Residual statics estimation by simulated annealing: field data example

Daniel Rothman

INTRODUCTION

I present an example of residual statics estimation for seismic data from the Wyoming Overthrust belt. The statics were estimated by *simulated annealing* (Rothman, 1984a). I used an adaptation of the *heat bath* method, which I describe elsewhere in this report (Rothman, 1984b).

THE DATA

The data I present here were the subject of a previous study by Johnson et al. (1983). Using data from the Wyoming Overthrust belt, Johnson et al. illustrated an intriguing example of a statics solution in the presence of severe lateral velocity anomalies in the near surface. Because the residual static shifts for these data are as large as 300 msec., conventional statics algorithms will fail due to poorly picked spatial correlations ("cycle-skips" or "leg-jumps"). Johnson et al. estimated their statics by visual interpretation of the data, and obtained a "hand-statics" solution. The example I present here is an early attempt at the automatic estimation of these large statics.

Figure 1a is a 24-fold stack, and Figure 1b shows common midpoint gathers 34 and 64. Both the stack and the gathers are displayed without static corrections. The data were collected with a 48-trace, split-spread cable. The group interval is 220 feet, and there is a 4 receiver group gap between the shot and the near traces. The source was Vibroseis, with an 8-55 Hz. sweep. The data in Figures 1a,b have undergone the following processing steps: (1) predictive deconvolution; (2) bandpass filtering, from 8-35 Hz.; and (3) normal-moveout corrections. One stacking velocity function was used for the entire line. No field static corrections were made. The cablelength extends over approximately 100 stacked traces, which is about 60% of the section. Both ends of the line exhibit the usual roll-on and roll-off, so the first and last 24 stacked traces are less than

24-fold.

PROCESSING PARAMETERS

Statics were estimated from the data between 2.9 and 3.9 seconds, thus concentrating on the prominent reflector at approximately 3.5 seconds. Static shifts were constrained to fall within ± 160 msec., in 8 msec. increments, for both shots and receivers. Using the heat bath method outlined in Rothman (1984b), estimates of statics were iteratively drawn from probability distributions constructed from the exponentiation of crosscorrelation functions. Thus the probability of choosing the value σ for a given static shift is

$$P(\sigma) = \frac{e^{\frac{-\phi(\sigma)}{T}}}{\sum_{\sigma} e^{\frac{-\phi(\sigma)}{T}}} ,$$

where ϕ is the normalized crosscorrelation of stacked traces against unstacked traces.

The temperature parameter T was chosen with the following objectives in mind. First, the starting temperature T_0 needs to be high enough to destroy the structure present in the input data; because the correct statics solution is assumed to yield a stack far different from the input stack, the destruction of the input stack removes any bias toward making only small, incremental changes. Also, the final estimate should be independent of where the algorithm starts. Once the character of the stack is destroyed, T can then quickly drop to a predetermined minimum, $T_{\min}$. $T_{\min}$ is chosen to be high enough so that local minima are escaped, but low enough so that convergence can occur only in the deepest (global) minimum, or in minima that are nearly as deep. This assumes a simple model in which the objective function, stack power, has a multitude of shallow minima, but only few relatively deep minima. Thus far this appears to be a viable model for data with large statics and moderate noise contamination.

The temperature function has the form

$$T_k = \begin{cases} \alpha^k T_0 , & \alpha^k T_0 > T_{\min} \\ T_{\min} , & \alpha^k T_0 \leq T_{\min} \end{cases}$$

where k is the number of iterations, and α, T_0, and $T_{\min}$ are input parameters. For the statics solution illustrated here, $\alpha = .99$, $T_0 = .045$, and $T_{\min} = .0265$. Choosing α and T_0 require just a few very quick tests. $T_{\min}$ is determined only after a fair amount of experimentation; 4 previous full-length test runs with different values for $T_{\min}$ were

necessary before I obtained the result shown here. It appears that $T_{\min}$ needs to be chosen correctly within a few percentage points - otherwise the algorithm will either never converge, or it will converge to a horrendous looking local minimum.

RESULTS

Figures 2a-e chart the progress of the statics estimation algorithm at different points in the iterative process. Each of these figures is a 24-fold stack that is performed after corrections are made with the current estimate of the statics. Figure 2a displays the stack after 5 iterations; we see that the choice of a high T_0 led to immediate obliteration of all spatial coherence in the stack. Figure 2b is the stack after 1000 iterations. By the 50th iteration, $T = T_{\min}$, but after 1000 iterations the stack still exhibits no obvious improvement over the result in Figure 2a. By iteration 1125, however, convergence begins; this is illustrated in Figure 2c. The algorithm then rapidly descended into a minimum, as is evident in Figure 2d, which is the stack after 1250 iterations. Figure 2e is the final solution, achieved after 1665 iterations. It should be compared with Figure 1a, the stack of the input data. We see now that not only has the deep reflector become continuous across most of the section, but also that the static corrections have revealed an interesting thrust fault in the more shallow data.

Figure 3 shows the final estimates of shot and receiver statics. The static shifts mostly vary smoothly, and extend the full range within ± 160 msec.

Figure 4 shows the same two gathers in Figure 1b; they are now shown after static corrections have been performed. In gather 34, the upward dips at far offsets have now been flattened. Gather 64, 3300 feet to the right, has had its downward dips also flattened.

Figure 5 is a graph of stack power as a function of iteration number. Stack power is computed only within the computation window (2.9-3.9 seconds), and the power of the input is normalized to one. Power quickly decreases to about .5, and does not begin to rise until after about 1000 iterations. After about 1150 iterations there is a sharp increase, with convergence finally occurring by about iteration 1400. The stacks produced by the remaining iterations were roughly equivalent except for behavior at the far right. The final solution, shown in Figure 2e, has a stack power of 1.471.

If one expects that the objective function, negative stack power, contains few or no local minima, then one may sequentially choose the static shifts that yield the greatest crosscorrelation coefficient, for each shot and receiver. This is optimization by *iterative improvement.* I ran a test of iterative improvement on the data of Figure 1, with

processing parameters identical to those used to generate the stacks in Figure 2. The result is shown in Figure 6. All reflections have been enhanced, but the poor stack in the region between traces 25 and 75 still remains. This "cycle-skipping" is most evident for the reflector at about 3.5 seconds, which should be continuous. The stack power for this result is 1.395. Although this is not much lower than the annealing result in Figure 2e, there is a substantial difference in the interpretation one might make from this section.

CONCLUSIONS AND REMARKS

When performing statics estimation by simulated annealing, the initial structure in the stack is irrelevant. If a potential user thinks that he or she can somehow produce a stack prior to statics estimation that is "close" to the stack that would be produced after static corrections, then he or she should try all the conventional (linear) techniques first. Simulated annealing allows one to start from nothing and still get something - simply compare Figures 2b and 2e for conclusive proof of this statement. One must be willing to start from nothing, however. If so, then the user may be justly rewarded, because the technique provides a general solution that is valid, in principle, for any degree of noise contamination or severity of statics.

