Chapter 1

Dip and offset together

[1]When dip and offset are combined, some serious complications arise. For many years it was common industry practice to ignore these complications and to handle dip and offset separately. Thus offset was handled by velocity analysis, normal moveout and stack (chapter **??**). And dip was handled by zero-offset migration after stack (chapters **??** and **??**). This practice is a good approximation only when the dips on the section are small. We need to handle large offset angles at the same time we handle large dip angles at the same time we are estimating rock velocity. It is confusing! Here we see the important steps of bootstrapping yourself towards both the velocity and the image.

1.1 PRESTACK MIGRATION

Prestack migration creates an image of the earth's reflectivity directly from prestack data. It is an alternative to the "**exploding reflector**" concept that proved so useful in zero-offset migration. In **prestack migration**, we consider both downgoing and upcoming waves.

A good starting point for discussing prestack migration is a reflecting point within the earth. A wave incident on the point from any direction reflects waves in all directions. This geometry is particularly important because any model is a superposition of such point scatterers. The point-scatterer geometry for a point located at (x,z) is shown in Figure 1.1. The equation for travel time t is the sum of the two travel paths is

$$t\,v \quad = \quad \sqrt{z^2 + (s-x)^2} + \sqrt{z^2 + (g-x)^2} \tag{1.1}$$

We could model field data with equation (1.1) by copying reflections from any point in (x,z)-space into (s,g,t)-space. The adjoint program would form an image stacked over all offsets. This process would be called prestack migration. The problem here is that the real problem is estimating velocity. In this chapter we will see that it is not satisfactory to use a horizontal layer

[1]Matt Schwab prepared a draft of the Gardner DMO derivation. Shuki Ronen gave me the "law of cosines" proof.

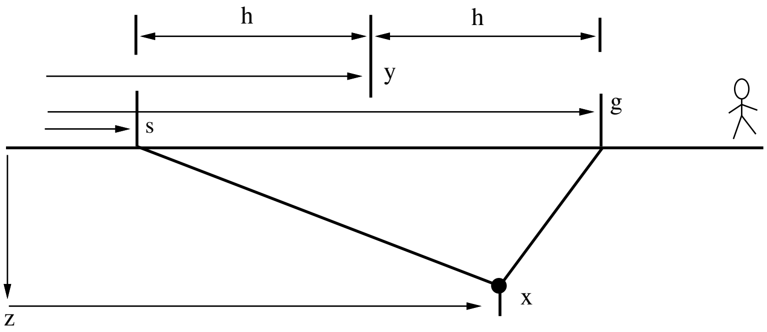

Figure 1.1: Geometry of a point scatterer. dpmv-pgeometry [NR]

approximation to estimate velocity, and then use equation (1.1) to do migration. Migration becomes sensitive to velocity when wide angles are involved. Errors in the velocity would spoil whatever benefit could accrue from prestack (instead of poststack) migration.

1.1.1 Cheops' pyramid

Because of the importance of the point-scatterer model, we will go to considerable lengths to visualize the functional dependence among t, z, x, s, and g in equation (1.1). This picture is more difficult—by one dimension—than is the conic section of the exploding-reflector geometry.

To begin with, suppose that the first square root in (1.1) is constant because everything in it is held constant. This leaves the familiar hyperbola in (g,t)-space, except that a constant has been added to the time. Suppose instead that the other square root is constant. This likewise leaves a hyperbola in (s,t)-space. In (s,g)-space, travel time is a function of s plus a function of g. I think of this as one coat hanger, which is parallel to the s-axis, being hung from another coat hanger, which is parallel to the g-axis.

A view of the traveltime pyramid on the (s,g)-plane or the (y,h)-plane is shown in Figure 1.2a. Notice that a cut through the pyramid at large t is a square, the corners of which have been smoothed. At very large t, a constant value of t is the square contoured in (s,g)-space, as in Figure 1.2b. Algebraically, the squareness becomes evident for a point reflector near the surface, say, $z \to 0$. Then (1.1) becomes

$$vt \quad = \quad |s - x| \; + \; |g - x| \tag{1.2}$$

The center of the square is located at $(s,g) = (x,x)$. Taking travel time t to increase downward from the horizontal plane of (s,g)-space, the square contour is like a horizontal slice through the Egyptian pyramid of Cheops. To walk around the pyramid at a constant altitude is to walk around a square. Alternately, the altitude change of a traverse over g (or s) at constant s (or g) is simply a constant plus an absolute-value function.

More interesting and less obvious are the curves on common-midpoint gathers and constant-offset sections. Recall the definition that the midpoint between the shot and geophone is y.

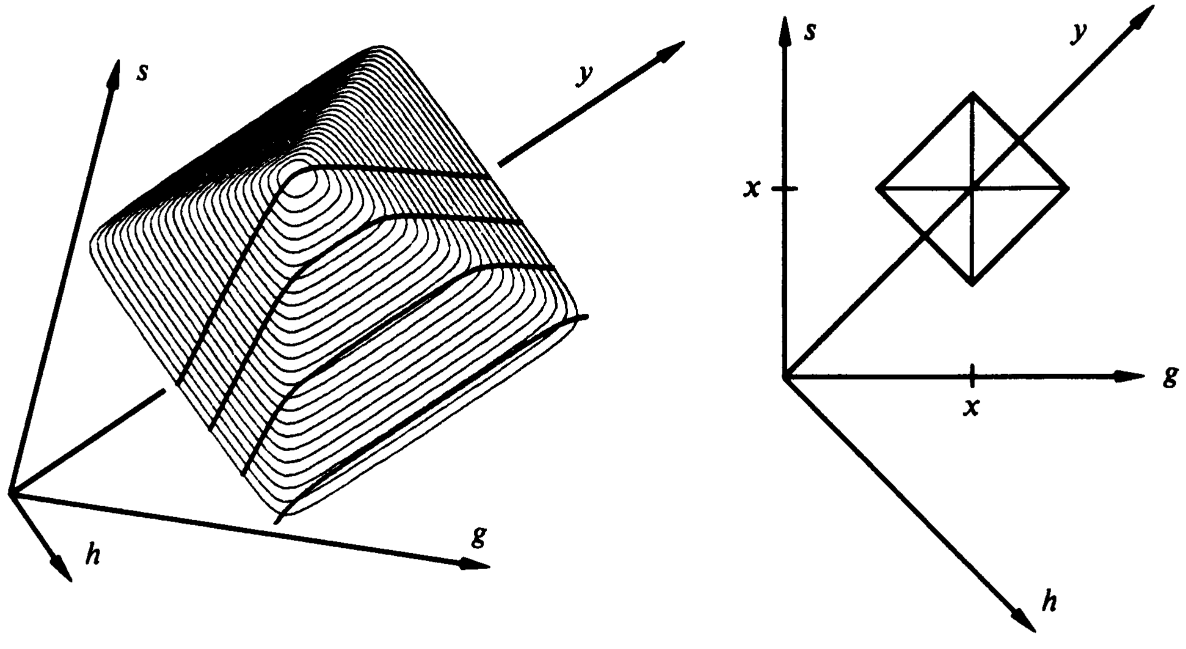

Figure 1.2: Left is a picture of the traveltime pyramid of equation ((1.1)) for fixed x and z. The darkened lines are constant-offset sections. Right is a cross section through the pyramid for large t (or small z). (Ottolini) dpmv-cheop [NR]

Also recall that h is half the horizontal offset from the shot to the geophone.

$$y = \frac{g + s}{2} \tag{1.3}$$

$$h = \frac{g - s}{2} \tag{1.4}$$

A traverse of y at constant h is shown in Figure 1.2. At the highest elevation on the traverse, you are walking along a flat horizontal step like the flat-topped hyperboloids of Figure 1.8. Some erosion to smooth the top and edges of the pyramid gives a model for nonzero reflector depth.

For rays that are near the vertical, the traveltime curves are far from the hyperbola asymptotes. Then the square roots in (1.1) may be expanded in Taylor series, giving a parabola of revolution. This describes the eroded peak of the pyramid.

1.1.2 Prestack migration ellipse

Equation (1.1) in (y,h)-space is

$$t\,v = \sqrt{z^2 + (y - y_0 - h)^2} + \sqrt{z^2 + (y - y_0 + h)^2} \tag{1.5}$$

A basic insight into equation (1.1) is to notice that at constant-offset h and constant travel time t the locus of possible reflectors is an ellipse in the (y,z)-plane centered at y_0. The reason it is an **ellipse** follows from the geometric definition of an ellipse. To draw an ellipse, place a nail or tack into s on Figure 1.1 and another into g. Connect the tacks by a string that is exactly long enough to go through (y_0, z). An ellipse going through (y_0, z) may be constructed by sliding a pencil along the string, keeping the string tight. The string keeps the total distance tv constant as is shown in Figure 1.3

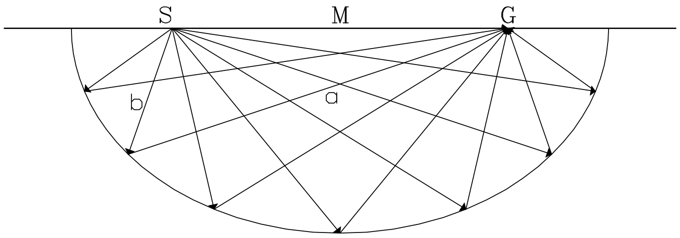

Figure 1.3: Prestack migration ellipse, the locus of all scatterers with constant traveltime for source S and receiver G. dpmv-ellipse1 [ER,M]

Replacing depth z in equation (1.5) by the vertical traveltime depth $\tau = 2z/v = z/v_{\text{half}}$ we get

$$t \quad = \quad \frac{1}{2}\left(\sqrt{\tau^2 \ + \ [(y-y_0)-h]^2/v_{\text{half}}^2} + \sqrt{\tau^2 \ + \ [(y-y_0)+h]^2/v_{\text{half}}^2}\right) \tag{1.6}$$

1.1.3 Constant offset migration

Considering h in equation (1.6) to be a constant, enables us to write a subroutine for migrating constant-offset sections. Subroutine `flathyp()` on this page is easily prepared from subroutine `kirchfast()` on page ?? by replacing its hyperbola equation with equation (1.6).

```
# Flat topped hyperbolas and constant-offset section migration
#
subroutine flathyp(   adj, add, vel    , h, t0,dt,dx, modl,nt,nx, data)
integer ix,iz,it,ib,  adj, add,                            nt,nx
real    t, amp, z,b,             vel(nt), h, t0,dt,dx, modl(nt,nx),data(nt,nx)
call adjnull(       adj, add,                        modl,nt*nx, data,nt*nx)
do ib= -nx, nx {        b = dx * ib             # b = midpt separation y-y0
    do iz= 2, nt {      z = t0 + dt * (iz-1)    # z = zero-offset time
        t = .5 * ( sqrt( z**2 +((b-h)*2/vel(iz))**2) +
                   sqrt( z**2 +((b+h)*2/vel(iz))**2)   )
        it = 1.5 + (t - t0) / dt
     if( it > nt )                      break
        amp = (z/t)/ sqrt(t)
        do ix= max0(1, 1-ib),  min0(nx, nx-ib)
                 if( adj == 0 )
                        data(it,ix+ib)= data(it,ix+ib) + modl(iz,ix   ) * amp
                 else
                        modl(iz,ix   )= modl(iz,ix   ) + data(it,ix+ib) * amp
        }
    }
return; end
```

The amplitude in subroutine `flathyp()` should be improved when we have time to do so. Forward and backward responses to impulses of subroutine `flathyp()` are found in Figures 1.4 and 1.5.

