Chapter 4

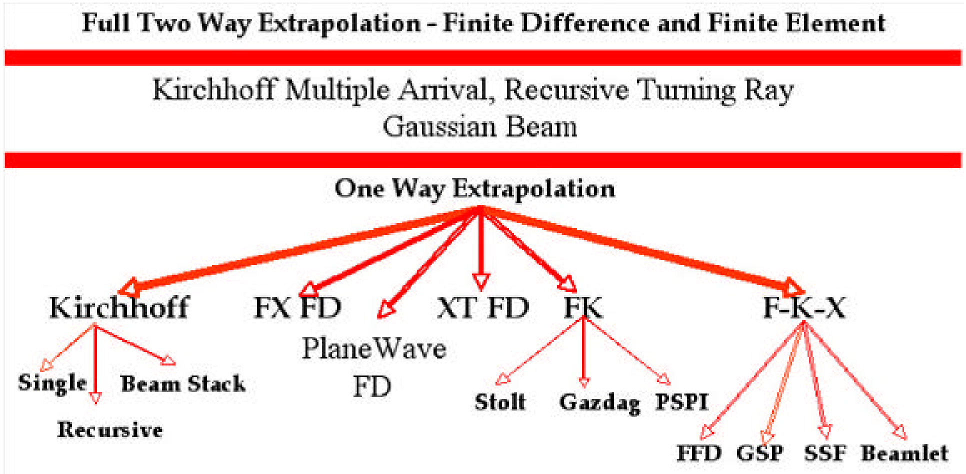
Zero Offset Migration Algorithms

The sections on imaging are clearly the heart of the book—this chapter contains a discussion of zero-offset poststack migration, including time migrations, and Chapter 6 discusses prestack migration. The wide selection of algorithms are from what we call the migration hierarchy, but not all of those algorithms are covered completely. Considerable importance is placed on the impact of aliasing and on how full wave-equation-based methods function in exactly the same manner as much simpler approaches, although they are more difficult to understand.

The Migration Algorithm Hierarchy

We again appeal to history to organize the material as we discuss various algorithms in terms of the Migration Hierarchy shown in Figure 4-1. This figure is a very simplified diagram of the variety of migration approaches available to image reflection seismic data. In a manner analogous to what was discussed in the modeling section (Chapter 3), this hierarchy refers to the theoretical assumptions made in algebraic manipulations of the initial propagator equations. It is important to note that, although we will address this hierarchy in a top down manner, this is definitely not the order in which the algorithms were developed. It is also interesting to observe that computational efficiency generally improves as we go down the figure; that is, full two-way extrapolation at the top is much more computationally intensive than almost any of the one-way methods at the bottom.

Figure 4-1. A modern hierarchy of migration algorithms.



Each symbol or phrase in the figure references a particular approach to migration. In general, capital *F* indicates a frequency domain method, K is a wavenumber method, X is a method in space, and T is a method in time. Thus, FX migration means an algorithm that works essentially in the frequency-space domain, while XT is a space-time method. FKX methods are the dual-domain methods that achieve their goals by bouncing back and forth between frequency, wavenumber, and space. We could conceivably have a FKXT or multi-domain method, but, to date, such methods have not gained much popularity.

All of the methods in the hierarchy, including all of the Kirchhoff approaches, were derived from a three-dimensional version of the one-dimensional forward modeling propagator. Fundamentally, the geometry dictated by the wave equation from which they arose is the same for each and every approach. Wave equation propagators are all derived from continuous versions of discrete equations like Equation 4-1, where v is the speed of sound in the medium.

$$\frac{u(x+h,t)-2u(x,t)+u(x-h,t)}{\Delta x^2} = \frac{1}{v^2}\left(\frac{u(x,t+\Delta t)-2u(x,t)+u(x,t-\Delta t)}{\Delta t^2}\right) \tag{4-1}$$

Equation 4-1 is written in continuous form as Equation 4-2.

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{v^2}\frac{\partial^2 u}{\partial t^2} \tag{4-2}$$

By extension, the basic three-dimensional partial differential equation then takes the form of Equation 4-3.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \frac{1}{v^2}\frac{\partial^2 u}{\partial t^2} \tag{4-3}$$

This wave equation is the simplest form from which all migration algorithms arise, but that does not mean it is the correct one. It just means that it was easier to start with this one rather than the general heterogeneous version shown in Equation 4-4, where ρ is density, v is sound speed, and ∇ is the vector differential operator given by Equation 2-19. Regardless of the complexity of the original equation, the fundamental migration concepts do not change.

$$\frac{\partial^2 U}{\partial t^2} - \rho(x,y,z)\, v^2(x,y,z)\, \nabla \cdot \frac{1}{\rho(x,y,z)} \nabla U = s(x_s, t) \tag{4-4}$$

Migration in Depth

This section describes a number of the methods used to perform migrations in depth. In depth migration, reflections in seismic data are moved to their correct locations in space. These methods include both explicit and implicit one- and two-way methods, two-way reverse time methods, Kirchhoff style methods, and plane wave techniques.

Explicit Two-Way XT-Reverse-Time Migration

Perhaps the easiest way to understand migration is to consider what we learned in the section on Zero Offset Modeling. To produce true zero-offset data, we started with a given model, halved the velocity, computed reflection amplitudes for each point on a reflector, and then simulated explosions at each such point reflector. These simulated explosions were ignited at $t = 0$. The surface wavefield at $z = 0$ for all possible recording times, t, became a zero-offset subsection. The trick of halving the velocity was all that was necessary to ensure proper arrival times in our zero-offset experiment. It does not take a lot of thought to see that if we simply reverse this process, we will arrive at an algorithm that will produce an image of the subsurface at the original $t = 0$.

“Reverse time migration” uses the reversed time data as a source term for a seismic modeling exercise. Thus, the “source” in this case is a multiplicity of traces. In 3D, it is a common-offset volume where the distance between the source and receiver is zero. Figure 4-2 is a schematic example of the process. The first and second lines of Figure 4-2 show what remains as each time slice is propagated into the output image, while the second and fourth lines show the current image at each propagation step. Thus, the upper first and second images in the upper left show the initial condition of the zero-offset wavefield and the subsurface. In practice, the initial subsurface subsection would be filed with zeros. Here, we just reversed the modeling images from the subsection on modeling. Note that as the process continues, the final subsurface at the bottom right contains the fully migrated image—all amplitudes in the original recorded zero-offset subsection have been exhausted and are now zero; that is, the entire recorded wavefield has been back propagated into the Earth. In these images, all subsurface reflectors are clearly being imaged from above and from below.

Figure 4-2. Reverse time migration snapshots.

