AIMS Lecture Notes 2006

Peter J. Olver

11. Numerical Solution of the Heat and Wave Equations

In this part, we study numerical solution methodss for the two most important equations of one-dimensional continuum dynamics. The *heat equation* models the diffusion of thermal energy in a body; here, we treat the case of a one-dimensional bar. The *wave equation* describes vibrations and waves in continuous media, including sound waves, water waves, elastic waves, electromagnetic waves, and so on. For simplicity, we restrict our attention to the case of waves in a one-dimensional medium, e.g., a string, bar, or column of air.

We begin with a general discussion of finite difference formulae for numerically approximating derivatives of functions. The basic *finite difference scheme* is obtained by replacing the derivatives in the equation by the appropriate numerical differentiation formulae. However, there is no guarantee that the resulting numerical scheme will accurately approximate the true solution, and further analysis is required to elicit bona fide, convergent numerical algorithms. In dynamical problems, the finite difference schemes replace the partial differential equation by an iterative linear matrix system, and the analysis of convergence relies on the methods covered in Section 7.1.

We will only introduce the most basic algorithms, leaving more sophisticated variations and extensions to a more thorough treatment, which can be found in numerical analysis texts, e.g., [**5**, **7**, **28**].

11.1. Finite Differences.

In general, to approximate the derivative of a function at a point, say $f'(x)$ or $f''(x)$, one constructs a suitable combination of sampled function values at nearby points. The underlying formalism used to construct these approximation formulae is known as the *calculus of finite differences*. Its development has a long and influential history, dating back to Newton. The resulting *finite difference numerical methods* for solving differential equations have extremely broad applicability, and can, with proper care, be adapted to most problems that arise in mathematics and its many applications.

The simplest finite difference approximation is the ordinary *difference quotient*

$$\frac{u(x+h)-u(x)}{h} \approx u'(x), \tag{11.1}$$

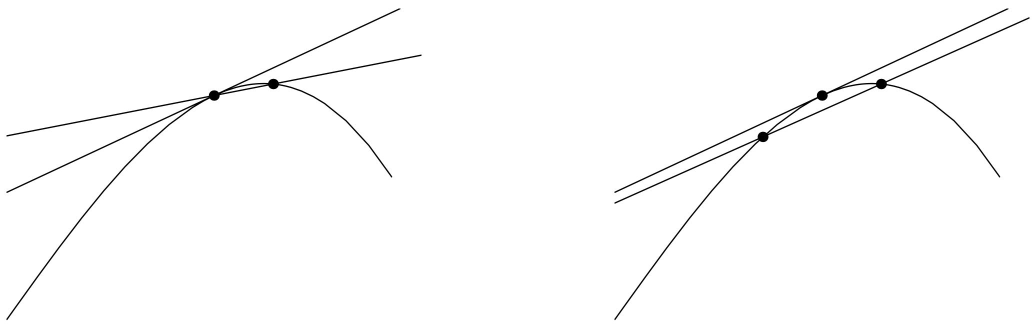

Figure 11.1. Finite Difference Approximations.

used to approximate the first derivative of the function $u(x)$. Indeed, if u is differentiable at x, then $u'(x)$ is, by definition, the limit, as $h \to 0$ of the finite difference quotients. Geometrically, the difference quotient equals the slope of the secant line through the two points $\big(x, u(x)\big)$ and $\big(x+h, u(x+h)\big)$ on the graph of the function. For small h, this should be a reasonably good approximation to the slope of the tangent line, $u'(x)$, as illustrated in the first picture in Figure 11.1.

How close an approximation is the difference quotient? To answer this question, we assume that $u(x)$ is at least twice continuously differentiable, and examine the first order Taylor expansion

$$u(x+h) = u(x) + u'(x)\,h + \tfrac{1}{2}\,u''(\xi)\,h^2. \tag{11.2}$$

We have used the Cauchy formula for the remainder term, in which ξ represents some point lying between x and $x+h$. The *error* or difference between the finite difference formula and the derivative being approximated is given by

$$\frac{u(x+h)-u(x)}{h} \;-\; u'(x) = \tfrac{1}{2}\,u''(\xi)\,h. \tag{11.3}$$

Since the error is proportional to h, we say that the finite difference quotient (11.3) is a *first order* approximation. When the precise formula for the error is not so important, we will write

$$u'(x) = \frac{u(x+h)-u(x)}{h} + \mathrm{O}(h). \tag{11.4}$$

The "big Oh" notation $\mathrm{O}(h)$ refers to a term that is proportional to h, or, more rigorously, bounded by a constant multiple of h as $h \to 0$.

Example 11.1. Let $u(x) = \sin x$. Let us try to approximate $u'(1) = \cos 1 = 0.5403023\ldots$ by computing finite difference quotients

$$\cos 1 \;\approx\; \frac{\sin(1+h) - \sin 1}{h}\,.$$

The result for different values of h is listed in the following table.

h	1	.1	.01	.001	.0001
approximation	0.067826	0.497364	0.536086	0.539881	0.540260
error	-0.472476	-0.042939	-0.004216	-0.000421	-0.000042

We observe that reducing the step size by a factor of $\frac{1}{10}$ reduces the size of the error by approximately the same factor. Thus, to obtain 10 decimal digits of accuracy, we anticipate needing a step size of about $h = 10^{-11}$. The fact that the error is more of less proportional to the step size confirms that we are dealing with a first order numerical approximation.

To approximate higher order derivatives, we need to evaluate the function at more than two points. In general, an approximation to the n^{th} order derivative $u^{(n)}(x)$ requires at least $n+1$ distinct sample points. For simplicity, we shall only use equally spaced points, leaving the general case to the exercises.

For example, let us try to approximate $u''(x)$ by sampling u at the particular points x, $x+h$ and $x-h$. Which combination of the function values $u(x-h), u(x), u(x+h)$ should be used? The answer to such a question can be found by consideration of the relevant Taylor expansions

$$\begin{aligned} u(x+h) &= u(x) + u'(x)\,h + u''(x)\,\frac{h^2}{2} + u'''(x)\,\frac{h^3}{6} + \mathrm{O}(h^4), \\ u(x-h) &= u(x) - u'(x)\,h + u''(x)\,\frac{h^2}{2} - u'''(x)\,\frac{h^3}{6} + \mathrm{O}(h^4), \end{aligned} \tag{11.5}$$

where the error terms are proportional to h^4. Adding the two formulae together gives

$$u(x+h) + u(x-h) = 2\,u(x) + u''(x)\,h^2 + \mathrm{O}(h^4).$$

Rearranging terms, we conclude that

$$u''(x) = \frac{u(x+h) - 2\,u(x) + u(x-h)}{h^2} + \mathrm{O}(h^2), \tag{11.6}$$

The result is known as the *centered finite difference approximation* to the second derivative of a function. Since the error is proportional to h^2, this is a second order approximation.

