Chapter 1

Finite-difference migration

This chapter is a condensation of wave extrapolation and finite-difference basics from IEI which is now out of print. On the good side, this new organization is more compact and several errors have been corrected. On the bad side, to follow up on many many interesting details you will need to find a copy of IEI (http://sepwww.stanford.edu/sep/prof/).

In 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 $\exp[ik_z(\omega,k_x)z]$. In this chapter instead of multiplying a wavefield by a function of k_x to downward continue waves, we will convolve them along the x-axis with a small filter that represents a differential equation. On space axes, a concern is the seismic velocity v. With **lateral velocity variation**, 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 in the spatial frequency domain are no longer multiplication, but convolution.) The alternative we choose here is to go to finite differences which are convolution in the physical x domain. This is what the wave equation itself does.

1.1 THE PARABOLIC EQUATION

Here we derive the most basic migration equation via the dispersion relation, equation (**??**). Recall this equation basically says $\cos\theta = \sqrt{1-\sin^2\theta}$.

$$k_z \quad = \quad \frac{\omega}{v}\sqrt{1 \;-\; \frac{v^2k_x^2}{\omega^2}} \tag{1.1}$$

The dispersion relation above is the foundation for downward continuing wavefields by Fourier methods in chapter **??**. Recall that nature extrapolates forward in time from $t=0$ whereas a geophysicist extrapolates information in depth from $z=0$. We get ideas for our task, and then we hope to show that our ideas are consistent with nature. Suppose we substitute $ik_z = \partial/\partial z$

into equation (1.1), multiply by P, and interpret velocity as depth variable.

$$\frac{\partial P}{\partial z} \quad = \quad \frac{i\,\omega}{v(z)} \sqrt{1 \;-\; \frac{v(z)^2 k_x^2}{\omega^2}}\; P \tag{1.2}$$

Since the above steps are unorthodox, we need to enquire about their validity. Suppose that equation (1.2) were valid. Then we could restrict it to constant velocity and take a trial solution $P = P_0 \exp(-i k_z z)$ and we would immediately have equation (1.1). Why do we believe the introduction of $v(z)$ in equation (1.2) has any validity? We can think about the phase shift migration method in chapter **??**. It handled $v(z)$ by having the earth velocity being a staircase function of depth. Inside a layer we had the solution to equation (1.2). To cross a layer boundary, we simply asserted that the wavefield at the bottom of one layer would be the same as the wavefield at the top of the next which is also the solution to equation (1.2). (Let $\Delta z \to 0$ be the transition from one layer to the next. Then $\Delta P = 0$ since $\partial P / \partial z$ is finite.) Although equation (1.2) is consistent with chapter **??**, it is an approximation of limited validity. It assumes there is no reflection at a layer boundary. Reflection would change part of a downgoing wave to an upcoming wave and the wave that continued downward would have reduced amplitude because of lost energy. Thus, by our strong desire to downward continue wavefields (extrapolate in z) whereas nature extrapolates in t, we have chosen to ignore reflection and transmission coefficients. Perhaps we can recover them, but now we have bigger fish to fry. We want to be able to handle $v(x,z)$, **lateral velocity variation**. This requires us to get rid of the square root in equation (1.2). Make a power series for it and drop higher terms.

$$\frac{\partial P}{\partial z} \quad = \quad \frac{i\,\omega}{v(z)} \left(1 \;-\; \frac{v(z)^2 k_x^2}{2\,\omega^2} \right) P \;+\; \cdots \tag{1.3}$$

The first dropped term is $\sin^4\theta$ where θ is the dip angle of a wavefront. The dropped terms increase slowly with angle, but they do increase, and dropping them will limit the range of angles that we can handle with this equation. This is a bitter price to pay for the benefit of handling $v(x,z)$, and we really will return to patch it up (unlike the transmission coefficient problem). There are many minus signs cropping up, so I give you another equation to straighten them out.

$$\frac{\partial P}{\partial z} \quad = \quad \left(\frac{i\,\omega}{v(z)} \;-\; \frac{v(z) k_x^2}{-i\,\omega 2} \right) P \tag{1.4}$$

Now we are prepared to leap to our final result, an equation for downward continuing waves in the presence of depth and **lateral velocity variation** $v(x,z)$. Substitute $\partial_{xx} = -k_x^2$ into equation (1.4) and revise interpretation of P from $P(\omega, k_x, z)$ to $P(\omega, x, z)$.

$$\frac{\partial P}{\partial z} \quad = \quad \frac{i\,\omega}{v(x,z)}\; P \;+\; \frac{v(x,z)}{-i\,\omega 2}\, \frac{\partial^2 P}{\partial x^2} \tag{1.5}$$

As with $v(z)$, there is a loss of lateral transmission and reflection coefficients. We plan to forget this minor problem. It is the price of being a data handler instead of a modeler. Equation (1.5) is the basis for our first program and examples.

1.2 SPLITTING AND 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.*

1.2.1 The heat-flow equation

We wish to solve equation (1.5) by a method involving **splitting**. Since equation (1.5) is an unfamiliar one, we turn to the **heat-flow equation** which besides being familiar, has no complex numbers. A two-sentence derivation of the **heat-flow equation** 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

$$\frac{\partial T}{\partial t} \quad = \quad \left(\sigma \, \frac{\partial^2}{\partial x^2} + \sigma \, \frac{\partial^2}{\partial y^2} \right) T \tag{1.6}$$

1.2.2 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:

$$\frac{\partial T}{\partial t} \quad = \quad 2\sigma \, \frac{\partial^2 T}{\partial x^2} \qquad (\text{all } y) \tag{1.7}$$

$$\frac{\partial T}{\partial t} \quad = \quad 2\sigma \, \frac{\partial^2 T}{\partial y^2} \qquad (\text{all } x) \tag{1.8}$$

In equation (1.7) 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 (1.8). At odd moments in time heat flows according to (1.7) and at even moments in time it flows according to (1.8). This solution by alternation between (1.7) and (1.8) can be proved mathematically to converge to the solution to (1.6) with errors of the order of Δt. Hence the error goes to zero as Δt goes to zero.

