Accurate finite-difference derivative operators by inversion

Joe Dellinger and *Peter Mora*

INTRODUCTION

The wave equation solved by the method of finite differences has numerical dispersion due to inaccuracies in the discrete derivative operator. Typically, derivative operators are obtained by Taylor's expansions which lead to schemes that are very accurate near zero frequency but decrease in accuracy for higher frequencies. One alternative method to obtain a derivative operator (proposed by Jon) is by solving for an operator of a specified length that minimizes the numerical grid dispersion. This can in principle be done using the conjugate gradient algorithm provided the adjoint operation can be calculated. The adjoint operator depends on the two spaces involved. In the case of normal seismic inversion, the spaces are the seismic data space and the space of physical parameters. Here the spaces are the seismic data space and the derivative operator space. The inversion will minimize the square error between the desired solution (without grid dispersion) and the solution obtained by using the finite difference scheme. Once a finite-difference representation of the derivative operator that minimizes grid dispersion for some range of velocity models has been found, it can be used again and again with different distributions of physical parameters. Possible problems with this scheme are nonlinearities leading to local minima and constraining the inversion to be stable.

THEORY

Introduction to adjoint operators

Consider a linearized forward problem

$$\delta d = D\delta m. \tag{1}$$

From the definition of the adjoint operator $< y, Ax > \; = \; < x, A^*y >$ it is clear that the adjoint operation is

$$\delta \hat{m} = D^* \delta d. \tag{2}$$

The "^" signifies that this is the dual space and does not have the same units as the original model space. It has instead the inverse units and can be made to have the same units by applying the inverse Hessian or an approximate inverse Hessian to scale the parameters appropriately.

In order to obtain the adjoint of the wave equation with respect to the derivative operator we must have a linearized wave equation of form (1). Then, the Frechet kernel (the elements of D) can be identified and the adjoint expression can be easily written (as was equation (2)).

The linearized forward problem

Consider the acoustic wave equation

$$\ddot{p} - \partial_{xx} p = f. \tag{3}$$

The velocity is assumed to be unity here to avoid cluttering the equation with unnecessary terms.

Writing the second derivative operator as a convolutional operator c we obtain

$$\ddot{p} - c * p = f. \tag{4}$$

Perturbing the pressure and convolutional operator yields

$$(\ddot{p} + \delta\ddot{p}) - (c + \delta c) * (p + \delta p) = f. \tag{5a}$$

Expanding, ignoring second order terms and reordering yields

$$(\ddot{p} - c * p - f) + \delta\ddot{p} - c * \delta p = \delta c * p. \tag{5b}$$

The parenthesized term is identically zero (see (4)) and so we have

$$\delta\ddot{p} - c * \delta p = \delta f, \qquad \text{where} \qquad \delta f = \delta c * p. \tag{6}$$

Using Green's functions to solve this wave equation we have

$$\delta p(x_r,t) = \int dx\ G(x_r,t;x,0) *_t \delta f(x,t), \tag{7a}$$

where $*_t$ is used to denote convolution over time. The Green's function $G(x_r,t;x,0)$ is the impulse response of the wave equation (4) when a shot is fired at location x at time $t=0$. The wavefield is observed as a function of receiver location x_r and time t.

Substituting the definition of δf into (7a) yields

$$\begin{aligned}
\delta p(x_r,t) &= \int dx\ G(x_r,t;x,0) *_t \Big[\delta c(x) *_x p(x,t)\Big] \\
&= \int dx\ G(x_r,t;x,0) *_t \int dX\ \delta c(X) p(x-X,t) \\
&= \int dX \Big[\int dx\ G(x_r,t;x,0) *_t p(x-X,t)\Big] \delta c(X) \\
&= \int dX\ K(x_r,t,X)\delta c(X).
\end{aligned} \tag{7b}$$

This has the form of the linearized forward problem (1) and defines the Frechet kernel $K(x_r,t,X)$.

