Chapter II

The Influence of Missing Data on Inverse Filtering

2.1 Missing data

Data recorded in a reflection seismic survey can be characterized by a multi-dimensional sampling process: sampling in space and sampling in time. The unique quality of a reflection seismic data set, in contrast to other geophysical records, can be ascribed to high volume and to the high sampling densities with which it is recorded. The ideal goal is to continuously record the wavefield arriving at the surface of the earth; but because the coverage of the ground surface by geophones is necessarily limited both in areal extent and density, seismic data will always be spatially incomplete to some degree. This incompleteness can be divided into three categories: lack of areal extent (cable length truncation), sparse areal sampling, and irregular gaps in the recording array (dead traces).

In contrast to sampling in space, sampling in time is much more controllable. In practice, there are no time sampling irregularities on seismic data, and the time dimension is usually sampled densely enough, with respect to the frequency content of the arriving signal, that for almost all practical purposes the data may be considered continuously sampled.

The channel capacity of state-of-the-art seismic recorders is continually increasing, but there are two limitations that may place an upper bound on future development in this direction. The first is the overwhelming amount of data collected. A data set in a linear survey has three dimensions: the shot and geophone axes, and the time dimension. Beginning with the use of 3-D surveys and areal

coverage, the dimensionality of the data jumps to five: two shot coordinates, two geophone coordinates, and a time coordinate. If the spatial sampling interval (the geophone and shot group spacing on the ground) is reduced by one-half, the amount of data in a linear survey will increase by a factor of four, and that of an areal survey by a factor of sixteen. This rapid expansion of the volume of data with reduction in sampling interval means that reflection seismic data may always be irregularly or incompletely sampled in the spatial dimensions, the limitations arising due to either the capacity of the recording equipment or the ability of subsequent processing to digest it.

The second problem of seismic data acquisition, which counteracts the spatial sampling problem, is a law of diminishing returns. Increasing the spatial sampling rate might increase the signal-to-noise ratio of poor data, but will not necessarily increase the content of information in the data if the signal-to-noise ratio is already high. One of the most striking characteristics of a good data set is the positive coherence between traces close to each other. The success of the common-midpoint stacking method is based on this fact: that coherent events exist in the data and can be tracked from trace to trace. A subsampled version of the collected data might be adequate to process; the remaining data being rejected as supplying redundant information. Human interpreters, as a case in point, are good at recognizing essential patterns in the data and passing off the remainder of the data as redundant. In contrast, non-human processors rely on a certain amount of redundancy of the data for good results; migration algorithms, for example, require the data to possess a fine spatial sampling interval.

2.2 Examples of missing data restoration

The motivation for restoring missing data arises from the need to present to any subsequent processing steps a complete, evenly sampled data set. An

"artifact" is defined as any undesired effect that arises because the data set is incomplete in some way. In the majority of cases this "incompleteness" translates directly into the implicit assumption that missing data values are set to zero. In this section the effects of missing data on some common seismic processing techniques are reviewed.

One technique so affected is wave equation migration. In migration, assuming that the wavefield is zero outside the computation grid is equivalent to specifying a zero-value boundary condition for the edge of the grid. This assumption introduces artificial edge diffractions, which arise from internal reflections at the grid boundary. A simple step that virtually eliminates this problem is to apply absorbing boundaries: that is, predict the values for the unknown data points immediately outside the edge of the grid (Clayton and Engquist, 1980), (Toldi and Hale, 1982). This method works well for the migration operator, because it is the integral over depth of a local operator, and the local operator needs input values about only a small neighborhood of the point it is interested in.

Another technique affected by missing data is spectral estimation. The spectral estimation of a short length of data is a classic problem; one of the most successful methods is that of maximum entropy spectral analysis (Burg, 1975). Within the framework of this discussion, the maximum entropy method first replaces the implied zeroes of missing data with a most likely set of values, then computes the spectrum from this extended data set. These "most likely" values are quantitatively evaluated by use of a minimum-information (or maximum-entropy) measure that is optimized by extrapolating the missing data. That is, the extrapolated values are designed precisely to supply as little additional information as possible.

Just as migration is affected by assuming zero values for missing data, so is resolution in dip filtering lost when the data set is spatially truncated. In the two-dimensional spectral domain, a sinc-convolved spread of energy will result from the

truncation, and cause events of nearly the same dip to overlap. A conceptually different situation causes aliasing problems. Insufficient spatial sampling of a steeply dipping event translates to wrap-around of the spectrum in the Fourier domain. In principle this spectrum can be unwrapped only if the true high-dip components do not overlap the true low-dip components; so interpolating the missing traces involves a decision on how to partition events in the Fourier domain into their low-dip and high-dip components. The all-important process of stacking is a special instance of dip filtering. The stacked trace should be influenced only by the zero-dip events on a moved-out gather, but spatial truncation and aliasing allow other undesired events to contribute to the stacked trace. Selectively weighting the traces before stacking is a means of suppressing undesired events on the stack (Larner, 1979), and is analogous to the taper-and-transform method of local spectral estimation.

2.3 Stacking: velocity stacks and slant stacks

To narrow the scope of our investigation, we will concentrate on the effects of missing data on two commonly used linear transformations: slant stacking and velocity stacking. It should be noted that no effort up to now has been made to define commonly used technical terms peculiar to reflection seismology. In this and the following sections, these terms will first appear in *italicized phrases* and may be accompanied with a brief definition, but most will be defined only by context.

Let us now define the stacking operations, and look at two typical data sets to which these operations could be applied. Figure 2.1(a) shows a *common-midpoint gather (CMG)* from a marine survey shot in the Grand Banks area of offshore Labrador. The horizontal axis, the *offset* from the shot-geophone midpoint, is in units of distance; the vertical axis is in units of time. The vertical axis has 750 sample points, and the horizontal axis is much more sparsely sampled, with 48 points.

