Chapter 10
Advection Equations and Hyperbolic Systems

Hyperbolic partial differential equations (PDEs) arise in many physical problems, typically whenever wave motion is observed. Acoustic waves, electromagnetic waves, seismic waves, shock waves, and many other types of waves can be modeled by hyperbolic equations. Often these are modeled by linear hyperbolic equations (for the propagation of sufficiently small perturbations), but modeling large motions generally requires solving nonlinear hyperbolic equations. Hyperbolic equations also arise in advective transport, when a substance is carried along with a flow, giving rise to an advection equation. This is a scalar linear first order hyperbolic PDE, the simplest possible case. See Appendix E for more discussion of hyperbolic problems and a derivation of the advection equation in particular.

In this chapter we will primarily consider the advection equation. This is sufficient to illustrate many (although certainly not all) of the issues that arise in the numerical solution of hyperbolic equations. Section 10.10 contains a very brief introduction to hyperbolic systems, still in the linear case. A much more extensive discussion of hyperbolic problems and numerical methods, including nonlinear problems and multidimensional methods, can be found in [66]. Those interested in solving more challenging hyperbolic problems may also look at the CLAWPACK software [64], which was designed primarily for hyperbolic problems. There are also a number of other books devoted to nonlinear hyperbolic equations and their solution, e.g., [58], [88].

10.1 Advection

In this section we consider numerical methods for the scalar advection equation

$$u_t + au_x = 0, \tag{10.1}$$

where a is a constant. See Section E.2.1 for a discussion of this equation. For the Cauchy problem we also need initial data

$$u(x, 0) = \eta(x).$$

This is the simplest example of a *hyperbolic* equation, and it is so simple that we can write down the exact solution,

$$u(x,t) = \eta(x - at). \tag{10.2}$$

One can verify directly that this is the solution (see also Appendix E). However, many of the issues that arise more generally in discretizing hyperbolic equations can be most easily seen with this equation.

The first approach we might consider is the analogue of the method (9.4) for the heat equation. Using the centered difference in space,

$$u_x(x,t) = \frac{u(x+h,t) - u(x-h,t)}{2h} + O(h^2) \tag{10.3}$$

and the forward difference in time results in the numerical method

$$\frac{U_j^{n+1} - U_j^n}{k} = -\frac{a}{2h}(U_{j+1}^n - U_{j-1}^n), \tag{10.4}$$

which can be rewritten as

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n). \tag{10.5}$$

This again has the stencil shown in Figure 9.1(a). In practice this method is not useful because of stability considerations, as we will see in the next section.

A minor modification gives a more useful method. If we replace U_j^n on the right-hand side of (10.5) by the average $\frac{1}{2}(U_{j-1}^n + U_{j+1}^n)$, then we obtain the *Lax–Friedrichs method*,

$$U_j^{n+1} = \frac{1}{2}(U_{j-1}^n + U_{j+1}^n) - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n). \tag{10.6}$$

Because of the low accuracy, this method is not commonly used in practice, but it serves to illustrate some stability issues and so we will study this method along with (10.5) before describing higher order methods, such as the well-known Lax–Wendroff method.

We will see in the next section that Lax–Friedrichs is Lax–Richtmyer stable (see Section 9.5) and convergent provided

$$\left|\frac{ak}{h}\right| \le 1. \tag{10.7}$$

Note that this stability restriction allows us to use a time step $k = O(h)$ although the method is explicit, unlike the case of the heat equation. The basic reason is that the advection equation involves only the first order derivative u_x rather than u_{xx} and so the difference equation involves $1/h$ rather than $1/h^2$.

The time step restriction (10.7) is consistent with what we would choose anyway based on accuracy considerations, and in this sense the advection equation is *not stiff*, unlike the heat equation. This is a fundamental difference between hyperbolic equations and parabolic equations more generally and accounts for the fact that hyperbolic equations are typically solved with explicit methods, while the efficient solution of parabolic equations generally requires implicit methods.

To see that (10.7) gives a reasonable time step, note that

$$u_x(x,t) = \eta'(x - at),$$

while

$$u_t(x,t) = -au_x(x,t) = -a\eta'(x - at).$$

The time derivative u_t is larger in magnitude than u_x by a factor of a, and so we would expect the time step required to achieve temporal resolution consistent with the spatial resolution h to be smaller by a factor of a. This suggests that the relation $k \approx h/a$ would be reasonable in practice. This is completely consistent with (10.7).

10.2 Method of lines discretization

To investigate stability further we will again introduce the method of lines (MOL) discretization as we did in Section 9.2 for the heat equation. To obtain a system of equations with finite dimension we must solve the equation on some bounded domain rather than solving the Cauchy problem. However, in a bounded domain, say, $0 \le x \le 1$, the advection equation can have a boundary condition specified on only one of the two boundaries. If $a > 0$, then we need a boundary condition at $x = 0$, say,

$$u(0,t) = g_0(t), \tag{10.8}$$

which is the *inflow* boundary in this case. The boundary at $x = 1$ is the *outflow* boundary and the solution there is completely determined by what is advecting to the right from the interior. If $a < 0$, we instead need a boundary condition at $x = 1$, which is the inflow boundary in this case.

The symmetric 3-point methods defined above can still be used near the inflow boundary but not at the outflow boundary. Instead the discretization will have to be coupled with some "numerical boundary condition" at the outflow boundary, say, a one-sided discretization of the equation. This issue complicates the stability analysis and will be discussed in Section 10.12.

For analysis purposes we can obtain a nice MOL discretization if we consider the special case of *periodic boundary conditions*,

$$u(0,t) = u(1,t) \quad \text{for } t \ge 0.$$

Physically, whatever flows out at the outflow boundary flows back in at the inflow boundary. This also models the Cauchy problem in the case where the initial data is periodic with period 1, in which case the solution remains periodic and we need to model only a single period $0 \le x \le 1$.

In this case the value $U_0(t) = U_{m+1}(t)$ along the boundaries is another unknown, and we must introduce one of these into the vector $U(t)$. If we introduce $U_{m+1}(t)$, then we have the vector of grid values

$$U(t) = \begin{bmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_{m+1}(t) \end{bmatrix}.$$

For $2 \le j \le m$ we have the ordinary differential equation (ODE)

$$U_j'(t) = -\frac{a}{2h}(U_{j+1}(t) - U_{j-1}(t)),$$

while the first and last equations are modified using the periodicity:

$$U_1'(t) = -\frac{a}{2h}(U_2(t) - U_{m+1}(t)),$$
$$U_{m+1}'(t) = -\frac{a}{2h}(U_1(t) - U_m(t)).$$

This system can be written as

$$U'(t) = AU(t) \tag{10.9}$$

with

$$A = -\frac{a}{2h}\begin{bmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 1 & & & & -1 & 0 \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)}. \tag{10.10}$$

Note that this matrix is skew-symmetric ($A^T = -A$) and so its eigenvalues must be pure imaginary. In fact, the eigenvalues are

$$\lambda_p = -\frac{ia}{h}\sin(2\pi ph) \quad \text{for } p = 1,\ 2,\ \ldots,\ m+1. \tag{10.11}$$

The corresponding eigenvector u^p has components

$$u_j^p = e^{2\pi ipjh} \quad \text{for } j = 1,\ 2,\ \ldots,\ m+1. \tag{10.12}$$

The eigenvalues lie on the imaginary axis between $-ia/h$ and ia/h.

For absolute stability of a time discretization we need the stability region $\mathcal{S}$ to include this interval. Any method that includes some interval iy, $|y| < b$ of the imaginary axis will lead to a stable method for the advection equation provided $|ak/h| \le b$. For example, looking again at the stability regions plotted in Figures 7.1 through 7.3 and Figure 8.5 shows that the midpoint method or certain Adams methods may be suitable for this problem, whereas the backward differentiation formula (BDF) methods are not.

10.2.1 Forward Euler time discretization

The method (10.5) can be viewed as the forward Euler time discretization of the MOL system of ODEs (10.9). We found in Section 7.3 that this method is stable only if $|1+k\lambda| \le 1$ and the stability region $\mathcal{S}$ is the unit circle centered at -1. No matter how small the ratio k/h is, since the eigenvalues λ_p from (10.11) are imaginary, the values $k\lambda_p$ will not lie in $\mathcal{S}$. Hence the method (10.5) is *unstable* for any fixed mesh ratio k/h; see Figure 10.1(a).

The method (10.5) will be convergent if we let $k \to 0$ faster than h, since then $k\lambda_p \to 0$ for all p and the zero-stability of Euler's method is enough to guarantee convergence. Taking k much smaller than h is generally not desirable and the method is not used in practice. However, it is interesting to analyze this situation also in terms of Lax–Richtmyer stability, since it shows an example where the Lax–Richtmyer stability uses a weaker bound of the form (9.22), $\|B\| \le 1 + \alpha k$, rather than $\|B\| \le 1$. Here $B = I + kA$. Suppose we take $k = h^2$, for example. Then we have

$$|1 + k\lambda_p|^2 \le 1 + (ka/h)^2$$

for each p (using the fact that λ_p is pure imaginary) and so

$$|1 + k\lambda_p|^2 \le 1 + a^2h^2 = 1 + a^2k.$$

Hence $\|I + kA\|_2^2 \le 1 + a^2k$ and if $nk \le T$, we have

$$\|(I + kA)^n\|_2 \le (1 + a^2k)^{n/2} \le e^{a^2T/2},$$

showing the uniform boundedness of $\|B^n\|$ (in the 2-norm) needed for Lax–Richtmyer stability.

10.2.2 Leapfrog

A better time discretization is to use the midpoint method (5.23),

$$U^{n+1} = U^{n-1} + 2kAU^n,$$

which gives the *leapfrog method* for the advection equation,

$$U_j^{n+1} = U_j^{n-1} - \frac{ak}{h}(U_{j+1}^n - U_{j-1}^n). \tag{10.13}$$

This is a 3-level explicit method and is second order accurate in both space and time.

Recall from Section 7.3 that the stability region of the midpoint method is the interval $i\alpha$ for $-1 < \alpha < 1$ of the imaginary axis. This method is hence stable on the advection equation provided $|ak/h| < 1$ is satisfied.

On the other hand, note that the $k\lambda_p$ will always be on the *boundary* of the stability region (the stability region for midpoint has no interior). This means the method is only marginally stable—there is no growth but also no decay of any eigenmode. The difference equation is said to be *nondissipative*. In some ways this is good—the true advection equation is also nondissipative, and any initial condition simply translates unchanged, no matter how oscillatory. Leapfrog captures this qualitative behavior well.

