Newton trace balancing II

Stewart A. Levin

SYNOPSIS

In SEP-42 I discussed the problem of trace balancing and derived a Newton iteration scheme for decomposing (possibly signed) balancing factors into surface consistent components. As with residual statics estimation, there are both free and poorly constrained parameters which cause the Newton equations to be poorly conditioned. I tried to handle this problem by means of constraint equations and penalty terms with unsatisfying results. I found that a naive "Gauss-Seidel" scheme gave comparable results with significantly less computation and concluded more study was needed.

MORE STUDY

I went to the literature, guided by suggestions of Gene Golub and Michael Saunders, and studied current methods for nonlinear least squares. Following Dennis (1977) let me define a framework in which to fit the various methods. Nonlinear least-squares tries to minimize a functional of the form

$$1/2 \quad || f ||^2 \qquad . \tag{1}$$

Setting its derivative to zero produces

$$J^T f \quad = \quad 0 \qquad . \tag{2}$$

The second derivative of the function may be written

$$J^T J \ + \ K \tag{3}$$

where

$$K \quad = \quad \sum_i f_i \nabla^2 f_i \qquad . \tag{4}$$

The Newton equations

$$(J^T J \ + \ K) \, \delta x \quad = \quad -J^T f \tag{5}$$

determine the location of a zero of the quadratic approximation to the functional obtained by truncating a Taylor expansion after second order.

In the Gauss-Newton method one assumes that K is small compared to $J^T J$ and looks instead at the equations

$$J^T J \, \delta x = -J^T f \qquad (6)$$

which are the normal equations for the classic linear least-squares problem

$$J \, \delta x \approx -f \quad . \qquad (7)$$

Various numerical conditioning techniques are then applied, the most popular being the Levenburg-Marquardt algorithm which appends a scalar multiple of the identity matrix to matrix J to bring it to full rank, and the resulting equations (7) are then solved, typically by numerically stable orthogonal rotations, e.g. the QR method (Golub, 1969), or related sparse matrix techniques, e.g. LSQR (Paige and Saunders, 1982).

The thesis of Nash (1982) outlines several other quasi-Newton techniques for general minimization all of them characterized by precisely solving approximate versions of the Newton equations. These methods are useful when exact second derivatives are not readily available or are very costly to compute.

Nash's thesis deals primarily with *truncated-Newton* methods where approximate solutions are sought for the exact Newton equations. These methods are desirable when exact second derivative information is easily available. Their local convergence properties are discussed by Dembo, Eisenstat and Steihaug (1982). In their analysis they measure the degree of approximation in the solution by the relative error in matching the right hand side gradient vector input to the Newton equations in, for example, equation (5). Linear convergence is obtained so long as this relative error is bounded by some constant less than one and stronger convergence rates hold if this bound trends towards zero sufficiently rapidly. Dembo and Steihaug (1983) selected a forcing sequence in combination with a conjugate gradient linear equation solver to turn Newton's method from a disappointingly slow nonlinear optimization technique into one more than competitive with other state-of-the-art optimization routines. Nash (1982) extended their method to handle rank deficiency.

Specializing now to trace balancing, f is a column vector containing the entries $a_{ij} - s_i g_j$ in some order and J contains nonzero entries g_j in the s_i'th column corresponding to the input amplitudes a_{ij} and vice-versa for the g_j'th column. The gradient $J^T f$ can be rewritten as another natural matrix-vector product

$$\begin{pmatrix} 0 & (SG^T - A)|_\Omega \\ (GS^T - A^T)|_\Omega & 0 \end{pmatrix} \begin{pmatrix} S \\ G \end{pmatrix} \quad . \qquad (8)$$

The product $J^T J$ is the matrix

$$\begin{pmatrix} \mathrm{diag}(\sum_{j \in \Omega_i} g_j^2) & SG^T \mid_\Omega \\ GS^T \mid_\Omega & \mathrm{diag}(\sum_{i \in \Omega_j} s_i^2) \end{pmatrix} \tag{9}$$

and the matrix K is the one that appeared in equation (8):

$$\begin{pmatrix} 0 & (SG^T - A) \mid_\Omega \\ (GS^T - A^T) \mid_\Omega & 0 \end{pmatrix} \quad . \tag{10}$$

In SEP-42 I also suggested a simple algorithm based upon the Gauss-Seidel iteration. This was based upon the algebraic relation

$$\begin{pmatrix} 0 & SG^T \mid_\Omega \\ GS^T \mid_\Omega & 0 \end{pmatrix} \begin{pmatrix} S \\ G \end{pmatrix} = \begin{pmatrix} \mathrm{diag}(\sum_{j \in \Omega_i} g_j^2) & 0 \\ 0 & \mathrm{diag}(\sum_{i \in \Omega_j} s_i^2) \end{pmatrix} \begin{pmatrix} S \\ G \end{pmatrix} \tag{11}$$

which allows me to approximately solve equation (2) by means of perturbations that satisfy

$$\begin{pmatrix} \mathrm{diag}(\sum_{j \in \Omega_i} g_j^2) & 0 \\ 0 & \mathrm{diag}(\sum_{i \in \Omega_j} s_i^2) \end{pmatrix} \begin{pmatrix} \delta S \\ \delta G \end{pmatrix} = -J^T f \tag{12}$$

i.e. a quasi-Newton method. The extremely simple solution of (12) combined with the experience reported in SEP-42 suggest the diagonal matrix might be useful as a preconditioner in the truncated Newton method.

TRYING IT OUT

Using the Vibroseis dataset I worked with for my SEP-41 report, I windowed 256 28-trace CMP gathers centered around the low-velocity sag. The resulting 7168 traces covered 155 shot stations and 156 receiver stations. I then applied John Toldi's dynamic time corrections (SEP-43) to remove the sag. These traces and their trace headers served as input to trace balancing. Figure 1 displays a stack of these data. We see that the time sag has been removed but, despite restimation of stacking velocities, an amplitude anomaly remains.

My first experiment was to tabulate the shot station, receiver station and RMS amplitude of each trace. This was then input to a truncated version of the Newton iteration I described in SEP-42. The modifications were to

1) abandon the additional norm constraint on the solution,

2) switch from LSQR to SYMMLQ (Paige and Saunders, 1975) which works directly with the symmetric Hessian rather than embedding it in a larger symmetric system,
3) solve the Newton step equations with increasing relative accuracy $\eta_{i+1} = \eta_i^2$ with $\eta_0 = 0.5$,
4) solve the homotopy step equations with relative accuracy 0.25, and
5) successfully terminate calculations when the Newton step changed the current estimate of the solution by less than 0.1 percent.