Example 11.2. Let $u(x) = e^{x^2}$, with $u''(x) = (4x^2+2)\,e^{x^2}$. Let us approximate $u''(1) = 6\,e = 16.30969097\ \ldots$ by using the finite difference quotient (11.6):

$$6\,e \approx \frac{e^{(1+h)^2} - 2\,e + e^{(1-h)^2}}{h^2}.$$

The results are listed in the following table.

h	1	.1	.01	.001	.0001
approximation	50.16158638	16.48289823	16.31141265	16.30970819	16.30969115
error	33.85189541	0.17320726	0.00172168	0.00001722	0.00000018

Each reduction in step size by a factor of $\frac{1}{10}$ reduces the size of the error by a factor of $\frac{1}{100}$ and results in a gain of two new decimal digits of accuracy, confirming that the finite difference approximation is of second order.

However, this prediction is not completely borne out in practice. If we take[†] $h = .00001$ then the formula produces the approximation 16.3097002570, with an error of 0.0000092863 — which is *less* accurate that the approximation with $h = .0001$. The problem is that round-off errors have now begun to affect the computation, and underscores the difficulty with numerical differentiation. Finite difference formulae involve dividing very small quantities, which can induce large numerical errors due to round-off. As a result, while they typically produce reasonably good approximations to the derivatives for moderately small step sizes, to achieve high accuracy, one must switch to a higher precision. In fact, a similar comment applied to the previous Example 11.1, and our expectations about the error were not, in fact, fully justified as you may have discovered if you tried an extremely small step size.

Another way to improve the order of accuracy of finite difference approximations is to employ more sample points. For instance, if the first order approximation (11.4) to the first derivative based on the two points x and $x+h$ is not sufficiently accurate, one can try combining the function values at three points x, $x+h$ and $x-h$. To find the appropriate combination of $u(x-h), u(x), u(x+h)$, we return to the Taylor expansions (11.5). To solve for $u'(x)$, we subtract[†] the two formulae, and so

$$u(x+h) - u(x-h) = 2\,u'(x)\,h + u'''(x)\,\frac{h^3}{3} + \mathrm{O}(h^4).$$

Rearranging the terms, we are led to the well-known *centered difference formula*

$$u'(x) = \frac{u(x+h) - u(x-h)}{2\,h} + \mathrm{O}(h^2), \qquad (11.7)$$

which is a second order approximation to the first derivative. Geometrically, the centered difference quotient represents the slope of the secant line through the two points $\bigl(x-h, u(x-h)\bigr)$ and $\bigl(x+h, u(x+h)\bigr)$ on the graph of u centered symmetrically about the point x. Figure 11.1 illustrates the two approximations; the advantages in accuracy in the centered difference version are graphically evident. Higher order approximations can be found by evaluating the function at yet more sample points, including, say, $x+2\,h, x-2\,h$, etc.

Example 11.3. Return to the function $u(x) = \sin x$ considered in Example 11.1. The centered difference approximation to its derivative $u'(1) = \cos 1 = 0.5403023\ \ldots$ is

$$\cos 1 \;\approx\; \frac{\sin(1+h) - \sin(1-h)}{2\,h}\,.$$

The results are tabulated as follows:

[†] This next computation depends upon the computer's precision; here we used single precision in MATLAB.

[†] The terms $\mathrm{O}(h^4)$ do *not* cancel, since they represent potentially different multiples of h^4.

h	.1	.01	.001	.0001
approximation	0.53940225217	0.54029330087	0.54030221582	0.54030230497
error	-0.00090005370	-0.00000900499	-0.00000009005	-0.00000000090

As advertised, the results are much more accurate than the one-sided finite difference approximation used in Example 11.1 at the same step size. Since it is a second order approximation, each reduction in the step size by a factor of $\frac{1}{10}$ results in two more decimal places of accuracy.

Many additional finite difference approximations can be constructed by similar manipulations of Taylor expansions, but these few very basic ones will suffice for our subsequent purposes. In the following subsection, we apply the finite difference formulae to develop numerical solution schemes for the heat and wave equations.

11.2. Numerical Algorithms for the Heat Equation.

Consider the heat equation

$$\frac{\partial u}{\partial t} = \gamma \, \frac{\partial^2 u}{\partial x^2} \, , \qquad 0 < x < \ell, \qquad t \geq 0, \tag{11.8}$$

representing a homogeneous diffusion process of, sqy, heat in bar of length ℓ and constant thermal diffusivity $\gamma > 0$. The solution $u(t,x)$ represents the temperature in the bar at time $t \geq 0$ and position $0 \leq x \leq \ell$. To be concrete, we will impose time-dependent Dirichlet boundary conditions

$$u(t,0) = \alpha(t), \qquad u(t,\ell) = \beta(t), \qquad t \geq 0, \tag{11.9}$$

specifying the temperature at the ends of the bar, along with the initial conditions

$$u(0,x) = f(x), \qquad 0 \leq x \leq \ell, \tag{11.10}$$

specifying the bar's initial temperature distribution. In order to effect a numerical approximation to the solution to this initial-boundary value problem, we begin by introducing a *rectangular mesh* consisting of points (t_i, x_j) with

$$0 = x_0 < x_1 < \cdots < x_n = \ell \qquad \text{and} \qquad 0 = t_0 < t_1 < t_2 < \cdots .$$

For simplicity, we maintain a uniform mesh spacing in both directions, with

$$h = x_{j+1} - x_j = \frac{\ell}{n} \, , \qquad k = t_{i+1} - t_i,$$

representing, respectively, the spatial mesh size and the time step size. It will be essential that we do *not* a priori require the two to be the same. We shall use the notation

$$u_{i,j} \approx u(t_i, x_j) \qquad \text{where} \qquad t_i = i\,k, \qquad x_j = j\,h, \tag{11.11}$$

to denote the numerical approximation to the solution value at the indicated mesh point.