The adjoint expression

The adjoint expression is clearly (see (7b), (1) and (2))

$$\begin{aligned}
\delta\hat{c}(X) &= \int dx_r \int dt\ K(x_r,t,X)\delta p(x_r,t) \\
&= \int dx_r \int dt \Big[\int dx\ G(x_r,t;x,0) *_t p(x-X,t)\Big] \delta p(x_r,t).
\end{aligned}$$

Applying the identity $\int dt f(t) * g(t)h(t) = \int dt f(-t)g(t) * h(-t)$, we continue with

$$\begin{aligned}
&= \int dx_r \int dt \Big[\int dx\ p(x-X,-t)G(x_r,t;x,0) *_t \delta p(x_r,-t)\Big] \\
&= \int dt \int dx\ p(x-X,-t)\Big[\int dx_r\ G(x,t;x_r,0) *_t \delta p(x_r,-t)\Big],
\end{aligned} \tag{7c}$$

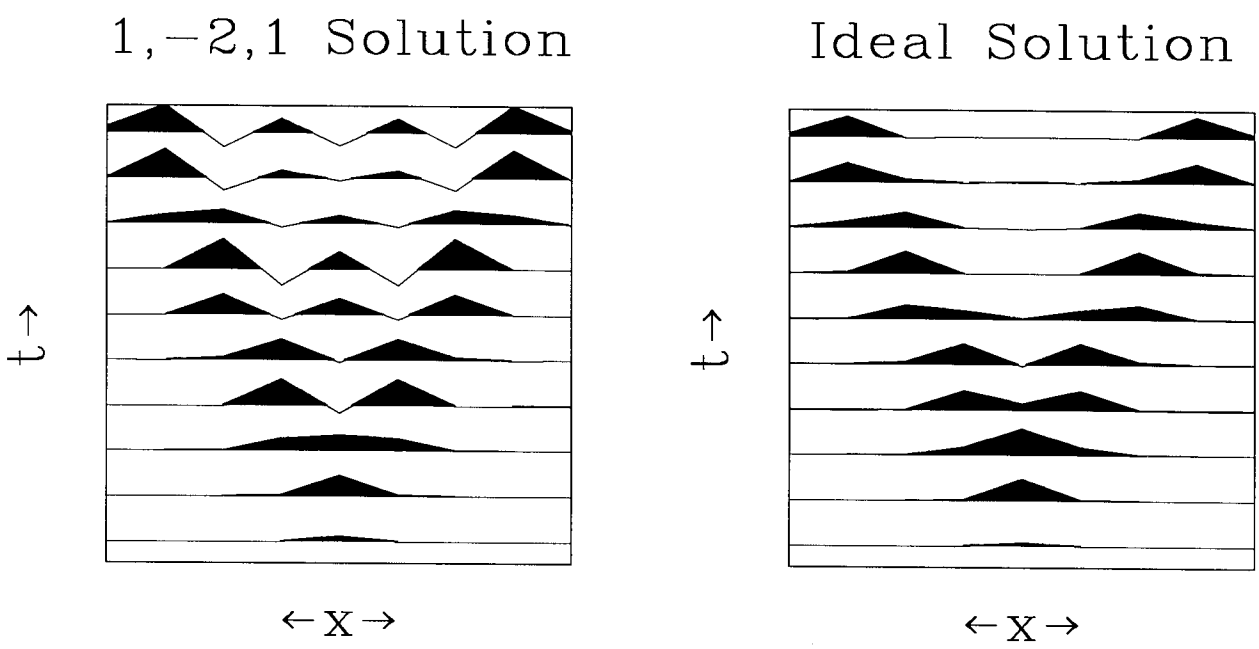

FIG. 1. Right: The "ideal solution", which our inversion scheme will attempt to converge on. Left: The standard three-point second derivative operator solution.

using the spatial reciprocity of G at this last step. Making the definition

$$\psi(x,t) = \int dx_r G(x,-t;x_r,0) *_t \delta p(x_r,t), \tag{8a}$$

we finally obtain for the adjoint expression

$$\delta\hat{c}(X) = \int dt \int dx\ p(x-X,t)\psi(x,t). \tag{8b}$$

The wavefield p is simply the wavefield that is calculated by finite differences when the forcing function f is applied. The wavefield ψ is the wavefield produced by doing finite differences applying the residual wavefield δp as a forcing term where the propagation is now carried out in negative time (i.e. δp is *back propagated* from time $t = t_{\max}$ to $t = 0$).

A SIMPLE EXAMPLE

We will now illustrate the preceding theory with a very simple example. Our forcing function will consist of a positive unit spike centered on the x axis at time $t = 0$, followed by a negative spike at the same position 2 time units later. Such a source is rich in the high frequencies that display the worst numerical dispersion. The time and spatial sampling rates will both be set to 1 everywhere, and the velocity to .4. For simplicity, we will restrict our convolutional representation of the spatial second derivative to a length of three points.

