
               CALL PMTVEC(Z,PZ,P,NP,NP,NZ,N)
            ENDIF
         ELSE
C
C CEM is zero, positive definite HP.
C Find Lagrange multipliers,decide on deletion.
            IF(INDA.EQ.0) THEN
               NEGLAG=.FALSE.
               GO TO 56
            ENDIF
            DO 252 I=1,NZ
               GPC(I)= -GP(I)
252         CONTINUE
            CALL FBSUBA(CL,TK,GPC,1.,NP,NZ)
            DO 253 I=1,NZ
               CDRCP(I)=TK(I)/CD(I)
253         CONTINUE
            CALL FBSUBA(CLT,PZ,CDRCP,-1.,NP,NZ)
            CALL PMTVEC(Z,PZ,P,NP,NP,N,NZ)
            CALL PMTVEC(H,P,HPK,NP,NP,N,N)
            DO 255 I=1,N
               HPKG(I)=G(I)+HPK(I)
255         CONTINUE
            CALL PMTVEC(Q1T,HPKG,Q1TG,NP,NP,MA,NA)
            CALL FBSUBB(R,VECLAG,Q1TG,-1.,SINGUL,NP,MA)
            IF(SINGUL) THEN
               WRITE(12,*)'********** Fatal ERROR: Singular R.'
               GO TO 99
            ENDIF
            NEGLAG=.FALSE.
            DO 55 I=1,MA
               IF((VECLAG(I).LT.0.).AND.(INDXA(I).GE.0)) THEN
                  IF(problm(1).AND.(I.GT.NECT)) THEN
                     NEGLAG=.TRUE.
                     GO TO 56
                  ENDIF
                  IF(problm(2).AND.(I.GT.NECXY)) THEN
                     NEGLAG=.TRUE.
                     GO TO 56
                  ENDIF
               ENDIF
55          CONTINUE
56          CONTINUE
            IF(.NOT.NEGLAG) THEN
               TERMNT=.TRUE.
               KEEPJG=.TRUE.
               GO TO 199
            ELSE
               IF(problm(1)) THEN
                  CALL MINVAL(VECLAG(NECT+1),VAL,IDEL,MA,MA-NECT)
                  IDEL=NECT+IDEL
               ENDIF
               IF(problm(2)) THEN
                  CALL MINVAL(VECLAG(NECXY+1),VAL,IDEL,MA,MA-NECXY)
                  IDEL=NECXY+IDEL
               ENDIF
            ENDIF
            CALL DELROW (A,B,INDXA,INDA,IDEL,NP,NP,MA,NA)
            IF(MA.EQ.0) THEN
               MA=NA
               CALL UNIMAT(Q,NP,NA)
               CALL UNIMAT(Z,NP,NA)
               CALL UNIMAT(R,NP,NA)
               CALL ZERO2D(AT,NP,NP,NA,MA)
               CALL TRANSP(Q,QT,NP,NP,NA,NA)
               CALL TRANSP(Z,ZT,NP,NP,NA,NA)
               NZ=NA
               CALL ZERO2D(Q1,NP,NP,NA,NA)
            ELSE
               CALL ZERO2D(AT,NP,NP,NA,NA)
               CALL TRANSP(A,AT,NP,NP,MA,NA)
               CALL QR(AT,QT,R,SAVEQ,NP,NP,NA,NA)
               CALL TRANSP(QT,Q,NP,NP,NA,NA)
               CALL PARTQ(Q,Q1,Z,NP,NA,INDA,NZ)
               CALL TRANSP(Q1,Q1T,NP,NP,NA,MA)
               CALL TRANSP(Z,ZT,NP,NP,NA,NZ)
            ENDIF
         ENDIF
C
C set logical variable DELION.
         DELION=.TRUE.
      ENDIF
C
C Stage 2. Deciding the step length.
      IF(DELION) THEN
         ALF=0.
         KEEPJG=.TRUE.
```

```fortran
      ELSE
         IF(INDB.EQ.0) THEN
            CHDIST=.FALSE.
            GO TO 64
         ENDIF
      ENDIF
      IF(problm(1)) THEN
         CALL ABSMXV(P,PMAXV,IPMX,NP,N)
         IF((ABS(PMAXV).GE.(.9*HSI)).AND.(ABS(PMAXV).LT.PSTPMN)) THEN
            DO 167 I=1,N
               P(I)=P(I)*PSTPMN/ABS(PMAXV)
167         CONTINUE
         ENDIF
         CALL BNP(P,N,SI,1)
         CALL FINDMX(AC1,BC1,INDXA,INDA,INDXB,NECT,X,Y,T,SI,P,
     +              DIST,CHDIST,DMAX,IADD,MC1P,NP,NA,NI,problm)
      ENDIF
      IF(problm(2)) CALL FINDMX(AC2,BC2,INDXA,INDA,INDXB,INDB,NECXY,
     +              X,Y,T,SI,P,DIST,CHDIST,DMAX,IADD,MC2P,NP,NA,NI,problm)
64    CONTINUE
      PNORM1=0.
      PNORM2=0.
      KX=0
      KY=0
      DO 177 I=1,N
         IF(problm(2)) THEN
            IF(MOD(I,2).GT.0) THEN
               KX=KX+1
               PNORM1=PNORM1+ABS(P(I))
            ELSE
               KY=KY+1
               PNORM2=PNORM2+ABS(P(I))
            ENDIF
         ELSE
            PNORM1=PNORM1+ABS(P(I))
         ENDIF
177   CONTINUE
      PCHNGED=.FALSE.