As a first attempt at designing a numerical method, we shall use the simplest finite difference approximations to the derivatives. The second order space derivative is approximated by (11.6), and hence

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}(t_i, x_j) &\approx \frac{u(t_i, x_{j+1}) - 2\,u(t_i, x_j) + u(t_i, x_{j-1})}{h^2} + \mathrm{O}(h^2) \\ &\approx \frac{u_{i,j+1} - 2\,u_{i,j} + u_{i,j-1}}{h^2} + \mathrm{O}(h^2),\end{aligned} \tag{11.12}$$

where the error in the approximation is proportional to h^2. Similarly, the one-sided finite difference approximation (11.4) is used for the time derivative, and so

$$\frac{\partial u}{\partial t}(t_i, x_j) \approx \frac{u(t_{i+1}, x_j) - u(t_i, x_j)}{k} + \mathrm{O}(k) \approx \frac{u_{i+1,j} - u_{i,j}}{k} + \mathrm{O}(k), \tag{11.13}$$

where the error is proportion to k. In practice, one should try to ensure that the approximations have similar orders of accuracy, which leads us to choose

$$k \approx h^2.$$

Assuming $h < 1$, this requirement has the important consequence that the time steps must be *much* smaller than the space mesh size.

Remark: At this stage, the reader might be tempted to replace (11.13) by the second order central difference approximation (11.7). However, this produces significant complications, and the resulting numerical scheme is not practical.

Replacing the derivatives in the heat equation (11.14) by their finite difference approximations (11.12), (11.13), and rearranging terms, we end up with the linear system

$$u_{i+1,j} = \mu\, u_{i,j+1} + (1 - 2\,\mu) u_{i,j} + \mu\, u_{i,j-1}, \qquad \begin{array}{l} i = 0, 1, 2, \ldots\,, \\ j = 1, \ldots, n-1, \end{array} \tag{11.14}$$

in which

$$\mu = \frac{\gamma\, k}{h^2}\,. \tag{11.15}$$

The resulting numerical scheme takes the form of an iterative linear system for the solution values $u_{i,j} \approx u(t_i, x_j)$, $j = 1, \ldots, n-1$, at each time step t_i.

The initial condition (11.10) means that we should initialize our numerical data by sampling the initial temperature at the mesh points:

$$u_{0,j} = f_j = f(x_j), \qquad j = 1, \ldots, n-1. \tag{11.16}$$

Similarly, the boundary conditions (11.9) require that

$$u_{i,0} = \alpha_i = \alpha(t_i), \qquad u_{i,n} = \beta_i = \beta(t_i), \qquad i = 0, 1, 2, \ldots\,. \tag{11.17}$$

For consistency, we should assume that the initial and boundary conditions agree at the corners of the domain:

$$f_0 = f(0) = u(0,0) = \alpha(0) = \alpha_0, \qquad f_n = f(\ell) = u(0,\ell) = \beta(0) = \beta_0.$$

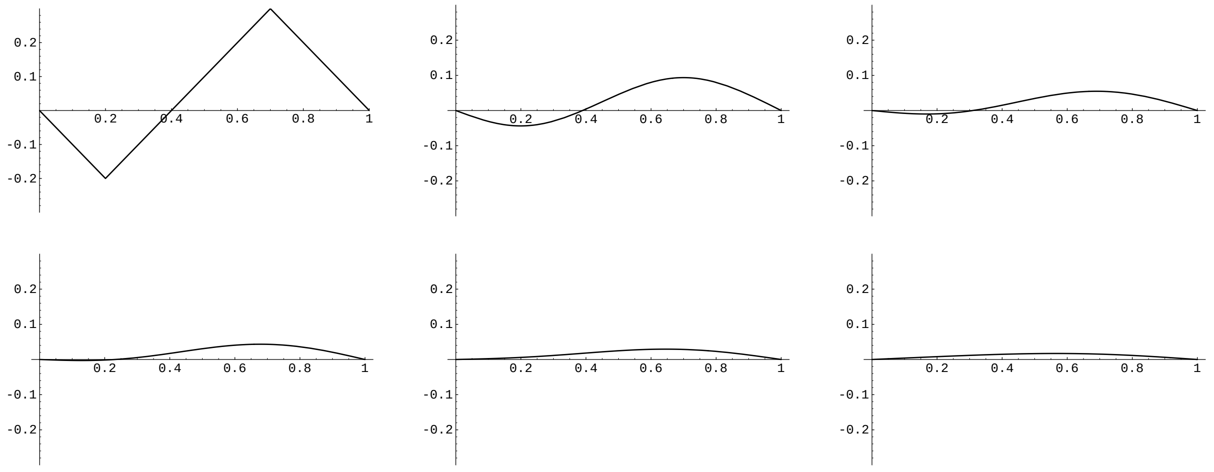

Figure 11.2. A Solution to the Heat Equation.

The three equations (11.14–17) completely prescribe the numerical approximation algorithm for solving the initial-boundary value problem (11.8–10).

Let us rewrite the scheme in a more transparent matrix form. First, let

$$\mathbf{u}^{(i)} = \big(u_{i,1}, u_{i,2}, \ldots, u_{i,n-1} \big)^T \approx \big(u(t_i, x_1), u(t_i, x_2), \ldots, u(t_i, x_{n-1}) \big)^T \tag{11.18}$$

be the vector whose entries are the numerical approximations to the solution values at time t_i at the *interior* nodes. We omit the boundary nodes $x_0 = 0$, $x_n = \ell$, since those values are fixed by the boundary conditions (11.9). Then (11.14) assumes the compact vectorial form

$$\mathbf{u}^{(i+1)} = A\,\mathbf{u}^{(i)} + \mathbf{b}^{(i)}, \tag{11.19}$$

where

$$A = \begin{pmatrix} 1-2\mu & \mu & & & & \\ \mu & 1-2\mu & \mu & & & \\ & \mu & 1-2\mu & \mu & & \\ & & \mu & \ddots & \ddots & \\ & & & \ddots & \ddots & \mu \\ & & & & \mu & 1-2\mu \end{pmatrix}, \qquad \mathbf{b}^{(i)} = \begin{pmatrix} \mu\,\alpha_i \\ 0 \\ 0 \\ \vdots \\ 0 \\ \mu\,\beta_i \end{pmatrix}. \tag{11.20}$$

The coefficient matrix A is symmetric and tridiagonal. The contributions (11.17) of the boundary nodes appear in the vector $\mathbf{b}^{(i)}$. This numerical method is known as an *explicit scheme* since each iterate is computed directly without relying on solving an auxiliary equation — unlike the implicit schemes to be discussed below.

