Chapter 13

Advection Equations

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

$$u_t + au_x = 0 \tag{13.1}$$

where a is a constant. See Section 10.3 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 is so simple that we can write down the exact solution,

$$u(x,t) = \eta(x - at). \tag{13.2}$$

One can verify directly that this is the solution (see also Chapter 11). However, many of the issues that arise more generally in discretizing hyperbolic equations can be most easily seen with this equation. Other hyperbolic systems will be discussed later.

The first approach we might consider is the analog of the method (12.4) for the heat equation. Using the centered difference in space and the forward difference in time results in

$$\frac{U_j^{n+1} - U_j^n}{k} = -\frac{a}{2h}(U_{j+1}^n - U_{j-1}^n), \tag{13.3}$$

which can be rewritten as

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n). \tag{13.4}$$

This again has the stencil shown in Figure 12.1(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 (13.4) by the average $\frac{1}{2}(U_{j-1}^n + U_{j+1}^n)$ then we obtain the *Lax-Friedrichs method*,

$$U_j^{n+1} = \frac{1}{2}(U_{j-1}^n + U_{j+1}^n) - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n). \tag{13.5}$$

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 (13.4) 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 12.5) and convergent provided

$$\left|\frac{ak}{h}\right| \le 1. \tag{13.6}$$

Note that this stability restriction allows us to use a time step $k = O(h)$ even though 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 (13.6) 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 (13.6) 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 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 (13.6).

13.1 MOL discretization

To investigate stability further we will again introduce the method of lines (MOL) discretization as we did in Section 12.2 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 only on one of the two boundaries. If $a > 0$ then we need a boundary condition at $x = 0$, say

$$u(0,t) = g_0(t), \tag{13.7}$$

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 later.

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 only need to model 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 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:

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

This system can be written as

$$U'(t) = AU(t) \tag{13.8}$$

with

$$A = -\frac{a}{2h}\begin{bmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 1 & & & & -1 & 0 \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)}. \tag{13.9}$$

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

$$\lambda_p = -\frac{ia}{h}\sin(2\pi ph), \quad \text{for } p = 1, 2, \ldots, m+1. \tag{13.10}$$

The corresponding eigenvector u^p has components

$$u_j^p = e^{2\pi ipjh}, \quad \text{for } j = 1, 2, \ldots, m+1. \tag{13.11}$$

The eigenvalues lie on the imaginary axis between $-ia/h$ and ia/h. Note how this relates to the eigenvalues and eigenfunctions of $-a\partial_x$.

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 be stable provided $|ak/h| \le b$.

13.1.1 Forward Euler time discretization

The method (13.4) can be viewed as the forward Euler time discretization of the MOL system of ODE's (13.8). We found in Section 8.3 that this method is only stable 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 (13.10) are imaginary, the values $k\lambda_p$ will not lie in $\mathcal{S}$. Hence the method (13.4) is *unstable* for any fixed mesh ratio k/h; see Figure 13.1(a).

The method (13.4) is 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 (12.19), $\|B\| \le 1 + \alpha k$, 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 $\|I + kA\|_2^2 \leq 1 + a^2k$ and if $nk \leq T$ we have

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

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

13.1.2 Leapfrog

A better time discretization is to use the midpoint method (6.17),

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

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

$$U_j^{n+1} = U_j^{n-1} - \frac{ak}{h}(U_{j+1}^n - U_{j-1}^n). \tag{13.12}$$

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

Recall from Section 8.3 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 satsified.

13.1.3 Lax-Friedrichs

Now consider the Lax-Friedrichs method (13.5). Note that we can rewrite (13.5) 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

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{1}{2}(U_{j-1}^n - 2U_j^n + U_{j+1}^n). \tag{13.13}$$

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 $\epsilon = h^2/2k$. Actually, we will see later that it is in fact a *second order* accurate method on a slightly different advection-diffusion equation. Viewing Lax-Friedrichs in this way allows us to investigate the diffusive nature of the method quite precisely.

For our present purposes, however, the crucial part is that we can now view (13.13) as resulting from a Forward Euler discretization of the system of ODE's

$$U'(t) = BU(t)$$

with

$$B = -\frac{a}{2h}\begin{bmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 1 & & & & -1 & 0 \end{bmatrix} + \frac{\epsilon}{h^2}\begin{bmatrix} -2 & 1 & & & & 1 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ 1 & & & & 1 & -2 \end{bmatrix} \tag{13.14}$$

where $\epsilon = h^2/2k$. The matrix B differs from the matrix A of (13.9) 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 B 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 (13.11) of the matrix A are also eigenvectors of the second difference operator (with periodic boundary conditions) that appears in (13.14), and hence these are also the eigenvectors of the full matrix B. We can easily compute that the eigenvalues of B are

$$\mu_p = -\frac{ia}{h}\sin(2\pi ph) - \frac{2\epsilon}{h^2}(1 - \cos(2\pi ph)). \tag{13.15}$$

The values $k\mu_p$ are plotted in the complex plane for various different values of ϵ in Figure 13.1. 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 $|ak/h| \leq 1$. (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 $|ak/h| \leq 1$.