However, there are problems with this. All modes translate without decay, but they do not all propagate at the correct velocity, as will be explained in Example 10.12. As a result initial data that contains high wave number components (e.g., if the data contains steep gradients) will disperse and can result in highly oscillatory numerical approximations.

The marginal stability of leapfrog can also turn into instability if a method of this sort is applied to a more complicated problem with variable coefficients or nonlinearities.

10.2.3 Lax–Friedrichs

Again consider the Lax–Friedrichs method (10.6). Note that we can rewrite (10.6) using the fact that

$$\frac{1}{2}(U_{j-1}^n + U_{j+1}^n) = U_j^n + \frac{1}{2}(U_{j-1}^n - 2U_j^n + U_{j+1}^n)$$

to obtain

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{1}{2}(U_{j-1}^n - 2U_j^n + U_{j+1}^n). \tag{10.14}$$

This can be rearranged to give

$$\frac{U_j^{n+1} - U_j^n}{k} + a\left(\frac{U_{j+1}^n - U_{j-1}^n}{2h}\right) = \frac{h^2}{2k}\left(\frac{U_{j-1}^n - 2U_j^n + U_{j+1}^n}{h^2}\right).$$

If we compute the local truncation error from this form we see, as expected, that it is consistent with the advection equation $u_t + au_x = 0$, since the term on the right-hand side vanishes as $k,\ h \to 0$ (assuming k/h is fixed). However, it looks more like a discretization of the advection-diffusion equation

$$u_t + au_x = \epsilon u_{xx},$$

where $\epsilon = h^2/2k$.

Later in this chapter we will study the diffusive nature of many methods for the advection equation. For our present purposes, however, the crucial part is that we can now view (10.14) as resulting from a forward Euler discretization of the system of ODEs

$$U'(t) = A_\epsilon U(t)$$

with

$$\begin{aligned} A_\epsilon = &-\frac{a}{2h}\begin{bmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 1 & & & & -1 & 0 \end{bmatrix} \\ &+ \frac{\epsilon}{h^2}\begin{bmatrix} -2 & 1 & & & & 1 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ 1 & & & & 1 & -2 \end{bmatrix}, \end{aligned} \tag{10.15}$$

where $\epsilon = h^2/2k$. The matrix A_ϵ differs from the matrix A of (10.10) by the addition of a small multiple of the second difference operator, which is symmetric rather than skew-symmetric. As a result the eigenvalues of A_ϵ are shifted off the imaginary axis and now lie

in the left half-plane. There is now some hope that each $k\lambda$ will lie in the stability region of Euler's method if k is small enough relative to h.

It can be verified that the eigenvectors (10.12) of the matrix A are also eigenvectors of the second difference operator (with periodic boundary conditions) that appears in (10.15), and hence these are also the eigenvectors of the full matrix A_ϵ. We can easily compute that the eigenvalues of A_ϵ are

$$\mu_p = -\frac{ia}{h}\sin(2\pi ph) - \frac{2\epsilon}{h^2}(1 - \cos(2\pi ph)). \tag{10.16}$$

The values $k\mu_p$ are plotted in the complex plane for various different values of ϵ in Figure 10.1. They lie on an ellipse centered at $-2k\epsilon/h^2$ with semi-axes of length $2k\epsilon/h^2$ in the x-direction and ak/h in the y-direction. For the special case $\epsilon = h^2/2k$ used in Lax–Friedrichs, we have $-2k\epsilon/h^2 = -1$ and this ellipse lies entirely inside the unit circle centered at -1, provided that $|ak/h| \le 1$. (If $|ak/h| > 1$, then the top and bottom of the ellipse would extend outside the circle.) The forward Euler method is stable as a time-discretization, and hence the Lax–Friedrichs method is Lax–Richtmyer stable, provided $|ak/h| \le 1$.

10.3 The Lax–Wendroff method

One way to achieve second order accuracy on the advection equation is to use a second order temporal discretization of the system of ODEs (10.9), since this system is based on a second order spatial discretization. This can be done with the midpoint method, for example, which gives rise to the leapfrog scheme (10.13) already discussed. However, this is a three-level method and for various reasons it is often much more convenient to use two-level methods for PDEs whenever possible—in more than one dimension the need to store several levels of data may be restrictive, boundary conditions can be harder to impose, and combining methods using fractional step procedures (as discussed in Chapter 11) may require two-level methods for each step, to name a few reasons. Moreover, the leapfrog method is nondissipative, leading to potential stability problems if the method is extended to variable coefficient or nonlinear problems.

Another way to achieve second order accuracy in time would be to use the trapezoidal method to discretize the system (10.9), as was done to derive the Crank–Nicolson method for the heat equation. But this is an implicit method and for hyperbolic equations there is generally no need to introduce this complication and expense.

Another possibility is to use a two-stage Runge–Kutta method such as the one in Example 5.11 for the time discretization. This can be done, although some care must be exercised near boundaries, and the use of a multistage method again typically requires additional storage.

One simple way to achieve a two-level explicit method with higher accuracy is to use the idea of Taylor series methods, as described in Section 5.6. Applying this directly to the linear system of ODEs $U'(t) = AU(t)$ (and using $U'' = AU' = A^2U$) gives the second order method

$$U^{n+1} = U^n + kAU^n + \frac{1}{2}k^2A^2U^n.$$

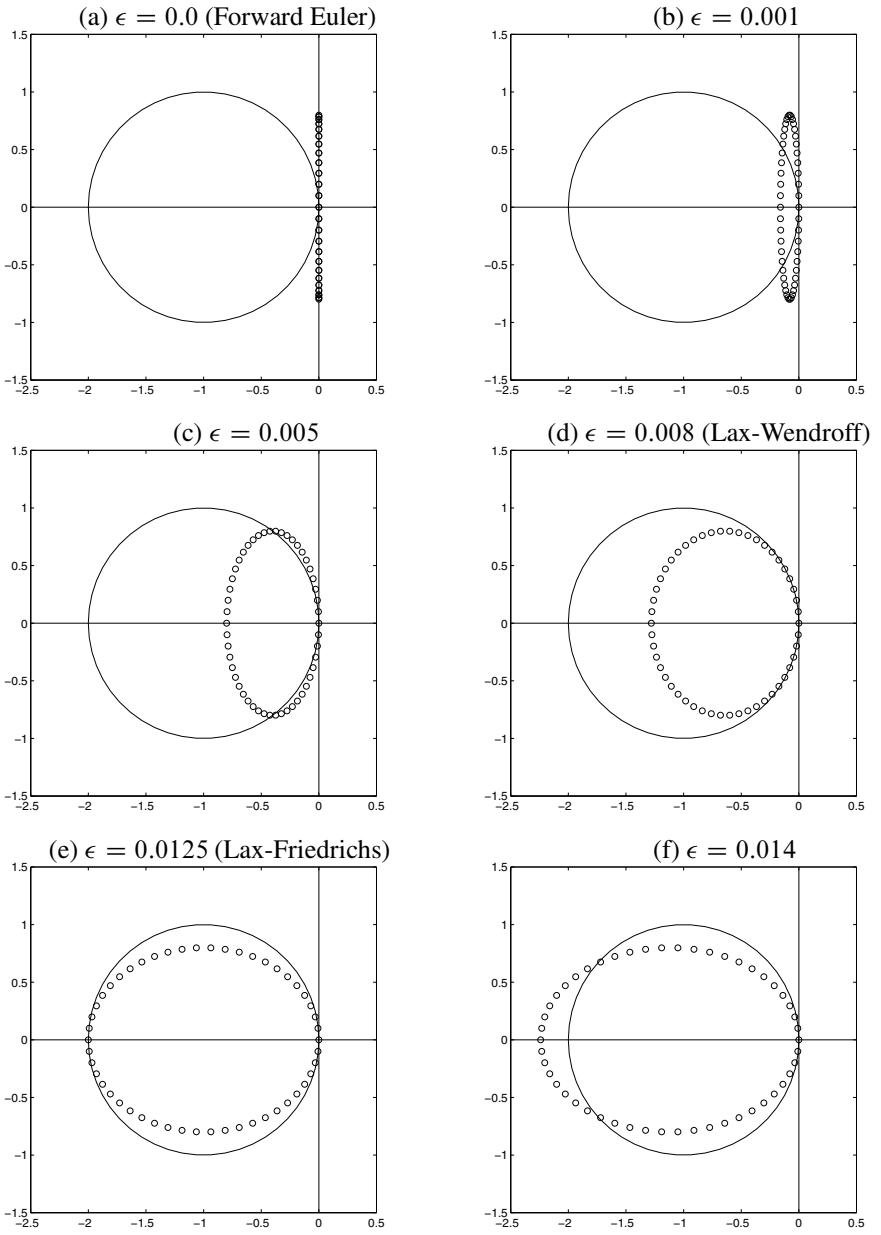

Figure 10.1. *Eigenvalues of the matrix A_ϵ in* (10.15)*, for various values of ϵ, in the case $h = 1/50$ and $k = 0.8h$, $a = 1$, so $ak/h = 0.8$.* (a) *shows the case $\epsilon = 0$ which corresponds to the forward Euler method* (10.5). (d) *shows the case $\epsilon = a^2k/2$, the Lax–Wendroff method* (10.18). (e) *shows the case $\epsilon = h^2/2k$, the Lax–Friedrichs method* (10.6). *The method is stable for ϵ between $a^2k/2$ and $h^2/2k$, as in* (d) *through* (e).

Here A is the matrix (10.10), and computing A^2 and writing the method at the typical grid point then gives

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{a^2k^2}{8h^2}(U_{j-2}^n - 2U_j^n + U_{j+2}^n). \tag{10.17}$$

This method is second order accurate and explicit but has a 5-point stencil involving the points U_{j-2}^n and U_{j+2}^n. With periodic boundary conditions this is not a problem, but with other boundary conditions this method needs more numerical boundary conditions than a 3-point method. This makes it less convenient to use and potentially more prone to numerical instability.