Example 11.4. Let us fix the diffusivity $\gamma = 1$ and the bar length $\ell = 1$. Consider the initial temperature profile

$$u(0,x) = f(x) = \begin{cases} -x, & 0 \le x \le \frac{1}{5}, \\ x - \frac{2}{5}, & \frac{1}{5} \le x \le \frac{7}{10}, \\ 1 - x, & \frac{7}{10} \le x \le 1, \end{cases} \tag{11.21}$$

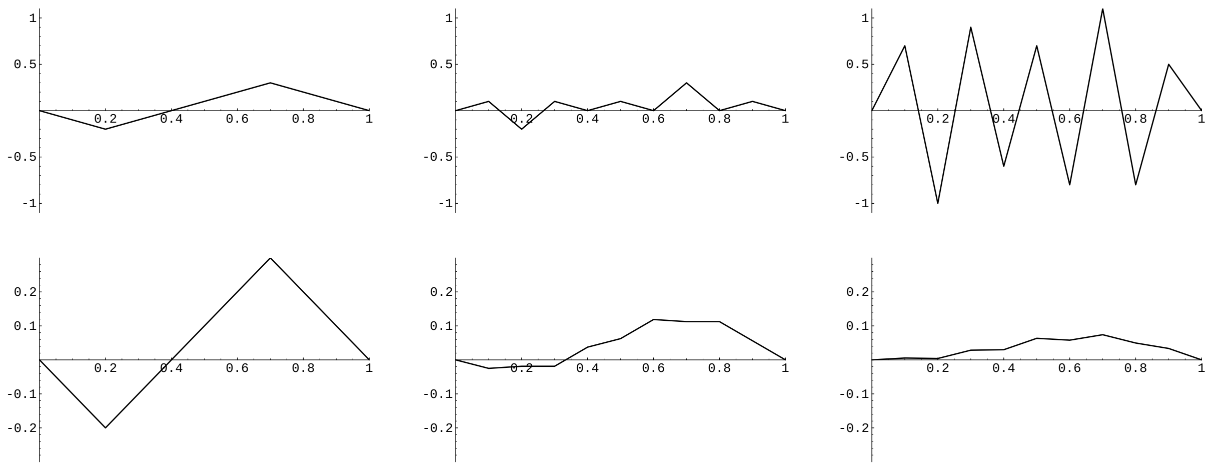

Figure 11.3. Numerical Solutions for the Heat Equation Based on the Explicit Scheme.

on a bar of length 1, plotted in the first graph in Figure 11.2. The solution is plotted at the successive times $t = ., .02, .04, \ldots, .1$. Observe that the corners in the initial data are immediately smoothed out. As time progresses, the solution decays, at an exponential rate of $\pi^2 \approx 9.87$, to a uniform, zero temperature, which is the equilibrium temperature distribution for the homogeneous Dirichlet boundary conditions. As the solution decays to thermal equilibrium, it also assumes the progressively more symmetric shape of a single sine arc, of exponentially decreasing amplitude.

In our numerical solution, we take the spatial step size $h = .1$. In Figure 11.3 we compare two (slightly) different time step sizes on the same initial data as used in (11.21). The first sequence uses the time step $k = h^2 = .01$ and plots the solution at times $t = 0., .02, .04$. The solution is already starting to show signs of instability, and indeed soon thereafter becomes completely wild. The second sequence takes $k = .005$ and plots the solution at times $t = 0., .025, .05$. (Note that the two sequences of plots have different vertical scales.) Even though we are employing a rather coarse mesh, the numerical solution is not too far away from the true solution to the initial value problem, which can be found in Figure 11.2.

In light of this calculation, we need to understand why our scheme sometimes gives reasonable answers but at other times utterly fails. To this end, let us specialize to homogeneous boundary conditions

$$u(t,0) = 0 = u(t,\ell), \qquad \text{whereby} \qquad \alpha_i = \beta_i = 0 \qquad \text{for all} \qquad i = 0,1,2,3,\ldots, \tag{11.22}$$

and so (11.19) reduces to a homogeneous, linear iterative system

$$\mathbf{u}^{(i+1)} = A\,\mathbf{u}^{(i)}. \tag{11.23}$$

According to Proposition 7.8, all solutions will converge to zero, $\mathbf{u}^{(i)} \to \mathbf{0}$ — as they are supposed to (why?) — if and only if A is a convergent matrix. But convergence depends upon the step sizes. Example 11.4 is indicating that for mesh size $h = .1$, the time step

$k = .01$ yields a non-convergent matrix, while $k = .005$ leads to a convergent matrix and a valid numerical scheme.

As we learned in Theorem 7.11, the convergence property of a matrix is fixed by its spectral radius, i.e., its largest eigenvalue in magnitude. There is, in fact, an explicit formula for the eigenvalues of the particular tridiagonal matrix in our numerical scheme, which follows from the following general result.

Lemma 11.5. *The eigenvalues of an $(n-1)\times(n-1)$ tridiagonal matrix all of whose diagonal entries are equal to a and all of whose sub- and super-diagonal entries are equal to b are*

$$\lambda_k = a + 2b\cos\frac{\pi k}{n}, \qquad k = 1, \ldots, n-1. \tag{11.24}$$

Proof: The corresponding eigenvectors are

$$\mathbf{v}_k = \left(\sin\frac{k\pi}{n}, \ \sin\frac{2k\pi}{n}, \ \ldots \ \sin\frac{nk\pi}{n} \right)^T.$$

Indeed, the j^{th} entry of the eigenvalue equation $A\mathbf{v}_k = \lambda_k \mathbf{v}_k$ reads

$$a\sin\frac{jk\pi}{n} + b\left(\sin\frac{(j-1)k\pi}{n} + \sin\frac{(j+1)k\pi}{n}\right) = \left(a + 2b\cos\frac{k\pi}{n}\right)\sin\frac{jk\pi}{n},$$

which follows from the trigonometric identity

$$\sin\alpha + \sin\beta = 2\cos\frac{\alpha-\beta}{2}\sin\frac{\alpha+\beta}{2}. \qquad \textit{Q.E.D.}$$

In our particular case, $a = 1 - 2\mu$ and $b = \mu$, and hence the eigenvalues of the matrix A given by (11.20) are

$$\lambda_k = 1 - 2\mu + 2\mu\cos\frac{\pi k}{n}, \qquad k = 1, \ldots, n-1.$$

Since the cosine term ranges between -1 and $+1$, the eigenvalues satisfy

$$1 - 4\mu < \lambda_k < 1.$$

Thus, assuming that $0 < \mu \leq \frac{1}{2}$ guarantees that all $|\lambda_k| < 1$, and hence A is a convergent matrix. In this way, we have deduced the basic stability criterion

$$\mu = \frac{\gamma k}{h^2} \leq \frac{1}{2}, \qquad \text{or} \qquad k \leq \frac{h^2}{2\gamma}. \tag{11.25}$$

With some additional analytical work, [**28**], it can be shown that this is sufficient to conclude that the numerical scheme (11.14–17) converges to the true solution to the initial-boundary value problem for the heat equation.