The statics estimated here were allowed to be only integer multiples of 8 msec., which was the sampling interval of the input data. This coarse discretization will probably miss the most optimal (non-discretized) solution, but not by much. I would recommend using the output of a simulated annealing run as the input to a conventional, least-squares traveltime decomposition algorithm to clean up any residual errors. The most efficient use of simulated annealing would have the algorithm only locate the region of the power maximum. Other, more efficient algorithms may then take over.

Inevitably one must somehow gauge the reliability of the statics solution that any algorithm produces. The best way to justify a statics solution is to demonstrate that the stack after the application of statics has improved not only within the computation window, but also outside it. In this example, the clear appearance of the thrust fault in the first 3 seconds of the data lends credence to the solution obtained from the data between 2.9 and 3.9 seconds. Two aspects of the solution should be viewed with suspicion, however. First, the structural trend of the deep reflector appears to have changed the direction of its dip. In the solution of Johnson et al., this reflector is flat. The correct answer, of course, is probably unknown, but one thing is sure: statics solutions are only reliable for spatial wavelengths less than about a cablelength (here, this spans about 100 stacked traces). Any interpretation of the data must bear this fact in mind.

The second cause for suspicion is the solution at the far right side of the section: the deep reflector loses its continuity. Examination of the statics in Figure 3 shows that the general trend at the right end of the line might actually require statics greater than 160 msec., which was the upper limit for the statics in this test. Thus the 160 msec. limit might have been too small. In any event, it should also be noted that statics solutions lose some reliability at the ends of lines, because the cdp multiplicity decreases.

The last 250 iterations produced stacks that differed slightly. Ordinarily I would recommend choosing the stack with the greatest power; in this example, however, I elected to choose the stack that was most plausible, from a geologic viewpoint. The only real differences occurred at the very right end. This of course raises the question of when the algorithm should stop. The simplest convergence criterion would have the algorithm stop after a fixed number of iterations.

ACKNOWLEDGMENTS

I thank Jim Johnson of Conoco for his generous help in providing the Wyoming Overthrust data.

REFERENCES

Johnson, J.H., 1983, Application of a statics solution, Wyoming Overthrust: in Seismic expression of structural styles, v. 3, A.W. Bally, ed., p. 3.4.1-35.

Rothman, D.H., 1984a, Nonlinear inversion, simulated annealing, and residual statics estimation: SEP 41.

Rothman, D.H., 1984b, Monte Carlo techniques: an overview: SEP 41.

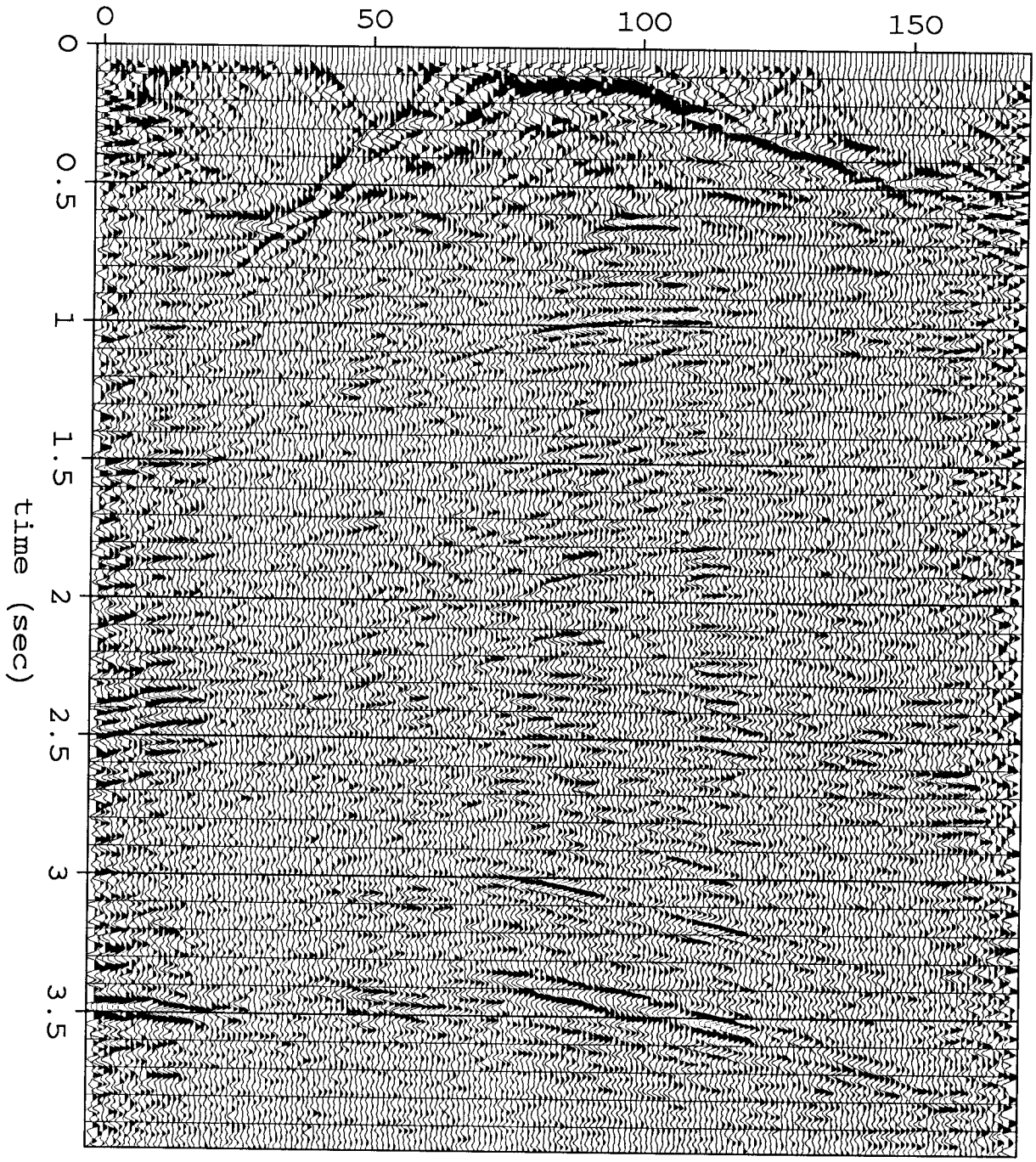

FIG. 1a. 24-fold stack of data from the Wyoming Overthrust belt; the stack is performed prior to statics estimation. One time-variable velocity function was used for the entire line. The data used for residual statics analysis are between 2.9 and 3.9 seconds. The strong reflections at the near surface are roughly indicative of the near-surface velocity variations. The first and last 24 traces are underfold due to the usual roll-on and roll-off.

