Chapter 5
Finite Differences

As one quickly learns, the differential equations that can be solved by explicit analytic formulas are few and far between. Consequently, the development of accurate numerical approximation schemes is an essential tool for extracting quantitative information as well as achieving a qualitative understanding of the possible behaviors of solutions to the vast majority of partial differential equations. (On the other hand, the successful design of numerical algorithms necessitates a fairly deep understanding of their basic analytic properties, and so exclusive reliance on numerics is not an option.) Even in cases, such as the heat and wave equations, in which explicit solution formulas (either in closed form or infinite series) exist, numerical methods can still be profitably employed. Indeed, one can accurately test a proposed numerical algorithm by running it on a known solution. As we will see, the lessons learned in the design and testing of numerical algorithms on simpler "solved" examples are of inestimable value when confronting more challenging problems.

Many of the basic numerical solution schemes for partial differential equations can be fit into two broad themes. The first, to be presented in the present chapter, is that of *finite difference methods*, obtained by replacing the derivatives in the equation by appropriate numerical differentiation formulae. We thus start with a brief discussion of some elementary finite difference formulas used to numerically approximate first- and second-order derivatives of functions. We then establish and analyze some of the most basic finite difference schemes for the heat equation, first-order transport equations, the second-order wave equation, and the Laplace and Poisson equations. As we will learn, not all finite difference schemes produce accurate numerical approximations, and one must confront issues of stability and convergence in order to distinguish reliable from worthless methods. In fact, inspired by Fourier analysis, the key numerical stability criterion is a consequence of the scheme's handling of complex exponentials.

The second category of numerical solution techniques comprises the *finite element methods*, which will be the topic of Chapter 10. These two chapters should be regarded as but a preliminary excursion into this vast and active area of contemporary research. More sophisticated variations and extensions, as well as other classes of numerical integration schemes, e.g., spectral, pseudo-spectral, multigrid, multipole, probabilistic (Monte Carlo, etc.), geometric, symplectic, and many more, can be found in specialized numerical analysis texts, including [**6**, **51**, **60**, **80**, **94**], and research papers. Also, the journal *Acta Numerica* is an excellent source of survey papers on state-of-the-art numerical methods for a broad range of disciplines.

P.J. Olver, *Introduction to Partial Differential Equations*, Undergraduate Texts in Mathematics,
DOI 10.1007/978-3-319-02099-0_5, © Springer International Publishing Switzerland 2014

5.1 Finite Difference Approximations

In general, a *finite difference* approximation to the value of some derivative of a scalar function $u(x)$ at a point x_0 in its domain, say $u'(x_0)$ or $u''(x_0)$, relies on a suitable combination of sampled function values at nearby points. The underlying formalism used to construct these approximation formulas is known as the *calculus of finite differences*. Its development has a long and influential history, dating back to Newton.

We begin with the first-order derivative. The simplest finite difference approximation is the ordinary *difference quotient*

$$\frac{u(x+h)-u(x)}{h} \approx u'(x), \tag{5.1}$$

which appears in the original calculus definition of the derivative. 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 measures the slope of the secant line through the two points $(x, u(x))$ and $(x+h, u(x+h))$ on its graph. For small enough 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 5.1. Throughout our discussion, h, the *step size*, which may be either positive or negative, is assumed to be small: $|\,h\,| \ll 1$. When $h > 0$, (5.1) is referred to as a *forward difference*, while $h < 0$ yields a *backward difference*.

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 its first-order Taylor expansion

$$u(x+h) = u(x) + u'(x)\,h + \tfrac{1}{2}\,u''(\xi)\,h^2 \tag{5.2}$$

at the point x. We have used Lagrange's formula for the remainder term, [**8**, **97**], in which ξ, which depends on both x and h, is a point lying between x and $x+h$. Rearranging (5.2), we obtain

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

Thus, the *error* in the finite difference approximation (5.1) can be bounded by a multiple of the step size:

$$\left|\,\frac{u(x+h)-u(x)}{h} - u'(x)\,\right| \le C\,|\,h\,|,$$

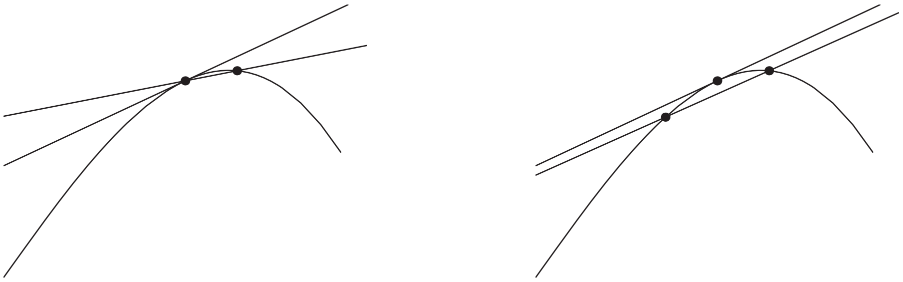
where $C = \max \tfrac{1}{2}\,|\,u''(\xi)\,|$ depends on the magnitude of the second derivative of the function over the interval in question. Since the error is proportional to the first power of h, we say that the finite difference quotient (5.1) is a *first-order* approximation to the derivative $u'(x)$. When the precise formula for the error is not so important, we will write

$$u'(x) = \frac{u(x+h)-u(x)}{h} + \mathrm{O}(h). \tag{5.3}$$

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

Example 5.1. Let $u(x) = \sin x$. Let us try to approximate

$$u'(1) = \cos 1 = .5403023\ldots$$



Forward difference Central difference

Figure 5.1. Finite difference approximations.

by computing finite difference quotients

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

The result for smaller and smaller (positive) values of h is listed in the following table.

h	.1	.01	.001	.0001
approximation	.497364	.536086	.539881	.540260
error	$-.042939$	$-.004216$	$-.000421$	$-.000042$

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 or 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 restrict our attention to equally spaced sample points, although the methods introduced can be readily extended to more general configurations.

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 is found by consideration of the relevant Taylor expansions[†]

$$\begin{aligned} u(x+h) &= u(x) + u'(x)\,h + u''(x)\,\frac{h^2}{2} + u'''(x)\,\frac{h^3}{6} + \mathrm{O}(h^4), \\ u(x-h) &= u(x) - u'(x)\,h + u''(x)\,\frac{h^2}{2} - u'''(x)\,\frac{h^3}{6} + \mathrm{O}(h^4), \end{aligned} \tag{5.4}$$

where the error terms are proportional to h^4. Adding the two formulas together yields

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

[†] Throughout, the function $u(x)$ is assumed to be sufficiently smooth so that any derivatives that appear are well defined and the expansion formula is valid.

Dividing by h^2 and rearranging terms, we arrive at the *centered finite difference approximation* to the second derivative of a function:

$$u''(x) = \frac{u(x+h) - 2\,u(x) + u(x-h)}{h^2} + \mathrm{O}(h^2). \tag{5.5}$$

Since the error is proportional to h^2, this forms a second-order approximation.

Example 5.2. Let $u(x) = e^{x^2}$, with $u''(x) = (4x^2 + 2)\,e^{x^2}$. Let us approximate

$$u''(1) = 6\,e = 16.30969097\ldots$$

using the finite difference quotient (5.5):

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

The results are listed in the following table.

h	.1	.01	.001	.0001
approximation	16.48289823	16.31141265	16.30970819	16.30969115
error	.17320726	.00172168	.00001722	.00000018

Each reduction in step size by a factor of $\frac{1}{10}$ reduces the size of the error by a factor of about $\frac{1}{100}$, thereby gaining two new decimal digits of accuracy, which confirms that the centered 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 .0000092863 — which is *less* accurate than the approximation with $h = .0001$. The problem is that round-off errors due to the finite precision of numbers stored in the computer (in the preceding computation we used single-precision floating-point arithmetic) have now begun to affect the computation. This highlights the inherent difficulty with numerical differentiation: Finite difference formulae inevitably require dividing very small quantities, and so round-off inaccuracies may produce noticeable numerical errors. Thus, while they typically produce reasonably good approximations to the derivatives for moderately small step sizes, achieving high accuracy requires switching to higher-precision computer arithmetic. Indeed, a similar comment applies to the previous computation in Example 5.1. Our expectations about the error were not, in fact, fully justified, as you may have discovered had 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 (5.3) to $u'(x)$ based on the two points x and $x+h$ is not sufficiently accurate, one can try combining the function values at three points, say x, $x+h$, and $x-h$. To find the appropriate combination of function values $u(x-h), u(x), u(x+h)$, we return to the Taylor expansions (5.4). To solve for $u'(x)$, we subtract the two formulas, and so

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

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

$$u'(x) = \frac{u(x+h) - u(x-h)}{2\,h} + \mathrm{O}(h^2), \tag{5.6}$$

which is a second-order approximation to the first derivative. Geometrically, the centered difference quotient represents the slope of the secant line passing through the two points $(x-h,u(x-h))$ and $(x+h,u(x+h))$ on the graph of u, which are centered symmetrically about the point x. Figure 5.1 illustrates the two approximations, and the advantage of the centered difference version is graphically evident. Higher-order approximations can be found by evaluating the function at yet more sample points, say, $x+2h,\ x-2h$, etc.

Example 5.3. Return to the function $u(x)=\sin x$ considered in Example 5.1. The centered difference approximation to its derivative $u'(1)=\cos 1=.5403023\ \ldots$ is

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

The results are tabulated as follows:

h	.1	.01	.001	.0001
approximation	.53940225217	.54029330087	.54030221582	.54030230497
error	−.00090005370	−.00000900499	−.00000009005	−.00000000090

As advertised, the results are much more accurate than the one-sided finite difference approximation used in Example 5.1 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 — up until the point where the effects of round-off error kick in.

Many additional finite difference approximations can be constructed by similar manipulations of Taylor expansions, but these few very basic formulas, along with a couple that are derived in the exercises, will suffice for our purposes. (For a thorough treatment of the calculus of finite differences, the reader can consult [**74**].) In the following sections, we will employ the finite difference formulas to devise numerical solution schemes for a variety of partial differential equations. Applications to the numerical integration of ordinary differential equations can be found, for example, in [**24**, **60**, **63**].

Exercises

♣ 5.1.1. Use the finite difference formula (5.3) with step sizes $h=.1, .01$, and $.001$ to approximate the derivative $u'(1)$ of the following functions $u(x)$. Discuss the accuracy of your approximation. (a) x^4, (b) $\dfrac{1}{1+x^2}$, (c) $\log x$, (d) $\cos x$, (e) $\tan^{-1} x$.

♣ 5.1.2. Repeat Exercise 5.1.1 using the centered difference formula (5.6). Compare your approximations with those in the previous exercise — are the values in accordance with the claimed orders of accuracy?

♣ 5.1.3. Approximate the second derivative $u''(1)$ of the functions in Exercise 5.1.1 using the finite difference formula (5.5) with $h\;=\;.1, .01$, and $.001$. Discuss the accuracy of your approximations.

5.1.4. Construct finite difference approximations to the first and second derivatives of a function $u(x)$ using its values at the points $x-k, x, x+h$, where $h,k \ll 1$ are of comparable size, but not necessarily equal. What can you say about the error in the approximation?

♠ 5.1.5. In this exercise, you are asked to derive some basic *one-sided finite difference formulas*, which are used for approximating derivatives of functions at or near the boundary of their domain. (a) Construct a finite difference formula that approximates the derivative $u'(x)$ using the values of $u(x)$ at the points $x, x+h$, and $x+2h$. What is the order of your formula? (b) Find a finite difference formula for $u''(x)$ that involves the same three function values. What is its order? (c) Test your formulas by computing approximations to the first and second derivatives of $u(x) = e^{x^2}$ at $x = 1$ using step sizes $h = .1, .01$, and $.001$. What is the error in your numerical approximations? Are the errors compatible with the theoretical orders of the finite difference formulas? Discuss why or why not.
(d) Answer part (c) at the point $x = 0$.

♣ 5.1.6. (a) Using the function values $u(x)$, $u(x+h)$, $u(x+3h)$, construct a numerical approximation to the derivative $u'(x)$. (b) What is the order of accuracy of your approximation? (c) Test your approximation on the function $u(x) = \cos x$ at $x = 1$ using the step sizes $h = .1, .01$, and $.001$. Are the errors consistent with your answer in part (b)?

♣ 5.1.7. Answer Exercise 5.1.6 for the second derivative $u''(x)$.

5.1.8. (a) Find the order of the five-point centered finite difference approximation

$$u'(x) \approx \frac{-u(x+2h) + 8u(x+h) - 8u(x-h) + u(x-2h)}{12h}.$$

(b) Test your result on the function $(1+x^2)^{-1}$ at $x = 1$ using the values $h = .1, .01, .001$.

5.1.9. (a) Using the formula in Exercise 5.1.8 as a guide, find five-point finite difference formulas to approximate (i) $u''(x)$, (ii) $u'''(x)$, (iii) $u^{(iv)}(x)$. What is the order of accuracy?
(b) Test your formulas on the function $(1+x^2)^{-1}$ at $x = 1$ using the values $h = .1, .01, .001$.

5.2 Numerical Algorithms for the Heat Equation

Consider the heat equation

$$\frac{\partial u}{\partial t} = \gamma \, \frac{\partial^2 u}{\partial x^2}\,, \qquad 0 < x < \ell, \qquad t > 0, \tag{5.7}$$

on an interval of length ℓ, with constant thermal diffusivity $\gamma > 0$. We impose time-dependent Dirichlet boundary conditions

$$u(t,0) = \alpha(t), \qquad u(t,\ell) = \beta(t), \qquad t > 0, \tag{5.8}$$

fixing the temperature at the ends of the interval, along with the initial conditions

$$u(0,x) = f(x), \qquad 0 \le x \le \ell, \tag{5.9}$$

specifying the 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 *nodes* $(t_j, x_m) \in \mathbb{R}^2$ with

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

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

$$\Delta t = t_{j+1} - t_j, \qquad \Delta x = x_{m+1} - x_m = \frac{\ell}{n}\,,$$

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

$$u_{j,m} \approx u(t_j, x_m), \qquad \text{where} \qquad t_j = j\,\Delta t, \qquad x_m = m\,\Delta x, \tag{5.10}$$

to denote the numerical approximation to the solution value at the indicated node.

