2.0 Why Space and Time?

In the previous 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 ω-domain and the spatial frequency k_x-domain. The new ingredient is the two-dimensional (ω, k_x)-space, which replaces the old one-dimensional ω-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. Those electrical engineers and old-time deconvolution experts who have pushed themselves into wave processing have turned out to be delighted by it. They hadn't realized their knowledge had so many applications!

Figure 1 illustrates the differences between Fourier domain calculations and time domain calculations. The figure was calculated on a 256×64 mesh to exacerbate for display the difficulties in either domain. Generally, you notice wraparound noise in the Fourier calculation, and frequency dispersion (Section 4.3) in the time domain calculation. (The "time domain" hyperbola in figure 1 is actually a frequency domain simulation — to wrap the entire hyperbola into view). In this Chapter we will see how to do the time domain calculations. A more detailed comparison of the domains is in Chapter 4.

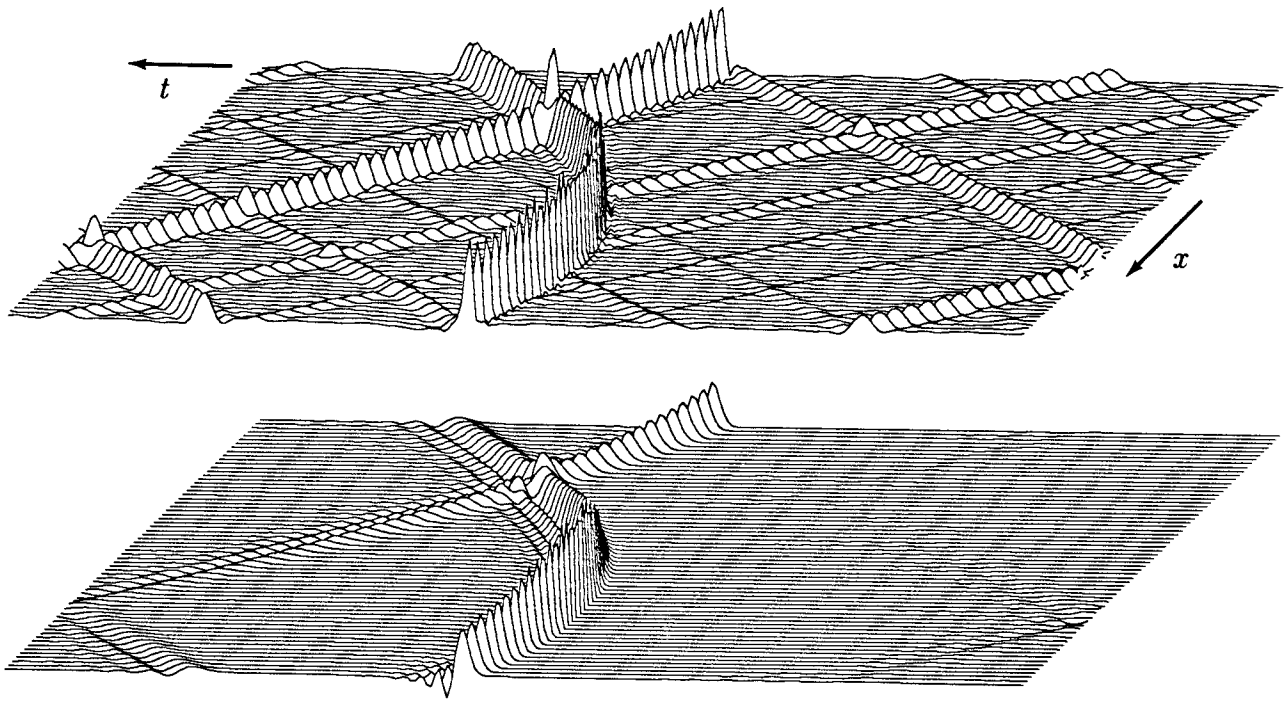


FIG. 2.0-1. Frequency domain hyperbola (top) and time domain hyperbola (bottom).

Even if you always migrate in the frequency domain, it is worth studying time domain methods to help you choose parameters to get a good time domain response. For example both parts of figure 1 were done in the frequency domain, but one simulated the time domain calculation to get a more causal response.

Lateral Variation

In ordinary linear filter theory, a filter can be made time-variable. This is 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.

In two or more dimensions, stretching tends to become more difficult and less satisfactory.

A less compelling circumstance of the same type that suggests finite differences rather than Fourier methods is lateral variation in channel location. If geophones somehow have become unevenly separated so that the Δx between channels is not independent of x, then there is a choice of (1) resampling the data at uniform intervals before Fourier analysis, or (2) processing the data directly with finite differences.

Stepout

Much of seismology amounts to measuring time shifts. The word *stepout* denotes a change of travel time with a change in location. Frequency-domain calculations usually conclude with a transform to the time domain to let us see the shifts. An advantage of time-domain computations is that time shifts of wave packets can be measured as the computation proceeds. In the frequency domain it is not difficult to reference one single time point, or to prescribe a shift of the whole time function. But it is not easy to access separate wavelets or wave packets without returning to the time domain.

The upward and downward wavefield extrapolation filter $\exp[i\, k_z(\omega, k_x)z]$ is basically a causal all-pass filter. (Under some circumstances it is anticausal). It moves energy around without amplification or attenuation. I suppose this is why migration filtering is more fun than minimum-phase filtering. Migration filters gather energy from all over and drop it in a good place, whereas minimum-phase filters hardly move things at all — they just scale some frequencies up and others down. Any filter of the form $\exp[i\,\phi(\omega)]$ is an all-pass filter. What are the constraints on the function $\phi(\omega)$ which make the time-domain representation of $\exp(i\,\phi)$ causal?

Causal all-pass filters turn out to have an attractive representation, with Z-transforms as $Z^N \bar{A}(1/Z)/A(Z)$. Those who are familiar with filter theory will realize that the division by $A(Z)$ raises a whole range of new issues: feedback, economy of parameterization, and possible instability. (Section 4.6 covers Z-transforms). These issues will all arise in using finite differences to downward extrapolate wavefields. It is a feedback process. The economy of parameterization is attractive. Taking $A(Z) = 1 + a_1 Z + a_2 Z^2$, the two adjustable coefficients are sufficient to select a frequency and a bandwidth for selective delay. Economy of parameterization also implies economy in application. That is nice. It is also nice having the functional form itself imply causality. On the other hand, the advantages of economy are offset by some dangers. Now we must learn and use some stability theory. $A(Z)$ must be minimum phase.

Being Too Clever in the Frequency Domain

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. Let me cite two examples of frequency-domain pitfalls in the field of one-dimensional time-series analysis.

In the frequency domain it is easy to specify sharp cutoff filters, say, a perfectly flat passband between 8 Hz and 80 Hz, zero outside. But such filters cause problems in the time domain. They are necessarily noncausal, giving a response before energy enters the filter. Another ugly aspect is that the time response drops off only inversely with t. Distant echoes that have amplitudes weakened as inverse time squared would get lost in the long filter response of the early echoes.

A more common problem arises with the 60 Hz powerline frequency rejection filters found in much recording equipment. Notch filters are easy to construct in the Z-transform domain. Start with a zero on the unit circle at exactly 60 Hz. That kills the noise but it distorts the passband at other frequencies. So, a tiny distance away, outside the unit circle, place a pole. The separation between the pole and the zero determines the bandwidth of the notch. The pole has the effect of nearly canceling the zero if the pair are seen from a distance. So there is an ideal flat spectrum away from the absorption zone. You record some data with this filter. Late echoes are weaker than early ones, so the plotting program increases the gain with time. After installing your powerline reject filters you discover that they have *increased* the powerline noise instead of decreasing it. Why? The reason is that you tried

to be too clever when you put the pole too close to the circle. The exponential gain effectively moved the unit circle away from the zero towards the pole. The pole may end up on the circle! Putting the pole further from the zero gives a broader notch, which is less attractive in the frequency domain, but at least the filter will work sensibly when the gain varies with time.

Zero Padding

When fast Fourier transforms came into use, one of the first applications was convolution. If a filter has more than about fifty coefficients, it may be faster to apply it by multiplication in the frequency domain. The result will be identical to convolution if care has been taken to pad the ends of the data and the filter with enough zeroes. They make invisible the periodic behavior of the discrete Fourier transform. For filtering time functions whose length is typically about one thousand, this is a small price in added memory to pay for the time saved. Seismic sections are often thousands of channels long. For migration, zero padding must simultaneously be done on the space axis and the time axis. There are three places where zeroes may be required, as indicated below:

data	0
0	0

Section 4.5 offers suggestions on how to alleviate the problems of Fourier domain migration techniques.

Looking Ahead

Some problems of the Fourier domain have just been summarized. The problems of the space domain will be shown in this chapter and Chapter 4. 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 3, where you can learn about shot-to-geophone offset, stacking, and migration before stack.

2.1 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 $P(z + \Delta z) = P(z) + \Delta z \; dP/dz$. So what is really needed is an expression for dP/dz. Two theoretical methods for finding dP/dz are the original *transformation* method and the newer *dispersion-relation* method.

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 the 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

$$P(t, x, z) \quad = \quad P_0 \; e^{-i\omega(t - z/v)} \tag{1}$$

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.

$$P(t, x, z) \quad = \quad Q(x, z)\ e^{-i\omega(t - z/v)} \tag{2}$$

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

$$\frac{\partial^2}{\partial x^2}\ Q \ + \ \left(\frac{i\omega}{v} + \frac{\partial}{\partial z} \right)^2 Q \quad = \quad - \frac{\omega^2}{v^2}\ Q$$

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

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*

$$\frac{\partial Q}{\partial z} \quad = \quad \frac{v}{-2\,i\omega}\ \frac{\partial^2 Q}{\partial x^2} \tag{4}$$

At the time it was first developed for use in seismology, the most important property of (4) was thought to be 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 Δz and thus be able to treat models more like the earth, and less like a drumhead.

It soon became apparent that the parabolic wave equation was also just what was needed for seismic imaging: it was a wave-extrapolation equation.

It is curious that equation (4) is the Schroedinger equation of quantum mechanics.

This approach, the transformation approach, was and is very useful. But it was soon replaced by the dispersion-equation approach — a way of getting equations to extrapolate waves at wider angles.

Muir Square-Root Expansion

When we use the newer method of finding wave extrapolators, we seek various approximations to a square-root dispersion relation. Then the approximate dispersion relation is inverse transformed into a differential equation. Thanks largely to Francis Muir, the dispersion approach has evolved considerably since the writing of *Fundamentals of Geophysical Data Processing.*

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

$$k_z^2 + k_x^2 = \frac{\omega^2}{v^2} \tag{5}$$

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

$$k_z = \frac{\omega}{v}\sqrt{1 - \frac{v^2 k_x^2}{\omega^2}} \tag{6a}$$

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,

$$\frac{\partial P}{\partial z} = i\,\frac{\omega}{v}\sqrt{1 - \frac{v^2 k_x^2}{\omega^2}}\,P \tag{6b}$$

Bringing equation (6b) 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 Section 4.6 that the Taylor series is a poor choice. Francis Muir showed that the original 15° and 45° methods were just truncations of a continued fraction expansion. To see this, let X and R be defined by writing (6a) as

$$k_z = \frac{\omega}{v}\sqrt{1 - X^2} = \frac{\omega}{v}\,R \tag{7}$$

The desired polynomial ratio of order n will be denoted R_n, and it will be determined by the recurrence

$$R_{n+1} = 1 - \frac{X^2}{1 + R_n} \tag{8}$$

To see what this sequence converges to (if it converges), set $n = \infty$ in (8) and solve

$$R_\infty = 1 - \frac{X^2}{1 + R_\infty}$$

$$R_\infty(1 + R_\infty) = 1 + R_\infty - X^2$$

$$R^2 = 1 - X^2 \tag{9}$$

The square root of (9) gives the required expression (7). Geometrically, (9) says that the cosine squared of the incident angle equals one minus the sine squared. Truncating the expansion leads to angle errors. Actually it is only the low-order terms in the expansion that are ever used. Beginning from $R_0 = 1$ the results in table 1 are found.

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

TABLE 2.1-1. First four truncations of Muir's continued fraction expansion.

For various historical reasons, the equations in table 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 $R_0 = \cos 45°$. Accuracy enthusiasts might even have R_0 a function of velocity, space coordinates, or frequency.

Dispersion Relations

Performing the substitutions of table 1 into equation (7) gives dispersion relationships for comparison to the exact expression (6a). These are shown in table 2.

5°	$k_z = \frac{\omega}{v}$
15°	$k_z = \frac{\omega}{v} - \frac{v\ k_x^2}{2\ \omega}$
45°	$k_z = \frac{\omega}{v} - \frac{k_x^2}{2\ \frac{\omega}{v} - \frac{v\ k_x^2}{2\ \omega}}$

TABLE 2.1-2. As displayed in figure 1, the dispersion relations of table 2 tend toward a semicircle.

Depth-Variable Velocity

Identification of ik_z with $\partial/\partial z$ converts the dispersion relations of table 2 into the differential equations of table 3.

The differential equations in table 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 (6b) or on table 3 are called *phase-shift methods.*

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 Section 2.6. Meanwhile, it is easy enough to introduce the time shift t_0 of a vertically propagating wave in a hypothetical medium of velocity $\overline{v}(z)$, namely,

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

TABLE 2.1-3. Extrapolation equations when velocity depends only on depth.

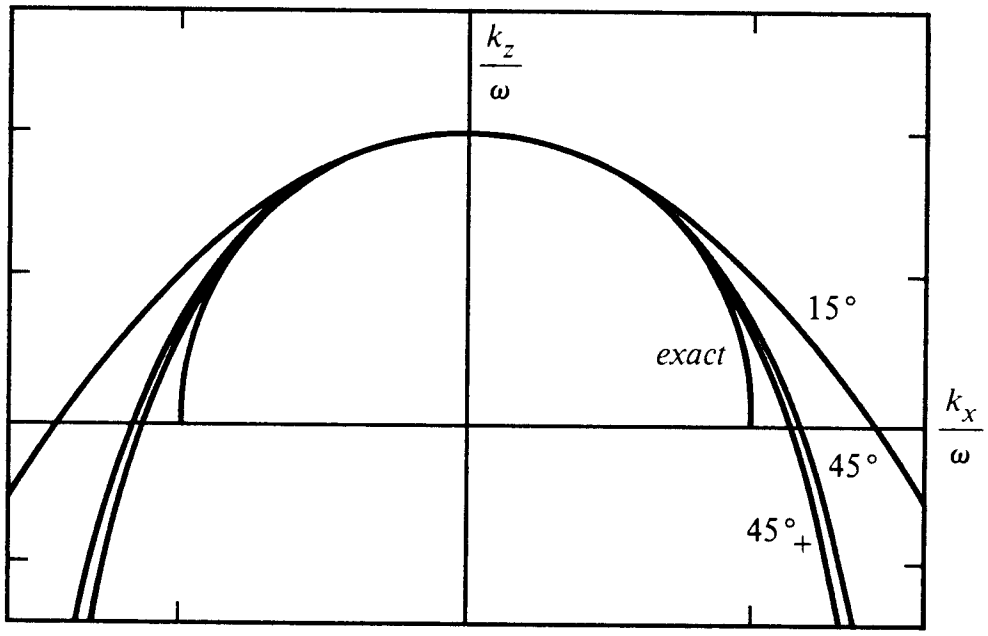


FIG. 2.1-1. Dispersion relation of equations (6a) and table 2. The curve labeled 45°_+ was constructed with $R_0 = \cos 45^\circ$. It fits exactly at 0° and 45°.

$$t_0 = \int_0^z \frac{dz}{\overline{v}(z)} \tag{10}$$

A time delay t_0 in the time domain corresponds to multiplication by $\exp(i\omega t_0)$ in the ω-domain. Thus, the actual wavefield P is related to the time-shifted wavefield Q by

$$P(z,\omega) = Q(z,\omega) \exp\left[i\omega \int_0^z \frac{dz}{\overline{v}(z)} \right] \tag{11a}$$

(Equation (11) applies in both x- and k_x-space). Differentiating with respect to z gives

$$\frac{\partial P}{\partial z} = \frac{\partial Q}{\partial z} \exp\left[i\omega \int_0^z \frac{dz}{\overline{v}(z)} \right] + Q(z,\omega) \frac{i\omega}{\overline{v}(z)} \exp\left[i\omega \int_0^z \frac{dz}{\overline{v}(z)} \right]$$

or

$$\frac{\partial P}{\partial z} = \exp\left[i\omega \int_0^z \frac{dz}{\overline{v}(z)} \right] \left[\frac{\partial}{\partial z} + \frac{i\omega}{\overline{v}(z)} \right] Q \tag{11b}$$

Next, substitute (11) into table 3 to obtain the retarded equations in table 4.