1.2.3 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 (1.7) and (1.8) in which, instead of alternating back and forth between them at alternate time steps, what is done is to march (1.7) through all time steps. Then this intermediate result is used as an initial condition for (1.8), 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 (1.6). But if σ is a constant function of x and y, it does. The process is depicted in Figure 1.1 for an impulsive initial disturbance. A differen-

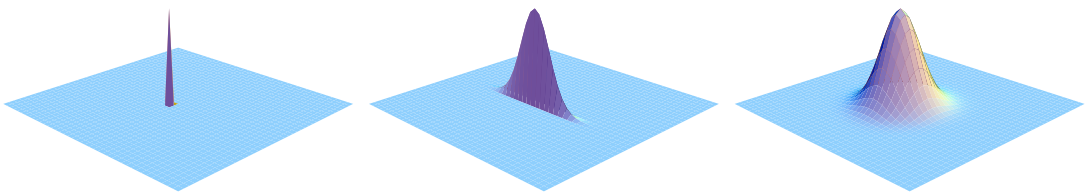


Figure 1.1: Temperature distribution in the (x, y)-plane beginning from a delta function (left). After heat is allowed to flow in the x-direction but not in the y-direction the heat is located in a "wall" (center). Finally allowing heat to flow for the same amount of time in the y-direction but not the 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). fdm-temperature [CR]

tial equation like (1.6) 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.

There are circumstances which dictate a middle road between **splitting** and **full 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 (1.7) before you transpose the data and switch over to (1.8). Circumstances like this one but with more geophysical interest arise with the wave-extrapolation equation that is considered next.

1.2.4 Splitting the parabolic equation

In discussing and solving the **parabolic wave equation** it is convenient to rearrange it to recognize the role of an averaged stratified medium of velocity of $\bar{v}(z)$ and departures from it.

$$\begin{aligned} \frac{\partial P}{\partial z} &= i\,\omega \left(\frac{1}{\bar{v}(z)}\right) P \;+\; i\,\omega \left(\frac{1}{v(x,z)} - \frac{1}{\bar{v}(z)}\right) P \;+\; \left(\frac{v(x,z)}{-i\,\omega 2}\,\frac{\partial^2}{\partial x^2}\right) P \\ &= \quad A\,P \quad + \quad B\,P \quad + \quad C\,P \\ &= \quad \text{shift} \quad + \quad \text{thin lens} \quad + \quad \text{diffraction} \end{aligned} \tag{1.9}$$

The shift term in (1.9) commutes with the thin-**lens term**, that is, $ABP = BAP$. the shift term also commutes with the diffraction term because $ACP = CAP$. But the thin-lens term and the diffraction term do not commute with one another because $(BC - CB)P \neq 0$, because

$$0 \quad \neq \quad (BC-CB)P \quad = \quad v(x,z)\left[\left(-2\frac{d^2}{dx^2}\frac{1}{v(x,z)}\right) + \frac{1}{v(x,z)^2}\frac{dv(x,z)}{dx}\frac{\partial}{\partial x}\right] P \tag{1.10}$$

Mathematicians look at the problem this way: Consider any fixed wave propagation angle so vk_x/ω is a constant. Now let frequency ω (and hence k_x) tend together to infinity. The terms in BCP and CBP grow in proportion to the second power of frequency, whereas those in $(BC - CB)P$ grow as lower powers. There is however, a catch. The material properties have a "wavelength" too. We can think of $(dv/dx)/v$ as a spatial wavenumber for the material just as k_x is a spatial wavenumber for the wave. If the material contains a step function change in its properties, that is an infinite spatial frequency $(dv/dx)/v$ for the material. Then the $(BC - CB)P$ terms dominate near the place where one material changes to another. If we drop the $(BC - CB)P$ terms, we'll get the transmission coefficient incorrect, although everything would be quite fine everywhere else except at the boundary.

A question is, to what degree do the 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 velocity variation** occurring in the **lens term** is known only to a limited accuracy and we often wish to determine $v(x)$ by the extrapolation procedure.

In practice it seems best to forget the $(BC - CB)P$ terms because we hardly ever know the material properties very well anyway. Then we split, doing the shift and the thin-lens part analytically while doing the diffraction part by a numerical method.

1.2.5 Validity of the splitting and full-separation concepts

Feel free to skip forward over this subsection which is merely a mathematical proof.

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.

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 differential operators or their finite-difference representations. 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 condition**s of a differential operator are incorporated in the corners of the matrix. The bottom line is whether $\mathbf{AB} = \mathbf{BA}$, so the question clearly involves the **boundary** conditions as well as the differential operators.

Extrapolation downward a short distance can be done with the operator $(\mathbf{I} + \mathbf{A}\,\Delta z)$. Let $\mathbf{p}$ denote a vector where components of the vector designate the wavefield at various locations on the x-axis. Numerical analysis gives us a matrix operator, say $\mathbf{A}$, which enables us to project forward, say,

$$\mathbf{p}(z + \Delta z) \quad = \quad \mathbf{A}_1\, \mathbf{p}(z) \tag{1.11}$$

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,

$$\mathbf{p}(z + 2\,\Delta z) \quad = \quad \mathbf{A}_2\, [\mathbf{A}_1\, \mathbf{p}(z)] \tag{1.12}$$

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

$$\mathbf{A}_2\, [\mathbf{A}_1\, \mathbf{p}(z)] \quad = \quad (\mathbf{A}_2\, \mathbf{A}_1)\, \mathbf{p}(z) \tag{1.13}$$

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

$$P(z_0 + \Delta z) \quad = \quad (\mathbf{I} + \mathbf{A}\,\Delta z)\,(\mathbf{I} + \mathbf{B}\,\Delta z)\,P(z_0) \tag{1.14}$$

Applying (1.14) in many stages gives a product of many operators. The operators **A** and **B** are subscripted with j to denote the possibility that they change with z.

$$P(z_1) \quad = \quad \prod_{j=1}^{N} [(\mathbf{I} + \mathbf{A}_j\,\Delta z)(\mathbf{I} + \mathbf{B}_j\,\Delta z)]\;P(z_0) \tag{1.15}$$

As soon as **A** and **B** are assumed to be commutative, the factors in (1.15) may be rearranged at will. For example, the **A** operator could be applied in its entirety before the **B** operator is applied:

$$P(z_1) \quad = \quad \left[\prod_{j=1}^{N} (\mathbf{I} + \mathbf{B}_j\,\Delta z)\right] \left[\prod_{j=1}^{N} (\mathbf{I} + \mathbf{A}_j\,\Delta z)\right] P(z_0) \tag{1.16}$$

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

1.3 FINITE DIFFERENCING IN (omega,x)-SPACE

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, a few easily learned tricks go a long way, and we will be covering them here.

