Chapter 4

Migration by finite differences

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

The situation is analogous to the one encountered in ordinary frequency filtering. Frequency filtering can be done as a product in the frequency domain or a convolution in the time domain. With wave extrapolation there are products in both the temporal frequency 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 where a filter can be made time-variable. Time-variable filters are useful in reflection seismology because the frequency content of echoes changes with time. An annoying aspect of time-variable filters is that they cannot be described by a simple product in the frequency domain. So when an application of time-variable filters comes along, the frequency domain is abandoned, or all kinds of contortions are made (stretching the time axis, for example) to try to make things appear timeinvariant.

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

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

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

4.1 WAVE-EXTRAPOLATION EQUATIONS

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

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

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

4.1.1 Meet the parabolic wave equation

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

The parabolic wave equation was originally introduced to speed finite-difference wave modeling. The following introduction to the parabolic wave equation is via my original transformation method.

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

A vertically downgoing plane wave is represented mathematically by the equation

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

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

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

Inserting (1.3) into the scalar wave equation $P_{xx} + P_{zz} = P_{tt}/v^2$ yields

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

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

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

When I first introduced equation (1.5) for use in seismology, I thought its most important property was this: For a wavefield close to a vertically propagating plane wave, the second x-derivative is small, hence the z-derivative is small. Thus, the finite-difference method should allow a very large Δz and thus be able to treat models more like the earth, and less like a drumhead. I soon realized that the parabolic wave equation is also just what is needed for seismic imaging because you can insert it in an equation like (1.1). (Curiously, equation (1.5) also happens to be the Schroedinger equation of quantum mechanics.)

I called equation (1.5) the 15° equation. After using it for about a year I discovered a way to improve on it by estimating the dropped ∂_{zz} term. Differentiate equation (1.5) with respect to z and substitute the result back into equation (1.4) getting

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

I named equation (1.6) the 45° migration equation. It is first order in ∂_z, so it requires only a single surface boundary condition, however, downward continuation will require something more complicated than equation (1.1).

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

4.1.2 Muir square-root expansion

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

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

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

Solve for k_z selecting the positive square root (thus for the moment selecting *down-going* waves).

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

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

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

Bringing equation (1.9) into the space domain is not simply a matter of substituting a second x derivative for k_x^2. The problem is the meaning of the square root of a differential operator. The square root of a differential operator is not defined in undergraduate calculus courses and there is no straightforward finite difference representation. The square root becomes meaningful only when the square root is regarded as some kind of truncated series expansion. It will be shown in chapter 8 that the Taylor series is a poor choice. Francis Muir showed that my original 15° and

45° methods were just truncations of a continued fraction expansion. To see this, define

$$X \quad = \quad \frac{v k_x}{\omega} \qquad \text{and} \qquad R \quad = \quad \frac{v k_z}{\omega} \tag{4.10}$$

So equation (1.8) is more simply and abstractly written as

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

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

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

The recurrence is a guess that we verify by seeing what it converges to (if it converges). Set $n \; = \; \infty$ in (1.12) and solve

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

The square root of (1.13) gives the required expression (1.11). Geometrically, (1.13) says that the cosine squared of the incident angle equals one minus the sine squared and truncating the expansion leads to angle errors. Muir said, and you can verify, that his recurrance relationship formalizes what I was doing by re-estimating the ∂_{zz} term. Although it is pleasing to think of large values of n, in real life only the low-order terms in the expansion are used. Table 1.1 shows the result of three Muir iterations beginning from $R_0 = 1$

For various historical reasons, the equations in Table 1.1 are often referred to as the 5°, 15°, and 45° equations, respectively, the names giving a reasonable qualitative (but poor quantitative) guide to the range of angles that are adequately handled. A trade-off between complexity and accuracy frequently dictates choice of the 45° equation. It then turns out that a slightly wider range of angles can be accommodated if the recurrence is begun with something like $R_0 = \cos 45°$. Figure 1.1 shows some plots.

4.1.3 Dispersion relations

Performing the substitutions of Table 1.1 into equation (1.10) gives dispersion relationships for comparison to the exact expression (1.8). These are shown in Table 1.2.

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

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

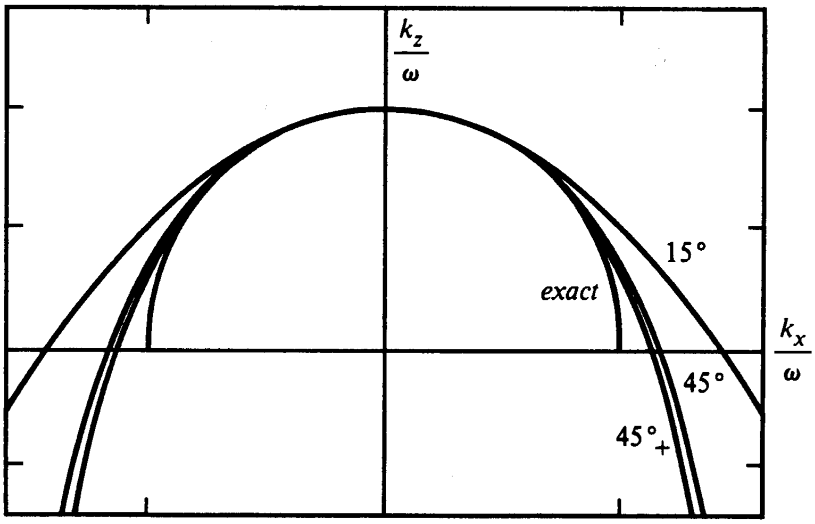


Figure 4.1: Dispersion relation of Table 1.2. The curve labeled 45°_+ was constructed with $R_0 = \cos 45^\circ$. It fits exactly at 0° and 45° . $\boxed{\text{omx-disper}}$ [NR]

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

Table 4.2: As displayed in Figure 1.1, the dispersion relations tend toward a semicircle.

