5. Show that the backward heat equation,

$$\frac{\partial \psi}{\partial t} = -\frac{\partial^2 \psi}{\partial x^2},$$

and the initial condition $\psi(x, 0) = f(x)$ do not constitute a well-posed problem on the domain $-\infty < x < \infty, t > 0$.

From Durran: Numerical Methods for Wave Equations in Geophysical Fluid Dynamics Springer 1999

2

Basic Finite-Difference Methods

As discussed in the preceding chapter, there are two conceptually different ways to represent continuous functions on digital computers: as a finite set of grid-point values or as a finite set of series-expansion functions. The grid-point approach is used in conjunction with finite-difference methods, which were widely implemented on digital computers somewhat earlier than the series-expansion techniques. In addition, the theory for these methods is somewhat simpler than that for series-expansion methods. We will parallel this historical development by studying finite-difference methods in this chapter and deferring the treatment of series expansion methods to Chapter 4. Moreover, it is useful to understand finite-difference methods before investigating series-expansion techniques because even when series expansions are used to represent the spatial dependence of some atmospheric quantity, the time dependence is almost always discretized and treated with finite differences.

The derivative of a function $f(x)$ at the point x_0 could be defined in any of the following three ways:

$$\frac{df}{dx}(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}, \tag{2.1}$$

$$\frac{df}{dx}(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x}, \tag{2.2}$$

$$\frac{df}{dx}(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}. \tag{2.3}$$

If the derivative of $f(x)$ is continuous at x_0, all three expressions produce the same unique answer. Suppose, however, that f is an approximation to some dif-

ferentiable function that is defined on a discrete grid; then the preceding expressions must be evaluated using a finite value of Δx. The approximations to the true derivative obtained by evaluating the algebraic expressions on the right side of (2.1)–(2.3) using finite Δx are known as *finite differences*. The basic idea behind finite-difference methods is to convert the differential equation into a system of algebraic equations by replacing each derivative with a finite difference.

When Δx is finite, the finite-difference approximations (2.1)–(2.3) are not equivalent; they differ in their accuracy, and when they are substituted for derivatives in differential equations they generate different algebraic equations. The differences in the structures of these algebraic equations can have a great influence on the stability of the numerical solution. In this chapter, we will examine the stability and accuracy of basic finite-difference methods.

2.1 Accuracy and Consistency

The exact derivative can be calculated to within an arbitrarily small error using any one of (2.1)–(2.3) by insuring that Δx is sufficiently small. However, since computer capacities always place a practical limit on the numerical resolution, it is necessary to consider the case when Δx is small but finite and to inquire whether one of the finite-difference formulas (2.1)–(2.3) is likely to be more accurate than the others. If $f(x)$ is sufficiently smooth, this question can be answered by expanding the terms like $f(x_0 \pm \Delta x)$ in Taylor series about x_0 and substituting these expansions into the finite-difference formula. For example, when

$$f(x_0 + \Delta x) = f(x_0) + \Delta x \frac{df}{dx}(x_0) + \frac{(\Delta x)^2}{2}\frac{d^2 f}{dx^2}(x_0) + \frac{(\Delta x)^3}{6}\frac{d^3 f}{dx^3}(x_0) + \cdots$$

is substituted into (2.1), one finds that

$$\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} - \frac{df}{dx}(x_0) = \frac{\Delta x}{2}\frac{d^2 f}{dx^2}(x_0) + \frac{(\Delta x)^2}{6}\frac{d^3 f}{dx^3}(x_0) + \cdots. \quad (2.4)$$

The right side of (2.4) is known as the *truncation error*. The lowest power of Δx in the truncation error determines the *order of accuracy* of the finite difference. Inspection of its truncation error shows that the one-sided difference (2.1) is first-order accurate. In contrast, the truncation error associated with the centered difference (2.3) is

$$\frac{(\Delta x)^2}{6}\frac{d^3 f}{dx^3}(x_0) + \frac{(\Delta x)^4}{120}\frac{d^5 f}{dx^5}(x_0) + \cdots,$$

and the centered difference is therefore second-order accurate. If the higher-order derivatives of f are bounded in some interval about x_0 (i.e., f is "smooth") and the grid spacing is reduced, the error in the second-order difference (2.3) will approach zero more rapidly than the error in the first-order difference (2.1). The fact

that the truncation error of the centered difference is of higher order does not, however, guarantee that it will always generate a more accurate estimate of the derivative. If the function is sufficiently rough and the grid spacing sufficiently coarse, neither formula is likely to produce a good approximation, and the superiority of one over the other will be largely a matter of chance.

Higher-order finite-difference approximations can be constructed by including additional grid points in the finite-difference formula. Suppose, for example, that one wishes to obtain a fourth-order approximation to df/dx by determining the five coefficients $a, b, \ldots, e$ that satisfy

$$\frac{df}{dx}(x_0) = af(x_0 + 2\Delta x) + bf(x_0 + \Delta x) + cf(x_0) + d f(x_0 - \Delta x) + ef(x_0 - 2\Delta x) + O\left[(\Delta x)^4\right]. \quad (2.5)$$

Expanding $f(x_0 \pm \Delta x)$ and $f(x_0 \pm 2\Delta x)$ in Taylor series, substituting those expansions into (2.5), and equating the coefficients of like powers of Δx yields five equations for the unknown coefficients:

$$\begin{aligned} a + b + c + d + e &= 0, \\ 2a + b - d - 2e &= 1/\Delta x, \\ 4a + b + d + 4e &= 0, \\ 8a + b - d - 8e &= 0, \\ 16a + b + d + 16e &= 0. \end{aligned}$$

The unique solution to this system requires $c = 0$ and yields an approximation to the derivative of the form

$$\frac{df}{dx}(x_0) = \frac{4}{3}\left(\frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}\right) - \frac{1}{3}\left(\frac{f(x_0 + 2\Delta x) - f(x_0 - 2\Delta x)}{4\Delta x}\right) + O\left[(\Delta x)^4\right]. \quad (2.6)$$

Similar procedures can be used to generate even higher-order formulae, off-centered formulae, and formulae for irregular grid intervals.

As an alternative to the brute force manipulation of Taylor series, the derivation of higher-order finite-difference formulae can be facilitated by the systematic use of operator notation and simple lower-order formulae. A simpler derivation of (2.6) may be obtained by defining a finite-difference operator δ_{nx} such that

$$\delta_{nx} f(x) = \frac{f(x + n\Delta x/2) - f(x - n\Delta x/2)}{n\Delta x}. \quad (2.7)$$

Using this notation, the second-order centered difference satisfies

$$\delta_{2x} f = \frac{df}{dx} + \frac{(\Delta x)^2}{6}\frac{d^3 f}{dx^3} + O\left[(\Delta x)^4\right]. \quad (2.8)$$

From the definition of δ_{nx},

$$\delta_x^2 f = \delta_x(\delta_x f) = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2},$$

and a conventional Taylor series analysis of the truncation error shows that

$$\delta_x^2 f = \frac{d^2 f}{dx^2} + O\left[(\Delta x)^2\right].$$

It follows that $\delta_{2x}\delta_x^2 f$ is a second-order approximation to the third derivative of f, since

$$\delta_{2x}\delta_x^2 f = \delta_{2x}\left(\frac{d^2 f}{dx^2} + O\left[(\Delta x)^2\right]\right) = \frac{d^3 f}{dx^3} + O\left[(\Delta x)^2\right].$$

Substitution of the preceding into (2.8) yields

$$\left(1 - \frac{(\Delta x)^2}{6}\delta_x^2\right)\delta_{2x} f = \frac{df}{dx} + O\left[(\Delta x)^4\right]. \tag{2.9}$$

Expansion of this formula via the operator definition (2.7) yields the centered fourth-order difference (2.6). Although it allows finite-difference equations to be expressed in a very compact form, operator notation will not be used for all finite-difference equations throughout the remainder of this book, but will be reserved for complicated formulae that become unwieldy when written in expanded form. Most of the finite-difference schemes considered in the remainder of this chapter are sufficiently simple that they will be expressed without using operator notation.

We now turn from the consideration of individual finite differences to examine the accuracy of an entire finite-difference scheme. Suppose that an approximation to the advection equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0 \tag{2.10}$$

is to be obtained at the grid points $(n\Delta t, j\Delta x)$, where n and j are integers. It is convenient to represent the numerical approximation to $\psi((n\Delta t, j\Delta x)$ in the shorthand notation ϕ_j^n. One possible finite-difference formula for the numerical approximation of (2.10) is

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + c\frac{\phi_j^n - \phi_{j-1}^n}{\Delta x} = 0; \tag{2.11}$$

when $c > 0$, this is known as the "upstream" or "donor-cell" scheme. The accuracy of a finite-difference scheme is characterized by the residual error with which

the solution of the continuous equation fails to satisfy the finite-difference formulation. Under the assumption that ψ is sufficiently smooth, its value at adjacent grid points can be obtained from a Taylor series expansion about $(n\Delta t, j\Delta x)$ and substituted into (2.11) to yield

$$\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} + c\frac{\psi_j^n - \psi_{j-1}^n}{\Delta x} = \frac{\Delta t}{2}\frac{\partial^2 \psi}{\partial t^2} - c\frac{\Delta x}{2}\frac{\partial^2 \psi}{\partial x^2} + \cdots. \tag{2.12}$$

The right side of (2.12) is the truncation error of the finite-difference scheme. The order of accuracy of the scheme is determined by the lowest powers of Δt and Δx appearing in the truncation error. According to (2.12), the upstream scheme is first-order accurate in space and time. If the truncation error of the finite-difference scheme approaches zero as $\Delta t \to 0$ and $\Delta x \to 0$, the scheme is *consistent*. Inspection of (2.12) clearly shows that the upstream scheme is consistent. Although it is not difficult to design consistent difference schemes, this property should not be taken for granted. One sometimes encounters methods that require additional relations between Δt and Δx, such as $\Delta t/\Delta x \to 0$, in order to achieve consistency.

2.2 Stability and Convergence

The preceding measures of accuracy do not describe the difference between the numerical solution ϕ_j^n and the true solution $\psi(n\Delta t, j\Delta x)$, which, of course, is the most direct measure of the quality of the numerical solution. Before discussing this error one needs a way to measure its size, i.e., one needs to define a *norm*. The general mathematical notation for a norm is a pair of vertical bars $\| \|$. In the following we will be concerned with the maximum norm and the Euclidean, or ℓ_2, norm. The maximum norm, defined as

$$\|\phi\|_\infty = \max_{1 \le j \le N} |\phi_j|, \tag{2.13}$$

is simply the extremum of the grid-point values. The Euclidean, or ℓ_2, norm is defined as

$$\|\phi\|_2 \equiv \left(\sum_{j=1}^{N} |\phi_j|^2 \Delta x\right)^{1/2}. \tag{2.14}$$

If the constant scaling factor Δx is ignored, (2.14) is just the length of an N-dimensional vector (hence the name Euclidean norm). The inclusion of the Δx factor makes (2.14) a numerical approximation to the square root of the spatial integral of the function times its complex conjugate, $\phi\phi^*$, whence the name ℓ_2

norm.[1] It might appear that the maximum norm is the most natural one to compute when working with grid-point values; however, the ℓ_2 norm is also useful, because it is more closely related to conserved physical quantities, such as the total energy.

A finite-difference scheme is said to be *convergent* of order (p, q) if in the limit $\Delta x, \Delta t \to 0$,

$$\|\psi(n\Delta t, j\Delta x) - \phi_j^n\| = O\left[(\Delta t)^p\right] + O\left[(\Delta x)^q\right].$$

The relationship between convergence and consistency is described by the *Lax equivalence theorem*, which states that *if a finite-difference scheme is linear, stable, and accurate of order* (p, q), *then it is convergent of order* (p, q) (Lax and Richtmyer 1956). Lax's theorem shows that mere consistency is not enough to assure the convergence of a numerical method. The method must also be *stable*. There are a great number of consistent finite-difference methods that are utterly useless because they are unstable. It is common practice to describe a finite-difference scheme as "unstable" if it generates a numerical solution that grows much more rapidly than the true solution. When "stable" and "unstable" are used in this sense, some reference must be made to the properties of the true solution, and since the true solution can exhibit a wide range of different behaviors, one can arrive at several different criteria for "stability."

The fundamental definition of stability makes no reference to the properties of the true solution and only identifies the least-restrictive additional constraint that must be satisfied in order to ensure the convergence of solutions generated by a consistent finite-difference scheme. A consistent linear finite-difference scheme will be convergent, and the Lax equivalence theorem will be satisfied, provided that for any time T there exists a constant C_T such that

$$\|\phi^n\| \le C_T \|\phi^0\| \quad \text{for all} \quad n\Delta t \le T \tag{2.15}$$

and all sufficiently small values of Δt and Δx. In the preceding, C_T may depend on the time T, but not on Δt, Δx, or the number of time steps n. This definition leaves the numerical solution tremendous latitude for growth with time, but it rules out solutions that grow as a function of the number of time steps. If a difference scheme is unstable in the sense that it fails to satisfy (2.15), repeated reductions in Δt and Δx may generate an unbounded amplification in the numerical approximation to the true solution at time T. In such a situation, the numerical solution could hardly be expected to converge to the true solution in the limit Δx, $\Delta t \to 0$.

[1] To better appreciate the notation used to represent the maximum and ℓ_2 norms, note that $\|\phi\|_\infty$ is essentially the integral

$$\left(\int |\phi|^\infty \, dx\right)^{1/\infty}$$

and $\|\phi\|_2$ is

$$\left(\int |\phi|^2 \, dx\right)^{1/2}.$$

The practical shortcoming of the preceding definition of stability is that it says nothing about the quality of the solution that might be obtained when using finite values of Δt and Δx; it only ensures that an accurate solution will be obtained in the limit $\Delta x, \Delta t \to 0$. Schemes that are stable according to the criteria (2.15) may, nevertheless, generate solutions that "blow up" in practical applications (see Section 3.4.3 for an example). In order to ensure that the numerical solution is qualitatively similar to the true solution when Δx and Δt are finite, it is often useful to impose stability constraints that are more stringent than (2.15). In many wave propagation problems, the norm of the true solution is constant with time, and in such instances it is appropriate to require that the numerical scheme satisfy

$$\|\phi^n\| \le \|\phi^0\| \quad \text{for all} \quad n. \tag{2.16}$$

In contrast to (2.15), this condition is not necessary for convergence, and it cannot be sensibly imposed without specific knowledge about the boundedness of the solutions to the associated partial differential equation. Nevertheless, (2.16) is a perfectly reasonable constraint to impose in applications where the true solution is not growing with time, and unlike (2.15), it guarantees that the solution will not blow up.

It is relatively easy to formulate consistent difference schemes and to determine their truncation error and order of accuracy. The analysis of stability can, however, be far more difficult, particularly when the finite-difference scheme and the associated partial differential equation are nonlinear. Thus, our initial discussion of stability will be focused on the simplest case—linear finite-difference schemes for the approximation of linear partial differential equations with constant coefficients. Nonlinear equations and linear equations with variable coefficients will be considered in Chapter 3.

2.2.1 *The Energy Method*

In practice, the energy method is used much less frequently than the Von Neumann method, which will be discussed in the next section. Nevertheless, the energy method is important, because unlike the Von Neumann method, it can be applied to nonlinear equations and to problems without periodic boundaries. The basic idea behind the energy method is to find a positive definite quantity like $\sum_j (\phi_j^n)^2$ and show that this quantity is bounded for all n. If $\sum_j (\phi_j^n)^2$ is bounded, the solution is stable with respect to the ℓ_2-norm.

As an example, let us investigate the stability of the upstream finite-difference scheme (2.11). Defining $\mu = c\Delta t/\Delta x$, the scheme may be written as

$$\phi_j^{n+1} = (1 - \mu)\phi_j^n + \mu\phi_{j-1}^n. \tag{2.17}$$

Squaring both sides and summing over all j gives

$$\sum_j (\phi_j^{n+1})^2 = \sum_j \left[(1-\mu)^2(\phi_j^n)^2 + 2\mu(1-\mu)\phi_j^n\phi_{j-1}^n + \mu^2(\phi_{j-1}^n)^2\right]. \tag{2.18}$$

Assuming cyclic boundary conditions,[2]

$$\sum_j (\phi_{j-1}^n)^2 = \sum_j (\phi_j^n)^2, \tag{2.19}$$

and using the Schwarz inequality (which states that for two vectors $\mathbf{u}$ and $\mathbf{v}$, $|\mathbf{u}\cdot\mathbf{v}| \le \|\mathbf{u}\|\|\mathbf{v}\|$),

$$\sum_j \phi_j^n\phi_{j-1}^n \le \left[\sum_j (\phi_j^n)^2\right]^{1/2}\left[\sum_j (\phi_{j-1}^n)^2\right]^{1/2} = \sum_j (\phi_j^n)^2. \tag{2.20}$$

If $\mu(1-\mu) \ge 0$, all three coefficients in (2.18) are positive, and (2.19) and (2.20) may be used to construct the inequality

$$\sum_j (\phi_j^{n+1})^2 \le \left[(1-\mu)^2 + 2\mu(1-\mu) + \mu^2\right]\sum_j (\phi_j^n)^2 = \sum_j (\phi_j^n)^2, \tag{2.21}$$

which requires $\|\phi^{n+1}\|_2 \le \|\phi^0\|_2$ and implies that the scheme is stable. The condition used to obtain (2.21),

$$\mu(1-\mu) \ge 0, \tag{2.22}$$

is therefore a sufficient condition for stability. Under the assumption that $\mu > 0$, division of (2.22) by μ leads to the relation $\mu \le 1$, and the total constraint on μ is therefore $0 < \mu \le 1$. A similar treatment of the case $\mu \le 0$ leads to the contradictory requirement that $\mu \ge 1$ and provides no additional solutions. Thus, recalling the definition of μ and noting that $\mu = 0$ satisfies (2.22), the stability condition may be written

$$0 \le \frac{c\Delta t}{\Delta x} \le 1.$$

As is typical with most conditionally stable difference schemes, there is a maximum limit on the time step beyond which the scheme is unstable, and the stability limit becomes more severe as the spatial resolution is increased.

[2] If more general boundary conditions are imposed at the edges of the spatial domain, a rigorous stability analysis becomes much more difficult. The determination of stability in the presence of nonperiodic boundaries is discussed in Section 8.1.6.

2.2.2 *Von Neumann's Method*

One drawback of the energy method is that each new problem requires fresh insight in order to define an appropriate energy and to show that the finite-difference scheme preserves a bound on that energy. Von Neumann's method has the advantage that it can be applied by following a prescribed procedure; however, it is applicable only to linear finite-difference equations with constant coefficients.[3] The basic idea of the Von Neumann method is to represent the discretized solution at some particular time step by a finite Fourier series of the form

$$\phi_j^n = \sum_{k=-N}^{N} a_k^n e^{ikj\Delta x},$$

and to examine the stability of the individual Fourier components. The total solution will be stable if and only if every Fourier component is stable. The use of finite Fourier series is strictly appropriate only if the spatial domain is periodic. When problems are posed with more general boundary conditions, a rigorous stability analysis is more difficult, but the Von Neumann method still provides a useful way of weeding out obviously unsuitable schemes.

A key property of Fourier series is that individual Fourier modes are eigenfunctions of the derivative operator, i.e.,

$$\frac{d}{dx}e^{ikx} = ike^{ikx}.$$

Finite Fourier series have an analogous property in that individual modes $e^{ikj\Delta x}$ are eigenfunctions of linear finite-difference operators. Thus, if the initial conditions for some linear, constant-coefficient finite-difference scheme are $\phi_j^n = e^{ikj\Delta x}$, after one iteration the solution will have the form

$$\phi_j^{n+1} = A_k e^{ikj\Delta x},$$

where A_k is a complex constant, known as the *amplification factor*, that is determined by the form of the finite-difference formulae. Since the analysis is restricted to linear constant-coefficient schemes, the amplification factor will not vary from time step to time step, and if a_k^n denotes the amplitude of the kth finite Fourier component at the nth time step, then

$$a_k^n = A_k a_k^{n-1} = (A_k)^n a_k^0.$$

[3] In order to apply Von Neumann's method to more general problems, the governing finite-difference equations must be linearized and any variable coefficients must be frozen at some constant value. The Von Neumann stability of the family of linearized, frozen-coefficient systems may then be examined. See Section 3.5.

It follows that the stability of each Fourier component is determined by the modulus of its amplification factor.

The *Von Neumann stability condition*, which is necessary and sufficient for the stability of a linear constant-coefficient finite-difference equation,[4] requires the amplification factor of every Fourier component resolvable on the grid to be bounded such that

$$|A_k| \leq 1 + \gamma \Delta t, \tag{2.23}$$

where γ is a constant independent of k, Δt, and Δx. This condition ensures that a consistent finite-difference scheme satisfies the minimum stability criteria for convergence in the limit Δx, $\Delta t \to 0$, (2.15). In applications where the true solution is bounded by the norm of the initial data, it is usually advantageous to enforce the more-stringent requirement that

$$|A_k| \leq 1, \tag{2.24}$$

which will guarantee satisfaction of the stability condition (2.16). When the Von Neumann condition is satisfied, every finite Fourier component is stable, and the full solution, being a linear combination of the individual Fourier components, must also be stable.