      IF(problm(1)) THEN
         IF(PNORM1.LE.1.E-7) THEN
            ALFMAX=ALFLMTT
         ELSE
            ALFMAX=AMIN1(ALFLMTT,ALFLMTT/PNORM1)
         ENDIF
         IF(DMAX.GT.0.) THEN
            DO 783 I=1,N
               IF((SI*((AINT(DMAX*P(I)/SI))/DMAX)).NE.P(I)) THEN
                  PCHNGED=.TRUE.
                  GO TO 784
               ENDIF
783         CONTINUE
         ENDIF
784      CONTINUE
      ENDIF
      IF(problm(2)) THEN
         IF((PNORM1.LE.1.E-7).AND.(PNORM2.LE.1.E-7)) THEN
            ALFMAX=AMIN1(ALFLMTX,ALFLMTY)
         ELSEIF(PNORM1.LE.1.E-7) THEN
            ALFMAX=AMIN1(ALFLMTX,ALFLMTY/PNORM2)
         ELSEIF(PNORM2.LE.1.E-7) THEN
            ALFMAX=AMIN1(ALFLMTX/PNORM1,ALFLMTY)
         ELSE
            ALFMAX=AMIN1(ALFLMTX/PNORM1,ALFLMTY/PNORM2)
         ENDIF
      ENDIF
      IF(CHDIST.AND.(DMAX.EQ.0.)) THEN
         ALF=DMAX
C Add IADD constraint. Keep the point, G, H and ER.
         IF(INDA.EQ.0) MA=INDA
         IF(problm(1)) CALL ADDROW(A,B,AC1,BC1,INDXA,INDXB(IADD),MC1P,
     +                 NP,MA,NA,MAC1)
         IF(problm(2)) CALL ADDROW(A,B,AC2,BC2,INDXA,INDXB(IADD),MC2P,
     +                 NP,MA,NA,MAC2)
C
C If add nonexistent constraint to A, or delete from empty INDXB exit.
         IF(problm(1).AND.(MA.GT.MAC1)) GO TO 99
         IF(problm(2).AND.(MA.GT.MAC2)) GO TO 99
         IF(INDB.LT.0) GO TO 99
         INDA=INDA+1
         CALL TRANSP(A,AT,NP,NP,MA,NA)
         CALL QR(AT,QT,R,SAVEQ,NP,NP,NA,MA)
         CALL TRANSP(QT,Q,NP,NP,NA,NA)
         CALL PARTQ(Q,Q1,Z,NP,NA,INDA,NZ)
         CALL TRANSP(Q1,Q1T,NP,NP,NA,MA)
         CALL TRANSP(Z,ZT,NP,NP,NA,NZ)
         ADDION=.TRUE.
         KEEPJG=.TRUE.
C
      ELSEIF(CHDIST.AND.(DMAX.LE.1.).AND.(DMAX.LE.ALFMAX)
     +       .AND.(.NOT.PCHNGED)) THEN
         ALF=DMAX
C Move step ALF, find ERT, if ERT<ER(K) and PGP1<0 Add IADD, else do LS.
         IF(problm(1)) THEN
            DO 780 I=1,N
               P(I)=ALF*P(I)
780         CONTINUE
            CALL BNP(P,N,SI,-1)
            DO 66 I=1,N
               TN(I)=T(I)+P(I)
66          CONTINUE
         ENDIF
         IF(problm(2)) THEN
            KX=0
            KY=0
            DO 65 I=1,N
               IF(MOD(I,2).GT.0) THEN
                  KX=KX+1
                  XN(KX)=X(KX)+ALF*P(I)
               ELSE
                  KY=KY+1
                  YN(KY)=Y(KY)+ALF*P(I)
               ENDIF
65          CONTINUE
         ENDIF
         IF(problm(1)) CALL RESPON(X,Y,TN,NP,NI,WV,LWP,LWV,SI,RSPI2,
     +                 MP,M,RSPT,MWP,ACOU,REF,NSWIN)
         IF(problm(2)) CALL RESPON(XN,YN,T,NP,NI,WV,LWP,LWV,SI,RSPI2,
     +                 MP,M,RSPT,MWP,ACOU,REF,NSWIN)
         CALL SUBVEC(RSPI2,RSPO,DSC2,MP,M)
         ERT=VNORM(DSC2,MP,M)/2.
         IF(problm(1)) CALL JCB(X,Y,TN,NI,WV,LWP,LWV,SI,RSPI2,RSPP,
     +                 RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,M,N,CENTRL,problm,NSWIN)
         IF(problm(2)) CALL JCB(XN,YN,T,NI,WV,LWP,LWV,SI,RSPI2,RSPP,
     +                 RSPM,ACOU,REF,RSPT,MWP,DJM2,MP,NP,M,N,CENTRL,problm,NSWIN)
         CALL TRANSP(DJM2,DJM2T,MP,NP,M,N)
         CALL PMTVEC(DJM2T,DSC2,GP1,MP,MP,N,M)
         PGP1=PRODIN(P,GP1,NP,N)
         PMAG=SQRT(VNORM(P,NP,N))
         IF((ABS(ERT-ER(K)).LE.ERRLMT).AND.((PMAG).LE.ERRLMT)) THEN
            K=K+1
            ER(K)=ERT
            KEEPJG=.TRUE.
         ELSEIF(ERT.LT.ER(K)) THEN
            IF(PGP1.LT.0.) THEN
               K=K+1
               ER(K)=ERT
C add IADD.
               IF(INDA.EQ.0) MA=INDA
               IF(problm(1)) CALL ADDROW(A,B,AC1,BC1,INDXA,INDXB(IADD),
     +                       MC1P,NP,MA,NA,MAC1)
```