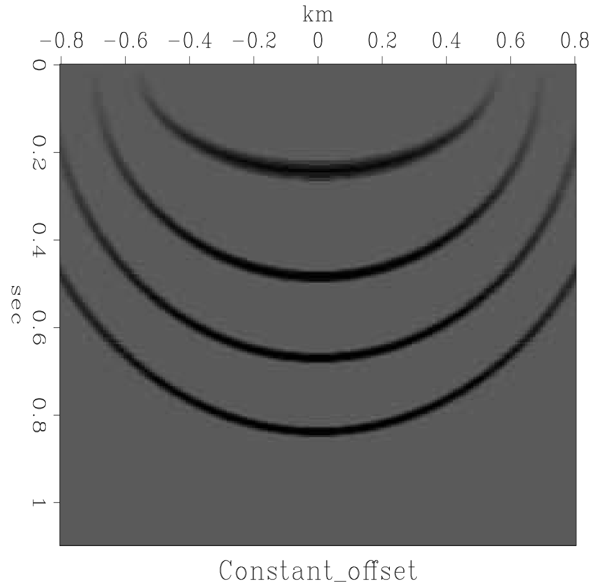

Figure 1.4: Migrating impulses on a constant-offset section with subroutine `flathyp()`. Notice that shallow impulses (shallow compared to h) appear ellipsoidal while deep ones appear circular. dpmv-Cos.1 [ER]

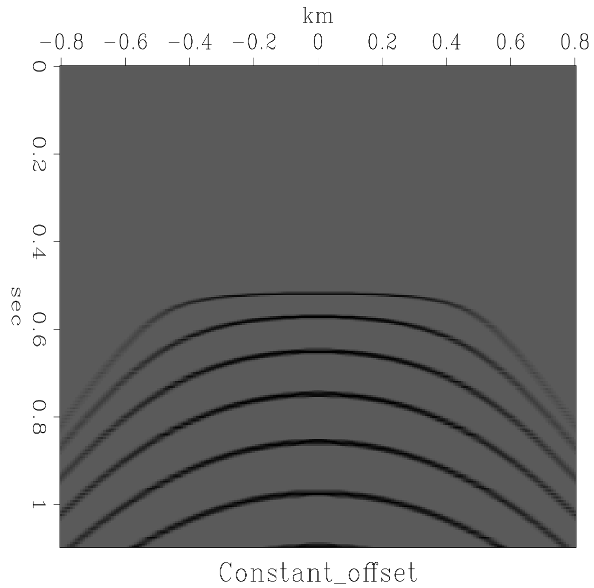

Figure 1.5: Forward modeling from an earth impulse with subroutine `flathyp()`. dpmv-Cos.0 [ER]

It is not easy to show that equation (1.5) can be cast in the standard mathematical form of an ellipse, namely, a stretched circle. But the result is a simple one, and an important one for later analysis. Feel free to skip forward over the following verification of this ancient wisdom. To help reduce algebraic verbosity, define a new y equal to the old one shifted by y_0. Also make the definitions

$$\begin{aligned} t\,v &= 2\,A \\ \alpha &= z^2 + (y+h)^2 \end{aligned} \tag{1.7}$$

$$
\begin{aligned}
\beta &= z^2 + (y-h)^2 \\
\alpha - \beta &= 4\, y\, h
\end{aligned}
$$

With these definitions, (1.5) becomes

$$
2\, A = \sqrt{\alpha} + \sqrt{\beta}
$$

Square to get a new equation with only one square root.

$$
4\, A^2 - (\alpha + \beta) = 2\sqrt{\alpha\beta}
$$

Square again to eliminate the square root.

$$
\begin{aligned}
16\, A^4 - 8\, A^2 (\alpha + \beta) + (\alpha + \beta)^2 &= 4\, \alpha\, \beta \\
16\, A^4 - 8\, A^2 (\alpha + \beta) + (\alpha - \beta)^2 &= 0
\end{aligned}
$$

Introduce definitions of α and β.

$$
16\, A^4 - 8\, A^2 \left[2z^2 + 2y^2 + 2h^2\right] + 16\, y^2 h^2 = 0
$$

Bring y and z to the right.

$$
\begin{aligned}
A^4 - A^2 h^2 &= A^2 (z^2 + y^2) - y^2 h^2 \\
A^2 (A^2 - h^2) &= A^2 z^2 + (A^2 - h^2)\, y^2 \\
A^2 &= \frac{z^2}{1 - \frac{h^2}{A^2}} + y^2 \qquad (1.8)
\end{aligned}
$$

Finally, recalling all earlier definitions and replacing y by $y - y_0$, we obtain the canonical form of an ellipse with semi-major axis A and semi-minor axis B:

$$
\frac{(y - y_0)^2}{A^2} + \frac{z^2}{B^2} = 1 \ , \qquad (1.9)
$$

where

$$
\begin{aligned}
A &= \frac{v\, t}{2} \qquad (1.10) \\
B &= \sqrt{A^2 - h^2} \qquad (1.11)
\end{aligned}
$$

Fixing t, equation (1.9) is the equation for a circle with a stretched z-axis. The above algebra confirms that the "string and tack" definition of an **ellipse** matches the "stretched circle" definition. An **ellipse** in earth model space corresponds to an impulse on a constant-offset section.

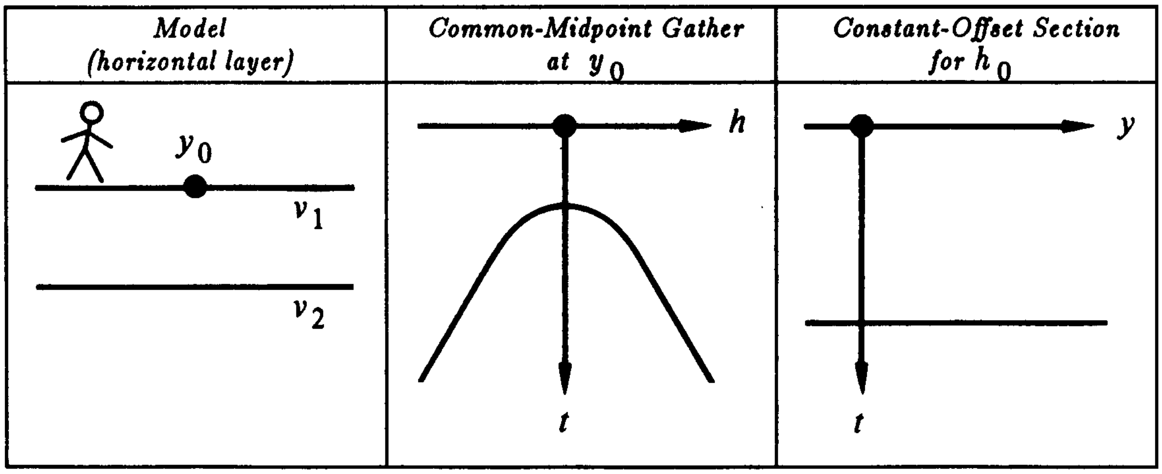

Figure 1.6: Simplest earth model. dpmv-simple [NR]

1.2 INTRODUCTION TO DIP

We can consider a data space to be a superposition of points and then analyze the point response, or we can consider data space or model space to be a superposition of planes and then do an analysis of planes. Analysis of points is often easier than planes, but planes, particularly local planes, are more like observational data and earth models.

The simplest environment for reflection data is a single horizontal reflection interface, which is shown in Figure 1.6. As expected, the zero-offset section mimics the earth model. The common-midpoint gather is a hyperbola whose asymptotes are straight lines with slopes of the inverse of the velocity v_1. The most basic data processing is called ***common-depth-point stack*** or **CDP stack**. In it, all the traces on the common-midpoint (CMP) gather are time shifted into alignment and then added together. The result mimics a zero-offset trace. The collection of all such traces is called the *CDP-stacked section.* In practice the CDP-stacked section is often interpreted and migrated as though it were a zero-offset section. In this chapter we will learn to avoid this popular, oversimplified assumption.

The next simplest environment is to have a planar reflector that is oriented vertically rather than horizontally. This might not seem typical, but the essential feature (that the rays run horizontally) really is common in practice (see for example Figure 1.9.) Also, the effect of dip, while generally complicated, becomes particularly simple in the extreme case. If you wish to avoid thinking of wave propagation along the air-earth interface you can take the reflector to be inclined a slight angle from the vertical, as in Figure 1.7.

Figure 1.7 shows that the travel time does not change as the offset changes. It may seem paradoxical that the travel time does not increase as the shot and geophone get further apart. The key to the paradox is that midpoint is held constant, not shotpoint. As offset increases, the shot gets further from the reflector while the geophone gets closer. Time lost on one path is gained on the other.

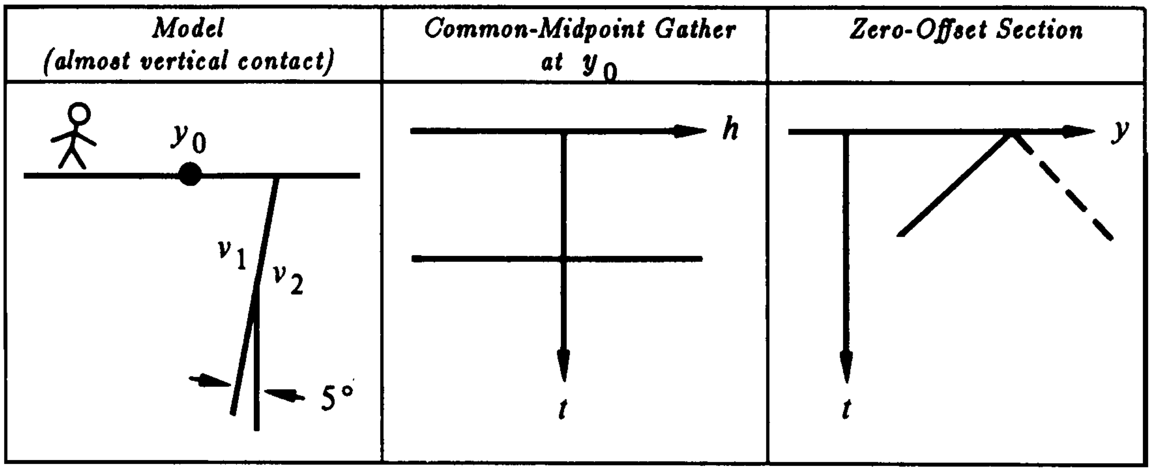

Figure 1.7: Near-vertical reflector, a gather, and a section. dpmv-vertlay [NR]

A planar reflector may have any dip between horizontal and vertical. Then the common-midpoint gather lies between the common-midpoint gather of Figure 1.6 and that of Figure 1.7. The zero-offset section in Figure 1.7 is a straight line, which turns out to be the asymptote of a family of hyperbolas. The slope of the asymptote is the inverse of the velocity v_1.