Since not all choices of space and time steps lead to a convergent scheme, the numerical method is called *conditionally stable*. The convergence criterion (11.25) places a severe restriction on the time step size. For instance, if we have $h = .01$, and $\gamma = 1$, then we can only use a time step size $k \leq .00005$, which is minuscule. It would take an inordinately large number of time steps to compute the value of the solution at even a moderate times, e.g., $t = 1$. Moreover, owing to the limited accuracy of computers, the propagation of round-off errors might then cause a significant reduction in the overall accuracy of the final solution values.

An unconditionally stable method — one that does not restrict the time step — can be constructed by using the backwards difference formula

$$\frac{\partial u}{\partial t}\,(t_i, x_j) \approx \frac{u(t_i, x_j) - u(t_{i-1}, x_j)}{k} + \mathrm{O}(h^k) \tag{11.26}$$

to approximate the temporal derivative. Substituting (11.26) and the same approximation (11.12) for u_{xx} into the heat equation, and then replacing i by $i+1$, leads to the iterative system

$$u_{i+1,j} - \mu\left(u_{i+1,j+1} - 2u_{i+1,j} + u_{i+1,j-1}\right) = u_{i,j}, \qquad \begin{aligned} &i = 0, 1, 2, \ldots\,, \\ &j = 1, \ldots, n-1, \end{aligned} \tag{11.27}$$

where the parameter $\mu = \gamma\, k/h^2$ is as above. The initial and boundary conditions also have the same form (11.16), (11.17). The system can be written in the matrix form

$$\widehat{A}\,\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \mathbf{b}^{(i+1)}, \tag{11.28}$$

where $\widehat{A}$ is obtained from the matrix A in (11.20) by replacing μ by $-\mu$. This defines an *implicit method* since we have to solve a tridiagonal linear system at each step in order to compute the next iterate $\mathbf{u}^{(i+1)}$. However, as we learned in Section 4.5, tridiagonal systems can be solved very rapidly, and so speed does not become a significant issue in the practical implementation of this implicit scheme.

Let us look at the convergence properties of the implicit scheme. For homogeneous Dirichlet boundary conditions (11.22), the system takes the form

$$\mathbf{u}^{(i+1)} = \widehat{A}^{-1}\,\mathbf{u}^{(i)},$$

and the convergence is now governed by the eigenvalues of $\widehat{A}^{-1}$. Lemma 11.5 tells us that the eigenvalues of $\widehat{A}$ are

$$\lambda_k = 1 + 2\mu - 2\mu\cos\frac{\pi k}{n}\,, \qquad\qquad k = 1, \ldots, n-1.$$

As a result, its inverse $\widehat{A}^{-1}$ has eigenvalues

$$\frac{1}{\lambda_k} = \frac{1}{1 + 2\mu\left(1 - \cos\dfrac{\pi k}{n}\right)}\,, \qquad\qquad k = 1, \ldots, n-1.$$

Since $\mu > 0$, the latter are *always* less than 1 in absolute value, and so $\widehat{A}$ is always a convergent matrix. The implicit scheme (11.28) is convergent for any choice of step sizes h, k, and hence *unconditionally stable*.

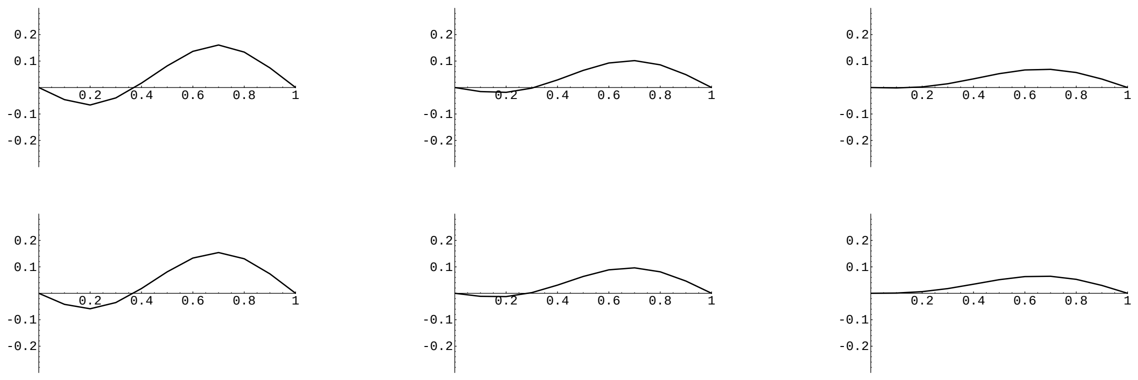

Figure 11.4. Numerical Solutions for the Heat Equation Based on the Implicit Scheme.

Example 11.6. Consider the same initial-boundary value problem considered in Example 11.4. In Figure 11.4, we plot the numerical solutions obtained using the implicit scheme. The initial data is not displayed, but we graph the numerical solutions at times $t = .2, .4, .6$ with a mesh size of $h = .1$. On the top line, we use a time step of $k = .01$, while on the bottom $k = .005$. Unlike the explicit scheme, there is very little difference between the two — both come much closer to the actual solution than the explicit scheme. Indeed, even significantly larger time steps give reasonable numerical approximations to the solution.

Another popular numerical scheme is the *Crank–Nicolson method*

$$u_{i+1,j} - u_{i,j} = \frac{\mu}{2}\left(u_{i+1,j+1} - 2\,u_{i+1,j} + u_{i+1,j-1} + u_{i,j+1} - 2\,u_{i,j} + u_{i,j-1}\right). \tag{11.29}$$

which can be obtained by averaging the explicit and implicit schemes (11.14, 27). We can write the iterative system in matrix form

$$B\,\mathbf{u}^{(i+1)} = C\,\mathbf{u}^{(i)} + \tfrac{1}{2}\left(\mathbf{b}^{(i)} + \mathbf{b}^{(i+1)}\right),$$

where

$$B = \begin{pmatrix} 1+\mu & -\frac{1}{2}\mu & & \\ -\frac{1}{2}\mu & 1+\mu & -\frac{1}{2}\mu & \\ & -\frac{1}{2}\mu & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}, \qquad C = \begin{pmatrix} 1-\mu & \frac{1}{2}\mu & & \\ \frac{1}{2}\mu & 1-\mu & \frac{1}{2}\mu & \\ & \frac{1}{2}\mu & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}. \tag{11.30}$$

Convergence is governed by the generalized eigenvalues of the tridiagonal matrix pair B, C, or, equivalently, the eigenvalues of the product $B^{-1}C$, which are

$$\lambda_k = \frac{1 - \mu\left(1 - \cos\dfrac{\pi k}{n}\right)}{1 + \mu\left(1 - \cos\dfrac{\pi k}{n}\right)}, \qquad k = 1, \ldots, n-1. \tag{11.31}$$

