Chapter V

Velocity Stack Stochastic Inversion

5.1 Introduction: the problem of unknown model variances

In section 1.4 the stochastic inverse was mentioned as an alternative to the generalized least squares inversion of the system of equations (1.4). Aki and Richards (1980, section 12.3.5) define the stochastic inverse as the solution to a set of normal equations weighted by an additional diagonal term, which is inversely proportional to the expected variance of the solution (equation 1.7). The normal equations, with the diagonal term, are commonly derived from the maximum a posteriori (MAP) estimator, in which Gaussian assumptions are made concerning the various probability densities involved. The next section will define the MAP estimator and outline such a derivation. But the advantage to formulating the stochastic inverse as a MAP estimator is that the constant diagonal term in (1.7) can be generalized to an arbitrary diagonal matrix. This generalization is made by simply specifying a model variance that differs from point to point in the model domain. Similarly, non-constant noise variances in the data domain may be incorporated into the normal equations with a diagonal weighting term. As long as the model and noise variances are specified a priori, the modified system of "normal equations", derived from MAP estimation, remains linear.

The requirement that model variances be specified beforehand is an overly restrictive one. This is the chief limitation in the use of the linear stochastic inverse: *a priori* model variances must be estimated beforehand, without knowledge of the solution, yet these variances can have a great effect on the final solution.

In velocity stack inversion, this requirement is equivalent to predicting which velocities will be present in the solution. In our case, we would like to assume nothing a priori about the solution, apart from specifying a realistic range of velocities to be encountered.

Estimating a priori noise variances, on the other hand, is not much of a problem because the common-midpoint gather (in the data domain) can be readily examined for noise. Independent noise analyses, recorded in the field, may be available. When there is a problem with a noisy trace, it may be ignored by setting its noise variance to infinity.

One way out of the problem of having to make a priori assumptions about model variances is by a bootstrap process: iteratively update the estimate of the model variances simultaneous with the solving of the MAP estimator. Theoretical arguments in support of such a procedure are given in the next section. Making the model variance functionally dependent upon the final solution turns the MAP estimation functional into a *parsimony* or sparseness measure. Solutions to the MAP estimator will have a tendency to be driven to sparseness: the envelope of the solution in the model domain will tend to cluster into a few, large peaks, but elsewhere will tend to be very small. Sparseness is therefore a desirable property for the velocity stack inverse to have.

A linear system of equations as large as (1.7), and even more so the corresponding nonlinear problem of bootstrapped model variances, cannot be solved directly, considering the dimensionality of the problems that we are dealing with. For their solution, a gradient descent algorithm shall be developed later in this chapter. Some examples of velocity stack inversion on field data shall follow, so that a comparison can be made between the stochastic approach and the generalized inverse approach to velocity stacking.

5.2 Parsimonious inversion and MAP estimation

In this section we shall derive the nonlinear system of equations that, when solved, yields the data set's so-called stochastic inverse in velocity space. In the terminology of chapter 2, this is a type I inverse: the inversion is applied in order to get a velocity panel **u** from the data **d**. For the rest of this chapter, the term *stochastic inverse* is generalized to mean the MAP estimator derived here.

MAP, or *maximum a posteriori*, estimation is defined to be the maximization of $p(\mathbf{u} \mid \mathbf{d})$, the conditional probability density of **u** given **d**, and is produced by variation over the model parameters **u**. The functional relation between **u** and **d** is given by $\mathbf{d} = \mathbf{L}\mathbf{u} + \mathbf{n}$. Applying Bayes' rule,

$$p(\mathbf{u} \mid \mathbf{d}) = \frac{p(\mathbf{d} \mid \mathbf{u})\, p(\mathbf{u})}{p(\mathbf{d})} \tag{5.1}$$

Now $p(\mathbf{d} \mid \mathbf{u})$, the conditional probability of **d** given **u**, can be interpreted as the probability density of the noise, $p(\mathbf{n})$, since the random variable **d** is the sum of a deterministic **Lu** and a random variable **n**. The probability density $p(\mathbf{d})$ enters only as the normalizing term

$$p(\mathbf{d}) = \int p(\mathbf{d} \mid \mathbf{u}) p(\mathbf{u})\, d\mathbf{u} \tag{5.2}$$

Thus, maximizing the a posteriori model density $p(\mathbf{u} \mid \mathbf{d})$ is equivalent to maximizing the product of the noise density $p(\mathbf{n})$ with the a priori model density $p(\mathbf{u})$. Let us now assume the probabilities of equation (5.1) to be in exponential (actually Gaussian) form; this fact makes it convenient to define the following functionals:

$$S_N \equiv -\ln p(\mathbf{u} \mid \mathbf{d}) \tag{5.3}$$

$$S_P \equiv -\ln p(\mathbf{u}) \tag{5.4}$$

The MAP estimation problem becomes the minimization problem

$$\min_{\mathbf{u}} \; (S_N + S_P) \tag{5.5}$$

Let us assume independent Gaussian noise for $p(\mathbf{n})$:

$$
\begin{aligned}
S_N &= -\ln\left[\frac{1}{C_1}\,e^{-\frac{1}{2}\mathbf{n}^T\,\mathrm{diag}(\sigma_n^{-2})\,\mathbf{n}}\right] \\
&= \frac{1}{2}(\mathbf{Lu}-\mathbf{d})^T\mathrm{diag}\left[\frac{1}{\sigma_n^2}\right](\mathbf{Lu}-\mathbf{d}) + C_2
\end{aligned}
\tag{5.6}
$$

If σ_n is constant, the factor σ_n^{-2} may be taken out of the diagonal term in the center of (5.6); a uniformly weighted least squares functional is left. The gradient vector $\mathbf{g}_N$ of the noise functional S_N is then given by its elements:

$$
g_{Ni} = \frac{\partial S_N}{\partial u_i} = \frac{1}{\sigma_{ni}^2}\,\mathbf{L}^T(\mathbf{Lu}-\mathbf{d})_i \tag{5.7}
$$

where the subscript i is an index to elements in model space. The normal equations $\mathbf{L}^T\mathbf{Lu} = \mathbf{L}^T\mathbf{d}$ result when the gradient is set to zero.

If the a priori model density $p(\mathbf{u})$ is similarly assumed to be independent and Gaussian, S_P is seen to equal $\mathbf{u}^T\boldsymbol{\sigma}^{-2}\mathbf{u}/2$ where σ^2 refers to the variance (a diagonal matrix) in the model domain. The normal equations (5.7) are modified to become

$$
0 = \mathbf{g} = \frac{1}{\boldsymbol{\sigma}^2}\mathbf{u} + \frac{1}{\sigma_n^2}\,\mathbf{L}^T(\mathbf{Lu}-\mathbf{d}) \tag{5.8}
$$

This is the proper stochastic inverse. The diagonal of the normal equations is weighted with the noise-to-signal ratio σ_n^2/σ^2, and enough weighting to the diagonal will guarantee that the solution $\mathbf{u}$ to (5.8) exists and is unique.

The variance $\boldsymbol{\sigma}^2$ in equation (5.8) is a deterministic variable. An alternative to a deterministic variance is to allow $\boldsymbol{\sigma}$ itself to be a random variable, able to take on, say, N discrete values $\{\sigma_k\}$ from $k = 1$ to N. Identifying $\boldsymbol{\sigma}$ with a random variable makes sense in the present case, because precise a priori knowledge of the variance of the solution will never be available. As a consequence of the random nature of $\boldsymbol{\sigma}$,

$$
p(\mathbf{u}) = \int p(\boldsymbol{\sigma})\,p(\mathbf{u}\,|\,\boldsymbol{\sigma})\,d\boldsymbol{\sigma} \tag{5.9}
$$

Now make the following assumption, that the joint conditional density $p(\mathbf{u}\,|\,\boldsymbol{\sigma})$ is independent (correlation free) and Gaussian with zero mean:

$$p(\mathbf{u}\,|\,\boldsymbol{\sigma}) = \prod_i p(u_i\,|\,\sigma_i) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp \frac{-u_i^2}{2\sigma_i^2} \tag{5.10}$$

When σ_i is made a member of the class of values $\{\sigma_k\}$, points in the model domain are implicitly partitioned into N distinct populations. Assumption (5.10) claims that each population (say the kth) has a Gaussian distribution, with a unique variance σ_k^2, and zero mean.

No constraints need to be put on the density $p(\boldsymbol{\sigma})$ at the moment. Specifically, $p(\boldsymbol{\sigma})$ does not have to be separable into a product of individual probability densities $p_i(u_i)$, one for each point u_i in model space.

Let us now calculate the gradient $\mathbf{g}_P$ of the *parsimony functional* S_P in equation (5.4). The ith term of the parsimony gradient is

$$\begin{aligned} g_{Pi} &= \frac{\partial S_P}{\partial u_i} = -\frac{1}{p(\mathbf{u})}\frac{\partial}{\partial u_i}p(\mathbf{u}) \\ &= -\frac{1}{p(\mathbf{u})}\int p(\boldsymbol{\sigma})\frac{\partial}{\partial u_i}p(\mathbf{u}\,|\,\boldsymbol{\sigma})\,d\boldsymbol{\sigma} \end{aligned} \tag{5.11}$$

Inserting the expression for the partial derivative with respect to u_i of $p(\mathbf{u}\,|\,\boldsymbol{\sigma})$ (equation 5.10), yields

$$\begin{aligned} g_{Pi} &= -\frac{1}{p(\mathbf{u})}\int p(\boldsymbol{\sigma})\frac{-u_i}{\sigma_i^2}p(\mathbf{u}\,|\,\boldsymbol{\sigma})\,d\boldsymbol{\sigma} \\ &= \int \frac{u_i}{\sigma_i^2}\,\frac{p(\boldsymbol{\sigma})p(\mathbf{u}\,|\,\boldsymbol{\sigma})}{p(\mathbf{u})}\,d\boldsymbol{\sigma} \\ &= u_i\int \frac{1}{\sigma_i^2}p(\boldsymbol{\sigma}\,|\,\mathbf{u})\,d\boldsymbol{\sigma} \end{aligned} \tag{5.12}$$

Bayes' rule has been used in equation (5.12) to define the new conditional density $p(\boldsymbol{\sigma}\,|\,\mathbf{u})$. The last integral of equation (5.12) defines the mathematical expectation of σ_i^{-2} conditional upon $\mathbf{u}$:

$$\tilde{\sigma}_i^{-2} \equiv \mathbf{E}\left[\frac{1}{\sigma_i^2}\middle|\,\mathbf{u}\right] = \int \frac{1}{\sigma_i^2}p(\boldsymbol{\sigma}\,|\,\mathbf{u})\,d\boldsymbol{\sigma} \tag{5.13}$$

Thus

$$g_{Pi} = \frac{1}{\tilde{\sigma}_i^2} u_i \qquad (5.14)$$

The mathematical expectation in (5.13) allows σ_i to have an explicit functional dependence on the entire model domain **u**, not only on the corresponding point u_i. The *stochastic inverse* can now be generalized to be the solution **u** of the set of equations resulting from setting the total gradient $\mathbf{g} = \mathbf{g}_P + \mathbf{g}_N$ to zero:

$$0 = \mathbf{g} = \frac{1}{\tilde{\boldsymbol{\sigma}}^2}\mathbf{u} + \frac{1}{\sigma_n^2}\mathbf{L}^T(\mathbf{L}\mathbf{u} - \mathbf{d}) \qquad (5.15)$$

When no functional dependence of $\tilde{\boldsymbol{\sigma}}$ on **u** is specified, the equations are linearized and become equivalent to the stochastic inverse (in the strict meaning) of equation (5.8).

5.3 Parsimony criteria

The selection of a parsimony criterion has now been reduced to the choice of an expression for the conditional expectation $\mathrm{E}[\sigma_i^{-2} \,|\, \mathbf{u}]$. It is logical to assume that the variance σ_i^2 at each point in the model domain depends on a local weighted average of **u** about that point:

$$\tilde{\sigma}_i^{-2} \equiv \mathrm{E}[\sigma_i^{-2} \,|\, \sum_j w_{ij}\, u_j^2,\ j \text{ near } i] \qquad (5.16)$$

where the w_{ij} are positive normalized weights, so that $\sum_j w_{ij} = 1$. Because the model domain by nature is two dimensional, it is convenient to index the fields $\boldsymbol{\sigma}$ and **u** by slowness p and time τ. The specification of (5.16) makes $\tilde{\sigma}$ a smooth function of **u**. For the indices p and τ, it is sensible to make the following smoothness assumptions:

(A) $\tilde{\sigma}_{p,\tau}^2$ does not depend on values $u_{p',\tau'}$ from adjacent traces $p' \neq p$.

(B) $\tilde{\sigma}^2_{p,\tau}$ depends only on values $u_{p,\tau'}$ within a limited time window about index τ.

In other words, $\tilde{\sigma}^2_{p,\tau}$ is only a function of $u_{p,\tau+j}$ over the time window $j = -T$ to T. The simplest rule in the form of equation (5.16) is a direct estimate of the variance from a local patch of data:

$$\tilde{\sigma}^{-2}_{p,\tau}(\mathbf{u}) = \left[\frac{1}{2T+1} \sum_{j=-T}^{T} u^2_{p,\tau+j} \right]^{-1} \tag{5.17}$$

With this rule, points in the velocity domain can be assigned to one of a discrete, finite set of populations; the $\tilde{\sigma}^2_{p,\tau}$ found by means of equation (5.17) can be rounded to the nearest member of a set of allowed variances $\{\sigma^2_k\}$. Each member of this set represents a variance of the corresponding Gaussianly-distributed population of points.

Assumptions (A) and (B) are meant to simplify the parsimony criterion $\mathbf{E}[\sigma^{-2}_{p,\tau}|\mathbf{u}]$ by making the estimate $\tilde{\sigma}^2_{p,\tau}$ independent of points far away from (p,τ). Recall that the dependence of $\mathbf{E}[\sigma^{-2}_{p,\tau}|\mathbf{u}]$ on the entire model $\mathbf{u}$ is a direct consequence of assuming that the joint density $p(\boldsymbol{\sigma})$ is nonseparable in model space. That is, correlations are allowed between the various random variables $\tilde{\sigma}^2_{p,\tau}$ over the velocity plane. Allowing only positive correlations between points that are close neighbors, results in expectations of the form of equation (5.16). Rules like equation (5.16) force the members of a population to group closely in the velocity plane: because the sum in equation (5.16) varies smoothly from point to point, $\tilde{\sigma}^2_{p,\tau}$ also varies smoothly. So when we distinguish between only k discrete populations $\{\sigma^2_k\}$, $\tilde{\sigma}^2_{p,\tau}$ must be rounded to the nearest member σ^2_k, and equation (5.16) will tend to put points in a local neighborhood into the same population, identified by the variance σ^2_k. Equations (5.16) and (5.17) are examples of what we may call a *clustering* criterion, because of their tendency to favor the clustering of points who are members of the same Gaussian population.

To get a better idea of how the conditional expectation $\mathbf{E}[\tilde{\sigma}_{p,\tau}^{-2}|\mathbf{u}]$ gives parsimonious behavior to **u**, let us assume for the moment that clustering rules such as equation (5.16) are not applied. We have yet to define precisely what characteristics a "parsimonious" solution **u** exhibits; a parsimonious solution to the equations (5.15) is pictured as having relatively few large elements scattered in a sea of small elements. A more precise definition of parsimony will soon be given.

By saying that no clustering is allowed, we mean that the joint density $p(\sigma)$ separates into a product of independent one-dimensional densities $p(\sigma_{p,\tau})$. The conditional expectation of the variance at each point in model space reduces to $\mathbf{E}[\sigma_{p,\tau}^{-2}|u_{p,\tau}]$; i.e., the rule for selecting a variance at the point (p,τ) is to depend only upon the current value $u_{p,\tau}$ at that point. Selecting such a rule fixes the functional form of the parsimony gradient g_P of equation (5.14); moreover, it establishes the form of the prior density $p(\mathbf{u})$ in the MAP estimator.

As an example, consider the simplest choice for $\tilde{\sigma}_{p,\tau}$ corresponding to equation (5.17):

Rule 1: $$\tilde{\sigma}_{p,\tau}^{-2}(u) = \mathbf{E}[\sigma_{p,\tau}^{-2}|u] = |u_{p,\tau}|^{-2} \tag{5.18}$$

To what does this rule fix the corresponding gradient and prior density $p(u)$ to be? For the remainder of this section, the subscripts (p,τ) are dropped so that $\tilde{\sigma}$ and u refer to a common element of the model domain. An element of the parsimony gradient $g_P(u)$, from equation (5.14), is simply

$$g_P(u) = \frac{1}{u} \tag{5.19}$$

Later, when we attempt to solve the nonlinear equations (5.15), we will employ by a gradient descent method. Such a method requires that the gradient be continuous and that it vanish at some point. By itself, equation (5.19) is inadequate for use as the parsimony component to the full gradient **g**, because it becomes infinitely discontinuous at the origin $u = 0$. We shall impose the following requirement on the parsimony gradient: $g_P(u)$ must go the zero continuously as u goes to zero. The easiest way to impose this requirement is to choose a limiting variance σ_o^2 below which the variance $\tilde{\sigma}^2(u)$ is not allowed to go. In other words, we assume that there is a background population of points described by a Gaussian distribution with zero mean and variance σ_o^2. In a similar way, it is reasonable to assume that very large values of u belong to a zero-mean, Gaussian population with the upper limiting variance σ_∞^2. With these assumptions, rule 1 (equation 5.18) is modified to become

Rule 2: $$\tilde{\sigma}^2(u) = \begin{cases} \sigma_o^2 & \sigma_o > |u| \\ |u|^2 & \sigma_o \leq |u| \leq \sigma_\infty \\ \sigma_\infty^2 & \sigma_\infty < |u| \end{cases} \tag{5.20}$$

The standard deviation $\tilde{\sigma}$ of (5.20) is plotted as a function of u in figure 5.1(a). The gradient, shown in figure 5.1(b), is consequently

$$g_P(u) = \begin{cases} u/\sigma_o^2 & \sigma_o > |u| \\ 1/u & \sigma_o \leq |u| \leq \sigma_\infty \\ u/\sigma_\infty^2 & \sigma_\infty < |u| \end{cases} \tag{5.21}$$

Recall that the gradient was defined to be the derivative of the logarithm of the a priori density $p(\mathbf{u})$ (equation 5.4):

$$p(u) = e^{-\tilde{S}_P}$$

$$\tilde{S}_P(u) \equiv \int^{u} g_P(u')\,du' \tag{5.22}$$

where $\tilde{S}_P$ is one term of the parsimony functional S_P of equation (5.4) at the point (p,τ). $g_P(u)$ can now be integrated to give

$$\tilde{S}_P(u) = \begin{cases} C_1 + u^2/2\sigma_o^2 & \sigma_o > |u| \\ C_1 + 1/2 + \ln(|u|/\sigma_o) & \sigma_o \leq |u| \leq \sigma_\infty \\ C_1 + \ln(\sigma_\infty/\sigma_o) + u^2/2\sigma_\infty^2 & \sigma_\infty < |u| \end{cases} \tag{5.23}$$

Consequently

$$p(u) = \begin{cases} C_2 e^{-u^2/2\sigma_0^2} & \sigma_0 > |u| \\ C_2 \dfrac{\sigma_0}{|u|} e^{-1/2} & \sigma_0 \leq |u| \leq \sigma_\infty \\ C_2 \dfrac{\sigma_0}{\sigma_\infty} e^{-u^2/2\sigma_\infty^2} & \sigma_\infty < |u| \end{cases} \qquad (5.24)$$

where C_1 and C_2, equal to exp $(-C_1)$, are chosen to normalize the probability density $p(u)$ in (5.24). Functions $\tilde{S}_P$ and $p(u)$ are illustrated in figures 5.1(c) and 5.1(d), respectively. The prior density $p(u)$ is easy to describe: it is Gaussian in form for low and high values of u, and, in the middle ranges, has a taper proportional to the inverse of u.