It is interesting to notice that at small dips, information about the earth velocity is essentially carried on the offset axis whereas at large dips, the velocity information is essentially on the midpoint axis.

1.2.1 The response of two points

Another simple geometry is a reflecting point within the earth. A wave incident on the point from any direction reflects waves in all directions. This geometry is particularly important because any model is a superposition of such point scatterers. Figure 1.8 shows an example. The curves in Figure 1.8 include flat spots for the same reasons that some of the curves in Figures 1.6 and 1.7 were straight lines.

1.2.2 The dipping bed

While the traveltime curves resulting from a dipping bed are simple, they are not simple to derive. Before the derivation, the result will be stated: for a bed dipping at angle α from the horizontal, the traveltime curve is

$$t^2 v^2 \quad = \quad 4(y-y_0)^2 \sin^2\alpha \; + \; 4h^2\cos^2\alpha \tag{1.12}$$

For $\alpha = 45°$, equation (1.12) is the familiar Pythagoras cone—it is just like $t^2 = z^2 + x^2$. For other values of α, the equation is still a cone, but a less familiar one because of the stretched axes.

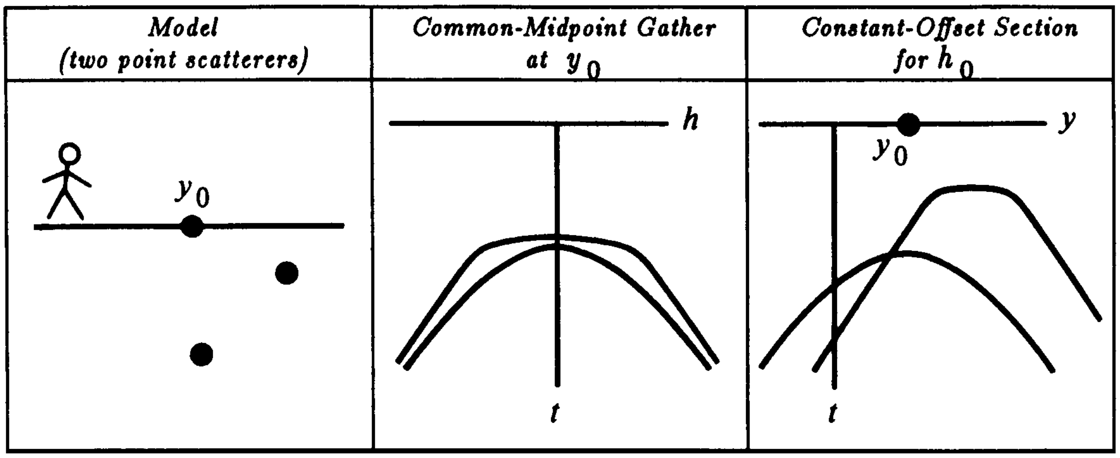

Figure 1.8: Response of two point scatterers. Note the flat spots. dpmv-twopoint [NR]

For a common-midpoint gather at $y = y_1$ in (h,t)-space, equation (1.12) looks like $t^2 = t_0^2 + 4h^2/v_{\text{apparent}}^2$. Thus the common-midpoint gather contains an *exact* hyperbola, regardless of the earth dip angle α. The effect of dip is to change the asymptote of the hyperbola, thus changing the apparent velocity. The result has great significance in applied work and is known as Levin's dip correction [1971]:

$$v_{\text{apparent}} \quad = \quad \frac{v_{\text{earth}}}{\cos(\alpha)} \tag{1.13}$$

(See also Slotnick [1959]). In summary, dip increases the stacking velocity.

Figure 1.10 depicts some rays from a common-midpoint gather. Notice that each ray strikes the dipping bed at a different place. So a common-*midpoint* gather is not a common-*depth-point* gather. To realize why the reflection point moves on the reflector, recall the basic geometrical fact that an angle bisector in a triangle generally doesn't bisect the opposite side. The reflection point moves *up* dip with increasing offset.

Finally, equation (1.12) will be proved. Figure 1.11 shows the basic geometry along with an "image" source on another reflector of twice the dip. For convenience, the bed intercepts the surface at $y_0 = 0$. The length of the line $s'g$ in Figure 1.11 is determined by the trigonometric Law of Cosines to be

$$\begin{aligned}
t^2\,v^2 \quad &= \quad s^2 \,+\, g^2 \,-\, 2s\,g\cos 2\alpha \\
t^2\,v^2 \quad &= \quad (y-h)^2 \,+\, (y+h)^2 \,-\, 2(y-h)(y+h)\cos 2\alpha \\
t^2\,v^2 \quad &= \quad 2(y^2+h^2) \,-\, 2(y^2-h^2)(\cos^2\alpha - \sin^2\alpha) \\
t^2\,v^2 \quad &= \quad 4\,y^2\sin^2\alpha \,+\, 4h^2\cos^2\alpha
\end{aligned}$$

which is equation (1.12).

Another facet of equation (1.12) is that it describes the constant-offset section. Surprisingly, the travel time of a dipping planar bed becomes curved at nonzero offset—it too becomes hyperbolic.

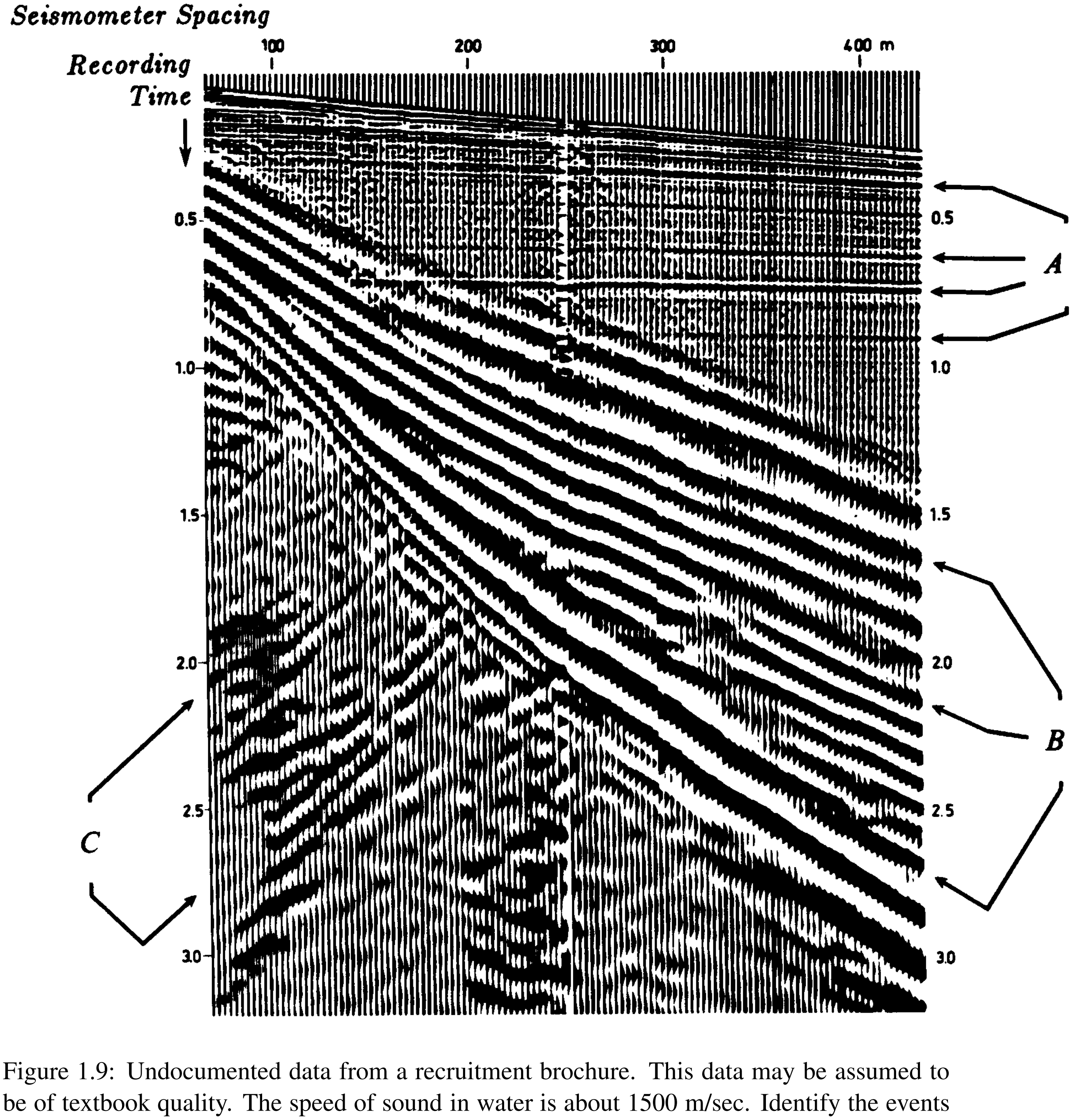

Figure 1.9: Undocumented data from a recruitment brochure. This data may be assumed to be of textbook quality. The speed of sound in water is about 1500 m/sec. Identify the events at A, B, and C. Is this a common-shotpoint gather or a common-midpoint gather? (Shell Oil Company) dpmv-shell [NR]

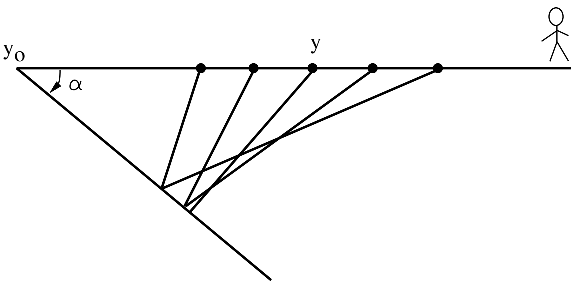

Figure 1.10: Rays from a common-midpoint gather. dpmv-dipray [NR]

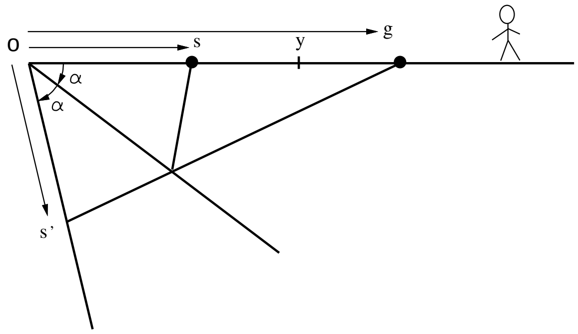

Figure 1.11: Travel time from image source at s' to g may be expressed by the law of cosines. dpmv-lawcos [NR]

1.2.3 Randomly dipping layers

On a horizontally layered earth, a common shotpoint gather looks like a common midpoint gather. For an earth model of random dipping planes the two kinds of gathers have quite different traveltime curves as we see in Figure 1.12.