Note that the last term in (10.17) is an approximation to $\frac{1}{2}a^2k^2u_{xx}$ using a centered difference based on step size $2h$. A simple way to achieve a second order accurate 3-point method is to replace this term by the more standard 3-point formula. We then obtain the standard *Lax–Wendroff method*:

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{a^2k^2}{2h^2}(U_{j-1}^n - 2U_j^n + U_{j+1}^n). \tag{10.18}$$

A cleaner way to derive this method is to use Taylor series expansions directly on the PDE $u_t + au_x = 0$, to obtain

$$u(x, t+k) = u(x,t) + ku_t(x,t) + \frac{1}{2}k^2u_{tt}(x,t) + \cdots.$$

Replacing u_t by $-au_x$ and u_{tt} by a^2u_{xx} gives

$$u(x, t+k) = u(x,t) - kau_x(x,t) + \frac{1}{2}k^2a^2u_{xx}(x,t) + \cdots.$$

If we now use the standard centered approximations to u_x and u_{xx} and drop the higher order terms, we obtain the Lax–Wendroff method (10.18). It is also clear how we could obtain higher order accurate explicit two-level methods by this same approach, by retaining more terms in the series and approximating the spatial derivatives (including the higher order spatial derivatives that will then arise) by suitably high order accurate finite difference approximations. The same approach can also be used with other PDEs. The key is to replace the time derivatives arising in the Taylor series expansion with spatial derivatives, using expressions obtained by differentiating the original PDE.

10.3.1 Stability analysis

We can analyze the stability of Lax–Wendroff following the same approach used for Lax–Friedrichs in Section 10.2. Note that with periodic boundary conditions, the Lax–Wendroff method (10.18) can be viewed as Euler's method applied to the linear system of ODEs $U'(t) = A_\epsilon U(t)$, where A_ϵ is given by (10.15) with $\epsilon = a^2k/2$ (instead of the value $\epsilon = h^2/2k$ used in Lax–Friedrichs). The eigenvalues of A_ϵ are given by (10.16) with the appropriate value of ϵ, and multiplying by the time step k gives

$$k\mu_p = -i\left(\frac{ak}{h}\right)\sin(p\pi h) + \left(\frac{ak}{h}\right)^2(\cos(p\pi h) - 1).$$

These values all lie on an ellipse centered at $-(ak/h)^2$ with semi-axes of length $(ak/h)^2$ and $|ak/h|$. If $|ak/h| \leq 1$, then all of these values lie inside the stability region of Euler's method. Figure 10.1(d) shows an example in the case $ak/h = 0.8$. The Lax–Wendroff method is stable with exactly the same time step restriction (10.7) as required for Lax–Friedrichs. In Section 10.7 we will see that this is a very natural stability condition to expect for the advection equation and is the best we could hope for when a 3-point method is used.

A close look at Figure 10.1 shows that the values $k\mu_p$ near the origin lie much closer to the boundary of the stability region for the Lax–Wendroff method (Figure 10.1(d)) than for the other methods illustrated in this figure. This is a reflection of the fact that Lax–Wendroff is second order accurate, while the others are only first order accurate. Note that a value $k\mu_p$ lying inside the stability region indicates that this eigenmode will be damped as the wave propagates, which is unphysical behavior since the true solution advects with no dissipation. For small values of μ_p (low wave numbers, smooth components) the Lax–Wendroff method has relatively little damping and the method is more accurate. Higher wave numbers are still damped with Lax–Wendroff (unless $|ak/h| = 1$, in which case all the $k\mu_p$ lie on the boundary of $\mathcal{S}$) and resolving the behavior of these modes properly would require a finer grid.

Comparing Figures 10.1(c), (d), and (e) shows that Lax–Wendroff has the minimal amount of numerical damping needed to bring the values $k\mu_p$ within the stability region. Any less damping, as in Figure 10.1(c) would lead to instability, while more damping as in Figure 10.1(e) gives excessive smearing of low wave numbers. Recall that the value of ϵ used in Lax–Wendroff was determined by doing a Taylor series expansion and requiring second order accuracy, so this makes sense.

10.4 Upwind methods

So far we have considered methods based on symmetric approximations to derivatives. Alternatively, one might use a nonsymmetric approximation to u_x in the advection equation, e.g.,

$$u_x(x_j, t) \approx \frac{1}{h}(U_j - U_{j-1}) \tag{10.19}$$

or

$$u_x(x_j, t) \approx \frac{1}{h}(U_{j+1} - U_j). \tag{10.20}$$

These are both *one-sided approximations*, since they use data only to one side or the other of the point x_j. Coupling one of these approximations with forward differencing in time gives the following methods for the advection equation:

$$U_j^{n+1} = U_j^n - \frac{ak}{h}(U_j^n - U_{j-1}^n) \tag{10.21}$$

or

$$U_j^{n+1} = U_j^n - \frac{ak}{h}(U_{j+1}^n - U_j^n). \tag{10.22}$$

These methods are first order accurate in both space and time. One might wonder why we would want to use such approximations, since centered approximations are more accurate.

For the advection equation, however, there is an asymmetry in the equations because the equation models translation at speed a. If $a > 0$, then the solution moves to the right, while if $a < 0$ it moves to the left. There are situations where it is best to acknowledge this asymmetry and use one-sided differences in the appropriate direction.

The choice between the two methods (10.21) and (10.22) should be dictated by the sign of a. Note that the true solution over one time step can be written as

$$u(x_j, t + k) = u(x_j - ak, t)$$

so that the solution at the point x_j at the next time level is given by data to the *left* of x_j if $a > 0$, whereas it is determined by data to the *right* of x_j if $a < 0$. This suggests that (10.21) might be a better choice for $a > 0$ and (10.22) for $a < 0$.

In fact the stability analysis below shows that (10.21) is stable only if

$$0 \le \frac{ak}{h} \le 1. \tag{10.23}$$

Since k and h are positive, we see that this method can be used only if $a > 0$. This method is called the *upwind method* when used on the advection equation with $a > 0$. If we view the equation as modeling the concentration of some tracer in air blowing past us at speed a, then we are looking in the correct upwind direction to judge how the concentration will change with time. (This is also referred to as an *upstream differencing* method in some literature.)

Conversely, (10.22) is stable only if

$$-1 \le \frac{ak}{h} \le 0 \tag{10.24}$$

and can be used only if $a < 0$. In this case (10.22) is the proper upwind method to use.

10.4.1 Stability analysis

The method (10.21) can be written as

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{ak}{2h}(U_{j+1}^n - 2U_j^n + U_{j-1}^n), \tag{10.25}$$

which puts it in the form (10.15) with $\epsilon = ah/2$. We have seen previously that methods of this form are stable provided $|ak/h| \le 1$ and also $-2 < -2\epsilon k/h^2 < 0$. Since $k,\ h > 0$, this requires in particular that $\epsilon > 0$. For Lax–Friedrichs and Lax–Wendroff, this condition was always satisfied, but for upwind the value of ϵ depends on a and we see that $\epsilon > 0$ only if $a > 0$. If $a < 0$, then the eigenvalues of the MOL matrix lie on a circle that lies entirely in the right half-plane, and the method will certainly be unstable. If $a > 0$, then the above requirements lead to the stability restriction (10.23).

If we think of (10.25) as modeling an advection-diffusion equation, then we see that $a < 0$ corresponds to a negative diffusion coefficient. This leads to an ill-posed equation, as in the "backward heat equation" (see Section E.3.4).

The method (10.22) can also be written in a form similar to (10.25), but the last term will have a minus sign in front of it. In this case we need $a < 0$ for any hope of stability and then easily derive the stability restriction (10.24).

The three methods, Lax–Wendroff, upwind, and Lax–Friedrichs, can all be written in the same form (10.15) with different values of ϵ. If we call these values ϵ_{LW}, ϵ_{up}, and ϵ_{LF}, respectively, then we have

$$\epsilon_{LW} = \frac{a^2k}{2} = \frac{ahv}{2}, \qquad \epsilon_{up} = \frac{ah}{2}, \qquad \epsilon_{LF} = \frac{h^2}{2k} = \frac{ah}{2v},$$

where $v = ak/h$. Note that

$$\epsilon_{LW} = v\epsilon_{up} \qquad \text{and} \qquad \epsilon_{up} = v\epsilon_{LF}.$$

If $0 < v < 1$, then $\epsilon_{LW} < \epsilon_{up} < \epsilon_{LF}$ and the method is stable for any value of ϵ between ϵ_{LW} and ϵ_{LF}, as suggested by Figure 10.1.

10.4.2 The Beam–Warming method

The upwind method is only first order accurate. A second order accurate method with the same one-sided character can be derived by following the derivation of the Lax–Wendroff method, but using one-sided approximations to the spatial derivatives. This results in the *Beam–Warming* method, which for $a > 0$ takes the form

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(3U_j^n - 4U_{j-1}^n + U_{j-2}^n) + \frac{a^2k^2}{2h^2}(U_j^n - 2U_{j-1}^n + U_{j-2}^n). \quad (10.26)$$

For $a < 0$ the Beam–Warming method is one-sided in the other direction:

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(-3U_j^n + 4U_{j+1}^n - U_{j+2}^n) + \frac{a^2k^2}{2h^2}(U_j^n - 2U_{j+1}^n + U_{j+2}^n). \quad (10.27)$$

These methods are stable for $0 \le v \le 2$ and $-2 \le v \le 0$, respectively.

10.5 Von Neumann analysis

We have analyzed the stability of various algorithms for the advection equation by viewing them as ODE methods applied to the MOL system (10.9). The same stability criteria can be obtained by using von Neumann analysis as described in Section 9.6. Recall that this is done by replacing U_j^n by $g(\xi)^n e^{i\xi jh}$ (where $i = \sqrt{-1}$ in this section). Canceling out common factors results in an expression for the amplification factor $g(\xi)$, and requiring that this be bounded by 1 in magnitude gives the stability bounds for the method.

Also recall from Section 9.6 that this can be expected to give the same result as our MOL analysis because of the close relation between the $e^{i\xi jh}$ factor and the eigenvectors of the matrix A. In a sense von Neumann analysis simply combines the computation of the eigenvalues of A together with the absolute stability analysis of the time-stepping method being used. Nonetheless we will go through this analysis explicitly for several of the methods already considered to show how it works, since for other methods it may be more convenient to work with this approach than to interpret the method as an MOL method.

For the von Neumann analysis in this section we will simplify notation slightly by setting $v = ak/h$, the Courant number.

Example 10.1. Following the procedure of Example 9.6 for the upwind method (10.21) gives

$$g(\xi) = 1 - \nu\left(1 - e^{-i\xi h}\right) = (1-\nu) + \nu e^{-i\xi h}. \tag{10.28}$$

As the wave number ξ varies, $g(\xi)$ moves around a circle of radius ν centered at $1-\nu$. These values stay within the unit circle if and only if $0 \le \nu \le 1$, the stability limit that was also found in Section 10.4.1.

