Chapter 1

Plane waves in three dimensions

In this chapter we seek a deeper understanding of **plane waves** in *three* dimensions where the examples and theory typically refer to functions of time t and two space coordinates (x, y), or to 3-D migration images where the t coordinate is depth or traveltime depth. As in chapter **??** we need to decompose data volumes into subcubes as in Figure 1.1.

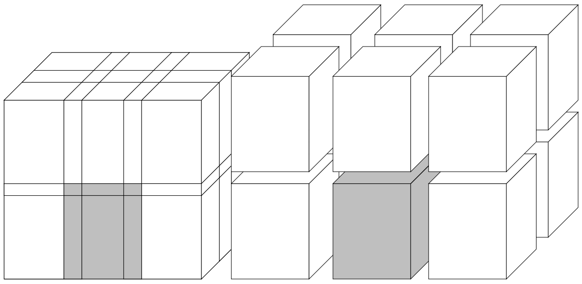

Figure 1.1: Left is space of inputs and outputs. Right is during analysis. lmn-rayab3D [NR]

In this chapter we will see that the wave model implies the 3-D whitener is not a cube filter but two planar filters. The wave model allows us to determine the scale factor of a signal, even where signals fluctuate in strength because of interference. Finally, we examine the local-monoplane concept that uses the superposition principle to distinguish a sedimentary model cube from a data **cube**.

1.1 THE LEVELER: A VOLUME OR TWO PLANES?

In *two* dimensions we took levelers to be PEFs, small rectangular planes of numbers where the time axis included enough points to include reasonable stepouts and the space axis contained one level plus another space level for each plane wave slope supposed to be present.

We saw that a whitening filter in *three* dimensions is a small volume with shape defined by subroutine `setfree()` (page **??**). It might seem natural that the number of points on the x- and y-axes be related to the number of plane waves present. Instead, I assert that if the volume contains plane waves, we don't want a *volume* filter to whiten it; we can use a *pair* of *planar* filters to do so and the order of those filters is the number of planes thought to be simultaneously present. I have no firm mathematical proofs, but I offer you some interesting discussions, examples, and computer tools for you to experiment with. It seems that some applications call for the volume filter while others call for the two planes. Since two planes of numbers generally contain many fewer adjustable values than a volume, statistical estimation reasons also favor the planes.

> What is the lowest order filter applied to a volume that will destroy one and only one slope of plane wave?

First we seek the answer to the question, "what is the lowest order filter that will destroy one and only one plane?" To begin with, we consider that plane to be horizontal so the volume of numbers is $f(t,x,y) = b(t)$ where $b(t)$ is an arbitrary function of time. One filter that has zero valued output (destroys the plane) is $\partial_x \equiv \partial/\partial x$. Another is the operator $\partial_y \equiv \partial/\partial y$. Still another is the **Laplacian operator** which is $\partial_{xx} + \partial_{yy} \equiv \partial^2/\partial x^2 + \partial^2/\partial y^2$.

The problem with $\partial/\partial x$ is that although it destroys the required plane, it also destroys $f(t,x,y) = a(t,y)$ where $a(t,y)$ is an *arbitrary* function of (t,y) such as a cylinder with axis parallel to the x-axis. The operator $\partial/\partial y$ has the same problem but with the axes rotated. The Laplacian operator not only destroys our desired plane but it also destroys the well known nonplanar function $e^{ax}\cos(ay)$ which is just one example of the many other interesting shapes that constitute solutions to Laplace's equation.

I remind you of a basic fact: When we set up the regression $\mathbf{0} \approx \mathbf{Af}$, the **quadratic form** minimized is $\mathbf{f'A'Af}$ which by differentiation with respect to $\mathbf{f'}$ gives us (in a constraint-free region) $\mathbf{A'Af} = \mathbf{0}$, so minimizing the volume integral (actually the sum) of the squares of the components of the gradient implies Laplace's equation is satisfied.

The lowest order filter in a volume that will destroy level planes and no other wave slope is a filter that has one input and *two outputs*. That filter is the gradient, $(\partial/\partial x, \partial/\partial y)$. Both outputs vanish if and only if the plane has the proper horizontal orientation. Other objects and functions are not extinguished (except for the non-wave-like function $f(t,x,y) = \text{const}$). It is annoying that we must deal with *two* outputs and that will be the topic of further discussion.

A wavefield of tilted parallel planes is $f(t,x,y) = g(\tau - p_x x - p_y y)$ where $g()$ is an arbitrary one-dimensional function. The operator that destroys these tilted planes is the two-component operator $(\partial_x + p_x\partial_t,\ \partial_y + p_y\partial_t)$.