The common-shot gather is more easily understood. Although a reflector is dipping, a spherical wave incident remains a spherical wave after reflection. The center of the reflected wave sphere is called the image point. The traveltime equation is again a cone centered at the image point. The traveltime curves are simply hyperbolas topped above the image point having the usual asymptotic slope. The new feature introduced by dip is that the hyperbola is laterally shifted which implies arrivals before the fastest possible straight-line arrivals at $vt = |g|$. Such arrivals cannot happen. These hyperbolas must be truncated where $vt = |g|$. This discontinuity has the physical meaning of a dipping bed hitting the surface at geophone location $|g| = vt$. Beyond the truncation, either the shot or the receiver has gone beyond the intersection. Eventually both are beyond. When either is beyond the intersection, there are no reflections.

On the common-midpoint gather we see hyperbolas all topping at zero offset, but with asymptotic velocities higher (by the Levin cosine of dip) than the earth velocity. Hyperbolas truncate, now at $|h| = tv/2$, again where a dipping bed hits the surface at a geophone.

On a CMP gather, some hyperbolas may seem high velocity, but it is the dip, not the earth velocity itself that causes it. Imagine Figure 1.12 with all layers at 90° dip (abandon curves and keep straight lines). Such dip is like the backscattered groundroll seen on the common-shot gather of Figure 1.9. The backscattered groundroll becomes a “flat top” on the CMP gather in Figure 1.12.

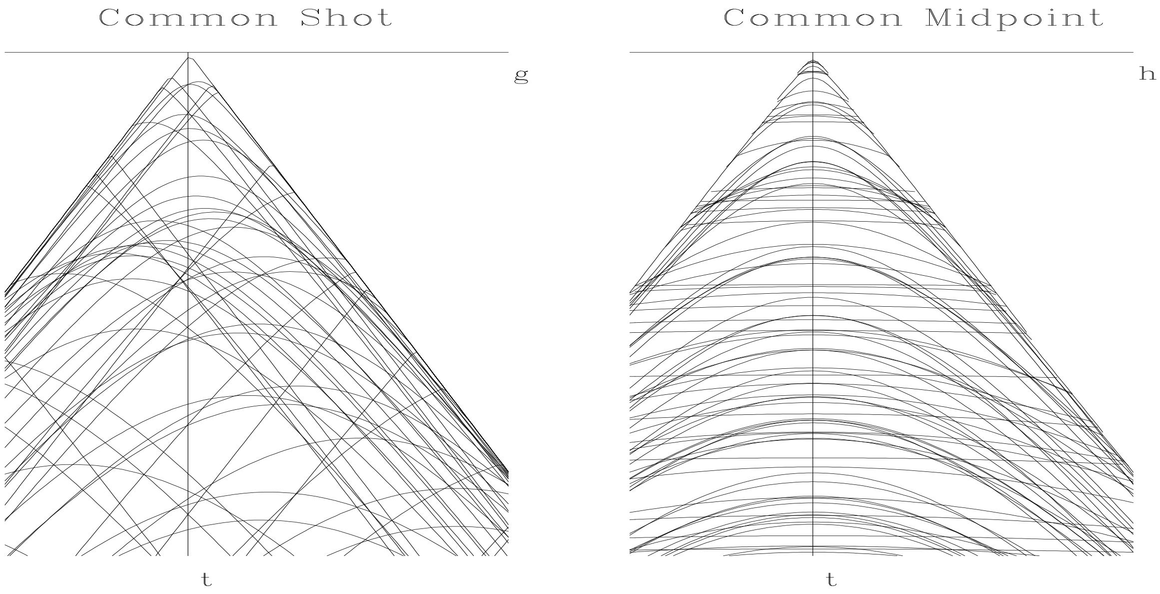

Figure 1.12: Seismic arrival times on an earth of random dipping planes. Left is for CSP. Right is for CMP. dpmv-randip [ER]

Such strong horizontal events near zero offset will match any velocity, particularly higher velocities such as primaries. Unfortunately such noise events thus make a strong contribution to a CMP stack. Let us see how these flat-tops in offset create the diagonal streaks you see in midpoint in Figure 1.13.

Consider 360 rocks of random sizes scattered in an exact circle of 2 km diameter on the ocean floor. The rocks are distributed along one degree intervals. Our survey ship sails from south to north towing a streamer across the exact center of the circle, coincidentally crossing directly over rock number 180 and number 0. Let us consider the common midpoint gather corresponding to the midpoint in the center of the circle. Rocks 0 and 180 produce flat-top hyperbolas. The top is flat for $0 < |h| < 1$ km. Rocks 90 and 270 are 90° out of the plane of the survey. Rays to those rocks propagate entirely within the water layer. Since this is a homogeneous media, the travel time expression of these rocks is a simple hyperbola of water velocity. Now our CMP gather at the circle center has a "flat top" and a simple hyperbola both going through zero offset at time $t = 2/v$ (diameter 2 km, water velocity). Both curves have the same water velocity asymptote and of course the curves are tangent at zero offset.

Now consider all the other rocks. They give curves inbetween the simple water hyperbola and the flat top. Near zero offset, these curves range in apparant velocity between water velocity and infinity. One of these curves will have an apparant velocity that matches that of sediment velocity. This rock (and all those near the same azimuth) will have velocities that are near the sediment velocity. This noise will stack very well. The CDP stack at sediment velocity will stack in a lot of water borne noise. This noise is propagating somewhat off the survey line but not very far off it.

Now let us think about the appearance of the CDP stack. We turn attention from offset

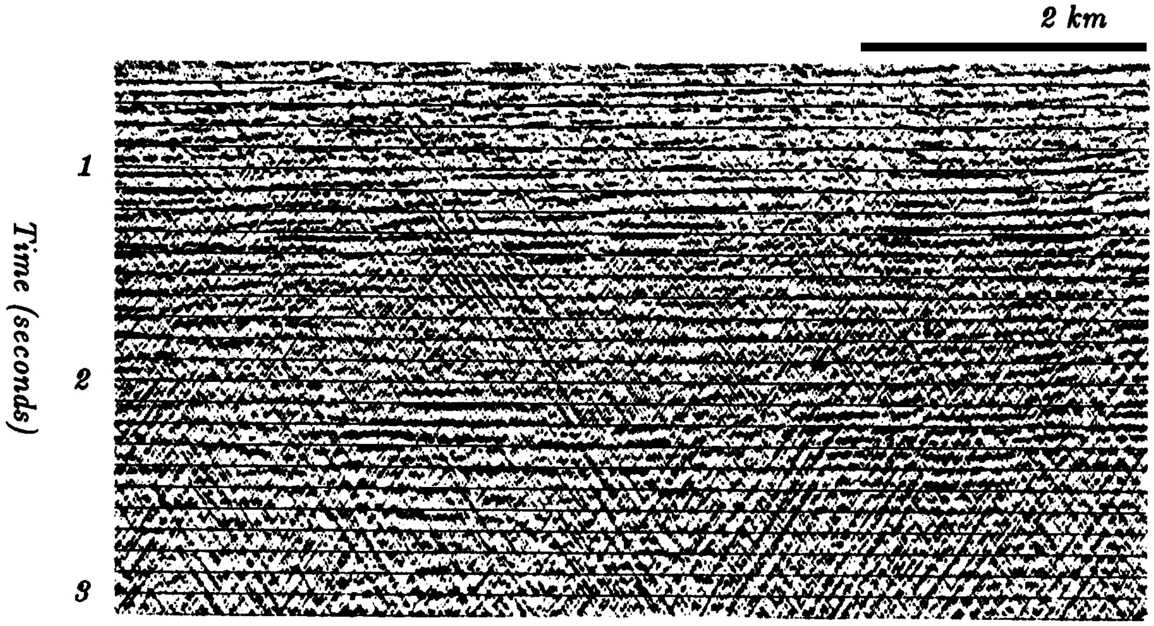

Figure 1.13: CDP stack with water noise from the Shelikof Strait, Alaska. (by permission from *Geophysics,* Larner et al.[1983]) dpmv-shelikof [NR]

to midpoint. The easiest way to imagine the CDP stack is to imagine the zero-offset section. Every rock has a water velocity asymptote. These asymptotes are evident on the CDP stack in Figure 1.13. This result was first recognized by Ken Larner.

Thus, backscattered low-velocity noises have a way of showing up on higher-velocity stacked data. There are two approaches to suppressing this noise: (1) mute the inner traces, and as we will see, (2) dip moveout processing.

1.3 TROUBLE WITH DIPPING REFLECTORS

The "standard process" is NMO, stack, and zero-offset migration. Its major shortcoming is the failure of NMO and stack to produce a section that resembles the true zero-offset section. In chapter **??** we derived the NMO equations for a stratified earth, but then applied them to seismic field data that was not really stratified. That this works at all is a little surprising, but it turns out that NMO hyperbolas apply to dipping reflectors as well as horizontal ones. When people try to put this result into practice, however, they run into a nasty conflict: reflectors generally require a *dip-dependent* NMO velocity in order to produce a "good" stack. Which NMO velocity are we to apply when a dipping event is near (or even crosses) a horizontal event? Using conventional NMO/stack techniques generally forces velocity analysts to choose which events they wish to preserve on the stack. This inability to simultaneously produce a good stack for events with *all* dips is a serious shortcoming, which we now wish to understand more quantitatively.

1.3.1 Gulf of Mexico example

Recall the Gulf of Mexico dataset presented in chapter **??**. We did a reasonably careful job of NMO velocity analysis in order to produce the stack shown in Figure **??**. But is this the best possible stack? To begin to answer this question, Figure 1.14 shows some constant-velocity stacks of this dataset done with subroutine `velsimp()` on page ??. This figure clearly shows that there are some very steeply-dipping reflections that are missing in Figure **??**. These steep reflections appear only when the NMO velocity is quite high compared with the velocity that does a good job on the horizontal reflectors. This phenomenon is consistent with the predictions of equation (1.12), which says that dipping events require a *higher* NMO velocity than nearby horizontal events.