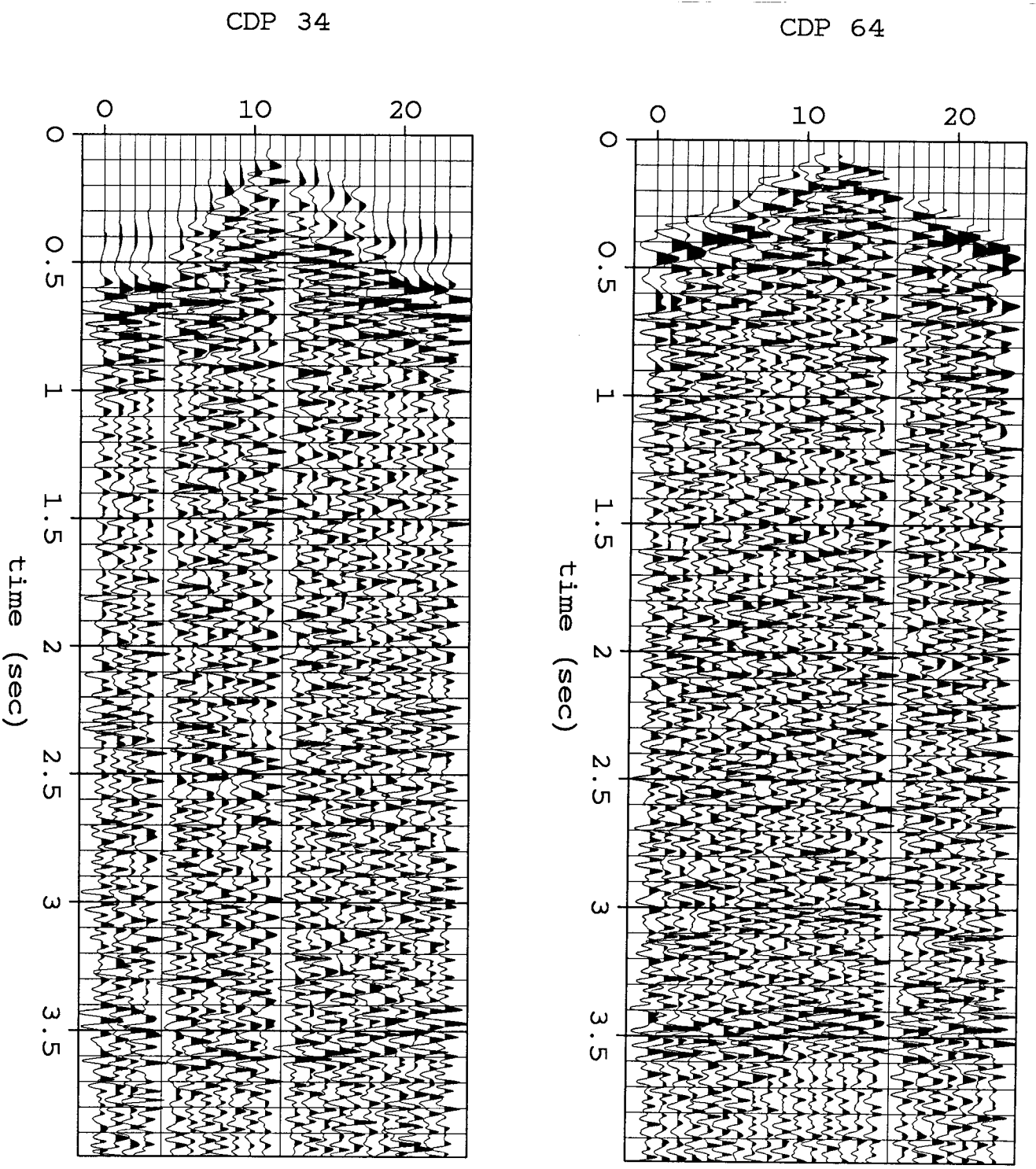

FIG. 1b. NMO-corrected common midpoint gathers 34 and 64. Offset increases in each plot from the center outward. The near-surface velocity anomalies have produced dipping structure in events that should be flat; this is most evident in the data near 3.5 sec. In gather 34, dip appears to bend upward with offset. In gather 64, just 3300 feet and about a third of a cablelength down the line, dip now appears to bend downward with offset.