As an illustration of the Von Neumann method, consider once again the finite-difference equation (2.17). The solutions to the associated partial differential equation (2.10) do not grow with time, so we will require $|A_k| \leq 1$. Substitution of an arbitrary Fourier component, of the form $e^{ikj\Delta x}$, into (2.17) yields

$$A_k e^{ikj\Delta x} = (1 - \mu)e^{ikj\Delta x} + \mu e^{ik(j-1)\Delta x}.$$

Dividing out the common factor $e^{ikj\Delta x}$ gives

$$A_k = 1 - \mu + \mu e^{-ik\Delta x}. \tag{2.25}$$

The magnitude of A_k is obtained by multiplying by its complex conjugate and taking the square root. Thus,

$$\begin{aligned} |A_k|^2 &= (1 - \mu + \mu e^{-ik\Delta x})(1 - \mu + \mu e^{ik\Delta x}) \\ &= 1 - 2\mu(1 - \mu)(1 - \cos k\Delta x). \end{aligned} \tag{2.26}$$

The Von Neumann condition (2.24) will therefore be satisfied if

$$1 - 2\mu(1 - \mu)(1 - \cos k\Delta x) \leq 1.$$

[4]The sufficiency of the Von Neumann condition holds only for single equations in one unknown. The stability of systems of finite-difference equations in several unknown variables is discussed in Section 3.1.

Since $1 - \cos k\Delta x > 0$ for all wave numbers except the trivial case $k = 0$, the preceding inequality reduces to

$$\mu(1 - \mu) \geq 0,$$

which is identical to the condition (2.22) obtained using the energy method. As discussed previously in connection with (2.22), this stability condition may be expressed as

$$0 \leq \frac{c\Delta t}{\Delta x} \leq 1.$$

Inspection of (2.26) shows that the $2\Delta x$ wave grows most rapidly in any integration performed with an unstable value of μ. Thus, as it "blows up," an unstable solution becomes dominated by large-amplitude $2\Delta x$ waves. Most other finite-difference approximations to the advection equation exhibit the same tendency: When solutions become unstable, they usually become contaminated by large-amplitude short waves. The upstream scheme is, nevertheless, unusual in that all waves become unstable for the same critical value of μ. In many other schemes, such as the leapfrog-time centered-space formulation (2.91), there exist values of μ for which only a few of the shorter wavelengths are unstable. One might suppose that such nominally unstable values of μ could still be used in numerical integrations if the initial data were filtered to remove all amplitude from the unstable finite Fourier components; however, even if the initial data have zero amplitude in the unstable modes, round-off error in the numerical computations will excite the unstable modes and trigger the instability.

2.2.3 *The Courant–Fredrichs–Lewy Condition*

The basic idea of the Courant–Fredrichs–Lewy (CFL) condition is that the solution of a finite-difference equation must not be independent of the data that determines the solution to the associated partial differential equation. The CFL condition can be made more precise by defining the *domain of influence* of a point (x_0, t_0) as that region of the x-t plane where the solution to some particular partial differential equation is influenced by the solution at (x_0, t_0). A related concept, the *domain of dependence* of a point (x_0, t_0), is defined as the set of points containing (x_0, t_0) within their domains of influence. The domain of dependence of (x_0, t_0) will therefore consist of all points (x, t) at which the solution has some influence on the solution at (x_0, t_0). A similar concept applicable to the discretized problem is the *numerical domain of dependence* of a grid point $(n_0\Delta t, j_0\Delta x)$, which consists of the set of all nodes on the space–time grid $(n\Delta t, j\Delta x)$ at which the value of the numerical solution influences the numerical solution at $(n_0\Delta t, j_0\Delta x)$. *The CFL condition requires that the numerical domain of dependence of a finite-difference scheme include the domain of dependence of the associated partial differential equation.* Satisfaction of the CFL condition is a necessary condition for stability, but is not sufficient to guarantee stability.

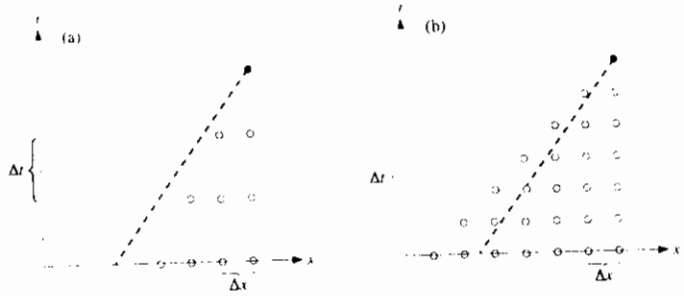

FIGURE 2.1. The influence of the time step on the relationship between the numerical domain of dependence of the upstream scheme (open circles) and the true domain of dependence of the advection equation (heavy dashed line): (a) unstable Δt, (b) stable Δt.

The nature of the CFL condition can be illustrated by considering the advection equation (2.10), which has general solutions of the form $\psi(x - ct)$. Thus the true domain of influence of a point (x_0, t_0) is the straight line

$$t = t_0 + \tfrac{1}{c}(x - x_0), \quad t \geq t_0.$$

The same "characteristic line" also defines the true domain of dependence of (x_0, t_0), except that one looks backward in time by requiring $t \leq t_0$. The true domain of dependence is plotted as a dashed line in Fig. 2.1, together with those grid points composing the numerical domain of dependence of the upstream finite-difference scheme (2.17). The two panels in this figure show the influence of two different time steps on the shape of the numerical domain of dependence. In Fig. 2.1a, the initial value of ψ along the x-axis, which determines the solution to the partial differential equation at $(n\Delta t, j\Delta x)$, plays no role in the determination of the finite-difference solution ϕ_j^n. The numerical solution can be in error by any arbitrary amount, and will not converge to the true solution as $\Delta t, \Delta x \to 0$ unless there is a change in the ratio $\Delta t/\Delta x$. Hence, the finite-difference method, which is consistent with the original partial differential equation, must be unstable (or else the Lax equivalence theorem would be violated).

The situation shown in Fig. 2.1b is obtained by halving the time step. Then the numerical domain of dependence contains the domain of dependence of the true solution, and it is possible for the numerical solution to be stable. In this example the CFL condition requires the slope of the characteristic curve to be greater than the slope of the left edge—and less than the slope of the right edge—of the domain of dependence. As evident from Fig. 2.1, the slope condition at the right edge of the domain is $1/c \leq \infty$, which is always satisfied. The slope condition at the left edge of the domain may be expressed as $\Delta t/\Delta x \leq 1/c$. If $c > 0$, this requires

$$c\Delta t/\Delta x \leq 1, \tag{2.27}$$

and the nonnegativity of Δt and Δx implies

$$c\Delta t/\Delta x \geq 0. \tag{2.28}$$

Simultaneous satisfaction of (2.27) and (2.28) is obtained when

$$0 \leq c\frac{\Delta t}{\Delta x} \leq 1.$$

In the case $c < 0$, similar reasoning leads to contradictory requirements, and the solution is unstable.

The preceding stability condition is identical to those already obtained using the energy and Von Neumann methods, but such agreement is actually rather unusual. The CFL condition is only a necessary condition for stability, and in many cases the sufficient conditions for stability are more restrictive than those required by the CFL condition. As an example, consider the following approximation to the advection equation,

$$\delta_{2t}\phi + c\left(\frac{4}{3}\delta_{2x}\phi - \frac{1}{3}\delta_{4x}\phi\right) = 0,$$

which uses the fourth-order accurate approximation to the spatial derivative (2.6). Since the spatial difference utilizes a five-grid-point-wide stencil, the CFL condition is satisfied when

$$\left|c\frac{\Delta t}{\Delta x}\right| \leq 2.$$

Yet the actual sufficient condition for stability is the much more restrictive condition

$$\left|c\frac{\Delta t}{\Delta x}\right| \leq 0.728,$$

which may be derived via a Von Neumann stability analysis.

2.3 Time-Differencing

A given partial differential equation can be approximated by an almost unlimited variety of different finite-difference formulae. In order to systematically examine the properties of various finite-difference schemes, let us begin by discussing possible approximations to the time derivative without explicitly considering the spatial derivatives. The primary reason for discussing time and space differencing separately is that they present the numerical analyst with rather different sets of practical problems. After the nth step of the integration, the numerical solution ϕ will be known at every point on the spatial mesh, and several grid-point values may be easily included in any finite-difference approximation to the spatial derivatives. It is easy, for example, to construct high-order centered approximations to

spatial derivatives. In contrast, storage limitations dictate that ϕ be retained at as few time levels as possible, and the only time levels available are those from previous iterations. Thus, higher-order finite-difference approximations to the time derivative are inherently one-sided.

The following discussion of the effects of time-differencing on the numerical solution is transferable, after minor modification, to situations where the spatial derivatives are approximated by centered differences. In addition, this discussion provides an exact analysis of the influence of time-differencing on schemes, such as the spectral method, where finite differences are not used to evaluate the spatial derivatives. Nevertheless, one must be careful not to assume that space and time differences are completely independent. Indeed, techniques such as the Lax–Wendroff method cannot be properly analyzed without understanding the interaction between space truncation error and time truncation error. The combined effects of space- and time-differencing will be discussed, together with schemes like the Lax–Wendroff method, in Section 2.5.

Time-differencing formulae used in the numerical solution of partial differential equations are related, naturally enough, to the numerical methods used to integrate ordinary differential equations. In comparison with typical ordinary differential equation solvers, the methods used to integrate partial differential equations are of very low order. Low-order schemes are used for two basic reasons. First, the approximation of the time derivative is not the only source of finite-differencing error in the solution of partial differential equations; other errors arise through the approximation of the spatial derivatives. In many circumstances the largest errors in the solution are introduced through the numerical evaluation of the spatial derivatives, so it is pointless to devote additional computational resources to higher-order time-differencing. The second reason for using low-order methods is that practical limitations on computational resources often leave no other choice.

2.3.1 *The Oscillation Equation: Phase-Speed and Amplitude Error*

Hundreds of papers have been written investigating various techniques for the finite-difference solution of the advection equation (2.10), many of which are listed in the extensive review by Rood (1987). The vastness of this body of literature is a testament to the subtle tradeoffs involved in the selection of the "best" numerical method for even very simple equations. It might be supposed that the relative accuracy of different methods could be easily determined by comparing their respective truncation errors. The analysis of truncation error is, however, most effective at predicting the behavior of well-resolved waves, and the most serious errors are often found in the poorly resolved waves. The accuracy of both well resolved and poorly resolved waves can be better examined by extending the standard Von Neumann analysis to study the phase-speed and amplitude error in each Fourier component of the numerical solution.

If the spatial structure of the solution to the advection equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0$$

is represented by a Fourier series (or a Fourier Integral), the amplitude of an individual Fourier mode $b_k(t)$ must satisfy

$$\frac{db_k}{dt} = -ikcb_k. \tag{2.29}$$

The preceding is an ordinary differential equation that can be used to examine the numerical error introduced by time-differencing in wave-propagation problems. Equation (2.29) is a specific example of the *oscillation equation*

$$\frac{d\psi}{dt} = i\kappa\psi, \tag{2.30}$$

where κ is a real constant representing a frequency. Integrating the oscillation equation over a time Δt yields

$$\psi(t_0 + \Delta t) = e^{i\kappa\Delta t}\psi(t_0) \equiv A_e\psi(t_0). \tag{2.31}$$

Here the last relation defines an "exact amplification factor" A_e, which is a complex number of modulus one. According to (2.31), ψ moves $\kappa\Delta t$ radians around a circle of radius $|\psi(t_0)|$ in the complex plane over the time interval Δt.

Suppose that a finite-difference scheme is used to compute an approximate solution ϕ to the oscillation equation. A numerical amplification factor may be defined such that $\phi^{n+1} = A\phi^n$, where ϕ^n is the numerical approximation to $\psi(n\Delta t)$. The standard Von Neumann stability analysis seeks a yes–no answer to the question, Is $|A| \le |A_e| = 1$? Additional information about the amplitude and phase-speed error in the numerical solution can, however, be obtained by writing the numerical amplification factor in the form $|A|e^{i\theta}$, where

$$|A| = (\Re\{A\}^2 + \Im\{A\}^2)^{1/2} \quad \text{and} \quad \theta = \arctan\left(\frac{\Im\{A\}}{\Re\{A\}}\right).$$

Phase-speed errors arise from the difference between the argument of the finite-difference amplification factor θ and the correct value of $\kappa\Delta t$. A useful way to characterize phase-speed error is through the relative phase change $R = \theta/(\kappa\Delta t)$, which is the ratio of the phase advance produced by one time step of the finite-difference scheme divided by the change in phase experienced by the true solution over the same time interval. If $R > 1$, the finite-difference scheme is *accelerating*; if $R < 1$, the scheme is *decelerating*.

Amplitude errors arise from the difference between the modulus of the finite-difference amplification factor $|A|$ and the correct value of unity. When $|A| = 1$, the scheme is *neutral*. If $|A| < 1$, the scheme is *damping*; and if $|A| > 1$, it is *amplifying*. Amplifying schemes are certainly unstable in the sense that they generate approximate solutions to the oscillation equation that blow up, whereas the correct solution remains bounded by $|\phi^0|$. A more subtle characterization of

the stability of amplifying schemes, which will be considered in Sections 2.3.2 and 2.5, concerns the extent to which they can produce approximate solutions to ordinary and partial differential equations that converge in the limit $\Delta t \to 0$ and $\Delta x \to 0$.

2.3.2 *Single-Stage Two-Level Schemes*

The simplest techniques for the solution of the differential equation

$$\frac{d\psi}{dt} = F(\psi) \tag{2.32}$$

are members of the general family of single-stage two-time-level schemes, which may be written in the form

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \alpha F(\phi^n) + \beta F(\phi^{n+1}). \tag{2.33}$$

Here ϕ^n is the numerical approximation to $\psi(n\Delta t)$, and $\alpha + \beta = 1$ for consistency with the original equation. The preceding general form includes several well-known elementary methods. When $\alpha = 1$, $\beta = 0$, the scheme is known as forward differencing or Euler's method. Backward differencing corresponds to the case $\alpha = 0$, $\beta = 1$, and the trapezoidal method is obtained when $\alpha = \beta = \frac{1}{2}$.

The finite-difference scheme (2.33) may be alternatively expressed in the form

$$\phi^{n+1} = \phi^n + \Delta t \left(\alpha F(\phi^n) + \beta F(\phi^{n+1})\right), \tag{2.34}$$

which is analogous to the integrated form of the original differential equation

$$\psi\left[(n+1)\Delta t\right] = \psi(n\Delta t) + \int_{n\Delta t}^{(n+1)\Delta t} F\left(\psi(t)\right)\, dt.$$

Although (2.33) and (2.34) are clearly equivalent, confusion sometimes arises in determining the accuracy of numerical methods expressed in the integrated form (2.34). If the numerical method is convergent of order r and if ψ is the solution to the continuous problem, expansion of ψ in a Taylor series and substitution of that series into the discrete derivative form (2.33) will yield a residual error of $O[(\Delta t)^r]$, whereas substitution into the discrete integral form (2.34) will yield an error of $O[(\Delta t)^{r+1}]$. Of course, if the solutions to (2.33) and (2.34) converge to the true solution as $\Delta t \to 0$, they must converge at the same rate, because the two schemes are algebraically equivalent. This rate of convergence is equal to the order of the *global truncation error*, which is the same order as the truncation error associated with (2.33). The $O[(\Delta t)^{r+1}]$ truncation error associated with the integral form (2.34) is the *local truncation error*, or *one-step error*, and represents the error introduced in each step of the integration, i.e., the difference between ψ^{n+1} as computed by one step of the finite-difference method and the exact solution to the differential equation at $t = (n+1)\Delta t$ subject to the initial condition $\psi(n\Delta t) = \psi^n$. The rate at which the solution of a *stable* finite-difference scheme converges to the true solution as $\Delta t \to 0$ is one power of Δt lower than the order of the local truncation error because, roughly speaking, the global error generated by a stable scheme during an integration over a time interval T is the cumulative sum of $T/\Delta t$ local errors. When the term "truncation error" is used without qualification in this book, it will refer to the global truncation error. The truncation error of all members of the family of schemes (2.33) is $O(\Delta t)$, except for the trapezoidal method, which is $O\left[(\Delta t)^2\right]$.

Application of (2.34) to the oscillation equation (2.30) yields

$$(1 - i\beta\kappa\Delta t)\phi^{n+1} = (1 + i\alpha\kappa\Delta t)\phi^n.$$

The amplification factor for this scheme is

$$A \equiv \frac{\phi^{n+1}}{\phi^n} = \frac{1 + i\alpha\kappa\Delta t}{1 - i\beta\kappa\Delta t}.$$

Multiplying A by its complex conjugate gives

$$|A|^2 = \frac{1 + \alpha^2\kappa^2\Delta t^2}{1 + \beta^2\kappa^2\Delta t^2}$$

$$= 1 + (\alpha^2 - \beta^2)\frac{\kappa^2\Delta t^2}{1 + \beta^2\kappa^2\Delta t^2}. \tag{2.35}$$

Inspection of (2.35) shows that the scheme is neutral when $\alpha = \beta$, damping when $\alpha < \beta$, and amplifying when $\alpha > \beta$. The amplification produced when $\alpha > \beta$ is clearly unstable in the sense that approximate solutions computed with finite Δt can generate floating-point overflows on digital computers, whereas the magnitude of the correct solution is bounded by $|\phi^0|$. Note, however, that the amplification factor for forward differencing satisfies the general Von Neumann stability condition (2.23), since for $\Delta t \leq 1$,

$$|A|_{\text{forward}} = 1 + (\kappa\Delta t)^2 \leq |A|^2_{\text{forward}} \leq 1 + \kappa^2\Delta t.$$

to yield

$$|A|_{\text{forward}} \leq |A|^2_{\text{forward}} = 1 + (\kappa\Delta t)^2 \leq 1 + \kappa^2\Delta t$$

As a consequence, the amplifying solutions obtained using forward differencing do converge to the correct solution of the oscillation equation as $\Delta t \to 0$. Forward differencing also generates convergent approximations to most other ordinary differential equations, but convergence is not guaranteed, and (2.23) is not satisfied, when forward time differencing is used in conjunction with centered-difference approximations to the spatial derivative in the advection equation. This point is discussed further in Section 2.5.

The amplitude and phase errors in the approximate solution are functions of the *numerical resolution*. The solution to the governing differential equation (2.30) oscillates with a period $T = 2\pi/\kappa$. An appropriate measure of numerical resolution is the number of time steps per oscillation period, $T/\Delta t$. The numerical resolution is improved by decreasing the step size. In the limit of very good numerical resolution, $T/\Delta t \to \infty$ and $\kappa\Delta t \to 0$. Assuming good numerical resolution,

Taylor series expansions, such as

$$(1+x)^{1/2} = 1 + \frac{x}{2} - \frac{x^2}{8} + \cdots \quad \text{for} \quad |x| < 1,$$

may be used to reduce (2.35) to

$$|A| \approx 1 + \tfrac{1}{2}(\alpha^2 - \beta^2)(\kappa\Delta t)^2.$$

It follows that

$$|A|_{\text{forward}} \approx 1 + \tfrac{1}{2}(\kappa\Delta t)^2 \qquad \text{and} \qquad |A|_{\text{backward}} \approx 1 - \tfrac{1}{2}(\kappa\Delta t)^2, \tag{2.36}$$

indicating that the spurious amplitude changes introduced by both forward differencing and backward differencing are $O[(\kappa\Delta t)^2]$.

The relative phase change in the family of single-stage two-level schemes is

$$R = \frac{1}{\kappa\Delta t} \arctan\left(\frac{(\alpha+\beta)\kappa\Delta t}{1 - \alpha\beta(\kappa\Delta t)^2}\right).$$

Thus,

$$R_{\text{forward}} = R_{\text{backward}} = \frac{\arctan \kappa\Delta t}{\kappa\Delta t}, \tag{2.37}$$

which ranges between 0 and 1, implying that both forward differencing and backward differencing are decelerating. Assuming, once again, that the numerical solution is well-resolved, the preceding expression for the phase-speed error may be approximated using the Taylor series expansions, such as

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \cdots \quad \text{for} \quad |x| < 1,$$

to obtain

$$R_{\text{forward}} = R_{\text{backward}} \approx 1 - \frac{(\kappa\Delta t)^2}{3}.$$

The phase-speed error, like the amplitude error, is $O[(\Delta t)^2]$.

The trapezoidal scheme gives the best results; it generates no amplitude error, and its relative phase change is

$$R_{\text{trapezoidal}} = \frac{1}{\kappa\Delta t} \arctan\left(\frac{\kappa\Delta t}{1 - \kappa^2\Delta t^2/4}\right).$$

For small values of $\kappa\Delta t$, this may be approximated using Taylor series expansions as

$$R_{\text{trapezoidal}} \approx \frac{1}{\kappa\Delta t} \arctan\left(\kappa\Delta t\left(1 + \frac{\kappa^2\Delta t^2}{4}\right)\right) \approx 1 - \frac{\kappa^2\Delta t^2}{12}.$$

As with forward differencing and backward differencing, the trapezoidal scheme retards the phase change of well-resolved oscillations. However, the deceleration is only $\frac{1}{4}$ as great as that produced by the other schemes.

Although the trapezoidal scheme is accurate and unconditionally stable, it suffers from one serious disadvantage: It requires the evaluation of $F(\phi^{n+1})$ during the computation of ϕ^{n+1}. A scheme such as the trapezoidal method, in which the calculation of ϕ^{n+1} depends on $F(\phi^{n+1})$, is known as an *implicit* method. If the calculation of ϕ^{n+1} does not depend on $F(\phi^{n+1})$, the scheme is *explicit*. In the case of the oscillation equation, implicitness is a trivial complication. However, if F is a nonlinear function, any implicit finite-difference scheme will convert the differential equation into a nonlinear algebraic equation for ϕ^{n+1}. In the general case, the solution to this nonlinear equation must be obtained by some iterative technique. Thus, implicit finite-difference schemes generally require much more computation per individual time step than do similar explicit methods. Some of that extra computation may be offset if the implicit method is unconditionally stable, in which case the step size is determined solely by accuracy considerations, and the implicit time step can sometimes be much larger than the maximum stable time step of comparable explicit schemes.

