The Third-Order Adams–Bashforth Method: An Attractive Alternative to Leapfrog Time Differencing

DALE R. DURRAN

Department of Atmospheric Sciences, University of Washington, Seattle, Washington

(Manuscript received 21 February 1990, in final form 21 September 1990)

ABSTRACT

The third-order Adams–Bashforth method is compared with the leapfrog scheme. Like the leapfrog scheme, the third-order Adams–Bashforth method is an explicit technique that requires just one function evaluation per time step. Yet the third-order Adams–Bashforth method is not subject to time splitting instability and it is more accurate than the leapfrog scheme. In particular, the $O[(\Delta t)^4]$ amplitude error of the third-order Adams–Bashforth method can be a marked improvement over the $O[(\Delta t)^2]$ amplitude error generated by the Asselin-filtered leapfrog scheme—even when the filter factor is very small. The $O[(\Delta t)^4]$ phase-speed errors associated with third-order Adams–Bashforth time differencing can also be significantly less than the $O[(\Delta t)^2]$ errors produced by the leapfrog method. The third-order Adams–Bashforth method does use more storage than the leapfrog method, but its storage requirements are not particularly burdensome. Several numerical examples are provided illustrating the superiority of third-order Adams–Bashforth time differencing. Other higher-order alternatives to the Adams–Bashforth method are also surveyed. A discussion is presented describing the general relationship between the truncation error of an ordinary differential solver and the amplitude and phase-speed errors that develop when the scheme is used to integrate oscillatory systems.

1. Introduction

Leapfrog time differencing is widely used to numerically simulate advective processes in low-viscosity fluids. The leapfrog scheme is an efficient method that achieves second-order accuracy with just one function evaluation per time step. Leapfrog time differencing does not artificially damp linear oscillatory motion nor does it produce instability by amplifying the oscillations. These advantages are somewhat diminished by the large phase-speed error of the leapfrog scheme, and the unsuitability of leapfrog differencing for the representation of diffusive and Rayleigh damping processes. However, the most serious problem associated with the leapfrog scheme is the "time splitting" instability that develops when the method is used to model nonlinear fluid dynamics. This time splitting instability is associated with the leapfrog scheme's undamped computational mode. Time splitting can be controlled through the use of an Asselin–Robert time filter (Asselin 1972; Robert 1966) or by periodically discarding the data at the oldest time level and reinitializing the leapfrog solution through a single integration step with a two-level scheme. Atmospheric modelers commonly follow the first strategy representing time derivatives with the Asselin-filtered leapfrog scheme. Asselin filtering does, nevertheless, degrade the accuracy of the calculations—reducing the $O[(\Delta t)^2]$ truncation error of the unfiltered leapfrog scheme to $O[\Delta t]$.

One can obviously avoid the problems with the leapfrog scheme's computational mode by using a different time differencing scheme. The purpose of this paper is to examine alternatives to leapfrog differencing; the primary emphasis will be on one particular alternative: the third-order Adams–Bashforth method. There is no lack of possible substitutes for leapfrog differencing. Any ordinary differential equation (ODE) solver is a potential time differencing scheme. In addition to the general ODE solvers, special differencing schemes that produce a beneficial cancellation between the temporal and spatial truncation error can be constructed for particular systems of partial differential equations, e.g., the Lax–Wendroff method for the advection equation. This second approach, in which time and space differencing formulas are developed simultaneously, will not be considered further in this paper because the results are inherently specific to the system of equations for which the scheme was constructed.

One class of ODE solvers that might serve as substitutes for the leapfrog scheme is the family of two time-level differencing methods, which have no computational mode. In order to achieve second-order accuracy, however, a two-level scheme must either be implicit (e.g., the trapezoidal method) or iterative. As a consequence, two-level schemes require more com-

Corresponding author address: Dr. Dale R. Durran, Dept. of Atmospheric Sciences, University of Washington, Seattle, WA 98195.

putational effort to achieve the same accuracy as the leapfrog scheme. In addition, most low-order two-level iterative schemes produce undesirable amplitude errors when applied to oscillatory problems. The first-order Matsuno scheme (Matsuno 1966) requires two function evaluations per time step and heavily damps the solution. Second-order Runge–Kutta methods (including the "Huen" method, Mesinger and Arakawa 1976) require two function evaluations per time step and generate unstable, amplifying oscillations. Young's method A (Young 1968), which is second order, requires three function evaluations per time step and is also unstable. The instabilities in these second-order methods are relatively slow growing and can sometimes be tolerated if one uses a sufficiently small time step and integrates for a limited time. Previous researchers (e.g., Young 1968) have also considered the stable and highly accurate fourth-order Runge–Kutta method. Unfortunately, the fourth-order Runge–Kutta scheme requires four function evaluations per time step and considerably more storage than the lower-order two time-level schemes.

A second class of potential replacements for leapfrog differencing are the multistep ordinary differential equation solvers with damped computational modes. The only member of this class that seems to have undergone much scrutiny is the second-order Adams–Bashforth method (Lilly 1965). Like leapfrog differencing, the second-order Adams–Bashforth method is a three-level scheme that requires only one function evaluation per time step. Unlike the leapfrog scheme, it is not subject to time-splitting instability and it can be used to model diffusive and Rayleigh damping processes. The disadvantage of the second-order Adams–Bashforth method is that when it is used to model oscillatory phenomena, the physical mode is subject to the same instability as the second-order Runge–Kutta method; i.e., the oscillation amplifies with time but the instability can be tolerated if short time steps are used and the total length of the integration is limited. Another disadvantage of the second-order Adams–Bashforth method is its large phase-speed error, which is 2½ times that of the leapfrog scheme (Mesinger and Arakawa 1976). Nevertheless, a minority of atmospheric modelers have preferred the weak amplification of the second-order Adams–Bashforth scheme to the time splitting problems of the leapfrog method and have employed the Adams–Bashforth scheme in their numerical models (e.g., Moeng 1984).

The third-order Adams–Bashforth method appears to be an even more attractive alternative. Like its second-order cousin, the third-order Adams–Bashforth method is an explicit scheme requiring one function evaluation per time step. The third-order Adams–Bashforth method is not subject to time splitting; its maximum stable time step is similar to the maximum stable time step permitted by the Asselin-filtered leapfrog scheme; and it allows a stable and accurate representation of diffusive and Rayleigh damping processes. The only potential drawback of the third-order Adams–Bashforth method is its storage requirement.

Concern about storage seems to be the main reason that higher-order multistep methods have not received more attention. The practical considerations that led to this preference for very low-storage algorithms have, however, changed over the last several years as the in-core memory in many computer systems has increased much more rapidly than CPU power. As an example, the original Cray 1 had one million words of memory and one processer with a 12-ns clock time. At present, the most powerful Cray has 128 million words of memory and eight processors, each operating with a 6-ns clock time. Total processing capacity has thus increased by a factor of 16 while storage capacity has increased by a factor of 128. As a consequence, most modest-sized atmospheric models are not limited by lack of memory, and they can easily accommodate the storage needs of the third-order Adams–Bashforth method. Nevertheless, suppose that memory truly is the limiting factor in a particular three-dimensional simulation. If the memory-limited model currently employs leapfrog time differencing and the code is optimized to minimize memory requirements, storage must be allocated for the complete set of prognostic variables at two different time levels plus a smaller temporary storage array. Switching to an efficiently coded third-order Adams–Bashforth method will increase the necessary storage by a factor of 3/2. This extra storage can be created by reducing the number of gridpoints in each of the three spatial dimensions by a factor of $(2/3)^{1/3}$, or equivalently by a 12% reduction in the spatial resolution. In some applications, the loss of accuracy associated with a 12% reduction in spatial resolution can easily be offset by the improvements associated with third-order time differencing.

The remainder of this paper is organized as follows. The third-order Adams–Bashforth method is described and analyzed in section 2, where its theoretical properties are compared with those of the Asselin-filtered leapfrog scheme. Section 3 contains a discussion of the relationship between time differencing error and the errors introduced by spatial differencing in wave-propagation problems. Several numerical tests, which contrast the performance of Adams–Bashforth and time-filtered leapfrog differencing, are presented in section 4. A brief survey of other higher-order time differencing schemes appears in section 5 together with a description of the general relationship between the truncation error of a finite-difference scheme and the amplitude and phase errors that occur when the scheme is applied to oscillatory problems. Section 6 contains a discussion of the difficulties that arise if the third-order Adams–Bashforth method is substituted for leapfrog differencing in semi-implicit models. The conclusions appear in section 7.

2. Comparative analysis of the third-order Adams–Bashforth method and the Asselin-filtered leapfrog scheme

The general Nth-order Adams–Bashforth approximation to the ordinary differential equation

$$d\psi/dt = F(\psi), \qquad (1)$$

has the form

$$\frac{\phi^{n+1}-\phi^{n}}{\Delta t} = \sum_{j=0}^{N-1} a_j F(\phi^{n-j}), \qquad (2)$$

where ϕ^n is the numerical approximation to $\psi(n\Delta t)$ (Gear 1971). The coefficients a_j may be determined by substituting the Taylor series expansions for ψ and $F(\psi)$ into (2) and choosing the a_j to cancel all terms less than order $(\Delta t)^N$. An alternative derivation of the coefficients can be obtained by writing (1) as the equivalent integral equation

$$\psi[(n+1)\Delta t] = \psi(n\Delta t) + \int_{n\Delta t}^{(n+1)\Delta t} F[\psi(t)]dt. \qquad (3)$$

The Adams–Bashforth scheme approximates the preceding integral as

$$\int_{n\Delta t}^{(n+1)\Delta t} F[\psi(t)]dt = \Delta t \sum_{j=0}^{N-1} a_j F(\phi^{n-j}). \qquad (4)$$

The coefficients a_j can be evaluated by requiring (4) to be exact for all polynomials less than order N.

The first-order Adams–Bashforth method, in which $a_0 = 1$, is the familiar Euler (or forward) difference. The second-order case, in which $a_0 = \frac{3}{2}$, $a_1 = -\frac{1}{2}$, is discussed by Lilly (1965) and Mesinger and Arakawa (1976). The third-order Adams–Bashforth scheme is

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

The family of Adams–Bashforth schemes are part of the classical literature. Numerical values for the coefficients of all Adams–Bashforth methods through order 6 are listed in Gear (1971, p. 109). Although it is a classical scheme, the third-order Adams–Bashforth method does not appear to have been widely employed in numerical models for the solution of partial differential equations.