Lateral Velocity Variation

Having approximated the square root by a polynomial ratio, table 3 or table 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. Ordinarily $\overline{v}(z)$ will be chosen to be some kind of horizontal average of $v(x,z)$. Permitting $\overline{v}$ to become a function of x generates many new terms. The terms are awkward to implement and ignoring them introduces unknown hazards. So $\overline{v}$ is usually taken to depend on z but not x.

Splitting

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

$$\frac{\partial Q}{\partial z} = \textit{lens term} \tag{12a}$$

$$\frac{\partial Q}{\partial z} = \textit{diffraction term} \tag{12b}$$

5°	$\frac{\partial Q}{\partial z} = \textit{zero}$	$+ i\omega \left(\frac{1}{v} - \frac{1}{\overline{v}(z)} \right) Q$
15°	$\frac{\partial Q}{\partial z} = -i \frac{v k_x^2}{2\omega} Q$	$+ i\omega \left(\frac{1}{v} - \frac{1}{\overline{v}(z)} \right) Q$
45°	$\frac{\partial Q}{\partial z} = -i \frac{k_x^2}{2\frac{\omega}{v} - \frac{v k_x^2}{2\omega}} Q$	$+ i\omega \left(\frac{1}{v} - \frac{1}{\overline{v}(z)} \right) Q$
general	$\frac{\partial Q}{\partial z} = \textit{diffraction}$	*+ thin lens*

TABLE 2.1-4. Retarded form of phase-shift equations.

Justification of the splitting process is found in Section 2.4. The first equation, called the *lens equation,* is solved analytically:

$$Q(z_2) = Q(z_1) \exp \left\{ i\omega \int_{z_1}^{z_2} \left[\frac{1}{v(x, z)} - \frac{1}{\overline{v}(z)} \right] dz \right\} \tag{13}$$

Observe that the diffraction parts of tables 3 and 4 are the same. Let us use them and equation (12b) to define a table of diffraction equations. Substitute $\partial/\partial x$ for ik_x and clear $\partial/\partial x$ from the denominators to obtain table 5.

Time Domain

To put the above equations in the time domain, it is necessary only to get ω into the numerator and then replace $-i\omega$ by $\partial/\partial t$. For example, the 15°, retarded, $v = \overline{v}$ equation from table 5 becomes

$$\frac{\partial^2}{\partial z \, \partial t} Q = \frac{v}{2} \frac{\partial^2}{\partial x^2} Q \tag{14}$$

Interpretation of time t for a retarded-time variable Q awaits further clarification in Section 2.6.

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

TABLE 2.1-5. Diffraction equations for laterally variable media.

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 (14), for example, takes the form

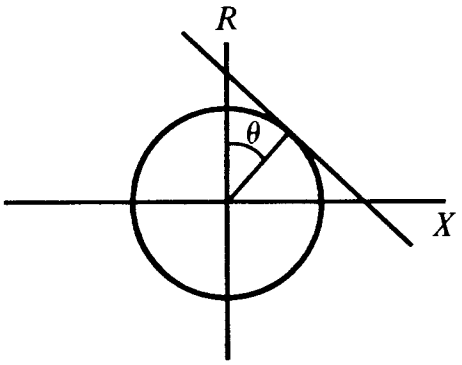
$\frac{\partial^2}{\partial z\,\partial t} D = +\frac{v}{2}\frac{\partial^2}{\partial x^2} D$
$\frac{\partial^2}{\partial z\,\partial t} U = -\frac{v}{2}\frac{\partial^2}{\partial x^2} U$

TABLE 2.1-6. Time-domain equations for downgoing and upcoming wave diffraction with retardation and the 15° approximation.

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 (see Section 5.7).

EXERCISE

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



2.2 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 should solve most stability problems.

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.

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 and the thickness of the lens 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.

First Derivatives, Explicit Method

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

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

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

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

Differencing Star *Data Table*

−1.1
+1.0

2.000
2.200
2.420
2.662

time ↓ (2)

Since the data in the data table satisfy the difference equation (1), 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 equation 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.

First Derivatives, Implicit Method

Let us solve the equation

$$\frac{dq}{dt} = 2\,r\,q \tag{3}$$

by numerical methods. Note that the inflation-of-money equation (1), where $2\,r = .1$, provides an approximation. But then note that in the inflation-of-money equation the expression of dq/dt is centered at $t+1/2$, whereas the expression of q by itself is at time t. There is no reason the q on the right side of equation (3) cannot be averaged at time t with time $t+1$, thus centering the whole equation at $t+1/2$. Specifically, a centered approximation of (3) is

$$q_{t+1} - q_t = 2\,r\,\Delta t\,\frac{q_{t+1} + q_t}{2} \tag{4a}$$

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

$$(1-\alpha)\,q_{t+1} - (1+\alpha)\,q_t = 0 \tag{4b}$$

which is representable as the difference star

$-1-\alpha$
$+1-\alpha$

t
$\downarrow$ (4c)

For a fixed Δt this star gives a more accurate solution to the differential equation (3) than does the star for the inflation of money.

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). Taking σ constant yields

$$\frac{\partial q}{\partial t} = \frac{\sigma}{C}\frac{\partial^2 q}{\partial x^2} \tag{5}$$

Implementing (5) in a computer requires some difference approximations for the partial differentials. The most obvious (but not the only) approach is the basic definition of elementary calculus. For the time derivative, this is

$$\frac{\partial q}{\partial t} \approx \frac{q(t+\Delta t) - q(t)}{\Delta t} \tag{6a}$$

It is convenient to use a subscript notation that allows (6a) to be compacted into

$$\frac{\partial q}{\partial t} \approx \frac{q_{t+1} - q_t}{\Delta t} \tag{6b}$$

In this notation $t+\Delta t$ is abbreviated by $t+1$, a convenience for more complicated equations. 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 equivalent as Δt tends to zero, but the more symmetrical arrangement will be more accurate when Δt is not zero. Using superscripts to describe x-dependence gives a finite-difference approximation to the second space derivative:

$$\frac{\partial^2 q}{\partial x^2} \approx \frac{q^{x+1} - 2\,q^x + q^{x-1}}{\Delta x^2} \tag{7}$$

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

$$\frac{q_{t+1}^x - q_t^x}{\Delta t} = \frac{\sigma}{C}\frac{q_t^{x+1} - 2\,q_t^x + q_t^{x-1}}{(\Delta x)^2} \tag{8}$$

Letting $\alpha = \sigma\,\Delta t/(C\,\Delta x^2)$ (8) can be arranged thus:

$$q_{t+1}^x - q_t^x - \alpha\,(q_t^{x+1} - 2\,q_t^x + q_t^{x-1}) = 0 \tag{9}$$

Equation (9) can be interpreted geometrically as a computational star in the (x, t)-plane, as depicted in figure 1. 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 other possible arrangements of initial and side conditions,

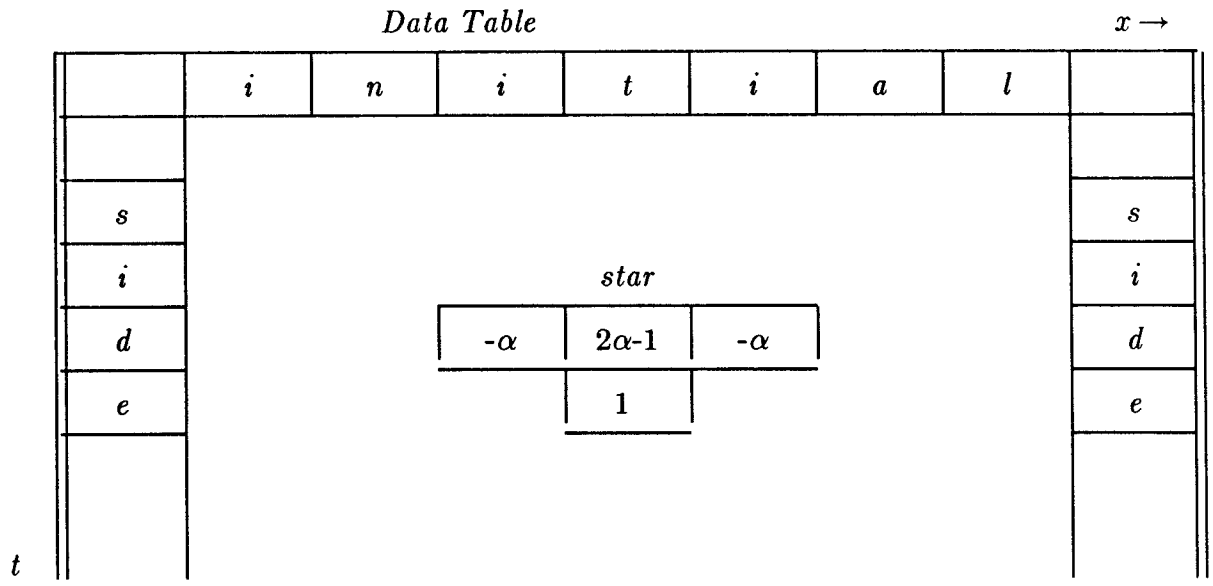


FIG. 2.2-1. Differencing star and table for one-dimensional heat-flow equation.

such as zero-value side conditions. Next is a computer program and a test example.

```
# 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 =",f4.2)') alpha
        do ix=1,6;      q(ix) = 0.      # Initial temperature step
        do ix=7,12;     q(ix) = 1.
        do it=1,6 {
                write(6,'(20f5.2)') (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)
                }
        }
stop;   end
```

alpha = .33

.00	.00	.00	.00	.00	.00	1.00	1.00	1.00	1.00	1.00	1.00
.00	.00	.00	.00	.00	.33	.67	1.00	1.00	1.00	1.00	1.00
.00	.00	.00	.00	.11	.33	.67	.89	1.00	1.00	1.00	1.00
.00	.00	.00	.04	.15	.37	.63	.85	.96	1.00	1.00	1.00
.00	.00	.01	.06	.19	.38	.62	.81	.94	.99	1.00	1.00
.00	.00	.02	.09	.21	.40	.60	.79	.91	.98	1.00	1.00

alpha = .67

.00	.00	.00	.00	.00	.00	1.00	1.00	1.00	1.00	1.00	1.00
.00	.00	.00	.00	.00	.67	.33	1.00	1.00	1.00	1.00	1.00
.00	.00	.00	.00	.44	.00	1.00	.56	1.00	1.00	1.00	1.00
.00	.00	.00	.30	-.15	.96	.04	1.15	.70	1.00	1.00	1.00
.00	.00	.20	-.20	.89	-.39	1.39	.11	1.20	.80	1.00	1.00
.13	.13	-.20	.79	-.69	1.65	-.65	1.69	.21	1.20	.87	.87

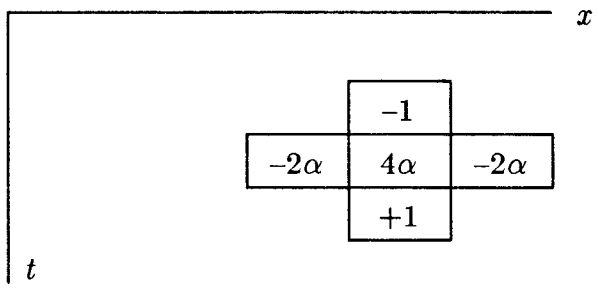
The Leapfrog Method

The difficulty with the given program is that it doesn't work for all possible numerical values of α. You can see that when α is too large (when Δx 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 done in Section 2.8. At wavelengths long compared to Δx or Δt, 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 $t + 1$:

$$\frac{\partial q}{\partial t} \approx \frac{q_{t+1} - q_{t-1}}{2\,\Delta t} \tag{10}$$

The resulting leapfrog differencing star is



(11)

Here the result is even worse. A later analysis shows that the solution is now divergent for *all* real numerical values of α. 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 α.

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.

The Crank-Nicolson Method

The Crank-Nicolson method solves both the accuracy and the stability problem.

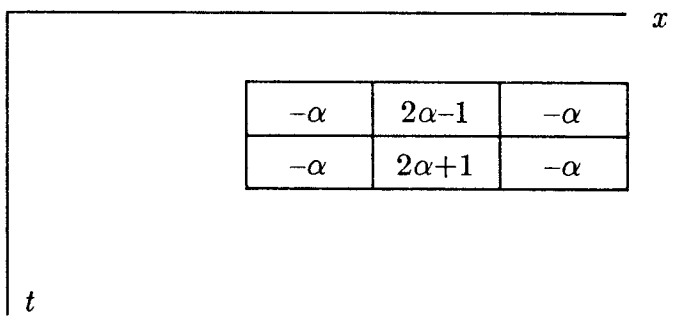
The heat-flow equation (6b) was represented as

$$q_{t+1}^{x} - q_t^{x} = a \left(q_t^{x+1} - 2\, q_t^{x} + q_t^{x-1} \right)$$

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

$$q_{t+1}^{x} - q_t^{x} = \frac{a}{2} \left[\left(q_t^{x+1} - 2q_t^{x} + q_t^{x-1} \right) + \left(q_{t+1}^{x+1} - 2q_{t+1}^{x} + q_{t+1}^{x-1} \right) \right] \quad (12a)$$

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



(12b)

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 (12a) to the left and the t terms to the right, obtaining

$$-\alpha q_{t+1}^{x+1} + (1+2\alpha)q_{t+1}^{x} - \alpha q_{t+1}^{x-1} = \alpha q_t^{x+1} + (1-2\alpha)q_t^{x} + \alpha q_t^{x-1} \qquad (13a)$$

Taking all the $t+1$ values to be unknown, while all the t values are known the right side of (13a) is known, say, d_t^x, and the left side is a set of simultaneous equations for the unknown q_{t+1}. In other words, (13a) does not give us each q_{t+1}^x *explicitly*. They are given *implicitly* by the solution of simultaneous equations. If the x-axis is limited to five points, these equations are

$$\begin{bmatrix} e_{lf} & -\alpha & 0 & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & 0 & 0 \\ 0 & -\alpha & 1+2\alpha & -\alpha & 0 \\ 0 & 0 & -\alpha & 1+2\alpha & -\alpha \\ 0 & 0 & 0 & -\alpha & e_{rt} \end{bmatrix} \begin{bmatrix} q_{t+1}^1 \\ q_{t+1}^2 \\ q_{t+1}^3 \\ q_{t+1}^4 \\ q_{t+1}^5 \end{bmatrix} = \begin{bmatrix} d_t^1 \\ d_t^2 \\ d_t^3 \\ d_t^4 \\ d_t^5 \end{bmatrix} \qquad (13b)$$

The values e_{lf} and e_{rt} 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 (13b) 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 (9). In fact, this implicit method turns out to be cheaper, since the increased accuracy of (13a) over (9) allows the use of a much larger numerical choice of Δt . A program that demonstrates the stability of the method, even for large Δt, is given next.

A tridiagonal simultaneous equation solving subroutine is used. It is explained subsequently.

```
# Implicit heat-flow equation
real q(12),d(12),e(12),f(12)
nx=12; a = 8.;      write(6,'(/"a =",f4.2)') 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,'(20f5.2)') (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,e,f)
      }
stop;  end

# real tridiagonal equation solver
subroutine rtris(n,endl,a,b,c,endr,d,q,e,f)
real q(n),d(n),f(n),e(n),a,b,c,den,endl,endr
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
```

```
a =8.00
 .00   .00   .00   .00   .00   .00  1.00  1.00  1.00  1.00  1.00  1.00
 .17   .17   .21   .30   .47   .76   .24   .53   .70   .79   .83   .83
 .40   .40   .42   .43   .40   .24   .76   .60   .57   .58   .60   .60
 .44   .44   .44   .44   .48   .68   .32   .52   .56   .56   .56   .56
```

Solving Tridiagonal Simultaneous Equations

Much of the world's 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

$$a_j \; q_{j+1} + b_j \; q_j + c_j \; q_{j-1} \;=\; d_j \tag{14}$$

Introduce new unknowns e_j and f_j, along with an equation

$$q_j \;=\; e_j \; q_{j+1} + f_j \tag{15}$$

Write (15) with shifted index:

$$q_{j-1} \;=\; e_{j-1} \, q_j + f_{j-1} \tag{16}$$

Insert (16) into (14):

$$a_j \; q_{j+1} + b_j \; q_j + c_j \; (e_{j-1} \, q_j + f_{j-1}) \;=\; d_j \tag{17}$$

Now rearrange (17) to resemble (15):

$$q_j = \frac{-a_j}{b_j + c_j e_{j-1}} q_{j+1} + \frac{d_j - c_j f_{j-1}}{b_j + c_j e_{j-1}} \tag{18}$$

Compare (18) to (15) to see recursions for the new unknowns e_j and f_j:

$$e_j = \frac{-a_j}{b_j + c_j e_{j-1}} \tag{19a}$$

$$f_j = \frac{d_j - c_j f_{j-1}}{b_j + c_j e_{j-1}} \tag{19b}$$

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 (15) 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 (19a) to compute all the e_j.