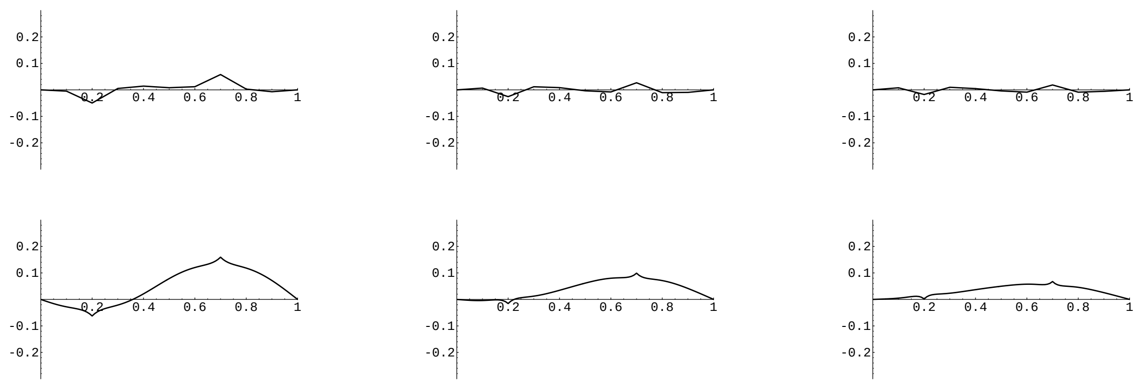

Figure 11.5. Numerical Solutions for the Heat Equation Based on the Crank–Nicolson Scheme.

Since $\mu > 0$, all of the eigenvalues are strictly less than 1 in absolute value, and so the Crank–Nicolson scheme is also unconditionally stable. A detailed analysis will show that the errors are of the order of k^2 and h^2, and so it is reasonable to choose the time step to have the same order of magnitude as the space step, $k \approx h$. This gives the Crank–Nicolson scheme one advantage over the previous two methods. However, applying it to the initial value problem considered earlier points out a significant weakness. Figure 11.5 shows the result of running the scheme on the initial data (11.21). The top row has space and time step sizes $h = k = .1$, and does a rather poor job replicating the solution. The second row uses $h = k = .01$, and performs better except near the corners where an annoying and incorrect local time oscillation persists as the solution decays. Indeed, since most of its eigenvalues are near -1, the Crank–Nicolson scheme does not do a good job of damping out the high frequency modes that arise from small scale features, including discontinuities and corners in the initial data. On the other hand, most of the eigenvalues of the fully implicit scheme are near zero, and it tends to handle the high frequency modes better, losing out to Crank–Nicolson when the data is smooth. Thus, a good strategy is to first evolve using the implicit scheme until the small scale noise is dissipated away, and then switch to Crank–Nicolson to use a much larger time step for final the large scale changes.

11.3. Numerical Solution Methods for the Wave Equation.

Let us now look at some numerical solution techniques for the wave equation. Although this is in a sense unnecessary, owing to the explicit d'Alembert solution formula, the experience we gain in designing workable schemes will serve us well in more complicated situations, including inhomogeneous media, and higher dimensional problems, when analytic solution formulas are no longer available.

Consider the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \, \frac{\partial^2 u}{\partial x^2} \,, \qquad 0 < x < \ell, \qquad t \geq 0, \tag{11.32}$$

modeling vibrations of a homogeneous bar of length ℓ with constant wave speed $c > 0$. We impose Dirichlet boundary conditions

$$u(t,0)=\alpha(t), \qquad u(t,\ell)=\beta(t), \qquad t\geq 0. \tag{11.33}$$

and initial conditions

$$u(0,x)=f(x), \qquad \frac{\partial u}{\partial t}(0,x)=g(x), \qquad 0\leq x\leq \ell. \tag{11.34}$$

We adopt the same uniformly spaced mesh

$$t_i=i\,k, \qquad x_j=j\,h, \qquad \text{where} \qquad h=\frac{\ell}{n}\,.$$

In order to discretize the wave equation, we replace the second order derivatives by their standard finite difference approximations (11.6), namely

$$\begin{aligned}\frac{\partial^2 u}{\partial t^2}(t_i,x_j) &\approx \frac{u(t_{i+1},x_j)-2\,u(t_i,x_j)+u(t_{i-1},x_j)}{k^2} \;+\; \mathrm{O}(h^2),\\ \frac{\partial^2 u}{\partial x^2}(t_i,x_j) &\approx \frac{u(t_i,x_{j+1})-2\,u(t_i,x_j)+u(t_i,x_{j-1})}{h^2} \;+\; \mathrm{O}(k^2),\end{aligned} \tag{11.35}$$

Since the errors are of orders of k^2 and h^2, we anticipate to be able to choose the space and time step sizes of comparable magnitude:

$$k\approx h.$$

Substituting the finite difference formulae (11.35) into the partial differential equation (11.32), and rearranging terms, we are led to the iterative system

$$u_{i+1,j}=\sigma^2\,u_{i,j+1}+2\,(1-\sigma^2)\,u_{i,j}+\sigma^2\,u_{i,j-1}-u_{i-1,j}, \qquad \begin{array}{l} i=1,2,\ldots\,, \\ j=1,\ldots,n-1, \end{array} \tag{11.36}$$

for the numerical approximations $u_{i,j}\approx u(t_i,x_j)$ to the solution values at the mesh points. The positive parameter

$$\sigma=\frac{c\,k}{h}\;>\;0 \tag{11.37}$$

depends upon the wave speed and the ratio of space and time step sizes. The boundary conditions (11.33) require that

$$u_{i,0}=\alpha_i=\alpha(t_i), \qquad u_{i,n}=\beta_i=\beta(t_i), \qquad i=0,1,2,\ldots\,. \tag{11.38}$$

This allows us to rewrite the system in matrix form

$$\mathbf{u}^{(i+1)}=B\,\mathbf{u}^{(i)}-\mathbf{u}^{(i-1)}+\mathbf{b}^{(i)}, \tag{11.39}$$

where

$$B=\begin{pmatrix} 2\,(1-\sigma^2) & \sigma^2 & & & \\ \sigma^2 & 2\,(1-\sigma^2) & \sigma^2 & & \\ & \sigma^2 & \ddots & \ddots & \\ & & \ddots & \ddots & \sigma^2 \\ & & & \sigma^2 & 2\,(1-\sigma^2) \end{pmatrix}, \quad \mathbf{u}^{(j)}=\begin{pmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{n-2,j} \\ u_{n-1,j} \end{pmatrix}, \quad \mathbf{b}^{(j)}=\begin{pmatrix} \sigma^2\,\alpha_j \\ 0 \\ \vdots \\ 0 \\ \sigma^2\,\beta_j \end{pmatrix}. \tag{11.40}$$