The "ideal solution" which our inversion will attempt to converge on was generated by oversampling the forcing function by a factor of 5, doing the finite differences with the usual 3-point second derivative operator $(1, -2, 1)$, and subsampling the resulting wavefield back onto the original grid. This wavefield is shown on the right-hand side of Figure 1. Note that on such a coarse grid it is impossible to perfectly represent a spike moving with a non-integral velocity.

The "standard" finite-difference three-point second derivative operator is the series $(1, -2, 1)$. If this operator is used, the left side of Figure 1 results. Note the severe numerical dispersion. Our inversion will attempt to find the 3-point convolutional operator that results in a wavefield as close to the "ideal solution" as possible when used on this coarse grid.

Various ways to descend

For this simple problem, we have only to search a three-dimensional space (the three coefficients of the second derivative operator). By starting with a symmetric operator which then stays symmetric during the descent, we reduce this further to a two-dimensional space (the two outer coefficients and the single center coefficient). A contour plot of this space is shown in Figure 2, along with steepest descent paths starting from various initial points. Out of the 10 different initial points, half of the runs do not find the global minimum at $(1.6, -3.1, 1.6)$, and instead settle in one of many of the small local minima which are present.

The nonlinear conjugate-gradient algorithm (see Figure 3) has even worse troubles. Linear conjugate gradients determines a direction and a magnitude to move for the next iteration. In a nonlinear problem, this calculated magnitude is likely to have the wrong scale, and so three different magnitudes are tried, a parabola is fitted through these points, and the distance to the minimum of the calculated parabola is used. This method reduces to ordinary conjugate gradients for a linear problem. If at any point conjugate gradients picks a poor direction due to some local nonlinearity, then the parabola fitting step will pick a zero (or nearly zero) magnitude in that direction as being best, and the descent will stop. This problem can be corrected by checking for this condition, and temporarily reverting back to steepest descents to get the direction to proceed in.

Linear conjugate gradients (see Figure 4) by consistently choosing too large a distance to move at each iteration avoids the smaller local minima by "bouncing out" of them. Since it is expected that the desired minima will always be much larger in size than the others, some such method may also prove useful for higher dimensional cases as well.