4.1.4 Depth-variable velocity

Identification of ik_z with $\partial/\partial z$ converts the dispersion relations of Table 1.2 into the differential equations of Table 1.3.

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

Migration methods based on equation (1.9) or on Table 1.3 are called *phase-shift methods.*

4.1.5 Retardation (frequency domain)

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

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

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

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

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

variable Q by

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

which is a generalization of equation (1.3) to depth-variable velocity. (Equations (1.15) and (1.17) apply in both x- and k_x-space). Differentiating with respect to z gives

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

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

Next, substitute (1.15) and (1.17) into Table 1.3 to obtain the retarded equations in Table 1.4.

4.1.6 Lateral velocity variation

Having approximated the square root by a polynomial ratio, Table 1.3 or Table 1.4 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.

5°	$\frac{\partial Q}{\partial z}$	$=$	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	$\frac{\partial Q}{\partial z}$	$=$	diffraction	$+$ thin lens

Table 4.4: Retarded form of phase-shift equations.

4.1.7 Two velocities

You may wonder where the two velocities $v(x, z)$ and $\bar{v}(z)$ came from. The first arises in the wave equation, and it must be x-variable if the model is x-variable. The second arises in a mathematical transformation, namely, equation (1.15), so it is purely a matter of definition. Experience shows that complications will abound if we try to make the defined velocity $\bar{v}$ depend on x. But it is nice if the two velocities can be equal so the term containing their difference drops out of the analysis. Thus ordinarily, $\bar{v}(z)$ is chosen to be some kind of horizontal average of $v(x, z)$.

4.1.8 Splitting diffraction and lens terms

The customary numerical solution to the x-domain forms of the equations in Tables 1.3 and 1.4 is arrived at by splitting. That is, you march forward a small Δz-step alternately with the two extrapolators

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

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

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

Table 4.5: Diffraction equations for laterally variable media.

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

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

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

Observe that the diffraction parts of Tables 1.3 and 1.4 are the same. Let us use them and equation (1.19) to define a table of diffraction equations. Substitute $\partial/\partial x$ for ik_x and clear $\partial/\partial x$ from the denominators to get Table 1.5.

4.1.9 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, $v = \bar{v}$ equation from Table 1.5 becomes

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

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

4.1.10 Upcoming waves

All the above equations are for *downgoing* waves. To get equations for *upcoming* waves you need only change the signs of z and $\partial/\partial z$. Letting D denote a *downgoing*

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

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

wavefield and U an *upcoming* wavefield, equation (1.21), for example, is found in Table 1.6.

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

EXERCISES:

1 Consider a tilted straight line tangent to a circle, as shown in figure 1.2 . Use this line to initialize the Muir square-root expansion. State equations and plot them ($-2 \le X \le +2$) for the next two Muir semicircle approximations.

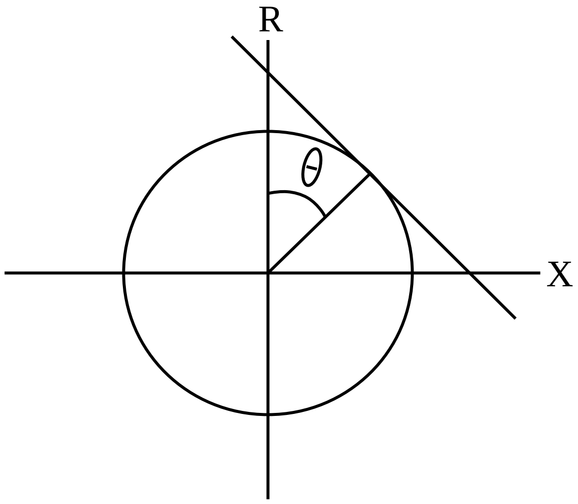


Figure 4.2: omx-tanexer [NR]

4.2 FINITE DIFFERENCING

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

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

4.2.1 The lens equation

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

4.2.2 First derivatives, explicit method

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

$$\begin{aligned} q_{t+1} \ - \ q_t \quad &= \quad .10\ q_t & (4.22) \\ (1.0)\ q_{t+1} \ + \ (-1.1)\, q_t \quad &= \quad 0 & (4.23) \end{aligned}$$

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

$-$ 1.1
$+$ 1.0

Data Table

2.000
2.200
2.420
2.662

time $\downarrow$

Since the data in the data table satisfy the difference equations (1.22) and (1.23), the differencing star may be laid anywhere on top of the data table, the numbers in the star may be multiplied by those in the underlying table, and the resulting cross products will sum to zero. On the other hand, if all but one number (the initial condition) in the data table were missing then the rest of the numbers could be filled in, one at a time, by sliding the star along, taking the difference equations to be true, and solving for the unknown data value at each stage.