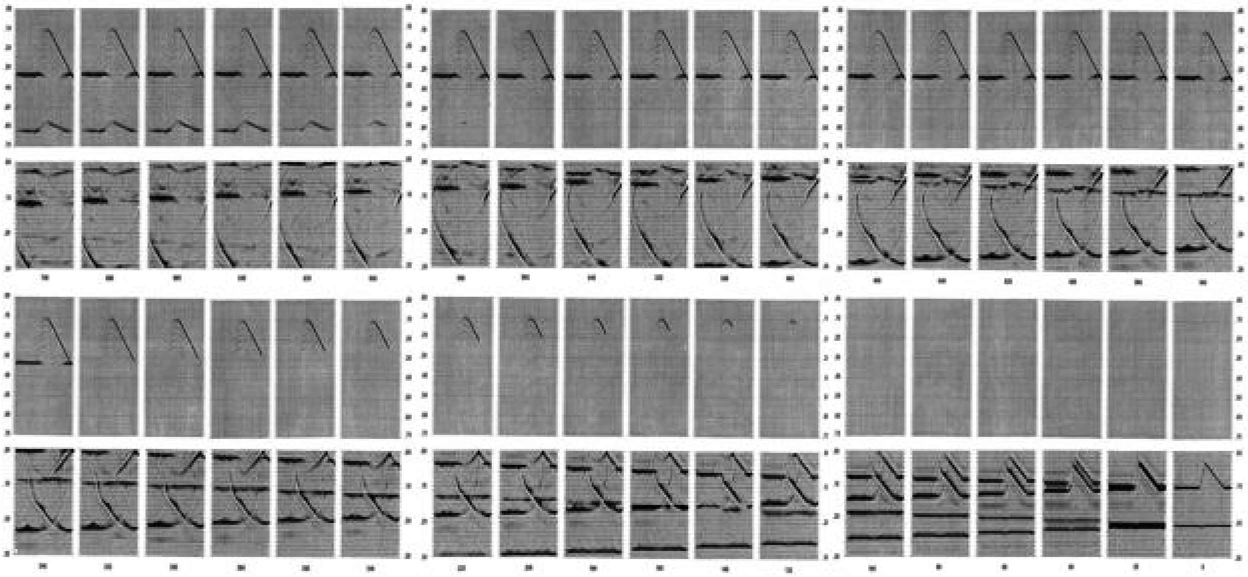
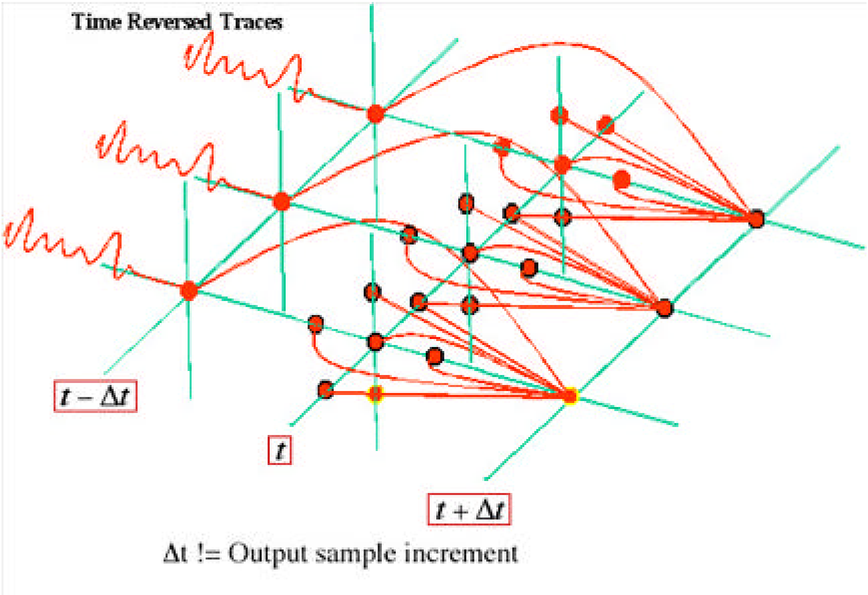


Figure 4-3 visualizes what happens locally during the process described by Figure 4-2 and its associated comments. Input traces are reversed in time and fed directly into the modeling algorithm to produce the final image. This kind of method is called an *explicit method* because it back propagates time-reversed data one step at a time.

Figure 4-3. Using time-reversed traces with a modeling propagator to produce subsurface images



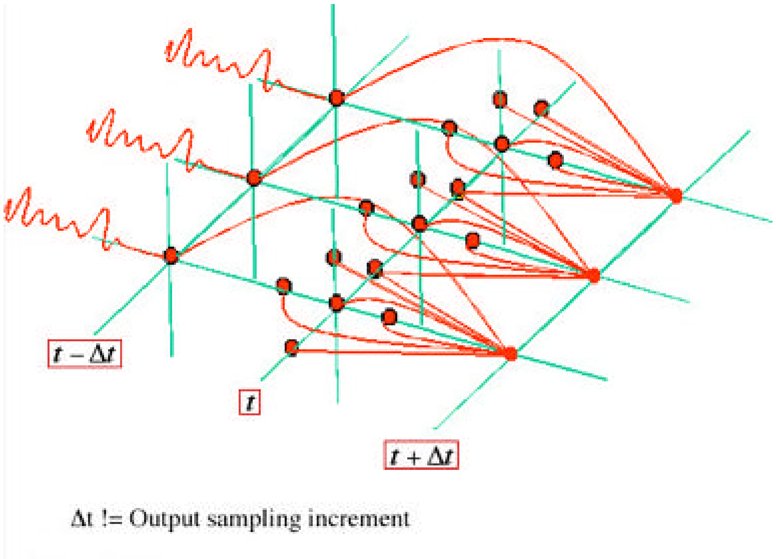
Implicit Two-Way XT Reverse-Time Migration

The only difference between explicit and implicit zero offset migration is that the former explicitly employs the mesh shown in Figure 4-3, while the latter inverts a rather large matrix to obtain the migrated image. The "sources" in both are the time-reversed stacked or zero-offset traces, so the only differences between the explicit and implicit versions involve the implementation.

Explicit One-Way XT Reverse-Time Migration

Again, going back to our modeling discussion, it is quite easy to make a one-way migration method. Figure 4-4 illustrates this point quite accurately. Note that three-dimensional, one-way-reverse-time migration differs from its two-way counterpart only in that the propagating stencil is free of coefficients below the current time or depth slice. Thus, unlike the two-way algorithm, one-way imaging proceeds one time or depth slice at a time. It is precisely this feature that makes one-way wave equation imaging vastly more efficient than its full two-way cousin.

Figure 4-4. Three-dimensional non-Kirchhoff time or depth migration.



Implicit One-Way XT Reverse-Time Migration

Here we only need to invert the appropriate upper-diagonal matrix to find the imaged solution. Because the process is based on a back-substitution method, it is extremely efficient and therefore much faster than the full two-way-implicit scheme.

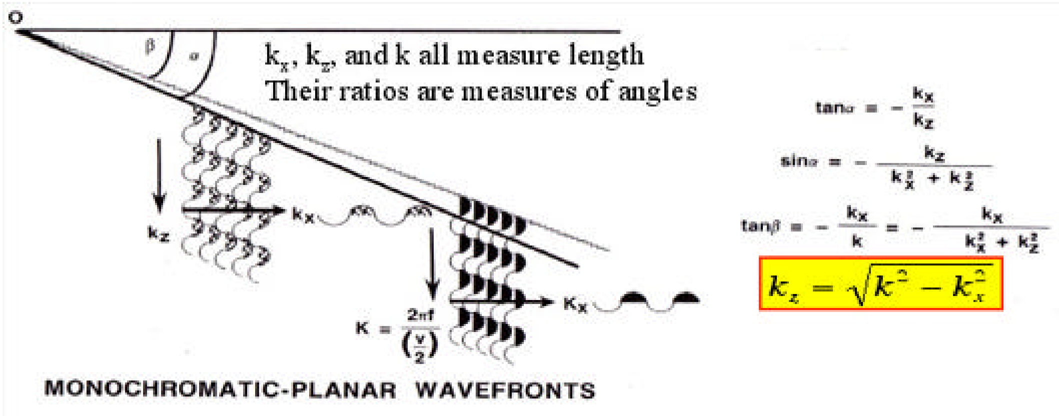
One-Way FK Reverse-Time Migration

One-Way-FK-Reverse-Time Migration methods can be divided into FK, phase shift methods, and phase shift plus interpolation methods.

FK or Stolt Migration

In the mid 1970's, R. H. Stolt at Conoco's Ponca City, Oklahoma research facility invented what has become known as Stolt or FK migration, as shown in Figure 4-5.

Figure 4-5. R. H. Stolt's frequency-wavenumber Migration.


O
β
α
k_x, k_z, and k all measure length
Their ratios are measures of angles
k_z
k_x
K = 2πf / (v/2)
K_x
MONOCHROMATIC-PLANAR WAVEFRONTS
tanα = − k_x / k_z
sinα = − k_z / (k_x² + k_z²)
tanβ = − k_x / k = − k_x / (k_x² + k_z²)
k_z = √(k² − k_x²)

What Stolt recognized was that in a normal zero-offset seismic experiment, we are given time data, $u(x, y, 0, t)$, measured at $z = 0$, and what we want is the exploding reflector depth data, $u(x, y, z, 0)$. He also understood that if he could compute the depth wavenumber, k_z, from frequency, $\omega = 2\pi f$, and the x and y wavenumbers, he could produce a very fast FK domain method.