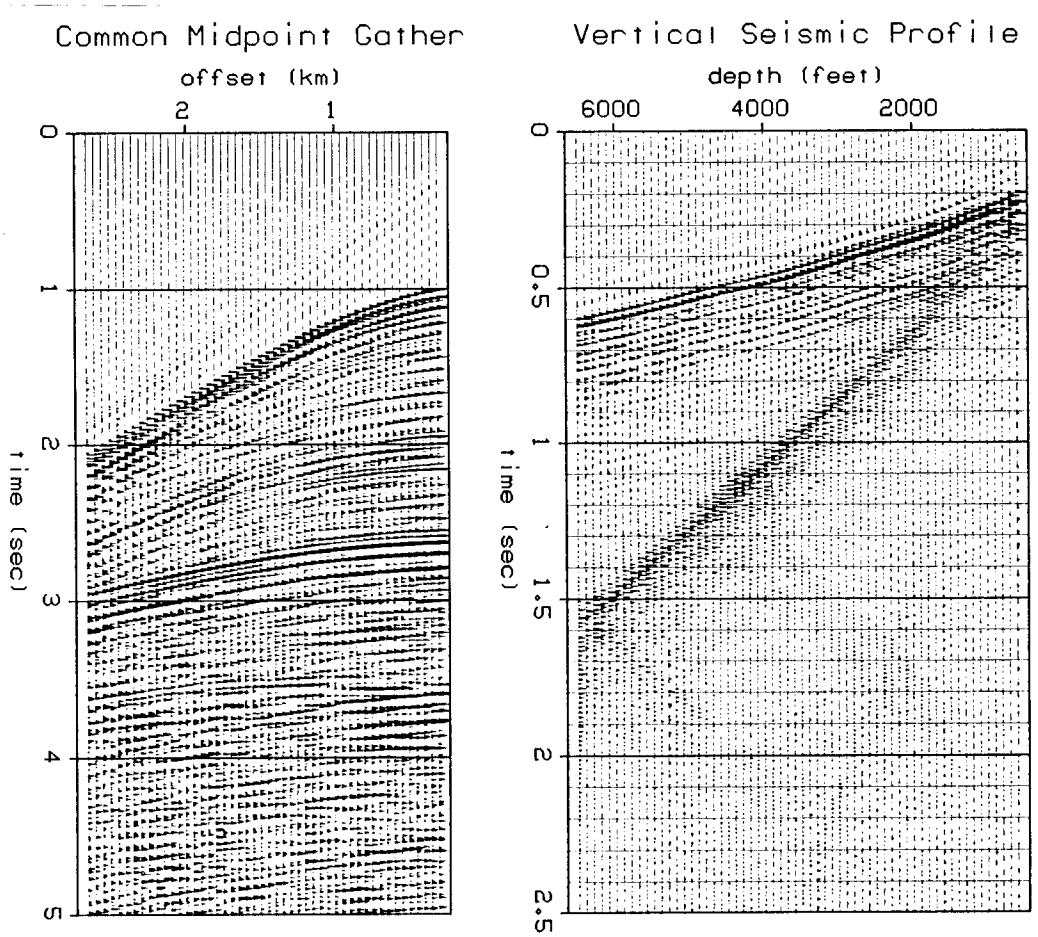

FIG. 2.1. Two data sets.

(a) A common-midpoint gather from Grand Banks, offshore Labrador, with an offset interval of 50 m, and sample rate of 8 ms. This is an example of input presented to the velocity stack transformation. This data is courtesy of Amoco, Inc.

(b) A vertical seismic profile, courtesy of Arco. The well-phone spacing is 100 feet, the sample rate 4 ms. In contrast to the Amoco gather, arriving events have linear moveout. The steeply dipping events are low-velocity tube waves propagating in the borehole fluid. For a discussion of vertical seismic profiles, see chapter 6.

Reflections from the subsurface typically possess *hyperbolic moveout*, that is, reflections on a CMG tend to correlate along hyperbolic trajectories described by the *normal moveout (NMO) equation*:

$$t(h) = \sqrt{\tau^2 + h^2/v^2} \qquad (2.1)$$

where h and t are the offset and time indices of the midpoint gather, τ is the zero-offset *intercept time*, and v is the *stacking velocity*.

A *velocity stack* is made by summing over such hyperbolic trajectories, and because each trajectory is uniquely determined by the stacking velocity v and the zero-offset intercept time $\tau = t(0)$, two degrees of freedom are available for each summation path; while the input space to a velocity stack transformation is indexed by t and h, the output space is indexed by τ and v. Let $d(h,t)$ refer to the common-midpoint gather, and $u(v,\tau)$ refer to the resulting velocity stack. Velocity stacking is then defined to be

$$u(v,\tau) = \sum_h d(h, \sqrt{\tau^2 + h^2/v^2}) \qquad (2.2)$$

Note that in this form, the data set has been assumed to be continuously sampled in time, but discretely sampled in offset h. An alternate definition of velocity stacking, and one that will be used in favor of equation 2.2, is made by setting the independent variable of the output space (*velocity space*) to *slowness* $p \equiv 1/v$. The new definition for velocity stacking is then

$$u(p,\tau) = \sum_h d(h, \sqrt{\tau^2 + p^2h^2}) \qquad (2.3)$$

The output $u(p,\tau)$ is also known as a *constant velocity panel*, because each output trace (i.e., **u** for fixed p) results from applying a normal moveout stretch appropriate to the corresponding velocity, to the midpoint gather $d(h,t)$. This process is then followed by a summation over offset (*stacking*). The transformation defined by equation (2.3) is linear, though the input and output spaces can have rather large dimensions: the discrete input space for the gather in figure 2.1(a) is 48 x 750 = 36,000-dimensional.

In what follows, all four parameters p, τ, h, and t are to be restricted to non-negative values. This is a natural restriction for slowness p, which represents a physical quantity which is always positive. A trace at negative offset $-h$, that is, one recorded in which shot and geophone locations have been interchanged, should be by reciprocity identical to the trace at offset $+h$. Therefore, negative offset

values theoretically are redundant and need not be included in the stack. Times $t = 0$ and $\tau = 0$ represent the instant that the shot goes off; of course, no events of interest occur before zero time, and the time axis can be restricted to be positive.

Velocity stacks are appropriate transformations to apply to data such as common-midpoint gathers. If reflections (the desired information) on the gather actually follow hyperbolic paths, summing over these paths will yield a large value in velocity space at that reflector's stacking velocity, while destructive interference will yield low-background values at velocities that do not correspond to detectable events on the gather. In other words, the ideal output will be focused: reflections should be transformed into points.

Slant stacking is defined in a manner completely analogous to that of velocity stacking. Sampling of the input data must be dense enough in time to warrant the assumption that the time axis is continuous. The *slant stack* $u(p,\tau)$ of the data $d(h,t)$ is defined to be the uniformly weighted sum

$$u(p,\tau) = \sum_h d(h, t = \tau + ph) \tag{2.4}$$

where h,t are offset and time indices to the data, τ again is a zero offset intercept, and p is now the slope along which points are summed. The relationship between slant stacking and velocity stacking (equation 2.3) is easy to see. Slant stacking is defined by replacing hyperbolic normal moveout (NMO) with linear moveout (LMO). Furthermore, velocity stacking may be reduced to slant stacking by applying a stretch transformation to the data **d** and output **u**:

$$\begin{aligned} t_v^2 &= t_s & \tau_v^2 &= \tau_s \\ h_v^2 &= h_s & p_v^2 &= p_s \end{aligned} \tag{2.5}$$

where the v subscript refers to the parameters of the velocity stack in equation (2.3), and the s subscript refers to the slant stack parameters in equation (2.4). In the earliest days of reflection seismology this transformation, known as a

t^2--x^2 *stretch*, was used to convert reflectors with hyperbolic moveout into events with linear moveout (Green, 1938). Transformation (2.5) is used in chapter 4 to derive a *generalized inverse* to the velocity stack of equation (2.3), by relating it to the slant stacking generalized inverse derived in chapter 3.

In contrast to the limited parameters of velocity stacking, the parameters p, τ, h, t in a slant stack will be allowed to range over all real values, both positive and negative.