2.3.3 *Multistage Methods*

All stable schemes of the form (2.33) have the disadvantage that they are implicit. Moreover, all except the trapezoidal scheme are only of first order. Is there a stable, accurate scheme that is not implicit? One may attempt to construct such a scheme by evaluating the function F in (2.32) at additional points in the interval $(n\Delta t, (n+1)\Delta t)$ and using this extra information to improve the accuracy of the calculation. These schemes are often referred to as multistage methods, because each integration step may require the estimation of ψ at several intermediate times, or "stages," before a final approximation to $\psi((n+1)\Delta t)$ is obtained. Each stage involves an additional evaluation of F, the right side of the differential equation. The family of consistent two-stage schemes may be written in the general form

$$\tilde{\phi}^{n+\alpha} = \phi^n + \alpha\Delta t F(\phi^n),$$
$$\phi^{n+1} = \phi^n + \beta\Delta t F(\tilde{\phi}^{n+\alpha}) + (1-\beta)\Delta t F(\phi^n).$$

Here $\tilde{\phi}^{n+\alpha}$ is an "intermediate" approximation to $\psi[(n+\alpha)\Delta t]$. One might attempt to choose α and β to maximize the *order* of the local truncation error. This criterion does not produce a unique solution, but rather leads to the requirement $\alpha\beta = \frac{1}{2}$. The family of schemes satisfying $\alpha\beta = \frac{1}{2}$ compose the set of second-order *Runge–Kutta* methods. One particular member of the Runge–Kutta family is the *Heun* method, obtained by setting $\alpha = 1$, $\beta = \frac{1}{2}$. The Heun method creates a trapezoidal-like approximation to the integral of F, but differs from the true trapezoidal method because $F(\phi^{n+1})$ is replaced by the estimate $F(\tilde{\phi}^{n+1})$. An-

other Runge–Kutta scheme is the *midpoint* method, in which $\alpha = \frac{1}{2}$ and $\beta = 1$. An example of a non-Runge–Kutta scheme is the *Matsuno*, or *forward–backward*, method, for which $\alpha = 1$ and $\beta = 1$ (Matsuno 1966b).

If the preceding multistage formula is applied to the oscillation equation, the result is

$$\phi^{n+1} = \phi^n + \beta i\kappa\Delta t(\phi^n + \alpha i\kappa\Delta t\phi^n) + (1 - \beta)i\kappa\Delta t\phi^n. \tag{2.38}$$

The amplification factor is

$$A = 1 + i\kappa\Delta t - \alpha\beta(\kappa\Delta t)^2,$$

and

$$|A|^2 = 1 + (1 - 2\alpha\beta)(\kappa\Delta t)^2 + \alpha^2\beta^2(\kappa\Delta t)^4, \tag{2.39}$$

which shows that the set of second-order Runge–Kutta schemes (i.e., those schemes for which $\alpha\beta = \frac{1}{2}$) have $O[(\Delta t)^4]$ amplitude error, whereas the amplitude error in the two-stage non-Runge–Kutta schemes is $O[(\Delta t)^2]$. Unfortunately, all the Runge–Kutta schemes are unstable, since in the limit of good numerical resolution,

$$|A|_{\text{R-K2}} \approx 1 + \tfrac{1}{8}(\kappa\Delta t)^4.$$

Although the Runge–Kutta schemes are unstable, the growth is $O[(\Delta t)^4]$. At a given step size, the erroneous amplification produced by the second-order Runge–Kutta methods will be much weaker than the $O[(\Delta t)^2]$ growth produced by forward time-differencing (see [2.36]). The slow growth generated by the Runge–Kutta methods can sometimes be tolerated if $\kappa\Delta t$ is sufficiently small and the total length of the integration is sufficiently short.[5]

Many physical systems contain several different modes, each oscillating at a different frequency. When simulating these systems, the highest-frequency components of the numerical solution are likely to be most seriously in error because of their poor numerical resolution. It is precisely these poorly resolved features that amplify most rapidly in the Runge–Kutta solutions. The amplification of the highest-frequency components can be prevented by choosing other values for α and β, although such a choice also increases the truncation error. According to (2.39), very low frequency oscillations ($\kappa\Delta t \ll 1$) will be stable whenever $\alpha\beta$ is greater than $\frac{1}{2}$. Matsuno (1966b) suggested setting $\alpha = 1$, $\beta = 1$, in which case (2.39) becomes

$$|A|^2_{\text{Matsuno}} = 1 - (\kappa\Delta t)^2 + (\kappa\Delta t)^4. \tag{2.40}$$

The Matsuno scheme damps the solution whenever $0 < \kappa\Delta t < 1$. Differentiation of (2.40) with respect to $\kappa\Delta t$ shows that the maximum damping occurs when

[5] As explored in Problem 15, the auxiliary relation $O(\Delta t) \le O[(\Delta x)^{4/3}]$ may be required to ensure the convergence of finite-difference approximations to the advection equation when the time difference is evaluated by a second-order Runge–Kutta scheme.

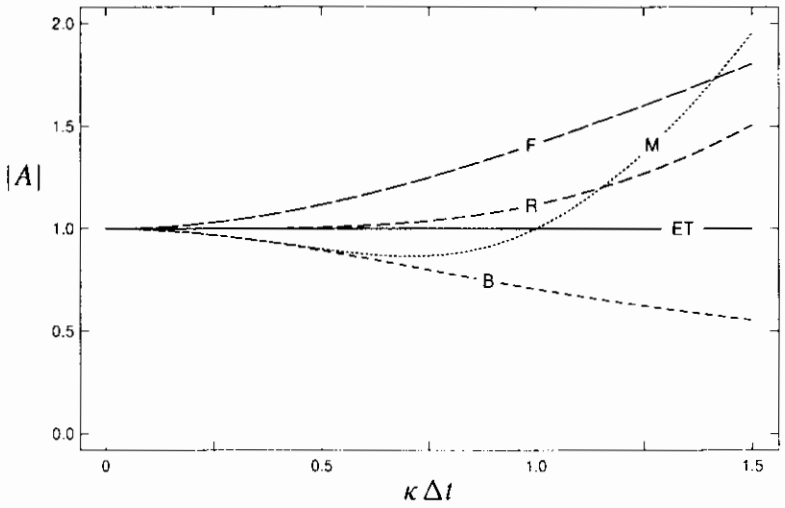

FIGURE 2.2. The modulus of the amplification factor $|A|$ as a function of temporal resolution $\kappa\Delta t$ for the true solution and five two-level schemes: exact solution and trapezoidal method (ET), forward differencing (F), backward differencing (B), second-order Runge–Kutta (R), and Matsuno (M).

$\kappa\Delta t = 1/\sqrt{2}$. Thus, if the time step is chosen such that $0 \le \kappa\Delta t \le 1/\sqrt{2}$ for all frequencies κ in the physical system, Matsuno time-differencing will preferentially damp the highest-frequency waves. The damping properties of the Matsuno scheme have been exploited to eliminate high-frequency gravity waves generated during the initialization of weather prediction models. The standard Matsuno scheme produces too much damping, however, for most nonspecialized applications. The fourth-order Runge–Kutta scheme (see Section 2.3.6) may also be used to preferentially damp high-frequency modes, and in most instances it would be a better choice than the Matsuno scheme because it is more efficient and far more accurate.

The amplitude errors generated by the preceding two-level schemes are compared in Fig. 2.2. The strong damping associated with the backward and Matsuno schemes is evident, along with the rapid amplification produced by forward differencing. These relatively large errors may be contrasted with the significantly weaker amplification produced by the second-order Runge–Kutta method and the neutral amplification of the trapezoidal method.

The relative phase change associated with the general two-stage scheme (2.38) is

$$R = \frac{1}{\kappa\Delta t}\arctan\left(\frac{\kappa\Delta t}{1 - \alpha\beta(\kappa\Delta t)^2}\right).$$

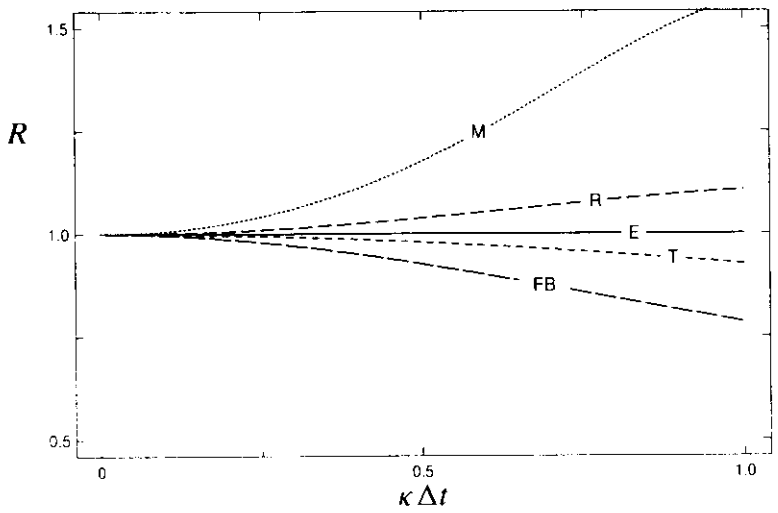

FIGURE 2.3. Relative phase change R as a function of temporal resolution $\kappa\Delta t$ for the true solution and five two-level schemes: exact solution (E), trapezoidal method (T), forward and backward differencing (FB), second-order Runge–Kutta (R), and Matsuno (M).

In the limit of good numerical resolution, the relative phase changes produced by second-order Runge–Kutta schemes and the Matsuno scheme are

$$R_{\text{R-K2}} \approx 1 + \tfrac{1}{6}(\kappa\Delta t)^2, \qquad R_{\text{Matsuno}} \approx 1 + \tfrac{2}{3}(\kappa\Delta t)^2.$$

The relative phase change for several two-level schemes is plotted as a function of temporal resolution in Fig. 2.3. The Matsuno and second-order Runge–Kutta schemes are accelerating, whereas the forward, backward, and trapezoidal schemes are decelerating.

2.3.4 *Three-Level Schemes*

As an alternative to multistage methods, information from several earlier time levels can be incorporated into the integration formula. This increases the storage requirements of the scheme, but it avoids the necessity of performing more than one evaluation of the right side per time step. According to the terminology developed for ordinary differential equations, these are *multistep* methods. A typical multistep ordinary differential equation solver might use data from a half dozen preceding time levels. The large storage requirements of many atmospheric and ocean models have, however, discouraged researchers from using data from more than two earlier time levels in their time-differencing schemes. Let us therefore consider the family of three-time-level schemes.

The general form for an explicit three-time-level method is

$$\phi^{n+1} = \alpha_1\phi^n + \alpha_2\phi^{n-1} + \beta_1\Delta t F(\phi^n) + \beta_2\Delta t F(\phi^{n-1}). \tag{2.41}$$

When formulating a three-level scheme, one seeks to improve upon the two-time-level methods, so it is reasonable to require that the global truncation error be of at least second order. The three-level scheme will be of at least second order if

$$\alpha_1 = 1 - \alpha_2, \qquad \beta_1 = \tfrac{1}{2}(\alpha_2 + 3), \qquad \beta_2 = \tfrac{1}{2}(\alpha_2 - 1), \tag{2.42}$$

where the coefficient α_2 remains a free parameter. One could choose α_2 to further reduce the truncation error, but the result is not a useful scheme (see problem 9). The most important explicit three-level schemes are obtained by choosing α_2 to minimize the amount of data that must be stored and carried over from the $n-1$ time level, i.e., by setting $\alpha_2 = 1$, in which case $\beta_2 = 0$, or by setting $\alpha_2 = 0$. If α_2 is set to one, (2.41) becomes the *leapfrog* scheme. The choice $\alpha_2 = 0$ gives the second-order *Adams–Bashforth* method. The remainder of this section will be devoted to an examination of these two schemes.

If the leapfrog scheme is applied to the oscillation equation, the result is

$$\phi^{n+1} = \phi^{n-1} + 2i\kappa\Delta t\phi^n. \tag{2.43}$$

Since the preceding is a linear finite-difference equation with constant coefficients, the amplification factor is constant from time step to time step and satisfies the quadratic equation

$$A^2 - 2i\kappa\Delta t A - 1 = 0.$$

The two roots are

$$A_\pm = i\kappa\Delta t \pm \left(1 - \kappa^2\Delta t^2\right)^{1/2}. \tag{2.44}$$

In the limit of good numerical resolution, $\kappa\Delta t \to 0$ and $A_+ \to 1$; $A_- \to -1$. Evidently, the numerical solution is capable of behaving in two very different ways, or *modes*. The mode associated with A_+ is known as the *physical mode* because it approximates the solution to the original differential equation. The mode associated with A_- is referred to as the *computational mode*, since it arises solely as an artifact of the numerical computation. If $|\kappa\Delta t| \le 1$, the second term in (2.44) is real and $|A_+| = |A_-| = 1$; i.e., both the physical and the computational modes are stable and neutral. In the case $\kappa\Delta t > 1$,

$$|A_+| = |i\kappa\Delta t + i\left(\kappa^2\Delta t^2 - 1\right)|^{1/2} > |i\kappa\Delta t| > 1,$$

and the scheme is unstable. When $\kappa\Delta t < -1$, a similar argument shows that $|A_-| > 1$. Note that when $\kappa\Delta t > 1$, A_+ lies on the positive imaginary axis in the complex plane, and thus each integration step produces a 90° shift in the phase of the oscillation. As a consequence, unstable leapfrog solutions grow with a period of $4\Delta t$.

The complete leapfrog solution can typically be written as a linear combination of the physical and computational modes. An exception occurs if $\kappa\Delta t = \pm 1$, in

which case $A_+ = A_-$, and the physical and computational modes are not linearly independent. In such circumstances, the general solution to (2.43) has the form

$$\phi^n = C_1(i\kappa\Delta t)^n + C_2 n(i\kappa\Delta t)^n.$$

Since the magnitude of the preceding solution grows as function of time step, the leapfrog scheme is *not* stable when $|\kappa\Delta t| = 1$. Nevertheless, the $O(n)$ growth of the solution that occurs when $\kappa\Delta t = \pm 1$ is far slower than the $O(A^n)$ amplification that is produced when $|\kappa\Delta t| > 1$.

The source of the computational mode is particularly easy to analyze in the trivial case of $\kappa = 0$; then the analytic solution to the oscillation equation (2.30) is $\psi(t) = C$, where C is a constant determined by the initial condition at $t = t_0$. Under these circumstances, the leapfrog scheme reduces to

$$\phi^{n+1} = \phi^{n-1}, \tag{2.45}$$

and the amplification factor has the roots $A_+ = 1$, $A_- = -1$. The initial condition requires $\phi^0 = C$, which, according to the difference scheme (2.45), also guarantees that $\phi^2 = \phi^4 = \phi^6 = \cdots = C$. The odd time levels are determined by a second, computational, initial condition imposed on ϕ^1. In practice, ϕ^1 is often obtained from ϕ^0 by taking a single time step with a two-level method, and the resulting approximation to $\psi(t_0 + \Delta t)$ will contain some error E. It is obvious that in our present example, the correct choice for ϕ^1 is C, but in order to mimic the situation in a more general problem, suppose that $\phi^1 = C + E$. Then the numerical solution at any subsequent time will be the sum of two modes,

$$\phi^n = (C + E/2) - (-1)^n E/2.$$

to yield $\phi^n = (C + E/2) - (-1)^n E/2$.

Here, the first term represents the physical mode, and the second term represents the computational mode. The computational mode oscillates with a period of $2\Delta t$, and does not decay with time. In this example, the amplitude of the computational mode is completely determined by the error in the specification of the computational initial condition ϕ^1. Since there is no coupling between the physical and computational modes in solutions to linear problems, the errors in the initial conditions also govern the amplitude of the computational mode in leapfrog solutions to most linear equations. If the governing equations are nonlinear, however, the nonlinear terms introduce a coupling between ϕ_+ and ϕ_- that often amplifies the computational mode until it eventually dominates the solution. This spurious growth of the computational mode can be avoided by periodically discarding the solution at ϕ^{n-1} and taking a single time step with a two-level scheme, or by filtering the high-frequency components of the numerical solution. Various techniques for controlling the leapfrog scheme's computational mode will be discussed in Section 2.3.5.

The relative phase changes in the two leapfrog modes are

$$R_{\pm_{\text{leapfrog}}} = \frac{1}{\kappa\Delta t}\arctan\left(\frac{\pm\kappa\Delta t}{(1-\kappa^2\Delta t^2)^{1/2}}\right).$$

The computational mode and the physical mode oscillate in opposite directions. In the limit of good time resolution,

$$R_{+_{\text{leapfrog}}} \approx 1 + \frac{(\kappa\Delta t)^2}{6},$$

showing that leapfrog time differencing is accelerating.

Now consider the second-order Adams–Bashforth method, which has the form

$$\phi^{n+1} = \phi^n + \Delta t\left(\tfrac{3}{2}F(\phi^n) - \tfrac{1}{2}F(\phi^{n-1})\right). \tag{2.46}$$

The second-order Adams–Bashforth formula may be interpreted as numerical integration via the midpoint method, except that the value of the integrand at the midpoint, $F(\phi^{n+1/2})$, is obtained by linear extrapolation. Application of the Adams–Bashforth method to the oscillation equation yields

$$\phi^{n+1} = \phi^n + i\kappa\Delta t\left(\tfrac{3}{2}\phi^n - \tfrac{1}{2}\phi^{n-1}\right).$$

The amplification factor associated with this scheme is given by the quadratic

$$A^2 - \left(1 + \frac{3i\kappa\Delta t}{2}\right)A + \frac{i\kappa\Delta t}{2} = 0,$$

in which case

$$A_\pm = \frac{1}{2}\left(1 + \frac{3i\kappa\Delta t}{2} \pm \left(1 - \frac{9(\kappa\Delta t)^2}{4} + i\kappa\Delta t\right)^{1/2}\right). \tag{2.47}$$

As the numerical resolution increases, $A_+ \to 1$ and $A_- \to 0$. Thus, the Adams–Bashforth method damps the computational mode. The highly desirable damping of the computational mode is somewhat offset by a weak instability in the physical mode. This instability is revealed if (2.47) is approximated under the assumption that $\kappa\Delta t$ is small; then

$$A_+ = \left(1 - \frac{(\kappa\Delta t)^2}{2} - \frac{(\kappa\Delta t)^4}{8} - \cdots\right) + i\left(\kappa\Delta t + \frac{(\kappa\Delta t)^3}{4} + \cdots\right),$$

$$A_- = \left(\frac{(\kappa\Delta t)^2}{2} + \frac{(\kappa\Delta t)^4}{8} + \cdots\right) + i\left(\frac{\kappa\Delta t}{2} - \frac{(\kappa\Delta t)^3}{4} - \cdots\right),$$

and

$$|A_+|_{\text{A-B2}} \approx 1 + \tfrac{1}{4}(\kappa\Delta t)^4, \qquad |A_-|_{\text{A-B2}} \approx \tfrac{1}{2}\kappa\Delta t.$$

The modulus of the amplification factor of the physical mode exceeds unity by an $O[(\kappa\Delta t)^4]$ term. As was the case for the second-order Runge–Kutta methods, this weak instability can sometimes be tolerated if the length of the integration is limited and the time step is sufficiently small. The dependence of $|A_+|$ and $|A_-|$

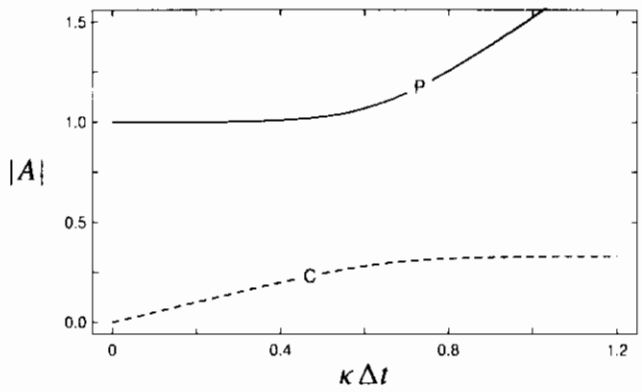

FIGURE 2.4. Modulus of the amplification factors for the second-order Adams–Bashforth scheme as a function of temporal resolution $\kappa \Delta t$. The solid and dashed lines represent the physical and the computational modes, respectively.

upon temporal resolution is plotted in Fig. 2.4. The relative phase change in the physical mode in the Adams–Bashforth method is

$$R_{\text{A-B2}} \approx 1 + \frac{5}{12}(\kappa \Delta t)^2,$$

where as before, it is assumed that $\kappa \Delta t \ll 1$.

Three-level schemes require information from two previous time levels, yet initial conditions for well-posed physical problems give information about the solution at only one time. It is therefore necessary to initialize the leapfrog and second-order Adams–Bashforth methods by taking a single time step using a two-level method. In most instances, a simple forward step is adequate. Although forward differencing is unstable, the amplification produced by a single step will generally not be large (see Problem 20). Moreover, even though the truncation error of a forward difference is $O(\Delta t)$, the execution of a single forward time step does not reduce the $O\left[(\Delta t)^2\right]$ global accuracy of leapfrog and Adams–Bashforth integrations. The basic reason that $O\left[(\Delta t)^2\right]$ accuracy is preserved is that forward differencing is used only over a Δt-long portion of the total integration. The contribution to the total error produced by the accumulation of $O\left[(\Delta t)^2\right]$ errors over a finite time interval is of the same order as the error arising from the accumulation of $O(\Delta t)$ errors over a time Δt.

2.3.5 *Controlling the Leapfrog Computational Mode*

The trapezoidal method is unconditionally stable, and it has the lowest truncation error of all the schemes presented in Sections 2.3.2–2.3.4. The weakness of the trapezoidal method is that it is implicit, which is often a serious disadvantage in computing numerical solutions to wave propagation problems. The best explicit scheme presented in the preceding sections might appear to be the leapfrog scheme. The leapfrog scheme is stable (unlike the second-order Runge–Kutta and Adams–Bashforth methods), it is of second order (unlike the Matsuno method), and it requires only one function evaluation per time step (unlike the Matsuno and Runge–Kutta schemes). The weakness of the leapfrog scheme is its undamped computational mode, which slowly amplifies during simulations of nonlinear wave propagation problems and generates an instability often known as "time splitting."