Example 10.2. Going through the same procedure for Lax–Friedrichs (10.6) gives

$$\begin{aligned} g(\xi) &= \frac{1}{2}\left(e^{-i\xi h} + e^{i\xi h}\right) - \frac{1}{2}\nu\left(e^{i\xi h} - e^{-i\xi h}\right) \\ &= \cos(\xi h) - \nu i \sin(\xi h) \end{aligned} \tag{10.29}$$

and so

$$|g(\xi)|^2 = \cos^2(\xi h) + \nu^2 \sin^2(\xi h), \tag{10.30}$$

which is bounded by 1 for all ξ only if $|\nu| \le 1$.

Example 10.3. For the Lax–Wendroff method (10.18) we obtain

$$\begin{aligned} g(\xi) &= 1 - \frac{1}{2}\nu\left(e^{i\xi h} - e^{-i\xi h}\right) + \frac{1}{2}\nu^2\left(e^{i\xi h} - 2 + e^{-i\xi h}\right) \\ &= 1 - i\nu\sin(\xi h) + \nu^2(\cos(\xi h) - 1) \\ &= 1 - i\nu[2\sin(\xi h/2)\cos(\xi h/2)] - \nu^2[2\sin^2(\xi h/2)], \end{aligned} \tag{10.31}$$

where we have used two trigonometric identities to obtain the last line. This complex number has modulus

$$\begin{aligned} |g(\xi)|^2 &= [1 - 2\nu^2\sin^2(\xi h/2)]^2 + 4\nu^2\sin^2(\xi h/2)\cos^2(\xi h/2) \\ &= 1 - 4\nu^2(1-\nu^2)\sin^4(\xi h/2). \end{aligned} \tag{10.32}$$

Since $0 \le \sin^4(\xi h/2) \le 1$ for all values of ξ, we see that $|g(\xi)|^2 \le 1$ for all ξ, and hence the method is stable provided that $|\nu| \le 1$, which again gives the expected stability bound (10.7).

Example 10.4. The leapfrog method (10.13) involves three time levels but can still be handled by the same basic approach. If we set $U_j^n = g(\xi)^n e^{i\xi jh}$ in the leapfrog method we obtain

$$g(\xi)^{n+1} e^{i\xi jh} = g(\xi)^{n-1} e^{i\xi jh} - \nu g(\xi)^n \left(e^{i\xi(j+1)h} - e^{i\xi(j-1)h}\right). \tag{10.33}$$

If we now divide by $g(\xi)^{n-1} e^{i\xi jh}$ we obtain a quadratic equation for $g(\xi)$,

$$g(\xi)^2 = 1 - 2\nu i \sin(\xi h) g(\xi). \tag{10.34}$$

Examining this in the same manner as the analysis of the stability region for the midpoint method in Example 7.7 yields the stability limit $|\nu| < 1$.

It is important to note the severe limitations of the von Neumann approach just presented. It is strictly applicable only in the constant coefficient linear case (with periodic boundary conditions or on the Cauchy problem). Applying von Neumann analysis to the "frozen coefficient" problem locally often gives good guidance to the stability properties of a method more generally, but it cannot always be relied on. A great deal of work has been done on proving that methods stable for frozen coefficient problems remain stable for variable coefficient or nonlinear problems when everything is sufficiently smooth; see, for example, [40], [75]. In the nonlinear case, where the solution can contain shocks, a nonlinear stability theory is needed that employs techniques very different from von Neumann analysis; see, e.g., [66].

10.6 Characteristic tracing and interpolation

The solution to the advection equation is given by (10.2). The value of u is constant along each characteristic, which for this example is a straight line with constant slope. Over a single time step we have

$$u(x_j, t_{n+1}) = u(x_j - ak, t_n). \tag{10.35}$$

Tracing this characteristic back over time step k from the grid point x_j results in the picture shown in Figure 10.2(a). Note that if $0 < ak/h < 1$, then the point $x_j - ak$ lies between x_{j-1} and x_j. If we carefully choose k and h so that $ak/h = 1$ exactly, then $x_j - ak = x_{j-1}$ and we would find that $u(x_j, t_{n+1}) = u(x_{j-1}, t_n)$. The solution should just shift one grid cell to the right in each time step. We could compute the *exact* solution numerically with the method

$$U_j^{n+1} = U_{j-1}^n. \tag{10.36}$$

Actually, all the two-level methods that we have considered so far reduce to the formula (10.36) in this special case $ak = h$, and each of these methods happens to be exact in this case.

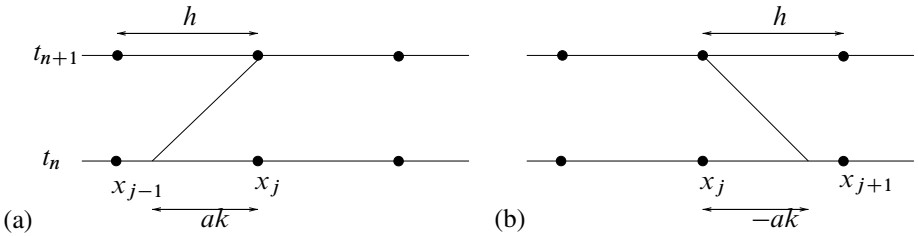

Figure 10.2. *Tracing the characteristic of the advection equation back in time from the point* (x_j, t_{n+1}) *to compute the solution according to* (10.35)*. Interpolating the value at this point from neighboring grid values gives the upwind method (for linear interpolation) or the Lax–Wendroff or Beam–Warming methods (quadratic interpolation).* (a) *shows the case* $a > 0$*,* (b) *shows the case* $a < 0$*.*

If $ak/h < 1$, then the point $x_j - ak$ is not exactly at a grid point, as illustrated in Figure 10.2. However, we might attempt to use the relation (10.35) as the basis for a numerical method by computing an approximation to $u(x_j - ak, t_n)$ based on interpolation from the grid values U_i^n at nearby grid points. For example, we might perform simple linear interpolation between U_{j-1}^n and U_j^n. Fitting a linear function to these points gives the function

$$p(x) = U_j^n + (x - x_j)\left(\frac{U_j^n - U_{j-1}^n}{h}\right). \tag{10.37}$$

Evaluating this at $x_j - ak$ and using this to define U_j^{n+1} gives

$$U_j^{n+1} = p(x_j - ak) = U_j^n - \frac{ak}{h}(U_j^n - U_{j-1}^n).$$

This is precisely the first order upwind method (10.21). Note that this also can be interpreted as a linear combination of the two values U_{j-1}^n and U_j^n:

$$U_j^{n+1} = \left(1 - \frac{ak}{h}\right)U_j^n + \frac{ak}{h}U_{j-1}^n. \tag{10.38}$$

Moreover, this is a *convex combination* (i.e., the coefficients of U_j^n and U_{j-1}^n are both nonnegative and sum to 1) provided the stability condition (10.23) is satisfied, which is also the condition required to ensure that $x_j - ak$ lies between the two points x_{j-1} and x_j. In this case we are *interpolating* between these points with the function $p(x)$. If the stability condition is violated, then we would be using $p(x)$ to *extrapolate* outside of the interval where the data lies. It is easy to see that this sort of extrapolation can lead to instability—consider what happens if the data U^n is oscillatory with $U_j^n = (-1)^j$, for example.

To obtain better accuracy, we might try using a higher order interpolating polynomial based on more data points. If we define a quadratic polynomial $p(x)$ by interpolating the values U_{j-1}^n, U_j^n, and U_{j+1}^n, and then define U_j^{n+1} by evaluating $p(x_j - ak)$, we simply obtain the Lax–Wendroff method (10.18). Note that in this case we are properly interpolating provided that the stability restriction $|ak/h| \le 1$ is satisfied. If we instead base our quadratic interpolation on the three points U_{j-2}^n, U_{j-1}^n, and U_j^n, then we obtain the Beam–Warming method (10.26), and we are properly interpolating provided $0 \le ak/h \le 2$.

10.7 The Courant–Friedrichs–Lewy condition

The discussion of Section 10.6 suggests that for the advection equation, the point $x_j - ak$ must be bracketed by points used in the stencil of the finite difference method if the method is to be stable and convergent. This turns out to be a *necessary* condition in general for any method developed for the advection equation: if U_j^{n+1} is computed based on values U_{j+p}^n, U_{j+p+1}^n, $\ldots$, U_{j+q}^n with $p \le q$ (negative values are allowed for p and q), then we must have $x_{j+p} \le x_j - ak \le x_{j+q}$ or the method cannot be convergent. Since $x_i = ih$, this requires

$$-q \le \frac{ak}{h} \le -p.$$

This result for the advection equation is one special case of a much more general principle that is called the *CFL condition*. This condition is named after Courant, Friedrichs, and Lewy, who wrote a fundamental paper in 1928 that was the first paper on the stability and convergence of finite difference methods for PDEs. (The original paper [17] is in German but an English translation is available in [18].) The value $\nu = ak/h$ is often called the *Courant number*.

To understand this general condition, we must discuss the *domain of dependence* of a time-dependent PDE. (See, e.g., [55], [66] for more details.) For the advection equation, the solution $u(X, T)$ at some fixed point (X, T) depends on the initial data η at only a single point: $u(X, T) = u(X - aT)$. We say that the domain of dependence of the point (X, T) is the point $X - aT$:

$$\mathcal{D}(X, T) = \{X - aT\}.$$

If we modify the data η at this point, then the solution $u(X, T)$ will change, while modifying the data at any other point will have no effect on the solution at this point.

This is a rather unusual situation for a PDE. More generally we might expect the solution at (X, T) to depend on the data at several points or over a whole interval. In Section 10.10 we consider hyperbolic systems of equations of the form $u_t + Au_x = 0$, where $u \in \mathbb{R}^s$ and $A \in \mathbb{R}^{s\times s}$ is a matrix with real eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_s$. If these values are distinct then we will see that the solution $u(X, T)$ depends on the data at the s distinct points $X - \lambda_1 T, \ \ldots, X - \lambda_s T$, and hence

$$\mathcal{D}(X, T) = \{X - \lambda_p T \text{ for } p = 1, 2, \ \ldots, \ s\}. \tag{10.39}$$

The heat equation $u_t = u_{xx}$ has a much larger domain of dependence. For this equation the solution at any point (X, T) depends on the data *everywhere* and the domain of dependence is the whole real line,

$$\mathcal{D}(X, T) = (-\infty, \infty).$$

This equation is said to have infinite propagation speed, since data at any point affects the solution everywhere at any small time in the future (although its effect of course decays exponentially away from this point, as seen from the Green's function (E.37)).