At most six Newton (outer) iterations were permitted before reducing the homotopy step and at most 25 SYMMLQ (inner) iterations were allowed per Newton step. No line search was performed to improve the approximate Newton steps, I relied on the homotopy adjustments to handle any divergence problems.

For these data calculations converged after one homotopy step, six outer Newton iterations and a total of 74 inner iterations. Figure 2 is a stack after trace balancing with these surface-consistent factors. Figure 3 shows the trace amplitudes a_{ij}, the initial fit obtained by averaging over shot or receiver, the final fit and the difference between the trace amplitudes and this final fit. The central low-amplitude zone of has been balanced but the right hand side of the section has been washed out.

Inspecting the RMS amplitudes and the residuals of Figure 3 shows the problem. High amplitudes cluster on the inner offsets, especially on the right side of the line. Further, positive residuals are concentrated on the inner offset and negative residuals on the outer offsets. The model does not fit the data well. This is also reflected in the rather large number of inner iterations. What has happened is the inner offset traces are dominated by low velocity, high amplitude surface waves. This suggests several alternatives. First, a suggestion of Shuki Ronen's, I could try to filter out the unwanted ground roll by discrimination against its lower frequency content. Second, I could add an offset-consistent term to my equations to make the model better match the data. Third, I could make the data better match the model by dividing each amplitude estimate by the average over all traces with the same offset. Fourth, I could compute balance factors using only the outer offsets.

I tried out the first suggestion by filtering out frequencies below 25 Hz before computing trace amplitudes. Trace balance factors were computed as before and then applied to the original, unfiltered traces. The results are shown in Figures 4 and 5. Convergence was significantly improved (32 inner iterations) and the residuals are now fairly well distributed throughout the survey. The balance on the right hand side of Fig. 4 is still a bit weak but a definite improvement over Figure 2.

Next I tried preconditioning by dividing the RMS amplitudes by the mean along common offsets prior to surface-consistent decomposition. As before, the trace balance factors were applied to the original traces without additional common-offset balancing. Figures 6 and 7 show the result. The residuals are now well distributed throughout all offsets. Computations were also a little bit faster, taking a total of 65 inner iterations to reach convergence.

For the practical reason that I found a good fit of the model to the preconditioned data I did not tackle the more expensive alternative of adding offset dependency to my model. Also, as there are few data gaps in the common offset direction, I'd expect the offset terms to decouple from the shot and geophone amplitudes in any case. I will, however, in the near future try to fit only the outer offsets, program bugs prevent me from having these available for this report.

Having a reasonable fit, I then worked on improving the Newton iteration. Ideally I'd want to attain convergence with at most, say, a dozen inner iterations in two or three outer iterations. To this end I tried the diagonal preconditioning proposed above. This was accompanied by a relaxation of the inner iteration accuracy requirements during the first few outer iterations. The first try with this "Jacobi" preconditioning flopped. The first Newton iteration diverged and two shorter homotopy steps were required to attain convergence. The problem was that the positive diagonal preconditioning matrix M changed how SYMMLQ measured the relative accuracy of the residual. Instead of the usual formula $(x^T x)^{1/2}$ norms were now computed as $(x^T Mx)^{1/2}$. In extreme cases this could decrease the relative accuracy of the computed Newton step by the ratio of the largest to the smallest diagonal element of M. To compensate, I reran the preconditioning test with all tolerances reduced by a factor of 10. This did the trick and I attained convergence with only 44 inner iterations. Figures 8 and 9 show the outputs.

SUMMARY

Surface-consistent trace balancing can be successfully implemented without resort to logarithms. The truncated-Newton method works for this problem despite the guaranteed rank deficiency of the Newton equations. Preconditioning is needed for reasonable convergence rates. For robustness I still need to incorporate additional safeguards such as insuring that I always select a step producing sufficient descent. Gill, Murray and Wright (1981) and Nash (1982) outline several ways of doing this.

In the larger goal of surface-consistent deconvolution, these results are encouraging. My chief worry is to improve the preconditioning so as to reduce the number of iterations by another factor or 2 or more. Most of the ideas

I've used in this scalar problem carry through to designing convolutional filters. In particular the analog of the diagonal preconditioning I used above is to design a deconvolution filter for each gather independently. In SEP-42 I proposed using this process in an iterative fashion until I reached convergence. I have tried this on the SEP-41 dataset I've used also in this paper and studied the stack after each iteration. I found that the results settled down in about 5 iterations to one of two alternating sections, the first if the last pass was over shots, the other if the last pass was over geophones. By using this as a preconditioning process, the conjugate-gradient and Newton iterations will concentrate on the "cross-feed" between shot and geophone gathers rather than wasting time on deconvolving individual gathers.

REFERENCES

Dembo, R.S., and Steihaug, T., 1983, Truncated-Newton algorithms for large scale unconstrained optimization: Math. Prog., 26, p 190-212

Dembo, R.S., Eisenstat, S.C., and Steihaug, T., 1982, Inexact Newton methods: SIAM J. Numer. Anal., 19, p 400-408

Dennis, J.E., Jr, 1977, Nonlinear least squares: *in* Jacobs, D., *ed.*, The state of the art in numerical analysis, Academic Press, p 269-312

Gill, P.E., Murray, W., and Wright, M.H., 1981, Practical Optimization: Academic Press, 401 p.

Golub, G., 1965, Numerical methods for solving linear least squares problems: Numerische Mathematik, 7, p 206-216

Levin, S.A., 1984, Surface-consistent deconvolution, SEP-41, p 1-26

Levin, S.A., 1985, Newton trace balancing, SEP-42, p 69-79

Nash, S.G., 1982, Truncated-Newton methods: Ph.D. thesis, Stanford Univ., 114 p

Paige, C.C., and Saunders, M.A., 1975, Solution of sparse indefinite systems of linear equations: SIAM J. Numer. Anal., 12, p 617-629

Paige, C.C., and Saunders, M.A., 1982, LSQR: an algorithm for sparse linear equations and sparse least squares: ACM Trans. Math. Software, 8, p 43-71