One way to control the growth of the computational mode is to periodically discard the data from the $n - 1$ time level (or alternatively, to average the n and $n - 1$ time-level solutions) and to restart the integration using a two-time-level method. Forward differencing is often used to reinitialize leapfrog integrations. Forward differencing is easy to implement, but since it is a first-order scheme, it degrades the second-order accuracy of the unadulterated leapfrog method. In addition, forward differencing is unstable and tends to amplify the high-frequency components of the solution. Moreover, it is difficult to quantify these adverse effects, since they vary according to the number of leapfrog steps between each forward step. Restarting with a second-order Runge–Kutta scheme is a far better choice, since this preserves second-order accuracy and produces less unstable amplification. The midpoint method is one second-order Runge–Kutta formulation that can be used to restart leapfrog integrations in complex numerical models without greatly complicating the model code. A midpoint-method restart may be implemented by taking a forward step of length $\Delta t/2$ followed by a single leapfrog step of length $\Delta t/2$.

In atmospheric science, it is common, though questionable, practice to control the computational mode through the use of a second-order time filter. Consider, therefore, the centered second-order time filter

$$\overline{\phi^n} = \phi^n + \gamma\left(\phi^{n+1} - 2\phi^n + \phi^{n-1}\right), \tag{2.48}$$

where ϕ^n denotes the solution at time $n\Delta t$ prior to time filtering, $\overline{\phi^n}$ is the solution after filtering, and γ is a positive real constant that determines the strength of the filter. The last term in (2.48) is the usual finite-difference approximation to the second derivative and preferentially damps the highest frequencies. Suppose that the unfiltered values are sampled from the exact solution to the oscillation equation; then

$$\overline{\phi^n} = \left(1 + \gamma\left(e^{i\kappa\Delta t} - 2 + e^{-i\kappa\Delta t}\right)\right)\phi^n.$$

Defining a *filter factor* $X = \overline{\phi^n}/\phi^n$, one obtains

$$X_{\text{centered}} = 1 - 2\gamma(1 - \cos\kappa\Delta t). \tag{2.49}$$

Since X_{centered} is real, it does not produce any change in the phase of the solution. In the limit $\kappa\Delta t \to 0$,

$$X_{\text{centered}} \approx 1 - \gamma(\kappa\Delta t)^2,$$

showing that well-resolved oscillations undergo an $O[(\Delta t)^2]$ damping. The centered filter has the greatest impact on the most poorly resolved component of the solution, the $2\Delta t$ oscillation. According to (2.49), each filter application reduces the amplitude of the $2\Delta t$ wave by a factor of $1 - 4\gamma$. If γ is specified to be $\frac{1}{4}$, each filtering operation will completely eliminate the $2\Delta t$ oscillation.

Robert (1966) and Asselin (1972) suggested a scheme to control the leapfrog computational mode by incorporating an approximate second-derivative time filter into the time integration cycle. They proposed following each leapfrog step

$$\phi^{n+1} = \overline{\phi^{n-1}} + 2\Delta t F(\phi^n)$$

by the filtering operation

$$\overline{\phi^n} = \phi^n + \gamma\left(\overline{\phi^{n-1}} - 2\phi^n + \phi^{n+1}\right). \tag{2.50}$$

A filter parameter of $\gamma = 0.06$ is typically used in global atmospheric models. Values of $\gamma = 0.2$ are common in convective cloud models; indeed, Schlesinger et al. (1983) recommend choosing γ in the range 0.25–0.3 for certain advection–diffusion problems.

If the Asselin-filtered leapfrog scheme is applied to the oscillation equation, the amplification factor is determined by the simultaneous equations

$$A^2\phi^{n-1} = \overline{\phi^{n-1}} + 2i\kappa\Delta t A\phi^{n-1}, \tag{2.51}$$

$$\overline{A\phi^{n-1}} = A\phi^{n-1} + \gamma\left(\overline{\phi^{n-1}} - 2A\phi^{n-1} + A^2\phi^{n-1}\right). \tag{2.52}$$

Under the assumption that $\overline{(A\phi^n)} = A(\overline{\phi^n})$, whose validity will be discussed shortly, (2.52) may be written

$$(A - \gamma)\overline{\phi^{n-1}} = A\left((1 - 2\gamma) + A\gamma\right)\phi^{n-1}. \tag{2.53}$$

Eliminating $\overline{\phi^{n-1}}$ between (2.51) and (2.53) yields

$$A_\pm = \gamma + i\kappa\Delta t \pm \left((1-\gamma)^2 - \kappa^2\Delta t^2\right)^{1/2}, \tag{2.54}$$

which reduces to the result for the standard leapfrog scheme when $\gamma = 0$. In the limit of small $\kappa\Delta t$, the amplification factor for the Asselin-filtered physical mode becomes

$$A_{\text{Asselin–LF}} = 1 + i\kappa\Delta t - \frac{(\kappa\Delta t)^2}{2(1-\gamma)} + O[(\kappa\Delta t)^4].$$

A comparison of this expression with the asymptotic behavior of the exact amplification factor

$$A_e = e^{i\kappa\Delta t} = 1 + i\kappa\Delta t - \frac{(\kappa\Delta t)^2}{2} - i\frac{(\kappa\Delta t)^3}{6} + O[(\kappa\Delta t)^4]$$

shows that the *local* truncation error of the Asselin-filtered leapfrog scheme is $O[(\kappa\Delta t)^2]$. In contrast, the local truncation error of the unfiltered leapfrog scheme ($\gamma = 0$) is $O[(\kappa\Delta t)^3]$. Thus, Asselin filtering degrades the *global* truncation error of the leapfrog scheme from second order to first order.

The preceding derivation was based on the assumption that $\overline{(A\phi^n)} = A(\overline{\phi^n})$. Is this justified? In practice, the initial condition is not time filtered; one simply defines $\overline{\phi^0} \equiv \phi^0$. Thus,

$$\overline{(A\phi^0)} - A(\overline{\phi^0}) = \overline{\phi^1} - \phi^1 = \gamma\left(\phi^0 - 2\phi^1 + \phi^2\right) \neq 0.$$

Nevertheless, an application of the Asselin time filter to ϕ^{n+1} gives

$$\begin{aligned}\overline{(A\phi^n)} &= A\phi^n + \gamma\left(\overline{(A\phi^{n-1})} - 2A\phi^n + A^2\phi^n\right)\\ &= A\left(\phi^n + \gamma\left(\overline{\phi^{n-1}} - 2\phi^n + A\phi^n\right)\right) + \gamma\left(\overline{(A\phi^{n-1})} - A\overline{\phi^{n-1}}\right)\\ &= A(\overline{\phi^n}) + \gamma\left(\overline{(A\phi^{n-1})} - A\overline{\phi^{n-1}}\right),\end{aligned}$$

from which it follows that

$$\overline{(A\phi^n)} - A(\overline{\phi^n}) = \gamma^n[\overline{\phi^1} - \phi^1]. \tag{2.55}$$

In all cases of practical interest, $n \gg 1$ and $\gamma \ll 1$; therefore, (2.55) implies that A may be factored out of the filtering operation with negligible error, and that (2.53) is indeed equivalent to (2.52).

In the limit of $\kappa\Delta t \ll 1$, the modulus of the amplification factor for the Asselin-filtered leapfrog scheme may be approximated as

$$|A_+|_{\text{Asselin–LF}} \approx 1 - \frac{\gamma}{2(1-\gamma)}(\kappa\Delta t)^2,$$

$$|A_-|_{\text{Asselin–LF}} \approx (1 - 2\gamma) + \frac{\gamma}{2 - 6\gamma + 4\gamma^2}(\kappa\Delta t)^2.$$

Like other first-order schemes, such as forward differencing and the Matsuno method, the physical mode in the Asselin-filtered leapfrog scheme has an $O[(\Delta t)^2]$ amplitude error. The behavior of the computational mode is also notable in that $|A_-|$ does not approach zero as $|\kappa\Delta t| \to 0$.

The asymptotic behavior of the relative phase change in the physical mode is

$$R_{+\text{Asselin–LF}} \approx 1 + \frac{1 + 2\gamma}{6(1-\gamma)}(\kappa\Delta t)^2.$$

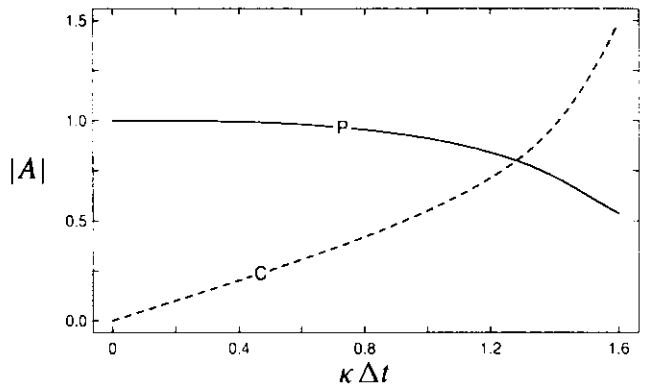

FIGURE 2.5. Modulus of the amplification factor for the leapfrog–trapezoidal method as a function of temporal resolution $\kappa\,\Delta t$. The solid and dashed lines represent the physical and the computational modes, respectively.

Asselin–Robert filtering increases the phase error; doubling it as γ increases from 0 to $\frac{1}{4}$.

The main problem with the Asselin-filtered leapfrog scheme is its first-order accuracy. There are two alternative techniques that control the leapfrog computational mode without sacrificing second-order accuracy—the leapfrog–trapezoidal method and the Magazenkov method. The leapfrog–trapezoidal method (Kurihara 1965; Zalesak 1979) is an iterative scheme in which a leapfrog predictor is followed by a trapezoidal correction step, i.e.,

$$\phi^* = \phi^{n-1} + 2\Delta t F(\phi^n),$$
$$\phi^{n+1} = \phi^n + \frac{\Delta t}{2}\left(F(\phi^n) + F(\phi^*)\right).$$

If this scheme is applied to the oscillation equation, the amplitude and relative phase changes in the physical mode are

$$|A|_{\text{LF–trap}} \approx 1 - \frac{(\kappa\Delta t)^4}{4}, \qquad R_{\text{LF–trap}} \approx 1 - \frac{(\kappa\Delta t)^2}{12},$$

where as usual, these approximations hold for small $\kappa\,\Delta t$. Leapfrog–trapezoidal integrations of the oscillation equation will be stable provided that $\kappa\,\Delta t \le \sqrt{2}$. The amplitude error associated with the leapfrog–trapezoidal scheme is plotted as a function of temporal resolution in Fig. 2.5.

Magazenkov (1980) suggested that the computational mode could be controlled by alternating each leapfrog step with a second-order Adams–Bashforth step. Since the Magazenkov method uses different schemes on the odd and even time steps, the amplification factor differs between the odd and even steps. In order to

analyze the behavior of the Magazenkov method, it is therefore, best to consider the averaged effect of a combined leapfrog–Adams–Bashforth cycle.

Thus, for analysis purposes, the scheme will be written as a system of equations that maps (ϕ^{n-2}, ϕ^{n-1}) into (ϕ^n, ϕ^{n+1}),

$$\phi^n = \phi^{n-2} + 2\Delta t F(\phi^{n-1}), \tag{2.56}$$
$$\phi^{n+1} = \left(\phi^{n-2} + 2\Delta t F(\phi^{n-1})\right) + \frac{\Delta t}{2}\left[3F\left(\phi^{n-2} + 2\Delta t F(\phi^{n-1})\right) - F(\phi^{n-1})\right]. \tag{2.57}$$

When actually implementing the Magazenkov method, however, (2.57) would be replaced by the equivalent expression (2.46). Application of (2.56) and (2.57) to the oscillation equation yields a system of two equations in two unknowns,

$$\begin{pmatrix} 1 & 2i\kappa\Delta t \\ 1 + \frac{3}{2}i\kappa\Delta t & \frac{3}{2}i\kappa\Delta t - 3(\kappa\Delta t)^2 \end{pmatrix}\begin{pmatrix} \phi^{n-2} \\ \phi^{n-1} \end{pmatrix} = \begin{pmatrix} \phi^n \\ \phi^{n+1} \end{pmatrix}.$$

The coefficient matrix in the preceding equation determines the combined amplification and phase-shift generated by each pair of leapfrog and Adams–Bashforth time steps. The eigenvalues of the coefficient matrix are determined by the characteristic equation

$$\lambda^2 - \left(\frac{3i\kappa\Delta t}{2} - 3(\kappa\Delta t)^2 + 1\right)\lambda - \frac{i\kappa\Delta t}{2} = 0.$$

The eigenvalues are distinct and have magnitudes less than one when $|\kappa\,\Delta t| < \frac{2}{3}$, implying that the method is conditionally stable. For well-resolved physical-mode oscillations, the average amplitude and relative phase change per single time step are

$$|A|_{\text{Mag}} = (|\lambda|)^{1/2} \approx 1 - \frac{(\kappa\Delta t)^4}{4}, \qquad R_{\text{Mag}} = 1 + \frac{(\kappa\Delta t)^2}{6}.$$

The average amplitude error per single time step is plotted as a function of temporal resolution in Fig. 2.6.

2.3.6 *Higher-Order Schemes*

Relatively little attention has been devoted to the incorporation of third- or fourth-order time differencing into schemes for the numerical solution of partial differential equations. A major reason for the lack of interest in higher-order time differencing is that in many applications the errors in the numerical representation of the spatial derivatives dominate the time-discretization error, and as a consequence it might appear unlikely that the accuracy of the solution could be improved through the use of higher-order time differences. Several higher-order schemes do, nevertheless, have attractive stability characteristics that merit further discussion. Schemes of particular interest are the third-order Adams–Bashforth method and the third- and fourth-order Runge–Kutta methods. Whereas the second-order

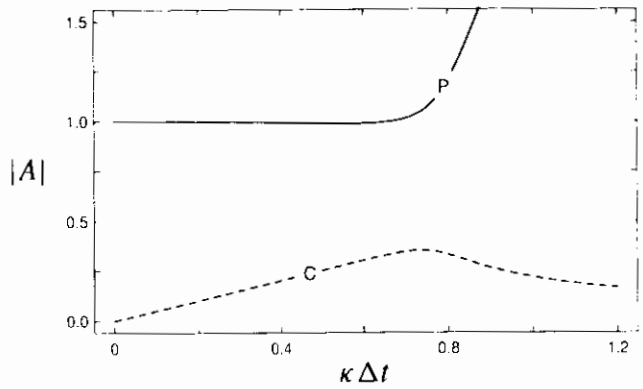

FIGURE 2.6. Modulus of the average amplification factor per single time step of the Magazenkov method plotted as a function of temporal resolution $\kappa \Delta t$. The solid and dashed lines represent the physical and the computational modes, respectively.

Runge–Kutta and Adams–Bashforth schemes produce amplifying solutions to the oscillation equation, their third- and fourth-order formulations are stable with strongly damped computational modes. As such, they offer additional possibilities for achieving better than second-order accuracy with a stable explicit time-differencing scheme. Moreover, they are better suited than the leapfrog–trapezoidal and Magazenkov schemes for the solution of a generalized oscillation equation in which κ has a positive imaginary part and the amplitude of the oscillation decays with time.

The distinctive advantage of the third-order Runge–Kutta scheme is the possibility of selecting a low-storage variant. As discussed earlier, there is no unique formula for the second-order Runge–Kutta method. Instead, the requirement of second-order accuracy leads to a family of schemes whose coefficients depend on the value of a free parameter. The coefficients of higher-order Runge–Kutta schemes are similarly nonunique. For example, the coefficients of the third- and fourth-order Runge–Kutta methods are determined by two independent parameters (Gear 1971, pp. 34–35). Williamson (1980) examined the subset of all possible third-order Runge–Kutta schemes that may be evaluated with minimal computer storage and recommended the following scheme:

$$
\begin{aligned}
q_1 &= \Delta t F(\phi^n), & \phi_1 &= \phi^n + q_1/3, \\
q_2 &= \Delta t F(\phi_1), -5q_1/9, & \phi_2 &= \phi_1 + 15q_2/16, \\
q_3 &= \Delta t F(\phi_2) - 153q_2/128, & \phi^{n+1} &= \phi_2 + 8q_3/15.
\end{aligned}
$$

In practical applications involving time-dependent partial differential equations, ϕ^n may be an extremely long vector of unknown variables (e.g., the velocity, temperature, and pressure at every node on a large three-dimensional mesh). It may therefore be difficult to store several copies of ϕ and $F(\phi)$ in the random

access memory (RAM) of a digital computer. If m is the number of unknowns in ϕ, the Williamson–Runge–Kutta scheme economizes on storage by allowing the integration to proceed using only $2m$ storage locations, divided between the arrays q and ϕ, which are overwritten three times during each integration step. The storage requirement of the Williamson–Runge–Kutta scheme is identical to that of forward time-differencing and the standard leapfrog scheme, and is less than that required for the Asselin-filtered leapfrog scheme.

Finite-difference formulae for several time-differencing schemes are summarized in Table 2.1. Although it is not apparent from their most common names, most of the schemes shown in Table 2.1 are either Adams–Bashforth, Adams–Moulton, or Runge–Kutta schemes. Adams–Bashforth schemes are explicit multilevel methods whose first-order variant is the forward difference. Adams–Moulton methods are implicit multilevel methods that include backward and trapezoidal differencing as their first- and second-order representatives. One way to approximate the solution of the implicit algebraic equations generated by an Adams–Moulton method is to estimate ϕ^{n+1} using an Adams–Bashforth scheme and then substitute this estimate into the Adams–Moulton formula. The third-order Adams–Bashforth–Moulton corrector is listed in Table 2.1. The particular second-order Runge–Kutta scheme appearing in Table 2.1 is also the second-order Adams–Bashforth–Moulton predictor corrector.

Several important properties of the schemes listed in Table 2.1 are given in Table 2.2. The column labeled "storage factor" indicates the number of full arrays that must be allocated for each unknown variable in order to implement each scheme. Storage factors are not provided for the implicit methods listed in Table 2.2 because the storage factor for implicit methods can vary from problem to problem, depending on the numerical algorithm used to solve the implicit system. Inspection of Table 2.2 clearly reveals the low-storage advantage of the third-order Runge–Kutta scheme. This advantage may, however, be slightly exaggerated, since the storage factors listed in Table 2.2 are upper limits that allow each method to be programmed in a completely straightforward manner. In many instances, it is possible to utilize less memory than that suggested by the storage factor if newly computed quantities are initially placed in a small, temporary storage array. As an example, when integrating a partial differential equation with forward time differencing, it is not generally possible to write the newly computed ϕ_j^{n+1} directly into the storage occupied by ϕ_j^n, because ϕ_j^n may be required for the computation of ϕ_{j+1}^{n+1}. However, at some point in the integration cycle, ϕ_j^n will no longer be needed, and at that stage it may be overwritten by ϕ_j^{n+1}. During the interim between the calculation of ϕ_j^{n+1} and the last use of ϕ_j^n, ϕ_j^{n+1} may be held in a temporary storage array. In many applications, the temporary storage array can be much smaller than the full array required to hold a complete set of ϕ^n, and use of such a temporary array will reduce the storage factor by almost one unit.

In applications where storage is not a problem, the third-order Adams–Bashforth scheme

Method	*Order*	*Formula*
Forward	1	$\phi^{n+1} = \phi^n + hF(\phi^n)$
Backward	1	$\phi^{n+1} = \phi^n + hF(\phi^{n+1})$
Asselin Leapfrog	1	$\phi^{n+1} = \overline{\phi^{n-1}} + 2hF(\phi^n)$ $\overline{\phi^n} = \phi^n + \gamma(\overline{\phi^{n-1}} - 2\phi^n + \phi^{n+1})$
Leapfrog	2	$\phi^{n+1} = \phi^{n-1} + 2hF(\phi^n)$
Adams–Bashforth	2	$\phi^{n+1} = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$
Trapezoidal	2	$\phi^{n+1} = \phi^n + \frac{h}{2}\left[F(\phi^{n+1}) + F(\phi^n)\right]$
Runge–Kutta	2	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1$ $q_2 = hF(\phi_1) - q_1, \quad \phi^{n+1} = \phi_1 + q_2/2$
Magazenkov	2	$\phi^n = \phi^{n-2} + 2hF(\phi^{n-1})$ $\phi^{n+1} = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$
Leapfrog–Trapezoidal	2	$\phi_1 = \phi^{n-1} + 2hF(\phi^n)$ $\phi^{n+1} = \phi^n + \frac{h}{2}\left[F(\phi_1) + F(\phi^n)\right]$
Adams–Bashforth	3	$\phi^{n+1} = \phi^n + \frac{h}{12}\left[23F(\phi^n) - 16F(\phi^{n-1}) + 5F(\phi^{n-2})\right]$
Adams–Moulton	3	$\phi^{n+1} = \phi^n + \frac{h}{12}\left[5F(\phi^{n+1}) + 8F(\phi^n) - F(\phi^{n-1})\right]$
ABM Predictor–Corrector	3	$\phi_1 = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$ $\phi^{n+1} = \phi^n + \frac{h}{12}\left[5F(\phi_1) + 8F(\phi^n) - F(\phi^{n-1})\right]$
Runge–Kutta	3	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1/3$ $q_2 = hF(\phi_1) - 5q_1/9, \quad \phi_2 = \phi_1 + 15q_2/16$ $q_3 = hF(\phi_2) - 153q_2/128, \quad \phi^{n+1} = \phi_2 + 8q_3/15$
Runge–Kutta	4	$q_1 = hF(\phi^n), \quad q_2 = hF(\phi^n + q_1/2)$ $q_3 = hF(\phi^n + q_2/2), \quad q_4 = hF(\phi^n + q_3)$ $\phi^{n+1} = \phi^n + (q_1 + 2q_2 + 2q_3 + q_4)/6$

TABLE 2.1. Summary of methods for the solution of ordinary differential equations. The second- and third-order Runge–Kutta methods are low-storage variants; $h = \Delta t$.