As a first attempt at designing a numerical solution scheme, we shall employ the simplest finite difference approximations to the derivatives appearing in the equation. The second-order space derivative is approximated by the centered difference formula (5.5), and hence

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2}(t_j, x_m) &\approx \frac{u(t_j, x_{m+1}) - 2\,u(t_j, x_m) + u(t_j, x_{m-1})}{(\Delta x)^2} + \mathrm{O}\big((\Delta x)^2\big) \\ &\approx \frac{u_{j,m+1} - 2\,u_{j,m} + u_{j,m-1}}{(\Delta x)^2} + \mathrm{O}\big((\Delta x)^2\big), \end{aligned} \tag{5.11}$$

where the error in the approximation is proportional to $(\Delta x)^2$. Similarly, the one-sided finite difference approximation (5.3) is used to approximate the time derivative, and so

$$\frac{\partial u}{\partial t}(t_j, x_m) \approx \frac{u(t_{j+1}, x_m) - u(t_j, x_m)}{\Delta t} + \mathrm{O}(\Delta t) \approx \frac{u_{j+1,m} - u_{j,m}}{\Delta t} + \mathrm{O}(\Delta t), \tag{5.12}$$

where the error is proportional to Δt. In general, one should try to ensure that the approximations have similar orders of accuracy, which leads us to require

$$\Delta t \approx (\Delta x)^2. \tag{5.13}$$

Assuming $\Delta x < 1$, this implies that the time steps must be *much* smaller than the space mesh size.

Remark: At this stage, the reader might be tempted to replace (5.12) by the second-order central difference approximation (5.6). However, this introduces significant complications, and the resulting numerical scheme is not practical; see Exercise 5.2.10.

Replacing the derivatives in the heat equation (5.14) by their finite difference approximations (5.11, 12) and rearranging terms, we end up with the linear system

$$u_{j+1,m} = \mu\, u_{j,m+1} + (1 - 2\,\mu) u_{j,m} + \mu\, u_{j,m-1}, \qquad \begin{aligned} &j = 0, 1, 2, \ldots\,, \\ &m = 1, \ldots, n-1, \end{aligned} \tag{5.14}$$

in which

$$\mu = \frac{\gamma\,\Delta t}{(\Delta x)^2}\,. \tag{5.15}$$

The resulting scheme is of iterative form, whereby the solution values $u_{j+1,m} \approx u(t_{j+1}, x_m)$ at time t_{j+1} are successively calculated, via (5.14), from those at the preceding time t_j.

The initial condition (5.9) indicates that we should initialize our numerical data by sampling the initial temperature at the nodes:

$$u_{0,m} = f_m = f(x_m), \qquad m = 1, \ldots, n-1. \tag{5.16}$$

Similarly, the boundary conditions (5.8) require that

$$u_{j,0} = \alpha_j = \alpha(t_j), \qquad u_{j,n} = \beta_j = \beta(t_j), \qquad j = 0, 1, 2, \ldots\,. \tag{5.17}$$

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

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

The three equations (5.14, 16, 17) completely prescribe the numerical approximation scheme for the solution to the initial-boundary value problem (5.7–9).

Let us rewrite the preceding equations in a more transparent vectorial form. First, let

$$\mathbf{u}^{(j)} = \bigl(u_{j,1}, u_{j,2}, \ldots, u_{j,n-1}\bigr)^T \approx \bigl(u(t_j,x_1), u(t_j,x_2), \ldots, u(t_j,x_{n-1})\bigr)^T \tag{5.18}$$

be the vector whose entries are the numerical approximations to the solution values at time t_j at the *interior* nodes. We omit the boundary nodes (t_j,x_0), (t_j,x_n), since those values are fixed by the boundary conditions (5.17). Then (5.14) takes the form

$$\mathbf{u}^{(j+1)} = A\,\mathbf{u}^{(j)} + \mathbf{b}^{(j)}, \tag{5.19}$$

where

$$A = \begin{pmatrix} 1-2\mu & \mu & & & & \\ \mu & 1-2\mu & \mu & & & \\ & \mu & 1-2\mu & \mu & & \\ & & \mu & \ddots & \ddots & \\ & & & \ddots & \ddots & \mu \\ & & & & \mu & 1-2\mu \end{pmatrix}, \qquad \mathbf{b}^{(j)} = \begin{pmatrix} \mu\,\alpha_j \\ 0 \\ 0 \\ \vdots \\ 0 \\ \mu\,\beta_j \end{pmatrix}. \tag{5.20}$$

The $(n-1)\times(n-1)$ coefficient matrix A is symmetric and tridiagonal, and only its nonzero entries are displayed. The contributions (5.17) of the boundary nodes appear in the vector $\mathbf{b}^{(j)} \in \mathbb{R}^{n-1}$. This numerical method is known as an *explicit scheme*, since each iterate is computed directly from its predecessor without having to solve any auxiliary equations — unlike the implicit schemes to be discussed next.

Example 5.4. Let us fix the diffusivity $\gamma = 1$ and the interval length $\ell = 1$. For illustrative purposes, we take a spatial step size of $\Delta x = .1$. We work with the initial data

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

used earlier in Example 4.1. In Figure 5.2 we compare the numerical solutions resulting from two (slightly) different time step sizes. The first row uses $\Delta t = (\Delta x)^2 = .01$ and plots the solution at the indicated times. The numerical solution is already showing signs of instability (the final plot does not even fit in the window), and indeed, soon thereafter, it becomes completely wild. The second row takes $\Delta t = .005$. 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 seen in Figure 4.1.

Stability Analysis

In light of the preceding calculation, we need to understand why our numerical scheme sometimes gives reasonable answers but sometimes utterly fails. To this end, we investigate

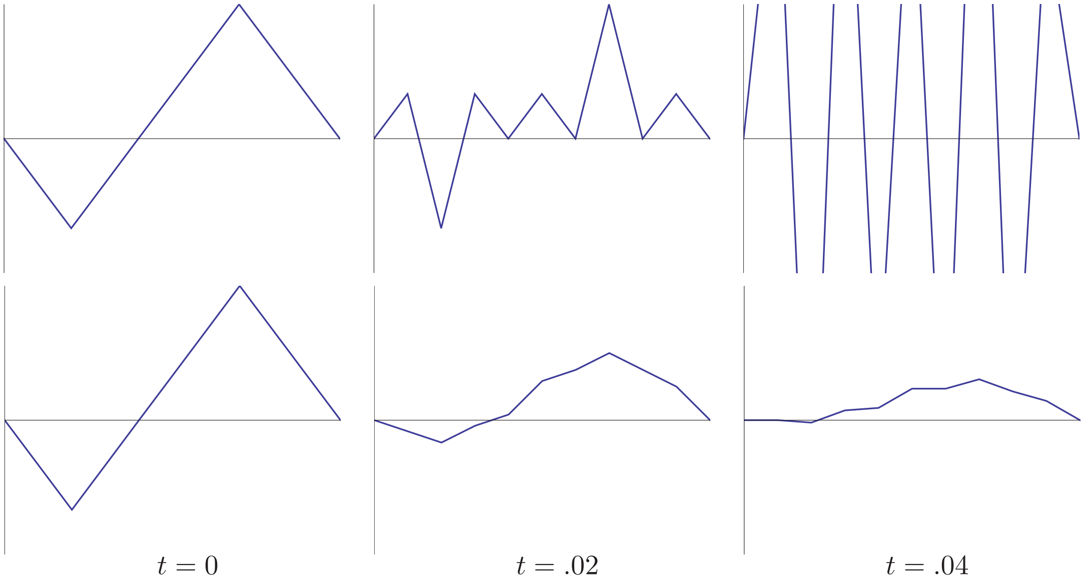


Figure 5.2. Numerical solutions for the heat equation based on the explicit scheme.

the effect of the numerical scheme on simple functions. As we know, the general solution to the heat equation can be decomposed into a sum over the various Fourier modes. Thus, we can concentrate on understanding what the numerical scheme does to an individual complex exponential,[†] bearing in mind that we can then reconstruct its effect on more general initial data by taking suitable linear combinations of exponentials.

To this end, suppose that, at time $t = t_j$, the solution is a sampled exponential

$$u(t_j, x) = e^{\,i\,k\,x}, \qquad \text{and so} \qquad u_{j,m} = u(t_j, x_m) = e^{\,i\,k\,x_m}, \tag{5.21}$$

where k is a real parameter. Substituting the latter values into our numerical equations (5.14), we find that the updated value at time t_{j+1} is also a sampled exponential:

$$\begin{aligned} u_{j+1,m} &= \mu\, u_{j,m+1} + (1 - 2\mu) u_{j,m} + \mu\, u_{j,m-1} \\ &= \mu\, e^{\,i\,k\,x_{m+1}} + (1 - 2\mu) e^{\,i\,k\,x_m} + \mu\, e^{\,i\,k\,x_{m-1}} \\ &= \mu\, e^{\,i\,k(x_m + \Delta x)} + (1 - 2\mu) e^{\,i\,k\,x_m} + \mu\, e^{\,i\,k(x_m - \Delta x)} \\ &= \lambda\, e^{\,i\,k\,x_m}, \end{aligned} \tag{5.22}$$

where

$$\begin{aligned} \lambda = \lambda(k) &= \mu\, e^{\,i\,k\,\Delta x} + (1 - 2\mu) + \mu\, e^{-\,i\,k\,\Delta x} \\ &= 1 - 2\mu\big[\,1 - \cos(k\,\Delta x)\,\big] = 1 - 4\mu \sin^2\big(\tfrac{1}{2}\,k\,\Delta x\big). \end{aligned} \tag{5.23}$$

Thus, the effect of a single step is to multiply the complex exponential (5.21) by the *magnification factor* λ:

$$u(t_{j+1}, x) = \lambda\, e^{\,i\,k\,x}. \tag{5.24}$$

[†] As usual, complex exponentials are easier to work with than real trigonometric functions.

In other words, $e^{\,\mathrm{i}\,k\,x}$ plays the role of an *eigenfunction*, with the magnification factor $\lambda(k)$ the corresponding *eigenvalue*, of the linear operator governing each step of the numerical scheme. Continuing in this fashion, we find that the effect of p further iterations of the scheme is to multiply the exponential by the p^{th} power of the magnification factor:

$$u(t_{j+p},x) = \lambda^p\, e^{\,\mathrm{i}\,k\,x}. \tag{5.25}$$

As a result, the stability is governed by the size of the magnification factor: If $|\,\lambda\,| > 1$, then λ^p grows exponentially, and so the numerical solutions (5.25) become unbounded as $p \to \infty$, which is clearly incompatible with the analytical behavior of solutions to the heat equation. Therefore, an evident necessary condition for the stability of our numerical scheme is that its magnification factor satisfy

$$|\,\lambda\,| \le 1. \tag{5.26}$$

This method of stability analysis was developed by the mid-twentieth-century Hungarian/American mathematician — and father of the electronic computer — John von Neumann. The *stability criterion* (5.26) effectively distinguishes the stable, and hence valid, numerical algorithms from the unstable, and hence ineffectual, schemes. For the particular case (5.23), the von Neumann stability criterion (5.26) requires

$$-1 \le 1 - 4\,\mu\sin^2\bigl(\tfrac{1}{2}\,k\,\Delta x\bigr) \le 1, \qquad \text{or, equivalently,} \qquad 0 \le \mu\sin^2\bigl(\tfrac{1}{2}\,k\,\Delta x\bigr) \le \tfrac{1}{2}.$$

Since this is required to hold for all possible k, we must have

$$0 \le \mu = \frac{\gamma\,\Delta t}{(\Delta x)^2} \le \frac{1}{2}\,, \qquad \text{and hence} \qquad \Delta t \le \frac{(\Delta x)^2}{2\gamma}\,, \tag{5.27}$$

since $\gamma > 0$. Thus, once the space mesh size is fixed, stability of the numerical scheme places a restriction on the allowable time step size. For instance, if $\gamma = 1$, and the space mesh size $\Delta x = .01$, then we must adopt a minuscule time step size $\Delta t \le .00005$. It would take an exorbitant number of time steps to compute the value of the solution at even moderate times, e.g., $t = 1$. Moreover, the accumulation of round-off errors might then cause a significant reduction in the overall accuracy of the final solution values. Since not all choices of space and time steps lead to a convergent scheme, the explicit scheme (5.14) is called *conditionally stable*.

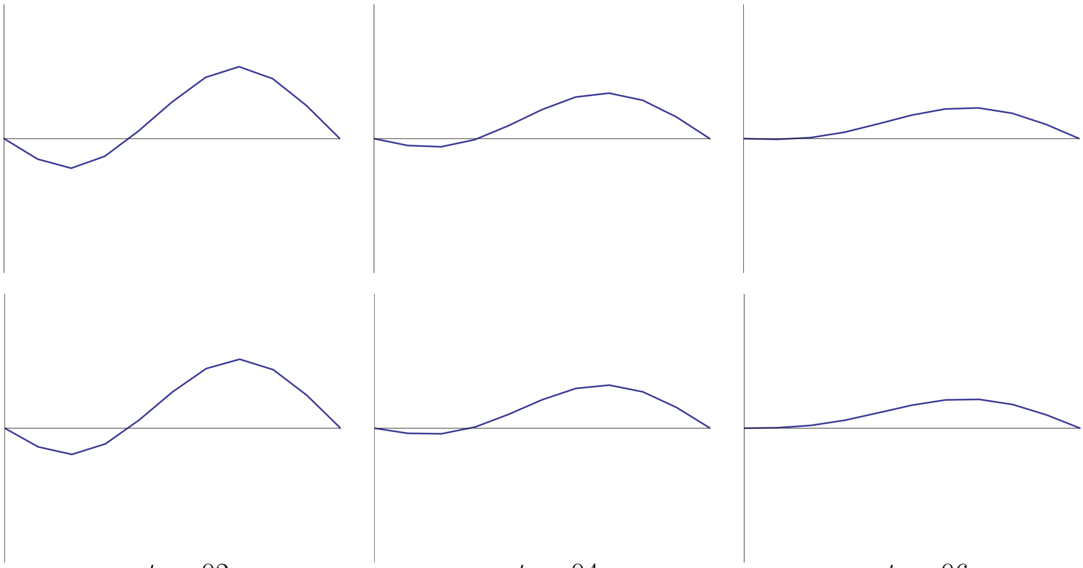
Implicit and Crank–Nicolson Methods

An unconditionally stable method — one that does not restrict the time step — can be constructed by replacing the forward difference formula (5.12) used to approximate the time derivative by the backwards difference formula

$$\frac{\partial u}{\partial t}\,(t_j,x_m) \approx \frac{u(t_j,x_m) - u(t_{j-1},x_m)}{\Delta t} + \mathrm{O}\bigl(\,(\Delta t)^2\,\bigr). \tag{5.28}$$

Substituting (5.28) and the same centered difference approximation (5.11) for u_{xx} into the heat equation, and then replacing j by $j+1$, leads to the iterative system

$$-\,\mu\,u_{j+1,m+1} + (1+2\,\mu)u_{j+1,m} - \mu\,u_{j+1,m-1} = u_{j,m}, \qquad \begin{aligned} j &= 0,1,2,\ldots\,, \\ m &= 1,\ldots,n-1, \end{aligned} \tag{5.29}$$



