Notes for Geoacoustic_TDFD

Ralph Stephen and Tom Bolmer
Woods Hole Oceanographic Institution

September 16, 2004
Revised: October 20, 2004

1. Introduction

These notes were written to help users run the WHOI TDFD (Time Domain Finite Difference) elastic wave equation code that was prepared for distribution through the SAIC Ocean Acoustics Library. The code and documentation are based on materials that were developed for a Numerical Wave Propagation class given at MIT in the Fall of 2000. The code used is the full two-dimensional time-domain finite-difference code developed at WHOI over the past 25 years, but in order to reduce the number of variables to a manageable size, we consider a two dimensional, isotropic problem with fixed parameters in time and space. For example, the source waveform in time for both beam and point sources is a Ricker wavelet, time units have been normalized to periods (defined at the peak frequency for pressure in water), space units have been normalized to water wavelengths (defined at the peak frequency of pressure in water - with compressional sound speed and density of 1.5km/sec and 1000kg/m^3) and the domain size has been fixed at 72 x 12 water wavelengths.

2. Background

The class approached the problem in three stages as described in Problem Sets #1, #3 and #5 (Appendices A,B, and C respectively). The first two stages are not necessary to just run the TDFD code, and they can be skipped for now. They could be useful in the future when the time comes to apply the code to a specific physical problem or to understand the details of the code.

In the first stage we studied the elastic wave equation for heterogeneous, isotropic media and the Ricker wavelet in the time and space domains (Appendix A). The code solves the wave equation in Equations 12, 13 and 14 of Appendix A -Part 1: Problem 1. The objective of Problem 2 is to understand the differences in wavelet shape and frequency content between displacement potential, displacement, velocity, and pressure. There is a lot of tedious algebra that is useful for completeness. There is a matlab code at the end of Appendix A - Part 2 called PS1.m that computes and displays time series and spectra for the various waveforms.

In the second stage we wrote a matlab code called tdfd_grid.m for defining physical models of interest (Appendix B). This is where we constrain the size of the model, the frequency of the source, and the minimum and maximum velocities. The parameters were selected to give reasonable stability, accuracy and run time. Running tdfd_grid.m generates Vp, Vs and density files and plots for each of the three models: a flat seafloor (model names ras01 and ras04), a sinusoidal seafloor (ras02 and ras05), and a tunnel beneath a flat seafloor (ras03 and ras06). [For each of the three models we used both a beam source (ras01, ras02, and ras03) and a point source (ras04, ras05 and ras06).]

The third stage is to run the TDFD code itself and to display the output. This is described in Appendix C.

After the class work (a fourth stage) we studied a quarter-space model (ras07 and ras08) to demonstrate different time series output strategies. These are discussed in Section 7.

We have assembled a directory of our TDFD code that compiles and runs our on our Sun workstations. It runs the three models defined in Appendix 2 and it contains batch files (.bch) for compiling the code on other UNIX machines. The snapshot and time series plotting codes (in matlab) are also included in a separate directory.

3. The Software Package

The software package for Geoacoustic_TDFD consists of three parts: a documentation directory and two code directories: TDFD and Plotting. The documentation directory includes these notes, notes for the plotting software, and the matlab m-files PS1.m and tdfd_grid.m that are described in Appendices A and B. The "TDFD" code directory consists of three sub-directories: PS1, PS3 and PS5 corresponding to the Problem Sets described above. PS5 includes nine sub-directories: the tdfd code itself in whoi_nsc and eight model directories (ras01 to ras08) which are discussed further below. The "Plotting" code directory includes the matlab code to generate snapshot and time series output in the same format as the figures in this report.

Three models each are given for point sources (ras04, ras05, ras06) and for beams at 15degree grazing angle (ras01, ras02, ras03). In each set the earth models correspond to a flat seafloor, a sinusoidal seafloor and a tunnel beneath a flat seafloor.

The code for compiling and running the **beam** software on a Sun Solaris workstation is in ras01/ras01.bch_long. Once the executables are created and ras01 runs successfully, ras02 and ras03 can be run without compiling new code (*.bch_short for example). There are four steps in the beam code: a preprocessor which does some quality control and defines array sizes for a given set of parameters (*.par) (bfprepa.f), a code to define the grid based on the matlab output from PS3 (bnymit.f), a code to generate an intermediate file for the beam (the beam is introduced to the grid as a time-dependent boundary condition)(fort.50)(bfsors.f) and the actual tdtd code (bfdif3a.f and bfdif2a.f). There is also a package of subroutines used in all four steps (bfsuba.f). We call the executable tdfd_beam. In the UNIX sh shell, the command for running a *.bch file, sending the run log to a text file and putting the job in the background is:

sh ras01.bch_long > & ras01.t1 &

Each of the four steps (bfprepa, bnymit, bfsors and bfdif3a) generate separate log files (*.LG1, *.LG2, *.LG3 and *.LG4, respectively).

The code for compiling and running the point source software is in ras04/ras04.bch_long. It is similar to the beam code except that there is no source step. We call this executable tdfd_point.

4. The Test Models

Results of the initial six test models are shown in Figures 1 to 6. These figures were generated using the plotting package plot_findif_1. Plotting results in the same format as this report helps tremendously in debugging. Time series plotting is discussed in Section 7 in the context of the quarter plane/step model.

5. Stability and Accuracy of TDFD Results

By editing the matlab file, **tdfd_grid.m**, one can define other seafloor and sub-seafloor models in order to study the bottom interaction for pulse point sources and beams (at 15degrees grazing angle). You can use any velocities you like for compressional waves in the upper medium and for compressional and shear waves in the lower medium with the following constraints:

1) The Courant stability condition requires that delt be less than [delr / (root(2) * vpmax)] where delt is the time increment, delr is the space increment, and vpmax is the largest velocity on the grid. In the examples, delt is 0.001sec and delr is 10m so vpmax is about 7,071m/s. This cannot be exceeded even locally.

2) For acceptable grid dispersion you need at least 10 points per wavelength at the slowest velocity on the grid and at the highest frequency of interest. In the examples, the slowest body wave velocity is water - 1500m/s. Since the peak frequency in pressure is 10Hz let's assume that the highest frequency of interest is 15Hz. At 15Hz the wavelength in water is 100m and since delx is 10m we have 10 points per wavelength. The interface (Stoneley) waves are a little slower than the water velocity, so they may suffer some significant grid dispersion. The effects of grid dispersion get worse with range, so if the interface waves don't travel very far, it is probably OK to under-sample them. Grid dispersion is not a catastrophic issue - the code still runs but the answers will be suspect. From tests, it appears to be OK to undersample locally. For example in small regions of low shear velocity - if the shear waves don't travel very far - you will not notice the grid dispersion. You don't want to sample at less than the Nyquist in any case - at 15Hz and a 10m grid interval you don't want to use a (shear) velocity less than 300m/s.

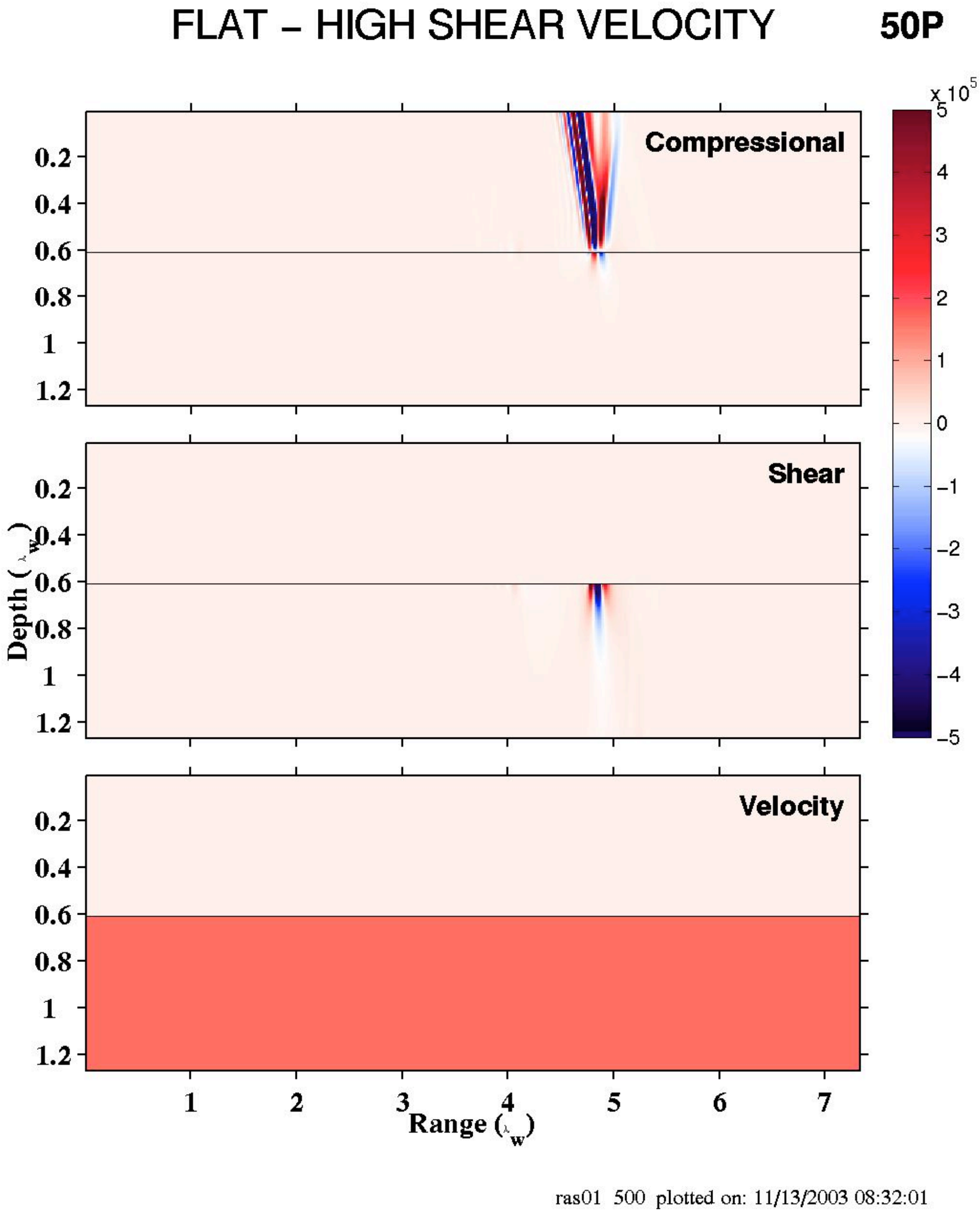

Figure 1: This figure shows a Gaussian beam incident on a flat seafloor at 15degrees grazing angle (sub-critical grazing angle for both compressional and shear waves). The direct and reflected beams can be seen above the seafloor in the compressional frame. The evanescent decay of the wave field beneath the seafloor can be seen in both the compressional and shear frames. Note that Figures 1 through 3 were plotted with a maximum amplitude set at +/-500000.

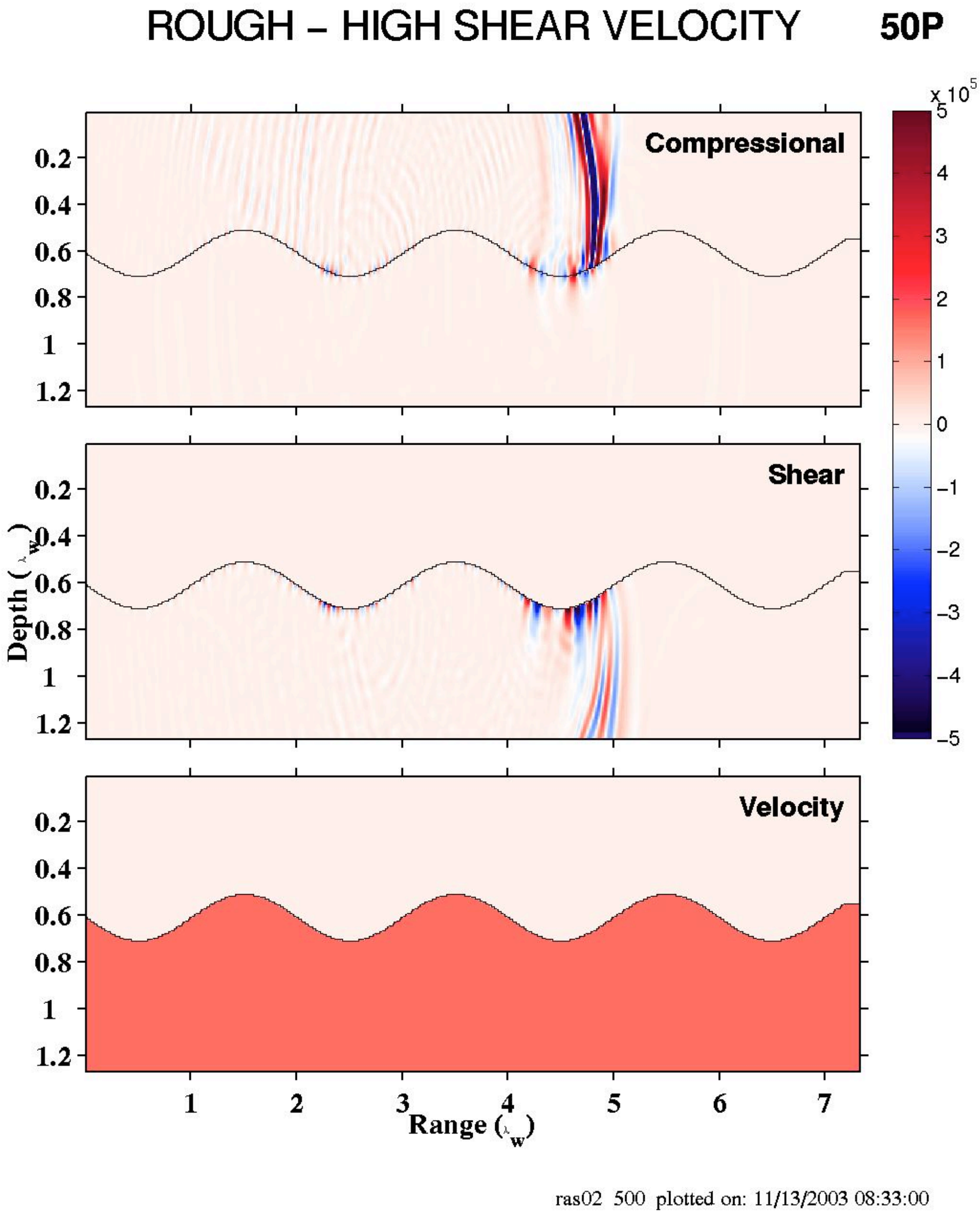

Figure 2: This figure shows the same beam as in Figure 1 but incident on a sinusoidal seafloor. Since the local grazing angle changes there are sections of the seafloor where transmitted (converted) shear body waves are generated in the bottom. The curvature of the seafloor as well as the stair step approximation to the rough seafloor also generates small diffracted body and interface waves.