Figure 4-5 shows how he did migration. The vertical wavenumber k_z in depth is directly computable from the 2D dispersion relation $k^2 = k_x^2 + k_z^2$ where $k = \frac{\omega}{v}$. Because the dispersion relation in Equation 4-5 holds in 3D, extension to three dimensions is straightforward.

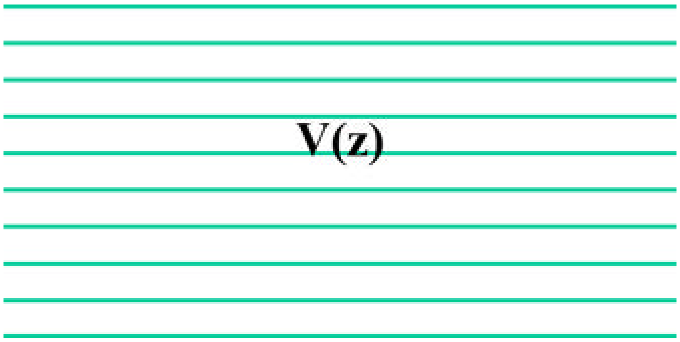
(4-5)
$$k^2 = k_x^2 + k_y^2 + k_z^2$$

The assumption of constant velocity is the Achilles' heel of Stolt's method. It cannot be derived mathematically unless the velocity is constant, and no method has ever been found to overcome this weakness. However, since it was Fourier-based, it was much faster than any of the other methods of the day. Researchers quickly tried to get around the constant velocity issue.

Phase Shift Migration

The FK domain phase shift method shown in Figure 4-6 was based purely on reverse-time modeling. It was one of the first approaches to successfully remove the constant velocity criteria imposed by the Stolt approach. It was also the first and, perhaps until very recently, the only method that was capable of imaging up to 90 degrees. The phase shift approach was invented by Jeno Gazdag shortly after the Stolt method appeared. It, in effect, applies the Stolt method to each slice in a Earth model where the propagation velocity varies only vertically. Thus, it is a recursive technique that repetitively applies the formula for k_z in Figure 4-5 to each constant velocity depth slice in the $v(z)$ model. It begins at zero depth ($z = 0$) and proceeds until the desired depth is reached. As long as the Earth's velocity varies only vertically, it does a remarkably good job of imaging steeply dipping events. This approach can be modified to include at least some two-way propagation, so it is also very versatile.

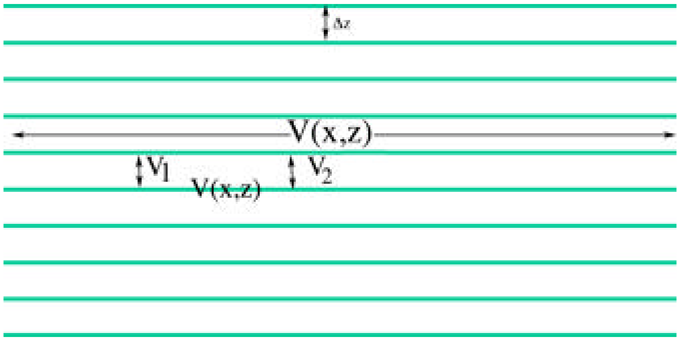
Figure 4-6. FK domain depth-slice by depth-slice migration



Phase Shift Plus Interpolation Migration

As discussed in the chapter on seismic modeling, phase shift plus interpolation (PSPI) modeling is easily adapted to produce a phase-shift-plus-interpolation style reverse-time migration, as shown in Figure 4-7. The sources in this case are again the time reversed traces and the result is a depth image of the zero-offset input data.

Figure 4-7. PSPI FK domain depth-slice by depth-slice migration



One-Way FKX-Reverse-Time Migration

One-way FKX-reverse-time migration methods can be divided into split-step and phase screen methods.

Split Step Imaging

The split-step imaging method is a direct application of the split-step modeling method in Split-Step Methods on page 67. Explaining this method in considerable detail is beyond the scope of this book. For the purposes here, it is enough to say that the method splits all downward propagation between the FK and FX domains.

Phase Screen Migration

The so-called phase-screen method, as envisioned by Ru Shan Wu at the University of California at Santa Cruz, is briefly discussed in Higher Order FKX Methods on page 68. This method, when compared to the split-step approach, is based on an improved set of approximations for utilization in the FX domain.

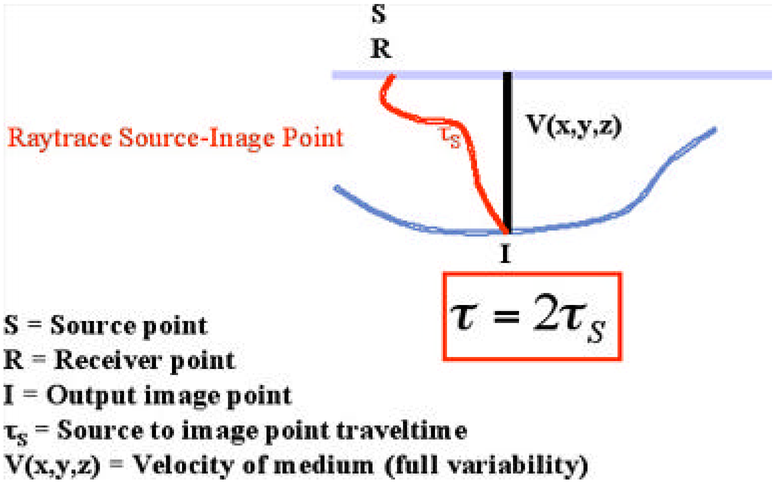
Kirchhoff Style Methods

Kirchhoff migrations are a method of seismic migration that uses the integral form (Kirchhoff equation) of the wave equation. Kirchhoff style methods can be separated into single-arrival methods and multiple-arrival methods.

Single-Arrival Kirchhoff Migration

In single arrival Kirchhoff depth migration, as shown in Figure 4-8, either a direct finite difference solution to the Eikonal equation or full 3D raytracing is used to calculate exact traveltimes from the source/receiver point to the output image point.

Figure 4-8. Single arrival Kirchhoff depth migration.


S
R
Raytrace Source-Inage Point
τ_S
V(x,y,z)
I
$\tau = 2\tau_S$
S = Source point
R = Receiver point
I = Output image point
τ_S = Source to image point traveltime
V(x,y,z) = Velocity of medium (full variability)

Because Eikonal-based methods are generally able to calculate only first arrivals, they are no longer popular as part of the general depth migration methodology.

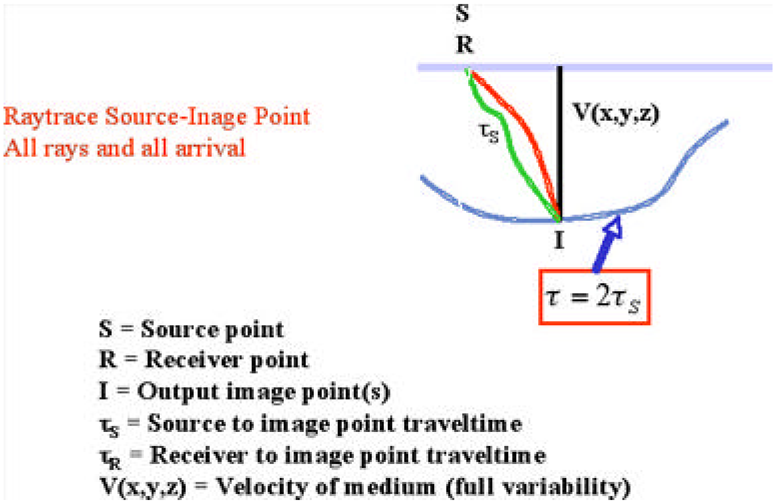
However, raytrace methods facilitate the calculation of multiple arrivals, and the selection of those arrivals better serves the migration process. Typical arrivals of this type are maximum energy, minimum distance, or minimum velocity. In areas with strong lateral velocity variations, such as salt regimes, the minimum velocity methodology is considered to be optimum since they avoid headwaves caused by proximity to salt or other high velocity structures.