The phase-speed and amplitude errors introduced by time differences in the numerical representation of nondissipative wave phenomena may be examined by analyzing solutions to the "oscillation equation"

$$d\psi/dt = i\omega\psi. \qquad (6)$$

Assuming ω is a real constant, numerical solutions to (6) can be obtained from finite-difference approximations of the form (2) such that

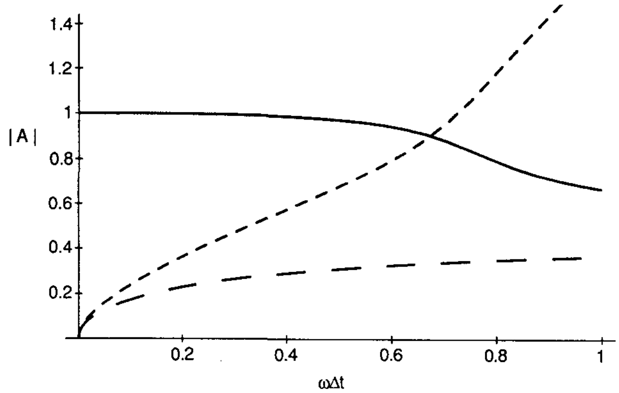

FIG. 1. Magnitude of the amplification factor for the third-order Adams–Bashforth scheme plotted as a function of $\omega\Delta t$. Solid line is the physical mode; dashed curves are the two computational modes.

$$\phi^{n+j} = A^j \phi^n, \qquad (7)$$

where A is the (complex valued) "amplification factor." The amplification factor associated with the exact solution to (6) is $A_e = e^{i\omega\Delta t}$. Thus, following Mesinger and Arakawa 1976, the relative amplitude error in the numerical solution may be defined as $|A|/|A_e| = |A|$, and the relative phase error as

$$R \equiv \frac{\arg A}{\arg A_e} = \frac{1}{\omega\Delta t}\arctan\left(\frac{\Im[A]}{\Re[A]}\right). \qquad (8)$$

The amplification factor for the third-order Adams–Bashforth solution to the oscillation equation (6) satisfies the cubic equation

$$A^3 - \left(1 + \frac{23}{12} i\omega\Delta t\right)A^2 + \frac{4}{3} i\omega\Delta t A - \frac{5}{12} i\omega\Delta t = 0. \qquad (9)$$

Figure 1 shows $|A|$, plotted as a function of $\omega\Delta t$, for each of the three roots of (9). Unlike its first- and second-order cousins, the third-order Adams–Bashforth scheme dampens the physical mode. One computational mode becomes unstable for $\omega\Delta t > 0.724$. (Note also that the damping rate of the physical mode exceeds the damping rate of one computational mode for $\omega\Delta t > 0.676$.) In the limit of $\omega\Delta t \ll 1$, both computational modes are strongly damped and the amplification factor for the physical mode is[1]

$$|A|_{\text{phys}} = 1 - \frac{3}{8}(\omega\Delta t)^4 + O[(\omega\Delta t)^6]. \qquad (10)$$

In the limit of $\omega\Delta t \ll 1$, the phase-speed error of the physical mode is

$$R_{\text{phys}} = 1 + \frac{289}{720}(\omega\Delta t)^4 + O[(\omega\Delta t)^6]. \qquad (11)$$

[1] The asymptotic results in (10), (11), and (17)–(19) were obtained using the symbolic manipulation capabilities of *Mathematica* (Wolfram 1988).

In agreement with the general proposition stated in section 5, the amplitude and phase-speed errors generated by the third-order Adams–Bashforth method are both fourth order.

Let us compare the amplitude and phase errors in (10) and (11) with the equivalent values for the Asselin-filtered leapfrog scheme:

$$\phi^{n+1} = \overline{\phi^{n-1}} + 2\Delta t F(\phi^n), \qquad (12)$$

$$\overline{\phi^n} = \phi^n + \gamma(\overline{\phi^{n-1}} - 2\phi^n + \phi^{n+1}). \qquad (13)$$

In the preceding, an overbar denotes a time-filtered value. Observe that the filtering coefficient γ is equivalent to $\nu/2$ in Asselin's (1972) notation. A value of $\gamma = 0.06$ is typically used in the NCAR community climate model (Williamson 1983). Values of $\gamma = 0.2$ are common in mesoscale convective models; indeed Schlesinger et al. (1983) recommend a value of γ in the range 0.25–0.3 for general advection-diffusion modeling. (Schlesinger et al. do not recommend leapfrog differencing for the diffusion terms.)

Suppose that (12) and (13) are used to obtain an approximate solution to the oscillation equation, and solutions are sought in the form (7). Under the assumption that $\overline{(A\phi^n)} = A(\overline{\phi^n})$, one obtains Asselin's expression for the amplification factor

$$A = \gamma + i\omega\Delta t \pm [(1 - \gamma^2) - (\omega\Delta t)^2]^{1/2}. \qquad (14)$$

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

$$\overline{(A\phi^0)} - A(\overline{\phi^0}) = \overline{\phi^1} - \phi^1$$
$$= \gamma(\phi^0 - 2\phi^1 + \phi^2) \neq 0. \qquad (15)$$

However, (13) may be rewritten

$$\overline{(A\phi^{n+1})} - A(\overline{\phi^{n+1}}) = \gamma[\overline{(A\phi^n)} - A(\overline{\phi^n})]$$
$$= \gamma^{n+1}[\overline{\phi^1} - \phi^1]. \qquad (16)$$

In all cases of practical interest $n \gg 1$ and $\gamma \ll 1$; therefore (16) implies that A may be factored out of the filtering operation with negligible error. Having justified the derivation of (14), an analysis of the amplitude and phase-speed errors of the Asselin-filtered leapfrog solution may be performed in the usual manner.

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

$$|A|_{\text{phys}} = 1 - \frac{\gamma(\omega\Delta t)^2}{2(1-\gamma)} - \frac{\gamma(1+\gamma-\gamma^2)(\omega\Delta t)^4}{8(1-\gamma)^3} + O[(\omega\Delta t)^5], \qquad (17)$$

$$|A|_{\text{comp}} = (1 - 2\gamma) + \frac{\gamma(\omega\Delta t)^2}{2 - 6\gamma + 4\gamma^2} + O[(\omega\Delta t)^3]. \qquad (18)$$

In the limit $\gamma \rightarrow 0$, both modes are neutral and the standard leapfrog result is recovered. As might be expected from the structure of (13), Asselin–Robert time filtering introduces a *second-order* error in the amplitude of the physical mode. Obviously, the amplitude error can be reduced by choosing a small value for γ. However, since the damping term in the third-order Adams–Bashforth method is fourth-order, it damps low-frequency oscillations much less than the second-order diffusion in the Asselin-filtered leapfrog method. If $\gamma = 0.2$ (the typical value for mesoscale convective models) all oscillations with periods greater than $11\Delta t$ are damped more heavily by the Asselin-filtered leapfrog method. If γ is reduced to 0.06 (the typical value in the community climate model) the Asselin-filtered leapfrog method produces more damping than the third-order Adams–Bashforth method whenever the period of the oscillation exceeds $22\Delta t$.

In the limit of $\omega\Delta t \ll 1$, the phase speed of the Asselin-filtered physical mode becomes

$$R_{\text{phys}} = 1 + \frac{1 + 2\gamma}{6(1-\gamma)}(\omega\Delta t)^2. \qquad (19)$$

As is the case for the standard leapfrog scheme, the phase speed is second-order accurate. Asselin–Robert filtering increases the phase-speed error doubling it when γ is 0.25. Once again, the third-order Adams–Bashforth scheme will follow the phase of low-frequency oscillations much more accurately than the Asselin-filtered leapfrog scheme. Even with $\gamma = 0$, the third-order Adams–Bashforth method is more accurate for all oscillations with periods greater than $10\Delta t$.

The amplification factors for all physical and computational modes of the Asselin-filtered leapfrog scheme and the third-order Adams–Bashforth method are plotted as a function of $\omega\Delta t$ in Fig. 2. Figure 2a shows the amplification factor for the true solution to (6), for $\omega\Delta t = 0.1, 0.2, \cdots, 0.9$. All values of A_e lie on the unit circle in the complex plane. The corresponding amplification factors for the third-order Adams–Bashforth scheme are plotted in Fig. 2b. The Adams–Bashforth scheme has two computational modes, both of which are strongly damped when $\omega\Delta t$ is small. The computational mode that becomes unstable for $\omega\Delta t > 0.724$ has an approximate period of $4\Delta t$. Amplification factors for the Asselin-filtered leapfrog scheme are plotted in Fig. 2c for the case $\gamma = 0.2$. Note how the damping introduced into the computational mode by Asselin filtering is relatively independent of the time step. In the case $\gamma = 0.2$, stability requires $\omega\Delta t \leq 0.816$, which is almost as stringent as the third-order Adams–Bashforth stability condition. As in the third-order Adams–Bashforth scheme, the period of unstable oscillations is approximately $4\Delta t$.

This study will now turn from the problem of modeling inviscid oscillatory systems to that of simulating dissipative processes. A prototype equation for the time

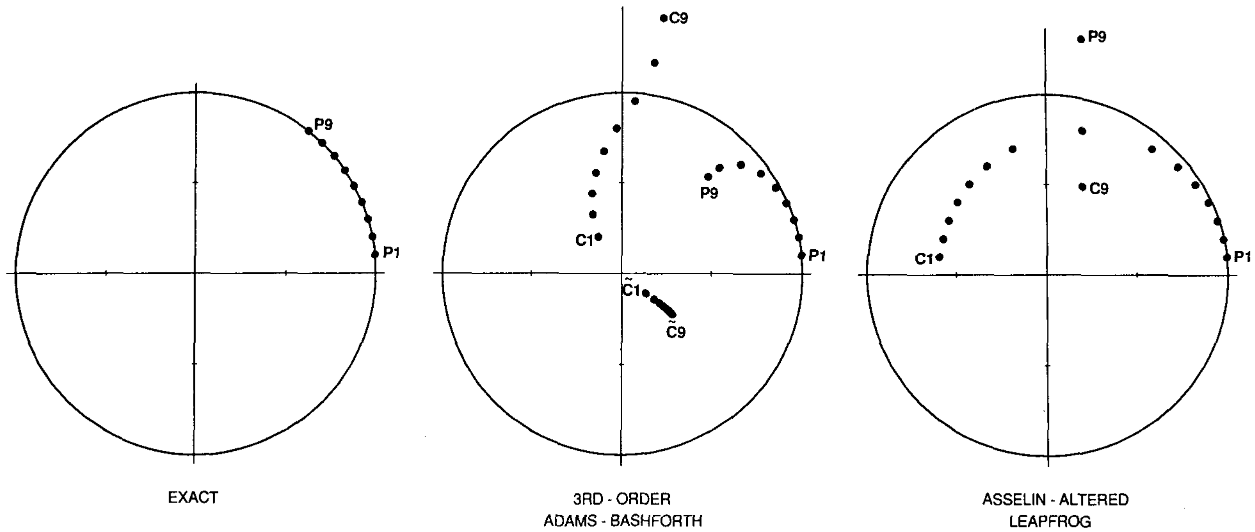

FIG. 2. Amplification factor plotted in the complex plane for the exact solution, the third-order Adams–Bashforth solution, and the Asselin-filtered leapfrog solution with $\gamma = 0.2$. Individual points associated with all physical and computational modes are plotted for $\omega\Delta t = 0.1, 0.2, \cdots, 0.9$. Points corresponding to a physical (computational) mode and a value of $\omega\Delta t = 0.1$ are labeled "P1" ("C1" or "C̃1"). Points obtained with $\omega\Delta t = 0.9$ are also labeled. The unit circle is graphed as a reference.

evolution of a dissipative process is the so-called "friction equation" (again following the terminology of Arakawa and Mesinger 1976):

$$d\psi/dt = -\kappa\psi, \tag{20}$$

where κ is a nonnegative real constant. The leapfrog technique cannot be used to integrate (20) without a transformation of variables because the resulting difference equations are unstable (Lilly 1965; Kreiss and Oliger 1973). Thus, if diffusion is included in models with leapfrog time differencing and if the diffusion terms are evaluated using an explicit scheme, those terms are generally lagged in time and integrated with a first-order Euler scheme (e.g., Durran and Klemp 1983). If the lagged Euler approach is used to difference (20), the result is

$$\phi^{n+1} = \phi^{n-1} - 2\Delta t\kappa\phi^{n-1}. \tag{21}$$

The stability criteria necessary to guarantee decaying, nonoscillatory solutions to (21) is $\kappa\Delta t < 0.5$.