The operator that destroys a family of dipping planes

$$f(t, x, y) \quad = \quad g(\tau - p_x x - p_y y)$$

is

$$\begin{bmatrix} \frac{\partial}{\partial x} & + & p_x \frac{\partial}{\partial t} \\ \\ \frac{\partial}{\partial y} & + & p_y \frac{\partial}{\partial t} \end{bmatrix}$$

1.1.1 PEFs overcome spatial aliasing of difference operators

The problem I found with finite-difference representations of differential operators is that they are susceptible to spatial aliasing. Even before they encounter spatial aliasing, they are susceptible to accuracy problems known in finite-difference wave propagation as "frequency dispersion". The aliasing problem can be avoided by using spatial prediction operators such as

$$\begin{array}{cc} \cdot & a \\ \cdot & b \\ 1 & c \\ \cdot & d \\ \cdot & e \end{array} \tag{1.1}$$

where the vertical axis is time; the horizontal axis is space; and the "$\cdot$"s are zeros. Another possibility is the 2-D whitening filter

$$\begin{array}{cc} f & a \\ g & b \\ 1 & c \\ \cdot & d \\ \cdot & e \end{array} \tag{1.2}$$

Imagine all the coefficients vanished but $d = -1$ and the given 1. Such filters would annihilate the appropriately sloping plane wave. Slopes that are not exact integers are also approximately extinguishable because the adjustable filter coefficients can interpolate in time. Filters like (1.2) do the operation $\partial_x + p_x \partial_t$ which is a component of the gradient in the plane of the wavefront *and* they include a temporal deconvolution aspect and a spatial coherency aspect. My experience shows that the operators (1.1) and (1.2) behave significantly differently in practice and I am not prepared to fully explain the difference but it seems to be similar to the gapping of one-dimensional filters.

You might find it alarming that your teacher is not fully prepared to explain the difference between a volume and two planes, but please remember that we are talking about the factorization of the volumetric inverse covariance matrix. Covariance matrices are well known to have structure, but books on theory typically handle them as simply $\lambda\mathbf{I}$. Anyway, where ever you see an $\mathbf{A}$ in a three-dimensional context, you may wonder whether it should be interpreted as a cubic filter that takes one volume to another, or as two planar filters that take one volume to two volumes as shown in Figure 1.2.

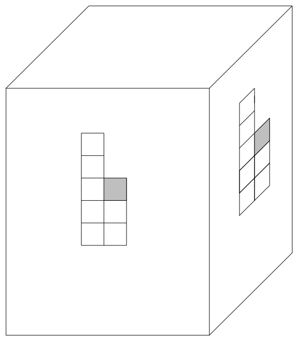

Figure 1.2: An inline 2-D PEF and a crossline 2-D PEF both applied throughout the volume. To find each filter, minimize each output power independently. lmn-rayab3Doper [NR]

1.1.2 My view of the world

I start from the idea that the four-dimensional world (t, x, y, z) is filled with expanding spherical waves and with quasispherical waves that result from reflection from quasiplanar objects and refraction through quasihomogeneous materials. In an observational data cube we rarely, if ever, see an entire expanding spherical wave but we normally have a two- or three-dimensional slice with some wavefront curvature. We analyze data subcubes that I call bricks. In any brick we see only local patches of apparent plane waves. I call them platelets. From the microview of this brick, the platelets come from the "great random point generator in the sky" which then somehow convolves the random points with a platelike impulse response. If we could deconvolve these platelets back to their random source points there would be nothing left inside the brick because the energy would have gone outside. We would have destroyed the energy inside the brick. If the platelets were coin shaped, then the gradient magnitude would convert each coin to its circular rim. The plate sizes and shapes are all different and they damp with distance from their centers like Gaussian beams. If we observed *rays* instead of wavefront platelets then we might think of the world as being filled with noodles, and then....

How is it possible that in a small brick we can do something realistic about deconvolving a spheroidal impulse response that is much bigger than the brick? The same as in one dimension where in a small time interval we can estimate the correct deconvolution filter of a long resonant signal. A three-point filter destroys a sinusoid.

The inverse filter to the expanding spherical wave might be a huge cube. Good approximations to this inverse at the brick level might be two small planes. Their time extent would be chosen to encompass the slowest waves, and their spatial extent could be two or three points representing the idea that normally we can listen to only one person at a time, occasionally we can listen to two, and we can never listen to three people talking at the same time.

1.2 WAVE INTERFERENCE AND TRACE SCALING

Although neighboring seismometers tend to show equal powers, the energy on one seismometer can differ greatly from that of a neighbor for both theoretical reasons and practical ones. Should a trace ever be rescaled to give it the same energy as its neighbors? Here we review the strong theoretical arguments against rescaling. In practice, however, especially on land where coupling is irregular, scaling seems a necessity. The question is, what can go wrong if we scale traces to have equal energy, and more basically, where the proper **scale factor** cannot be recorded, what should we do to get the best scale factor? A related question is how to make good measurements of amplitude versus offset. To understand this we review the fundamentals of wave interference.

Theoretically, a scale factor problem arises because *locally,* wavefields add, not energies. Nodes on standing waves are familiar from theory but they could give you the wrong idea that the concept of node is one that applies only with sinusoids. Actually, destructive interference arises anytime a polarity reversed waveform bounces back and crosses itself. Figure 1.3 shows two waves of opposite polarity crossing each other. Observe that one seismogram has zero valued signal while its neighbors