13.2 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 ODE's (13.8) 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 (13.12) already discussed. However, this is a 3-level method and for various reasons it is often much more convenient to use 2-level methods for PDE's 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 15) may require 2-level methods for each step, to name a few reasons. Moreover, the Leapfrog method is "nondissipative" in a sense that will be discussed in Section 13.6, 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 dicretize the system (13.8), 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 2-stage Runge-Kutta method such as the one in Example 6.11 for the time discretization. This can be done, though some care must be exercised near boundaries and the use of a multi-stage method again typically requires additional storage.

One simple way to achieve a 2-level explicit method with higher accuracy is to use the idea of Taylor series methods, as described in Section 6.5. Applying this directly to the linear system of ODE's $U'(t) = AU(t)$ (and using $U'' = AU' = A^2U$) gives the second order method

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

Here A is the matrix (13.9) and computing A^2 and writing the method at the typical grid point then gives

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{a^2k^2}{8h^2}(U_{j-2}^n - 2U_j^n + U_{j+2}^n). \tag{13.16}$$

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.

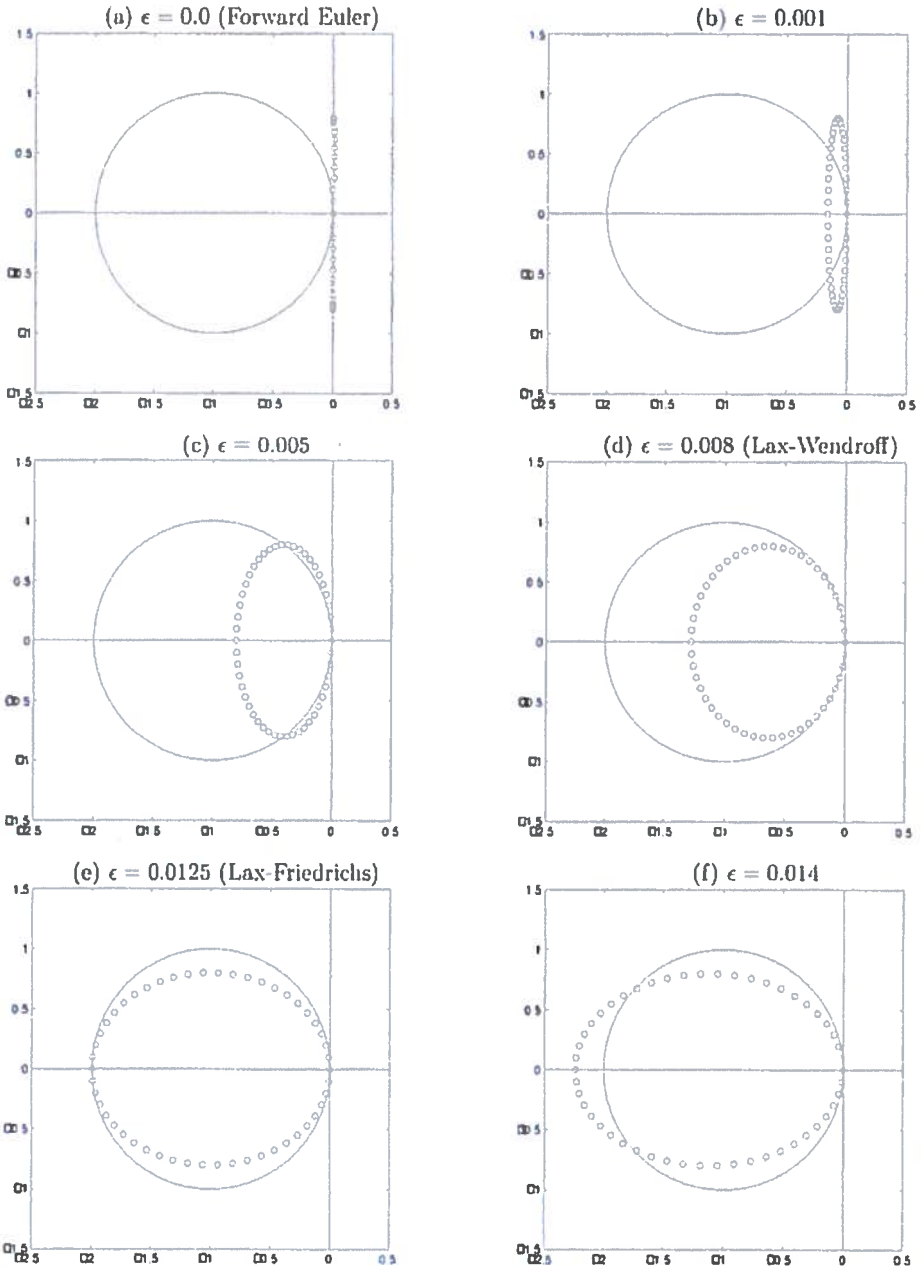

Figure 13.1. Eigenvalues of the matrix B in (13.14), for various values of ϵ, in the case $h = 1/50$ and $k = 0.8h$, $a = 1$, so $ak/h = 0.8$. (a) shows the case $\epsilon = 0$ which corresponds the forward Euler method (13.4). (d) shows the case $\epsilon = a^2k/2$, the Lax-Wendroff method (13.17). (e) shows the case $\epsilon = h^2/2k$ the Lax-Friedrichs method (13.5). The method is stable for ϵ between $a^2k/2$ and $h^2/2k$, as in figures (d) through (e).

Note that the last term in (13.16) is an approximation to $\frac{1}{2}a^2k^2u_{xx}$ using a centered difference based on stepsize $2h$. 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*:

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{a^2k^2}{2h^2}(U_{j-1}^n - 2U_j^n + U_{j+1}^n). \tag{13.17}$$

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 (13.17). It is also clear how we could obtain higher-order accurate explicit 2-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 PDE's. 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.

13.2.1 Stability analysis