All Adams–Bashforth methods may be applied to (20) without encountering the instability generated by the leapfrog scheme. The third-order Adams–Bashforth method will produce stable nonoscillatory solutions to (20) whenever $\kappa\Delta t < 0.545$. (One computational mode is stable, but less strongly damped than the physical mode for $0.359 < \kappa\Delta t < 0.545$.) Thus, the third-order Adams–Bashforth method permits the explicit integration of diffusion terms with a slightly larger stable time step than that allowed by the time-lagged Euler method. The third-order Adams–Bashforth solution should also be far more accurate than one produced by the first-order lagged Euler method because it would be obtained using a higher-order scheme and a smaller time step.

3. The interaction between time-differencing error and errors in spatial differencing

The relative importance of temporal and spatial differencing error in the numerical representation of a traveling sinusoidal wave is largely determined by the absolute value of the Courant number,

$$|\mu| \equiv \left|\frac{c\Delta t}{\Delta x}\right| = \frac{|ck|\Delta t}{|k|\Delta x} = \frac{\Delta t/T}{\Delta x/L}. \tag{22}$$

Here k is the wavenumber, L is the wavelength, c is the phase speed, and $T \equiv 2\pi/(kc)$ is the period of the wave. According to (22), the Courant number may be interpreted as the ratio of the temporal resolution to the spatial resolution in a wave propagating at phase speed c. The familiar Courant–Fredricks–Lewy (CFL) stability condition requires $|\mu| \leq O(1)$ for any explicit difference scheme. In some atmospheric models, the most stringent CFL time step restrictions are imposed by fast-moving waves that have little meteorological significance (i.e., gravity waves in a global primitive equation model). Then, the Courant number associated with the slower-moving, meteorologically important modes can be very small and the effect of time truncation error on the accuracy of the important modes will be negligible. If, however, the stability constraints imposed by the fastest moving waves are eliminated by the use of filtered equations or by numerical techniques like the semi-implicit method (Kwizak and Robert 1971) or the two time-step method (Klemp and Wilhelmson 1978), the Courant number associated with the features of interest may approach unity and time-truncation errors may become significant.

Consider, therefore, the importance of time-truncation error in simulations performed with Courant numbers near the stability limit. Some basic techniques for solving the one-dimensional advection equation,

$$\frac{\partial \psi}{\partial t} + c\frac{\partial \psi}{\partial x} = 0, \qquad (23)$$

are exact when $\mu = 1$. Examples of this type include upstream differencing, the Lax–Wendroff method, and the leapfrog scheme with centered second-order spatial differencing. Computations utilizing these schemes are most accurate when performed with the largest stable time step. Unfortunately, when choosing the optimum time step for the widely used leapfrog-time fourth-order space approximation to (23),

$$\frac{\phi_j^{n+1} - \phi_j^{n-1}}{2\Delta t} + c\left[\frac{4}{3}\left(\frac{\phi_{j+1}^n - \phi_{j-1}^n}{2\Delta x}\right) - \frac{1}{3}\left(\frac{\phi_{j+2}^n - \phi_{j-2}^n}{4\Delta x}\right)\right] = 0, \qquad (24)$$

one must strike a compromise between efficiency and accuracy since the largest stable time step does not give the most accurate result. The dependence of solutions to (24) on the Courant number is illustrated in Fig. 3. In this example, the initial condition is

$$\phi^0(x) = \begin{cases} \{64[(x-1/2)^2 - 1/64]\}^2, & \text{if} \quad 3/8 \leqslant x \leqslant 5/8; \qquad (25) \\ 0, & \text{otherwise,} \end{cases}$$

the horizontal mesh size $\Delta x = 1/32$, and ϕ is periodic over the interval $0 \leqslant x \leqslant 1$. The advection velocity c is one-fourth. The results in Fig. 3 are for a nondimensional time of 3, where one unit of nondimensional time is the time required for the flow to complete one circuit around the periodic domain. As indicated in Fig. 3a, very poor accuracy is obtained using $\mu = 0.7272$, a Courant number just below the stability limit. Much better results are produced when $\mu = 0.5$ (Fig. 3b); although the phase speed of the spike is overestimated, the amplitude of the spike is nearly correct, and the dispersive ripples are small. When $\mu = 0.3$ (Fig. 3c), the leading phase-speed error of the leapfrog time difference no longer dominates the lagging phase-speed error of the spatial difference and the spike trails the true solution. A further increase in the phase lag and the amplitude of the dispersive ripples is evident in Fig. 3d, which shows the case $\mu = 0.1$. Continued reductions in μ produce results very similar to Fig. 3d. In Fig. 3a, the numerical error is dominated by time differencing; in Fig. 3d the dominant error is produced by spatial differencing. Some beneficial cancellation between the temporal- and spatial-differencing error is evident in Figs. 3b,c.

The dependence of the phase-speed error on μ can easily be understood by evaluating the discrete dispersion relation satisfied by traveling wave solutions to (24) (Haltiner and Williams 1980). For reasonably resolved waves $k\Delta x < 1$, $ck\Delta t < 1$, and the leading terms in the phase-speed error are

$$c_{4lf} = c\left[1 - \frac{k^2\Delta x^2}{6}\left(\frac{k^2\Delta x^2}{5} - \mu^2\right)\right]. \qquad (26)$$

In contrast to the more familiar example where leapfrog differencing is combined with centered second-order spatial differences (Haltiner and Williams 1980), the value of μ that minimizes the phase-speed error is dependent on the wavenumber. In order to minimize the phase-speed error for an $8\Delta x$ wave, Δt should be just 21% of the maximum stable time step. The optimal value of μ decreases linearly with decreasing wavenumber. As a result, very small time steps are required to minimize the phase-speed error of long, well-resolved waves. In practice, the phase-speed error of well-resolved waves will be small, but it will be dominated by the effects of time differencing.

One can obtain phase-speed errors that are truly fourth-order by combining third-order Adams–Bashforth time differencing with centered fourth-order spatial differences. The leading-order terms in the resulting discrete dispersion relation are

$$c_{4ab3} = c\left[1 - \frac{k^4\Delta x^4}{30}\left(1 - \frac{289\mu^4}{24}\right)\right] \qquad (27)$$

where it is again assumed that $k\Delta x < 1$, $ck\Delta t < 1$. According to (27), the optimal value of μ is independent of k and the fourth-order phase-speed error is eliminated when $\mu = 0.537$. This value is slightly larger than the maximum stable time step of 0.527. Thus, when third-order Adams–Bashforth time differencing is used in conjunction with centered fourth-order spatial differences, the total phase-speed error is always dominated by spatial differencing, and the minimum phase-speed error is achieved by taking the largest stable time step.

The amplitude of a traveling wave is also influenced by the Courant number. As previously discussed in section 2, the third-order Adams–Bashforth method damps the solution at a rate proportional to $(\omega\Delta t)^4$, whereas Asselin time filtering introduces an $O[(\omega\Delta t)^2]$ dissipation. Since $\omega\Delta t = \mu k\Delta x$, the dissipation of traveling waves generated by temporal filtering is equivalent to damping from a spatial filter, and Asselin time filtering effectively introduces a second-order spatial filter on all propagating features. Many numerical models employ fourth-order spatial filters to selectively damp the shortest wavelengths. Some of this scale selectivity will be lost if the same model uses Asselin time filtering,

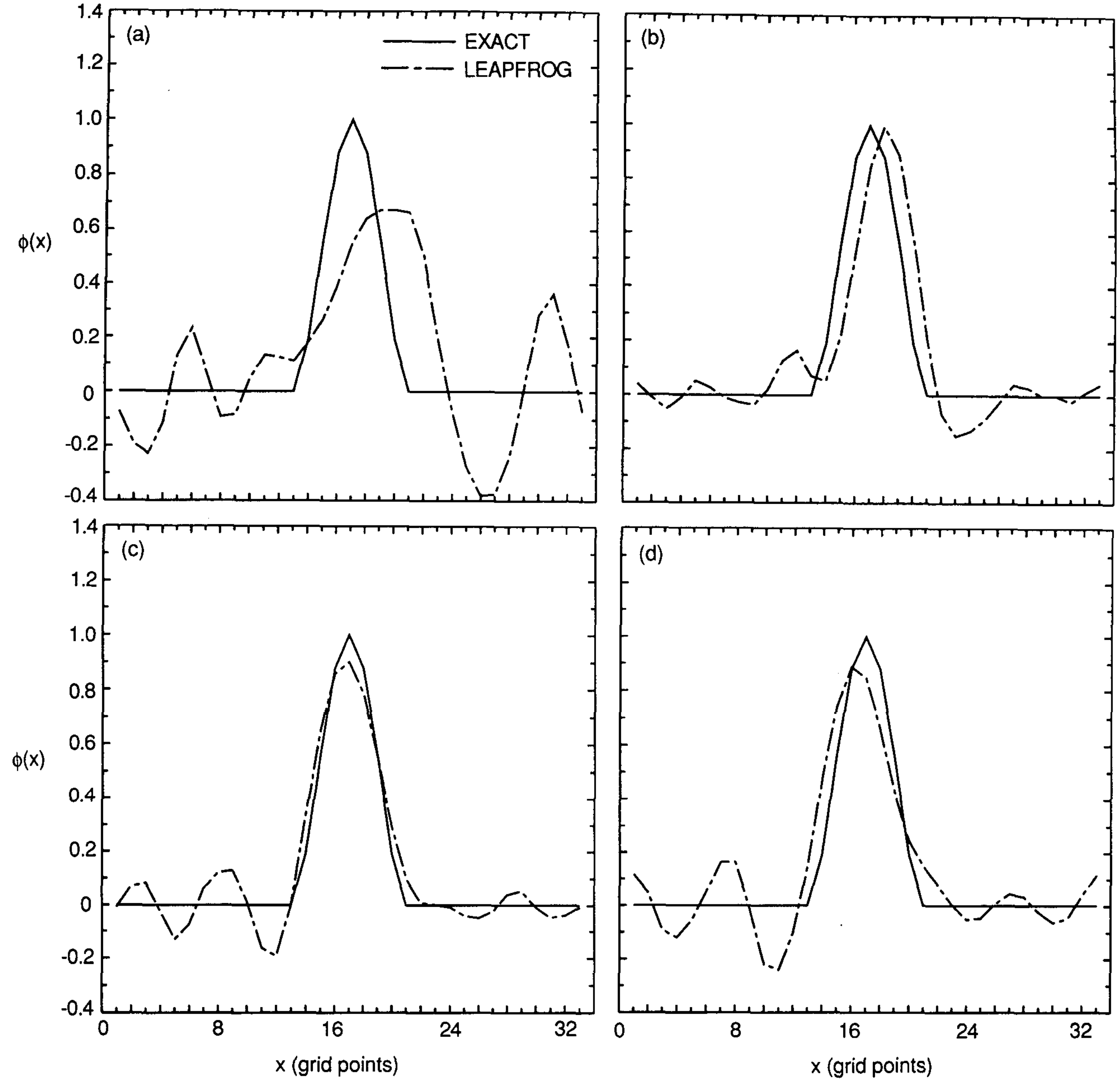

FIG. 3. Effect of leapfrog stepsize on the accuracy of fourth-order centered-difference solution to the advection equation. Shown are the exact and numerical solutions computed using Courant numbers of (a) 0.7272, (b) 0.5, (c) 0.3, and (d) 0.1. All results are for a nondimensional time of 3.

since the dissipation acting on the longest waves will be dominated by the second-order time filter. This problem can be avoided with third-order Adams–Bashforth time differencing because the dissipation in that scheme is equivalent to a fourth-order spatial filter.