A finite difference method also has a domain of dependence. On a particular fixed grid we define the domain of dependence of a grid point (x_j, t_n) to be the set of grid points x_i at the initial time $t = 0$ with the property that the data U_i^0 at x_i has an effect on the solution U_j^n. For example, with the Lax–Wendroff method (10.18) or any other 3-point method, the value U_j^n depends on U_{j-1}^{n-1}, U_j^{n-1}, and U_{j+1}^{n-1}. These values depend in turn on U_{j-2}^{n-2} through U_{j+2}^{n-2}. Tracing back to the initial time we obtain a triangular array of grid points as seen in Figure 10.3(a), and we see that U_j^n depends on the initial data at the points $x_{j-n}, \ \ldots, x_{j+n}$.

Now consider what happens if we refine the grid, keeping k/h fixed. Figure 10.3(b) shows the situation when k and h are reduced by a factor of 2, focusing on the same value of (X, T) which now corresponds to U_{2j}^{2n} on the finer grid. This value depends on twice as many values of the initial data, but these values all lie within the same interval and are merely twice as dense.

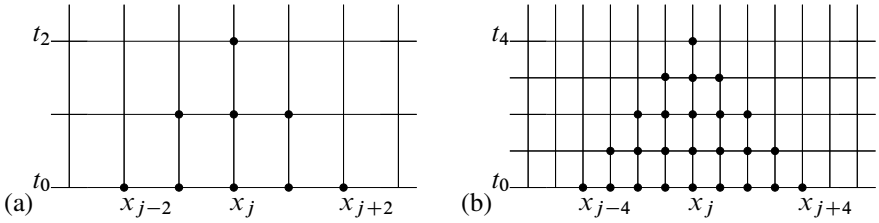

Figure 10.3. (a) *Numerical domain of dependence of a grid point when using a 3-point explicit method.* (b) *On a finer grid.*

If the grid is refined further with $k/h \equiv r$ fixed, then clearly the numerical domain of dependence of the point (X, T) will fill in the interval $[X - T/r,\ X + T/r]$. As we refine the grid, we hope that our computed solution at (X, T) will converge to the true solution $u(X, T) = \eta(X - aT)$. Clearly this can be possible only if

$$X - T/r \le X - aT \le X + T/r. \tag{10.40}$$

Otherwise, the true solution will depend only on a value $\eta(X - aT)$ that is never seen by the numerical method, no matter how fine a grid we take. We could change the data at this point and hence change the true solution without having any effect on the numerical solution, so the method cannot be convergent for general initial data.

Note that the condition (10.40) translates into $|a| \le 1/r$ and hence $|ak/h| \le 1$. This can also be written as $|ak| \le h$, which just says that over a single time step the characteristic we trace back must lie within one grid point of x_j. (Recall the discussion of interpolation versus extrapolation in Section 10.6.)

The CFL condition generalizes this idea:

The CFL condition: A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as k and h go to zero.

For the Lax–Friedrichs, leapfrog, and Lax–Wendroff methods the condition on k and h required by the CFL condition is exactly the stability restriction we derived earlier in this chapter. But it is important to note that in general the CFL condition is only a *necessary* condition. If it is violated, then the method cannot be convergent. If it is satisfied, then the method *might* be convergent, but a proper stability analysis is required to prove this or to determine the proper stability restriction on k and h. (And of course consistency is also required for convergence—stability alone is not enough.)

Example 10.5. The 3-point method (10.5) has the same stencil and numerical domain of dependence as Lax–Wendroff but is unstable for any fixed value of k/h even though the CFL condition is satisfied for $|ak/h| \le 1$.

Example 10.6. The upwind methods (10.21) and (10.22) each have a 2-point stencil and the stability restrictions of these methods, (10.23) and (10.24), respectively, agree precisely with what the CFL condition requires.

Example 10.7. The Beam–Warming method (10.26) has a 3-point one-sided stencil. The CFL condition is satisfied if $0 \le ak/h \le 2$. When $a < 0$ the method (10.27) is used and the CFL condition requires $-2 \le ak/h \le 0$. These are also the stability regions for the methods, which must be verified by appropriate stability analysis.

Example 10.8. For the heat equation the true domain of dependence is the whole real line. It appears that any 3-point explicit method violates the CFL condition, and indeed it does if we fix k/h as the grid is refined. However, recall from Section 10.2.1 that the 3-point explicit method (9.5) is convergent as we refine the grid, provided we have $k/h^2 \le 1/2$. In this case when we make the grid finer by a factor of 2 in space it will become finer by a factor of 4 in time, and hence the numerical domain of dependence will cover a wider interval at time $t = 0$. As $k \to 0$ the numerical domain of dependence will spread to cover the entire real line, and hence the CFL condition is satisfied in this case.

An implicit method such as the Crank–Nicolson method (9.7) satisfies the CFL condition for any time step k. In this case the numerical domain of dependence is the entire real line because the tridiagonal linear system couples together all points in such a manner that the solution at each point depends on the data at all points (i.e., the inverse of a tridiagonal matrix is dense).

10.8 Some numerical results

Figure 10.4 shows typical numerical results obtained with three of the methods discussed in the previous sections. The initial data at time $t = 0$, shown in Figure 10.4(a), are smooth and consist of two Gaussian peaks, one sharper than the other:

$$u(x, 0) = \eta(x) = \exp(-20(x - 2)^2) + \exp(-(x - 5)^2). \tag{10.41}$$

The remaining frames in this figure show the results obtained when solving the advection equation $u_t + u_x = 0$ up to time $t = 17$, so the exact solution is simply the initial data shifted by 17 units. Note that only part of the computational domain is shown; the computation was done on the interval $0 \le x \le 25$. The grid spacing $h = 0.05$ was used, with time step $k = 0.8h$ so the Courant number is $ak/h = 0.8$. On this grid one peak is fairly well resolved and the other is poorly resolved.

Figure 10.4(b) shows the result obtained with the upwind method (10.21) and illustrates the extreme numerical dissipation of this method. Figure 10.4(c) shows the result obtained with Lax–Wendroff. The broader peak remains well resolved, while the dispersive nature of Lax–Wendroff is apparent near the sharper peak. Dispersion is even more apparent when the leapfrog method is used, as seen in Figure 10.4(d).

The "modified equation" analysis of the next section sheds more light on these results.

10.9 Modified equations

Our standard tool for estimating the accuracy of a finite difference method has been the "local truncation error." Seeing how well the true solution of the PDE satisfies the difference equation gives an indication of the accuracy of the difference equation. Now we will study a slightly different approach that can be very illuminating since it reveals much more about the structure and behavior of the numerical solution.

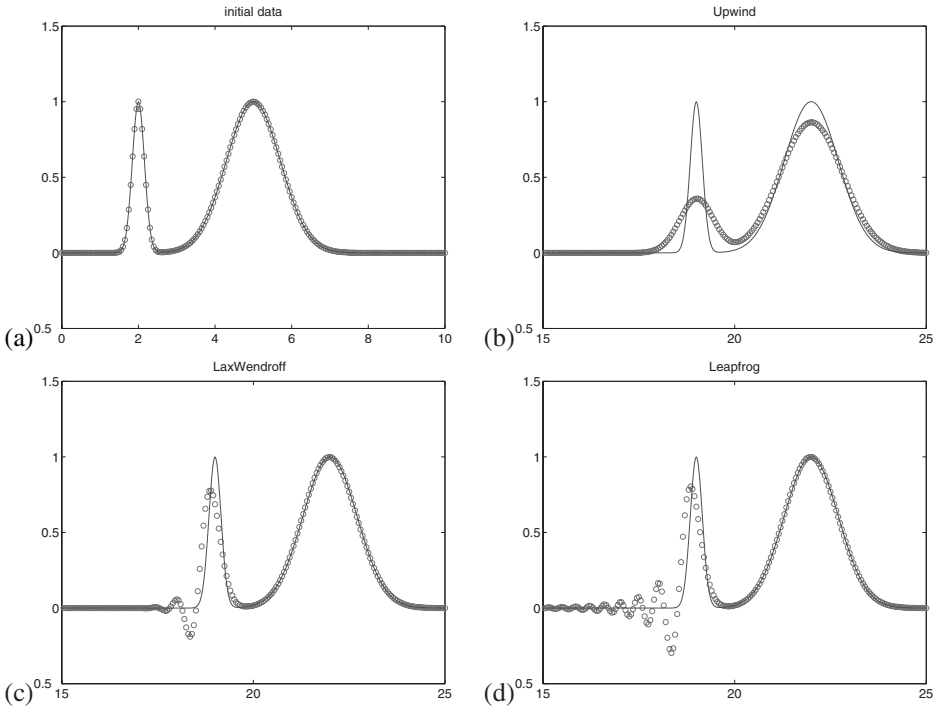

Figure 10.4. *The numerical experiments on the advection equation described in Section* 10.8.

The idea is to ask the following question: is there a PDE $v_t = \cdots$ such that our numerical approximation U_j^n is actually the *exact* solution to this PDE, $U_j^n = v(x_j, t_n)$? Or, less ambitiously, can we at least find a PDE that is better satisfied by U_j^n than the original PDE we were attempting to model? If so, then studying the behavior of solutions to this PDE should tell us much about how the numerical approximation is behaving. This can be advantageous because it is often easier to study the behavior of PDEs than of finite difference formulas.

In fact it is possible to find a PDE that is exactly satisfied by the U_j^n by doing Taylor series expansions as we do to compute the local truncation error. However, this PDE will have an infinite number of terms involving higher and higher powers of k and h. By truncating this series at some point we will obtain a PDE that is simple enough to study and yet gives a good indication of the behavior of the U_j^n.

The procedure of determining a modified equation is best illustrated with an example. See [100] for a more detailed discussion of the derivation of modified equations.