We can analyze the stability of Lax-Wendroff following the same approach used for Lax-Friedrichs in Section 13.1. Note that with periodic boundary conditions, the Lax-Wendroff method (13.17) can be viewed as Euler's method applied to the linear system of ODE's $U'(t) = BU(t)$ where B is given by (13.14) with $\epsilon = a^2k/2$ (instead of the value $\epsilon = h^2/2k$ used in Lax-Friedrichs). The eigenvalues of B are given by (13.15) 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 semiaxes of length $(ak/h)^2$ and $|ak/h|$. If $|ak/h| \le 1$ then all of these values lie inside the stability region of Euler's method. Figure 13.1(d) shows an example in the case $ak/h = 0.8$ The Lax-Wendroff method is stable with exactly the same time step restriction (13.6) as required for Lax-Friedrichs. In Section 13.5 we will see that this is a very natural stability condition to expect for the advection equation, and the best we could hope for when a 3-point method is used.

A close look at Figure 13.1 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 13.1(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 S) and resolving the behavior of these modes properly would require a finer grid.

Comparing Figures 13.1(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 13.1(c) would lead to instability, while more damping as in Figure 13.1(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.

13.2.2 Von Neumann analysis

The stability criterion for the Lax-Wendroff method can also be determined using von Neumann analysis as described in Section 12.6. Setting $\nu = ak/h$ and following the procedure of Example 12.5 for the Lax-Wendroff method (13.17) gives

$$\begin{aligned} g &= 1 - \frac{1}{2}i\nu(e^{i\xi h} - e^{-i\xi h}) + \frac{1}{2}\nu^2(e^{i\xi h} - 2 + e^{-i\xi h}) \\ &= 1 - i\nu\sin(\xi h) + \nu^2(\cos(\xi h) - 1) \\ &= 1 - i\nu[2\sin(\xi h/2)\cos(\xi h/2)] + \nu^2[2\sin^2(\xi h/2)] \end{aligned} \tag{13.18}$$

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

$$\begin{aligned} |g|^2 &= [1 - 2\nu^2\sin^2(\xi h/2)]^2 + 4\sin^2(\xi h/2)\cos^2(\xi h/2) \\ &= 1 - 4\nu^2(1 - \nu^2)\sin^4(\xi h/2). \end{aligned} \tag{13.19}$$

Since $0 \le \sin^4(\xi h/2) \le 1$ for all values of ξ, we see that $|g|^2 \le 1$ for all ξ and hence the method is stable provided that $|\nu| \le 1$, which again gives the expected stability bound (13.6).

13.3 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.*,

$$u_x(x_j, t) \approx \frac{1}{h}(U_j - U_{j-1}) \tag{13.20}$$

or

$$u_x(x_j, t) \approx \frac{1}{h}(U_{j+1} - U_j). \tag{13.21}$$

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:

$$U_j^{n+1} = U_j^n - \frac{ak}{h}(U_j^n - U_{j-1}^n) \tag{13.22}$$

or

$$U_j^{n+1} = U_j^n - \frac{ak}{h}(U_{j+1}^n - U_j^n). \tag{13.23}$$

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 due to the fact that 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. We will see that there are situations where it is best to acknowledge this asymmetry and use one-sided differences in the appropriate direction. In practice we often want to also use more accurate approximations, however, and we will see later how to extend these methods to higher order accuracy.

The choice between the two methods (13.22) and (13.23) 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 (13.22) might be a better choice for $a > 0$ and (13.23) for $a < 0$.