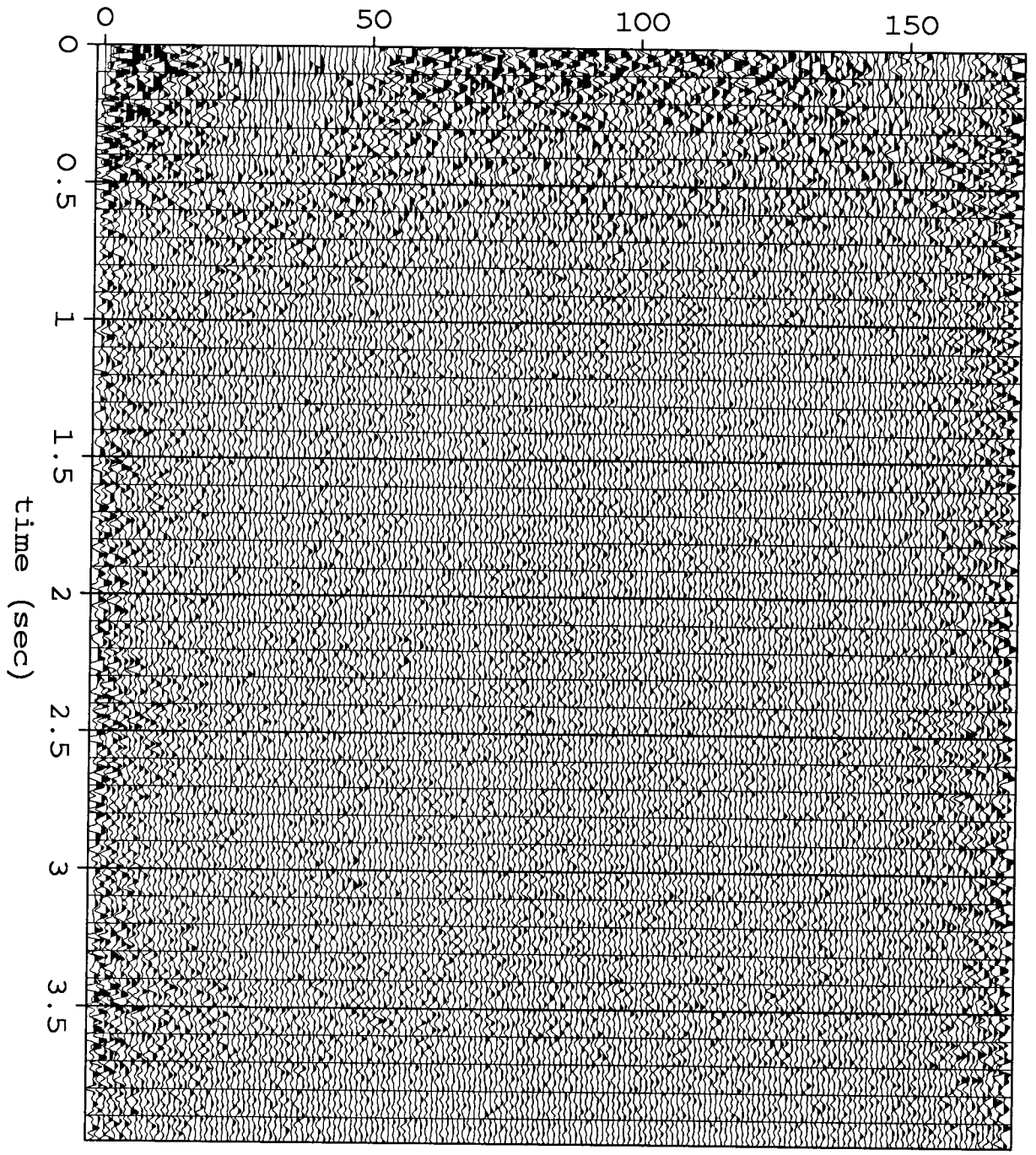

FIG. 2a. Stack after 5 iterations of the statics estimation algorithm. T_0 was chosen high enough so that all reflection events are now obliterated. This removes any tendency for the algorithm to make only incremental improvements in the stack, as would be expected with a linearized technique. Amplitudes are greater at the sides due to the low cdp multiplicity at the ends of the line.

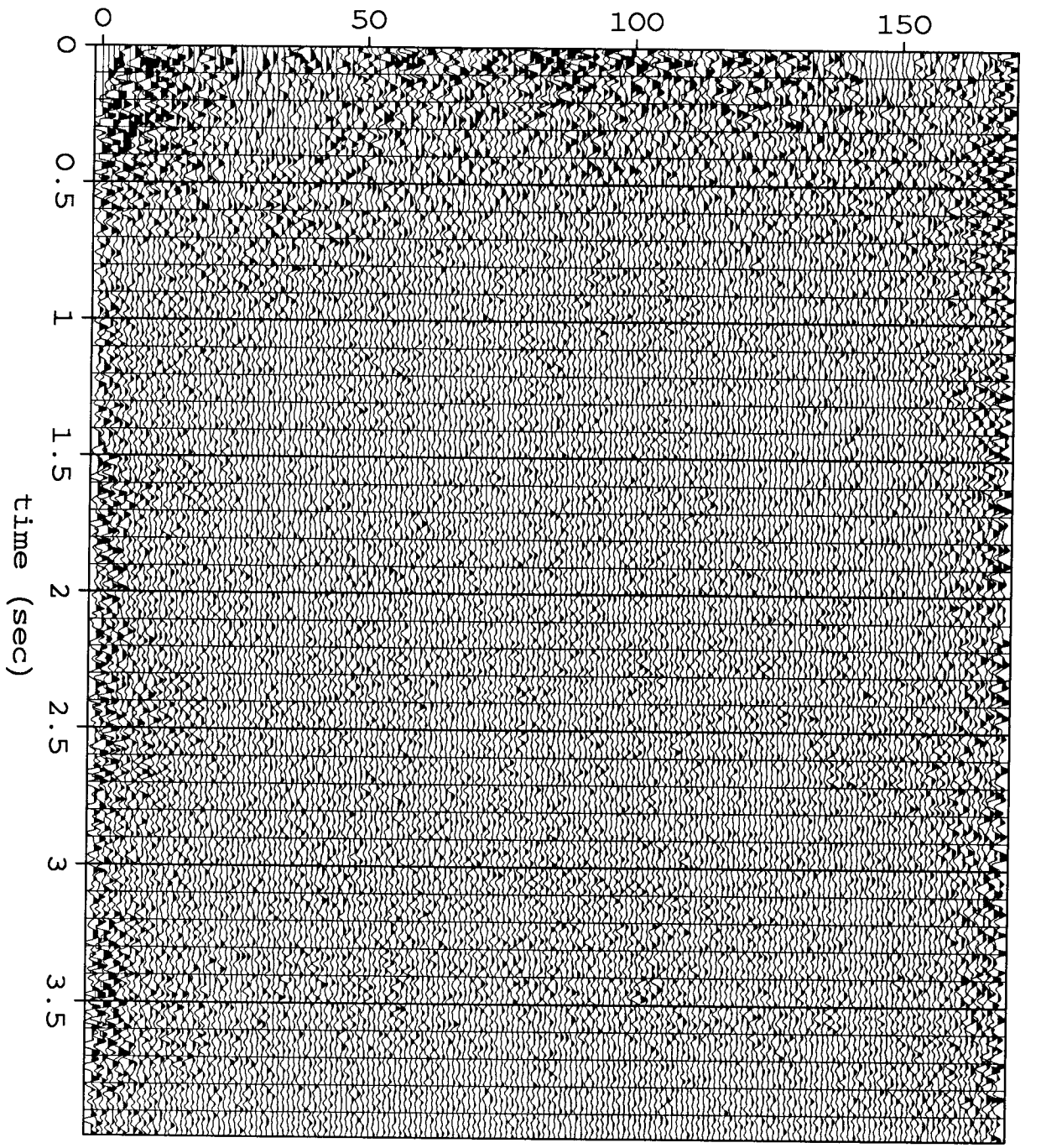

FIG. 2b. Stack after 1000 iterations of the statics estimation algorithm. T is held constant from iteration 50 onward. No appreciable differences are evident when comparing this with the result after 5 iterations (Figure 2a).