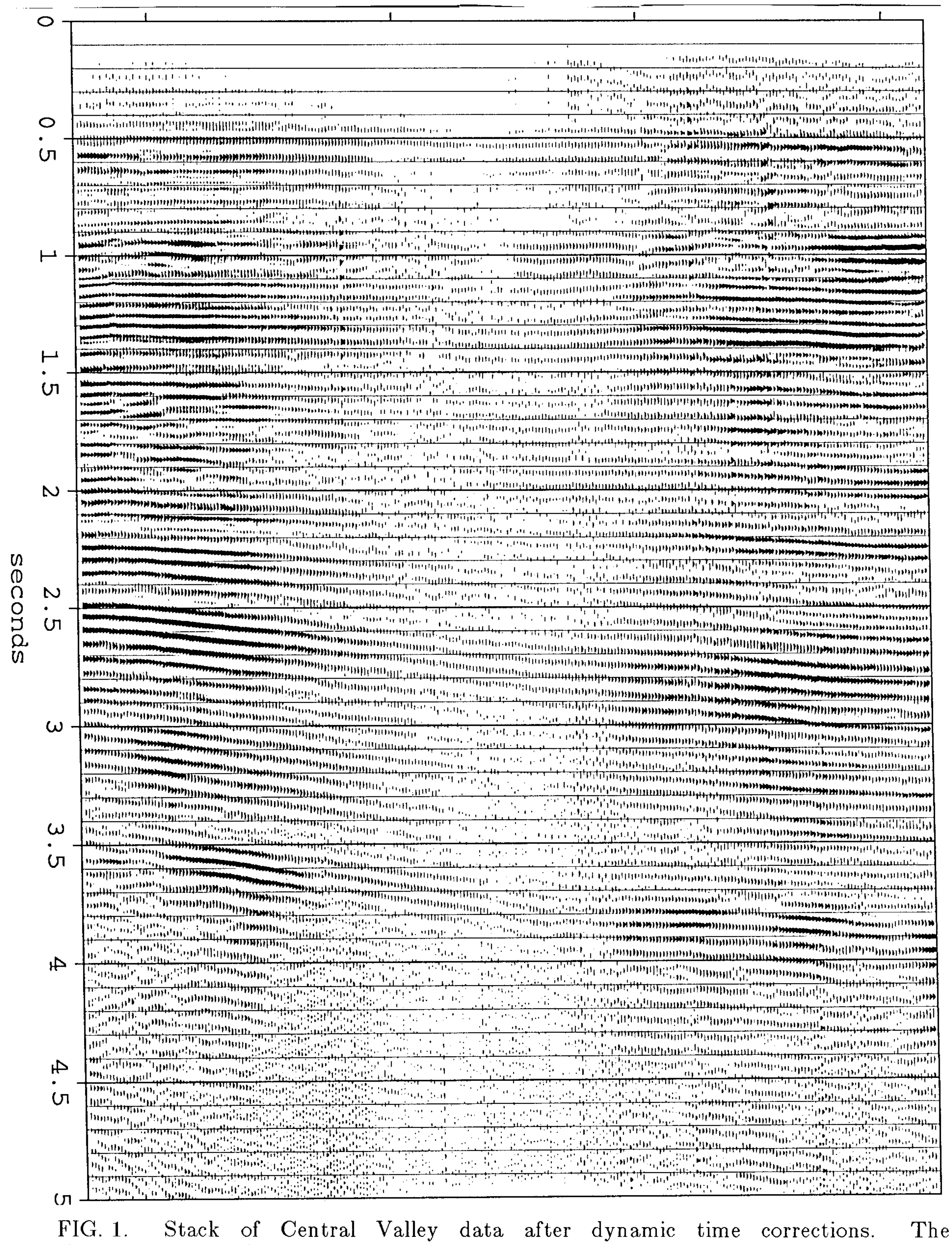

FIG. 1. Stack of Central Valley data after dynamic time corrections. The time corrections have compensated for a low velocity sag in the middle traces but low amplitudes remain. There were 256 28-fold CMP gathers in this dataset. Trace length is 5 seconds sampled at 4 msec. CMP interval is 67.5 feet.