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

$$c_{n-1} q_{n-1} + e_{rt} q_n = d_n \tag{20}$$

Equation (20) includes as special cases the zero-value and zero-slope boundary conditions. Equation (20) can be compared to equation (16) at its end.

$$q_{n-1} = e_{n-1} q_n + f_{n-1} \tag{21}$$

Both q_n and q_{n-1} are unknown, but in equations (20) and (21) we have two equations, so the solution is easy. The final step is to take the value of q_n and use it in (16) to compute q_{n-1}, q_{n-2}, q_{n-3}, etc.

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).

The $\partial^3/\partial x^2\,\partial z$-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

$$\frac{1}{\Delta x^2\,\Delta z}\begin{array}{|c|c|c|}\hline -1 & 2 & -1 \\ \hline 1 & -2 & 1 \\ \hline \end{array} \tag{22}$$

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

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 (13a). Introducing the abbreviation $\delta_{xx}\, q = q^{x+1} - 2q^x + q^{x-1}$, equation (13a) becomes

$$\left(1 - \alpha\,\delta_{xx}\right)\, Q_{t+1} \;=\; \left(1 + \alpha\,\delta_{xx}\right)\, Q_t \tag{23}$$

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 α denote the numerical constant in (23), the equation for stepping forward in time is

$$\left[\,1 - \alpha\,(\delta_{xx} + \delta_{yy})\,\right]\, Q_{t+1} \;=\; \left[1 + \alpha(\delta_{xx} + \delta_{yy})\,\right]\, Q_t \tag{24}$$

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

$$\begin{array}{cccc} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{array} \tag{25}$$

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 (25) with vector components by the column arrangement

$$\begin{array}{cccc} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{array} \tag{26}$$

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

$$\begin{array}{ccc} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{array} \tag{27}$$

Lay this star down in the (x, y)-plane (26) 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 (27) over the 1 in the upper left corner of (26). Observe 1's on the 2 and the 5. Copy the 1's into the top row of table 1 into the second and fifth columns. Then put the –4 in (27) over the 2 in (26). Observe 1's on the 1, 3, and 6. Copy the 1's into the next row of table 1. Then put the –4 over the 3. Observe 1's on the 2, 4, and 7. Continue likewise. The 16×16 square matrix that results is shown in table 1.

Now that table 1 has been constructed we can return to the interpretation of equation (24). 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 16×16 matrix. Clearly, the matrix contains zeroes everywhere that table 1 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 (16). 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.

-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 2.2-1. The two-dimensional matrix of coefficients for the Laplacian operator.

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 ω in the form of (12b).

2.3 Monochromatic Wave Programs

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

> *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 first problems relate to the computer program in figure 1. 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 (when you are the only user on the computer).

Analysis of Film Loop Program

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 the program in figure 1, 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

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

The pulses are a superposition of sinusoids of *nw* frequencies, namely, $\Delta\omega$, $2\,\Delta\omega$, ..., nw $\Delta\omega$. The lowest frequency dw $= \Delta\omega$ has a wavelength inverse to *lambda*. Thus the definition

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

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

$$\text{dt} \quad = \quad \text{pi2} \,/\, (\, \text{nt} * \text{dw} \,)$$

The differential equation solved by the program is

$$\frac{\partial P}{\partial z} = \frac{i\,\omega}{v(x,z)}\,P + \frac{v}{-i\,\omega\,2}\,\frac{\partial^2 P}{\partial x^2} \tag{1}$$

For each Δz-step the calculation is done in two stages. The first stage is to solve

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

Using the Crank-Nicolson differencing method this becomes

$$\frac{q_{z+1}^{x} - q_z^x}{\Delta z} =$$

$$= \frac{v}{-i\,\omega\,2}\left(\frac{q_z^{x+1} - 2\,q_z^x + q_z^{x-1}}{2\,\Delta x^2} + \frac{q_{z+1}^{x+1} - 2\,q_{z+1}^x + q_{z+1}^{x-1}}{2\,\Delta x^2} \right)$$

Absorb all the constants into one and define

$$\alpha = \frac{v\,\Delta z}{-i\,\omega\,4\,\Delta x^2} \tag{3}$$

getting

$$q_{z+1}^x - q_z^x = \alpha\left[(q_z^{x+1} - 2\,q_z^x + q_z^{x-1}) + (q_{z+1}^{x+1} - 2\,q_{z+1}^x + q_{z+1}^{x-1}) \right]$$

Bring the unknowns to the left:

$$-\alpha q_{z+1}^{x+1} + (1+2\alpha) q_{z+1}^x - \alpha q_{z+1}^{x-1} = \alpha q_z^{x+1} + (1-2\alpha) q_z^x + \alpha q_z^{x-1} \tag{4}$$

The second stage is to solve the equation

$$\frac{\partial Q}{\partial z} = \frac{i\,\omega}{v}\,Q \tag{5}$$

analytically by

$$Q(z+\Delta z) = Q(z)\,e^{i\,\frac{\omega}{v}\,\Delta z} \tag{6}$$

The program closely follows the notation of equations (3), (4), and (6).

To make a wave pulse, some frequency components are added together. In this program, only two frequencies nw=2 were used. If you try a single frequency nw=1 several things become less clear. Waves reflected at side boundaries (see especially exercise 2) look more like *standing waves.* If you try more frequencies, the program will take longer, but you might like the movie better, because the quiet zones between the pulses will get longer and quieter. Frequency components can be weighted differently.

```
# Wave field extrapolation program
implicit undefined (a-z)
complex cd(48),ce(48),cf(48),q(48),aa,a,b,c,cshift
real p(96,48,12),phase,pi2,dx,dz,v,z0,x0,dt,dw,lambda,w,wov,x
integer ix,nx,iz,nz,iw,nw,it,nt
open(3,file='plot30',status='new',access='direct',form='unformatted',recl=1)

nt=12; nx=48; nz=96; dx=2; dz=1; pi2=2.*3.141592
v=1; lambda=nz*dz/4; dw=v*pi2/lambda; dt=pi2/(nt*dw); nw=2

do iz=1,nz;  do ix=1,nx;  do it=1,nt  { p(iz,ix,it) = 0. }
do iw = 1,nw {                          # superimpose nw frequencies
        w = iw*dw;      wov = w/v               # frequency / velocity
        x0 = nx*dx/3; z0 = nz*dz/3
        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 = dz/(4.*(0.,-1.)*wov*dx**2)         # 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,ce,cf)
                        # "ctris" solves complex tridiagonal equations
                        # i.e. "rtris" with complex variables
                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
                        }
                }
        }
write(3,rec=1) (((p(iz,ix,it),iz=1,nz),ix=1,nx),it=1,nt)
stop;           end
```

FIG. 2.3-1. Computer program to make a movie of a sum of monochromatic waves. (Lynn, Gonzalez, JFC, Hale)

Phase Shift

Theory predicts that in two dimensions waves going through a focus suffer a 90° phase shift. You should be able to notice that a symmetrical waveform is incident on the focus, but an asymmetrical waveform emerges. (This is best seen in figure 6, but is clearer in a movie). In migrations, waves go just *to* a focus, not *through* it. So the migration impulse response in two

dimensions carries a 45° phase shift. Even though real life is three dimensional, the two dimensional response is appropriate for migrating seismic lines where focusing is presumed to arise from cylindrical, not spherical, reflectors.

Lateral Velocity Variation

Lateral velocity variation $v = v(x)$ has not been included in the program, but it is not difficult to install. It enters in two places. It enters first in equation (6). If the data is such that k_x is small enough to be neglectable, then equation (6) is the only place it is needed. Second, it enters in the tridiagonal coefficients. The so-called thin-lens approximation of optics seems to amount to including the equation (6) part only.

Side-Boundary Analysis

In geophysics, we usually wish the side-boundary question would go away. 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 about them. The program of figure 1 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

```
endl = endr = 10^30  ≈  ∞
```

The above approach is slightly wasteful of computer memory, because the end zero is stored, and the zero slope is explicitly visible as two identical traces. This waste is avoided in Dave Hale's coding of the boundary conditions as given, but not derived, below:

```
q0 = bl * q(1);        qnxp1 = br * q(nx)

cd(1)  = aa * q(2)    + ( 1. - 2. * aa ) * q(1) + aa * q0
cd(nx) = aa * q(nx-1) + ( 1. - 2. * aa ) * q(nx) + aa * qnxp1

endl = c * bl + b;
endr = a * br + b

call ctris(nx,endl,a,b,c,endr,cd,q,ce,cf)
```

Note that bl = br = 0 for zero-value boundaries, and bl = br = 1 for zero-slope boundaries. Absorbing side boundaries, derived in Section 4.4, are

obtained by letting bl and br be complex.

Variations on the Film Loop Program

Keep a record of your progress through these exercises. It will be helpful when preparing for the final exam. And several years hence you will be able to refresh your memory.

Get a three-ring notebook. Cut all plots and program listings to 8-*1/2* by 11 size and three-hole punch them. If algebraic analysis is required, do it on the same size paper. Avoid leaving important bits of analysis on scraps of paper. Either keep this material with your lecture notes or maintain it as a laboratory notebook, filing consistently by date.

For each of these exercises, hand in a program listing and a plot of the first frame.

1. Specify program changes that give an initial plane wave propagating downward at an angle of 15° to the right of vertical.

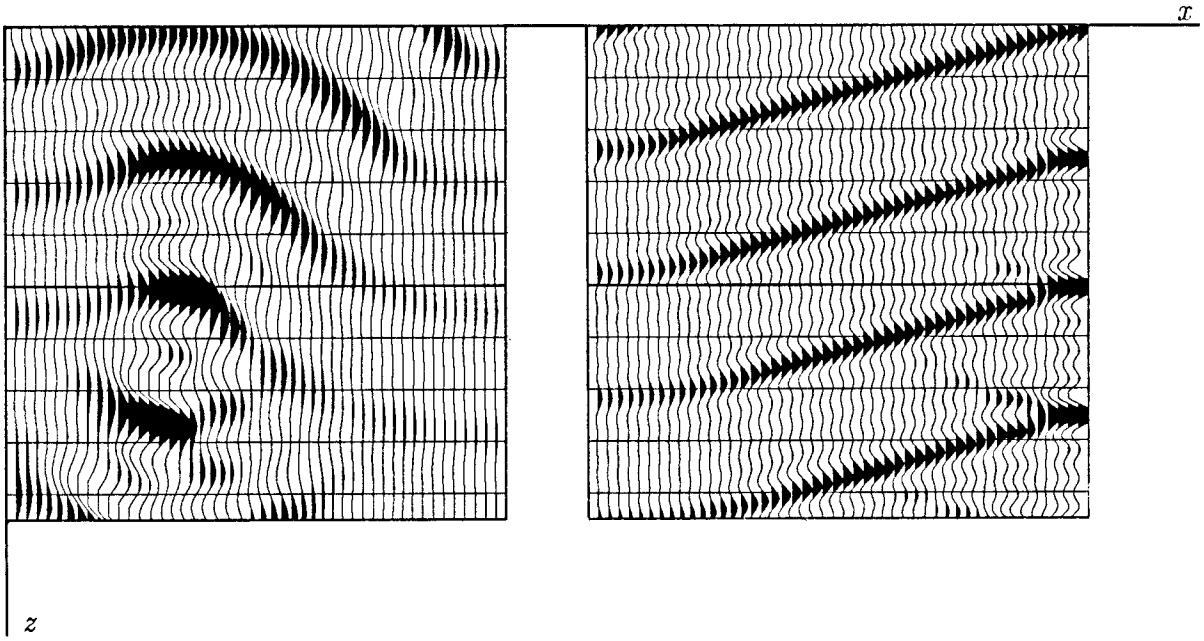


FIG. 2.3-2. Left, first frame of movie generated by figure 1. Right, solution to exercise 1. (Li Zhiming).

2. Given that the domain of computation is $0 < x <= \text{xmax}$ and $0 < z \leq \text{zmax}$, how would you modify the initial conditions at $z = 0$ to simulate a point source at $(x, z) = (\text{xmax}/3, -\text{zmax}/2)$? Try it.
3. Modify the program so that zero-slope side boundaries are replaced by zero-value side boundaries.

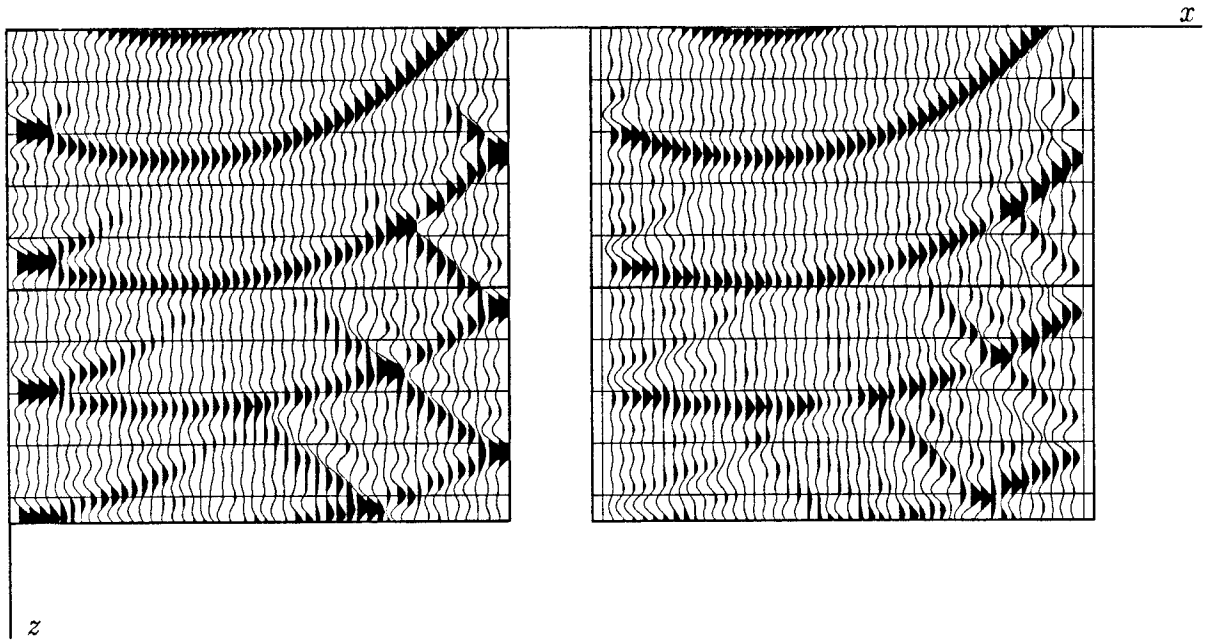


FIG. 2.3-3. Left, exercise 2, expanding spherical wave. Right, exercise 3, zero-value side boundaries. (Li Zhiming).

4. Incorporate the 45° term, ∂_{xxz}, for the collapsing spherical wave. Use zero-slope sides. Compare your result with the 15° result obtained via the program in figure 1. Mark an X at the theoretical focus location.
5. Make changes to the program to include a thin-lens term with a lateral velocity change of 40% across the frame produced by a constant slowness gradient. Identify other parts of the program which are affected by lateral velocity variation. You need not make these other changes. Why are they expected to be small?
6. Observe and describe various computational artifacts by testing the program using a point source at $(x, z) = (\text{xmax}/2, 0)$. Such a source is rich in the high spatial frequencies for which difference equations do not mimic their differential counterparts.