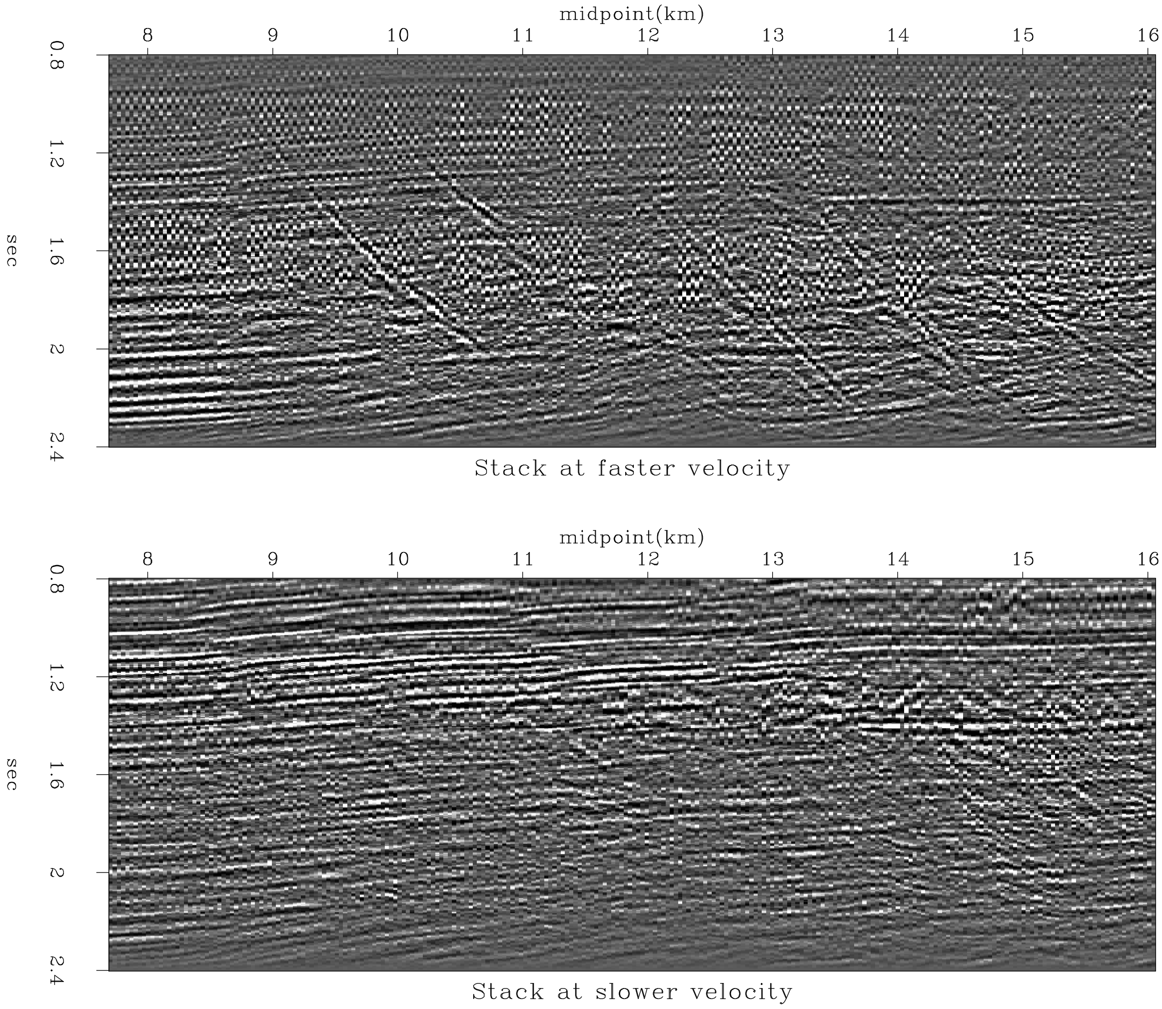

Figure 1.14: Stacks of Gulf of Mexico data with two different constant NMO velocities. Press button to see a **movie** in which each frame is a stack with a different constant velocity. dpmv-cvstacks [ER,M]

Another way of seeing the same conflict in the data is to look at a velocity-analysis panel at a single common-midpoint location such as the panel shown in Figure 1.15 made by subroutine `velsimp()` on page ??. In this figure it is easy to see that the velocity which is good for the dipping event at 1.5 sec is too high for the horizontal events in its vicinity.

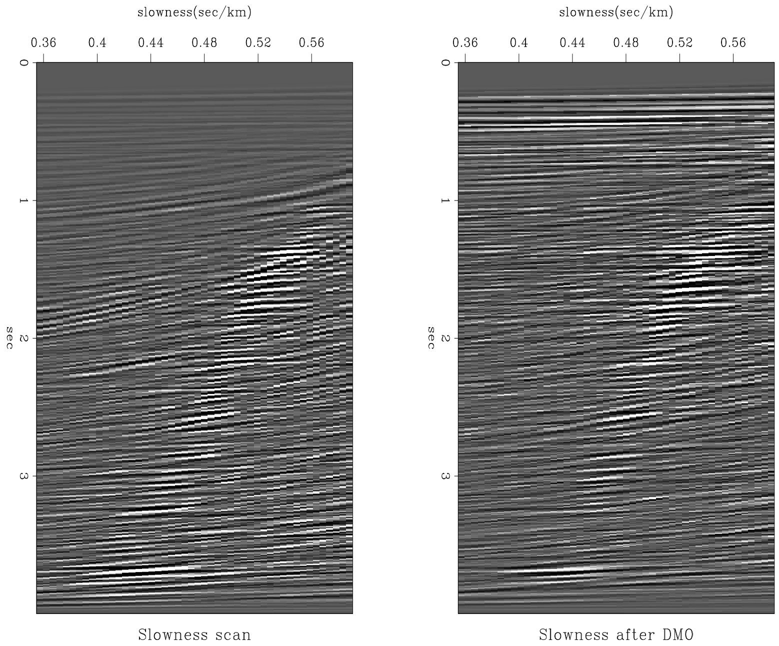

Figure 1.15: Velocity analysis panel of one of the panels in Figure 1.14 before (left) and after (right) DMO. Notice two velocities at the same time before DMO. dpmv-velscan [ER,M]

1.4 SHERWOOD'S DEVILISH

The migration process should be thought of as being interwoven with the velocity estimation process. J.W.C. **Sherwood** [1976] indicated how the two processes, migration and velocity estimation, should be interwoven. The moveout correction should be considered in two parts, one depending on offset, the NMO, and the other depending on dip. This latter process was conceptually new. Sherwood described the process as a kind of filtering, but he did not provide implementation details. He called his process *Devilish,* an acronym for "dipping-event velocity inequalities licked." The process was later described more functionally by **Yilmaz** as ***prestack partial migration,*** and now the process is usually called *dip moveout* (**DMO**) although some call it MZO, migration to zero offset. We will first see Sherwood's results, then Rocca's conceptual model of the **DMO** process, and finally two conceptually distinct, quantitative specifications of the process.

Figure 1.16 contains a panel from a stacked section. The panel is shown several times; each time the stacking velocity is different. It should be noted that at the low velocities, the horizontal events dominate, whereas at the high velocities, the steeply dipping events dominate. After the *Devilish* correction was applied, the data was restacked as before. Figure 1.17 shows that the stacking velocity no longer depends on the dip. This means that after *Devilish,* the velocity may be determined without regard to dip. In other words, events with all dips contribute to the same consistent velocity rather than each dipping event predicting a different velocity. So the *Devilish* process should provide better velocities for data with conflicting dips. And we can expect a better final stack as well.

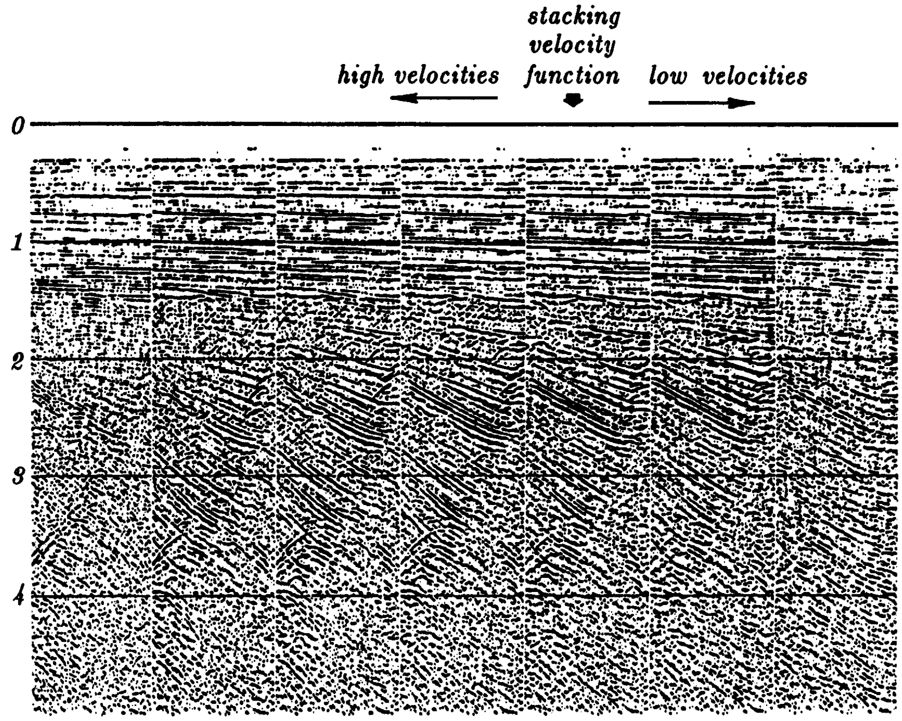

Figure 1.16: Conventional stacks with varying velocity. (distributed by Digicon, Inc.) $\boxed{\text{dpmv-digicon}}$ [NR]

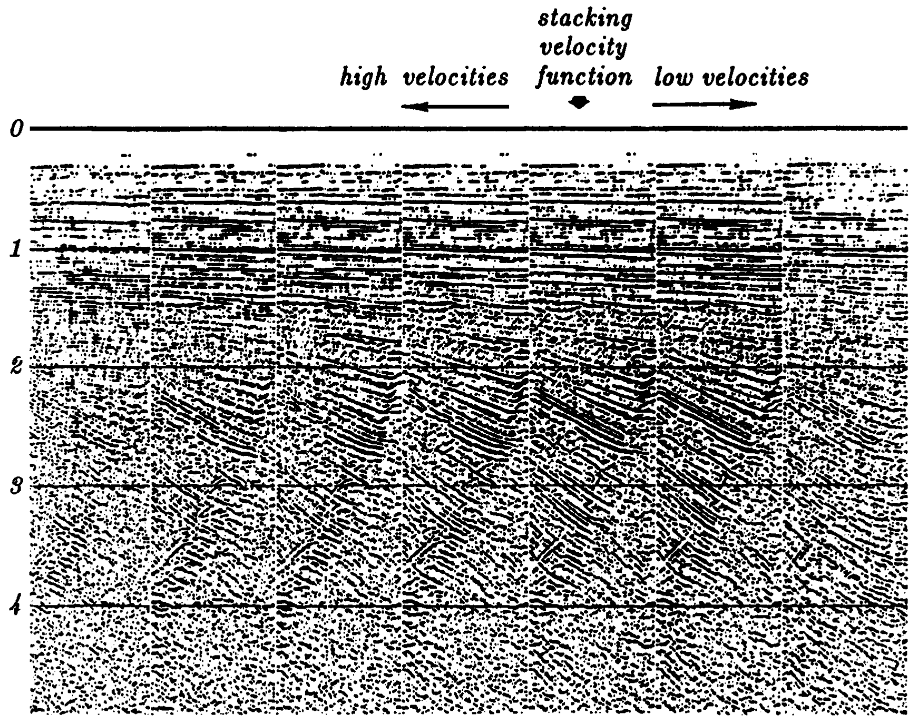

Figure 1.17: *Devilish* stacks with varying velocity. (distributed by Digicon, Inc.) $\boxed{\text{dpmv-devlish}}$ [NR]