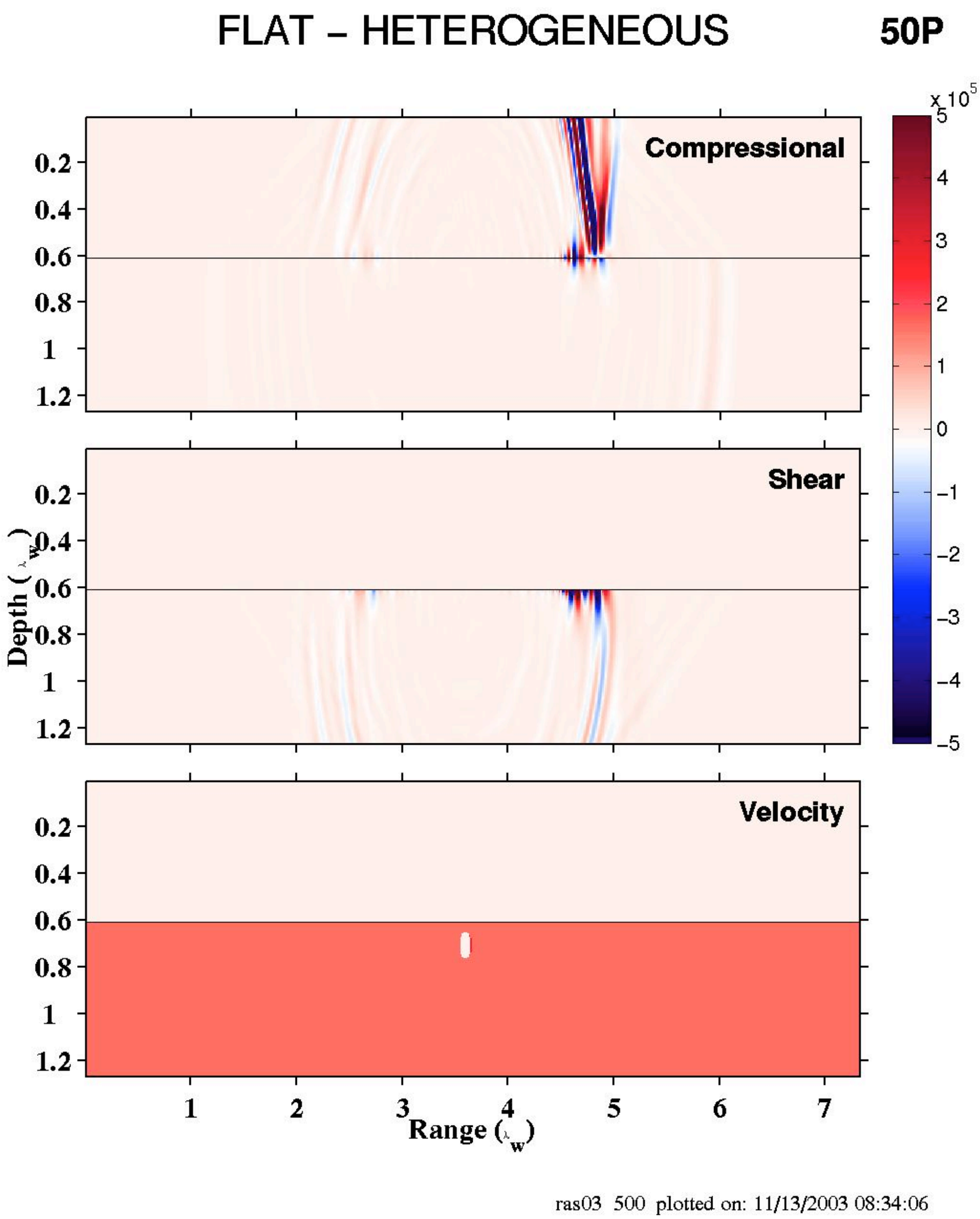

Figure 3: This is the same model and beam source as in Figure 1 except that a "tunnel" has been added beneath the seafloor as shown in the lower panel. The interaction of the compressional and shear evanescent components creates diffractions from the tunnel which are analagous to a point source.

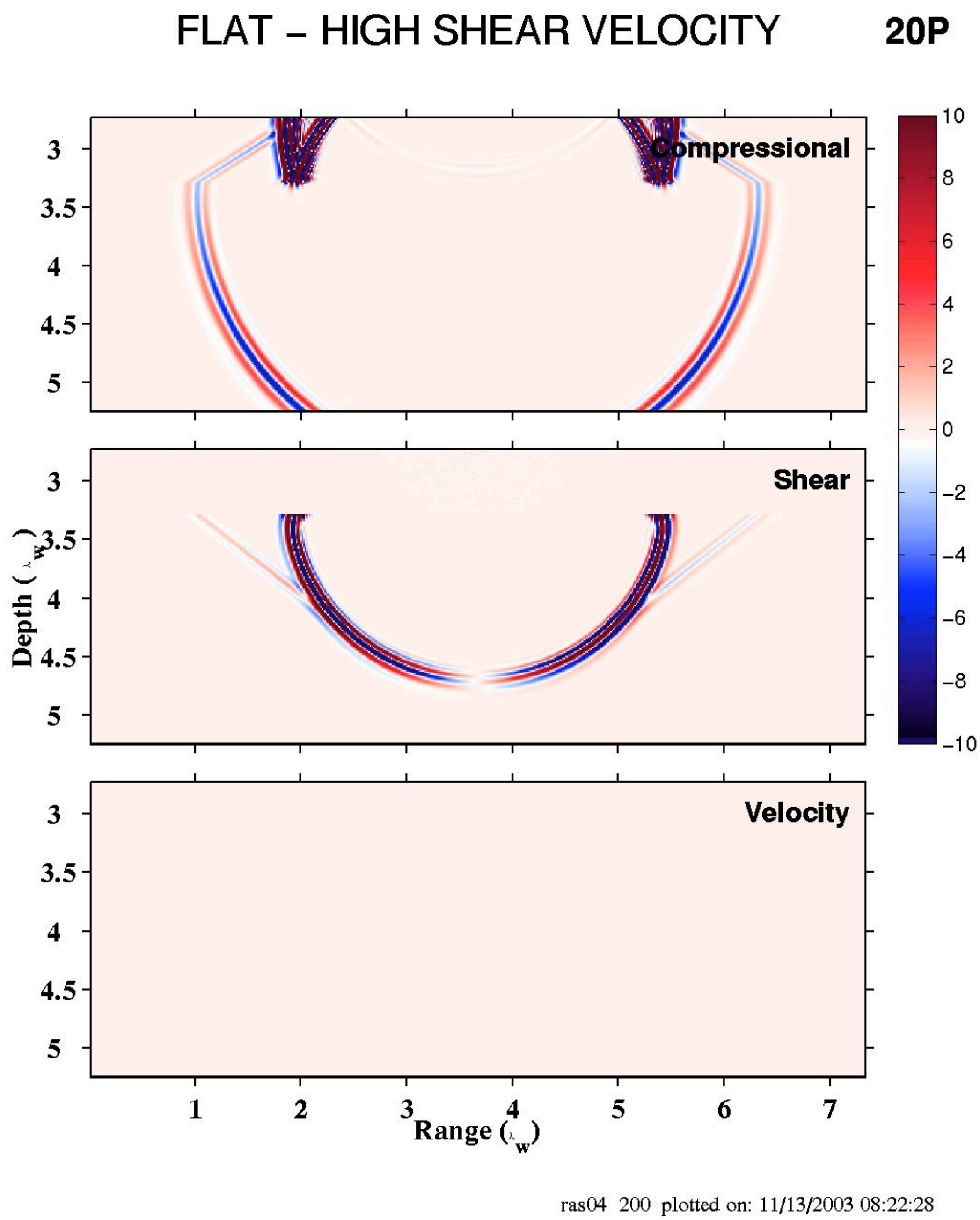

Figure 4: This is the same model as Figure 1 but a point source in the water has been used instead of a Gaussian beam. The compressional direct, reflected, transmitted and head waves can be seen in the top frame. The converted shear and head waves (P1P2S2) can be seen in the middle frame. There is something wrong in the plotting of the velocity field in the lower frame. These frames for Figures 4, 5 and 6 should be identical to the frames in Figures 1, 2 and 3 respectively. Figures 4 through 6 were plotted with a maximum amplitude set at +/-10.

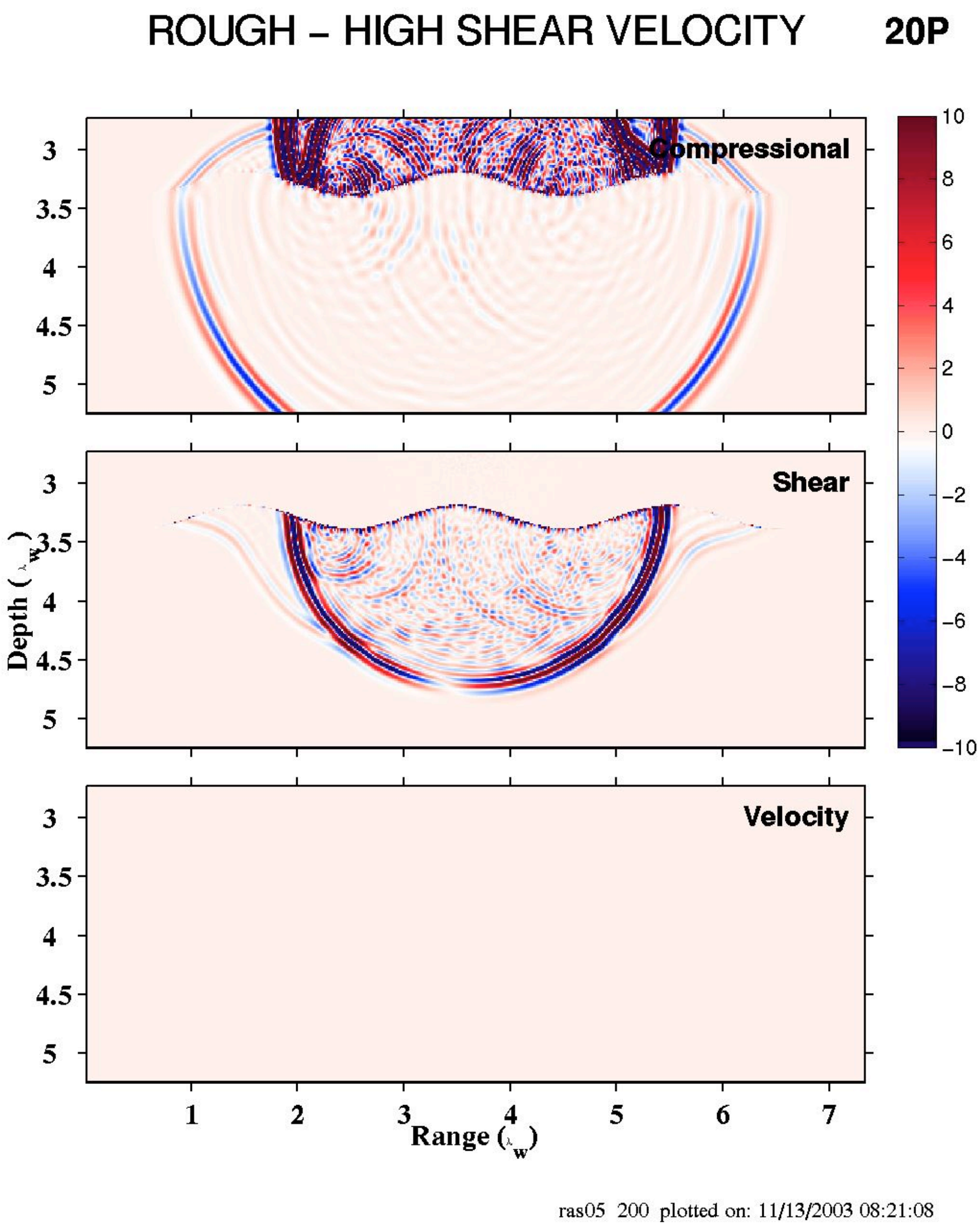

Figure 5: This shapshot corresponds to a point source over a sinusoidal seafloor. The nominal wavefronts from the flat seafloor are distorted; there are real diffractions from the curvature of the seafloor; and there is "noise" from scattering from the stair steps.

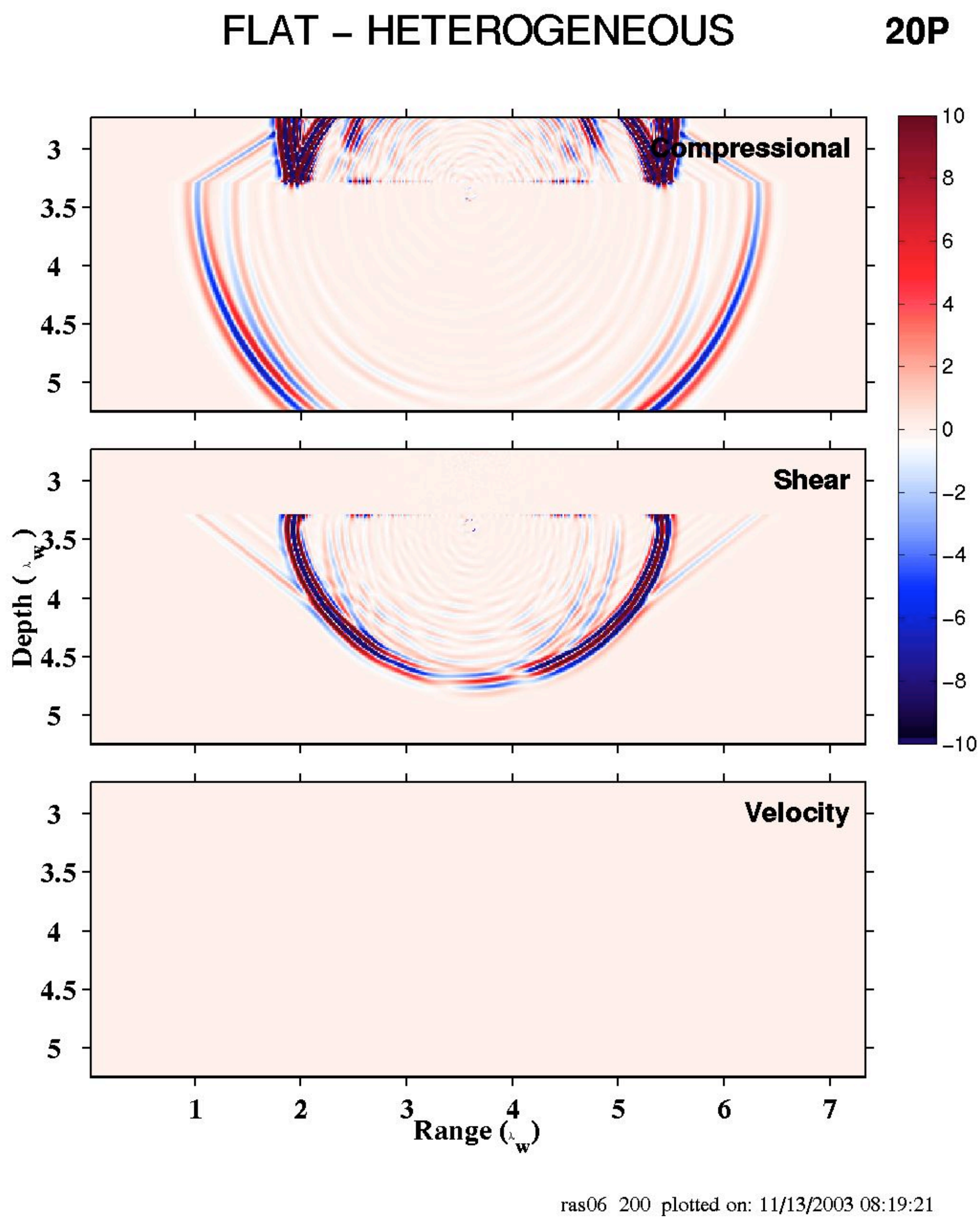

Figure 6: The effect of the tunnel on the response of a point source on a flat sea floor can be seen by comparing this to Figure 4.

So the parameters in the six examples were chosen to model water/rock interaction where the shear velocity in the rock is greater than the water velocity and the compressional velocity in the rock does not exceed 7km/s. The code itself works fine for softer materials (but with different parameters). Models have been run with shear velocities of 140m/s. Note that at 150m/s and 15Hz (the upper frequency of interest from above) the wavelength is 10m and to get 10 points per shear wavelength requires a delx of 1m - so at the nominal frequency of 10Hz you have 150 points per water wavelength! These become large models but the code does work and gives reasonable results.

6. More Detail on the Code and Parameters

For learning and testing it is best to stick with the models ras01 through ras06 which do not require changes to the parameter files. The snapshot and time series plotting codes (in matlab) are also included. It is convenient to look at results in the same format as this report. When you are familiar with these examples, you can start making changes to the parameter file. An annotated .par file is included in Appendix D. This explains all of the parameters and flags, some options, the code layout, and the definitions for the snapshot and time series outputs. Snapshots are the best format for debugging and for understanding the physics. Time series are useful for interpreting field and lab data. Examples of time series are shown below for the quarter-space problem.

7. A Quarter Space Model and Time Series Output

The problem of a point source field incident on a quarter space demonstrates diffraction effects and has analytical solutions that can be compared with the numerical ones. Because of the way that absorbing boundaries are treated in this code the strict "quarter space" problem is difficult to do, but we can model a "big step".

The matlab code to generate the "big step" model is called tdfd_grid_qspace_RAS and is given in Appendix E. The graphical output is given below in Figure 7. This code puts a step in the middle of the usual 12x72 lambda box. The left side is all water except for the bottom two rows of basalt. (This is necessary because of the way the absorbing boundaries work.) The right side is water for the first two wavelengths in depth and then basalt for the rest. The point in the water just at the apex of the step is at (540,31) in transition zone indexes. This should be at (540,33) in the physical domain. (See Appendix D for a discussion of the transition zone and physical domain indexes.)