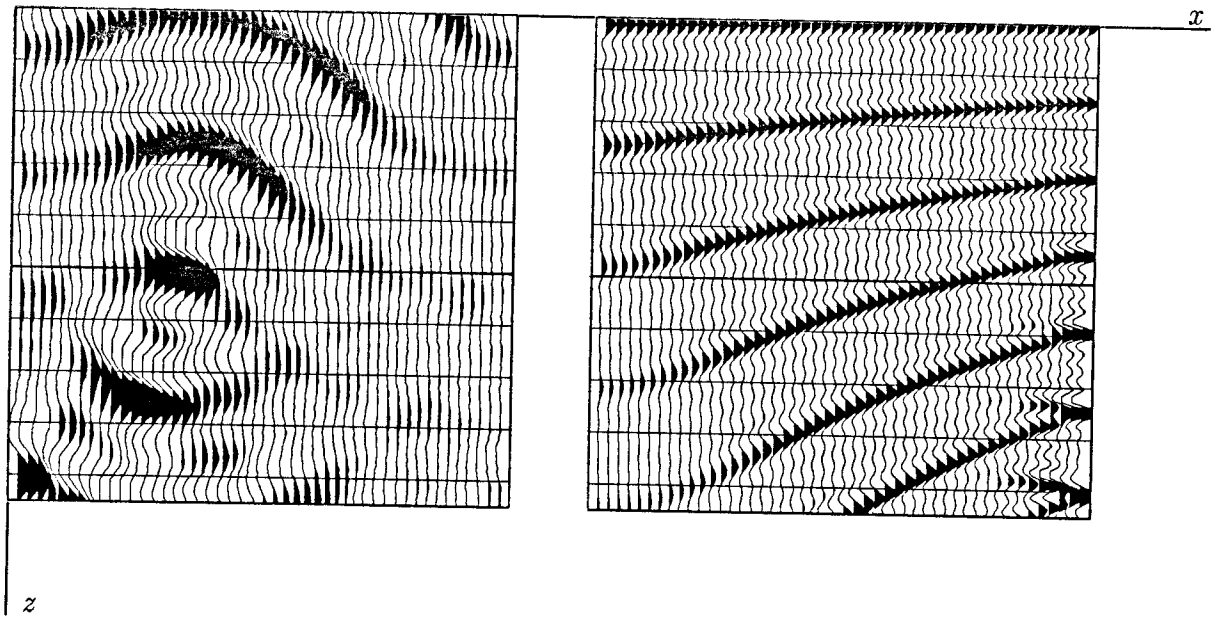


FIG. 2.3-4. Left, exercise 4, 45° term. Right, exercise 5, lateral velocity variation. (Li Zhiming).

7. Section 4.4 explains how to absorb energy at the side boundaries. Make the necessary changes to the program.
8. The accuracy of the x-derivative may be improved by a technique that is analyzed later in Section 4.3. Briefly, instead of representing $k_x^2 \Delta x^2$ by the tridiagonal matrix $\mathbf{T}$ with $(-1, 2, -1)$ on the main diagonal, you use $\mathbf{T}/(\mathbf{I}-\mathbf{T}/6)$. Modify the extrapolation analysis by multiplying through by the denominator. Make the necessary changes to the 45° collapsing wave program.

Migration Program in the (ω,x,z)-Domain (Kjartansson, Jacobs)

The migration program is similar to the film loop program. But there are some differences. The film loop program has "do loops" nested four deep. It produces results for many values of t. Migration requires a value only at $t = 0$. So one loop is saved, which means that for the same amount of computer time, the space volume can be increased. Unfortunately, loss of a loop seems also to mean loss of a movie. With ω-domain migration, it seems that the only interesting thing to view is the input and the output.

The input for this process will probably be field data, unlike for the film loop movie, so there will not be an analytic representation in the ω-domain.

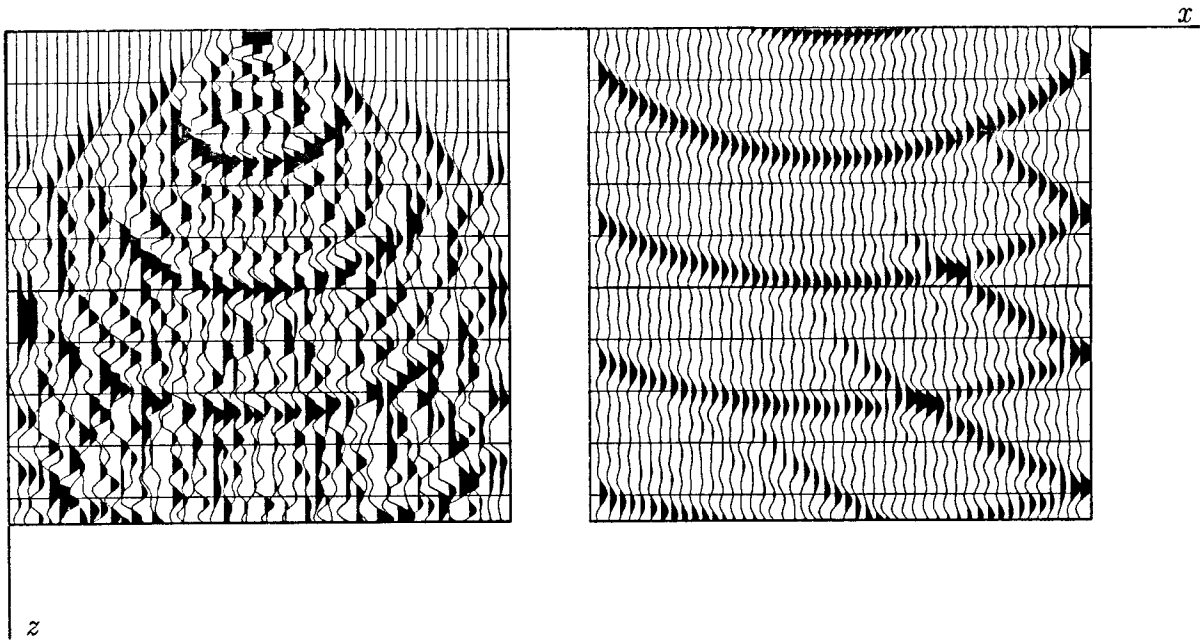


FIG. 2.3-5. Left, exercise 6, computational artifacts with point source. Right, exercise 7, absorbing side. (Li Zhiming).

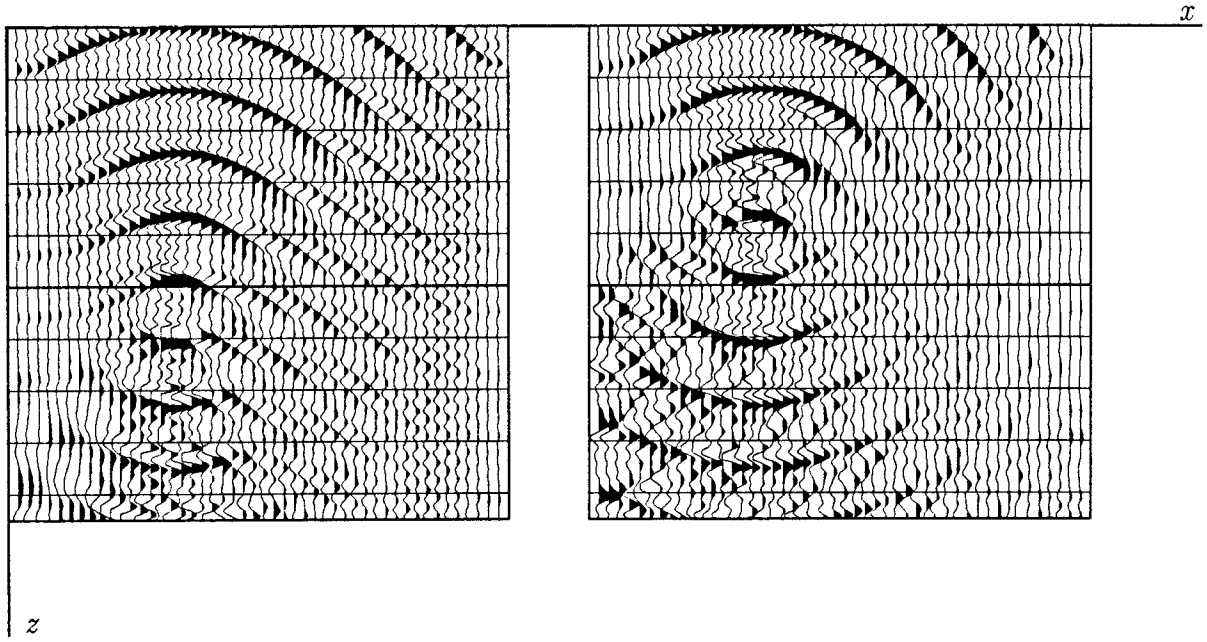


FIG. 2.3-6. Left, exercise 8, without 1/6 trick; right, with 1/6 trick. (Li Zhiming).

The input will be in the time domain and will have to be Fourier transformed. The beginning of the program in figure 6 defines some pulses to simulate field data. The pulses are broadened impulses and should migrate to approximate semicircles. Exact impulses were not used because the departure of difference operators from differential operators would make a noisy mess.

Next the program Fourier transforms the pseudodata from the time domain into the ω-frequency domain.

Then comes the downward continuation of each frequency. This is a loop on depth z and on frequency ω. Either of these loops may be on the inside. The choice can be made for machine-dependent efficiency.

```
# Migration in the (omega,x,z)-domain
real q(48,64),pi2,alpha,dt,dtau,dw
complex cq(48,64),cd(48),ce(48),cf(48),aa,a,b,c,cshift
integer ix,nx,iz,nz,iw,nw,it,nt
open(4,file='plot36',status='new',access='direct',form='unformatted',recl=1)

nt = 64;  nz = nt;   nx = 48;     pi2=2.*3.141592
dt=1.;  dtau=1.;   dw=pi2/(dt*nt);   nw=nt/2;
alpha = .25                                   # alpha = v*v*dtau/(4*dx*dx)
do iz=1,nz;  do ix=1,nx;  { q(ix,iz) = 0.;   cq(ix,iz)=0. }
do it=nt/3,nt,nt/4
        do ix=1,4                             # Broadened impulse source
                { cq(ix,it) = (5.-ix);    cq(ix,it+1) = (5.-ix) }
call rowcc(nx,nt,cq,+1.,+1.)                  # F.T. over time.
do iz = 1,nz {                                # iz and iw loops interchangeable
do iw = 2,nw {                                # iz and iw loops interchangeable
        aa = - alpha /( (0.,-1.)*(iw-1)*dw )
        a = -aa;     b = 1.+2.*aa;     c = -aa
        do ix = 2,nx-1
                cd(ix) = aa*cq(ix+1,iw) + (1.-2.*aa)*cq(ix,iw) + aa*cq(ix-1,iw)
        cd(1) = 0.;     cd(nx) = 0.
        call ctris(nx,-a,a,b,c,-c,cd,cq(1,iw),ce,cf)
        cshift = cexp(cmplx(0.,-(iw-1)*dw*dtau))
        do ix=1,nx
                cq(ix,iw) = cq(ix,iw) * cshift
        do ix = 1,nx
                q(ix,iz) = q(ix,iz)+cq(ix,iw)      # q(t=0) = Σ Q(ω)
        }}
write(4,rec=1) ((q(ix,iz),iz=1,nz),ix=1,nx)
stop;   end
```

FIG. 2.3-7. Migration program in the (ω, x, z)-domain.

For migration an equation for upcoming waves is required, unlike the downgoing wave equation required for the film loop program. Change the sign of the z-axis in equation (1). This affects the sign of *aa* and the sign of the phase of *cshift*.

Another difference with the film loop program is that the input now has a time axis whereas the output is still a depth axis. It is customary and convenient to reorganize the calculation to plot travel-time depth, instead of depth, making the vertical axes on both input and output the same. Using $\tau = z/v$, equivalently $d\tau/dz = 1/v$, the chain rule gives

$$\frac{\partial}{\partial z} = \frac{\partial \tau}{\partial z}\frac{\partial}{\partial \tau} = \frac{1}{v}\frac{\partial}{\partial \tau} \tag{7}$$

Substitution into (1) gives

$$\frac{\partial P}{\partial \tau} = -i\,\omega\,P - \frac{v^2}{-i\,\omega\,2}\frac{\partial^2 P}{\partial x^2} \tag{8}$$

In the program, the time sample size $\mathrm{dt} = \Delta t$ and the travel-time depth sample $\mathrm{dtau} = \Delta\tau$ are taken to be unity, so the maximum frequency is the Nyquist. Notice that the frequency loop covers only the positive frequency axis. The negative frequencies serve only to keep the time function real, a task that is more easily done by simply taking the real part.

The output of the program is shown in figure 8. Mainly, you see semicircle approximations. There are also some artifacts at late time that may be ω-domain wraparounds. The input pulses were apparently sufficiently broadbanded in dip that the figure provides a preview of the fact, to be proved later, that the actual semicircle approximation is an ellipse going through the origin.

Notice that the waveform of the original pulses was a symmetric function of time, whereas the semicircles exhibit a waveform that is neither symmetric nor antisymmetric, but is a 45° phase-shifted pulse. Waves from a point in a three-dimensional world would have a phase shift of 90°. Waves from a two-dimensional exploding reflector in a three-dimensional world have the 45° phase shift.

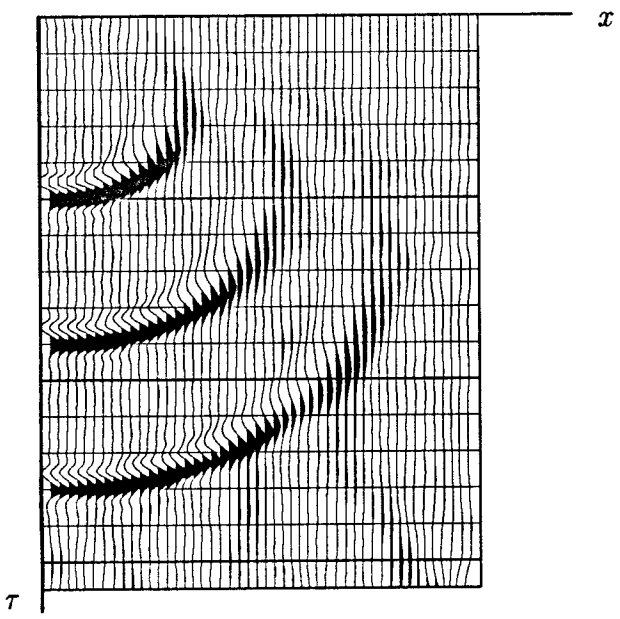


FIG. 2.3-8. Output of figure 7 program: semicircle approximations.

2.4 Splitting and Full Separation

Two processes, **A** and **B**, which ordinarily act simultaneously, may or may not be interconnected. The case where they are independent is called *full separation.* In this case it is often useful, for thought and for computation, to imagine process **A** going to completion before process **B** is begun. Where the processes are interconnected it is possible to allow **A** to run for a short while, then switch to **B**, and continue in alternation. This alternation approach is called *splitting.*

The Heat-Flow Equation

The diffraction or migration equation could be called the "wavefront healing" equation. It smooths back together any lateral breaks in the wavefront that may have been caused by initial conditions or by the lens term. The 15° migration equation has the same mathematical form as the heat-flow equation. But the heat-flow equation has all real numbers, and its physical behavior is more comprehensible. This makes it a worthwhile detour. A two-sentence derivation of it follows. (1) The heat flow H_x in the x-

direction equals the negative of the gradient $-\partial/\partial x$ of temperature T times the heat conductivity σ. (2) The decrease of temperature $-\partial T/\partial t$ is proportional to the divergence of the heat flow $\partial H_x/\partial x$ divided by the heat storage capacity C of the material. Combining these, extending from one dimension to two, taking σ constant and $C=1$, gives the equation

$$\frac{\partial T}{\partial t} = \sigma \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) T \tag{1}$$

Splitting

The splitting method for numerically solving the heat-flow equation is to replace the two-dimensional heat-flow equation by two one-dimensional equations, each of which is used on alternate time steps:

$$\frac{\partial T}{\partial t} = 2\sigma \frac{\partial^2 T}{\partial x^2} \qquad \textit{(all } y\textit{)} \tag{2a}$$

$$\frac{\partial T}{\partial t} = 2\sigma \frac{\partial^2 T}{\partial y^2} \qquad \textit{(all } x\textit{)} \tag{2b}$$

In equation (2a) the heat conductivity σ has been doubled for flow in the x-direction and zeroed for flow in the y-direction. The reverse applies in equation (2b). At odd moments in time heat flows according to (2a) and at even moments in time it flows according to (2b). This solution by alternation between (2a) and (2b) can be proved mathematically to converge to the solution to (1) with errors of the order of Δt. Hence the error goes to zero as Δt goes to zero. The motivation for splitting is the infeasibility of higher-dimensional implicit methods (end of Section 2.2).