$t = .02$ $t = .04$ $t = .06$

Figure 5.3. Numerical solutions for the heat equation based on the implicit scheme.

where the parameter $\mu = \gamma\,\Delta t/(\Delta x)^2$ is as before. The initial and boundary conditions have the same form (5.16, 17). The latter system can be written in the matrix form

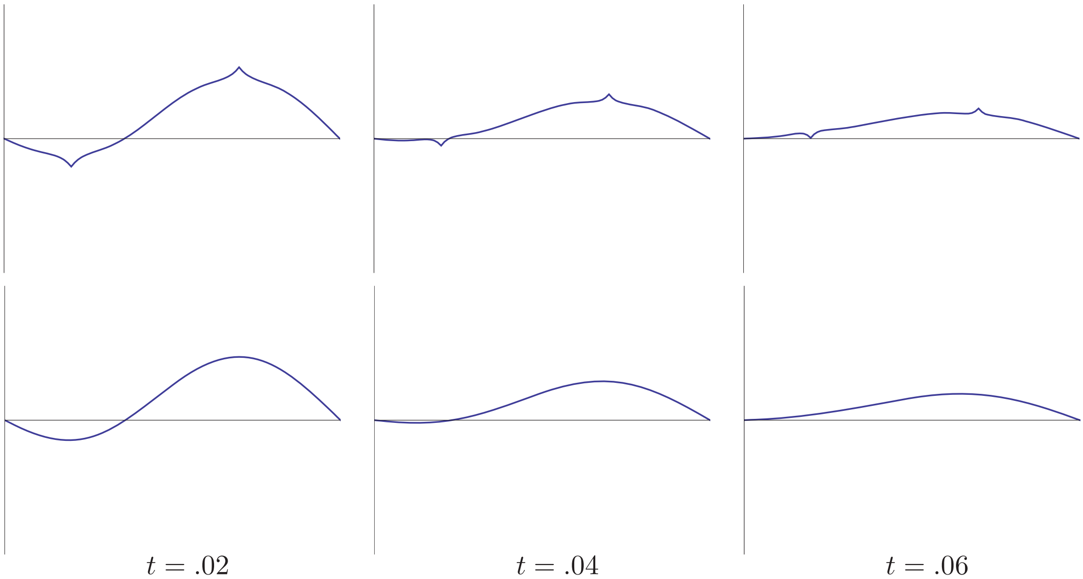
$$\widehat{A}\,\mathbf{u}^{(j+1)} = \mathbf{u}^{(j)} + \mathbf{b}^{(j+1)}, \tag{5.30}$$

where $\widehat{A}$ is obtained from the matrix A in (5.20) by replacing μ by $-\mu$. This serves to define an *implicit scheme*, since we have to solve a linear system of algebraic equations at each step in order to compute the next iterate $\mathbf{u}^{(j+1)}$. However, since the coefficient matrix $\widehat{A}$ is tridiagonal, the solution can be computed extremely rapidly, [**89**], and so its calculation is not an impediment to the practical implementation of this implicit scheme.

Example 5.5. Consider the same initial-boundary value problem considered in Example 5.4. In Figure 5.3, we plot the numerical solutions obtained using the implicit scheme. The initial data is not displayed, but we graph the numerical solutions at times $t = .02, .04, .06$ with a mesh size of $\Delta x = .1$. In the top row, we use a time step of $\Delta t = .01$, while in the bottom row $\Delta t = .005$. In contrast to the explicit scheme, there is very little difference between the two — indeed, both come much closer to the actual solution than the explicit scheme. In fact, even significantly larger time steps yield reasonable numerical approximations to the solution.

Let us apply the von Neumann analysis to investigate the stability of the implicit scheme. Again, we need only look at the effect of the scheme on a complex exponential. Substituting (5.21, 24) into (5.29) and canceling the common exponential factor leads to the equation

$$\lambda\,(-\mu\,e^{\,\mathrm{i}\,k\,\Delta x} + 1 + 2\,\mu - \mu\,e^{-\,\mathrm{i}\,k\,\Delta x}) = 1.$$



$t = .02$ $t = .04$ $t = .06$

Figure 5.4. Numerical Solutions for the heat equation based on the Crank–Nicolson scheme.

We solve for the magnification factor

$$\lambda = \frac{1}{1 + 2\,\mu\bigl(1 - \cos(k\,\Delta x)\bigr)} = \frac{1}{1 + 4\,\mu \sin^2\bigl(\frac{1}{2}\,k\,\Delta x\bigr)}\,. \tag{5.31}$$

Since $\mu > 0$, the magnification factor is *always* less than 1 in absolute value, and so the stability criterion (5.26) is satisfied *for any choice of step sizes*. We conclude that the implicit scheme (5.14) is *unconditionally stable*.

Another popular numerical scheme for solving the heat equation is the *Crank–Nicolson method*, due to the British numerical analysts John Crank and Phyllis Nicolson:

$$u_{j+1,m} - u_{j,m} = \tfrac{1}{2}\,\mu\,(u_{j+1,m+1} - 2\,u_{j+1,m} + u_{j+1,m-1} + u_{j,m+1} - 2\,u_{j,m} + u_{j,m-1}), \tag{5.32}$$

which can be obtained by averaging the explicit and implicit schemes (5.14) and (5.29). We can write (5.32) in vectorial form

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

where

$$\widehat{B} = \begin{pmatrix} 1+\mu & -\frac{1}{2}\,\mu & & \\ -\frac{1}{2}\,\mu & 1+\mu & -\frac{1}{2}\,\mu & \\ & -\frac{1}{2}\,\mu & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}, \qquad B = \begin{pmatrix} 1-\mu & \frac{1}{2}\,\mu & & \\ \frac{1}{2}\,\mu & 1-\mu & \frac{1}{2}\,\mu & \\ & \frac{1}{2}\,\mu & \ddots & \ddots \\ & & \ddots & \ddots \end{pmatrix}, \tag{5.33}$$

are both tridiagonal.

Applying the von Neumann analysis as before, we deduce that the magnification factor has the form

$$\lambda = \frac{1 - 2\,\mu \sin^2\bigl(\frac{1}{2}\,k\,\Delta x\bigr)}{1 + 2\,\mu \sin^2\bigl(\frac{1}{2}\,k\,\Delta x\bigr)}\,. \tag{5.34}$$

Since $\mu > 0$, we see that $|\lambda| \leq 1$ for all choices of step size, and so the Crank–Nicolson scheme is also unconditionally stable. A detailed analysis based on a Taylor expansion of the solution reveals that the errors are of order $(\Delta t)^2$ and $(\Delta x)^2$, and so it is reasonable to choose the time step to have the same order of magnitude as the space step: $\Delta t \approx \Delta x$. This gives the Crank–Nicolson scheme a significant advantage over the previous two methods, in that one can get away with far fewer time steps. However, applying it to the initial value problem considered above reveals a subtle weakness. The top row in Figure 5.4 has space and time step sizes $\Delta t = \Delta x = .01$, and does a reasonable job of approximating the solution except near the corners, where an annoying and incorrect local oscillation persists as the solution decays. The bottom row uses $\Delta t = \Delta x = .001$, and performs much better, although a similar oscillatory error can be observed at much smaller times. Indeed, unlike the implicit scheme, the Crank–Nicolson method fails to rapidly damp out the high-frequency Fourier modes associated with small-scale features such as discontinuities and corners in the initial data, although it performs quite well in smooth regimes. Thus, when dealing with irregular initial data, a good strategy is to first run the implicit scheme until the small-scale noise is dissipated away, and then switch to Crank–Nicolson with a much larger time step to determine the later large scale dynamics.

Finally, we remark that the finite difference schemes developed above for the heat equation can all be readily adapted to more general parabolic partial differential equations. The stability criteria and observed behaviors are fairly similar, and a couple of illustrative examples can be found in the exercises.

Exercises

5.2.1. Suppose we seek to approximate the solution to the initial-boundary value problem

$$u_t = 5\,u_{xx}, \qquad u(t,0) = u(t,3) = 0, \qquad u(0,x) = x(x-1)(x-3), \qquad 0 \leq x \leq 3,$$

by employing the explicit scheme (5.14). (*a*) Given the spatial mesh size $\Delta x = .1$, what range of time steps Δt can be used to produce an accurate numerical approximation? (*b*) Test your prediction by implementing the scheme using one value of Δt in the allowed range and one value outside.

5.2.2. Solve the following initial-boundary value problem

$$u_t = u_{xx}, \qquad u(t,0) = u(t,1) = 0, \qquad u(0,x) = f(x), \qquad 0 \leq x \leq 1,$$

with initial data $f(x) = \begin{cases} 2\left|x - \frac{1}{6}\right| - \frac{1}{3}, & 0 \leq x \leq \frac{1}{3}, \\ 0, & \frac{1}{3} \leq x \leq \frac{2}{3}, \\ \frac{1}{2} - 3\left|x - \frac{5}{6}\right|, & \frac{2}{3} \leq x \leq 1, \end{cases}$ using

(*i*) the explicit scheme (5.14); (*ii*) the implicit scheme (5.29); and (*iii*) the Crank–Nicolson scheme (5.32). Use space step sizes $\Delta x = .1$ and $.05$, and suitably chosen time steps Δt. Discuss which features of the solution can be observed in your numerical approximations.

5.2.3. Repeat Exercise 5.2.2 for the initial-boundary value problem $u_t = 3u_{xx}$, $u(0,x) = 0$, $u(t,-1) = 1$, $u(t,1) = -1$, using space step sizes $\Delta x = .2$ and $.1$.

5.2.4. (*a*) Solve the initial-boundary value problem

$$u_t = u_{xx}, \qquad u(t,-1) = u(t,1) = 0, \qquad u(0,x) = |x|^{1/2} - x^2, \qquad -1 \leq x \leq 1,$$

using (*i*) the explicit scheme (5.14); (*ii*) the implicit scheme (5.29); (*iii*) the Crank–Nicolson scheme (5.32). Use $\Delta x = .1$ and an appropriate time step Δt. Compare your numerical solutions at times $t = 0, .01, , .02, .05, .1, .3, .5, 1.0$, and discuss your findings. (*b*) Repeat part (*a*) for the implicit and Crank-Nicolson schemes with $\Delta x = .01$. Why aren't you being asked to implement the explicit scheme?

5.2.5. Use the implicit scheme with spatial mesh sizes $\Delta x = .1$ and $.05$ and appropriately chosen values of the time step Δt to investigate the solution to the periodically forced boundary value problem $u_t = u_{xx}$, $u(0,x) = 0$, $u(t,0) = \sin 5\pi t$, $u(t,1) = \cos 5\pi t$. Is your solution periodic in time?

♡ 5.2.6. (*a*) How would you modify (*i*) the explicit scheme; (*ii*) the implicit scheme; to deal with Neumann boundary conditions? *Hint*: Use the one-sided finite difference formulae found in Exercise 5.1.5 to approximate the derivatives at the boundary.
(*b*) Test your proposals on the boundary value problem

$$u_t = u_{xx}, \qquad u(0,x) = \tfrac{1}{2} + \cos 2\pi x - \tfrac{1}{2}\cos 3\pi x, \qquad u_x(t,0) = 0 = u_x(t,1),$$

using space step sizes $\Delta x = .1$ and $.01$ and appropriate time steps. Compare your numerical solution with the exact solution at times $t = .01, .03, .05$, and explain any discrepancies.

5.2.7. (*a*) Design an explicit numerical scheme for approximating the solution to the initial-boundary value problem

$$u_t = \gamma\, u_{xx} + s(x), \qquad u(t,0) = u(t,1) = 0, \qquad u(0,x) = f(x), \qquad 0 \le x \le 1,$$

for the heat equation with a *source term* $s(x)$. (*b*) Test your scheme when

$$\gamma = \tfrac{1}{6}, \qquad s(x) = x(1-x)(10-22x), \qquad f(x) = \begin{cases} 2\left|x - \tfrac{1}{6}\right| - \tfrac{1}{3}, & 0 \le x \le \tfrac{1}{3}, \\ 0, & \tfrac{1}{3} \le x \le \tfrac{2}{3}, \\ \tfrac{1}{2} - 3\left|x - \tfrac{5}{6}\right|, & \tfrac{2}{3} \le x \le 1, \end{cases}$$

using space step sizes $\Delta x = .1$ and $.05$, and a suitably chosen time step Δt. Are your two numerical solutions close? (*c*) What is the long-term behavior of the solution? Can you find a formula for its eventual profile? (*d*) Design an implicit scheme for the same problem. Does this affect the behavior of your numerical solution? What are the advantages of the implicit scheme?

5.2.8. Consider the initial-boundary value problem for the *lossy diffusion equation*

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - \alpha\, u, \qquad u(t,0) = u(t,1) = 0, \qquad u(0,x) = f(x), \qquad \begin{array}{c} t \ge 0, \\ 0 \le x \le 1, \end{array}$$

where $\alpha > 0$ is a positive constant. (*a*) Devise an explicit finite difference method for computing a numerical approximation to the solution. (*b*) For what mesh sizes would you expect your method to provide a good approximation to the solution?
(*c*) Discuss the case when $\alpha < 0$.

5.2.9. Consider the initial-boundary value problem for the *diffusive transport equation*

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + 2\,\frac{\partial u}{\partial x}, \qquad u(t,0) = u(t,1) = 0, \qquad u(0,x) = x(1-x), \qquad \begin{array}{c} t \ge 0, \\ 0 \le x \le 1. \end{array}$$

(*a*) Devise an explicit finite difference scheme for computing numerical approximations to the solution. *Hint*: Make sure your approximations are of comparable order. (*b*) For what range of time step sizes would you expect your method to provide a decent approximation to the solution? (*c*) Test your answer in part (*b*) for the spatial step size $\Delta x = .1$.

♢ 5.2.10. (*a*) Show that using the centered difference approximation (5.6) to approximate the time derivative leads to *Richardson's method* for numerically solving the heat equation:

$$u_{j+1,m} = u_{j-1,m} + 2\mu\,(u_{j,m+1} - 2u_{j,m} + u_{j,m-1}), \qquad \begin{array}{c} j = 1, 2, \ldots\,, \\ m = 1, \ldots, n-1, \end{array}$$

where $\mu = \gamma\,\Delta t/(\Delta x)^2$ is as in (5.15). (*b*) Discuss how to start Richardson's method. (*c*) Discuss the stability of Richardson's method. (*d*) Test Richardson's method on the initial-boundary value problem in Exercise 5.2.2. Does your numerical solution conform with your expectations from part (*b*)?