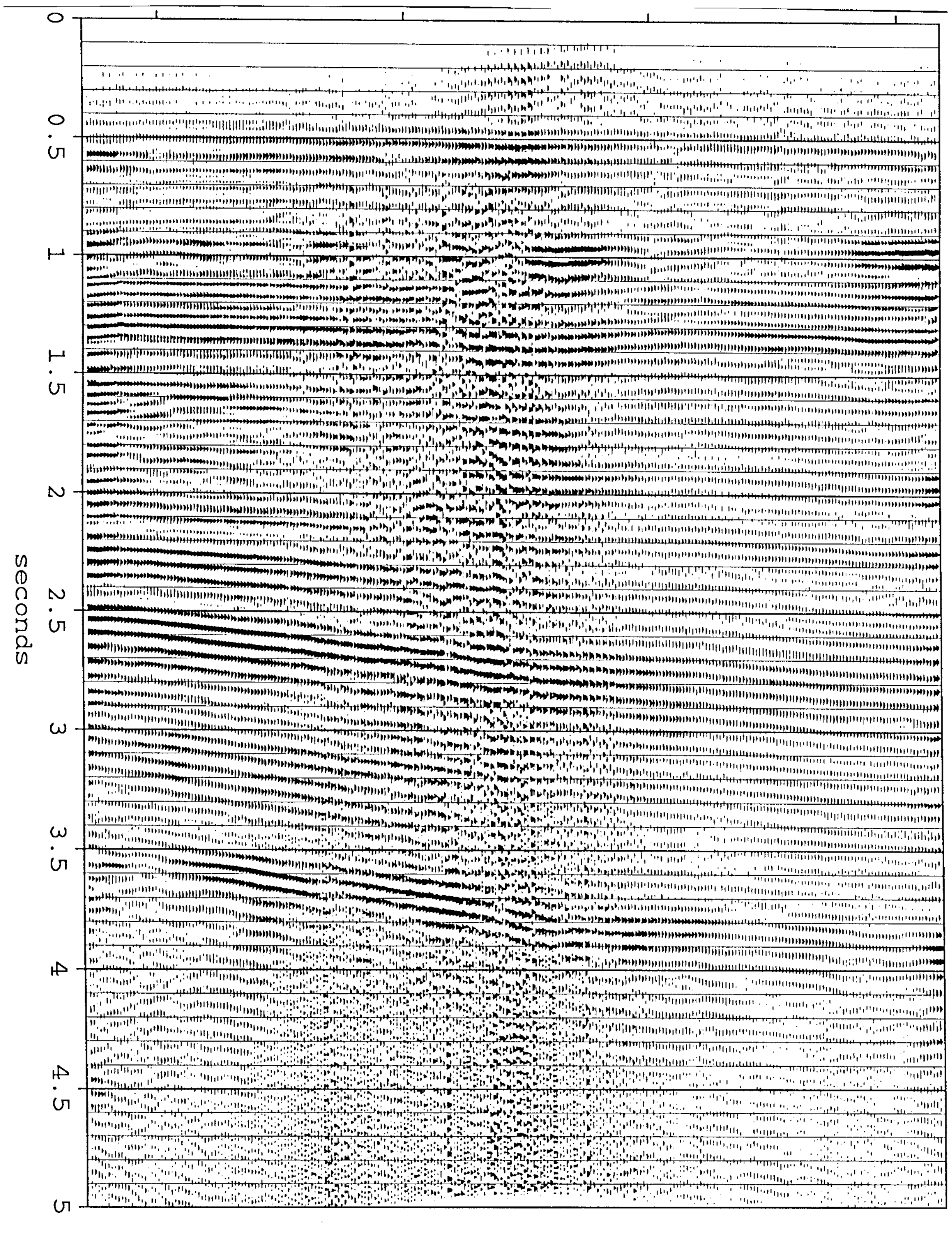

FIG. 2. Stack after surface-consistent amplitude balancing. Compared with Fig 1, the central amplitudes have been gained up O.K. but the right hand traces are improperly scaled down.

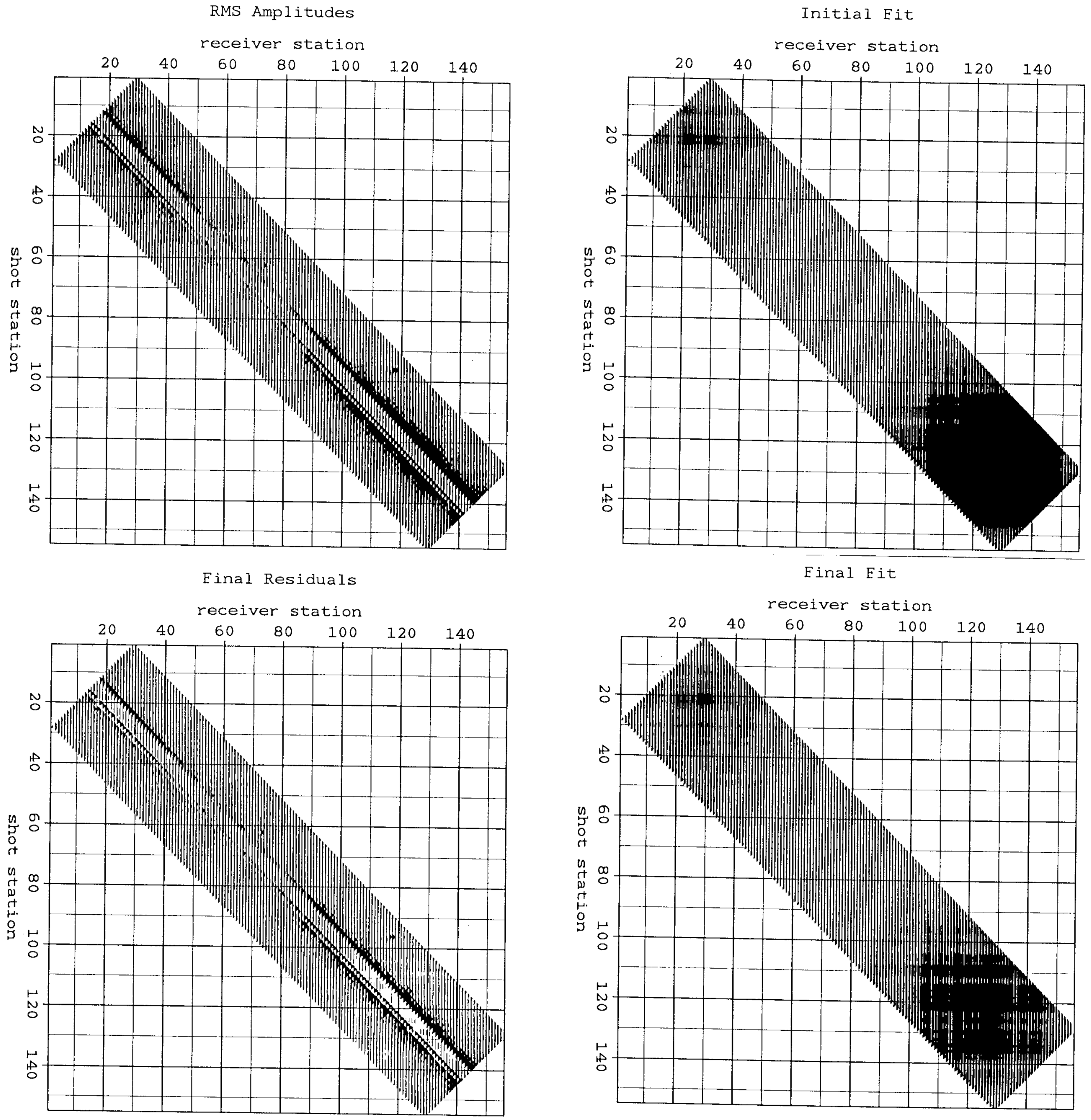

FIG. 3. Trace amplitudes, initial and final fits and the difference between the trace amplitudes and the final fit corresponding to Figure 2. Here we see that high amplitudes concentrated on the inner offsets are causing the least-squares residuals to trend from positive to negative from inner to outer offset. This is undesirable and reflects the offset dependency left out of my model.