In fact the stability analysis below shows that (13.22) is stable only if

$$0 \le \frac{ak}{h} \le 1. \tag{13.24}$$

Since k and h are positive, we see that this method can only be used 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, (13.23) is stable only if

$$-1 \le \frac{ak}{h} \le 0, \tag{13.25}$$

and can only be used if $a < 0$. In this case (13.23) is the proper upwind method to use.

13.3.1 Stability analysis

The method (13.22) can be written as

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(U_{j+1}^n - U_{j-1}^n) + \frac{ak}{2h}(U_{j+1}^n - 2U_j^n + U_{j-1}^n), \tag{13.26}$$

which puts it in the form (13.14) 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 (13.24).

If we think of (13.26) 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 Chapter 11).

The method (13.23) can also be written in a form similar to (13.26), 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 (13.25).

The three methods Lax-Wendroff, upwind, and Lax-Friedrichs, can all be written in the same form (13.14) with different values of ϵ. If we call these values ϵ_{LW}, ϵ_{up}, and ϵ_{LF} respectively, then we have

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

where $\nu = ak/h$. Note that

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

If $0 < \nu < 1$ 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 13.1.

13.3.2 The Beam-Warming method

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

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(3U_j^n - 4U_{j-1}^n + U_{j-2}^n) + \frac{a^2k^2}{2h^2}(U_j^n - 2U_{j-1}^n + U_{j-2}^n). \tag{13.27}$$

Figure 13.2: Tracing the characteristic of the advection equation back in time from the point (x_j, t_{n+1}) to compute the solution according to (13.29). 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 $a < 0$.

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

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}(-3U_j^n + 4U_{j+1}^n - U_{j+2}^n) + \frac{a^2k^2}{2h^2}(U_j^n - 2U_{j+1}^n + U_{j+2}^n). \tag{13.28}$$

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

13.4 Characteristic tracing and interpolation

The solution to the advection equation is given by (13.2). The value of u is constant along each characteristic, which for this example are straight lines with constant slope. Over a single time step we have

$$u(x_j, t_{n+1}) = u(x_j - ak, t_n). \tag{13.29}$$

Tracing this characteristic back over time step k from the grid point x_j results in the picture shown in Figure 13.2(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

$$U_j^{n+1} = U_{j-1}^n. \tag{13.30}$$

Actually, all of the 2-level methods that we have considered so far reduce to the formula (13.30) in this special case $ak = h$, and each of these methods happens to be exact in this case.

If $ak/h < 1$ then the point $x_j - ak$ is not exactly at a grid point, as illustrated in Figure 13.2. However, we might attempt to use the relation (13.29) 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

$$p(x) = U_j^n + (x - x_j)\left(\frac{U_j^n - U_{j-1}^n}{h}\right). \tag{13.31}$$

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 (13.22). Note that this can also be interpreted as a linear combination of the two values U_{j-1}^n and U_j^n:

$$U_j^{n+1} = \left(1 - \frac{ak}{h}\right)U_j^n + \frac{ak}{h}U_{j-1}^n. \tag{13.32}$$

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 (13.24) is satisfied, which is also the condition required to insure 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_j^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 (13.17). Note that in this case we are properly interpolating provided that the stability restriction $|ak/h| \leq 1$ 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 (13.27), and we are properly interpolating provided $0 \leq ak/h \leq 2$.

13.5 The CFL Condition

The discussion of Section 13.4 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 $p \leq q$ (negative values are allowed for p and q), then we must have $x_{j+p} \leq x_j - ak \leq x_{j+q}$ or the method cannot be convergent. Since $x_i = ih$, this requires

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

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 essentially the first paper on the stability and convergence of finite difference methods for partial differential equations. (The original paper [CFL28] is in German but an English translation is available in [CFL67].) 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.*, [Kev90], [LeV02] 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 13.8 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

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

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 is felt everywhere at any small time in the future (though its effect of course decays exponentially away from this point).

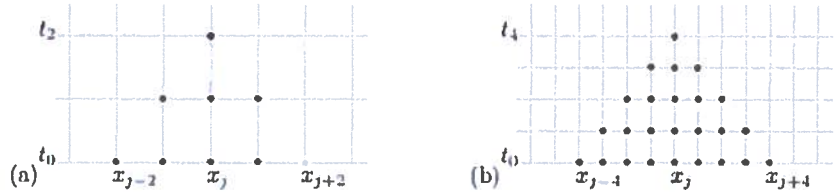

Figure 13.3: (a) Numerical domain of dependence of a grid point when using a 3-point explicit method. (b) On a finer grid.

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 (13.17) 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 13.3(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 13.3(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.

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 only be possible if

$$X - T/r \le X - aT \le X + T/r. \tag{13.33}$$

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 (13.33) 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 like within one grid point of x_j (recall the discussion of interpolation vs. extrapolation in Section 13.4).