5.3 Numerical Algorithms for First–Order Partial Differential Equations

Let us next apply the method of finite differences to construct some basic numerical methods for first-order partial differential equations. As noted in Section 4.4, first-order partial differential equations are prototypes for hyperbolic equations, and so many of the lessons learned here carry over to the general hyperbolic regime, including the second-order wave equation, which we analyze in detail in the following section.

Consider the initial value problem for the elementary transport equation

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = 0, \qquad u(0,x) = f(x), \qquad -\infty < x < \infty, \tag{5.35}$$

with constant wave speed c. Of course, as we learned in Section 2.2, the solution is a simple traveling wave

$$u(t,x) = f(x - c\,t) \tag{5.36}$$

that is constant along the characteristic lines of slope c in the (t,x)–plane. Although the analytical solution is completely elementary, there will be valuable lessons to be learned from our attempt to reproduce it by numerical approximation. Indeed, each of the numerical schemes developed below has an evident adaptation to transport equations with variable wave speeds $c(t,x)$, and even to nonlinear transport equations whose wave speed depends on the solution u, and so admit shock-wave solutions.

As before, we restrict our attention to a rectangular mesh (t_j, x_m) with uniform time step size $\Delta t = t_{j+1} - t_j$ and space mesh size $\Delta x = x_{m+1} - x_m$. We use $u_{j,m} \approx u(t_j, x_m)$ to denote our numerical approximation to the solution $u(t,x)$ at the indicated node. The simplest numerical scheme is obtained by replacing the time and space derivatives by their first-order finite difference approximations (5.1):

$$\frac{\partial u}{\partial t}\,(t_j, x_m) \approx \frac{u_{j+1,m} - u_{j,m}}{\Delta t} + \mathrm{O}(\Delta t), \qquad \frac{\partial u}{\partial x}\,(t_j, x_m) \approx \frac{u_{j,m+1} - u_{j,m}}{\Delta x} + \mathrm{O}(\Delta x). \tag{5.37}$$

Substituting these expressions into the transport equation (5.35) leads to the explicit numerical scheme

$$u_{j+1,m} = -\,\sigma\, u_{j,m+1} + (\sigma + 1) u_{j,m}, \tag{5.38}$$

in which the parameter

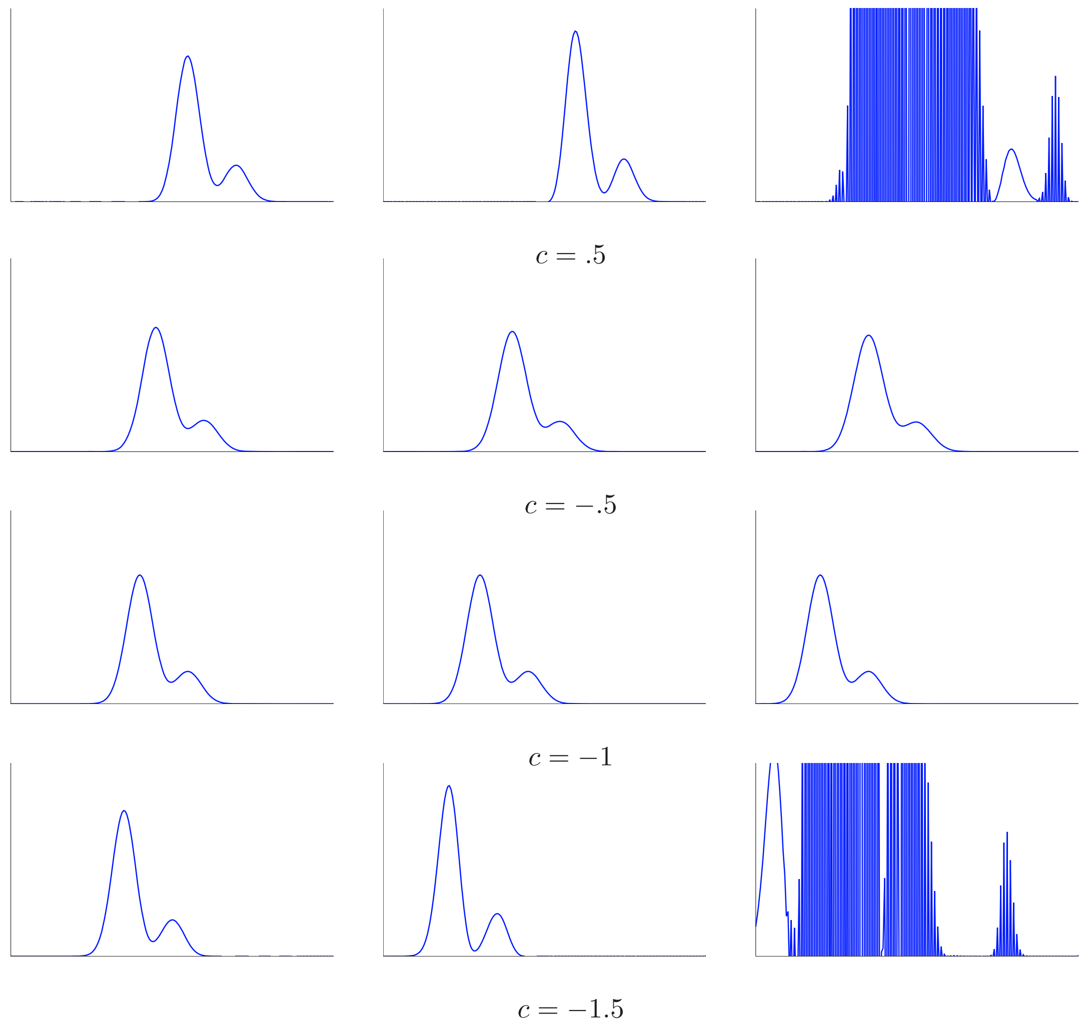
$$\sigma = \frac{c\,\Delta t}{\Delta x} \tag{5.39}$$

depends on the wave speed and the ratio of time to space step sizes. Since we are employing first-order approximations to both derivatives, we should choose the step sizes to be comparable: $\Delta t \approx \Delta x$. When working on a bounded interval, say $0 \le x \le \ell$, we will need to specify a value for the numerical solution at the right end, e.g., setting $u_{j,n} = 0$, which corresponds to imposing the boundary condition $u(t,\ell) = 0$.

In Figure 5.5, we plot the numerical solutions, at times $t = .1, .2, .3$, arising from the following initial condition:

$$u(0,x) = f(x) = .4\, e^{-300(x-.5)^2} + .1\, e^{-300(x-.65)^2}. \tag{5.40}$$

We use step sizes $\Delta t = \Delta x = .005$, and try four different values of the wave speed. The cases $c = .5$ and $c = -1.5$ clearly exhibit some form of numerical instability. The numerical



$c = .5$

$c = -.5$

$c = -1$

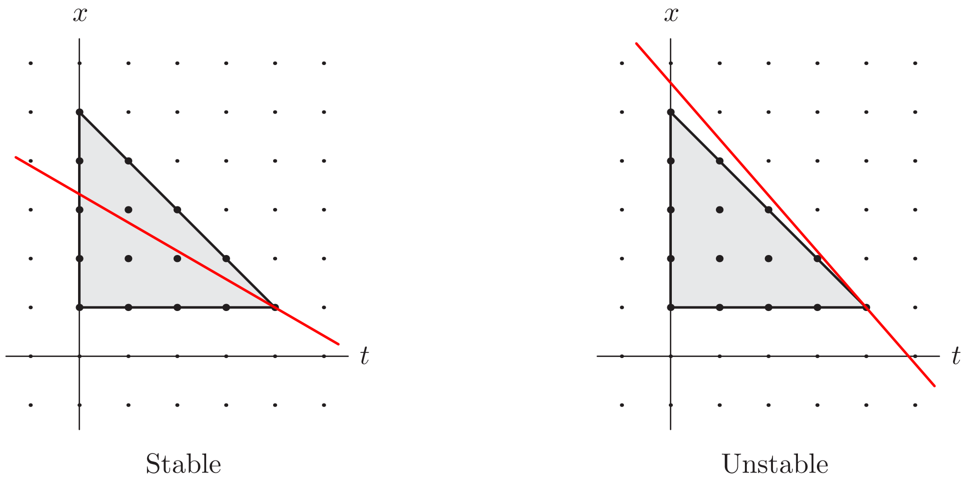
$c = -1.5$

Figure 5.5. Numerical solutions to the transport equation.

solution when $c = -.5$ is a bit more reasonable, although one can already observe some degradation due to the relatively low accuracy of the scheme. This can be alleviated by employing a smaller step size. The case $c = -1$ looks exceptionally good, and you are asked to provide an explanation in Exercise 5.3.6.

The CFL Condition

There are two ways to understand the observed numerical instability. First, we recall that the exact solution (5.36) is constant along the characteristic lines $x = ct + \xi$, and hence the value of $u(t, x)$ depends only on the initial value $f(\xi)$ at the point $\xi = x - ct$. On the other hand, at time $t = t_j$, the numerical solution $u_{j,m} \approx u(t_j, x_m)$ computed using (5.38) depends on the values of $u_{j-1,m}$ and $u_{j-1,m+1}$. The latter two values have



Stable

Unstable

Figure 5.6. The CFL condition.

been computed from the previous approximations $u_{j-2,m}$, $u_{j-2,m+1}$, $u_{j-2,m+2}$. And so on. Going all the way back to the initial time $t_0 = 0$, we find that $u_{j,m}$ depends on the initial values $u_{0,m} = f(x_m), \ \ldots\ , u_{0,m+j} = f(x_m + j\,\Delta x)$ at the nodes lying in the interval $x_m \leq x \leq x_m + j\,\Delta x$. On the other hand, the actual solution $u(t_j, x_m)$ depends only on the value of $f(\xi)$, where

$$\xi = x_m - c\,t_j = x_m - c\,j\,\Delta t.$$

Thus, if ξ lies outside the interval $[\,x_m, x_m + j\,\Delta x\,]$, then varying the initial condition near the point $x = \xi$ will change the actual solution value $u(t_j, x_m)$ without altering its numerical approximation $u_{j,m}$ at all! So the numerical scheme cannot possibly provide an accurate approximation to the solution value. As a result, we must require

$$x_m \leq \xi = x_m - c\,j\,\Delta t \leq x_m + j\,\Delta x, \qquad \text{and hence} \qquad 0 \leq -c\,\Delta t \leq \Delta x,$$

which we rewrite as

$$0 \geq \sigma = \frac{c\,\Delta t}{\Delta x} \geq -1, \qquad \text{or, equivalently,} \qquad -\frac{\Delta x}{\Delta t} \leq c \leq 0. \tag{5.41}$$

This is the simplest manifestation of what is known as the *Courant–Friedrichs–Lewy condition*, or *CFL condition* for short, which was established in the groundbreaking 1928 paper [**33**] by three of the pioneers in the development of numerical methods for partial differential equations: the German (soon to be American) applied mathematicians Richard Courant, Kurt Friedrichs, and Hans Lewy. Note that the CFL condition requires that the wave speed be *negative*, and the time step size not too large. Thus, for allowable wave speeds, the finite difference scheme (5.38) is conditionally stable.

The CFL condition can be recast in a more geometrically transparent manner as follows. For the finite difference scheme (5.38), the *numerical domain of dependence* of a point (t_j, x_m) is the triangle

$$T_{(t_j,x_m)} = \left\{\, (t,x) \;\middle|\; 0 \leq t \leq t_j,\ \ x_m \leq x \leq x_m + t_j - t \,\right\}. \tag{5.42}$$

The reason for this nomenclature is that, as we have just seen, the numerical approximation to the solution at the node (t_j, x_m) depends on the computed values at the nodes lying within its numerical domain of dependence; see Figure 5.6. The CFL condition (5.41) requires that, for all $0 \le t \le t_j$, the characteristic passing through the point (t_j, x_m) lie entirely within the numerical domain of dependence (5.42). If the characteristic ventures outside the domain, then the scheme will be numerically unstable. With this geometric reformulation, the CFL criterion can be applied to both linear and nonlinear transport equations that have nonuniform wave speeds.

The CFL criterion (5.41) is reconfirmed by a von Neumann stability analysis. As before, we test the numerical scheme on an exponential function. Substituting

$$u_{j,m} = e^{\,\mathrm{i}\,k\,x_m}, \qquad u_{j+1,m} = \lambda\, e^{\,\mathrm{i}\,k\,x_m}, \tag{5.43}$$

into (5.38) leads to

$$\lambda e^{\,\mathrm{i}\,k\,x_m} = -\,\sigma\, e^{\,\mathrm{i}\,k\,x_{m+1}} + (\sigma+1) e^{\,\mathrm{i}\,k\,x_m} = \bigl(-\,\sigma\, e^{\,\mathrm{i}\,k\,\Delta x} + \sigma + 1\bigr) e^{\,\mathrm{i}\,k\,x_m}.$$

The resulting (complex) magnification factor

$$\lambda = 1 + \sigma\bigl(1 - e^{\,\mathrm{i}\,k\,\Delta x}\bigr) = \bigl(\,1 + \sigma - \sigma\cos(k\,\Delta x)\,\bigr) -\, \mathrm{i}\,\sigma\sin(k\,\Delta x)$$

satisfies the stability criterion $|\,\lambda\,| \le 1$ if and only if

$$\begin{aligned} |\,\lambda\,|^2 &= \bigl(\,1 + \sigma - \sigma\cos(k\,\Delta x)\,\bigr)^2 + \bigl(\,\sigma\sin(k\,\Delta x)\,\bigr)^2 \\ &= 1 + 2\sigma(\sigma+1)\bigl(\,1 - \cos(k\,\Delta x)\,\bigr) = 1 + 4\sigma(\sigma+1)\sin^2\bigl(\tfrac{1}{2}\,k\,\Delta x\bigr) \le 1 \end{aligned}$$

for all k. Thus, stability requires that $\sigma(\sigma+1) \le 0$, and thus $-1 \le \sigma \le 0$, in complete accord with the CFL condition (5.41).

Upwind and Lax–Wendroff Schemes

To obtain a finite difference scheme that can be used for positive wave speeds, we replace the forward finite difference approximation to $\partial u/\partial x$ by the corresponding backwards difference quotient, namely, (5.1) with $h = -\,\Delta x$, leading to the alternative first-order numerical scheme

$$u_{j+1,m} = -\,(\sigma - 1)\,u_{j,m} + \sigma\,u_{j,m-1}, \tag{5.44}$$

where $\sigma = c\,\Delta t/\Delta x$ is as before. A similar analysis, left to the reader, produces the corresponding CFL stability criterion

$$0 \le \sigma = \frac{c\,\Delta t}{\Delta x} \le 1,$$

and so this scheme can be applied for suitable positive wave speeds.