This model is called ras07. An example of the par file is given in Appendix F. The point source is at (525,18), one wavelength to the left and above the apex of the step. The time series receiver locations and flag have been changed to give two rows of receivers at depths of 18 and 48, one wavelength above and below the top of the step.

The ordinates in Figure 7 run from 1 to 181 from bottom to top. This is a consequence of the matlab plotting routine and needs to be fixed. The actual indices run from the top to the bottom. The units here are index numbers. Since the .par file is set-up to run a 10Hz source (the peak frequency of the pressure pulse), the water wavelength is 150m. Since delx is 10m there are 15 points per wavelength. Some plots are given in indices and some are given in wavelengths. To convert just divide or multiply by 15. Figure 7 is in transition zone coordinates. Once this is read into the TDFD code the range indices stay the same but the depth indices in the physical domain are two points larger than the depth indices in the transition zone.

Figures 8 and 9 show snapshots at 10 and 20 Periods respectively. Since a period at 10Hz is 100msec these correspond to 1 and 2 seconds. The bottom panel in each case shows the (compressional) velocity field. The units here are wavelengths (in the physical domain). The point source is one wavelength up and one wavelength to the left of the upper corner of the step. These wavefront figures show the evolution with time of the direct, reflected, transmitted and head waves as well as all of the diffractions, including interface waves, from the corner. Kinematically at least all of these wavefronts make sense physically.

Figure 10 shows the pressure time series (TSP, the compressional energy density) for a row of receivers one wavelength above the upper step. (This is the same depth as the point source, where there is a singularity.) The ranges correspond to the same ranges as in the snapshots. TSP is the pressure time series, so there are pulses in the time series at every event in the "compressional" snapshot. One can compute velocities for the major arrivals to confirm that they correspond to direct waves, compressional and shear head waves, etc.

Figure 11 shows the compressional energy density for a row of receivers one wavelength below the upper step. In water, to the left of the step, the compressional energy density is the pressure. Figure 12 shows the time series of the shear energy density (TSS). Since this is zero in the water only traces for the right side of the lower row of receivers are shown.

Note that the first four time series in the TSP and TSS files are "dummy" series. They exist because sometimes we use them to generate vertical columns of receivers. You don't need to deal with that right now.

The time series up to about 2 sec are the same for this step model as for a quarter plane. The cluster of energy starting at about 2 seconds for the middle traces are the effects of reflections from the left side of the step and diffractions from the lower corner. If necessary the step could be made deeper to move these events later in time.

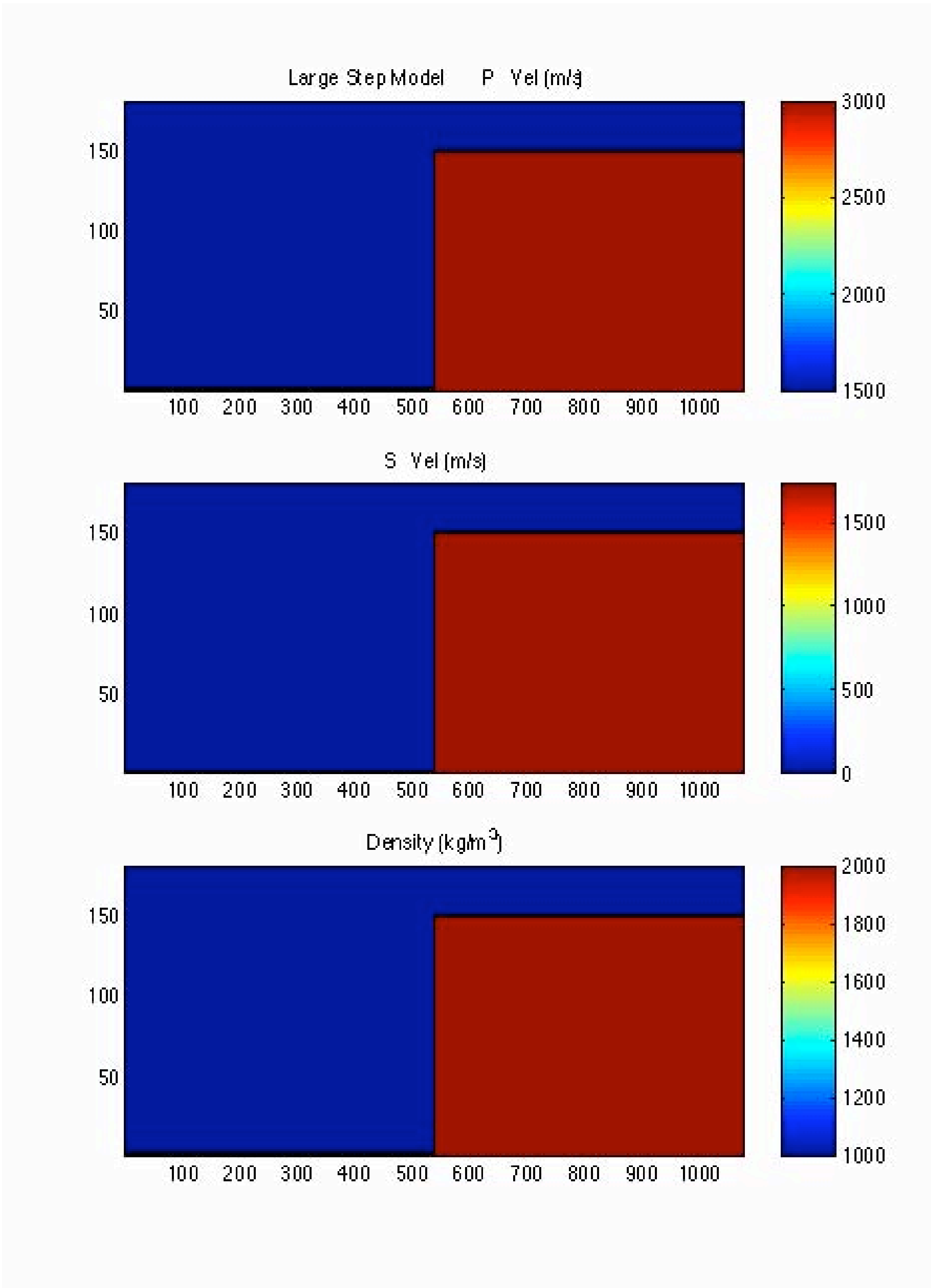

Figure 7: Graphical output of tdfd_grid_qspace_RAS.m showing the distribution of compressional sound speed, shear sound speed and density for the quarter plane model.

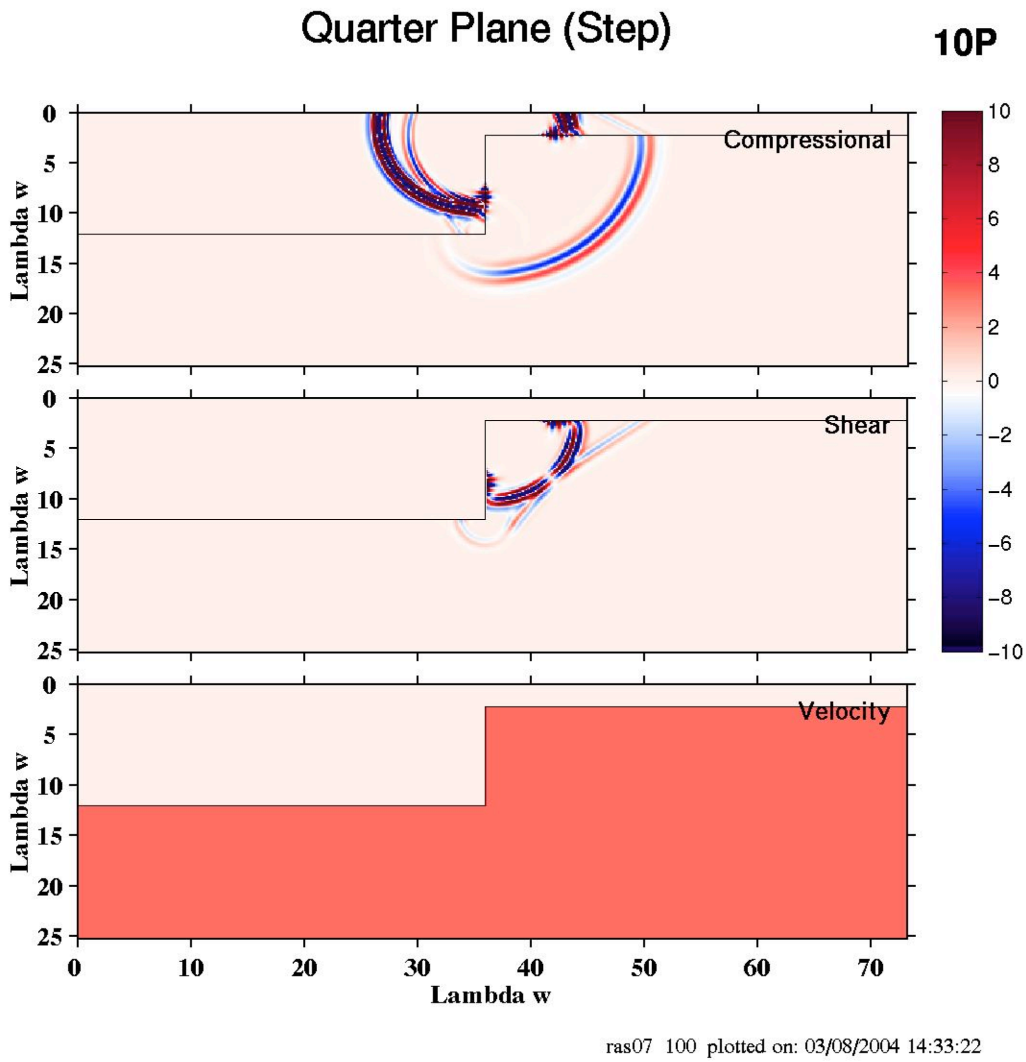

Figure 8: Snapshot of the quarter plane (step) solution at 10 periods (1 sec for a pulse with a peak frequency of 10Hz). The point source is located one wavelength above and to the left of the upper corner. In addition to the direct wave from the point source, diffracted, transmitted, reflected, interface and head waves at both the vertical and upper horizontal interfaces can be observed. Until the energy hits the bottom corner (just before 1sec) the solution to the quarter plane and step models is the same.

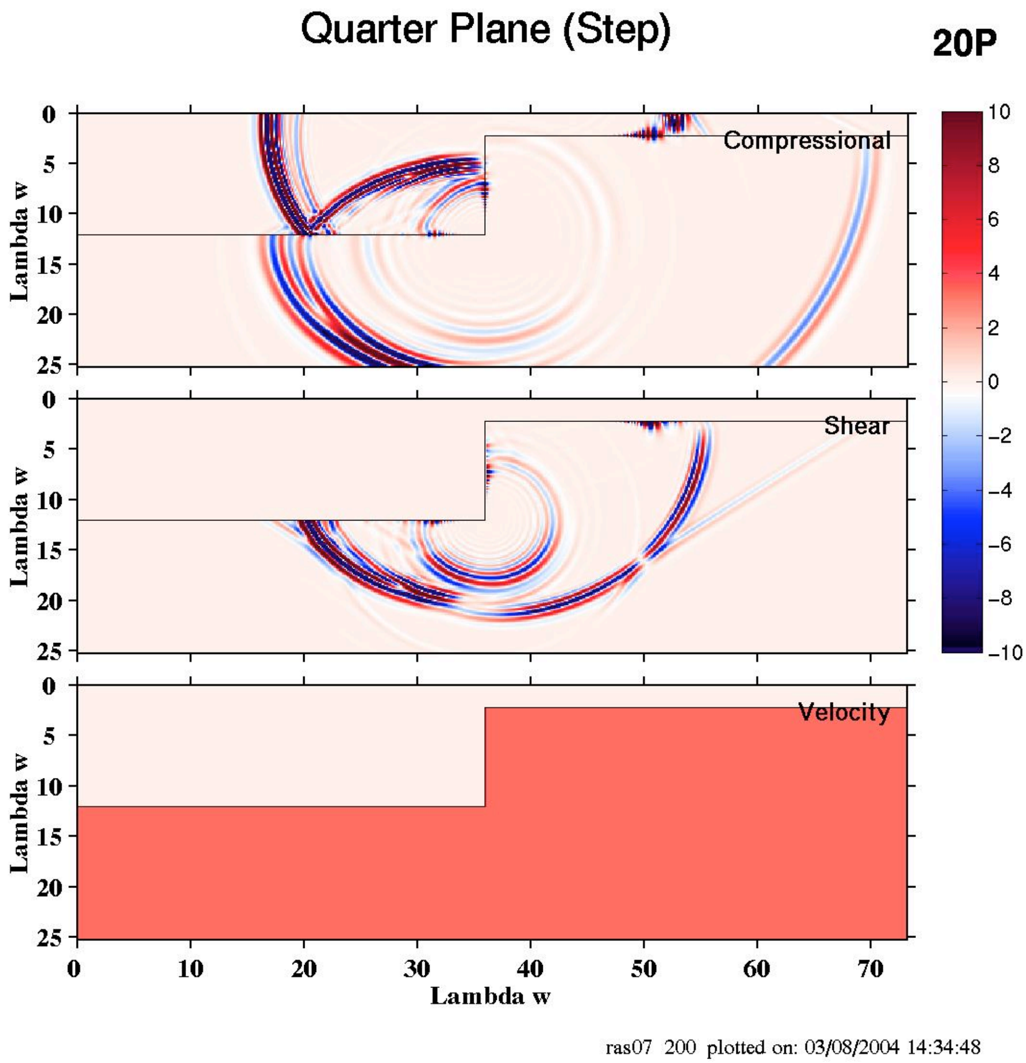

Figure 9: Snapshot of the quarter plane solution at 20 periods (2 sec for a pulse with a peak frequency of 10Hz). In addition to the wave types in Figure 8, diffracted, transmitted, reflected, interface and head waves from the lower interface and corner can be observed.

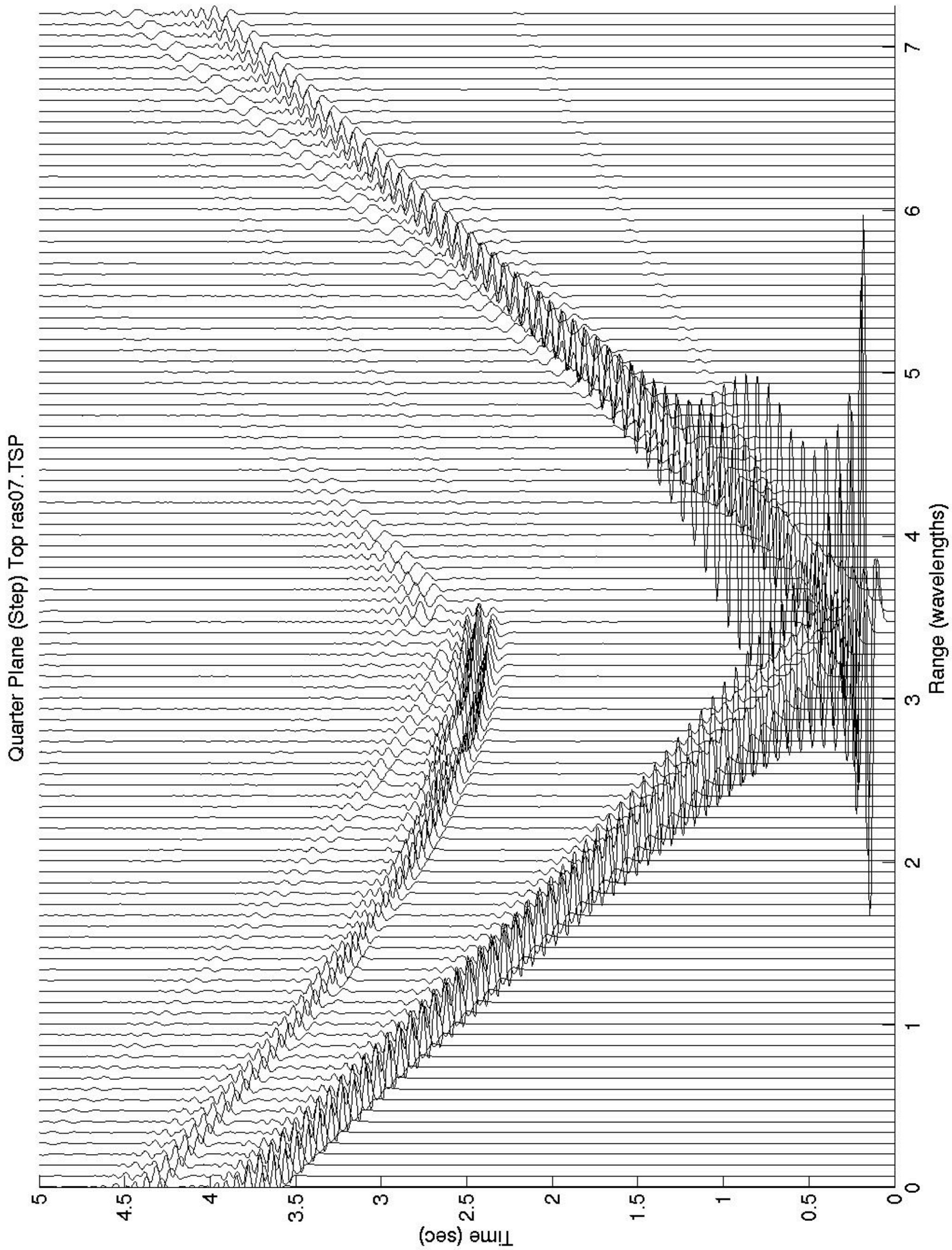

Figure 10: Pressure time series for a row of receivers one wavelength above the upper step. Range is in 10's of wavelengths. Total range is 72 wavelengths. To convert to meters multiply by 150.