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

4.2.3 First derivatives, implicit method

Let us solve the equation

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

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

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

Using this in equation (1.24) yields the the inflation-of-money equations (1.22) and (1.23), where $2\,r = .1$. Thus in the inflation-of-money equation the expression of dq/dt is centered at $t+\Delta t/2$, whereas the expression of q by itself is at time t. There is no reason the q on the right side of equation (1.24) cannot be averaged at time t with time $t + \Delta t$, thus centering the whole equation at $t + \Delta t/2$. When writing difference equations, it is customary to write $q(t + \Delta t)$ more simply as q_{t+1}. (Formally one should say $t = n\Delta t$ and write q_{n+1} instead of q_{t+1}, but helpful mnemonic information is carried by using t as the subscript instead of some integer like n.) Thus, a centered approximation of (1.24) is

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

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

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

which is representable as the difference star

$$\begin{array}{c} \\ \\ \\ t \\ \downarrow \end{array} \begin{array}{|c|} \hline -1-\alpha \\ \hline +1-\alpha \\ \hline \end{array}$$

For a fixed Δt this star gives a more accurate solution to the differential equation (1.24) than does the star for the inflation of money. The reasons for the names "explicit method" and "implicit method" above will become clear only after we study a more complicated equation such as the heat-flow equation.

4.2.4 The explicit heat-flow equation

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

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

Implementing (1.28) in a computer requires some difference approximations for the partial differentials. As before we use a subscript notation that allows (1.25) to be compacted into

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

where $t + \Delta t$ is denoted by $t + 1$. The second-derivative formula may be obtained by doing the first derivative twice. This leads to $q_{t+2} - 2\,q_{t+1} + q_t$. The formula is usually

Data Table $x \rightarrow$

	i	n	i	t	i	a	l	
s								s
i				star				i
d			$-\alpha$	$2\alpha - 1$	$-\alpha$			d
e				1				e

t

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

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:

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

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

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

(Of course it is not justified to use $=$ to denote $\approx$, but the study of errors must be deferred until the concepts have been laid out. Errors are studied in chapter 8.) Letting $\alpha = \sigma\,\Delta t/(C\,\Delta x^2)$, equation (1.31) becomes

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

Equation (1.32) can be *explicitly* solved for q for any x at the particular time $t+1$ given q at all x for the particular time t and hence the name *explicit* method.

Equation (1.32) can be interpreted geometrically as a computational star in the (x,t)-plane, as depicted in Table 1.7. By moving the star around in the data table you will note that it can be positioned so that only one number at a time (the 1) lies over an unknown element in the data table. This enables the computation of subsequent rows beginning from the top. By doing this you are solving the partial-differential equation by the finite-difference method. There are many possible arrangements of initial and side conditions, such as zero-value side conditions. Next is a computer program and its result.

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

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

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

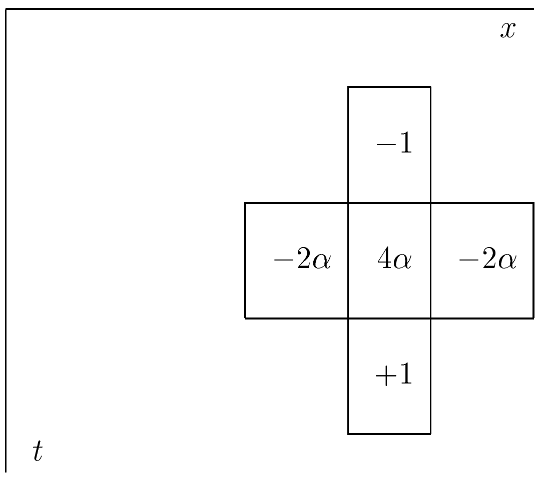
4.2.5 The leapfrog method

The difficulty with the given program is that it doesn't work for all possible numerical values of 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 postponed till page **??**. 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 $t + 1$:

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

The resulting leapfrog differencing star is



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

4.2.6 The Crank-Nicolson method

The Crank-Nicolson method solves both the accuracy and the stability problem. Recall the difference representation of the heat-flow equation (1.32).

$$q_{t+1}^{x} \ - \ q_t^x \quad = \quad a \ \left(q_t^{x+1} - 2q_t^x + q_t^{x-1} \right) \tag{4.34}$$

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

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

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

$$\begin{array}{l|ccc} & & & x \\ \hline & -\alpha & 2\alpha - 1 & -\alpha \\ & -\alpha & 2\alpha + 1 & -\alpha \\ t & & & \end{array} \tag{4.36}$$

When placing this star over the data table, note that, typically, three elements at a time cover unknowns. To say the same thing with equations, move all the $t+1$ terms in (1.35) to the left and the t terms to the right, obtaining

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

Now think of the left side of equation (1.37) as containing all the *unknown* quantities and the right side as containing all *known* quantities. Everything on the right can be combined into a single known quantity, say, d_t^x. Now we can rewrite equation (1.37) as a set of simultaneous equations. For definiteness, take the x-axis to be limited to five points. Then these equations are:

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

Equation (1.37) does not give us each q_{t+1}^x *explicitly*, but equation (1.38) gives them *implicitly* by the solution of simultaneous equations.

