Appendix B

Verification of the elastic adjoint operation

B.1 Overview

B.1.1 Testing adjoint implementations

One way to test an implementation of an adjoint operation is to see whether the answer appears to be correct. This is done in chapter 4 where one iteration of an inversion ($\approx$ the adjoint operation) applied to a diffraction yields a filtered point.

Another more quantitative way is by checking if the implementation of the adjoint numerically obeys the definition of the adjoint. This is also known as the dot product test and is the test preferred by Jon Claerbout. The dot product test consists of numerically checking that

$$\mathbf{x}^T \mathbf{A}^T \mathbf{y} = \mathbf{y}^T \mathbf{A} \mathbf{x} \pm \varepsilon \quad , \tag{B.1}$$

where $\mathbf{A}$ denotes a forward operator, $\mathbf{A}^T$ denotes the presumed adjoint operator to $\mathbf{A}$ that is to be checked, $\mathbf{x}$ and $\mathbf{y}$ are arbitrary vectors, and ε is the expected numerical error which is related to computer precision and accumulation of roundoff errors in doing the operations $\mathbf{A}$ and $\mathbf{A}^T$.

B.1.2 1D acoustic tutorial

This appendix uses a dot product test to provide a numerical verification that the adjoint operation given by equations 3.36 through 3.44 has been correctly implemented. Before doing this for the general elastic case, I will specialize the formulas to the constant-velocity acoustic case in 1D (i.e. solve for density only). This tutorial should help provide the reader with a good understanding of exactly how to compute the adjoint equations.

The 1D density adjoint program is based on the interpretation of the adjoint given in section 3.4.6. As expected, it yields the same program that one could obtain by reordering loops in the linearized forward problem (an adjoint operation can be expressed as multiplication by a transposed forward problem matrix $\mathbf{A}^T$ and can thus be achieved by carefully reordering loops in the linearized forward problem code $\mathbf{A}$).

B.1.3 1D acoustic results

For small sized problems, the dot product test is passed to computer precision (i.e. $\varepsilon \approx 10^{-7}$) verifying the numerical implementation is correct. Several tests indicate that two factors can increase ε, the size of the problem and the way that the forward problem handles roundoff. For larger 1D problems (of a size equivalent to a small 2D problem) and using the same eight-point convolutional derivative operator used in the 2D elastic modeling program, the size of ε increases to 10^{-5}. This implies that the level of accumulated roundoff noise in the elastic finite difference program used to do inversions in this thesis is about 10^{-5}.

B.1.4 2D elastic results

The elastic inversion code computes the linearized forward problem as the difference between two nonlinear problems (as a finite difference). This results in an accumulated roundoff level of about 10^{-3} rather than a level of 10^{-5} implied by the 1D acoustic tutorial. Thus, the value of ε, the expected precision of the dot product test is also 10^{-3}. A dot product test done using the elastic program passes to within this expected precision of 10^{-3} verifying the 2D elastic adjoint implementation is correct to within the accuracy of the test.

B.2 Tutorial: the 1D constant-velocity acoustic case

In order to gain a knowledge of the dot product precision parameter ε one may expect in an elastic inversion and to gain an understanding of the adjoint calculations, I will first treat the special case of a 1D constant-velocity acoustic medium.

B.2.1 Mathematical proof of the adjoint

From equation 3.33, the linearized elastic forward problem specialized to a constant velocity and variable density 1D medium is

$$\delta u(x_r,t) \;=\; \int dx\; \delta\rho(x)\; \dot{G}(x_r,t;x,0) * \dot{u}(x,t) \;\equiv\; \mathbf{Ax} \quad . \tag{B.2}$$