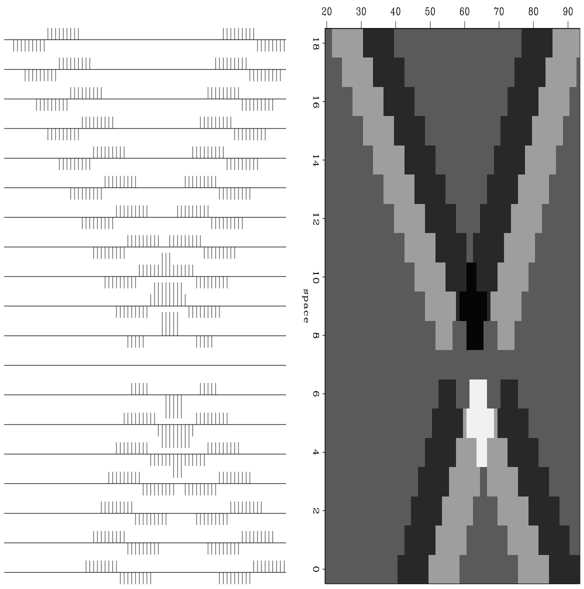

Figure 1.3: Superposition of plane waves of opposite polarity. lmn-super [NR]

have anomalously higher amplitudes and higher energies than are found far away from the interference. The situation shown in figure 1.3 does not occur easily in nature. Reflection naturally comes to mind, but usually the reflected wave crosses the incident wave at a later time and then they don't extinguish. Approximate extinguishing occurs rather easily when waves are quasi monochromatic. We will soon see, however, that methodologies for finding scales all begin with deconvolution and that eliminates the monochromatic waves.

1.2.1 Computing the proper scale factor for a seismogram

With data like Figure 1.3, rescaling traces to have equal energy would obviously be wrong. The question is, "how can we determine the proper scale factor?" As we have seen, a superposition of N plane waves exactly satisfies an N-th order (in x) difference equation. Given a 2-D wave field we can find its PEF by minimizing output power. Then we ask the question, could rescaling the traces give a lower output power? To answer this, we set up an optimization problem: given the leveler (be it a cubic PEF or two planar ones), find the best trace scales. (After solving this, we could return to re-estimate the leveler, and iterate.) To solve for the scales, we need a subroutine that scales traces and the only tricky part is that the adjoint should bring us back to the space of scale factors. This is done by `scaletrace()` (page 6)

```
subroutine scaletrace( adj,add, data,n1,n2,  scale,     sdata )
integer                adj,add,      n1,n2,                     i1,i2
real                              data(n1,n2), scale(n2), sdata( n1,n2)
call adjnull(          adj,add,                scale,n2,  sdata, n1*n2)
do i2= 1, n2
do i1= 1, n1
        if( adj == 0)
                sdata(i1,i2) = sdata(i1,i2) + scale(   i2) * data(i1,i2)
        else
                scale(   i2) = scale(   i2) + sdata(i1,i2) * data(i1,i2)
return; end
```

Notice the adjoint estimates scales by forming an inner product of the raw data on the previously scaled data. Let the operator implemented by `scaletrace()` (page 6) be denoted by $\mathbf{D}$ which is mnemonic for "data" and for "diagonal matrix" and let the vector of scale factors be denoted by $\mathbf{s}$ and the leveler by $\mathbf{A}$. Now we consider the regression $\mathbf{0} \approx \mathbf{ADs}$. The trouble with this regression is that the solution is obviously $\mathbf{s} = \mathbf{0}$. To avoid the trivial solution $\mathbf{s} = \mathbf{0}$ we can choose from a variety of supplemental regressions. One possibility is that for the i-th scale factor we could add the regression $s_i \approx 1$. Another possibility, perhaps better if some of the signals have the opposite of the correct polarity, is that the sum of the scales should be approximately unity. I regret that time has not yet allowed me to identify some interesting examples and work them through.

1.3 LOCAL MONOPLANE ANNIHILATOR

LOMOPLAN (LOcal MOno PLane ANnihilator) is a data-adaptive filter that extinguishes a local monoplane, but cannot extinguish a superposition of several planes. We presume an ideal sedimentary model as made of (possibly curved) parallel layers.

Because of the superposition principle, data can be a superposition of several plane waves, but the ideal model should consist locally of only a single plane. Thus, LOMOPLAN extinguishes an ideal model, but not typical data. I conceived of LOMOPLAN as the "ultimate" optimization criterion for inversion problems in reflection seismology (?) but it has not yet demonstrated that it can attain that lofty goal. Instead, however, in two dimensions it is useful in data interpretation and in data quality inspection.

The main way we estimate parameters in reflection seismology is that we maximize the coherence of theoretically redundant measurements. Thus, we estimate velocity and statics shifts by maximizing something like the power in the stacked data. Here I propose another optimization criterion for estimating model parameters and missing data. An interpreter looking at a migrated section containing two dips in the same place suspects wave superposition more likely than bedding texture superposition. To minimize the presence of multiple dipping events in the same place, we should use the mono plane annihilator (**MOPLAN**) filter as the weighting operator in any regression. Since the filter is intended for use on images or migrated data, not on data directly, I call it a *plane* annihilator, not a *planewave* annihilator. (A time-migration or merely a stack, however, might qualify as an image.) We should avoid using the word "wavefront" because waves readily satisfy the superposition principle, whereas images do not, and it is this aspect of images that I advocate and formulate as "prior information."

An example of a MOPLAN in two dimensions, $(\partial_x + p_x \partial_\tau)$, is explored in chapter 4 of **PVI** (?) where the main goal is to estimate the (τ, x)-variation of p_x. Another family of MOPLANs arise from multidimensional prediction-error filtering described earlier in this book and in PVI chapter 8.

Here I hypothesize that a MOPLAN may be a valuable weighting function for many estimation problems in seismology. Perhaps we can estimate statics, interval velocity, and missing data using the principle of minimizing the power out of a LOcal MOno PLane ANnihilator (LOMOPLAN) on a migrated section. Thus, those embarrassing semicircles that we have seen for years on our migrated sections may hold one of the keys for unlocking the secrets of statics and lateral velocity variation. I do not claim this concept is as powerful as our traditional methods. I merely claim that we have not yet exploited this concept in a systematic way and that it might prove useful where traditional methods break.