1.5 ROCCA'S SMEAR OPERATOR

Fabio **Rocca** developed a clear conceptual model for Sherwood's dip corrections. Start with an impulse on a common offset section, and migrate it getting ellipses like in Figure 1.4. We did this with subroutine `flathyp()` on page 4 using some constant-offset `h`. Although the result is an ellipsoidal curve, think of it as a row of many points along an ellipsoidal curve. Then diffract the image thus turning each of the many points into a hyperbola. We do this with the return path of the same subroutine `flathyp()`, however the path back is taken with `h=0`. The result is shown in Figure 1.18. To enhance the appearance of the figure, I injected an intermediate step of converting the ellipsoid curve into a trajectory of dots on the ellipse. Notice that the hyperbola tops are not on the strong smear function that results from the superposition.

The strong smear function that you see in Figure 1.18 is Rocca's **DMO**+NMO operator, the operator that converts a point on a constant-offset section to a zero-offset section. The important feature of this operator is that the bulk of the energy is in a much narrower region than the big ellipse of migration. The narrowness of the Rocca operator is important since it means that energies will not move far, so the operator will not have a drastic effect and be unduly affected by remote data. (Being a small operator also makes it cheaper to apply). The little signals you see away from the central burst in Figure 1.18 result mainly from my modulating the ellipse curve into into a sequence of dots. However, noises from sampling and nearest-neighbor interpolation also yield a figure much like Figure 1.18. This warrants a more careful theoretical study to see how to represent the Rocca operator directly (rather than as a sequence of two nearly opposite operators).

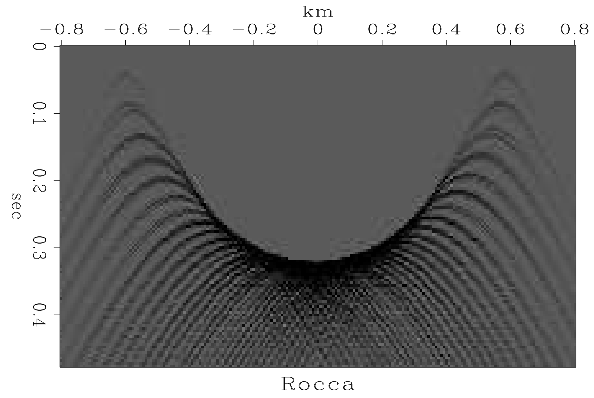

Figure 1.18: Rocca's prestack partial-migration operator is a superposition of hyperbolas, each with its top on an ellipse. dpmv-frocca [ER]

To get a sharper, more theoretical view of the Rocca operator, Figure 1.19 shows line drawings of the curves in a Rocca construction. It happens, and we will later show, that the Rocca operator lies along an ellipse that passes through $\pm h$ (and hence is independent of velocity!) Curiously, we see something we could not see on Figure 1.18, that the Rocca curve ends part way up the ellipse and it does not reach the surface. The place where the Rocca operator ends and the velocity independent ellipse continues is, however, velocity dependent as we will see. The Rocca operator is along the curve of osculation in Figure 1.19, i.e., the smile-shaped curve where the hyperbolas reinforce one another.

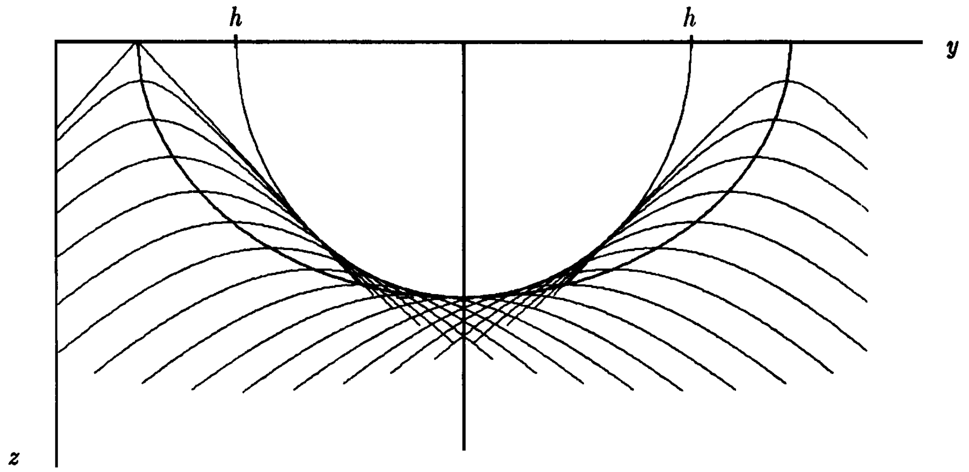

Figure 1.19: Rocca's smile. (Ronen) dpmv-rocca2 [NR]

1.5.1 Push and pull

Migration and diffraction operators can be conceived and programmed in two different ways. Let $\vec{\mathbf{t}}$ denote data and $\vec{\mathbf{z}}$ denote the depth image. We have

$$\vec{\mathbf{z}} = \mathbf{C}_h \vec{\mathbf{t}} \quad \text{spray or push an ellipse into the output} \tag{1.14}$$
$$\vec{\mathbf{t}} = \mathbf{H}_h \vec{\mathbf{z}} \quad \text{spray or push a flattened hyperbola into the output} \tag{1.15}$$

where h is half the shot-geophone offset. The adjoints are

$$\vec{\mathbf{t}} = \mathbf{C}'_h \vec{\mathbf{z}} \quad \text{sum or pull a semiCircle from the input} \tag{1.16}$$
$$\vec{\mathbf{z}} = \mathbf{H}'_h \vec{\mathbf{t}} \quad \text{sum or pull a flattened Hyperbola from the input} \tag{1.17}$$

In practice we can choose either of $\mathbf{C} \approx \mathbf{H}'$. A natural question is which is more correct or better. The question of "more correct" applies to modeling and is best answered by theoreticians (who will find more than simply a hyperbola; they will find its waveform including its amplitude and phase as a function of frequency). The question of "better" is something else. An important practical issue is that the transformation should not leave miscellaneous holes in the output. It is typically desirable to write programs that loop over all positions in the *output* space, "pulling" in whatever inputs are required. It is usually less desirable to loop over all positions in the *input* space, "pushing" or "spraying" each input value to the appropriate location in the output space. Programs that push the input data to the output space might leave the output too sparsely distributed. Also, because of gridding, the output data might be irregularly positioned. Thus, to produce smooth outputs, we usually *prefer the summation operators* $\mathbf{H}'$ for migration and $\mathbf{C}'$ for diffraction modeling. Since one could always force smooth outputs by lowpass filtering, what we really seek is the highest possible resolution.

Given a nonzero-offset section, we seek to convert it to a zero-offset section. Rocca's concept is to first migrate the constant offset data with an ellipsoid push operator $\mathbf{C}_h$ and then take each point on the ellipsoid and diffract it out to a zero-offset hyperbola with a push operator $\mathbf{H}_0$. The product of push operators $\mathbf{R} = \mathbf{H}_0\mathbf{C}_h$ is known as Rocca's smile. This smile operator includes both normal moveout and dip moveout. (We could say that dip moveout is defined by Rocca's smile after restoring the normal moveout.)

Because of the approximation $\mathbf{H} \approx \mathbf{C}'$, we have four different ways to express the Rocca smile:

$$\mathbf{R} \quad = \quad \mathbf{H}_0\mathbf{C}_h \quad \approx \quad \mathbf{H}_0\mathbf{H}'_h \quad \approx \quad \mathbf{C}'_0\mathbf{H}'_h \quad \approx \quad \mathbf{C}'_0\mathbf{C}_h \tag{1.18}$$

$\mathbf{H}_0\mathbf{H}'_h$ says sum over a flat-top and then spray a regular hyperbola.

The operator $\mathbf{C}'_0\mathbf{H}'_h$, having two pull operators should have smoothest output. Sergey Fomel suggests an interesting illustration of it: Its adjoint is two push operators, $\mathbf{R}' = \mathbf{H}_h\mathbf{C}_0$. $\mathbf{R}'$ takes us from zero offset to nonzero offset first by pushing a data point to a semicircle and then by pushing points on the semicircle to flat-topped hyperbolas. As before, to make the hyperbolas more distinct, I broke the circle into dots along the circle and show the result in Figure 1.20. The whole truth is a little more complicated. Subroutine `flathyp()` on page 4 implements $\mathbf{H}$ and $\mathbf{H}'$. Since I had no subroutine for $\mathbf{C}$, figures 1.18 and 1.20 were actually made with only $\mathbf{H}$ and $\mathbf{H}'$. We discuss the $\mathbf{C}'_0\mathbf{C}_h$ representation of $\mathbf{R}$ in the next section.

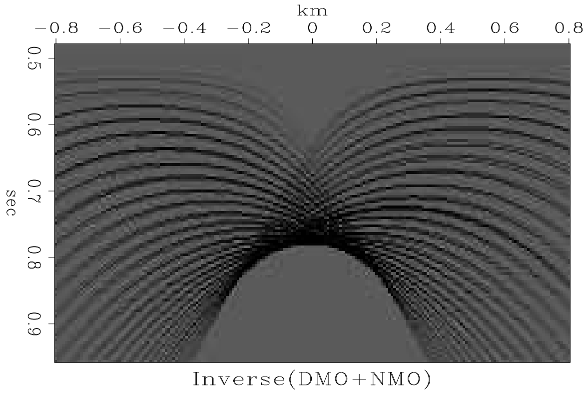

Figure 1.20: The adjoint of Rocca's smile is a superposition of flattened hyperbolas, each with its top on a circle. dpmv-sergey [ER]

1.5.2 Dip moveout with $v(z)$

It is worth noticing that the concepts in this section are not limited to constant velocity but apply as well to $v(z)$. However, the circle operator $\mathbf{C}$ presents some difficulties. Let us see why. Starting from the Dix moveout approximation, $t^2 = \tau^2 + x^2/v(\tau)^2$, we can directly solve for $t(\tau,x)$ but finding $\tau(t,x)$ is an iterative process at best. Even worse, at wide offsets, hyperbolas cross one another which means that $\tau(t,x)$ is multivalued. The spray (push) operators $\mathbf{C}$ and $\mathbf{H}$ loop over inputs and compute the location of their outputs. Thus $\vec{\mathbf{z}} = \mathbf{C}_h\,\vec{\mathbf{t}}$ requires we compute τ from t so it is one of the troublesome cases. Likewise, the sum (pull) operators $\mathbf{C}'$ and $\mathbf{H}'$ loop over outputs. Thus $\vec{\mathbf{t}} = \mathbf{C}'_h\,\vec{\mathbf{z}}$ causes us the same trouble. In both cases, the circle operator turns out to be the troublesome one. As a consequence, most practical work is done with the hyperbola operator.