1.3.1 The lens equation

The parabolic wave-equation operator 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.

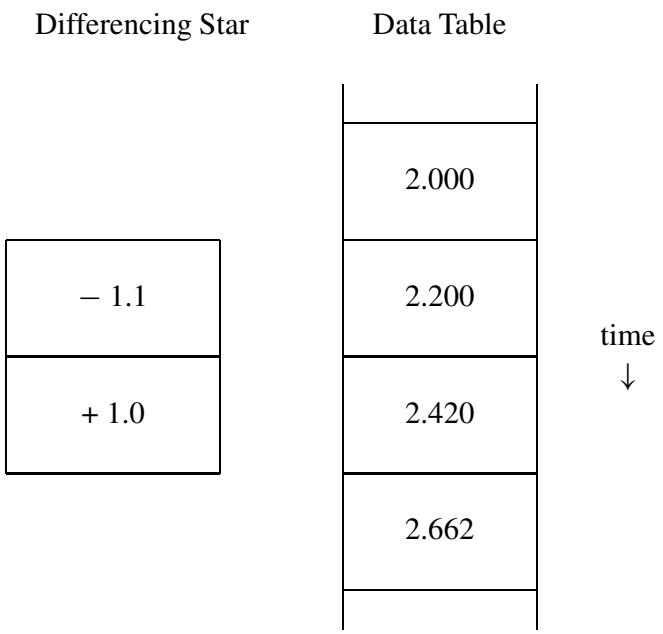
1.3.2 First derivatives, explicit method

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

$$q_{t+1} - q_t = .10\, q_t \tag{1.17}$$

$$(1.0)\, q_{t+1} + (-1.1) q_t = 0 \tag{1.18}$$

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



Since the data in the data table satisfy the difference equations (1.17) and (1.18), 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.

1.3.3 First derivatives, implicit method

Let us solve the equation

$$\frac{dq}{dt} = 2\, r\, q \tag{1.19}$$

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

$$\frac{dq}{dt} \approx \frac{q(t+\Delta t) - q(t)}{\Delta t} \tag{1.20}$$

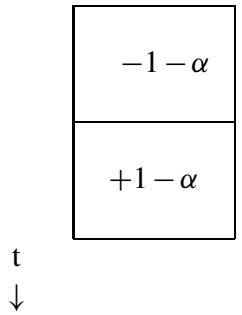
Using this in equation (1.19) yields the the inflation-of-money equations (1.17) and (1.18), where $2r = .1$. Thus in the inflation-of-money equation the expression of dq/dt is centered at $t + \Delta t/2$, whereas the expression of q by itself is at time t. There is no reason the q on the right side of equation (1.19) 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 (1.19) is

$$q_{t+1} - q_t = 2r\,\Delta t\ \frac{q_{t+1} + q_t}{2} \tag{1.21}$$

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

$$(1-\alpha)\,q_{t+1} - (1+\alpha)\,q_t = 0 \tag{1.22}$$

which is representable as the difference star



For a fixed Δt this star gives a more accurate solution to the differential equation (1.19) 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**.

1.3.4 Explicit heat-flow equation

The **heat-flow equation** (1.6) is a prototype for migration. Let us recopy the heatflow equation letting q denote the temperature.

$$\frac{\partial q}{\partial t} = \frac{\sigma}{C}\,\frac{\partial^2 q}{\partial x^2} \tag{1.23}$$

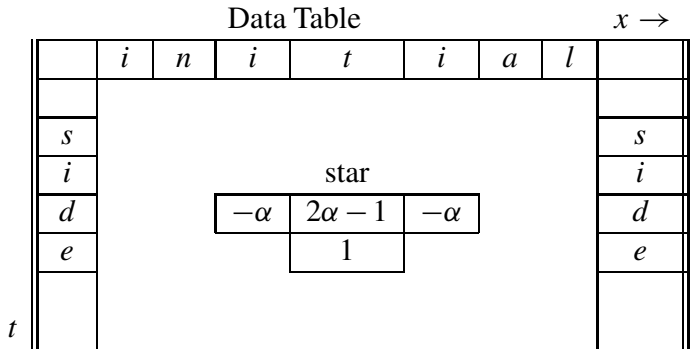


Table 1.1: Differencing star and table for one-dimensional **heat-flow equation**.

Implementing (1.23) in a computer requires some difference approximations for the partial differentials. As before we use a subscript notation that allows (1.20) to be compacted into

$$\frac{\partial q}{\partial t} \approx \frac{q_{t+1} - q_t}{\Delta t} \tag{1.24}$$

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 $q_{t+2} - 2q_{t+1} + q_t$. The formula is usually treated more symmetrically by shifting it to $q_{t+1} - 2q_t + q_{t-1}$. 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:

$$\frac{\partial^2 q}{\partial x^2} \approx \frac{q^{x+1} - 2q^x + q^{x-1}}{\Delta x^2} \tag{1.25}$$

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

$$\frac{q_{t+1}^x - q_t^x}{\Delta t} = \frac{\sigma}{C} \frac{q_t^{x+1} - 2q_t^x + q_t^{x-1}}{\Delta x^2} \tag{1.26}$$