The values e_{left} and e_{right} are adjustable and have to do with the side boundary conditions. The important thing to notice is that the matrix is tridiagonal, that is, except for three central diagonals all the elements of the matrix in (1.38) are zero. The solution to such a set of simultaneous equations may be economically obtained. It turns out that the cost is only about twice that of the explicit method given by (1.32). In fact, this implicit method turns out to be cheaper, since the increased accuracy of (1.37) over (1.32) 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 `rtris()` explained in the next section. The results are stable, as you can see.

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

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

4.2.7 Solving tridiagonal simultaneous equations

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

Let the simultaneous equations be written as a difference equation

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

Introduce new unknowns e_j and f_j, along with an equation

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

Write (1.40) with shifted index:

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

Insert (1.41) into (1.39):

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

Now rearrange (1.42) to resemble (15):

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

Compare (1.43) to (1.40) to see recursions for the new unknowns e_j and f_j:

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

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

First a boundary condition for the left-hand side must be given. This may involve one or two points. The most general possible end condition is a linear relation like equation (1.40) at $j = 0$, namely, $q_0 \;=\; e_0 q_1 + f_0$. Thus, the boundary condition must give us both e_0 and f_0. With e_0 and all the a_j, b_j, c_j, we can use (1.44) to compute all the e_j.

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

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

Equation (1.46) includes as special cases the zero-value and zero-slope boundary conditions. Equation (1.46) can be compared to equation (1.41) at its end.

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

Both q_n and q_{n-1} are unknown, but in equations (1.46) and (1.47) we have two equations, so the solution is easy. The final step is to take the value of q_n and use it in (1.41) to compute q_{n-1}, q_{n-2}, q_{n-3}, etc. The subroutine `rtris()` solves equation (1.38) for q where `n=5`, `endl=` e_{left}, `endr=` e_{right}, `a=c=` $-\alpha$, and `b=` $1 - 2\alpha$.

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

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

4.2.8 The xxz derivative

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

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

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

4.2.9 Difficulty in higher dimensions

So far we have had no trouble obtaining cheap, safe, and accurate difference methods for solving *partial-differential equations* (PDEs). The implicit method has met all needs. But in space dimensions higher than one, the implicit method becomes prohibitively costly. For the common example of problems in which $\partial^2/\partial x^2$ becomes generalized to $\partial^2/\partial x^2 + \partial^2/\partial y^2$, we will learn the reason why. The simplest case is the heat-flow equation for which the Crank-Nicolson method gave us (1.37). Introducing the abbreviation $\delta_{xx} q = q^{x+1} - 2q^x + q^{x-1}$, equation (1.37) becomes

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

The nested expression on the left represents a tridiagonal matrix. The critical stage is in solving the tridiagonal simultaneous equations for the vector of unknowns Q_{t+1}. Fortunately there is a special algorithm for this solution, and the cost increases only linearly with the size of the matrix. Now turn from the one-dimensional physical space of x to two-dimensional (x, y)-space. Letting α denote the numerical constant in (1.48), the equation for stepping forward in time is

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

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

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

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

For algebraic purposes these sixteen numbers can be mapped into a vector. There are many ways to do this. A simple way would be to associate the locations in (1.50) with vector components by the column arrangement

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

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

$$\begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & -4 & 1 \\ \hline & 1 & \\ \hline \end{array} \tag{4.52}$$

Lay this star down in the (x, y)-plane (1.51) and move it around. Unfortunately, with just sixteen points, much of what you see is dominated by edges and corners. Try every position of the star that allows the center -4 to overlay one of the sixteen points. Never mind the 1's going off the sides. Start with the -4 in (1.52) over the 1 in the upper left corner of (1.51). Observe 1's on the 2 and the 5. Copy the 1's into the top row of Table 4.8 into the second and fifth columns. Then put the -4 in (1.52) over the 2 in (1.51). Observe 1's on the 1, 3, and 6. Copy the 1's into the next row of Table 4.8. Then put the -4 over the 3. Observe 1's on the 2, 4, and 7. Continue likewise. The 16×16 square matrix that results is shown in Table 4.8.

Now that Table 4.8 has been constructed we can return to the interpretation of equation (1.49). The matrix of unknowns Q_{t+1} has been mapped into a sixteen-point column vector, and the bracketed expression multiplying Q_{t+1} can be mapped into a 16×16 matrix. Clearly, the matrix contains zeroes everywhere that Table 4.8 contains dots. It seems fortunate that the table contains many zeroes, and we are led to hope for a rapid solution method for the simultaneous equations. The bad news is that no good method has ever been found. The best methods seem to require effort proportional to N^3, where in this case $N = 4$. Based on our experience in one dimension, those of us who worked on this problem hoped for a method proportional to N^2, which is the cost of an explicit method—essentially the cost of computing the right side of (1.41). Even all the features of implicit methods do not justify an additional cost of a factor of N. The next best thing is the splitting method.

EXERCISES:

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

2 Write the 45° diffraction equation in (x, z)-space for fixed ω in the form of (1.36).

4.3 WAVEMOVIE PROGRAM

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

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

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

Your problems all relate to the computer subroutine `wavemovie()` (Lynn, Gonzalez, JFC, Hale, Li, Karrenbach). It makes a movie of a sum of monochromatic waves. As it stands it will produce a movie (three-dimensional matrix) of waves propagating through a focus. The whole process from compilation through computation to finally viewing the film loop takes about a minute. A sample frame of the movie is in Figure 1.3. It shows a snapshot of the (x, z)-plane. Collapsing spherical waves enter from the top, go through a focus and then expand again. Notice that the wavefield is small but not zero in the region of geometrical shadow. In the shadow region you see waves that appear to be circles emanating from point sources at the top corners. Notice that the amplitudes of expanding spherical waves drop off with distance and collapsing spherical waves grow towards the focus. We will study the program that made this figure and see many features of waves and mathematics.

