downward continued to a depth $\Delta z$ by multiplying by

$$e^{i\,k_z \Delta z} \quad = \quad \exp\left\{ -\,i\,\frac{\omega}{v}\,\sqrt{1\;-\;\left(\frac{v\,k_x}{\omega}\right)^2}\;\Delta z \right\} \tag{3.20}$$

Ordinarily the time-sample interval $\Delta\tau$ for the output-migrated section is chosen equal to the time-sample rate of the input data (often 4 milliseconds). Thus, choosing the depth $\Delta z = v\Delta\tau$, the downward-extrapolation operator for a single time unit is

$$C \quad = \quad \exp\left\{ -\,i\,\omega\;\Delta\tau\,\sqrt{1\;-\;\left(\frac{v\,k_x}{\omega}\right)^2} \right\} \tag{3.21}$$

Data will be multiplied many times by $C$, thereby downward continuing it by many steps of $\Delta\tau$.

Next is the task of imaging. At each depth an inverse Fourier transform is followed by selection of its value at $t = 0$. (Reflectors explode at $t = 0$). Luckily, only the Fourier transform at one point, $t = 0$, is needed, so that is all that need be computed. The computation is especially easy since the value at $t = 0$ is merely a summation of each $\omega$ frequency component. (This may be seen by substituting $t = 0$ into the inverse Fourier integral). Finally, inverse Fourier transform $k_x$ to $x$. The migration process, computing the image from the upcoming wave $u$, may be summarized in the following pseudo code:

$U(\omega, k_x, \tau = 0) = FT[u(t, x)]$
For $\tau = \triangle\tau, 2\triangle\tau, \ldots,$ end of time axis on seismogram
    For all $k_x$
        For all $\omega$
            $C =$exp$(-i\omega\triangle\tau\sqrt{1 - v^2k_x^2/\omega^2}$
            $U(\omega, k_x, \tau) = U(\omega, k_x, \tau - \Delta\tau) * C$
    For all $k_x$
        Image$(k_x, \tau) = 0.$
        For all $\omega$
            Image$(k_x, \tau) =$Image$(k_x, \tau) + U(\omega, k_x, \tau)$
    image$(x, \tau) = FT[$Image$(k_x, \tau)]$

This pseudo code Fourier transforms a wavefield observed at the earth's surface $\tau = 0$, and then it marches that wavefield down into the earth ($\tau > 0$) filling up a three-dimensional function, $U(\omega, k_x, \tau)$. Then it selects $t = 0$, the time of the exploding reflectors by summing over all frequencies $\omega$. (Mathematically, this resembles finding the signal at $\omega = 0$ by summing over all $t$).

Turning from pseudocode to real code, an important practical reality is that computer memories are not big enough for the three-dimensional function $U(\omega, k_x, \tau)$. But it is easy to intertwine the downward continuation with the summation over $\omega$ so a three-dimensional function need not be kept in memory. This is done in the real code in subroutine `phasemig()`.

```
subroutine phasemig( u, nt, nx, dt, dx, image, ntau, dtau, v)
integer nt,nw,nx,ntau, iw,ikx,itau
real    dt,dx, w,w0,dw, kx,kx0,dkx,dtau, kzkz, v, sig1,sig2,pi, signum
complex u(nt,nx), image(ntau,nx), c
                    pi = 3.14159265;        sig1 =+1.;  sig2 =-1.
call ft1axis( 0, sig1, nt, nx, u)
call ft2axis( 0, sig2, nt, nx, u)

w0  = -pi/dt;   dw = 2.*pi/(nt*dt);     nw = nt
kx0 = -pi/dx;   dkx= 2.*pi/(nx*dx)

call zero( ntau*nx, image)
do itau = 1, ntau {
do ikx  = 2, nx   {        kx = kx0 + (ikx-1) * dkx
do iw = 2, nw     {        w  = w0  + (iw-1)  * dw
            kzkz = w*w - v*v * kx*kx
            if (kzkz >0) { c = cexp(cmplx(0., -signum(w)*dtau*sqrt(kzkz))) }
            else         { c = 0.                                          }
            u(iw,ikx) = u(iw,ikx) * c
            image(itau,ikx) = image(itau,ikx) + u(iw,ikx)
            }}}
call ft2axis( 1, sig2, ntau, nx, image)
return; end
```

An aspect of the computation that was hidden in the pseudo code that you can see in the real code is that we must also handle negative frequencies. When doing so, we use the signum function $\mathrm{sgn}(\omega) = \omega/|\omega|$ to choose the sign of $k_z$ so that the sign of $k_z$ is always opposite that of $\omega$.

Inverse migration (modeling) proceeds in much the same way. Beginning from an upcoming wave that is zero at great depth, the wave is marched upward in steps by multiplication with $\exp(i\, k_z\, \Delta\, z)$. As each level in the earth is passed, exploding reflectors from that level are added into the upcoming wave. Pseudo code for modeling the upcoming wave $u$ is

$$
\begin{aligned}
&\text{Image}(k_x, z) = FT[\text{image}(x, z)] \\
&\text{For all } \omega \text{ and all } k_x \\
&\quad U(\omega, k_x) = 0. \\
&\text{For all } \omega \ \{ \\
&\text{For all } k_x \ \{ \\
&\text{For } z = z_{\max}, z_{\max} - \triangle z, z_{\max} - 2\triangle z, \ \ldots, \ 0 \ \{ \\
&\quad C = \exp(+i \triangle z \omega \sqrt{v^{-2} - {k_x}^2/\omega^2} \\
&\quad U(\omega, k_x) = U(\omega, k_x) * C \\
&\quad U(\omega, k_x) = U(\omega, k_x) + \text{Image}(k_x, z) \\
&\quad \} \ \} \ \} \\
&u(t, x) = FT[U(\omega, k_x)]
\end{aligned}
$$

Some real code for this job is in subroutine `phasemod()`.

```
subroutine phasemod( image, nz, nx, dz, dx, u, nt, dt, v)
integer nt,nw,nx,nz, iw,ikx,iz
real    dt,dx,dz, w,w0,dw, kx,kx0,dkx, kzkz, v, sig1,sig2,pi, signum
complex u(nt,nx), image(nz,nx), c
                pi = 3.14159265;        sig1=+1.;       sig2=-1.
call ft2axis( 0, sig2, nz, nx, image)

w0  = -pi/dt;   dw = 2.*pi/(nt*dt);     nw = nt
kx0 = -pi/dx;   dkx= 2.*pi/(nx*dx)
call zero( nw*nx, u)
do iw  = 2, nw     {  w  = w0  + (iw-1)  * dw
do ikx = 2, nx     {  kx = kx0 + (ikx-1) * dkx
do iz  = nz, 1, -1 {
            kzkz = w*w - (v*v) * kx*kx
            if (kzkz >0) { c = cexp(cmplx(0., signum(w)*dz*sqrt(kzkz))) }
            else         { c = 0.                                       }
            u(iw,ikx) = u(iw,ikx) * c
            u(iw,ikx) = u(iw,ikx)   +  image(iz,ikx)
            }}}
call ft1axis( 1, sig1, nt, nx, u)
call ft2axis( 1, sig2, nt, nx, u)
return; end
```

The positive sign in the complex exponential is a combination of two negatives, the *up* coming wave and the *up*ward extrapolation. The three loops on $\omega$, $k_x$, and $z$ are interchangeable. When the velocity $v$ is a constant function of depth the program can be speeded by moving the computation of the complex exponential $C$ out of the inner loop on $z$.

## 3.4 THE STOLT METHOD

On most computers the Stolt [1978] method of migration is the fastest one—by a wide margin. For many applications, this will be its most important attribute. For

a constant-velocity earth it incorporates the Huygens wave source exactly correctly. Like the other methods, this migration method can be reversed and made into a modeling program. One drawback, a matter of principle, is that the Stolt method does not handle depth variation in velocity. This drawback is largely offset in practice by an approximate correction that uses an axis-stretching procedure. A practical problem is the periodicity of all the Fourier transforms. In principle this is no problem at all, since it can be solved by adequately surrounding the data by zeroes.

A single line sketch of the Stolt method is this:

$$p(x,t) \quad \rightarrow \quad P(k_x,\omega) \quad \rightarrow \quad P'(k_x,k_z \;=\; \sqrt{\omega^2/v^2 \;-\; k_x^2}) \quad \rightarrow \quad p'(x,z)$$

To see why this works, begin with the input-output relation for downward extrapolation of wavefields:

$$P(\omega,k_x,z) \quad = \quad e^{i\,k_z z}\; P(\omega,k_x,z=0) \tag{3.22}$$

Perform a two-dimensional inverse Fourier transform:

$$p(t,x,z) \quad = \quad \int\int \; e^{i\,k_x x - i\omega t + i\,k_z z}\; P(\omega,k_x,0)\; d\omega\; dk_x \tag{3.23}$$

Apply the idea that the image at $(x,z)$ is the exploding-reflector wave at time $t=0$:

$$\text{Image}(x,z) \quad = \quad \int\int \; e^{i\,k_x x}\; e^{i\,k_z(\omega,k_x)\,z}\; P(\omega,k_x,0)\; d\omega\; dk_x \tag{3.24}$$

Equation (3.24) gives the final image, but it is in an unattractive form, since it implies that a two-dimensional integration must be done for each and every $z$-level. The Stolt procedure converts the three-dimensional calculation thus implied by (3.24) to a single two-dimensional Fourier transform.

So far nothing has been done to specify an *upcoming* wave instead of a downgoing wave. The direction of the wave is defined by the relationship of $z$ and $t$ that is required to keep the phase constant in the expression $\exp(-i\omega t \;+\; ik_z z)$. If $\omega$ were always positive, then $+k_z$ would always refer to a downgoing wave and $-k_z$ to an upcoming wave. Negative frequencies $\omega$ as well as positive frequencies are needed to describe waves that have real (not complex) values. So the proper description for a downgoing wave is that the signs of $\omega$ and $k_z$ must be the same. The proper description for an upcoming wave is the reverse. With this clarification the integration variable in (3.24) will be changed from $\omega$ to $k_z$.

$$\omega \quad = \quad -\,\text{sgn}(k_z)\; v\sqrt{k_x^2 \;+\; k_z^2} \tag{3.25}$$

$$\frac{d\omega}{dk_z} \quad = \quad -\,\text{sgn}(k_z)\; v\; \frac{k_z}{\sqrt{k_x^2 \;+\; k_z^2}} \tag{3.26}$$

$$\frac{d\omega}{dk_z} \quad = \quad \frac{-\,v|k_z|}{\sqrt{k_x^2 \;+\; k_z^2}} \tag{3.27}$$

Put (3.25), (3.26) and (3.27) into (3.24), and include also a minus sign so that the integration on $k_z$ goes from minus infinity to plus infinity as was the integration on $\omega$.

$$\text{Image}(x,z) \quad = \quad \int\int e^{ik_z z + ik_x x} \left\{ P[\omega(k_x,k_z),k_x,0] \frac{v|k_z|}{\sqrt{k_x^2+k_z^2}} \right\} dk_z\, dk_x \qquad (3.28)$$

Equation (3.28) states the result as a two-dimensional inverse Fourier transform. The Stolt migration method is a direct implementation of (3.28). The steps of the algorithm are

- Double Fourier transform data from $p(t,x,0)$ to $P(\omega,k_x,0)$.
- Interpolate $P$ onto a new mesh so that it is a function of $k_x$ and $k_z$. Multiply $P$ by the scale factor (which has the interpretation $\cos\theta$).
- Inverse Fourier transform to $(x,z)$-space.

Samples of Stolt migration of impulses are shown in Figure 3.9. You can see the

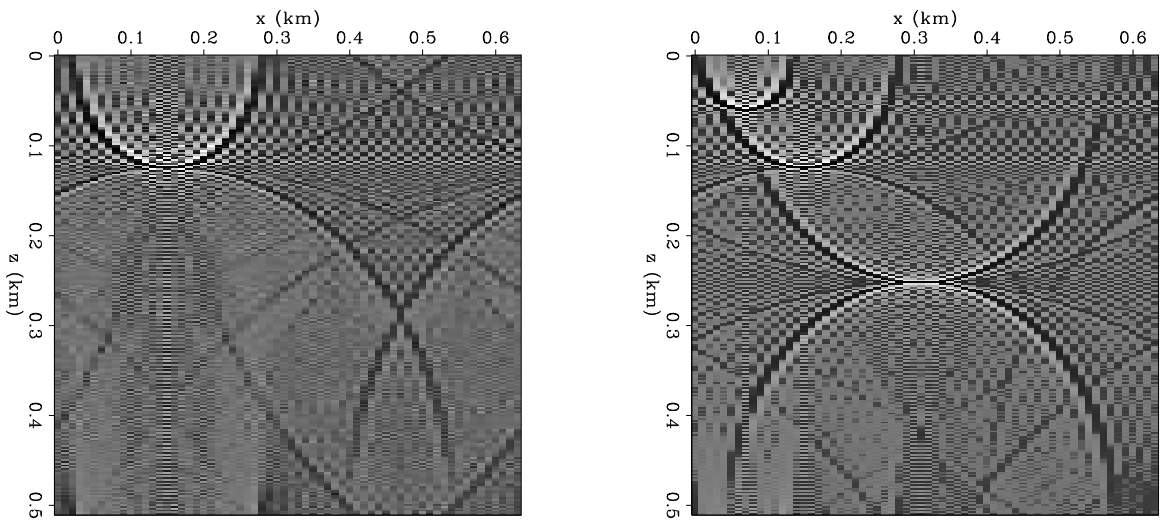



Figure 3.9: Response of Stolt method to data with impulses. Semicircles are seen, along with computation artifacts. omk-stolt [ER]

expected semicircular smiles. You can also see a semicircular frown hanging from the bottom of each semicircle. The worst frown is on the deepest spike. The semicircular mirrors have centers not only at the earth's surface $z = 0$ but also at the bottom of the model $z = z_{\max}$. It is known that these frowns can be suppressed by interpolating more carefully. (Interpolation is the way you convert from a uniform mesh in $\omega$, to a uniform mesh in $k_z$). Interpolate with say a *sinc* function instead of a linear interpolator. A simpler alternative is to stay away from the bottom of the model, i.e. pad with lots of zeroes.

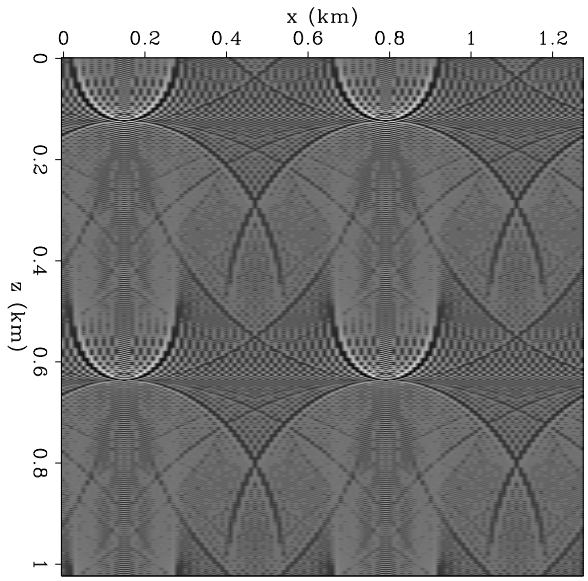



Figure 3.10: Periodicity of the output of the Stolt migration program. omk-stolt4 [ER]

It seems that an extraordinary amount of zero padding is required on the time axis. To keep memory requirements reasonable, the algorithm can be reorganized as described in an exercise. Naturally, the periodicity in $x$ also requires padding the $x$-axis with zeroes.