(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 IEI chapter 4. Letting $\alpha = \sigma\,\Delta t/(C\,\Delta x^2)$, equation (1.26) becomes

$$q_{t+1}^x - q_t^x - \alpha(q_t^{x+1} - 2q_t^x + q_t^{x-1}) = 0 \tag{1.27}$$

Equation (1.27) 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 (1.27) can be interpreted geometrically as a computational star in the (x, t)-plane, as depicted in Table 1.1. 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

many possible arrangements of initial and side conditions, such as zero-value side conditions. Next is a computer program for the job 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
```

1.3.5 The leapfrog method

A 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 mathematical reason the divergence occurs is the subject of mathematical analysis found in IEI section 2.8. Intuitively, 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 $t+1$:

$$\frac{\partial q}{\partial t} \quad \approx \quad \frac{q_{t+1} \; - \; q_{t-1}}{2 \; \Delta t} \tag{1.28}$$

Here the result is even worse. An analysis found in IEI 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***.

1.3.6 The Crank-Nicolson method

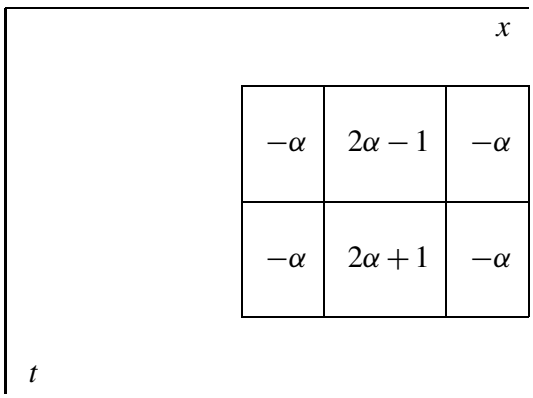
The **Crank-Nicolson method** solves both the accuracy and the stability problem. Recall the difference representation of the **heat-flow equation** (1.27).

$$q_{t+1}^{x} \; - \; q_t^x \quad = \quad a \; \left(q_t^{x+1} - 2q_t^x + q_t^{x-1} \right) \tag{1.29}$$

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

$$q_{t+1}^{x} - q_t^x \quad = \quad \frac{a}{2} \left[\left(q_t^{x+1} - 2q_t^x + q_t^{x-1} \right) + \left(q_{t+1}^{x+1} - 2q_{t+1}^x + q_{t+1}^{x-1} \right) \right] \tag{1.30}$$

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



(1.31)

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 (1.30) to the left and the t terms to the right, obtaining

$$-\alpha q_{t+1}^{x+1} + (1+2\alpha) q_{t+1}^{x} - \alpha q_{t+1}^{x-1} \quad = \quad \alpha q_t^{x+1} + (1-2\alpha) q_t^x + \alpha q_t^{x-1} \tag{1.32}$$

Now think of the left side of equation (1.32) 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 (1.32) as a set of simultaneous equations. For definiteness, take the x-axis to be limited to five points. Then these equations are:

$$\begin{bmatrix} e_{\text{left}} & -\alpha & 0 & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & 0 & 0 \\ 0 & -\alpha & 1+2\alpha & -\alpha & 0 \\ 0 & 0 & -\alpha & 1+2\alpha & -\alpha \\ 0 & 0 & 0 & -\alpha & e_{\text{right}} \end{bmatrix} \begin{bmatrix} q_{t+1}^1 \\ q_{t+1}^2 \\ q_{t+1}^3 \\ q_{t+1}^4 \\ q_{t+1}^5 \end{bmatrix} = \begin{bmatrix} d_t^1 \\ d_t^2 \\ d_t^3 \\ d_t^4 \\ d_t^5 \end{bmatrix} \tag{1.33}$$

Equation (1.32) does not give us each q_{t+1}^x *explicitly*, but equation (1.33) 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 condition**s. 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 (1.33) 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 (1.27). In fact, this **implicit method** turns out to be cheaper, since the increased accuracy of (1.32) over (1.27) 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
```

1.3.7 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

$$a_j\, q_{j+1} \;+\; b_j\, q_j \;+\; c_j\, q_{j-1} \quad = \quad d_j \tag{1.34}$$

Introduce new unknowns e_j and f_j, along with an equation

$$q_j \quad = \quad e_j\, q_{j+1} \;+\; f_j \tag{1.35}$$

Write (1.35) with shifted index:

$$q_{j-1} \quad = \quad e_{j-1}\, q_j \;+\; f_{j-1} \tag{1.36}$$

Insert (1.36) into (14):

$$a_j\, q_{j+1} \;+\; b_j\, q_j \;+\; c_j\, (e_{j-1}\, q_j \;+\; f_{j-1}) \quad = \quad d_j \tag{1.37}$$

Now rearrange (1.37) to resemble (15):

$$q_j \quad = \quad \frac{-\,a_j}{b_j \;+\; c_j\, e_{j-1}}\; q_{j+1} \;+\; \frac{d_j \;-\; c_j\, f_{j-1}}{b_j \;+\; c_j\, e_{j-1}} \tag{1.38}$$

Compare (1.38) to (1.35) to see recursions for the new unknowns e_j and f_j:

$$e_j \quad = \quad \frac{-\,a_j}{b_j \;+\; c_j\, e_{j-1}} \tag{1.39}$$

$$f_j \quad = \quad \frac{d_j \;-\; c_j\, f_{j-1}}{b_j \;+\; c_j\, e_{j-1}} \tag{1.40}$$

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 (1.35) 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 (1.39) to compute all the e_j.

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

$$c_{n-1}\, q_{n-1} \;+\; e_{\text{right}}\, q_n \quad = \quad d_n \tag{1.41}$$

Equation (1.41) includes as special cases the zero-value and zero-slope **boundary condition**s. Equation (1.41) can be compared to equation (1.36) at its end.

$$q_{n-1} \quad = \quad e_{n-1}\, q_n \;+\; f_{n-1} \tag{1.42}$$

Both q_n and q_{n-1} are unknown, but in equations (1.41) and (1.42) we have two equations, so the solution is easy. The final step is to take the value of q_n and use it in (1.36) to compute q_{n-1}, q_{n-2}, q_{n-3}, etc. The subroutine `rtris()` solves equation (1.33) for q where `n=5`, `endl`$= e_{\text{left}}$, `endr`$= e_{\text{right}}$, `a=c`$= -\alpha$, and `b`$= 1 - 2\alpha$.

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

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

1.3.8 Finite-differencing in the time domain

IEI develops time-domain finite differencing methods. Since the earth velocity is unvarying in time, a "basics only" book such as this omits this topic since you can, in principle, accomplish the same goals in the ω-domain. There are some applications, however, that give rise to time-variable coefficients in their partial differential equations. Recursive dip filtering is one application. Residual migration is another. Some formulations of DMO are another.

1.4 WAVEMOVIE PROGRAM

Here we see solutions to exercises stated in figure captions. The problems and solutions were worked over by former teaching assistants. (Lynn, Gonzalez, JFC, Hale, Li, Karrenbach, Fomel). The various figures are all variations of the computer subroutine `wavemovie()`. 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 a few seconds. A sample frame of the **movie** is in Figure 1.2. 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.