−4	1	·	·	1	·	·	·	·	·	·	·	·	·	·	·
1	−4	1	·	·	1	·	·	·	·	·	·	·	·	·	·
·	1	−4	1	·	·	1	·	·	·	·	·	·	·	·	·
·	·	1	−4	·	·	·	1	·	·	·	·	·	·	·	·
1	·	·	·	−4	1	·	·	1	·	·	·	·	·	·	·
·	1	·	·	1	−4	1	·	·	1	·	·	·	·	·	·
·	·	1	·	·	1	−4	1	·	·	1	·	·	·	·	·
·	·	·	1	·	·	1	−4	·	·	·	1	·	·	·	·
·	·	·	·	1	·	·	·	−4	1	·	·	1	·	·	·
·	·	·	·	·	1	·	·	1	−4	1	·	·	1	·	·
·	·	·	·	·	·	1	·	·	1	−4	1	·	·	1	·
·	·	·	·	·	·	·	1	·	·	1	−4	·	·	·	1
·	·	·	·	·	·	·	·	1	·	·	·	−4	1	·	·
·	·	·	·	·	·	·	·	·	1	·	·	1	−4	1	·
·	·	·	·	·	·	·	·	·	·	1	·	·	1	−4	1
·	·	·	·	·	·	·	·	·	·	·	1	·	·	1	−4

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

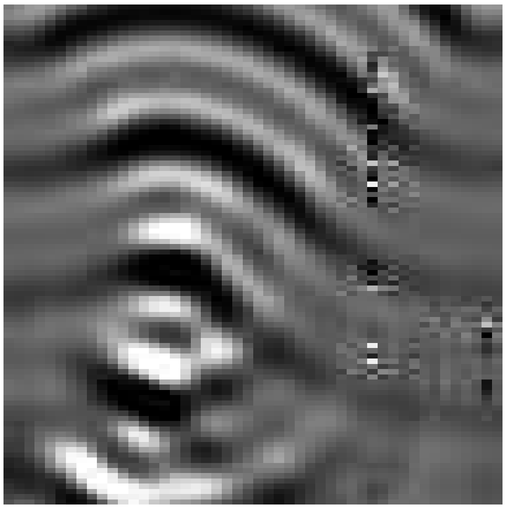


Figure 4.3: First frame of movie generated by `wavemovie()`. omx-Mfocus15 [ER]

4.3.1 Earth's surface boundary condition

The program that created Figure 1.3 begins with an initial condition along the top boundary, and then this initial wavefield is extrapolated downward. So, the first question is: what is the mathematical function of x that describes a collapsing spherical (actually cylindrical) wave? An expanding spherical wave has an equation $\exp[-i\omega(t - r/v)]$, where the radial distance is $r = \sqrt{(x - x_0)^2 + (z - z_0)^2}$ from the source. For a collapsing spherical wave we need $\exp[-i\omega(t + r/v)]$. Paranthetically, I'll add that the theoretical solutions are not really these, but something more like these divided by $\sqrt{r}$, actually they should be a Hankel functions, but the picture is little different when the exact initial condition is used. If you have been following this analysis, you should have little difficulty changing the initial conditions in the program to create a downgoing plane wave shown in Figure 1.4.

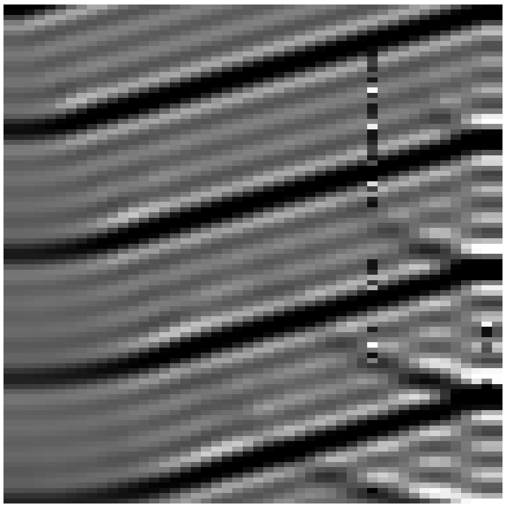


Figure 4.4: Specify program changes that give an initial plane wave propagating downward at an angle of 15° to the right of vertical. omx-Mdipplane [ER]

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

4.3.2 Time-domain analysis

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

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

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

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

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

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

This latter equation defines the time interval on the line

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

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

```
# from par:  integer n3:nt=12, n2:nx=48, n1:nz=96,  nw=2, nlam=4
# from par:  real dx=2, dz=1, v=1
#
subroutine wavemovie( nt, nx, nz, nw, nlam, dx,dz,v, p, cd, q)
integer it,nt,ix,nx,iz,nz,iw,nw, nlam
real dx,dz,v, phase,pi2,z0,x0,dt,dw,lambda,w,wov,x, p(nz,nx,nt)
complex aa,a,b,c,cshift, cd(nx),q(nx)
lambda=nz*dz/nlam;  pi2=2.*3.141592; dw=v*pi2/lambda;  dt=pi2/(nt*dw)
```

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

4.3.3 Internals of the film-loop program

The differential equation solved by the program is

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

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

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

Using the Crank-Nicolson differencing method this becomes

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

Absorb all the constants into one and define

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

getting

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

Bring the unknowns to the left:

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

We will solve this as we solved equations (1.37) and (1.38). The second stage is to solve the equation

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

analytically by

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

By alternating between (1.58) and (1.60), which are derived from (1.54) and (1.59), the program solves (1.53) by a splitting method. Formal justification of the splitting method follows in chapter 5. The program uses the tridiagonal solver discussed earlier, except the version here, `ctris()`, has all the real variables changed to complex.