Example 10.9. Consider the upwind method (10.21) for the advection equation $u_t + au_x = 0$ in the case $a > 0$,

$$U_j^{n+1} = U_j^n - \frac{ak}{h}(U_j^n - U_{j-1}^n). \tag{10.42}$$

The process of deriving the modified equation is very similar to computing the local truncation error, only now we insert the formula $v(x,t)$ into the difference equation. This is supposed to be a function that agrees exactly with U_j^n at the grid points and so, unlike $u(x,t)$, the function $v(x,t)$ satisfies (10.42) exactly:

$$v(x,t+k) = v(x,t) - \frac{ak}{h}(v(x,t) - v(x-h,t)).$$

Expanding these terms in Taylor series about (x,t) and simplifying gives

$$\left(v_t + \frac{1}{2}kv_{tt} + \frac{1}{6}k^2 v_{ttt} + \cdots\right) + a\left(v_x - \frac{1}{2}hv_{xx} + \frac{1}{6}h^2 v_{xxx} + \cdots\right) = 0.$$

We can rewrite this as

$$v_t + av_x = \frac{1}{2}(ahv_{xx} - kv_{tt}) + \frac{1}{6}(ah^2 v_{xxx} - k^2 v_{ttt}) + \cdots.$$

This is the PDE that v satisfies. If we take k/h fixed, then the terms on the right-hand side are $O(k)$, $O(k^2)$, etc., so that for small k we can truncate this series to get a PDE that is quite well satisfied by the U_j^n.

If we drop all the terms on the right-hand side, we just recover the original advection equation. Since we have then dropped terms of $O(k)$, we expect that U_j^n satisfies this equation to $O(k)$, as we know to be true since this upwind method is first order accurate.

If we keep the $O(k)$ terms, then we get something more interesting:

$$v_t + av_x = \frac{1}{2}(ahv_{xx} - kv_{tt}). \tag{10.43}$$

This involves second derivatives in both x and t, but we can derive a slightly different modified equation with the same accuracy by differentiating (10.43) with respect to t to obtain

$$v_{tt} = -av_{xt} + \frac{1}{2}(ahv_{xxt} - kv_{ttt})$$

and with respect to x to obtain

$$v_{tx} = -av_{xx} + \frac{1}{2}(ahv_{xxx} - kv_{ttx}).$$

Combining these gives

$$v_{tt} = a^2 v_{xx} + O(k).$$

Inserting this in (10.43) gives

$$v_t + av_x = \frac{1}{2}(ahv_{xx} - a^2 kv_{xx}) + O(k^2).$$

Since we have already decided to drop terms of $O(k^2)$, we can drop these terms here also to obtain

$$v_t + av_x = \frac{1}{2}ah\left(1 - \frac{ak}{h}\right)v_{xx}. \tag{10.44}$$

This is now a familiar advection-diffusion equation. The grid values U_j^n can be viewed as giving a *second order accurate* approximation to the true solution of this equation (whereas they give only first order accurate approximations to the true solution of the advection equation).

The fact that the modified equation is an advection-diffusion equation tells us a great deal about how the numerical solution behaves. Solutions to the advection-diffusion equation translate at the proper speed a but also diffuse and are smeared out. This is clearly visible in Figure 10.4(b).

Note that the diffusion coefficient in (10.44) is $\frac{1}{2}(ah - a^2k)$, which vanishes in the special case $ak = h$. In this case we already know that the exact solution to the advection equation is recovered by the upwind method.

Also note that the diffusion coefficient is positive only if $0 < ak/h < 1$. This is precisely the stability limit of upwind. If this is violated, then the diffusion coefficient in the modified equation is negative, giving an ill-posed problem with exponentially growing solutions. Hence we see that even some information about stability can be extracted from the modified equation.

Example 10.10. If the same procedure is followed for the Lax–Wendroff method, we find that all $O(k)$ terms drop out of the modified equation, as is expected since this method is second order accurate on the advection equation. The modified equation obtained by retaining the $O(k^2)$ term and then replacing time derivatives by spatial derivatives is

$$v_t + av_x + \frac{1}{6}ah^2\left(1 - \left(\frac{ak}{h}\right)^2\right)v_{xxx} = 0. \tag{10.45}$$

The Lax–Wendroff method produces a *third order* accurate solution to this equation. This equation has a very different character from (10.43). The v_{xxx} term leads to *dispersive* behavior rather than diffusion. This is clearly seen in Figure 10.4(c), where the U_j^n computed with Lax–Wendroff are compared to the true solution of the advection equation. The magnitude of the error is smaller than with the upwind method for a given set of k and h, since it is a higher order method, but the dispersive term leads to an oscillating solution and also a shift in the location of the main peak, a *phase error*. This is similar to the dispersive behavior seen in Figure E.1 for an equation very similar to (10.45).

In Section E.3.6 the propagation properties of dispersive waves is analyzed in terms of the dispersion relation of the PDE and the phase and group velocities of different wave numbers. Following the discussion there, we find that for the modified equation (10.45), the group velocity for wave number ξ is

$$c_g = a - \frac{1}{2}ah^2\left(1 - \left(\frac{ak}{h}\right)^2\right)\xi^2,$$

which is less than a for all wave numbers. As a result the numerical result can be expected to develop a train of oscillations behind the peak, with the high wave numbers lagging farthest behind the correct location.

Some care must be used here, however, when looking at highly oscillatory waves (relative to the grid, i.e., waves for which ξh is far from 0). For ξh sufficiently small the modified equation (10.45) is a reasonable model, but for larger ξh the terms we have

neglected in this modified equation may play an equally important role. Rather than determining the dispersion relation for a method from its modified equation, it is more reliable to determine it directly from the numerical method, which is essentially what we have done in von Neumann stability analysis. This is pursued further in Example 10.13 below.

If we retain one more term in the modified equation for Lax–Wendroff, we would find that the U_j^n are fourth order accurate solutions to an equation of the form

$$v_t + av_x + \frac{1}{6}ah^2\left(1 - \left(\frac{ak}{h}\right)^2\right)v_{xxx} = -\epsilon v_{xxxx}, \tag{10.46}$$

where the ϵ in the fourth order dissipative term is $O(k^3+h^3)$ and positive when the stability bound holds. This higher order dissipation causes the highest wave numbers to be damped (see Section E.3.7), so that there is a limit to the oscillations seen in practice.

The fact that this method can produce oscillatory approximations is one of the reasons that the first order upwind method is sometimes preferable in practice. In some situations nonphysical oscillations may be disastrous, for example, if the value of u represents a concentration that cannot become negative or exceed some limit without difficulties arising elsewhere in the modeling process.

Example 10.11. The Beam–Warming method (10.26) has a similar modified equation,

$$v_t + av_x = \frac{1}{6}ah^2\left(2 - \frac{3ak}{h} + \left(\frac{ak}{h}\right)^2\right)v_{xxx}. \tag{10.47}$$

In this case the group velocity is greater than a for all wave numbers in the case $0 < ak/h < 1$, so that the oscillations move ahead of the main hump. If $1 < ak/h < 2$, then the group velocity is less than a and the oscillations fall behind. (Again the dispersion relation for (10.47) gives an accurate idea of the dispersive properties of the numerical method only for ξh sufficiently small.)

Example 10.12. The modified equation for the leapfrog method (10.13) can be derived by writing

$$\left(\frac{v(x,t+k) - v(x,t-k)}{2k}\right) = a\left(\frac{v(x+h,t) - v(x-h,t)}{2h}\right) = 0 \tag{10.48}$$

and expanding in Taylor series. As in Example 10.9 we then further differentiate the resulting equation (which has an infinite number of terms) to express higher spatial derivatives of v in terms of temporal derivatives. The dominant terms look just like Lax–Wendroff, and (10.45) is again obtained.

However, from the symmetric form of (10.48) in both x and t we see that all even-order derivatives drop out. If we derive the next term in the modified equation we will find an equation of the form

$$v_t + av_x + \frac{1}{6}ah^2\left(1 - \left(\frac{ak}{h}\right)^2\right)v_{xxx} = \epsilon v_{xxxxx} + \cdots \tag{10.49}$$

for some $\epsilon = O(h^4 + k^4)$, and higher order modified equations will also involve only odd-order derivatives and even powers of h and k. Hence the numerical solution produced with the leapfrog method is a fourth order accurate solution to the modified equation (10.45).

Moreover, recall from Section E.3.6 that all higher order odd-order derivatives give dispersive terms. We conclude that the leapfrog method is *nondissipative* at all orders. This conclusion is consistent with the observation in Section 10.2.2 that $k\lambda_p$ is on the boundary of the stability region for all eigenvalues of A from (10.10), and so we see neither growth nor decay of any mode. However, we also see from the form of (10.49) that high wave number modes will not propagate with the correct velocity. This was also true of Lax–Wendroff, but there the fourth order dissipation damps out the worst offenders, whereas with leapfrog this dispersion of often much more apparent in computational results, as observed in Figure 10.4(d).

Example 10.13. Since leapfrog is nondissipative it serves as a nice example for calculating the true dispersion relation of the numerical method. The approach is very similar to the von Neumann stability analysis of Section 10.5, only now we use $e^{i(\xi x_j - \omega t_n)}$ as our Ansatz (so that the g from von Neumann analysis is replaced by $e^{-i\omega k}$). Following the same procedure as in Example 10.4, we find that

$$e^{-i\omega k} = e^{i\omega k} - \frac{ak}{h}\left(e^{i\xi h} - e^{-i\xi h}\right), \tag{10.50}$$

which can be simplified to yield

$$\sin(\omega k) = \frac{ak}{h}\sin(\xi h). \tag{10.51}$$

This is the dispersion relation relating ω to ξ. Note that $|\xi h| \le \pi$ for waves that can be resolved on our grid and that for each such ξh there are two corresponding values of ωk. The dispersion relation is multivalued because leapfrog is a three-level method, and different temporal behavior of the same spatial wave can be seen, depending on the relation between the initial data chosen on the two initial levels. For well-resolved waves ($|\xi h|$ small) and reasonable initial data we expect ωk also near zero (not near $\pm\pi$, where the other solution is in this case).

Solving for ω as a function of ξ and expanding in Taylor series for small ξh would show this agrees with the dispersion relation of the infinite modified equation (10.49). We do not need to do this, however, if our goal is to compute the group velocity for the leapfrog method. We can differentiate (10.51) with respect to ξ and solve for

$$\frac{d\omega}{d\xi} = \frac{a\cos(\xi h)}{\cos(\omega k)} = \pm\frac{a\cos(\xi h)}{\sqrt{1-\nu^2\sin^2(\xi h)}}, \tag{10.52}$$

where $\nu = ak/h$ is the Courant number and again the $\pm$ arises from the multivalued dispersion relation. The velocity observed would depend on how the initial two levels are set.