Method	*Storage Factor*	*Efficiency Factor*	*Amplification Factor*	*Phase Error*	*Max s*
Forward	2	0	$1 + \frac{s^2}{2}$	$1 - \frac{s^2}{3}$	0
Backward	*	∞	$1 - \frac{s^2}{2}$	$1 - \frac{s^2}{3}$	∞
Asselin Leapfrog	3	< 1	$1 - \frac{\gamma s^2}{2(1-\gamma)}$	$1 + \frac{(1+2\gamma)s^2}{6(1-\gamma)}$	< 1
Leapfrog	2	1	1	$1 + \frac{s^2}{6}$	1
Adams Bashforth–2	3	0	$1 + \frac{s^4}{4}$	$1 + \frac{5}{12}s^2$	0
Trapezoidal	*	∞	1	$1 - \frac{s^2}{12}$	∞
Runge–Kutta–2	2	0	$1 + \frac{s^4}{8}$	$1 + \frac{s^2}{6}$	0
Magazenkov	3	0.67	$1 - \frac{s^4}{4}$	$1 + \frac{s^2}{6}$	0.67
Leapfrog–Trapezoidal	3	0.71	$1 - \frac{s^4}{4}$	$1 - \frac{s^2}{12}$	1.41
Adams–Bashforth–3	4	0.72	$1 - \frac{3}{8}s^4$	$1 + \frac{289}{720}s^4$	0.72
Adams–Moulton–3	*	0	$1 + \frac{s^4}{24}$	$1 - \frac{11}{720}s^4$	0
ABM Predictor–Corrector–3	4	0.60	$1 - \frac{19}{144}s^4$	$1 + \frac{1243}{8640}s^4$	1.20
Runge–Kutta–3	2	0.58	$1 - \frac{s^4}{24}$	$1 + \frac{s^4}{30}$	1.73
Runge–Kutta–4	4†	0.70	$1 - \frac{s^6}{144}$	$1 - \frac{s^4}{120}$	2.82

†A storage factor of 3 may be achieved following the algorithm of Blum (1962).

TABLE 2.2. Characteristics of the schemes listed in Table 2.1. The amplification factor and relative phase change are for well-resolved solutions to the oscillation equation, and $s = \kappa \Delta t$. "Max s" is the maximum value of $\kappa \Delta t$ for which the solution is nonamplifying. The storage and efficiency factors are defined in the text. No storage factor is given for implicit schemes.

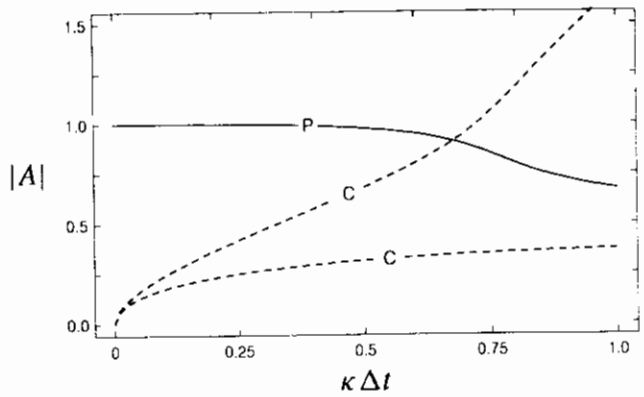

FIGURE 2.7. Modulus of the amplification factor for the third-order Adams–Bashforth method plotted as a function of temporal resolution $\kappa \Delta t$. The solid line represents the physical mode and the dashed lines the two computational modes.

$$\phi^{n+1} = \phi^n + \frac{\Delta t}{12}\left[23F(\phi^n) - 16F(\phi^{n-1}) + 5F(\phi^{n-2})\right] \tag{2.58}$$

can be an attractive alternative. The primary advantage of the third-order Adams–Bashforth scheme is its relative efficiency. In most practical applications involving partial differential equations, the bulk of the computational effort is associated with the evaluation of F, the function that determines the time derivative. Thus, a rough measure of the comparative efficiency of each method may be obtained by defining an efficiency factor as the maximum stable time step with which the oscillation equation can be integrated, divided by the number of evaluations of $F(\phi)$ that each scheme requires to perform a single integration step. Inspection of Table 2.2 shows that with the exception of the leapfrog scheme and its time-filtered variant, the third-order Adams–Bashforth scheme has the highest efficiency factor. The amplitude error in the third-order Adams–Bashforth solution to the oscillation equation is plotted in Fig. 2.7. Unlike the second-order Adams–Bashforth method, instability is not associated with unstable growth of the physical mode; instead, it is one of the computational modes that becomes unstable for $\kappa \Delta t > 0.724$.

Other schemes with efficiency factors almost as large as the third-order Adams–Bashforth method are the leapfrog–trapezoidal method and the fourth-order Runge–Kutta method. The leapfrog–trapezoidal scheme, being a lower-order scheme, is not a particularly attractive alternative. On the other hand, the fourth-order Runge–Kutta scheme is of higher order and potentially attractive, but its high efficiency factor is somewhat misleading. Figure 2.8 shows the amplification factor plotted as a function of temporal resolution for both the third- and fourth-order Runge–Kutta schemes. As shown in Fig. 2.8, once the time step ex-

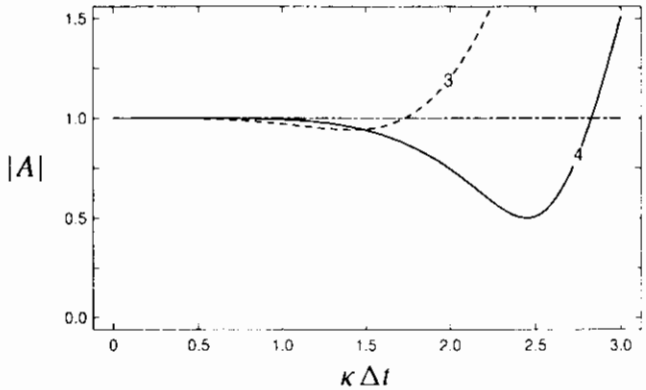

FIGURE 2.8. Modulus of the amplification factor plotted as a function of temporal resolution $\kappa \Delta t$ for higher-order Runge–Kutta solutions to the oscillation equation. Dashed line: third-order scheme; solid line: fourth-order method.

ceeds the maximum stable time step for the third-order scheme, the *fourth-order* method becomes highly damping. In some circumstances it may be desirable to selectively damp the highest-frequency modes, and in such cases the fourth-order Runge–Kutta method would appear to be much preferable to the first-order Matsuno method. On the other hand, if one wishes to avoid excessive damping of the the high-frequency components, it will not be possible to use the full stable time step of the fourth-order Runge–Kutta scheme, and as a result, the practical efficiency factor of the scheme will drop.

Tables 2.1 and 2.2 also list the order of accuracy of each scheme and give expressions for the relative phase change and amplitude error generated when each scheme is applied to the oscillation equation (2.30). The relationship between a scheme's order of accuracy and the orders of the amplitude and phase error is not entirely intuitive. As discussed in Section 2.3.1, the exact amplification factor for solutions to the oscillation equation is

$$A_e = e^{i\kappa k \Delta t} = 1 + i\kappa \Delta t - \frac{(\kappa \Delta t)^2}{2} - i\frac{(\kappa \Delta t)^3}{6} + \frac{(\kappa \Delta t)^4}{24} + \cdots.$$

The amplification factor A of an nth-order time-differencing scheme will match all terms in the preceding expression through order $(\kappa \Delta t)^n$. The amplitude error and the phase error characterize the errors in the modulus and the argument of A, respectively, and as such, their order of accuracy may differ from the general order of accuracy of the scheme. In particular, since amplitude and phase errors are special aspects of the total error, it is possible for either of these quantities to be smaller than the total error. The general relationship between the truncation error and the amplitude and phase errors may be stated as follows (Durran 1991):

If the oscillation equation (2.30) is integrated using a linear finite-difference scheme and if the truncation error of the resulting finite-difference approximation to the oscillation equation is of order r, then as $\kappa \Delta t \to 0$ the amplitude change in each step of the numerical solution is no worse than

$$1 + O[(\kappa \Delta t)^n], \quad \text{where} \quad \begin{cases} n = r + 1, & \text{if } r \text{ is odd;} \\ n \geq r + 2, & \text{if } r \text{ is even;} \end{cases}$$

and the relative phase change is no worse than

$$1 + O[(\kappa \Delta t)^m], \quad \text{where} \quad \begin{cases} m \geq r + 1, & \text{if } r \text{ is odd;} \\ m = r, & \text{if } r \text{ is even.} \end{cases}$$

Switching from an even- to an odd-order scheme increases the order of accuracy of the relative phase change without improving the order of accuracy of the amplitude error. Switching from odd to even order reduces the asymptotic amplitude error without altering the order of the error in the relative phase change.

2.4 Space-Differencing

Having examined the errors associated with time-differencing in Section 2.3, let us now consider the errors introduced when spatial derivatives are replaced with finite differences. In order to isolate the influence of the spatial differencing, the time dependence will not be discretized. Once again, our investigation will focus on the constant-wind-speed advection equation

$$\frac{\partial \psi}{\partial t} + c \frac{\partial \psi}{\partial x} = 0. \tag{2.59}$$

If the x-domain is periodic or unbounded, the spatial structure of the solution may be represented by a Fourier series or a Fourier integral, and a solution for each individual mode may be sought in the form of a traveling wave

$$\psi(x, t) = e^{i(kx - \omega t)}.$$

Here k is the *wave number*, and ω is the *frequency*. Substitution of this assumed solution into (2.59) shows that the traveling wave will satisfy the governing equation only if its frequency satisfies the *dispersion relation*

$$\omega = ck.$$

The wave travels with constant amplitude at a *phase speed* $\omega / k = c$. These waves are *nondispersive*, meaning that their phase speed is independent of the wave number. The energy associated with an isolated "packet" of waves propagates at the group velocity $\partial \omega / \partial k = c$, which is also independent of wave number. Readers unfamiliar with the concept of group velocity may wish to consult Gill (1982) or Whitham (1974).

2.4.1 *Differential–Difference Equations and Wave Dispersion*

Suppose that the spatial derivative in the advection equation is replaced with a second-order centered difference. Then (2.59) becomes the *differential–difference* equation:[6]

$$\frac{d\phi_j}{dt} + c\left(\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x}\right) = 0. \tag{2.60}$$

Individual wave-like solutions to this equation may be obtained in the form

$$\phi_j(t) = e^{i(kj\Delta x - \omega_{2c} t)}, \tag{2.61}$$

where ω_{2c} denotes the frequency associated with centered second-order spatial differencing. Substitution of (2.61) into the differential–difference equation yields

$$-i\omega_{2c}\phi_j = -c\left(\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2\Delta x}\right)\phi_j,$$

from which one obtains the dispersion relation

$$\omega_{2c} = c\frac{\sin k\Delta x}{\Delta x}. \tag{2.62}$$

Because ω_{2c} is real, there is no change in wave amplitude with time, and therefore no amplitude error. However, the phase speed,

$$c_{2c} \equiv \frac{\omega_{2c}}{k} = c\frac{\sin k\Delta x}{k\Delta x}, \tag{2.63}$$

is a function of k, so unlike the solutions to the original advection equation, these waves are dispersive. If the numerical resolution is good, $k\Delta x \ll 1$, and the Taylor series expansion $\sin x \approx x - x^3/6$ may be used to obtain

$$c_{2c} \approx c\left[1 - \tfrac{1}{6}(k\Delta x)^2\right],$$

showing that the phase-speed error is second-order in $k\Delta x$. Although the error for a well-resolved wave is small, the phase-speed error does become significant as the spatial resolution decreases. The least well-resolved wave on a numerical grid has wavelength $2\Delta x$ and wave number $k = \pi/\Delta x$. According to (2.63), *the phase speed of the* $2\Delta x$ *wave is zero.* Needless to say, this is a considerable error. The situation with the group velocity

$$\frac{\partial \omega_{2c}}{\partial k} = c\cos k\Delta x \tag{2.64}$$

is, however, even worse. The group velocity of well-resolved waves is approximately correct, but the group velocity of the poorly resolved waves is severely

[6] The set of differential–difference equations (2.60) for ϕ_j at every grid point constitute a large system of ordinary differential equations that could, in principle, be evaluated numerically using standard packages. This procedure, known as the *method of lines*, is usually not the most efficient approach.

retarded. The group velocity of the $2\Delta x$ wave is $-c$; its energy propagates backwards!

If the spatial derivative in the advection equation is replaced with a fourth-order centered difference, the resulting differential–difference equation

$$\frac{d\phi_j}{dt} + c\left[\frac{4}{3}\left(\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x}\right) - \frac{1}{3}\left(\frac{\phi_{j+2} - \phi_{j-2}}{4\Delta x}\right)\right] = 0 \tag{2.65}$$

has wave solutions of the form (2.61), provided that the frequency ω_{4c} satisfies the dispersion relation

$$\omega_{4c} = \frac{c}{\Delta x}\left(\frac{4}{3}\sin k\Delta x - \frac{1}{6}\sin 2k\Delta x\right).$$

As is the case for centered second-order differences, there is no amplitude error, only phase-speed error. Once again, the waves are dispersive, and the phase speed of the $2\Delta x$ wave is zero. The phase-speed error of a well-resolved wave is, however, reduced to $O\left[(k\Delta x)^4\right]$, since for $k\Delta x$ small,

$$c_{4c} = \frac{\omega_{4c}}{k} \approx c\left(1 - \frac{(k\Delta x)^4}{30}\right).$$

The group velocity

$$\frac{\partial \omega_{4c}}{\partial k} = c\left(\frac{4}{3}\cos k\Delta x - \frac{1}{3}\cos 2k\Delta x\right) \tag{2.66}$$

is also fourth-order accurate for well-resolved waves, but the group velocity of the $2\Delta x$ wave is $-5c/3$, an even greater error than that obtained using centered second-order differences.

The influence of spatial differencing on the frequency is illustrated in Fig. 2.9. As suggested by the preceding analysis, ω_{4c} approaches the true frequency more rapidly than ω_{2c} as $k\Delta x \to 0$, but both finite-difference schemes completely fail to capture the oscillation of $2\Delta x$ waves. The greatest advantages of the fourth-order difference over the second-order formulation are evident at "intermediate" wavelengths on the order of three to eight Δx. The improvements in the frequencies of these intermediate waves also generates a considerable improvement in their phase speeds and group velocities. The variation in the phase speed of a Fourier mode as a function of wave number is shown in Fig. 2.10. The improvement in the phase speed associated with an increase from second- to fourth-order accurate spatial differences is apparent even in the $3\Delta x$ wave. The fourth-order difference does not, however, improve the phase speed of the $2\Delta x$ wave. In fact, almost all finite-difference schemes fail to propagate the $2\Delta x$ wave. The basic problem is that there are only two possible configurations, differing by a phase angle of 180°, in which $2\Delta x$ waves can appear on a finite mesh.Thus, as shown

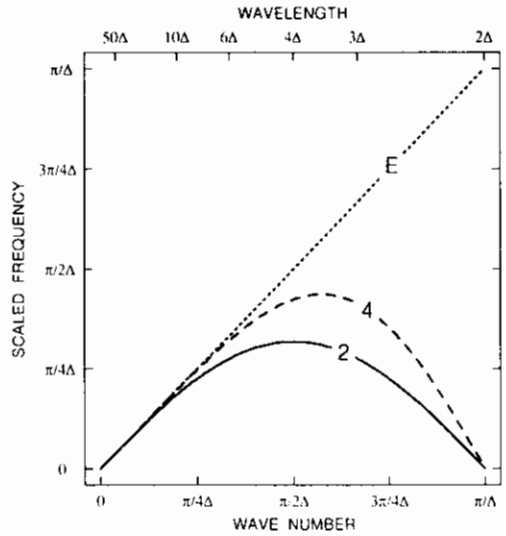

FIGURE 2.9. Scaled frequency (ω/c) as a function of wave number for the analytic solution of the advection equation (dotted line) and for corresponding differential–difference approximations using second- (solid line) and fourth-order (dashed line) centered differences.

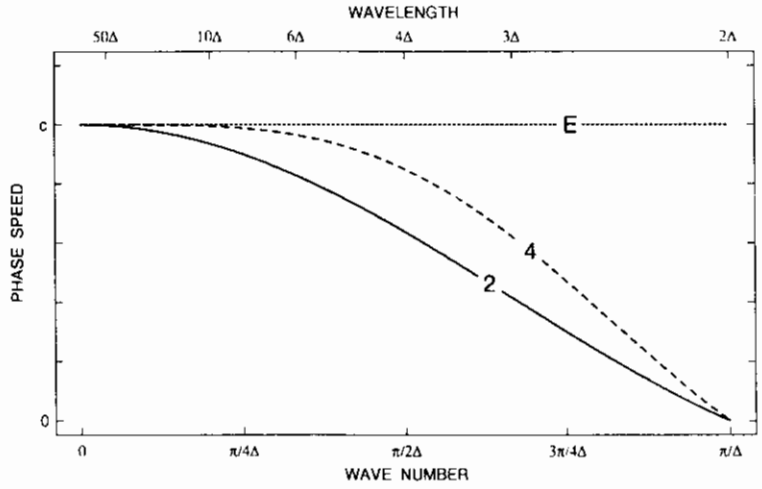

FIGURE 2.10. Phase speed as a function of numerical resolution for the analytic solution of the advection equation (dotted line) and for corresponding differential–difference approximations using second- (solid line) and fourth-order (dashed line) centered differences.

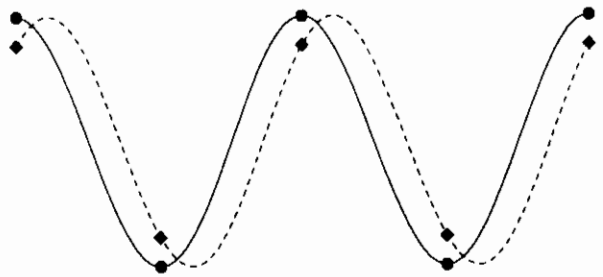

FIGURE 2.11. Misrepresentation of a $2\Delta x$ wave translating to the right as a decaying standing wave when the wave is sampled at fixed grid points on a numerical mesh. The grid-point values are indicated by dots at the earlier time and diamonds at the later time.

in Fig. 2.11, the grid-point representation of a translating $2\Delta x$ wave will be misinterpreted as a decaying standing wave.

The group velocities for the true solution and for the solutions of the second- and fourth-order differential–difference equations are plotted as a function of wave number in Fig. 2.12. The fourth-order scheme allows a better approximation of the group velocity for all but the shortest wavelengths. As discussed previously, the group velocity of the $2\Delta x$ wave produced by the fourth-order finite difference

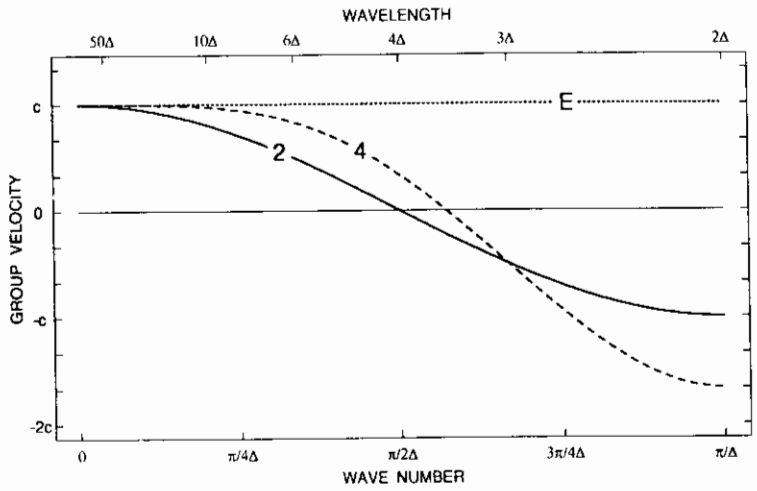

FIGURE 2.12. Group velocity as a function of numerical resolution for the analytic solution of the advection equation (dotted line) and corresponding differential–difference approximations using second- (solid line) and fourth-order (dashed line) centered differences.

is actually worse than that obtained with the second-order method. The degradation of the $2\Delta x$ group velocity in the higher-order scheme—or, equivalently, the increase in $\partial\omega/\partial k$ at $k = \pi/\Delta x$ in Fig. 2.9—is an unfortunate by-product of the inability of all finite-difference schemes to propagate the $2\Delta x$ wave and the otherwise desirable tendency of higher-order schemes to better approximate ω for wavelengths slightly longer than $2\Delta x$. In the absence of dissipation, the large negative group velocities associated with the $2\Delta x$ wave rapidly spread short-wavelength noise away from regions where $2\Delta x$ waves are forced.