Full Separation

Splitting can turn out to be much more accurate than might be imagined. In many cases there is *no* loss of accuracy. Then the method can be taken to an extreme limit. Think about a radical approach to equations (2a) and (2b) in which, instead of alternating back and forth between them at alternate time steps, what is done is to march (2a) through all time steps. Then this intermediate result is used as an initial condition for (2b), which is marched through all time steps to produce a final result. It might seem surprising that this radical method can produce the correct solution to equation (1). But if σ is a constant function of x and y, it does. The process is depicted in figure 1 for an impulsive initial disturbance. A differential equation like (1) is said to be *fully separable* when the correct solution is obtainable by the radical method. It should not be too surprising that full

separation works when σ is a constant, because then Fourier transformation may be used, and the two-dimensional solution $\exp[-\sigma\,(k_x^{\,2} + k_y^{\,2})t\,]$ equals the succession of one-dimensional solutions $\exp(-\sigma\,k_x^{\,2}t\,)\ \exp(-\sigma\,k_y^{\,2}t\,)$. It turns out, and will later be shown, that the condition required for applicability of full separation is that $\sigma\,\partial^2/\partial x^2$ should commute with $\sigma\,\partial^2/\partial y^2$, that is, the order of differentiation should be irrelevant. Technically there is also a boundary-condition requirement, but it creates no difficulty when the disturbance dies out before reaching a boundary.

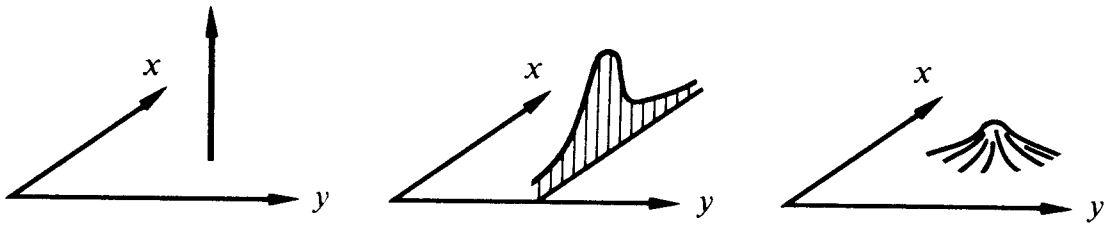


FIG. 2.4-1. Temperature distribution in the $(x\,,\,y\,)$-plane beginning from a delta function (left). After heat is allowed to flow in the x-direction but not in the y-direction the heat is located in a "wall" (center). Finally allowing heat to flow for the same amount of time in the y-direction but not the x-direction gives the same symmetrical Gaussian result that would have been found if the heat had moved in x- and y-directions simultaneously (right).

Surprisingly, no notice is made of full separability in many textbooks on numerical solutions. Perhaps this is because the total number of additions and multiplications is the same whether a solution is found by splitting or by full separation. But as a practical matter, the cost of solving large problems does not mount up simply according to the number of multiplications. When the data base does not fit entirely into the random-access memory, as is almost the definition of a *large problem,* then each step of the splitting method demands that the data base be transposed, say, from $(x\,,\,y\,)$ storage order to $(y\,,\,x\,)$ storage order. Transposing requires no multiplications, but in many environments transposing would be by far the most costly part of the whole computation. So if transposing cannot be avoided, at least it should be reduced to a practical minimum.

There are circumstances which dictate a middle road between splitting and separation — for example, if σ were a slowly variable function of x or y. Then you might find that although $\sigma\,\partial^2/\partial x^2$ does not strictly commute

with $\sigma\,\partial^2/\partial y^2$, it comes close enough that a number of time steps may be made with (2a) before you transpose the data and switch over to (2b). Circumstances like this one but with more geophysical interest arise with the wave-extrapolation equation that is considered next. The significance in seismology of the splitting and full separation concepts was first recognized by Brown [1983].

Application to Lateral Velocity Variation

A circumstance in which the degree of noncommutativity of two differential operators has a simple physical meaning and an obviously significant geophysical application is the so-called monochromatic 15° wave-extrapolation equation in inhomogeneous media. Taking $v \approx \overline{v}$ this equation is

$$\frac{\partial U}{\partial z} = \left\{ \frac{i\omega}{\overline{v}(z)} + i\omega \left[\frac{1}{v(x,z)} - \frac{1}{\overline{v}(z)} \right] - \frac{\overline{v}(z)}{2\,i\omega} \frac{\partial^2}{\partial x^2} \right\} U \quad (3)$$

$$= (\textit{retardation} + \textit{thin lens} + \textit{diffraction})\; U$$

Inspection of (3) shows that the retardation term commutes with the thin-lens term and with the free-space diffraction term. But the thin-lens term and the diffraction term do not commute with one another. In practice it seems best to split, doing the thin-lens part analytically and the diffraction part by the Crank-Nicolson method. Then stability is assured because the stability of each separate problem is known. Also, the accuracy of the analytic solution is an attractive feature. Now the question is, to what degree do these two terms commute?

The problem is just that of focusing a slide projector. Adjusting the focus knob amounts to repositioning the thin-lens term in comparison to the free-space diffraction term. There is a small range of knob positions over which no one can notice any difference, and a larger range over which the people in the back row are not disturbed by misfocus. Much geophysical data processing amounts to downward extrapolation of data. The lateral variation of velocity occurring in the lens term is known only to a limited accuracy. The application could be to determine $v(x)$ by the extrapolation procedure.

For long lateral spatial wavelengths the terms commute. Then diffraction may proceed in ignorance of the lateral variation in v. At shorter wavelengths the diffraction and lensing effects must be interspersed. So the real issue is not merely computational convenience but the interplay between data accuracy and the possible range for velocity in the underlying model.

Application to 3-D Downward Continuation

The operator for migration of zero-offset reflection seismic data in three dimensions is expandable to second order by Taylor series expansion to the so-called 15° approximation

$$\left[\frac{(-i\omega)^2}{v^2} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \right]^{1/2} \approx \frac{-i\omega}{v} - \frac{v}{-2i\omega} \frac{\partial^2}{\partial x^2} - \frac{v}{-2i\omega} \frac{\partial^2}{\partial y^2} \tag{4}$$

The most common case is when v is slowly variable or independent of x and y. Then the conditions of full separation do apply. This is good news because it means that we can use ordinary 2-D wave-extrapolation programs for 3-D, doing the in-line data and the out-of-line data in either order. The bad news comes when we try for more accuracy. Keeping more terms in the Taylor series expansion soon brings in the cross term $\partial^4/\partial x^2 \partial y^2$. Such a term allows neither full separation nor splitting. Fortunately, present-day marine data-acquisition techniques are sufficiently crude in the out-of-line direction that there is little justification for out-of-line processing beyond the 15° equation. Francis Muir had the good idea of representing the square root as

$$\left[\frac{(-i\omega)^2}{v^2} - \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \right]^{1/2} \approx$$

$$\approx \left[\frac{(-i\omega)^2}{v^2} - \frac{\partial^2}{\partial x^2} \right]^{1/2} - \frac{v}{-2i\omega} \frac{\partial^2}{\partial y^2} \tag{5}$$

There may be justification for better approximations with land data. Fourier transformation of at least one of the two space axes will solve the computational problem. This should be a good approach when the medium velocity does not vary laterally so rapidly as to invalidate application of Fourier transformation.

Separability of 3-D Migration (the Jakubowicz Justification)

In an operations environment, 3-D is much harder to cope with than 2-D. Therefore, it may be expedient to suppose that 3-D migration can be achieved merely by application of 2-D migration twice, once in the x-direction and once in the y-direction. The previous section would lead you to believe that such an expedient process would result in a significant degradation of accuracy. In fact, the situation is *much* better than might be supposed. It has been shown by Jakubowicz and Levin [1983] that, wonder of wonders, for a constant-velocity medium, the expedient process is exact.

The explanation is this: migration consists of more than downward continuation. It also involves imaging, that is, the selection of data at $t=0$. In

principle, downward continuation is first completed, for both the x and the y directions. After that, the imaging condition is applied. In the expedient process there are four steps: downward continuation in x, imaging, downward continuation in y, and finally a second imaging. Why it is that the expedient procedure gives the correct result seems something of a puzzle, but the validity of the result is easy to demonstrate.

First note that substitution of (6) into (7) gives (8) where

$$t_1^2 = t_0^2 + (x-x_0)^2/v^2 \tag{6}$$

$$t^2 = t_1^2 + (y-y_0)^2/v^2 \tag{7}$$

$$t^2 = t_0^2 + (x-x_0)^2/v^2 + (y-y_0)^2/v^2 \tag{8}$$

Equation (8) represents travel time to an arbitrary point scatterer. For a 2-D survey recorded along the y-axis, i.e., at constant x, equation (7) is the travel-time curve. In-line hyperbolas cannot be distinguished from sideswipe hyperbolas. 2-D migration with equation (7) brings the energy up to t_1. Subsequently migrating the other direction with equation (6) brings the energy up the rest of the way to t_0. This is the same result as the one given by the more costly 3-D procedure migrating with (8).

The Jakubowicz justification is somewhat more mathematical, but may be paraphrased as follows. First note that substitution of (9) into (10) gives (11) where

$$k_\tau^2 = \omega^2 - v^2 k_x^2 \tag{9}$$

$$k_z^2 = \frac{k_\tau^2}{v^2} - k_y^2 \tag{10}$$

$$k_z^2 = \frac{\omega^2}{v^2} - k_x^2 - k_y^2 \tag{11}$$

Two-dimensional Stolt migration over x may be regarded as a transformation from travel-time depth t to a pseudodepth τ by use of equation (9). The second two-dimensional migration over y may be regarded as a transformation from pseudodepth τ to true depth z by use of equation (10). The composite is the same as equation (11), which depicts 3-D migration.

The validity of the Jakubowicz result goes somewhat beyond its proof. Our two-dimensional geophysicist may be migrating other offsets besides zero offset. (In Chapter 3 nonzero-offset data is migrated). If a good job is done, all the reflected energy moves up to the apex of the zero-offset hyperbola. Then the cross-plane migration can handle it if it can handle zero offset. So offset is not a problem. But can a good job be done of bringing all the energy

up to the apex of the zero-offset hyperbola?

Difficulty arises when the velocity of the earth is depth-dependent, as it usually is. Then the Jakubowicz proof fails, and so does the expedient 3-D method. With a 2-D survey you have the problem that the sideswipe planes require a different migration velocity than the vertical plane. Rays propagating to the side take longer to reach the high-velocity media deep in the earth. So sideswipes usually require a lower migration velocity. If you really want to do three-dimensional migration with $v(z)$, you should forget about separation and do it the hard way. Since we know how to transpose (Section 1.6), the hard way really isn't much harder.

Separability in Shot-Geophone Space

Reflection seismic data gathering is done on the earth's surface. One can imagine the appearance of the data that would result if the data were generated and recorded at depth, that is, with deeply buried shots and geophones. Such buried data could be synthesized from surface data by first downward extrapolating the geophones, then using the reciprocal principle to interchange sources and receivers, and finally downward extrapolating the surface shots (now the receivers). A second, equivalent approach would be to march downward in steps, alternating between shots and geophones. This latter approach is developed in Chapter 3, but the result is simply stated by the equation

$$\frac{\partial U}{\partial z} = \left\{ \left[\frac{(-i\omega)^2}{v(s)^2} - \frac{\partial^2}{\partial s^2} \right]^{1/2} + \left[\frac{(-i\omega)^2}{v(g)^2} - \frac{\partial^2}{\partial g^2} \right]^{1/2} \right\} U \qquad (12)$$

The equivalence of the two approaches has a mathematical consequence. The shot coordinate s and the geophone coordinate g are independent variables, so the two square-root operators commute. Thus the same solution is obtained by splitting as by full separation.

Validity of the Splitting and Full-Separation Concepts

When Fourier transformation is possible, extrapolation operators are complex numbers like $e^{ik_z z}$. With complex numbers a and b there is never any question that $ab = ba$. Then both splitting and full separation are always valid, but the proof will be given only for a more general arrangement.

Suppose Fourier transformation has not been done, or could not be done because of some spatial variation of material properties. Then extrapolation operators are built up by combinations of the finite-differencing operators described in previous sections. Let $\mathbf{A}$ and $\mathbf{B}$ denote two such operators.

For example, $\mathbf{A}$ could be a matrix containing the second x differencing operator. Seen as matrices, the boundary conditions of a differential operator are incorporated in the corners of the matrix. The bottom line is whether $\mathbf{A}\,\mathbf{B} = \mathbf{B}\,\mathbf{A}$, so the question clearly involves the boundary conditions as well as the differential operators.

Extrapolation forward a short distance can be done with the operator $(\mathbf{I}+\mathbf{A}\,\Delta z)$. In two-dimensional problems $\mathbf{A}$ was seen to be a four-dimensional matrix. For convenience the terms of the four-dimensional matrix can be arranged into a super-large, ordinary two-dimensional matrix. Implicit finite-differencing calculations gave extrapolation operators like $(\mathbf{I}+\mathbf{A}\,\Delta z)/(\mathbf{I}-\mathbf{A}\,\Delta z)$. Let $\mathbf{p}$ denote a vector where components of the vector designate the wavefield at various locations. As has been seen, the locations need not be constrained to the x-axis but could also be distributed throughout the (x, y)-plane. Numerical analysis gives us a matrix operator, say $\mathbf{A}$, which enables us to project forward, say,

$$\mathbf{p}(z + \Delta z) \quad = \quad \mathbf{A}_1\, \mathbf{p}(z)$$

The subscript on $\mathbf{A}$ denotes the fact that the operator may change with z. To get a step further the operator is applied again, say,

$$\mathbf{p}(z + 2\,\Delta z) \quad = \quad \mathbf{A}_2\, [\mathbf{A}_1\, \mathbf{p}(z)]$$

From an operational point of view the matrix $\mathbf{A}$ is never squared, but from an analytical point of view, it really is squared.

$$\mathbf{A}_2\, [\mathbf{A}_1\, \mathbf{p}(z)] \quad = \quad (\mathbf{A}_2\, \mathbf{A}_1)\, \mathbf{p}(z)$$

To march some distance down the z-axis we apply the operator many times. Take an interval $z_1 - z_0$, to be divided into N subintervals. Since there are N intervals, an error proportional to $1/N$ in each subinterval would accumulate to an unacceptable level by the time z_1 was reached. On the other hand, an error proportional to $1/N^2$ could only accumulate to a total error proportional to $1/N$. Such an error would disappear as the number of subintervals increased.

To prove the validity of splitting, we take $\Delta z = (z_1 - z_0)/N$. Observe that the operator $\mathbf{I}+(\mathbf{A}+\mathbf{B})\Delta z$ differs from the operator $(\mathbf{I}+\mathbf{A}\,\Delta z)(\mathbf{I}+\mathbf{B}\,\Delta z)$ by something in proportion to Δz^2 or $1/N^2$. So in the limit of a very large number of subintervals, the error disappears.

It is much easier to establish the validity of the full-separation concept. Commutativity is whether or not $\mathbf{A}\,\mathbf{B} = \mathbf{B}\,\mathbf{A}$. Commutativity is always true for scalars. With finite differencing the question is whether the two matrices commute. Taking $\mathbf{A}$ and $\mathbf{B}$ to be differential operators,

commutativity is defined with the help of the family of all possible wavefields P. Then $\mathbf{A}$ and $\mathbf{B}$ are commutative if $\mathbf{A}\,\mathbf{B}\,P = \mathbf{B}\,\mathbf{A}\,P$.