In this manner, we have produced one numerical scheme that works for negative wave speeds, and an alternative scheme for positive speeds. The question arises — particularly when one is dealing with equations with variable wave speeds — whether one can devise a scheme that is (conditionally) stable for *both* positive and negative wave speeds. One might be tempted to use the centered difference approximation (5.6):

$$\frac{\partial u}{\partial x}\,(t_j, x_m) \approx \frac{u_{j,m+1} - u_{j,m-1}}{\Delta x} + \mathrm{O}\bigl(\,(\Delta x)^2\,\bigr). \tag{5.45}$$

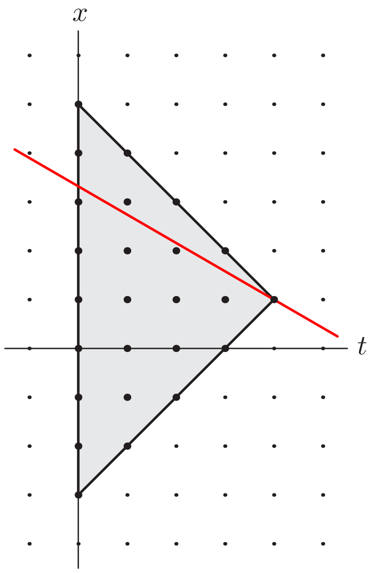


Figure 5.7. The CFL condition for the centered difference scheme.

Substituting (5.45) and the previous approximation to the time derivative (5.37) into (5.35) leads to the numerical scheme

$$u_{j+1,m} = -\tfrac{1}{2}\sigma\, u_{j,m+1} + u_{j,m} + \tfrac{1}{2}\sigma\, u_{j,m-1}, \tag{5.46}$$

where, as usual, $\sigma = c\,\Delta t/\Delta x$. In this case, the *numerical domain of dependence* of the node (t_j, x_m) consists of the nodes in the triangle

$$\widetilde{T}_{(t_j,x_m)} = \left\{ \, (t,x) \;\middle|\; 0 \le t \le t_j,\ x_m - t_j + t \le x \le x_m + t_j - t \, \right\}. \tag{5.47}$$

The CFL condition requires that, for $0 \le t \le t_j$, the characteristic going through (t_j, x_m) lie within this triangle, as in Figure 5.7, which imposes the condition

$$|\,\sigma\,| = \left| \frac{c\,\Delta t}{\Delta x} \right| \le 1, \qquad \text{or, equivalently,} \qquad |\,c\,| \le \frac{\Delta x}{\Delta t}\,. \tag{5.48}$$

Unfortunately, although it satisfies the CFL condition over this range of wave speeds, the centered difference scheme is, in fact, always *unstable*! For instance, the instability of the numerical solution to the preceding initial value problem (5.40) for $c = 1$ can be observed in Figure 5.8. This is confirmed by applying a von Neumann analysis: substitute (5.43) into (5.46), and cancel the common exponential factors. Provided $\sigma \ne 0$, which means that $c \ne 0$, the resulting magnification factor

$$\lambda = 1 - \,\mathrm{i}\,\sigma \sin(k\,\Delta x)$$

satisfies $|\,\lambda\,| > 1$ for all k with $\sin(k\,\Delta x) \ne 0$. Thus, for $c \ne 0$, the centered difference scheme (5.46) is unstable for all (nonzero) wave speeds!

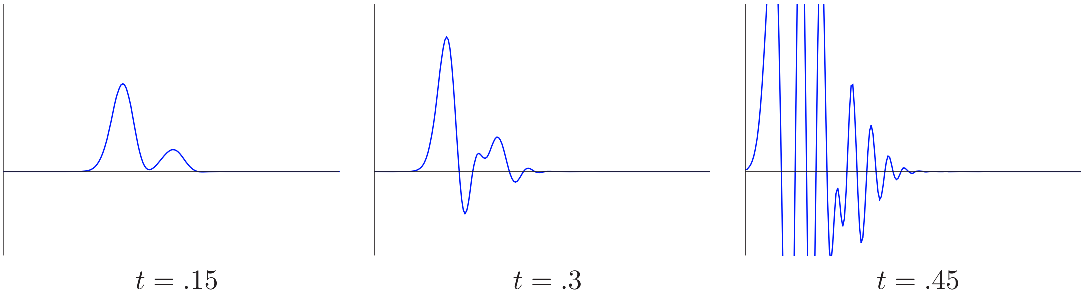


Figure 5.8. Centered difference numerical solution to the transport equation.

One possible means of overcoming the sign restriction on the wave speed is to use the forward difference scheme (5.38) when the wave speed is negative and the backwards scheme (5.44) when it is positive. The resulting scheme, valid for varying wave speeds $c(t,x)$, takes the form

$$u_{j+1,m} = \begin{cases} -\sigma_{j,m} u_{j,m+1} + (\sigma_{j,m}+1) u_{j,m}, & c_{j,m} \leq 0, \\ -(\sigma_{j,m}-1) u_{j,m} + \sigma_{j,m} u_{j,m-1}, & c_{j,m} > 0, \end{cases} \tag{5.49}$$

where

$$\sigma_{j,m} = c_{j,m} \frac{\Delta t}{\Delta x}, \qquad c_{j,m} = c(t_j, x_m). \tag{5.50}$$

This is referred to as an *upwind scheme*, since the second node always lies "upwind" — that is, away from the direction of motion — from the reference point (t_j, x_m). The upwind scheme works reasonably well over short time intervals, assuming that the space step size is sufficiently small and the time step satisfies the CFL condition $\Delta x / \Delta t \leq |\, c_{j,m} \,|$ at each node, cf. (5.41). However, over longer time intervals, as we already observed in Figure 5.5, the simple upwind scheme tends to produce a noticeable damping of waves or, alternatively, require an unacceptably small step size. One way of overcoming this defect is to use the popular *Lax–Wendroff scheme*, which is based on second-order approximations to the derivatives. In the case of constant wave speed, the iterative step takes the form

$$u_{j+1,m} = \tfrac{1}{2}\sigma(\sigma-1)\, u_{j,m+1} - (\sigma^2-1)\, u_{j,m} + \tfrac{1}{2}\sigma(\sigma+1)\, u_{j,m-1}. \tag{5.51}$$

The stability analysis of the Lax–Wendroff scheme is relegated to the exercises. Extensions to variable wave speeds are more subtle, and we refer the reader to [**80**] for a detailed derivation.

Exercises

5.3.1. Solve the initial value problem $u_t = 3 u_x$, $u(0,x) = 1/(1+x^2)$, on the interval $[-10, 10]$ using an upwind scheme with space step size $\Delta x = .1$. Decide on an appropriate time step size, and graph your solution at times $t = .5, 1, 1.5$. Discuss what you observe.

5.3.2. Solve Exercise 5.3.1 for the nonuniform transport equations

(*a*) $u_t + 4\,(1+x^2)^{-1}\,u_x = 0$, (*b*) $u_t = \left(3 - 2\,e^{-x^2/4}\right) u_x$,

(*c*) $u_t + 7\,x\,(1+x^2)^{-1}\,u_x = 0$, (*d*) $u_t + \left(2\tan^{-1}\tfrac{1}{2}\,x\right) u_x = 0$.

5.3.3. Consider the initial value problem

$$u_t + \frac{3\,x}{x^2+1}\,u_x = 0, \qquad u(0,x) = \left(1 - \tfrac{1}{2}\,x^2\right) e^{-x^2/3}.$$

On the interval $[-5,5]$, using space step size $\Delta x = .1$ and time step size $\Delta t = .025$, apply (*a*) the forward scheme (5.38) (suitably modified for variable wave speed), (*b*) the backward scheme (5.44) (suitably modified for variable wave speed), and (*c*) the upwind scheme (5.49). Graph the resulting numerical solutions at times $t = .5, 1, 1.5$, and discuss what you observe in each case. Which of the schemes are stable?

5.3.4. Use the centered difference scheme (5.46) to solve the initial value problem in Exercise 5.3.1. Do you observe any instabilities in your numerical solution?

5.3.5. Use the Lax–Wendroff scheme (5.51) to solve the initial value problem in Exercise 5.3.1. Discuss the accuracy of your solution in comparison with the upwind scheme.

♢ 5.3.6. Can you explain why, in Figure 5.5, the numerical solution in the case $c = -1$ is significantly better than for $c = -.5$, or, indeed, for any other c in the stable range.

5.3.7. Nonlinear transport equations are often solved numerically by writing them in the form of a conservation law, and then applying the finite difference formulas directly to the conserved density and flux. (*a*) Devise an upwind scheme for numerically solving our favorite nonlinear transport equation, $u_t + \frac{1}{2}\,(u^2)_x = 0$.

(*b*) Test your scheme on the initial value problem $u(0,x) = e^{-x^2}$.

5.3.8. (*a*) Design a stable numerical solution scheme for the damped transport equation $u_t + \frac{3}{4}\,u_x + u = 0$. (*b*) Test your scheme on the initial value problem with $u(0,x) = e^{-x^2}$.

♢ 5.3.9. Analyze the stability of the numerical scheme (5.44) by applying (*a*) the CFL condition; (*b*) a von Neumann analysis. Are your conclusions the same?

♢ 5.3.10. For what choices of step size $\Delta t, \Delta x$ is the Lax–Wendroff scheme (5.51) stable?

5.4 Numerical Algorithms for the Wave Equation

Let us now develop some basic numerical solution techniques for the second-order wave equation. As above, although we are in possession of the explicit d'Alembert solution formula (2.82), the lessons learned in designing viable schemes here will carry over to more complicated situations, including inhomogeneous media and higher-dimensional problems, for which analytic solution formulas may no longer be readily available.

Consider the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2\,\frac{\partial^2 u}{\partial x^2}\,, \qquad 0 < x < \ell, \qquad t \geq 0, \tag{5.52}$$

on a bounded interval of length ℓ with constant wave speed $c > 0$. For specificity, we impose (possibly time-dependent) Dirichlet boundary conditions

$$u(t,0) = \alpha(t), \qquad u(t,\ell) = \beta(t), \qquad t \geq 0, \tag{5.53}$$

along with the usual initial conditions

$$u(0,x) = f(x), \qquad \frac{\partial u}{\partial t}(0,x) = g(x), \qquad 0 \le x \le \ell. \tag{5.54}$$

As usual, we adopt a uniformly spaced mesh

$$t_j = j\,\Delta t, \qquad x_m = m\,\Delta x, \qquad \text{where} \qquad \Delta x = \frac{\ell}{n}\,.$$

Discretization is implemented by replacing the second-order derivatives in the wave equation by their standard finite difference approximations (5.5):

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2}(t_j,x_m) &\approx \frac{u(t_{j+1},x_m) - 2\,u(t_j,x_m) + u(t_{j-1},x_m)}{(\Delta t)^2} + \mathrm{O}\big((\Delta t)^2\big), \\ \frac{\partial^2 u}{\partial x^2}(t_j,x_m) &\approx \frac{u(t_j,x_{m+1}) - 2\,u(t_j,x_m) + u(t_j,x_{m-1})}{(\Delta x)^2} + \mathrm{O}\big((\Delta x)^2\big). \end{aligned} \tag{5.55}$$

Since the error terms are both of second order, we anticipate being able to choose the space and time step sizes to have comparable magnitudes: $\Delta t \approx \Delta x$. Substituting the finite difference formulas (5.55) into the partial differential equation (5.52) and rearranging terms, we are led to the iterative system

$$u_{j+1,m} = \sigma^2\,u_{j,m+1} + 2\,(1-\sigma^2)\,u_{j,m} + \sigma^2\,u_{j,m-1} - u_{j-1,m}, \qquad \begin{array}{l} j = 1,2,\ldots\,, \\ m = 1,\ldots,n-1, \end{array} \tag{5.56}$$

for the numerical approximations $u_{j,m} \approx u(t_j,x_m)$ to the solution values at the nodes. The parameter

$$\sigma = \frac{c\,\Delta t}{\Delta x} > 0 \tag{5.57}$$

depends on the wave speed and the ratio of space and time step sizes. The boundary conditions (5.53) require that

$$u_{j,0} = \alpha_j = \alpha(t_j), \qquad u_{j,n} = \beta_j = \beta(t_j), \qquad j = 0,1,2,\ldots\,. \tag{5.58}$$

This allows us to rewrite the iterative system in vectorial form

$$\mathbf{u}^{(j+1)} = B\,\mathbf{u}^{(j)} - \mathbf{u}^{(j-1)} + \mathbf{b}^{(j)}, \tag{5.59}$$

where

$$B = \begin{pmatrix} 2\,(1-\sigma^2) & \sigma^2 & & & \\ \sigma^2 & 2\,(1-\sigma^2) & \sigma^2 & & \\ & \sigma^2 & \ddots & \ddots & \\ & & \ddots & \ddots & \sigma^2 \\ & & & \sigma^2 & 2\,(1-\sigma^2) \end{pmatrix}, \quad \mathbf{u}^{(j)} = \begin{pmatrix} u_{j,1} \\ u_{j,2} \\ \vdots \\ u_{j,n-2} \\ u_{j,n-1} \end{pmatrix}, \quad \mathbf{b}^{(j)} = \begin{pmatrix} \sigma^2\,\alpha_j \\ 0 \\ \vdots \\ 0 \\ \sigma^2\,\beta_j \end{pmatrix}. \tag{5.60}$$

The entries of $\mathbf{u}^{(j)} \in \mathbb{R}^{n-1}$ are, as in (5.18), the numerical approximations to the solution values at the *interior* nodes. Note that (5.59) describes a *second-order iterative scheme*, since computing the subsequent iterate $\mathbf{u}^{(j+1)}$ requires knowing the values of the preceding two: $\mathbf{u}^{(j)}$ and $\mathbf{u}^{(j-1)}$.

The one subtlety is how to get the method started. We know $\mathbf{u}^{(0)}$, since its entries $u_{0,m} = f_m = f(x_m)$ are determined by the initial position. However, we also need $\mathbf{u}^{(1)}$

in order to launch the iteration and compute $\mathbf{u}^{(2)}, \mathbf{u}^{(3)}, \ldots$. Its entries $u_{1,m} \approx u(\Delta t, x_m)$ approximate the solution at time $t_1 = \Delta t$, whereas the initial velocity $u_t(0,x) = g(x)$ prescribes the derivatives $u_t(0,x_m) = g_m = g(x_m)$ at the initial time $t_0 = 0$. To resolve this difficulty, a first thought might be to use the finite difference approximation

$$g_m = \frac{\partial u}{\partial t}(0, x_m) \approx \frac{u(\Delta t, x_m) - u(0, x_m)}{\Delta t} \approx \frac{u_{1,m} - f_m}{\Delta t} \tag{5.61}$$

to compute the required values $u_{1,m} = f_m + g_m \, \Delta t$. However, the approximation (5.61) is accurate only to order Δt, whereas the rest of the scheme has errors proportional to $(\Delta t)^2$. The effect would be to introduce an unacceptably large error at the initial step, and the resulting solution would fail to conform to the desired order of accuracy.