For an image model of nonoverlapping curved planes a suitable choice of weighting function for regression (model covariance) is the local filter that destroys the best fitting local plane.

1.3.1 Mono-plane deconvolution

The coefficients of a 2-D monoplane annihilator filter are defined to be the same as those of a 2-D PEF of spatial order unity, in other words, those defined by either (1.1) or (1.2).

The filter can be lengthened in time but not in space. The choice of exactly two columns is a choice to have an analytic form that can exactly destroy a single plane, but cannot destroy two. With two signals that are statistically independent, the filter (1.2) reduces to the well-known prediction-error filter in the left column and zeros in the right column. If the filter coefficients were extended in both directions on t and to the right on x, it would flatten the two-dimensional spectrum of the input.

1.3.2 Monoplanes in local windows

The earth dip changes rapidly with location. In a small region there is a local dip and dip bandwidth which determines the best LOMOPLAN (LOcal MOPLAN). To see how to cope with the edge effects of filtering in a small region, and to see how to patch together these small regions, recall subroutine `patch2()` (page ??) and the weighting subroutines that work with it.

Figure 1.4 shows a synthetic model that illustrates local variation in bedding. Notice dipping bedding, curved bedding, unconformity between them, and a fault in the curved bedding. Also, notice the image has its amplitude tapered to zero on the left and right sides. After local monoplane annihilation (LOMOPLAN) the continuous bedding is essentially gone. The fault and unconformity remain.

> The local spatial prediction-error filters contain the essence of a factored form of the inverse covariance matrix of the model.

The local spatial prediction-error filters contain the essence of a factored form of the inverse covariance matrix of the model. A covariance matrix has two aspects: a spectral aspect in the prediction-error filter and an amplitude aspect as in gain leveling. Thus a possible additional step is gaining by the inverse square root of the local power.

Since the plane waves are Local, so the illustrations were made with subroutine `lopef2()` (page ??).

1.3.3 Crossing dips

Figure 1.5 deserves careful study. The input frame is dipping events with amplitudes slowly changing as they cross the frame. The dip of the events is not commensurate with the mesh so linear interpolation is used which accounts for the irregularity along an event. The output panel tends to be small where there is only a single dip present.

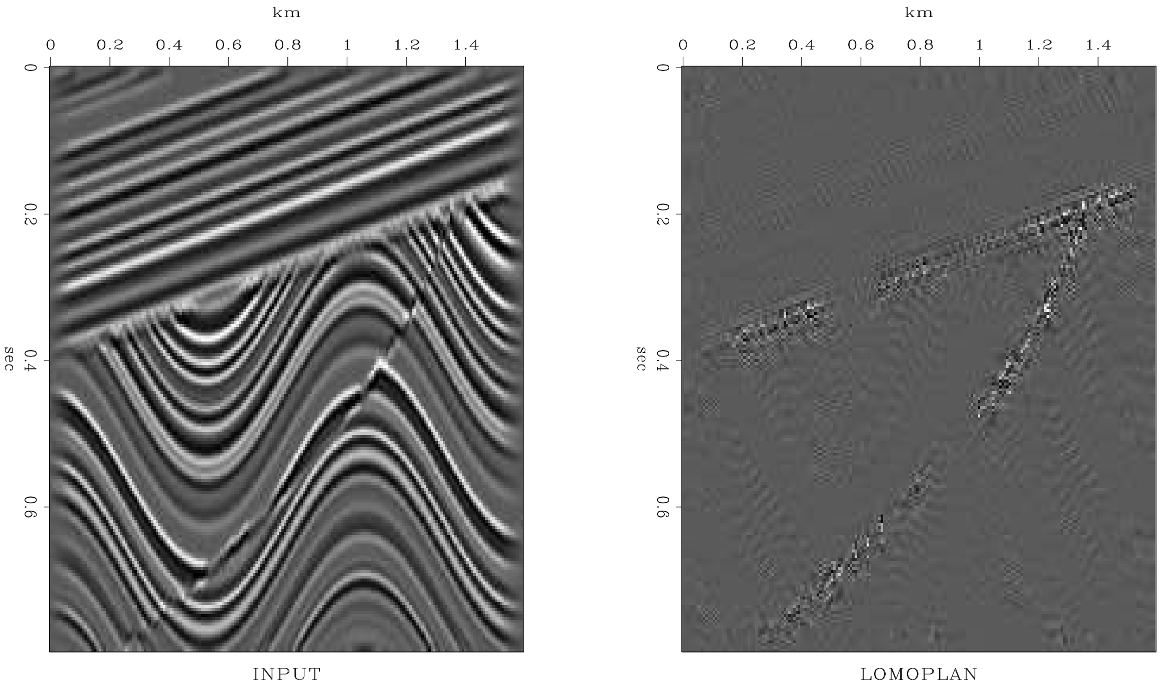

Figure 1.4: Left is a synthetic reflectivity model. Right is the result of local monoplane annihilation. lmn-sigmoid0 [NR]