The lower limiting variance σ_0^2 is needed to enforce the continuity of the derivative g_P at the origin $u = 0$. Similarly, the upper limiting variance σ_∞^2 forces the gradient to be linear at large values of u, which imposes Gaussian behavior upon the population of very large elements. Otherwise the gradient (in the form of equation 5.19) would converge to zero as $u \rightarrow \infty$. This tendency toward zero would impose poor convergence characteristics upon any gradient descent method attempting to solve the system (5.15).

A *parsimonious* distribution is defined as one whose kurtosis is higher than that of a normal (Gaussian) distribution. Kurtosis, usually defined to be the ratio of the fourth moment to the second moment of the distribution, is a measure of the tail weight of a non-Gaussian distribution (Gray, 1979). The distribution $p(u)$ of figure 5.1(d) has the weighting of its tails enhanced by the inverse taper between σ_0 and σ_∞, and so, in the sense that has been defined, is parsimonious.

An alternative method for deriving a choice for $\mathbf{E}[\sigma^{-2}|\mathbf{u}]$, is to specify a one-dimensional density $p(u_{p,\tau})$ with the desired quality of large tails on the density. Such a density can be chosen out of the family of *generalized Gaussians* defined by Gray (1979):

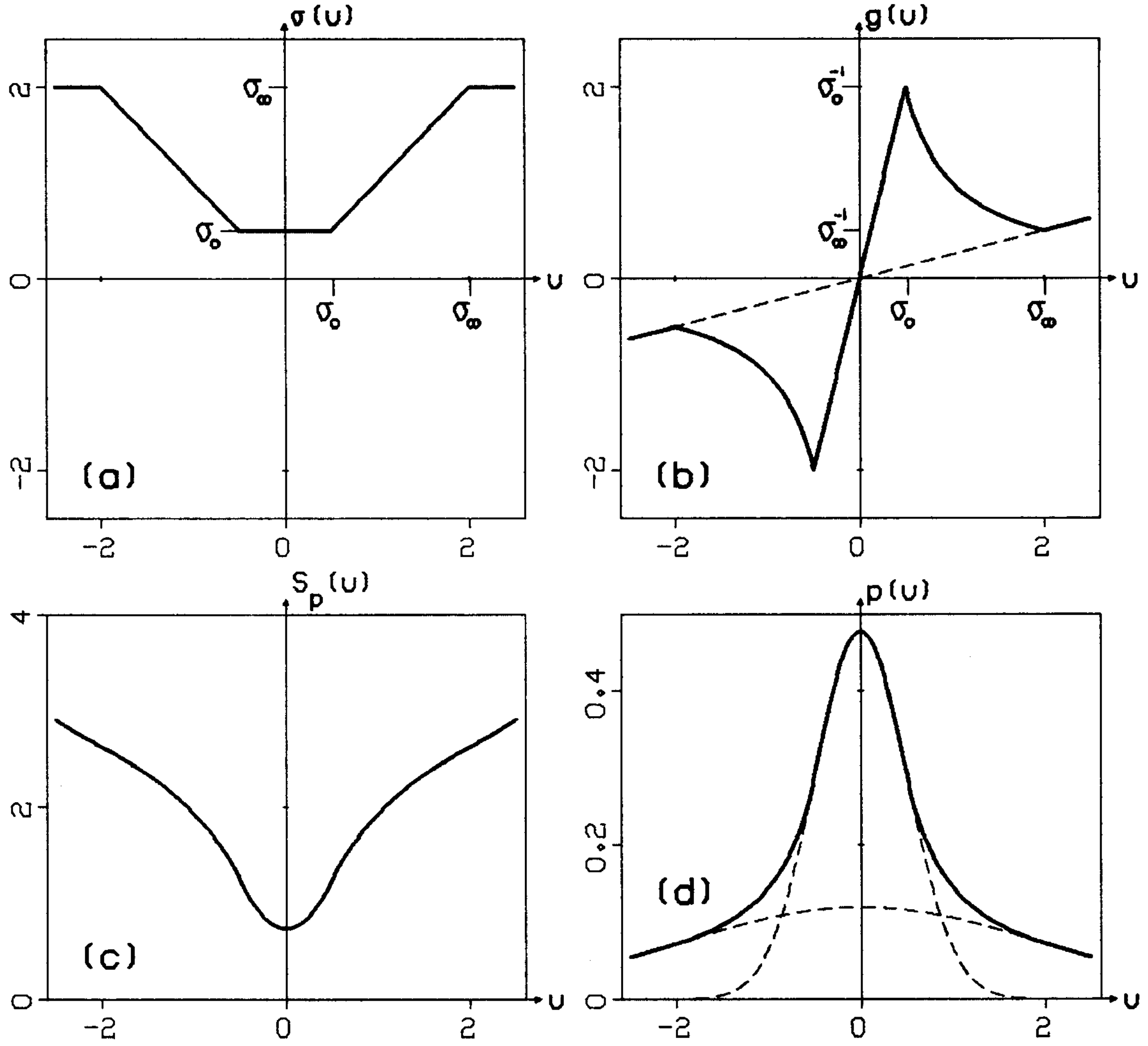

FIG. 5.1. The parsimony gradient $g_P(u)$ and prior density $p(u)$ for the continuous-variance case.

(a) Expected value of the standard deviation given u, from equation (5.20). Variances are clipped by the minimum and maximum values of σ_0 and σ_∞. Here, $\sigma_0 = 0.5$ and $\sigma_\infty = 2.0$.

(b) One-dimensional gradient of the parsimony functional, equation (5.21). The gradient is linear outside the range (σ_0, σ_∞), and decays as u^{-1} within this range.

(c) The parsimony functional, the integral of the gradient in (b), from equation (5.23). The slope of this potential "well" is steepest at $u = \sigma_0$. If there were no upper limit σ_∞, the gradient of this surface would vanish as $u \to \infty$ and there would be no incentive for a descent routine to force large values of u to be smaller.

(d) The resulting one-dimensional prior density, the exponent of the parsimony functional (equation 5.24). The two dashed curved are the limiting Gaussian envelopes whose standard deviations are σ_0 and σ_∞. The curve between the envelopes is an inverse-u decay.

$$p(u) = \frac{\alpha}{2\beta\Gamma(\alpha^{-1})} \exp\left[-\frac{|u|^{\alpha}}{\beta^{\alpha}}\right] \qquad (5.25)$$

$\Gamma(x)$ is the gamma function. When $\alpha = 2$, $p(u)$ is Gaussian. As $\alpha \to 0$, the kurtosis of $p(u)$ increases as the distribution becomes more and more "parsimonious", or spiky in appearance. Thus, any member of the generalized Gaussian family whose shape parameter α lies between 0 and 2 can be used as a parsimonious prior density $p(u)$. For example, if $\alpha = 1$, the corresponding functional $\tilde{S}_P(u)$ is linear in u, and the gradient is constant. The presence of these characteristics implies that the standard deviation can be estimated as the square root of $|u|$. The limiting variances σ_o^2 and σ_∞^2 are again needed in this case to enforce continuity of the gradient.

An additional incentive to placing limits on the range of possible variances is to keep the resulting set of equations (5.15) well-conditioned. When the terms $\tilde{\sigma}_{p,r}^{-2}$ are considered to be elements of the diagonal matrix $\boldsymbol{\sigma}^{-2}$ in equation (5.15), limiting the range of the elements to between σ_o^{-2} and σ_∞^{-2} places an upper limit on the condition number of $\boldsymbol{\sigma}^{-2}$: namely, $\sigma_\infty^2 / \sigma_o^2$.

Rule 2 in equation (5.20) assumed the existence of a continuous (infinite) set of populations characterized by a variance $\tilde{\sigma}^2$ between σ_o^2 and σ_∞^2. Let us now go back to our original assumption of a discrete, finite set of populations $\{\sigma_k^2\}$, $k = 0, n-1$, and observe the form the prior density $p(u)$ takes:

Rule 3: $$\tilde{\sigma}^2(u) = \sigma_k^2, \qquad u \;\varepsilon\; \Phi_k, \quad k = 0, n-1 \qquad (5.26)$$

where

$$\Phi_k = \left\{ u \text{ for which } \sigma_{k-1} \le |u| < \sigma_k \right\}$$
$$\Phi_o = \left\{ u \text{ for which } |u| < \sigma_o \right\}$$
$$\Phi_{n-1} = \left\{ u \text{ for which } \sigma_{n-2} \le |u| \right\} \qquad (5.27)$$

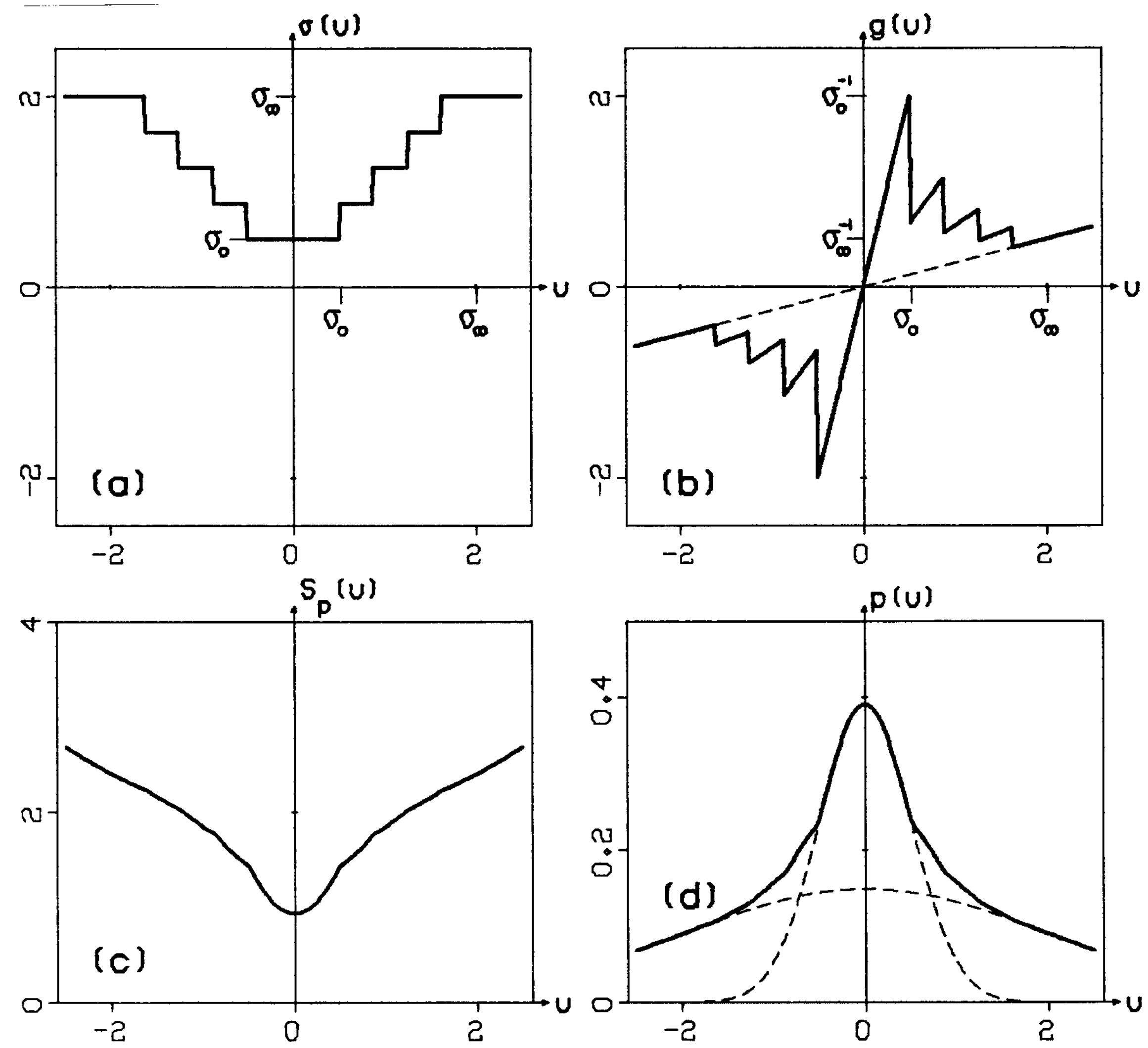

FIG. 5.2. Parsimony gradient and prior density for the discrete-variance case.

(a) Expected value of the standard deviation conditional upon u, equation (5.26). The limiting variances $\sigma_o = 0.5$ and $\sigma_\infty = \sigma_4 = 2.0$ have been chosen, as in figure 5.1. The set of $\{\sigma_k\}$ in this example are evenly spaced from σ_o to σ_4.

(b) One-dimensional gradient of the parsimony functional, equation (5.28). The gradient is linear with a zero-intercept within each segment $\sigma_{k-1} \leq u < \sigma_k$, but is discontinuous at the edges of each segment.

(c) Parsimony functional, equation (5.29). Though having a discontinuous derivative, it has the same overall shape as the corresponding parsimony functional of figure 5.1(c).

(d) The one-dimensional prior density, the exponential of (c) (equation 5.30). The dashed curves are identical to the limiting Gaussian envelopes shown in figure 5.1(d).

$\tilde{\sigma}(u)$ is the step function in u illustrated in figure 5.2(a). The corresponding gradient is

$$g_P(u) \;=\; \frac{u}{\sigma_k^2} \qquad u \;\varepsilon\; \Phi_k, \quad k = 0, n-1 \tag{5.28}$$

which is illustrated in figure 5.2(b). There is a discontinuity in the gradient at each point σ_k on the u axis, but as the subdivisions σ_k become finer, the gradient approaches the continuous gradient of figure 5.1(b). The parsimony functional $\tilde{S}_P(u)$, and the prior density $p(u)$, are illustrated in figures 5.2(c) and (d). They are given by the expressions

$$\tilde{S}_P(u) \;=\; C_k + \frac{1}{2}\frac{u^2}{\sigma_k^2} \qquad u \;\varepsilon\; \Phi_k, \quad k = 0, n-1 \tag{5.29}$$

$$p(u) \;=\; \exp\left[-C_k - \frac{1}{2}\frac{u^2}{\sigma_k^2}\right] \qquad u \;\varepsilon\; \Phi_k, \quad k = 0, n-1 \tag{5.30}$$

The normalizing terms C_k may be found by the recursive relation

$$C_k \;=\; C_{k-1} + \frac{1}{2}\left(\frac{\sigma_k^2}{\sigma_{k-1}^2} - 1\right) \qquad k = 1, n-1 \tag{5.31}$$

while C_0 is uniquely determined by the constraint

$$\int p(u)\, du \;=\; 1$$

Although the two gradients of figures 5.1 and 5.2 appear to be grossly different, there is relatively little difference between the shapes of the resulting prior densities. The smoothness of $p(u)$ in the discrete case justifies the use of a more convenient gradient: the continuous gradient of figure 5.1(b).

To summarize this section: instead of deriving the nonlinear system of equations (5.15) from a least-squares functional, they can be derived directly from the MAP estimator. As a consequence of this derivation, prior information about the solution can be introduced by definition of a rule for estimating variances as a function of **u**.

No explicit covariance is imposed on the solution **u**; rather, correlation between points in the model domain is possible because the variances $\sigma^2_{p,\tau}$ themselves can be defined to be random variables. The clustering property is imposed by allowing positive correlations to exist between neighboring variances $\sigma^2_{p,\tau}$ in model space.

In this section, various one-dimensional prior densities $p(u)$ have been considered as candidates for inclusion into the MAP estimator $p(\mathbf{d}\,|\,\mathbf{u})p(\mathbf{u})$. The feature they share in common is that the "variances" in the diagonal matrix $\boldsymbol{\sigma}^{-2}$ are monotonic increasing functions of $|u|$. For example, if $\tilde{\sigma}(u)$ is a constant function, the prior density is Gaussian; if $\tilde{\sigma}(u)$ is given by equation (5.20), the prior density is given by equation (5.24). For the general one-dimensional case, the dependence of $p(u)$ on $\tilde{\sigma}(u)$ can be written as

$$p(u_{p,\tau}) \;=\; C\exp\left[-\int \frac{u_{p,\tau}}{\tilde{\sigma}^2(u_{p,\tau})}\,du_{p,\tau}\right] \tag{5.32}$$

As long as $\tilde{\sigma}(u)$ is a monotonic increasing function of $|u|$, the proportion of area in the tails to area in the central peak of the probability density in (5.32) will be higher than it is in the Gaussian case. In such a case, the distribution is parsimonious in the sense defined in this section.

Finally, we impose the following requirements on the parsimony gradient $g = u/\sigma^2$: it must be continuous, vanish at $u = 0$, and become linear at high values of $|u|$. These requirements cannot be met with the use of any single member of the generalized Gaussian family of equation (5.25) (except the case $\alpha = 2$), but they can be met by defining limiting variances σ_0^2 and σ_∞^2, between which the monotonic function $\tilde{\sigma}^2(u)$ must be constrained to lie.

5.4 The multidimensional parsimony functional $S_P(\mathbf{u})$

The previous section described various choices for the one-dimensional parsimony functional $\widetilde{S}_P(u_{p,\tau})$. When correlations are allowed between points in model space, the total parsimony functional $S_P(\mathbf{u})$ is no longer the sum of independent terms $\widetilde{S}_P(u_{p,\tau})$. Let us now briefly examine what form $S_P(\mathbf{u})$ takes when Rule 2 (equation 5.20) of section 5.3 is in effect, but with the clustering criterion (equation 5.16) activated.