Below is a subroutine for Stolt migration. Its input is assumed to be data after 2-D Fourier transformation to $(\omega, k_x)$-space, and its output is the image in to $(k_z, k_x)$-space, so it needs a final transformation to $(z, x)$-space.

```
subroutine stoltmig( nt,nz,nx, dt,dz,dx, vel, cftdata, cftimage)
integer it,iz,ix, nt,nz,nx, iw,ikz,ikx, nw,nkz,nkx
real    dt,dz,dx, w0,kz0,kx0, dw,dkz,dkx, pi, vel,vhalf, w,kz,kx, signum
complex ckzkx, cftdata(nt,nx), cftimage(nz,nx)
pi = 3.1415926
nw  = nt;       w0  = -pi/dt;   dw  = 2.*pi/(nt*dt);
nkz = nz;       kz0 = -pi/dz;   dkz = 2.*pi/(nz*dz);
nkx = nx;       kx0 = -pi/dx;   dkx = 2.*pi/(nx*dx);
do ikz= 1, nkz {        cftimage( ikz,   1) = (0.,0.) } # zero the Nyquist.
do ikx= 1, nkx {        cftimage(   1, ikx) = (0.,0.) } # zero the Nyquist.
vhalf = 0.5 * vel
do ikx = 2, nkx {       kx = kx0 + dkx * (ikx-1)
do ikz = 2, nkz {       kz = kz0 + dkz * (ikz-1)
        w = - signum( kz) * vhalf * sqrt( kx*kx + kz*kz)
        call cinterp1( w, nw,w0,dw, cftdata(1,ikx), ckzkx)
        if( w != 0.)
                cftimage(ikz,ikx) = ckzkx * abs( kz / w)
        else
                cftimage(ikz,ikx) = 0.
        }}
return;         end
```

`stoltmig()` uses the linear interpolation routine `cinterp1()` (page 57) and the signum function $\mathrm{sgn}(x) = x/|x|$ done by routine `signum()` (page 57). In `stoltmig()` I have omitted the theoretically required $\cos\theta$ multiplication. This angle-dependent scaling can be left as an exercise. The effect of such an angle-dependent multiplier is

to scale the signal amplitude along the hyperbola according to the local slope of the hyperbola.

```
# Nearest neighbor interpolation would do this:    cbx = cb( 1.5 + (x-x0)/dx)
# Linear interpolation with the same definitions of variables is:
#
subroutine cinterp1( x, nx,x0,dx, cb, cbx)
integer ix,ixc,         nx
real    x, xc, x0, dx, fraction
complex cb(nx), cbx
xc      = (x-x0) / dx
ixc     =  xc
fraction = xc - ixc
ix      = 1 + ixc
if( ix < 1 )
                cbx = cb( 1)
else if( ix+1 > nx)
                cbx = cb( nx)
else
                cbx = (1.-fraction) * cb(ix)  +  fraction * cb(ix+1)
return; end


real function signum(x)
real x
        if      (x>0)  { signum =  1 }
        else if (x<0)  { signum = -1 }
        else           { signum =  0 }
return;end
```

## 3.5 INFLUENCES ON MIGRATION QUALITY

Several things can degrade of the quality of a migrated image. Some, such as spatial aliasing and velocity uncertainty affect all migration methods. Other quality differences stem from the migration method.

### 3.5.1 Spatial aliasing concept

*Spatial aliasing* means insufficient sampling of the data along the space axis. This difficulty is so universal, that all migration methods must consider it.

Data should be sampled at more than two points per wavelength. Otherwise the wave arrival direction becomes ambiguous. Figure 3.11 shows synthetic data that is sampled with insufficient density along the $x$-axis. You can see that the problem becomes more acute at high frequencies and steep dips.

There is no generally-accepted, automatic method for migrating spatially aliased data. In such cases, human beings may do better than machines, because of their skill in recognizing true slopes. When the data is adequately sampled, however, computer migration based on the wave equation gives better results than manual methods. Contemporary surveys are usually adequately sampled along the line of the survey, but there is often difficulty in the perpendicular direction.

The hyperbola-sum-type methods run the risk of the migration operator itself becoming spatially aliased. This should be avoided by careful implementation. The

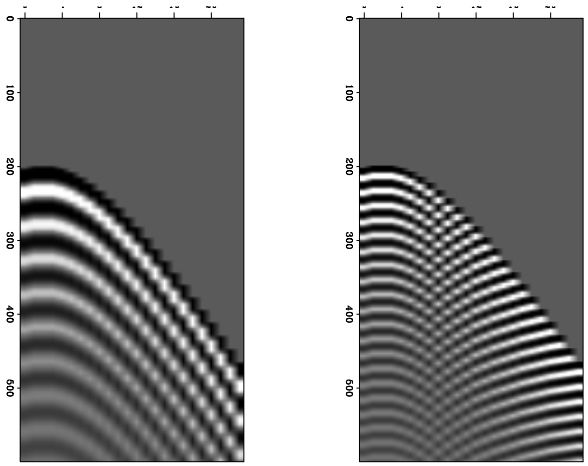


Figure 3.11: Insufficient spatial sampling of synthetic data. To better perceive the ambiguity of arrival angle, view the figures at a grazing angle from the side. omk-alias [ER]

first thing to realize is that you should be *integrating* along a hyperbolic trajectory. A summation incorporating only one point per trace is a poor approximation. It is better to incorporate more points, as depicted in Figure 3.12. The likelihood of getting

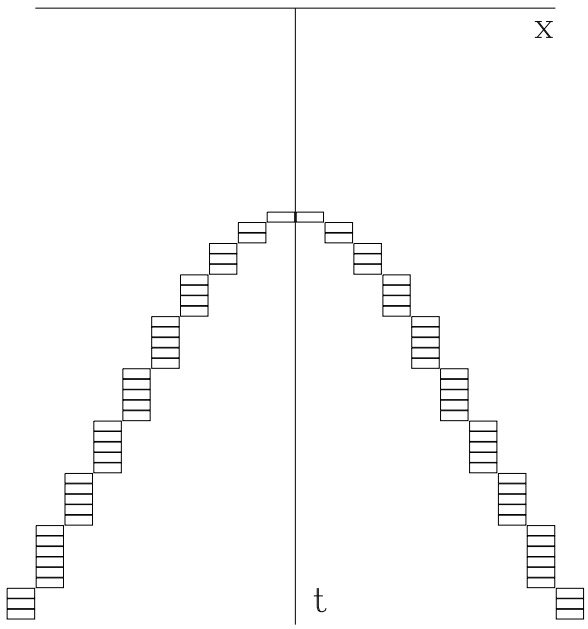



Figure 3.12: For a low-velocity hyperbola, integration will require more than one point per channel. omk-lineint [ER]

an aliased operator increases where the hyperbola is steeply sloped. In production examples an aliased operator often stands out above the sea-floor reflection, where—although the sea floor may be flat—it acquires a noisy precursor due to the steeply flanked hyperbola crossing the sea floor.

### 3.5.2 Sensitivity of migration to velocity error

Figure 3.13 shows how the migration impulse response depends on velocity. Recall

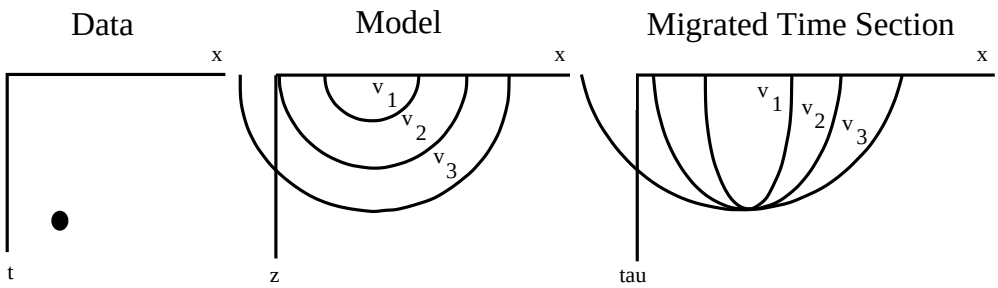



Figure 3.13: Velocity error sensitivity increases with angle up to 90°. Migration of a data impulse as a function of velocity. Three possible choices of constant velocity are shown superposed on one plane. omk-sensitive [NR]

that migrated data is ordinarily displayed as a *time* section. Arbitrary velocity error makes no difference for horizontal bedding.

Different people have different accuracy criteria. A reasonable criterion is that the positioning error of the energy in the semicircles should be less than a half-wavelength. For the energy moving horizontally, the positioning error is simply related to the dominant period $\Delta T$ and the travel time $T$. The ratio $T/\Delta T$ is rarely observed to exceed 100. This 100 seems to be a fundamental observational parameter of reflection seismology in sedimentary rock. (Theoretically, it might be related to the "$Q$" of sedimentary rock or it may relate to generation of chaotic internal multiple reflections. Larger values than 100 occur when (1) much of the path is in water or (2) at time depths greater than about 4 seconds). Figure 3.14 compares two nearby migration velocities. The separation of the curves increases with angle. For the separation to

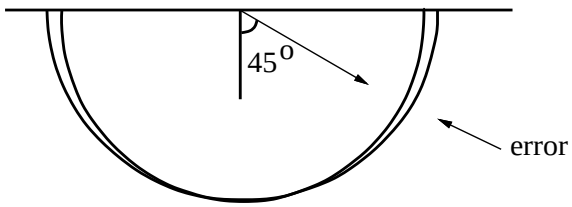



Figure 3.14: Timing error of the wrong velocity increases with angle. omk-senser [NR]

be less than a wavelength, for 90° dip the velocity error must be less than one part in 100. For 45° migration velocity error could be larger by $\sqrt{2}$.

Velocities are rarely known this accurately. So we may question the value of migration at wide angles.

| | Hyperbola Sum or Semicircle Superpose | Phase Shift | Stolt |
|---|---|---|---|
| $v(z)$ | ray tracing | easily | approximately by stretching |
| wide angle? | Beware of data alias and operator alias. | Beware of data alias. | Beware of data alias. |
| correct phase and obliquity? | possible with some effort for const $v$ | easy for any $v(z)$ | for const $v$ |
| wraparound noise? | no | on $x$ | on $x$ |
| $v(x)$ | Production programs have serious pitfalls. | approximately by iteration and interpolation | no known program |
| side boundaries and irregular spaceing | excellent | poor | poor |
| Speed | slow | average | very fast |

Table 3.1: Subjective comparison of three wide-angle migration methods.

### 3.5.3 Subjective comparison and evaluation of methods

The three basic methods of migration described in this section are compared subjectively in the following table.

The perspective of later chapters makes it possible to remark on the quality of the wide-angle methods as a group, and it is useful to do that now. Their greatest weakness is their difficulties with lateral velocity variation. Their greatest strength, wide-angle capability, is reduced by the weakness of other links in the data collection and processing chain, namely:

- Shot-to-geophone offset angles are often large but ignored. A CDP stack is not a zero-offset section.
- Why process to the very wide angles seen in the survey line when even tiny angles perpendicular to the line are being ignored?
- Data is often not sampled densely enough to represent steeply dipping data without aliasing.
- Accuracy in the knowledge of velocity is seldom enough to justify processing to wide angles.

- Noise eventually overpowers all echoes, say at time $t_{\max}$. Given a depth of interest, the maximum angle is $\cos(z/vt)$. For example, imagine oil reservoirs at a time depth of two seconds, where data recording stopped at four seconds. The implied angle cutoff is at 60°.

### 3.5.4 Evanescence and ground roll

Completing the physical derivation of the dispersion relation,

$$k_x^2 \; + \; k_z^2 \quad = \quad \frac{\omega^2}{v^2} \tag{3.29}$$

we can now have a new respect for it. It carries more meaning than could have been anticipated by the earlier geometrical derivation. The dispersion relation was originally regarded merely as $\sin^2\theta + \cos^2\theta = 1$ where $\sin\theta = vk_x/\omega$. There was no meaning in sin $\theta$ exceeding unity, in other words, in $v\,k_x$ exceeding $\omega$. Now there is. There was a hidden ambiguity in two of the previous migration methods. Since data could be an arbitrary function in the $(t, x)$-plane, its Fourier transform could be an arbitrary function in the $(\omega, k_x)$-plane. In practice then, there is always energy with an angle sine greater than one. This is depicted in Figure 3.15. What should be done with this energy?

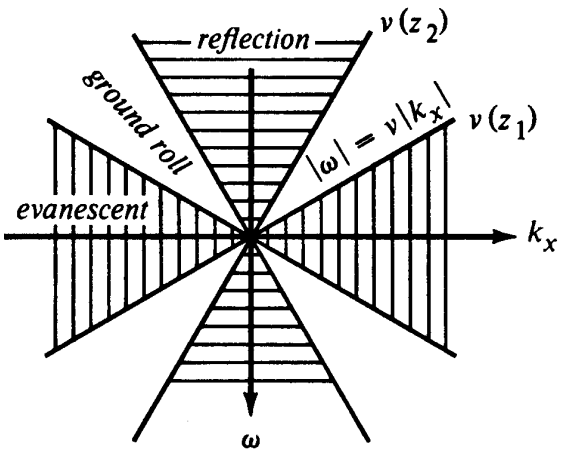



Figure 3.15: The triangle(s) of reflection energy $|\omega| > v(z)|k_x|$ become narrower with velocity, hence with depth. Ground roll is energy that is propagating at the surface, but evanescent at depth. omk-evtheory [NR]

When $v\,k_x$ exceeds $\omega$, the familiar downward-extrapolation expression is better rewritten as

$$e^{\pm\, i\sqrt{\omega^2/v^2 - k_x^2}\, z} \quad = \quad e^{\pm\sqrt{k_x^2 - \omega^2/v^2}\, z} \tag{3.30}$$

This says that the depth-dependence of the physical solution is a growing or a damped exponential. These solutions are termed *evanescent* waves. In the most extreme case, $\omega = 0$, $k_x$ is real, and $k_z = \pm ik_x$. For elastic waves, that would be the deformation of the ground under a parked airplane. Only if the airplane can move faster than the

speed of sound in the earth will a wave be radiated into the earth. If the airplane moves at a subsonic speed the deformation is said to be *quasi-static.*

Perhaps a better physical description is a thought experiment with a sinusoidally corrugated sheet. Such metallic sheet is sometimes used for roofs or garage doorways. The wavelength of the corrugation fixes $k_x$. Moving such a sheet past your ear at velocity $V$ you would hear a frequency of oscillation equal to $V\,k_x$, regardless of whether $V$ is larger or smaller than the speed of sound in air. But the sound you hear would get weaker exponentially with distance from the sheet unless it moved very fast, $V > v$, in which case the moving sheet would be radiating sound to great distances. This is why supersonic airplanes use so much fuel.

What should a migration program do with energy that moves slower than the sound speed? Theoretically, such energy should be exponentially damped in the direction going away from the source. The damping in the offending region of $(\omega, k_x)$-space is, quantitatively, extremely rapid. Thus, simple exploding-reflector theory predicts that there should be almost no energy in the data at these low velocities.