Figure 1.5 shows a change of initial conditions where the incoming wave on the top frame is defined to be an impulse, namely, $p(x, z=0) = (\cdots, 0, 0, 1, 0, 0, \cdots)$. The result is alarmingly noisy. What is happening is that for any frequencies anywhere near the Nyquist frequency, the *difference* equation departs from the *differential* equation that it should mimic. This problem is addressed, analyzed, and ameliorated in chapter 8. For now, the best thing to do is to avoid sharp corners in the initial wave field.

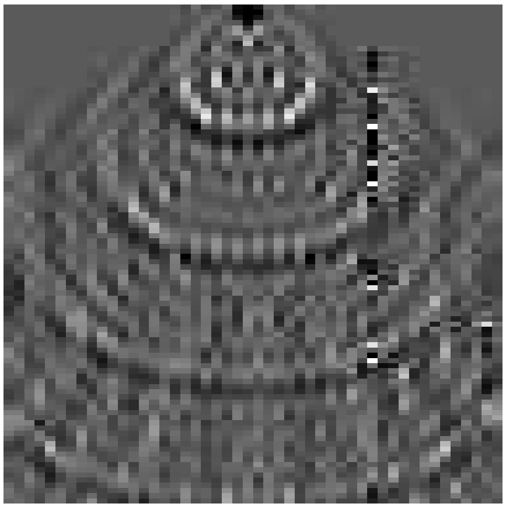


Figure 4.5: Observe and describe various computational artifacts by testing the program using a point source at $(x, z) =$ `(xmax/2,0)`. Such a source is rich in the high spatial frequencies for which difference equations do not mimic their differential counterparts. omx-Mcompart [ER]

4.3.4 Side-boundary analysis

In geophysics, we usually wish the side-boundary question did not arise. The only real reason for side boundaries is that either our survey or our processing activity is necessarily limited in extent. Given that side boundaries are inevitable, we must think about them. The subroutine `wavemovie()` included zero-slope boundary conditions. This type of boundary treatment resulted from taking

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

and in the call to `ctris` taking

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

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

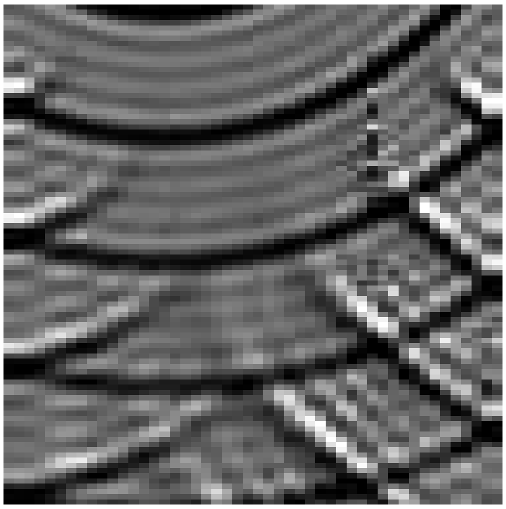


Figure 4.6: Given that the domain of computation is $0 \leq x \leq$ `xmax` and $0 \leq z \leq$`zmax`, how would you modify the initial conditions at $z = 0$ to simulate a point source at $(x, z) =$ `(xmax/3, -zmax/2)`? omx-Mexpandsphere [ER]

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

Compare the side-boundary behavior of Figures 1.6 and 1.7.

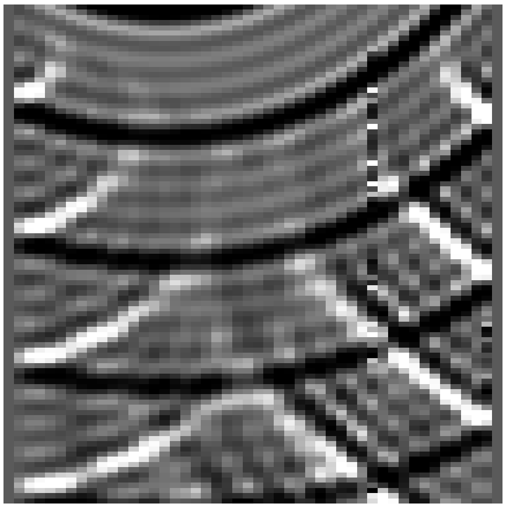


Figure 4.7: Modify the program so that zero-slope side boundaries are replaced by zero-value side boundaries. omx-Mzeroslope [ER]