For convenience let points in the model domain now be indexed by a single subscript i. The clustering criterion of equation (5.16) claims that the standard deviation σ_i may be estimated by

$$\sigma_i = \left[\sum_k w_{ik} u_k^2 \right]^{1/2} \tag{5.33}$$

where the values w_{ik} comprise a local smoothing filter. All coefficients w_{ik} are positive, symmetric and sum to unity. If $S_P(\mathbf{u})$ is assumed to depend explicitly on the averaged variances $(\sigma_1, \cdots, \sigma_n)$, then S_P can be differentiated, using the chain rule, to yield the gradient:

$$g_i = \sum_j \frac{\partial S_P}{\partial \sigma_j} \frac{\partial \sigma_j}{\partial u_i} \tag{5.34}$$

But, from (5.33),

$$\frac{\partial \sigma_j}{\partial u_i} = \frac{\partial}{\partial u_i} \left[\sum_k w_{jk} u_k^2 \right]^{1/2} = \frac{w_{ji} u_i}{\sigma_j} \tag{5.35}$$

so that

$$g_i = u_i \sum_j \frac{\partial S_P}{\partial \sigma_j} \frac{w_{ji}}{\sigma_j} \tag{5.36}$$

The desired form for $g_i(u_i)$, given by equation (5.14), is

$$g_i = \frac{u_i}{\widetilde{\sigma}_i^2} \tag{5.37}$$

and we can make the following definitions for the variance and functional derivative:

$$\frac{1}{\tilde{\sigma}_i^2} = \sum_j w_{ij}\frac{1}{\sigma_j^2} \quad \text{and} \quad \frac{\partial S_P}{\partial \sigma_j} = \frac{1}{\sigma_j} \tag{5.38}$$

The definitions in equations (5.37) and (5.38) are therefore consistent with the gradient in (5.36). However, equation (5.38) is not entirely consistent with Rule 2 because the above assumptions yield the following functional dependence of $\tilde{\sigma}_i^2$ on **u**:

$$\tilde{\sigma}_i^2 = \left[\sum_j w_{ji}\left[\sum_k w_{jk}u_k^2\right]^{-1}\right]^{-1} \tag{5.39}$$

That is, $\tilde{\sigma}_i^2$ is the average of the inverse of the average of u_i^2. If the weights w_{ij} are smooth enough, equation (5.39) is approximated by use of the direct estimation

$$\tilde{\sigma}_i^2(\mathbf{u}) = \sum_j \tilde{w}_{ij}u_j^2 \tag{5.40}$$

Integrating the right-hand term of equation (5.38) we obtain the following expression for $S_P(\mathbf{u})$:

$$S_P(\mathbf{u}) = \sum_j \ln\sigma_j = \frac{1}{2}\sum_j \ln\left[\sum_k w_{jk}u_k^2\right] \tag{5.41}$$

The threshold variances σ_o^2 and σ_∞^2 can be easily introduced if σ_j^2 is constrained to lie between them. Using equation (5.39) in favor of (5.40) for the estimation of $\tilde{\sigma}^2$, the gradient becomes:

$$g_i = u_i\sum_j w_{ji}\begin{cases} \dfrac{1}{\sigma_o^2} & \sigma_j < \sigma_o \\ \dfrac{1}{\sigma_j}\dfrac{\partial S_P}{\partial\sigma_j} & \sigma_o \le \sigma_j \le \sigma_\infty \\ \dfrac{1}{\sigma_\infty^2} & \sigma_j > \sigma_\infty \end{cases} \tag{5.42}$$

Consequently S_P takes on the form

$$S_P(\sigma_1, \cdots, \sigma_n) = \sum_j \begin{cases} \dfrac{C_0}{2}\,\dfrac{\sigma_j^2}{\sigma_0^2} & \sigma_j < \sigma_0 \\ \ln \sigma_j & \sigma_0 \le \sigma_j \le \sigma_\infty \\ \dfrac{C_\infty}{2}\,\dfrac{\sigma_j^2}{\sigma_\infty^2} & \sigma_j > \sigma_\infty \end{cases} \qquad (5.43)$$

in which σ_j^2 are given by equation (5.33). The constants $C_0 = \ln \sigma_0^2$ and $C_\infty = \ln \sigma_\infty^2$ are required to make S_P continuous at the threshold points σ_0 and σ_∞. This multidimensional version of $S_p(\mathbf{u})$ is the simplest generalization of the one-dimensional parsimony functional $\tilde{S}_P(u)$ given in equation (5.23).

5.5 Scale invariance and entropy

If **u** is uniformly scaled by the factor a, the parsimony functional of equation (5.23) becomes:

$$\tilde{S}_P(u) = \begin{cases} C_1 + u^2/2a^2\sigma_0^2 & a\sigma_0 > |u| \\ C_1 + 1/2 + \ln(|u|/a\sigma_0) & a\sigma_0 \le |u| \le a\sigma_\infty \\ C_1 + \ln(\sigma_\infty/\sigma_0) + u^2/2a^2\sigma_\infty^2 & a\sigma_\infty < |u| \end{cases} \quad (5.44)$$

where u now refers to the new, rescaled value. Because of the presence of the logarithm, the gradient in the middle range of u remains unchanged (equal to $|u|^{-1}$); scaling affects the gradient only in redefining the threshold standard deviations σ_0 and σ_∞ to $a\sigma_0$ and $a\sigma_\infty$. In this respect the parsimony functional is scale-invariant. If the limiting values $a\sigma_{0,\infty}$ still bracket the range of "interesting" amplitudes of u, rescaling u does not affect the functional. In a like manner the full-dimensional parsimony functional $S_P(\mathbf{u})$, defined by equation (5.41), is scale-invariant. Among the possible functionals considered in the last section, even the ones derived from the generalized Gaussian family, only the functionals defined in Rules 1 and 2 possess the property of scale invariance. Except for the shift of the threshold values σ_0, σ_∞, the parsimony gradient is unaffected by an overall scaling factor applied to the model.

The multidimensional functional $S_P(\mathbf{u})$ derived in section 5.4 reduces to a simple form: the extensive quantity $\sum_j \ln \sigma_j$. The variable σ_j is an estimate of the standard deviation of the local population about the point u_j. Treating model space as N sample points of an ensemble, we may view the sum as an estimate of the expected value of the logarithm of σ_j^2:

$$S_P = \frac{1}{2}\sum_{j=1}^{N} \ln \sigma_j^2 \approx \frac{N}{2}\mathrm{E}[\ln \sigma^2] \tag{5.45}$$

There is a close relationship between this expression for $S_P(\mathbf{u})$ and the formal definition of entropy. In fact, expression (5.45) is precisely the entropy of a statistically independent, Gaussianly distributed set of N random variables defined by Burg (1975, section III-A). This is Burg's so-called *spectral entropy*. We might wonder whether there is a connection between spectral entropy and the more formal definition of statistical entropy, which is (Shore and Johnson, 1980)

$$S_E \equiv -\int p(\mathbf{u}) \ln p(\mathbf{u})\, d\mathbf{u} \tag{5.46}$$

The answer is yes: expressions (5.45) and (5.46) are equivalent, up to constant terms, if $p(\mathbf{u})$ is jointly Gaussian and independent. The proof of this fact is given in appendix 5.A. The important point to make here is that the variances describing $p(\mathbf{u})$ are themselves variable. Minimizing the parsimony functional $S_P(\mathbf{u})$ is therefore equivalent to minimizing the entropy S_E of the underlying probability density in model space.

Our problem has a direct physical analogy to that of a vibrating lattice of atoms, the so-called Einstein model (Careri, 1984). Each atom occupies a potential well, with the potential energy of each atom being a quadratic function of its displacement from the center of the well. At high enough temperatures, the motions of the atoms at each site are independent, so that no direct correlation can be made between lattice sites. The statistical state of the lattice in this case is completely described by the mean displacements of all atoms from their rest positions, that is, their variances. Let us assume the crystal to be adiabatically isolated from its surroundings, so that its total thermal energy is conserved. By examining the expression for entropy given in equation (5.45), we see that the crystal has highest entropy when the distribution of vibrational energy throughout the crystal is most uniform, or when σ_j is nearly constant throughout. The entropy of the crystal can grow negative without bounds as the vibrational energy characterized by $\sigma_j(u_j)$ becomes concentrated at a few points in the crystal.

In using stochastic inversion, our goal has been to concentrate the solution in model space to as small an area as possible: stochastic inversion, as we have defined it, is a process that drives the solution to *minimum entropy.*

Of course, by specifying a low-end cutoff variance σ_o^2, we are adding the constraint that entropy cannot go to negative infinity. In terms of our analogy of the crystal lattice, atoms in all parts of the lattice must retain a residual amount of vibrational energy.

Burg (1975) proposed maximizing the entropy measure of equation (5.45) in order to make the variances σ_j^2 as uniform as possible, within the constraints imposed by the time series data (in the form of expected values). In the present case, the only physical justification for *minimizing* the entropy functional of equation (5.45) is that we expect the solution to be sparse and clumped. The *maximum entropy method* developed by Burg (1975) and Jaynes (1957) attempts to draw the least inference from available data. *Minimum entropy methods,* such as the present method, attempt to draw the greatest inference from the information at hand.

Before leaving this section, let us digress once more into the subject of maximum entropy. This principle, first formulated by Jaynes in 1956, can be compared to

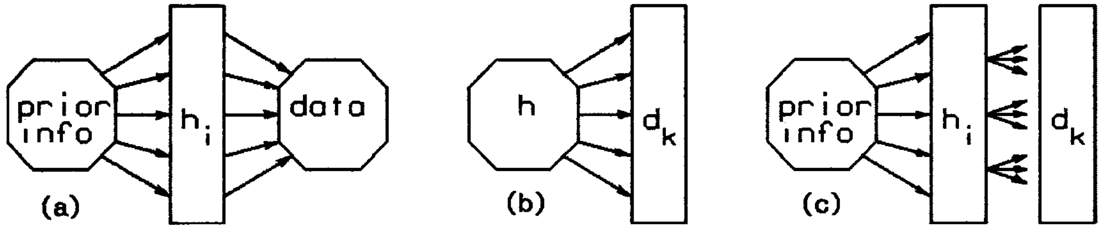

FIG. 5.3. Maximum entropy versus orthodox statistics.

(a) Maximum entropy: one out of a class of possible hypothesis h_i gave rise to the observed data. Each hypothesis is a random process characterized by unknown parameters. Prior information for the class of hypotheses may be available in the form of probabilities. Pick the h_i that has the greatest entropy (or, is most noncommittal).

(b) Orthodox statistics: the underlying hypothesis is given, and a set of data is available. Determine the parameters of the hypothesis, using various statistical measures on the data.

(c) A combination of the two methods: h_i is not completely known, and an additional noise process, which is assumed to be known, causes variations in the data.

orthodox statistics in the following way (Jaynes, 1981).

The principle is to specify a probability $p(h_i)$ over the class of all hypotheses h_i that could have given the data set that was actually observed. See figure 5.3(a). The data are assumed to be perfectly reliable and without noise. In comparison, orthodox statistics considers the observed data set to be embedded in an imaginary class of data sets generated by an underlying hypothesis (figure 5.3(b)).

What are the underlying hypotheses for our case? They can be identified with the family of prior densities $p(\mathbf{u} \mid \boldsymbol{\sigma})$ that are Gaussian and independent, but have unknown variances σ_i^2. The class of all possible hypotheses $\{h_i\}$ is characterized by the N-dimensional space of variances σ_i^2, $i = 1, N$. Each variance may range from σ_0^2 to σ_∞^2. Besides the assumption that the densities are Gaussian, other prior information is provided by $p(\boldsymbol{\sigma})$, or equivalently (as seen in section 5.2) by the rule

E[σ_i^{-2} | u].

We can also assume that the true data set is embedded in a class of data sets $\{\mathbf{d}_k\}$ formed by perturbing the underlying data with some additive zero-mean Gaussian noise. Graphically presented, the estimation problem described in this chapter is a combination of the first two cases shown in figure 5.3. The starting point common to all three principles illustrated in figure 5.3 is the MAP estimator derived from Bayes' rule.

5.6 Gradient descent algorithms

As shown in section 5.2, the nonlinear system of equations to solve for the stochastic inverse is

$$\mathbf{g} = \frac{1}{\tilde{\sigma}^2}\mathbf{u} + \frac{1}{\sigma_n^2}\mathbf{L}^T(\mathbf{Lu} - \mathbf{d}) = 0 \tag{5.15}$$

in which $\tilde{\sigma}^2(\mathbf{u})$ is a monotonic function of the averaged **u**. The noise variance σ_n^2, on the other hand, is assumed to be constant, and we can multiply through equation (5.15) by σ_n^2. The minimum and maximum values in the diagonal matrix (call it $\mathbf{D} \equiv \sigma_n^2/\boldsymbol{\sigma}_o^2$) are given by $\sigma_n^2/\sigma_\infty^2$ and σ_n^2/σ_o^2, respectively. The upper limit on the condition number of **D** is therefore $\sigma_\infty^2/\sigma_o^2$. Projection $\mathbf{P}_d$, which limits the range in offset (the input space of $\mathbf{L}^T$), has in this chapter been incorporated into **L**. Now the threshold variances σ_o^2 and σ_∞^2 influence the solution in two fundamentally different ways. If $\mathbf{L}^T\mathbf{L}$ is invertible, σ_∞ has a minimal effect on the solution, but if $\mathbf{L}^T\mathbf{L}$ has a null space that lies principally in a region where σ^2 equals σ_∞^2, the presence of the σ_∞^2 terms (small diagonal elements) makes the system ill-conditioned. The terms $\sigma^2 = \sigma_o^2$ have another effect: they drive the corresponding terms of the solution to zero. The smaller σ_o^2 is, the greater this effect (i.e., the greater the weighting on the diagonal of system 5.15).

Because an expression for the gradient is already known, the most straightforward approach to solving equation (5.15) is to apply a steepest descent algorithm. However, the classical formula for the step size α,

$$\alpha = \frac{\mathbf{g}^T\mathbf{g}}{\mathbf{g}^T(\mathbf{D} + \mathbf{L}^T\mathbf{L})\mathbf{g}} \tag{5.47}$$

is not strictly valid because the system is nonlinear. Such an explicit formula for the step size must be replaced by a line search which minimizes the objective functional along the direction of the current gradient. For a continuous gradient surface, the line search is equivalent to finding a new gradient normal to the starting gradient along the line parametrized by, say, α. For our nonlinear system, the stopping point for the line search is the point where

$$0 = \mathbf{g}_o^T\mathbf{g} = \mathbf{g}_o^T(\mathbf{D} + \mathbf{L}^T\mathbf{L})\mathbf{u} - \mathbf{g}_o^T\mathbf{L}^T\mathbf{d} \tag{5.48}$$

where $\mathbf{g}_o$ is the initial gradient, $\mathbf{u}_o$ is the initial point, and $\mathbf{u}$ is the desired stopping point along the line $\mathbf{u} = \mathbf{u}_o - \alpha\mathbf{g}_o$. Now $\mathbf{D}$ is a function of $\mathbf{u}$, and consequently of α:

$$0 = \mathbf{g}_o^T\mathbf{g} = \mathbf{g}_o^T(\mathbf{D}[\mathbf{u}] - \mathbf{D}[\mathbf{u}_o])\mathbf{u}_o + \mathbf{g}_o^T\mathbf{g}_o - \alpha\mathbf{g}_o^T(\mathbf{D}[\mathbf{u}] + \mathbf{L}^T\mathbf{L})\mathbf{g}_o \tag{5.49}$$

One way to solve for α is by the recursion formula

$$\alpha = \frac{\mathbf{g}_o^T\mathbf{g}_o + \mathbf{g}_o^T(\mathbf{D}[\mathbf{u}] - \mathbf{D}[\mathbf{u}_o])\mathbf{u}_o}{\mathbf{g}_o^T\mathbf{L}^T\mathbf{L}\mathbf{g}_o + \mathbf{g}_o^T\mathbf{D}[\mathbf{u}]\mathbf{g}_o} \tag{5.50}$$

The extra work (over the classical formula) at each iteration involves calculating the terms $\mathbf{D}$ and forming the inner products $\mathbf{g}_o^T\mathbf{D}\mathbf{u}_o$.

The actual algorithm implemented for the examples that follow is a *projected gradient descent* algorithm. The descent direction used is a projection of the gradient onto some subspace of model space. By constraining the descent direction with this projection we can focus on what we know to be the region in model space where most of the solution is going to concentrate.

There are some theoretical reasons why the projected gradient method may perform better than a standard steepest descent method in solving the nonlinear system (5.15). Let us first outline the algorithm and derive the expected convergence properties of the projected gradient algorithm; we can then compare these with the well-known convergence properties of steepest descent.

The projected gradient algorithm used in the stochastic inversion examples of the following sections is described in table 5.1. The basic idea of the algorithm is to first concentrate on convergence in the region of the velocity panel in which most of the solution is expected to lie. This region, at the start, is the region of highest variances, which must be determined in a bootstrap manner since it is not known a priori. The gradient is in turn projected onto a family of subspaces in the model domain with gradually decreasing variances; each of these subspaces (hopefully) defines a population of points with a Gaussian distribution uniquely characterized by its variance. Within these subspaces, the gradient is linear and we can use the classical formula for the step size in equation (5.47), therefore bypassing the problem of having to use the nonlinear line search in equation (5.50). Search directions are set up by defining a family of projections $\mathbf{P}_k$ that are defined by the current estimate of the variance σ^2. The projections are ordered from high- to low-variance regimes, and line searches are taken in directions $\mathbf{Q}_k \mathbf{g}$ over increasingly larger sections of the velocity panel. For the details of the algorithm, see table 5.1.

Let us now examine a few convergence properties of the algorithm of table 5.1. The convergence rate of steepest descent is given by the classical formula [Luenberger, sec. 7.6]

$$E(\mathbf{u}_{k+1}) \leq \left(\frac{1 - a/A}{1 + a/A}\right)^2 E(\mathbf{u}_k) \qquad (5.51)$$

in which $E(\mathbf{u}_k)$ is the error between the estimated solution $\mathbf{u}_k$ and the true solution $\mathbf{u}$ given by

$$E(\mathbf{u}_k) \equiv \frac{1}{2}(\mathbf{u}_k - \mathbf{u})^T(\mathbf{D} + \mathbf{L}^T\mathbf{L})(\mathbf{u}_k - \mathbf{u}) \qquad (5.52)$$

Scalars a and A are the smallest and largest eigenvalues, respectively, of the positive definite linear system $\mathbf{D} + \mathbf{L}^T\mathbf{L}$. The error bound for a projected gradient algorithm is comparatively larger (Luenberger, 1973, sec. 7.11):

$$E(\mathbf{u}_{k+1}) \le \left[1 - \beta_k \frac{a}{A}\right] E(\mathbf{u}_k) \tag{5.53}$$

where the additional term β_k is the ratio of the gradient energy in the projection, $\mathbf{g}_k^T \mathbf{P}_k \mathbf{g}_k$, to the total gradient energy, $\mathbf{g}_k^T \mathbf{g}_k$.