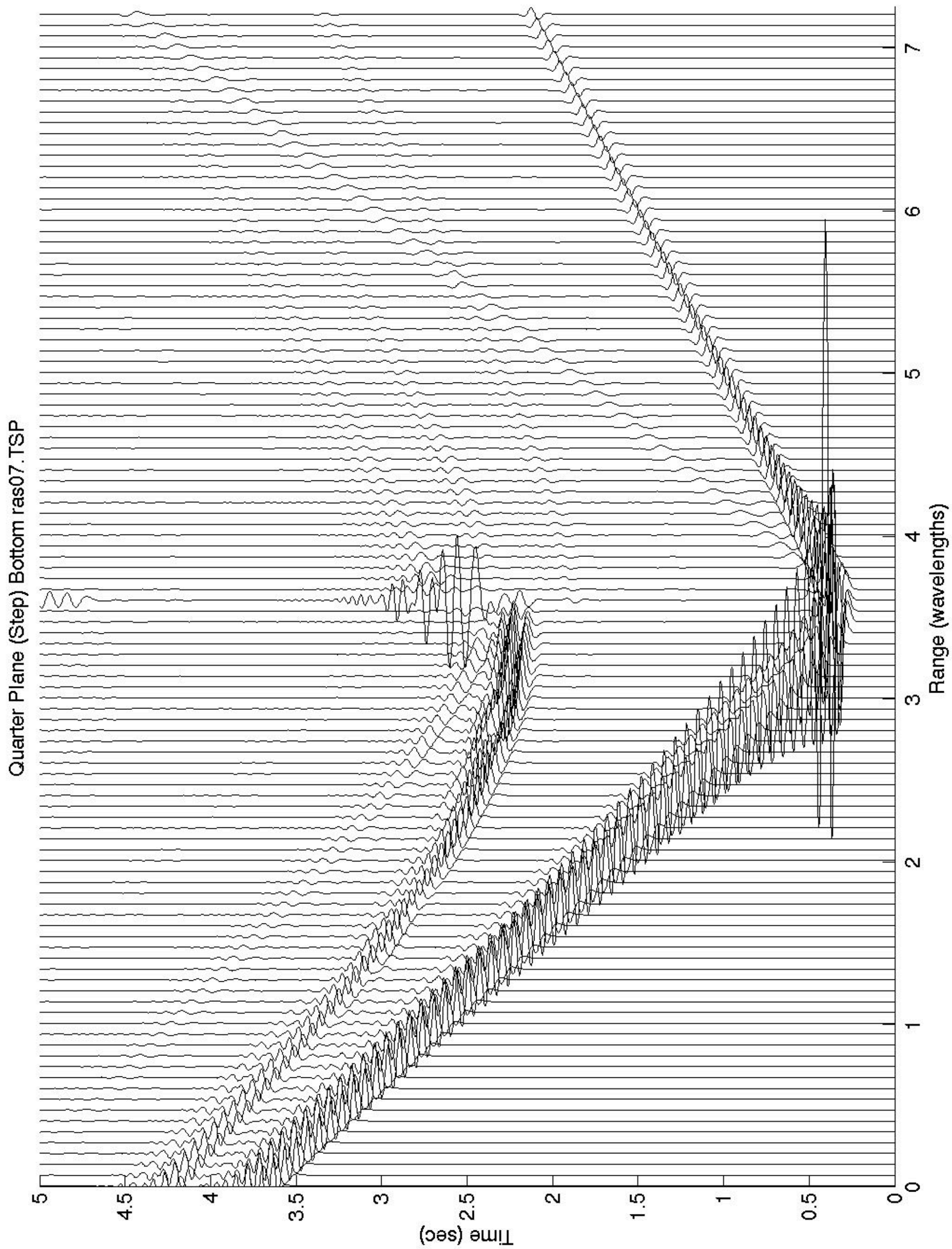

Figure 11: The compressional energy density for a row of receivers one wavelength below the upper step. In water, to the left of the step, the compressional energy density is the pressure.

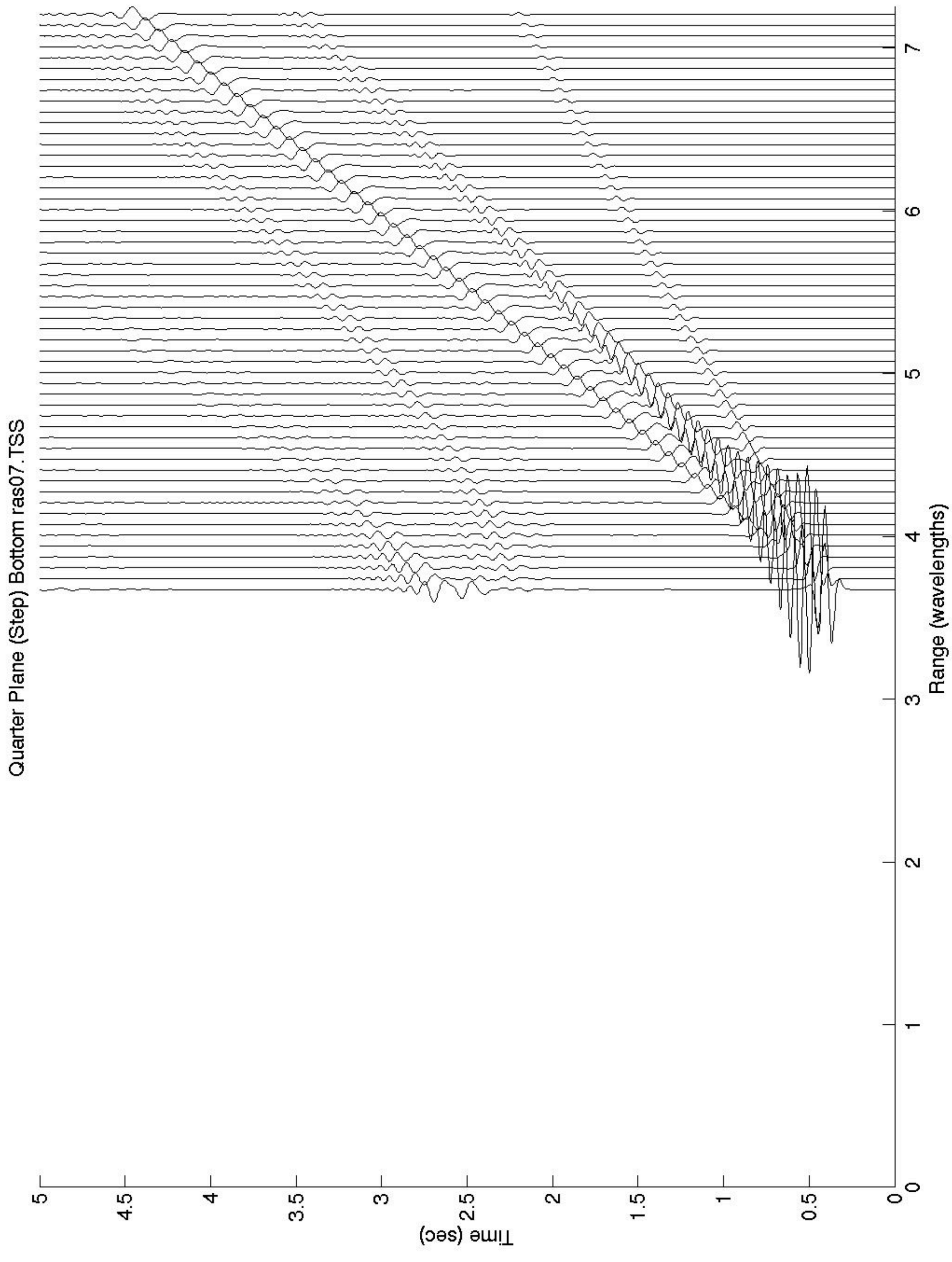

Figure 12: The shear energy density for a row of receivers one wavelength below the upper step. Since this is zero in the water only traces for the right side of the lower row of receivers are shown.

8. Displacement Time Series

In the quarter space model, ras07, time series were output as normalized divergence (pressure) and curl of the displacement field. Since seismometers record vertical and horizontal velocities or accelerations, we have added an option to the code (IVERT=3) to output pressure when the shear modulus at the receiver is zero and to output horizontal and vertical displacement when the shear modulus at the receiver is non-zero. These notes present the new time series results. The modified parameter file, ras08.par, is given in Appendix G.

In the earlier example, the code output two time series files (IVERT=2). *.TSP contained the normalized divergence of displacement at the receiver. When the shear modulus is zero this is the same as pressure. *.TSS contained the curl of the displacement. When the shear modulus is zero this is zero. (Note that in the time series summary at the end of the *.LG4 file, traces that are identically zero are not listed.) In this example (IVERT=3) we have kept the same file names and when the shear modulus at the receiver is zero we output the same pressure (*.TSP) and zero (*.TSS) traces as before. When the shear modulus is non-zero at the receiver, however, we output horizontal displacement in *.TSP and vertical displacement in *.TSS. The user can convert displacement to velocity or acceleration in post-processing.

The model and snapshot output from ras08 are the same as for ras07. Figures 13 through 16 show the new time series results. Details are discussed in the captions.

9. A Note on Units

Pressure is the compressibility times the divergence of displacement (see Appendix E of Stephen et al, 1985. Geophysics, v50, p1588-1609). If the shear modulus is zero the compressibility is the same as the Lame coefficient lambda and it equals the velocity squared times the density. In the actual code (as opposed to the graphical output) all distances are in units of meters, all times are in units of seconds and density has units of kilograms per meter cubed. Since velocity has units of m/sec, the compressibility has units of (m/sec)^2 * kg/m^3 or kg/(m-sec^2) or Newtons/m^2. Now divergence has units of 1/m and displacement has units of meters so the divergence of displacement is dimensionless. So if we assume that the displacement time series are in meters then the pressure is in Newtons/m^2 or Pascals.

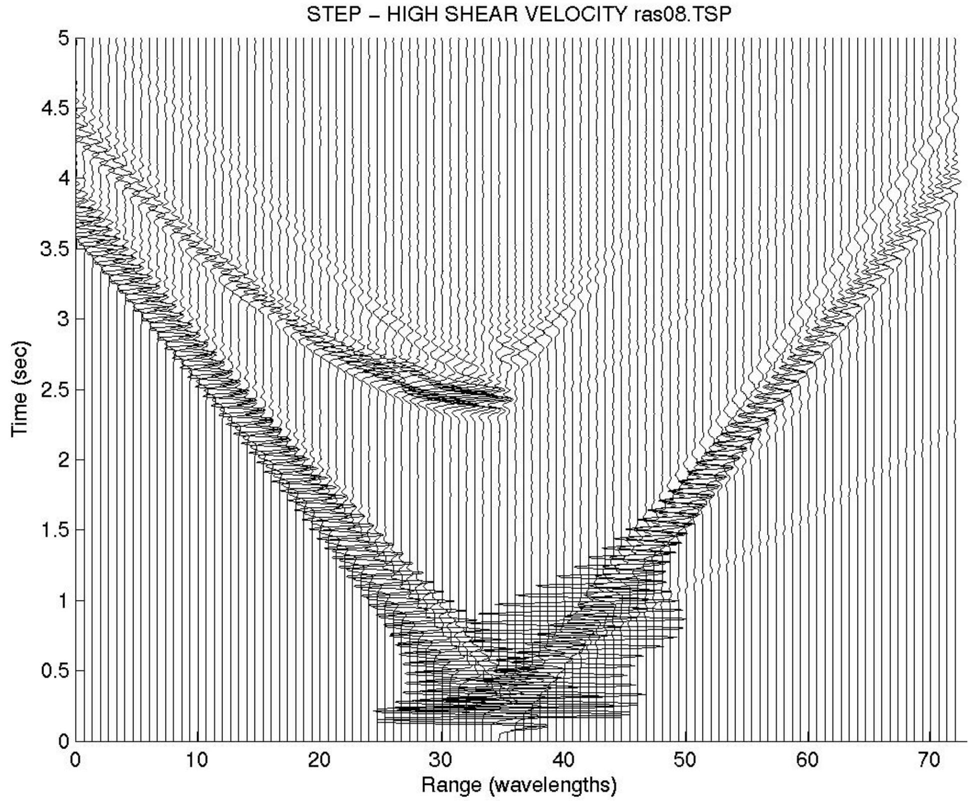

Figure 13: This figure shows the traces in the top row of ras08.TSP (with IVERT=3). These are the same traces as in Figure 10. The top row is located one wavelength above the top of the step and goes through the point source. (The point source is also offset one wavelength to the left from the edge of the step.) Since these receivers are all in the water (shear modulus is zero) the traces are **pressure**. Every fifth trace from Traces Numbers 5 to 545 is plotted with an amplitude scale of 0.000002. To get range in wavelengths from M divide by 15. For this time scale there are 150m per wavelength.

Figure 14: This figure shows the traces in the bottom row of ras08.TSP (IVERT=3). The left half are the same traces as the left half in Figure 11. The bottom row is located one wavelength below the top of the step. The left half is in the water and the right half is in the solid. Since the receivers on the **left half** are all in the water (shear modulus is zero) the traces are **pressure**. Every fifth trace from Traces Numbers 546 to 1086 is plotted with an amplitude scale of 0.000002. The traces on the right half represent horizontal displacement, which is much smaller in absolute value than pressure and appears to be zero. The traces on the right are shown in an expanded scale in Figure 15.

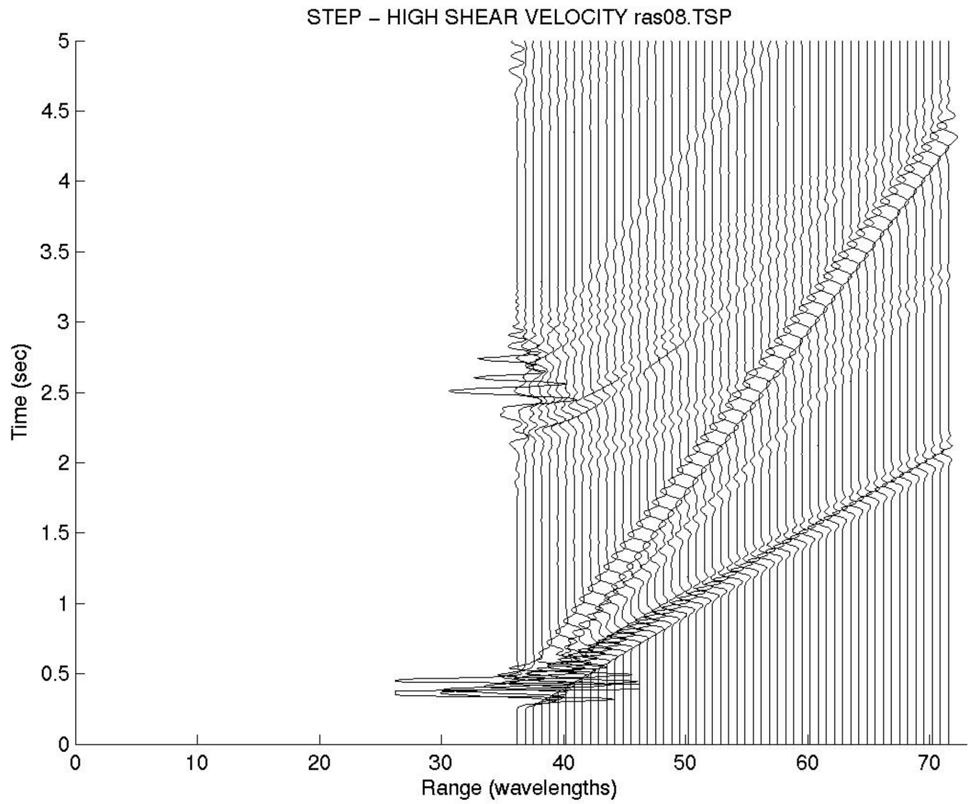

Figure 15: This figure shows the right half of the traces in the bottom row of ras08.**TSP**. The amplitude scale is 1000 compared to the amplitude scale for the pressure traces in Figures 13 and 14 of 0.000002. Since the right half is in the solid these traces represent **horizontal displacement**. Every fifth trace from Traces Numbers 817 to 1086 are plotted with an amplitude scale of 1000.