The zero slope boundary condition is explicitly visible as identical signal on the two end columns. Likewise, the zero-value side boundary condition has a column of zero-valued signal on each side. In each case, the end trace need not be explicitly present, and it is not present in Dave Hale's coding of the boundary conditions which I give, but do not derive, below.

```
p0 = bl * p(1);          pnxp1 = br * p(nx)

cd(1)   =  aa * p(2)    + ( 1. - 2. * aa ) * p( 1) + aa * p0
cd(nx)  =  aa * p(nx-1) + ( 1. - 2. * aa ) * p(nx) + aa * pnxp1

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

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

Note that `bl=br=0` for zero-value boundaries, and `bl=br=1` for zero-slope boundaries. Absorbing side boundaries, derived in chapter 9 are obtained by letting `bl` and `br` be complex.

4.3.5 Including 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 (1.60). If the wavefield is such that k_x is small enough, then equation (1.60) is the only place it is needed. Second, it enters in the tridiagonal coefficients through the v in equation (1.56). The so-called thin-lens approximation of optics seems to amount to including the equation (1.60) part only. An example of lateral velocity variation is in Figure 1.8.

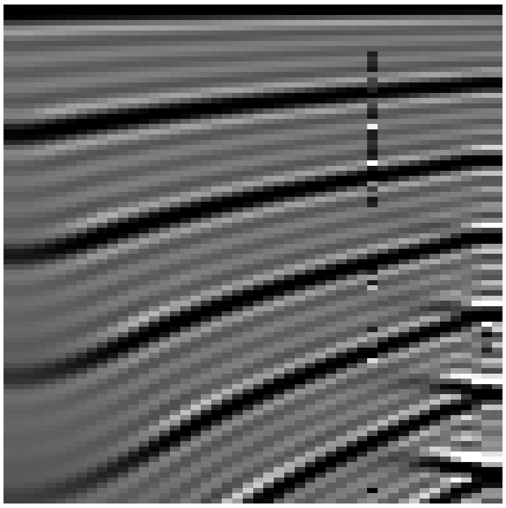


Figure 4.8: Make changes to the program to include a thin-lens term with a lateral velocity change of 40% across the frame produced by a constant slowness gradient. Identify other parts of the program which are affected by lateral velocity variation. You need not make these other changes. Why are they expected to be small? omx-Mlateralvel [ER]

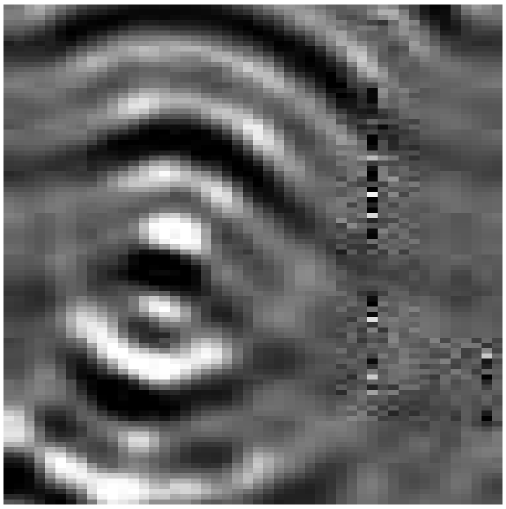


Figure 4.9: Incorporate the 45° term, ∂_{xxz}, for the collapsing spherical wave. Use zero-slope sides. Compare your result with the 15° result obtained via the program `wavemovie`. Mark an X at the theoretical focus location. omx-Mfortyfive [ER]

4.3.6 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 antisymmetrical waveform emerges. This is easily seen in Figure 1.10. In migrations, waves go just *to* a focus, not *through* it. So the migration impulse

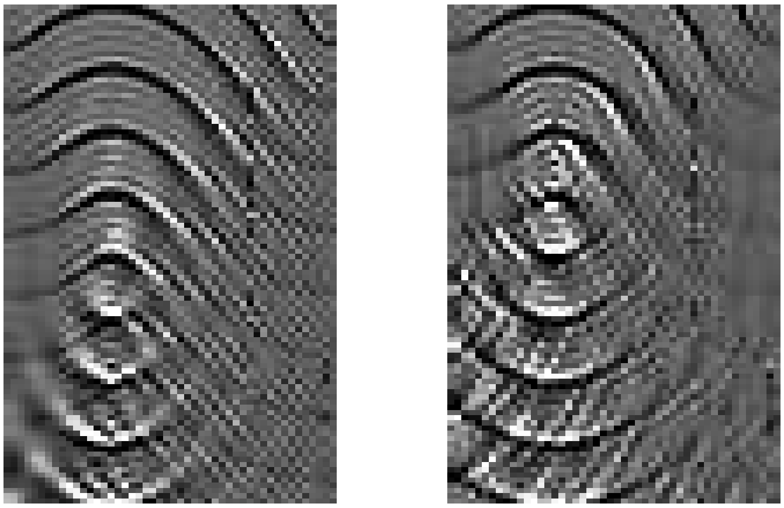


Figure 4.10: The accuracy of the x-derivative may be improved by a technique that is analyzed later in chapter 8. Briefly, instead of representing $k_x^2\,\Delta x^2$ by the tridiagonal matrix $\mathbf{T}$ with $(-1, 2, -1)$ on the main diagonal, you use $\mathbf{T}/(\mathbf{I} - \mathbf{T}/6)$. Modify the extrapolation analysis by multiplying through by the denominator. Make the necessary changes to the 45° collapsing wave program. Left without 1/6 trick; right, with 1/6 trick. omx-trick [ER]

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.

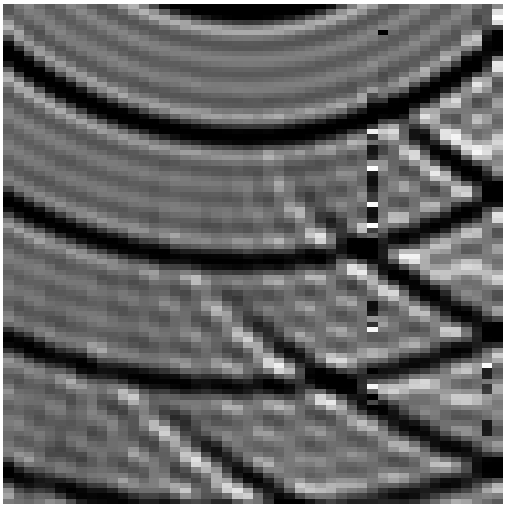


Figure 4.11: Chapter 9 explains how to absorb energy at the side boundaries. Make the necessary changes to the program to absorb waves incident on the left-side boundary. omx-Mabsorbside [ER]

4.4 MIGRATION IN THE (omega,x)-DOMAIN (KJ)

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. The input will be in the time domain and will have to be Fourier transformed. The beginning of the program 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.

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 (1.53). 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

$$\frac{\partial}{\partial z} \quad = \quad \frac{\partial \tau}{\partial z}\,\frac{\partial}{\partial \tau} \quad = \quad \frac{1}{v}\,\frac{\partial}{\partial \tau} \tag{4.61}$$

Substitution into (1.53) gives

$$\frac{\partial P}{\partial \tau} \quad = \quad -\,i\,\omega\;P \quad - \quad \frac{v^2}{-\,i\,\omega\,2}\;\frac{\partial^2 P}{\partial x^2} \tag{4.62}$$

In the program, the time sample size `dt` $= \Delta t$ and the travel-time depth sample `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. A program accompanies.