From equation 3.36, the adjoint operation is

$$\begin{aligned} \delta\rho(x) \;&=\; -\int dt \int dx_r\; \dot{G}(x_r,t;x,0) * \dot{u}(x,t)\; \delta u(x_r,t) \\ &=\; -\int dt \int dx_r\; \dot{u}(x,t)\; \dot{\psi}(x,t) \;\equiv\; \mathbf{A}^T\mathbf{y} \quad , \end{aligned} \tag{B.3}$$

where

$$\psi(x,t) \;=\; \int dx_r\; G(x,-t;x_r,0) * \delta u(x_r,t) \quad , \tag{B.4}$$

and

$$u(x,t) \;=\; \int dx'\; G(x,t;x',0) * f(x',t) \quad , \tag{B.5}$$

is the nonlinear forward problem (the Green's function $G(x,t;x',0)$ is the impulse response of a delta source at $t=0$ and $x=x'$).

Mathematically, equation B.3 is the adjoint operation corresponding to equation B.2 as can be seen by the following

$$\begin{aligned} \mathbf{y}^T\mathbf{Ax} \;&\equiv\; \int dt \int dx_r\; \delta u(x_r,t) \int dx\; \delta\rho(x)\; \dot{G}(x_r,t;x,0) * \dot{u}(x,t) \\ &=\; \int\int\int dt\; dx_r\; dx\; \delta\rho(x)\; \delta u(x_r,t)\; \dot{G}(x_r,t;x,0) * \dot{u}(x,t) \quad . \end{aligned} \tag{B.6}$$

and

$$\begin{aligned} \mathbf{x}^T\mathbf{A}^T\mathbf{y} \;&\equiv\; \int dx\; \delta\rho(x) \int dt \int dx_r \dot{G}(x_r,t;x,0) * \dot{u}(x,t)\; \delta u(x_r,t) \\ &=\; \int\int\int dt\; dx_r\; dx\; \delta\rho(x)\; \delta u(x_r,t)\; \dot{G}(x_r,t;x,0) * \dot{u}(x,t) \quad . \end{aligned} \tag{B.7}$$

Equations B.6 and B.7 are equal verifying

$$\mathbf{x}^T\mathbf{Ay} \;=\; \mathbf{y}^T\mathbf{Ax} \tag{B.8}$$

which is the definition of the adjoint given in elementary mathematical texts. This test is mathematically trivial considering I derived the adjoint expression to obey the definition of an adjoint. A more interesting test is whether the dot product test is passed numerically to the expected precision.

B.2.2 Numerical calculation

Linearized forward problem

In practice, it is too expensive (in the 2D calculations in this thesis) to evaluate the integrals over the Green's functions to compute the forward problem. Rather than evaluating δu using,

$$\delta u = \int dx\ \delta\rho\ \dot{G} * \dot{u} \quad , \tag{B.9}$$

I first compute a wavefield $\delta v(x,t)$ by solving the wave equation

$$\rho\ \delta\ddot{v} - \lambda\ \partial_{xx}\delta v = -\delta\rho\ \ddot{u} \quad , \tag{B.10}$$

and then interpolate the data at the receiver locations using

$$\delta u(x_r,t) = \int dx\ \delta(x-x_r)\delta v(x,t) \quad . \tag{B.11}$$

Equation B.9 corresponds to equation 3.31 and equation B.11 corresponds to equation 3.28.

Linearized adjoint

Similarly, to do the adjoint operation, I first compute wavefield ψ by solving the wave equation

$$\rho\ \ddot{\psi} - \lambda\ \partial_{xx}\psi = -\delta u(x_r,-t) \quad , \tag{B.12}$$

followed by calculation of the integral

$$\delta\rho = \int dt\ \dot{u}\ \dot{\psi} \quad . \tag{B.13}$$

Equation B.12 corresponds to the wave equation solution to 3.43 and equation B.13 corresponds to equation 3.36.

B.2.3 Algorithm

Introduction

In this section I will define the algorithm to compute the linearized forward and adjoint operations. (Note that in my thesis I solve the nonlinear forward problem. I use the linearized forward problem here only because it is required in the dot product test.)