One might attempt to improve the representation of extremely short waves by avoiding centered differences. If the spatial derivative in the advection equation is replaced with a first-order one-sided difference, (2.59) becomes

$$\frac{d\phi_j}{dt} + c\left(\frac{\phi_j - \phi_{j-1}}{\Delta x}\right) = 0. \tag{2.67}$$

Substitution of a wave solution of the form (2.61) into (2.67) yields the dispersion relation for the frequency associated with one-sided spatial differencing,

$$\omega_{1s} = \frac{c}{i\Delta x}\left(1 - e^{-ik\Delta x}\right) = \frac{c}{\Delta x}\left(\sin k\Delta x + i(\cos k\Delta x - 1)\right). \tag{2.68}$$

The real part of ω_{1s} is identical to the real part of ω_{2c}, and hence one-sided spatial differencing introduces the same dispersion error as centered second-order spatial differencing. Unlike centered differencing, however, the one-sided difference also generates amplitude error through the imaginary part of ω_{1s}. The amplitude of the differential–difference solution will grow or decay at the rate

$$\exp\left(-\frac{c}{\Delta x}(1 - \cos k\Delta x)t\right).$$

Thus, poorly resolved waves change amplitude most rapidly. If $c > 0$, the solution damps; the solution amplifies when $c < 0$. Note that if $c < 0$, the numerical domain of dependence does not include the domain of dependence of the original partial differential equation, so instability could also be predicted from the CFL condition.

A comparison of the performance of first-order, second-order, and fourth-order spatial differencing is provided in Fig. 2.13, which shows analytic solutions to the advection equation and numerical solutions to the corresponding differential–difference problem. The differential–difference equations are solved numerically on a periodic spatial domain using a fourth-order Runge–Kutta scheme to integrate (2.60), (2.65), and (2.67) with a very small time step.

Fig. 2.13a shows the distribution of ϕ that develops when the initial condition is a narrow spike, such that $\phi_j(t = 0)$ is zero everywhere except at the midpoint of the domain. Although the numerical domain is periodic, large-amplitude perturbations have not reached the lateral boundaries at the time shown in Fig. 2.13a. The narrow initial spike is formed by the superposition of many waves of different wavelengths; however, the Fourier components with largest amplitude are all of very short wavelength. The large diffusive error generated by one-sided differencing rapidly damps these short wavelengths and reduces the spike to a highly

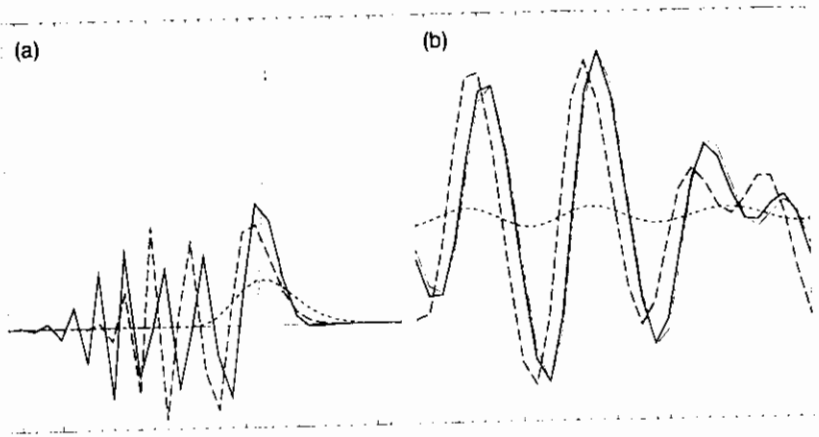

FIGURE 2.13. Exact solution and differential-difference solutions for (a) advection of a spike over a distance of five grid points, and (b) advection of the sum of equal-amplitude $7.5\Delta x$ and $10\Delta x$ sine waves over a distance of twelve grid points. Exact solution (dot-dashed), one-sided first-order (short-dashed), centered second-order (long-dashed), and centered fourth-order (solid). The distribution is translating to the right. Grid-point locations are indicated by the tick marks at the top and bottom of the plot.

smoothed low-amplitude disturbance. The second- and fourth-order centered differences also produce a dramatic distortion in the amplitude of the solution. Although the centered schemes preserve the amplitude of each individual Fourier component, the various components propagate at different speeds, and thus the superposition of these components ceases to properly represent the true solution. Consistent with the values of the group velocity given by (2.64) and (2.66), the energy in the shortest waves propagates back upstream from the initial location of the spike. As predicted by theory, the upstream propagation of the $2\Delta x$ wave is most rapid for the fourth-order method. Switching to a higher-order scheme does not improve the performance of finite-difference methods when they are used to model poorly resolved features like the spike in Fig. 2.13a; in fact, in many respects the fourth-order solution is worse than the second-order result.

The spike test is an extreme example of a common problem for which many numerical schemes are poorly suited, namely, the task of properly representing solutions with near discontinuities. As such the spike test provides a reference point that characterizes a scheme's ability to properly model poorly resolved waves. A second important reference point is provided by the test in Fig. 2.13b, which examines each scheme's ability to approximate features at an intermediate numerical resolution. The solution in Fig.2.13b is the sum of equal-amplitude $7.5\Delta x$ and $10\Delta x$ waves; in all other respects the problem is identical to that in Fig.2.13a. Unlike the situation with the spike test, the higher-order schemes are clearly superior in their treatment of the waves in Fig. 2.13b. Whereas the first-order difference generates substantial amplitude error and is distinctly inferior to the other two

schemes, the second-order difference produces a reasonable approximation to the correct solution. Second-order centered differencing does, however, generate a noticeable lag in the phase speed of the disturbance (as in (2.63)). Moreover, since the phase lag of the $7.5\Delta x$ wave differs from that of the $10\Delta x$ wave in the second-order solution, the relative phase of the two waves changes during the simulation, and a significant error develops in the amplitude of the two rightmost wave crests. This example serves to emphasize that although centered differences do not produce amplitude errors in individual Fourier components, they still generate amplitude errors in the total solution. Finally, in contrast to the first- and second-order schemes, the errors introduced by fourth-order differencing are barely detectable at this time in the simulation.

The damping associated with the first-order upstream scheme (2.67) can be significantly reduced by using a higher-order one-sided difference. The differential–difference equation

$$\frac{d\phi_j}{dt} + \frac{c}{6}\left(\frac{2\phi_{j+1} + 3\phi_j - 6\phi_{j-1} + \phi_{j-2}}{\Delta x}\right) = 0 \tag{2.69}$$

may be obtained by replacing the spatial derivative in the advection equation with a third-order difference. The dispersion relation associated with this differential–difference equation is

$$\omega_{3s} = \frac{c}{\Delta x}\left[\left(\frac{4}{3}\sin k\Delta x - \frac{1}{6}\sin 2k\Delta x\right) - \frac{i}{3}(1 - \cos k\Delta x)^2\right]. \tag{2.70}$$

The real part of ω_{3s} is identical to that of ω_{4c}, and the phase-speed errors associated with the third- and fourth-order schemes are therefore identical. As was the case with first-order one-sided differencing, the sign of the imaginary part of ω_{3s} is determined by the sign of c such that solutions amplify for $c < 0$ and damp for $c > 0$. The damping associated with the third-order scheme is considerably less than that of the first-order scheme. According to (2.68) and (2.70),

$$\frac{\Im(\omega_{3s})}{\Im(\omega_{1s})} = \frac{1}{3}(1 - \cos k\Delta x).$$

As might be expected with a higher-order scheme, the well-resolved waves are damped much more slowly by the third-order approximation. Even the short waves show substantial improvement.

Some idea of the relative performance of the first-, second-, and third-order differences is provided in Fig. 2.14, which is identical to Fig. 2.13, except that the solid curve now represents the third-order solution. As indicated in Fig. 2.14, the damping produced by the third-order scheme is much weaker than that generated by first-order upstream differencing. Moreover, the third-order solution to the spike test is actually better than the second- and fourth-order results (compare Figs. 2.13a and 2.14a). In problems with extremely poor resolution, such as the spike test, the tendency of the third-order scheme to damp short wavelengths can

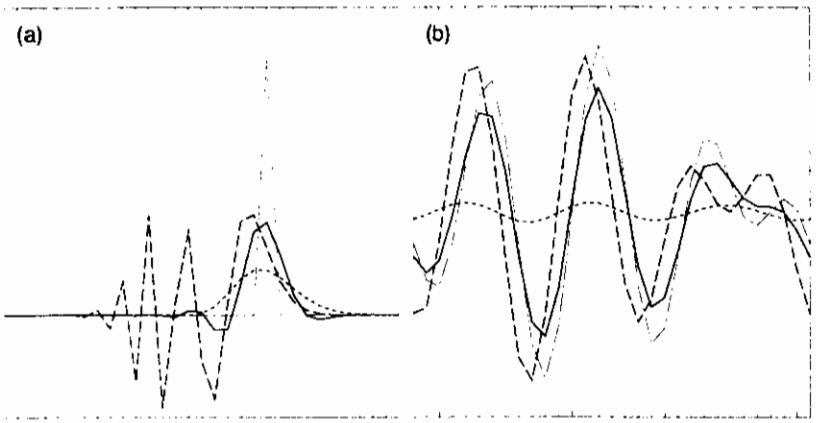

FIGURE 2.14. Exact solution and differential-difference solutions for (a) advection of a spike over a distance of five grid points, and (b) advection of the sum of equal-amplitude $7.5\Delta x$ and $10\Delta x$ sine waves over a distance of twelve grid points. Exact solution (dot-dashed), one-sided first-order (short-dashed), centered second-order (long-dashed), and one-sided third-order (solid).

be beneficial, because it largely eliminates the dispersive trail of waves found in the centered difference solutions. On the other hand, the damping of intermediate wavelengths is sufficiently weak that the third-order solution retains almost the same amplitude in the region of the spike as the "nondamping" second- and fourth-order schemes. The situation in Fig. 2.14b is somewhat different, and it is not entirely obvious whether the third-order results should be preferred over the second-order scheme. The third-order scheme clearly exhibits less phase-speed error, but it also shows more amplitude error than the centered second-order method.

2.4.2 *Dissipation, Dispersion, and the Modified Equation*

One way to estimate phase-speed and amplitude error is to derive the differential-difference dispersion relation, as described in the preceding section. Another way to characterize the relative magnitude of the these errors is to examine the lowest-order terms in the truncation error of the finite-difference formula. The truncation errors for each of the finite-difference approximations considered in the preceding section are as follows. One-sided, first-order:

$$\frac{\psi_j - \psi_{j-1}}{\Delta x} = \frac{\partial \psi}{\partial x} - \frac{\Delta x}{2}\frac{\partial^2 \psi}{\partial x^2} + \frac{\Delta x^2}{6}\frac{\partial^3 \psi}{\partial x^3} + O\left[(\Delta x)^3\right]. \tag{2.71}$$

Centered, second-order:

$$\frac{\psi_{j+1} - \psi_{j-1}}{2\Delta x} = \frac{\partial \psi}{\partial x} + \frac{\Delta x^2}{6}\frac{\partial^3 \psi}{\partial x^3} + O\left[(\Delta x)^4\right].$$

One-sided, third-order:

$$\frac{2\psi_{j+1} + 3\psi_j - 6\psi_{j-1} + \psi_{j-2}}{6\Delta x} = \frac{\partial \psi}{\partial x} + \frac{\Delta x^3}{12}\frac{\partial^4 \psi}{\partial x^4} - \frac{\Delta x^4}{30}\frac{\partial^5 \psi}{\partial x^5} + O\left[(\Delta x)^5\right].$$

Centered, fourth-order:

$$\frac{4}{3}\left(\frac{\psi_{j+1} - \psi_{j-1}}{2\Delta x}\right) - \frac{1}{3}\left(\frac{\psi_{j+2} - \psi_{j-2}}{4\Delta x}\right) = \frac{\partial \psi}{\partial x} - \frac{\Delta x^4}{30}\frac{\partial^5 \psi}{\partial x^5} + O\left[(\Delta x)^6\right].$$

If one of these formulae is used to determine the truncation error in a differential–difference approximation to the advection equation and the resulting scheme is $O\left[(\Delta x)^m\right]$ accurate, the same differential–difference scheme will approximate the *modified equation*

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = a(\Delta x)^m \frac{\partial^{m+1} \psi}{\partial x^{m+1}} + b(\Delta x)^{m+1}\frac{\partial^{m+2} \psi}{\partial x^{m+2}} \tag{2.72}$$

to $O\left[(\Delta x)^{m+2}\right]$, where a and b are rational numbers determined by the particular finite-difference formula. Thus, as $\Delta x \to 0$, the numerical solution to the differential–difference equation will approach the solution to the modified equation more rapidly than it approaches the solution to the advection equation. A qualitative description of the effects of the leading-order errors in the differential–difference equation may therefore be obtained by examining the prototypical response generated by each of the forcing terms on the right side of the modified equation (2.72).

The term with the even-order derivative in (2.72) introduces a forcing identical to that in the prototypical equation

$$\frac{\partial \xi}{\partial t} = (-1)^{m+1}\frac{\partial^{2m} \xi}{\partial x^{2m}},$$

whose solutions

$$\xi(x, t) = Ce^{ikx}e^{-k^{2m}t}$$

become smoother with time because the shorter-wavelength modes decay more rapidly than the longer modes. Thus, the term with the lowest-order even derivative produces amplitude error, or *numerical dissipation*, in the approximate solution of the advection equation. The odd-order derivative on the right side of (2.72) introduces a forcing identical to that in the prototypical equation

$$\frac{\partial \xi}{\partial t} = -\frac{\partial^{2m+1} \xi}{\partial x^{2m+1}},$$

whose solutions are waves of the form

$$\xi(x,t) = Ce^{i(kx-\omega t)}, \quad \text{where} \quad \omega = (-1)^m k^{2m+1}.$$

For $m > 0$, these waves are dispersive, because their phase speed ω/k depends on the wave number k. As a consequence, the lowest-order odd derivative on the right side of (2.72) produces a wave-number-dependent phase speed error known as *numerical dispersion.*

Centered spatial differences do not produce numerical dissipation because there are no even derivatives in the truncation error of a centered difference scheme. Numerical dissipation is, however, produced by the leading-order term in the truncation error of the one-sided differences. There is a pronounced qualitative difference between the solutions generated by schemes with leading-order dissipative and leading-order dispersive errors. The modified equations associated with the preceding first- and second-order spatial differences both include identical terms in $\partial^3\psi/\partial x^3$. As a consequence, both schemes produce essentially the same dispersive error. The dispersion errors in the third- and fourth-order schemes are also very similar because the truncation error associated with each of these schemes includes identical terms in $\partial^5\psi/\partial x^5$. Yet, as was illustrated in Figs. 2.13 and 2.14, the impact of dispersion on even- and odd-order schemes is very different. Numerical dispersion is the only error in the centered even-order differences, so when short-wavelength modes are present, the dispersion is quite evident. In contrast, the numerical dispersion generated by the one-sided odd-order schemes is largely obscured by the lower-order dissipative errors that dominate the total error in these schemes.

2.4.3 *Artificial Dissipation*

As suggested by the test problems shown in Figs. 2.13 and 2.14, the lack of dissipation in centered-spatial differences can sometimes be a disadvantage. In particular, the error produced by the dispersion of poorly resolved Fourier components is free to propagate throughout the solution without loss of amplitude. It is therefore often useful to add scale-selective dissipation to otherwise nondissipative schemes in order to damp the shortest resolvable wavelengths. Moreover, in nonlinear problems it is often necessary to remove energy from the shortest spatial scales to prevent the development of numerical instabilities that can arise through the nonlinear interaction of short-wavelength modes (see Section 3.6).

The centered finite-difference approximations to even spatial derivatives of order two or higher provide potential formulae for scale-selective smoothers. Consider the isolated effect of a second-derivative smoother in an equation of the form

$$\frac{d\phi_j}{dt} = \gamma_2\left(\phi_{j+1} - 2\phi_j + \phi_{j-1}\right), \tag{2.73}$$

where γ_2 is a parameter that determines the strength of the smoother. Substitution of solutions of the form

$$\phi_j = A(t)e^{ikj\Delta x} \tag{2.74}$$

into (2.73) yields

$$\frac{dA}{dt} = -2\gamma_2(1 - \cos k\Delta x)A,$$

implying that $2\Delta x$ waves are damped most rapidly, and that well-resolved waves undergo an $O[(k\Delta x)^2]$ dissipation. Indeed, if the second-derivative smoother is combined with the standard second-order centered difference,[7] the total truncation error in the smoothed difference becomes

$$\frac{\psi_{j+1} - \psi_{j-1}}{2\Delta x} - \gamma_2\left(\psi_{j+1} - 2\psi_j + \psi_{j-1}\right)$$

$$= \frac{\partial\psi}{\partial x} - (\Delta x)^2\left(\gamma_2\frac{\partial^2\psi}{\partial x^2} - \frac{1}{6}\frac{\partial^3\psi}{\partial x^3}\right) + O\left[(\Delta x)^4\right].$$

Thus, the smoothed difference remains of second order, but the leading-order truncation error becomes both dissipative and dispersive. Note that as $\Delta x \to 0$, the preceding scheme will generate less dissipation than one-sided differencing, because as indicated by (2.71), one-sided differencing produces $O(\Delta x)$ dissipation. Furthermore, the addition of a separate smoother allows the dissipation rate to be explicitly controlled through the specification of γ_2.

Greater scale selectivity can be obtained using a fourth-derivative filter of the form

$$\frac{d\phi_j}{dt} = \gamma_4\left(-\phi_{j+2} + 4\phi_{j+1} - 6\phi_j + 4\phi_{j-1} - \phi_{j-2}\right), \tag{2.75}$$

or the sixth-derivative filter

$$\frac{d\phi_j}{dt} = \gamma_6\left(\phi_{j+3} - 6\phi_{j+2} + 15\phi_{j+1} - 20\phi_j + 15\phi_{j-1} - 6\phi_{j-2} + \phi_{j-3}\right). \tag{2.76}$$

Substituting a single wave of the form (2.74) into any of the preceding smoothers (2.73), (2.75), or (2.76) yields

$$\frac{dA}{dt} = -\gamma_n\left[2(1 - \cos k\Delta x)\right]^{n/2} A, \tag{2.77}$$

where $n = 2$, 4, or 6 is the order of the derivative in each of the respective smoothers. In all cases, the $2\Delta x$ wave is damped most rapidly, and long waves are relatively unaffected. The actual scale selectivity of these filters is determined by the factor $(1 - \cos k\Delta x)^{n/2}$, which for well-resolved waves is $O\left[(k\Delta x)^n\right]$. This scale selectivity is illustrated in Fig. 2.15, in which the exponential decay rate as-

[7] If a dissipative filter is used in conjunction with leapfrog time-differencing, the terms involved in the filtering calculation must be evaluated at the $t - \Delta t$ time level to preserve stability. Time-differencing schemes appropriate for the simulation of diffusive processes are examined in Section 3.4.

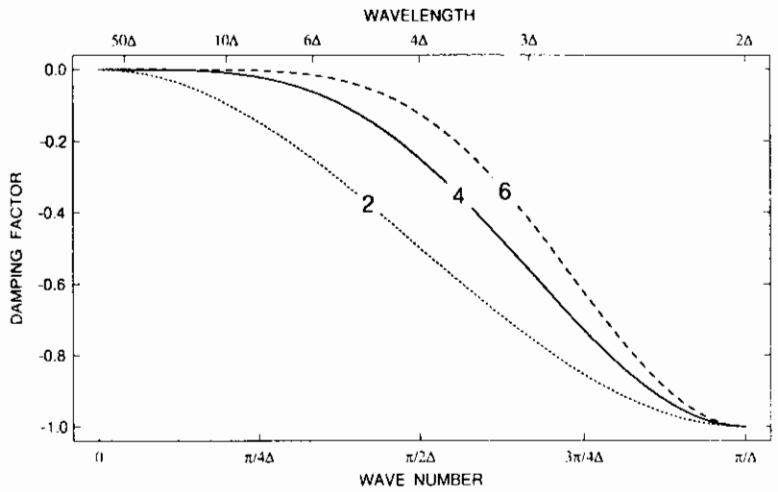

FIGURE 2.15. Normalized damping rate as a function of horizontal wave number for second- (dotted line), fourth- (solid line), and sixth-order (dashed-line) diffusive filters.

sociated with each smoother is plotted as a function of wave number. In order to facilitate the comparison of these filters, the decay rate of the $2\Delta x$ wave has been normalized to unity by choosing $\gamma_n = 2^{-n}$.

The test problems shown in Figs. 2.13 and 2.14 were repeated using fourth-order centered differencing in combination with fourth-order and sixth-order spatial smoothers and the results plotted in Fig. 2.16. The filtering coefficients were set such that $\gamma_4 = 0.2$, and $\gamma_6 = \gamma_4/4$; this choice for γ_6 insures that both filters will damp a $2\Delta x$ wave at the same rate. As evident in a comparison of Figs. 2.13a and 2.16a, both the fourth- and the sixth-order filters remove much of the dispersive train of short waves that were previously present behind the isolated spike in the unfiltered solution. Those waves that remain behind the spike in the smoothed solutions have wavelengths near $4\Delta x$. Since γ_4 and γ_6 have been chosen to damp $2\Delta x$ waves at the same rate, the $4\Delta x$ waves in the dispersive train are not damped as rapidly by the sixth-order smoother, and as is evident in Fig. 2.16a, the sixth-order smoother leaves more amplitude in the wave train behind the spike. Although the scale selectivity of the sixth-order smoother interferes with the damping of the dispersive wave train behind the spike, it significantly improves the simulation of the moderately resolved waves shown in Fig. 2.16b. The solution obtained using the sixth-order filter is almost perfect, whereas the fourth-order filter generates significant damping. In fact, the general character of the solution obtained with the fourth-order filter is reminiscent of that obtained with the third-order one-sided finite-difference approximation. This similarity is not coincidental; the phase-speed errors produced by the third- and fourth-order finite differences are identical, and the leading-order numerical dissipation in the third-

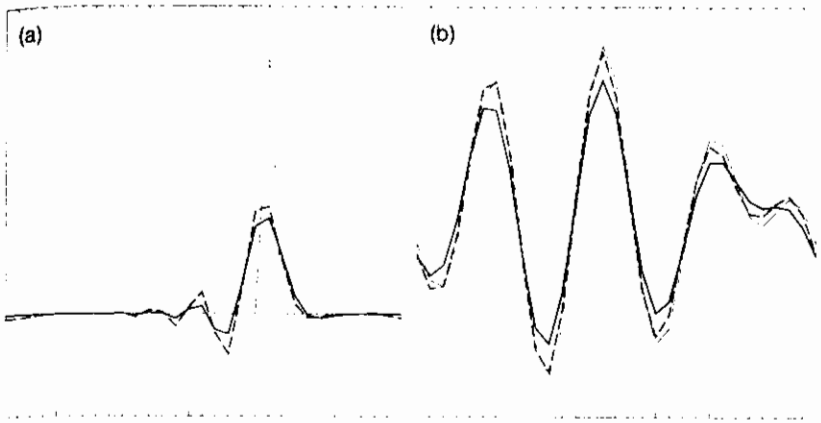

FIGURE 2.16. Exact solution and differential–difference solutions for (a) advection of a spike over a distance of five grid points, and (b) advection of the sum of equal-amplitude $7.5\Delta x$ and $10\Delta x$ sine waves over a distance of twelve grid points. Exact solution (dot-dashed), and fourth-order centered difference solutions in combination with a fourth-derivative filter (solid) or a sixth-derivative filter (dashed).

order difference, which is proportional to the fourth derivative, has the same scale selectivity as the fourth-order smoother.