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 13.1. The 3-point method (13.4) 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 13.2. The upwind methods (13.22) and (13.23) each have a 2-point stencil and the stability restrictions of these methods, (13.24) and (13.25) respectively, agree precisely with what the CFL condition requires.

Example 13.3. The Beam-Warming method (13.27) has a 3-point one-sided stencil. The CFL condition is satisfied if $0 \le ak/h \le 2$. When $a < 0$ the method (13.28) 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 (Exercise 13.4).

Example 13.4. 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 13.1.1 that the 3-point explicit method (12.5) 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 (12.7) 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).

13.6 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.

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 PDE's 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.

13.6.1 Upwind

This is best illustrated with an example. Consider the upwind method (13.22) for the advection equation $u_t + au_x = 0$ in the case $a > 0$,

$$U_j^{n+1} = U_j^n - \frac{ak}{h}(U_j^n - U_{j-1}^n). \tag{13.34}$$

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 (13.34) 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:

$$v_t + av_x = \frac{1}{2}(ahv_{xx} - kv_{tt}) \tag{13.35}$$

This involves second derivatives in both x and t, but we can derive a slightly different modified equation with the same accuracy by differentiating (13.35) 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 (13.35) 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

$$v_t + av_x = \frac{1}{2}ah\left(1 - \frac{ak}{h}\right)v_{xx}. \tag{13.36}$$

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 only give 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 13.4.

Note that the diffusion coefficient in (13.36) is $\frac{1}{2}(ah - a^2 k)$, 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.

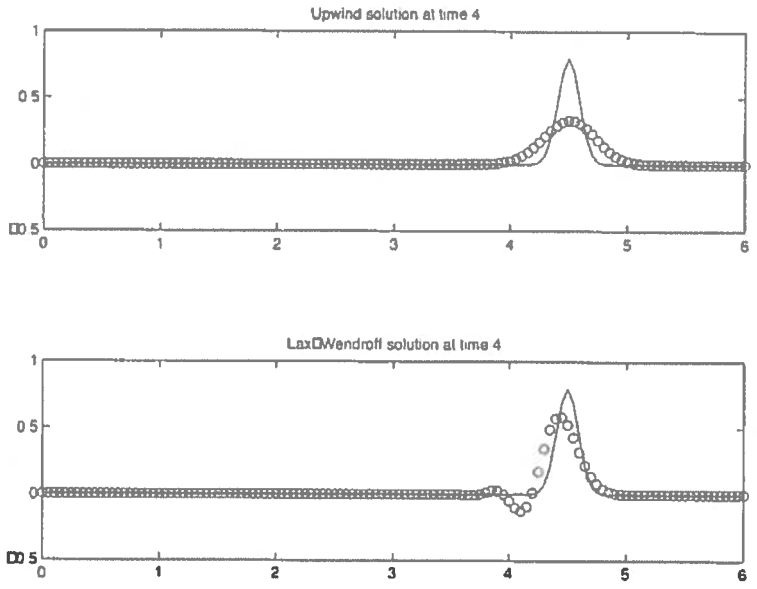

Figure 13.4: Numerical solution using upwind (diffusive) and Lax-Wendroff (dispersive) methods.

13.6.2 Lax-Wendroff

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

$$v_t + av_x + \frac{1}{6}ah^2\left(1 - \left(\frac{ak}{h}\right)^2\right)v_{xxx} = 0. \tag{13.37}$$

The Lax-Wendroff method produces a *third order* accurate solution to this equation. This equation has a very different character from (13.35). The v_{xxx} term leads to *dispersive* behavior rather than diffusion. This is clearly seen in Figure 13.4, 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*.

The group velocity for wave number ξ under Lax-Wendroff 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. (The concept of group velocity is explained in Section 13.7.) 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.

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

$$v_t + av_x + \frac{1}{6}ah^2\left(1 - \left(\frac{ak}{h}\right)^2\right)v_{xxx} = -\epsilon v_{xxxx}, \tag{13.38}$$

where the ϵ in the fourth order dissipative term is $O(h^3)$ and positive when the stability bound holds. This higher order dissipation causes the highest wave numbers to be damped, 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 disasterous, for example if the value of u represents a concentration that cannot go negative or exceed some limit without difficulties arising elsewhere in the modeling process.

13.6.3 Beam-Warming

The Beam-Warming method (13.27) has a similar modified equation,

$$v_t + av_x = \frac{1}{6}ah^2\left(2 - \frac{3ak}{h} + \left(\frac{ak}{h}\right)^2\right)v_{xxx}. \tag{13.39}$$

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 bump. If $1 < ak/h < 2$ then the group velocity is less than a and the oscillations fall behind.

13.7 Dispersive waves

This section contains a short introduction to the theory of dispersive waves, useful in understanding the behavior of many finite difference methods. See, e.g, [Kev90], [Str89], [Whi74] for further details.