Definitions

I make use of the following definitions

$$L = (\rho\, \partial_{tt} - \lambda\, \partial_{xx}) \equiv wave\ \ equation\ \ operator \quad , \tag{B.14}$$

and

$$W = window + interpolate\ a\ wavefield\ to\ get\ data\ at\ the\ receiver\ locations\ . \tag{B.15}$$

The linearized forward problem

Using the definitions, the linearized forward problem becomes

$$\begin{aligned} L\, u &= f \quad , \\ L\, \delta v &= \ddot{u}\, \delta\rho \quad , \\ \delta u &= W\, \delta v \quad . \end{aligned} \tag{B.16}$$

Putting these operations together we have mathematically that

$$\delta u = W\, L^{-1}\, \ddot{u}\, \delta\rho \quad . \tag{B.17}$$

The linearized adjoint

The linearized adjoint in terms of the wave equation and interpolation operators is

$$\begin{aligned} L\, u &= f \quad , \\ \delta f &= W^T\, \delta u \quad , \\ L^T\, \psi &= \delta f \quad , \\ \delta\hat{\rho} &= \ddot{u}\, \psi = \dot{u}\, \dot{\psi} \quad . \end{aligned} \tag{B.18}$$

Putting these operations together we have mathematically that

$$\delta\hat{\rho} = \ddot{u}\, L^{-T}\, W^T\, \delta u \quad , \tag{B.19}$$

which is clearly the adjoint of equation B.17 because $\ddot{u} = \ddot{u}^T$. Note that the transpose wave equation program corresponding to L^T can be obtained by reversing the order of the time loop in the wave equation program corresponding to L.

B.2.4 Pseudo code

The linearized forward problem

Some pseudo code to perform the forward equations given by B.16 using finite differences to solve the wave equation is

$\delta v \; = \; 0$

$do\ t = 1, T \, \{$

$\quad \delta v_{xx}(t,x) = \partial_x(\Delta x/2) * \partial_x(-\Delta x/2) * dv(t,x)$

$\quad do\ x = 1, X \, \{$

$\qquad \delta v(t,x) = 2\delta v(t-1,x) - \delta v(t-2,x) + s^2 \delta v_{xx}(t,x) - \ddot{u}(t,x)\delta\rho(x)$

$\qquad \}$

$\quad \}$

$\delta v \; = \; 0$

$do\ t = 1, T \, \{$

$\quad do\ x = 1, X \, \{$

$\qquad do\ x' = -X', X' \, \{$

$\qquad\quad \delta u(t,x) = \delta u(t,x) + w(t,x,x')\delta v(t, x - x')$

$\qquad \}$

$\quad \}$

$\}$

where $\partial_x(\Delta x/2)$ convolutional derivative operator centered at $\Delta x/2$, $s^2 = (\lambda \Delta t^2)/(\Delta x^2)$, Δt is the time step and Δx is the grid spacing in the finite differences (the 1D finite difference stability criterion is $s^2 \; \leq \; 1$).

The linearized adjoint problem

Similarly, some pseudo code to perform the adjoint equations given by B.18 is

$\delta f \; = \; 0$

$do\ t = 1, T \, \{$

$\quad do\ x = 1, X \, \{$

$\qquad do\ x' = -X', X' \, \{$

$\qquad\quad \delta f(t,x) = \delta f(t,x) + w(t,x,x')\delta u(t, x - x')$

$\qquad \}$

```
    }
}
ψ = 0
δρ = 0
do t = T,1 {
    do x = 1,X {
        ψ_xx(t,x) = ∂_x(Δx/2) * ∂_x(−Δx/2) * ψ(t,x)
        ψ(t,x) = 2ψ(t+1,x) − ψ(t+2,x) + s²ψ_xx(t,x) − δf(t,x)
        δρ(x) = δρ(x) + ψ(t,x)ü(t,x)
    }
}
```

B.2.5 Numerical test

Input data

Typical vectors used in the dot product test are illustrated in Figures B.1 through B.5. They were obtained as described below.