If all members of the projection family $\{\mathbf{P}_k\}$ $(k = 1, n)$ are disjoint and sum to the identity, then $\sum \beta_k = 1$. When the projection family $\{\mathbf{P}_k\}$ is used, the projected gradient algorithm approaches the performance of one steepest descent iteration only after all n projected directions have been searched. To see this, compare the nth cascaded error in equation (5.53) with the steepest descent error in equation (5.51):

$$\prod_{k=1}^{n} \left[1 - \beta_k \frac{a}{A}\right] \ge \left[1 - \frac{a}{A}\right] \ge \left[\frac{1 - a/A}{1 + a/A}\right]^2 \tag{5.54}$$

Equality in this expression is approached as $\beta_k a/A \to 0$. Each step in a projected gradient algorithm may cost as much as one full step of the steepest descent algorithm, which is bad news. To justify using the projected gradient method over steepest descent, we must get a better bound than (5.53) on the local convergence rate of the projected gradient algorithm.

Let us trace the steps of the proof of the error bound in equation (5.53). Consider the linearized problem $\mathbf{Au} = \mathbf{d}$ where $\mathbf{A} = \mathbf{D} + \mathbf{L}^T\mathbf{L}$ for some fixed choice of the diagonal variance matrix $\mathbf{D}$. The relative decrease in error E_r from the kth to the $k+1$st iteration is defined to be

$$E_r \equiv \frac{E(\mathbf{u}_k) - E(\mathbf{u}_{k+1})}{E(\mathbf{u}_k)} \tag{5.55}$$

Let $\mathbf{x}_k \equiv \mathbf{u}_k - \mathbf{u}$ be the error vector. The kth gradient is then $\mathbf{g}_k = \mathbf{A}\mathbf{x}_k$. The

Projected gradient descent algorithm

```
Begin

(1) Unconstrained gradient step.
    $\mathbf{g} := \mathbf{L}^T\mathbf{L}\mathbf{u} - \mathbf{L}^T\mathbf{d}$
    $\alpha := \frac{\mathbf{g}^T\mathbf{g}}{\mathbf{g}^T\mathbf{L}^T\mathbf{L}\mathbf{g}}$
    $\mathbf{u} := \mathbf{u} - \alpha\mathbf{g}$

Loop

(2) Estimate variances by applying an averaging time window to u.
    $\sigma^2_{p,\tau} := \frac{1}{2T+1}\sum_{j=-T}^{T} u^2_{p,\tau+j}$
    $\mathbf{D}_{p,\tau} := \sigma^2_n / \sigma^2_{p,\tau}$

(3) Partition u into $n+1$ constant-variance regions from 0 to $\infty$ by defining projections $\mathbf{P}_k$. The variances that divide the regions are spaced in equal log increments.
    $q_0 := \ln\sigma_0$
    $q_n := \ln\sigma_\infty$
    $\Delta q := (q_n - q_0)/n$
    for $k = 1$ to $n-1$
        $q_k := q_0 + k\Delta q$
    next $k$

    $q_{-1} := 0$
    $q_n := \infty$
    for $k = 0$ to $n$
        $(\mathbf{P}_k)_{p,\tau} := \begin{cases} 1 & q_{k-1} \le \ln\sigma_{p,\tau} < q_k \\ 0 & \text{otherwise} \end{cases}$
    next $k$
```

(Continued on next page)

TABLE 5.1. Projected gradient algorithm to solve the nonlinear system of equations (5.15) with σ specified by (5.17). Initially σ_n, σ_∞, σ_0 must be selected, but may be allowed to relax to fit the data. For example σ_∞ may be set to the 99th quantile of **u**, and σ_0 set to a certain fixed log range below σ_∞. Lower case bold characters are vectors indexed by p,τ (data **d** is indexed by h,t) and upper case bold characters are diagonal matrices (**D**, **P**, **Q**) or operators ($\mathbf{L}^T$, **L**). The unconstrained step at the beginning is needed to get an initial estimate of σ.

```
(4) Projected gradient steps.
    $\mathbf{u}_{n+1} := \mathbf{u}$
    for $k = n$ to 0 decremented
        $\mathbf{Q}_k := \sum_{j=k}^{n} \mathbf{P}_j$
        $\mathbf{g} := \mathbf{Q}_k(\mathbf{L}^T\mathbf{L} + \mathbf{D})\mathbf{u}_{k+1} - \mathbf{Q}_k\mathbf{L}^T\mathbf{d}$
        $\alpha := \frac{\mathbf{g}^T\mathbf{g}}{\mathbf{g}^T(\mathbf{L}^T\mathbf{L} + \mathbf{D})\mathbf{g}}$
        $\mathbf{u}_k := \mathbf{u}_{k+1} - \alpha\mathbf{g}$
    next $k$
    $\mathbf{u} := \mathbf{u}_o$
Stopping criterion goes here.
Go to Loop.
```

TABLE 5.1 (continued). Projected gradient descent algorithm.

search will be made in a direction defined by the projection of the gradient, $\mathbf{h}_k \equiv \mathbf{P}_k\mathbf{g}_k$, and the corresponding step size is $\alpha = \mathbf{h}_k^T\mathbf{h}_k / \mathbf{h}_k^T\mathbf{A}\mathbf{h}_k$. Under these definitions E_r becomes

$$E_r = \frac{\mathbf{x}_k^T\mathbf{A}\mathbf{x}_k - (\mathbf{x}_k - \alpha\mathbf{h}_k)^T\mathbf{A}(\mathbf{x}_k - \alpha\mathbf{h}_k)}{\mathbf{x}_k^T\mathbf{A}\mathbf{x}_k} = \frac{2\alpha\mathbf{h}_k^T\mathbf{A}\mathbf{x}_k - \alpha^2\mathbf{h}_k^T\mathbf{A}\mathbf{h}_k}{\mathbf{g}_k^T\mathbf{A}^{-1}\mathbf{g}_k} \qquad (5.56)$$

Substituting for α gives

$$E_r = \left[\frac{\mathbf{h}_k^T\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{A}\mathbf{h}_k}\right]\left[\frac{\mathbf{h}_k^T\mathbf{h}_k}{\mathbf{g}_k^T\mathbf{A}^{-1}\mathbf{g}_k}\right] \qquad (5.57)$$

Now $\mathbf{g}_k^T\mathbf{A}^{-1}\mathbf{g}_k \le a^{-1}\mathbf{g}_k^T\mathbf{g}_k$ where a is the smallest eigenvalue of A. Together with the definition $\beta_k \equiv (\mathbf{h}_k^T\mathbf{h}_k)/(\mathbf{g}_k^T\mathbf{g}_k)$, the following bound for the right-hand factor in (5.57) can be given:

$$E_r \ge \frac{\mathbf{h}_k^T\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{A}\mathbf{h}_k}\,\beta_k a \qquad (5.58)$$

A bound better than $1/A$ can be given for the left-hand factor of equation (5.57) by noting that $\mathbf{h}_k^T\mathbf{A}\mathbf{h}_k = \mathbf{h}_k^T\mathbf{P}_k\mathbf{A}\mathbf{P}_k\mathbf{h}_k$, so that

$$E_T \geq \frac{1}{B_k} \beta_k a \tag{5.59}$$

where B_k is the largest eigenvalue of the truncated operator $\mathbf{P}_k \mathbf{A} \mathbf{P}_k$. The convergence inequality is found by substituting the definition of E_T (equation 5.55) into equation (5.59). This gives us our better error bound:

$$E(\mathbf{u}_{k+1}) \leq \left[1 - \beta_k \frac{a}{B_k} \right] E(\mathbf{u}_k) \tag{5.60}$$

Much of the spread in the range of eigenvalues of **A** can be attributed to the diagonal matrix **D**. If we normalize the velocity stack operator by dividing through by the number of summed traces, the resulting maximum eigenvalue of $\mathbf{L}^T\mathbf{L}$ cannot be much more that unity. The minimum eigenvalue of $\mathbf{L}^T\mathbf{L}$ is of course zero. In contrast, the terms making up **D** range from $\sigma_n^2 / \sigma_\infty^2$, which might be very small, to σ_n^2 / σ_0^2 which might be very large.

If we can assume that most of the scaling imbalance in the linear system **A** is due to the range of variances in **D**, then, with a suitable choice for the family of projections $\{\mathbf{P}_k\}$ it is possible to get faster convergence with the projected gradient algorithm (compared to that of steepest descent). In table 5.1 the family of projections $\{\mathbf{Q}_k\}$ is defined as the sequence of partial sums of $\{\mathbf{P}_k\}$ that ranges from 0 to the identity **I** as the iterations proceed. The expected maximum eigenvalues of $\mathbf{Q}_k \mathbf{A} \mathbf{Q}_k$ are

$$B_k \approx 1 + \frac{\sigma_\infty^2}{\sigma_k^2} \tag{5.61}$$

as σ_k^2 ranges from σ_∞^2 to σ_0^2. At the same time, β_k ranges from 0 to 1. If the energy in the gradient is concentrated in the high-variance populations of **u**, then the growth of β_k against B_k, as the iteration loop proceeds, is something like that shown in figure 5.4(a). The corresponding plot of $\beta_k a / B_k$ versus β_k is sketched in figure 5.4(b). By comparing equations (5.51) and (5.60), we can see that wherever the curve rises above the approximate horizontal line $\beta_k a / B_k = 4a / A$, the con-

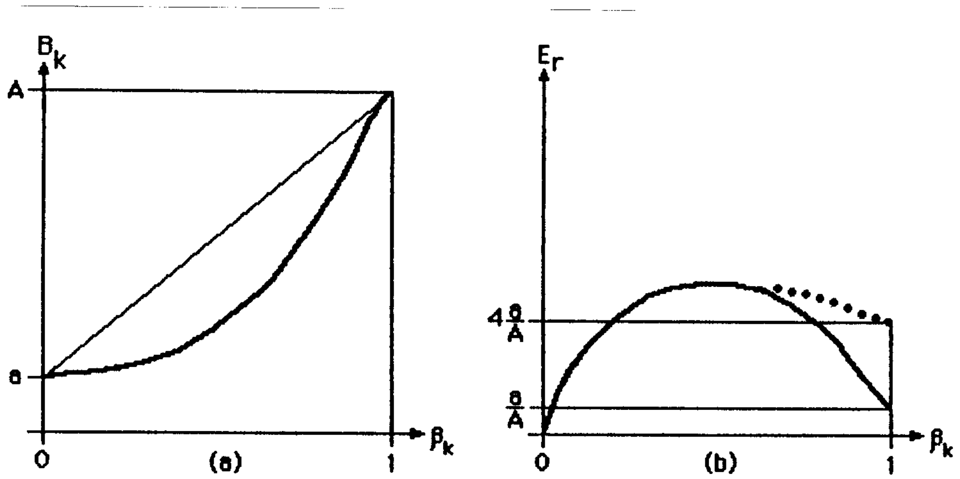

FIG. 5.4. Hypothetical behavior of the norm of $\mathbf{Q}_k\mathbf{A}\mathbf{Q}_k$.

(a) β_k is the ratio of projected gradient energy to total gradient energy. B_k is the norm of $\mathbf{Q}_k\mathbf{A}\mathbf{Q}_k$. As the projections $\mathbf{Q}_k$ grow from 0 to $\mathbf{I}$, β_k ranges from 0 to 1 while B_k ranges from a to A, the smallest to largest eigenvalue of **A**. If the energy in the gradient concentrates in the high-variance populations of **u**, then the growth of β_k will be greater than that of B_k.

(b) As a result, the quantity $E_r = \beta_k a / B_k$, which determines the convergence rate (equation 5.60), may be large in the middle ranges of β_k. To get superconvergence (greater that steepest descent), E_r must surpass $4a/A$. Finally as $\mathbf{Q}_k$ approaches the identity matrix, the performance of the projected gradient algorithm will approach the performance of steepest descent (the dotted line) from above.

vergence rate is faster than that of steepest descent.

In summary, the projected gradient algorithm allows one to concentrate on matching the data set's high-amplitude events before turning to the smaller amplitude events. Being able to do this is especially important in the present case: the dimensionality of the system to be solved is so large, that any iterative algorithm will have to eventually stop far short of complete convergence.

5.7 Synthetic data inversion

Before presenting examples of the stochastic inversion algorithm on actual data, let us first consider two synthetic cases. The first case, an ideal noise-free data set, is shown in figure 5.5. The data set **d** of figure 5.5 has been generated by velocity stacking (via **L**) the set of six "point sources" in the velocity panel labeled Syn. The waveform for each source is a sinc ($\sin(t)/t$) function in time. The stochastic inversion algorithm was subsequently applied to **d**, and after ten iterations resulted in the estimate **u**. The estimate **u** is very close to the exact solution Syn; in fact, if **u** is subsequently stacked with the operator **L**, the residual **d** − **Lu** (again, on figure 5.5) is nearly zero.

The inverse velocity stack **u** has converged to all six "reflections" in velocity space, excepting the reflector with the shallowest zero-offset intercept time. The shallower a reflector is, the more sensitive its moveout is to velocity variations. This sensitivity to velocity to shallow intercept times, coupled with the possibility of not sampling the velocity axis fine enough, may prevent the gradient from converging to the true solution at shallow times.

A solution to the problem of unfocused reflectors at shallow times would be to sample velocities in velocity space with a sampling interval proportional to time. There are no constraints on the sampling scheme for **L**, and as long as the adjoint $\mathbf{L}^T$ is known, the sampling scheme in velocity space might be freely chosen to adapt to the problem at hand.

The initial data set **d** + **n** of the second synthetic case we shall consider is shown in figure 5.6. In this case, Gaussian noise has been added to the data of figure 5.5. The signal-to-noise RMS amplitude ratio is 1:2. Five out of the six original events have been detected by the stochastic estimate **u** (shown in figure 5.6), despite the high noise level. In addition, a few spurious events at late times have been added to the estimate. The remaining two panels of figure 5.6, **Lu** and **d** − **Lu**, are, respectively, the resulting modeled data and the residual.

The stochastic inversion method in table 5.1 requires the assignments of a high and low signal variance σ_∞^2 and σ_o^2, together with a noise variance σ_n^2. For the noise-free case of figure 5.5, the noise variance could be set to an arbitrarily low,

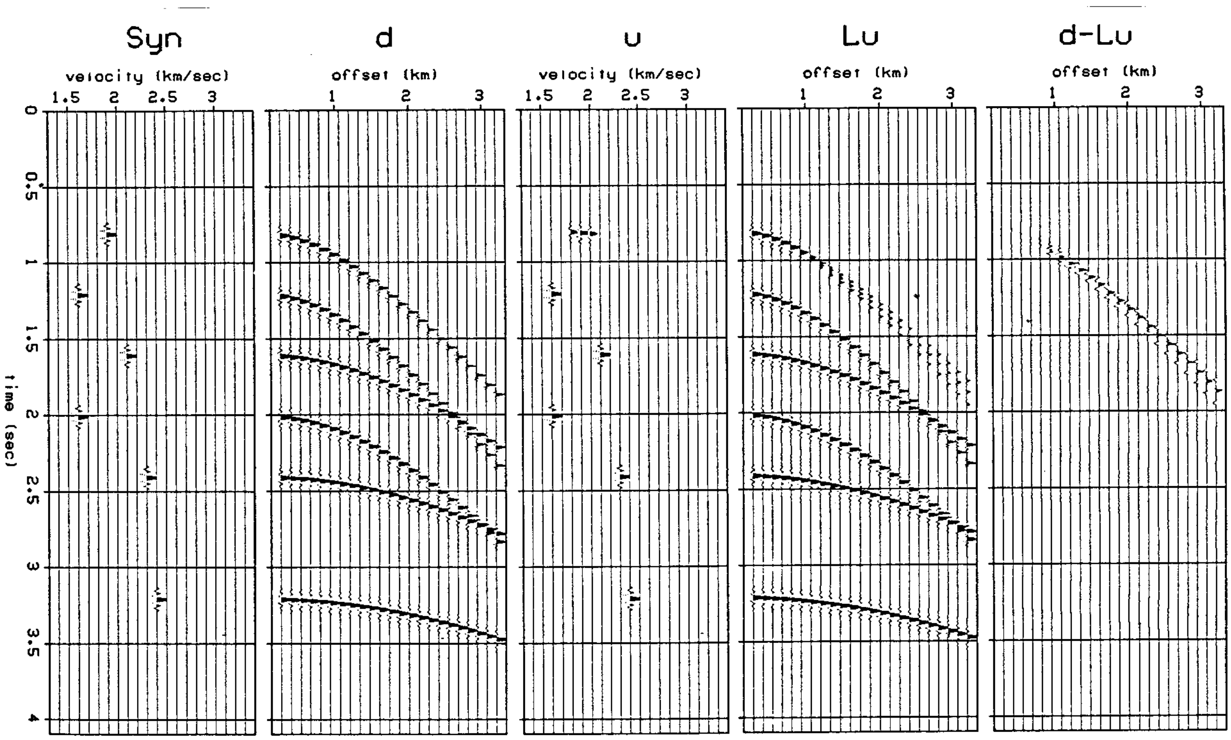

FIG. 5.5. Inversion on a synthetic noise-free model. Panel **Syn** is the desired solution; it was used to create (via velocity stack **L**) the synthetic data **d** on the next panel. Ten iterations of the gradient descent algorithm of table 5.1, applied to **d** yields **u** in the center panel. Panel **Lu**, the velocity stack of **u**, approximates the initial data set; the rightmost panel is the difference **d** − **Lu**.

nonzero value. For the noisy case of figure 5.6, the noise variance was iteratively estimated from the error measure as the algorithm proceeded. The signal variances were also allowed to adapt to the model: σ_∞ was either set to the greatest value of $|u|$ seen in the current estimate of **u**, or to some high quantile of **u**; while σ_0 was set to a small fraction of σ_∞, typically $10^{-4}\sigma_\infty$. The remaining examples of this chapter follow this procedure to determine the variance values of σ_∞^2, σ_0^2, and σ_n^2 at each iteration.