The reality is that, instead of tiny amounts of energy in the evanescent region of $(\omega, k_x)$-space, there is often a great deal. This is another breakdown of the exploding-reflector concept. The problem is worst with land data. Waves that are evanescent in deep, fast rocks of interest can be propagating in the low-velocity soil layer. This energy is called *ground roll.* Figure 3.16 shows an example. This data is not a zero-

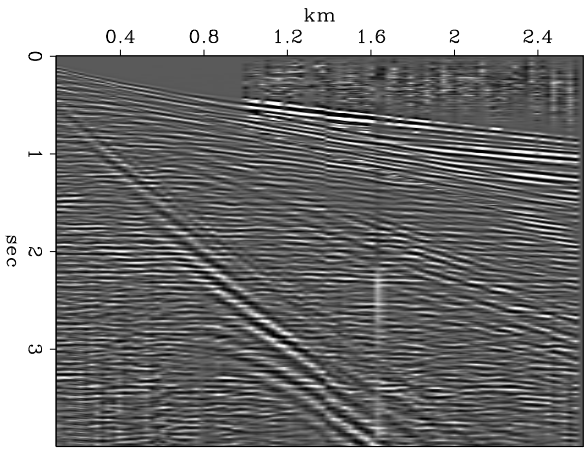



Figure 3.16: Florida shallow marine profile, exhibiting ground roll with frequency dispersion. (Conoco, Yedlin) omk-evdata [ER]

offset section. The shot is on the left, and the traces to the right are from geophones at increasing distances from the shot. First draw a line on the figure going through origin and at a slope of the water velocity, 1.5 km/sec. Steeper slopes are slower surface waves. Geological events have lesser slopes. Like the surface of the earth, which varies greatly from place to place, the immediate subsurface which controls the ground roll varies substantially. So although Figure 3.16 is a nice example, no example can really be typical. In this figure there are two types of ground roll, one at about half of water velocity, and a stronger one at about a quarter of water velocity. The later and stronger one shows an interesting feature known as *frequency dispersion.*

Viewing the data from the side, you should be able to notice that the high frequencies arrive before the low frequencies.

Ground roll is unwanted noise since its exponential decay effectively prevents it from being influenced by deep objects of interest. In practice, energy in the offending region of $(\omega, k_x)$-space should be attenuated. A mathematical description is to say that the composite mapping from model space to data space and back to model space again is not an identity transformation but an idempotent transformation.

EXERCISES:

1 Why does `phasemig()` (page 52) use a negative sign on $k_z$ whereas `phasemod()` (page 53) uses a positive sign even though both are dealing with *upcoming* waves?

2 The wave modeling program sketch assumes that the exploding reflectors are impulse functions of time. Modify the program sketch for wave modeling to include a source waveform $s(t)$.

3 The migration program sketch allows the velocity to vary with depth. However the program could be speeded considerably when the velocity is a constant function of depth. Show how this could be done.

4 Define the program sketch for the inverse to the Stolt algorithm—that is, create synthetic data from a given model.

5 The Stolt algorithm can be reorganized to reduce the memory requirement of zero padding the time axis. First Fourier transform $x$ to $k_x$. Then select, from the $(t, k_x)$-plane of data, vectors of constant $k_x$. Each vector can be moved into the space of a long vector, then zero padded and interpolated. Sketch the implied program.

6 Given seismic data that is cut off at four seconds, what is the deepest travel time depth from which 80° dips can be observed?

## 3.6 THE PARAXIAL WAVE EQUATION

The scalar wave equation, unlike Fourier equations, allows arbitrary spatial variation in density and velocity. Because of this you might expect that it would be used directly in the manufacture of migrated sections. But it is used little for migration, and we will first review why this is so. Then we will meet the *paraxial wave equation,* which is the basis for most production migration.

Philosophically, the paraxial wave equation is an intermediary between the simple concepts of rays and plane waves and deeper concepts embodied in the wave equation. (The paraxial wave equation is also called the *single-square-root* equation, or a *parabolic* wave equation). The derivation of the parabolic wave equation does not

proceed from simple concepts of classical physics. Its development is more circuitous, like the Schroedinger equation of quantum physics. You must study it for a while to see why it is needed. When I introduced the parabolic wave equation to seismic calculations in 1970, it met with considerable suspicion. Fortunately for you, years of experience have enabled me to do a better job of explaining it, and fortunately for me, its dominance of the industrial scene will give you the interest to persevere.

The paraxial equation will be introduced by means of Fourier methods. Fourier methods are incompatible with space-variable coefficients. Since we want to incorporate spatial variations in velocity, this limitation is ultimately to be avoided, so after getting the paraxial equation in the Fourier domain, $ik_z$ is replaced by $\partial/\partial z$, and $ik_x$ is replaced by $\partial/\partial x$. Then, being in the space domain, the velocity can be space-variable. The result is a partial differential equation often solved by the finite-differencing method. This procedure turns out to be valid, but new students of migration understandably regard it with misgiving. Thus, the final part of this section is a derivation of the paraxial wave equation which makes no use of Fourier methods.

### 3.6.1 Why the wave equation is rarely used for migration

Life would be simpler if migration could be done with the scalar wave equation instead of the paraxial equation. Indeed, migration can be done with the scalar wave equation, and there are some potential advantages (Hemon [1978], Kosloff and Baysal [1983]). But more than 99% of current industrial migration is done with the paraxial equation.

The main problem with the scalar wave equation is that it will generate unwanted internal multiple reflections. The exploding-reflector concept cannot deal with multiple reflections. Primary reflections can be modeled with only upcoming waves, but multiple reflections involve both up and downgoing paths. The multiple reflections observed in real life are completely different from those predicted by the exploding-reflector concept. For the sea-floor multiple reflection, a sea-floor two-way travel-time depth of $t_0$ yields sea-floor multiple reflections at times $2t_0,\ 3t_0,\ 4t_0, \ldots$. In the exploding-reflector conceptual model, a sea-floor one-way travel-time depth of $t_0$ yields sea-floor multiple reflections at times $3t_0,\ 5t_0,\ 7t_0, \ldots$. In building a telescope, microscope, or camera, the designer takes care to suppress backward reflected light because it creates background noise on the image. Likewise, in building a migration program we do not want to have energy moving around that does not contribute to the focused image. The scalar wave equation with space-variable coefficients will generate such energy. This unwanted energy is especially troublesome if it is coherent and migrates to a time when primaries are weak. It is annoying, as the reflection of a bright window seen on a television screen is annoying. So if you were trying to migrate with the scalar wave equation, you would make the velocity as smooth as possible.

Another difficulty of imaging with the scalar wave equation arises with evanescent waves. These are the waves that are exponentially growing or decaying with depth.

Nature extrapolates waves forward in time, but we are extrapolating them in depth. Growing exponentials can have tiny sources, even numerical round-off, and because they grow rapidly, some means must be found to suppress them.

A third difficulty of imaging with the scalar wave equation derives from initial conditions. The scalar wave equation has a second depth $z$-derivative. This means that two boundary conditions are required on the $z$-axis. Since data is recorded at $z = 0$, it seems natural that these boundary conditions should be knowledge of $P$ and $\partial P/\partial z$ at $z = 0$. But $\partial P/\partial z$ isn't recorded.

Luckily, in building an imaging device that operates wholly within a computer, we have ideal materials to work with, i.e., reflectionless lenses. Instead of the scalar wave equation of the real world we have the paraxial wave equation.

### 3.6.2 Fourier derivation of the paraxial wave equation

Start from the dispersion relation of the scalar wave equation:

$$k_x^2 \quad + \quad k_z^2 \quad = \quad \frac{\omega^2}{v^2} \tag{3.31}$$

Take a square root.

$$k_z \quad = \quad \pm \sqrt{\frac{\omega^2}{v^2} \quad - \quad k_x^2} \tag{3.32}$$

The simple act of selecting the minus sign in (3.32) includes the upcoming waves and eliminates the downgoing waves. Equation (3.31) is the three-dimensional Fourier transform of the scalar wave equation. Inverse transforming (3.32) will give us an equation for upcoming (or downgoing) waves only, without the other. Inverse Fourier transformation over a dimension is just a matter of selecting one or more of the following substitutions:

$$\frac{\partial}{\partial t} \quad = \quad -\, i\,\omega \tag{3.33}$$

$$\frac{\partial}{\partial x} \quad = \quad i\,k_x \tag{3.34}$$

$$\frac{\partial}{\partial z} \quad = \quad i\,k_z \tag{3.35}$$

After inverse transformation over $z$ there is a differential equation in $z$ in which the velocity may be taken to be $z$-variable. Likewise for $x$. Any equation resulting from any of the substitutions of (3.33), (3.34) and (3.35) into (3.32) is called a paraxial equation. Here we will go into great detail about the meaning of these equations. Before beginning this interpretation the paraxial wave equation will be derived without the use of Fourier transformation. Besides giving a clear path to the basic migration equation, this derivation also gives a better understanding of what the equation really does, and how it differs from the scalar wave equation.

### 3.6.3 Snell waves

It is natural to begin studies of waves with equations that describe plane waves in a medium of constant velocity. However, in reflection seismic surveys the velocity contrast between shallowest and deepest reflectors ordinarily exceeds a factor of two. Thus depth variation of velocity is almost always included in the analysis of field data. Seismological theory needs to consider waves that are just like plane waves except that they bend to accommodate the velocity stratification $v(z)$. Figure 3.17 shows this in an idealized geometry: waves radiated from the horizontal flight of a supersonic airplane. The airplane passes location $x$ at time $t_0(x)$ flying horizontally

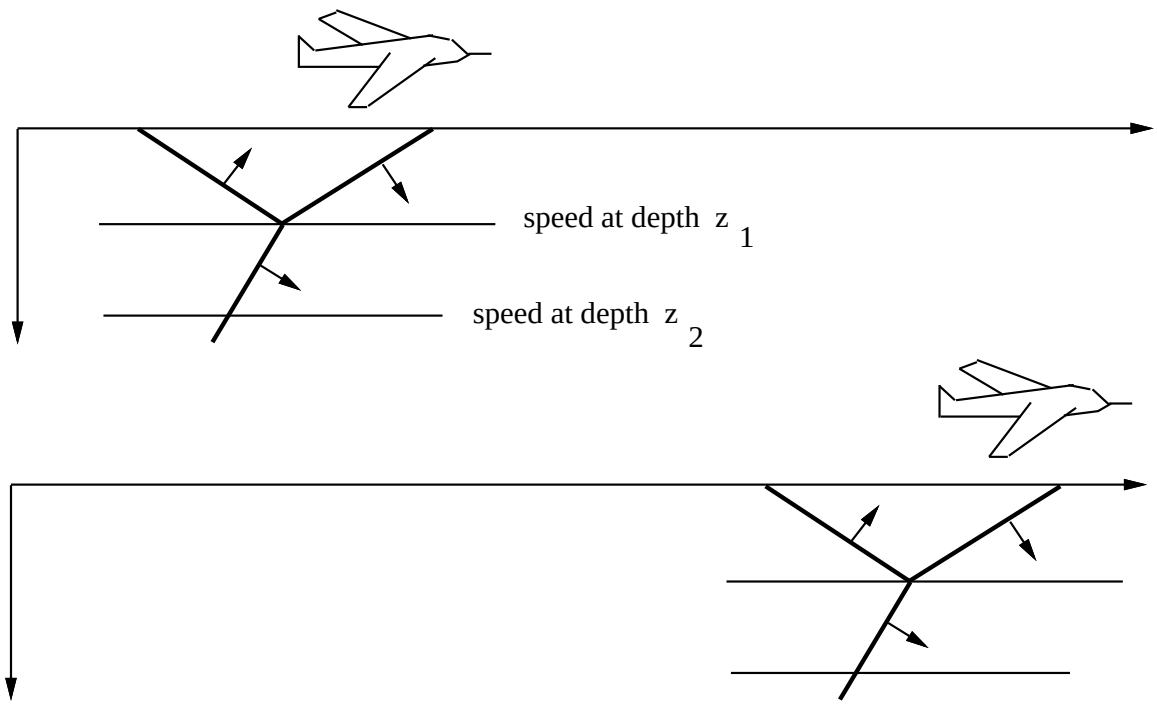



Figure 3.17: Fast airplane radiating a sound wave into the earth. From the figure you can deduce that the horizontal speed of the wavefront is the same at depth $z_1$ as it is at depth $z_2$. This leads (in isotropic media) to Snell's law. omk-airplane [NR]

at a constant speed. Imagine an earth of horizontal plane layers. In this model there is nothing to distinguish any point on the $x$-axis from any other point on the $x$-axis. But the seismic velocity varies from layer to layer. There may be reflections, head waves, shear waves, and multiple reflections. Whatever the picture is, it moves along with the airplane. A picture of the wavefronts near the airplane moves along with the airplane. The top of the picture and the bottom of the picture both move laterally at the same speed even if the earth velocity increases with depth. If the top and bottom didn't go at the same speed, the picture would become distorted, contradicting the presumed symmetry of translation. This horizontal speed, or rather its inverse $\partial t_0/\partial x$, has several names. In practical work it is called the *stepout.* In theoretical work it is called the *ray parameter.* It is very important to note that $\partial t_0/\partial x$ does not change with depth, even though the seismic velocity does change with depth. In a constant-

velocity medium, the angle of a wave does not change with depth. In a stratified medium, $\partial t_0/\partial x$ does not change with depth.

Figure 3.18 illustrates the differential geometry of the wave. The diagram shows

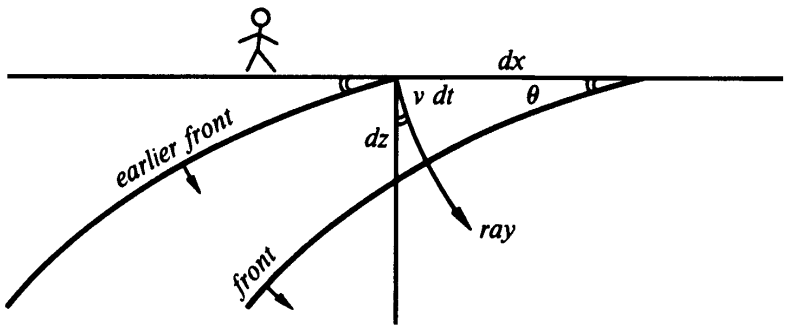



Figure 3.18: Downgoing fronts and rays in stratified medium $v(z)$. The wavefronts are horizontal translations of one another. omk-frontz [NR]

that

$$\frac{\partial t_0}{\partial x} = \frac{\sin\theta}{v} \tag{3.36}$$

$$\frac{\partial t_0}{\partial z} = \frac{\cos\theta}{v} \tag{3.37}$$

These two equations define two (inverse) speeds. The first is a horizontal speed, measured along the earth's surface, called the *horizontal phase velocity.* The second is a vertical speed, measurable in a borehole, called the *vertical phase velocity.* Notice that both these speeds *exceed* the velocity $v$ of wave propagation in the medium. Projection of wave *fronts* onto coordinate axes gives speeds larger than $v$, whereas projection of *rays* onto coordinate axes gives speeds smaller than $v$. The inverse of the phase velocities is called the *stepout* or the *slowness.*

Snell's law relates the angle of a wave in one layer with the angle in another. The constancy of equation (3.36) in depth is really just the statement of Snell's law. Indeed, we have just derived Snell's law. All waves in seismology propagate in a velocity-stratified medium. So they cannot be called plane waves. But we need a name for waves that are near to plane waves. A *Snell wave* will be defined to be the generalization of a plane wave to a stratified medium $v(z)$. A plane wave that happens to enter a medium of depth-variable velocity $v(z)$ gets changed into a Snell wave. While a plane wave has an angle of propagation, a Snell wave has instead a *Snell parameter* $p = \partial t_0/\partial x$.

It is noteworthy that Snell's parameter $p = \partial t_0/\partial x$ is directly observable at the surface, whereas neither $v$ nor $\theta$ is directly observable. Since $p = \partial t_0/\partial x$ is not only observable, but constant in depth, it is customary to use it to eliminate $\theta$ from equations (3.36) and (3.37):

$$\frac{\partial t_0}{\partial x} = \frac{\sin\theta}{v} = p \tag{3.38}$$

$$\frac{\partial t_0}{\partial z} \quad = \quad \frac{\cos\theta}{v} \quad = \quad \sqrt{\frac{1}{v(z)^2} \; - \; p^2} \tag{3.39}$$

Taking the Snell wave to go through the origin at time zero, an expression for the arrival time of the Snell wave at any other location is given by

$$t(x,z) \quad = \quad \frac{\sin\theta}{v}\, x \; + \; \int_0^z \frac{\cos\theta}{v}\, dz \tag{3.40}$$

$$t(x,z) \quad = \quad p\, x \; + \; \int_0^z \sqrt{\frac{1}{v(z)^2} \; - \; p^2}\; dz \tag{3.41}$$

The validity of (3.41) is readily checked by computing $\partial t_0/\partial x$ and $\partial t_0/\partial z$, then comparing with (3.38) and (3.39).