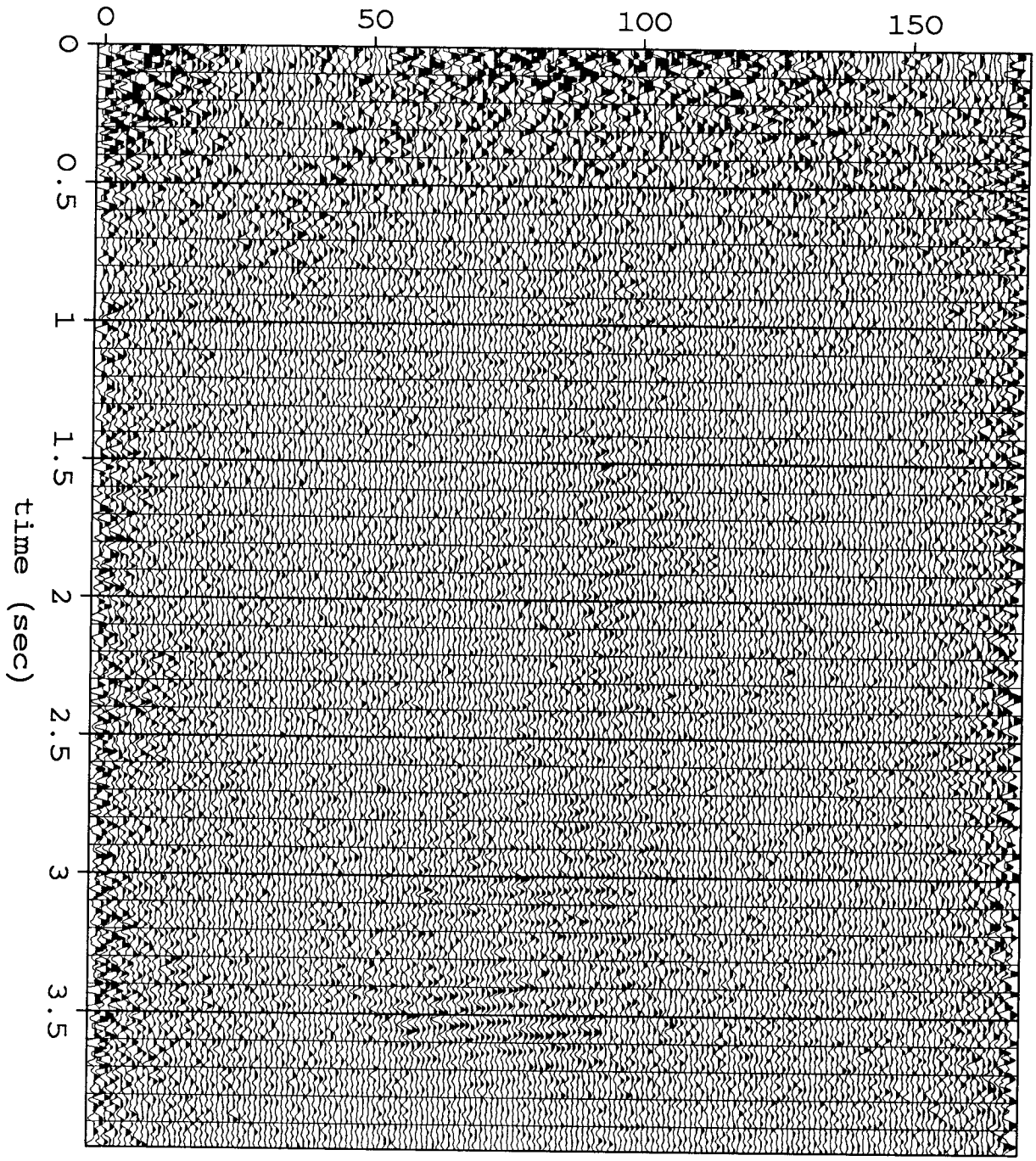

FIG. 2c. Stack after 1125 iterations of the statics estimation algorithm. The faint spatial coherence in the middle of the section shows that convergence is now beginning.

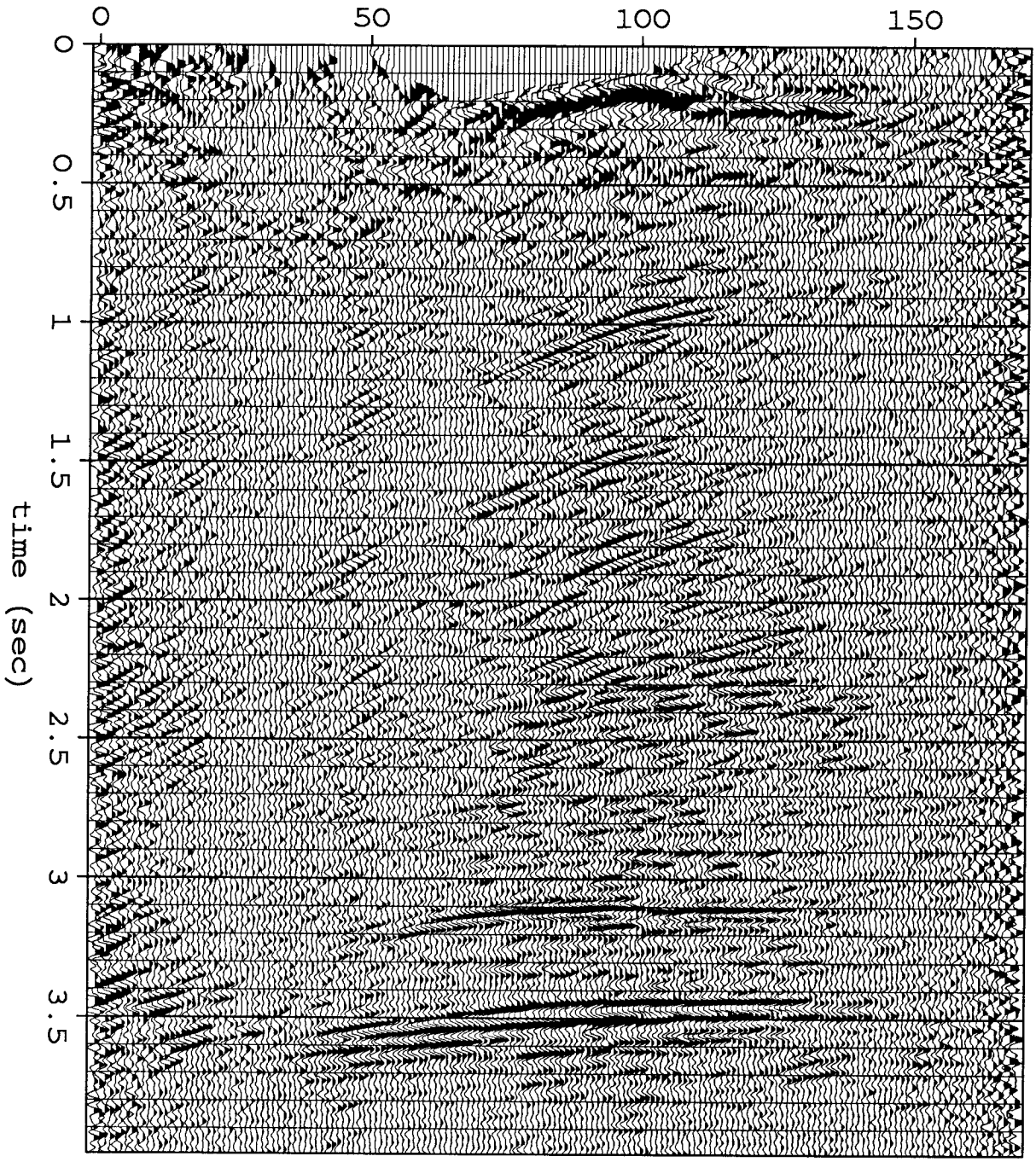

FIG. 2d. Stack after 1250 iterations of the statics estimation algorithm. Convergence is now almost complete, except for the ends of the line.