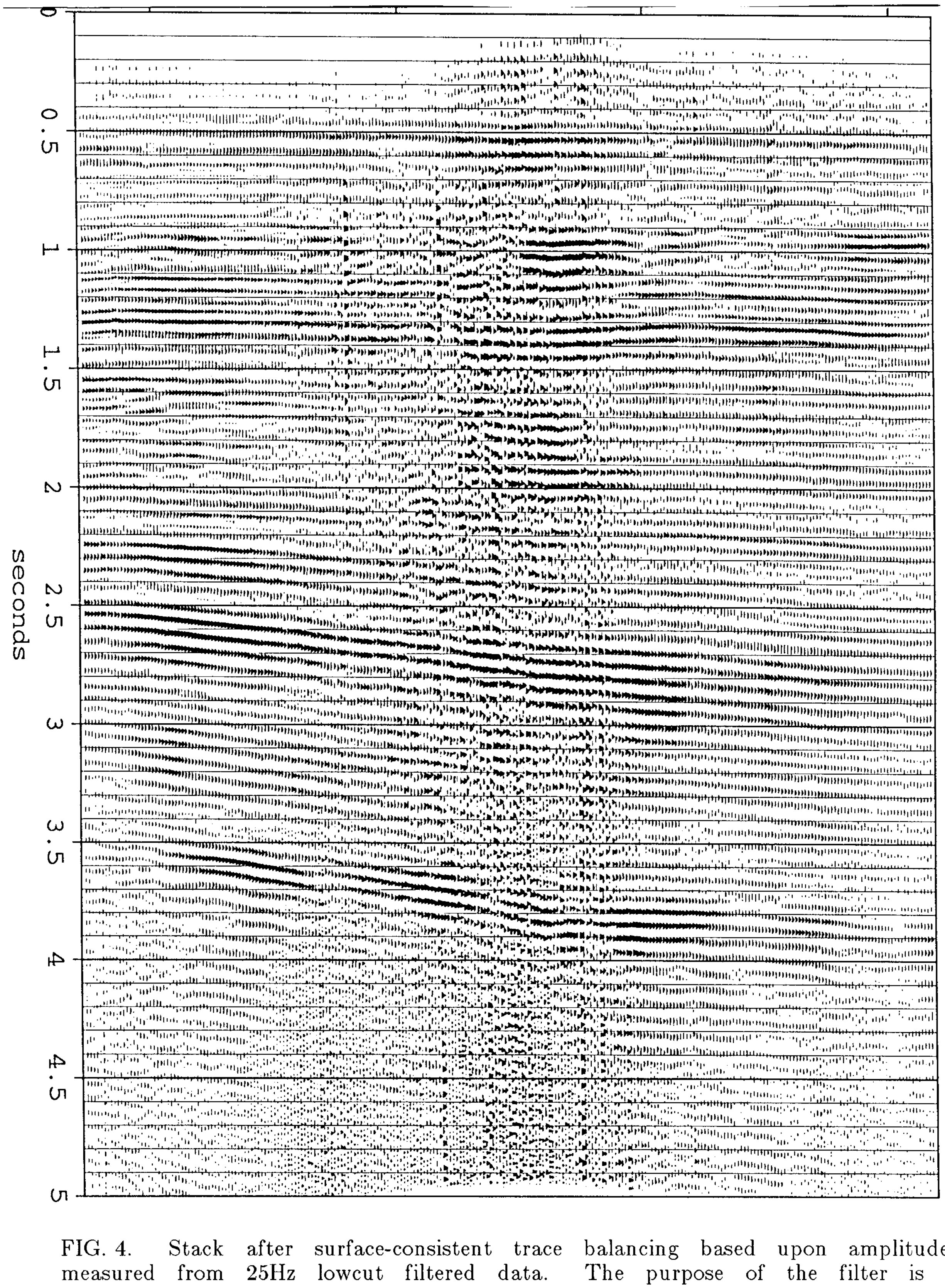

FIG. 4. Stack after surface-consistent trace balancing based upon amplitudes measured from 25Hz lowcut filtered data. The purpose of the filter is to discriminate sharply against low frequency ground roll concentrated on the inner offsets. The gain factors thus computed were applied to the unfiltered input traces to produce this stack.

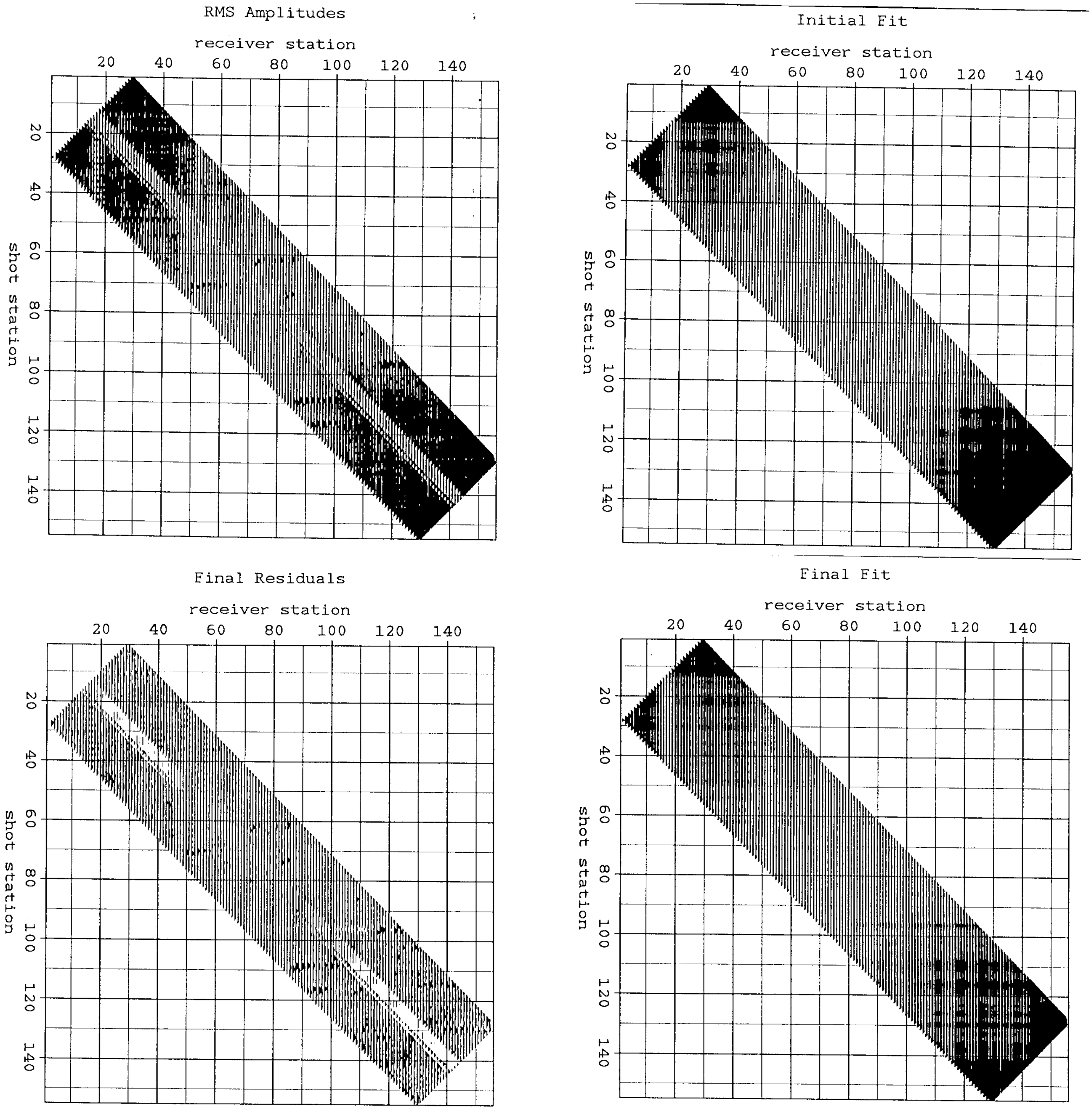

FIG. 5. Trace amplitudes, initial and final fits and the residuals for Fig. 4.