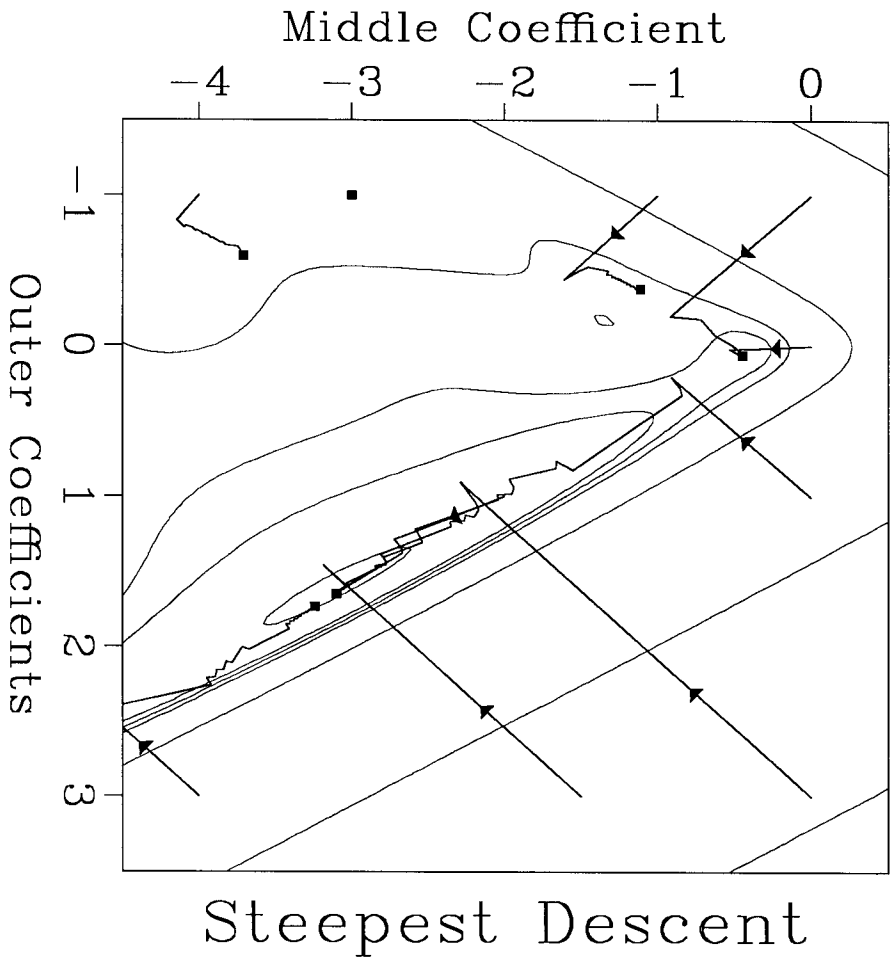

FIG. 2. Contour plot of the mean square error between the computed and desired wavefield as a function of the 3 coefficients of the second derivative operator used to do the finite-differencing, with superimposed steepest descent paths starting from various initial points. The end of each descent path is marked by a square. The contours are *not* at regular intervals. The contours were chosen so as to show the direction of slope of the surface everywhere.

Results

Figure 5 shows the calculated wavefields at various steps of the inversion for two different runs, the ones marked by a script "L" and "R" in Figure 4. The left half of the figure shows an unsuccessful run which gets stuck in a local minimum. The right half of the figure shows a successful run which finds the global minimum. Comparing the lower-rightmost plot in Figure 5 with the left plot in Figure 1, one can see that the inversion has determined a better operator than the standard $(1, -2, 1)$.

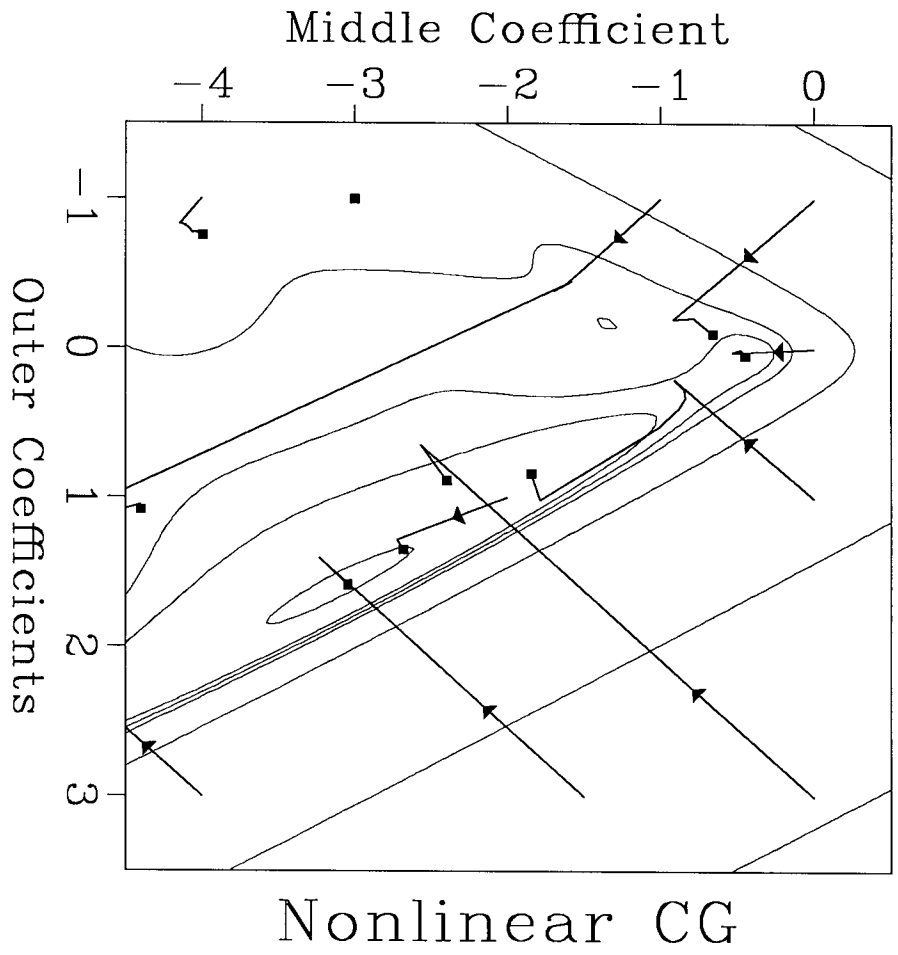

FIG. 3. As Figure 2, but the nonlinear conjugate gradient algorithm was used to do the descent.