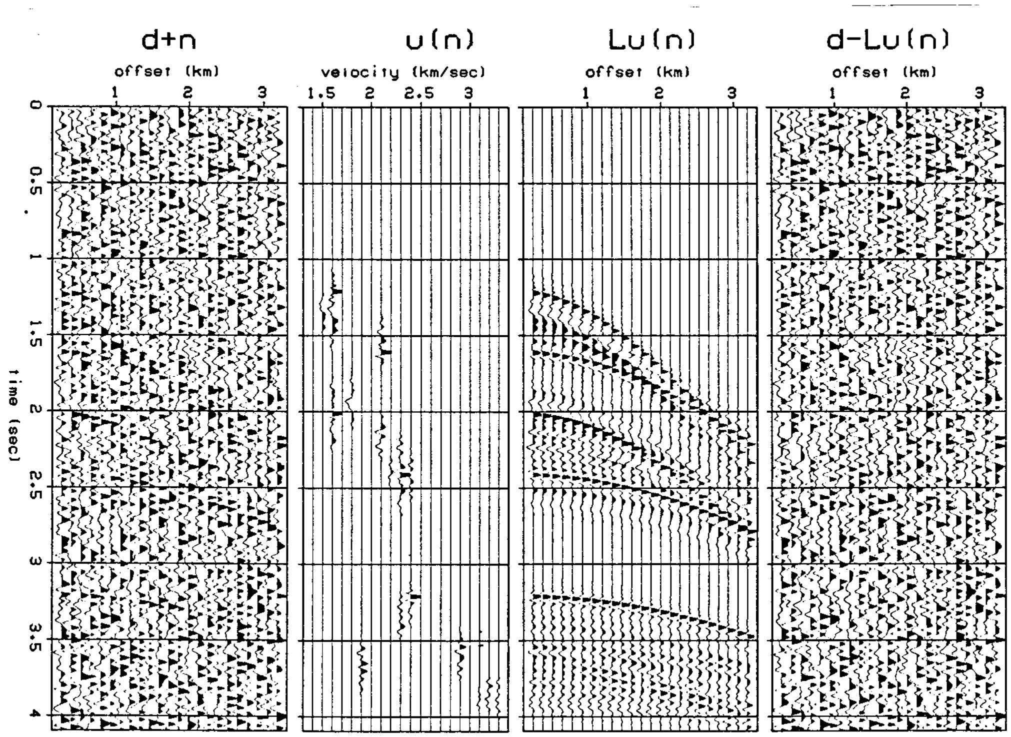

FIG. 5.6. Inversion on a synthetic noisy model. Bandlimited noise was added to panel **d** of figure 5.5 to create the noisy data on the left panel, **d** + **n**. The signal-to-noise ratio on **d** + **n** is 0.5. Panel **u(n)** is the result of 10 iterations of the gradient descent algorithm of table 5.1 on the noisy data. This panel should be compared to **Syn** of figure 5.5. Panel **Lu(n)** is the velocity stack of **u**, and panel **d** − **Lu(n)** is the difference between the first and third panels.

5.8 Real data inversion: Amoco Grand Banks

The next example deals with the real data of figure 5.7: a common-midpoint gather collected by Amoco in 1974 on the Grand Banks of offshore Newfoundland. The resulting velocity stack **u** after seven iterations through the stochastic inversion algorithm appears on the left in figure 5.8. For comparison, the forward velocity stack on the data, $\mathbf{L}^T\mathbf{d}$, is shown on the right. The comparison of **Lu** to **d** is shown in figure 5.7, and the comparison of **d** with the residual **d** − **Lu** is shown in figure 5.9.

There are some obvious events on both the original gather and the residual of figure 5.9 that seem not to satisfy the hyperbolic moveout assumption. These

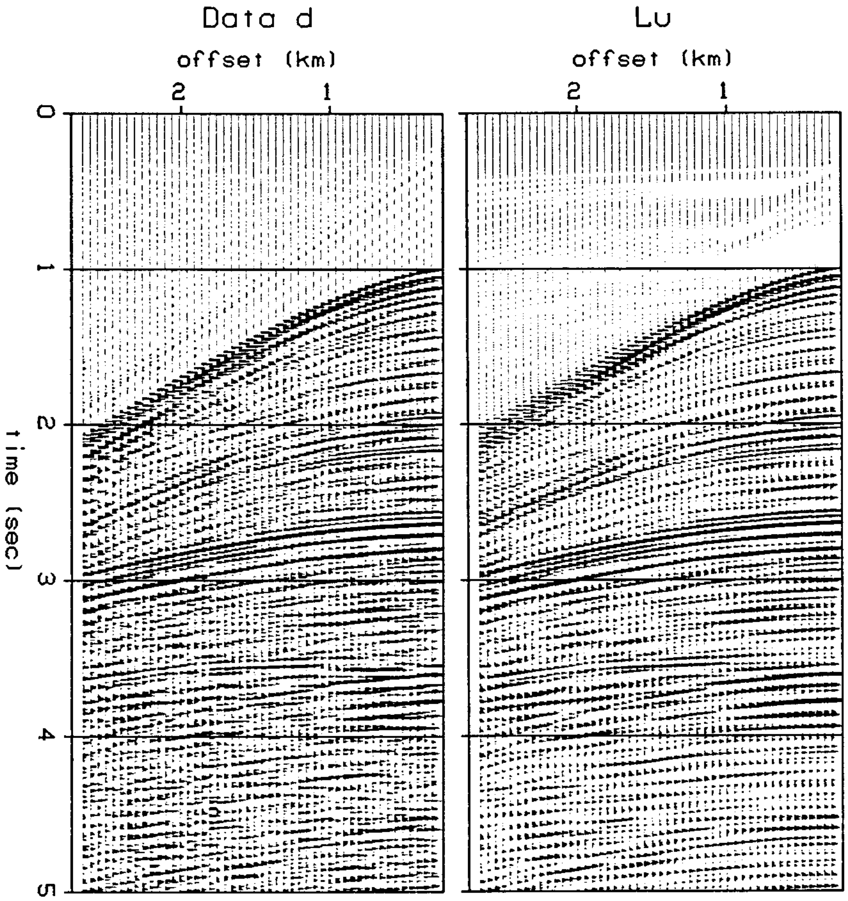

FIG. 5.7. Amoco Grand Banks: data **d** and **Lu**. The left panel is a common midpoint gather (courtesy of Amoco) from Grands Banks, offshore Newfoundland. The offset interval is 50 meters, the time sample interval is 8 msec. This gather was shot in an area of strong sea-floor multiples and with large variations in sea-floor elevation. The right panel is the modeled common midpoint gather created by velocity stacking the stochastic inverse **u** of figure 5.8.

events, at approximately 3.6 seconds on the gather, are water-layer (or *peg-leg*) multiples off the strong primary reflector at 2.6 seconds. The nonhyperbolic nature of the arrival times of these events can be explained by modeling the traveltimes with ray tracing. The model shown in figure 5.10 was generated with the aid of two pieces of information: first, velocities measured from the velocity stack inverse **u**; second, the sea floor topography measured from the stacked section. The interval velocities below the sea floor were chosen to increase linearly from 1.6 to 3

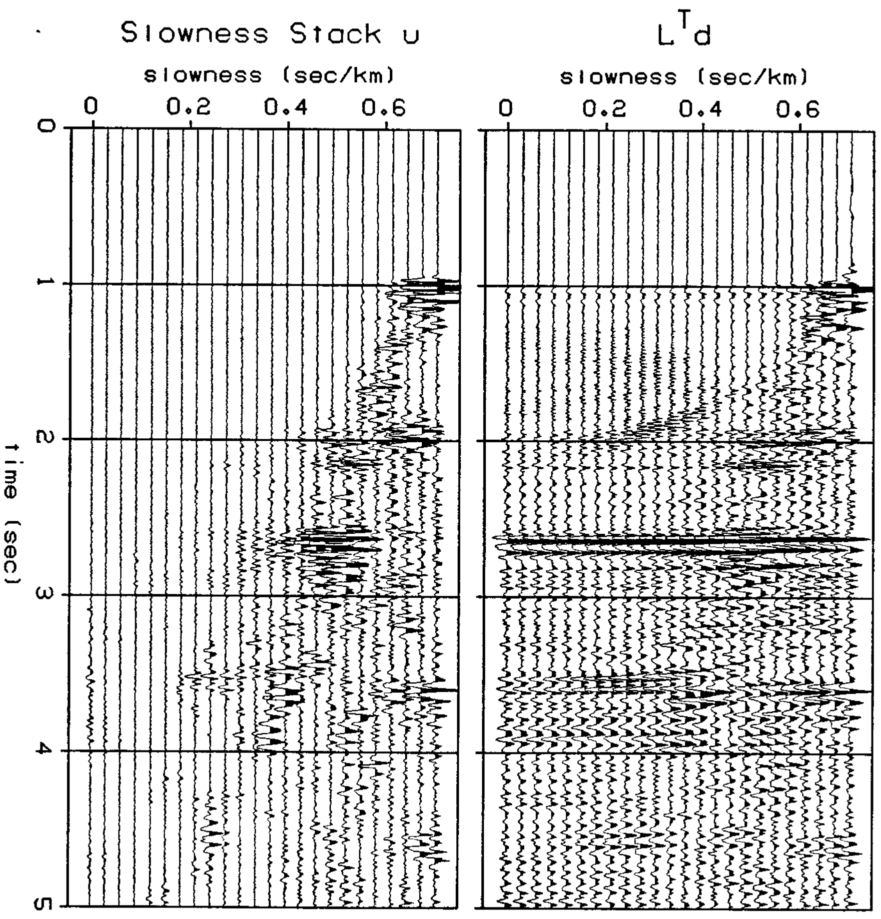

FIG. 5.8. Amoco Grand Banks: **u** and $\mathbf{L}^T\mathbf{d}$. The left panel is the inverse velocity stack **u**, the result of 7 iterations of the gradient descent algorithm using data **d** of figure 5.7 as input. The right panel is the velocity stack $\mathbf{L}^T$ applied directly to the data **d**. The slowness interval on each plot is 0.03 sec/km.

km/sec, which are consistent with the stacking velocities seen on **u**.

The two traced rays on figure 5.10 model the two possible paths the water-layer multiple may take at the highest offset in figure 5.9 (about 2.6 km). Because of the variable depth of the sea floor, the two events, representing energy traveling from a common source on the right to a common receiver on the left, take different, unsymmetric paths. Consequently the events have different traveltimes.

In figure 5.11 the two independent peg-leg paths are modeled for all offsets on the original gather. As the offset from source to receiver decreases to zero, the

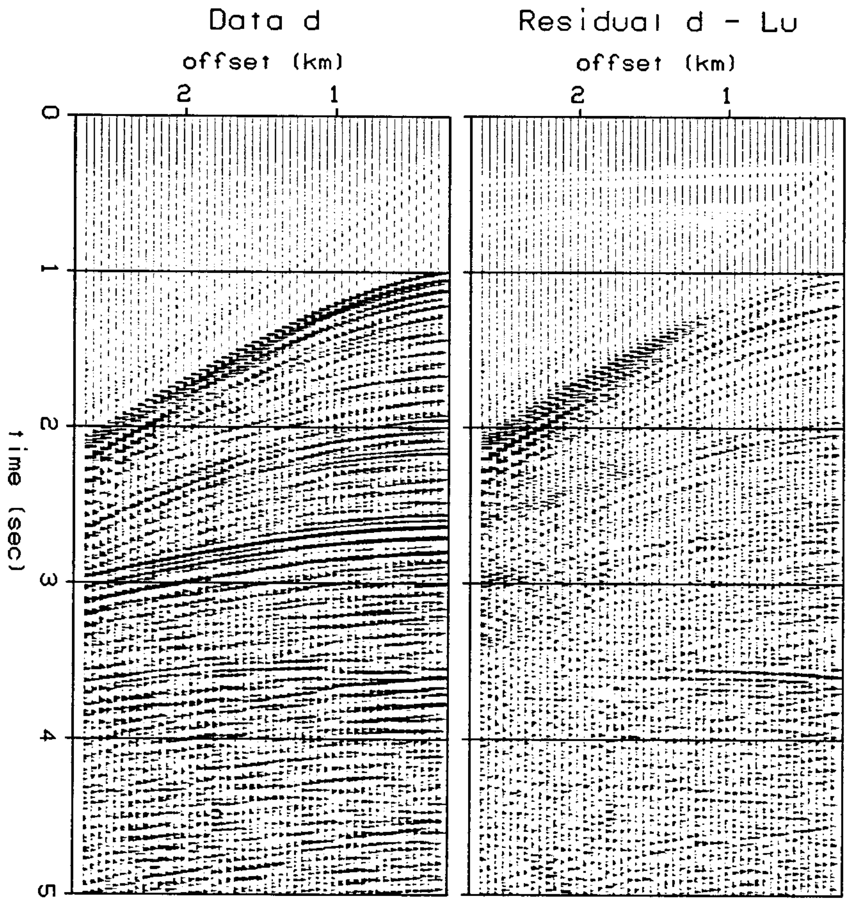

FIG. 5.9. Amoco Grand Banks: comparison of data with residual. On the right is the difference between the data and modeled data panels of figure 5.7. For comparison the gather of figure 5.7 is re-illustrated on the left.

two pegleg paths, labeled slow and fast, converge to a common zero-offset path. From the traveltimes of all rays on figure 11 can be generated a synthetic gather: it is shown on the right in figure 5.12. The synthetically-generated peg-leg multiples match well all features of the corresponding multiples on the real data (the left panel of figure 5.12). In particular, the two multiples at high offsets are separated by a traveltime of 0.3 seconds. The peg-leg multiples might be better fit by a pair of hyperbolas whose axes are translated out to offsets of approximately ±1 km.

The solution **u** in figure 5.8 was essentially found by fitting a family of hyperbolae, with a common axis at zero offset, to the data. Moreover, the solution is able

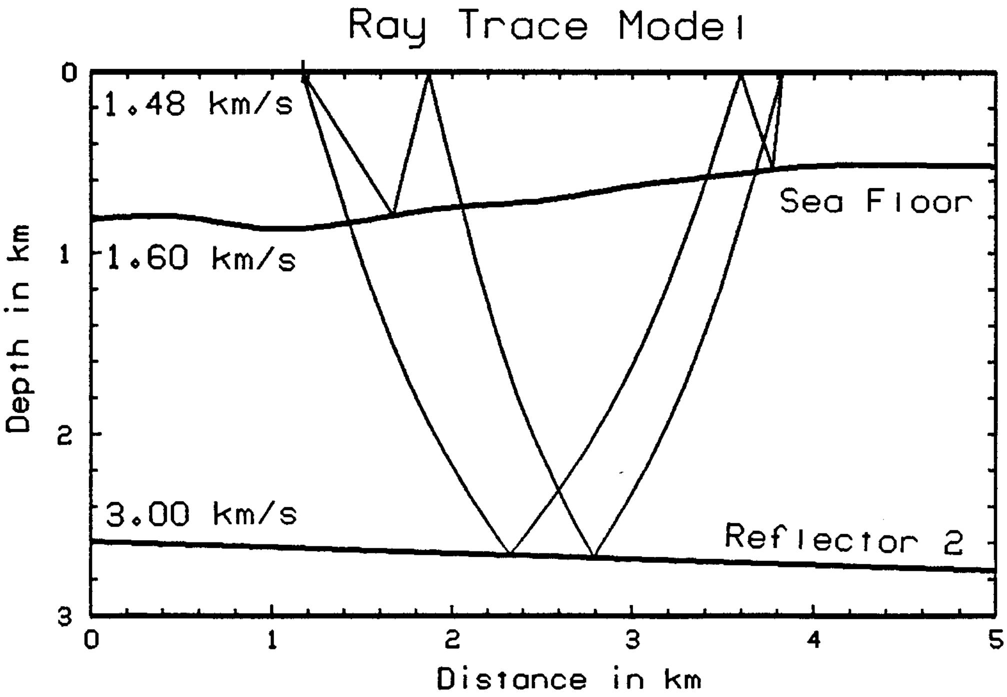

FIG. 5.10. Amoco Grand Banks: ray tracing model. This panel models the sea floor topography and the topography of the strong reflector at 2.6 seconds (determined from the stacked section) in the vicinity of the common midpoint gather of figure 5.7. In fact, the two rays shown above model the propagation from the source (on the left) to the farthest-offset receiver (on the right) of the gather, the ray being allowed to bounce once off the sea floor. Water velocity is 1.48 km/sec, and the subsurface velocity to the strong reflector is a linear function of depth from 1.60 to 3.00 km/sec.

to model to a good degree the peg-leg multiples which cannot be fit by any one member of the hyperbolic family. Compare the left and right panels of figure 5.9: that portion of the peg-leg multiple which is least successfully fit is obviously the part with a reverse slope.

Returning to the velocity stack of figure 5.8, we can see the consequences of the attempt to fit the nonhyperbolic peg-leg events at approximately 3.6 seconds. At this point on the velocity stack, apparent events range from a slowness of 0.7 sec/km (below water velocity) to 0.2 sec/km: i.e., the energy is not localized on the stack. By comparison, the strong event at 2.6 seconds on the velocity stack,

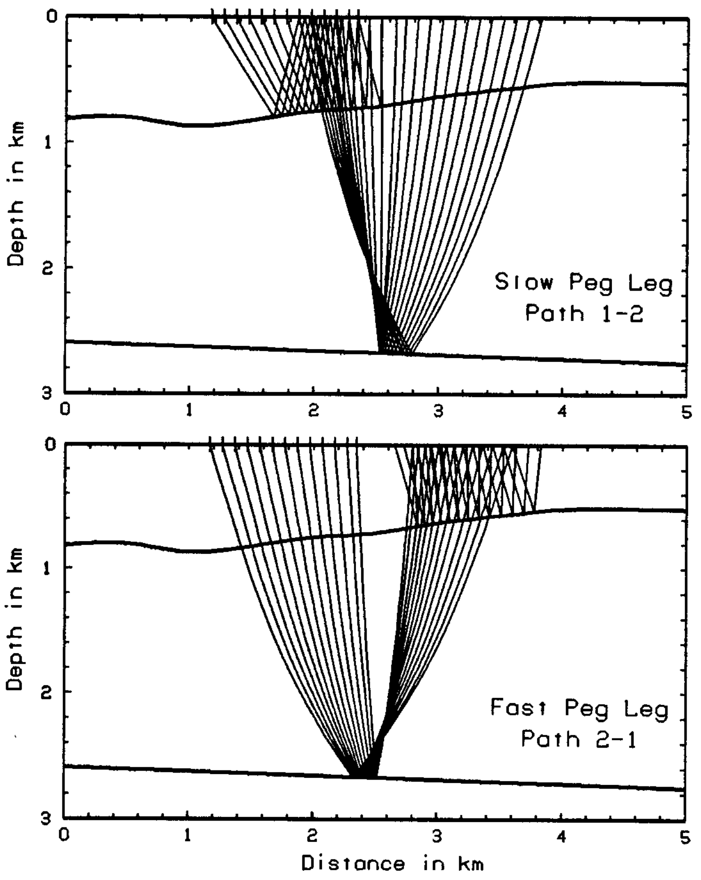

FIG. 5.11. Amoco Grand Banks: the two possible peg-leg paths. Ray tracings of the two possible peg-leg raypaths from the strong reflector '2' that bounce once off the sea floor reflector '1'. All rays shown have a common midpoint. The family of raypaths in the upper panel have slower traveltimes than the family of raypaths in the lower panel because of the greater water depth that the slower peg-leg travels through. The model of figure 5.10 was used to generate the rays.

responsible for the formation of the pegleg multiples, has focused well at 0.5 sec/km.