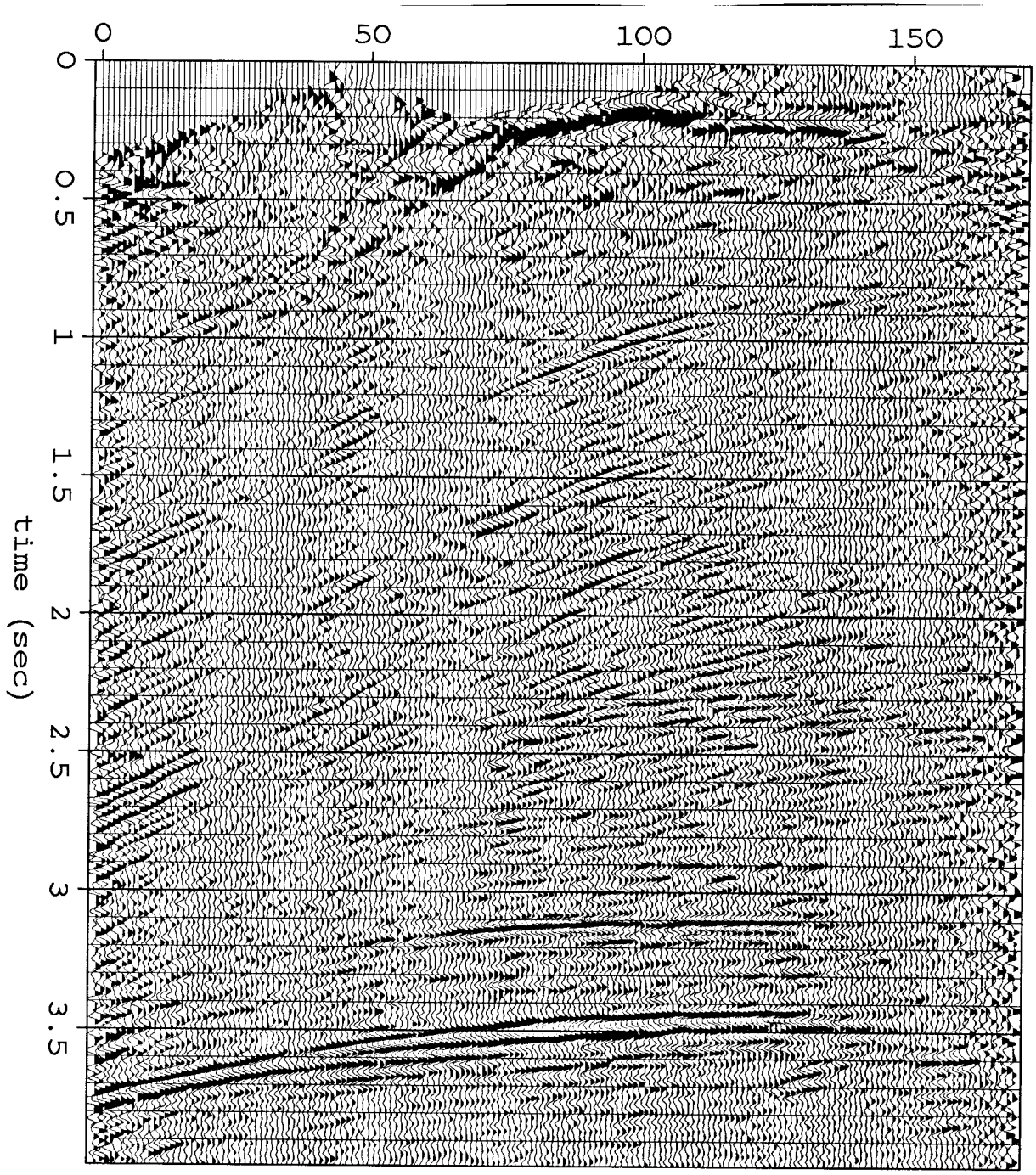

FIG. 2e. Stack after 1665 iterations of the statics estimation algorithm. This is the final solution, and should be compared with the stack of the input data in Figure 1a. Reflections are now continuous throughout much of the section. Moreover, this statics solution has uncovered structure that appears to be a thrust fault dipping upward from left to right.