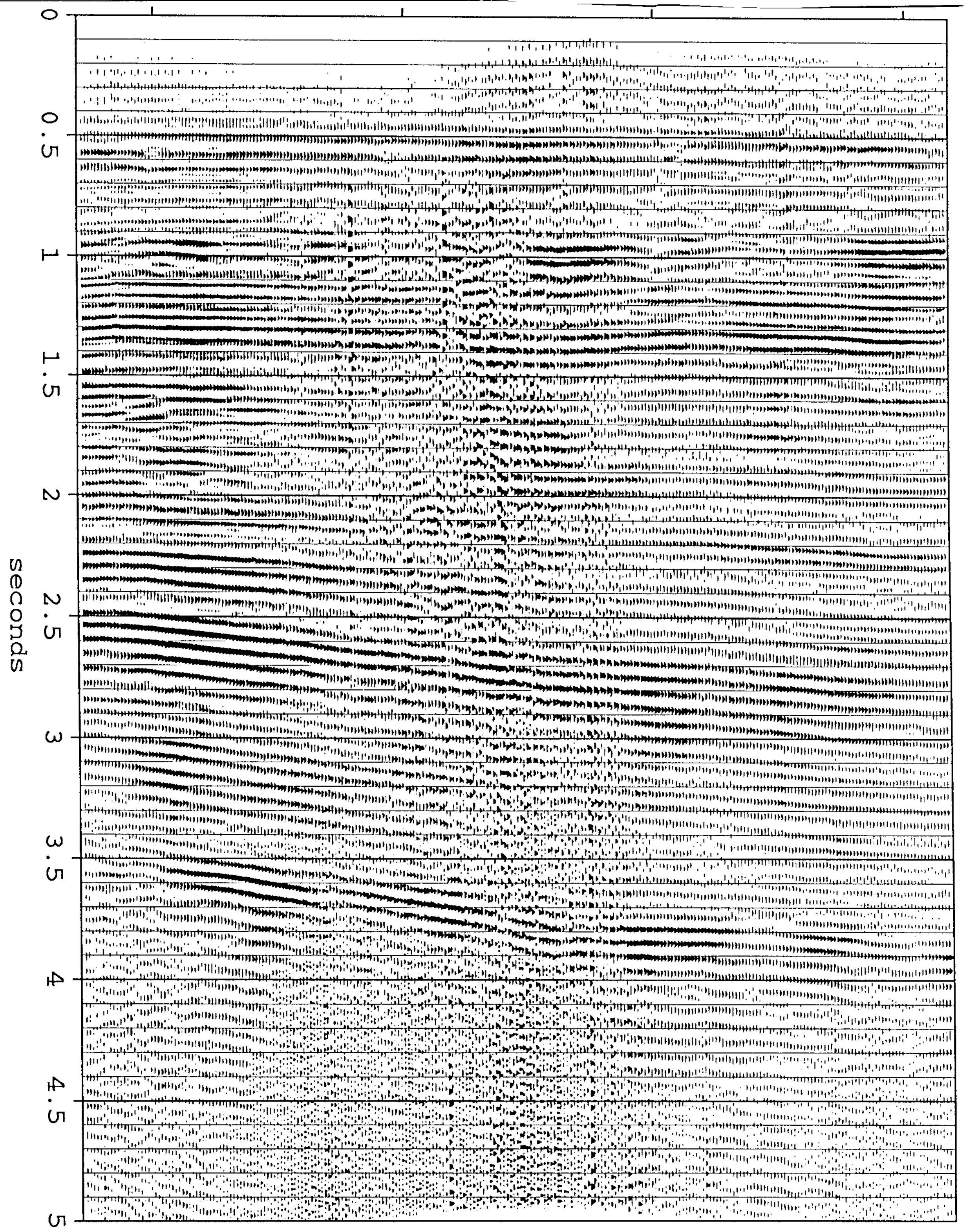

FIG. 6. Stack after surface-consistent trace balancing based upon amplitudes measured after offset balancing. An average amplitude was computed for each common-offset panel and divided into the amplitudes along that offset. The gain factors thus computed were applied to the unfiltered input traces to produce this stack. The stack is now well balanced.