Figure 2.1(b) illustrates a data set that is appropriate for slant stacking. This data, a vertical seismic profile (VSP), will be used in chapter 6 as a test case for slant stack inversion. The horizontal axis, again in units of distance and having a coarse sampling interval, represents the depth of the recording geophone in a well. When the velocity of the surrounding medium is constant and the reflectors have no dip, a shot set off at the earth's surface near the well produces events whose arrival time is almost a linear function of geophone depth in the well. These events exhibit *linear moveout*: arrival time is directly proportional to distance. The VSP in figure 2.1(b) shows a slight flattening of the downgoing events with depth; this flattening indicates that the average velocity of the medium increases with depth.

Slant stacking is an appropriate transform to apply to a VSP when wavefronts arriving at the well bore (both downgoing and upcoming), are to be mapped into points in the slant stack domain. Just as velocity stacking of a CMP gather should result in a sparse output, so should slant stacking transform a VSP into a sparse, focused record in the new domain. One reason that this does not happen in practice is that the inadequate sampling of the data introduces aliasing and truncation artifacts into the transform.

2.4 Characterizing missing data by projections

In this section we will show that the discretization of continuous data, and all other effects of missing data, may be represented by projection operators $\mathbf{P}_d$ and $\mathbf{P}_u$. These operators will be used extensively in the development of the theory in this and the following chapters.

The data set **d** of equations (2.3) and (2.4) is continuous in time and discrete in offset. Let us assume that the data result from sampling a wavefield $\hat{\mathbf{d}}$ that is continuous both in time and offset; this sampling may be represented by a *projection* operator $\mathbf{P}_d$, the subscript d indicating that the sampling takes place in data space. $\mathbf{P}_d$, then, is an operation that samples the continuous wavefield $\hat{\mathbf{d}}$. The sample points in offset are given by, say, h_j, $j = 1, \cdots, N_h$. In operator notation,

$$\mathbf{d} = \mathbf{P}_d \hat{\mathbf{d}} \tag{2.6}$$

or,

$$d(h,t) = \left\{ \begin{array}{ll} 1 & h = h_j \\ 0 & h \neq h_j \end{array} \right\} \cdot \hat{d}(h,t) \tag{2.7}$$

The h_j may or may not be evenly spaced points in offset.

Note that $\mathbf{P}_d$ differs from what is normally defined to be a sampling function, the so-called Shah function (Bracewell, 1965, p. 77):

$$\mathrm{III}(h) = \sum_j \delta(h - h_j)$$

where δ is the Dirac delta. To account for this difference, and to maintain consistency in what follows, we claim $\hat{\mathbf{d}}$ to be *continuous* in h when $\hat{\mathbf{d}}$ is actually arbitrarily finely sampled in h, by a uniform sample interval Δh. That is, assume that $\hat{d}(h,t)$ is known at points $h = i\Delta h$, i being an integer from $-\infty$ to ∞. Therefore $\{ h_j \}$ is taken to be a subset of the points $\{ i\Delta h \}$. Integrals are approximated as sums: that is,

$$\int dh \approx \sum \Delta h$$

The time integral may be approximated in the same way by a discrete sum. In what follows, however, the distinction between integral and discrete sum over time will not be made, and $\hat{\mathbf{d}}$ will be considered continuous in time.

The projection operator $\mathbf{P}_d$ thus represents a subsampling process: it sets to zero all values of $\hat{\mathbf{d}}$ where h is not equal to h_j. $\mathbf{P}_d$ has all the properties of a formal projection: $\mathbf{P}_d = \mathbf{P}_d\mathbf{P}_d$; its formal inverse $\mathbf{P}_d^{-1}$ does not exist; and if depicted as a large matrix, $\mathbf{P}_d$ consists of 1's and 0's positioned along its diagonal. It is also self-adjoint: $\mathbf{P}_d^T = \mathbf{P}_d$.

Sampling in velocity space can similarly be represented by a projection operator. Subsampling $\hat{u}(p,\tau)$ to get $u(p,\tau)$ is performed by applying the corresponding projection $\mathbf{P}_u$:

$$\mathbf{u} = \mathbf{P}_u\hat{\mathbf{u}} \tag{2.8}$$

where

$$P_u(p,\tau) = \left\{ \begin{array}{ll} 1 & p = p_k \\ 0 & p \neq p_k \end{array} \right\} \qquad k = 1, \cdots, N_p \tag{2.9}$$

The output space of $\mathbf{u}$ shall be referred to as the *model domain* or the *model space*.

We will note two special cases of a projection. An *aliased* data set is an evenly subsampled version of the original data set:

$$P_d(h,t) = \left\{ \begin{array}{ll} 1 & h = jm\Delta h \\ 0 & h \neq jm\Delta h \end{array} \right\} \quad \text{for } j = 1, \cdots, N_h; \text{ and } m > 1 \tag{2.10}$$

A projection that is set equal to zero outside a specified range of h defines a *truncated* data set:

$$P_d(h,t) = \left\{ \begin{array}{ll} 1 & h_1 \leq h \leq h_2 \\ 0 & \text{otherwise} \end{array} \right\} \tag{2.11}$$

With the use of the projections $\mathbf{P}_d$ and $\mathbf{P}_u$, the discrete transformations of equations (2.3) and (2.4) may be replaced by their continuous counterparts. Veloc-

ity stacking and its discrete counterpart are now defined as

$$u(p,\tau) = \int_{-\infty}^{\infty} dh\ d(h, \sqrt{\tau^2 + p^2 h^2})$$

$$\approx \sum_{j=-\infty}^{\infty} \Delta h\ d(j\Delta h, \sqrt{\tau^2 + k\Delta p^2 j\Delta h^2}) \qquad (2.12)$$

Similarly, the continuous and discrete versions of slant stacking are defined to be

$$u(p,\tau) = \int_{-\infty}^{\infty} dh\ d(h, \tau + ph)$$

$$\approx \sum_{j=-\infty}^{\infty} \Delta h\ d(j\Delta h, \tau + k\Delta p\, j\Delta h) \qquad (2.13)$$

Note that the difference between these definitions and those of (2.3) and (2.4) lies in the extra factor Δh, the infinitesimal (but nonzero) offset sample interval that has now been included in the sums.

The missing sections of the data and model domains are taken into account by, first, applying $\mathbf{P}_d$ to the data; second, stacking with either equation (2.12) or (2.13); and third, applying $\mathbf{P}_u$ to the output. For example, the finite velocity stack can be represented as

$$u(p_k,\tau) = P_u(p,\tau) \int dh\ P_d(h,t)\ d(h, \sqrt{\tau^2 + p^2 h^2})$$

$$= \sum_j \Delta h\ d(h_j, \sqrt{\tau^2 + p_k^2 h_j^2}) \qquad (2.14)$$

The finite slant stack is likewise

$$u(p_k,\tau) = P_u(p,\tau) \int dh\ P_d(h,t)\ d(h, \tau + ph)$$

$$= \sum_j \Delta h\ d(h_j, \tau + p_k h_j) \qquad (2.15)$$

where

$$\{ h_j \} \subseteq \{ j\,\Delta h \} \quad \text{and} \quad \{ p_k \} \subseteq \{ k\,\Delta p \}$$

That is, the infinite sums of (2.12) and (2.13) are now replaced by the finite sums

of (2.14) and (2.15).