An arbitrary waveform $f(t)$ may be carried by the Snell wave. Use (3.40) and (3.41) to *define* the time $t_0$ for a delayed wave $f[t - t_0(x,z)]$ at the location $(x,z)$.

$$\text{SnellWave}(t,x,z) \quad = \quad f\left( \; t \; - \; p\, x \; - \; \int_0^z \sqrt{\frac{1}{v(z)^2} \; - \; p^2}\; dz \; \right) \tag{3.42}$$

### 3.6.4 Time-shifting equations

An important task is to predict the wavefield inside the earth given the waveform at the surface. For a downgoing plane wave this can be done by the time-shifting partial differential equation

$$\frac{\partial P(t,z)}{\partial z} \quad = \quad -\,\frac{1}{v}\,\frac{\partial P(t,z)}{\partial t} \tag{3.43}$$

as may be readily verified by substituting either of the trial solutions

$$P(t,z) \;=\; f\left( \; t \; - \; \frac{z}{v} \; \right) \qquad \text{for constant } \; v \tag{3.44}$$

$$P(t,z) \;=\; f\left( \; t \; - \; \int_0^z \frac{dz}{v(z)} \; \right) \qquad \text{for } v(z) \tag{3.45}$$

This also works for nonvertically incident waves with the partial differential equation

$$\frac{\partial P(t,x,z)}{\partial z} \quad = \quad -\,\frac{\partial t_0}{\partial z}\,\frac{\partial P(t,x,z)}{\partial t} \tag{3.46}$$

which has the solution

$$P(t,x,z) \quad = \quad f(t \; - \; p\, x \; - \; \int_0^z \frac{\partial t_0}{\partial z}\, dz) \tag{3.47}$$

In interpreting (3.46) and (3.47) recall that $1/(\partial t_0/\partial z$ is the apparent velocity in a borehole. The partial derivative of wavefield $P(t,x,z)$ with respect to depth $z$ is taken at constant $x$, i.e., the wave is extrapolated down the borehole. The idea that

downward extrapolation can be achieved by merely time shifting holds only when a *single* Snell wave is present; that is, the *same* arbitrary time function must be seen at all locations.

Substitution from (3.38) and (3.39) also enables us to rewrite (3.46) in the various forms

$$\frac{\partial P(t,x,z)}{\partial z} = -\frac{\cos\theta}{v}\,\frac{\partial P(t,x,z)}{\partial t} \tag{3.48}$$

$$\frac{\partial P(t,x,z)}{\partial z} = -\sqrt{\frac{1}{v(z)^2} - p^2}\;\frac{\partial P(t,x,z)}{\partial t} \tag{3.49}$$

$$\frac{\partial P(t,x,z)}{\partial z} = -\sqrt{\frac{1}{v(z)^2} - \left(\frac{\partial t_0}{\partial x}\right)^2}\;\frac{\partial P(t,x,z)}{\partial t} \tag{3.50}$$

Equations (3.48), (3.49) and (3.50) are paraxial wave equations. Since $\partial t_0/\partial x = p$ can be measured along the surface of the earth, it seems that equation (3.50), along with an assumed velocity $v(z)$ and some observed data $P(t,x,z=0)$, would enable us to determine $\partial P/\partial z$, which is the necessary first step of downward continuation. But the presumption was that there was only a single Snell wave and not a superposition of several Snell waves. Superposition of different waveforms on different Snell paths would cause different time functions to be seen at different places. Then a mere time shift would not achieve downward continuation. Luckily, a complicated wavefield that is variable from place to place may be decomposed into many Snell waves, each of which can be downward extrapolated with the differential equations (3.48), (3.49) and (3.50) or their solution (3.47). One such decomposition technique is *Fourier analysis.*

### 3.6.5 Fourier decomposition

Fourier analyzing the function $f(x,t,z=0)$, seen on the earth's surface, requires the Fourier kernel $\exp(-i\omega t + i\,k_x x)$. Moving on the earth's surface at an inverse speed of $\partial t_0/\partial x = k_x/\omega$, the phase of the Fourier kernel, hence the kernel itself, remains constant. Only those sinusoidal components that move at the same speed as the Snell wave can have a nonzero correlation with it. So if the disturbance is a single Snell wave, then all Fourier components vanish except for those that satisfy $p = k_x/\omega$. You should memorize these basic relations:

$$\boxed{\frac{\partial t_0}{\partial x} = \frac{\sin\theta}{v} = p = \frac{k_x}{\omega}} \tag{3.51}$$

In theoretical seismology a square-root function often appears as a result of using (3.51) to make a cosine.

Utilization of this Fourier domain interpretation of Snell's parameter $p$ enables us to write the square-root equations (3.48), (3.49) and (3.50) in an even more useful form. But first the square-root equation must be reexpressed in the Fourier domain.

This is done by replacing the $\partial/\partial t$ operator in (3.48), (3.49) and (3.50) by $-i\omega$. The result is

$$\frac{\partial P(\omega, k_x, z)}{\partial z} \quad = \quad + i\omega \sqrt{\frac{1}{v(z)^2} \; - \; \frac{k_x^2}{\omega^2}} \; P(\omega, k_x, z) \tag{3.52}$$

At present it is equivalent to specify either the differential equation (3.52) or its solution (3.47) with $f$ as the complex exponential:

$$P(\omega, k_x, z) \quad = \quad \exp\left( i\omega \int_0^z \sqrt{\frac{1}{v(z)^2} \; - \; \frac{k_x^2}{\omega^2}} \; dz \right) \tag{3.53}$$

Later, when we consider lateral velocity variation $v(x)$, the solution (3.53) becomes wrong, whereas the differential equation (3.50) is a valid description of any *local* plane-wave behavior. But before going to lateral velocity gradients we should look more carefully at vertical velocity gradients.

### 3.6.6 Velocity gradients

Inserting the Snell wavefield expression into the scalar wave equation, we discover that our definition of a Snell wave does not satisfy the scalar wave equation. The discrepancy arises only in the presence of velocity gradients. In other words, if there is a shallow constant velocity $v_1$ and a deep constant velocity $v_2$, the equation is satisfied everywhere except where $v_1$ changes to $v_2$. Solutions to the scalar wave equation must show amplitude changes across an interface, because of transmission coefficients. Our definition of a Snell wave is a wave of constant amplitude with depth. The paraxial wave equation could be modified to incorporate a transmission coefficient effect. The reason it rarely is modified may be the same reason that density gradients are often ignored. They add clutter to equations while their contribution to better results—namely, more correct amplitudes and possible tiny phase shifts—has marginal utility. Indeed, if they are included, then other deeper questions should also be included, such as the question, why use the acoustic equation instead of various other forms of scalar elastic equations?

Even if the paraxial wave equation were modified to incorporate a transmission coefficient effect, its solution would still fail to satisfy the scalar wave equation because of the absence of the reflected wave. But that is just fine, because it is the paraxial equation, with its reflection-free lenses, that is desired for data processing.

EXERCISES:

1 Devise a mathematical expression for a plane wave that is an impulse function of time with a propagation angle of 15° from the vertical $z$-axis in the plus $z$ direction. Express the result in the domain of

(a) $(t, x, z)$

(b) $(\omega, x, z)$

(c) $(\omega, k_x, z)$

(d) $(\omega, p, z)$

2 Find an amplitude function $A(z)$ which, when multiplied by $f$ in equation (3.47), yields an approximate solution to the scalar wave equation for stratified media $v(z)$.

# Chapter 4

# Migration by finite differences

In the last chapter we learned how to extrapolate wavefields down into the earth. The process proceeded simply, since it is just a multiplication in the frequency domain by $\exp[ik_z(\omega, k_x)z]$. Finite-difference techniques will be seen to be complicated. They will involve new approximations and new pitfalls. Why should we trouble ourselves to learn them? To begin with, many people find finite-difference methods more comprehensible. In $(t, x, z)$-space, there are no complex numbers, no complex exponentials, and no "magic" box called FFT.

The situation is analogous to the one encountered in ordinary frequency filtering. Frequency filtering can be done as a product in the frequency domain or a convolution in the time domain. With wave extrapolation there are products in both the temporal frequency $\omega$-domain and the spatial frequency $k_x$-domain. The new ingredient is the two-dimensional $(\omega, k_x)$-space, which replaces the old one-dimensional $\omega$-space. Our question, why bother with finite differences?, is a two-dimensional form of an old question: After the discovery of the fast Fourier transform, why should anyone bother with time-domain filtering operations?

Our question will be asked many times and under many circumstances. Later we will have the axis of offset between the shot and geophone and the axis of midpoints between them. There again we will need to choose whether to work on these axes with finite differences or to use Fourier transformation. It is not an all-or-nothing proposition: for each axis separately either Fourier transform or convolution (finite difference) must be chosen.

The answer to our question is many-sided, just as geophysical objectives are many-sided. Most of the criteria for answering the question are already familiar from ordinary filter theory where a filter can be made time-variable. Time-variable filters are useful in reflection seismology because the frequency content of echoes changes with time. An annoying aspect of time-variable filters is that they cannot be described by a simple product in the frequency domain. So when an application of time-variable filters comes along, the frequency domain is abandoned, or all kinds of contortions are made (stretching the time axis, for example) to try to make things appear time-invariant.

All the same considerations apply to the horizontal space axis $x$. On space axes, a new concern is the seismic velocity $v$. If it is space-variable, say $v(x)$, then the operation of extrapolating wavefields upward and downward can no longer be expressed as a product in the $k_x$-domain. Wave-extrapolation procedures must abandon the spatial frequency domain and go to finite differences. The alternative again is all kinds of contortions (such as stretching the $x$-axis) to try to make things appear to be space-invariant.

Fourier methods are *global.* That is, the entire dataset must be in hand before processing can begin. Remote errors and truncations can have serious local effects. On the other hand, finite-difference methods are *local.* Data points are directly related only to their neighbors. Remote errors propagate slowly.

Some problems of the Fourier domain have just been summarized. The problems of the space domain will be shown in this chapter and chapter 9. Seismic data processing is a multidimensional task, and the different dimensions are often handled in different ways. But if you are sure you are content with the Fourier domain then you can skip much of this chapter and jump directly to chapter 6 where you can learn about shot-to-geophone offset, stacking, and migration before stack.

## 4.1 WAVE-EXTRAPOLATION EQUATIONS

A wave-extrapolation equation is an expression for the derivative of a wavefield (usually in the depth $z$ direction). When the wavefield and its derivative are known, extrapolation can proceed by various numerical representations of

$$P(z + \Delta z) \quad = \quad P(z) \; + \; \Delta z \; \frac{dP}{dz} \tag{4.1}$$

Extrapolation is moving information from $z$ to $z + \Delta z$ and what we need to do it is a way to find $dP/dz$. Two theoretical methods for finding $dP/dz$ are the original *transformation* method and the newer *dispersion-relation* method.

### 4.1.1 Meet the parabolic wave equation

At the time the parabolic equation was introduced to petroleum prospecting (1969), it was well known that "wave theory doesn't work." At that time, petroleum prospectors analyzed seismic data with rays. The wave equation was not relevant to practical work. Wave equations were for university theoreticians. (Actually, wave theory did work for the surface waves of massive earthquakes, scales 1000 times greater than in exploration). Even for university workers, finite-difference solutions to the wave equation didn't work out very well. Computers being what they were, solutions looked more like "vibrations of a drum head" than like "seismic waves in the earth." The parabolic wave equation was originally introduced to speed finite-difference wave modeling. The following introduction to the parabolic wave equation is via my original transformation method.

The difficulty prior to 1969 came from an inappropriate assumption central to all then-existing seismic wave theory, namely, the horizontal layering assumption. Ray tracing was the only way to escape this assumption, but ray tracing seemed to ignore waveform modeling. In petroleum exploration almost all wave theory further limited itself to vertical incidence. The road to success lay in expanding ambitions from *vertical incidence* to include a small *angular bandwidth* around vertical incidence. This was achieved by abandoning much known, but cumbersome, seismic theory.

A vertically downgoing plane wave is represented mathematically by the equation

$$P(t,x,z) \quad = \quad P_0 \;\; e^{-\,i\omega\,(t\,-\,z/v)} \tag{4.2}$$

In this expression, $P_0$ is absolutely constant. A small departure from vertical incidence can be modeled by replacing the constant $P_0$ with something, say, $Q(x,z)$, which is not strictly constant but varies slowly.

$$P(t,x,z) \quad = \quad Q(x,z) \;\; e^{-\,i\,\omega\,(t\,-\,z/v)} \tag{4.3}$$

Inserting (4.3) into the scalar wave equation $P_{xx} + P_{zz} = P_{tt}/v^2$ yields

$$\begin{aligned} \frac{\partial^2}{\partial x^2}\;\; Q \;\; + \;\; \left( \frac{i\omega}{v} \; + \frac{\partial}{\partial z} \right)^2 \; Q \quad &= \quad - \; \frac{\omega^2}{v^2} \;\; Q \\ \frac{\partial^2 Q}{\partial x^2} \;\; + \;\; \frac{2\,i\omega}{v} \; \frac{\partial Q}{\partial z} \;\; + \;\; \frac{\partial^2 Q}{\partial z^2} \quad &= \quad 0 \end{aligned} \tag{4.4}$$

The wave equation has been reexpressed in terms of $Q(x,z)$. So far no approximations have been made. To require the wavefield to be near to a plane wave, $Q(x,z)$ must be near to a constant. The appropriate means (which caused some controversy when it was first introduced) is to drop the highest depth derivative of $Q$, namely, $Q_{zz}$. This leaves us with the *parabolic wave equation*

$$\frac{\partial Q}{\partial z} \quad = \quad \frac{v}{-\,2\,i\omega} \; \frac{\partial^2 Q}{\partial x^2} \tag{4.5}$$

When I first introduced equation (4.5) for use in seismology, I thought its most important property was this: For a wavefield close to a vertically propagating plane wave, the second $x$-derivative is small, hence the $z$-derivative is small. Thus, the finite-difference method should allow a very large $\Delta z$ and thus be able to treat models more like the earth, and less like a drumhead. I soon realized that the parabolic wave equation is also just what is needed for seismic imaging because you can insert it in an equation like (4.1). (Curiously, equation (4.5) also happens to be the Schroedinger equation of quantum mechanics.)

I called equation (4.5) the 15° equation. After using it for about a year I discovered a way to improve on it by estimating the dropped $\partial_{zz}$ term. Differentiate equation (4.5) with respect to $z$ and substitute the result back into equation (4.4) getting

$$\frac{\partial^2 Q}{\partial x^2} \;\; + \;\; \frac{2\,i\omega}{v} \; \frac{\partial Q}{\partial z} \;\; + \;\; \frac{v}{-\,2\,i\omega} \; \frac{\partial^3 Q}{\partial z \partial x^2} \quad = \quad 0 \tag{4.6}$$

I named equation (4.6) the 45° migration equation. It is first order in $\partial_z$, so it requires only a single surface boundary condition, however, downward continuation will require something more complicated than equation (4.1).

The above approach, the transformation approach, was and is very useful. But people were confused by the dropping and estimating of the $\partial_{zz}$ derivative, and a philosophically more pleasing approach was invented by Francis Muir, a way of getting equations to extrapolate waves at wider angles by fitting the dispersion relation of a semicircle by polynomial ratios.