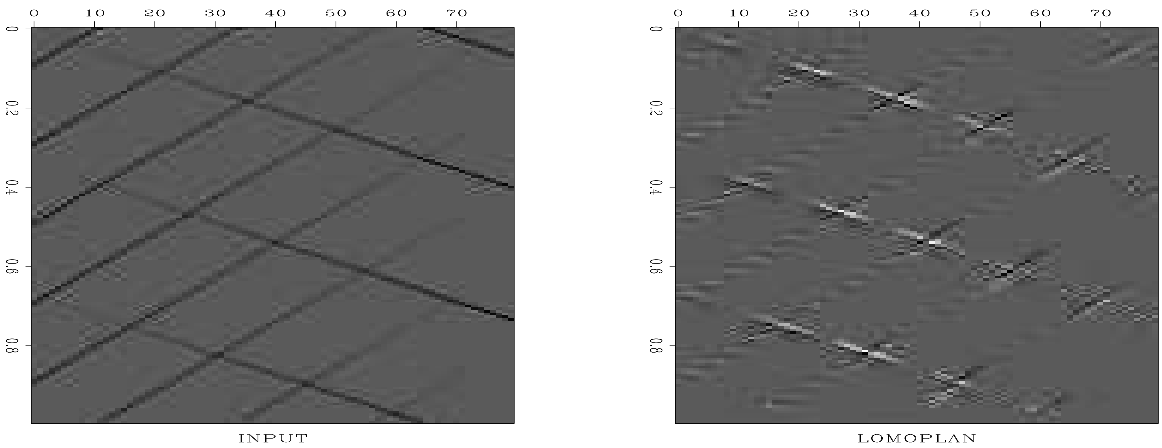

Figure 1.5: Conflicting dips before and after application of a local monoplane annihilator. Press button for movie. The movie sequence is: 1: data, 2: data after LOMOPLAN, 3: like previous but windows not overlapping, 4: predicted data lmn-conflict [NR]

Where two dips cross, they tend to be equal in magnitude. Studying the output more carefully, we notice that of the two dips, the one that is strongest on the input becomes irregular and noisy on the output whereas the other dip tends to remain phase coherent.

I could rebuild Figure 1.5 to do a better job of suppressing monodip areas if I passed the image through a lowpass filter, and then designed a gapped deconvolution operator. Instead, I preferred to show you high frequency noise in the place of an attenuated wavefront.

The residual of prediction-error deconvolution tends to have a white spectrum in time. This aspect of deconvolution is somewhat irritating and in practice it requires us to postfilter for display to regain continuity of signals. As is well known (PVI, for example), an alternative to postfiltering is to put a gap in the filter. This should work with 2-D filters too, but it is too early to describe how experimenters will ultimately choose to arrange gaps, if any, in 2-D filters. There are some interesting possibilities. (Inserting a gap also reduces the required number of CG iterations.)

1.3.4 2-D LOMOPLAN field data tests

Although the LOMOPLAN concept was developed for geophysical *models*, not raw *data*, initial experience showed that the LOMOPLAN program is effective for quality testing data and data interpretation.

Some field data examples are in Figures 1.6 and 1.7. These results are not surprising. A dominant local plane is removed, leaving noise or the second from strongest local plane. These data sets fit the local plane model so well that subtracting the residual noise from the data made little improvement. These Figures are clearer on a video screen. To facilitate examination of the residual on Figure 1.6 on paper (which has a lesser dynamic range than video) I recolored the white residual with a short triangle filter on the time axis.

The residual in Figure 1.7 is large at the dead trace and wherever the data contains crossing events. Also, closer examination showed that the strong residual trace near 1.1 km offset is apparently slightly time shifted, almost certainly a cable problem, perhaps resulting from a combination of the stepout and a few dead pickups. Overall, the local-plane-residual shows a low frequency water velocity wave seeming to originate from the ship.

1.4 GRADIENT ALONG THE BEDDING PLANE

The LOMOPLAN (LOcal MOnoPLane ANnihilator) filter in three dimensions is a deconvolution filter that takes a volume in and produces two volumes out. The x-output volume results from a first order prediction-error filter on the x-axis, and the y-output volume is likewise on the y-axis.