Equations (2.14) and (2.15) shall now replace (2.3) and (2.4) as the proper definitions of discrete velocity and slant stack. The operations of discrete slant stacking and discrete velocity stacking can now be written in operator notation: projections $\mathbf{P}_d$ and $\mathbf{P}_u$ are used, and $\mathbf{L}^T$ is either of the two "continuous" linear transformations defined in (2.12) and (2.13), so that

$$\mathbf{u} = \mathbf{P}_u \mathbf{L}^T \mathbf{P}_d \hat{\mathbf{d}} \tag{2.16}$$

To summarize this section, wavefields $\hat{\mathbf{d}}$ and outputs $\hat{\mathbf{u}}$, if continuous, formally require that a discretization step $\mathbf{P}_u$ or $\mathbf{P}_d$ be applied before they can be used in further discrete processing. The use of projections $\mathbf{P}_u$ and $\mathbf{P}_d$ is essentially a notational convenience to indicate to what degree the data set is truncated and aliased.

2.5 Adjoint operators

In the operator notation introduced in the last section, $\mathbf{L}^T$ may refer to either a slant stack or a velocity stack. Let us now address the question, what are the operators $\mathbf{L}$ that are adjoint to $\mathbf{L}^T$ for each choice of stacking operator. We shall subsequently use the terms *adjoint* and *transpose* interchangeably, though adjoint normally will refer to a continuous operator and transpose will refer to its discrete counterpart. Operators $\mathbf{L}$ and $\mathbf{L}^T$ are *adjoint* if they satisfy

$$(\mathbf{u}, \mathbf{L}^T\mathbf{d})_p = (\mathbf{L}\mathbf{u}, \mathbf{d})_h \tag{2.17}$$

for all $\mathbf{u}$ in the (p,τ) domain and all $\mathbf{d}$ in the (h,t) domain that have finite norms. The definition of an adjoint implies that an inner product exists in each of the input (data) and output (model) spaces of $\mathbf{L}^T$. $(\cdot,\cdot)_p$ denotes the inner product in model space; likewise an h subscript denotes data space. For slant stacks, the inner products are defined with unit weighting:

[slant] $$(\mathbf{u}_1, \mathbf{u}_2)_p \equiv \int\int_{-\infty}^{\infty} dp\, d\tau\, u_1(p,\tau)\, u_2(p,\tau) \tag{2.18}$$

$$(\mathbf{d}_1, \mathbf{d}_2)_h \equiv \int\int_{-\infty}^{\infty} dh\, dt\, d_1(h,t)\, d_2(h,t) \tag{2.19}$$

The adjoint **L** in the case of slant stacking is found by combining relation (2.17) with the definition for the slant stack, equation (2.13), and interchanging the order of integration:

$$\begin{aligned}(\mathbf{Lu}, \mathbf{d})_h &= \int\int_{-\infty}^{\infty} dp\, d\tau\, u(p,\tau) \int_{-\infty}^{\infty} dh\, d(h, \tau + ph) \\ &= \int\int\int_{-\infty}^{\infty} dp\, dt\, dh\, u(p, t - ph)\, d(h,t) \\ &= \int\int_{-\infty}^{\infty} dh\, dt\, d(h,t) \int_{-\infty}^{\infty} dp\, u(p, t - ph)\end{aligned} \tag{2.20}$$

A change of variable $t = \tau + ph$ was made, and the order of integration may be interchanged if all of the integrals are assumed to exist; that is, assuming

$$\| \mathbf{u} \|_p^2 \equiv (\mathbf{u}, \mathbf{u})_p \qquad \text{and} \qquad \| \mathbf{d} \|_h^2 \equiv (\mathbf{d}, \mathbf{d})_h$$

are both finite. The form for **Lu** is apparent in the last expression of (2.20).

Summarizing, a consistent slant stack pair **L** and $\mathbf{L}^T$ can be defined as

Slant **L**: $$d(h,t) = \int_{-\infty}^{\infty} dp\, u(p, t - ph) \tag{2.21}$$

Slant $\mathbf{L}^T$: $$u(p,\tau) = \int_{-\infty}^{\infty} dh\, d(p, \tau + ph) \tag{2.22}$$

The only difference between a slant stack and its adjoint is thus seen to be a change in sign of the dips to be summed over.

The transpose for the discrete slant stack of equation (2.15) is now immediately found by referring to expression (2.16):

$$\mathbf{d} = (\mathbf{P}_u \mathbf{L}^T \mathbf{P}_d)^T \mathbf{u} = \mathbf{P}_d \mathbf{L} \mathbf{P}_u \mathbf{u} \tag{2.23}$$

or,

$$d(h_j, t) = \sum_k u(p_k, t - p_k h_j)$$

To derive an adjoint L to the velocity stack operator, it is convenient to first define an inner product weighted by the time variable in each space:

[velocity] $$(\mathbf{u}_1, \mathbf{u}_2)_p \equiv \int_0^\infty\int_0^\infty \tau \, dp \, d\tau \, u_1(p,\tau) \, u_2(p,\tau) \tag{2.24}$$

[offset] $$(\mathbf{d}_1, \mathbf{d}_2)_h \equiv \int_0^\infty\int_0^\infty t \, dh \, dt \, d_1(h,t) \, d_2(h,t) \tag{2.25}$$

The adjoint is then found by combining these inner product definitions with equation (2.12), the definition for the velocity stack:

$$(\mathbf{Lu}, \mathbf{d})_h = (\mathbf{u}, \mathbf{L}^T\mathbf{d})_p = \int_0^\infty\int_0^\infty \tau \, dp \, d\tau \, u(p,\tau) \int_0^\infty dh \, d(h, \sqrt{\tau^2 + p^2h^2})$$

$$= \int_0^\infty\int_0^\infty\int_0^\infty \tau \, dp \, d\tau \, dh \, u(p,\tau) \, d(h, \sqrt{\tau^2 + p^2h^2}) \tag{2.26}$$

Now let $t = \sqrt{\tau^2 + p^2h^2}$; then $\tau = \sqrt{t^2 - p^2h^2}$ and $d\tau = t/\tau \, dt$, so that

$$(\mathbf{Lu}, \mathbf{d})_h = \int_0^\infty\int_{ph}^\infty dp \, dh \, t \, dt \, u(p, \sqrt{t^2 - p^2h^2}) \, d(h,t) \tag{2.27}$$

If we assume that $u(p, \sqrt{t^2 - p^2h^2}) = 0$ for $0 \le t \le ph$ (that is, $u(p,\tau)$ is zero for an imaginary time argument τ), the lower limit on the time integral may be replaced by zero, giving

$$(\mathbf{Lu}, \mathbf{d})_h = \int_0^\infty\int_0^\infty t \, dh \, dt \, d(h,t) \int_0^\infty dp \, u(p, \sqrt{t^2 - p^2h^2}) \tag{2.28}$$

The integral form for **Lu** can be identified as the innermost integral of equation (2.28), by comparing this equation with the previous definition for $(\cdot,\cdot)_h$ given in

equation (2.24). As was done in the slant stack case, all integrals are assumed to exist. Let us summarize the velocity stack case:

Velocity **L**:
$$d(h,t) = \int_0^\infty dp\ u(p, \sqrt{t^2 - p^2h^2}) \tag{2.29}$$

Velocity $\mathbf{L}^T$:
$$u(p,\tau) = \int_0^\infty dh\ d(h, \sqrt{\tau^2 + p^2h^2}) \tag{2.30}$$

Finally, the transpose for the discrete velocity stack is found by again using (2.23):

$$d(h_j,t) = \sum_k u(p_k, \sqrt{t^2 - p_k^2h_j^2}) \tag{2.31}$$

assuming that $u(p_k,\tau)$ is zero for imaginary time arguments.