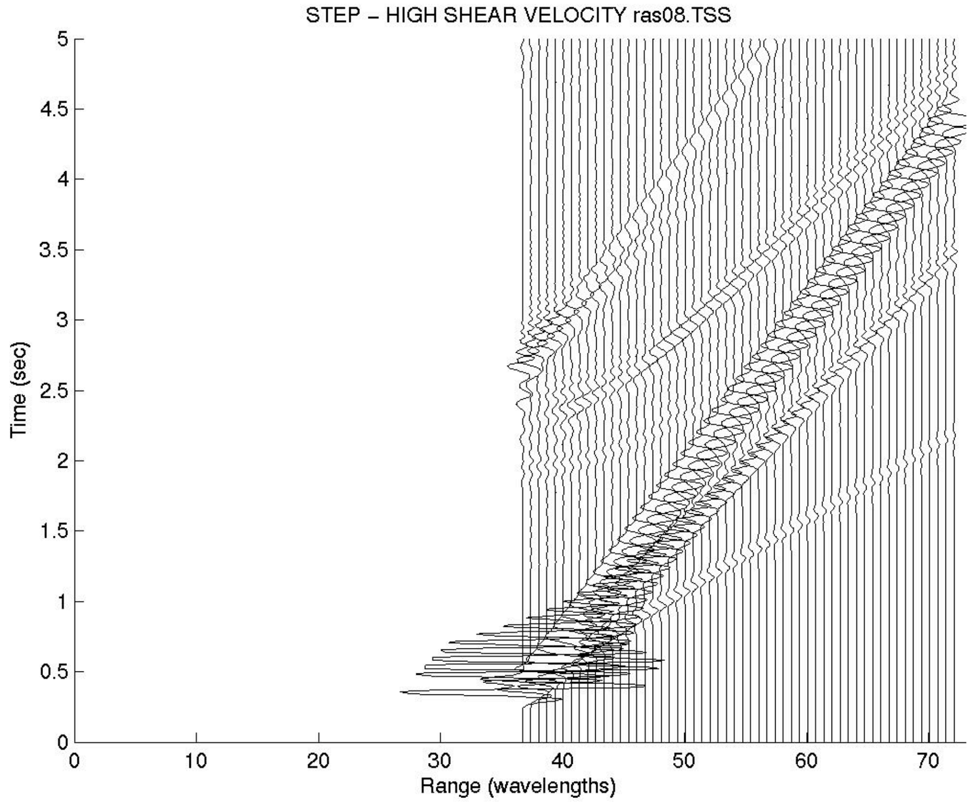

Figure 16: This figure shows the right half of the traces in the bottom row of ras08.**TSS**. Since the top row and left half of the bottom row are in water, their "normalized curl" are all zero. These are the only non-zero traces in ras08.TSS. The amplitude scale is 1000 compared to the amplitude scale for the pressure traces in Figures 13 and 14 of 0.000002. Since the right half is in the solid these traces represent **vertical displacement**. Every fifth trace from Traces Numbers 817 to 1086 are plotted with an amplitude scale of 1000.

Acknowledgements

We would like to thank the Office of Naval Research for funding the development and applications of this code for over twenty years. The actual production of this report was funded under ONR Award No. N00014-04-1-0090.

Selected References

[*Greaves and Stephen*, 2000; *Greaves and Stephen*, 2003; *Stephen*, 1990; *Stephen*, 1996; *Stephen*, 2000; *Stephen and Swift*, 1994a; *Stephen and Swift*, 1994b; *Swift and Stephen*, 1994]

Greaves, R.J., and R.A. Stephen, Low-grazing-angle monostatic acoustic reverberation from rough and heterogeneous seafloors, *Journal of the Acoustical Society of America*, *108*, 1013-1025, 2000.

Greaves, R.J., and R.A. Stephen, The influence of large-scale seafloor slope and average bottom velocity on low-grazing-angle monostatic acoustic reverberation, *Journal of the Acoustical Society of America*, *113*, 2548-2561, 2003.

Stephen, R.A., Solutions to range-dependent benchmark problems by the finite difference method, *Journal of the Acoustical Society of America*, *87* (No. 4), 1527-1534, 1990.

Stephen, R.A., Modeling sea surface scattering by the time-domain finite difference method, *Journal of the Acoustical Society of America*, *100*, 2070-2078, 1996.

Stephen, R.A., Optimum and standard beam widths for numerical modeling of interface scattering problems, *Journal of the Acoustical Society of America*, *107*, 1095-1102, 2000.

Stephen, R.A., and S.T. Bolmer, Notes for Geoacoustic_TDFD, Woods Hole Oceanographic Institution, Woods Hole, MA, in prep.

Stephen, R.A., and S.A. Swift, Finite difference modeling of geoacoustic interaction at anelastic seafloors, *Journal of the Acoustical Society of America*, *95* (1), 60-70, 1994a.

Stephen, R.A., and S.A. Swift, Modeling seafloor geoacoustic interaction with a numerical scattering chamber, *Journal of the Acoustical Society of America*, 96, 973-990, 1994b.

Swift, S.A., and R.A. Stephen, The scattering of a low-angle pulse beam by seafloor volume heterogeneities, Journal of the Acoustical Society of America, 96, 991-1001, 1994.

Appendix A: Wave Equations and Source Waveforms

(Based on Course 12.571 - Numerical Wave Propagation - Fall '00)

Appendix A - Part 1: Problem Set #1

1. Given the relations below, derive the elastic wave equation for heterogeneous, isotropic media: 1) as a second order partial differential equation in terms of displacements in a) vector notation (grad, div, curl) and b) tensor notation, and 2) as a system of first order equations in terms of particle velocity and stress in tensor notation.

Equation of Motion: $\rho \ddot{u}_i = \tau_{ij,j}$

Constitutive Relation: $\tau_{ij} = c_{ijpq} e_{pq}$

General Form of Isotropic Tensor: $c_{ijpq} = \lambda \delta_{ij} \delta_{pq} + \mu \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right)$

Definition of Infinitessimal Strain: $e_{pq} = \frac{1}{2} \left(u_{p,q} + u_{q,p} \right)$

2. Consider a compressional point source in a homogeneous fluid (which is a good representation of an explosive or a single airgun source in the ocean). The solution to the wave equation for the compressional displacement potential in a homogeneous liquid in cylindrical co-ordinates (r,z) is:

$$\phi(r,z,t) = \frac{A}{4\pi\rho\alpha^2 R} g(t - R/\alpha)$$

where: $R = \sqrt{r^2 + z^2}$ is the distance between the source and the observation point;
α is the compressional wave velocity in the fluid;
ρ is the density of the fluid; and
A is a unit constant with dimensions of (mass x length^2 / time).
(Note that A is frequently omitted from published solutions but it is necessary to keep the solutions dimensionally correct.)

A frequently used source waveform in synthetic seismogram work is the Gaussian curve:

$$g(t) = -2\zeta T e^{-\zeta T^2}, T = t - t_s$$

where ζ is the pulse width parameter and t_s is a time shift parameter.
a) What are the corresponding solutions for displacement ($\vec{u} = \nabla\phi$) and pressure ($p = -\alpha^2 \rho \nabla \cdot \vec{u}$) assuming the Gaussian curve above for the potential waveform? Show that the solutions are dimensionally correct. (Do this analytically.)

b) What is the frequency spectrum of the pressure waveform? What are the peak frequency and the upper and lower half-power frequencies as functions of ζ ? (This may be done numerically, in Matlab for example.)

Appendix A - Part 2: Solutions to Problem Set #1

1. Given the relations below, derive the elastic wave equation for heterogeneous, isotropic media: 1) as a second order partial differential equation in terms of displacements in a) vector notation (grad, div, curl) and b) tensor notation, and 2) as a system of first order equations in terms of particle velocity and stress in tensor notation.

Equation of Motion: $\rho \ddot{u}_i = \tau_{ij,j}$ (1)

Constitutive Relation: $\tau_{ij} = c_{ijpq} e_{pq}$ (2)

General Form of Isotropic Tensor: $c_{ijpq} = \lambda \delta_{ij} \delta_{pq} + \mu \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right)$ (3)

Definition of Infinitessimal Strain: $e_{pq} = \frac{1}{2} \left(u_{p,q} + u_{q,p} \right)$ (4)

Substitute the form of the isotropic tensor and the definition of infinitessimal strain into the constitutive relation (Generalized Hooke's Law):

$$
\begin{aligned}
\tau_{ij} &= c_{ijpq} e_{pq} \\
&= \lambda \delta_{ij} \delta_{pq} e_{pq} + \mu \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) e_{pq} \\
&= \lambda \tfrac{1}{2} \left(u_{p,q} + u_{q,p} \right) \delta_{ij} \delta_{pq} + \mu \tfrac{1}{2} \left(u_{p,q} + u_{q,p} \right) \left(\delta_{ip} \delta_{jq} + \delta_{iq} \delta_{jp} \right) \\
&= \quad \lambda A \quad + \quad \mu B
\end{aligned}
\tag{5}
$$

For the first term:

$$
\begin{aligned}
&if \quad i = j, \quad A = u_{p,p} = u_{1,1} + u_{2,2} + u_{3,3} \\
&\qquad\qquad A_{,j} = \frac{\partial A}{\partial x_j} = u_{1,1j} + u_{2,2j} + u_{3,3j} = u_{1,1i} + u_{2,2i} + u_{3,3i} = u_{p,pi} \\
&if \quad i \neq j, \quad A = 0
\end{aligned}
\tag{6}
$$

Now for the second term, since we always sum through all values of p and q (1, 2, 3), each term in the Kronecker delta expression will be unity and non-zero once for each pair of (i,j).

when $p = i$ *and* $q = j$

$$Ba = \tfrac{1}{2}\left(u_{i,j} + u_{j,i}\right)$$

$$Ba_{,j} = \frac{\partial B}{\partial x_j} = \tfrac{1}{2}\left(u_{i,jj} + u_{j,ij}\right) = \tfrac{1}{2}\left[\frac{\partial^2 u_i}{\partial x_j^2} + \frac{\partial}{\partial x_i}\left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}\right)\right] \qquad (7)$$

and when $p = j$ *and* $q = i$

$$Bb = \tfrac{1}{2}\left(u_{j,i} + u_{i,j}\right)$$

$$Bb_{,j} = \tfrac{1}{2}\left(u_{i,jj} + u_{j,ij}\right)$$

since Ba and Bb reduce to the same displacement expression, the effect of the Kronecker delta expresion is either to multiply the displacement expression by zero or two.   So:

$$\begin{aligned} B &= \left(u_{j,i} + u_{i,j}\right) \\ B_{,j} &= \left(u_{i,jj} + u_{j,ij}\right) \end{aligned} \qquad (8)$$

Now substituting into the equation of motion:

$$\begin{aligned} \rho\ddot{u}_i &= \tau_{ij,j} \\ &= \lambda_{,j} A + \lambda A_{,j} + \mu_{,j} B + \mu B_{,j} \\ &= \lambda_{,j} u_{p,p}\delta_{ij} + \lambda u_{p,pj}\delta_{ij} + \mu_{,j}\left(u_{i,j} + u_{j,i}\right) + \mu\left(u_{i,jj} + u_{j,ij}\right) \end{aligned} \qquad (9)$$

$$\rho\ddot{u}_i = \lambda u_{p,pi} + \mu\left(u_{i,jj} + u_{p,pi}\right) + \lambda_{,i} u_{p,p} + \mu_{,j}\left(u_{i,j} + u_{j,i}\right)$$

1-1-b. The last equation is the wave equation for heterogeneous, isotropic, perfectly elastic media in tensor notation. If the last two terms are dropped we get the wave equation for homogeneous, isotropic, perfectly elastic media in tensor notation. It is a second order partial differential equation in terms of displacement.

To convert (9) to vector notation we have the following identities:

$$
\begin{aligned}
&\nabla\lambda = \lambda_{,i} \\
&\nabla\cdot\mathbf{u} = \frac{\partial u_p}{\partial x_p} = u_{p,p} \\
&(\nabla\times\mathbf{u})_i = \varepsilon_{ijk}\frac{\partial u_k}{\partial xj} = \varepsilon_{ijk}u_{k,j} \\
&\nabla^2\mathbf{u} = \frac{\partial^2 u_i}{\partial x^2} = u_{i,jj} \\
&\nabla(\nabla\cdot\mathbf{u}) = u_{p,pi} \\
&\left[\nabla\mu\times(\nabla\times\mathbf{u})\right]_i = \varepsilon_{ijk}(\nabla\mu)_j(\nabla\times\mathbf{u})_k \\
&\qquad = \varepsilon_{ijk}\frac{\partial\mu}{\partial x_j}\varepsilon_{klm}\frac{\partial u_m}{\partial x_l} \\
&\qquad = \left(\delta_{il}\delta_{jm} - \delta_{im}\delta_{jl}\right)\frac{\partial\mu}{\partial x_j}\frac{\partial u_m}{\partial x_l} \\
&\qquad = \frac{\partial\mu}{\partial x_j}\frac{\partial u_j}{\partial x_i} - \frac{\partial\mu}{\partial x_j}\frac{\partial u_i}{\partial x_j} \\
&\qquad = \mu_{,j}u_{j,i} - (\nabla\mu\cdot\nabla)u_i \\
&(\nabla\mu\cdot\nabla)u_i = \frac{\partial\mu}{\partial x_j}\frac{\partial u_i}{\partial x_j} = \mu_{,j}u_{i,j} \\
&\left[\nabla\mu\times(\nabla\times\mathbf{u})\right] + (\nabla\mu\cdot\nabla)\mathbf{u} = \mu_{,j}u_{j,i}
\end{aligned}
\tag{10}
$$

where we have used the epsilon notation for the vector product:

$$
\begin{aligned}
&A\times B = \varepsilon_{ikm}A_kB_m \\
&\varepsilon_{ikm} = 0, \quad \textit{any two subscripts equal} \\
&\quad = 1, \quad \textit{subscripts in the order } 1,2,3,1,2,\ldots \\
&\quad = -1, \quad \textit{subscripts in the order } 2,1,3,2,1,\ldots \\
&\varepsilon_{ikm} = \varepsilon_{mik} = \varepsilon_{kmi} \\
&\varepsilon_{ikm}\varepsilon_{psm} = \delta_{ip}\delta_{ks} - \delta_{is}\delta_{kp}
\end{aligned}
\tag{11}
$$

1-1-a: Substituting expressions from (10) into (9) gives:

$$
\rho\ddot{\mathbf{u}} = (\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) + \mu\nabla^2\mathbf{u} + \nabla\lambda(\nabla\cdot\mathbf{u}) + \nabla\mu\times\nabla\times\mathbf{u} + 2(\nabla\mu\cdot\nabla)\mathbf{u} \tag{12}
$$

which is the wave equation for heterogeneous, isotropic, perfectly elastic media in vector notation. If the last three terms are dropped we get the wave equation for homogeneous, isotropic, perfectly elastic media in vector notation.

1-2: From the equation of motion (1) and the time derivative of (5) we get:

$$
\begin{aligned}
\dot{v}_i &= \frac{1}{\rho}\tau_{ij,j} \\
\dot{\tau}_{ij} &= c_{ijpq}\frac{1}{2}\left(v_{i,j} + v_{j,i}\right) \\
&= \frac{1}{2}\lambda\left(v_{p,q} + v_{q,p}\right)\delta_{pq}\delta_{ij} + \frac{1}{2}\mu\left(v_{p,q} + v_{q,p}\right)\left(\delta_{ip}\delta_{jq} + \delta_{iq}\delta_{jp}\right) \\
&= \lambda v_{p,p}\delta_{ij} + \mu\left(v_{i,j} + v_{j,i}\right)
\end{aligned} \tag{13}
$$

This expresses the wave equation for heterogeneous, isotropic, perfectly elastic media as a system of two first-order partial differential equation in terms of velocity and stress in tensor notation. Note that the equations are the same for homogeneous and heterogeneous media.