4. Comparative tests of the third-order Adams–Bashforth method and the Asselin-filtered leapfrog scheme

In this section, several numerical tests will be presented in order to illustrate the improvements that can accompany the replacement of leapfrog time differences by the third-order Adams–Bashforth method.

a. Linear advection

The first numerical comparison will be for simulations of conservative-tracer transport by a constant-velocity flow. As in section 3, solutions to (23) are computed on the periodic domain $0 \leqslant x \leqslant 1$ using a wind speed $c = 1/4$ and the initial distribution given by (25). The behavior of third-order Adams–Bashforth

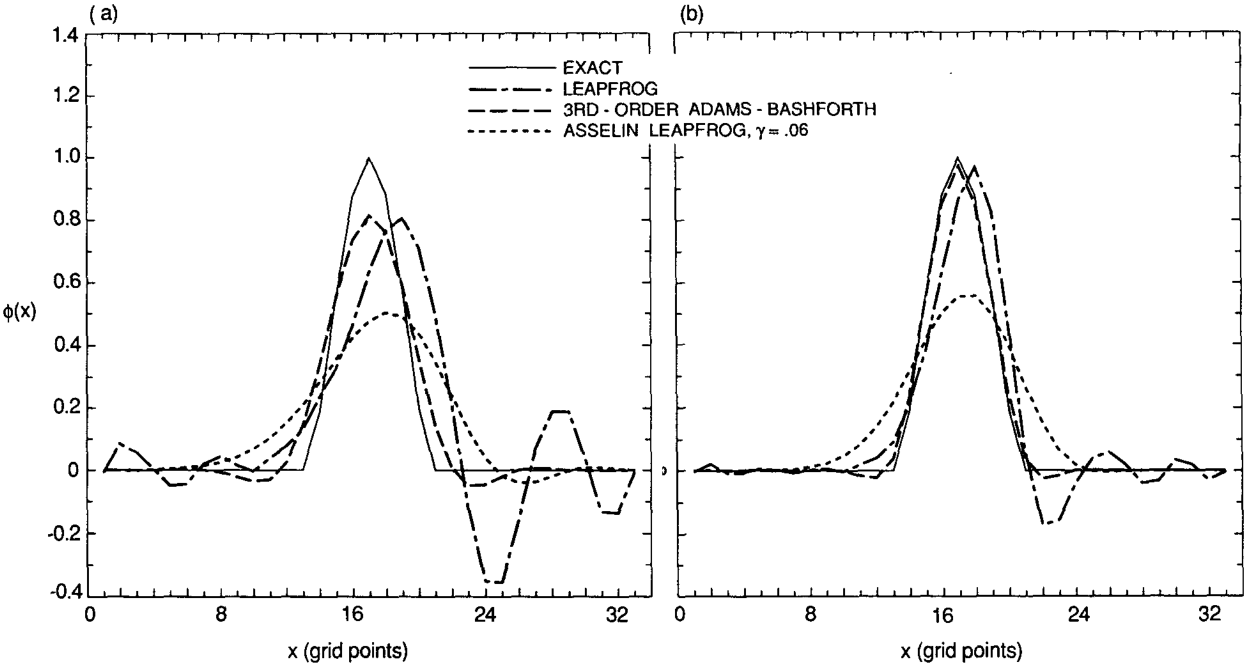

FIG. 4. Comparison of an exact solution to the advection equation with results obtained using leapfrog, Adams–Bashforth, and Asselin-filtered leapfrog time differencing in a spectral model: (a) for nondimensional time $t_{nd} = 20$ and Courant number $\mu = 0.2$, and (b) for $t_{nd} = 40$ and $\mu = 0.08$.

and leapfrog time differencing will be compared in two different models: a spectral model and a fourth-order finite-difference model. In the following tests, the leapfrog solution was started by taking a single forward time step; the Adams–Bashforth solution was started by taking a forward time step, followed by a second-order Adams–Bashforth step.

Figure 4 shows the numerical solutions computed using a 32-wavenumber spectral representation of the spatial derivatives. The four curves represent the exact solution and results obtained using the third-order Adams–Bashforth method, the unfiltered leapfrog method, and the leapfrog method with Asselin-filtering coefficient $\gamma = 0.06$. The value $\gamma = 0.06$ was selected because it is typical of the filtering employed in global spectral models. The solutions plotted in Fig. 4a were calculated using a Courant number of 0.2, which implies the time step is 63% of the maximum permitted by the stability requirements of the unfiltered leapfrog scheme. Figure 4a, which shows results at a nondimensional time of 20, indicates that even weak time filtering strongly damps the solution. The unfiltered leapfrog solution retains more amplitude, but the large, frequency-dependent phase-speed error of the leapfrog scheme disperses the solution, creating substantial negative concentrations ahead of the main spike. The third-order Adams–Bashforth scheme exhibits negligible phase-speed error; its amplitude error is slightly less than the error in the unfiltered leapfrog solution. Although the unfiltered leapfrog scheme is nominally free from "amplitude" error, the dispersion generated by leapfrog differencing reduces the maxima in Fig. 4 faster than the $O[(\omega\Delta t)^4]$ damping produced by the third-order Adams–Bashforth method.

Figure 4b shows results obtained with the time step reduced so that $\mu = 0.08$. Although the simulations in Fig. 4b have been run out to a nondimensional time of 40 (twice the time interval over which the integrations were performed in Fig. 4a), the errors in Fig. 4b are considerably smaller than those in Fig. 4a. Nevertheless, the relative performance of the three schemes is unchanged. Asselin filtering with $\gamma = 0.06$ still produces heavy damping. Significant phase-speed errors and negative concentrations are produced by the leapfrog scheme. The third-order Adams–Bashforth scheme produces the smallest amplitude and phase-speed errors; indeed the Adams–Bashforth solution looks nearly perfect.

The spectral method is essentially an infinite-order finite-difference scheme. How do the preceding results change if the accuracy of the spatial differences are reduced to fourth-order? The answer appears in Figs. 5 and 6. Figure 5 shows computations performed using centered fourth-order differences [see (24)] with a spatial mesh size of $1/32$. The exact, third-order Adams–Bashforth, and two Asselin-filtered leapfrog solutions are shown at a nondimensional time of 3. The filtering coefficients are 0.06 and 0.2; the case $\gamma = 0.06$ may be compared with the preceding results; the value $\gamma = 0.2$ is typical of the filtering used in finite-difference me-

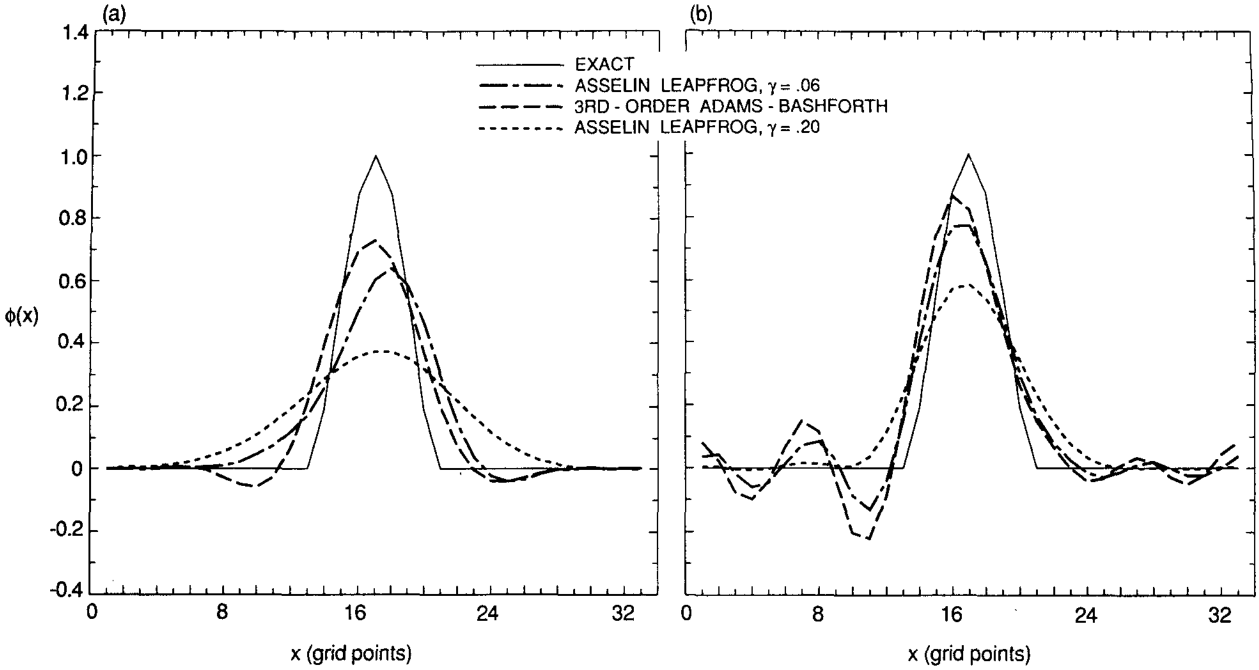

FIG. 5. Comparison of an exact solution to the advection equation with results obtained using Adams–Bashforth and Asselin-filtered leapfrog time differencing in a fourth-order finite-difference model at a nondimensional time of 3, for (a) $\mu = 0.5$ and (b) $\mu = 0.2$.

soscale models. The calculations shown in Fig. 5a were performed with a Courant number of 0.5, implying that the time step was 68%, the maximum allowed by the unfiltered leapfrog scheme. (This time step is also 84% of the maximum allowed by the filtered leapfrog scheme when $\gamma = 0.2$, and 95% of the maximum allowed by the third-order Adams–Bashforth method.) As per the discussion in section 4, when the time step is close to the stability limit, the phase-speed error introduced by fourth-order spatial differencing almost

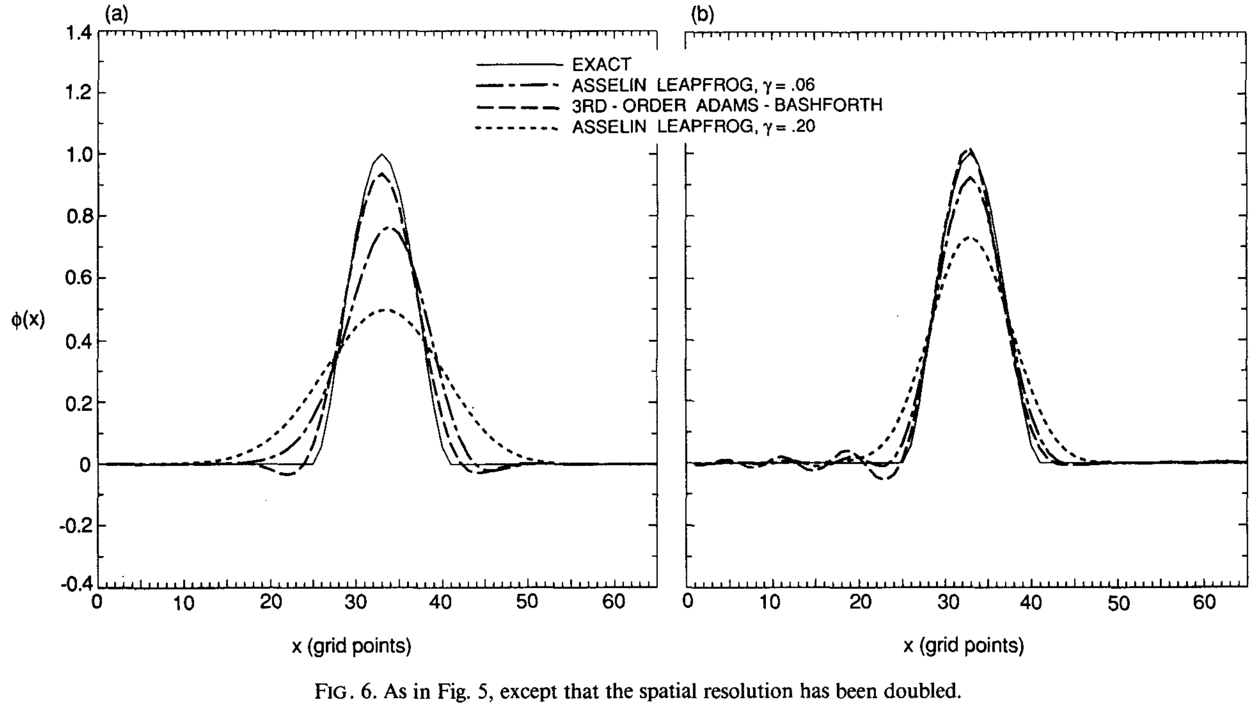

FIG. 6. As in Fig. 5, except that the spatial resolution has been doubled.