Future directions

The global minimum in Figure 4 shows a pronounced elongation from the lower left to the upper right. This "groove" in fact lies exactly on the line given by $\sum c_i = 0$. It makes sense include constraints such as this one in the inversion from the very beginning. The theory of incorporating constraints has been worked out but no tests have yet been made. This will be the subject of a future paper.

CONCLUSIONS

In a regular mesh such as an evenly spaced one dimensional mesh, the convolutional

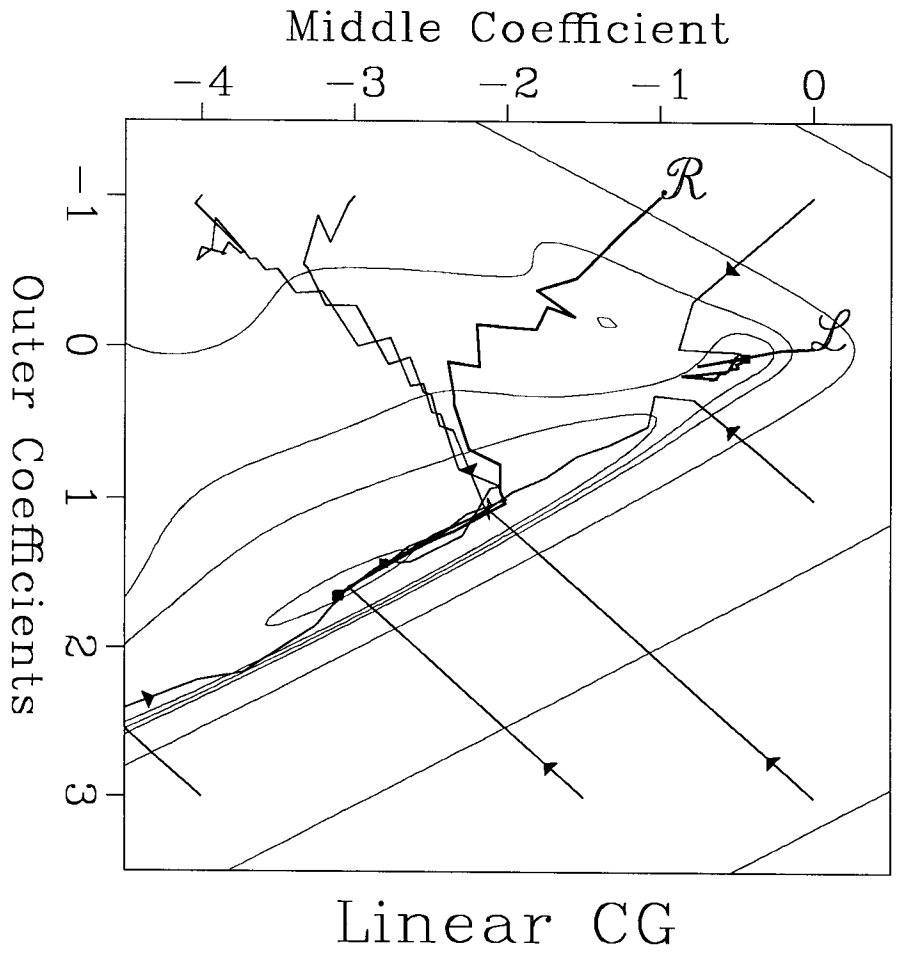

FIG. 4. As Figure 2, but the linear conjugate gradient algorithm was used to do the descent. The script "R" and "L" refer to the right and left hand sides of Figure 5, respectively.

operator $c \approx \partial_{xx}$ can be obtained by applying the conjugate gradient algorithm using the adjoint expression (8). The adjoint requires *only two* finite difference calculations, one to obtain p and the other to obtain ψ.

Chuck's problem of finding operators for an icosahedral grid (Sword, 1986) may involve more finite difference calculations because the mesh is irregular and many different differencing stars are required. Just as Chuck does not need to recalculate the shape of his grid anew each time he puts in a different source, an appropriate operator need only be calculated once for each mesh point in an irregular grid, and after that it can be used unchanged.