### 4.1.2 Muir square-root expansion

Muir's method of finding wave extrapolators seeks polynomial ratio approximations to a square-root dispersion relation. Then fractions are cleared and the approximate dispersion relation is inverse transformed into a differential equation.

Substitution of the plane wave $\exp(-i\omega t + ik_x x + ik_z z)$ into the two-dimensional scalar wave equation (1.18) yields the dispersion relation

$$k_z^2 \;+\; k_x^2 \quad = \quad \frac{\omega^2}{v^2} \tag{4.7}$$

Solve for $k_z$ selecting the positive square root (thus for the moment selecting *downgoing* waves).

$$k_z \quad = \quad \frac{\omega}{v}\sqrt{1 \;-\; \frac{v^2\, k_x^2}{\omega^2}} \tag{4.8}$$

To inverse transform the $z$-axis we only need to recognize that $ik_z$ corresponds to $\partial/\partial z$. The resulting expression is a wavefield extrapolator, namely,

$$\frac{\partial}{\partial z}\; P(\omega, k_x, z) \quad = \quad i\,\frac{\omega}{v}\sqrt{1 \;-\; \frac{v^2\, k_x^2}{\omega^2}}\; P(\omega, k_x, z) \tag{4.9}$$

Bringing equation (4.9) into the space domain is not simply a matter of substituting a second $x$ derivative for $k_x^2$. The problem is the meaning of the square root of a differential operator. The square root of a differential operator is not defined in undergraduate calculus courses and there is no straightforward finite difference representation. The square root becomes meaningful only when the square root is regarded as some kind of truncated series expansion. It will be shown in chapter 8 that the Taylor series is a poor choice. Francis Muir showed that my original 15° and 45° methods were just truncations of a continued fraction expansion. To see this, define

$$X \quad = \quad \frac{vk_x}{\omega} \qquad \text{and} \qquad R \quad = \quad \frac{vk_z}{\omega} \tag{4.10}$$

So equation (4.8) is more simply and abstractly written as

$$R \quad = \quad \sqrt{1 \;-\; X^2} \tag{4.11}$$

| | |
|---|---|
| $5^\circ$ | $R_0 = 1$ |
| $15^\circ$ | $R_1 = 1 - \frac{X^2}{2}$ |
| $45^\circ$ | $R_2 = 1 - \frac{X^2}{2 - \frac{X^2}{2}}$ |
| $60^\circ$ | $R_3 = 1 - \frac{X^2}{2 - \frac{X^2}{2 - \frac{X^2}{2}}}$ |

Table 4.1: First four truncations of Muir's continued fraction expansion.

which you recognize as meaning that cosine is the square root of one minus sine squared. The desired polynomial ratio of order $n$ will be denoted $R_n$, and it will be determined by the recurrence

$$R_{n+1} = 1 - \frac{X^2}{1 + R_n} \tag{4.12}$$

The recurrence is a guess that we verify by seeing what it converges to (if it converges). Set $n = \infty$ in (4.12) and solve

$$\begin{aligned} R_\infty &= 1 - \frac{X^2}{1 + R_\infty} \\ R_\infty (1 + R_\infty) &= 1 + R_\infty - X^2 \\ R^2 &= 1 - X^2 \end{aligned} \tag{4.13}$$

The square root of (4.13) gives the required expression (4.11). Geometrically, (4.13) says that the cosine squared of the incident angle equals one minus the sine squared and truncating the expansion leads to angle errors. Muir said, and you can verify, that his recurrance relationship formalizes what I was doing by re-estimating the $\partial_{zz}$ term. Although it is pleasing to think of large values of $n$, in real life only the low-order terms in the expansion are used. Table 4.1 shows the result of three Muir iterations beginning from $R_0 = 1$

For various historical reasons, the equations in Table 4.1 are often referred to as the 5°, 15°, and 45° equations, respectively, the names giving a reasonable qualitative (but poor quantitative) guide to the range of angles that are adequately handled. A trade-off between complexity and accuracy frequently dictates choice of the 45° equation. It then turns out that a slightly wider range of angles can be accommodated if the recurrence is begun with something like $R_0 = \cos 45°$. Figure 4.1 shows some plots.

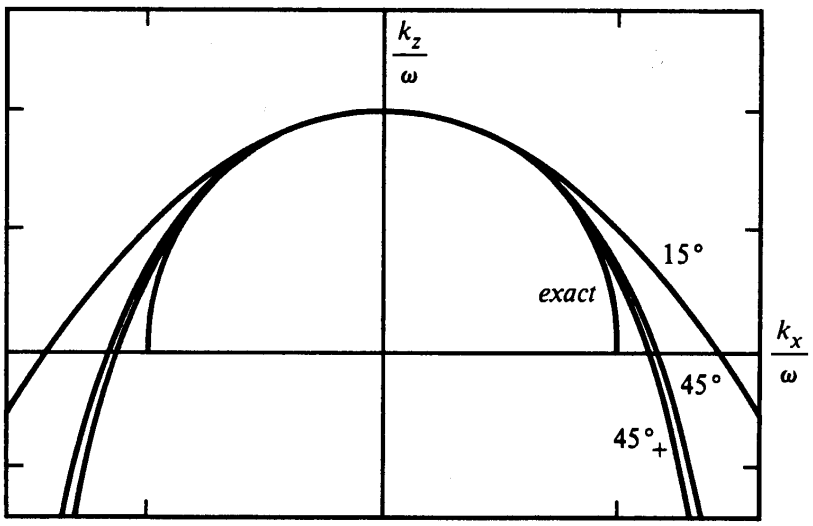



Figure 4.1: Dispersion relation of Table 4.2. The curve labeled $45^\circ_+$ was constructed with $R_0 = \cos 45°$ . It fits exactly at 0° and 45° . omx-disper [NR]

### 4.1.3 Dispersion relations

Performing the substitutions of Table 4.1 into equation (4.10) gives dispersion relationships for comparison to the exact expression (4.8). These are shown in Table 4.2.

### 4.1.4 Depth-variable velocity

Identification of $ik_z$ with $\partial/\partial z$ converts the dispersion relations of Table 4.2 into the differential equations of Table 4.3.

The differential equations in Table 4.3 were based on a dispersion relation that in turn was based on an assumption of constant velocity. So you might not expect that the equations have substantial validity or even great utility when the velocity is depth-variable, $v = v(z)$. The actual limitations are better characterized by their inability, by themselves, to describe reflection.

Migration methods based on equation (4.9) or on Table 4.3 are called *phase-shift methods.*

| | |
|---|---|
| $5^\circ$ | $k_z = \dfrac{\omega}{v}$ |
| $15^\circ$ | $k_z = \dfrac{\omega}{v} - \dfrac{v k_x^2}{2\omega}$ |
| $45^\circ$ | $k_z = \dfrac{\omega}{v} - \dfrac{k_x^2}{2\,\dfrac{\omega}{v} - \dfrac{v k_x^2}{2\omega}}$ |

Table 4.2: As displayed in Figure 4.1, the dispersion relations tend toward a semicircle.

| | |
|---|---|
| $5^\circ$ | $\dfrac{\partial P}{\partial z} = i\left(\dfrac{\omega}{v}\right) P$ |
| $15^\circ$ | $\dfrac{\partial P}{\partial z} = i\left(\dfrac{\omega}{v} - \dfrac{v k_x^2}{2\omega}\right) P$ |
| $45^\circ$ | $\dfrac{\partial P}{\partial z} = i\left(\dfrac{\omega}{v} - \dfrac{k_x^2}{2\,\dfrac{\omega}{v} - \dfrac{v k_x^2}{2\omega}}\right) P$ |

Table 4.3: Extrapolation equations when velocity depends only on depth.

### 4.1.5 Retardation (frequency domain)

It is often convenient to arrange the calculation of a wave to remove the effect of overall translation, thereby making the wave appear to "stand still." This subject, wave retardation, will be examined more thoroughly in chapter 5. Meanwhile, it is easy enough to introduce the time shift $t_0$ of a vertically propagating wave in a hypothetical medium of velocity $\bar{v}(z)$, namely,

$$t_0 \quad = \quad \int_0^z \frac{dz}{\bar{v}(z)} \tag{4.14}$$

A time delay $t_0$ in the time domain corresponds to multiplication by $\exp(i\omega t_0)$ in the $\omega$-domain. Thus, the wave pressure $P$ is related to the time-shifted mathematical variable $Q$ by

$$P(z,\omega) \quad = \quad Q(z,\omega)\ \exp\left( i\omega \int_0^z \frac{dz}{\bar{v}(z)} \right) \tag{4.15}$$

which is a generalization of equation (4.3) to depth-variable velocity. (Equations (4.15) and (4.17) apply in both $x$- and $k_x$-space). Differentiating with respect to $z$ gives

$$\frac{\partial P}{\partial z} \quad = \quad \frac{\partial Q}{\partial z}\ \exp\left( i\,\omega \int_0^z \frac{dz}{\bar{v}(z)} \right) \ + \ Q(z,\omega)\ \frac{i\omega}{\bar{v}(z)}\ \exp\left( i\,\omega \int_0^z \frac{dz}{\bar{v}(z)} \right) \tag{4.16}$$

$$\frac{\partial P}{\partial z} \quad = \quad \exp\left( i\omega \int_0^z \frac{dz}{\bar{v}(z)} \right) \left( \frac{\partial}{\partial z} \ + \ \frac{i\omega}{\bar{v}(z)} \right) \ Q \tag{4.17}$$

Next, substitute (4.15) and (4.17) into Table 4.3 to obtain the retarded equations in Table 4.4.

### 4.1.6 Lateral velocity variation

Having approximated the square root by a polynomial ratio, Table 4.3 or Table 4.4 can be inverse transformed from the horizontal wavenumber domain $k_x$ to the horizontal space domain $x$ by substituting $(ik_x)^2 = \partial^2/\partial x^2$. As before, the result has a wide range of validity for $v = v(x,z)$ even though the derivation would not seem to permit this.

### 4.1.7 Two velocities

You may wonder where the two velocities $v(x,z)$ and $\bar{v}(z)$ came from. The first arises in the wave equation, and it must be $x$-variable if the model is $x$-variable. The second arises in a mathematical transformation, namely, equation (4.15), so it is purely a matter of definition. Experience shows that complications will abound if we try to make the defined velocity $\bar{v}$ depend on $x$. But it is nice if the two velocities can be equal so the term containing their difference drops out of the analysis. Thus ordinarily, $\bar{v}(z)$ is chosen to be some kind of horizontal average of $v(x,z)$.

| | | | | |
|---|---|---|---|---|
| $5^\circ$ | $\dfrac{\partial Q}{\partial z}$ | $=$ | zero | $+\, i\omega \left( \dfrac{1}{v} - \dfrac{1}{\overline{v}(z)} \right) Q$ |
| $15^\circ$ | $\dfrac{\partial Q}{\partial z}$ | $=$ | $-\, i \dfrac{v k_x^2}{2\omega}\, Q$ | $+\, i\omega \left( \dfrac{1}{v} - \dfrac{1}{\overline{v}(z)} \right) Q$ |
| $45^\circ$ | $\dfrac{\partial Q}{\partial z}$ | $=$ | $-\, i \dfrac{k_x^2}{2\,\dfrac{\omega}{v} - \dfrac{v k_x^2}{2\omega}}\, Q$ | $+\, i\omega \left( \dfrac{1}{v} - \dfrac{1}{\overline{v}(z)} \right) Q$ |
| general | $\dfrac{\partial Q}{\partial z}$ | $=$ | diffraction | + thin lens |

Table 4.4: Retarded form of phase-shift equations.

### 4.1.8 Splitting diffraction and lens terms

The customary numerical solution to the $x$-domain forms of the equations in Tables 4-3 and 4-4 is arrived at by splitting. That is, you march forward a small $\Delta z$-step alternately with the two extrapolators

$$\frac{\partial Q}{\partial z} \quad = \quad \text{lens term} \tag{4.18}$$

$$\frac{\partial Q}{\partial z} \quad = \quad \text{diffraction term} \tag{4.19}$$

Formal justification of the splitting process is found in chapter 5. The first equation, called the *lens equation,* is solved analytically:

$$Q(z_2) \quad = \quad Q(z_1)\, \exp \left\{ i\omega \int_{z_1}^{z_2} \left( \frac{1}{v(x,z)} \; - \; \frac{1}{\overline{v}(z)} \right) dz \right\} \tag{4.20}$$

Migrations that include the lens equation are called *depth* migrations. The term is often omitted, giving a *time* migration.

Observe that the diffraction parts of Tables 4.3 and 4.4 are the same. Let us use them and equation (4.19) to define a table of diffraction equations. Substitute $\partial/\partial x$ for $ik_x$ and clear $\partial/\partial x$ from the denominators to get Table 4.5.

| | |
|---|---|
| $5^\circ$ | $\dfrac{\partial Q}{\partial z} = \text{zero}$ |
| $15^\circ$ | $\dfrac{\partial Q}{\partial z} = \dfrac{v(x,z)}{-2i\omega}\dfrac{\partial^2 Q}{\partial x^2}$ |
| $45^\circ$ | $\left[1 - \left(\dfrac{v(x,z)}{-2i\omega}\right)^2 \dfrac{\partial^2}{\partial x^2}\right] \dfrac{\partial Q}{\partial z} = \dfrac{v(x,z)}{-2i\omega}\dfrac{\partial^2 Q}{\partial x^2}$ |

Table 4.5: Diffraction equations for laterally variable media.

### 4.1.9 Time domain

To put the above equations in the time domain, it is necessary only to get $\omega$ into the numerator and then replace $-i\omega$ by $\partial/\partial t$. For example, the $15^\circ$, retarded, $v = \bar{v}$ equation from Table 4.5 becomes

$$\frac{\partial^2}{\partial z\, \partial t}\, Q \quad = \quad \frac{v}{2}\, \frac{\partial^2}{\partial x^2}\, Q \tag{4.21}$$

Interpretation of time $t$ for a retarded-time variable $Q$ awaits further clarification in chapter 5.

### 4.1.10 Upcoming waves

All the above equations are for *downgoing* waves. To get equations for *upcoming* waves you need only change the signs of $z$ and $\partial/\partial z$. Letting $D$ denote a *downgoing* wavefield and $U$ an *upcoming* wavefield, equation (4.21), for example, is found in Table 4.6.

Using the exploding-reflector concept, it is the upcoming wave equation that is found in both migration and diffraction programs. The downgoing wave equation is useful for modeling and migration procedures that are more elaborate than those based on the exploding-reflector concept (chapter 6).

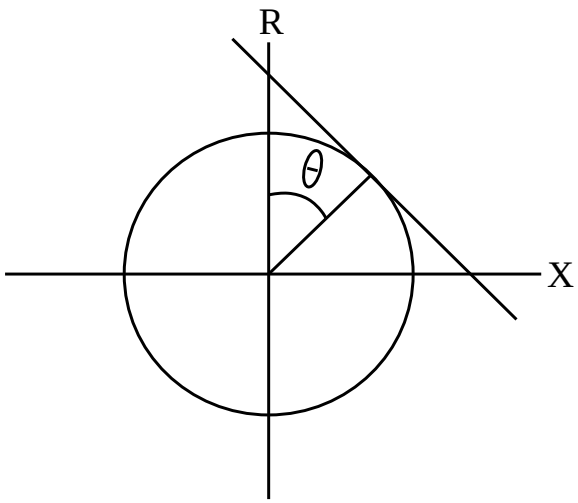
EXERCISES:

1 Consider a tilted straight line tangent to a circle, as shown in figure 4.2 . Use this line to initialize the Muir square-root expansion. State equations and plot them ($-2 \le X \le +2$) for the next two Muir semicircle approximations.

| | | |
|---|---|---|
| $\frac{\partial^2}{\partial z \partial t} D$ | $=$ | $+ \frac{v}{2} \frac{\partial^2}{\partial x^2} D$ |
| $\frac{\partial^2}{\partial z \partial t} U$ | $=$ | $- \frac{v}{2} \frac{\partial^2}{\partial x^2} U$ |

Table 4.6: Time-domain equations for downgoing and upcoming wave diffraction with retardation and the 15° approximation.