Note that the group velocity can be negative and near $-a$ for $|\xi h| \approx \pi$. This is not surprising since the leapfrog method has a 3-point centered stencil, and it is possible for numerical waves to travel from right to left although physically there is advection only to the right. This can be observed in some computations, for example, in Figure 10.5 as discussed in Example 10.14.

10.10 Hyperbolic systems

The advection equation $u_t + au_x = 0$ can be generalized to a first order linear system of equations of the form

$$\begin{aligned} u_t + Au_x &= 0, \\ u(x,0) &= \eta(x), \end{aligned} \tag{10.53}$$

where $u : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^s$ and $A \in \mathbb{R}^{s \times s}$ is a constant matrix. (Note that this is not the matrix A from earlier in this chapter, e.g., (10.10).)

This is a system of conservation laws (see Section E.2) with the flux function $f(u) = Au$. This system is called *hyperbolic* if A is diagonalizable with real eigenvalues, so that we can decompose

$$A = R\Lambda R^{-1}, \tag{10.54}$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_s)$ is a diagonal matrix of eigenvalues and $R = [r_1|r_2|\cdots|r_s]$ is the matrix of right eigenvectors. Note that $AR = R\Lambda$, i.e.,

$$Ar_p = \lambda_p r_p \quad \text{for } p = 1,\ 2,\ \ldots,\ s. \tag{10.55}$$

The system is called *strictly hyperbolic* if the eigenvalues are distinct.

10.10.1 Characteristic variables

We can solve (10.53) by changing to the "characteristic variables"

$$w = R^{-1}u \tag{10.56}$$

in much the same way we solved linear systems of ODEs in Section 7.4.2. Multiplying (10.53) by R^{-1} and using (10.54) gives

$$R^{-1}u_t + \Lambda R^{-1}u_x = 0 \tag{10.57}$$

or, since R^{-1} is constant,

$$w_t + \Lambda w_x = 0. \tag{10.58}$$

Since Λ is diagonal, this decouples into s independent scalar equations

$$(w_p)_t + \lambda_p (w_p)_x = 0, \qquad p = 1,\ 2,\ \ldots,\ s. \tag{10.59}$$

Each of these is a constant coefficient linear advection equation with solution

$$w_p(x,t) = w_p(x - \lambda_p t, 0). \tag{10.60}$$

Since $w = R^{-1}u$, the initial data for w_p is simply the pth component of the vector

$$w(x,0) = R^{-1}\eta(x). \tag{10.61}$$

The solution to the original system is finally recovered via (10.56):

$$u(x,t) = Rw(x,t). \tag{10.62}$$

Note that the value $w_p(x,t)$ is the coefficient of r_p in an eigenvector expansion of the vector $u(x,t)$, i.e., (10.62) can be written out as

$$u(x,t) = \sum_{p=1}^{s} w_p(x,t) r_p. \tag{10.63}$$

Combining this with the solutions (10.60) of the decoupled scalar equations gives

$$u(x,t) = \sum_{p=1}^{s} w_p(x - \lambda_p t, 0) r_p. \tag{10.64}$$

Note that $u(x,t)$ depends only on the initial data at the s points $x - \lambda_p t$. This set of points is the domain of dependent $\mathcal{D}(x,t)$ of (10.39).

The curves $x = x_0 + \lambda_p t$ satisfying $x'(t) = \lambda_p$ are the "characteristics of the pth family," or simply "p-characteristics." These are straight lines in the case of a constant coefficient system. Note that for a strictly hyperbolic system, s distinct characteristic curves pass through each point in the x-t plane. The coefficient $w_p(x,t)$ of the eigenvector r_p in the eigenvector expansion (10.63) of $u(x,t)$ is constant along any p-characteristic.

10.11 Numerical methods for hyperbolic systems

Most of the methods discussed earlier for the advection equation can be extended directly to a general hyperbolic system by replacing a with A in the formulas. For example, the Lax–Wendroff method becomes

$$U_j^{n+1} = U_j^n - \frac{k}{2h} A(U_{j+1}^n - U_{j-1}^n) + \frac{k^2}{2h^2} A^2 (U_{j-1}^n - 2U_j^n + U_{j+1}^n). \tag{10.65}$$

This is second order accurate and is stable provided the Courant number is no larger than 1, where the *Courant number* is defined to be

$$\nu = \max_{1 \le p \le s} |\lambda_p k / h|. \tag{10.66}$$

For the scalar advection equation, there is only one eigenvalue equal to a, and the Courant number is simply $|ak/h|$. The Lax–Friedrichs and leapfrog methods can be generalized in the same way to systems of equations and remain stable for $\nu \le 1$.

The upwind method for the scalar advection equation is based on a one-sided approximation to u_x, using data in the upwind direction. The one-sided formulas (10.21) and (10.22) generalize naturally to

$$U_j^{n+1} = U_j^n - \frac{k}{h} A(U_j^n - U_{j-1}^n) \tag{10.67}$$

and

$$U_j^{n+1} = U_j^n - \frac{k}{h} A(U_{j+1}^n - U_j^n). \tag{10.68}$$

For a system of equations, however, neither of these is useful unless all the eigenvalues of A have the same sign, so that the upwind direction is the same for all characteristic variables. The method (10.67) is stable only if

$$0 \le \frac{k\lambda_p}{h} \le 1 \qquad \text{for all } p = 1, 2, \ldots, s, \tag{10.69}$$

while (10.68) is stable only if

$$-1 \le \frac{k\lambda_p}{h} \le 0 \qquad \text{for all } p = 1, 2, \ldots, s. \tag{10.70}$$

It is possible to generalize the upwind method to more general systems with eigenvalues of both signs, but to do so requires decomposing the system into the characteristic variables and upwinding each of these in the appropriate direction. The resulting method can also be generalized to nonlinear hyperbolic systems and generally goes by the name of *Godunov's method*. These methods are described in much more detail in [66].

10.12 Initial boundary value problems

So far we have studied only numerical methods for hyperbolic problems on a domain with periodic boundary conditions (or the Cauchy problem, if we could use a grid with an infinite number of grid points).

Most practical problems are posed on a bounded domain with nonperiodic boundary conditions, which must be specified in addition to the initial conditions to march forward in time. These problems are called *initial boundary value problems* (IBVPs).

Consider the advection equation $u_t + au_x = 0$ with $a > 0$, corresponding to flow to the right, on the domain $0 \le x \le 1$ where some initial conditions $u(x, 0) = \eta(x)$ are given. This data completely determine the solution via (10.2) in the triangular region $0 \le x - at \le 1$ of the x-t plane. Outside this region, however, the solution is determined only if we also impose boundary conditions at $x = 0$, say, (10.8). Then the solution is

$$u(x,t) = \begin{cases} \eta(x - at) & \text{if } 0 \le x - at \le 1, \\ g_0(t - x/a) & \text{otherwise.} \end{cases} \tag{10.71}$$

Note that boundary data are required only at the *inflow boundary* $x = 0$, not at the *outflow boundary* $x = 1$, where the solution is determined via (10.71). Trying to impose a different value on $u(1, t)$ would lead to a problem with no solution.

If $a < 0$ in the advection equation, then $x = 1$ is the inflow boundary, the solution is transported to the left, and $x = 0$ is the outflow boundary.

10.12.1 Analysis of upwind on the initial boundary value problem

Now suppose we apply the upwind method (10.21) to this IBVP with $a > 0$ on a grid with $h = 1/(m + 1)$ and $x_i = ih$ for $i = 0, 1, \ldots, m + 1$. The formula (10.21) can be applied for $i = 1, \ldots, m + 1$ in each time step, while $U_0^n = g(t_n)$ is set by the boundary condition. Hence the method is completely specified.

When is this method stable? Intuitively we expect it to be stable if $0 \le ak/h \le 1$. This is the stability condition for the problem with periodic boundary conditions, and here we are using the same method at every point except $i = 0$, where the exact solution is being set in each time step. Our intuition is correct in this case and the method is stable if $0 \le ak/h \le 1$.

Notice, however, that von Neumann analysis cannot be used in this case, as discussed already in Section 10.5: the Fourier modes $e^{i\xi h}$ are no longer eigengridfunctions. But von Neumann analysis is still useful because it generally gives a necessary condition for stability. In most cases a method that is unstable on the periodic domain or Cauchy problem will not be useful on a bounded domain either, since locally on a fine grid, away from the boundaries, any instability indicated by von Neumann analysis is bound to show up.

Instead we can use MOL stability analysis, although it is sometimes subtle to do so correctly. We have a system of ODEs similar to (10.9) for the vector $U(t)$, which again has $m + 1$ components corresponding to $u(x_i, t)$. But now we must incorporate the boundary conditions, and so we have a system of the form

$$U'(t) = AU(t) + g(t), \tag{10.72}$$

where

$$A = -\frac{a}{h}\begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & & \\ & & & -1 & 1 \end{bmatrix}, \qquad g(t) = \begin{bmatrix} g_0(t)a/h \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{10.73}$$

The upwind method corresponds to using Euler's method on the ODE (10.72).

The change from the matrix of (10.10) to (10.73) may seem trivial but it completely changes the character of the matrix. The matrix (10.10) is circulant and normal and has eigenvalues uniformly distributed about the circle of radius a/h in the complex plane centered at $z = -a/h$. The matrix (10.73) is a defective Jordan block with all its eigenvalues at the point $-a/h$. The eigenvalues have moved distance $a/h \to \infty$ as $h \to 0$.

Suppose we attempt to apply the usual stability analysis of Chapter 7 to this system and require that $k\lambda_p \in \mathcal{S}$ for all eigenvalues of A, where S is the stability region for Euler's method. Since $\mathcal{S}$ contains the interval $[-2, 0]$ on the real axis, this would suggest the stability restriction

$$0 \le ak/h \le 2 \tag{10.74}$$

for the upwind method on the IBVP. This is wrong by a factor of 2. It is a necessary condition but not sufficient.

The problem is that A in (10.73) is highly nonnormal. It is essentially a Jordan block of the sort discussed in Section D.5.1, and on a fine grid its ϵ-pseudospectra roughly fill up the circle of radius a/h about $-a/h$, even for very small ϵ. This is a case where we need to apply a more stringent requirement than simply requiring that $k\lambda$ be inside the stability region for all eigenvalues; we also need to require that

$$\operatorname{dist}(k\lambda_\epsilon, \mathcal{S}) \le C\epsilon \tag{10.75}$$

holds for the ϵ-pseudoeigenvalues (see Section D.5), where C is a modest constant, as suggested in [47], [92]. Requiring (10.75) shows that the expected requirement $0 \le ak/h \le 1$ is needed rather than (10.74).