cancels that generated by the third-order Adams–Bashforth method. As a consequence, the phase error of the Adams–Bashforth solution in Fig. 5a is very small. On the other hand, when $\mu = 0.5$, the leading phase-speed error of the leapfrog method dominates the smaller, lagging phase-speed error produced by fourth-order differencing, and both leapfrog solutions move too fast. In addition, the Asselin time filter damps both leapfrog solutions more rapidly than the Adams–Bashforth solution; the damping is particularly severe when $\gamma = 0.2$.

Figure 5b indicates that when the Courant number is reduced to 0.2, the amplitude error in all solutions is reduced significantly. The third-order Adams–Bashforth scheme continues to produce the least dissipation. Asselin filtering with $\gamma = 0.2$ generates heavy damping. In contrast to Fig. 5a, all three numerical solutions lag the exact solution. The leading phase-speed error introduced by all three time differencing schemes is reduced as μ is decreased, and as a result the lagging phase-speed error generated by the fourth-order spatial difference dominates the total phase error. The phase error in the Adams–Bashforth solution is slightly worse than that in either leapfrog solution because, at the relatively short wavelengths that dominate this test problem, the larger phase-speed errors introduced by leapfrog time differencing are better able to compensate the error introduced by spatial differencing.

As expected, the finite-difference results are less accurate than those obtained with the spectral method. A considerable increase in the quality of the finite-difference solution can be achieved by doubling the horizontal resolution. The degree of improvement may be assessed by comparing Fig. 5 with Fig. 6. The simulations plotted in Fig. 6 are identical to those in Fig. 5, except that the horizontal mesh spacing and the time step were halved. As in Fig. 5, the third-order Adams–Bashforth method produces less amplitude error than the Asselin-filtered leapfrog scheme. The phase error in the Adams–Bashforth solution is better than that of the leapfrog method in Fig. 6a and slightly worse in Fig. 6b. In all cases, the improvement in the solution accompanying the increase in resolution is significantly greater for the third-order Adams–Bashforth method. This last result is not surprising; the Adams–Bashforth solution should show greater improvement because it is a higher-order method.

b. Nonlinear convection

Now consider a situation in which the leapfrog solution must be filtered in order to avoid spurious amplification of the computational mode. Lorenz (1963) has shown that the equations governing two-dimensional Rayleigh convection can be truncated to three components whose amplitudes are governed by the system of equations

$$\dot{X} = \sigma Y - [\sigma X], \tag{28}$$

$$\dot{Y} = -XZ + rX - [Y], \tag{29}$$

$$\dot{Z} = XY - [bZ] \tag{30}$$

where the dot represents differentiation with respect to a nondimensional time, and the square brackets are used to indicate the diffusion terms. The behavior of this system will be examined for the case $\sigma = 12$, $r = 12$, $b = 6$, and initial conditions $(X_0, Y_0, Z_0) = (-10, -10, 25)$. From the standpoint of predictability, this is an uninteresting case—the solution decays with time; there is no sensitive dependence on the initial conditions. Chaotic solutions to (28)–(30) are inappropriate for these tests because they would exaggerate the difference between the various numerical schemes. The goal is to examine a nonchaotic solution illustrating the susceptibility of leapfrog time differencing to instability.

Equations (28)–(30) were integrated using a fourth-order Runge–Kutta scheme (see Table 1), the third-order Adams–Bashforth scheme, and the Asselin-filtered leapfrog method. The application of the first two schemes to these equations is straightforward. The leapfrog scheme, however, is not suitable for the direct integration of (28)–(30). Following the practice in most mesoscale convective models, the diffusion terms, enclosed by the square brackets in (28)–(30), were integrated using lagged forward differences [as per (21)]. The remaining terms in (28)–(30) represent advection and buoyancy forces and, again following the practice in most mesoscale models, these were differenced using the leapfrog scheme. The Asselin time filter was set to $\gamma = 0.2$.

The numerical solution for X is plotted in Fig. 7. The results in Fig. 7a were obtained using a time step of 0.03; a 0.015 time step was used in Fig. 7b. The fourth-order Runge–Kutta solution is essentially identical in both Figs. 7a,b, suggesting that the Runge–Kutta scheme has converged to the exact solution. Thus, the Runge–Kutta solution provides a reference against which the other methods may be compared. The error in the third-order Adams–Bashforth solution is much smaller than the error in the Asselin-filtered leapfrog solution; and as might be expected from a higher-order scheme, the error in the Adams–Bashforth result decreases more rapidly than that in the Asselin-filtered leapfrog solution as the time step is halved. The least accurate solutions are those produced by unfiltered leapfrog differencing, which rapidly generates a $2\Delta t$ computational mode. In these simulations the computational mode does not grow without bound; it remains at a steady amplitude as the low-frequency oscillation in the physical mode decays. It is easy to find other values of (σ, r, b) and other initial conditions that generate significant $2\Delta t$ noise in the leapfrog solution. Examples that produce catastrophic instability are, however, rare. (The solution will "blow up" if σ

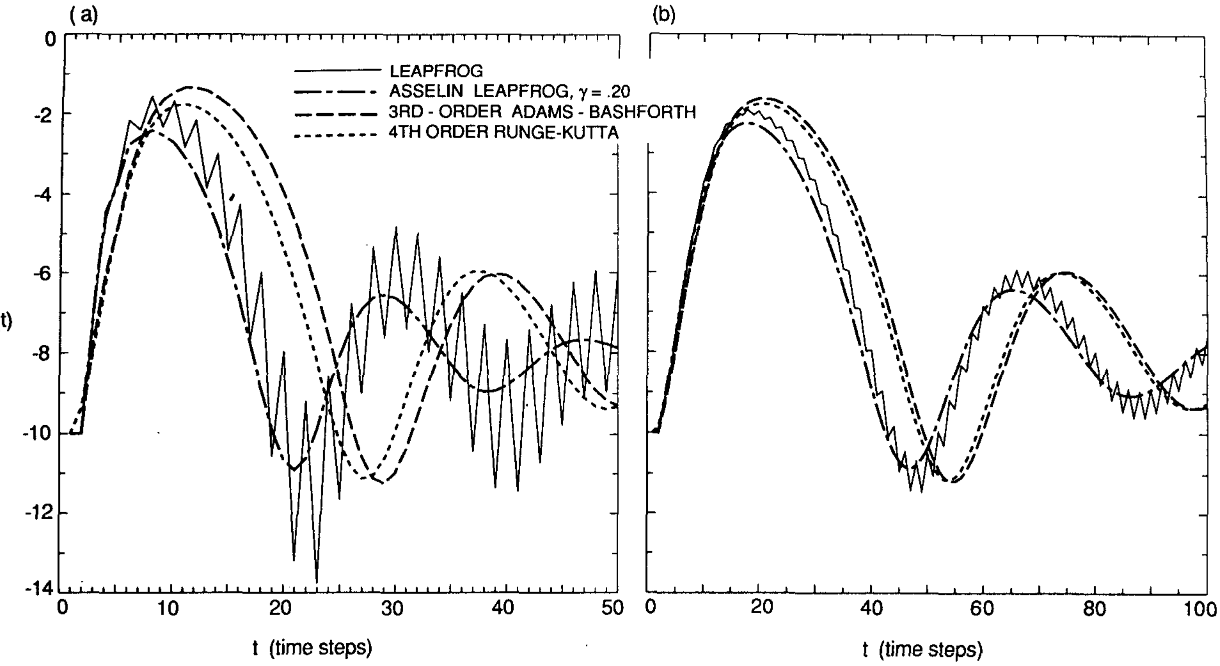

FIG. 7. Numerical approximations to a nonchaotic solution of the Lorenz system. The variable plotted is X; the time step is (a) 0.03 and (b) 0.015.

and r are changed to 10, and $\Delta t = 0.03$.) Practical experience with less highly truncated convective models suggests that time splitting instabilities will inevitably arise in leapfrog integrations. The highly truncated nature of Lorenz's model apparently prevents the additional interactions responsible for the continued amplification of the computational mode.

The initial development of the computational mode in Fig. 7 is dependent on nonlinear processes. The importance of nonlinearity is demonstrated by Fig. 8, which shows an integration equivalent to that in Fig. 7a, except that the Lorenz equations were linearized about (X_0, Y_0, Z_0). The qualitative character of the linear solution is similar to the nonlinear solution in that both are damped oscillations, yet the unfiltered leapfrog solution does not develop a computational mode. Other larger-amplitude linear solutions also failed to generate a discernable computational mode. Although there is no time splitting in the linearized problem, the leapfrog solution is, once again, less accurate than that obtained with the third-order Adams–Bashforth method.

5. Other time differencing schemes

Are there other high-order time differencing schemes that might be even more attractive than the third-order Adams–Bashforth method? Are there second-order schemes that provide a better way to control time splitting than Asselin–Robert time filtering? A partial answer to these questions can be provided by a brief survey of other ordinary-differential-equation solvers. In order to organize the survey in a systematic manner, it is helpful to describe the relationship between the overall truncation error of an ODE solver and the amplitude and phase-speed errors introduced by that solver when it is used to model nondissipative oscillatory phenomena.

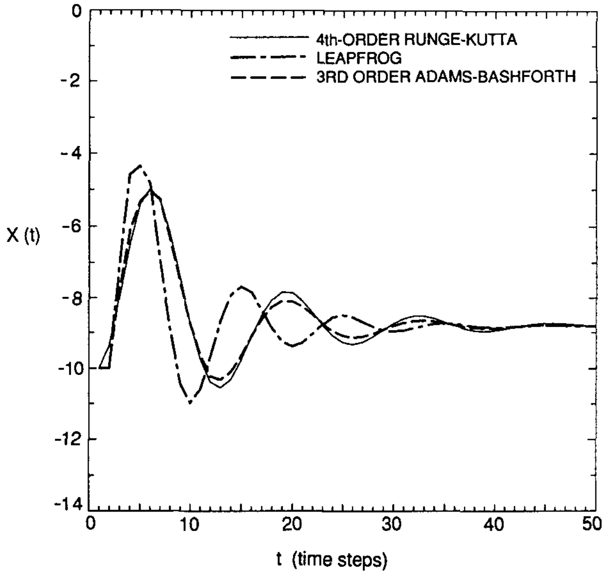

FIG. 8. As in Fig. 7a, except that the governing equations have been linearized.