Figure 4.2: omx-tanexer [NR]



## 4.2 FINITE DIFFERENCING

The basic method for solving differential equations in a computer is *finite differencing.* The nicest feature of the method is that it allows analysis of objects of almost any shape, such as earth topography or geological structure. Ordinarily, finite differencing is a straightforward task. The main pitfall is instability. It often happens that a seemingly reasonable approach to a reasonable physical problem leads to wildly oscillatory, divergent calculations. Luckily, there is a fairly small body of important and easily learned tricks that solves many stability problems, and we will be covering them here.

Of secondary concern are the matters of cost and accuracy. These must be considered together since improved accuracy can be achieved simply by paying the higher price of a more refined computational mesh. Although the methods of the next several pages have not been chosen for their accuracy or efficiency, it turns out that in these areas they are excellent. Indeed, to my knowledge, some cannot be improved on at all, while others can be improved on only in small ways. By "small" I mean an improvement in efficiency of a factor of five or less. Such an improvement is rarely of consequence in research or experimental work; however, its importance in matters of production will justify pursuit of the literature far beyond the succeeding pages.

### 4.2.1 The lens equation

The various wave-extrapolation operators can be split into two parts, a complicated part called the *diffraction* or *migration* part, and an easy part called the *lens* part. The lens equation applies a time shift that is a function of $x$. The lens equation acquires its name because it acts just like a thin optical lens when a light beam enters on-axis (vertically). Corrections for nonvertical incidence are buried somehow in the diffraction part. The lens equation has an analytical solution, namely, $\exp[i\omega t_0(x)]$. It is better to use this analytical solution than to use a finite-difference solution because there are no approximations in it to go bad. The only reason the lens equation is mentioned at all in a chapter on finite differencing is that the companion diffraction equation must be marched forward along with the lens equation, so the analytic solutions are marched along in small steps.

### 4.2.2 First derivatives, explicit method

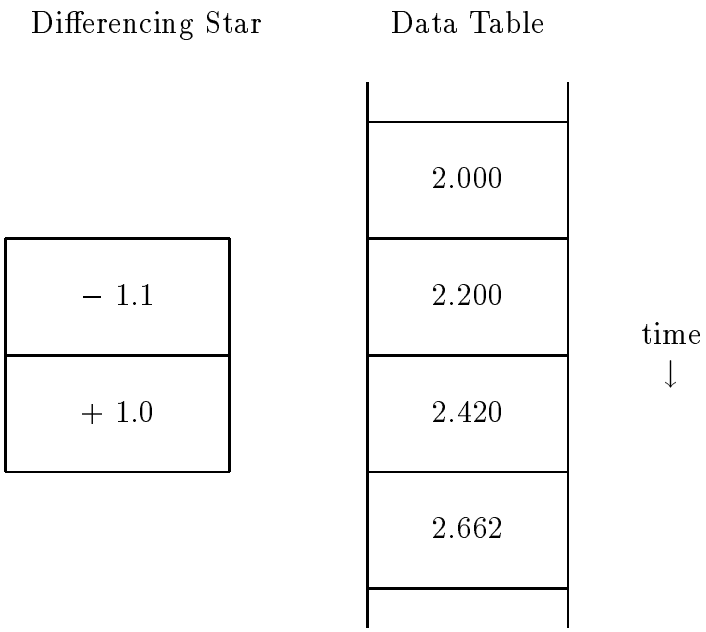
The inflation of money $q$ at a 10% rate can be described by the difference equation

$$q_{t+1} - q_t = .10\, q_t \tag{4.22}$$

$$(1.0)\, q_{t+1} + (-1.1)\, q_t = 0 \tag{4.23}$$

This one-dimensional calculation can be reexpressed as a differencing star and a data table. As such it provides a prototype for the organization of calculations with two-

dimensional partial-differential equations. Consider




Since the data in the data table satisfy the difference equations (4.22) and (4.23), the differencing star may be laid anywhere on top of the data table, the numbers in the star may be multiplied by those in the underlying table, and the resulting cross products will sum to zero. On the other hand, if all but one number (the initial condition) in the data table were missing then the rest of the numbers could be filled in, one at a time, by sliding the star along, taking the difference equations to be true, and solving for the unknown data value at each stage.

Less trivial examples utilizing the same differencing star arise when the numerical constant .10 is replaced by a complex number. Such examples exhibit oscillation as well as growth and decay.

### 4.2.3 First derivatives, implicit method

Let us solve the equation

$$\frac{dq}{dt} \quad = \quad 2\,r\,q \tag{4.24}$$

by numerical methods. The most obvious (but not the only) approach is the basic definition of elementary calculus. For the time derivative, this is

$$\frac{dq}{dt} \quad \approx \quad \frac{q(t+\Delta t) \;-\; q(t)}{\Delta t} \tag{4.25}$$

Using this in equation (4.24) yields the the inflation-of-money equations (4.22) and (4.23), where $2\,r = .1$. Thus in the inflation-of-money equation the expression of $dq/dt$ is centered at $t + \Delta t/2$, whereas the expression of $q$ by itself is at time $t$. There is no reason the $q$ on the right side of equation (4.24) cannot be averaged at time $t$ with

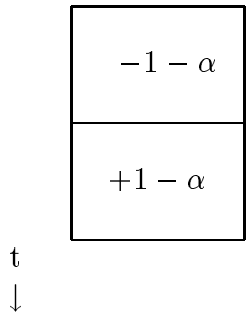
time $t + \Delta t$, thus centering the whole equation at $t + \Delta t/2$. When writing difference equations, it is customary to write $q(t + \Delta t)$ more simply as $q_{t+1}$. (Formally one should say $t = n\Delta t$ and write $q_{n+1}$ instead of $q_{t+1}$, but helpful mnemonic information is carried by using $t$ as the subscript instead of some integer like $n$.) Thus, a centered approximation of (4.24) is

$$q_{t+1} \; - \; q_t \quad = \quad 2\, r\, \Delta t \;\; \frac{q_{t+1} \; + \; q_t}{2} \tag{4.26}$$

Letting $\alpha = r\Delta t$, this becomes

$$(1 - \alpha)\; q_{t+1} \quad - \; (1 + \alpha)\; q_t \quad = \quad 0 \tag{4.27}$$

which is representable as the difference star




For a fixed $\Delta t$ this star gives a more accurate solution to the differential equation (4.24) than does the star for the inflation of money. The reasons for the names "explicit method" and "implicit method" above will become clear only after we study a more complicated equation such as the heat-flow equation.

### 4.2.4 The explicit heat-flow equation

The heat-flow equation controls the diffusion of heat. This equation is a prototype for migration. The 15° migration equation is the same equation but the heat conductivity constant is imaginary. (The migration equation is really the Schroedinger equation, which controls the diffusion of probability of atomic particles).

$$\frac{\partial q}{\partial t} \quad = \quad \frac{\sigma}{C}\, \frac{\partial^2 q}{\partial x^2} \tag{4.28}$$

Implementing (4.28) in a computer requires some difference approximations for the partial differentials. As before we use a subscript notation that allows (4.25) to be compacted into

$$\frac{\partial q}{\partial t} \quad \approx \quad \frac{q_{t+1} \; - \; q_t}{\Delta t} \tag{4.29}$$

where $t + \Delta t$ is denoted by $t + 1$. The second-derivative formula may be obtained by doing the first derivative twice. This leads to $q_{t+2} - 2\, q_{t+1} + q_t$. The formula is usually treated more symmetrically by shifting it to $q_{t+1} - 2\, q_t + q_{t-1}$. These two versions are

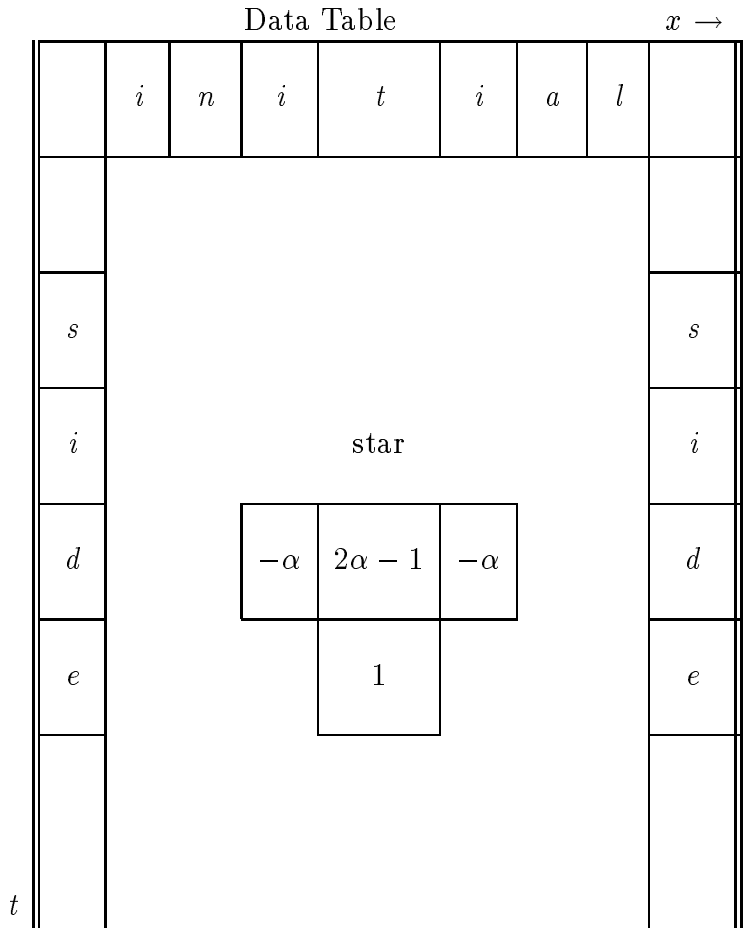



Table 4.7: Differencing star and table for one-dimensional heat-flow equation.

equivalent as $\Delta t$ tends to zero, but the more symmetrical arrangement will be more accurate when $\Delta t$ is not zero. Using superscripts to describe $x$-dependence gives a finite-difference approximation to the second space derivative:

$$\frac{\partial^2 q}{\partial x^2} \quad \approx \quad \frac{q^{x+1} \; - \; 2\,q^x \; + \; q^{x-1}}{\Delta x^2} \tag{4.30}$$

Inserting the last two equations into the heat-flow equation (and using $=$ to denote $\approx$) gives

$$\frac{q_{t+1}^x \; - \; q_t^x}{\Delta t} \quad = \quad \frac{\sigma}{C} \; \frac{q_t^{x+1} \; - \; 2\,q_t^x \; + \; q_t^{x-1}}{(\Delta x)^2} \tag{4.31}$$

(Of course it is not justified to use $=$ to denote $\approx$, but the study of errors must be deferred until the concepts have been laid out. Errors are studied in chapter 8.) Letting $\alpha = \sigma\,\Delta t/(C\,\Delta x^2)$, equation (4.31) becomes

$$q_{t+1}^x \; - \; q_t^x \; - \; \alpha\,(q_t^{x+1} \; - \; 2\,q_t^x \; + \; q_t^{x-1}\,) \quad = \quad 0 \tag{4.32}$$

Equation (4.32) can be *explicitly* solved for $q$ for any $x$ at the particular time $t+1$ given $q$ at all $x$ for the particular time $t$ and hence the name *explicit* method.

Equation (4.32) can be interpreted geometrically as a computational star in the $(x,t)$-plane, as depicted in Table 4.7. By moving the star around in the data table you will note that it can be positioned so that only one number at a time (the 1) lies over an unknown element in the data table. This enables the computation of subsequent rows beginning from the top. By doing this you are solving the partial-differential equation by the finite-difference method. There are many possible arrangements of initial and side conditions, such as zero-value side conditions. Next is a computer program and its result.

```
# Explicit heat-flow equation
real q(12), qp(12)
nx = 12
do ia= 1, 2 {           # stable and unstable cases
        alpha = ia*.3333;       write(6,'(/"alpha =",f5.2)') alpha
        do ix= 1,6  { q(ix) = 0.}       # Initial temperature step
        do ix= 7,12 { q(ix) = 1.}
        do it= 1, 6 {
                write(6,'(20f6.2)') (q(ix),ix=1,nx)
                do ix= 2, nx-1
                        qp(ix) = q(ix) + alpha*(q(ix-1)-2.*q(ix)+q(ix+1))
                qp(1) = qp(2);  qp(nx) = qp(nx-1)
                do ix= 1, nx
                        q(ix) = qp(ix)
                }
        }
call exit(0);   end
```

```
alpha = 0.33
  0.00  0.00  0.00  0.00  0.00  0.00  1.00  1.00  1.00  1.00  1.00  1.00
  0.00  0.00  0.00  0.00  0.00  0.33  0.67  1.00  1.00  1.00  1.00  1.00
  0.00  0.00  0.00  0.00  0.11  0.33  0.67  0.89  1.00  1.00  1.00  1.00
  0.00  0.00  0.00  0.04  0.15  0.37  0.63  0.85  0.96  1.00  1.00  1.00
  0.00  0.00  0.01  0.06  0.19  0.38  0.62  0.81  0.94  0.99  1.00  1.00
  0.00  0.00  0.02  0.09  0.21  0.40  0.60  0.79  0.91  0.98  1.00  1.00

alpha = 0.67
  0.00  0.00  0.00  0.00  0.00  0.00  1.00  1.00  1.00  1.00  1.00  1.00
  0.00  0.00  0.00  0.00  0.00  0.67  0.33  1.00  1.00  1.00  1.00  1.00
  0.00  0.00  0.00  0.00  0.44  0.00  1.00  0.56  1.00  1.00  1.00  1.00
  0.00  0.00  0.00  0.30 -0.15  0.96  0.04  1.15  0.70  1.00  1.00  1.00
  0.00  0.00  0.20 -0.20  0.89 -0.39  1.39  0.11  1.20  0.80  1.00  1.00
  0.13  0.13 -0.20  0.79 -0.69  1.65 -0.65  1.69  0.21  1.20  0.87  0.87
```

### 4.2.5 The leapfrog method

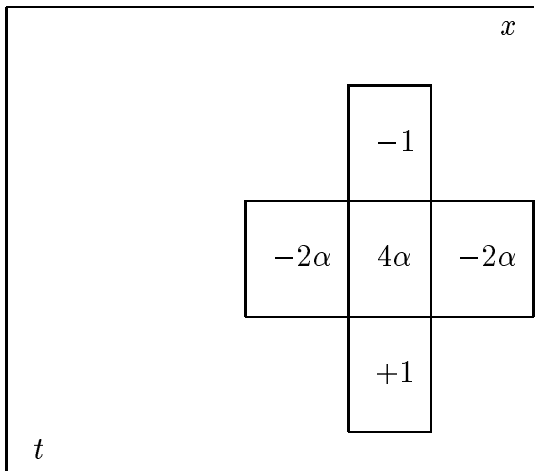
The difficulty with the given program is that it doesn't work for all possible numerical values of $\alpha$. You can see that when $\alpha$ is too large (when $\Delta x$ is too small) the solution in the interior region of the data table contains growing oscillations. What is happening is that the low-frequency part of the solution is OK (for a while), but the high-frequency part is diverging. The precise reason the divergence occurs is the subject of some mathematical analysis that will be postponed till page 131. At wavelengths long compared to $\Delta x$ or $\Delta t$, we expect the difference approximation

to agree with the true heat-flow equation, smoothing out irregularities in temperature. At short wavelengths the wild oscillation shows that the difference equation can behave in a way almost opposite to the way the differential equation behaves. The short wavelength discrepancy arises because difference operators become equal to differential operators only at long wavelengths. The divergence of the solution is a fatal problem because the subsequent round-off error will eventually destroy the low frequencies too.