Indeed, the proper choice of γ_4 will produce an exact equivalence between the solution obtained with the third-order scheme and the result produced by the combination of a fourth-order centered difference and a fourth-order smoother. The third-order differential–difference equation (2.69) can be expressed in a form that remains upstream independent of the sign of c as

$$\frac{d\phi_j}{dt} + \frac{c}{12\Delta x}\left(-\phi_{j+2} + 8(\phi_{j+1} - \phi_{j-1}) + \phi_{j-2}\right) = -\frac{|c|}{12\Delta x}\left(\phi_{j+2} - 4\phi_{j+1} + 6\phi_j - 4\phi_{j-1} + \phi_{j-2}\right), \quad (2.78)$$

which is the combination of a fourth-order centered spatial difference and a fourth-order filter with a filter coefficient $\gamma_4 = |c|/(12\Delta x)$. Note that the value of the fourth-derivative filter in the preceding is an inverse function of Δx. The implicit Δx-dependence of the filtering coefficient in (2.78) makes the scheme $O\left[(\Delta x)^3\right]$, whereas the dissipation introduced by the explicit fourth-order filter (2.75) is $O\left[(\Delta x)^4\right]$.

In practical applications, the time derivatives in (2.73) and (2.75) are replaced by finite differences, and the maximum values for γ_2 and γ_4 will be determined by stability considerations. If the differencing is forward in time, the maximum useful smoothing coefficients are determined by the relations $\gamma_2 \Delta t \leq 0.25$ and

$\gamma_4 \Delta t \leq 0.0625$. When $\gamma_2 \Delta t = 0.25$ (or $\gamma_4 \Delta t = 0.0625$) any $2\Delta x$ wave will be completely removed by a single application of the second-order (or fourth-order) filter.

2.4.4 *Compact Differencing*

Further improvements in the filtered solutions shown in Fig. 2.16 can be obtained by using more accurate finite-difference schemes. Simply switching to a higher-order explicit scheme, such as the centered sixth-order difference

$$\frac{df}{dx} = \frac{3}{2}\delta_{2x} f - \frac{3}{5}\delta_{4x} f + \frac{1}{10}\delta_{6x} f + O\left[(\Delta x)^6\right] \tag{2.79}$$

(where the operator δ_{nx} is defined by (2.7)), provides only marginal improvement. More significant improvements can be obtained using *compact differencing*, in which the desired derivative is given implicitly by a matrix equation. Our attention will be restricted to compact schemes in which this implicit coupling leads to tridiagonal matrices, since tridiagonal systems can be evaluated with modest computational effort (see Appendix).

The simplest compact scheme is obtained by rewriting the expression for the truncation error in the centered second-order difference (2.8) in the form

$$\delta_{2x} f = \left(1 + \frac{(\Delta x)^2}{6}\delta_x^2\right)\frac{df}{dx} + O\left[(\Delta x)^4\right]. \tag{2.80}$$

Expanding the finite-difference operators in the preceding expression yields the following $O\left[(\Delta x)^4\right]$ accurate expression for the derivative:

$$\frac{f_{j+1} - f_{j-1}}{2\Delta x} = \frac{1}{6}\left[\left(\frac{df}{dx}\right)_{j+1} + 4\left(\frac{df}{dx}\right)_j + \left(\frac{df}{dx}\right)_{j-1}\right]. \tag{2.81}$$

This scheme allows fourth-order-accurate derivatives to be calculated on a three-point stencil. At intermediate numerical resolution, the fourth-order compact scheme is typically more accurate than the sixth-order explicit difference (2.79).

If one is going to the trouble to solve a tridiagonal matrix, it can be advantageous to do a little extra work and use the sixth-order tridiagonal scheme. The formula for the sixth-order tridiagonal compact scheme may be derived by first noting that the truncation error in the fourth-order explicit scheme (2.9) is

$$\left(1 - \frac{(\Delta x)^2}{6}\delta_x^2\right)\delta_{2x} f = \frac{df}{dx} - \frac{(\Delta x)^4}{30}\frac{d^5 f}{dx^5} + O\left[(\Delta x)^6\right],$$

and the truncation error in the fourth-order compact scheme (2.80) is

$$\delta_{2x} f = \left(1 + \frac{(\Delta x)^2}{6}\delta_x^2\right)\frac{df}{dx} - \frac{(\Delta x)^4}{180}\frac{d^5 f}{dx^5} + O\left[(\Delta x)^6\right].$$

Eliminating the $O\left[(\Delta x)^4\right]$ term between these two expressions, one obtains

$$\left(1 + \frac{(\Delta x)^2}{30}\delta_x^2\right)\delta_{2x} f = \left(1 + \frac{(\Delta x)^2}{5}\delta_x^2\right)\frac{df}{dx} + O\left[(\Delta x)^6\right].$$

Expanding the operators in the preceding yields the following $O\left[(\Delta x)^6\right]$-accurate tridiagonal system for df/dx:

$$\frac{1}{15}\left(14\delta_{2x} f_j + \delta_{4x} f_j\right) = \frac{1}{5}\left[\left(\frac{df}{dx}\right)_{j+1} + 3\left(\frac{df}{dx}\right)_j + \left(\frac{df}{dx}\right)_{j-1}\right]. \tag{2.82}$$

When compact schemes are used to approximate partial derivatives in complex equations in which one must compute several different spatial derivatives, such as the multidimensional advection equation, it is simplest to solve either (2.81) or (2.82) as a separate tridiagonal system for each derivative. However, in very simple problems, such as the one-dimensional advection equation (2.59), the spatial derivatives in the compact formulae may be replaced directly by $-(1/c)\partial\psi/\partial t$. Thus, in order to analyze the phase-speed error associated with compact spatial differencing, the fourth-order compact approximation to the advection equation may be written

$$\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x} = \frac{-1}{6c}\left[\left(\frac{\partial\phi}{\partial t}\right)_{j+1} + 4\left(\frac{\partial\phi}{\partial t}\right)_j + \left(\frac{\partial\phi}{\partial t}\right)_{j-1}\right]. \tag{2.83}$$

Substitution of a wave solution of the form (2.61) into the preceding yields the following expression for the phase speed of the differential–difference solution:

$$c_{4c} = \frac{\omega_{4p}}{k} = \frac{3c}{2 + \cos k\Delta x}\left(\frac{\sin k\Delta x}{k\Delta x}\right). \tag{2.84}$$

The phase speeds for the sixth-order compact scheme,

$$c_{6c} = \frac{c}{3(3 + 2\cos k\Delta x)}\left(14\frac{\sin k\Delta x}{k\Delta x} + \frac{\sin 2k\Delta x}{2k\Delta x}\right),$$

may be obtained through a similar derivation. These phase speeds are plotted as a function of $k\Delta x$, together with the curves for second-, fourth-, and sixth-order explicit centered differences, in Fig. 2.17. It is apparent that the compact schemes are superior to the explicit schemes. In particular, the phase speeds associated with the sixth-order compact differencing are almost perfect for wavelengths as short as $4\Delta x$. Note that although the order of accuracy of a scheme determines the rate at which the phase-speed curves in Fig. 2.17 asymptotically approach the correct value as $k\Delta x \to 0$, the order of accuracy does not reliably predict a scheme's ability to represent the poorly resolved waves. Lele (1992) observed that a better treatment of the shorter waves can be obtained by perturbing the coefficients in

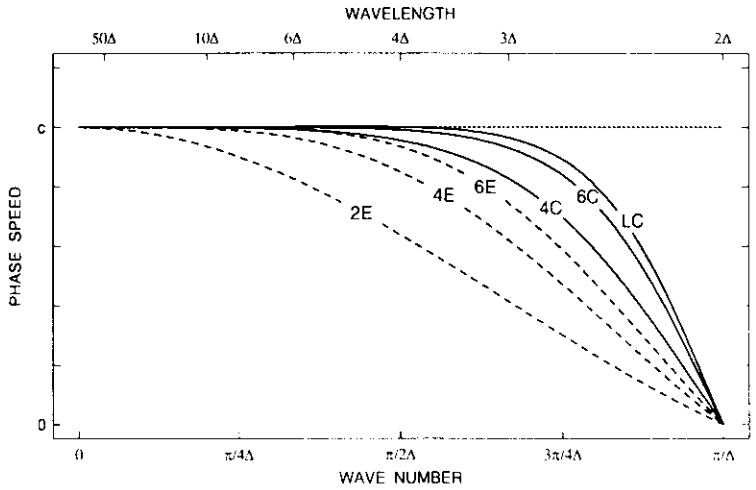

FIGURE 2.17. Phase speed as a function of numerical resolution for the analytic solution of the advection equation (dotted line) and for corresponding differential–difference approximations using second-, fourth-, and sixth-order explicit differences (dashed lines), fourth- and sixth-order compact differences (solid lines), and the low-phase-speed error fourth-order compact scheme of Lele (solid line labeled "LC").

the sixth-order compact scheme to create the fourth-order method

$$\frac{1}{12}\left(11\delta_{2x}f_j + \delta_{4x}f_j\right) = \frac{1}{24}\left[5\left(\frac{df}{dx}\right)_{j+1} + 14\left(\frac{df}{dx}\right)_j + 5\left(\frac{df}{dx}\right)_{j-1}\right]. \tag{2.85}$$

The phase speeds associated with this differencing scheme are plotted as the solid curve labeled "LC"in Fig. 2.17. Observe that Lele's compact scheme produces phase-speed errors in a $3\Delta x$ wave that are comparable to the errors introduced in a $6\Delta x$ wave by explicit fourth-order differences.

The performance of Lele's compact scheme on the test problems considered previously in connection with Figs. 2.13, 2.14, and 2.16 is illustrated in Fig. 2.18. Since they accurately capture the frequency of very short waves while still failing to detect any oscillations at $2\Delta x$, compact schemes propagate the energy in the $2\Delta x$ wave backwards at very large group velocities (i.e., $-\partial\omega/\partial k$ is large near $k = 2\Delta x$). The preceding compact schemes are also nondamping because they are centered in space. It is therefore necessary to use a spatial filter in conjunction with these schemes when modeling problems with significant short-wavelength features. In these tests, a sixth-order filter (2.76) was used in combination with both the compact scheme (2.85) and the fourth-order explicit method. In all cases $\gamma_6 = 0.05$, which is the same value used in the computations shown in Fig. 2.16.

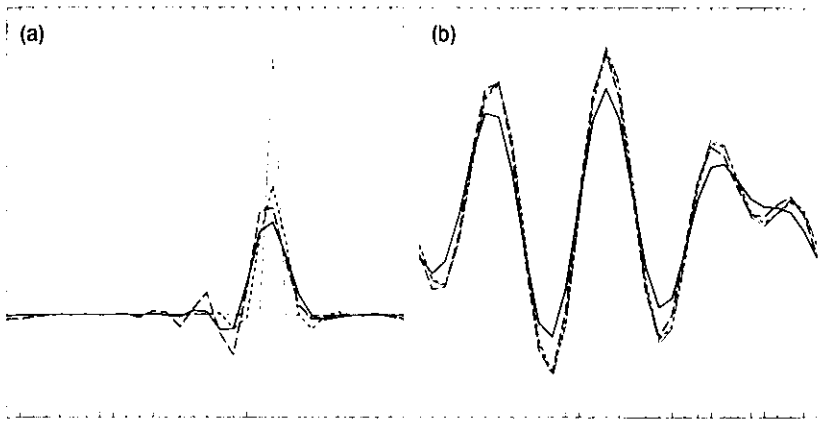

FIGURE 2.18. Exact solution and differential–difference solutions for (a) advection of a spike over a distance of five grid points, and (b) advection of the sum of equal-amplitude $7.5\Delta x$ and $10\Delta x$ sine waves over a distance of twelve grid points. Exact solution (dot-dashed), third-order one-sided solution (solid), fourth-order centered explicit solution (long dashed), and the solution obtained using Lele's low phase-speed-error compact scheme (short dashed). A sixth-order smoother, with $\gamma_6 = .05$ was used in combination with the fourth- and sixth-order differences.

In fact, the fourth-order solutions shown in these tests are identical to those shown previously in Fig. 2.16. Also plotted in Fig. 2.18 are the exact solution and the third-order one-sided solution (previously plotted in Fig. 2.14). As evident in Fig. 2.18a, the smoothed sixth-order compact scheme exhibits less of a $4\Delta x$ dispersive trail than either the third- or fourth-order scheme. Since the dissipation applied to the compact solution is identical to that used with the fourth-order scheme (and less than that inherent in the third-order method), the relative absence of dispersive ripples in compact solution indicates a relative lack of dispersive error at the $4\Delta x$ wavelength. This, of course, is completely consistent with the theoretical phase speed analysis shown in Fig. 2.17. The compact scheme also performs best on the two-wave test, Fig. 2.18b. Although the filtered compact scheme is the best-performing method considered in this section, it is also the most computationally burdensome. Other approaches to the problem of creating methods that can adequately represent short-wavelength features without sacrificing accuracy in smoother parts of the flow will be discussed in connection with the concept of flux-corrected transport in Chapter 5.

2.5 Combined Time- and Space-Differencing

The error introduced by time-differencing in ordinary differential equations was examined in Section 2.3. In Section 2.4, the error generated by spatial differencing was isolated and investigated through the use of differential–difference equations.

We now consider finite-difference approximations to the complete partial differential equation and analyze the total error that arises from the combined effects of both temporal and spatial differencing.

In some instances, the fundamental behavior of a scheme can be deduced from the characteristics of its constituent spatial and temporal differences. For example, suppose that the advection equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0 \tag{2.86}$$

is approximated using forward time-differencing in combination with centered spatial differencing. The result should be amplifying because forward time-differencing is amplifying and centered spatial differencing is neutral. Their combined effect will therefore produce amplification. On the other hand, it might be possible to combine forward time-differencing with one-sided space-differencing because the one-sided spatial difference is damping—provided that it is computed using "upstream" data. If this damping dominates the amplification generated by the forward time difference, it will stabilize the scheme. Further analysis would be required to determine the actual stability condition and the phase-speed error.

As another example, consider the use of leapfrog time-differencing and centered spatial differencing to approximate the advection equation. Since both differences are neutral, it seems likely that such a scheme would be conditionally stable. Once again, further analysis is required to determine the exact stability condition and the phase-speed error. In the absence of such analysis, the sign of the phase-speed error is in doubt, since the leapfrog scheme is accelerating, whereas centered spatial differencing is decelerating. Finally, suppose that leapfrog differencing is combined with one-sided spatial differences. The result should be unstable because the leapfrog solution consists of two modes (the physical and computational modes) each propagating in the opposite direction. If the one-sided difference is "upstream" with respect to one mode, it will be "downstream" with respect to the second mode, thereby amplifying the second mode.

Although as just noted, the forward-time and centered-space scheme

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + c\frac{\phi_{j+1}^n - \phi_{j-1}^n}{2\Delta x} = 0 \tag{2.87}$$

will produce a nonphysical amplification of the approximate solution to the advection problem, one might wonder whether this amplification is sufficiently weak that the scheme nevertheless satisfies the more general Von Neumann stability condition

$$|A_k| \le 1 + \gamma \Delta t, \tag{2.88}$$

where γ is a constant independent of k, Δt, and Δx. If so, then (2.87) will still generate convergent approximations to the correct solution in the limit $\Delta x \to 0$, $\Delta t \to 0$, because it is a consistent approximation to the advection equation. Recall that as discussed in Section 2.3.2, forward differencing produces amplifying solutions that nevertheless converge to the correct solution of the oscillation equation as $\Delta t \to 0$. The amplification factor arising from a Von Neumann stability analysis of (2.87) satisfies

$$|A_k|^2 = 1 + \left(\frac{c\sin(k\Delta x)}{\Delta x}\right)^2 (\Delta t)^2.$$

Here, in contrast to the results obtained if ordinary differential equations are approximated with a forward difference, the coefficient of Δt includes a factor of $(\Delta x)^{-2}$ that cannot be bounded by a constant independent of Δx as $\Delta x \to 0$. As a consequence, the forward-time centered-space scheme does not satisfy the Von Neumann condition (2.88) and is both unstable in the sense that it generates growing solutions to a problem where the true solution is bounded, and unstable in the more general sense that it does not produce convergent solutions as $\Delta x \to 0$ and $\Delta t \to 0$. Note in particular that after N time steps the amplitude of a $4\Delta x$ wave increases by a factor of $(1 + \mu^2)^{N/2}$ (where $\mu = c\Delta t/\Delta x$). Thus, if a series of integrations are performed in which the space–time grid is refined while holding μ constant, the cumulative amplification of the $4\Delta x$ wave occurring over a fixed interval of physical time increases as $\Delta x \to 0$ and $\Delta t \to 0$.

2.5.1 *The Discrete-Dispersion Relation*

Although the preceding discussion suggests that useful deductions can be made by examining temporal and spatial differences independently, that discussion also reveals the need to rigorously analyze the combined effects of all finite differences in a specific formula in order to determine the complete behavior of the numerical solution. A useful tool in the analysis of errors in wave propagation problems is the *discrete-dispersion relation*, which is just the finite-difference analogue to the dispersion relation associated with the original continuous problem. The discrete-dispersion relation is obtained by substituting a traveling wave solution of the form

$$\phi_j^n = e^{i(kj\Delta x - \omega n\Delta t)} \tag{2.89}$$

into the finite-difference formula and solving for ω. If the frequency is separated into its real and imaginary parts ($\omega_r + i\omega_i$), (2.89) becomes

$$\phi_j^n = e^{\omega_i n\Delta t} e^{i(kj\Delta x - \omega_r n\Delta t)} = |A|^n e^{i(kj\Delta x - \omega_r n\Delta t)}. \tag{2.90}$$

The determination of the imaginary part of ω is tantamount to a Von Neumann stability analysis, since ω_i determines the amplification factor and governs the rate of numerical dissipation. Information about the phase-speed error can be obtained from ω_r.

Suppose that the advection equation is approximated with leapfrog-time and second-order centered-space differencing such that

$$\frac{\phi_j^{n+1} - \phi_j^{n-1}}{2\Delta t} + c\frac{\phi_{j+1}^n - \phi_{j-1}^n}{2\Delta x} = 0. \tag{2.91}$$

Substitution of (2.89) into this finite-difference scheme gives

$$\left(\frac{e^{-i\omega\Delta t} - e^{i\omega\Delta t}}{2\Delta t}\right)\phi_j^n = -c\left(\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2\Delta x}\right)\phi_j^n,$$

or, equivalently,

$$\sin\omega\Delta t = \mu \sin k\Delta x, \tag{2.92}$$

where $\mu = c\Delta t/\Delta x$. Inspection of (2.92) demonstrates that if $|\mu| < 1$, ω will be real and the scheme will be neutral. The scheme also appears to be neutral when $|\mu| = 1$, but this is a special case. When $\mu = 1$, (2.92) reduces to

$$c^* = \frac{\Delta x}{\Delta t} = c,$$

showing that the numerical solution propagates at the correct phase speed. Although there are no phase-speed errors when $|\mu| = 1$, the two roots of (2.92) become identical if $k\Delta x = \pi/2$, and as a consequence of this double root, the scheme admits a weakly unstable $4\Delta x$ wave. When $\mu = 1$, the weakly growing mode has the form

$$\phi_j^n = n\cos[\pi(j-n)/2]. \tag{2.93}$$

The distinction between the sufficient condition for stability $|\mu| < 1$ and the more easily derived necessary condition $|\mu| \leq 1$ is, however, of little practical significance because uncertainties about the magnitudes of the spatially and temporally varying velocities in real-world applications usually make it impossible to choose a time step such that $|\mu| = 1$.

The frequencies resolvable in the discretized time domain lie in the interval $0 \leq \omega_r \leq \pi/\Delta t$. Except for the special case just considered when $|\mu| = 1$ and $k\Delta x = \pi/2$, there are two resolvable frequencies that satisfy (2.92). Dividing these frequencies by k gives the phase speed of the physical and computational modes

$$c^*_{\text{phys}} \equiv \frac{\omega_{\text{phys}}}{k} = \frac{1}{k\Delta t}\arcsin(\mu \sin k\Delta x)$$

and

$$c^*_{\text{comp}} \equiv \frac{\omega_{\text{comp}}}{k} = \frac{1}{k\Delta t}[\pi - \arcsin(\mu \sin k\Delta x)].$$

As in the differential–difference problem, the $2\Delta x$ physical mode does not propagate. The $2\Delta x$ computational mode flips sign each time step, or equivalently, it moves at the speed $\Delta x/\Delta t$. In the limit of good spatial resolution ($k\Delta x \to 0$),

the Taylor series approximations

$$\sin x \approx x - \tfrac{1}{6}x^3 \quad \text{and} \quad \arcsin x \approx x + \tfrac{1}{6}x^3$$

can be used to obtain

$$c^*_{\text{phys}} \approx c\left(1 - \frac{k^2\Delta x^2}{6}(1-\mu^2)\right). \tag{2.94}$$

If the time step is chosen to ensure stability, then $\mu^2 < 1$, $|c^*| < |c|$, and the decelerating effect of centered spatial differencing dominates the accelerating effects of leapfrog-time differencing. As suggested by (2.94), in practical computations the most accurate results are obtained using a time step such that the maximum value of $|\mu|$ is slightly less than one.