Proposition: if the oscillation Eq. (6) is integrated using a linear finite-difference scheme and if the truncation error of the resulting finite-difference approximation to the oscillation equation is order r, then as $\omega\Delta t \rightarrow 0$ the amplitude error in the numerical solution is

$$1 + O[(\omega\Delta t)^n], \quad \text{where} \quad \begin{cases} n = r + 1, & \text{if } r \text{ is odd;} \\ n \geqslant r + 2, & \text{if } r \text{ is even;} \end{cases} \tag{31}$$

and the phase-speed error is

$$1 + O[(\omega\Delta t)^m], \quad \text{where} \quad \begin{cases} m \geqslant r + 1, & \text{if } r \text{ is odd;} \\ m = r, & \text{if } r \text{ is even.} \end{cases} \tag{32}$$

The proof of this proposition, which was partially anticipated by Takacs (1985), appears in the Appendix. It follows that second-order time differencing produces second-order phase-speed error and fourth-order (or better) amplitude error. Switching to a third-order scheme will not improve the fourth-order amplitude error, but will reduce the phase error to fourth order or better.

A summary of the properties of several elementary methods appears in Table 1. Note that for each entry, the relationship between the total truncation error and the phase speed and amplitude error is consistent with the preceding proposition. In Table 1, the column labeled "efficiency factor" lists the maximum stable time step with which the oscillation equation can be integrated, divided by the number of function evaluations per time step.[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. The storage factor for implicit methods can vary from problem to problem, depending on the algorithm used to solve the implicit system. Therefore, storage factors are not provided for the implicit methods in Table 1.

The storage factors given in Table 1 are upper limits that allow each method to be programmed in a 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, consider the third-order Adams–Bashforth method. When integrating partial differential equations, 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 is much smaller than the full array required to hold a complete set of ϕ^n. This procedure may be used to reduce the storage requirement of the third-order Adams–Bashforth method from four full arrays to three full arrays and a smaller, temporary storage array.

A survey of Table 1 reveals that, whereas the first- and second-order Adams–Bashforth and Runge–Kutta techniques[3] are unstable, the third- and fourth-order versions of these schemes are stable. Conversely, the implicit Adams–Moulton schemes are stable at the first and second orders (the backward and trapezoidal methods), but unstable at the third and fourth orders. Table 1 also indicates that, among the explicit methods, the efficiency factor of the third-order Adams–Bashforth scheme is second only to the leapfrog scheme and its Asselin-filtered variant. This high efficiency factor implies that, with the exception of the leapfrog scheme, the third-order Adams–Bashforth method is capable of producing a stable solution with less computational effort than any other explicit scheme listed in Table 1. One would not, however, select the third-order Adams–Bashforth method solely on the basis of its high efficiency factor. If the efficiency factor was all that mattered, one would use backward differencing or the trapezoidal method and a very large time step. The other important consideration is accuracy and as defined in this study the efficiency factor only indicates the minimum work required to advance the solution in a stable manner; it does not measure the work required to achieve a given accuracy.

The preceding sections of this paper have demonstrated that the third-order Adams–Bashforth method can be significantly more accurate than the Asselin-filtered leapfrog scheme. Most of the following discussion will be devoted to other higher-order methods, but first consider the stable explicit second-order methods listed in Table 1. Some improvement relative to Asselin-filtered leapfrog differencing might be obtained from the leapfrog-trapezoidal method (Kurihara 1965) or the Magazenkov scheme because they both damp the computational mode without the loss of $O[(\Delta t)^2]$ accuracy introduced by the Asselin filter. The leapfrog-trapezoidal scheme is an iterative method however, and if a given numerical application will accommodate an iterative method, the (low storage) fourth-order Runge–Kutta scheme might be an even

[2] The definition of the efficiency factor is based on the assumption that the bulk of the computational effort is associated with the evaluation of the function that determines the time derivative. This assumption is appropriate when integrating all but the simplest partial difference equations.

[3] The first-order Adams–Bashforth and Runge–Kutta schemes are both equivalent to forward differencing.

TABLE 1. Comparison of time differencing schemes. The amplitude, phase speed, and time step limitations are those associated with the application of each scheme to the oscillation Eq. (4). Storage and efficiency factors are defined in the text. Here, $h \equiv \Delta t$ and $p \equiv \omega h$.

Method	Order	Formula	Storage factor	Efficiency factor	Amplitude error	Phase error	Maximum $\omega\Delta t$
Forward (Adams–Bashforth)	1	$\phi^{n+1} = \phi^n + hF(\phi^n)$	2	0	$1 + \frac{p^2}{2}$	$1 - \frac{p^2}{3}$	0
Backward (Adams–Moulton)	1	$\phi^{n+1} = \phi^n + hF(\phi^{n+1})$	Implicit	∞	$1 - \frac{p^2}{2}$	$1 - \frac{p^2}{3}$	∞
Matsuno	1	$\phi^* = \phi^n + hF(\phi^n)$ $\phi^{n+1} = \phi^n + hF(\phi^*)$	2	.5	$1 - \frac{p^2}{2}$	$1 + \frac{2}{3}p^2$	1
Asselin-filtered leapfrog	1	$\phi^{n+1} = \overline{\phi^{n-1}} + 2hF(\phi^n)$ $\overline{\phi^n} = \phi^n + \gamma(\overline{\phi^{n-1}} - 2\phi^n + \phi^{n+1})$	3	<1	$1 - \frac{\gamma}{2(1-\gamma)}p^2$	$1 + \frac{1+2\gamma}{6(1-\gamma)}p^2$	<1
Leapfrog	2	$\phi^{n+1} = \phi^{n-1} + 2hF(\phi^n)$	2	1	1	$1 + \frac{p^2}{6}$	1
Runge–Kutta (Williamson/Huen)	2	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1$ $q_2 = hF(\phi_1) - q_1, \quad \phi^{n+1} = \phi_1 + q_2/2$	2	0	$1 + \frac{p^4}{8}$	$1 + \frac{p^2}{6}$	0
Adams–Bashforth	2	$\phi^{n+1} = \phi^n + \frac{h}{2}[3F(\phi^n) - F(\phi^{n-1})]$	3	0	$1 + \frac{p^4}{4}$	$1 + \frac{5}{12}p^2$	0
Adams–Moulton (Trapizoidal)	2	$\phi^{n+1} = \phi^n + \frac{h}{2}[F(\phi^{n+1}) + F(\phi^n)]$	Implicit	∞	1	$1 - \frac{p^2}{12}$	∞
Leapfrog, then Adams–Bashforth (Magazenkov)	2	$\phi^n = \phi^{n-2} + 2hF(\phi^{n-1})$ $\phi^{n+1} = \phi^n + \frac{h}{2}[3F(\phi^n) - F(\phi^{n-1})]$	3	.67	$1 - \frac{p^4}{4}$	$1 + \frac{p^2}{6}$	.67
Leapfrog-Trapizoidal (Kurihara)	2	$\phi^* = \phi^{n-1} + 2hF(\phi^n)$ $\phi^{n+1} = \phi^n + \frac{h}{2}[F(\phi^n) + F(\phi^*)]$	3	.71	$1 - \frac{p^4}{4}$	$1 - \frac{p^2}{12}$	1.41
Young's method A	2	$\phi_1 = \phi^n + hF(\phi^n)/2$ $\phi_2 = \phi_1 + hF(\phi_1)/2$ $\phi^{n+1} = \phi^n + hF(\phi_2)$	3	0	$1 + \frac{p^6}{128}$	$1 + \frac{p^2}{24}$	0
Runge–Kutta (Williamson)	3	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1/3_1$ $q_2 = hF(\phi_1) - 5q_1/9, \quad \phi_2 = \phi^1 + 15q_2/16$ $q_3 = hF(\phi_2) - 153q_2/128, \quad \phi^{n+1} = \phi_2 + 8q_3/15$	2	.58	$1 - \frac{p^4}{24}$	$1 + \frac{p^4}{30}$	1.73
ABM predictor-corrector	3	$\phi^* = \phi^n + \frac{h}{2}[3F(\phi^n) - F(\phi^{n-1})]$ $\phi^{n+1} = \phi^* + \frac{5h}{12}[F(\phi^*) - 2F(\phi^n) + F(\phi^{n-1})]$	4	.60	$1 - \frac{19}{144}p^4$	$1 + \frac{1243}{8640}p^4$	1.20
Adams–Moulton	3	$\phi^{n+1} = \phi^n + \frac{h}{12}[5F(\phi^{n+1}) + 8F(\phi^n) - F(\phi^{n-1})]$	Implicit	0	$1 + \frac{p^4}{24}$	$1 - \frac{11}{720}p^4$	0

TABLE 1. (*Continued*)

Method	Order	Formula	Storage factor	Efficiency factor	Amplitude error	Phase error	Maximum $\omega\Delta t$
Adams–Bashforth	3	$\phi^{n+1} = \phi^n + \frac{h}{12}[23F(\phi^n) - 16F(\phi^{n+1}) + 5F(\phi^{n-2})]$	4	.72	$1 - \frac{3}{8}p^4$	$1 + \frac{289}{720}p^4$	0.72
Runge–Kutta (Classical)	4	$k_0 = hF(\phi^n)$ $k_1 = hF(\phi^n + k_0/2)$ $k_2 = hF(\phi^n + k_1/2)$ $k_3 = hF(\phi^n + k_2)$ $\phi^{n+1} = \phi^n + \frac{1}{6}(k_0 + 2k_1 + 2k_2 + k_3)$	3*	.70	$1 - \frac{p^6}{144}$	$1 - \frac{p^4}{120}$	2.82
ABM predictor-corrector	4	$\phi^* = \phi^n + \frac{h}{12}[23F(\phi^n) - 16F(\phi^{n-1}) + 5F(\phi^{n-2})]$ $\phi^{n+1} = \phi^* + \frac{3h}{8}[F(\phi^*) - 3F(\phi^n) + 3F(\phi^{n-1}) - F(\phi^{n-2})]$	5	.59	$1 - \frac{265}{1536}p^6$	$1 - \frac{329}{2880}p^4$	1.18
Adams–Moulton	4	$\phi^{n+1} = \phi^n + \frac{h}{24}[9F(\phi^{n+1}) + 19F(\phi^n) - 5F(\phi^{n-1}) + F(\phi^{n-2})]$	Implicit	0	$1 + \frac{p^6}{48}$	$1 + \frac{19}{720}p^4$	0
Adams–Bashforth	4	$\phi^{n+1} = \phi^n + \frac{h}{24}[55F(\phi^n) - 59F(\phi^{n-1}) + 37F(\phi^{n-2}) - 9F(\phi^{n-3})]$	5	.43	$1 - \frac{13}{24}p^6$	$1 - \frac{251}{720}p^4$	.43

*Storage factor of three is achieved following the algorithm of Blum (1962).

better choice since it is higher order and has a similar efficiency factor. Magazenkov's method is possibly the most attractive second-order alternative to Asselin-filtered leapfrog time differencing. Magazenkov (1980) suggested alternating each leapfrog step with a second-order Adams–Bashforth step; the result is a fully explicit, stable, second-order scheme with a damped computational mode. Note, however, that neither of these second-order schemes achieve quite the same efficiency factor as the third-order Adams–Bashforth method. The remainder of this section will, therefore, be devoted to the consideration of other higher-order schemes.