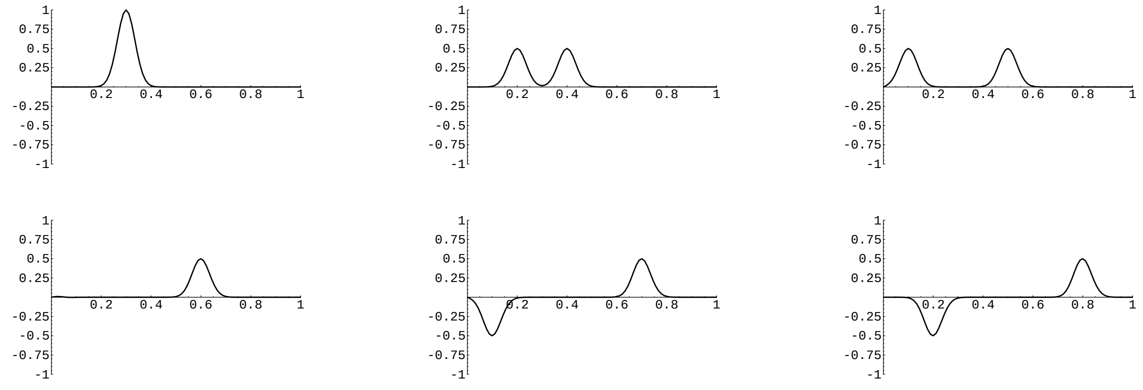

Figure 11.6. Numerically Stable Waves.

The entries of $\mathbf{u}^{(i)}$ are, as in (11.18), the numerical approximations to the solution values at the *interior* nodes. Note that the system (11.39) is a second order iterative scheme, since computing the next iterate $\mathbf{u}^{(i+1)}$ requires the value of the preceding two, $\mathbf{u}^{(i)}$ and $\mathbf{u}^{(i-1)}$.

The one difficulty is getting the method started. We know $\mathbf{u}^{(0)}$ since $u_{0,j} = f_j = f(x_j)$ is determined by the initial position. However, we also need to find $\mathbf{u}^{(1)}$ with entries $u_{1,j} \approx u(k, x_j)$ at time $t_1 = k$ in order launch the iteration, but the initial velocity $u_t(0,x) = g(x)$ prescribes the derivatives $u_t(0,x_j) = g_j = g(x_j)$ at time $t_0 = 0$ instead. One way to resolve this difficult would be to utilize the finite difference approximation

$$g_j = \frac{\partial u}{\partial t}(0,x_j) \approx \frac{u(k,x_j) - u(0,x_j)}{k} \approx \frac{u_{1,j} - g_j}{k} \qquad (11.41)$$

to compute the required values

$$u_{1,j} = f_j + k\, g_j.$$

However, the approximation (11.41) is only accurate to order k, whereas the rest of the scheme has error proportional to k^2. Therefore, we would introduce an unacceptably large error at the initial step.

To construct an initial approximation to $\mathbf{u}^{(1)}$ with error on the order of k^2, we need to analyze the local error in more detail. Note that, by Taylor's theorem,

$$\frac{u(k,x_j) - u(0,x_j)}{k} \approx \frac{\partial u}{\partial t}(0,x_j) + \frac{k}{2}\,\frac{\partial^2 u}{\partial t^2}(0,x_j) = \frac{\partial u}{\partial t}(0,x_j) + \frac{c^2\,k}{2}\;\frac{\partial^2 u}{\partial x^2}(0,x_j)\,,$$

where the error is now of order k^2, and, in the final equality, we have used the fact that u is a solution to the wave equation. Therefore, we find

$$\begin{aligned} u(k,x_j) &\approx u(0,x_j) + k\;\frac{\partial u}{\partial t}(0,x_j) + \frac{c^2\,k^2}{2}\;\frac{\partial^2 u}{\partial x^2}(0,x_j) \\ &= f(x_j) + k\,g(x_j) + \frac{c^2\,k^2}{2}\,f''(x_j) \approx f_j + k\,g_j + \frac{c^2\,k^2}{2\,h^2}(f_{j+1} - 2\,f_j + f_{j-1})\,, \end{aligned}$$

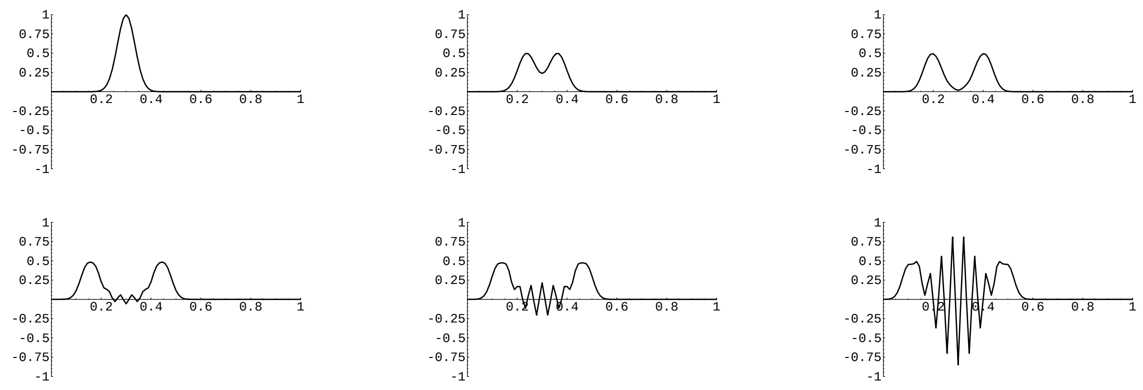

Figure 11.7. Numerically Unstable Waves.

where we can use the finite difference approximation (11.6) for the second derivative of $f(x)$ if no explicit formula is known. Therefore, when we initiate the scheme by setting

$$u_{1,j} = \tfrac{1}{2}\,\sigma^2\, f_{j+1} + (1-\sigma^2) f_j + \tfrac{1}{2}\,\sigma^2\, f_{j-1} + k\, g_j, \tag{11.42}$$

or, in matrix form,

$$\mathbf{u}^{(0)} = \mathbf{f}, \qquad\qquad \mathbf{u}^{(1)} = \tfrac{1}{2}\, B\,\mathbf{u}^{(0)} + k\,\mathbf{g} + \tfrac{1}{2}\,\mathbf{b}^{(0)}, \tag{11.43}$$

we will have maintained the desired order k^2 (and h^2) accuracy.