The transpose to NMO and stacking can be recognized as reverse NMO and stacking. All points lying on the common elliptical path defined by $\tau^2 = t^2 - p^2h^2$ are summed into one offset. Programming the transpose operations is virtually identical to programming the original operations of slant stacking and velocity stacking; in fact, the only difference between the routines lies in the calculation of the moveout functions.

2.6 Two inverse problems

Assume that the discrete data set **d** is modeled by the following equation:

$$\mathbf{d} = \mathbf{P}_d\mathbf{L}\hat{\mathbf{u}} + \mathbf{n} \tag{2.32}$$

where $\hat{\mathbf{u}}$ is a continuous function in model space and **n** is an independent noise term. This discussion of **L** need not be restricted to the two cases already discussed: slant stacking and velocity stacking. For example, **L** may be a diffraction operator, or any other linear operator that models seismic data. The model $\hat{\mathbf{u}}$ then represents input parameters to the modeling scheme.

Let us denote as a *type I problem* the problem of inverting (2.32) to obtain $\hat{\mathbf{u}}$, or at best to obtain a sampled version $\mathbf{u}$ of $\hat{\mathbf{u}}$. First, assume that noise is absent; i.e., $\mathbf{n} = 0$. As long as the operator $\mathbf{L}$ is nonsingular, a unique $\hat{\mathbf{u}}$ would be obtained if "all" the data $\hat{\mathbf{d}}$ were available. But because of the limitations of sampling, the corresponding operator $\mathbf{P}_d\mathbf{L}$ might be noninvertible; that is, $\mathbf{P}_d\mathbf{L}$ can have a nontrivial null space. For example, it will be seen in the next chapter that a nontrivial null space always arises from data sampling when $\mathbf{L}$ is defined to be a slant stack.

A discrete linear system can be formed from (2.32) if the following assumptions are made: noise is absent, and $\hat{\mathbf{u}}$ is constrained to be zero outside the sampled region in the model domain, i.e., $\hat{\mathbf{u}} = \mathbf{u} = \mathbf{P}_u\mathbf{u} = \mathbf{P}_u\hat{\mathbf{u}}$. Equation (2.32) then reduces to

$$\mathbf{d} = \mathbf{P}_d\mathbf{L}\mathbf{P}_u\mathbf{u} \tag{2.33}$$

When the sampling projections $\mathbf{P}_d$ and $\mathbf{P}_u$ are discrete, equation (2.33) is a discrete linear system, though it can be very large in dimension.

When there is noise $\mathbf{n}$, equation (2.32) may be solved by minimizing

$$\| \mathbf{n} \|^2 = \| \mathbf{P}_d\mathbf{L}\mathbf{P}_u\mathbf{u} - \mathbf{d} \|^2 \tag{2.34}$$

The corresponding least squares system is

$$\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u\mathbf{u} = \mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{d} \tag{2.35}$$

or dropping the redundant projection term $\mathbf{P}_d$ on the right hand side,

$$\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}}\mathbf{u} = \hat{\mathbf{L}}^T\mathbf{d} \tag{2.36}$$

where $\hat{\mathbf{L}}$ is defined to be $\mathbf{P}_u\mathbf{L}$, an operator with a continuous input space and discrete output space. As seen by making a simple substitution, any solution $\mathbf{u}$ that satisfies (2.33) will also satisfy the least squares system (2.36).

If the inverse $(\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}})^{-1}$ does not exist, equation (2.35) is nonunique. There are two ways to overcome this difficulty. The first method is to derive the generalized inverse, or *pseudoinverse*, of $\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}}$, denoted by the plus-sign superscript:

[Type I Pseudoinverse] $$\mathbf{u} = (\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}})^+\hat{\mathbf{L}}^T\mathbf{d} \tag{2.37}$$

The pseudoinverse guarantees that the components of **u** in the null space of $\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}}$ are set equal to zero. A second method is to derive the *stochastic inverse*, which consists of adding positive weights to the diagonal of $\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}}$ until the modified linear system becomes nonsingular, and then inverting:

[Type I Stochastic Inverse] $$\mathbf{u} = (\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}} + \alpha^2\mathbf{I})^{-1}\hat{\mathbf{L}}^T\mathbf{d} \qquad (2.38)$$

The stochastic inverse (Aki & Richards, 1980, p. 695) allows the existence of nonzero values of **u** within the null space of $\hat{\mathbf{L}}^T\mathbf{P}_d\hat{\mathbf{L}}$.

How suitable the model (equation 2.32) is to the data at hand, depends on both the data and the choice made for **L**. Let us assume that the model $\hat{\mathbf{u}}$, from which the data were generated, satisfies the following properties: first, that the domain of nonzero $\hat{\mathbf{u}}$ be bounded, i.e. $\hat{\mathbf{u}} = \mathbf{u}$; second, that $\hat{\mathbf{u}}$ be sparse. By *sparseness* we mean the tendency of events to cluster together in model space.

A slant stack of a sparse model is easy to visualize: it consists of a few strong events possessing linear moveout. In a velocity stack, one example of a sparsely distributed model is a single-valued velocity function $v(\tau)$ embedded in the velocity plane. Figure 2.2(a) shows such a velocity distribution. When this model **u** is run through a velocity stacking operation, the generated data set simulates a common-midpoint gather (figure 2.2(b)). In chapter 5, we will impose the property of sparseness as a constraint, or more precisely, as a penalty function to the least squares system of equations (2.36), in order to obtain a more realistic estimate of the solution to the least squares system.