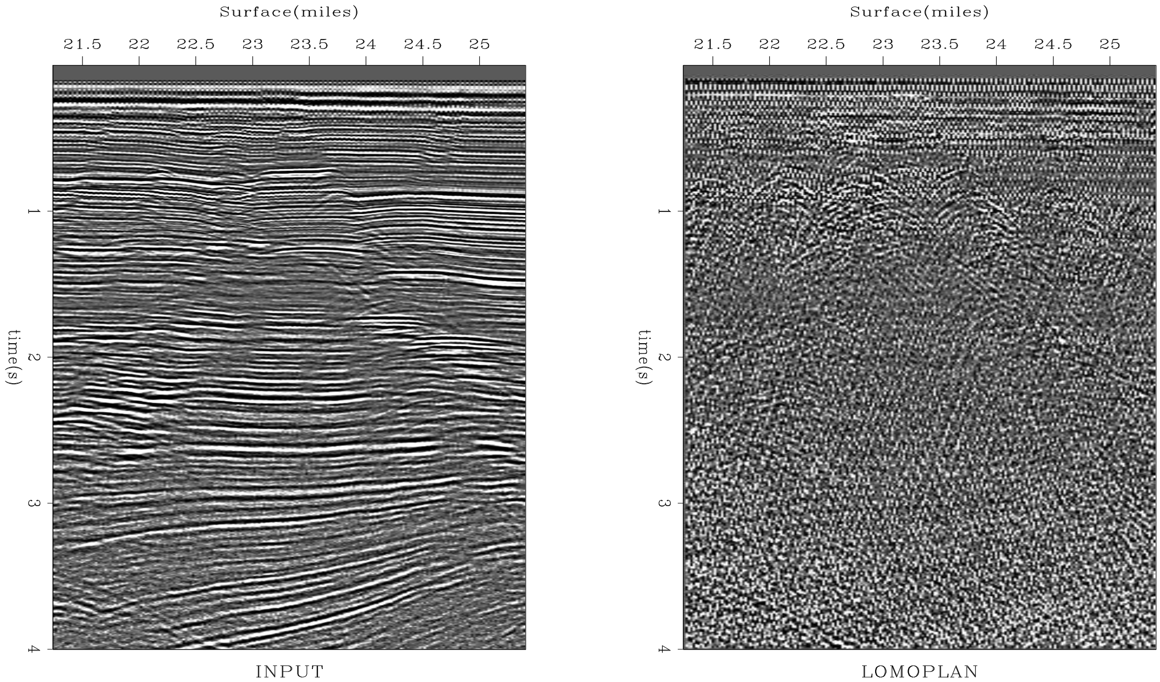

Figure 1.6: Data section from the Gulf of Mexico (left) and after LOMOPLAN (right) Press button for movie. lmn-dgulf [NR]

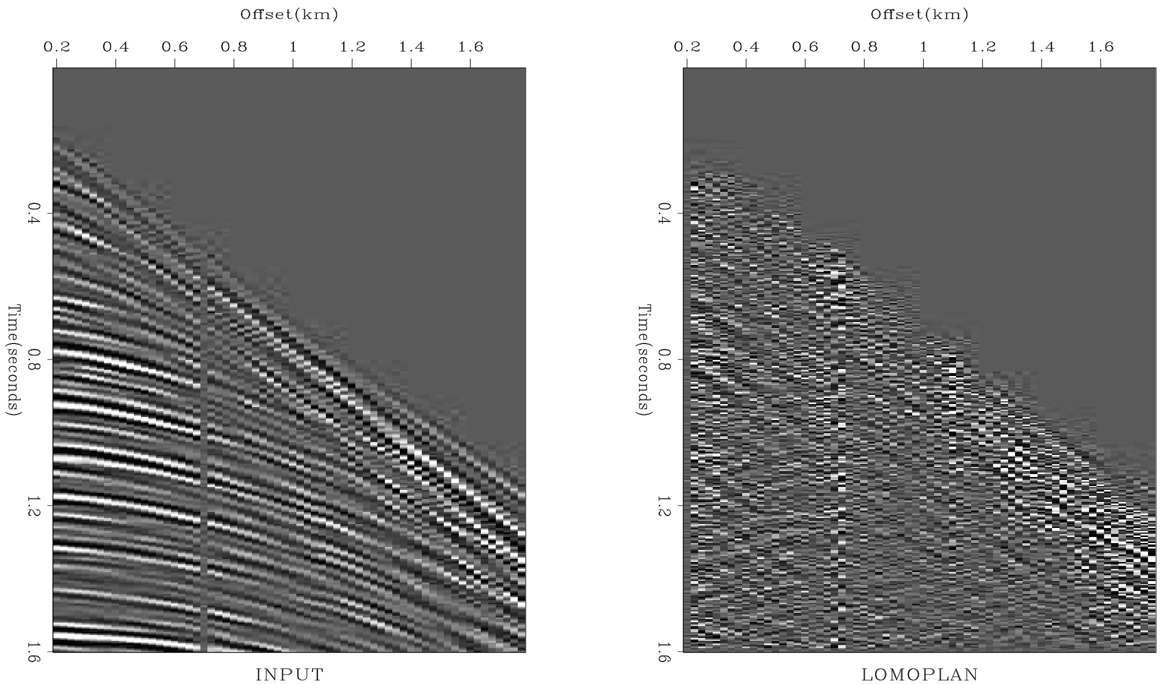

Figure 1.7: Portion of Yilmaz and Cumro data set 27 (left) and after LOMOPLAN (right). Press button for movie. lmn-yc27 [NR]

Although I conceived of 2-D LOMOPLAN as the "ultimate" optimization criterion for inversion problems in reflection seismology of sedimentary section, it turned out that it was more useful in data interpretation and in data quality inspection. In this study, I sought to evaluate usefulness with *three*-dimensional data such as 3-D stacks or migrated volumes, or 2-D prestack data.

In experimenting with 3-D LOMOPLAN, I came upon a conceptual oversimplification, which although it is not precisely correct, gives a suitable feeling of the meaning of the operator. Imagine the earth was flat horizontal layers, except for occasional faults. Then, to find the faults you might invoke the horizontal gradient of the 3-D continuum of data. The horizontal components of gradient vanish except at a fault, where their relative magnitudes tell you the orientation of the fault. Instead of using the gradient vector, you could use prediction-error filters of first order (two components) along x and y directions. 3-D LOMOPLAN is like this, but the flat horizontal **bedding** may be dipping or curved. No output is produced (ideally) except at faults. The 3-D LOMOPLAN is like the gradient *along the plane of the bedding.* It is nonzero where the bedding has an intrinsic change.

LOMOPLAN flags the bedding where there is an intrinsic change.