The third-order Runge–Kutta scheme is stable and has less amplitude and phase-speed error than the Adams–Bashforth method. However, since the Runge–Kutta scheme requires three function evaluations per time step, it may best be compared with the Adams–Bashforth method by assuming the Adams–Bashforth integration proceeds with one-third the time step of the Runge–Kutta scheme. The adjective "equal work" will be used to describe this type of comparison in which the time step used with each method is adjusted to hold constant the ratio of the step size to the number of function evaluations per step. When the two schemes are compared on this equal-work basis, the amplitude and phase-speed errors of the third-order Adams–Bashforth method are much less than those of the Runge–Kutta method. This accuracy comparison is based solely on the truncation error however, so it only applies to well-resolved solutions. In order to assess the relative accuracy of the two methods when poorly resolved high-frequency modes are present, the spectral-model advection test presented in Fig. 4 was repeated using a third-order Runge–Kutta integrator with $\mu = ½$, and the third-order Adams–Bashforth method with μ set to the equal-work value of ⅙. The Courant number of ½ is near the maximum stable time step permitted by the third-order Runge–Kutta scheme. The two solutions are compared at a nondimensional time of 20 in Fig. 9, which demonstrates that the Adams–Bashforth method produces a significantly better equal-work solution.

Although the third-order Runge–Kutta method is less efficient and less accurate than the third-order Adams–Bashforth method, it is noteworthy because of its low storage requirements, which are actually smaller than those of the Asselin-filtered leapfrog scheme. The third-order Runge–Kutta method presented in Table 1 is just one member of a large family of possible Runge–Kutta schemes. All third-order Runge–Kutta schemes produce the same truncation error if they are used to integrate the simple oscillation equation. Different Runge–Kutta formulae do produce different truncation errors in more complex nonlinear problems and most third-order formulae have a storage factor of 3. Willimson (1980) discovered a small class of low-storage third-order Runge–Kutta methods and recommended the particular scheme shown in Table 1. The Williamson–Runge–Kutta method appears to be distinctly superior to two other time differencing techniques that have appeared in the meteorological literature—Young's method A (Young 1968) and the Lorenz three-cycle method (Lorenz 1971). All three methods are iterative schemes requiring three function evaluations per time step. The Williamson–Runge–Kutta method is preferred over Young's method A because the latter is unstable, lower order, and seems to require more storage. The amplification factors for the third-order Runge–Kutta method and Young's method A are compared in Fig. 10. The Williamson–Runge–Kutta method is similar to the Lorenz three-cycle method in that both require the same storage and, when used to integrate the oscillation equation, both generate the same truncation error. However, the Williamson–Runge–Kutta method is truly third order, whereas the three-cycle method is just second order. The three-cycle method achieves third-order accuracy only in the special case where the governing differential equation is linear (Lorenz 1971).

The other stable third-order method appearing in Table 1 is the Adams–Bashforth–Moulton predictor corrector. It is curious that two unstable schemes (second-order Adams–Bashforth and third-order Adams–Moulton) combine to yield a stable predictor-corrector formula. As was the case with the third-order Runge–Kutta scheme, the Adams–Bashforth–Moulton predictor corrector is inferior to the Adams–Bashforth method in equal-work comparisons. Moreover, the Adams–Bashforth–Moulton predictor corrector requires the same storage as the third-order Adams–Bashforth method.

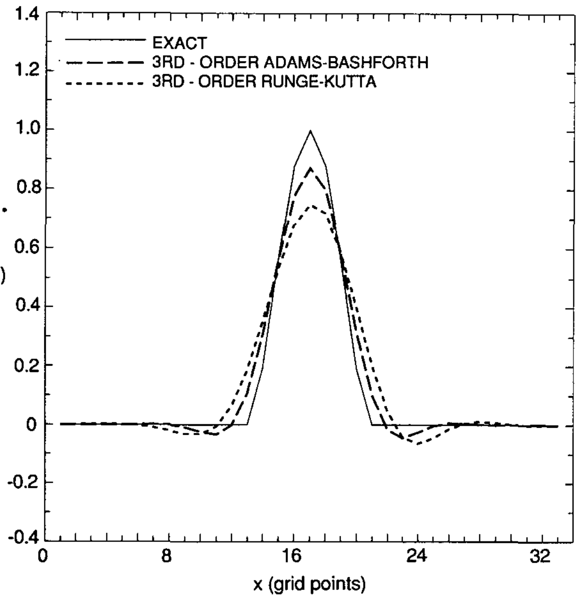

FIG. 9. Equal-work comparison of a third-order Runge–Kutta solution (short-dashed line) computed using $\mu = 1/2$, with a third-order Adams–Bashforth solution (long-dashed lined) calculated using $\mu = 1/6$. Otherwise identical to Fig. 4a.

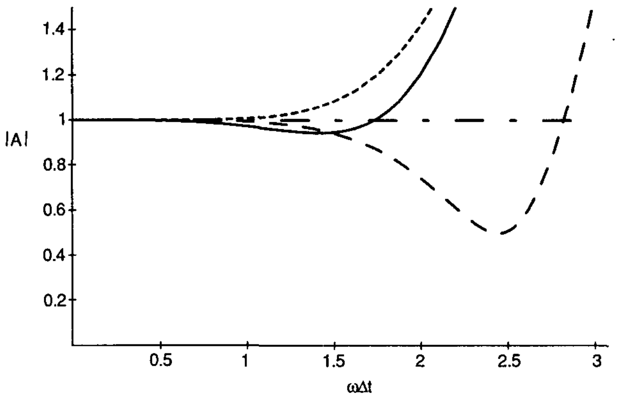

FIG. 10. Magnitude of the amplification factor plotted as a function of $\omega\Delta t$, for the third-order Runge–Kutta scheme (solid line), the fourth-order Runge–Kutta scheme (long-dashed line), Young's method A (short-dashed line), and the exact solution (dash-dotted line).

Turning to the fourth-order methods, inspection of Table 1 shows that the fourth-order Runge–Kutta method has the lowest storage factor[4] and the highest efficiency factor of all listed fourth-order schemes. Being fourth order, it will produce $O[(\omega\Delta t)^6]$ amplitude errors, which will be smaller than the $O[(\omega\Delta t)^4]$ amplitude errors produced by the third-order Adams–Bashforth method whenever $\omega\Delta t$ is sufficiently small. This improvement in amplitude error may be lost however, if the method is used in combination with the fourth-order spatial filters commonly employed to control short-wavelength noise in atmospheric models. At a fixed value of $\omega\Delta t$, the phase-speed error of the fourth-order Runge–Kutta scheme is less than that of the third-order Adams–Bashforth method; however, if the comparison is made on an equal-work basis (four Adams–Bashforth steps per each Runge–Kutta step), the phase-speed error of the Runge–Kutta scheme is five times worse that the error in the Adams–Bashforth scheme. In addition, the lagging phase-speed error of the fourth-order Runge–Kutta scheme will reinforce the lagging phase-speed error introduced by any spatial differencing. The preceding discussion applies only to the errors in well-resolved solutions. Some idea of the quality of a poorly resolved solution, obtained using time steps near the stability limit, is provided by the plot of amplification factor versus $\omega\Delta t$ in Fig. 10. The fourth-order Runge–Kutta method strongly damps the poorly resolved modes; its region of high accuracy is similar to the region in which the *third*-order Runge–Kutta scheme is both accurate and stable. Thus, if one

[4] A storage factor of three is achieved using the algorithm of Blum (1962). In order to save space, the classical fourth-order Runge–Kutta method is listed in Table 1.

wishes to accurately represent the highest frequencies, the maximum step selected for a fourth-order Runge–Kutta integration should be similar to the maximum step used in the third-order Runge–Kutta scheme. Nevertheless, in some situations one might welcome the opportunity to damp the highest-frequency oscillations and in those circumstances, one could take full advantage of the large stable time step permitted by the fourth-order Runge–Kutta method. An additional assessment of the relative accuracy of the fourth-order Runge–Kutta and third-order Adams–Bashforth methods is provided by the equal-work advection test shown in Fig. 11. Like Figs. 4 and 9, Fig. 11 shows results obtained by integrating a spectral advection model to a nondimensional time of 20. The fourth-order Runge–Kutta solution was computed using $\mu = 0.8$, and, in this equal-work test, the third-order Adams–Bashforth solution was calculated using $\mu = 0.2$. The fourth-order Runge–Kutta solution retains slightly more amplitude, but exhibits larger dispersion and phase-speed errors than the third-order Adams–Bashforth result. On balance, the third-order Adams–Bashforth solution looks the best.

One other scheme worthy of discussion is the Lorenz four-cycle method (Lorenz 1971). Like the previously mentioned three-cycle scheme, the four-cycle scheme was designed to achieve high accuracy for linear differential equations while conserving storage. When applied to the oscillation equation, the four-cycle scheme produces truncation errors identical to those of a fourth-order Runge–Kutta method. Yet when applied to nonlinear differential equations, the four-cycle scheme is second-order. The advantage of the four-cycle scheme is that its storage factor is only two. Although the four-cycle scheme will perform very well on the linear test problem, because of its relatively lower order it is unlikely to be a better general purpose scheme that the low-storage, higher-order Runge–Kutta formulae and the third-order Adams–Bashforth method.

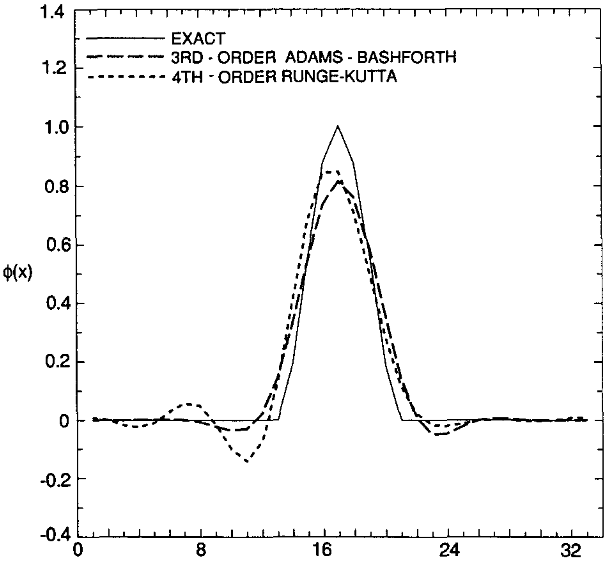

FIG. 11. Equal-work comparison of a fourth-order Runge–Kutta solution (short-dashed line) computed using $\mu = 0.8$, with a third-order Adams–Bashforth solution (long-dashed line) calculated using $\mu = 0.2$. Otherwise identical to Figs. 4a and 9.

In summary, the third-order Adams–Bashforth method can produce a stable solution to the oscillation equation with fewer function evaluations than any of the other third- or fourth-order methods listed in Table 1. The third-order Adams–Bashforth method is also more accurate than the other third-order methods when compared on an equal work basis. The most attractive fourth-order method appears to be the (low storage) Runge–Kutta scheme. Being fourth order, this scheme will exhibit less overall truncation error than the third-order Adams–Bashforth method as $\Delta t \rightarrow 0$. When applied to the oscillation equation, the fourth-order accuracy of the Runge–Kutta method manifests itself as a reduction in amplitude error. The phase-speed error of the fourth-order Runge–Kutta method is actually larger than that of the third-order Adams–Bashforth method when the two are compared on an equal-work basis.