A *type II* problem arises when a data set has been slant stacked or velocity stacked (for example, so that a filtering operation could be performed in the stack domain), but a return to the original domain is desired. This is an inversion problem for **d** given **u**, and the objective is to return to the data domain with as much of the original data as possible. The roles of data space and model space are interchanged from the "Type I" problem to a "Type II" problem. The noise term in this

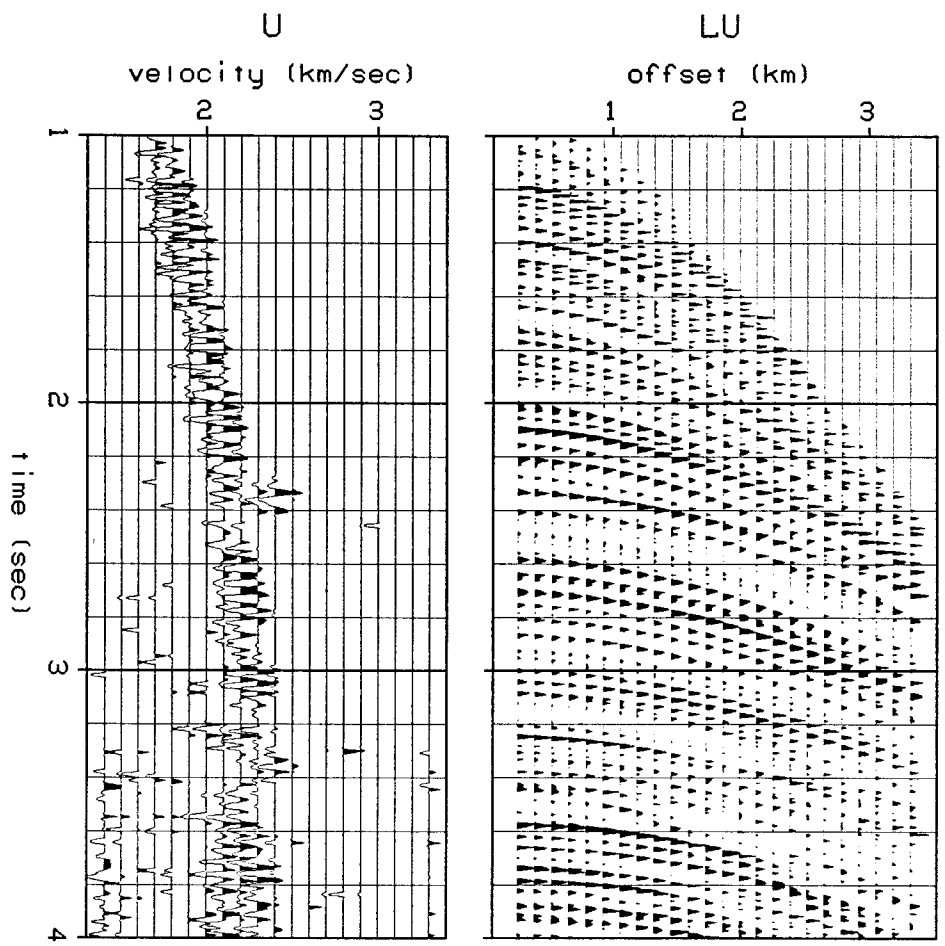

FIG. 2.2. Sparseness

(a) Example of a sparse model **u**, to be input into the velocity stack **L** in equation (2.32). In this example, events can be seen to cluster about a particular velocity function $v(\tau)$. The sparseness of **u** implies that power in the model domain (subject to the constraints imposed on **u**) moves as far away as possible from a uniform distribution. This velocity panel was actually generated from the real data of figure 1.1 by the "type I" inversion procedure discussed in chapter 5.

(b) The modeled output **Lu**. It has the characteristics of a typical common-midpoint gather; that is, events have hyperbolic moveout.

case is identically zero, since **u** comes directly from **d**:

$$\mathbf{u} = \mathbf{L}^T\mathbf{d} \tag{2.39}$$

The corresponding pseudoinversion and stochastic inversion formulas are:

[Type II Pseudoinverse] $$\mathbf{d} = (\hat{\mathbf{L}}\mathbf{P}_u\hat{\mathbf{L}}^T)^+\hat{\mathbf{L}}\mathbf{u} \tag{2.40}$$

[Type II Stochastic Inverse] $\mathbf{d} = (\hat{\mathbf{L}}\mathbf{P}_u\hat{\mathbf{L}}^T + \alpha^2\mathbf{I})^{-1}\hat{\mathbf{L}}\mathbf{u}$ (2.41)

where $\hat{\mathbf{L}}$ is assumed to have a continuous input space $\hat{\mathbf{u}}$ and a discrete output space **d**:

$$\hat{\mathbf{L}} \equiv \mathbf{P}_d\mathbf{L}$$

It shall be seen that there is virtually no formal difference between the expressions for the two operators $\mathbf{L}^T\mathbf{P}_d\mathbf{L}$ and $\mathbf{L}\mathbf{P}_u\mathbf{L}^T$ for slant stacks. There is a significant difference, however, between the two operators for velocity stacks.

2.7 Pseudoinverses

Pseudoinverses, also called generalized inverses (Lanczos, 1961), are useful means of solving least squares systems for which the solution **u** that minimizes the least squares functional is nonunique. Among all those **u** that satisfy the least squares system, the pseudoinverse is the unique **u** having minimum energy.

Consider the type I inverse problem of the last section (equation 2.32):

$$\mathbf{d} = \mathbf{P}_d\mathbf{L}\mathbf{P}_u\mathbf{u} + \mathbf{n} \tag{2.42}$$

The pseudoinverse of this problem will be denoted by

$$\mathbf{u} = (\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+\mathbf{d} \tag{2.43}$$

A pseudoinverse is uniquely determined by its *singular value decomposition (SVD)*. In an SVD, the discrete linear system $\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ may be decomposed into

$$\mathbf{P}_d\mathbf{L}\mathbf{P}_u = \mathbf{U}\Sigma\mathbf{V}^T \tag{2.44}$$

where **U** and **V** are unitary matrices and Σ is diagonal. **U** has the dimensionality of data space and **V** has the dimensionality of model space; consequently **V** and **U** do not necessarily have to be matrices of equal dimension. The matrix Σ is a (possibly nonsquare) diagonal matrix composed of singular values, all nonnegative and ordered from large to small along the diagonal. In this form the SVD is essentially unique, apart from arbitrary rotations of the orthogonal basis vectors in **U** and **V**

which correspond to repeated singular values (Stewart, 1973, p. 319).

The presence of zero singular values indicates the existence of a nontrivial null space for the operator $\mathbf{P}_d\mathbf{L}\mathbf{P}_u$; almost all the operators that we shall consider have some singular values equal to zero. Let us divide the SVD system of (2.44) into the zero and nonzero singular value portions:

$$\mathbf{P}_d\mathbf{L}\mathbf{P}_u = \begin{bmatrix}\mathbf{U}_p & \mathbf{U}_o\end{bmatrix}\begin{bmatrix}\Sigma_p & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T\\ \mathbf{V}_o^T\end{bmatrix} \tag{2.45}$$

where $[\mathbf{U}_p\ \mathbf{U}_o]$ is the partition of $\mathbf{U}$ into two sets of column vectors $\mathbf{U}_p$ and $\mathbf{U}_o$. Σ_p is square, and contains the positive nonzero singular values. In fact, equation (2.45) could be rewritten as

$$\mathbf{P}_d\mathbf{L}\mathbf{P}_u = \mathbf{U}_p\Sigma_p\mathbf{V}_p^T \tag{2.46}$$

However we shall continue to write the pseudoinverse in the form of equation (2.45). The column vectors of $\mathbf{U}_p$, $\mathbf{U}_o$, $\mathbf{V}_p$, $\mathbf{V}_o$ happen to be orthogonal basis vectors that respectively span the four fundamental subspaces of $\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ (Strang, 1980, p. 139):

$\mathbf{V}_o$ spans the input (model) null space,

$\mathbf{U}_o$ spans the output (data) null space,

$\mathbf{V}_p$ spans the input (model) non-null space,

$\mathbf{U}_p$ spans the output (data) non-null space.