To construct an initial approximation to $\mathbf{u}^{(1)}$ with error on the order of $(\Delta t)^2$, we need to analyze the error in the approximation (5.61) in more depth. Note that, by Taylor's Theorem,

$$\begin{aligned} \frac{u(\Delta t, x_m) - u(0, x_m)}{\Delta t} &= \frac{\partial u}{\partial t}(0, x_m) + \frac{1}{2}\frac{\partial^2 u}{\partial t^2}(0, x_m)\Delta t + \mathrm{O}\big((\Delta t)^2\big) \\ &= \frac{\partial u}{\partial t}(0, x_m) + \frac{c^2}{2}\,\frac{\partial^2 u}{\partial x^2}(0, x_m)\,\Delta t + \mathrm{O}\big((\Delta t)^2\big), \end{aligned}$$

since $u(t,x)$ solves the wave equation. Therefore,

$$\begin{aligned} u_{1,m} = u(\Delta t, x_m) &\approx u(0, x_m) + \frac{\partial u}{\partial t}(0, x_m)\Delta t + \frac{c^2}{2}\,\frac{\partial^2 u}{\partial x^2}(0, x_m)(\Delta t)^2 \\ &= f(x_m) + g(x_m)\,\Delta t + \frac{c^2}{2}\, f''(x_m)(\Delta t)^2 \\ &\approx f_m + g_m\,\Delta t + \frac{c^2(f_{m+1} - 2\,f_m + f_{m-1})(\Delta t)^2}{2\,(\Delta x)^2}\,, \end{aligned}$$

where the last line, which employs the finite difference approximation (5.5) to the second derivative, can be used if the explicit formula for $f''(x)$ is either not known or too complicated to evaluate directly. Therefore, we initiate the scheme by setting

$$u_{1,m} = \tfrac{1}{2}\sigma^2 f_{m+1} + (1 - \sigma^2) f_m + \tfrac{1}{2}\sigma^2 f_{m-1} + g_m\,\Delta t, \tag{5.62}$$

or, in vectorial form,

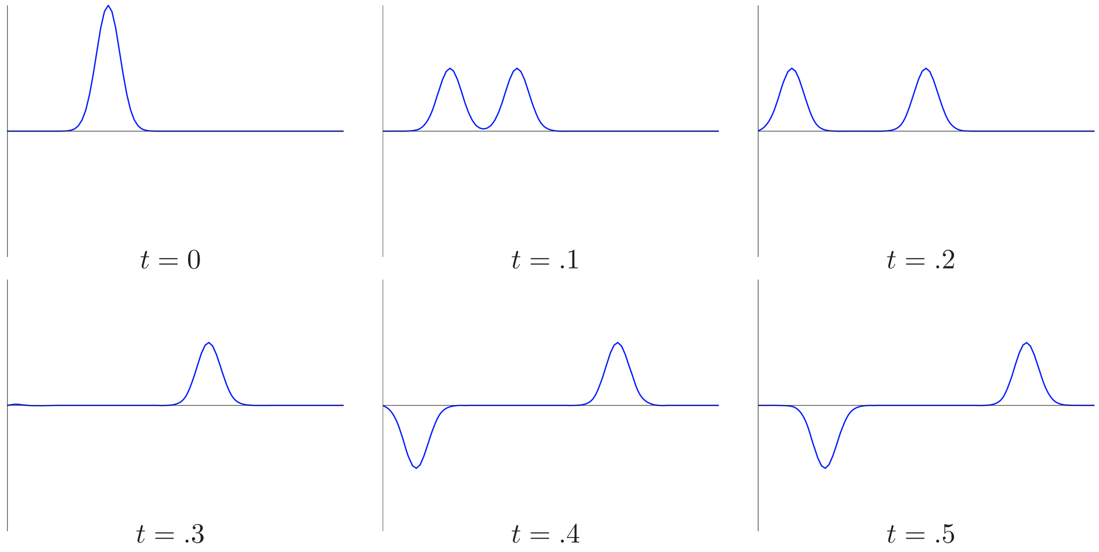
$$\mathbf{u}^{(0)} = \mathbf{f}, \qquad \mathbf{u}^{(1)} = \tfrac{1}{2} B\,\mathbf{u}^{(0)} + \mathbf{g}\,\Delta t + \tfrac{1}{2}\mathbf{b}^{(0)}, \tag{5.63}$$

where $\mathbf{f} = \big(f_1, f_2, \ldots, f_{n-1}\big)^T$, $\mathbf{g} = \big(g_1, g_2, \ldots, g_{n-1}\big)^T$, are the sampled values of the initial data. This serves to maintain the desired second-order accuracy of the scheme.

Example 5.6. Consider the particular initial value problem

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

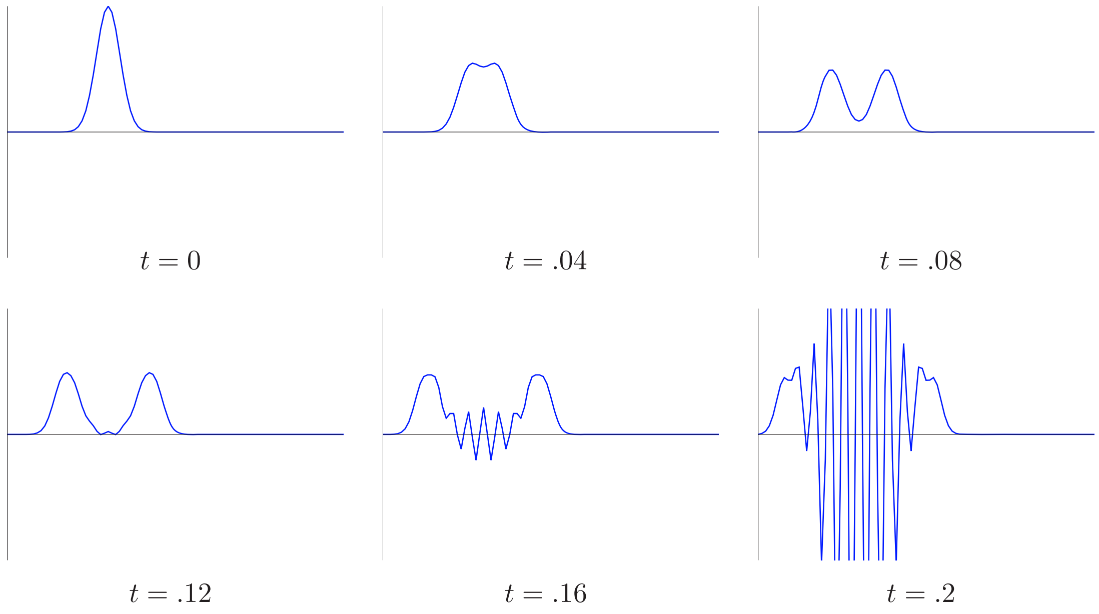
subject to homogeneous Dirichlet boundary conditions on the interval $[0,1]$. The initial data is a fairly concentrated hump centered at $x = .3$. As time progresses, we expect the initial hump to split into two half-sized humps, which then collide with the ends of the interval, reversing direction and orientation.



$t = 0$ $t = .1$ $t = .2$

$t = .3$ $t = .4$ $t = .5$

Figure 5.9. Numerically stable waves. ⊎



$t = 0$ $t = .04$ $t = .08$

$t = .12$ $t = .16$ $t = .2$

Figure 5.10. Numerically unstable waves. ⊎

For our numerical approximation, let us use a space discretization consisting of 90 equally spaced points, and so $\Delta x = \frac{1}{90} = .0111\ldots$. If we choose a time step of $\Delta t = .01$, whereby $\sigma = .9$, then we obtain a reasonably accurate solution over a fairly long time range, as plotted in Figure 5.9. On the other hand, if we double the time step, setting $\Delta t = .02$, so $\sigma = 1.8$, then, as shown in Figure 5.10, we induce an instability that eventually

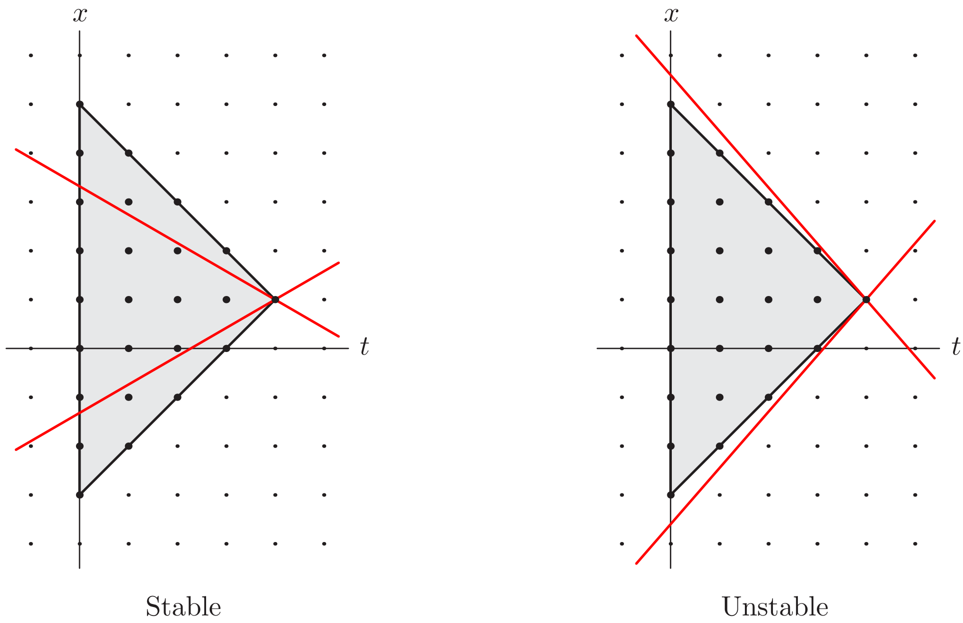


Figure 5.11. The CFL condition for the wave equation.

overwhelms the numerical solution. Thus, the preceding numerical scheme appears to be only conditionally stable.

Stability analysis proceeds along the same lines as in the first-order case. The CFL condition requires that the characteristics emanating from a node (t_j, x_m) remain, for times $0 \le t \le t_j$, in its numerical domain of dependence, which, for our particular numerical scheme, is the same triangle

$$\widetilde{T}_{(t_j, x_m)} = \left\{ (t, x) \;\middle|\; 0 \le t \le t_j,\ x_m - t_j + t \le x \le x_m + t_j - t \right\},$$

now plotted in Figure 5.11. Since the characteristics are the lines of slope $\pm c$, the CFL condition is the same as in (5.48):

$$\sigma = \frac{c\,\Delta t}{\Delta x} \le 1, \qquad \text{or, equivalently,} \qquad 0 < c \le \frac{\Delta x}{\Delta t}. \tag{5.64}$$

The resulting stability criterion explains the observed difference between the numerically stable and unstable cases.

However, as we noted above, the CFL condition is, in general, only necessary for stability of the numerical scheme; sufficiency requires that we perform a von Neumann stability analysis. To this end, we specialize the calculation to a single complex exponential $e^{\,\mathrm{i}\,k\,x}$. After one time step, the scheme will have the effect of multiplying it by the *magnification factor* $\lambda = \lambda(k)$, after another time step by λ^2, and so on. To determine λ, we substitute the relevant sampled exponential values

$$u_{j-1,m} = e^{\,\mathrm{i}\,k\,x_m}, \qquad u_{j,m} = \lambda\, e^{\,\mathrm{i}\,k\,x_m}, \qquad u_{j+1,m} = \lambda^2\, e^{\,\mathrm{i}\,k\,x_m}, \tag{5.65}$$

into the scheme (5.56). After canceling the common exponential, we find that the magnification factor satisfies the following quadratic equation:

$$\lambda^2 = \left(2 - 4\sigma^2 \sin^2\left(\tfrac{1}{2}\,k\,\Delta x\right) \right)\lambda - 1,$$

whence

$$\lambda = \alpha \pm \sqrt{\alpha^2 - 1}\,, \qquad \text{where} \qquad \alpha = 1 - 2\sigma^2 \sin^2\left(\tfrac{1}{2}\,k\,\Delta x\right). \tag{5.66}$$

Thus, there are *two* different magnification factors associated with each complex exponential — which is, in fact, a consequence of the scheme being of second order. Stability requires that *both* be ≤ 1 in modulus. Now, if the CFL condition (5.64) holds, then $|\,\alpha\,| \leq 1$, which implies that both magnification factors (5.66) are complex numbers of modulus $|\,\lambda\,| = 1$, and thus the numerical scheme satisfies the stability criterion (5.26). On the other hand, if $\sigma > 1$, then $\alpha < -1$ over a range of values of k, which implies that the two magnification factors (5.66) are both real and one of them is < -1, thus violating the stability criterion. Consequently, the CFL condition (5.64) does indeed distinguish between the stable and unstable finite difference schemes for the wave equation.

Exercises

5.4.1. Suppose you are asked to numerically approximate the solution to the initial-boundary value problem

$$u_{tt} = 64\,u_{xx}, \quad u(t,0) = u(t,3) = 0, \quad u(0,x) = \begin{cases} 1 - 2|\,x - 1\,|, & \tfrac{1}{2} \leq x \leq \tfrac{3}{2}, \\ 0, & \text{otherwise,} \end{cases} \quad u_t(0,x) = 0,$$

on the interval $0 \leq x \leq 3$, using (5.56) with space step size $\Delta x = .1$. (*a*) What range of time steps Δt are allowed? (*b*) Test your answer by implementing the numerical solution for one value of Δt in the allowable range and one value outside. Discuss what you observe in your numerical solutions. (*c*) In the stable range, compare your numerical solution with that obtained using the smaller step size $\Delta x = .01$ and a suitable time step Δt.