Section 11.4 contains a brief discussion of Fourier analysis of the dispersive equation $u_t = u_{xxx}$. Extending that analysis to an equation of the form

$$u_t + au_x + bu_{xxx} = 0, \tag{13.40}$$

we find that the solution can be written as

$$u(x,t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\hat{\eta}(\xi)e^{i\xi(x-(a-b\xi^2)t)}\,d\xi,$$

where $\hat{\eta}(\xi)$ is the Fourier transform of the initial data $\eta(x)$. Each Fourier mode $e^{i\xi x}$ propagates at velocity $a - b\xi^2$, called the phase velocity of this wave number. In general the initial data $\eta(x)$ is a linear combination of infinitely many different Fourier modes. For $b \neq 0$ these modes propagate at different speeds relative to one another. Their peaks and troughs will be shifted relative to other modes and they will no longer add up to a shifted version of the original data. The waves are called dispersive since the different modes do not move in tandem. Moreover we will see below that the "energy" associated with different wave numbers also disperses.

13.7.1 The dispersion relation

Consider a more general PDE of the form

$$u_t + a_1u_x + a_3u_{xxx} + a_5u_{xxxxx} + \cdots = 0 \tag{13.41}$$

that contains only odd-order derivative in x. The Fourier transform $\hat{u}(\xi,t)$ satisfies

$$\hat{u}_t(\xi,t) + a_1i\xi\hat{u}(\xi,t) - a_3i\xi^3\hat{u}(\xi,t) + a_5i\xi^5\hat{u}(\xi,t) + \cdots = 0,$$

and hence

$$\hat{u}(\xi,t) = e^{-i\omega t}\hat{\eta}(\xi),$$

where

$$\omega = \omega(\xi) = a_1\xi - a_3\xi^3 + a_5\xi^5 - \cdots. \tag{13.42}$$

The solution can thus be written as

$$u(x,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{\eta}(\xi) e^{i(\xi x - \omega(\xi)t)}\, d\xi. \tag{13.43}$$

The relation (13.42) between ξ and ω is called the *dispersion relation* for the PDE. Once we've gone through this full Fourier analysis a couple times we realize that since the different wave numbers ξ decouple, the dispersion relation for a linear PDE can be found simply by substituting a single Fourier mode of the form

$$u(x,t) = e^{-i\omega t} e^{i\xi x} \tag{13.44}$$

into the PDE and cancelling the common terms in order to find the relation between ω and ξ. This is similar to what we found when applying von Neumann analysis in Section 12.6. In fact there is a close relation between determining the dispersion relation and doing von Neumann analysis, and the dispersion relation for a finite difference method can be defined by an approach similar to von Neumann analysis by setting $U_j^n = e^{-i\omega nk} e^{i\xi jh}$, *i.e.*, using $e^{-i\omega k}$ in place of g.

Note that this same analysis can also be done for equations that involve even order derivatives, such as

$$u_t + a_1 u_x + a_2 u_{xx} + a_3 u_{xxx} + a_4 u_{xxxx} + \cdots = 0,$$

but then we find that

$$\omega(\xi) = a_1\xi + ia_2\xi^2 - a_3\xi^3 - ia_4\xi^4 - \cdots.$$

The even order derivatives give imaginary terms in $\omega(\xi)$ so that

$$e^{-i\omega t} = e^{(a_2\xi^2 - a_4\xi^4 + \cdots)t}\, e^{i(a_1\xi - a_3\xi^3 + \cdots)t}.$$

The first term gives exponential growth or decay, as we expect from Section 11.3, rather than dispersive behavior. For this reason we call the PDE (purely) dispersive only if $\omega(\xi)$ is real for all $\xi \in \mathbb{R}$. Informally we also speak of an equation like $u_t = u_{xx} + u_{xxx}$ as having both a diffusive and a dispersive term.

In the purely dispersive case (13.41) the single Fourier mode (13.44) can be written as

$$u(x,t) = e^{i\xi(x - (\omega/\xi)t)}$$

and so a pure mode of this form propagates at velocity ω/ξ. This is called the *phase velocity* for this wave number,

$$c_p(\xi) = \frac{\omega(\xi)}{\xi}. \tag{13.45}$$

Most physical problems have data $\eta(x)$ that is not simply sinusoidal for all $x \in (-\infty, \infty)$ but instead is concentrated in some restricted region, *e.g.*, a Gaussian pulse,

$$\eta(x) = e^{-\beta x^2} \tag{13.46}$$

The Fourier transform of this function is a Gaussian in ξ,

$$\hat{\eta}(\xi) = \frac{1}{\sqrt{2\beta}} e^{-\xi^2/4\beta}. \tag{13.47}$$

Note that for β small, $\eta(x)$ is a broad and smooth Gaussian with a Fourier transform that is sharply peaked near $\xi = 0$. In this case $\eta(x)$ consists primarily of low wave number smooth components. For β large $\eta(x)$ is sharply peaked while the transform is broad. More high wave number components are needed to represent the rapid spatial variation of $\eta(x)$ in this case. Note the nice duality here, since $\eta(x)$ can also be viewed as essentially (up to the sign in the exponent) the Fourier transform of $\hat{\eta}(\xi)$.