Notice that equation (2.46) shows that the non-null input and output spaces have the same dimensionality.

The adjoint to (2.45) can be immediately written down:

$$\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d = \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{U}_p^T\\ \mathbf{U}_o^T\end{bmatrix} \tag{2.47}$$

Because $\mathbf{U}^T\mathbf{U}$ is equal to the identity $\mathbf{I}$, the least squares system of equation (2.35) possesses a similar SVD:

$$\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u = \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^2 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T\\ \mathbf{V}_o^T\end{bmatrix} \tag{2.48}$$

Equation (2.48) also is the eigenvalue decomposition of the least squares system $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$.

By definition, the pseudoinverse of $\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ is formed by inverting the *nonzero* singular values comprising the diagonal matrix Σ_p:

$$(\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+ = \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^{-1} & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{U}_p^T\\ \mathbf{U}_o^T\end{bmatrix} \tag{2.49}$$

Likewise, the pseudoinverse of $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ is seen to be:

$$(\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+ = \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^{-2} & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T\\ \mathbf{V}_o^T\end{bmatrix} \tag{2.50}$$

Therefore, the relationship between the two pseudoinverses is

$$(\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+ = (\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+(\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d) \tag{2.51}$$

The process of applying the pseudoinverse operator to **d** (equation 2.43) can be broken down into the following steps:

1) Apply the discrete transpose operation $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d$,

2) Transform into the eigenvalue space of $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ using $\mathbf{V}^T$,

3) Invert the nonzero eigenvalues of Σ_p,

4) Apply the inverse transform operator **V**.

If the eigenvalue decomposition of the least squares operator in equation (2.48) is known, following this procedure will yield the pseudoinverse. Of course, the solution will have a zero component in the input null space of $\mathbf{P}_d\mathbf{L}\mathbf{P}_u$. The dimension of the transform pair ($\mathbf{V}^T$, **V**) is proportional to the size of the model space. Because of the large dimensions of a typical seismic model, it is generally impossible to determine the singular value decomposition of any operator in model space. Fortunately there are exceptions: the operator might separate into smaller dimensional components, or **V** might happen to coincide with a common transformation, such as a

Fourier transform. Slant stacking is such an exception, but only if a particular structure exists for $\mathbf{P}_d$, the truncation operator in data space. The next chapter shall discuss this aspect of slant stacks: the effects that $\mathbf{P}_d$ has on the data and the form of the pseudoinverse.

An alternative to using the pseudoinverse is to add a small value to the diagonal of the least squares operator $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$. Doing this easily removes the problem of the input null space. For example, consider what happens to the SVD when a constant diagonal $\alpha^2\mathbf{I}$ is added to $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$:

$$\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u + \alpha^2\mathbf{I} = \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^2+\alpha^2\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \alpha^2\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T \\ \mathbf{V}_o^T\end{bmatrix} \tag{2.52}$$

Because all the singular values of equation (2.52) are positive, the least squares operator is now nonsingular. When α^2 is small, the inverse to (2.52) is approximately defined as

$$(\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u + \alpha^2\mathbf{I})^{-1} \approx \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^{-2} & \mathbf{0} \\ \mathbf{0} & \alpha^{-2}\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T \\ \mathbf{V}_o^T\end{bmatrix} \tag{2.53}$$

This operator will behave properly only if there is no component of the input lying in the null space $\mathbf{V}_o$. If there were such a component, α^{-2} would amplify it greatly. Null space components introduced by numeric round-off will have the same effect. However, the stochastic inverse solution $\mathbf{u}$ (equation 2.38) guarantees that no component reaches the null space, at least under the constraints of exact arithmetic:

$$\mathbf{u} = (\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u + \alpha^2\mathbf{I})^{-1}\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{d} \tag{2.54}$$

To see this, expand equation (2.54) in terms of the singular value decomposition:

$$\begin{aligned}\mathbf{u} &= \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}(\Sigma_p^2+\alpha^2\mathbf{I})^{-1} & \mathbf{0} \\ \mathbf{0} & \alpha^{-2}\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T \\ \mathbf{V}_o^T\end{bmatrix}\begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{U}_p^T \\ \mathbf{U}_o^T\end{bmatrix}\mathbf{d} \\ &= \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}(\Sigma_p^2+\alpha^2\mathbf{I})^{-1}\Sigma_p & \mathbf{0} \\ \mathbf{0} & \alpha^{-2}\mathbf{I}\cdot\mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{U}_p^T \\ \mathbf{U}_o^T\end{bmatrix}\mathbf{d}\end{aligned} \tag{2.55}$$

As $\alpha \to 0$, this expression becomes identical to the pseudoinverse (2.48), and the

stochastic inverse yields the same solution as the pseudoinverse. The choice of using one inversion over another becomes a matter of convenience: whichever one happens to be easier to implement can be used.

In general, the preceding conclusion is not true when $\alpha^2\mathbf{I}$ is replaced by an arbitrary diagonal **D**. Adding non-constant positive values to the diagonal of $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ changes the singular value decomposition: **V** will be altered.

We shall now examine the effect of the projection $\mathbf{P}_d$ on the pseudoinverse, and on the singular value decomposition. Let

$$\mathbf{L}\mathbf{P}_u = \mathbf{U}\Sigma\mathbf{V}^T \tag{2.56}$$

be the SVD of an operator that has no truncation or aliasing effects in the data domain; i.e., $\mathbf{P}_d$ is absent. In the presence of truncation,

$$\mathbf{P}_d\mathbf{L}\mathbf{P}_u = \mathbf{P}_d\mathbf{U}\Sigma\mathbf{V}^T \tag{2.57}$$

However this expression is not in the form of an SVD, because $\mathbf{P}_d\mathbf{U}$ is not unitary. If it can now be assumed that the columns of **U** are the eigenvectors of $\mathbf{P}_d$, expression (2.57) can be easily converted into an SVD by using the relation $\mathbf{P}_d\mathbf{U} = \mathbf{U}\Lambda$, where Λ contains the eigenvalues of $\mathbf{P}_d$:

$$\mathbf{P}_d\mathbf{L}\mathbf{P}_u = \mathbf{U}\Lambda\Sigma\mathbf{V}^T \tag{2.58}$$

The eigenvalues Λ of the projection operator (zeros and ones) can be arbitrarily rearranged by freely permuting the columns of **U**. It remains only to permute, say, the columns of **U** into $\tilde{\mathbf{U}}$, in order to separate the zeros from the ones in Λ. Once this arrangement is made, $\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ can be written in the form of a singular value decomposition:

$$\begin{aligned}\mathbf{P}_d\mathbf{L}\mathbf{P}_u &= \tilde{\mathbf{U}}\tilde{\Lambda}\Sigma\mathbf{V}^T \\ &= \begin{bmatrix}\tilde{\mathbf{U}}_p & \tilde{\mathbf{U}}_o\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\Sigma_p & \mathbf{0} \\ \mathbf{0} & \Sigma_o\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T \\ \mathbf{V}_o^T\end{bmatrix}\end{aligned} \tag{2.59}$$

It is obvious that the effect of $\mathbf{P}_d$ in this case is to "expand" the size of the null

space, by zeroing out the otherwise nonzero singular values Σ_o of $\mathbf{LP}_u$. The remaining singular values Σ_p are unaffected by $\mathbf{P}_d$.