```
#%  Migration in the (omega,x,z)-domain
subroutine kjartjac()
real p(48,64), pi, alpha, dt, dtau, dw, w0, omega
complex cp(48,64), cd(48), ce(48), cf(48), aa, a, b, c, cshift
integer ix, nx, iz, nz, iw, nw, it, nt, esize
nt= 64;   nz= nt;    nx= 48;      pi= 3.141592
dt= 1.;   dtau= 1.;  w0=-pi/dt;   dw= 2*pi/(dt*nt);    nw= nt/2;
alpha = .25                                     #  alpha = v*v*dtau/(4*dx*dx)
do iz= 1, nz { do ix=1,nx { p(ix,iz) = 0.;    cp(ix,iz)=0. }}
do it= nt/3, nt, nt/4{
        do ix= 1, 4                             # Broadened impulse source
                { cp(ix,it) = (5.-ix);     cp(ix,it+1) = (5.-ix) }}
call ft2axis( 0, 1., nx,nt, cp)
do iz= 1, nz {
do iw= 2, nw {    omega = w0 + dw*(iw-1)
        aa = - alpha /( (0.,-1.)*omega )
        a = -aa;      b = 1.+2.*aa;      c = -aa
        do ix= 2, nx-1
                cd(ix) = aa*cp(ix+1,iw) + (1.-2.*aa)*cp(ix,iw) + aa*cp(ix-1,iw)
        cd(1) = 0.;      cd(nx) = 0.
        call ctris( nx, -a, a, b, c, -c, cd, cp(1,iw))
        cshift = cexp( cmplx( 0.,-omega*dtau))
        do ix= 1, nx
                cp(ix,iw) = cp(ix,iw) * cshift
        do ix= 1, nx
                p(ix,iz) = p(ix,iz)+cp(ix,iw)   # p(t=0) = Sum P(omega)
        }}
esize=4
to history:     integer n1:nx, n2:nz, esize
call hclose()
call srite( 'out', p, nx*nz*4 )
return; end
```

The output of the program is shown in Figure 1.12. 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 broad-banded 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.

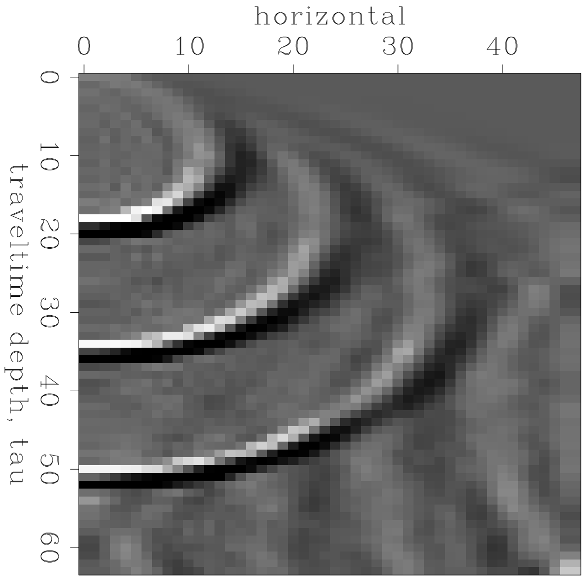


Figure 4.12: Output of the program `kjartjac`: semicircle approximations. omx-kjartjac [ER]

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.