Example 11.7. Consider the particular initial value problem

$$u_{tt} = u_{xx}, \qquad \begin{matrix} u(0,x) = e^{-400\,(x-.3)^2}, \quad & u_t(0,x) = 0, \qquad & 0 \le x \le 1, \\ u(t,0) = u(1,0) = 0, & & t \ge 0, \end{matrix}$$

subject to homogeneous Dirichlet boundary conditions on the interval $[0,1]$. The initial data is a fairly concentrated single hump centered at $x = .3$, and we expect it to split into two half sized humps, which then collide with the ends. Let us choose a space discretization consisting of 90 equally spaced points, and so $h = \frac{1}{90} = .0111\ldots$. If we choose a time step of $k = .01$, whereby $\sigma = .9$, then we get reasonably accurate solution over a fairly long time range, as plotted in Figure 11.6 at times $t = 0, .1, .2, \ldots, .5$. On the other hand, if we double the time step, setting $k = .02$, so $\sigma = 1.8$, then, as plotted in Figure 11.7 at times $t = 0, .05, .1, .14, .16, .18$, we observe an instability eventually creeping into the picture that eventually overwhelms the numerical solution. Thus, the numerical scheme appears to only be conditionally stable.

The stability analysis of this numerical scheme proceeds as follows. We first need to recast the second order iterative system (11.39) into a first order system. In analogy with Example 7.4, this is accomplished by introducing the vector $\mathbf{z}^{(i)} = \begin{pmatrix} \mathbf{u}^{(i)} \\ \mathbf{u}^{(i-1)} \end{pmatrix} \in \mathbb{R}^{2n-2}$. Then

$$\mathbf{z}^{(i+1)} = C\,\mathbf{z}^{(i)} + \mathbf{c}^{(i)}, \qquad \text{where} \qquad C = \begin{pmatrix} B & -\mathrm{I} \\ \mathrm{I} & \mathrm{O} \end{pmatrix}. \tag{11.44}$$

Therefore, the stability of the method will be determined by the eigenvalues of the coefficient matrix C. The eigenvector equation $C\mathbf{z} = \lambda\,\mathbf{z}$, where $\mathbf{z} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}$, can be written out in its individual components:

$$B\mathbf{u} - \mathbf{v} = \lambda\,\mathbf{u}, \qquad \mathbf{u} = \lambda\,\mathbf{v}.$$

Substituting the second equation into the first, we find

$$(\lambda B - \lambda^2 - 1)\,\mathbf{v} = \mathbf{0}, \qquad \text{or} \qquad B\mathbf{v} = \left(\lambda + \frac{1}{\lambda}\right)\mathbf{v}.$$

The latter equation implies that $\mathbf{v}$ is an eigenvector of B with $\lambda + \lambda^{-1}$ the corresponding eigenvalue. The eigenvalues of the tridiagonal matrix B are governed by Lemma 11.5, in which $a = 2(1-\sigma^2)$ and $b = \sigma^2$, and hence are

$$\lambda + \frac{1}{\lambda} = 2\left(1 - \sigma^2 + \sigma^2 \cos\frac{\pi k}{n}\right), \qquad k = 1, \ldots, n-1.$$

Multiplying both sides by λ leads to a quadratic equation for the eigenvalues,

$$\lambda^2 - 2a_k\lambda + 1 = 0, \qquad \text{where} \qquad 1 - 2\sigma^2 < a_k = 1 - \sigma^2 + \sigma^2\cos\frac{\pi k}{n} < 1. \tag{11.45}$$

Each pair of solutions to these $n-1$ quadratic equations, namely

$$\lambda_k^{\pm} = a_k \pm \sqrt{a_k^2 - 1}\,, \tag{11.46}$$

yields two eigenvalues of the matrix C. If $a_k > 1$, then one of the two eigenvalues will be larger than one in magnitude, which means that the linear iterative system has an exponentially growing mode, and so $\|\,\mathbf{u}^{(i)}\,\| \to \infty$ as $i \to \infty$ for almost all choices of initial data. This is clearly incompatible with the wave equation solution that we are trying to approximate, which is periodic and hence remains bounded. On the other hand, if $|\,a_k\,| < 1$, then the eigenvalues (11.46) are complex numbers of modulus 1, indicated stability (but not convergence) of the matrix C. Therefore, in view of (11.45), we should require that

$$\sigma = \frac{c\,k}{h} < 1, \qquad \text{or} \qquad k < \frac{h}{c}\,, \tag{11.47}$$

which places a restriction on the relative sizes of the time and space steps. We conclude that the numerical scheme is conditionally stable.

The stability criterion (11.47) is known as the *Courant condition*, and can be assigned a simple geometric interpretation. Recall that the wave speed c is the slope of the characteristic lines for the wave equation. The Courant condition requires that the *mesh slope*, which is defined to be the ratio of the space step size to the time step size, namely h/k, must be strictly greater than the characteristic slope c. A signal starting at a mesh point (t_i, x_j) will reach positions $x_j \pm k/c$ at the next time $t_{i+1} = t_i + k$, which are still between the mesh points x_{j-1} and x_{j+1}. Thus, characteristic lines that start at a mesh point are not allowed to reach beyond the neighboring mesh points at the next time step.

For instance, in Figure 11.8, the wave speed is $c = 1.25$. The first figure has equal mesh spacing $k = h$, and does not satisfy the Courant condition (11.47), whereas the

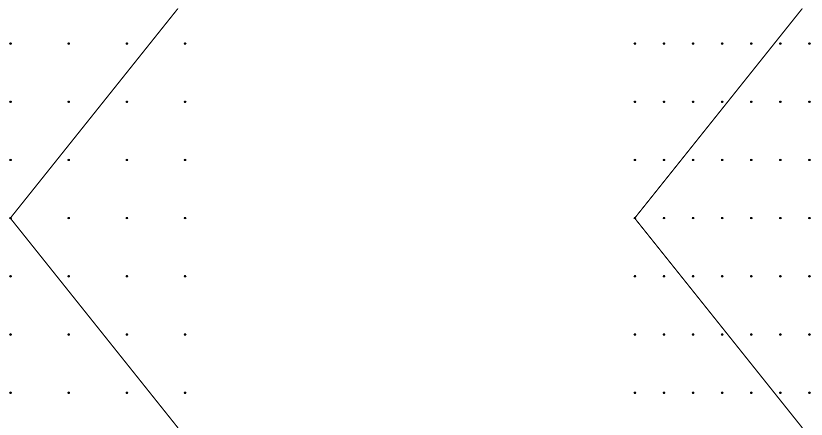

Figure 11.8. The Courant Condition.

second figure has $k = \frac{1}{2} h$, which does. Note how the characteristic lines starting at a given mesh point have progressed beyond the neighboring mesh points after one time step in the first case, but not in the second.