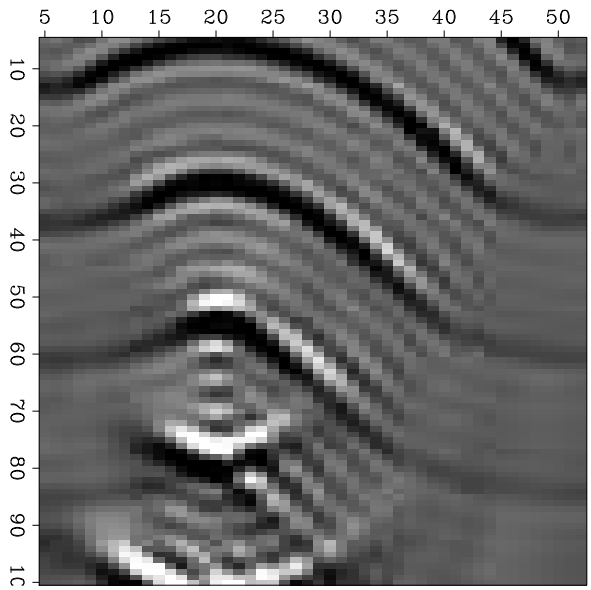


Figure 1.2: First frame of movie generated by `wavemovie()`. (Press button for movie.) fdm-Mfocus1590 [ER,M]

1.4.1 Earth surface boundary condition

The program that created Figure 1.2 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 $\exp[-i\omega(t+r/v)]$. Parenthetically, 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 function**s, but the picture is hardly 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 the downgoing plane wave shown in Figure 1.3. Notice the weakened waves

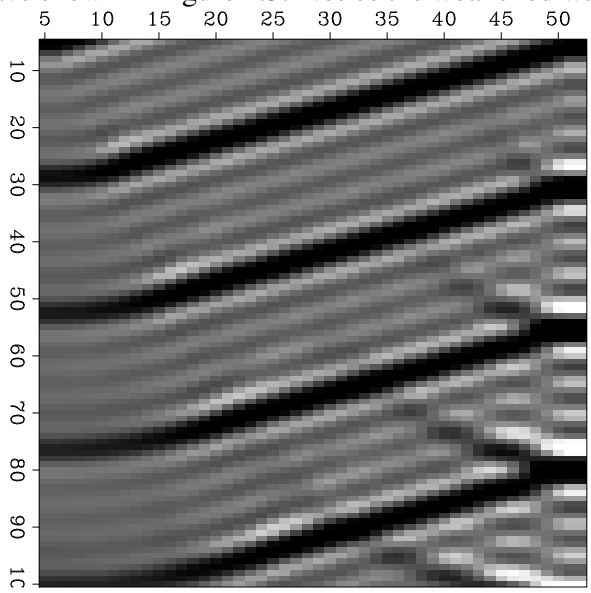


Figure 1.3: Specify program changes that give an initial plane wave propagating downward at an angle of 15° to the right of vertical. (Movie) fdm-Mdipplane90 [ER,M]

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" appearance near the reflecting boundary. Even this figure with `nw=4` shows the tendency.

1.4.2 Frames changing with time

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 $\omega_j = \Delta\omega, 2\,\Delta\omega$, ..., `nw` $\Delta\omega$. The lowest frequency `dw` = $\Delta\omega$ must be inversely proportional to the wavelength `lambda` $= \lambda$

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

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

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

This latter equation defines the time interval on the line

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

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

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

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

1.4.3 Internals of the film-loop program

The differential equation solved by the program is equation (1.5), copied here as

$$\frac{\partial P}{\partial z} \quad = \quad \frac{i\,\omega}{v(x,z)}\ P \ + \ \frac{v}{-i\,\omega 2}\,\frac{\partial^2 P}{\partial x^2} \tag{1.43}$$

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

$$\frac{\partial P}{\partial z} \quad = \quad \frac{v}{-i\,\omega 2}\,\frac{\partial^2 P}{\partial x^2} \tag{1.44}$$

Using the **Crank-Nicolson** differencing method this becomes

$$\frac{p_{z+1}^x - p_z^x}{\Delta z} \quad = \quad \frac{v}{-i\omega 2}\left(\frac{p_z^{x+1} - 2p_z^x + p_z^{x-1}}{2\,\Delta x^2} + \frac{p_{z+1}^{x+1} - 2p_{z+1}^x + p_{z+1}^{x-1}}{2\,\Delta x^2}\right) \tag{1.45}$$

Absorb all the constants into one and define

$$\alpha \quad = \quad \frac{v\,\Delta z}{-i\,\omega 4\,\Delta x^2} \tag{1.46}$$

getting

$$p_{z+1}^x - p_z^x \quad = \quad \alpha\left[\,(\,p_z^{x+1} - 2\,p_z^x + p_z^{x-1}) + (\,p_{z+1}^{x+1} - 2\,p_{z+1}^x + p_{z+1}^{x-1})\right] \tag{1.47}$$

Bring the unknowns to the left:

$$-\alpha p_{z+1}^{x+1} + (1+2\alpha) p_{z+1}^{x} - \alpha p_{z+1}^{x-1} \quad = \quad \alpha p_{z}^{x+1} + (1-2\alpha) p_{z}^{x} + \alpha p_{z}^{x-1} \tag{1.48}$$

We will solve this as we solved equations (1.32) and (1.33). The second stage is to solve the equation

$$\frac{\partial P}{\partial z} \quad = \quad \frac{i\,\omega}{v}\; P \tag{1.49}$$

analytically by

$$P(z+\Delta z) \quad = \quad P(z)\; e^{i \Delta z\; \omega / v} \tag{1.50}$$

By alternating between (1.48) and (1.50), which are derived from (1.44) and (1.49), the program solves (1.43) by a **splitting** method. The program uses the **tridiagonal** solver discussed earlier, like subroutine `rtris()` on page 14 except that version needed here, `ctris()`, has all the real variables declared complex.

Figure 1.4 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 IEI. For now, the best thing to do is to avoid sharp corners in the initial wave field.