$\mathbf{x} \equiv \delta\rho$: random density perturbations.

$\mathbf{x}_1 \equiv \delta\rho_1$: random density perturbations (note that $\delta\rho_1 \neq \delta\rho$)

$\mathbf{y} = \mathbf{A}\mathbf{x}_1 \equiv \delta u$: computed by modeling with density perturbations $\delta\rho_1 = \mathbf{x}_1$.

$\mathbf{A}\mathbf{x} \equiv W\ L^{-1}\ \ddot{u}\ \delta\rho$: computed by modeling with density perturbations $\delta\rho = \mathbf{x}$.

$\mathbf{A}^T\mathbf{y} \equiv \ddot{u}\ L^{-T}\ W^T\ \delta u$: the adjoint applied to displacement perturbations $\delta u = \mathbf{y}$.

Note that the background wavefield $u(x,t)$, the perturbation wavefield $\delta v(x,t)$ and the back-propagated residual wavefield $\psi(x,t)$ are shown in Figures B.6 through B.8 illustrating that the boundaries have been reached by the waves (hence, the dot product test is also checking whether the boundary conditions are adjoint).

Dot product test

I performed a dot product test several times using different random input vectors $\mathbf{x}$ and $\mathbf{x}_1$ for some different cases to gain an understanding of the dot product precision parameter ε.

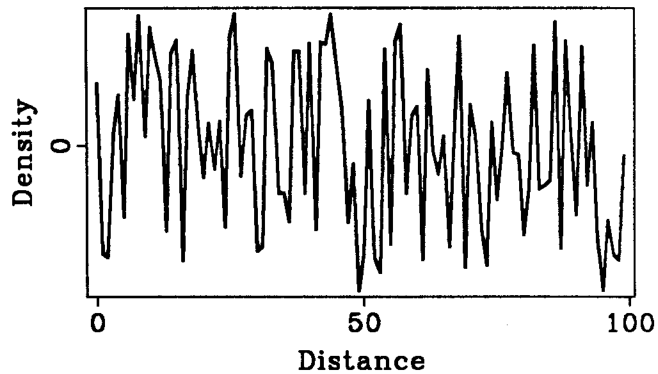

Figure B.1: The density perturbation vector $\delta\rho = \mathbf{x}$.

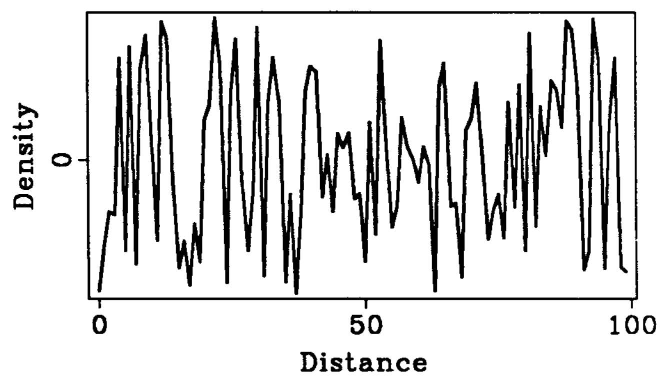

Figure B.2: The density perturbation vector $\delta\rho_1 = \mathbf{x}_1$.

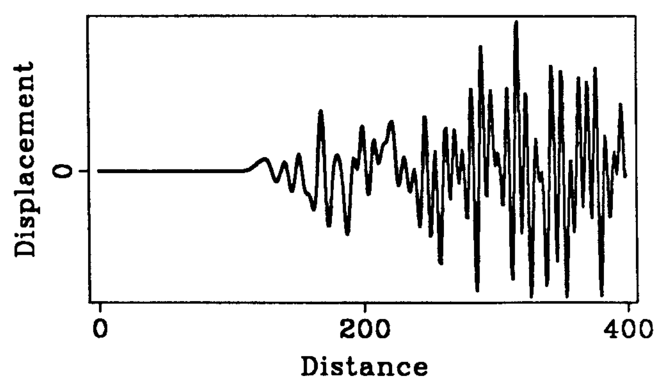

Figure B.3: The displacement perturbation vector $\delta u = \mathbf{y} = \mathbf{A}\mathbf{x}_1$.