5.4.2. Solve Exercise 5.4.1 for the boundary value problem

$$u_{tt} = 64\,u_{xx}, \quad u(t,0) = 0 = u(t,3), \quad u(0,x) = 0, \quad u_t(0,x) = \begin{cases} 1 - 2|\,x - 1\,|, & \tfrac{1}{2} \leq x \leq \tfrac{3}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

5.4.3. Solve the following initial-boundary value problem

$$u_{tt} = 9\,u_{xx}, \quad u(t,0) = u(t,1) = 0, \quad u(0,x) = \tfrac{1}{2} + \left|\,x - \tfrac{1}{4}\,\right| - \left|\,2x - \tfrac{3}{4}\,\right|, \quad u_t(0,x) = 0,$$

on the interval $0 \leq x \leq 1$, using the numerical scheme (5.56) with space step sizes $\Delta x = .1, .01$ and $.001$ and suitably chosen time steps. Discuss which features of the solution can be observed in your numerical approximations.

5.4.4. (*a*) Use a numerical integrator with space step size $\Delta x = .05$ to solve the periodically forced boundary value problem

$$u_{tt} = u_{xx}, \qquad u(0,x) = u_t(0,x) = 0, \qquad u(t,0) = \sin t, \qquad u(t,1) = 0.$$

Is your solution periodic? (*b*) Repeat the computation using the alternative boundary condition $u(t,0) = \sin \pi\, t$. Discuss any observed differences between the two problems.

5.4.5. (*a*) Design an explicit numerical scheme for solving the initial-boundary value problem

$$u_{tt} = c^2 u_{xx} + F(t,x), \quad u(t,0) = u(t,1) = 0, \quad u(0,x) = f(x), \quad u_t(0,x) = g(x), \quad 0 \leq x \leq 1,$$

for the wave equation with an external *forcing term* $F(t,x)$. Clearly state any stability conditions that need to be imposed on the time and space step sizes.

(*b*) Test your scheme on the particular case $c = \tfrac{1}{4}$, $F(t,x) = 3\,\text{sign}\left(x - \tfrac{1}{2}\right) \sin \pi\, t$, $f(x) \equiv g(x) \equiv 0$, using space step sizes $\Delta x = .05$ and $.01$, and suitably chosen time steps.

5.4.6. Let $\beta > 0$. (a) Design a finite difference scheme for approximating the solution to the initial-boundary value problem

$$u_{tt} + \beta\, u_t = c^2 u_{xx}, \qquad u(t,0) = u(t,1) = 0, \qquad u(0,x) = f(x), \qquad u_t(0,x) = g(x),$$

for the damped wave equation on the interval $0 \leq x \leq 1$. (b) Discuss the stability of your scheme. What choice of step sizes will ensure stability? (c) Test your scheme with $c = 1$, $\beta = 1$, using the initial data $f(x) = e^{-(x-.7)^2}$, $g(x) = 0$.

5.5 Finite Difference Algorithms for the Laplace and Poisson Equations

Finally, let us discuss the implementation of finite diffference numerical schemes for elliptic boundary value problems. We concentrate on the simplest cases: the two-dimensional Laplace and Poisson equations. The basic issues are already apparent in this particular context, and extensions to more general equations, higher dimensions, and higher-order schemes are all reasonably straightforward. In Chapter 10, we will present a competitor — the renowned finite element method — which, while relying on more sophisticated mathematical machinery, enjoys several advantages, including more immediate adaptability to variable mesh sizes and more sophisticated geometries.

For specificity, we concentrate on the Dirichlet boundary value problem

$$\begin{aligned} -\Delta u = -u_{xx} - u_{yy} &= f(x,y), \\ u(x,y) &= g(x,y), \end{aligned} \qquad \text{for} \qquad \begin{aligned} &(x,y) \in \Omega, \\ &(x,y) \in \partial\Omega, \end{aligned} \tag{5.67}$$

on a bounded planar domain $\Omega \subset \mathbb{R}^2$. The first step is to discretize the domain Ω by constructing a rectangular mesh. Thus, the finite difference method is particularly suited to domains whose boundary lines up with the coordinate axes; otherwise, the mesh nodes do not, generally, lie exactly on $\partial\Omega$, making the approximation of the boundary data more challenging — although not insurmountable.

For simplicity, let us study the case in which

$$\Omega = \{\, a < x < b,\ c < y < d \,\}$$

is a rectangle. We introduce a regular rectanglar mesh, with x and y spacings given, respectively, by

$$\Delta x = \frac{b-a}{m}\,, \qquad \Delta y = \frac{c-d}{n}\,,$$

for positive integers m, n. Thus, the interior of the rectangle contains $(m-1)(n-1)$ *interior nodes*

$$(x_i, y_j) = (a + i\,\Delta x, c + j\,\Delta y) \qquad \text{for} \qquad 0 < i < m, \quad 0 < j < n.$$

In addition, the $2m + 2n$ *boundary nodes* $(x_0, y_j) = (a, y_j)$, $(x_m, y_j) = (b, y_j)$, $(x_i, y_0) = (x_i, c)$, $(x_i, y_n) = (x_i, d)$, lie on the boundary of the rectangle.

At each interior node, we employ the centered difference formula (5.5) to approximate the relevant second-order derivatives:

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2}(x_i, y_j) &= \frac{u(x_{i+1}, y_j) - 2\,u(x_i, y_j) + u(x_{i-1}, y_j)}{(\Delta x)^2} + \mathrm{O}\big((\Delta x)^2\big), \\ \frac{\partial^2 u}{\partial y^2}(x_i, y_j) &= \frac{u(x_i, y_{j+1}) - 2\,u(x_i, y_j) + u(x_i, y_{j-1})}{(\Delta y)^2} + \mathrm{O}\big((\Delta y)^2\big). \end{aligned} \tag{5.68}$$

Substituting these finite difference formulae into the Poisson equation produces the linear system

$$-\frac{u_{i+1,j}-2\,u_{i,j}+u_{i-1,j}}{(\Delta x)^2}-\frac{u_{i,j+1}-2\,u_{i,j}+u_{i,j-1}}{(\Delta y)^2}=f_{i,j},\qquad \begin{aligned} i&=1,\ldots,m-1,\\ j&=1,\ldots,n-1,\end{aligned} \tag{5.69}$$

in which $u_{i,j}$ denotes our numerical approximation to the solution values $u(x_i,y_j)$ at the nodes, while $f_{i,j}=f(x_i,y_j)$. If we set

$$\rho=\frac{\Delta x}{\Delta y}\,, \tag{5.70}$$

then (5.69) can be rewritten in the form

$$\begin{aligned} 2\,(1+\rho^2)u_{i,j}-(u_{i-1,j}+u_{i+1,j})-\rho^2(u_{i,j-1}+u_{i,j+1})=(\Delta x)^2 f_{i,j},&\\ i=1,\ldots,m-1,\qquad j=1,\ldots,n-1.&\end{aligned} \tag{5.71}$$

Since both finite difference approximations (5.68) are of second order, one should choose Δx and Δy to be of comparable size, thus keeping ρ around 1.

The linear system (5.71) forms the finite difference approximation to the Poisson equation at the interior nodes. It is supplemented by the discretized Dirichlet boundary conditions

$$\begin{aligned} u_{i,0}&=g_{i,0}, &\quad u_{i,n}&=g_{i,n}, &\quad i&=0,\ldots,m,\\ u_{0,j}&=g_{0,j}, &\quad u_{m,j}&=g_{m,j}, &\quad j&=0,\ldots,n.\end{aligned} \tag{5.72}$$

These boundary values can be substituted directly into the system, making (5.71) a system of $(m-1)(n-1)$ linear equations involving the $(m-1)(n-1)$ unknowns $u_{i,j}$ for $1\le i\le m-1$, $1\le j\le n-1$. We impose some convenient ordering for these entries, e.g., from left to right and then bottom to top, forming the column vector of unknowns

$$\begin{aligned}\mathbf{w}&=(w_1,w_2,\ \ldots\ ,w_{(m-1)(n-1)})^T\\ &=(u_{1,1},u_{2,1},\ldots,u_{m-1,1},u_{1,2},u_{2,2},\ldots,u_{m-1,2},u_{1,3},\ldots,u_{m-1,n-1})^T.\end{aligned} \tag{5.73}$$

The combined linear system (5.71–72) can then be rewritten in matrix form

$$A\,\mathbf{w}=\widehat{\mathbf{f}}, \tag{5.74}$$

where the right-hand side is obtained by combining the column vector $\mathbf{f}=(\ \ldots\ f_{i,j}\ \ldots\ )^T$ with the boundary data provided by (5.72) according to where they appear in the system. The implementation will become clearer once we work through a small-scale example.

Example 5.7. To better understand how the process works, let us look at the case in which $\Omega=\{\,0<x<1,\ 0<y<1\,\}$ is the unit square. In order to write everything in full detail, we start with a very coarse mesh with $\Delta x=\Delta y=\frac{1}{4}$; see Figure 5.12. Thus $m=n=4$, resulting in a total of nine interior nodes. In this case, $\rho=1$, and hence the

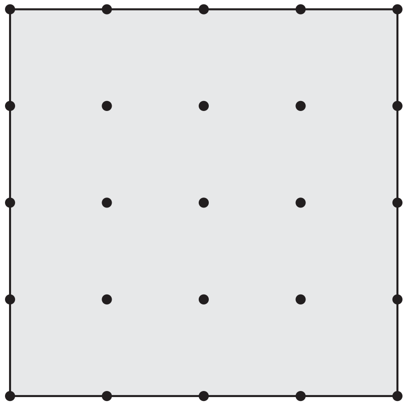


Figure 5.12. Square mesh with $\Delta x = \Delta y = \frac{1}{4}$.

finite difference system (5.71) consists of the following nine equations:

$$\begin{aligned}
-u_{1,0} - u_{0,1} + 4u_{1,1} - u_{2,1} - u_{1,2} &= \tfrac{1}{16}\, f_{1,1},\\
-u_{2,0} - u_{1,1} + 4u_{2,1} - u_{3,1} - u_{2,2} &= \tfrac{1}{16}\, f_{2,1},\\
-u_{3,0} - u_{2,1} + 4u_{3,1} - u_{4,1} - u_{3,2} &= \tfrac{1}{16}\, f_{3,1},\\
-u_{1,1} - u_{0,2} + 4u_{1,2} - u_{2,2} - u_{1,3} &= \tfrac{1}{16}\, f_{1,2},\\
-u_{2,1} - u_{1,2} + 4u_{2,2} - u_{3,2} - u_{2,3} &= \tfrac{1}{16}\, f_{2,2},\\
-u_{3,1} - u_{2,2} + 4u_{3,2} - u_{4,2} - u_{3,3} &= \tfrac{1}{16}\, f_{3,2},\\
-u_{1,2} - u_{0,3} + 4u_{1,3} - u_{2,3} - u_{1,4} &= \tfrac{1}{16}\, f_{1,3},\\
-u_{2,2} - u_{1,3} + 4u_{2,3} - u_{3,3} - u_{2,4} &= \tfrac{1}{16}\, f_{2,3},\\
-u_{3,2} - u_{2,3} + 4u_{3,3} - u_{4,3} - u_{3,4} &= \tfrac{1}{16}\, f_{3,3}.
\end{aligned} \tag{5.75}$$

(Note that the values at the four corner nodes, $u_{0,0}, u_{4,0}, u_{0,4}, u_{4,4}$, do not appear.) The boundary data imposes the additional conditions (5.72), namely

$$\begin{matrix}
u_{0,1} = g_{0,1}, & u_{0,2} = g_{0,2}, & u_{0,3} = g_{0,3}, & u_{1,0} = g_{1,0}, & u_{2,0} = g_{2,0}, & u_{3,0} = g_{3,0},\\
u_{4,1} = g_{4,1}, & u_{4,2} = g_{4,2}, & u_{4,3} = g_{4,3}, & u_{1,4} = g_{1,4}, & u_{2,4} = g_{2,4}, & u_{3,4} = g_{3,4}.
\end{matrix}$$

The system (5.75) can be written in matrix form $A\mathbf{w} = \widehat{\mathbf{f}}$, where

$$A = \begin{pmatrix}
4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\
-1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0\\
0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0\\
-1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0\\
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0\\
0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1\\
0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0\\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1\\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4
\end{pmatrix}, \tag{5.76}$$

and

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \end{pmatrix} = \begin{pmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{1,2} \\ u_{2,2} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{pmatrix}, \qquad \widehat{\mathbf{f}} = \begin{pmatrix} \frac{1}{16} f_{1,1} + g_{1,0} + g_{0,1} \\ \frac{1}{16} f_{2,1} + g_{2,0} \\ \frac{1}{16} f_{3,1} + g_{3,0} + g_{4,1} \\ \frac{1}{16} f_{1,2} + g_{0,2} \\ \frac{1}{16} f_{2,2} \\ \frac{1}{16} f_{3,2} + g_{4,2} \\ \frac{1}{16} f_{1,3} + g_{0,3} + g_{1,4} \\ \frac{1}{16} f_{2,3} + g_{2,4} \\ \frac{1}{16} f_{3,3} + g_{4,3} + g_{3,4} \end{pmatrix}.$$

Note that the known boundary values, namely $u_{i,j} = g_{i,j}$ when i or j equals 0 or 4, have been incorporated into the right-hand side $\widehat{\mathbf{f}}$ of the finite difference linear system (5.74). The resulting linear system is easily solved by Gaussian Elimination, [**89**]. Finer meshes lead to correspondingly larger linear systems, all endowed with a common overall structure, as discussed below.

For example, the function

$$u(x,y) = y \sin(\pi x)$$

solves the particular boundary value problem

$$-\Delta u = \pi^2 y \sin(\pi x), \quad u(x,0) = u(0,y) = u(1,y) = 0, \quad u(x,1) = \sin(\pi x), \quad 0 < x, y < 1.$$

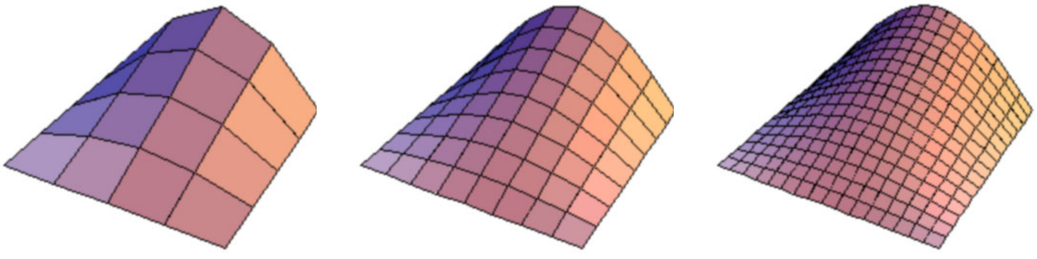
Setting up and solving the linear system (5.75) produces the finite difference solution values

$$\begin{array}{lll} u_{1,1} = .1831, & u_{1,2} = .2589, & u_{1,3} = .1831, \\ u_{2,1} = .3643, & u_{2,2} = .5152, & u_{2,3} = .3643, \\ u_{3,1} = .5409, & u_{3,2} = .7649, & u_{3,3} = .5409, \end{array}$$

leading to the numerical approximation plotted in the first graph[†] of Figure 5.13. The maximal error between the numerical and exact solution values is .01520, which occurs at the center of the square. In the second and third graphs, the mesh spacing is successively reduced by half, so there are, respectively, $m = n = 8$ and 16 nodes in each coordinate direction. The corresponding maximal numerical errors at the nodes are .004123 and .001035. Observe that halving the step size reduces the error by a factor of $\frac{1}{4}$, which is consistent with the numerical scheme being of second order.