1.4.1 LOMOPLAN definition in 3-D

Three-dimensional LOMOPLAN is somewhat like multiple passes of two-dimensional LOMOPLAN, i.e., we first LOMOPLAN the (t, x)-plane for each y, and then we LOMOPLAN the (t, y)-plane for each x. Actually, 3-D LOMOPLAN is a little more complicated than this. Each LOMOPLAN filter is designed on all the data in a small (t, x, y) volume.

I began from my earlier two-dimensional code and made the obvious extensions to three dimensions. For example, converting subroutine `icaf2()` (page ??) to the 3-D version `icaf3()` gives a subroutine that convolves a volume over a volume to get a volume. For a 2-D filter we need only pick `a3=1` and with it, subroutine `icaf3()` convolves the planar filter throughout the input volume giving an output volume. For a monoplane filter orthogonally oriented, we take `a2=1` but `a3=2`. With the adjoint being the filter, this crosscorrelates two volumes with the output being a plane, so the averaging is done in cubes and *that* is what makes 3-D LOMOPLAN different from two orthogonal passes of 2-D LOMOPLAN.

To put the LOcal in LOMOPLAN we use subcubes (bricks). Recall we can do 2-D LOMOPLAN with the prediction-error subroutine `lopef2()` (page ??). For 3-D LOMOPLAN we need to make two calls to subroutine `lopef3()`, one for the x-axis in-line planar filters and one for the y-axis crossline filters. That is what I will try next time I install this book on a computer with a bigger memory.

1.4.2 The quarterdome 3-D synthetic (qdome)

Figure 1.4 used a model called "**Sigmoid**". Using the same modeling concepts, I set out to make a three-dimensional model. The model has horizontal layers near the top, a Gaussian appearance in the middle, and dipping layers on the bottom with horizontal unconformities between the three regions. Figure 1.8 shows a vertical slice through the 3-D "qdome" model and components of its LOMOPLAN. There is also a fault that will be described later. The most interesting part of the qdome model

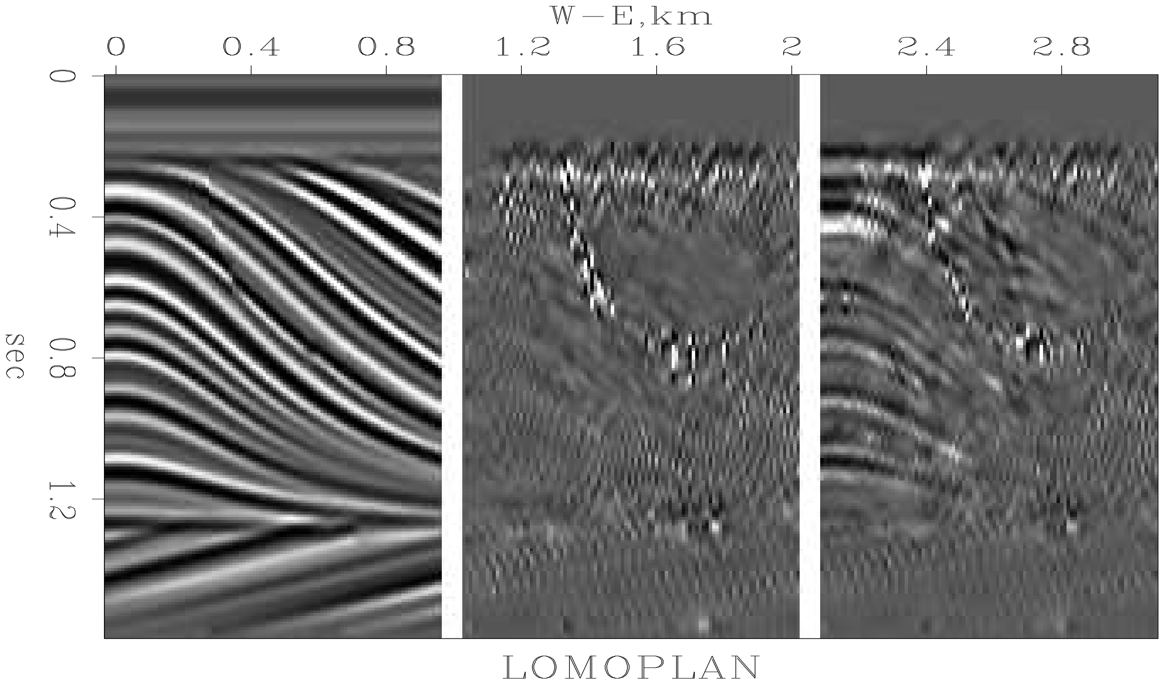

Figure 1.8: Left is a vertical slice through the 3-D "qdome" model. Center is the in-line component of the LOMOPLAN. Right is the cross-line component of the LOMOPLAN. lmn-qdomesico [NR]