[optional] Now if we assume that all vectors are column vectors and use the transpose to convert column vectors to row vectors, and we take care to distinguish the two, we get the first order system in vector notation as:

$$
\begin{aligned}
\dot{\mathbf{v}} &= \frac{1}{\rho}\nabla^T \cdot \bar{\tau} \\
\dot{\bar{\tau}} &= \lambda\, I\,\nabla^T \cdot \mathbf{v} + \mu\left[\nabla \cdot \mathbf{v}^T + \left(\nabla \cdot \mathbf{v}^T\right)^T\right]
\end{aligned} \tag{14}
$$

where

$$
\begin{aligned}
\bar{\tau} &= \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \\
\nabla^T \cdot \mathbf{v} &= \mathit{scalar} \qquad\qquad \textit{inner product} \\
\nabla \cdot \mathbf{v}^T &= \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} & \frac{\partial v_z}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} & \frac{\partial v_z}{\partial y} \\ \frac{\partial v_x}{\partial z} & \frac{\partial v_y}{\partial z} & \frac{\partial v_z}{\partial z} \end{bmatrix} \qquad \textit{outer product}
\end{aligned} \tag{15}
$$

---

2. Consider a compressional point source in a homogeneous fluid (which is a good representation of an explosive or a single airgun source in the ocean). The solution to the wave equation for the compressional displacement potential in a homogeneous liquid in cylindrical co-ordinates (r,z) is:

$$\phi(r,z,t) = \frac{A}{4\pi\rho\alpha^2 R} g(t - R/\alpha) \tag{16}$$

where: $R = \sqrt{r^2 + z^2}$ is the distance between the source and the observation point;

α is the compressional wave velocity in the fluid;

ρ is the density of the fluid; and

A is a unit constant with dimensions of (mass x length^2 / time).

(Note that A is frequently omitted from published solutions but it is necessary to keep the solutions dimensionally correct.)

A frequently used source waveform in synthetic seismogram work is the Gaussian curve:

$$g(t) = -2\zeta T e^{-\zeta T^2}, T = t - t_s \tag{17}$$

where ζ is the pulse width parameter and t_s is a time shift parameter.

a) What are the corresponding solutions for displacement ($\vec{u} = \nabla\phi$) and pressure ($p = -\alpha^2\rho\nabla\cdot\vec{u}$) assuming the Gaussian curve above for the potential waveform? Show that the solutions are dimensionally correct. (Do this analytically.)

2-a) Start by taking the derivative of the potential with respect to radial distance, R:

$$\begin{aligned}\frac{\partial\phi}{\partial R} &= \frac{A}{4\pi\rho\alpha^2}\left[\frac{\partial(1/R)}{\partial R} g(t - R/\alpha) + \frac{1}{R}\frac{\partial g}{\partial R}(t - R/\alpha)\right] \\ &= \frac{A}{4\pi\rho\alpha^2}\left[-\frac{1}{R^2} g(t - R/\alpha) - \frac{1}{R\alpha} g'(t - R/\alpha)\right]\end{aligned} \tag{18}$$

Now displacement which is the gradient of the potential is:

$$
\begin{aligned}
\mathbf{u}(r,z,t) &= \begin{bmatrix} \dfrac{\partial}{\partial r} \\ \dfrac{\partial}{\partial z} \end{bmatrix} \phi(r,z,t) \\
&= \begin{bmatrix} \dfrac{\partial}{\partial r} \\ \dfrac{\partial}{\partial z} \end{bmatrix} R \cdot \frac{\partial \phi}{\partial R} \\
&= \frac{-A}{4\pi\rho\alpha^2} \begin{bmatrix} \dfrac{r}{R} \\ \dfrac{z}{R} \end{bmatrix} \cdot \left[\frac{g(t - R/\alpha)}{R^2} + \frac{g'(t - R/\alpha)}{R\alpha} \right]
\end{aligned} \tag{19}
$$

Now for pressure, using the wave equation for potential, we get:

$$
\begin{aligned}
p &= -\alpha^2 \rho \nabla \cdot \mathbf{u} = -\alpha^2 \rho \nabla^2 \phi = -\rho \frac{\partial^2 \phi}{\partial t^2} \\
&= \frac{-A}{4\pi\alpha^2 R} g''(t - R/\alpha)
\end{aligned} \tag{20}
$$

The derivatives of the Gaussian are:

$$
\begin{aligned}
g'(t) &= -2\zeta \left(\frac{\partial T}{\partial t} e^{-\zeta T^2} + T \frac{\partial}{\partial t} e^{-\zeta T^2} \right) \\
&= -2\zeta \left(e^{-\zeta T^2} + T(-2\zeta T) e^{-\zeta T^2} \right) \\
&= -2\zeta \left(1 - 2\zeta T^2 \right) e^{-\zeta T^2} \\
g''(t) &= -2\zeta \left(\frac{\partial (1 - 2\zeta T^2)}{\partial t} e^{-\zeta T^2} + (1 - 2\zeta T^2) \frac{\partial}{\partial t} e^{-\zeta T^2} \right) \\
&= -2\zeta \left(-2\zeta 2 T e^{-\zeta T^2} + (1 - 2\zeta T^2)(-2\zeta T) e^{-\zeta T^2} \right) \\
&= -2\zeta \left(-4\zeta T - 2\zeta T + 4\zeta^2 T^3 \right) e^{-\zeta T^2} \\
&= 4\zeta^2 \left(3T - 2\zeta T^3 \right) e^{-\zeta T^2}
\end{aligned} \tag{21}
$$

---

b) What is the frequency spectrum of the pressure waveform? What are the peak frequency and the upper and lower half-power frequencies as functions of ζ ? (This may be done numerically, in Matlab for example.)

```
%
%   PS1.m
%
%   This file computes the frequency spectrum (amplitude
amd phase) of the
%   pressure time series and estimates the peak frequency
and upper and lower
%   half-power frequencies as functions of the pulse-width
parameter.
%
zeta=1.;  %  Some initial values (guessed)
ts=0;
delt=0.001;  % Sample interval
Fs=1/delt;  % Sampling rate
t=-1.023:delt:1.024;  %  2048 points
length(t)    %  Make this a power of two for later fft's
%
%  Compute the pressure time series
%
T=t-ts;
gppatt=(4*zeta^2).*(3.*T-(2*zeta).*T.^3).*exp(-zeta.*T.^2);
plot(t,gppatt)
xlabel('Time (sec)')
ylabel('Amplitude (pressure units)')
title('First guess:  ts=0, zeta=1')
%
pause
%
%  The waveform above does not return to zero at the ends.
Try zeta=10.
%
zeta=10;
T=t-ts;
gppatt=(4*zeta^2).*(3.*T-(2*zeta).*T.^3).*exp(-zeta.*T.^2);
plot(t,gppatt)
xlabel('Time (sec)')
ylabel('Amplitude (pressure units)')
title('Second guess:  ts=0, zeta=10, NN=2048')
%
pause
%
%   This time series looks better;  let's take the fft
%
gppatf=fft(gppatt);
freq=(0:length(t)-1)/length(t)*Fs;
plot(freq,abs(gppatf))
axis([0 10 0 100000])
```

```
xlabel('Frequency (Hz)')
ylabel('Amplitude (pressure units)')
title('Amplitude spectra:  ts=0, zeta=10, NN=2048')
%
pause
%
%   Not smooth enough in frequency, try a longer time
series
%
zeta=10;
t=-4.095:delt:4.096;
T=t-ts;
gppatt=(4*zeta^2).*(3.*T-(2*zeta).*T.^3).*exp(-zeta.*T.^2);
plot(t,gppatt)
xlabel('Time (sec)')
ylabel('Amplitude (pressure units)')
title('Third guess:  ts=0, zeta=10, NN=8192')
%
pause
%
%   This time series looks better;  let's take the fft
%
gppatf=fft(gppatt);
freq=(0:length(t)-1)/length(t)*Fs;
plot(freq,abs(gppatf))
axis([0 10 0 100000])
xlabel('Frequency (Hz)')
ylabel('Amplitude (pressure units)')
title('Amplitude spectra:  ts=0, zeta=10, NN=8192')
%
pause
%
%   This spectra looks smooth and nice.  Let's get a
relationship between
%   zeta and the maximum value.
%
zet(1)=zeta;
pmax=max(abs(gppatf));
ind=find(abs(gppatf)>(pmax-1));
peak_freq(1)=freq(min(ind));
%
for index=2:6
%
   zeta=10*2^index;
   gppatt=(4*zeta^2).*(3.*T-(2*zeta).*T.^3).*exp(-
zeta.*T.^2);
   plot(t,gppatt)
   xlabel('Time (sec)')
```

```
   ylabel('Amplitude (pressure units)')
   title(['Third guess:  ts=0, zeta=
',num2str(zet(index)),', NN=8192'])
   pause
%
   gppatf=fft(gppatt);
   freq=(0:length(t)-1)/length(t)*Fs;
   plot(freq,abs(gppatf))
   axis([0 30 0 4000000])
   xlabel('Frequency (Hz)')
   ylabel('Amplitude (pressure units)')
   title(['Amplitude spectra: ts=0, zeta=
',num2str(zet(index)),', NN=8192'])
   pause
%
   zet(index)=zeta;
   pmax=max(abs(gppatf));
   ind=find(abs(gppatf)>(pmax-1));
   peak_freq(index)=freq(min(ind));
%
end
plot(zet,peak_freq,'b+',zet,peak_freq.^2,'ro')
axis([0 700 0 100])
xlabel('zeta (Hz^2)')
ylabel('Peak frequency (Hz) or Peak frequency squared
(Hz^2)')
title('Peak frequency as a function of zeta')
%
%  Zeta appears to be linearly related to the peak
frequency squared
%
coef=polyfit(zet,peak_freq.^2,1);
text(75,8.5,['Peak frequency squared =
',num2str(coef(2)),...
 ' plus zeta times ', num2str(coef(1))]);
%
%  Find and plot upper and lower half power frequencies
%
for index=1:6;
%
   zeta=10*2^index;
   gppatt=(4*zeta^2).*(3.*T-(2*zeta).*T.^3).*exp(-
zeta.*T.^2);
%
   gppatf=fft(gppatt);
   plot(freq,abs(gppatf))
   hold on
   axis([0 30 0 4000000])
```

```
   xlabel('Frequency (Hz)')
   ylabel('Amplitude (pressure units)')
   title(['Amplitude spectra: ts=0, zeta=
',num2str(zeta),', NN=8192'])
%
   pmax1=max(abs(gppatf));
   ind=find(abs(gppatf)>(pmax1-1));
   peak=freq(min(ind));
   ind1=find(abs(gppatf(1:(length(gppatf)/2)))>(pmax1/2-
1));
   low_pow=freq(min(ind1));
   high_pow=freq(max(ind1));
   r1(index)=peak/low_pow;
   r2(index)=peak/high_pow;
%

plot(peak,pmax1,'r+',low_pow,pmax1/2,'g+',high_pow,pmax1/2,
'g+')
   text(13,pmax,['Peak frequency = ',num2str(peak,3),'Hz'])
   text(13,pmax*0.95,['Lower half-power frequency =
',num2str(low_pow,3),'Hz'])
   text(13,pmax*0.9,['Upper half-power frequency =
',num2str(high_pow,3),'Hz'])
   text(13,pmax*0.85,['Peak/LHP = ',num2str(r1(index),3)])
   text(13,pmax*0.8,['Peak/UHP = ',num2str(r2(index),3)])
   hold off
   pause
%
end
%
%  End of script
%
```

Appendix A - Part 3: Solutions to Problem Set #1 (continued):

The direct calculation of pressure in Problem 2

2. Consider a compressional point source in a homogeneous fluid (which is a good representation of an explosive or a single airgun source in the ocean). The solution to the wave equation for the compressional displacement potential in a homogeneous liquid in cylindrical co-ordinates (r,z) is:

$$\phi(r,z,t) = \frac{A}{4\pi\rho\alpha^2 R} g(t - R/\alpha) \qquad (16)$$

where: $R = \sqrt{r^2 + z^2}$ is the distance between the source and the observation point;

α is the compressional wave velocity in the fluid;
ρ is the density of the fluid; and
A is a unit constant with dimensions of (mass x length^2 / time).
(Note that A is frequently omitted from published solutions but it is necessary to keep the solutions dimensionally correct.)

A frequently used source waveform in synthetic seismogram work is the Gaussian curve:

$$g(t) = -2\zeta T e^{-\zeta T^2}, T = t - t_s \tag{17}$$

where ζ is the pulse width parameter and t_s is a time shift parameter.
a) What are the corresponding solutions for displacement ($\vec{u} = \nabla\phi$) and pressure ($p = -\alpha^2\rho\nabla\cdot\vec{u}$) assuming the Gaussian curve above for the potential waveform? Show that the solutions are dimensionally correct. (Do this analytically.)

2-a) Start by taking the derivative of the potential with respect to radial distance, R:

$$\begin{aligned}\frac{\partial\phi}{\partial R} &= \frac{A}{4\pi\rho\alpha^2}\left[\frac{\partial(1/R)}{\partial R}g(t-R/\alpha) + \frac{1}{R}\frac{\partial g}{\partial R}(t-R/\alpha)\right] \\ &= \frac{A}{4\pi\rho\alpha^2}\left[-\frac{1}{R^2}g(t-R/\alpha) - \frac{1}{R\alpha}g'(t-R/\alpha)\right]\end{aligned} \tag{18}$$

Now displacement which is the gradient of the potential is:

$$\begin{aligned}\mathbf{u}(r,z,t) &= \begin{bmatrix}\frac{\partial}{\partial r} \\ \frac{\partial}{\partial z}\end{bmatrix}\phi(r,z,t) \\ &= \begin{bmatrix}\frac{\partial}{\partial r} \\ \frac{\partial}{\partial z}\end{bmatrix}R\cdot\frac{\partial\phi}{\partial R} \\ &= \frac{-A}{4\pi\rho\alpha^2}\begin{bmatrix}\frac{r}{R} \\ \frac{z}{R}\end{bmatrix}\cdot\left[\frac{g(t-R/\alpha)}{R^2} + \frac{g'(t-R/\alpha)}{R\alpha}\right]\end{aligned} \tag{19}$$

Now for pressure, using the divergence in **cylindrical coordinates** and substituting for the displacement from (19), we get:

$$
\begin{aligned}
p &= -\alpha^2 \rho \nabla \cdot \mathbf{u} \\
&= -\alpha^2 \rho \left\{ \frac{1}{r} \frac{\partial (r u_r)}{\partial r} + \frac{\partial u_z}{\partial z} \right\} \\
&= \frac{A}{4\pi} \left\{ \frac{1}{r} \frac{\partial}{\partial r} \left[\frac{r^2}{R} (h(R)) \right] + \frac{\partial}{\partial z} \left[\frac{z}{R} (h(R)) \right] \right\} \\
&= \frac{A}{4\pi} \left\{ \frac{1}{r} \frac{\partial}{\partial r} \left(\frac{r^2}{R} \right) \cdot h(R) + \frac{r}{R} \frac{\partial h(R)}{\partial r} + \frac{\partial}{\partial z} \left(\frac{z}{R} \right) \cdot h(R) + \frac{z}{R} \frac{\partial h(R)}{\partial z} \right\} \\
&= \frac{A}{4\pi} \left\{ \frac{1}{r} \frac{\partial}{\partial r} \left(\frac{r^2}{R} \right) \cdot h(R) + \frac{r}{R} \frac{\partial R}{\partial r} \frac{\partial h(R)}{\partial R} + \frac{\partial}{\partial z} \left(\frac{z}{R} \right) \cdot h(R) + \frac{z}{R} \frac{\partial R}{\partial z} \frac{\partial h(R)}{\partial R} \right\} \\
&= \frac{A}{4\pi} \left\{ \left[\frac{1}{r} \frac{\partial}{\partial r} \left(\frac{r^2}{R} \right) + \frac{\partial}{\partial z} \left(\frac{z}{R} \right) \right] \cdot h(R) + \left[\frac{r}{R} \frac{\partial R}{\partial r} + \frac{z}{R} \frac{\partial R}{\partial z} \right] \cdot \frac{\partial h(R)}{\partial R} \right\}
\end{aligned}
\tag{22}
$$

where

$$
h(R) = \frac{g(t - R/\alpha)}{R^2} + \frac{g'(t - R/\alpha)}{R\alpha}
$$

Using the relations in Table 1 we have:

$$
\begin{aligned}
p &= \frac{A}{4\pi} \left\{ \left[\frac{R^2 + z^2}{R^3} + \frac{r^2}{R^3} \right] \cdot h(R) + \left[\frac{r}{R} \cdot \frac{r}{R} + \frac{z}{R} \cdot \frac{z}{R} \right] \cdot \frac{\partial h(R)}{\partial R} \right\} \\
&= \frac{A}{4\pi} \Bigg\{ \left[\frac{2}{R} \right] \cdot \left[\frac{g(t - R/\alpha)}{R^2} + \frac{g'(t - R/\alpha)}{R\alpha} \right] \\
&\qquad - \frac{2}{R^3} g(t - R/\alpha) - \frac{2}{R^2\alpha} g'(t - R/\alpha) - \frac{1}{R\alpha^2} g''(t - R/\alpha) \Bigg\} \\
&= \frac{A}{4\pi} \left\{ -\frac{1}{R\alpha^2} g''(t - R/\alpha) \right\} \\
&= \frac{-A}{4\pi\alpha^2 R} g''(t - R/\alpha)
\end{aligned}
\tag{24}
$$

which is the same as equation (20) in Part 2. The derivatives of the Gaussian were given by equation (21) in Part 2.