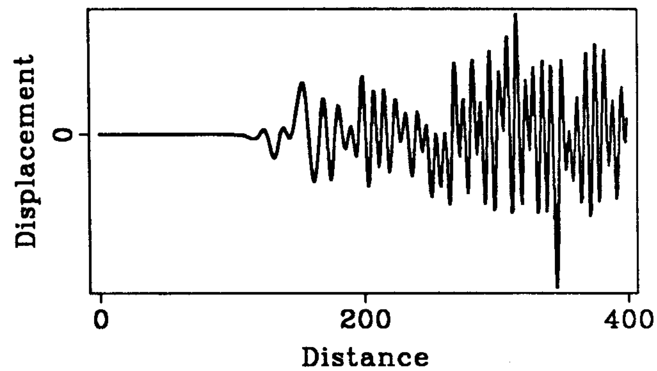

Figure B.4: The displacement perturbation vector $\mathbf{Ax}$.

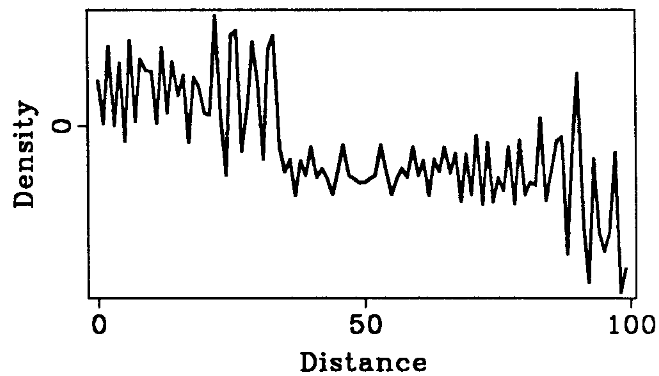

Figure B.5: The adjoint vector $\delta\hat{\rho} = \mathbf{A}^T\mathbf{x}$.

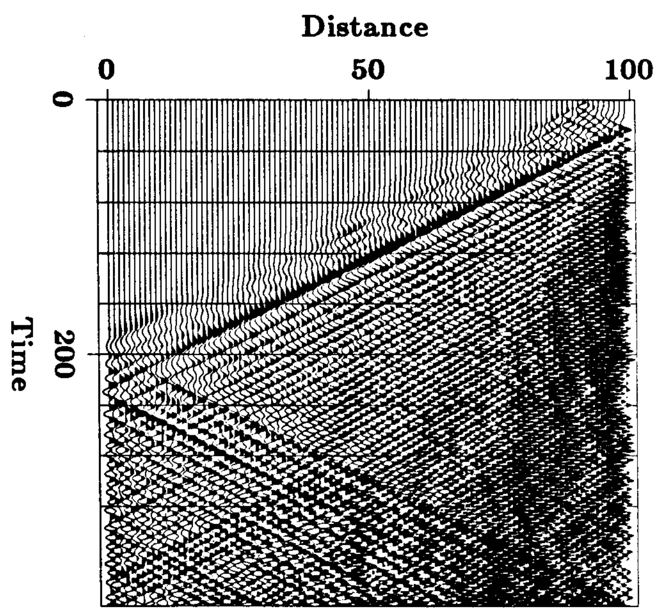

Figure B.6: The background wavefield $u(x,t)$.