A summary of the meaning of the Rocca smile and its adjoint is found in Figures 1.21 and 1.22, which were computed using subroutine `flathyp()` on page 4.

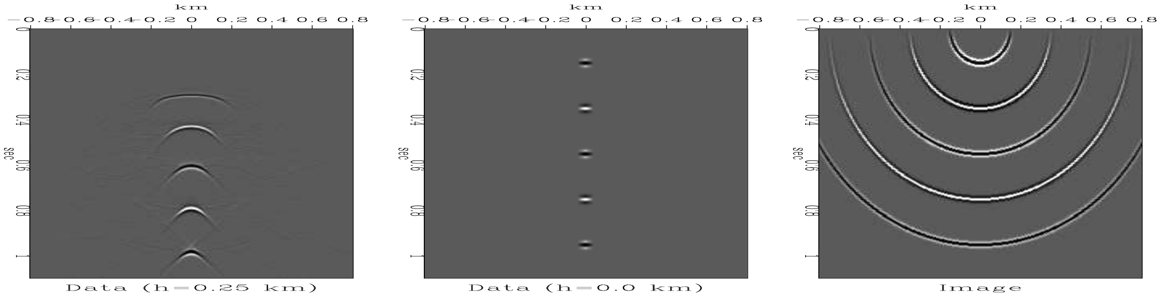

Figure 1.21: Impulses on a zero-offset section migrate to semicircles. The corresponding constant-offset section contains the adjoint of the Rocca smile. dpmv-yalei2 [ER]

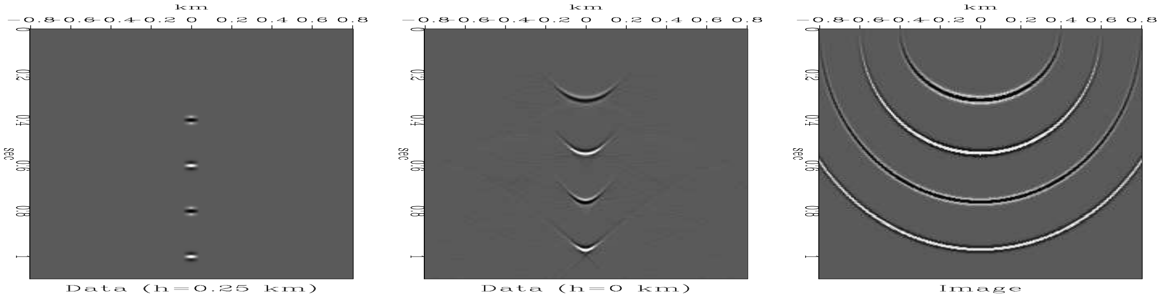

Figure 1.22: Impulses on a constant-offset section become ellipses in depth and Rocca smiles on the zero-offset section. dpmv-yalei1 [ER]

1.6 GARDNER'S SMEAR OPERATOR

A task, even in constant velocity media, is to find analytic expressions for the travel time in the Rocca operator. This we do now.

The Rocca operator $\mathbf{R} = \mathbf{C}_0'\mathbf{C}_h$ says to spray out an ellipse and then sum over a circle. This approach, associated with Gerry **Gardner**, says that we are interested in all circles that are inside and tangent to an ellipse, since only the ones that are tangent will have a constructive interference.

The Gardner formulation answers this question: Given a single nonzero offset impulse, which events on the zero-offset section will result in the same migrated subsurface picture? Since we know the migration response of a zero and nonzero offset impulses (circle and ellipse) we can rephrase this question: Given an ellipse corresponding to a nonzero offset impulse, what are the circles tangent to it that have their centers at the earth's surface? These circles if superposed will yield the ellipse. Furthermore, each of these circles corresponds to an impulse on the zero-offset section. The set of these impulses in the zero offset section is the **DMO**+NMO impulse response for a given nonzero offset event.

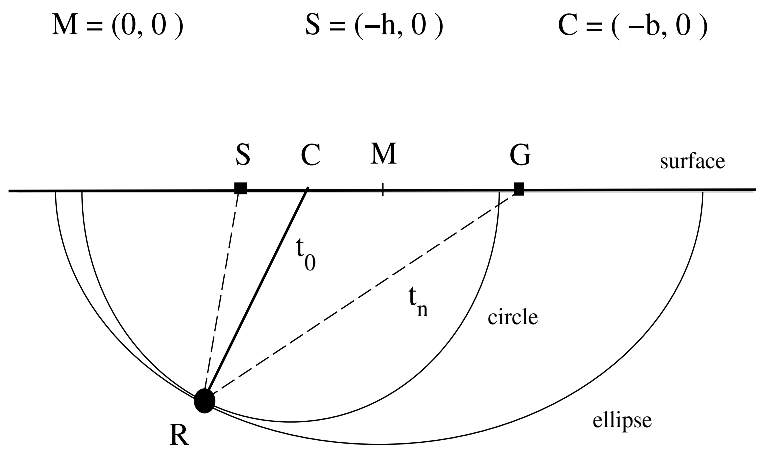

Figure 1.23: The nonzero offset migration impulse response is an ellipse. This ellipse can be mapped as a superposition of tangential circles with centers along the survey line. These circles correspond to zero offset migration impulse responses. dpmv-ell [NR]

1.6.1 Restatement of ellipse equations

Recall equation (1.9) for an ellipse centered at the origin.

$$0 \quad = \quad \frac{y^2}{A^2} + \frac{z^2}{B^2} - 1. \tag{1.19}$$

where

$$A \quad = \quad v_{\text{half}}\, t_h, \tag{1.20}$$

$$B^2 \quad = \quad A^2 - h^2. \tag{1.21}$$

The ray goes from the shot at one focus of the ellipse to anywhere on the ellipse, and then to the receiver in traveltime t_h. The equation for a circle of radius $R = t_0 v_{\text{half}}$ with center on the surface at the source-receiver pair coordinate $x = b$ is

$$R^2 \quad = \quad (y-b)^2 + z^2, \tag{1.22}$$

where

$$R \quad = \quad t_0\, v_{\text{half}}. \tag{1.23}$$

To get the circle and ellipse tangent to each other, their slopes must match. Implicit differentiation of equation (1.19) and (1.22) with respect to y yields:

$$0 \quad = \quad \frac{y}{A^2} + \frac{z}{B^2}\,\frac{dz}{dy} \tag{1.24}$$

$$0 \quad = \quad (y-b)+z\,\frac{dz}{dy} \tag{1.25}$$

Eliminating dz/dy from equations (1.24) and (1.25) yields:

$$y \quad = \quad \frac{b}{1-\frac{B^2}{A^2}}. \tag{1.26}$$

At the point of tangency the circle and the ellipse should coincide. Thus we need to combine equations to eliminate x and z. We eliminate z from equation (1.19) and (1.22) to get an equation only dependent on the y variable. This y variable can be eliminated by inserting equation (1.26).

$$R^2 \quad = \quad B^2\left(\frac{A^2-B^2-b^2}{A^2-B^2}\right). \tag{1.27}$$

Substituting the definitions (1.20), (1.21), (1.23) of various parameter gives the relation between zero-offset traveltime t_0 and nonzero traveltime t_h:

$$t_0^2 \quad = \quad \left(t_h^2-\frac{h^2}{v_{\rm half}^2}\right)\left(1-\frac{b^2}{h^2}\right). \tag{1.28}$$

As with the Rocca operator, equation (1.28) includes both dip moveout **DMO** and NMO.

1.7 DMO IN THE PROCESSING FLOW

Instead of implementing equation (1.28) in one step we can split it into two steps. The first step converts raw data at time t_h to NMOed data at time t_n.

$$t_n^2 \quad = \quad t_h^2-\frac{h^2}{v_{\rm half}^2} \tag{1.29}$$

The second step is the **DMO** step which like Kirchhoff migration itself is a convolution over the x-axis (or b-axis) with

$$t_0^2 \quad = \quad t_n^2\left(1-\frac{b^2}{h^2}\right) \tag{1.30}$$

and it converts time t_n to time t_0. Substituting (1.29) into (1.30) leads back to (1.28). As equation (1.30) clearly states, the **DMO** step itself is essentially velocity independent, but the NMO step naturally is not. Now the program. Backsolving equation (1.30) for t_n gives

$$t_n^2 \quad = \quad \frac{t_0^2}{1-b^2/h^2}. \tag{1.31}$$

Like subroutine `flathyp()` on page 4, our **DMO** subroutine `dmokirch()` on this page is based on subroutine `kirchfast()` on page ??. It is just the same, except where `kirch-fast()` has a hyperbola we put equation (1.31). In the program, the variable t_0 is called `z` and the variable t_h is called `t`. Note, that the velocity `velhalf` does exclusively occur in the break condition (which we have failed to derive, but which is where the circle and ellipse touch at $z=0$).

```
subroutine dmokirch( adj, add, velhalf, h, t0,dt,dx, modl,nt,nx, data)
integer ix,iz,it,ib, adj, add,                              nt,nx
real    amp,t,z,b,              velhalf, h, t0,dt,dx, modl(nt,nx),data(nt,nx)
call adjnull(        adj, add,                         modl,nt*nx, data,nt*nx)
if( h == 0)     call erexit('h=0')
do ib= -nx, nx {        b = dx * ib              # b = midpt separation
    do iz= 2, nt {       z = t0 + dt * (iz-1)     # z = zero-offset time

        if( h**2 <= b**2 )                    next
        t= sqrt(  z**2 / (1-b**2/h**2) )
        amp= sqrt(t) * dx/h
        if( velhalf*abs(b) * t*t > h**2*z)  break
        it = 1.5 + (t - t0) / dt
        if( it > nt )                         break
        do ix= max0(1, 1-ib),  min0(nx, nx-ib)
                if( adj == 0 )
                        data(it,ix+ib) = data(it,ix+ib) + modl(iz,ix   ) * amp
                else
                        modl(iz,ix   ) = modl(iz,ix   ) + data(it,ix+ib) * amp
        }
   }
return; end
```

Figures 1.24 and 1.25 were made with subroutine `dmokirch()` on the facing page. Notice the big noise reduction over Figure 1.18.