The utilization of a single arrival in Kirchhoff depth migration technology is one of the chief reasons these methods cannot image below complex structure.

Multiple-Arrival Kirchhoff Migration

When we use all of the possible arrivals, as shown in Figure 4-9, we can achieve a superior result to what is achieved with the single arrival approach. While multiple arrivals complicate the migration algorithm and generally make it more computationally costly, the benefit of including more arrivals usually outweighs the increased cost.

Figure 4-9. Multiple arrival Kirchhoff depth migration.


S
R
Raytrace Source-Inage Point
All rays and all arrival
τ_S
V(x,y,z)
I
$\tau = 2\tau_S$
S = Source point
R = Receiver point
I = Output image point(s)
τ_S = Source to image point traveltime
τ_R = Receiver to image point traveltime
V(x,y,z) = Velocity of medium (full variability)

Plane-Wave Migration

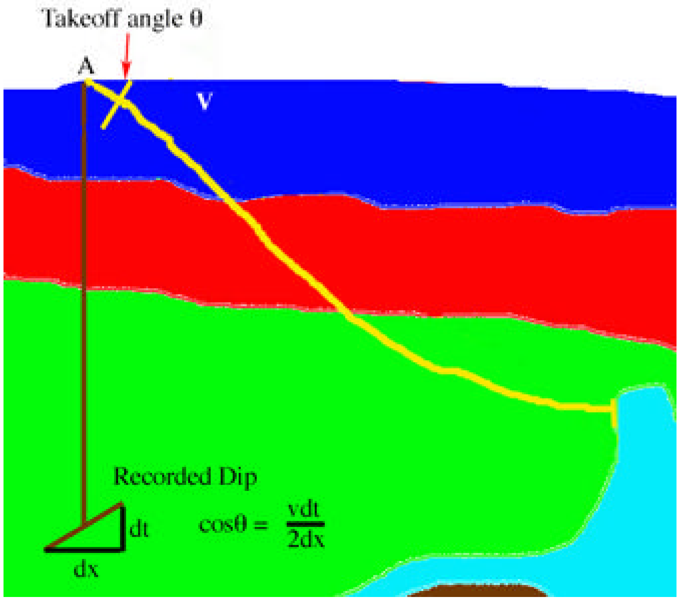
Plane-wave migration methods can be divided into pure plane wave methods, beam stack methods, and Gaussian beam methods.

Pure Plane Wave Migration

The hand migration discussed in the chapter on historical methods and displayed in Figure 3-5 is in actuality one of the first plane wave migration techniques. The constant velocity trigonometric solution in that figure provides the formulas for calculating the migration time, τ, and the lateral shift, x, to determine the migrated position, S'.

Figure 4-10 shows how you can avoid the constant velocity assumption completely. In this figure, the $\frac{dt}{dx}$ measured from the unmigrated data, together with the near surface velocity, determines the cosine of the plane wave takeoff angle, θ, at the zero-offset location of the reflected arrival from the steeply dipping structure. If we shoot a ray with this takeoff angle into the subsurface model and continue it until the recorded time at A is exhausted, the orthogonal to the ray will be the dip of the reflected event and the position of the end of the ray will be its location. Repeating this process for every estimated dip ultimately produces a full migration of the recorded zero-offset or stacked subsection.

Figure 4-10. Hand migration as a beam or plane wave migration.



Beam Stack Migration

Kirchhoff methods are converted to a beam or slant stack approach by simplifying the manner in which traveltimes are calculated. The idea is to try to use the distance from a central source or receiver to calculate another traveltime at a nearby location. In two dimensions, the formula is something like that shown in Equation 4-6, where $\tau_{x_{s0}}$ is the raytraced traveltime from the central location, x_{s0}, and $p_{x_{s0}}$ are suitably chosen scalars.

(4-6)
$$\tau_{x_s} = \tau_{x_{s0}} + p_{x_{s0}}(x - x_s)$$

Noting that $p_{x_{s0}}$ must have units of time over distance, we can easily infer that this value must tell us how $\tau_{x_{s0}}$ changes incrementally versus an incremental change in the central source position. That is, $p_{x_{s0}}$ must have the form in Equation 4-7, which is just the derivative (gradient in 3D) of the traveltime with respect to source position.

(4-7)
$$p_{x_{s0}} = \frac{\Delta\tau_{x_{s0}}}{\Delta x}$$

Traveltimes at a position close to the central point can then be calculated from Equation 4-8.

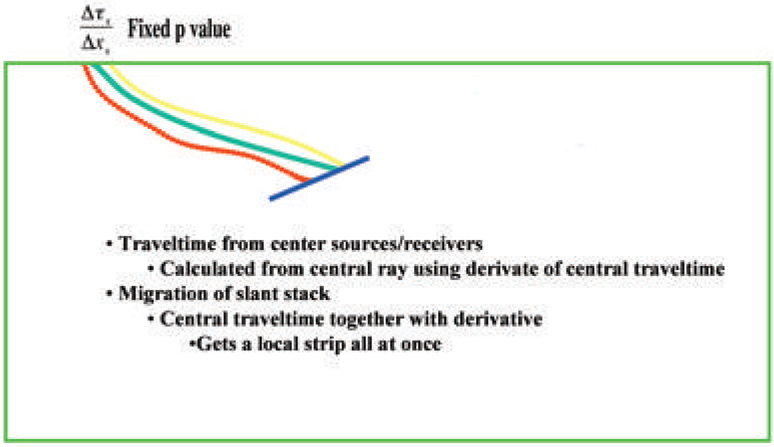
(4-8)
$$\tau_{x_s} = \tau_{x_{s0}} + \frac{\Delta\tau_{x_{s0}}}{\Delta x}(x - x_s) = \tau_{x_{s0}} + \frac{d\tau_{x_{s0}}}{dx}$$

Equation 4-9 performs a standard zero-offset Kirchhoff migration in 2D, where A is a traveltime correction factor, $\tau_{x_s}(x,z)$ is the traveltime from the surface location x_s to (x,z), and $D(x_s,t)$ is the zero-offset data to be imaged as $I(x,z)$.

(4-9) $$I(x,z) = \Sigma_{x_s} A(x_s) D(x_s, \tau_{x_s}(x,z))$$

The basic ideas for this kind of traveltime computation is shown in Figure 4-11.

Figure 4-11. Traveltime computation in terms of local distance from a given centrally located shot.



- Traveltime from center sources/receivers
    - Calculated from central ray using derivate of central traveltime
- Migration of slant stack
    - Central traveltime together with derivative
        - Gets a local strip all at once

Inserting the formula for calculating τ_{x_s} results in Equation 4-10.

(4-10) $$I(x,z) = \Sigma_{x_s} A(x_s) D(x_s, \tau_{x_{s_0}} + p_{x_{s_0}}(x - x_s))$$

Equation 4-10 suggests that we can perform this operation in two independent steps. First, we calculate the slant stack of our input zero-offset data, $D(x,t)$, using Equation 4-11. Then we simply Kirchhoff migrate this slant stack bundle using Equation 4-12, where the sum is taken over all x_s that are sufficiently close to x_{s_0}. The image $I(x,z)$ is then just the sum of all the bundles.