We might summarize the behavior of the stochastic inverse in this example with the following three points: hyperbolic events focus in velocity space; nonhyperbolic events with positive moveout (or, positive dip) are modeled, but by "unfocused" events extending over a range of slownesses in velocity space; finally, events with reverse moveout have a tendency to be left on the residual.

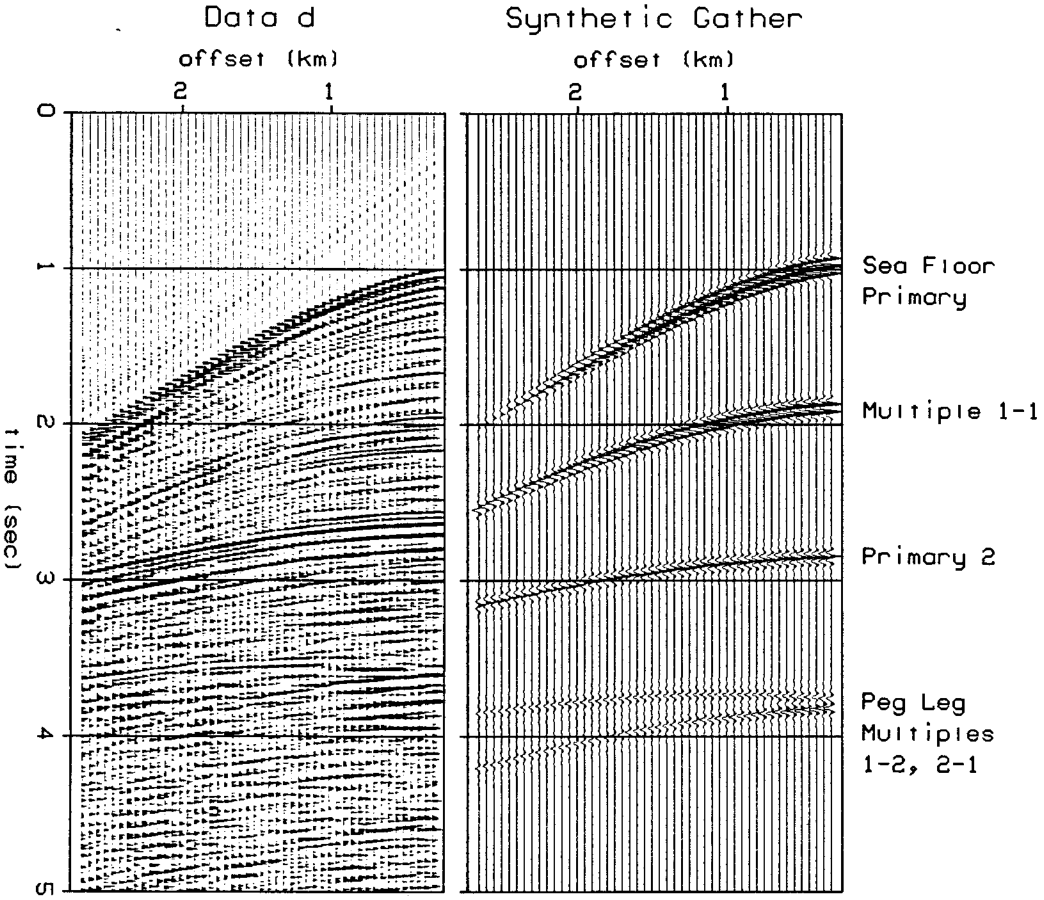

FIG. 5.12. Amoco Grand Banks: synthetic gather. The modeled common midpoint gather on the right panel is a summary of the traveltimes of various rays traced from figure 5.10. For comparison the real common midpoint gather of figure 5.7, which the synthetic is meant to fit, is shown on the left.

The standard procedure of stacking is equivalent to sampling the velocity stack $\mathbf{L}^T\mathbf{d}$ of figure 5.8 along a preselected time-velocity curve, which defines the stacking velocities. Given the opportunity to increase the resolution of events on the velocity stack **u**, we can expect to attain a better separation between primaries and multiples on the stack; thus an alternative to stacking is to sample the velocity-stack panel **u** of figure 5.8 along the time-velocity curve. Figure 5.13 is an example of such a process: the dark borders outline a window in the velocity panel from which the desired events are pulled to make the stack. Outside of this

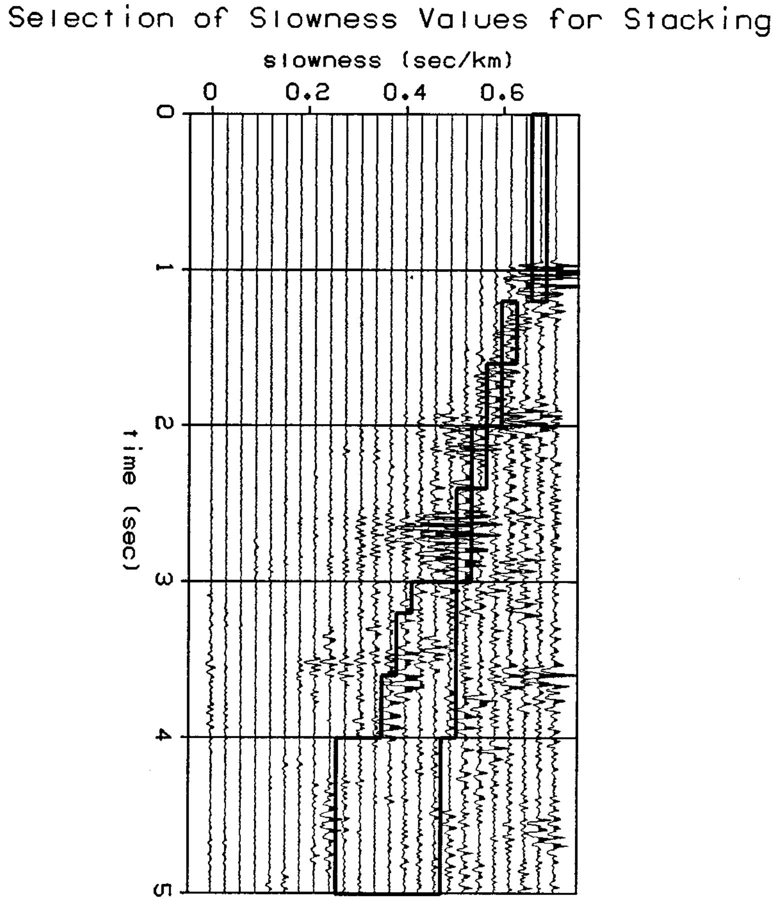

FIG. 5.13. Amoco Grand Banks: velocity stacking function. This panel is a re-illustration of the inverse velocity stack **u** of figure 5.8. The portions of traces inside the boxed outline were summed (over slowness) to produce the stacked section of figure 5.14. In that figure, each trace is the result of inverse velocity stacking a common midpoint gather to obtain **u**, followed by summing over the subregion defined by the box outlines above.

window, for example, lies most of the energy from the first sea-floor multiple at 2.0 seconds.

To create the stack in figure 5.14, 90 adjacent common-midpoint gathers from the Grand Banks area were processed with the stochastic inversion algorithm of Table 5.1, then windowed and stacked with the implied velocity function of figure 5.13. For comparison, a standard stack made over the same gathers with the same time-velocity curve is shown in figure 5.15. The differences between the two

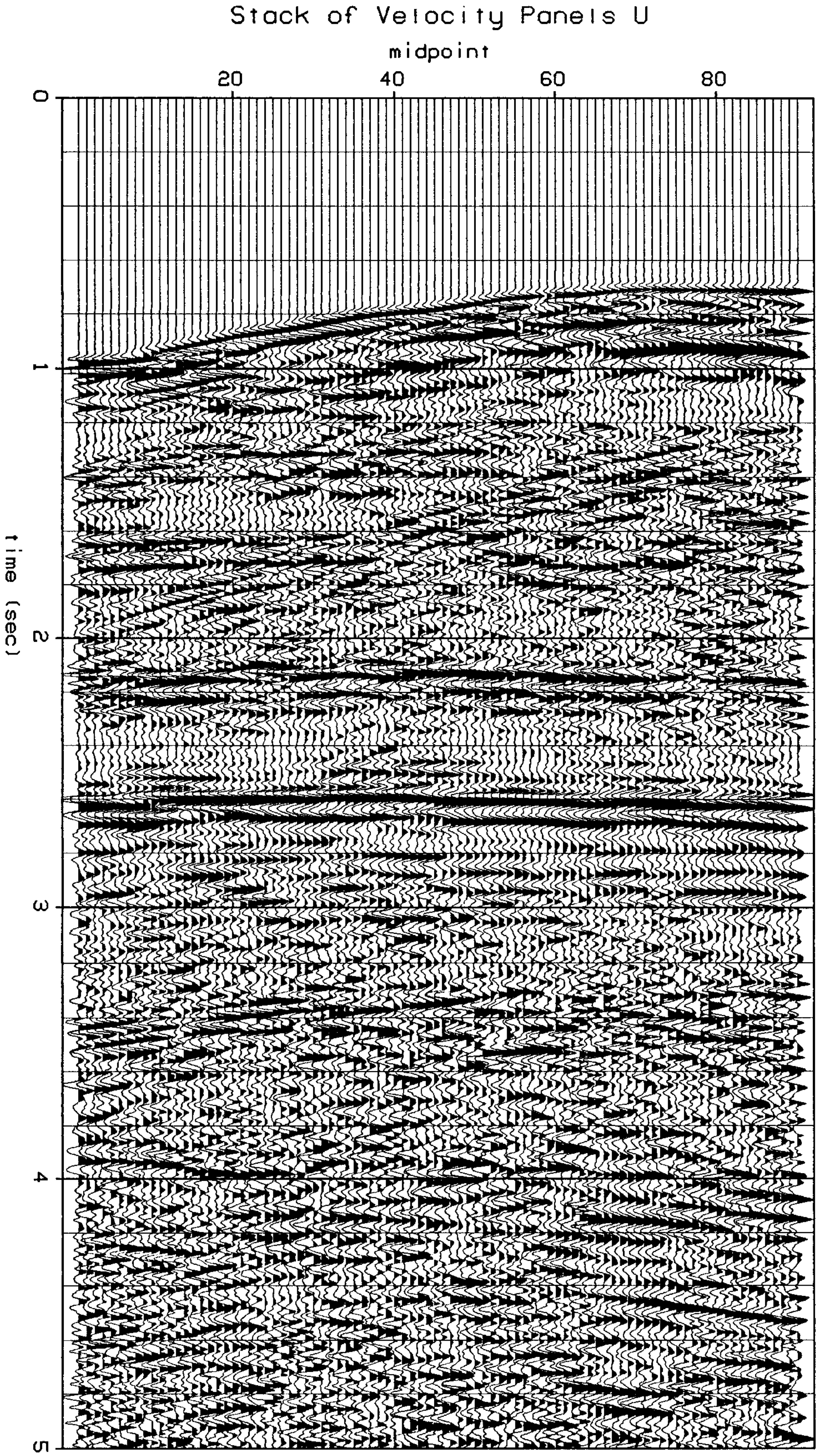

FIG. 5.14. Amoco Grand Banks: a stack of the velocity panels **u** for 90 adjacent gathers using the velocity function defined in figure 5.13.

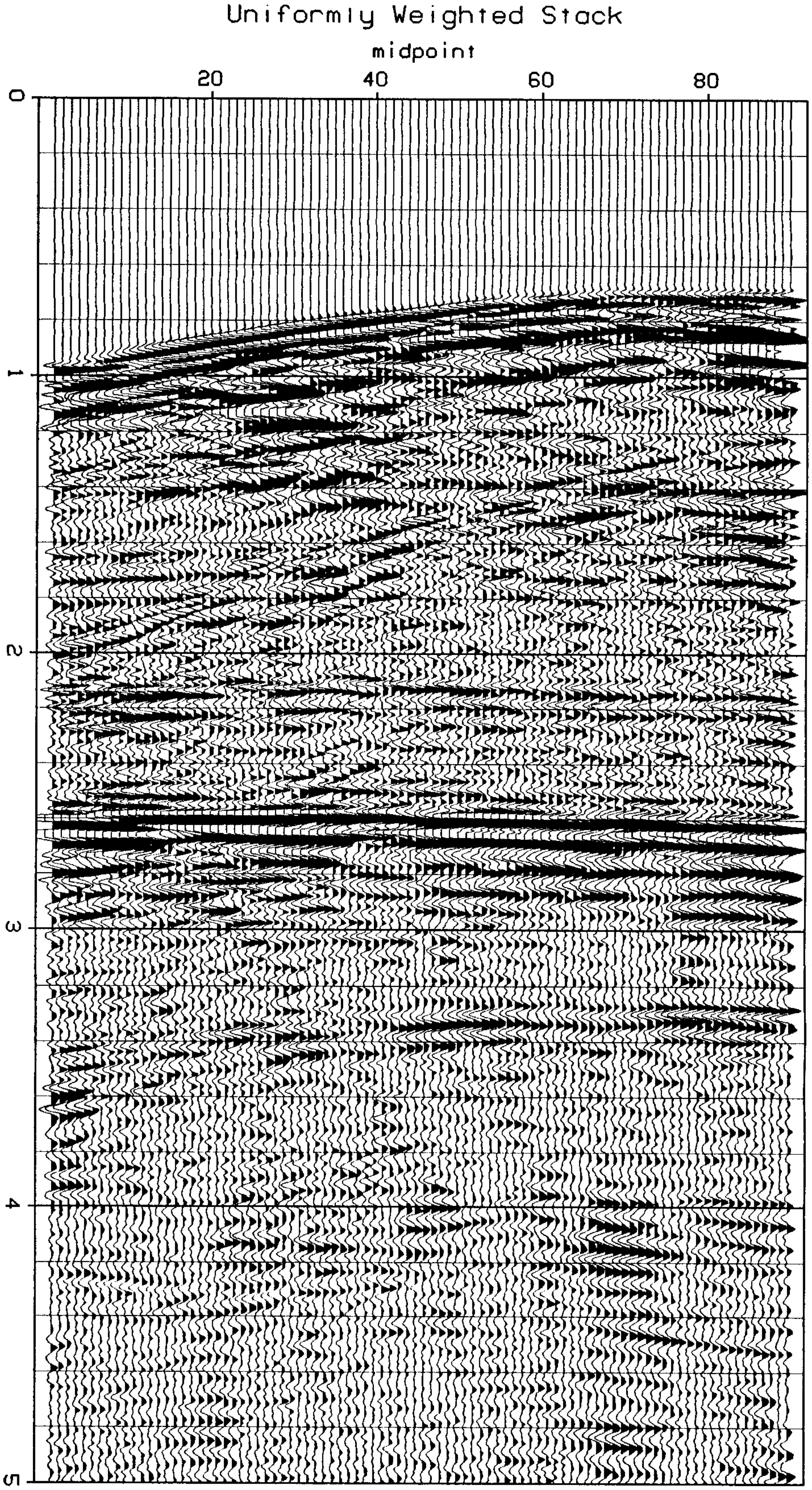

FIG. 5.15. Amoco Grand Banks: a standard, uniformly-weighted stack of 90 adjacent common-midpoint gathers.

stacks are, unfortunately, small. The stack of figure 5.14 has slightly improved the strength of primary events with respect to the sea floor multiples in the region of 1.5 to 2.0 seconds, and has better resolved the deep events from 4.0 to 4.5 seconds. An obvious improvement to the stochastic inversion process would be to sample velocities more finely at shallow times than at deep times, in order to increase the discrimination between primaries and multiples whose stacking velocities at shallow times are very nearly the same. We are free to select an arbitrary sampling geometry in velocity space; the only constraint is that the transformation **L** from velocity space to data space must be linear.

5.9 Real data inversion: Western peg-legs

Not only common-midpoint gathers, but common-shot gathers may be transformed into velocity space, provided the events on each gather are approximately hyperbolic. Figure 5.16 illustrates a common-shot gather, courtesy of Western Geophysical Company. It consists of many orders of peg-leg multiples, originating from primaries no deeper than 2.5 seconds on the gather, and formed by reverberations within a shallow water layer of about 100 meters depth. The reverberations become clear when we look at the velocity stack inverse **u** in figure 5.17. On the left panel is a contoured envelope of **u**, on the right panel **u** itself. Four dominant primaries are visible in the left panel, and are connected by a line representing the time-velocity curve of the primaries. By assuming a water depth of 100 meters, the apparent time-velocity curve each peg-leg multiple must exhibit on the velocity stack can be calculated: these are the curves A, B, C in figure 5.17; each curve emanates from its own strong primary. A water layer peg-leg multiple characteristically decreases in apparent velocity with time, as the time spent by the ray path in the water layer increases relative to the time spent in the sediments. Curves B and C match well the actual peg-leg events seen on **u**.

In order to demonstrate velocity filtering, let us divide the velocity plane of figure 5.17 up into the light- and dark-colored areas shown. The velocity stack operation **L** can then be applied to the portion of the velocity plane corresponding to the dark colored area; the resulting high-velocity modeled events are shown in figure 5.18. Figure 5.19 is the result of subtracting the high-velocity events of figure 5.18 from the original data; as a result it represents the data with one train of pegleg multiples (and its generating primary) filtered out. The most visible change in the filtered data is the removal of the destructive-interference "dead zone" seen in figure 5.16 at the 1.9 km offset, from 3 to 4 seconds.

Originally, the data of figure 5.16 were sampled at two different offset intervals: the near offset traces out to 1.8 kilometers were sampled 25 meters apart, while the far offset traces from 1.8 to 3.2 kilometers had a 50 meter sampling interval. When solving for the stochastic inverse **u** of this data set, we prevented the uneven sampling from biasing the solution **u** by effectively setting the noise variance of the missing traces to infinity. Once **u** was found, the missing traces were estimated by applying a velocity stack **L** with an even sampling interval of 25 meters: thus every other trace on figure 5.16 from 1.8 km to the end of the cable has been interpolated.