By supposing that the instability arises because the time derivative is centered at a slightly different time $t + 1/2$ than the second $x$-derivative at time $t$, we are led to the so-called *leapfrog* method, in which the time derivative is taken as a difference between $t - 1$ and $t + 1$:

$$\frac{\partial q}{\partial t} \quad \approx \quad \frac{q_{t+1} \; - \; q_{t-1}}{2 \; \Delta t} \tag{4.33}$$

The resulting leapfrog differencing star is




Here the result is even worse. A later analysis shows that the solution is now divergent for *all* real numerical values of $\alpha$. Although it was a good idea to center both derivatives in the same place, it turns out that it was a bad idea to express a first derivative over a span of more mesh points. The enlarged operator has two solutions in time instead of just the familiar one. The numerical solution is the sum of the two theoretical solutions, one of which, unfortunately (in this case), grows and oscillates for all real values of $\alpha$.

To avoid all these problems (and get more accurate answers as well), we now turn to some slightly more complicated solution methods known as *implicit* methods.

### 4.2.6 The Crank-Nicolson method

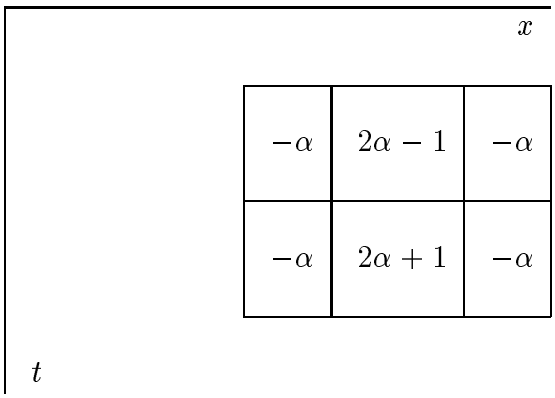
The Crank-Nicolson method solves both the accuracy and the stability problem. Recall the difference representation of the heat-flow equation (4.32).

$$q^x_{t+1} \; - \; q^x_t \quad = \quad a \; \left( q^{x+1}_t - 2q^x_t + q^{x-1}_t \right) \tag{4.34}$$

Now, instead of expressing the right-hand side entirely at time $t$, it will be averaged at $t$ and $t+1$, giving

$$q_{t+1}^x - q_t^x \quad = \quad \frac{a}{2}\left[\left(q_t^{x+1} - 2q_t^x + q_t^{x-1}\right) + \left(q_{t+1}^{x+1} - 2q_{t+1}^x + q_{t+1}^{x-1}\right)\right] \tag{4.35}$$

This is called the Crank-Nicolson method. Defining a new parameter $\alpha = a/2$, the difference star is




| | | |
|---|---|---|
| $-\alpha$ | $2\alpha - 1$ | $-\alpha$ |
| $-\alpha$ | $2\alpha + 1$ | $-\alpha$ |

(4.36)

When placing this star over the data table, note that, typically, three elements at a time cover unknowns. To say the same thing with equations, move all the $t+1$ terms in (4.35) to the left and the $t$ terms to the right, obtaining

$$-\alpha q_{t+1}^{x+1} + (1+2\alpha)q_{t+1}^x - \alpha q_{t+1}^{x-1} \quad = \quad \alpha q_t^{x+1} + (1-2\alpha)q_t^x + \alpha q_t^{x-1} \tag{4.37}$$

Now think of the left side of equation (4.37) as containing all the *unknown* quantities and the right side as containing all *known* quantities. Everything on the right can be combined into a single known quantity, say, $d_t^x$. Now we can rewrite equation (4.37) as a set of simultaneous equations. For definiteness, take the $x$-axis to be limited to five points. Then these equations are:

$$\begin{bmatrix} e_{\text{left}} & -\alpha & 0 & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & 0 & 0 \\ 0 & -\alpha & 1+2\alpha & -\alpha & 0 \\ 0 & 0 & -\alpha & 1+2\alpha & -\alpha \\ 0 & 0 & 0 & -\alpha & e_{\text{right}} \end{bmatrix} \begin{bmatrix} q_{t+1}^1 \\ q_{t+1}^2 \\ q_{t+1}^3 \\ q_{t+1}^4 \\ q_{t+1}^5 \end{bmatrix} \quad = \quad \begin{bmatrix} d_t^1 \\ d_t^2 \\ d_t^3 \\ d_t^4 \\ d_t^5 \end{bmatrix} \tag{4.38}$$

Equation (4.37) does not give us each $q_{t+1}^x$ *explicitly*, but equation (4.38) gives them *implicitly* by the solution of simultaneous equations.

The values $e_{\text{left}}$ and $e_{\text{right}}$ are adjustable and have to do with the side boundary conditions. The important thing to notice is that the matrix is tridiagonal, that is, except for three central diagonals all the elements of the matrix in (4.38) are zero. The solution to such a set of simultaneous equations may be economically obtained. It turns out that the cost is only about twice that of the explicit method given by (4.32). In fact, this implicit method turns out to be cheaper, since the increased accuracy of (4.37) over (4.32) allows the use of a much larger numerical choice of $\Delta t$ . A program that demonstrates the stability of the method, even for large $\Delta t$, is given next.

A tridiagonal simultaneous equation solving subroutine `rtris()` explained in the next section. The results are stable, as you can see.

```
a = 8.00
  0.00  0.00  0.00  0.00  0.00  0.00  1.00  1.00  1.00  1.00  1.00  1.00
  0.17  0.17  0.21  0.30  0.47  0.76  0.24  0.53  0.70  0.79  0.83  0.83
  0.40  0.40  0.42  0.43  0.40  0.24  0.76  0.60  0.57  0.58  0.60  0.60
  0.44  0.44  0.44  0.44  0.48  0.68  0.32  0.52  0.56  0.56  0.56  0.56
```

```
# Implicit heat-flow equation
real q(12),d(12)
nx=12; a = 8.;  write(6,'(/"a =",f5.2)') a;     alpha = .5*a
do ix= 1,6  {   q(ix) = 0.}     # Initial temperature step
do ix= 7,12 {   q(ix) = 1.}
do it= 1,4 {
        write(6,'(20f6.2)') (q(ix),ix=1,nx)
        d(1) = 0.;      d(nx) = 0.
        do ix= 2, nx-1
                d(ix) = q(ix) + alpha*(q(ix-1)-2.*q(ix)+q(ix+1))
        call rtris( nx, alpha, -alpha, (1.+2.*alpha), -alpha, alpha, d, q)
        }
call exit(0);   end
```

### 4.2.7 Solving tridiagonal simultaneous equations

Much of the world's scientific computing power gets used up solving tridiagonal simultaneous equations. For reference and completeness the algorithm is included here.

Let the simultaneous equations be written as a difference equation

$$a_j\, q_{j+1} \;+\; b_j\, q_j \;+\; c_j\, q_{j-1} \quad = \quad d_j \tag{4.39}$$

Introduce new unknowns $e_j$ and $f_j$, along with an equation

$$q_j \quad = \quad e_j\, q_{j+1} \;+\; f_j \tag{4.40}$$

Write (4.40) with shifted index:

$$q_{j-1} \quad = \quad e_{j-1}\, q_j \;+\; f_{j-1} \tag{4.41}$$

Insert (4.41) into (4.39):

$$a_j\, q_{j+1} \;+\; b_j\, q_j \;+\; c_j\,(e_{j-1}\, q_j \;+\; f_{j-1}) \quad = \quad d_j \tag{4.42}$$

Now rearrange (4.42) to resemble (15):

$$q_j \quad = \quad \frac{-\,a_j}{b_j \;+\; c_j\, e_{j-1}}\; q_{j+1} \;+\; \frac{d_j \;-\; c_j\, f_{j-1}}{b_j \;+\; c_j\, e_{j-1}} \tag{4.43}$$

Compare (4.43) to (4.40) to see recursions for the new unknowns $e_j$ and $f_j$:

$$e_j \quad = \quad \frac{-\,a_j}{b_j \;+\; c_j\, e_{j-1}} \tag{4.44}$$

$$f_j \quad = \quad \frac{d_j \;-\; c_j\, f_{j-1}}{b_j \;+\; c_j\, e_{j-1}} \tag{4.45}$$

First a boundary condition for the left-hand side must be given. This may involve one or two points. The most general possible end condition is a linear relation like equation (4.40) at $j = 0$, namely, $q_0 = e_0 q_1 + f_0$. Thus, the boundary condition must give us both $e_0$ and $f_0$. With $e_0$ and all the $a_j, b_j, c_j$, we can use (4.44) to compute all the $e_j$.

On the right-hand boundary we need a boundary condition. The general two-point boundary condition is

$$c_{n-1}\, q_{n-1} \;+\; e_{\text{right}}\, q_n \quad = \quad d_n \tag{4.46}$$

Equation (4.46) includes as special cases the zero-value and zero-slope boundary conditions. Equation (4.46) can be compared to equation (4.41) at its end.

$$q_{n-1} \quad = \quad e_{n-1}\, q_n \;+\; f_{n-1} \tag{4.47}$$

Both $q_n$ and $q_{n-1}$ are unknown, but in equations (4.46) and (4.47) we have two equations, so the solution is easy. The final step is to take the value of $q_n$ and use it in (4.41) to compute $q_{n-1}$, $q_{n-2}$, $q_{n-3}$, etc. The subroutine `rtris()` solves equation (4.38) for $q$ where `n=5`, `endl`$= e_{\text{left}}$, `endr`$= e_{\text{right}}$, `a=c`$= -\alpha$, and `b`$= 1 - 2\alpha$.

```
# real tridiagonal equation solver
subroutine rtris( n, endl, a, b, c, endr, d, q)
integer i,        n
real q(n), d(n), a, b, c, den, endl, endr
temporary real f(n), e(n)
e(1) = -a/endl;   f(1) = d(1)/endl
do i= 2, n-1 {
        den = b+c*e(i-1);   e(i) = -a/den;   f(i) = (d(i)-c*f(i-1))/den
        }
q(n) = (d(n)-c*f(n-1)) / (endr+c*e(n-1))
do i= n-1, 1, -1
        q(i) = e(i) * q(i+1) + f(i)
return;          end
```

If you wish to squeeze every last ounce of power from your computer, note some facts about this algorithm. (1) The calculation of $e_j$ depends on the *medium* through $a_j$, $b_j$, $c_j$, but it does not depend on the *solution* $q_j$ (even through $d_j$). This means that it may be possible to save and reuse $e_j$. (2) In many computers, division is much slower than multiplication. Thus, the divisor in (19a,b) can be inverted once (and perhaps stored for reuse).

### 4.2.8 The xxz derivative

The 45° diffraction equation differs from the 15° equation by the inclusion of a $\partial^3/\partial x^2 \partial z$ -derivative. Luckily this derivative fits on the six-point differencing star

$$\frac{1}{\Delta x^2\, \Delta z}$$

| | | |
|---|---|---|
| $-1$ | $2$ | $-1$ |
| $1$ | $-2$ | $1$ |

So other than modifying the six coefficients on the star, it adds nothing to the computational cost.

### 4.2.9 Difficulty in higher dimensions

So far we have had no trouble obtaining cheap, safe, and accurate difference methods for solving *partial-differential equations* (PDEs). The implicit method has met all needs. But in space dimensions higher than one, the implicit method becomes prohibitively costly. For the common example of problems in which $\partial^2/\partial x^2$ becomes generalized to $\partial^2/\partial x^2 + \partial^2/\partial y^2$, we will learn the reason why. The simplest case is the heat-flow equation for which the Crank-Nicolson method gave us (4.37). Introducing the abbreviation $\delta_{xx} q = q^{x+1} - 2q^x + q^{x-1}$, equation (4.37) becomes

$$(1 - \alpha\, \delta_{xx})\, Q_{t+1} \quad = \quad (1 + \alpha\, \delta_{xx})\, Q_t \tag{4.48}$$

The nested expression on the left represents a tridiagonal matrix. The critical stage is in solving the tridiagonal simultaneous equations for the vector of unknowns $Q_{t+1}$. Fortunately there is a special algorithm for this solution, and the cost increases only linearly with the size of the matrix. Now turn from the one-dimensional physical space of $x$ to two-dimensional $(x, y)$-space. Letting $\alpha$ denote the numerical constant in (4.48), the equation for stepping forward in time is

$$[\,1 - \alpha\, (\delta_{xx} + \delta_{yy})\,]\, Q_{t+1} \quad = \quad [1 + \alpha(\delta_{xx} + \delta_{yy})\,]\, Q_t \tag{4.49}$$

The unknowns $Q_{t+1}$ are a two-dimensional function of $x$ and $y$ that can be denoted by a matrix. Next we will interpret the bracketed expression on the left side. It turns out to be a four-dimensional matrix!

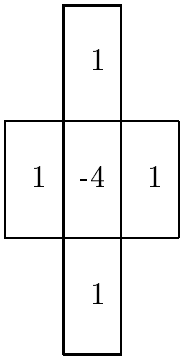
To clarify the meaning of this matrix, a mapping from two dimensions to one will be illustrated. Take the temperature $Q$ to be defined on a 4×4 mesh. A natural way of numbering the points on the mesh is

$$\begin{array}{cccc} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \\ 14 & 24 & 34 & 44 \end{array} \tag{4.50}$$

For algebraic purposes these sixteen numbers can be mapped into a vector. There are many ways to do this. A simple way would be to associate the locations in (4.50) with vector components by the column arrangement

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array} \tag{4.51}$$

The second difference operator has the following star in the $(x, y)$-plane:




(4.52)

Lay this star down in the $(x, y)$-plane (4.51) and move it around. Unfortunately, with just sixteen points, much of what you see is dominated by edges and corners. Try every position of the star that allows the center $-4$ to overlay one of the sixteen points. Never mind the 1's going off the sides. Start with the $-4$ in (4.52) over the 1 in the upper left corner of (4.51). Observe 1's on the 2 and the 5. Copy the 1's into the top row of Table 4.8 into the second and fifth columns. Then put the $-4$ in (4.52) over the 2 in (4.51). Observe 1's on the 1, 3, and 6. Copy the 1's into the next row of Table 4.8. Then put the $-4$ over the 3. Observe 1's on the 2, 4, and 7. Continue likewise. The $16 \times 16$ square matrix that results is shown in Table 4.8.

Now that Table 4.8 has been constructed we can return to the interpretation of equation (4.49). The matrix of unknowns $Q_{t+1}$ has been mapped into a sixteen-point column vector, and the bracketed expression multiplying $Q_{t+1}$ can be mapped into a $16 \times 16$ matrix. Clearly, the matrix contains zeroes everywhere that Table 4.8 contains dots. It seems fortunate that the table contains many zeroes, and we are led to hope for a rapid solution method for the simultaneous equations. The bad news is that no good method has ever been found. The best methods seem to require effort proportional to $N^3$, where in this case $N = 4$. Based on our experience in one dimension, those of us who worked on this problem hoped for a method proportional to $N^2$, which is the cost of an explicit method—essentially the cost of computing the right side of (4.41). Even all the features of implicit methods do not justify an additional cost of a factor of $N$. The next best thing is the splitting method.