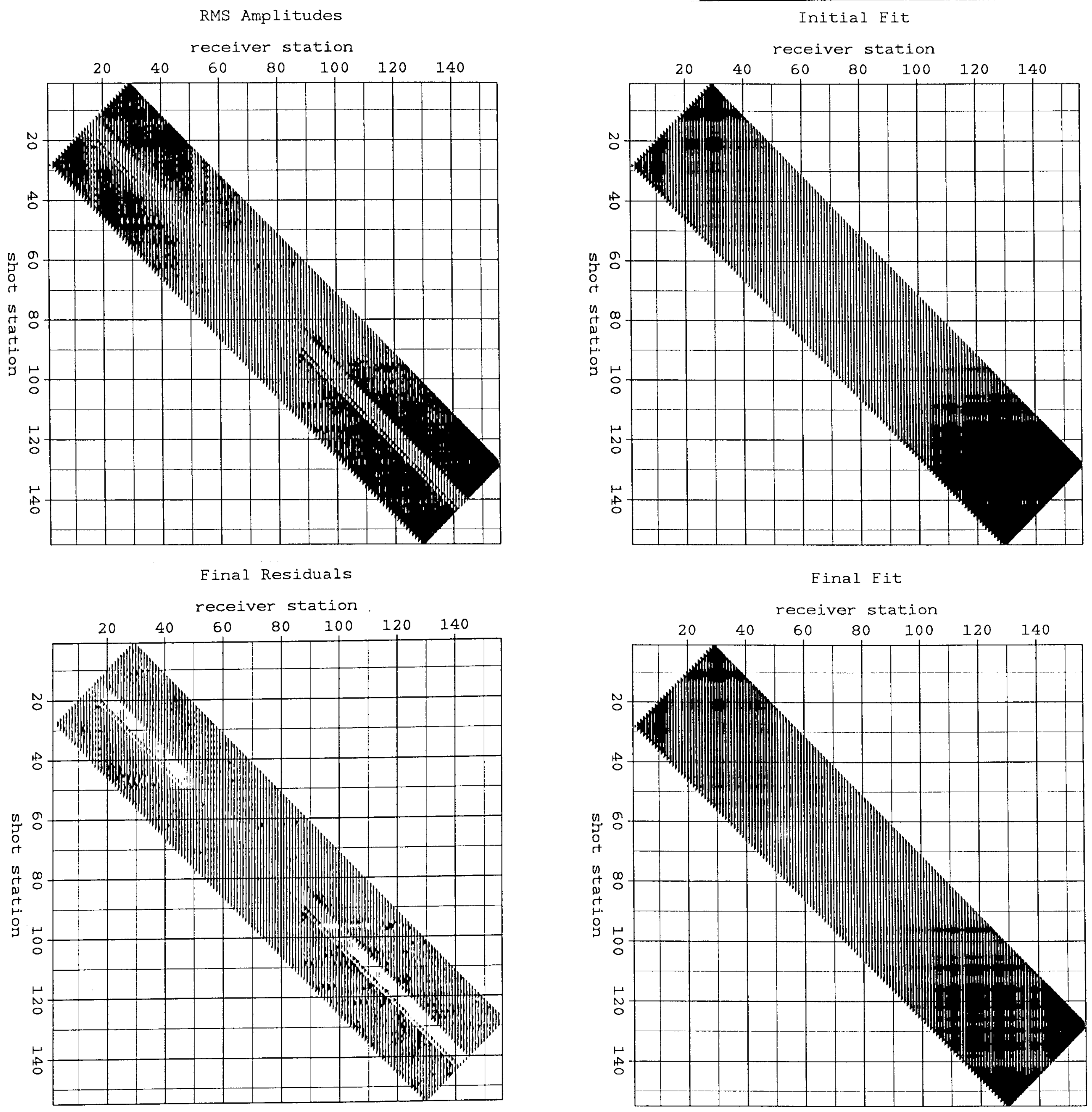

FIG. 7. Trace amplitudes, initial and final fits and the residuals for Fig. 6. The residual scatter is quite uniform, reflecting a good fit of the model to the offset-balanced amplitudes.

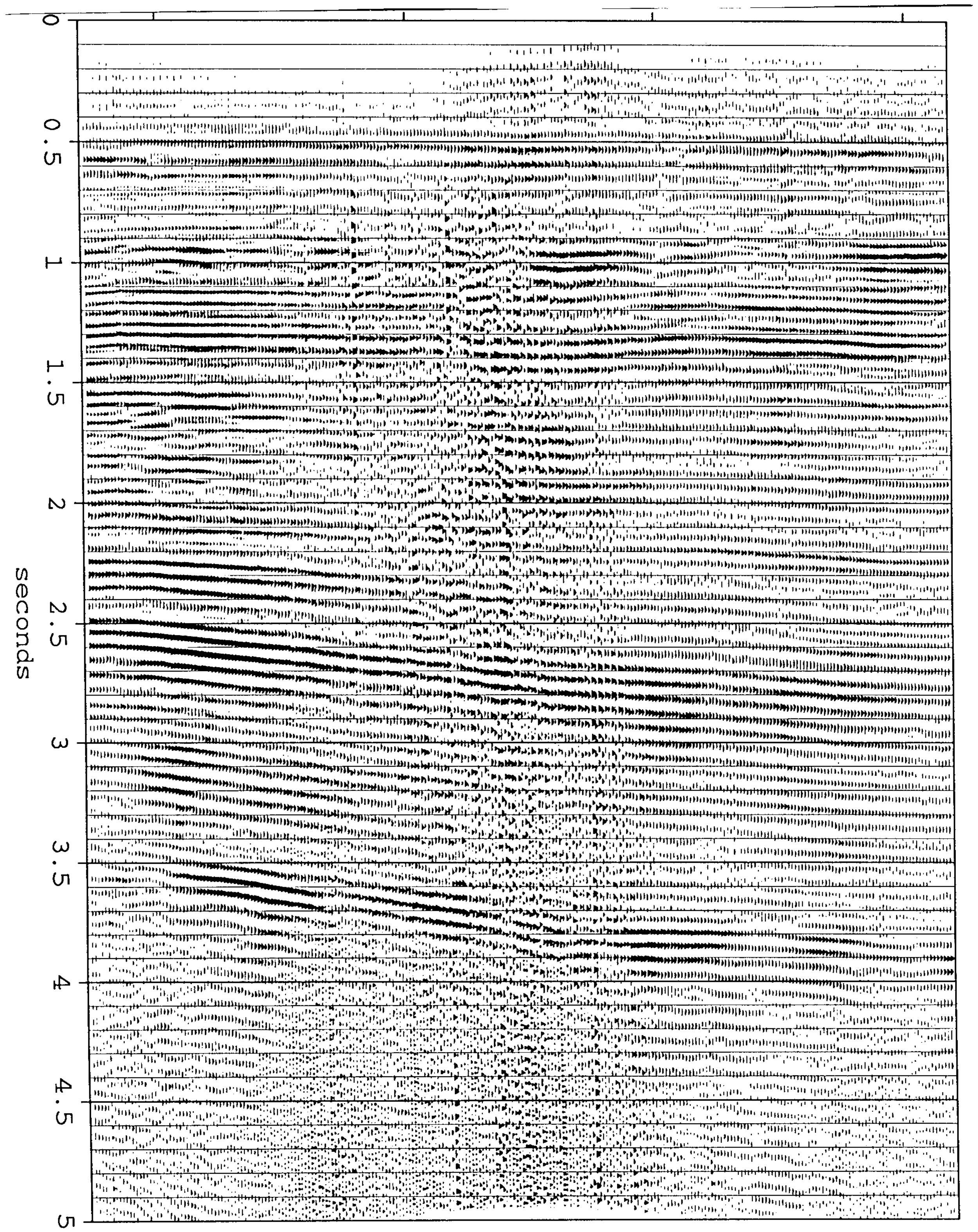

FIG. 8. Stack after surface-consistent trace balancing using the offset-balanced amplitudes of Figs 6 and 7 plus diagonal preconditioning to improve convergence. Again the stack shows good lateral balance.