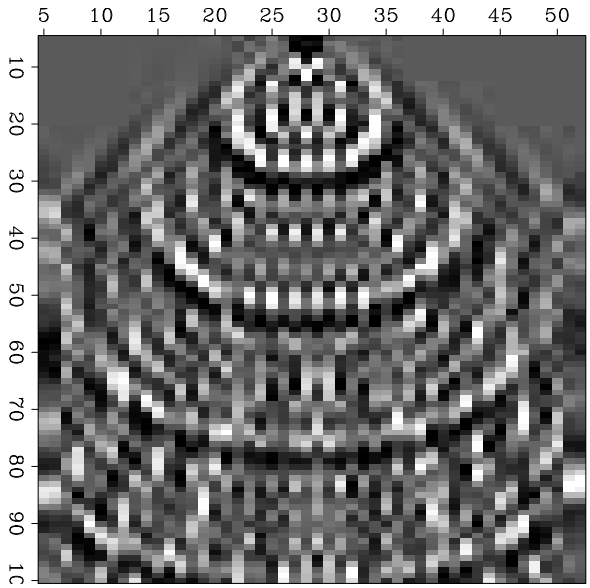


Figure 1.4: Observe and describe various computational artifacts by testing the program using a point source at (x,z) = `(xmax/2,0)`. Such a source is rich in the high spatial frequencies for which difference equations do not mimic their differential counterparts. (Movie) fdm-Mcompart90 [ER,M]

1.4.4 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

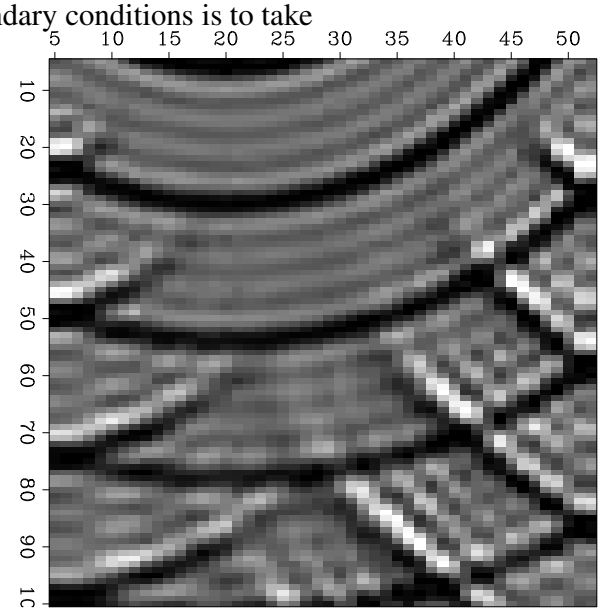


Figure 1.5: Given that the domain of computation is $0 \le x \le$ `xmax` and $0 \le z \le$`zmax`, how would you modify the initial conditions at $z = 0$ to simulate a point source at $(x,z) =$ `(xmax/3, -zmax/2)`? (Movie) fdm-Mexpandsphere90 [ER,M]

```
endl = endr =  10^30  ≈  ∞
```

Compare the side-boundary behavior of Figures 1.5 and 1.6.

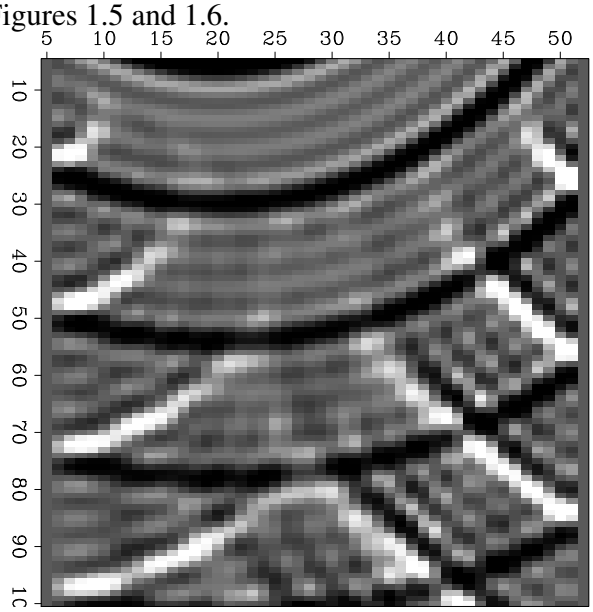


Figure 1.6: Modify the program so that zero-slope side boundaries are replaced by zero-value side boundaries. (Movie) fdm-Mzeroslope90 [ER,M]

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.

1.4.5 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 (1.50). If the wavefield is such that k_x is small enough, then equation (1.50) is the only place it is needed. Second, it enters in the **tridiagonal** coefficients through the v in equation (1.46). The so-called thin-lens approximation of optics seems to amount to including the equation (1.50) part only. An example of **lateral velocity variation** is in Figure 1.7.

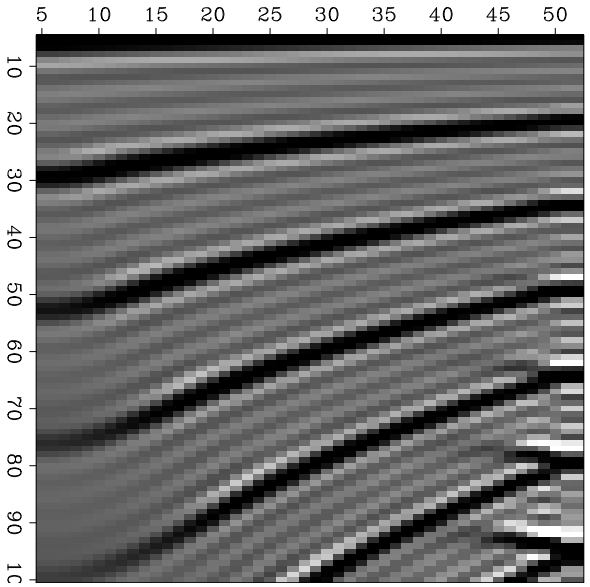


Figure 1.7: Make changes to the program to include a thin-**lens term** with a lateral velocity change of 40% across the frame produced by a constant slowness gradient. Identify other parts of the program which are affected by lateral velocity variation. You need not make these other changes. Why are they expected to be small? (Movie) fdm-Mlateralvel90 [ER,M]

1.4.6 Migration in (omega,x)-space

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 (1.43). 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 traveltime 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

$$\frac{\partial}{\partial z} \quad = \quad \frac{\partial \tau}{\partial z}\,\frac{\partial}{\partial \tau} \quad = \quad \frac{1}{v}\,\frac{\partial}{\partial \tau} \tag{1.51}$$

Substitution into (1.43) gives

$$\frac{\partial P}{\partial \tau} \quad = \quad -i\,\omega\,P \;-\; \frac{v^2}{-i\,\omega 2}\,\frac{\partial^2 P}{\partial x^2} \tag{1.52}$$

In the program, the time sample size `dt` $= \Delta t$ and the traveltime 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 negative frequency axis. The positive frequencies serve only to keep the time function real, a task that is more quickly done by simply taking the real part. A program listing follows

```
#%  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 srite( 'out', p, nx*nz*4 )
call hclose()
return; end
```

The output of the program is shown in Figure 1.8. 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 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.