EXERCISES:

1 Interpret the inflation-of-money equation when the interest rate is the imaginary number $i/10$.

2 Write the 45° diffraction equation in $(x, z)$-space for fixed $\omega$ in the form of (4.36).

## 4.3 WAVEMOVIE PROGRAM

An old professor of education had a monochromatic theme. It was his only theme and the topic of his every lecture. It was this:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| −4 | 1 | · | · | 1 | · | · | · | · | · | · | · | · | · | · | · |
| 1 | −4 | 1 | · | · | 1 | · | · | · | · | · | · | · | · | · | · |
| · | 1 | −4 | 1 | · | · | 1 | · | · | · | · | · | · | · | · | · |
| · | · | 1 | −4 | · | · | · | 1 | · | · | · | · | · | · | · | · |
| 1 | · | · | · | −4 | 1 | · | · | 1 | · | · | · | · | · | · | · |
| · | 1 | · | · | 1 | −4 | 1 | · | · | 1 | · | · | · | · | · | · |
| · | · | 1 | · | · | 1 | −4 | 1 | · | · | 1 | · | · | · | · | · |
| · | · | · | 1 | · | · | 1 | −4 | · | · | · | 1 | · | · | · | · |
| · | · | · | · | 1 | · | · | · | −4 | 1 | · | · | 1 | · | · | · |
| · | · | · | · | · | 1 | · | · | 1 | −4 | 1 | · | · | 1 | · | · |
| · | · | · | · | · | · | 1 | · | · | 1 | −4 | 1 | · | · | 1 | · |
| · | · | · | · | · | · | · | 1 | · | · | 1 | −4 | · | · | · | 1 |
| · | · | · | · | · | · | · | · | 1 | · | · | · | −4 | 1 | · | · |
| · | · | · | · | · | · | · | · | · | 1 | · | · | 1 | −4 | 1 | · |
| · | · | · | · | · | · | · | · | · | · | 1 | · | · | 1 | −4 | 1 |
| · | · | · | · | · | · | · | · | · | · | · | 1 | · | · | 1 | −4 |

Table 4.8: The two-dimensional matrix of coefficients for the Laplacian operator.

- *People learn by solving problems. Solving problems is the only way people learn, etc., etc., etc.....*

All he ever did was lecture; he never assigned any problems.

Your problems all relate to the computer subroutine `wavemovie()` (Lynn, Gonzalez, JFC, Hale, Li, Karrenbach). It makes a movie of a sum of monochromatic waves. As it stands it will produce a movie (three-dimensional matrix) of waves propagating through a focus. The whole process from compilation through computation to finally viewing the film loop takes about a minute. A sample frame of the movie is in Figure 4.3. It shows a snapshot of the $(x, z)$-plane. Collapsing spherical waves enter

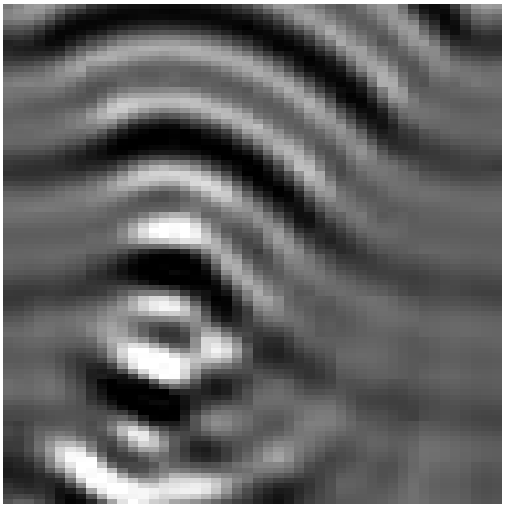


Figure 4.3: First frame of movie generated by `wavemovie()`. omx-Mfocus15 [ER]

from the top, go through a focus and then expand again. Notice that the wavefield is small but not zero in the region of geometrical shadow. In the shadow region you see waves that appear to be circles emanating from point sources at the top corners. Notice that the amplitudes of expanding spherical waves drop off with distance and collapsing spherical waves grow towards the focus. We will study the program that made this figure and see many features of waves and mathematics.

### 4.3.1 Earth's surface boundary condition

The program that created Figure 4.3 begins with an initial condition along the top boundary, and then this initial wavefield is extrapolated downward. So, the first question is: what is the mathematical function of $x$ that describes a collapsing spherical (actually cylindrical) wave? An expanding spherical wave has an equation $\exp[-i\omega(t - r/v)]$, where the radial distance is $r = \sqrt{(x - x_0)^2 + (z - z_0)^2}$ from the source. For a collapsing spherical wave we need $\exp[-i\omega(t + r/v)]$. Parantheticaly, I'll add that the theoretical solutions are not really these, but something more like these divided by $\sqrt{r}$, actually they should be a Hankel functions, but the picture

is little different when the exact initial condition is used. If you have been following this analysis, you should have little difficulty changing the initial conditions in the program to create a downgoing plane wave shown in Figure 4.4.

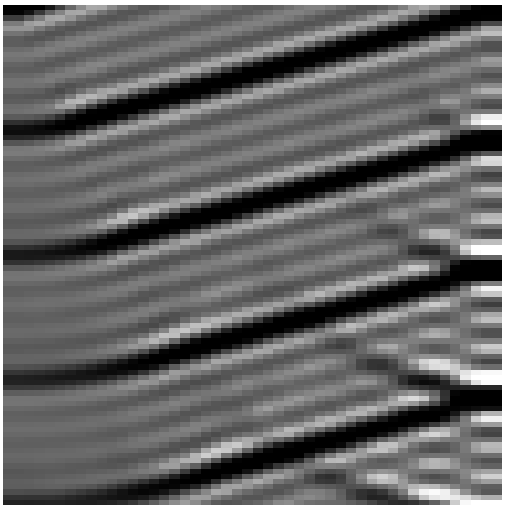


Figure 4.4: Specify program changes that give an initial plane wave propagating downward at an angle of 15° to the right of vertical. omx-Mdipplane [ER]

Notice the weakened waves in the zone of theoretical shadow that appear to arise from a point source on the top corner of the plot. You have probably learned in physics classes of "standing waves". This is what you will see near the reflecting side boundary if you recompute the plot with a single frequency `nw=1`. Then the plot will acquire a "checkerboard" appearence near the reflecting boundary. Even this figure with `nw=4` shows the tendency.

### 4.3.2 Time-domain analysis

For a film loop to make sense to a viewer, the subject of the movie must be periodic, and organized so that the last frame leads naturally into the first. In the movie created by `wavemovie()` there is a parameter `lambda` that controls the basic repetition rate of wave pulses fired onto the screen from the top. When a wavelet travels one-quarter of the way down the frame, another is sent in. This is defined by the line

$$\texttt{lambda = nz * dz / 4} \quad = \quad \frac{N_z \, \Delta z}{4}$$

Take any point in $(x, z)$-space. The signal there will be a superposition of sinusoids of various frequencies, $\omega_j$. We can choose what frequencies we will use in the calculation and what amplitudes and phases we will attach to the initial conditions at those frequencies. Here we will simply take uniformly spaced sinusoids of unit amplitude and no phase. The `nw` frequencies are $\omega_j = \Delta\omega$, $2\,\Delta\omega$, ..., $\texttt{nw}\,\Delta\omega$. The lowest frequency $\texttt{dw} = \Delta\omega$ must be inversely proportional to the wavelength `lambda` $= \lambda$

$$\texttt{dw = v * pi2 / lambda} = \quad \frac{2\,\pi\,v}{\lambda}$$

Finally, the time duration of the film loop must equal the period of the lowest-frequency sinusoid

$$N_t \, \Delta t \quad = \quad \frac{2\,\pi}{\Delta \omega}$$

This latter equation defines the time interval on the line

```
dt = pi2 / ( nt * dw )
```

If you use more frequencies, you might like the result better because the wave pulses will be shorter, and the number of wavelengths between the pulses will increase. Thus the quiet zones between the pulses will get quieter. The frequency components can be weighted differently—but this becomes a digression into simple Fourier analysis.

```
# from par:  integer n3:nt=12, n2:nx=48, n1:nz=96,  nw=2, nlam=4
# from par:  real dx=2, dz=1, v=1
#
subroutine wavemovie( nt, nx, nz, nw, nlam, dx,dz,v, p, cd, q)
integer it,nt,ix,nx,iz,nz,iw,nw, nlam
real dx,dz,v, phase,pi2,z0,x0,dt,dw,lambda,w,wov,x, p(nz,nx,nt)
complex aa,a,b,c,cshift, cd(nx),q(nx)
lambda=nz*dz/nlam;  pi2=2.*3.141592; dw=v*pi2/lambda;  dt=pi2/(nt*dw)
x0 = nx*dx/3; z0 = nz*dz/3
call null( p, nz*nx*nt)
do iw = 1,nw {                                  # superimpose nw frequencies
        w =  iw*dw;   wov = w/v                 # frequency / velocity
        do ix = 1,nx {                          # initial conditions for a
                x = ix*dx-x0;                   # collapsing spherical wave
                phase = -wov*sqrt( z0**2+x**2)
                q(ix) = cexp( cmplx( 0.,phase))
                }
        aa = (0.,1.)*dz/(4.*dx**2*wov)          # tridiagonal matrix coefficients
        a = -aa;      b = 1.+2.*aa;      c = -aa
        do iz = 1,nz {                          # extrapolation in depth
                do ix = 2,nx-1                  # diffraction term
                        cd(ix) = aa*q(ix+1) + (1.-2.*aa)*q(ix) + aa*q(ix-1)
                cd(1) = 0.;      cd(nx) = 0.
                call ctris( nx,-a,a,b,c,-c,cd,q)
                        # Solves complex tridiagonal equations
                cshift = cexp( cmplx( 0.,wov*dz))
                do ix = 1,nx                    # shifting term
                        q(ix) = q(ix) * cshift
                do it=1,nt {                    # evolution in time
                        cshift = cexp( cmplx( 0.,-w*it*dt))
                        do ix = 1,nx
                                p(iz,ix,it) = p(iz,ix,it) + q(ix)*cshift
                        }
                }
        }
return;         end
```

### 4.3.3 Internals of the film-loop program

The differential equation solved by the program is

$$\frac{\partial P}{\partial z} \quad = \quad \frac{i\,\omega}{v(x,z)} \; P \quad + \quad \frac{v}{-\,i\,\omega\,2} \; \frac{\partial^2 P}{\partial x^2} \tag{4.53}$$

For each $\Delta z$-step the calculation is done in two stages. The first stage is to solve

$$\frac{\partial P}{\partial z} \quad = \quad \frac{v}{-i\,\omega\,2}\,\frac{\partial^2 P}{\partial x^2} \tag{4.54}$$

Using the Crank-Nicolson differencing method this becomes

$$\frac{p_{z+1}^x - p_z^x}{\Delta z} \quad = \quad \frac{v}{-i\omega 2}\left(\frac{p_z^{x+1} - 2p_z^x + p_z^{x-1}}{2\,\Delta x^2} + \frac{p_{z+1}^{x+1} - 2p_{z+1}^x + p_{z+1}^{x-1}}{2\,\Delta x^2}\right) \tag{4.55}$$

Absorb all the constants into one and define

$$\alpha \quad = \quad \frac{v\,\Delta z}{-i\,\omega\,4\,\Delta x^2} \tag{4.56}$$

getting

$$p_{z+1}^x - p_z^x \quad = \quad \alpha\,\left[(\,p_z^{x+1} - 2\,p_z^{x+} p_z^{x-1}) + (\,p_{z+1}^{x+1} - 2\,p_{z+1}^x + p_{z+1}^{x-1})\right] \tag{4.57}$$

Bring the unknowns to the left:

$$-\,\alpha p_{z+1}^{x+1} + (1+2\alpha)p_{z+1}^x - \alpha p_{z+1}^{x-1} \quad = \quad \alpha p_z^{x+1} + (1-2\alpha)p_z^x + \alpha p_z^{x-1} \tag{4.58}$$

We will solve this as we solved equations (4.37) and (4.38). The second stage is to solve the equation

$$\frac{\partial P}{\partial z} \quad = \quad \frac{i\,\omega}{v}\;P \tag{4.59}$$

analytically by

$$P(z+\Delta z) \quad = \quad P(z)\;e^{i\Delta z\;\omega/v} \tag{4.60}$$

By alternating between (4.58) and (4.60), which are derived from (4.54) and (4.59), the program solves (4.53) by a splitting method. Formal justification of the splitting method follows in chapter 5. The program uses the tridiagonal solver discussed earlier, except the version here, `ctris()`, has all the real variables changed to complex.

Figure 4.5 shows a change of initial conditions where the incoming wave on the top frame is defined to be an impulse, namely, $p(x, z=0) = (\cdots, 0, 0, 1, 0, 0, \cdots)$. The result is alarmingly noisy. What is happening is that for any frequencies anywhere near the Nyquist frequency, the *difference* equation departs from the *differential* equation that it should mimic. This problem is addressed, analyzed, and ameliorated in chapter 8. For now, the best thing to do is to avoid sharp corners in the initial wave field.

### 4.3.4 Side-boundary analysis

In geophysics, we usually wish the side-boundary question did not arise. The only real reason for side boundaries is that either our survey or our processing activity is necessarily limited in extent. Given that side boundaries are inevitable, we must think

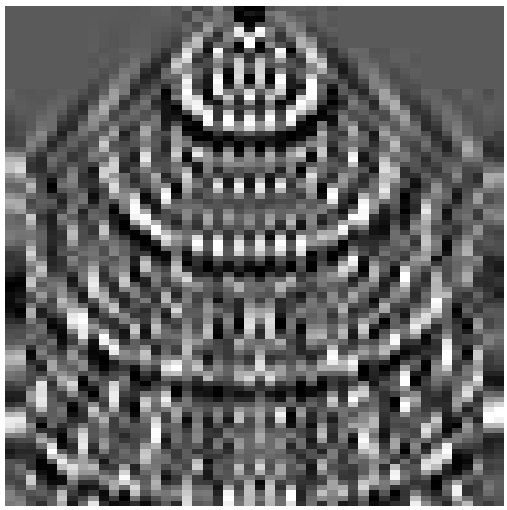


Figure 4.5: Observe and describe various computational artifacts by testing the program using a point source at $(x, z) =$ `(xmax/2,0)`. Such a source is rich in the high spatial frequencies for which difference equations do not mimic their differential counterparts. omx-Mcompart [ER]

about them. The subroutine `wavemovie()` included zero-slope boundary conditions. This type of boundary treatment resulted from taking

```
d(1) = 0.   ; d(nx) = 0.
```

and in the call to `ctris` taking

```
endl = - a ; endr = - c
```

A quick way to get zero-value side-boundary conditions is to take

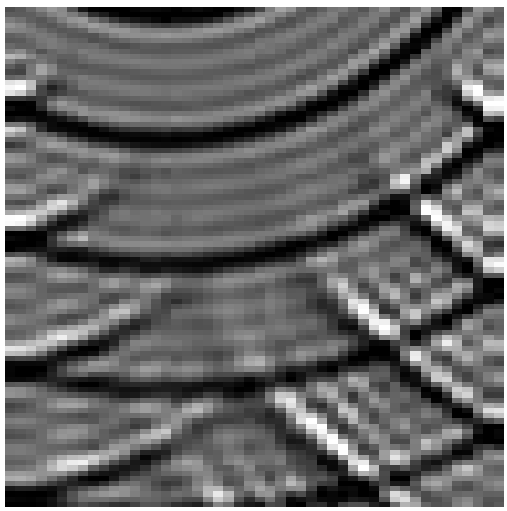


Figure 4.6: Given that the domain of computation is $0 \leq x \leq$ `xmax` and $0 \leq z \leq$`zmax`, how would you modify the initial conditions at $z = 0$ to simulate a point source at $(x, z) =$ `(xmax/3, -zmax/2)`? omx-Mexpandsphere [ER]

`endl = endr =` $10^{30} \approx \infty$

Compare the side-boundary behavior of Figures 4.6 and 4.7.