10.12.2 Outflow boundary conditions

When the upwind method is used for the advection equation, as in the previous section, the outflow boundary poses no problem. The finite difference formula is one sided and the value U^n_{m+1} at the rightmost grid point is computed by the same formula that is used in the interior.

However, if we use a finite difference method whose stencil extends to the right as well as the left, we will need to use a different formula at the rightmost point in the domain. This is called a *numerical boundary condition* or *artificial boundary condition* since it is required by the method, not for the PDE. Numerical boundary conditions may also be required at boundaries where a physical boundary condition is imposed, if the numerical method requires more conditions than the equation. For example, if we use the Beam–Warming method for advection, which has a stencil that extends two grid points in the upwind direction, then we can use this only for updating U^{n+1}_2, U^{n+1}_3, ... The value U^{n+1}_0 will be set by the physical boundary condition but U^{n+1}_1 will have to be set by some other method, which can be viewed as a numerical boundary condition for Beam–Warming.

Naturally some care must be used in choosing numerical boundary conditions, both in terms of the accuracy and stability of the resulting method. This is a difficult topic, particularly the stability analysis of numerical methods for IBVPs, and we will not pursue it here. See, for example, [40], [84], [89].

Example 10.14. We will look at one example simply to give a flavor of the potential difficulties. Suppose we use the leapfrog method to solve the IBVP for the advection equation $u_t + au_x = 0$. At the left (inflow) boundary we can use the given boundary condition, but at the right we will need a numerical boundary condition. Suppose we use the first order upwind method at this point,

$$U^{n+1}_{m+1} = U^n_{m+1} - \frac{ak}{h}(U^n_{m+1} - U^n_m), \tag{10.76}$$

which is also consistent with the advection equation. Figure 10.5 shows four snapshots of the solution when the initial data is $u(x,0) = \eta(x) = \exp(-5(x-2)^2)$ and the first two time levels for leapfrog are initialized based on the exact solution $u(x,t) = \eta(x-at)$. We see that as the wave passes out the right boundary, a reflection is generated that moves to the left, back into the domain. The dispersion relation for the leapfrog method found in Example 10.13 shows that waves with $\xi h \approx \pi$ can move to the left with group velocity approximately equal to $-a$, and this wave number corresponds exactly to the sawtooth wave seen in the figure.

For an interesting discussion of the relation of numerical dispersion relations and group velocity to the stability of numerical boundary conditions, see [89].

Outflow boundaries are often particularly troublesome. Even if a method is formally stable (as the leapfrog method in the previous example is with the upwind boundary condition), it is often hard to avoid spurious reflections. We have seen this even for the advection equation, where in principle flow is entirely to the right, and it can be even more difficult

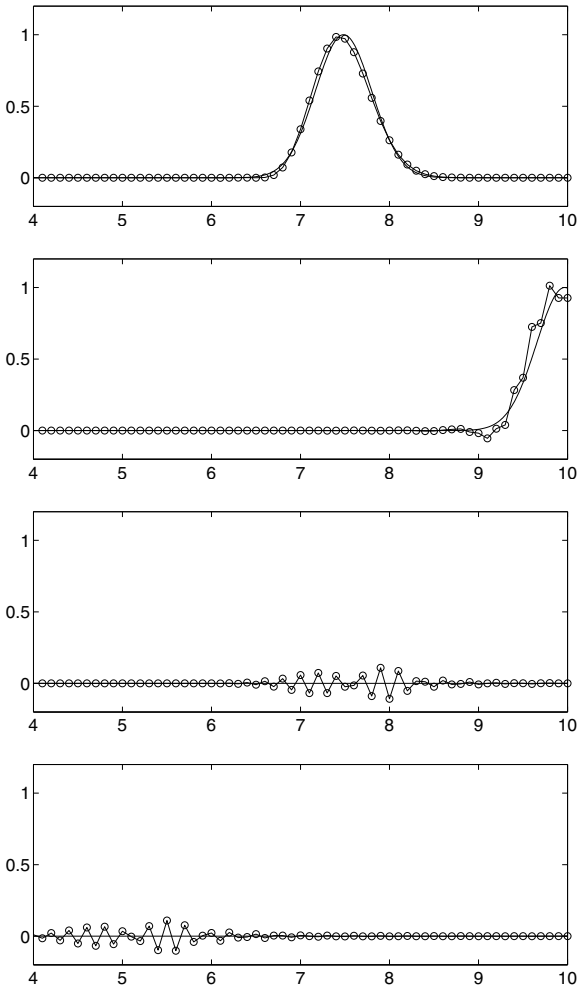

Figure 10.5. *Numerical solution of the advection equation using the leapfrog method in the interior and the upwind method at the right boundary. The solution is shown at four equally spaced times, illustrating the generation and leftward propagation of a sawtooth mode.*

to develop appropriate boundary conditions for wave propagation or fluid dynamics equations that admit wave motion in all directions. Yet in practice we always have to compute over a finite domain and this often requires setting artificial boundaries around the region of interest. The assumption is that the phenomena of interest happen within this region, and the hope is that any waves hitting the artificial boundary will leave the domain with no reflection. Numerical boundary conditions that attempt to achieve this are often called *nonreflecting* or *absorbing* boundary conditions.

10.13 Other discretizations

As in the previous chapter on parabolic equations, we have concentrated on a few basic methods in order to explore some fundamental ideas. We have also considered only the simplest case of constant coefficient linear hyperbolic equations, whereas in practice most hyperbolic problems of interest have variable coefficients (e.g., linear wave propagation in heterogeneous media) or are nonlinear. These problems give rise to a host of new difficulties, not least of which is the fact that the solutions of interest are often discontinuous since nonlinearity can lead to shock formation. There is a well-developed theory of numerical methods for such problems that we will not delve into here; see, for example, [66].

Even in the case of constant coefficient linear problems, there are many other discretizations possible beyond the ones presented here. We end with a brief overview of just a few of these:

- Higher order discretizations of u_x can be used in place of the discretizations considered so far. If we write the MOL system for the advection equation as

$$U_j'(t) = -aW_j(t), \tag{10.77}$$

  where $W_j(t)$ is some approximation to $u_x(x_j, t)$, then there are many ways to approximate $W_j(t)$ from the U_j values beyond the centered approximation (10.3). One-sided approximations are one possibility, as in the upwind method. For sufficiently smooth solutions we might instead use higher order accurate centered approximations, e.g.,

$$W_j = \frac{4}{3}\left(\frac{U_{j+1} - U_{j-1}}{2h}\right) - \frac{1}{3}\left(\frac{U_{j+2} - U_{j-2}}{4h}\right). \tag{10.78}$$

  This and other approximations can be determined using the `fdcoeffV.m` routine discussed in Section 1.5; e.g., `fdcoeffV(1,0,-2:2)` produces (10.78). Centered approximations such as (10.78) generally lead to skew-symmetric matrices with pure imaginary eigenvalues, at least when applied to the problem with periodic boundary conditions. In practice most problems are on a finite domain with nonperiodic boundary conditions, and other issues arise as already seen in Section 10.12. Note that the discretization (10.78) requires more numerical boundary conditions than the upwind or second order centered operator.

- An interesting approach to obtaining better accuracy in W_j is to use a so-called compact method, in which the W_j are determined by solving a linear system rather than explicitly. A simple example is

$$\frac{1}{4}W_{j-1} + W_j + \frac{1}{4}W_{j+1} = \frac{3}{2}\left(\frac{U_{j+1} - U_{j-1}}{2h}\right). \tag{10.79}$$

  This gives a tridiagonal system of equations to solve for the W_j values, and it can be shown that the resulting values will be $O(h^4)$ approximations to $u_x(x_j, t)$. Higher order methods of this form also exist; see Lele [63] for an in-depth discussion. In addition to giving higher order of accuracy with a compact stencil, these approximations also typically have much better dispersion properties than standard finite difference approximations of the same order.

- Spectral approximations to the first derivative can be used, based on the same ideas as in Section 2.21. In this case $W = DU$ is used to obtain the approximations to the first derivative, where D is the dense spectral differentiation matrix. These methods can also be generalized to variable coefficient and even nonlinear problems and often work very well for problems with smooth solutions.

  Stability analysis of these methods can be tricky. To obtain reasonable results a nonuniform distribution of grid points must be used, such as the Chebyshev extreme points as discussed in Section 2.21. In this case the eigenvalues of the matrix D turn out to be $O(1/h^2)$, rather than $O(1/h)$ as is expected for a fixed-stencil discretization of the first derivative. If instead we use the roots of the Legendre polynomial (see Section B.3.1), another popular choice of grid points for spectral methods, it can be shown that the eigenvalues are $O(1/h)$, which appears to be better. In both cases, however, the matrix D is highly nonnormal and the eigenvalues are misleading, and in fact a time step $k = O(h^2)$ is generally required if an explicit method is used for either choice of grid points; see, e.g., [92].

- Other time discretizations can be used in place of the ones discussed in this chapter. In particular, for spectral methods the MOL system is stiff and it may be beneficial to use an implicit method as discussed in Chapters 8 and 9. Another possibility is to use an exponential time differencing method, as discussed in Section 11.6.

- For conservation laws (see Section E.2) numerical methods are often more naturally derived using the integral form (E.9) than by using finite difference approximations to derivatives. Such methods are particularly important for nonlinear hyperbolic problems, where shock waves (discontinuous solutions) can develop spontaneously even from smooth initial data. In this case the discrete value U_i^n is viewed as an approximation to the cell average of $u(x, t_n)$ over the grid cell $[x_{i-1/2},\ x_{i+1/2}]$ of length h centered about x_i,

$$U_i^n \approx \frac{1}{h} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t_n)\, dx. \tag{10.80}$$

  According to (E.9) this cell average evolves at a rate given by the difference of fluxes at the cell edges, and a particular numerical method is obtained by approximating these fluxes based on the current cell averages. Methods of this form are often called *finite volume methods* since the spatial domain is partitioned into volumes of finite size. Simple finite volume methods often look identical to finite difference methods, but the change in viewpoint allows the development of more sophisticated methods that are better suited to solving nonlinear conservation laws. These methods also have some advantages for linear problems, particularly if they have variable coefficients with jump discontinuities, as often arises in solving wave propagation problems in heterogeneous media. See [66] for a detailed description of such methods.