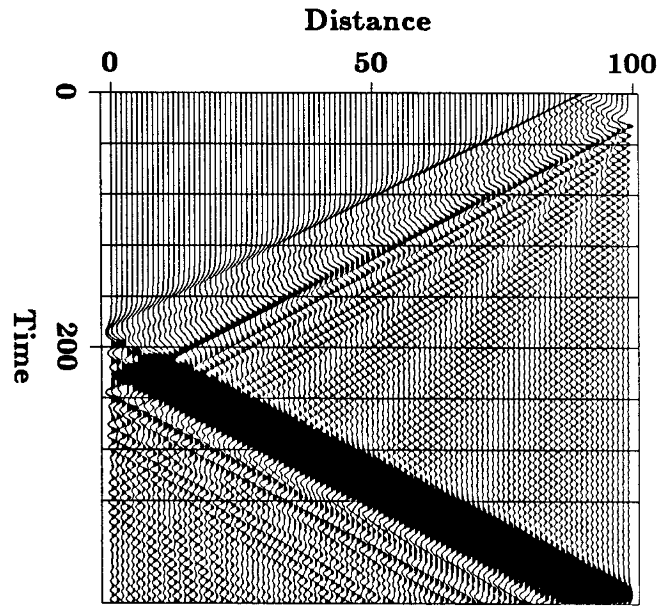

Figure B.7: The perturbation wavefield $\delta v(x,t)$.

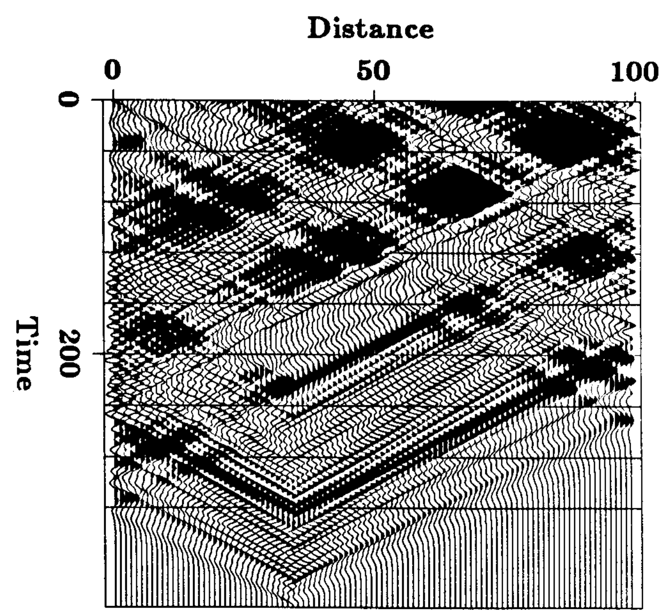

Figure B.8: The back-propagated residual wavefield $\psi(x,t)$.

Case 1: A small example (i.e. $X = 10$ and $T = 40$) using a two point derivative operator in the finite difference scheme. Typical results of dot product tests are

$$\mathbf{y}^T \mathbf{A} \mathbf{x} = 1.0000000 \quad ,$$

and

$$\mathbf{x}^T \mathbf{A}^T \mathbf{y} = 1.0000005 \quad .$$

The results indicate 7 figure precision implying a correct adjoint implementation.

Case 2: A big example (i.e. $X = 100$ and $T = 400$) using a two point derivative operator in the finite difference scheme. Typical results of dot product tests are

$$\mathbf{y}^T \mathbf{A} \mathbf{x} = 1.0000027 \quad ,$$

and

$$\mathbf{x}^T \mathbf{A}^T \mathbf{y} = 1.0000000 \quad .$$

These results indicate that there is some accumulation of roundoff when the problem is bigger so the dot product test has less precision. Note by less precision I mean that the dot product test has less precision (i.e. ε is larger) and not that the program contains bugs. This is clear because the small example presented in case 1 had a precision equal to computer accuracy.

Case 3: Same as case 2 but using the same eight point derivative operator as was used in the 2D elastic finite difference program. Therefore, the value of ε here is approximately the expected precision of a 2D elastic dot product test. Typical results of dot product tests are

$$\mathbf{y}^T\mathbf{A}\mathbf{x} = 1.0000000 \quad ,$$

and

$$\mathbf{x}^T\mathbf{A}^T\mathbf{y} = 1.0000320 \quad .$$

This 5 figure precision indicates that relative to the two-point derivative operator (case 2), the eight-point derivative operator causes a greater accumulation of roundoff.