5.10 Real data inversion: Western surface waves

Besides discriminating hyperbolic events by their associated moveout, the stochastic inverse is also able to discriminate between hyperbolic and non-hyperbolic events. This is especially true of surface waves, whose characteristics on the shot record are high dip (i.e., slow velocities) and dispersion. A data set (courtesy of Western Geophysical Co.) with such a surface wave component is shown in figure 5.20. It is a shot profile recorded somewhere in Saudi Arabia. As in all previous examples, it is convenient to compare the stochastic inverse **u** with a "standard"

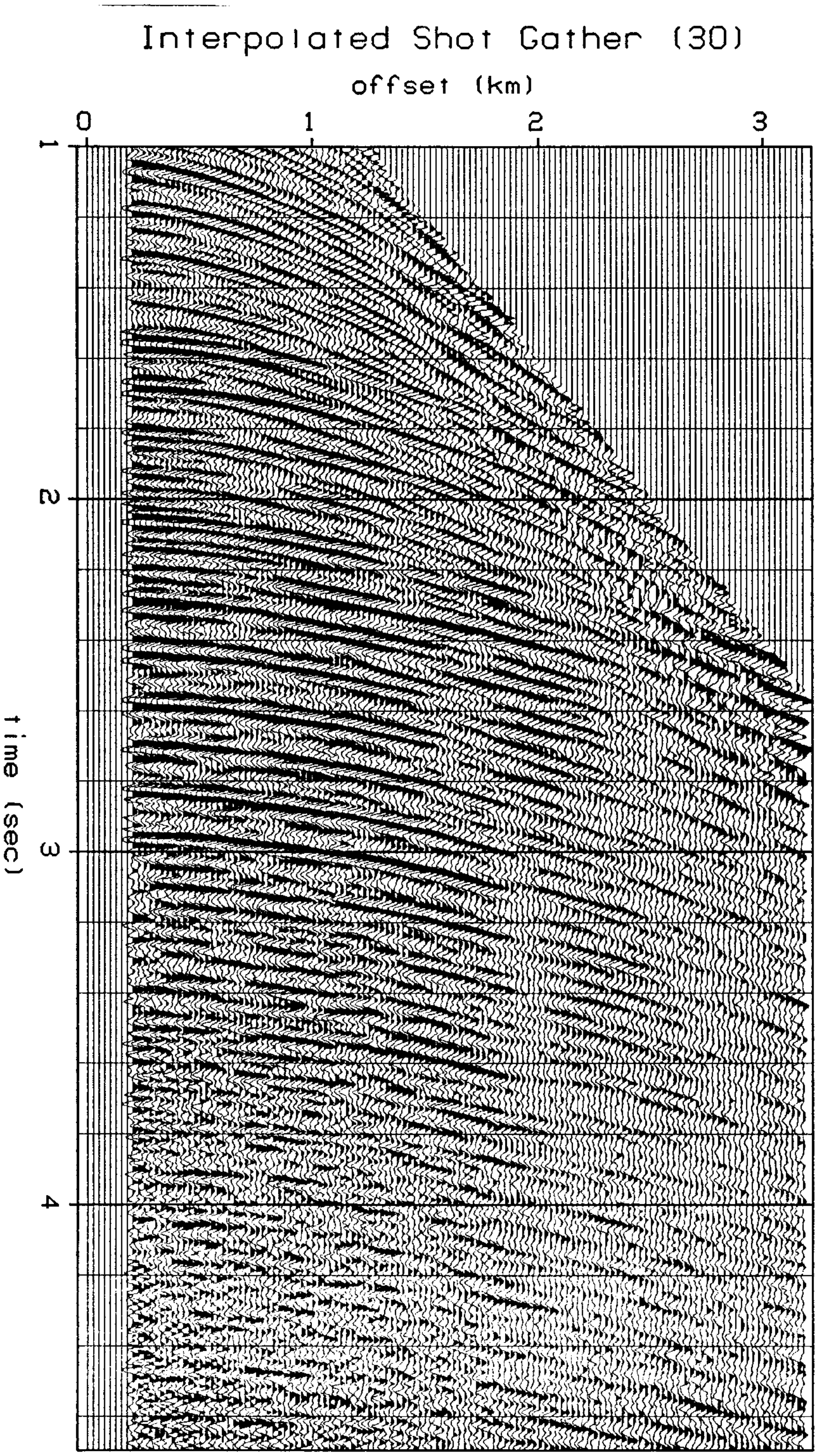

FIG. 5.16. Western peg-legs: interpolated shot gather. The original group interval on this shot profile (courtesy of Western Geophysical) was 25 meters for the near offsets out to 1.8 km, and 50 meters from then on. Far-offset traces were interpolated, to make a uniform group interval of 25 meters, by use of the velocity stack inverse **u** of figure 5.17. The time sampling interval is 8 msec.

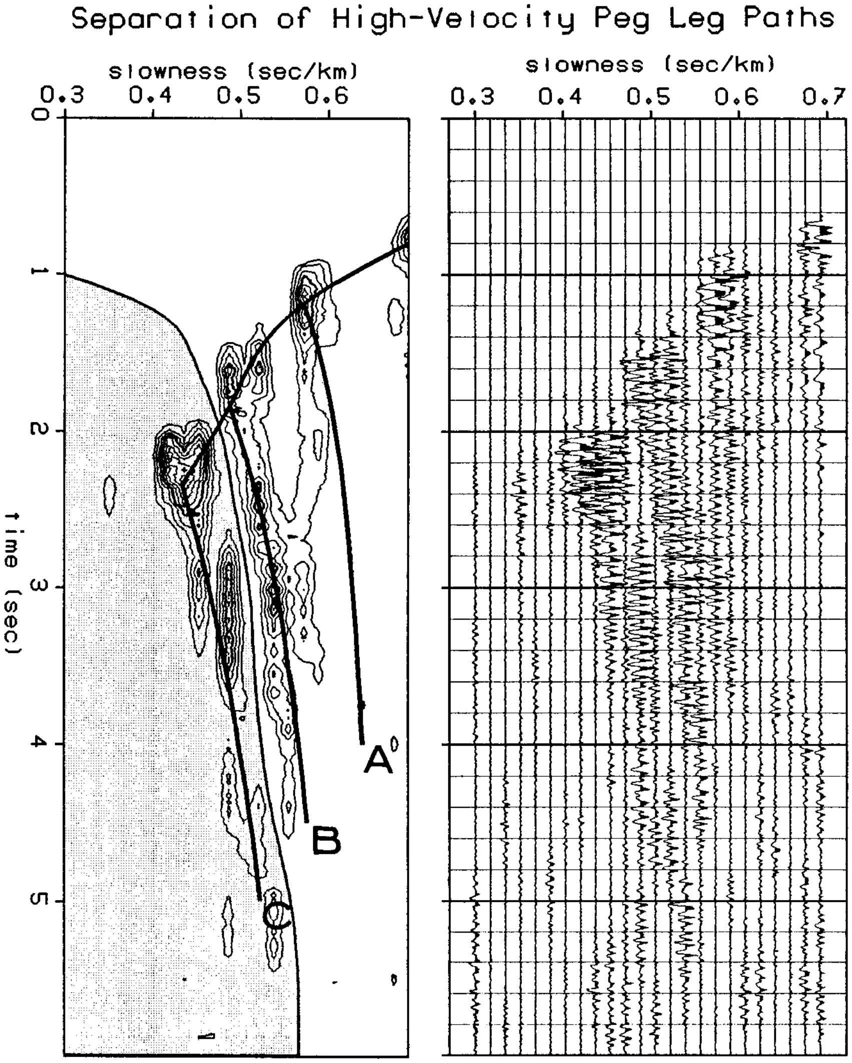

FIG. 5.17. Western peg-legs: the separation of peg-leg paths by velocity. The panel on the right is the velocity stack inverse **u**, the result of applying the algorithm of table 5.1 to the uninterpolated portion of the gather of figure 5.16. The panel on the left shows contours of $\sigma(\mathbf{u})$, the envelope of **u**; the contour interval is 10% of max(σ). The upper bold line connects primaries from 0.8 seconds to 2.4 seconds; the lower bold lines, labeled A, B, C, are theoretical peg-leg multiple paths, calculated using the assumption that water depth equals 100 meters. The shaded area has been designed to enclose only the highest-velocity peg-leg multiple path C.

velocity stack $\mathbf{L}^T\mathbf{d}$ (figure 5.21). The modeled shot gather **Lu** and the residual **d** − **Lu** are shown in figures 5.22 and 5.23, respectively. Virtually all of the surface wave energy present on the shot record remains in the residual. On the other hand,

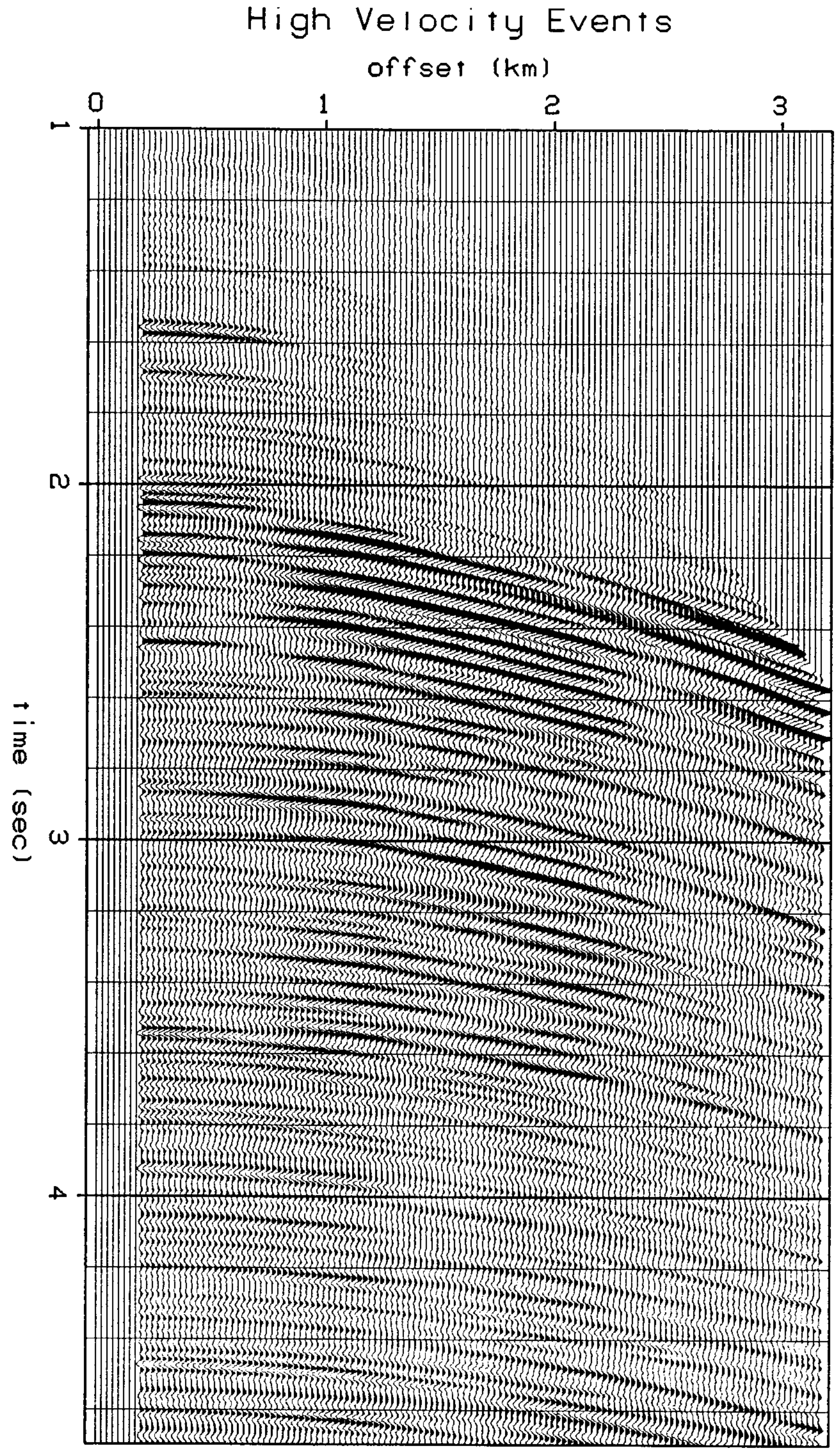

FIG. 5.18. Western peg-legs: the high velocity multiple paths modeled by velocity stacking **u** (with L) *only over the shaded* area in figure 5.17.

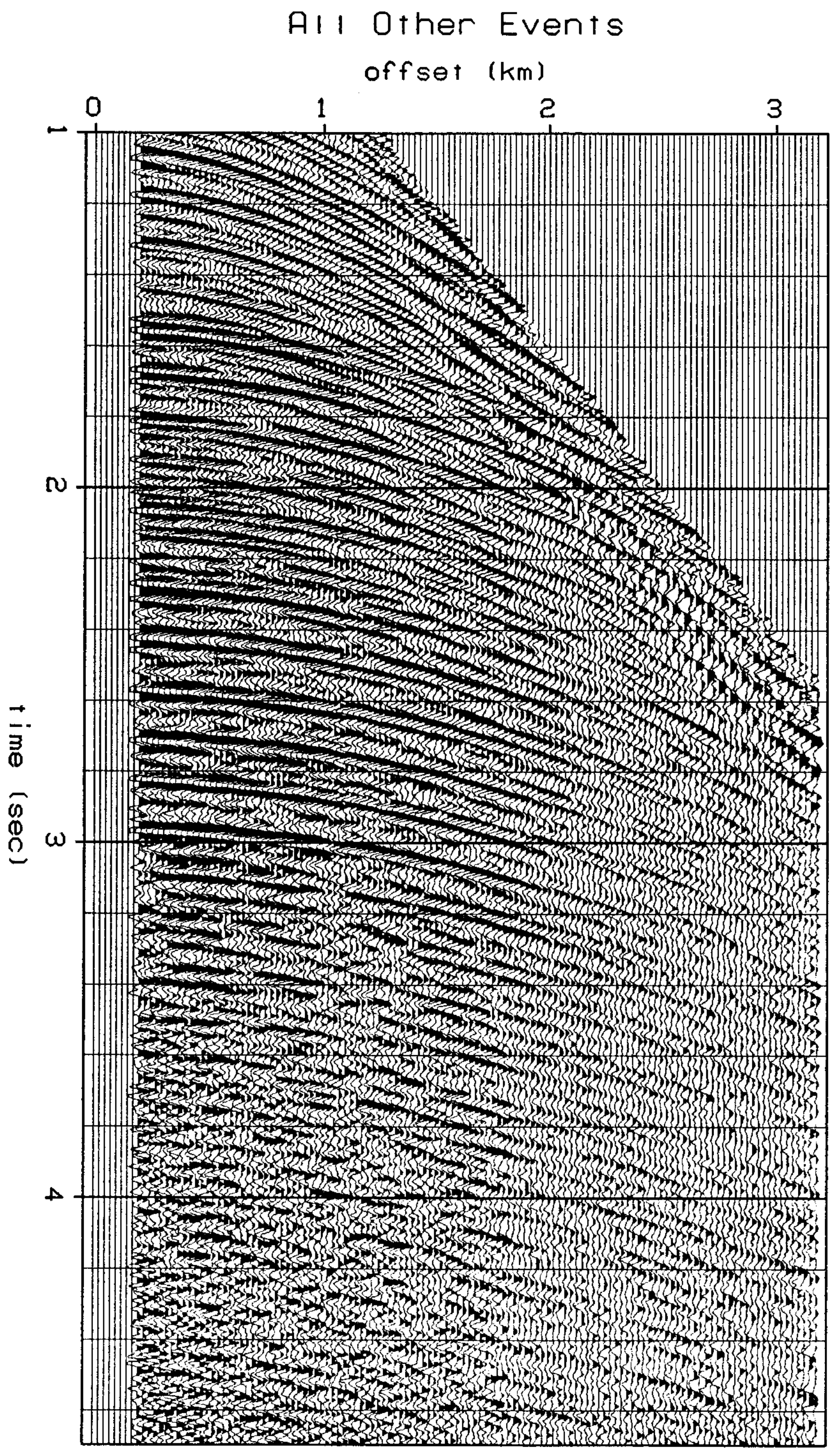

FIG. 5.19. Western peg-legs: The result of subtracting the data of figure 5.18 from the data of figure 5.16. It represents a version of the common shot profile with the high-velocity peg-legs (path C of figure 5.17) filtered out.

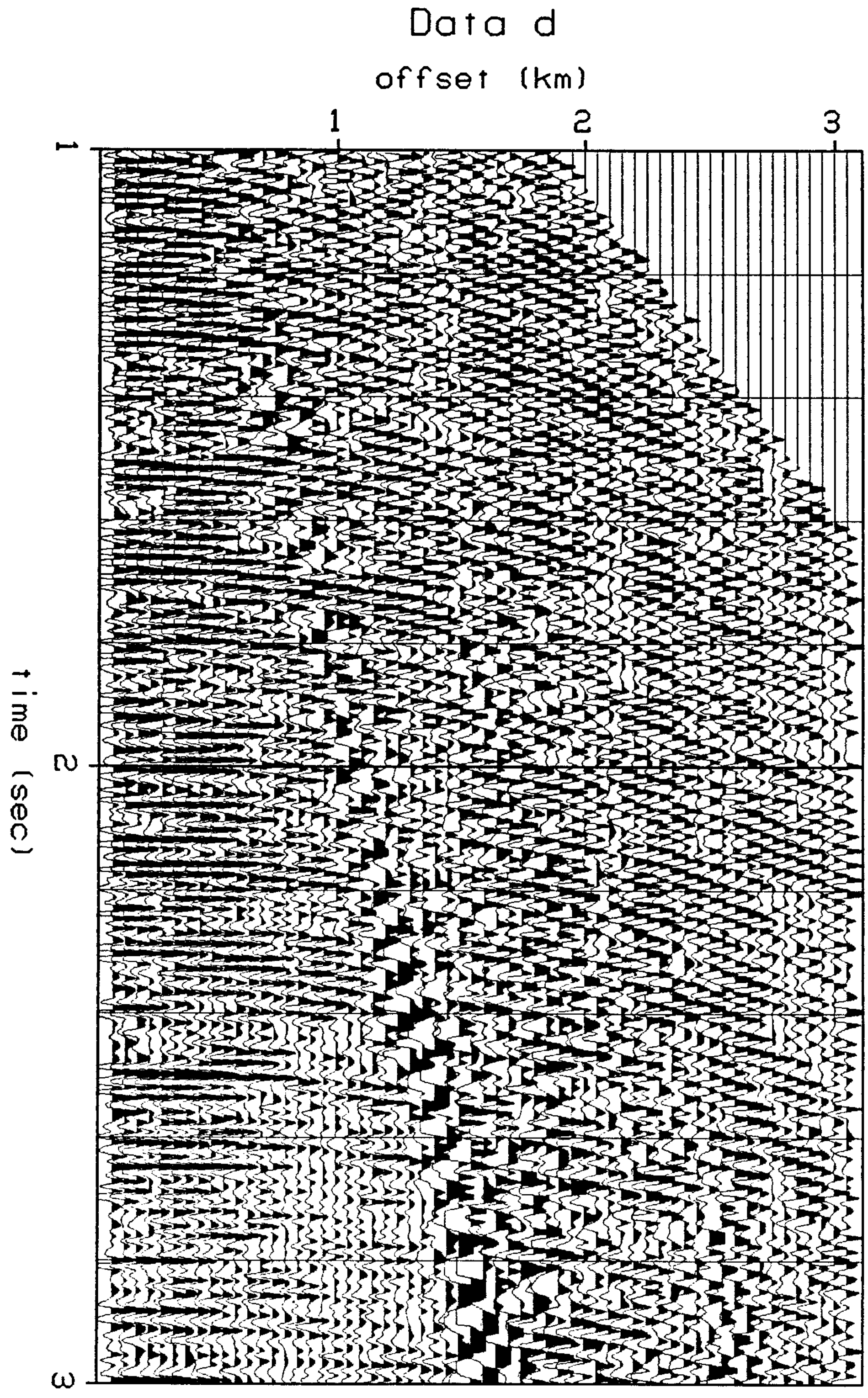

FIG. 5.20. Western surface waves: a common-shot land profile (courtesy of Western Geophysical) from Saudi Arabia. The sampling interval is 4 msec, and the group interval is 50 meters. The profile has been gained with a function proportional to time squared.