(4-11) $$D_S(x_{s_0}, p, \tau) = \Sigma_{x_{s_0}} D(x_{s_0}, \tau + p(x - x_{s_0}))$$

(4-12) $$I(x_s, z) = \Sigma A(x_{s_0}) D_S(x_{s_0}, p_{x_{s_0}}, \tau_{x_{s_0}})$$

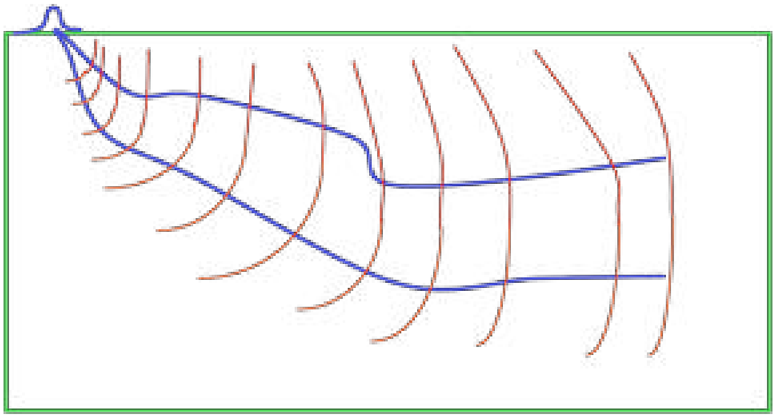
Since the slant stack can be calculated independently of the migration, and since the derivative of the traveltime can be calculated during the raytracing, the computational cost of performing the necessary steps prior to the migration stage is only slightly greater than that required for a standard Kirchhoff migration.

On the other hand, since the entire slant stack bundle replaces many traces in the migration process, Kirchhoff-beam-stack methods can be significantly more efficient than their traditional straight-forward implementations. Typically, these Kirchhoff-beam-stack migrations are about 10 times as fast as a traditional Kirchhoff approach.

Gaussian Beam Migration

Figure 4-12 demonstrates graphically how we can construct a significant portion of the required impulse response for a zero-offset trace using what are called *Gaussian beams*. The mathematics associated with Gaussian beams is beyond the scope of this book, but the concepts are worth some explanation. Although somewhat of an over-simplification, we can think of a Gaussian beam as being equivalent to a phase shift calculation using a $v(\rho)$ velocity function extracted using the arc-length along the ray, ρ, as the depth parameter.

Figure 4-12. A partial wave response due to two rays.



Several corrections to this phase shift are made to ensure that the process works correctly. First, as the ray calculations proceed, two special functions are calculated at each increment, $\Delta\rho$, of the arc length. These functions provide the necessary amplitude decays parallel and orthogonal to the ray at any given location. Second, the beam is weighted with the normal Gaussian function to ensure that the sum of all such beams accurately represents the true impulse response of the migration operator. The result of applying the corrections is called a Gaussian beam, and its amplitudes are then summed into corresponding subsurface locations on either side of the central ray. The sum of all such beams produces the completed impulse response.

Migration in Time

Time migration moves dipping events from their apparent locations to their true locations in time. The resulting image is shown in terms of traveltime rather than depth, and must then be converted to depth with an accurate velocity model. Because of the clear and direct connection to data synthesis over media described in terms of depth, it is easy to explain depth migration in the time-reversed sense.

However, even though it can be one of the most accurate imaging methods, depth migration has not yet reached the acceptance level it deserves. There are many reasons for this. One reason is the belief by many people that finding the most accurate velocity is still beyond the reach of most depth-migration methods. This was certainly true when actually performing a prestack migration of any form was beyond the reach of the computational capacities of the period. Historically, imaging in time by hand (see Chapter 1) was easy to explain with virtually no need to even consider partial differential equations. Interpreters were required to have only limited mathematical ability to do the migration and could perform all depth conversions as a simple vertical stretch.

Moreover, as shown in Figure 3-17 and discussed in Chapter 1, interpreters could compute the migration position and time quite easily using only an appropriate RMS velocity and locally estimated dip. Consequently, the coupling of readily available computer power with multi-fold data for estimating stacking velocities led directly to a plethora of time-migration versions of the depth migration algorithms listed in the previous sections.

Converting to Vertical Time

Converting depth migration algorithms into time migration algorithms is not difficult, but the process is not easy to explain fully without detailed mathematical analysis.

In 2D, vertical time, τ, is defined by Equation 4-13, where z is depth, and $v(x, z)$ is a two-dimensional velocity function.

(4-13)
$$\tau(z) = \int_0^z \frac{dz}{v(x, z)}$$

There is, of course, a 3D version of this equation, but for simplicity we will only consider the 2D case.

With this as our variable of choice, the two-dimensional equation governing our wavefield takes the form of Equation 4-14, which, on the surface, appears to be little different from the depth equation on which all modeling and depth imaging is based.

(4-14)
$$\frac{\partial^2 u}{\partial \tau^2} + \frac{\partial^2 u}{\partial x^2} = \frac{1}{v^2}\frac{\partial^2 u}{\partial t^2}$$

The problem is that this equation has two time variables. One, t, is the actual time governing wavefield propagation, and it is what we record in the form of arrival times. The other, τ, is simply the conversion of vertical depth to time. The key point to all of this is that wavefields do not travel in vertical time, they propagate in space. Therefore, the problem is how can we solve Equation 4-14 for the image at time τ.

Jon Claerbout (1976) found a unique solution to this problem. He first recognized that vertically propagating solutions of this equation are constant along level lines, $t + \tau = constant$. He then made a change of variables from t to $t' = t + \tau$, and then ignored all higher order derivatives. The result is

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{v^2}\frac{\partial u}{\partial t'}\frac{\partial u}{\partial \tau} \tag{4-15}$$

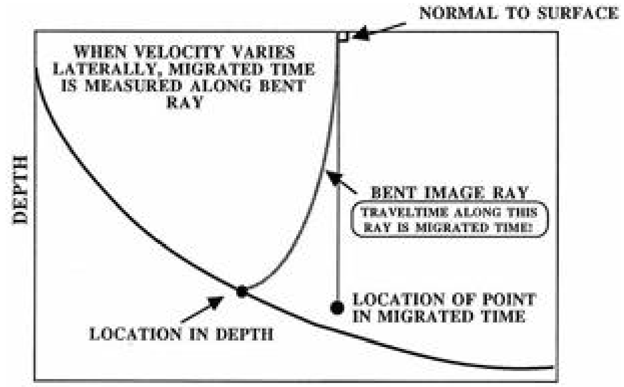
This equation can be solved discretely quite easily. The transformation from Equation 4-14 to 4-15 is not unlike the factorization into one-way equations in the Modeling chapter (Chapter 2), and so virtually all of the concepts and ideas incorporated in depth migration algorithms are also part and parcel of the time migration scene. In the same manner that factoring the full two-way equation into two one-way equations theoretically limited propagation to no more than 90 degrees, approximating Equation 4-14, another easily solvable equation, also reduces the dip response of any image.

Perhaps the only significant difference between the one-way depth approach and the one-way time migration approach is that the latter is almost always cast as an implicit scheme. Because the required matrix inversion is relatively simple, this does not result in a computationally intense algorithm. Thus, time migration as practiced in the past is quite efficient.

The Major Difference between Time and Depth Migration

The major difference between time and depth migration is directly related to the transformation from depth, z, to vertical time, τ. This difference is illustrated in Figure 4-13, which provides a graphic of what happens when the velocity varies in any direction. Depth migration accurately places the imaged event at its image ray time location, while time migration continues to place the migrated event at vertical time. Large lateral placement errors are the rule when the velocity varies strongly. This is a major reason to prefer depth migration over pure time-migration approaches, but it is not the only reason. Ray bending is also extremely limited because time migrations utilize RMS velocities instead of true interval velocities.

Figure 4-13. The major difference between time and depth.

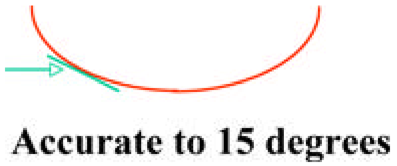

NORMAL TO SURFACE
WHEN VELOCITY VARIES LATERALLY, MIGRATED TIME IS MEASURED ALONG BENT RAY
DEPTH
BENT IMAGE RAY
TRAVELTIME ALONG THIS RAY IS MIGRATED TIME!
LOCATION OF POINT IN MIGRATED TIME
LOCATION IN DEPTH

Dip Limits

As we saw in the chapter on modeling, factoring the full two-way equation limited one-way propagation to 90 degrees, since almost any implementation of a one-way method decreases the dip response. In depth migration, the basic problem arises from the need to approximate the square-root term in the one-way equation. The approximation is usually the result of truncating a Taylor series, but it can also result from other approximations.