Case 4: Same as case 3 but for the small sized 10x40 example (i.e. $X = 10$ and $T = 40$). Typical results of dot product tests are

$$\mathbf{y}^T\mathbf{A}\mathbf{x} = 1.0000014 \quad ,$$

and

$$\mathbf{x}^T\mathbf{A}^T\mathbf{y} = 1.0000000 \quad .$$

The dot product test here has 6 figures of precision which is greater than the 5 figure precision of case 3. This implies that the program using an eight-point operator is also a correct adjoint implementation but with more accumulated roundoff than a two point scheme (i.e. compare case 4 with case 1).

The conclusions so far are that there two factors affecting the dot product precision parameter ε). Namely, that an increase in the size and/or accumulation of roundoff in the forward problem results in an increase in ε.

Precision of dot product tests

The examples above indicated that there is considerable variation in the dot product precision parameter ε which is required to know whether or not a dot product test is passed. The precision parameter is ideally computer precision (7 figures) when the size of the problem is small. As the size increases, this precision parameter may increase by a factor of 10. Another factor is how much roundoff is accumulated in the forward problem (finite difference scheme). Ideally, a finite difference scheme attenuates the roundoff to ensure stability but the different schemes can attenuate the roundoff at different rates.

This variation could account for different precisions in dot product tests. The eight-point scheme I use in the elastic inversions accumulates 10 times more roundoff than a two-point scheme (note that this is not as bad as it seems considering the eight-point scheme can use 1/4 the number grid-points per wavelength required by a two-point scheme to solve a given problem).

B.3 2D elastic dot product test

B.3.1 Summary

Now I will check the implementation of the elastic adjoint operation (equations 3.48 through 3.50 and 3.59 through 3.61) using a dot product test. Recall that the dot product test numerically checks that $\mathbf{x}^T\mathbf{A}^T\mathbf{y} = \mathbf{y}^T\mathbf{A}\mathbf{x} \pm \varepsilon$. Case 3 of the previous section indicated that ε for a small 2D example should be approximately 10^{-5} provided the linearized forward problem is done as it was in the 1D example. However, for ease of implementation, I will do the linearized forward problem as the difference between two nonlinear problems leading to 100 times less precision in the linearized forward problem. This results in a dot product precision parameter of $\varepsilon = 10^{-3}$. I achieve this expected precision in a dot product test verifying that my adjoint implementation is correct (as well as can be tested using this implementation of the linearized forward problem).

B.3.2 Linearized forward problem

The dot product test requires a linear forward operator $\mathbf{A}$ and its adjoint. In my elastic inversion I use the nonlinear forward problem. This is because:

(i) I wish to solve the more difficult nonlinear inverse problem which requires the nonlinear forward problem, and

(ii) the linearized forward problem involves the same kind of computations as the nonlinear problem (see thesis equations 3.28 through 3.30 and drop the O^2 terms) so it is not advantageous to program the linearized equations (i.e. the linear problem is the same speed as the nonlinear problem).

To do the dot product test without programming the linearized forward problem separately, (i.e. without writing extra code that saves the background wavefield and applies a forcing function according to thesis equations 3.29 and 3.30 but without the second order terms), I will do the linear forward problem by finite differencing the nonlinear forward problem $\mathbf{f}(\mathbf{x})$, i.e.

$$(\partial \mathbf{f}/\partial \mathbf{x})\mathbf{x} \;=\; \mathbf{A}\mathbf{x} \;=\; \mathbf{f}(\mathbf{x}_0+\mathbf{x})-\mathbf{f}(\mathbf{x}_0) \;+\; O^2(\mathbf{x}) \;=\; \Delta\mathbf{f} \;+\; O^2(\mathbf{x}) \quad .$$

I found that the velocity perturbations $\mathbf{x}$ must be less than about one percent of the background velocity to ensure the O^2 terms are small.