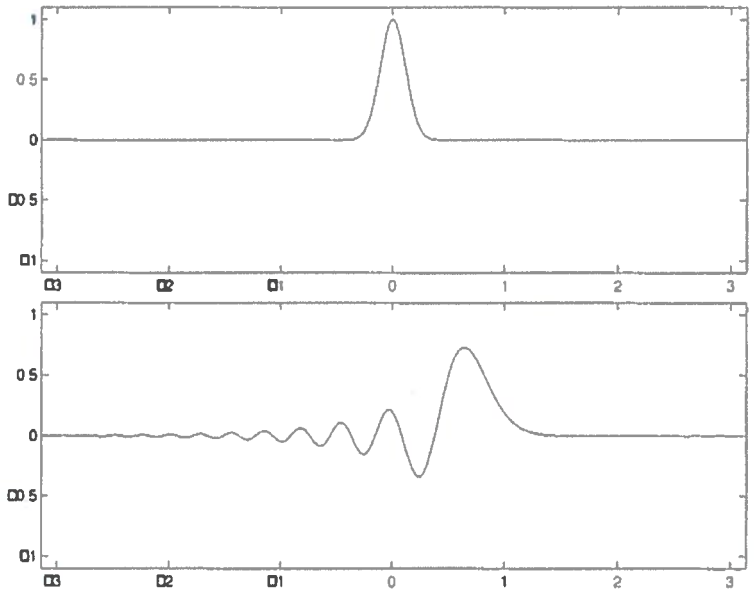

Figure 13.5: Gaussian initial data propagating with dispersion.

If we solve the dispersive equation with data of this form then the different modes propagate at different phase velocities and will no longer sum to a Gaussian, and the solution evolves as shown in Figure 13.5, forming "dispersive ripples". Note that for large times it is apparent that the wave length of the ripples is changing through this wave, and that the energy associated with the low wave numbers is apparently moving faster than the energy associated with larger wave numbers. The propagation velocity of this energy is not, however, the phase velocity $c_p(\xi)$. Instead it is given by the *group velocity*

$$c_g(\xi) = \frac{d\omega(\xi)}{d\xi}. \tag{13.48}$$

For the advection equation $u_t + au_x = 0$ the dispersion relation is $\omega(\xi) = a\xi$ and the group velocity agrees with the phase velocity (since all waves propagate at the same velocity a), but more generally the two do not agree. For the dispersive equation (13.40), $\omega(\xi) = a\xi - b\xi^3$ and we find that

$$c_g(\xi) = a - 3b\xi^2$$

whereas

$$c_p(\xi) = a - b\xi^2.$$

13.7.2 Wave packets

The notion and importance of group velocity is easiest to appreciate by considering a "wave packet" with data of the form

$$\eta(x) = e^{i\xi_0 x} e^{-\beta x^2} \tag{13.49}$$

or the real part of such a wave,

$$\eta(x) = \cos(\xi_0 x) e^{-\beta x^2}. \tag{13.50}$$

This is a single Fourier mode modulated by a Gaussian.