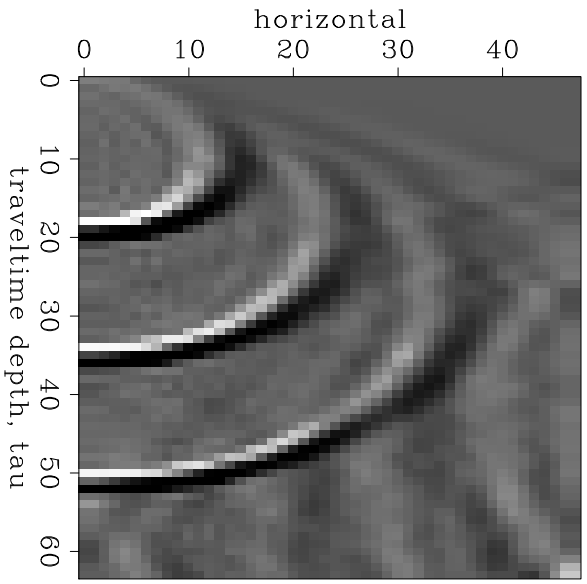


Figure 1.8: Output of the program `kjartjac`: semicircle approximations. fdm-kjartjac [ER]

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.

1.5 HIGHER ANGLE ACCURACY

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

$$P(z+\Delta z) \quad = \quad P(z) + \Delta z \, \frac{dP}{dz} \tag{1.53}$$

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.

1.5.1 Another way to the parabolic wave equation

Here we review the historic "transformation method" of deriving the parabolic wave equation.

A vertically downgoing plane wave is represented mathematically by the equation

$$P(t,x,z) \quad = \quad P_0 \; e^{-i\omega(t-z/v)} \tag{1.54}$$

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.

$$P(t,x,z) \quad = \quad Q(x,z) \; e^{-i\omega(t-z/v)} \tag{1.55}$$

Inserting (1.55) into the scalar wave equation $P_{xx} + P_{zz} = P_{tt}/v^2$ yields

$$\begin{aligned} \frac{\partial^2}{\partial x^2} \; Q \; + \; \left(\frac{i\omega}{v} + \frac{\partial}{\partial z} \right)^2 Q \quad &= \quad - \frac{\omega^2}{v^2} \; Q \\ \frac{\partial^2 Q}{\partial x^2} \; + \; \frac{2i\omega}{v} \; \frac{\partial Q}{\partial z} \; + \; \frac{\partial^2 Q}{\partial z^2} \quad &= \quad 0 \end{aligned} \tag{1.56}$$

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***

$$\frac{\partial Q}{\partial z} \quad = \quad \frac{v}{-2i\omega} \; \frac{\partial^2 Q}{\partial x^2} \tag{1.57}$$

I called equation (1.57) 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 (1.57) with respect to z and substitute the result back into equation (1.56) getting

$$\frac{\partial^2 Q}{\partial x^2} \; + \; \frac{2i\omega}{v} \; \frac{\partial Q}{\partial z} \; + \; \frac{v}{-2i\omega} \; \frac{\partial^3 Q}{\partial z \partial x^2} \quad = \quad 0 \tag{1.58}$$

I named equation (1.58) 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 (1.53).

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.

1.5.2 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. Recall equation (1.1)

$$k_z = \frac{\omega}{v}\sqrt{1 - \frac{v^2 k_x^2}{\omega^2}} \tag{1.59}$$

To inverse transform the z-axis we only need to recognize that ik_z corresponds to $\partial/\partial z$. Getting into the x-domain, however, 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 it is regarded as some kind of truncated series expansion. It is shown in IEI 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

$$X = \frac{vk_x}{\omega} \quad \text{and} \quad R = \frac{vk_z}{\omega} \tag{1.60}$$

With the definitions (1.60) equation (1.59) is more compactly written as

$$R = \sqrt{1 - X^2} \tag{1.61}$$

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

$$R_{n+1} = 1 - \frac{X^2}{1 + R_n} \tag{1.62}$$

The recurrence is a guess that we verify by seeing what it converges to (if it converges). Set $n = \infty$ in (1.62) and solve

$$\begin{aligned} R_\infty &= 1 - \frac{X^2}{1 + R_\infty} \\ R_\infty(1 + R_\infty) &= 1 + R_\infty - X^2 \\ R^2 &= 1 - X^2 \end{aligned} \tag{1.63}$$

The square root of (1.63) gives the required expression (1.61). Geometrically, (1.63) 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 recurrence 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. The first four truncations of **Muir**'s continued fraction expansion beginning from $R_0 = 1$ are

$$\begin{aligned} 5^\circ: \quad R_0 &= 1 \\ 15^\circ: \quad R_1 &= 1 - \frac{X^2}{2} \\ 45^\circ: \quad R_2 &= 1 - \frac{X^2}{2 - \frac{X^2}{2}} \end{aligned} \tag{1.64}$$

$$60^\circ: \qquad R_3 \quad = \quad 1 - \cfrac{X^2}{2 - \cfrac{X^2}{2 - \cfrac{X^2}{2}}}$$

For various historical reasons, the equations in the above equations 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 $R_0 = \cos 45^\circ$. Figure 1.9 shows some plots.