Figure B.9 was computed using the elastic finite difference program and Figure B.10 is a plot of the same data as Figure B.9 but at 1/1000-th the clip. Random noise can be seen at the bottom of the time axis in Figure B.10. Thus, the linearized forward problem is accurate to about three figures. Note that the forward problem $\mathbf{f}(\mathbf{x}_0)$ has a direct wave that is about 100 times stronger than the reflections shown in Figure B.9 so my finite difference method to calculate $\mathbf{f}(\mathbf{x}_0)$ has five figure accuracy as expected from case 3 of the previous section.

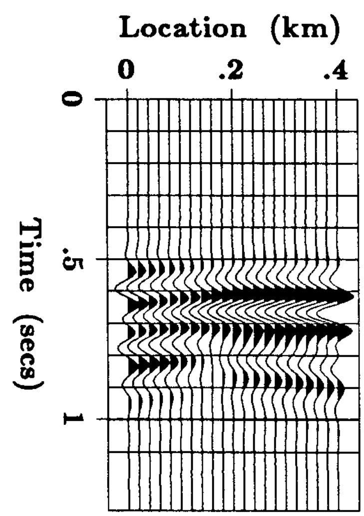

Figure B.9: The linearized forward problem **Ax**.

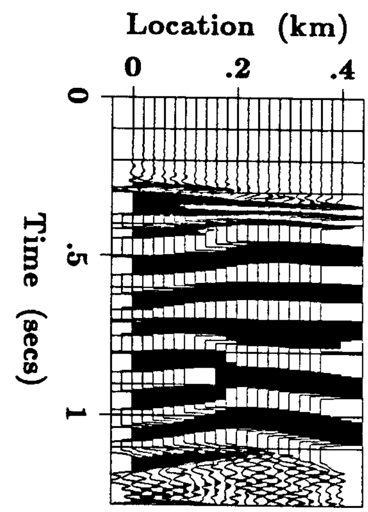

Figure B.10: The linearized forward problem **Ax** shown in Figure B.9 boosted by a factor of 10^3.

B.3.3 Numerical result

Several random P-wave velocity, S-wave velocity and density models were generated on a 40x40 grid. They were filtered to the seismic bandwidth and used in dot product tests in the same manner as was done for the 1D tests in the previous section. A typical dot product result

$$\mathbf{x}^T \mathbf{A}^T \mathbf{y} = 1.0036 \quad ,$$

$$\mathbf{y}^T \mathbf{A} \mathbf{x} = 1.0000 \quad .$$

Therefore, considering the precision parameter for the elastic dot product test is about 10^{-3}, the dot product test is passed and the adjoint implementation is verified. This is not an accurate verification because the linearized forward problem has only three figure precision. However, the dot product test does say that the level of noise in the gradient due to an inaccurate implementation is at most 10^{-3}. This is easily adequate for inversions of real seismic data which typically contain seismic noise levels around 50% or more.

B.4 Conclusions

The dot product test is the numerical equivalent of the mathematical definition of an adjoint and is useful to check adjoint implementations. Tests using a 1D constant velocity acoustic program show the dot product test has a precision parameter related to the level of accumulated roundoff in the forward problem. This parameter depends on the size of the forward problem as well as the details of the calculations.

The implementation of the linearized elastic forward problem used to test the 2D elastic adjoint operation has a roundoff noise level of 10^{-3} implying an expected dot-product precision of 10^{-3}. A numerical test verifies that the elastic adjoint implementation passes the dot product test to the expected precision. This 10^{-3} precision implies that the implementation noise in the adjoint can be at a level no greater than 10^{-3}. This is well below the usual 50% seismic noise level so, from a practical standpoint, this dot product test is a satisfactory verification of the adjoint implementation.