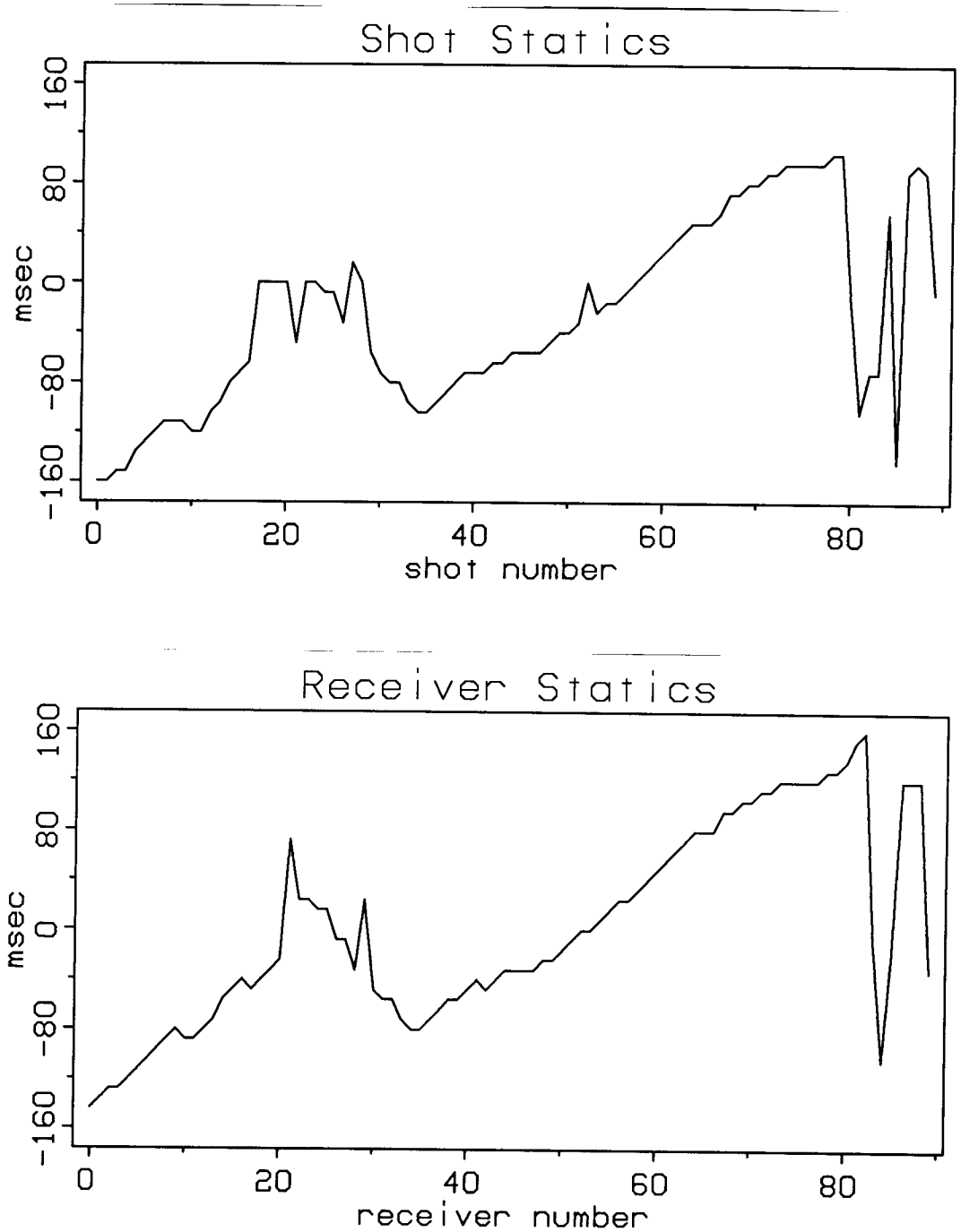

FIG. 3. Final estimates of shot and receiver statics. Shots 17-20 and shot 24 were skipped; these are plotted as zeroes. Both shot and receiver statics are as large as 160 msec, so bulk static shifts of individual traces are as large as 320 msec.

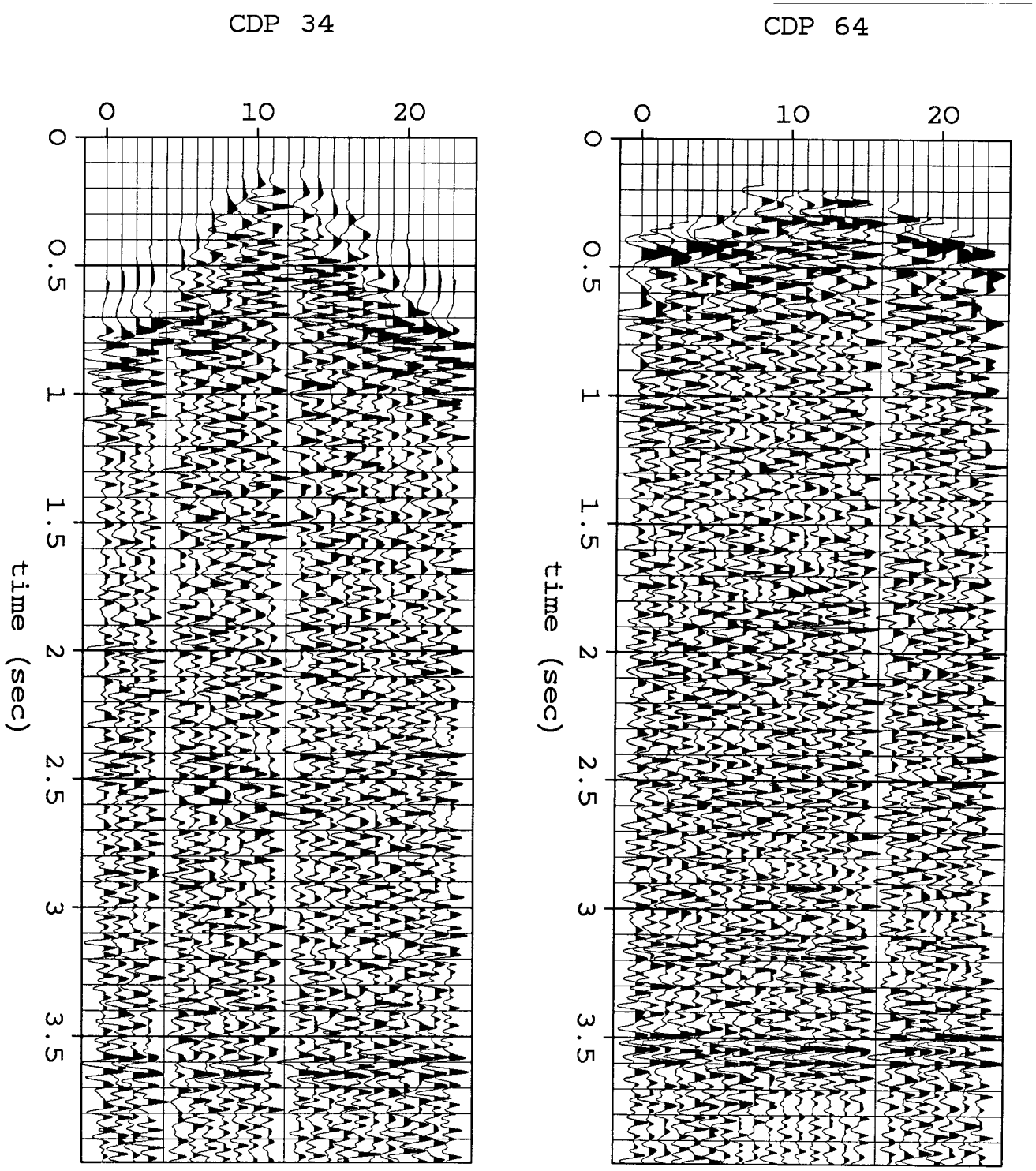

FIG. 4. The same two common midpoint gathers in Figure 1b, now shown after static corrections have been made. Both show substantial flattening after application of the statics solution.