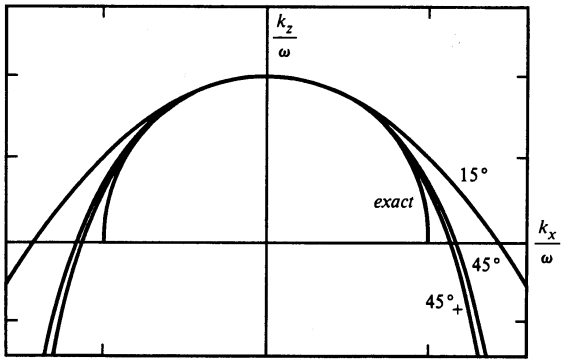


Figure 1.9: Dispersion relation of equation (1.65). The curve labeled 45°_+ was constructed with $R_0 = \cos 45^\circ$. It fits exactly at 0° and 45°. fdm-disper [NR]

1.5.3 Dispersion relations

Substituting the definitions (1.60) into equation (1.65) et. seq. gives dispersion relationships for comparison to the exact expression (1.59).

$$\begin{aligned} 5^\circ: \qquad k_z &= \frac{\omega}{v} \\ 15^\circ: \qquad k_z &= \frac{\omega}{v} - \frac{v k_x^2}{2\omega} \\ 45^\circ: \qquad k_z &= \frac{\omega}{v} - \cfrac{k_x^2}{2\,\cfrac{\omega}{v} - \cfrac{v k_x^2}{2\omega}} \end{aligned} \tag{1.65}$$

Identification of ik_z with $\partial/\partial z$ converts the dispersion relations (1.65) into the differential equations

$$5^\circ: \qquad \frac{\partial P}{\partial z} \quad = \quad i\left(\frac{\omega}{v}\right) P \tag{1.66}$$

$$15^\circ: \qquad \frac{\partial P}{\partial z} = i\left(\frac{\omega}{v} - \frac{vk_x^2}{2\omega}\right) P$$

$$45^\circ: \qquad \frac{\partial P}{\partial z} = i\left(\frac{\omega}{v} - \frac{k_x^2}{2\dfrac{\omega}{v} - \dfrac{vk_x^2}{2\omega}}\right) P$$

which are extrapolation equations for when velocity depends only on depth.

The differential equations above in Table 1.4 were based on a dispersion relation that in turn was based on an assumption of constant velocity. Surprisingly, these equations also have validity and great utility when the velocity is depth-variable, $v = v(z)$. The limitation is that the velocity be constant over each depth "slab" of width Δz over which the downward-continuation is carried out.

1.5.4 The xxz derivative

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

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

−1	2	−1
1	−2	1

So other than modifying the six coefficients on the star, it adds nothing to the computational cost. Using this extra term allows in programs like subroutine `wavemovie()` on page 17 yields wider angles.

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 1.11.

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.

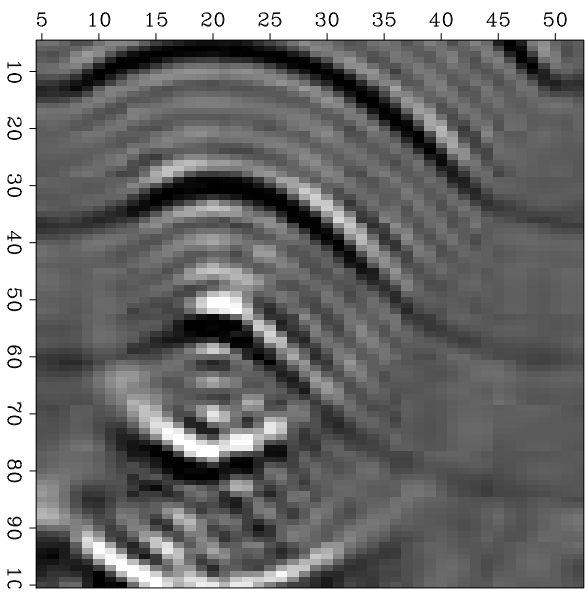


Figure 1.10: Figure 1.2 including the 45° term, ∂_{xxz}, for the collapsing spherical wave. What changes must be made to subroutine `wavemovie()` to get this result? Mark an X at the theoretical focus location. fdm-Mfortyfive90 [ER,M]

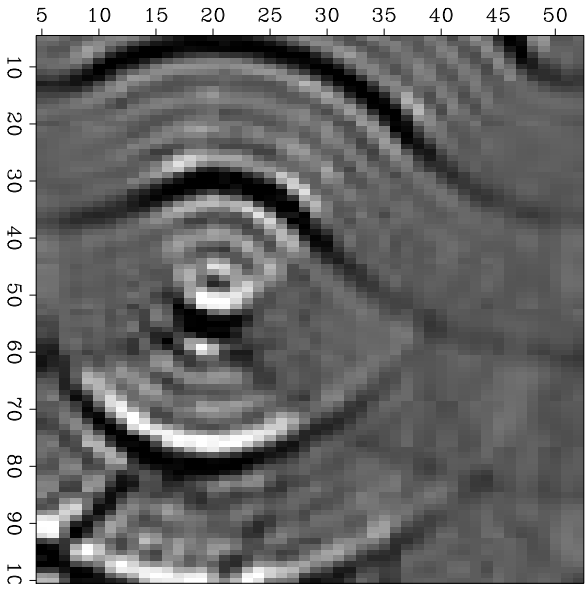


Figure 1.11: The accuracy of the x-derivative may be improved by a technique that is analyzed in IEI p 262-265. Briefly, instead of representing $k_x^2 \Delta x^2$ by the **tridiagonal** matrix $\mathbf{T}$ with $(-1, 2, -1)$ on the main diagonal, you use $\mathbf{T}/(\mathbf{I} - \mathbf{T}/6)$. Modify 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. fdm-Mhi45b90 [ER,M]

Index

45 degree phase angle, 27

full separation, 3

boundary, 6
boundary condition, 6, 13, 14

Crank-Nicolson method, 12, 18

explicit method, 9, 10, 13

finite difference, 1
full separation, 3, 4, 6, 7

Hankel function, 16
heat-flow equation, 3, 9–12

implicit method, 9, 12, 13

lateral velocity variation, 1, 2, 5, 21
leapfrog method, 12
lens equation, 7
lens term, 5, 21

movie, 15, 17, 21
Muir, 24, 25

parabolic wave equation, 1, 5, 24

rtris subroutine, 14

splitting, 3, 4, 6, 7, 19
square root, 25
subroutine
    `rtris`, real tridiagonal solver, 14
    `wavemovie`, 2-D wave movie, 17

tridiagonal, 13, 19, 21, 28

velocity
    laterally variable, 1, 2, 5, 21

wavemovie subroutine, 17