Now form the SVDs of the corresponding normal equations with $\mathbf{P}_d$ either present or absent:

$$\begin{aligned} \mathbf{P}_u\mathbf{L}^T\mathbf{L}\mathbf{P}_u &= \mathbf{V}\Sigma^2\mathbf{V}^T \\ &= \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^2 & 0 \\ 0 & \Sigma_o^2\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T \\ \mathbf{V}_o^T\end{bmatrix} \end{aligned} \tag{2.60}$$

and

$$\begin{aligned} \mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u &= \mathbf{V}\Sigma^2\tilde{\mathbf{U}}^T\mathbf{P}_d\tilde{\mathbf{U}}\Sigma\mathbf{V}^T = \mathbf{V}\Sigma^2\tilde{\Lambda}\mathbf{V}^T \\ &= \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^2 & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T \\ \mathbf{V}_o^T\end{bmatrix} \end{aligned} \tag{2.61}$$

Comparing these two SVDs reveals that, requiring $\mathbf{U}$ to consist of the eigenvectors of $\mathbf{P}_d$, is equivalent to requiring that $\mathbf{P}_u\mathbf{L}^T\mathbf{L}\mathbf{P}_u$ and $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$ share eigenvectors. This requirement might seem to place excessive restrictions on $\mathbf{P}_d$; in general, it cannot be satisfied. However, as shall be seen in the next chapter, it is automatically satisfied for slant stacks.

If the pseudoinverse of equation (2.60) is now multiplied to $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d$, the resulting expression is identical to the pseudoinverse $(\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+$:

$$\begin{aligned} (\mathbf{P}_u\mathbf{L}^T\mathbf{L}\mathbf{P}_u)^+\mathbf{P}_u\mathbf{L}\mathbf{P}_d &= \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^{-2} & 0 \\ 0 & \Sigma_o^{-2}\end{bmatrix}\begin{bmatrix}\mathbf{V}_p^T \\ \mathbf{V}_o^T\end{bmatrix}\begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{U}_p^T \\ \mathbf{U}_o^T\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{V}_p & \mathbf{V}_o\end{bmatrix}\begin{bmatrix}\Sigma_p^{-1} & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{U}_p^T \\ \mathbf{U}_o^T\end{bmatrix} \end{aligned} \tag{2.62}$$

This result happens to be identical to the pseudoinverse of equation (2.49). This is a third way to find the pseudoinverse solution $\mathbf{u}$ to (2.42): i.e., apply the transpose operation $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d$, then apply the pseudoinverse $(\mathbf{P}_u\mathbf{L}^T\mathbf{L}\mathbf{P}_u)^+$, which is derived from an operator not restricted by the missing data projection $\mathbf{P}_d$. For the stacking operators we have considered up to now, the pseudoinverse in equation (2.62) is

much easier to construct than the truncated version $(\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+$.

The following **theorem** summarizes the various ways we have discussed to solve the noisy inversion problem of equation (2.42):

Let **d** be given data, **L** be a continuous linear system, and **n** an independent noise term such that

$$\mathbf{d} = \mathbf{P}_d\mathbf{L}\mathbf{P}_u\mathbf{u} + \mathbf{n} \tag{2.63}$$

(a) The pseudoinverse **u** given by

$$\mathbf{u} = (\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u)^+\mathbf{P}_u\mathbf{L}^T\mathbf{d} \tag{2.64}$$

is the unique minimum-energy least squares solution to (2.63).

(b) As the positive scalar α goes to zero, the stochastic inverse

$$\mathbf{u} = (\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u + \alpha^2\mathbf{I})^{-1}\mathbf{P}_u\mathbf{L}^T\mathbf{d} \tag{2.65}$$

yields the same unique estimate **u** as does equation (2.64).

(c) Furthermore, if $\mathbf{P}_u\mathbf{L}^T\mathbf{L}\mathbf{P}_u$ shares eigenvectors (or eigenspaces) with $\mathbf{P}_u\mathbf{L}^T\mathbf{P}_d\mathbf{L}\mathbf{P}_u$, then the estimate **u** given by

$$\mathbf{u} = (\mathbf{P}_u\mathbf{L}^T\mathbf{L}\mathbf{P}_u)^+\mathbf{P}_u\mathbf{L}^T\mathbf{d} \tag{2.66}$$

is equal to the estimates **u** given in (2.64) and (2.65) above.

2.8 Summary

The generalized inversion procedures of the last section always begin with the application of, first, the adjoint operator $\mathbf{L}^T$, and then one of a choice of least squares inverses. In the following chapters, we shall see that applying the pseudo-inverse $(\mathbf{L}^T\mathbf{P}_d\mathbf{L})^+$ reduces to a relatively local filtering operation; it may be said that $\mathbf{L}^T$ does "most" of the work of the inversion. The operation performed by $(\mathbf{L}^T\mathbf{P}_d\mathbf{L})^+$ is cosmetic; for example, it is able to restore the original frequency content to the model.

The projectors $\mathbf{P}_d$ and $\mathbf{P}_u$ have a different effect on a type I problem than they have on a type II problem. For type I, the data set is assumed to have been generated (via **L**) from a model that is bounded in p. Thus the effect of $\mathbf{P}_u$ is nil when $\mathbf{P}_u\mathbf{u} = \mathbf{u}$: that is, when the true solution **u** lies within the limits of the sampled model space defined by $\mathbf{P}_u$. If it does not, the model space can always be expanded by a finer sampling (i.e., $\mathbf{P}_u$ can be extended) until the condition $\mathbf{P}_u\mathbf{u} = \mathbf{u}$ is once again met. In contrast, finer sampling in the data domain (extending $\mathbf{P}_d$) is not possible unless the experiment that originally created the data is reconducted with the required denser sampling. If $\mathbf{P}_u\mathbf{u} = \mathbf{u}$, $\mathbf{P}_u$ may be dropped altogether from the equations of section 2.7, including the formulas for the pseudoinverse and stochastic inverse. $\mathbf{P}_d$ on the other hand can never really be dropped, and missing data, accounted for by $\mathbf{P}_d$, can have a crucial effect on the various inversions. Therefore, for a type I problem, the presence of enough offsets is crucial for a good inversion.

The opposite conditions hold in a type II problem. The data set has already been transformed into model space, and a successful recovery of the original data set depends on there having been adequate sampling in the model domain. Having enough velocities (or dips as the case may be) is crucial for the inversion of a type II problem.