The operator representing $\partial P/\partial z$ will be taken to be $\mathbf{A}+\mathbf{B}$. The simplest numerical integration scheme using the splitting method is

$$P(z_0 + \Delta z) \;=\; (\mathbf{I} + \mathbf{A}\,\Delta z)\,(\mathbf{I} + \mathbf{B}\,\Delta z)\,P(z_0) \tag{13}$$

Applying (13) in many stages gives a product of many operators. The operators $\mathbf{A}$ and $\mathbf{B}$ are subscripted with j to denote the possibility that they change with z.

$$P(z_1) \;=\; \prod_{j=1}^{N} [(\mathbf{I} + \mathbf{A}_j\,\Delta z)(\mathbf{I} + \mathbf{B}_j\,\Delta z)]\,P(z_0) \tag{14}$$

As soon as $\mathbf{A}$ and $\mathbf{B}$ are assumed to be commutative, the factors in (14) may be rearranged at will. For example, the $\mathbf{A}$ operator could be applied in its entirety before the $\mathbf{B}$ operator is applied:

$$P(z_1) \;=\; \left[\prod_{j=1}^{N} (\mathbf{I} + \mathbf{B}\,\Delta z)\right] \left[\prod_{j=1}^{N} (\mathbf{I} + \mathbf{A}\,\Delta z)\right] P(z_0) \tag{15}$$

Thus the full-separation concept is seen to depend on the commutativity of operators.

EXERCISES

1. With a splitting method, Ma Zaitian (Ma [1981]) showed how very wide-angle representations may be implemented with successive applications of an equation like a 45° equation. This avoids the band matrix solving inherent in the high-order Muir expansion. Specifically, one chooses coefficients a_j and b_j, in the square-root fitting function

$$i\,k_z \;=\; \sum_{j=1}^{n-1} \frac{k_x^{\,2}\,b_j}{-i\,\omega + a_j\,ik_x}$$

   The general n^{th}-order case is somewhat complicated, so your job is simply to find a_1, a_2, b_1, and b_2, to make the fitting function match the 45° equation.

2. Migrate a two dimensional data set with velocity v_1. Then migrate the migrated data set with a velocity v_2. Rocca pointed out that this double migration simulates a migration with a third velocity v_3. Using a method of deduction similar to the Jakubowicz deduction equations (9), (10), and (11) find v_3 in terms of v_1 and v_2.

3. Consider migration of zero-offset data $P(x, y, t)$ recorded in an area of the earth's surface plane. Assume a computer with a random access memory (RAM) large enough to hold several planes (any orientation) from the data volume. (The entire volume resides in slow memory devices). Define a migration algorithm by means of a program sketch (such as in Section 1.3). Your method should allow velocity to vary with depth.

2.5 Recursive Dip Filters

Recursive filtering is a form of filtering where the output of the filter is fed back as an input. This can achieve a long impulse response for a tiny computational effort. It is particularly useful in computing a running mean. A running mean could be implemented as a low-pass filter in the frequency domain, but it is generally much better to avoid transform space. Physical space is cheaper, it allows for variable coefficients, and it permits a more flexible treatment of boundaries. Geophysical datasets are rarely stationary over long distances in either time or space, so recursive filtering is particularly helpful in statistical estimation.

The purpose of most filters is to make possible the observation of important weak events that are obscured by strong events. *One*-dimensional filters can do this only by the selection or rejection of *frequency* components. In *two* dimensions, a different criterion is possible, namely, selection by *dip*.

Dip filtering is a process of long-standing interest in geophysics (Embree, Burg, and Backus [1963]). Steep dips are often ground-roll noise. Horizontal dips can also be noise. For example, weak fault diffractions carry valuable information, but they may often be invisible because of the dominating presence of flat layers.

To do an ordinary dip-filtering operation ("pie slice"), you simply transform data into (ω, k)-space, multiply by any desired function of k/ω, and transform back. Pie-slice filters thus offer complete control over the filter response in k/ω dip space. While the recursive dip filters are not controlled so easily, they do meet the same general needs as pie-slice filters and offer the

additional advantages of

1. time- and space-variability
2. causality
3. ease of implementation
4. orders of magnitude more economic than (ω, k)-implementation

The causality property offers an interesting opportunity during data recording. Water-velocity rejection filters could be built into the recording apparatus of a modern high-density marine cable.

Definition of a Recursive Dip Filter

Let P denote raw data and Q denote filtered data. When seismic data is quasimonochromatic, dip filtering can be achieved with spatial frequency filters. The table below shows filters with an adjustable cutoff parameter α.

Dip Filters for Monochromatic Data ($\omega \approx$ *Const*)	
Low Pass	*High Pass*
$Q = \dfrac{\alpha}{\alpha + k^2} P$	$Q = \dfrac{k^2}{\alpha + k^2} P$

To apply these filters in the space domain it is necessary only to interpret k^2 as the tridiagonal matrix $\mathbf{T}$ with $(-1, 2, -1)$ on the main diagonal. Specifically, for the low-pass filter it is necessary to solve a tridiagonal set of simultaneous equations like

$$(\alpha \mathbf{I} + \mathbf{T})\,\mathbf{q} = \alpha\,\mathbf{p} \tag{1}$$

in which $\mathbf{q}$ and $\mathbf{p}$ are column vectors whose elements denote different places on the x-axis. Previously, this was done while solving the heat-flow equation. To make the filter space-variable, the parameter α can be taken to depend on x so that $\alpha \mathbf{I}$ is replaced by an arbitrary diagonal matrix. It doesn't matter whether $\mathbf{p}$ and $\mathbf{q}$ are represented in the ω-domain or the t-domain.

Turn your attention from narrow-band data to data with a somewhat

broader spectrum and consider

Dip Filters for Moderate Bandwidth Data ($\Delta\omega$)	
Low Pass	*High Pass*
$Q = \dfrac{\alpha}{\alpha + \dfrac{k^2}{-i\omega}} \; P$	$Q = \dfrac{\dfrac{k^2}{-i\omega}}{\alpha + \dfrac{k^2}{-i\omega}} \; P$

Naturally these filters can be applied to data of any bandwidth. However the filters are appropriately termed "dip filters" only over a modest bandwidth.

To understand these filters look in the (ω, k)-plane at contours of constant k^2/ω, i.e. $\omega \approx k^2$. Such contours, examples of which are shown in figure 1, are curves of constant attenuation and constant phase shift. The low-pass filter has no phase shift in the pass zone, but there is time differentiation in the attenuation zone. This is apparent from the defining equation. The high-pass filter has no phase shift in the flat pass zone, but there is time integration in the attenuating zone.

An interesting feature of these dip filters is that the low-pass and the high-pass filters constitute a pair of filters which sum to unity. So nothing is lost if a dataset is partitioned by them in two. The high-passed part could be added to the low-passed part to recover the original dataset. Alternately, once the low-pass output is computed, it is much easier to compute the high-pass output, because it is just the input minus the low-pass.

Recursive-Dip-Filter Implementation

Implementation of the moderate bandwidth dip filters is, again, a straightforward matter. For example, clearing fractions, the low-pass filter becomes

$$(-i\,\omega\,\alpha\,\mathbf{I} + \mathbf{T})\,\mathbf{Q} \;=\; -i\,\omega\,\alpha\,\mathbf{P} \tag{2}$$

The main trick is to realize that the differentiation implied by $-i\,\omega$ is performed in a Crank-Nicolson sense. That is, terms not differentiated are averaged over adjacent values.

$$\mathbf{I}\,\frac{\alpha}{\Delta t}\,(\,\mathbf{q}_{t+1} - \mathbf{q}_t\,) \;+\; \mathbf{T}\,\frac{\mathbf{q}_{t+1} + \mathbf{q}_t}{2} \;=\; \frac{\alpha}{\Delta t}\,(\mathbf{p}_{t+1} - \mathbf{p}_t) \tag{3}$$

Gathering the unknowns to the left gives

$$\left(\frac{\alpha}{\Delta t}\,\mathbf{I} + \frac{1}{2}\,\mathbf{T}\right)\mathbf{q}_{t+1} \;=\; \left(\frac{\alpha}{\Delta t}\,\mathbf{I} - \frac{1}{2}\,\mathbf{T}\right)\mathbf{q}_t \;+\; \frac{\alpha}{\Delta t}\,(\mathbf{p}_{t+1} - \mathbf{p}_t) \tag{4}$$

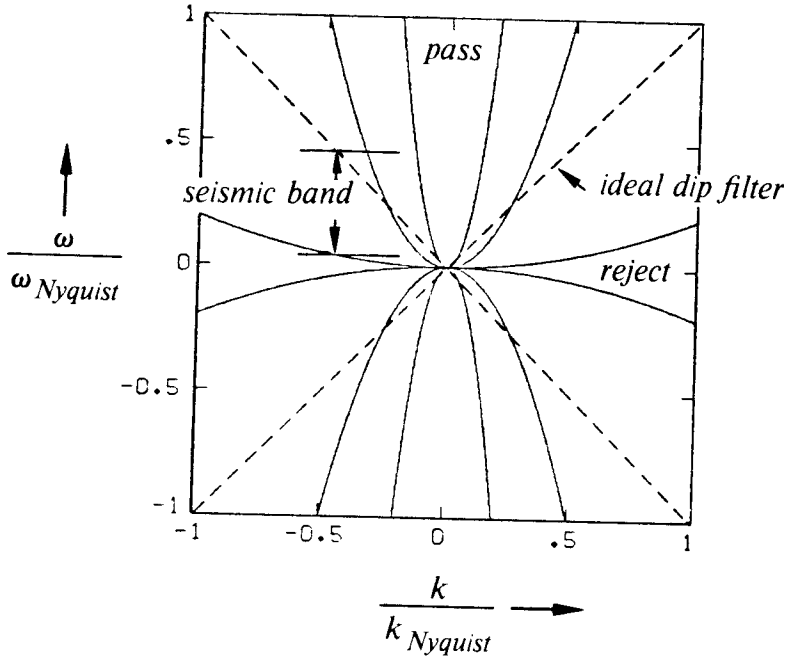


FIG. 2.5-1. Constant-attenuation contours of dip filters. Over the seismic frequency band these parabolas may be satisfactory approximations to the dashed straight line. Pass/reject zones are indicated for the low-pass filter. (Hale)

Equation (4) is a tridiagonal system of simultaneous equations for the unknowns $\mathbf{q}_{t+1}$. The system may be solved recursively for successive values of t.

The parameter α determines the filter cutoff. It can be chosen to be any function of time and space. However, if the function is to vary extremely rapidly, then it may be necessary to incorporate some of the stability analysis that is developed in a later chapter for use with wave equations.

Side Boundaries

Usually geophysicists wish that there were no boundaries on the sides, or that they were infinitely far away. There are two kinds of side conditions to think about, those in x, and those in k.

Often the side conditions on x are best approximated by zero-slope side conditions. It is possible to use more general side conditions because we have previously learned to solve any tridiagonal system of equations.

The side conditions in k-space relate to the steepest dips. A way to handle these dips is to use $\mathbf{T}/(\mathbf{I}-\beta\mathbf{T})$ to represent k^2. This introduces another adjustable parameter β, which must be kept less than 1/4. Details are

studied in Section 4.3.

Slicing Pies

Naturally we may prefer true dip filters, that is, functions of k/ω instead of the functions of k^2/ω described above. But it can be shown that replacing k^2/ω in the above expressions by k^2/ω^2 gives recursions that are unstable.

Sharper pie slices (filters which are more strictly a rectangle function of k/ω), may be defined through a variety of approximation methods described by Hale and Claerbout [1983]. Generally, $|k|$ can be expanded in a power series in $\partial^2/\partial x^2$. If the approximation to $|k|$ is ensured positive, you can expect stability of the recursion that represents $|k|/i\omega$.

More simply, you might be willing to Fourier transform time or space, but not both. In the remaining dimension (the one not transformed) the required operation is a highpass or lowpass filter. This is readily implemented by a variety of techniques, such as the Butterworth filter.

Higher Dimensionality

It is natural to think of a recursive three-dimensional low-pass dip filter as the functional form

$$\frac{\alpha}{\alpha + \dfrac{k_x^2 + k_y^2}{-i\omega}} \tag{5}$$

This, however, leads to an infeasible Crank-Nicolson situation. Multidimensional low-pass filtering *is* possible with

$$\left(\frac{\alpha_x}{\alpha_x + \dfrac{k_x^2}{-i\omega}} \right) \left(\frac{\alpha_y}{\alpha_y + \dfrac{k_y^2}{-i\omega}} \right) \tag{6}$$

2.6 Retarded Coordinates

To examine running horses it may be best to jump on a horse. Likewise, to examine moving waves, it may be better to move along with them. So to describe waves moving downward into the earth we might abandon (x, z)-coordinates in favor of moving (x, z')-coordinates, where $z' = z + t\ v$.

An alternative to the moving coordinate system is to define *retarded coordinates* (x, z, t') where $t' = t - z/v$. The classical example of retarded coordinates is solar time. Time seems to stand still on an airplane that moves westward at the speed of the sun.

The migration process resembles the simulation of wave propagation in either a moving coordinate frame or a retarded coordinate frame. Retarded coordinates are much more popular than moving coordinates. Here is the reason: In solid-earth geophysics, velocity may depend on both x and z, but the earth doesn't change with time t during our seismic observations. In a moving coordinate system the velocity could depend on all three variables, thus unnecessarily increasing the complexity of the calculations. Fourier transformation is a popular means of solving the wave equation, but it loses most of its utility when the coefficients are nonconstant.

Definition of Independent Variables

The specific definition of retarded coordinates is a matter of convenience. Often the retardation is based on hypothetical rays moving straight down with velocity $\overline{v}(z)$. The definition of these coordinates has utility even in problems in which the earth velocity varies laterally, say $v(x, z)$, even though there may be no rays going exactly straight down. In principle, any coordinate system may be used to describe any circumstance, but the utility of the retarded coordinate system generally declines as the family of rays defining it departs more and more from the actual rays.

Despite the simple case at hand it is worthwhile to be somewhat formal and precise. Define the retarded coordinate system (t', x', z') in terms of

ordinary Cartesian coordinates (t, x, z) by the set of equations

$$t' = t'(t, x, z) = t - \int_0^z \frac{dz}{\overline{v}(z)} \tag{1a}$$

$$x' = x'(t, x, z) = x \tag{1b}$$

$$z' = z'(t, x, z) = z \tag{1c}$$

The purpose of the integral is to accumulate the travel time from the surface to depth z. The reasons to define (x', z') when it is just set equal to (x, z) are, first, to avoid confusion during partial differentiation and, second, to prepare for later work in which the family of rays is more general.

Definition of Dependent Variables

There are two kinds of *dependent* variables, those that characterize the medium and those that characterize the waves. The medium is characterized by its velocity v and its reflectivity c. The waves are characterized by using U for an upcoming wave, D for a downgoing wave, P for the pressure, and Q for a modulated form of pressure. Let us say $P(t, x, z)$ is the mathematical function to find pressure, given (t, x, z); and $P'(t', x', z')$ is the mathematical function given (t', x', z'). The statement that the two mathematical functions P and P' both refer to the same physical variable is this:

$$P(t, x, z) = P'[t'(t, x, z),\ x'(t, x, z),\ z'(t, x, z)] \tag{2}$$
$$P(t, x, z) = P'(t', x', z')$$

Obviously there are analogous expressions for the other dependent variables and medium parameters like velocity $v(x, z)$.

The Chain Rule and the High Frequency Limit

The familiar partial-differential equations of physics come to us in (t, x, z)-space. The chain rule for partial differentiation will convert the partial derivatives to (t', x', z')-space. For example, differentiating (2) with respect to z gives

$$\frac{\partial P}{\partial z} = \frac{\partial P'}{\partial t'}\frac{\partial t'}{\partial z} + \frac{\partial P'}{\partial x'}\frac{\partial x'}{\partial z} + \frac{\partial P'}{\partial z'}\frac{\partial z'}{\partial z} \tag{3a}$$

Using (1) to evaluate the coordinate derivatives gives

$$\frac{\partial P}{\partial z} = -\frac{1}{\overline{v}}\frac{\partial P'}{\partial t'} + \frac{\partial P'}{\partial z'} \tag{3b}$$

There is nothing special about the variable P in (3). We could as well write

$$\frac{\partial}{\partial z} = -\frac{1}{\overline{v}}\frac{\partial}{\partial t'} + \frac{\partial}{\partial z'} \tag{4}$$

where the left side is for operation on functions that depend on (t, x, z) and the right side is for functions of (t', x', z'). Differentiating twice gives

$$\frac{\partial^2}{\partial z^2} = \left(\frac{-1}{\overline{v}}\frac{\partial}{\partial t'} + \frac{\partial}{\partial z'}\right)\left(\frac{-1}{\overline{v}}\frac{\partial}{\partial t'} + \frac{\partial}{\partial z'}\right) \tag{5}$$

Using the fact that the velocity is always time-independent results in

$$\frac{\partial^2}{\partial z^2} = \frac{1}{\overline{v}^2}\frac{\partial^2}{\partial t'^2} - \frac{2}{\overline{v}}\frac{\partial^2}{\partial t'\partial z'} + \frac{\partial^2}{\partial z'^2} + \left[\frac{1}{\overline{v}^2}\frac{d\overline{v}}{dz'}\right]\frac{\partial}{\partial t'} \tag{6}$$

Except for the rightmost term with the square brackets it could be said that "squaring" the operator (4) gives the second derivative. This last term is almost always neglected in data processing. The reason is that its effect is similar to the effect of other first-derivative terms with material gradients for coefficients. Such terms, as described in Section 1.5, cause amplitudes to be more carefully computed. If the last term in (6) is to be included, then it would seem that all such terms should be included, from the beginning.