Table 1: Some useful relations

$$
\begin{aligned}
\frac{\partial R}{\partial r} &= \frac{\partial}{\partial r}\left(r^2+z^2\right)^{1/2} = \frac{1}{2}\left(r^2+z^2\right)^{-1/2}2r = \frac{r}{R} \\
\frac{\partial R}{\partial z} &= \frac{z}{R} \\
\frac{\partial}{\partial z}\left(\frac{z}{R}\right) &= \frac{\partial z}{\partial z}\cdot\frac{1}{R} + z\frac{\partial}{\partial z}\left(r^2+z^2\right)^{-1/2} \\
&= \frac{1}{R} + z\left(-\frac{1}{2}\right)\left(r^2+z^2\right)^{-3/2}2z \\
&= \frac{1}{R} - \frac{z^2}{R^3} = \frac{R^2-z^2}{R^3} = \frac{r^2}{R^3} \\
\frac{\partial}{\partial r}\left(\frac{r^2}{R}\right) &= \frac{\partial r^2}{\partial r}\cdot\frac{1}{R} + r^2\frac{\partial}{\partial r}\left(r^2+z^2\right)^{-1/2} \\
&= \frac{2r}{R} - r^2\left(r^2+z^2\right)^{-3/2}r \\
&= \frac{2r}{R} - \frac{r^3}{R^3} = \frac{r\left(2R^2-r^2\right)}{R^3} = \frac{r\left(R^2+z^2\right)}{R^3} \\
\frac{\partial h(R)}{\partial R} &= \frac{\partial}{\partial R}\left(\frac{1}{R^2}\right)\cdot g(t-R/\alpha) + \frac{1}{R^2}\frac{\partial g(t-R/\alpha)}{\partial R} \\
&\quad + \frac{\partial}{\partial R}\left(\frac{1}{R\alpha}\right)\cdot g'(t-R/\alpha) + \frac{1}{R\alpha}\frac{\partial g'(t-R/\alpha)}{\partial R} \\
&= \frac{-2}{R^3}g(t-R/\alpha) + \frac{1}{R^2}\left(-\frac{1}{\alpha}\right)g'(t-R/\alpha) \\
&\quad + \left(\frac{-1}{R^2\alpha}\right)g'(t-R/\alpha) + \frac{1}{R\alpha}\left(-\frac{1}{\alpha}\right)g''(t-R/\alpha) \\
&= \frac{-2}{R^3}g(t-R/\alpha) - \frac{2}{R^2\alpha}g'(t-R/\alpha) - \frac{1}{R\alpha^2}g''(t-R/\alpha)
\end{aligned}
\tag{23}
$$

Appendix B: Defining Models Within the Grid

(Based on Course 12.571 - Numerical Wave Propagation - Fall '00)

Appendix B - Part 1: Problem Set #3

This problem will prepare materials for a finite-difference calculation to be done later. Consider a vertical, two-dimensional (rectangular) cross-section of the earth. The region will be about 12 wavelengths deep and 72 wavelengths long. We have absorbing boundaries on each edge of the region. We have the option of placing a point compressional source anywhere within the grid and of defining a structure arbitrarily on the grid.

We will assume that the source has a time dependence discussed in Problem #2 of Problem Set #1. For convenience, the peak frequency in pressure will be 10Hz, and we will define the wavelength based on the peak frequency in water. Problems at other frequencies can be scaled easily since our results will always be in wavelengths and periods.

At the top of the grid we have a layer of water (Vp=1500m/s, density=1000kg/m^3). At the bottom of the grid we have a basaltic basement (Vp=3000m/s, Vs=1730m/s and density=2000kg/m^3).

1. Write a code (in matlab, C, ...) to output a table of ascii values of compressional velocity, shear velocity and density at each grid point. Make the sampling interval a variable in your code, but assume for this exercise that we sample at 20 points per water wavelength. Display the three parameter grids as contour plots or color plots. Write code and display the parameters for the following models:

a) A flat, horizontal interface between water and basalt located mid-way in the grid.

b) A sinusoidally rough bottom with an amplitude of one water wavelength and a structural wavelength of 20 water wavelengths. Make the amplitude and structural wavelengths variables in your code.

c) Keeping the water layer at the top and the basaltic layer at the bottom, define a structure that you would like to see studied by finite differences. The minimum velocity (compressional or shear) should be 250m/s and the highest velocity should be 5,000m/s.

Appendix B - Part 2: tdfd_grid.m

```
%  This file creates ascii arrays to define models for
input to a
%  a two dimensional finite difference code (12.571 Problem
Set #3)
%
Nlx=72;  %  Number of wavelengths long
Nlz=12;  %  Number of wavelengths deep
delx=1/20; %  Grid interval in x and z in units of
wavelengths
%
Nx=Nlx/delx+1;  %  Number of grid points long
Nz=Nlz/delx+1;  %  Number of grid points deep
Nzh=((Nz-1)/2); %  Number of points to half-depth
%
%  Loop over the three models
%
for ind=1:3
%
   velp=ones(Nz,Nx);  %  Initialize arrays
   vels=ones(Nz,Nx);
   ro=ones(Nz,Nx);
   velp(1:2,:)=velp(1:2,:)*1500;  %  Top two rows are water
   vels(1:2,:)=vels(1:2,:)*0;
   ro(1:2,:)=ro(1:2,:)*1000;
%  Bottom two rows are basalt
   velp((Nz-1):Nz,:)=velp((Nz-1):Nz,:)*3000;
   vels((Nz-1):Nz,:)=vels((Nz-1):Nz,:)*1730;
   ro((Nz-1):Nz,:)=ro((Nz-1):Nz,:)*2000;
%
   if ind==1
%
%  Model a:  Flat, horizontal interface between water and
basalt
%
      velp(3:Nzh,:)=velp(3:Nzh,:)*1500;  %  Top half is
water
      vels(3:Nzh,:)=vels(3:Nzh,:)*0;
      ro(3:Nzh,:)=ro(3:Nzh,:)*1000;
%
%  Bottom half is basalt
      velp((Nzh+1):(Nz-2),:)=velp((Nzh+1):(Nz-2),:)*3000;
      vels((Nzh+1):(Nz-2),:)=vels((Nzh+1):(Nz-2),:)*1730;
      ro((Nzh+1):(Nz-2),:)=ro((Nzh+1):(Nz-2),:)*2000;
%
   elseif ind==2
%
```

```
%  Model b:  Sinusoidal interface between water and basalt
%
      amp=1;    %  Amplitude of structure in acoustic
wavelengths
      wave=20;  %  Wavelength of structure in acoustic
wavelengths
      Namp=amp/delx;
      Nwave=wave/delx;
      range=0:delx:Nlx;
      Nzp=round(Nzh+Namp.*sin(2.*pi.*range./wave));
      for index=1:Nx

velp(3:Nzp(index),index)=velp(3:Nzp(index),index)*1500;  %
Top half is water

vels(3:Nzp(index),index)=vels(3:Nzp(index),index)*0;

ro(3:Nzp(index),index)=ro(3:Nzp(index),index)*1000;
%
%  Bottom half is basalt
         velp((Nzp(index)+1):(Nz-
2),index)=velp((Nzp(index)+1):(Nz-2),index)*3000;
         vels((Nzp(index)+1):(Nz-
2),index)=vels((Nzp(index)+1):(Nz-2),index)*1730;
         ro((Nzp(index)+1):(Nz-
2),index)=ro((Nzp(index)+1):(Nz-2),index)*2000;
      end
%
   elseif ind==3
%
%  Model c:  Cylinder beneath a flat, horizontal interface
%
      velp(3:Nzh,:)=velp(3:Nzh,:)*1500;  %  Top half is
water
      vels(3:Nzh,:)=vels(3:Nzh,:)*0;
      ro(3:Nzh,:)=ro(3:Nzh,:)*1000;
%
%  Bottom half is basalt
      velp((Nzh+1):(Nz-2),:)=velp((Nzh+1):(Nz-2),:)*3000;
      vels((Nzh+1):(Nz-2),:)=vels((Nzh+1):(Nz-2),:)*1730;
      ro((Nzh+1):(Nz-2),:)=ro((Nzh+1):(Nz-2),:)*2000;
%
%  Add a cylinder of water below the interface
%
      radius=0.5;    %  Radius of cylinder
      cylx=36;  %  Range of cylinder (center)
      cylz=1.0;  %  Depth of cylinder (center) below
interface
```

```
      Nrad=round(radius/delx);
      Ncylx=round(cylx/delx);
      Ncylz=round(cylz/delx);
      Nrange=-Nrad:1:Nrad;
      Ndepth=round(sqrt(Nrad^2-Nrange.^2));
      for indx=1:length(Nrange)
         indxp=Nrange(indx);
         if Ndepth(indx)~=0
            for indz=-Ndepth(indx):Ndepth(indx)
               velp(indz+Ncylz+Nzh,indxp+Ncylx)=1500;
               vels(indz+Ncylz+Nzh,indxp+Ncylx)=0;
               ro(indz+Ncylz+Nzh,indxp+Ncylx)=1000;
            end
         end
      end
   end
%
%  Plot the three parameters
   set(gcf,'PaperPosition',[1.,1.,6.5,9.0]);
   subplot(3,1,1)
   v=[1500 1501 3000];
   [c,h] = contourf(velp,v);
   if ind==1
   title('Model a: Flat Interface  -  P-Vel (m/s)')
   elseif ind==2
   title('Model b: Sinusoidal Interface  -  P-Vel (m/s)')
   elseif ind==3
   title('Model c: Tunnel  -  P-Vel (m/s)')
   end
   colorbar
   subplot(3,1,2)
   v=[0 1 1730];
   [c,h] = contourf(vels,v);
   title('S-Vel (m/s)')
   colorbar
   subplot(3,1,3)
   v=[1000 1001 2000];
   [c,h] = contourf(ro,v);
   title('Density (kg/m^3)')
   colorbar
%
   if ind~=3
      pause
   end
end
%
%  End of script
%
```

Appendix C: Running the TDFD Code

(Based on Course 12.571 - Numerical Wave Propagation - Fall '00)

Problem Set #5 - Rev 1 (25/10/00)

The objective of this problem set is to run a two dimensional finite difference code on the models that were defined in Problem Set #3 (Appendix B). The goal is to carry out these runs on a UNIX machine with a Fortran 77 compiler. The input will be the compressional velocity, shear velocity and density arrays that were computed in the matlab code in PS#3. The output will be snapshots of compressional and shear energy density at various times during the simulation. Follow these steps:

1) Define a five character name for each of your three models. I suggest three initials followed by a model number, for example, **abc01 (01 represents the number one)**. In what follows you should replace abc01 with the name you have chosen for your model.

2) Modify your matlab code from PS#3 to output compressional velocity, shear velocity and density on a grid at 15 points per wavelength. It should still be 12 wavelengths deep and 72 wavelengths long. Each array will have dimensions 181x1081. Call these arrays velp, vels and ro. Add the following code to your matlab script to output the three arrays for each model.

```
[n,m]=size(velp);
save -ascii abc01_vp.dat  n m velp
save -ascii abc01_vs.dat  n m vels
save -ascii abc01_ro.dat  n m ro
```

NB: IT IS IMPORTANT THAT THE ARRAYS velp, vels and ro HAVE DIMENSIONS [n,m]=[181,1081]. n IS THE NUMBER OF ROWS (=181). m IS THE NUMBER OF COLUMNS (=1081).

3) Create a directory in your working area for each model and change directories to this working area:

```
mkdir abc01
cd abc01
```

4) An example directory exists in the tar file at:

../GeoAcoustic_TDFD/PS5/ras01

Call this PATH. Copy the following two files from the example directory into your working directory:

```
cp PATH/ras01_long.bch abc01.bch
```

```
cp PATH/ras01.par abc01.par
```

5) Assuming that the data files that you created in Step 2) are in YOUR_PATH copy these files to your working area:

```
cp YOUR_PATH/abc01_vp.dat .
cp YOUR_PATH/abc01_vs.dat .
cp YOUR_PATH/abc01_ro.dat .
```

6) Edit the script file, abc01.bch, and the parameter file, abc01.par, to customize them for your run. Replace all occurrences of ras01 with abc01 in both files.

7) You are ready to run the script file. This takes about an hour on a 2000 vintage work station and will create 100Mbytes of files in your working directory. The command below will start the script running in the background and save the log output of the program and other files into your work area. This code assumes a Guassian beam incident at 15 degrees grazing angle.

```
sh abc01.bch >  & abc01.t1 &
```

8) Upon completion of the program you can plot the snapshot data (with names like abc01****.DIV and abc01****.CRL) with the matlab code plot_findif_1 that is included in a separate tar file. In your abc01 directory start matlab and set the path to the directory containing plot_findif_1.m. Then enter the following command:

```
[lfpar2]=plot_findif_1;
```

a) Check that the full path to the files you want to plot is in the top window. Check the model name. The script will then find the DIV and CRL files and it will create a menu of the snapshot numbers next to "TIME STEP".

b) Use an amplitude scale of plus/minus 1,000,000 for the beam source. This is all that you need to do for now. The plot title will not make sense, but the label at the bottom of the plot page will. Hit the plot button. A plot of the DIV, CRL and VEL files should come up. You may need to change the max and min values used for "contouring".

c) Plot hardcopies by using the plot window's print button in the file menu. On some machines it may be necessary to save the plot as a file and then print it from the command line outside of matlab.