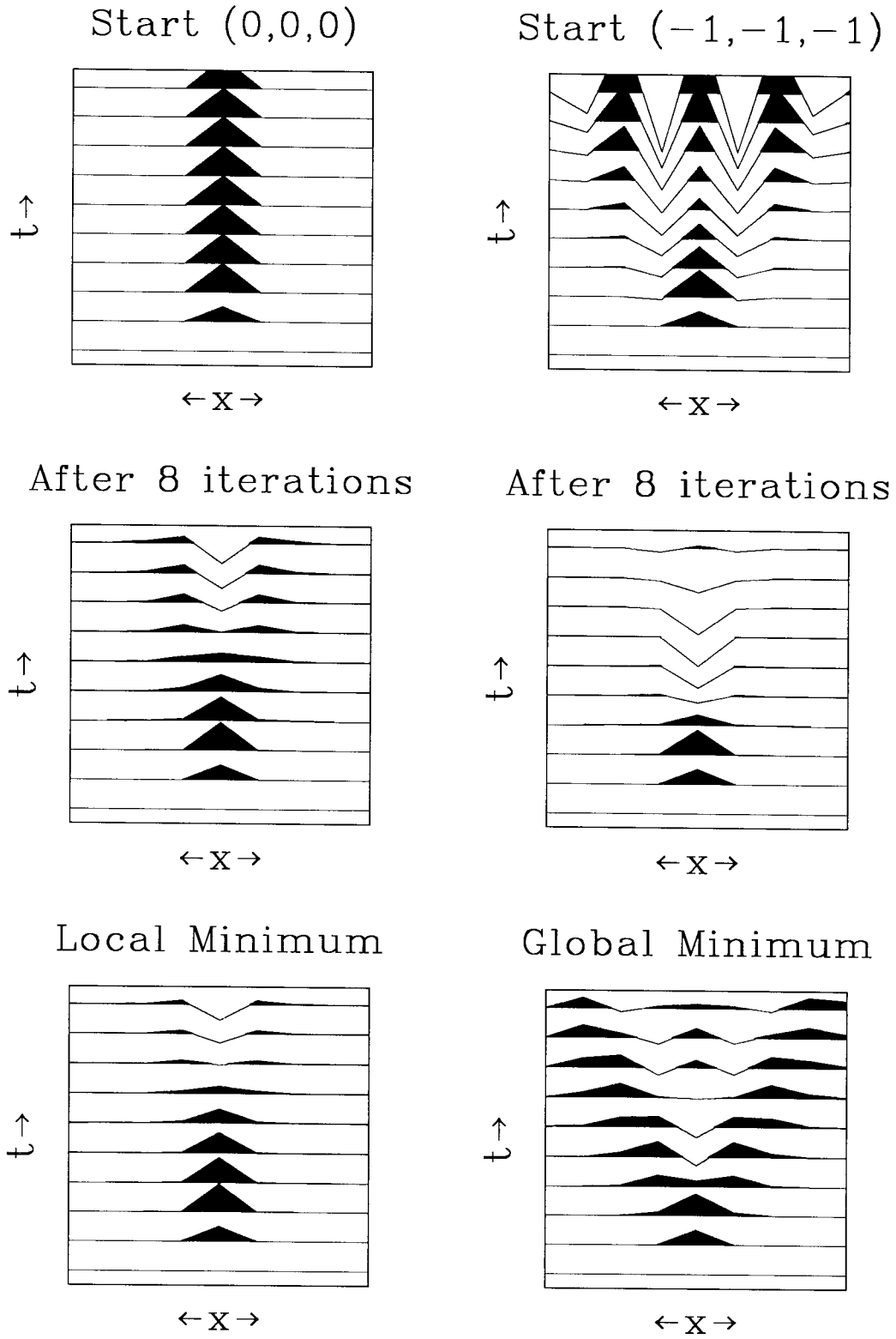

FIG. 5. Wavefields produced at the beginnings, middles, and ends of two different inversion runs.

REFERENCES

Sword, C., 1986, Finite element propagation of acoustic waves on a spherical shell: SEP-50.

APPENDIX

Proof of relationship $\int dt\ f(t) * g(t)h(t) = \int dt\ f(-t)g(t) * h(-t)$

$$
\begin{aligned}
&\int dt\ f(t) * g(t)h(t) \\
&= \int dt \int d\tau\ f(\tau)g(t-\tau)h(t) \\
&= \int d\tau\ f(\tau) \int dt\ g(t-\tau)h(t) \\
&= \int dT\ f(-T) \int dt'\ g(T-t')h(-t') \\
&= \int dt\ f(-t) \int dt'\ g(t-t')h(-t') \\
&= \int dt\ f(-t)g(t) * h(-t)
\end{aligned}
$$

Q.E.D.