In vertical-time migrations, an entirely different equation replaces the fundamental wave equation, but its approximation is the result of truncation and also results in a decreased dip response. This decrease is clearly evident on any comparison between the algorithmic and exact impulse responses. It is not surprising that the recognized definition of dip limits was based on where the impulse response breaks down. Figure 4-14 shows how dip limits are defined.

Figure 4-14. Algorithmic dip limit definitions.

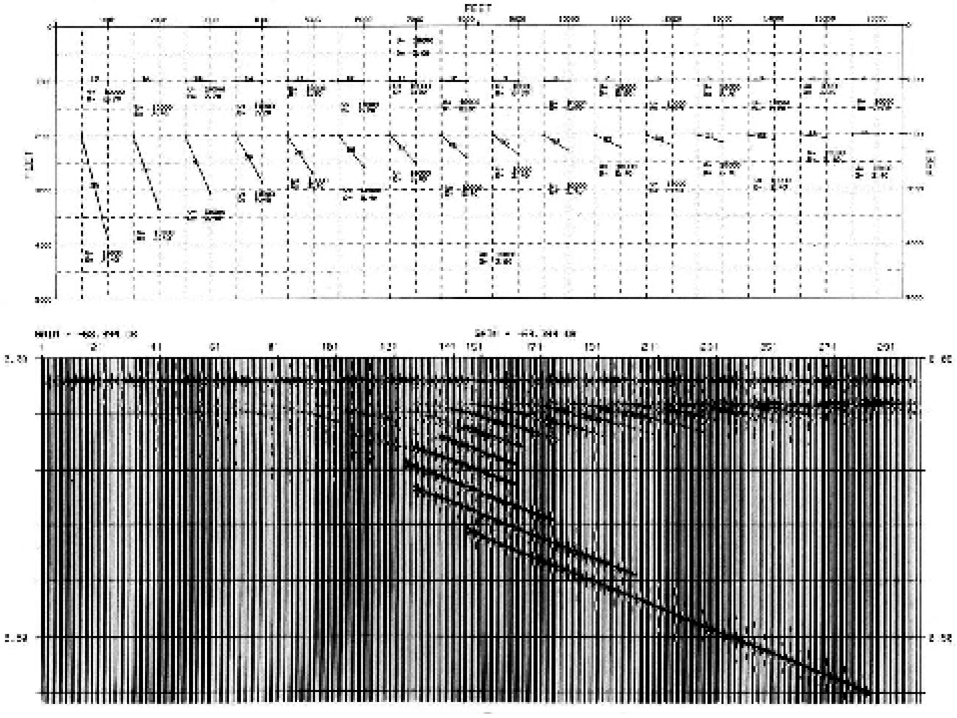


Using this definition, it is possible to show that the dip limit of early time migration algorithms did not exceed 15 degrees. Consequently, they were called 15 degree algorithms. Increasing the approximation accuracy by using additional terms in series approximations produced algorithms with 45 and even in some cases 60 degree limits. Using three or more terms ultimately resulted in diminishing returns and there was an upper limit of around 80 degrees. Although there are many different domains of application, the process almost always revolves around some type of series expansion of either the square root term or its time-domain equivalent.

One-Way XT Time Migration

The simple model shown at the top of Figure 4-15 was designed to test the basic concepts of zero-offset seismic migration or imaging. The model consists of 16 flat and 16 dipping events in a constant velocity medium, where the first dipping event is actually flat. Since the medium has constant velocity, there is absolutely no difference between running a time or a depth migration algorithm. We will test our first migration by running one-way time migration algorithms that are limited to 15 and 45 degree dip responses. The idea is to see just how good of a job we can do with a more or less naive approach.

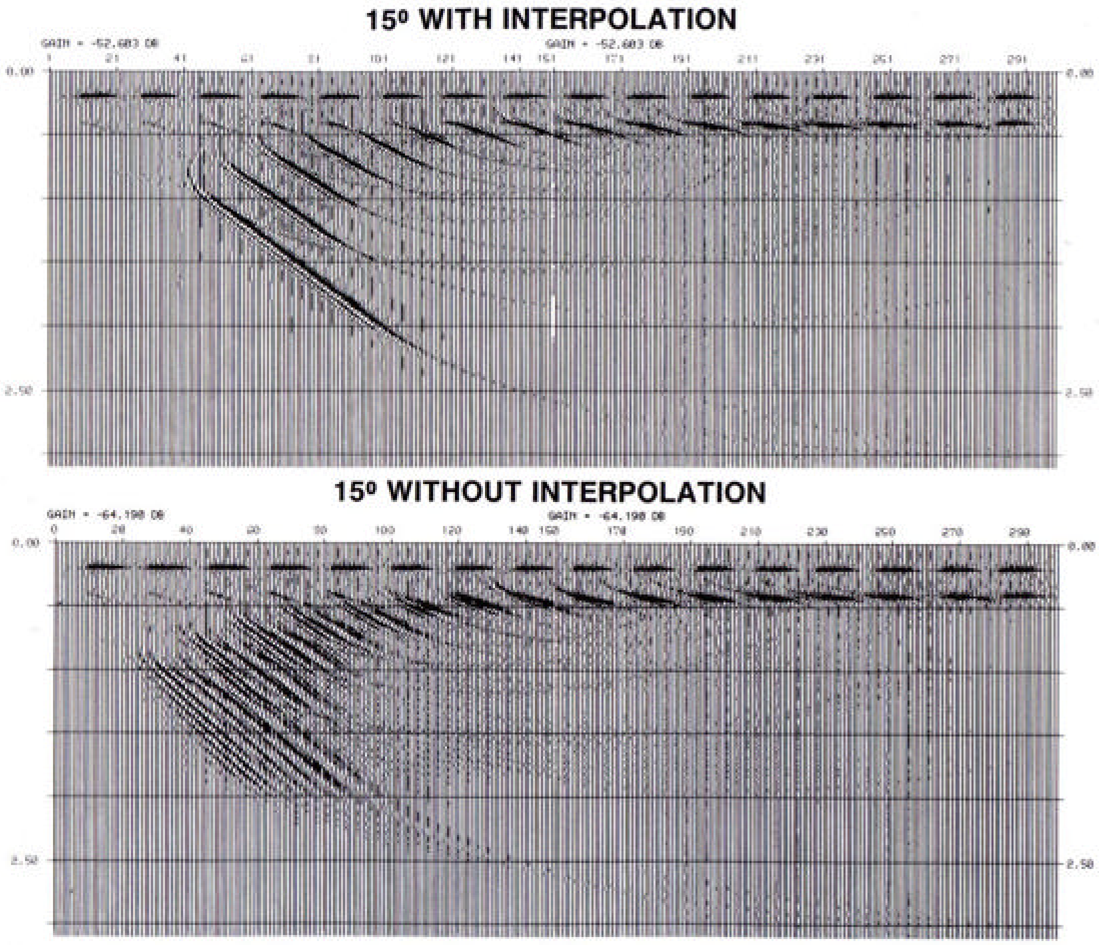
Figure 4-15. Simple constant velocity model with dipping events from 0 to 70 degrees.



The data shown at the bottom of Figure 4-15 demonstrates several of the usual rules of thumb concerning seismic data. In this case, since the data was synthesized numerically, it is pure zero-offset data; that is, there is no separation between the source and receiver used to synthesize it.

The 15 degree approach shown in Figure 4-16 exhibits one of the most difficult aspects of migration. Imaging steeply dipping events is very much a function of the accuracy of the approximations used to make the algorithm workable.