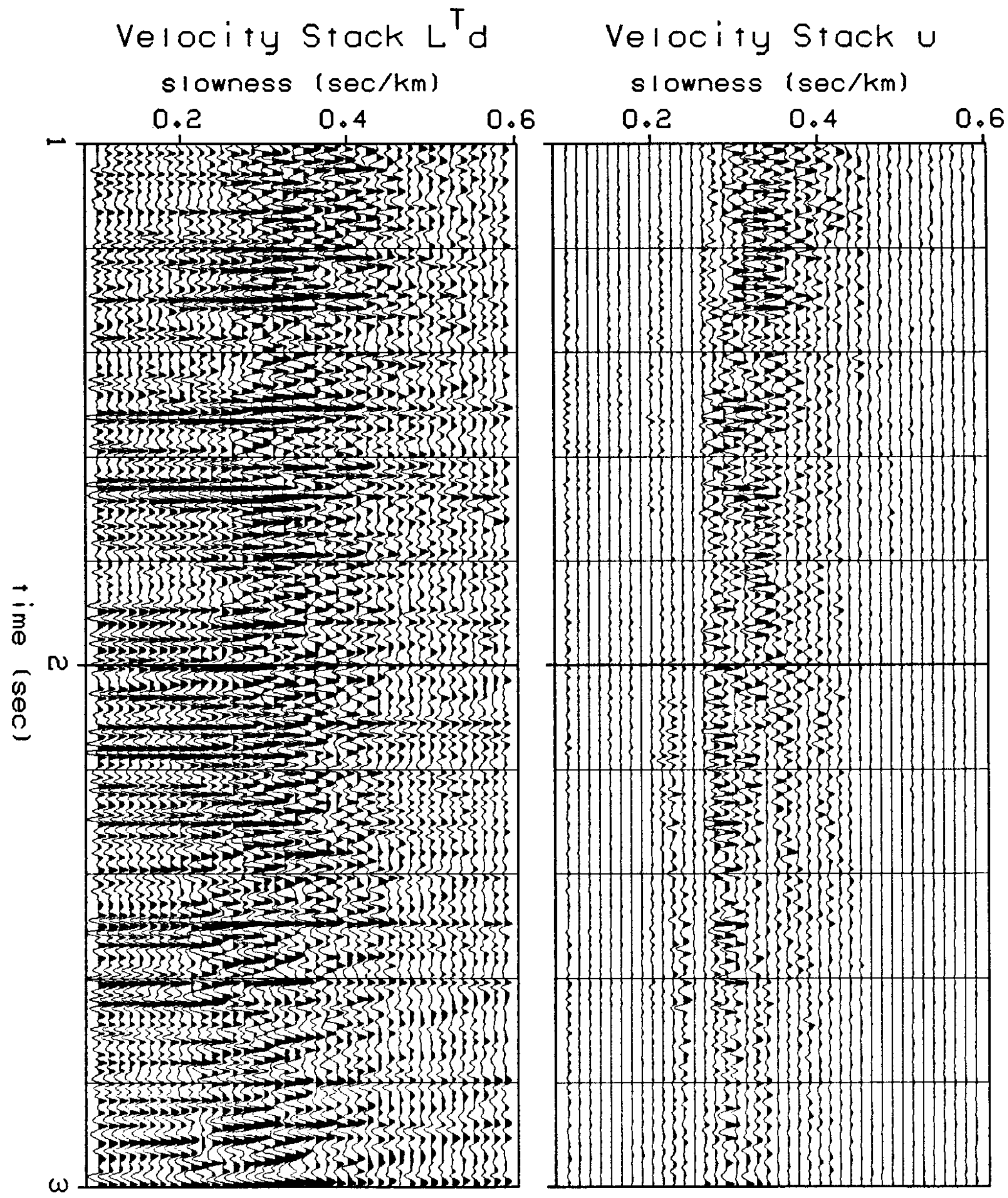

FIG. 5.21. Western surface waves: a comparison of a standard velocity stack $\mathbf{L}^T\mathbf{d}$ of the data of figure 5.20, with the inverse velocity stack **u**. The slowness interval on each panel is 0.0125 sec/km.

reflectors previously masked by the surface waves have been recovered on figure 5.22.

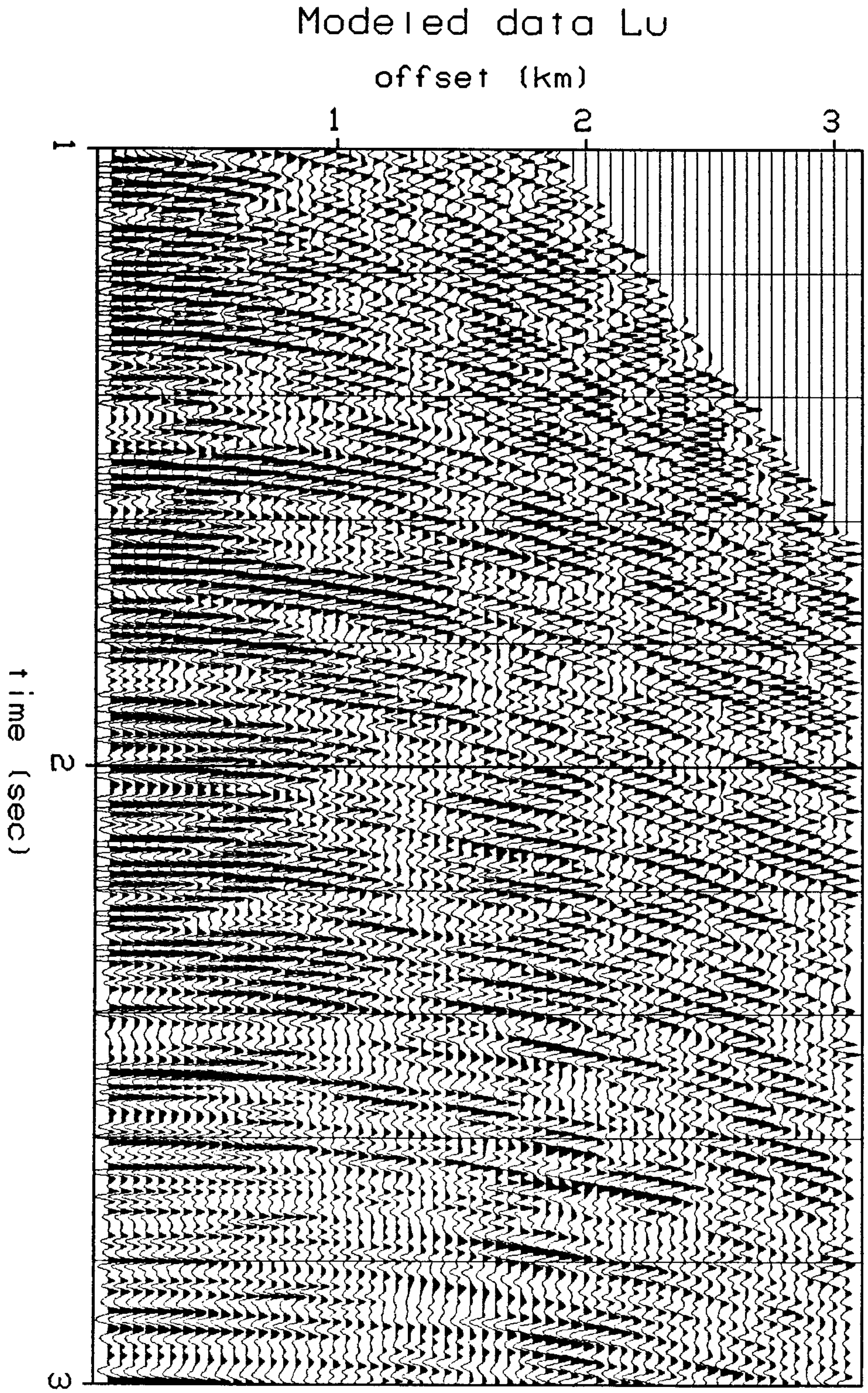

FIG. 5.22. Western surface waves: the velocity stack **Lu** of the panel **u** in figure 5.21. Compare this to the original data of figure 5.20.

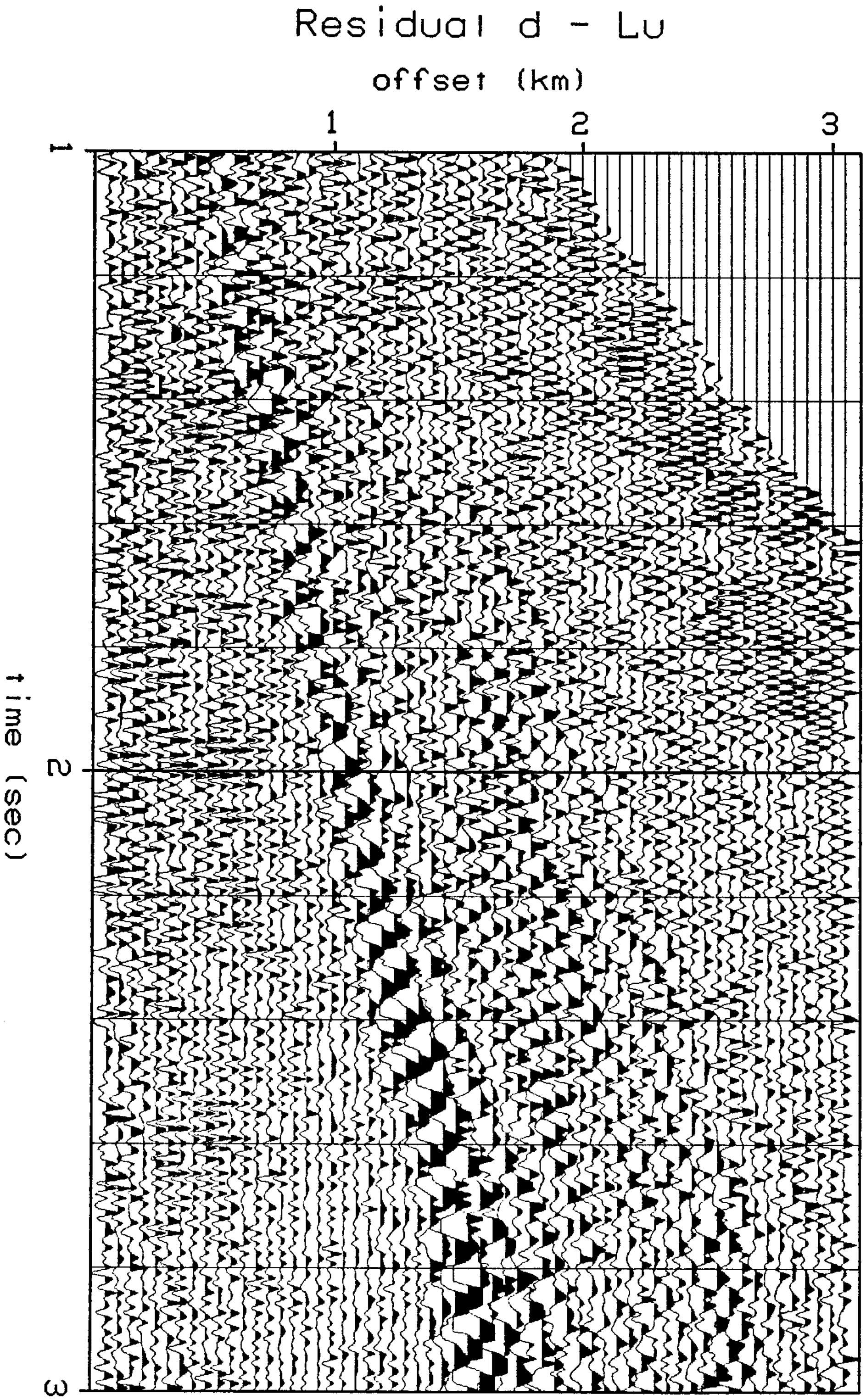

FIG. 5.23. Western surface waves: the difference between the modeled profile of figure 5.22 and the original data of figure 5.20.

5.11 Summary: conditions favorable to inverse velocity stacking

The stochastic inversion process described in this chapter is basically a dip-decomposition scheme. It assumes that the given data set is a linear superposition of events with hyperbolic moveout. The validity of this assumption can be easily destroyed by applying an automatic gain correction to the data, whose function is to make events visible by gaining the traces in a nonuniform manner. The examples considered in the last three sections have all avoided this problem: instead of an automatic gain, identical spherical divergence corrections were applied with a gain proportional to t^2.

Stochastic inversion is a global process; it does not model well changes in amplitude or phase along the reflector from one offset to another. Up to a point it can accommodate reflectivity variations with offset, for it seems to do a good job at reproducing smooth variations, but this is done at the expense of the image's sharpness in velocity space. A replacement for the velocity stacking operator **L** which is less global in nature will certainly be more successful when there are large deviations from the hyperbolic-moveout model, as there were for the example of figure 5.7.

In any case, the stochastic inverse **u** has proven to be equal in performance to nonlinear semblance velocity analysis in resolving event velocities. The common assumption that both methods are based on is the presence of events whose traveltimes fit a hyperbolic moveout curve. Moreover, the stochastic inverse has the added advantage of being quasi-invertible: the transformed velocity panel **Lu** in many cases fits the original gather very well.

5.A Appendix: the equivalence of Burg's spectral entropy to statistical entropy

In this section we shall show the equivalence of functional S_P (equation 5.45) and entropy S_E (equation 5.46) under the conditions that $p(\mathbf{u})$ is independent, and has a known Gaussian distribution with zero mean. The density $p(\mathbf{u})$ is actually conditional upon σ, the variances of the Gaussians, and here we may consider S_P and S_E to be functions of $\boldsymbol{\sigma}$. With these assumptions, $p(\mathbf{u})$ is

$$p(\mathbf{u}) = \prod_{i=1}^{N} A_i \exp\left[-\frac{1}{2}\frac{u_i^2}{\sigma_i^2}\right] \tag{5.A1}$$

where

$$A_i \equiv \frac{1}{\sqrt{2\pi}\sigma_i} \tag{5.A2}$$

are the normalizing terms. By direct substitution,

$$\begin{aligned} S_E &= -\int p(\mathbf{u}) \ln p(\mathbf{u})\, d\mathbf{u} \\ &= -\int \left[\prod_{i=1}^{N} A_i \exp\left[-\frac{1}{2}\frac{u_i^2}{\sigma_i^2}\right]\right]\left[\sum_{j=1}^{N} \ln A_j - \frac{1}{2}\frac{u_j^2}{\sigma_j^2}\right] du_1 \cdots du_N \\ &= -\sum_{j=1}^{N} \int \left[\ln A_j - \frac{1}{2}\frac{u_j^2}{\sigma_j^2}\right] \prod_{i=1}^{N} A_i \exp\left[-\frac{1}{2}\frac{u_i^2}{\sigma_i^2}\right] du_1 \cdots du_N \\ &= -\sum_{j=1}^{N} \int_{-\infty}^{\infty} \left[\ln A_j - \frac{1}{2}\frac{u_j^2}{\sigma_j^2}\right] A_j \exp\left[-\frac{1}{2}\frac{u_j^2}{\sigma_j^2}\right] du_j \\ &\qquad * \prod_{\substack{i=1 \\ i \neq j}}^{N} \int_{-\infty}^{\infty} A_i \exp\left[-\frac{1}{2}\frac{u_i^2}{\sigma_i^2}\right] du_i \end{aligned} \tag{5.A3}$$

Each factor in the product on the right equals unity, because the independent densities are normalized by A_i. Each integral within the sum over j consists of two terms. The integral of the first term is seen by inspection to be $\ln(A_j)$; the integral of the second term can be evaluated by parts:

$$\begin{aligned} S_E &= -\sum_{j=1}^{N}\left[\ln A_j + A_j\int_{-\infty}^{\infty} u_j\left(-\frac{1}{2}\,\frac{u_j^2}{\sigma_j^2}\right)\exp\left(-\frac{1}{2}\,\frac{u_j^2}{\sigma_j^2}\right)du_j\,\right] \\ &= -\sum_{j=1}^{N}\left[\ln A_j + A_j\,\frac{u_j}{2}\,\exp\left(-\frac{1}{2}\,\frac{u_j^2}{\sigma_j^2}\right)\Bigg|_{-\infty}^{\infty}\right. \\ &\qquad\qquad \left. - A_j\int_{-\infty}^{\infty}\frac{1}{2}\,\exp\left(-\frac{1}{2}\,\frac{u_j^2}{\sigma_j^2}\right)du_j\,\right] \end{aligned} \tag{5.A4}$$

The integrated part vanishes, while the remaining integral is $1/2A_j$.

$$\begin{aligned} S_E &= -\sum_{j=1}^{N}\left[\ln A_j - \frac{1}{2}\right] = \frac{N}{2} - \sum_{j=1}^{N}\ln\frac{1}{\sqrt{2\pi}\sigma_j} \\ &= \frac{1}{2}\sum_{j=1}^{N}\ln\sigma_j^2 + \frac{N}{2}(\ln 2\pi + 1) \end{aligned} \tag{5.A5}$$

Apart from the constant term, S_E is therefore seen to be equivalent to S_P:

$$S_E(\boldsymbol{\sigma}) = S_P(\boldsymbol{\sigma}) + \frac{N}{2}(\ln 2\pi + 1) \tag{5.A6}$$