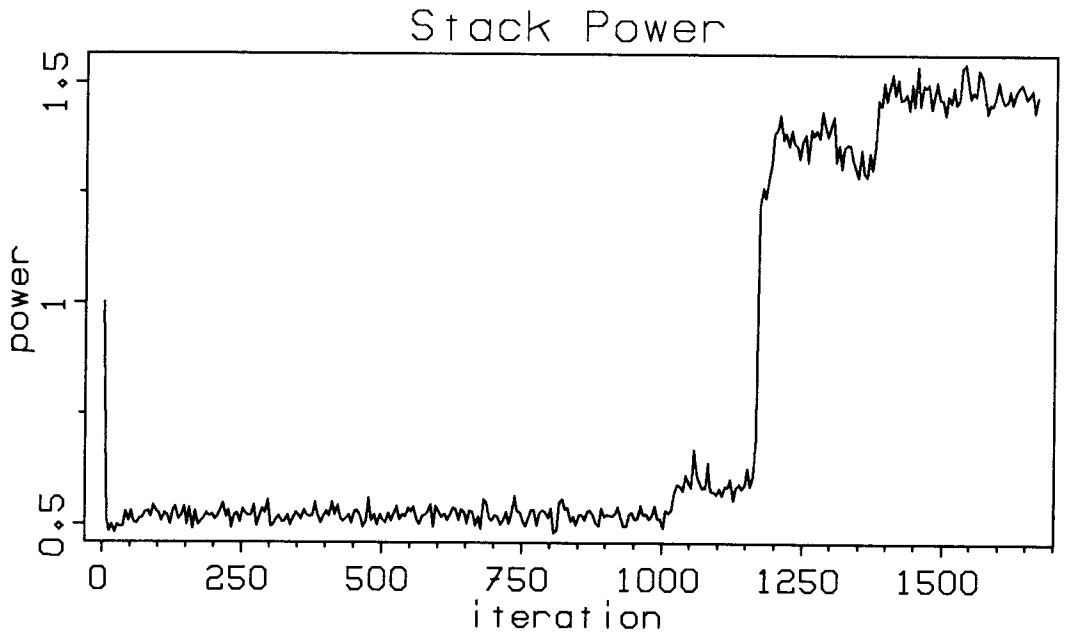

FIG. 5. Stack power as a function of iteration number for the test illustrated in Figures 2a-e. The input stack power is normalized to 1. Power is computed every 5 iterations within the computation window, which extends from 2.9 to 3.9 sec. Power initially decreases quickly to about .5. Temperature decreases from .042 to .0265 during the first 50 iterations; thereafter it remains constant. Convergence begins after about 1000 iterations. A sharp increase in power occurs after about 1150 iterations; this rapid change is analogous to crystallization. Global convergence occurs after about 1400 iterations. The power of the final solution displayed in Figure 2e is 1.471. Power is occasionally greater during the last 200 iterations, but the solution in Figure 2e was chosen because of its superior appearance at the far right end of the stack.

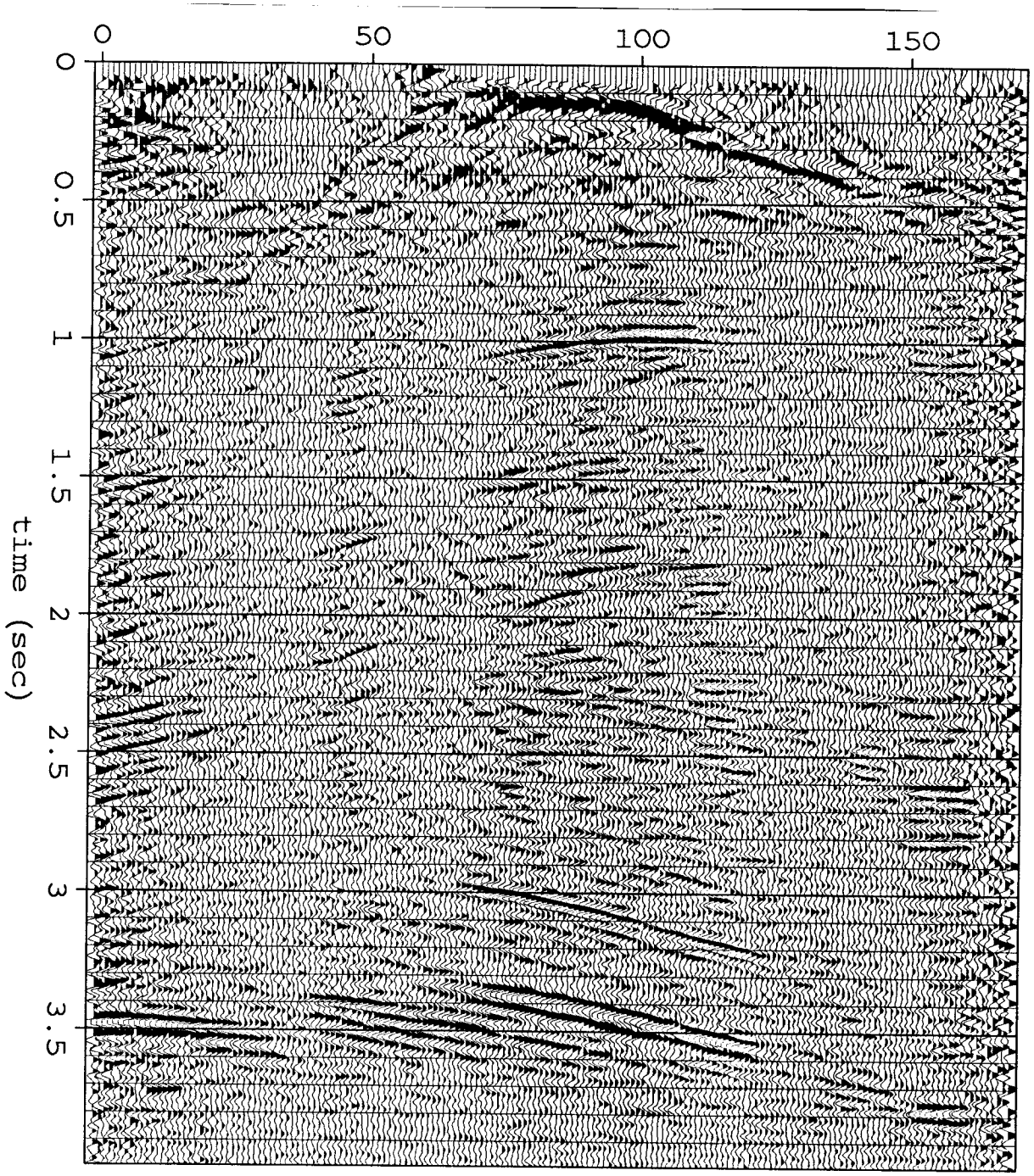

FIG. 6. Stack after application of a statics solution obtained by iteratively choosing the best value for each shot and receiver static until convergence to a maximum. Convergence occurred after only 13 iterations. This should be compared with the solution obtained by simulated annealing, shown in Figure 2e. Since this technique of statics estimation converges to the nearest power maximum, cycle skipping can be a serious problem if the true statics are large. This is evident in the solution here, especially in the region near 3.5 seconds, which should exhibit a continuous reflector.