Now consider the forward-time one-sided space scheme

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + c\frac{\phi_j^n - \phi_{j-1}^n}{\Delta x} = 0, \tag{2.95}$$

sometimes referred to as the *donor-cell* scheme. Substitution of (2.89) into (2.95) gives

$$e^{-i\omega\Delta t} - 1 = \mu\left(e^{-ik\Delta x} - 1\right). \tag{2.96}$$

It follows that the exact dispersion relation and the exact solution are obtained in the special case when $\mu = 1$. Further analysis is facilitated by separating (2.96) into its real and imaginary parts

$$|A|\cos\omega_r\Delta t - 1 = \mu(\cos k\Delta x - 1) \tag{2.97}$$

and

$$|A|\sin\omega_r\Delta t = \mu \sin k\Delta x, \tag{2.98}$$

where, $\omega = \omega_r + i\omega_i$, and $|A| \equiv e^{\omega_i\Delta t}$ is the modulus of the amplification factor. Squaring both sides of (2.97) and (2.98) and adding yields

$$|A|^2 = 1 - 2\mu(1-\mu)(1-\cos k\Delta x),$$

which implies that the donor-cell scheme is stable and damping for $0 \leq \mu \leq 1$, and that the maximum damping per time step occurs at $\mu = \frac{1}{2}$.[8]

The discrete dispersion relation

$$\omega_r = \frac{1}{\Delta t}\arctan\left(\frac{\mu \sin k\Delta x}{1 + \mu(\cos k\Delta x - 1)}\right)$$

[8] This stability condition is identical to that obtained via the standard Von Neumann stability analysis in Section 2.3.3.

may be obtained after dividing (2.98) by (2.97). The function $\arctan \omega_r \Delta t$ is single-valued over the range of resolvable frequencies $0 \leq \omega_r \leq \pi/\Delta t$, so as expected for a two-time-level scheme, there is no computational mode. In the limit of good numerical resolution,

$$c^* \equiv \frac{\omega_r}{k} \approx c\left[1 - \frac{(k\Delta x)^2}{6}(1-\mu)(1-2\mu)\right],$$

showing that phase-speed error is minimized by choosing either $\mu = 1$ or $\mu = \frac{1}{2}$. The donor cell scheme is decelerating for $0 < \mu < \frac{1}{2}$, and accelerating for $\frac{1}{2} < \mu < 1$. The phase-speed error in the donor-cell scheme may be minimized by choosing a time step such that $\mu_{\text{avg}} \approx \frac{1}{2}$. Under such circumstances, the donor-cell method will generate less phase-speed error than the leapfrog centered-space scheme. Unfortunately, the good phase-speed characteristics of the donor-cell method are overshadowed by its large dissipation.

It is somewhat surprising that there are values of μ for which the donor cell scheme is accelerating, since forward time-differencing is decelerating and one-sided spatial differencing reduces the phase speed of solutions to the differential–difference advection equation. This example illustrates the danger of relying too heavily on results obtained through the independent analysis of space- and time-truncation error.

2.5.2 *The Modified Equation*

As an alternative to the discrete dispersion equation, numerical dissipation and dispersion can be analyzed by examining a "modified" partial differential equation whose solution satisfies the finite-difference equation to a higher order of accuracy than the solution to the original partial differential equation. This technique is similar to that described in Section 2.4.2 except that since the truncation error includes derivatives with respect to both space and time, all the time derivatives must be expressed as spatial derivatives in order to isolate those terms responsible for numerical dissipation and dispersion. As an example, consider (2.95), the upstream approximation to the constant-wind-speed advection equation, which is a third-order accurate approximation to the modified equation

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = \frac{c\Delta x}{2}(1-\mu)\frac{\partial^2 \psi}{\partial x^2} - \frac{c(\Delta x)^2}{6}(1-\mu)(1-2\mu)\frac{\partial^3 \psi}{\partial x^3}. \tag{2.99}$$

Examination of this equation shows that upstream differencing generates numerical dissipation of $O\left[(\Delta x)^2\right]$ and numerical dispersion of $O\left[(\Delta x)^3\right]$. Both the dissipation and dispersion are minimized as $\mu \to 1$, and the dispersion is also eliminated when $\mu = \frac{1}{2}$.

In deriving the modified equation, the original partial differential equation cannot be used to express all the higher-order time derivatives as spatial derivatives because the finite-difference scheme must approximate the modified equation more accurately than the original partial differential equation (Warming and Hyett 1974). The upstream method (2.99) provides a first-order approximation to the advection equation (2.86), a second-order approximation to

$$\frac{\partial \psi}{\partial t} = -c\frac{\partial \psi}{\partial x} - \frac{\Delta t}{2}\frac{\partial^2 \psi}{\partial t^2} + c\frac{\Delta x}{2}\frac{\partial^2 \psi}{\partial \psi^2}, \tag{2.100}$$

and a third-order approximation to

$$\frac{\partial \psi}{\partial t} = -c\frac{\partial \psi}{\partial x} - \frac{\Delta t}{2}\frac{\partial^2 \psi}{\partial t^2} - \frac{(\Delta t)^2}{6}\frac{\partial^3 \psi}{\partial t^3} + c\frac{\Delta x}{2}\frac{\partial^2 \psi}{\partial x^2} - c\frac{(\Delta x)^2}{6}\frac{\partial^3 \psi}{\partial x^3}. \tag{2.101}$$

The third-order accurate modified equation (2.99) is obtained by repeatedly substituting derivatives of (2.100) into (2.101) until all the first-order terms involving time derivatives are eliminated. The time derivatives in the remaining second-order terms can then be eliminated using the first-order-accurate relation (2.86).

2.5.3 *The Lax–Wendroff Method*

None of the schemes considered previously achieves $O\left[(\Delta t)^2\right]$ accuracy without multistage computation or implicitness or the use of data from two or more previous time levels. Lax and Wendroff (1960) proposed a general method for creating $O\left[(\Delta t)^2\right]$ schemes in which the time derivative is approximated by forward differencing and the $O(\Delta t)$ truncation error generated by that forward difference is canceled by terms involving finite-difference approximations to spatial derivatives. Needless to say, it is impossible to analyze the behavior of a Lax–Wendroff method properly without considering the combined effects of space- and time-differencing.

One important example of a Lax–Wendroff scheme is the following approximation to the advection equation (2.86):

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + c\left(\frac{\phi_{j+1}^n - \phi_{j-1}^n}{2\Delta x}\right) = \frac{c^2 \Delta t}{2}\left(\frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{(\Delta x)^2}\right). \tag{2.102}$$

The lowest-order truncation error in the first term of (2.102), the forward time difference, is

$$\frac{\Delta t}{2}\frac{\partial^2 \psi}{\partial t^2}.$$

However, since ψ is the exact solution to the continuous problem (2.86),

$$\frac{\Delta t}{2}\frac{\partial^2 \psi}{\partial t^2} = \frac{\Delta t}{2}\frac{\partial}{\partial t}\left(-c\frac{\partial \psi}{\partial x}\right) = \frac{c^2 \Delta t}{2}\frac{\partial^2 \psi}{\partial x^2}.$$

The term on the right side of (2.102) will therefore cancel the $O(\Delta t)$ truncation error in the forward time difference to within $O\left[\Delta t(\Delta x)^2\right]$, and as a consequence, the entire scheme is $O\left[(\Delta t)^2\right] + O\left[(\Delta x)^2\right]$ accurate.

The second-order nature of (2.102) may also be demonstrated by expressing it as a two-step formula in which each individual step is centered in space and time. In the first step, intermediate values staggered in space and time are calculated from the relations

$$\frac{\phi_{j+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{1}{2}\left(\phi_{j+1}^{n} + \phi_{j}^{n}\right)}{\frac{1}{2}\Delta t} = -c\left(\frac{\phi_{j+1}^{n} - \phi_{j}^{n}}{\Delta x}\right), \tag{2.103}$$

$$\frac{\phi_{j-\frac{1}{2}}^{n+\frac{1}{2}} - \frac{1}{2}\left(\phi_{j}^{n} + \phi_{j-1}^{n}\right)}{\frac{1}{2}\Delta t} = -c\left(\frac{\phi_{j}^{n} - \phi_{j-1}^{n}}{\Delta x}\right). \tag{2.104}$$

In the second step, ϕ_j^{n+1} is computed from

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} = -c\left(\frac{\phi_{j+\frac{1}{2}}^{n+\frac{1}{2}} - \phi_{j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x}\right). \tag{2.105}$$

The single-step formula (2.102) may be recovered by using (2.103) and (2.104) to eliminate $\phi^{n+\frac{1}{2}}$ and $\phi^{n-\frac{1}{2}}$ from (2.105). One advantage of the two-step formulation is that its extension to more complex problems can be immediately apparent. For example, if the wind speed is a function of the spatial coordinate, c is replaced by $c_{j+\frac{1}{2}}$, $c_{j-\frac{1}{2}}$, and c_j in (2.103), (2.104), and (2.105), respectively. In contrast, the equivalent modification of the single-step formula (see (2.107)) is slightly less obvious.

The amplitude and phase-speed errors of the Lax–Wendroff approximation to the constant-wind-speed advection equation may be examined by substituting a solution of the form (2.90) into (2.102), which yields

$$|A|(\cos\omega_r\Delta t - i\sin\omega_r\Delta t) = 1 + \mu^2(\cos k\Delta x - 1) - i\mu\sin k\Delta x. \tag{2.106}$$

Equating the real and imaginary parts of the preceding equation, and then eliminating $|A|$, one obtains the discrete-dispersion relation

$$\omega_r = \frac{1}{\Delta t}\arctan\left(\frac{\mu\sin k\Delta x}{1 + \mu^2(\cos k\Delta x - 1)}\right).$$

In the limit $k\Delta x \ll 1$, ω_r/k reduces to (2.94), showing that for well-resolved waves, the phase-speed error of the Lax–Wendroff method is identical to that of the leapfrog centered-space scheme. Eliminating ω_r from the real and imaginary parts of (2.106), one obtains

$$|A|^2 = 1 - \mu^2(1 - \mu^2)(1 - \cos k\Delta x)^2,$$

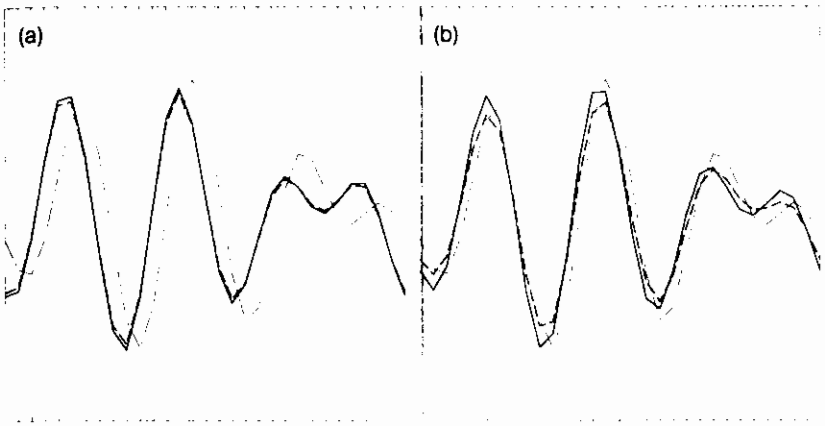

FIGURE 2.19. Leapfrog, second-order space (solid), Lax–Wendroff (dashed), and exact solution (dot-dashed) for the advection of the sum of equal-amplitude $7.5\Delta x$ and $10\Delta x$ sine waves over a distance of twelve grid points using a Courant number of (a) 0.1, and (b) 0.75.

from which it follows that the Lax–Wendroff scheme is stable for $\mu^2 \leq 1$. Short wavelengths are damped most rapidly; the $2\Delta x$ wave is completely eliminated in a single time step if $|\mu| = 1/\sqrt{2}$. Since the shortest wavelengths are seriously in error—once again the phase speed of the $2\Delta x$ wave is zero—this scale-selective damping can be advantageous. Indeed, the scale-selectivity of the dissipation in the Lax–Wendroff scheme is the same as that of a fourth-order spatial filter. Unfortunately, the numerical analyst has little control over the actual magnitude of the dissipation because it is a function of the Courant number, and in most practical problems, μ will vary throughout the computational domain. The dependence of the damping on the Courant number is illustrated in Fig. 2.19, which compares solutions generated by the Lax–Wendroff method and the leapfrog scheme (2.91) using Courant numbers of 0.75 and 0.1. When $\mu = 0.1$, the leapfrog and Lax–Wendroff schemes give essentially the same result, but when μ is increased to 0.75, the damping of the Lax–Wendroff solution relative to the leapfrog scheme is clearly evident. Fig. 2.19 also demonstrates how the phase-speed error in both numerical solutions is reduced as the Courant number increases toward unity.

The term that cancels the $O(\Delta t)$ truncation error in a Lax–Wendroff scheme must be specifically reformulated for each new problem. The following three examples illustrate the general approach. If the flow velocity in (2.102) is a function of x, then

$$\frac{\partial^2\psi}{\partial t^2} = c\frac{\partial}{\partial x}\left(c\frac{\partial\psi}{\partial x}\right),$$

and the right side of (2.102) becomes

$$\frac{c_j \Delta t}{2}\left(\frac{c_{j+\frac{1}{2}}(\phi^n_{j+1}-\phi^n_j)-c_{j-\frac{1}{2}}(\phi^n_j-\phi^n_{j-1})}{(\Delta x)^2}\right). \tag{2.107}$$

If the flow is two-dimensional, the advection problem becomes

$$\frac{\partial \psi}{\partial t}+u\frac{\partial \psi}{\partial x}+v\frac{\partial \psi}{\partial y}=0,$$

and if u and v are constant,

$$\frac{\partial^2 \psi}{\partial t^2}=u\frac{\partial^2 \psi}{\partial x^2}+v\frac{\partial^2 \psi}{\partial y^2}+2uv\frac{\partial^2 \psi}{\partial x \partial y},$$

which must be approximated by a second-order spatial difference. Finally, consider a general system of "conservation laws" of the form

$$\frac{\partial \mathbf{v}}{\partial t}+\frac{\partial}{\partial x}\mathbf{F}(\mathbf{v})=\mathbf{0},$$

where $\mathbf{v}$ and $\mathbf{F}$ are column vectors. Then

$$\frac{\partial^2 \mathbf{v}}{\partial t^2}=\frac{\partial}{\partial x}\left(\mathbf{J}\frac{\partial \mathbf{F}}{\partial x}\right), \tag{2.108}$$

where $\mathbf{J}$ is the Jacobian matrix whose ijth element is $\partial F_i/\partial v_j$. Once again, this matrix operator must be approximated by second-order spatial differences.

In many applications the Lax–Wendroff method can be implemented more easily and more efficiently using the two-step method (2.103)–(2.105), or the following variant of the two-step method suggested by MacCormack (1969):

$$\tilde{\mathbf{v}}_j=\mathbf{v}^n_j-\frac{\Delta t}{\Delta x}\left[\mathbf{F}(\mathbf{v}^n_j)-\mathbf{F}(\mathbf{v}^n_{j-1})\right],$$

$$\tilde{\tilde{\mathbf{v}}}_j=\tilde{\mathbf{v}}_j-\frac{\Delta t}{\Delta x}\left[\mathbf{F}(\tilde{\mathbf{v}}_{j+1})-\mathbf{F}(\tilde{\mathbf{v}}_j)\right],$$

$$\mathbf{v}^{n+1}_j=\frac{1}{2}\left(\tilde{\mathbf{v}}_j+\tilde{\tilde{\mathbf{v}}}_j\right).$$

These two-step methods generate numerical approximations to the higher-order spatial derivatives required to cancel the $O(\Delta t)$ truncation error in the forward time difference without requiring the user to explicitly evaluate complex expressions like (2.108). The MacCormack method is particularly useful, since it easily generalizes to problems in two or more spatial dimensions.

In the classical Lax–Wendroff method, the spatial derivatives are approximated using centered differences, but other approximations are also possible. If the spatial dependence of ψ is not discretized, the Lax–Wendroff approximation to the advection equation (2.102) may be written

$$\frac{\phi^{n+1}-\phi^n}{\Delta t}+c\frac{\partial \phi^n}{\partial x}=\frac{c^2 \Delta t}{2}\frac{\partial \phi^n}{\partial x^2}.$$

Warming and Beam (1976) proposed the following upwind approximation to the preceding:

$$\phi^{n+1}_j=\phi^n_j-\mu\left(\phi^n_j-\phi^n_{j-1}\right)-\frac{\mu}{2}(1-\mu)\left(\left(\phi^n_j-2\phi^n_{j-1}+\phi^n_{j-2}\right)\right), \tag{2.109}$$

which is $O\left[(\Delta t)^2\right]+O\left[(\Delta x)^2\right]$ accurate and is stable for $0 \le \mu \le 2$.

2.6 Summary Discussion of Elementary Methods

In this chapter we have investigated the performance of schemes for approximating the constant-wind-speed advection equation. Let us now recapitulate the better methods discussed in this chapter and briefly summarize the conditions under which they might be expected to yield good results in more complicated problems. Further analysis of the performance of these schemes in more complex situations will be presented in the following chapters of this book.

First consider the relatively atypical class of problems in which the solution is sufficiently smooth that it can always be properly resolved on the numerical mesh.[9] Under these circumstances any stable method can be expected to converge to the correct result as the space–time grid is refined. Higher-order schemes will converge to *smooth* solutions more rapidly than low-order methods as the mesh size is decreased. Thus, even though higher-order methods require more computations per grid point per time step, genuinely high accuracy (i.e., several significant digits) can usually be achieved more efficiently by using a high-order scheme on a relatively coarse mesh than by using a low-order scheme on a finer mesh. Suitable high-order time differences include the third-order Adams–Bashforth method and the third- and fourth-order Runge–Kutta methods. Spatial differences might be computed using either explicit or compact fourth- or sixth-order differences; however, spectral methods, which will be discussed in Chapter 4, can be a better choice when very high accuracy is desired.

Most low-viscosity flows do not remain completely smooth. Instead, they develop at least some features with spatial scales shorter than or equal to that of an individual grid cell. Such small-scale features cannot be accurately captured by any numerical scheme, and the unavoidable errors in these small scales can feed back on the larger-scale flow and thereby exert a significant influence on the overall solution. In such circumstances there is no hope of computing an approximation to the correct solution that is accurate to several significant digits.

[9] An example of this type is provided by the barotropic vorticity equation, which will be discussed in Section 3.6.2.

Although the larger-scale features may be approximated with considerable quantitative accuracy, generally one must either be content with a qualitatively correct representation of the shortest-scale features or must remove these features with some type of numerical smoothing. Since it is not realistic to expect convergence to the correct solution in such problems, it is not particularly important to use high-order methods. Instead, one generally employs the finest possible numerical grid, selects a method that captures the behavior of moderately resolved waves with reasonable fidelity, and ensures that any spurious poorly resolved waves are eliminated by either explicit or implicit numerical dissipation. The numerical dissipation associated with all the schemes considered in this chapter is applied throughout the entire numerical domain. An alternative approach will be considered in Chapter 5, in which the implicit dissipation is primarily limited to those regions where the approximate solution is discontinuous or very poorly resolved.

Given that some degree of dissipation must generally be included to generalize the methods described in this chapter to practical problems involving low-viscosity flow, the neutral amplification factors associated with leapfrog time differencing and centered spatial differences are less advantageous than they may first appear. The difficulties associated with time splitting that can arise in nonlinear problems make the leapfrog scheme relatively unattractive in comparison with the third-order Adams–Bashforth or third- or fourth-order Runge–Kutta methods. The advantages of these relatively high-order methods are not primarily associated with their small truncation error (since some features will be poorly resolved) but arise from their stability and relative efficiency. The second-order Magazenkov and leapfrog–trapezoidal methods are also possible alternatives to the leapfrog scheme. Even forward differencing is a possibility, provided that it is used in a Lax–Wendroff method and that the implicit diffusion in the Lax–Wendroff scheme is limited by using a sufficiently small time step.

Now consider the choice of spatial difference approximations. Approximations based on centered spatial differences typically require the use of an explicit fourth- or sixth-derivative dissipative filter and are therefore less efficient than a third-order upstream approximation. This lack of efficiency is compensated by two practical advantages. First, it is not necessary to determine the upstream direction at each grid point when formulating the computer algorithm to evaluate a centered spatial difference. The determination of the upstream direction is not particularly difficult in advection problems where all signal propagation is directed along a clearly defined flow, but it can be far more difficult in problems admitting wave solutions that propagate both to the right and to the left. The second advantage of a centered difference used in conjunction with a spatial filter is that one can explicitly control the magnitude of the artificial dissipation, whereas the magnitude of the numerical dissipation associated with an upstream difference is implicitly determined by the local wind speed. The compact schemes appear to provide particularly good formulae for the evaluation of centered spatial differences because they remain accurate at relatively short wavelengths ($3\Delta x$ or $4\Delta x$) and use information at a minimum number of spatial grid points, which reduces the amount of special coding required near the boundaries of the spatial domain.