is the Gaussian center. I started from the equation of a **Gaussian**

$$z(x,y,t) \quad = \quad e^{-(x^2+y^2)/t^2} \tag{1.3}$$

and backsolved for t

$$t(x,y,z) \quad = \quad \sqrt{\frac{x^2+y^2}{-\ln z}} \tag{1.4}$$

Then I used a random number generator to make a blocky one-dimensional impedance function of t. At each (x,y,z) location in the model I used the impedance at time $t(x,y,z)$, and finally defined reflectivity as the logarithmic derivative of the impedance. Without careful interpolation (particularly where the beds pinch out) a variety of curious artifacts appear. I hope to find time to make a tutorial lesson on interpolation from the experience of making the qdome model. As a refinement to the model, within a certain subvolume the time $t(x,y,z)$ is given a small additive constant. This gives a fault along the edge of the subvolume. Ray **Abma** defined the subvolume for me in the qdome model. The fault looks quite realistic, and it is easy to make faults of any shape, though I wonder how they would relate to realistic fault dynamics. Figure ?? shows a top view of the 3-D qdome model and components of its LOMOPLAN. Notice the cross-line spacing has been chosen to be double the in-line spacing. Evidently a consequence of this in both Figure 1.8 and Figure ?? is that the Gaussian dome is not so well suppressed on the crossline cut as on the in-line cut. By comparison, notice that the horizontal bedding above the dome is perfectly suppressed, whereas the dipping bedding below the dome is imperfectly suppressed.

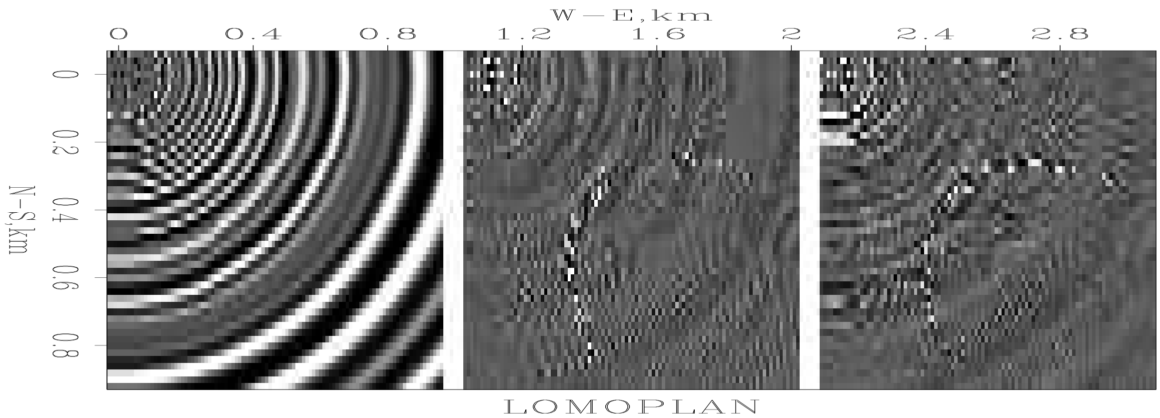

Figure 1.9: Left is a horizontal slice through the 3-D qdome model. Center is the in-line component of the LOMOPLAN. Right is the cross-line component of the LOMOPLAN. Press button for volume view. lmn-qdometoco [NR]

Finally, I became irritated at the need to look at *two* output volumes. Since I rarely if ever interpreted the polarity of the LOMOPLAN components, I formed their sum of squares and show the square root in Figure ??.

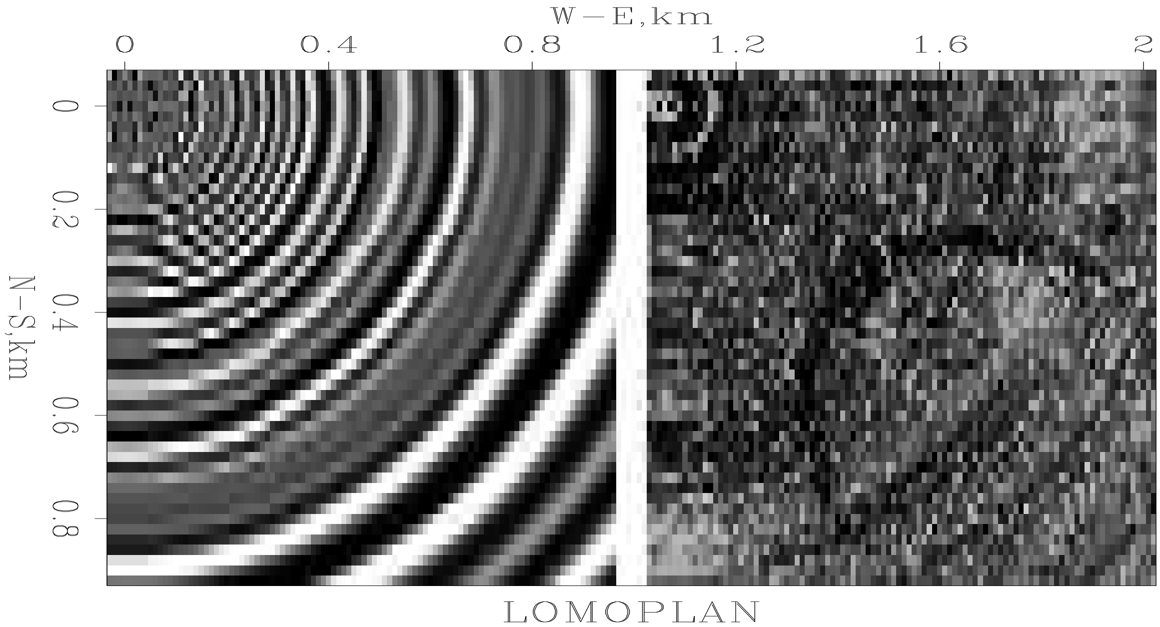

Figure 1.10: Left is the model. Right is the magnitude of the LOMOPLAN components in Figure ??. Press button for volume view. lmn-qdometora [NR]