Remark: The preceding test is a particular instance of the method of *manufactured solutions*, in which one starts with a preselected function that almost certainly is not a solution to the exact problem at hand. Nevertheless, substituting this function into the differential equation and the relevant initial and/or boundary conditions leads to an inhomogeneous problem of the same character as the original. After running the numerical scheme on the modified problem, one can test for accuracy by comparing the numerical output with the preselected function.

[†] We are using flat triangles to interpolate the nodal data. Smoother interpolation schemes, e.g., splines, [**102**], will produce a more realistic reproduction of the analytic solution graph.



$\Delta x = \Delta y = .25$ $\Delta x = \Delta y = .125$ $\Delta x = \Delta y = .0625$

Figure 5.13. Finite difference solutions to a Poisson boundary value problem.

Solution Strategies

The linear algebraic system resulting from a finite difference discretization can be rather large, and it behooves us to devise efficient solution strategies. The general finite difference coefficient matrix A has a very structured form, which can already be inferred from the very simple case (5.76). When the underlying domain is a rectangle, it assumes a *block tridiagonal form*

$$A = \begin{pmatrix} B_\rho & -\rho^2\,\mathrm{I} & & & & \\ -\rho^2\,\mathrm{I} & B_\rho & -\rho^2\,\mathrm{I} & & & \\ & -\rho^2\,\mathrm{I} & B_\rho & -\rho^2\,\mathrm{I} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\rho^2\,\mathrm{I} & B_\rho & -\,\rho^2\,\mathrm{I} \\ & & & & -\rho^2\,\mathrm{I} & B_\rho \end{pmatrix}, \tag{5.77}$$

where I is the $(m-1)\times(m-1)$ identity matrix, while

$$B_\rho = \begin{pmatrix} 2\,(1+\rho^2) & -\rho^2 & & & & & \\ -\rho^2 & 2\,(1+\rho^2) & -\rho^2 & & & & \\ & -\rho^2 & 2\,(1+\rho^2) & -\rho^2 & & & \\ & & -\rho^2 & 2\,(1+\rho^2) & -\rho^2 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -\rho^2 & 2\,(1+\rho^2) & -\,\rho^2 \\ & & & & & -\rho^2 & 2\,(1+\rho^2) \end{pmatrix} \tag{5.78}$$

is itself an $(m-1)\times(m-1)$ tridiagonal matrix. (Here and below, all entries not explicitly indicated are zero.) There are $n-1$ blocks in both the row and column directions.

When the finite difference linear system is of moderate size, it can be efficiently solved by Gaussian Elimination, which effectively factorizes $A = LU$ into a product of lower and upper triangular matrices. (This follows since A is symmetric and nonsingular, as guaranteed by Theorem 5.8 below.) In the present case, the factors are *block bidiagonal* matrices:

$$L = \begin{pmatrix} \mathrm{I} & & & & & \\ L_1 & \mathrm{I} & & & & \\ & L_2 & \mathrm{I} & & & \\ & & \ddots & \ddots & & \\ & & & L_{n-3} & \mathrm{I} & \\ & & & & L_{n-2} & \mathrm{I} \end{pmatrix},$$
$$U = \begin{pmatrix} U_1 & -\rho^2\,\mathrm{I} & & & & \\ & U_2 & -\rho^2\,\mathrm{I} & & & \\ & & U_3 & -\rho^2\,\mathrm{I} & & \\ & & & \ddots & \ddots & \\ & & & & U_{n-2} & -\rho^2\,\mathrm{I} \\ & & & & & U_{n-1} \end{pmatrix}, \tag{5.79}$$

where the individual blocks are again of size $(m-1)\times(m-1)$. Indeed, multiplying out the matrix product $L U$ and equating the result to (5.77) leads to the iterative matrix system

$$U_1 = B_\rho, \qquad L_j = -\rho^2\, U_j^{-1}, \qquad U_{j+1} = B_\rho + \rho^2 L_j, \qquad j = 1,\ldots,n-2, \tag{5.80}$$

which produces the individual blocks.

With the $L U$ factors in place, we can apply Forward and Back Substitution to solve the block tridiagonal linear system $A\,\mathbf{w} = \widehat{\mathbf{f}}$ by solving the block lower and upper triangular systems

$$L\,\mathbf{z} = \widehat{\mathbf{f}}, \qquad U\,\mathbf{w} = \mathbf{z}. \tag{5.81}$$

In view of the forms (5.79) of L and U, if we write

$$\mathbf{w} = \begin{pmatrix} \mathbf{w}^{(1)} \\ \mathbf{w}^{(2)} \\ \vdots \\ \mathbf{w}^{(n-1)} \end{pmatrix}, \qquad \mathbf{z} = \begin{pmatrix} \mathbf{z}^{(1)} \\ \mathbf{z}^{(2)} \\ \vdots \\ \mathbf{z}^{(n-1)} \end{pmatrix}, \qquad \widehat{\mathbf{f}} = \begin{pmatrix} \widehat{\mathbf{f}}^{(1)} \\ \widehat{\mathbf{f}}^{(2)} \\ \vdots \\ \widehat{\mathbf{f}}^{(n-1)} \end{pmatrix},$$

so that each $\mathbf{w}^{(j)}, \mathbf{z}^{(j)}, \widehat{\mathbf{f}}^{(j)}$, is a vector with $m-1$ entries, then we must successively solve

$$\begin{aligned} &\mathbf{z}^{(1)} = \widehat{\mathbf{f}}^{(1)}, && \mathbf{z}^{(j+1)} = \widehat{\mathbf{f}}^{(j+1)} - L_j \mathbf{z}^{(j)}, && j = 1,2,\ldots,n-2, \\ &\mathbf{w}^{(n-1)} = \mathbf{z}^{(n-1)}, && U_j \mathbf{w}^{(k)} = \mathbf{z}^{(k)} - \rho^2\,\mathbf{w}^{(k+1)}, && k = n-2, n-3, \ldots, 1, \end{aligned} \tag{5.82}$$

in the prescribed order. In view of the identification of L_j with $-\rho^2$ times the inverse of U_j, the last set of equations in (5.82) is perhaps better written as

$$\mathbf{w}^{(k)} = L_j \big(\mathbf{w}^{(k+1)} - \rho^{-2}\,\mathbf{z}^{(k)} \big), \qquad k = n-2, n-3, \ldots, 1. \tag{5.83}$$

As the number of nodes becomes large, the preceding elimination/factorization approach to solving the linear system becomes increasingly inefficient, and one often switches to an iterative solution method such as Gauss–Seidel, Jacobi, or, even better, Successive Over–Relaxation (SOR); indeed, SOR was originally designed to speed up the solution of the large-scale linear systems arising from the numerical solution of elliptic partial differential equations. Detailed discussions of iterative matrix methods can be found in

[**89**; Chapter 10] and [**118**]. For the SOR method, a good choice for the relaxation parameter is

$$\omega = \frac{4}{2 + \sqrt{4 - \cos^2(\pi/m) - \cos^2(\pi/n)}} \,. \tag{5.84}$$

Iterative solution methods are even more attractive in dealing with irregular domains, whose finite difference coefficient matrix, while still sparse, is less structured than in the rectangular case, and hence less amenable to fast Gaussian Elimination algorithms.

Finally, let us address the question of unique solvability of the finite difference linear system obtained by discretization of the Poisson equation on a bounded domain subject to Dirichlet boundary conditions. As in the Uniqueness Theorem 4.10 for the original boundary value, this will follow from an easily established Maximum Principle for the discrete system that directly mimics the Laplace equation maximum principle of Theorem 4.9.

Theorem 5.8. *Let Ω be a bounded domain. Then the finite difference linear system* (5.74) *has a unique solution.*

Proof: The result will follow if we can prove that the only solution to the corresponding homogeneous linear system $A\mathbf{w} = \mathbf{0}$ is the trivial solution $\mathbf{w} = \mathbf{0}$. The homogeneous system corresponds to discretizing the Laplace equation subject to zero Dirichlet boundary conditions.

Now, in view of (5.71), each equation in the homogeneous linear system can be written in the form

$$u_{i,j} = \frac{u_{i-1,j} + u_{i+1,j} + \rho^2 u_{i,j-1} + \rho^2 u_{i,j+1}}{2\,(1+\rho^2)} \,. \tag{5.85}$$

If $\rho = 1$, then (5.85) says that the value of $u_{i,j}$ at the node (x_i, y_j) is equal to the *average* of the values at the four neighboring nodes. For general ρ, it says that $u_{i,j}$ is a *weighted average* of the four neighboring values. In either case, the value of $u_{i,j}$ must lie *strictly* between the maximum and minimum values of $u_{i-1,j}, u_{i+1,j}, u_{i,j-1}$ and $u_{i,j+1}$ — unless all these values are the same, in which case $u_{i,j}$ also has the same value. This observation suffices to establish a *Maximum Principle* for the finite difference system for the Laplace equation — namely, that its solution cannot achieve a local maximum or minimum at an interior node.

Now suppose that the homogeneous finite difference system $A\mathbf{w} = \mathbf{0}$ for the domain has a nontrivial solution $\mathbf{w} \neq \mathbf{0}$. Let $u_{i,j} = w_k$ be the maximal entry of this purported solution. The Maximum Principle requires that all four of its neighboring values must have the *same* maximal value. But then the same argument applies to the neighbors of those entries, to their neighbors, and so on. Eventually one of the neighbors is at a boundary node, but, since we are dealing with the homogeneous Dirichlet boundary value problem, its value is zero. This immediately implies that all the entries of $\mathbf{w}$ must be zero, which is a contradiction. *Q.E.D.*

Rigorously establishing convergence of the finite difference solution to the analytic solution to the boundary value problem as the step size goes to zero will not be discussed here, and we refer the reader to [**6**, **80**] for precise results and proofs.

Exercises

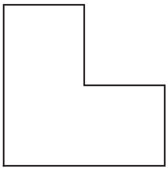
♠ 5.5.1. Solve the Dirichlet problem $\Delta u = 0$, $u(x,0) = \sin^3 x$, $u(x,\pi) = 0$, $u(0,y) = 0$, $u(\pi,y) = 0$, numerically using a finite difference scheme. Compare your approximation with the solution you obtained in Exercise 4.3.10(*a*).

♠ 5.5.2. Solve the Dirichlet problem $\Delta u = 0$, $u(x,0) = x$, $u(x,1) = 1 - x$, $u(0,y) = y$, $u(1,y) = 1 - y$, numerically via finite differences. Compare your approximation with the solution you obtained in Exercise 4.3.12(*d*).

♠ 5.5.3. Consider the Dirichlet boundary value problem $\Delta u = 0 \quad u(x,0) = \sin x$, $u(x,\pi) = 0$, $u(0,y) = 0$, $u(\pi,y) = 0$, on the square $\{\, 0 < x, y < \pi \,\}$. (*a*) Find the exact solution. (*b*) Set up and solve the finite difference equations based on a square mesh with $m = n = 2$ squares on each side of the full square. How close is this value to the exact solution at the center of the square: $u\left(\frac{1}{2}\pi, \frac{1}{2}\pi\right)$? (*c*) Repeat part (*b*) for $m = n = 4$ squares per side. Is the value of your approximation at the center of the unit square closer to the true solution? (*d*) Use a computer to find a finite difference approximation to $u\left(\frac{1}{2}\pi, \frac{1}{2}\pi\right)$ using $m = n = 8$ and 16 squares per side. Is your approximation converging to the exact solution as the mesh becomes finer and finer? Is the convergence rate consistent with the order of the finite difference approximation?

♠ 5.5.4. (*a*) Use finite differences to approximate a solution to the Helmholtz boundary value problem $\Delta u = u$, $u(x,0) = u(x,1) = u(0,y) = 0$, $u(1,y) = 1$, on the unit square $0 < x, y < 1$. (*b*) Use separation of variables to construct a series solution. Do your analytic and numerical solutions match? Explain any discrepancies.

♠ 5.5.5. A drum is in the shape of an L, as in the accompanying figure, whose short sides all have length 1. (*a*) Use a finite difference scheme with mesh spacing $\Delta x = \Delta y = .1$ to find and graph the equilibrium configuration when the drum is subject to a unit upwards force while all its sides are fixed to the (x,y)–plane. What is the maximal deflection, and at which point(s) does it occur? (*b*) Check the accuracy of your answer in part (*a*) by reducing the step size by half: $\Delta x = \Delta y = .05$.



♣ 5.5.6. A metal plate has the shape of a 3 cm square with a 1 cm square hole cut out of the middle. The plate is heated by making the inner edge have temperature 100° while keeping the outer edge at 0°. (*a*) Find the (approximate) equilibrium temperature using finite differences with a mesh width of $\Delta x = \Delta y = .5$ cm. Plot your approximate solution using a three-dimensional graphics program. (*b*) Let C denote the square contour lying midway between the inner and outer square boundaries of the plate. Using your finite difference approximation, determine at what point(s) on C the temperature is (*i*) minimized; (*ii*) maximimized; (*iii*) equal to the average of the two boundary temperatures.
(*c*) Repeat part (*a*) using a smaller mesh width of $\Delta x = \Delta y = .2$. How much does this affect your answers in part (*b*)?

♣ 5.5.7. Answer Exercise 5.5.6 when the plate is additionally subjected to a constant heat source

$$f(x,y) = 600\,x + 800\,y - 2400.$$

♠ 5.5.8. (*a*) Explain how to adapt the finite difference method to a mixed boundary value problem on a rectangle with inhomogeneous Neumann conditions. *Hint*: Use a one-sided difference formula of the appropriate order to approximate the normal derivative at the boundary. (*b*) Apply your method to the problem

$$\Delta u = 0, \qquad u(x,0) = 0, \qquad u(x,1) = 0, \qquad \frac{\partial u}{\partial x}(0,y) = y(1-y), \qquad u(1,y) = 0,$$

using mesh sizes $\Delta x = \Delta y = .1, .01$, and $.001$. Compare your answers. (*c*) Solve the boundary value problem via separation of variables, and compare the value of the solution and the numerical approximations at the center of the square.