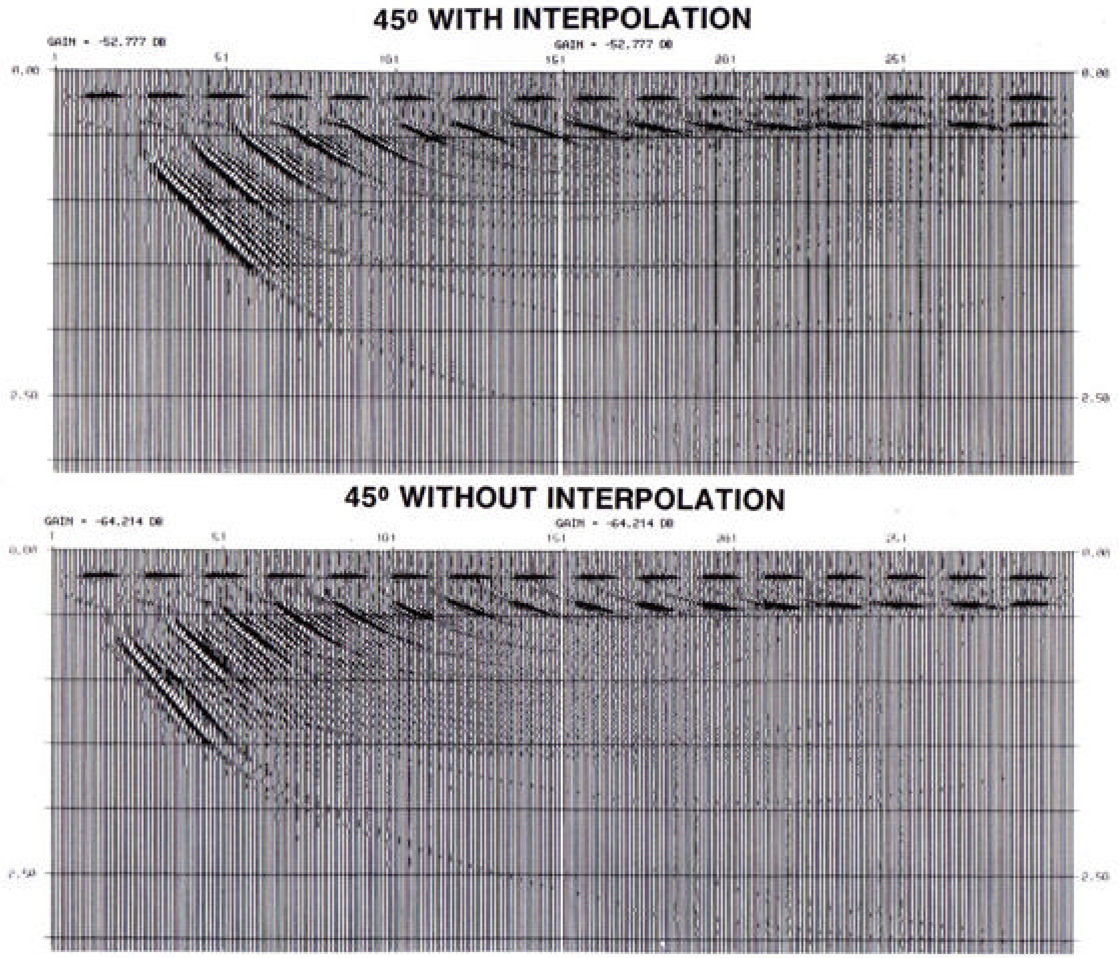
Figure 4-16. Application of a 15 degree equation and the impact of interpolation.



In the lower half of this figure, we see under-migrated events. Visual inspection reveals that the highest correct dip on the section is approximately 20 degrees. This is a bit higher than the 15 degrees the model study suggests is possible, but well within the expected error of the method. The bottom of this figure also demonstrates something called *grid dispersion*. This phenomenon is a complex part of the algorithm's implementations, but is easily handled by interpolating to a finer sample interval. The top half of the figure demonstrates that although the grid dispersion can be removed quite easily, the improper placement of the steep dips is still apparent.

If we move to a more accurate algorithm, in this case a 45 degree approximation, as shown in Figure 4-17, we see that we can easily achieve proper placement up to approximately 55 degrees. In this case, grid dispersion is not as much of a problem, even though it is still present. The more accurate implementation has achieved much better overall results.

Figure 4-17. Application of a 45 degree equation and the impact of interpolation.



One-Way FK Time Migration

One-way FK time migration can be separated into a phase shift migration method and a phase shift plus interpolation (PSPI) method.

Phase Shift Migration

Phase shift migration in time is actually based on an approximation of the transformed wave equation, Equation 4-15. However, the process of down shifting the surface exploding reflector data is given by Equation 4-16, where φ is defined by Equation 4-17. Variable v_{rms} is the RMS velocity between τ and $\tau + \Delta\tau$, and is virtually identical to its depth counterpart given by Equation 2-127.

$$U(k_x, \tau + \Delta\tau) = \exp\left(-\imath\omega\varphi\Delta\tau\right) U(k_x, \omega) \tag{4-16}$$

$$\varphi = k_x^2\left(\frac{v_{rms}}{\omega}\right)^2 \tag{4-17}$$

Phase Shift Plus Interpolation

Phase-shift-plus-interpolation (PSPI) time migration is identical to the depth migration explained in Phase Shift Plus Interpolation Migration on page 139. The only noticeable difference is that the down shift takes place in migrated time rather than migrated or imaged depth. Multiple velocities can be phase shifted to form the basis of an interpolation scheme in either space-time or frequency-space to make sure that the process is accurate, effective, and fast.

Kirchhoff Style Time Migration

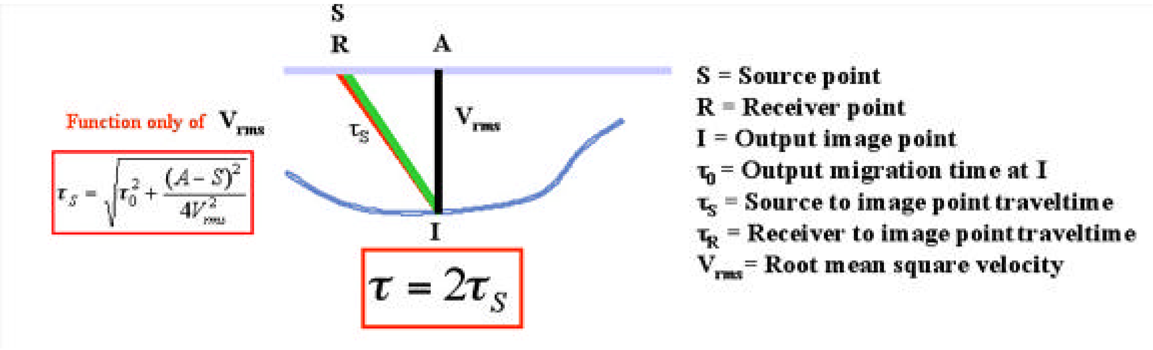
Kirchhoff style time migration can be separated into straight-ray Kirchhoff time migration and curved-ray Kirchhoff time migration.

Straight-Ray Kirchhoff Time Migration

Straight ray Kirchhoff migration does not really do what the name implies. The method is patterned on Figure 3-17 and the RMS velocity as defined in Equation 3-4.

As shown in Figure 4-18, the method selects an RMS velocity function at the output image location, I, and then uses it in the traditional traveltime Equation 4-18 to compute the required traveltime from the source to the image point and back to the coincident receiver. Once this time, $2\tau_S$, is available, the method selects the amplitude from the trace at surface location S, and adds the output to the location A at time τ.

Figure 4-18. Straight ray Kirchhoff Migration.


S
R
A
Function only of V_{rms}
$\tau_S = \sqrt{\tau_0^2 + \frac{(A-S)^2}{4V_{rms}^2}}$
τ_S
V_{rms}
I
$\tau = 2\tau_S$
S = Source point
R = Receiver point
I = Output image point
τ_0 = Output migration time at I
τ_S = Source to image point traveltime
τ_R = Receiver to image point traveltime
V_{rms} = Root mean square velocity

The blue line indicates all image points with the same output migration time. In a constant velocity medium, this equal-traveltime curve is actually a circle and represents the set of points which are equally likely points from which energy might be reflected. This process was the *de facto* Kirchhoff algorithm for many years and was one of the first to be computerized.