Fourier Transforms in Retarded Coordinates

Given a pressure field $P(t, x, z)$, we may Fourier transform it with respect to any or all of its independent variables (t, x, z). Likewise, if the pressure field is specified in retarded coordinates, we may Fourier transform with respect to (t', x', z'). Since the Fourier dual of (t, x, z) is (ω, k_x, k_z), it seems appropriate for the dual of (t', x', z') to be (ω', k_x', k_z'). Now the question is, how are (ω', k_x', k_z') related to the familiar (ω, k_x, k_z)? The answer is contained in the chain rule for partial differentiation. Any expression like

$$\frac{\partial}{\partial z} = -\frac{1}{\overline{v}}\frac{\partial}{\partial t'} + \frac{\partial}{\partial z'} \tag{4}$$

on Fourier transformation says

$$i\,k_z = -\frac{-i\,\omega'}{\overline{v}} + i\,k_z' \tag{7}$$

Computing all the other derivatives, we have the transformation

$$\omega = \omega' \tag{8a}$$

$$k_x = k_x' \tag{8b}$$

$$k_z = k_z' + \frac{\omega'}{\overline{v}} \tag{8c}$$

Recall the dispersion relation for the scalar wave equation:

$$\frac{\omega^2}{v^2} = k_x^2 + k_z^2 \tag{9}$$

Performing the substitutions from (8) into (9) we have the expression of the scalar wave equation in retarded time, namely,

$$\left(\frac{\omega'}{v} \right)^2 = k_x'^2 + \left(k_z' + \frac{\omega'}{\overline{v}} \right)^2 \tag{10}$$

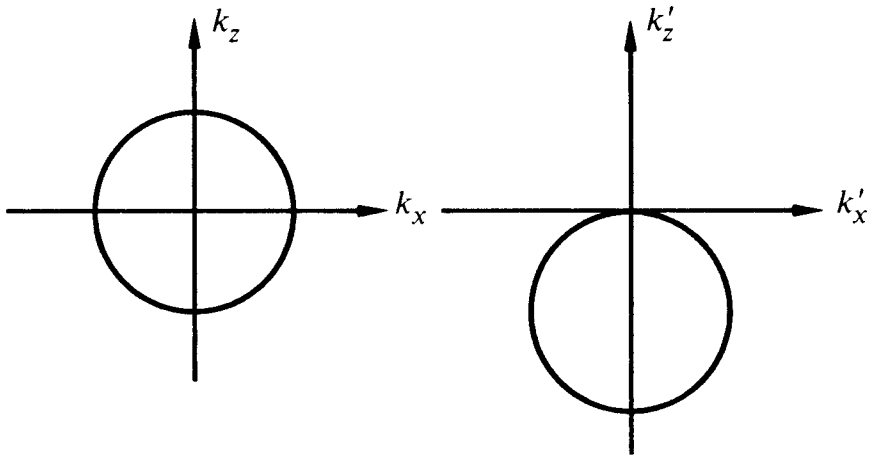


FIG. 2.6-1. Dispersion relation of the wave equation in usual coordinates (left) and retarded time coordinates (right).

These two dispersion relations are plotted in figure 1 for the retardation velocity chosen equal to the medium velocity.

Figure 1 graphically illustrates that retardation can reduce the cost of finite-difference calculations. Waves going straight down are near the top of the dispersion curve (circle). The effect of retardation is to shift the circle's top down to the origin. Discretizing the x- and z-axes will cause spatial frequency aliasing on them. The larger the frequency ω, the larger the circle. Clearly the top of the shifted circle is further from folding. Alternately, Δz may be increased (for the sake of economy) before k_z' exceeds the Nyquist frequency $\pi/\Delta z$.

Interpretation of the Modulated Pressure Variable Q

Earlier a variable Q was defined from the pressure P by the equation

$$P(\omega) \quad = \quad Q(\omega) \exp\left[i\omega \int_0^z \frac{dz}{\overline{v}(z)} \right] \tag{11}$$

The right side is a product of two functions of ω. At constant velocity (11) is expressed as

$$P(\omega) \quad = \quad Q(\omega)\, e^{i\omega z/v} \quad = \quad Q(\omega)\, e^{i\omega t_0} \tag{12}$$

In the time domain $e^{i\omega t_0}$ becomes a delta function $\delta(t - t_0)$. Equation (12) is a product in the frequency domain, so in the time domain it is the convolution

$$\begin{aligned} p(t) \quad &= \quad q(t) * \delta(t - z/v) \\ &= \quad q(t - z/v) \\ &= \quad q(t') \end{aligned} \tag{13}$$

This confirms that the definition of a dependent variable Q is equivalent to introducing retarded time t'.

Einstein's Special Relativity Theory

There is no known application of Einstein's theory of special relativity to seismic imaging. But some of the mathematical methods are related, and now is the appropriate time to take a peek at this famous theory.

In 1887 the Michelson-Morley interferometer experiment established with high accuracy that light travels in all directions at the same speed, day and night, winter and summer. We have seen that the dispersion relation of the scalar wave equation is a circle centered at the origin, meaning that waves go the same speed in all directions. But if the coordinate system is moving with respect to the medium, then the dispersion relation loses directional symmetry. For light propagating in the vacuum of outer space, there seems to be no natural reference coordinate system. If the earth is presumed to be at rest in the summer, then by winter, the earth is moving around the sun in the opposite direction. The summer coordinates relate to the winter coordinates by something like $x' = x - 2\, v_{earth}\, t$. While analysis of the Michelson-Morley experiment shows that such motion should have a measurable asymmetry, measurements show that the predicted asymmetry is absent. Why? One theory is the "ether" theory. Ether is a presumed substance that explains the paradox of the Michelson-Morley experiment. It is presumed to be of minuscule density and viscosity, allowing us to imagine that it is

somehow dragged around the earth in such a way that earthbound experimenters are always moving at the same speed as it is. Other measurements, however, also contradict the presumption of ether. Just as wind refracts atmospheric sound waves, ether should cause a measurable refraction of starlight, but this is not observed.

Einstein's explanation of the experiments is based on a mathematical fact that you can easily verify. Let a coordinate frame be defined by

$$z' = z - v \frac{t}{\sqrt{1 - v^2/c^2}} \tag{14a}$$

$$x' = x \tag{14b}$$

$$t' = \frac{t - \frac{v}{c^2} z}{\sqrt{1 - v^2/c^2}} \tag{14c}$$

The amazing thing about this transformation, which you can easily prove, is that it converts the equation $P_{xx} + P_{zz} = c^{-2} P_{tt}$ to the equation $P_{x'x'} + P_{z'z'} = c^{-2} P_{t't'}$. The transformed wave equation is independent of velocity v which is what led Einstein to his surprising conclusions.

2.7 Finite Differencing in (t, x, z)-Space

Much, if not most, production migration work is done in (t, x, z)-space. To avoid being overwhelmed by the complexity of this three-dimensional space, we will first look at migration in (z, t)-space for fixed k_x.

Migration in (z, t)-Space

Migration and data synthesis may be envisioned in (z', t')-space on the following table, which contains the upcoming wave U:

						z'
	c_0					
	u_1	c_1				
	u_2		c_2			
	u_3			c_3		
	u_4				c_4	
	0	0	0	0	0	
t'						

(1)

In this table the observed upcoming wave at the earth's surface $z' = 0$ is denoted by u_t. The migrated section, denoted by c_t, is depicted along the diagonal because the imaging condition of exploding reflectors at time $t = 0$ is represented in retarded space as

$$z' = z \tag{2a}$$

$$t' = t + z/v \qquad (+ \text{ for } up) \tag{2b}$$

$$0 = t = t' - z'/v \tag{3}$$

The best-focused migration need not fall on the 45° line as depicted in (1); it might be on any line or curve as determined by the earth velocity. This curve forms the basis for velocity determination (Section 3.5). You couldn't determine velocity this way in the frequency-domain.

From Section 2.1, the equation for upcoming waves U in retarded coordinates (t', x', z') is

$$\frac{\partial^2 U}{\partial z' \, \partial t'} = -\frac{v}{2} \frac{\partial^2 U}{\partial x'^2} \tag{4}$$

Next, Fourier transform the x-axis. This assumes that v is a constant function of x and that the x-dependence of U is the sinusoidal function $\exp(ik_x x)$. Thus,

$$0 = \left(\frac{v}{2} k_x^2 - \frac{\partial^2}{\partial z' \, \partial t'} \right) U \tag{5}$$

Now this partial-differential equation will be discretized with respect to t' and z'. Matrix notation will be used, but the notation does not refer to matrix algebra. Instead the matrices refer to differencing stars that may be placed on the (t', z')-plane of (1). Let $*$ denote convolution in (z, t)-space. A succession of derivatives is really a convolution, so the concept of

$\partial/\partial z \; \partial/\partial t = \partial^2/\partial z \, \partial t$ is expressed by is expressed by

$$[\,-1 \quad +1\,] \; * \; \begin{bmatrix} -1 \\ 1 \end{bmatrix} \; = \; \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{6}$$

Thus, the differenced form of (5) is

$$0 \; = \; \left\{ \frac{v}{2} \, \frac{\Delta z' \, \Delta t'}{4} \; k_x{}^2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right\} * U \tag{7}$$

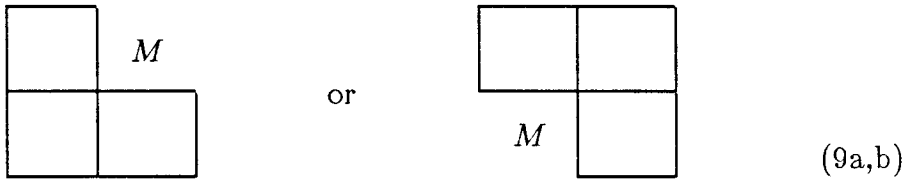
The $1/4$ enters in because the average of U is taken over four places on the mesh.

The sum of the two operators always has $|\,b\,| \geq |\,s\,|$ in the form

$$0 \; = \; \begin{bmatrix} s & b \\ b & s \end{bmatrix} \; * \; U \tag{8}$$

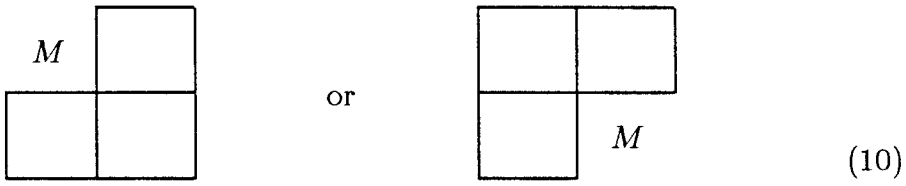
Now the differencing star in (8) will be used to fill table (1) with values for U.

Given the three values of U in the boxes, a missing one, M, may be determined by either of the implied two operations



(9a,b)

It turns out that because $|\,b\,| \geq |\,s\,|$, the implied filling operations by



(10)

are unstable. It is obvious that there would be a zero-divide problem if s were equal to 0, and it is not difficult to do the stability analysis that shows that (10) causes exponential growth of small disturbances.

It is a worthwhile exercise to make the zero-dip assumption ($k_x = 0$) and use the numerical values in the operator of (8) to fill in the elements of table (1). It will be found that the values of u_t move laterally in z across the table with no change, predicting, as the table should, that $c_t = u_t$. Slow change in z suggests that we have oversampled the z-axis. In practice, effort is saved by sampling the z-axis with fewer points than are used to sample the t-axis.

(t, x, z)-Space, 15° Diffraction Program

The easiest way to understand 15° migration in (t, x, z)-space is to refer to the (z, t)-space migration. Instead of a scalar function $U(k_x)$, we use $\mathbf{u}$, a vector whose components u_j measure pressure at $x = j\,\Delta x$. Think of k_x^2 as a tridiagonal matrix, call it $\mathbf{T}$, with $(-1, 2, -1)$ on the main diagonal. Note that k_x^2 is positive, and that $\mathbf{T}$ is a positive definite matrix with a positive element on its main diagonal. Take equation (7) and use α to denote the left constant. This gives

$$0 = \left\{ \alpha\,\mathbf{T} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix} \right\} * \mathbf{u} = \begin{bmatrix} \alpha\mathbf{T}-\mathbf{I} & \alpha\mathbf{T}+\mathbf{I} \\ \alpha\mathbf{T}+\mathbf{I} & \alpha\mathbf{T}-\mathbf{I} \end{bmatrix} * \mathbf{u} \qquad (11)$$

Consider a modeling program. It begins down inside the earth with the differencing star (9b). Solving (11) for the unknown $\mathbf{u}_{t'+1,\,z'}$ and dropping all primes yields

$$(\alpha\mathbf{T}+\mathbf{I})\,\mathbf{u}_{t+1,\,z} = -\left[(\alpha\mathbf{T}+\mathbf{I})\,\mathbf{u}_{t,\,z+1} + (\alpha\mathbf{T}-\mathbf{I})(\mathbf{u}_{t,\,z} + \mathbf{u}_{t+1,\,z+1})\right] \qquad (12)$$

First, evaluate the expression on the right. The left side is a tridiagonal system to be solved for the unknown $\mathbf{u}_{t+1,\,z}$. Allowable sequences in which (12) may be applied are dictated by the differencing star (9b).

Heeding the earlier remark that with waves of modest dip, the z'-axis need not be sampled so densely as the t'-axis, we do a computation that skips alternate levels of z'. The specific order chosen in the computer program in figure 1 is indicated by the numbers in the following table:

t' \ z'				
	c_0	c_0		
	5	c_1		
	6	c_2	c_2	
	7	2	c_3	
	8	3	c_4	c_4
	9	4	1	c_5

(13)

An inescapable practical problem shown in the table when the number of points in t'-space is not exactly equal that in z'-space is that the earth image must be *interpolated* along a diagonal on the mesh. The crude interpolation in (13) illustrates the assumption that the wave field changes rapidly in t' but slowly in z', i.e. the small-angle assumption.

```
# Time Domain 15-degree Diffraction Movie
# Star:          w=p(t  ,z)      y=p(t  ,z+1)
# Star:          u=p(t+1,z)      v=p(t+1,z+1)
real p(36,96),u(36),w(36),v(36),y(36),e(36),f(36),d(36),z(96),alfa,beta
integer ix,nx,iz,nz,it,nt,kbyte
nx = 36;  nz = 96;  nt = 96;     kbyte=1
alfa = .125                  # v*dz*dt/(8*dx*dx)
beta = .140                  # accurate x derivative parameter; simplest case b=0.
open(3,file='plot40',status='new',access='direct',form='unformatted',recl=1)
do iz=1,nz;  do ix=1,nx;  p(ix,iz) = 0.                  # clear space
do iz=nz/5,nz,nz/4                              # Set up initial model
        do it=1,15                              # of 4 band limited
                do ix=1,4                       # "point" scatterers.
                        p(ix,it+iz) = (5.-ix)*(8-it)*exp(-.1*(it-8)**2)
apb = alfa+beta; amb = alfa-beta                # tridiagonal coefficients
diag = 1.+2.*amb;  offdi = -amb
do iz=nz,2,-2 {                                         # Climb up in steps of 2 z-levels
        do i=1,nz;  z(i)=0.;  z(iz)=1.          # Pointer to current z-level
        write(3,rec=kbyte) (z(i),i=1,nz),((p(ix,i),i=1,nz),ix=1,nx)
        kbyte = kbyte + nx*nz*4 + nz*4
        do ix=1,nx
                { u(ix) = p(ix,iz-1);           v(ix) = u(ix) }
        do it=iz,nt {
                do ix=1,nx      #update the differencing star
                        { w(ix) = u(ix);  y(ix) = v(ix);  v(ix) = p(ix,it) }
                dd = (1.-apb)*(v(1)+w(1))+apb*(v(2)+w(2))
                d(1) = dd-diag*y(1)-offdi*(y(1)+y(2))
                do ix=2,nx-1 {
                        dd   = (1.-2.*apb)*(v(ix)+w(ix))
                        dd   = dd + apb*(v(ix-1)+w(ix-1)+v(ix+1)+w(ix+1))
                        d(ix) = dd-diag*y(ix)-offdi*(y(ix-1)+y(ix+1))  }
                dd = (1.-apb)*(v(nx)+w(nx))+apb*(v(nx-1)+w(nx-1))
                d(nx) = dd-diag*y(nx)-offdi*(y(nx)+y(nx-1))
                call rtris(nx,diag+offdi,offdi,diag,offdi,diag+offdi,d,u,e,f)
                do ix=1,nx
                        p(ix,it) = u(ix)
                }
        }
do i=1,nz;  z(i)=0.;  z(1)=1.
write(3,rec=kbyte) (z(i),i=1,nz),((p(ix,i),i=1,nz),ix=1,nx)
stop;   end
```

FIG. 2.7-1. Time-domain diffraction movie program. (Clayton, Gonzalez, JFC, Hale)

Figure 2 shows the last frame in the movie produced by the test program. Exercise 1 suggests minor changes to the program of figure 1 to convert it from diffraction to migration. As modified, the program is essentially the original wave equation migration program introduced by Johnson and Claerbout [1971] and Doherty and Claerbout [1972].