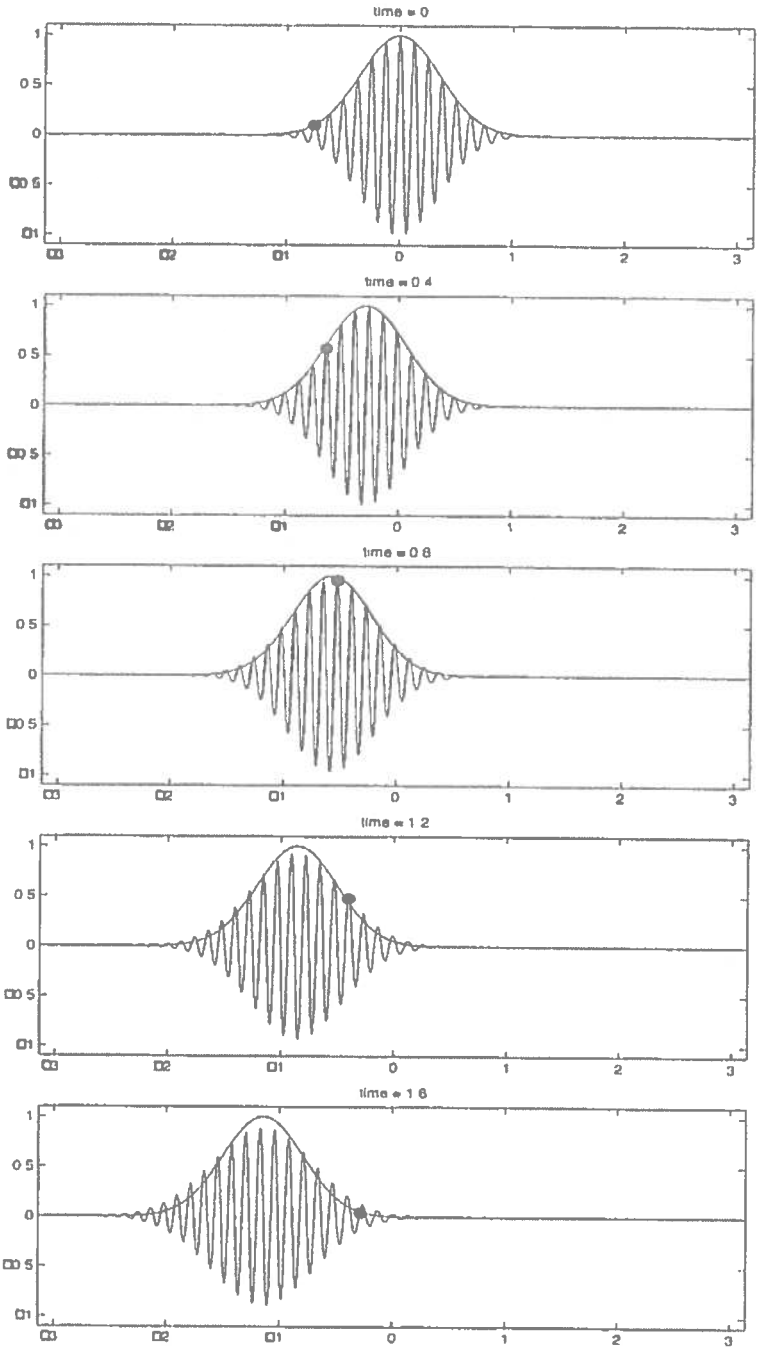

Figure 13.6: The oscillatory wave packet satisfies the dispersive equation $u_t + au_x + bu_{xxx} = 0$. Also shown is a black dot, translating at the phase velocity $c_p(\xi_0)$ and a Gaussian that is translating at the group velocity $c_g(\xi_0)$.

The Fourier transform of (13.49) is

$$\hat{\eta}(\xi) = \frac{1}{\sqrt{2\beta}} e^{-(\xi-\xi_0)^2/4\beta}, \tag{13.51}$$

a Gaussian centered about $\xi = \xi_0$. If the packet is fairly broad (β small) then the Fourier transform is concentrated near $\xi = \xi_0$ and hence the propagation properties of the wave packet are well approximated in terms of the phase velocity $c_p(\xi)$ and the group velocity $c_g(\xi)$. The wave crests propagate at the speed $c_p(\xi_0)$ while the evelope of the packet propagates at the group velocity $c_g(\xi_0)$.

To get some idea of why the packet propagates at the group velocity, consider the expression (13.43),

$$u(x,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{\eta}(\xi) e^{i(\xi x - \omega(\xi)t)}\, d\xi.$$

For a concentrated packet, we expect $u(x,t)$ to be very close to zero for most x, except near some point ct where c is the propagation velocity of the packet. To estimate c we will ask where this integral could give something nonzero. At each fixed x the integral is a Gaussian in ξ (the function $\hat{\eta}(\xi)$) multiplied by an oscillatory function of ξ (the exponential factor). Integrating this product will given essentially zero at a particular x provided that the oscillatory part is oscillating rapidly enough in ξ that it averages out to zero, even though it is modulated by the Gaussian $\hat{\eta}(\xi)$. This happens provided the function $\xi x - \omega(\xi)t$ appearing as the phase in the exponential is rapidly varying as a function of ξ at this x. Conversely, we expect the integral to be significantly different from zero only near points x where this phase function is stationary, *i.e.*, where

$$\frac{d}{d\xi}(\xi x - \omega(\xi)t) = 0.$$

This occurs at

$$x = \omega'(\xi)t,$$

showing that the wave packet propagates at the group velocity $c_g = \omega'(\xi)$. This approach to studying oscillatory integrals is called the "method of stationary phase" and is useful in other applications as well.

13.8 Hyperbolic systems

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

$$\begin{aligned} u_t + Au_x &= 0 \\ u(x,0) &= u_0(x) \end{aligned} \tag{13.52}$$

where $u : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times m}$ is a constant matrix. This is a system of conservation laws with the flux function $f(u) = Au$. This system is called **hyperbolic** if A is diagonalizable with real eigenvalues, so that we can decompose

$$A = R\Lambda R^{-1} \tag{13.53}$$

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

$$Ar_p = \lambda_p r_p \quad \text{for } p = 1, 2, \ldots, m. \tag{13.54}$$

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