Curved-Ray Kirchhoff Time Migration

Equation 4-18 is the traditional traveltime formula for the time between a source at a distance from a surface point, m, and an image point located directly below m with a vertical traveltime of τ_0.

(4-18)
$$\tau = \sqrt{\tau_0^2 + \frac{h^2}{4v^2}}$$

This is a truncation of a series of the form shown in Equation 4-19, where $c_1 = \frac{1}{v^2}$ is the reciprocal of the RMS velocity, v. As shown in Figure 4-19, the rest of the c_i values are complicated modes of the interval velocity function, $v(i\Delta z)$.

(4-19)
$$\tau^2 = \tau_0^2 + c_1(m-S)^2 - c_2(m-S)^4 + c_3(m-S)^6 - \ldots$$

Figure 4-19. Curved ray traveltimes.

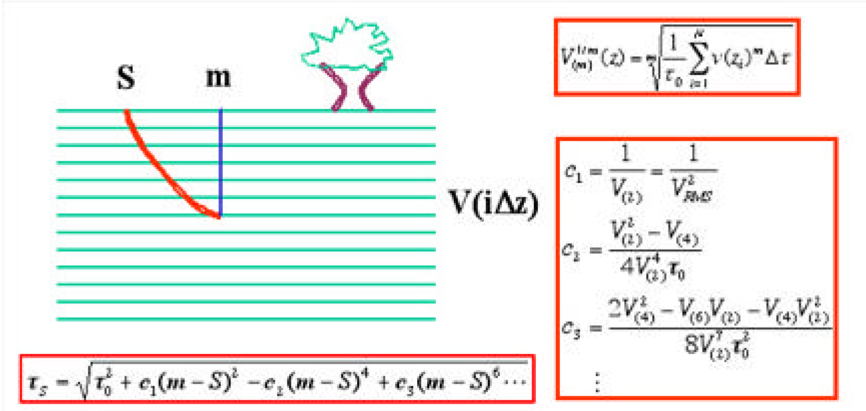
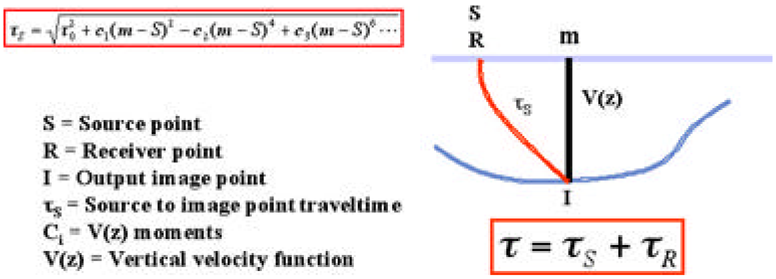

S
m
V(iΔz)
$V_{(m)}^{1/m}(z) = \sqrt[m]{\frac{1}{\tau_0}\sum_{i=1}^{N} v(z_i)^m \Delta\tau}$
$c_1 = \frac{1}{V_{(2)}} = \frac{1}{V_{RMS}^2}$
$c_2 = \frac{V_{(2)}^2 - V_{(4)}}{4V_{(2)}^4 \tau_0}$
$c_3 = \frac{2V_{(4)}^2 - V_{(6)}V_{(2)} - V_{(4)}V_{(2)}^2}{8V_{(2)}^7 \tau_0^2}$
$\vdots$
$\tau_S = \sqrt{\tau_0^2 + c_1(m-S)^2 - c_2(m-S)^4 + c_3(m-S)^6 \cdots}$

Figure 4-20 shows how curved ray Kirchhoff migration works using the full series representation shown in Figure 4-19.

Figure 4-20. Curved ray Kirchhoff time migration.

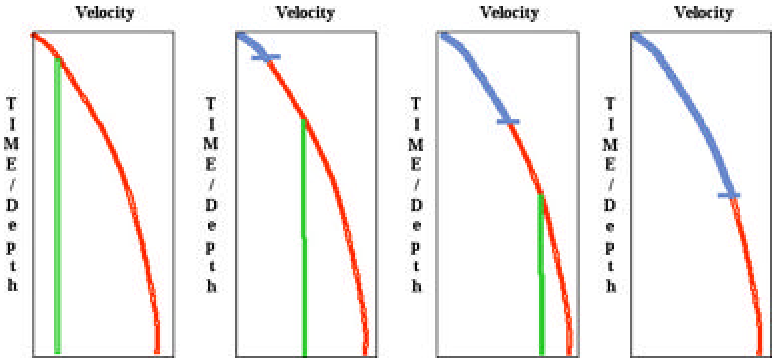

$\tau_S = \sqrt{\tau_0^2 + c_1(m-S)^2 - c_2(m-S)^4 + c_3(m-S)^6 \cdots}$
S = Source point
R = Receiver point
I = Output image point
τ_S = Source to image point traveltime
C_i = V(z) moments
V(z) = Vertical velocity function
S
R
m
τ_S
V(z)
I
$\tau = \tau_S + \tau_R$

Calculating the traveltime through a velocity medium that varies only in the vertical direction ensures that the migration is identical to those that would be obtained using a raytracer. In effect, a curved ray time migration is identical to performing a depth migration using a different $v(z)$ for each output location and then outputting the result in vertical time.

Cascaded Migration

Figure 4-21 shows how migration algorithms can be cascaded to improve the overall response of the final migration.

Figure 4-21. Cascaded Migration.



In some cases, the combination of several poor migration techniques produces a migration that is superior to each individual migration methodology. Each successive migration takes place with a migration field that is constructed from a small portion of the final velocity field and a constant velocity. Usually, the piece of the final field is chosen so as not to have any strong variations. After migrating with the constructed field, the next migration begins where the last one left off. That is, the new time zero is defined by the time at which the constant velocity functions intersect that portion of the final velocity field used during this stage of the cascade.

Almost any migration algorithm can be cascaded. The basic problem is that cascaded migrations are theoretically correct only when the cascade is performed after a constant velocity migration. This, in effect, means that the cascade concept can only be applied effectively after a completely straight-ray time migration.

Migration Summary

There is no question that two-way finite difference methods offer the potential for the most accurate images any migration algorithm can produce. If we know the exact subsurface wavefield at the last recorded time, the output from this technique provides an exact answer. It images all events from which reflections occur at precisely the correct reflectivity. Consequently, it is the ultimate goal of all migration algorithm developers. Its only algorithmic problems are its extreme computational requirements and that it can produce unusual artifacts that are difficult to explain when incorrect Earth models are used.

However, full two-way reverse time migration has no velocity sensitivities, it has no dip limitations, and, in the prestack sense, it is the only approach that accurately handles all amplitude issues.

Phase-shift-plus-interpolation (PSPI) represents a very simple extension to the pure phase-shift approach. It uses the phase shift algorithm with multiple constant velocities, and then interpolates as needed to achieve the proper image at each image point on the current depth or time slice. It was very likely the first FK-style method that was able to at least partially remove both the $v(z)$ assumption and still retain a reasonable dip response. The quality of PSPI algorithms is still a function of the accuracy of the implementation, but, nevertheless, most such algorithms are quite good. Like its phase-shift counterpart, it can be extended to include some full two-way propagation. Because the interpolation step in PSPI can be somewhat difficult, and because each application of the phase shift method adds to the overall cost, alternatives to the method have been sought.

Split-step methods attempt to avoid the interpolation step of PSPI methods through a different approximation to the underlying wave-equation. In effect, this approach was one of the first to use a dual domain approach. The “shift” is accomplished in the FK domain, while the modification of the interpolation step is accomplished in the FX domain. When this method was published, it was thought to provide an improved approach to PSPI, but this did not prove to be the case.

Generalized phase screens are really split-step algorithms with additional terms to increase the overall dip-response and improve accuracy. The key difference between a split-step algorithm and a phase screen method is that the phase screen methods have additional correction terms in (F, X) space, and, as a result, should produce a more accurate, less sensitive algorithm when properly implemented. Again, like the original phase shift method, generalized phase screens can be modified to include some forms of two-way propagation.