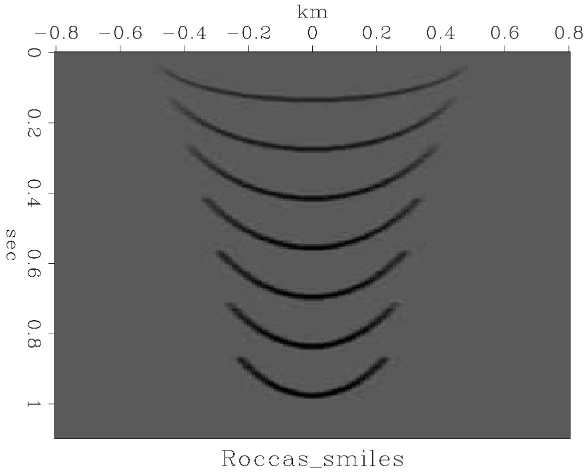

Figure 1.24: Impulse response of DMO and NMO dpmv-dmatt [ER]

1.7.1 Residual NMO

Unfortunately, the theory above shows that **DMO** should be performed *after* NMO. **DMO** is a convolutional operator, and significantly more costly than NMO. This is an annoyance because it would be much nicer if it could be done once and for all, and not need to be redone for each new NMO velocity.

Much practical work is done with using constant velocity for the DMO process. This is

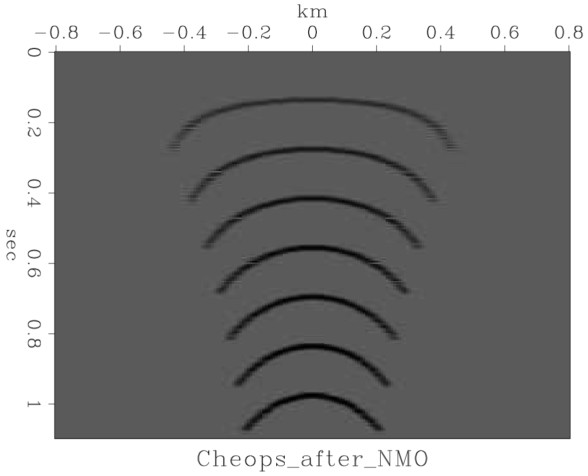

Figure 1.25: Synthetic Cheop's pyramid dpmv-coffs [ER]

roughly valid since DMO, unlike NMO, does little to the data so the error of using the wrong velocity is much less.

It is not easy to find a theoretical impulse response for the DMO operator in $v(z)$ media, but you can easily compute the impulse response in $v(z)$ by using $\mathbf{R} = \mathbf{H}_0\mathbf{H}'_h$ from equation (1.18).

1.7.2 Results of our DMO program

We now return to the field data from the Gulf of Mexico, which we have processed earlier in this chapter and in chapter **??**.

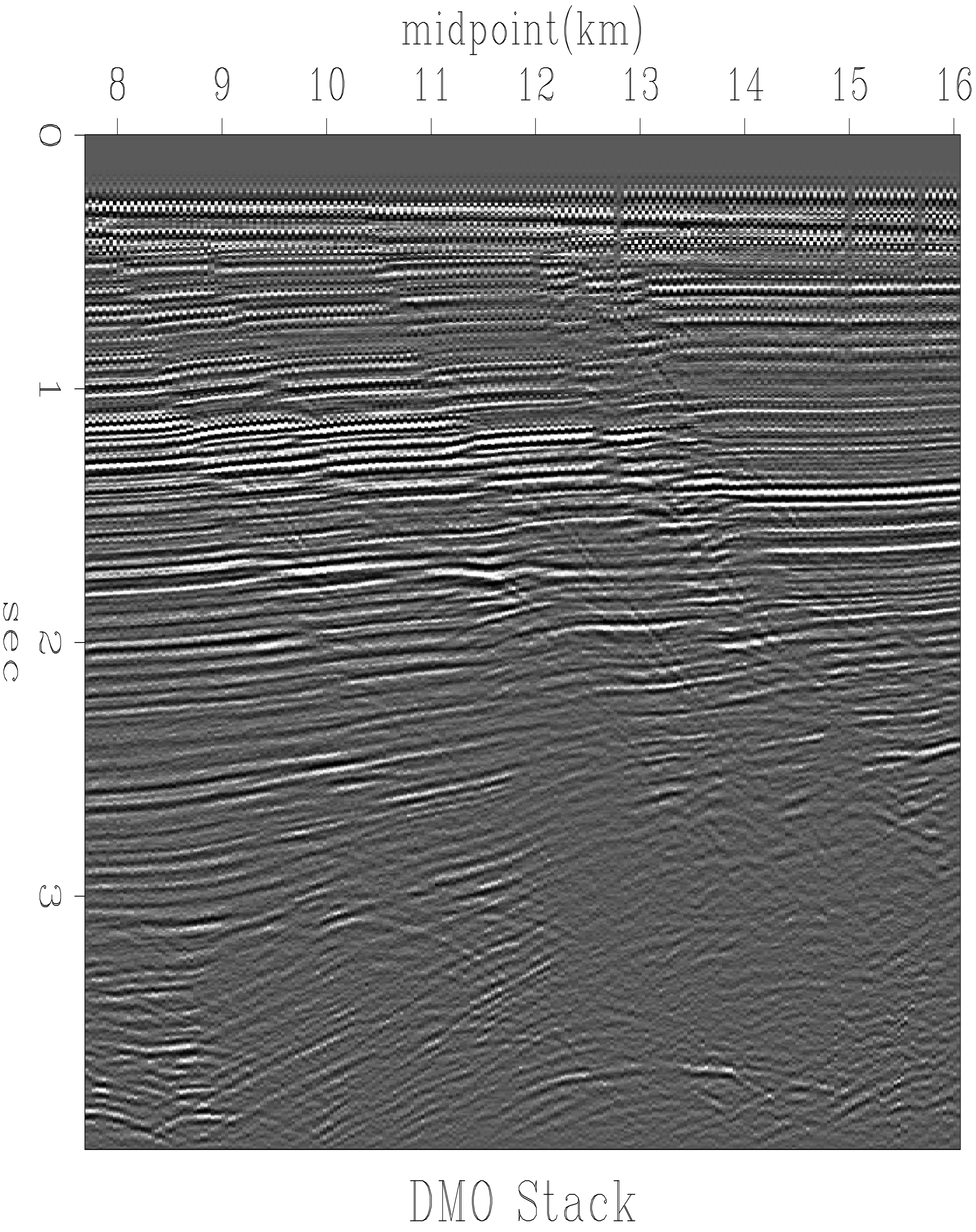

Figure 1.26: Stack after the dip-moveout correction. Compare this result with Figure **??**. This one has fault plane reflections to the right of the faults. dpmv-wgdmostk [ER,M]

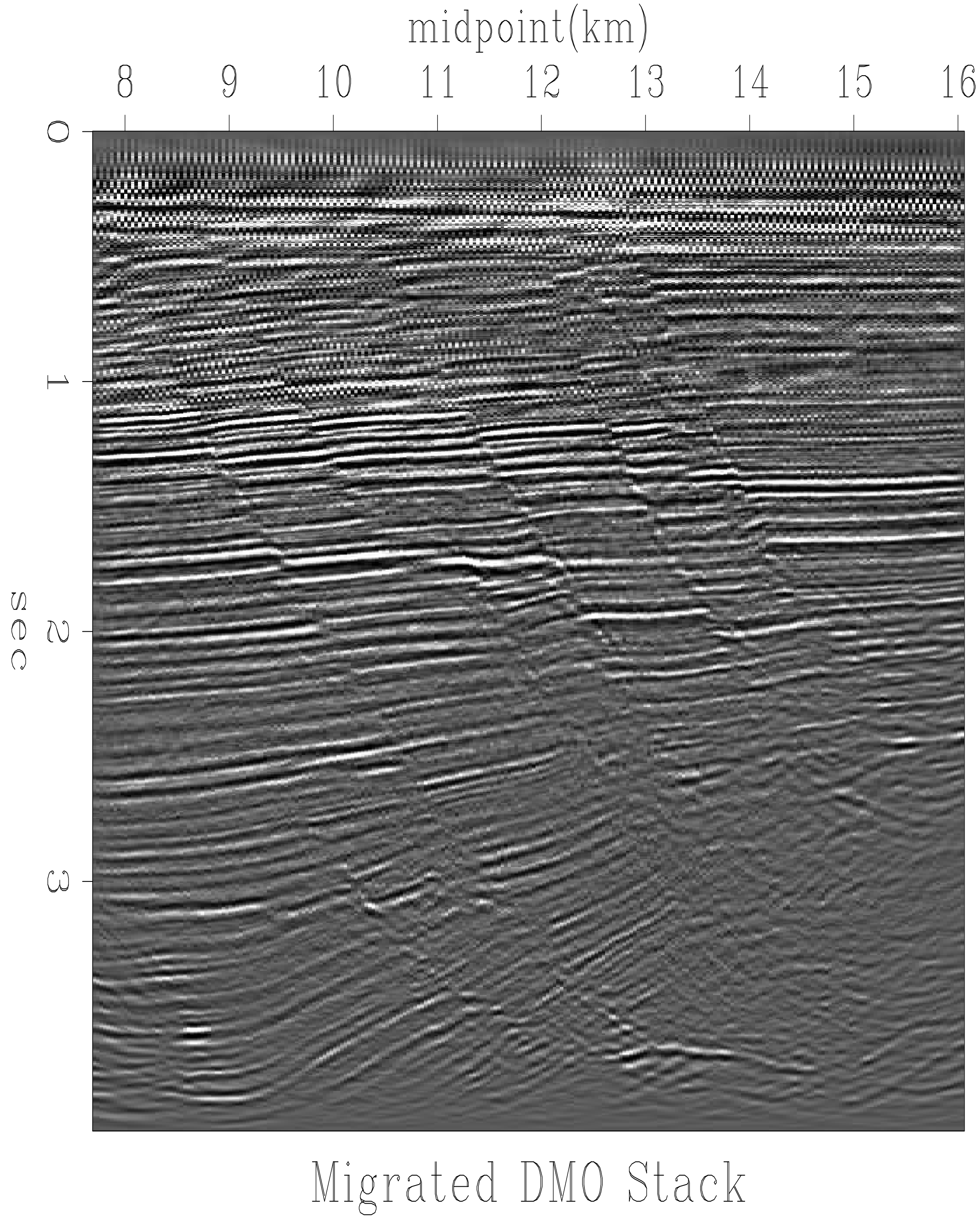

Figure 1.27: Kirchhoff migration of the previous figure. Now the fault plane reflections jump to the fault. dpmv-wgdmomig [ER,M]

Index