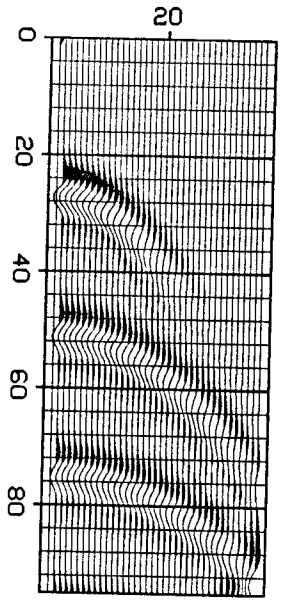


FIG. 2.7-2. Diffractions in the last frame of the downward-continuation movie.

You Can't Time Shift in the Time Domain.

You might wish to do migration in (x, z, t)-space with lateral velocity variation. Then the thin-lens stage would be implemented by time shifting instead of by multiplying by $\exp\{i\omega[v(x, z)^{-1} - \bar{v}(z)^{-1}]\Delta z\}$. Time shifting is a delightfully easy operation when what is needed is to shift data by an integral number of sample units. Repetitive time shifting by a fractional number of digital units, however, is a nightmare. Multipoint interpolation operators are required. Even then, pulses tend to disperse. So the lens term is probably best left in the frequency domain.

(t, x, z)-Space, 45° Equation

The 45° migration is a little harder than the 15° migration because the operator in the time domain is higher order, but the methods are similar to those of the 15° equation and the recursive dip filter. The straightforward approach is just to write down the differencing stars. When I did this kind of work I found it easiest to use the Z-transform approach where $1/(-i\omega\Delta t)$ is represented by the bilinear transform $1/2(1+Z)/(1-Z)$. There are various ways to keep the algebra bearable. One way is to bring all powers of Z to the numerator and then collect powers of Z. Another way, called the integrated approach, is to keep $1/(1-Z)$ with some of the terms. Terms including $1/(1-Z)$ are represented in the computer by buffers that contain

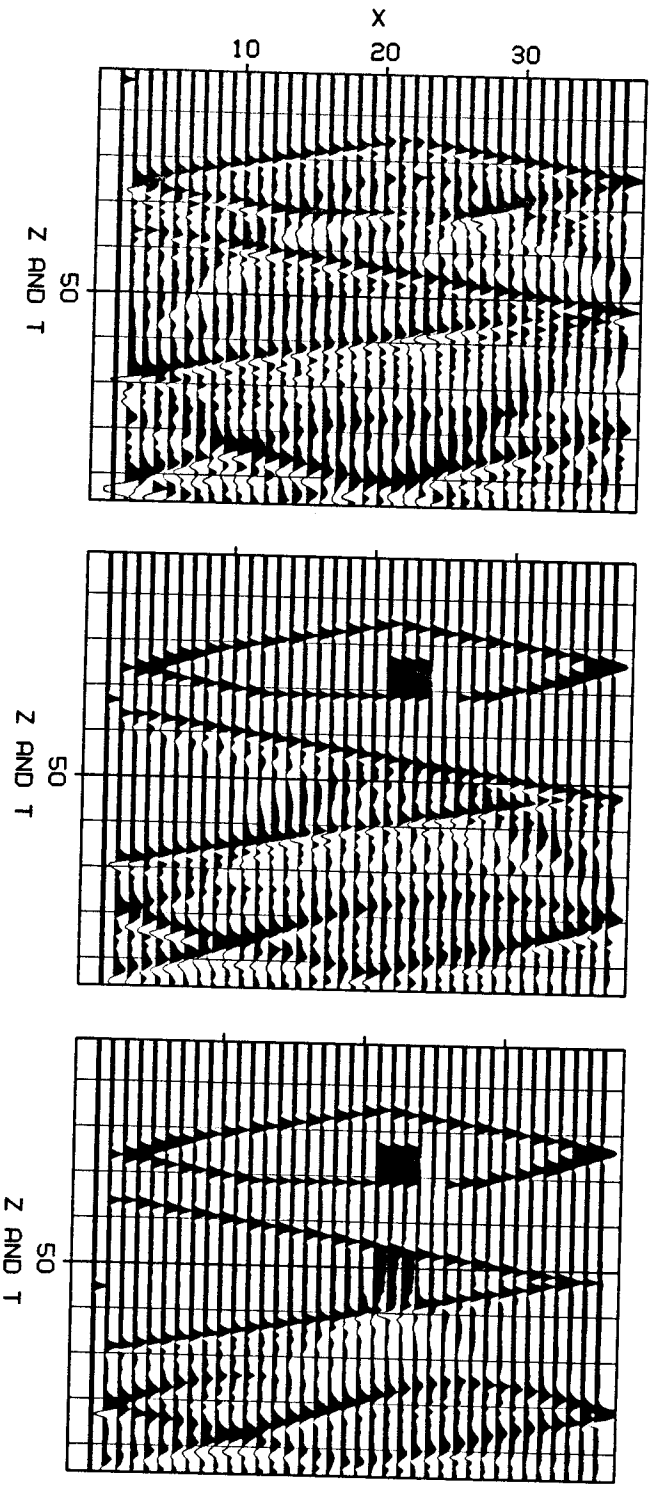


FIG. 2.7-3. The program of figure 1 was used to create synthetic data from a more complicated model. The three frames depict three depths. The leftmost channel of each frame depicts the depth of the frame. (Hale)

the sum from infinite time to time t. The Z-transform approach is developed in Section 4.6. Its real advantage is that it systematizes the stability analysis.

EXERCISES

1. Alter the program given in figure 1 so that it does migration. The delta-function inputs should turn into approximate semicircles.
2. Perform major surgery on the program in figure 1 so that it becomes a low-pass dip filter.
3. Consider a 45° migration program in the space of (z, t, k_x). Find the coefficients in a 6-point differencing star, three points in time and two points in depth. For simplicity, take $v=1$, $\Delta t=1$, and $\Delta z=1$. Suppose this analysis were transformed into the x-domain ($\Delta x=1$) by replacing k_x^2 with $\mathbf{T}$. What set of tridiagonal equations would have to be solved?

2.8 Introduction to Stability

Experience shows that as soon as you undertake an application that departs significantly from textbook situations, stability becomes a greater concern than accuracy. Stability, or its absence, determines whether the goal is achievable at all, whereas accuracy merely determines the price of achieving it. Here we will look at the stability of the heat-flow equation with real and with imaginary heat conductivity. Since the latter case corresponds to seismic migration, these two cases provide a useful background for stability analysis.

Most stability analysis is based on Fourier transformation. More simply, single sinusoidal or complex exponential trial solutions are examined. If a method becomes unstable for any frequency, then it will be unstable for any realistic case, because realistic functions are just combinations of all frequencies. Begin with the sinusoidal function

$$P(x) = P_0 e^{i k x} \tag{1}$$

The second derivative is

$$\frac{\partial^2 P}{\partial x^2} = -k^2 P \tag{2}$$

An expression analogous to the second difference operator defines $\hat{k}$:

$$\frac{\delta^2 P}{\delta x^2} = \frac{P(x+\Delta x) - 2P(x) + P(x-\Delta x)}{\Delta x^2} \tag{3a}$$

$$= -\hat{k}^2 P \tag{3b}$$

Ideally $\hat{k}$ should equal k. Inserting the complex exponential (1) into (3a) gives an expression for $\hat{k}$:

$$-\hat{k}^2 P = \frac{P_0}{\Delta x^2}\left[e^{i k (x+\Delta x)} - 2 e^{i k x} + e^{i k (x-\Delta x)}\right] \tag{4a}$$

$$(\hat{k}\,\Delta x)^2 = 2[1 - \cos(k\,\Delta x)] \tag{4b}$$

It is a straightforward matter to make plots of (4b) or its square root. The square root of (4b), through the half-angle trig identity, is

$$\hat{k}\,\Delta x = 2 \sin\frac{k\,\Delta x}{2} \tag{4c}$$

Series expansion shows that $\hat{k}$ matches k well at low spatial frequencies. At the Nyquist frequency, defined by $k\,\Delta x = \pi$, the value of $\hat{k}\,\Delta x = 2$ is a poor approximation to π. As with any Fourier transform on the discrete domain, $\hat{k}$ is a periodic function of k above the Nyquist frequency. Although k ranges from minus infinity to plus infinity, $\hat{k}^2$ is compressed into the range zero to four. The limits to the range are important since instability often starts at one end of the range.

Explicit Heat-Flow Equation

Begin with the heat-flow equation and Fourier transform over space. Thus $\partial^2/\partial x^2$ becomes simply $-k^2$, and

$$\frac{\partial q}{\partial t} = -\frac{\sigma}{c} k^2 q \tag{5}$$

Finite differencing explicitly over time gives an equation that is identical in form to the inflation-of-money equation:

$$\frac{q_{t+1} - q_t}{\Delta t} = -\frac{\sigma}{c} k^2 q_t \tag{6a}$$

$$q_{t+1} = \left(1 - \frac{\sigma\,\Delta t}{c} k^2\right) q_t \tag{6b}$$

For stability, the magnitude of q_{t+1} should be less than or equal to the magnitude of q_t. This requires the factor in parentheses to have a magnitude less than or equal to unity. The dangerous case is when the factor is more negative than -1. There is instability when $k^2 > 2c/(\sigma \Delta t)$. This means that the high frequencies are diverging with time. The explicit finite differencing on the time axis has caused disaster for short wavelengths on the space axis. Surprisingly, this disaster can be recouped by differencing the space axis coarsely enough! The second space derivative in the Fourier transform domain is $-k^2$. When the x-axis is discretized it becomes $-\hat{k}^2$. So, to discretize (5) and (6), just replace k by $\hat{k}$. Equation (4c) shows that $\hat{k}^2$ has an upper limit of $\hat{k}^2 = 4/\Delta x^2$ at the Nyquist frequency $k \Delta x = \pi$. Finally, the factor in (6b) will be less than unity and there will be stability if

$$\hat{k}^2 \quad = \quad \frac{4}{\Delta x^2} \quad \leq \quad \frac{2c}{\sigma \Delta t} \tag{7}$$

Evidently instability can be averted by a sufficiently dense sampling of time compared to space. Such a solution becomes unbearably costly, however, when the heat conductivity $\sigma(x)$ takes on a wide range of values. For problems in one space dimension, there is an easy escape in implicit methods. For problems in higher-dimensional spaces, explicit methods must be used.

Explicit 15° Migration Equation

We saw in Section 2.1 that the retarded 15° wave-extrapolation equation is like the heat-flow equation with the exception that the heat conductivity σ must be replaced by the purely imaginary number i. The amplification factor (the magnitude of the factor in parentheses in equation (6b)) is now the square root of the sum squared of real and imaginary parts. Since the real part is already one, the amplification factor exceeds unity for all nonzero values of k^2. The resulting instability is manifested by the growth of dipping plane waves. The more dip, the faster the growth. Furthermore, discretizing the x-axis does not solve the problem.

Implicit Equations

Recall that the inflation-of-money equation

$$q_{t+1} - q_t \quad = \quad r \; q_t \tag{8}$$

is a simple explicit finite differencing of the differential equation $dq/dt \approx q$. And recall that a better approximation to the differential equation is given by the Crank-Nicolson form

$$q_{t+1} - q_t \quad = \quad r \; \frac{q_{t+1} + q_t}{2} \tag{9a}$$

that may be rearranged to

$$\left(1 - \frac{r}{2}\right) q_{t+1} \quad = \quad \left(1 + \frac{r}{2}\right) q_t \tag{9b}$$

or

$$\frac{q_{t+1}}{q_t} \quad = \quad \frac{1 + r/2}{1 - r/2} \tag{9c}$$

The amplification factor (9c) has magnitude less than unity for all negative r values, even r equal to minus infinity. Recall that the heat-flow equation corresponds to

$$r \quad = \quad - \frac{\sigma \, \Delta t}{c} \, k^2 \tag{10}$$

where k is the spatial frequency. Since (9c) is good for all negative r, the heat-flow equation, implicitly time-differenced, is good for all spatial frequencies k. The heat-flow equation is stable whether or not the space axis is discretized (then $k \rightarrow \hat{k}$) and regardless of the sizes of Δt and Δx. Furthermore, the 15° wave-extrapolation equation is also unconditionally stable. This follows from letting r in (9c) be purely imaginary: the amplification factor (9c) then takes the form of some complex number $1+r/2$ divided by its complex conjugate. Expressing the complex number in polar form, it becomes clear that such a number has a magnitude exactly equal to unity. Again there is unconditional stability.

At this point it seems right to add a historical footnote. When finite-difference migration was first introduced many objections were raised on the basis that the theoretical assumptions were unfamiliar. Despite these objections finite-difference migration quickly became popular. I think the reason for its popularity was that, compared to other methods of the time, it was a gentle operation on the data. More specifically, since (9c) is of exactly unit magnitude, the output has the same (ω, k)-spectrum as the input. There may be a wider lesson to be learned from this experience: any process acting on data should do as little to the data as possible.

Leapfrog Equations

The leapfrog method of finite differencing, it will be recalled, requires expressing the time derivative over two time steps. This keeps the centers of the differencing operators in the same place. For the heat-flow equation Fourier-transformed over space,

$$\frac{q_{t+1} - q_{t-1}}{2\,\Delta t} = -\frac{\sigma}{c}\,k^2\,q_t \tag{11}$$

It is a bit of a nuisance to analyze this equation because it covers times $t-1$, t, and $t+1$ and requires slightly more difficult analytical techniques. Therefore, it seems worthwhile to state the results first. The result for the heat-flow equation is that the solution always diverges. The result for the wave-extrapolation equation is much more useful: there is stability provided certain mesh-size restrictions are satisfied, namely, Δz must be less than some factor times Δx^2. This result is not exciting in one space dimension (where implicit methods seem ideal), but in higher-dimensional space, such as in the so-called 3-D prospecting surveys, we may be thankful to have the leap-frog method.

The best way to analyze equations like (11) which range over three or more time levels is to use Z-transform filter analysis. Converted to a Z-transform filter problem, the question posed by (11) becomes whether the filter has zeroes inside (or outside) the unit circle. Z-transform stability analysis is described in Section 4.6. Such analysis is necessary for all possible numerical values of k^2. Its result is that there is always trouble if k^2 ranges from zero to infinity. But with the wave-extrapolation equation, instability can be avoided with certain mesh-size restrictions, because $(\hat{k}\ \Delta x)^2$ lies between zero and four.

Tridiagonal Equation Solver

The tridiagonal algorithm is stable for all positive definite matrices. If you have any problems with the tridiagonal solver, you should question the validity of your problem formulation. What is there about your application that seems to demand division by zero?