Appendix D: Model Parameters

This is an example parameter file for the point source over a flat, seafloor model (ras04). The code has been modified many times for many different problems. Some of the parameters are irrelevant for some problems. There is some documentation in the code itself.

```
'ras04'
  2001, 401, 5001, 1
  0.001, 10, 10
  1500,  000,  1000
  3000, 1730, 2000
  00, 0, 00
  4, 185, 100, 15.,  3.91
  14, 657.0, 0.0
  1, 5001,   4
  11, 1091, 2
  5, 185,  1
  0, 5, 1000, 1000
  3, -2, 3
  0.01, 12.5, 180
  -1,  -3, 93, 551, 5
  20, 20, 0.0002, 2.0
  0.0, 0.000001, 1.0E-10, 1000, 2650, 3.6E+10, 2.25E+09,...
4.36E+07, 2.61E+07
```

This .par file is read in the subroutine RDMPAR in bfsub.f:

```
C
C     READ .PAR FILE
C
      READ( IUNIT, *) FILEID
      READ( IUNIT, *) MM,NN,KK,KSTRT
      READ( IUNIT, *) DELT,DELR,DELZ
      READ( IUNIT, *) VP1,VS1,RO1
      READ( IUNIT, *) VP2,VS2,RO2
      READ( IUNIT, *) VPT,VST,ROT
      READ( IUNIT, *) NA,NB,NDEP,ANGLE,WIDTH
      READ( IUNIT, *) NSORCE,PLSWID,TSWAVE
      READ( IUNIT, *) KOUTST,KOUTEN,KINC
      READ( IUNIT, *) MOUTST,MOUTEN,MINC
      READ( IUNIT, *) NWOUTST,NWOUTEN,NWINC
      READ( IUNIT, *) NSFREQ,NSFINC,KMARK,KMINC
      READ( IUNIT, *) ISORB,IVERT,ISNST
      READ( IUNIT, *) BALP,BALPBOT,NALPWTH
      READ( IUNIT, *) KLOOPS,KLOOPE,NDEPSORS, MD, ND
      READ( IUNIT, *) QP,QS,TAU1,TAU2
      READ( IUNIT, *, END=999, ERR=999)
```

```
     *      PORO,VISC,PERM,ROF,ROS,RKS,RKF,RKB,RNB
C
```

There are also some notes on flags in the comments section of bfdif3.f:

```
C
C***********************************************************
************
C
C       SOME NOTES ON FLAGS (ISORB, IVERT, ISNST):
C
C
C      IF ISORB IS LESS THAN ZERO THEN THE CODE COMPUTES
ATTENUATION.
C
C      IF IABS(ISORB) IS 1, THEN THE CODE READS THE SOURCE
FUNCTION
C      FROM AN OUTSIDE FILE IN THE MANNER USED FOR THE
NUMERICAL
C      SCATTERING CHAMBER.
C
C      IF IABS(ISORB) IS 2, THEN THE CODE READS THE SOURCE
FUNCTION
C      FROM AN OUTSIDE FILE IN THE MANNER USED BY THORSOS
FOR TEST CASE 1
C      OF THE 1994 BENCHMARK WORKSHOP.
C
C      IF IABS(ISORB) IS 3, THEN THE CODE COMPUTES A POINT
SOURCE.
C
C*********
C
C      IF IVERT.LT.0 THEN THE TIME SERIES ARE COMPUTED
AROUND THE
C      TOP OF THE BOX, OTHERWISE THEY ARE COMPUTED AROUND
THE WHOLE BOX.
C
C      IF ABS(IVERT) IS 1, THEN THE TIME SERIES OUTPUT IS
NORMALIZED
C      DILATATION (PRESSURE).
C
C      IF ABS(IVERT) IS 2, THEN THE TIME SERIES OUTPUT IS
NORMALIZED
C      DILATATION (PRESSURE) AND NORMALIZED ROTATION.
C
C********
C
```

```
C      IF ISNST IS LESS THAN ZERO THEN CODE IS SET-UP FOR CW
SOURCE,
C      OTHERWISE PULSE IS COMPUTED.
C
C      IF ABS(ISNST) IS 1 THEN SNAPSHOTS ARE OF VERTICAL
DISPLACEMENT.
C
C      IF ABS(ISNST) IS 2 THEN SNAPSHOTS ARE OF HORIZONTAL
DISPLACEMENT.
C
C      IF ABS(ISNST) IS 3 THEN SNAPSHOTS ARE OF
COMPRESSIONAL AND SHEAR
C      AMPLITUDE DENSITY.
C
C********
C
C      IF NALPWTH IS GREATER THAN 700 THEN IHIG=NALPWTH-720
C      AND NALPWTH IS SET TO 720
C
C      IF IHIG IS 0 THEN HIGDON ABSORBING BOUNDARIES AND
T.E. ARE
C      USED EVERYWHERE
C
C      IF IHIG IS 1 THEN HIGDON ABSORBING BOUNDARIES ARE
DISABLED
C      ON THE LEFT HAND SIDE.
C
C      IF IHIG IS 2 THEN HIGDON ABSORBING BOUNDARIES ARE
DISABLED
C      EVERYWHERE.
C
C*****************************************************
*************
C
C      PORO IS USED AS A FLAG TO TRIGGER THE BIOT CODE
C
```

There are three "grids": the computational grid, the transition zone grid, and the snapshot grid. There should really be some figures to describe how the various grids and indices overlap but I will do my best with prose for now.

The computational grid consists of four sub-domains: the physical domain and three absorbing boundary domains. The physical domain has dimensions MM x NN. Absorbing boundaries are handled by a combination of a tapered telegraph equation over a width of NALPWTH and a parabolic approximation at the edge (Higdon for example). On the right and bottom edges absorbing boundaries are applied within the physical domain. So the effective size of the physical domain (the region without any absorbing boundary calculations) is (MM-NALPWTH) x (NN-NALPWTH). The three absorbing boundary domains are attached to the top and left sides and the top- left corner of the physical domain. The actual width of the absorbing boundary domains is different from NALPWTH. These have dimensions of MM x NDEP1, NDEP x NN and NDEP x NDEP1 respectively. [In our example, NDEP = 100, NALPWTH = 180 and NDEP1=NALPWTH*5/9.]

The physical domain is separated into three layers: a homogeneous layer on top, a transition zone layer which can have arbitrary variability in 2-D for compressional and shear velocity and density, and a homogeneous layer on the bottom. Since the homogeneous layers can be arbitrarily thin, you can still handle fairly general two dimensional structures. [The bottom absorbing boundary is within the homogeneous bottom layer so this must be at least NALPWTH thick. Absorbing boundaries on the right side (the last NALPWTH points in range) are handled separately for the three layers. The top side and top-left corner absorbing boundary domains are for homogeneous media. The left absorbing boundary domain is handled separately for the three layers.] The top layer is always a fluid, but the parameters can be chosen to make it look like a "free" medium.

The location of the transition zone grid is defined by NA and NB as depths within the physical domain. To allow for overlap between the transition zone and the homogeneous layers the "width" of the transition zone is NBNDY = NB - NA + 3, the top two rows of the transition zone must have parameters corresponding to the top homogeneous layer and the bottom two rows of the transition zone must have parameters corresponding to the bottom homogeneous layer. The matlab code tdfd_grid.m computes values for the transition zone. The range values (M) are the same in the transition zone and the physical domain, but the depth values (N) are not. N_TZ = N_PD - NA + 2. In our examples NA=4 so the top row of the transition zone (N_TZ=1) corresponds to (N_PD) row 3 in the physical domain.

The snapshot grid represents a coarse subset of the physical domain. For convenience in displaying results we try to use a constant display format for the snapshots. See the examples (Figures 1 to 6). This display shows (from top to bottom) the divergence of the displacement field, the curl of the displacement field and the compressional velocity model. Various options for outputting the snapshot fields are presented in the subroutine ZDIVCRL in bfdif2.f. It is convenient for each snapshot field to have dimensions 550 x 190. In range snapshot fields start at MMST and are incremented by MMINC until there are 550 points. In depth snapshot fields start at NNST and are incremented by NNINC until there are 190 points. The depth indices are in the physical domain, not the transition zone (or TDFD_grid) domain.

Time series are output for given receiver locations which are defined by KOUTST, KOUTEN, KINC, MOUTST, MOUTEN, MINC, NWOUTST, NWOUTEN, and NWINC. You can easily put a vertical or horizontal line of receivers anywhere. The logic was based on putting a box of receivers around the whole grid for computing scattering functions.

Here is a description of each parameter:

FILEID - This is the model name (5 digits) and is used in the filename of most output files.
MM - Total number of range points in the physical domain. Includes NALPWTH for the absorbing region at the right end.
NN - Total number of depth points in the physical domain. Includes NALPWTH for the absorbing region at the bottom.
KK - Total number of time steps for the calculation.
KSTRT - The start time index. Always equals 1.
DELT - The time step in seconds. Needs to satisfy the stability criterion.
DELR - The range step in meters. Needs to satisfy the dispersion criterion.
DELZ - The depth step in meters. Always equals the range step.
VP1 - Compressional velocity in the top homogeneous region in meters/sec.
VS1 - Shear velocity in the top homogeneous region in meters/sec.
RO1 - Density in the top homogeneous region in kg/m^3.
VP2 - Compressional velocity in the bottom homogeneous region in meters/sec.
VS2 - Shear velocity in the bottom homogeneous region in meters/sec.
RO2 - Density in the bottom homogeneous region in kg/m^3.
VPT - Dummy value included for historical reasons.
VST - Dummy value included for historical reasons.
ROT - Dummy value included for historical reasons.
NA - Index in the physical domain for the top of the transition zone.
NB - Index in the physical domain for the bottom of the transition zone.
NDEP - Thickness of the absorbing boundary domain on the left side.
ANGLE - Grazing angle of the Gaussian beam for beam calculations in degrees. Usually 15 degrees.
WIDTH - Spatial half-width parameter for the Gaussian beam in "wavelengths".
NSORCE - 2^NSORCE is the length of the source time series used for FFT's in the implementation of Gaussian beams.
PLSWID - Time width parameter for the Ricker wavelet. 657 corresponds to a peak frequency in pressure of 10Hz.
TSWAVE - Time shift parameter for the Ricker wavelet. If = 0 then this gets computed in the code.
KOUTST - Starting time index for the output time series. Usually equals 1.
KOUTEN - Stopping time index for the output time series. Usually equals KK.
KINC - Time index increment for the output time series.
MOUTST - Starting range index (in the physical domain) for the output time series.
MOUTEN - Stopping range index for the output time series.
MINC - Range index increment for the output time series.
NWOUTST - Starting depth index (in the physical domain) for the output time series.
NWOUTEN - Stopping depth index for the output time series.
NWINC - Depth index increment for the output time series.
NSFREQ - Dummy value included for historical reasons.
NSFINC - Dummy value included for historical reasons.
KMARK - Time index of first snapshot.

KMINC - Time index increment for snapshots.
ISORB - Flag used to select intrinsic attenuation (don't confuse with the absorbing boundaries) and type of source function. See the notes above. For a point source without intrinsic attenuation this should equal 3.
IVERT - Flag to control the output time series. The philosophy here is based on a box. The top and bottom of the box are horizontal lines of receivers at NWOUTST and NWOUTEN with ranges given by MOUTST, MOUTEN and MINC. The left and right sides of the box are vertical lines of receivers at MOUTST and MOUTWEN with depths given by NWOUTST, NWOUTEN and NWINC. For a horizontal line of pressure receivers (in the water) choose IVERT = -1 and set NWOUTEN = NWOUTST + NINC. The time series are output by the subroutine TSOUT in bfsub.f. The coordinates of each receiver in the physical domain are given by MLOC and NLOC in the header preceding each time series.
ISNST - Flag to select CW or pulse sources and the type of snapshots. ISNST = 3 selects a pulse source and snapshots of compressional and shear amplitude density. To not output any snapshots set KMARK to a number greater than KK.
BALP - Parameter to adjust the weights for the telegraph equation in the absorbing boundary.
BALPBOT - Parameter to adjust the weights for the telegraph equation in the absorbing boundary.
NALPWTH - Parameter to set the width the telegraph equation in the absorbing boundary. These three absorbing boundary parameters should not be changed unless there is a problem with the absorbing boundaries.
KLOOPS - Dummy value included for historical reasons.
KLOOPE - Dummy value included for historical reasons. Unfortunately MOD(KLOOPE-KOUTST,KINC) must equal zero.
NDEPSORS - Used for the beam source to set how far down the left edge (in the physical domain) the source should be introduced. Not used for point sources.
MD - Range increment in the physical domain of the point source.
ND - Depth increment in the physical domain of the point source.
QP - Parameter for intrinsic attenuation code.
QS - Parameter for intrinsic attenuation code.
TAU1 - Parameter for intrinsic attenuation code.
TAU2 - Parameter for intrinsic attenuation code.
PORO - Parameter for BIOT code. PORO = 0 indicates no Biot calculations are performed.
VISC - Parameter for BIOT code.
PERM - Parameter for BIOT code.
ROF - Parameter for BIOT code.
ROS - Parameter for BIOT code.
RKS - Parameter for BIOT code.
RKF - Parameter for BIOT code.
RKB - Parameter for BIOT code.
RNB - Parameter for BIOT code.

Appendix E: tdfd_grid_qspace_RAS.m

```
%
%
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* *
%
%         Copyright (c) 2003 material of
%
%         Woods Hole Oceanographic Institution
%         All rights reserved.
%
%         This material cannot be distributed or sold
without
%         prior permission of the author(s).
%
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* *
%
%  Special version of tdfd_grid.m to generate a quarter
space model
%  for Dennis.  We will call it ras07.  (Ralph Stephen -
Feb 25, 2004).
%
figure(1)
clf
%
Nlx=72;  %  Number of wavelengths long
Nlz=12;  %  Number of wavelengths deep
delx=1/15; %  Grid interval in x and z in units of
wavelengths
%
Nx=Nlx/delx+1;  %  Number of grid points long
Nz=Nlz/delx+1;  %  Number of grid points deep
Nzh=((Nz-1)/2); %  Number of points to half-depth
%
Nxh=((Nx-1)/2); %  Number of points to half-range
Nzd2=2/delx+1; %  Number of points to two wavelength depth
%
%  Loop over the three models
%
for ind=1:1
%
   velp=ones(Nz,Nx);  %  Initialize arrays
   vels=ones(Nz,Nx);
   ro=ones(Nz,Nx);
```

```
%
   velp(1:2,:)=velp(1:2,:)*1500;  %  Top two rows are water
   vels(1:2,:)=vels(1:2,:)*0;
   ro(1:2,:)=ro(1:2,:)*1000;
%
%  Bottom two rows are basalt
   velp((Nz-1):Nz,:)=velp((Nz-1):Nz,:)*3000;
   vels((Nz-1):Nz,:)=vels((Nz-1):Nz,:)*1730;
   ro((Nz-1):Nz,:)=ro((Nz-1):Nz,:)*2000;
%
   if ind==1
%
%  Large step model
%
      velp(3:Nzd2,:)=velp(3:Nzd2,:)*1500;  %  Top two
wavelengths is all water
      vels(3:Nzd2,:)=vels(3:Nzd2,:)*0;
      ro(3:Nzd2,:)=ro(3:Nzd2,:)*1000;
%
%  The rest is half water and half basalt
      velp((Nzd2+1):(Nz-2),1:Nxh)=velp((Nzd2+1):(Nz-
2),1:Nxh)*1500;
      vels((Nzd2+1):(Nz-2),1:Nxh)=vels((Nzd2+1):(Nz-
2),1:Nxh)*0;
      ro((Nzd2+1):(Nz-2),1:Nxh)=ro((Nzd2+1):(Nz-
2),1:Nxh)*1000;
%
      velp((Nzd2+1):(Nz-2),(Nxh+1):Nx)=velp((Nzd2+1):(Nz-
2),(Nxh+1):Nx)*3000;
      vels((Nzd2+1):(Nz-2),(Nxh+1):Nx)=vels((Nzd2+1):(Nz-
2),(Nxh+1):Nx)*1730;
      ro((Nzd2+1):(Nz-2),(Nxh+1):Nx)=ro((Nzd2+1):(Nz-
2),(Nxh+1):Nx)*2000;
%
   end
%
%  Plot the three parameters
%
   set(gcf,'PaperPosition',[1.,1.,6.5,9.0]);
   subplot(3,1,1)
   v=[1500 1501 3000];
   [c,h] = contourf(flipud(velp),v);
   if ind==1
   title('Large Step Model  -  P-Vel (m/s)')
   end
   colorbar
   subplot(3,1,2)
   v=[0 1 1730];
```

```
   [c,h] = contourf(flipud(vels),v);
   title('S-Vel (m/s)')
   colorbar
   subplot(3,1,3)
   v=[1000 1001 2000];
   [c,h] = contourf(flipud(ro),v);
   title('Density (kg/m^3)')
   colorbar
%
%  Output arrays for bfdif3 tdfd code
%
   if ind==1
       [n,m]=size(velp);
       save -ascii ras07_vp.dat  n m velp
       save -ascii ras07_vs.dat  n m vels
       save -ascii ras07_ro.dat  n m ro
       print -dpsc2 ras07_fig.ps
   end
end
%
%  End of script
%
```

Appendix F: ras07.par

```
'ras07'
  2001, 401, 5001, 1
  0.001, 10, 10
  1500,  000,  1000
  3000, 1730, 2000
  00, 0, 00
  4, 185, 100, 15.,  3.91
  14, 657.0, 0.0
  1, 5001,   4
  11, 1091, 2
  18, 48,  30
  0, 5, 1000, 1000
  3, 2, 3
  0.01, 12.5, 180
  -1,  -3, 93, 525, 18
  20, 20, 0.0002, 2.0
  0.0, 0.000001, 1.0E-10, 1000, 2650, 3.6E+10, 2.25E+09,...
4.36E+07, 2.61E+07
```

Appendix G: ras08.par

```
 'ras08'
  2001, 401, 5001, 1
  0.001, 10, 10
  1500,  000,  1000
  3000, 1730, 2000
  00, 0, 00
  4, 185, 100, 15.,  3.91
  14, 657.0, 0.0
  1, 5001,   4
  1, 1081, 2
  18, 48,  30
  0, 5, 1000, 1000
  3, 3, 3
  0.01, 12.5, 180
  -1,  -3, 93, 525, 18
  20, 20, 0.0002, 2.0
 0.0, 0.000001, 1.0E-10, 1000, 2650, 3.6E+10, 2.25E+09, 4.36E+07, 2.61E+07
```