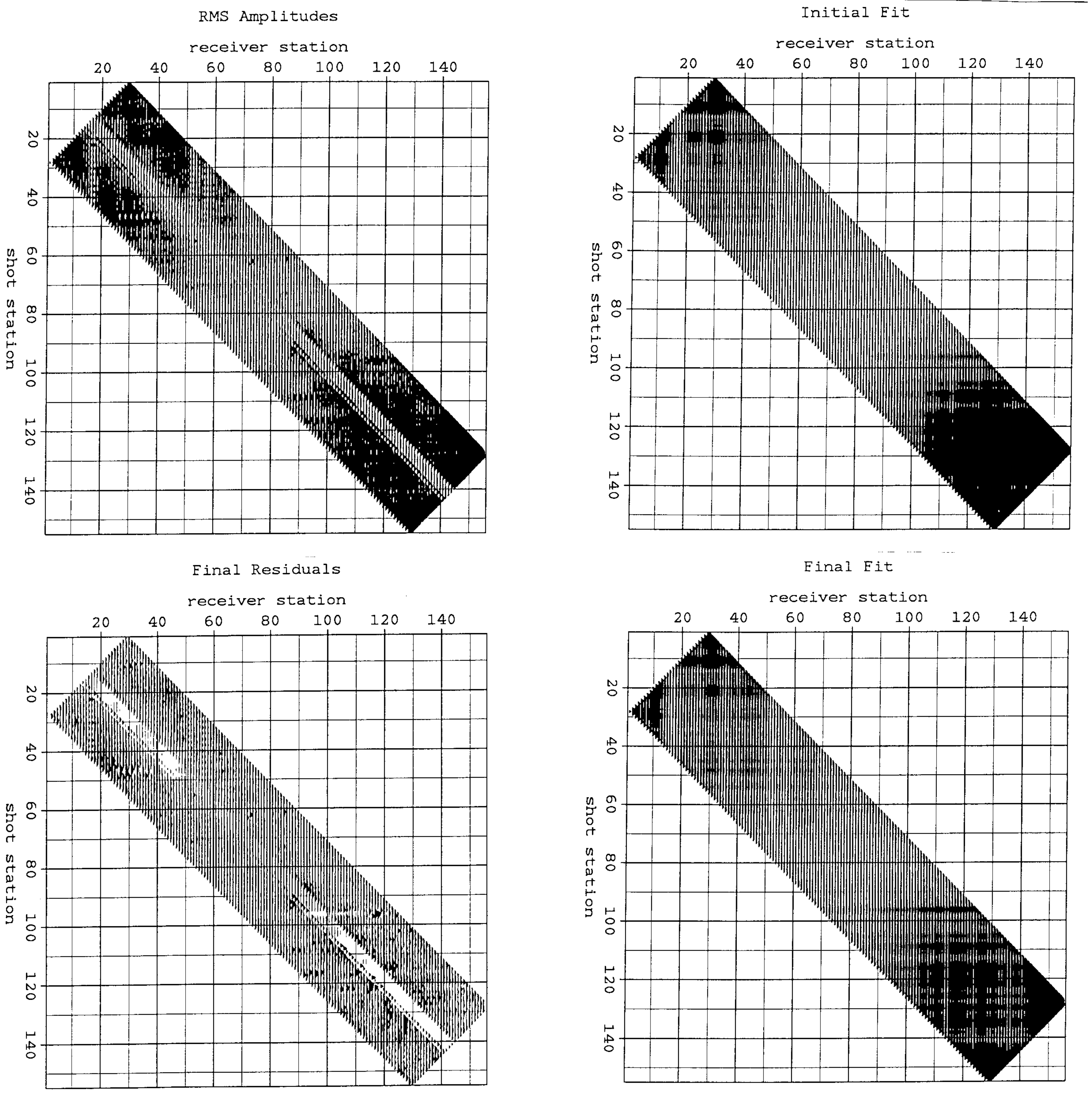

FIG. 9. Trace amplitudes, initial and final fits and the residuals for Fig. 8. The final fit is reassuringly the same as in Figure 7 but was reached in 30 percent iterations due to the diagonal preconditioning.

Local Application Programs Usage: 1/84 - 8/85

Rank	*#Used*	*Name*	*Rank*	*#Used*	*Name*
1	48874	vaon	51	3696	TEX
2	44427	Window	52	3675	catimp
3	43758	Gpen	53	3598	plas
4	34991	Graph	54	3530	aedreset
5	32765	RENICE	55	3444	KILL
6	30443	findtty	56	3412	gpen
7	30149	Tube	57	3389	ctomat
8	27304	vasoftoff	58	3172	pen
9	23576	rows	59	3159	Bandpass
10	19006	graph	60	3155	Pad
11	18503	apq	61	3085	Transp
12	16389	gcolor	62	3080	pause
13	16016	Epen	63	2837	therm
14	14332	Taplot	64	2818	Tiplot
15	13883	disfil	65	2810	Cat
16	13785	ipr	66	2559	Fft2d
17	13592	Movie	67	2528	gpp
18	13443	sywind	68	2468	apen
19	13149	man	69	2416	Disfil
20	13008	bandpass	70	2287	conmat
21	12579	Teplot	71	2240	sysort
22	12306	itroff	72	2158	untab
23	11926	Pen	73	2154	hyphen
24	11482	Merge	74	2141	bigmatlab
25	10096	aplink	75	2071	where
26	9713	tube	76	2025	syxfft
27	9076	Thplot	77	1968	hoggraph
28	8658	Wiggle	78	1922	Tpow
29	8618	max	79	1828	Vzoom
30	8442	Cp	80	1803	vpick
31	8248	dviimp	81	1764	ftplot
32	7930	catdvi	82	1749	Zero
33	7292	systrip	83	1739	Reverse
34	7116	Rm	84	1703	ranlib
35	7112	matlab	85	1703	errstrip
36	6900	Append	86	1682	Nmo
37	6652	Rmsubcmd	87	1641	apc
38	6203	Contour	88	1630	Epen2
39	5764	epen	89	1612	tex
40	5386	apuser	90	1603	epen2
41	5111	Real	91	1549	Add
42	5064	mattoc	92	1548	transp
43	4906	Ipen	93	1527	sydmo
44	4835	Cftplot	94	1526	apal
45	4684	Trplot	95	1517	systack
46	4596	Rtoc	96	1449	Ctek
47	4240	setup	97	1415	ipen
48	4119	Pchain	98	1388	dvi-imagen
49	3989	Apen	99	1373	sysxfft
50	3965	imprint	100	1360	Cabs