6. Use of the Adams–Bashforth method in semi-implicit schemes

Many large meteorological models employ semi-implicit time differencing, in which the Asselin-filtered leapfrog scheme is used to represent the advection terms while the pressure gradient and divergence terms are integrated using the trapezoidal method. The basic properties of this technique may be illustrated by the following semi-implicit approximation to the oscillation Eq. (6):

$$\frac{\phi^{n+1} - \phi^{n-1}}{2\Delta t} = i\omega_1\phi^n + i\omega_2\left(\frac{\phi^{n+1} + \phi^{n-1}}{2}\right). \quad (33)$$

Here $\omega = \omega_1 + \omega_2$, and the Asselin filter is neglected. Kwizak and Robert (1971) have shown that (33) will be stable whenever

$$\omega_1{}^2\Delta t^2 \leqslant 1 + \omega_2{}^2\Delta t^2. \quad (34)$$

A sufficient condition for the satisfaction of (34) is $|\omega_1| \leqslant |\omega_2|$. In atmospheric applications, ω_1 is identified with those frequencies characteristic of advection, ω_2 is identified with the frequencies characteristic of gravity wave propagation, and the condition $|\omega_1| \leqslant |\omega_2|$ is almost always satisfied. Since the accurate simulation of high-speed gravity waves is thought to be unnecessary in large-scale atmospheric models, use of the semi-implicit scheme allows such models to be efficiently integrated with a large time step that would violate the CFL condition for gravity waves ($|\omega_2\Delta t| < 1$) and lead to instability in a fully explicit scheme.

One might expect to obtain a more accurate semi-implicit scheme by replacing the leapfrog difference in (33) with a third-order Adams–Bashforth formula; in which case

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \frac{i\omega_1}{12}(23\phi^n - 16\phi^{n-1} + 5\phi^{n-2}) + \frac{i\omega_2}{2}(\phi^{n+1} + \phi^n). \quad (35)$$

This modification not only reduces the truncation error in the ω_1 component of the solution, it also improves the accuracy of the ω_2 component by halving the step-size of the trapezoidal integration. Unfortunately, the robust stability of the original semi-implicit scheme is not shared by the Adams–Bashforth trapezoidal formula (35). The amplification factor for (35) satisfies

$$\left(1 - \frac{i\omega_2\Delta t}{2}\right)A^3 - \left(1 + \frac{23i\omega_1\Delta t}{12} + \frac{i\omega_2\Delta t}{2}\right)A^2 + \frac{4i\omega_1\Delta t}{3}A - \frac{5i\omega_1\Delta t}{12} = 0. \quad (36)$$

The magnitude of the three roots of (36) are plotted in Fig. 12 as a function of $\omega_1\Delta t$ for the case $\omega_2\Delta t = 2.0$. Note that the physical mode is always unstable. Stable integrations can be achieved by reducing $\omega_2\Delta t$, but there is little point in using a semi-implicit scheme unless the high-frequency modes can be stably integrated at Courant numbers ($\omega_2\Delta t$) much greater than unity.

7. Conclusions

An analysis of the third-order Adams–Bashforth method has been presented, together with a detailed comparison of the third-order Adams–Bashforth method with the widely used Asselin-filtered leapfrog scheme. Not surprinsingly, the third-order Adams–Bashforth method is more accurate than the leapfrog method: phase-speed errors are smaller and amplitude errors are considerably less than those generated by Asselin time filtering. The third-order Adams–Bashforth method is also superior to the leapfrog scheme in that it can be easily used to integrate diffusive and Rayleigh damping processes. The preceding comments notwithstanding, the greatest advantage of the third-order Adams–Bashforth method may simply be that it is a simple, efficient way to avoid leapfrog time-splitting instability.

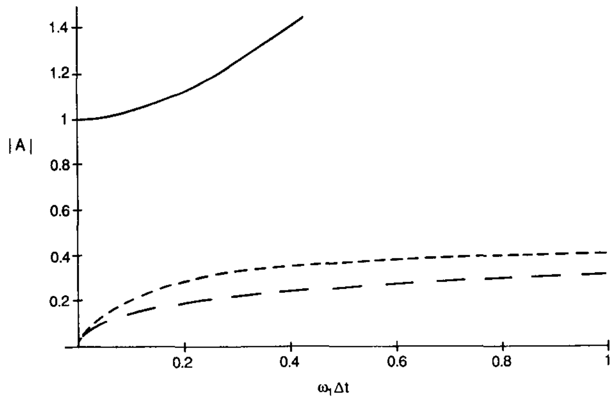

FIG. 12. Magnitude of the amplification factor for the Adams–Bashforth trapezoidal semi-implicit scheme (36), as a function of $\omega_1\Delta t$ for the case $\omega_2\Delta t = 2.0$. Solid line is the physical mode; dashed curves are computational modes.

As an alternative to Asselin time filtering, one might attempt to control leapfrog time splitting by periodically discarding data from the oldest time level, and restarting the integration using a two time-level method. The simple Euler or forward-difference method is commonly used to reinitialize leapfrog integrations. This scheme is easy to implement, but like Asselin filtering, 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. Restarting with a second-order scheme, such as the second-order Runge–Kutta method, would preserve the second-order accuracy of the overall integration, but the implementation of such a scheme is more complex, and most explicit second-order schemes are also subject to slowly growing instability. Both Asselin time filtering and the periodic reinitialization of the leapfrog solution involve arbitrary parameters: the filtering coefficient and the number of time steps between restarts. The influence of these parameter values on the numerical solution can be rather obscure. Use of the third-order Adams–Bashforth method eliminates the computational mode without the introduction of arbitrary parameters that need to be engineered for each application.

Other accurate time-integration techniques have also been surveyed. In connection with this survey, it has been demonstrated that if the truncation error of a difference scheme is order r and the scheme is applied to the oscillation equation, the amplitude error will be order $r + 1$ and the phase-speed error will be order $r + 1$, or better, provided that r is odd.[5] If r is even, the amplitude error will be order $r + 2$, or better, and the phase-speed error will be order r. Among the more attractive schemes revealed by the survey are the Magazenkov method and the low-storage third- and fourth-order Runge–Kutta methods. The Magazenkov method is an explicit, noniterative method that damps the computational mode while retaining second-order

[5] Note that the local truncation error of an $O(r)$ approximation to a differential equation is $O(r + 1)$ as per (A2). Thus, even though the local amplitude and phase-speed errors of an odd-order scheme might both be $O(r + 1)$, the scheme itself will be accurate only to $O(r)$.

accuracy. The storage requirements of the Magazenkov method are similar to those of the Asselin-filtered leapfrog scheme. If the numerical application will accommodate an iterative method and storage is critical, one might utilize the Williamson–Runge–Kutta scheme, which is both higher order and requires less storage than the Magazenkov and Asselin-filtered leapfrog schemes. The most accurate and most efficient iterative technique examined in this survey would appear to be the low-storage fourth-order Runge–Kutta scheme. The third-order Adams–Bashforth scheme is, however, more efficient than any of these schemes in that it requires the fewest function evaluations to stably integrate the solution over a given time interval. When it is compared on an equal-work basis (in which the ratio of the step size to the number of function evaluations per step is held constant by adjusting the time step used with each method), the third-order Adams–Bashforth method is more accurate than all the preceding schemes, except the fourth-order Runge–Kutta method. In addition, the third-order Adams–Bashforth method has a number of practical advantages that make it particularly suitable as a direct substitute for the leapfrog scheme. Unlike the Magazenkov method, the same algorithm is executed every time step. Unlike the Runge–Kutta schemes, there is no need to iterate, which can simplify the coding in complex modeling applications. It does appear, however, that the third-order Adams–Bashforth method is not a suitable substitute for leapfrog differencing in semi-implicit models.

Acknowledgments. The author has benefited from a review by Piotr Smolarkiewicz, and from discussions with Rashid Akmaev and Mark Borges. Joe Barsugli and Chris Bretherton introduced the author to *Mathematica.* Harry Edmon assisted with the computer graphics. Grace Gudmundson edited the manuscript and typed Table 1. Kay Dewar drafted the figures. This research was supported by NSF Grants ATM-8796281 and ATM-8914852.

APPENDIX

Proof of Proposition

Suppose that approximate solutions to the oscillation Eq. (6) are computed by the numerical scheme

$$\phi^{n+1} + L(\phi) = 0, \tag{A1}$$

where ϕ^n is the numerical approximation to $\psi(n\Delta t)$, and $L(\phi)$ is a linear function of ϕ at previous times. Substitution of ψ into (A1) yields an expression of the form

$$\psi^{n+1} + L(\psi) = O[(\Delta t)^{r+1}], \tag{A2}$$

where the right-hand side of (A2) is the local truncation error and r is the order of accuracy of the numerical scheme. It is assumed that ω is constant in (6) and that the coefficients of L are therefore independent of the time step, in which case solutions may be sought in the form (7). In order to prove the proposition, it is useful to first establish that for an rth order scheme

$$A = e^{i\omega\Delta t} + e_{r+1}(i\omega\Delta t)^{r+1} + e_{r+2}(i\omega\Delta t)^{r+2} + O[(\omega\Delta t)^{r+3}], \tag{A3}$$

where e_{r+1} and e_{r+2} are real coefficients. The derivation of (A3) is particularly easy for single-step methods and proceeds as follows. The truncation error, obtained by substituting Taylor series expansions for ψ into (A1), has the form

$$\psi^{n+1} + L(\psi^n) = \sum_{j=r+1}^{\infty} c_j \frac{d^j\psi^n}{dt^j}(\Delta t)^j = \psi^n \sum_{j=r+1}^{\infty} c_j(i\omega\Delta t)^j, \tag{A4}$$

where the c_j are determined by the finite-difference formula. Since ψ is the exact solution to the oscillation equation $\psi^{n+1} = e^{i\omega\Delta t}\psi^n$. Furthermore, in the case of a two-level scheme, $A\psi^n = -L(\psi^n)$ by the definition of the amplification factor. Using these relations, (A4) reduces to an expression of the form (A3) with $e_{r+1} = -c_{r+1}$ and $e_{r+2} = -c_{r+2}$. If the scheme is a multistep method, (A3) applies to the "principle root" (or the physical mode), and the derivation follows Gear (1971). The details of Gear's proof, which must be extended to $O[(\omega\Delta t)^{r+3}]$, are somewhat tedious and will not be repeated here.

The important difference between even- and odd-order schemes lies in the position of the imaginary factor i in the truncation error. The amplification factor for odd-order schemes can be expressed in the form

$$A = e^{i\omega\Delta t} + C_1(\omega\Delta t)^{r+1} + iC_2(\omega\Delta t)^{r+2} + O[(\omega\Delta t)^{r+3}], \tag{A5}$$

and the amplification factor for even-order methods can be written

$$A = e^{i\omega\Delta t} + iC_1(\omega\Delta t)^{r+1} + C_2(\omega\Delta t)^{r+2} + O[(\omega\Delta t)^{r+3}], \tag{A6}$$

where C_1 and C_2 are real constants. Evaluating $(AA^*)^{1/2}$, it follows that, for odd-order schemes

$$|A| = 1 + C_1(\omega\Delta t)^{r+1} + O[(\omega\Delta t)^{r+2}], \tag{A7}$$

whereas for even-order schemes

$$|A| = 1 + (C_1 + C_2)(\omega\Delta t)^{r+2} + O[(\omega\Delta t)^{r+3}]. \tag{A8}$$

This proves the first part of the proposition. Note that the amplitude error associated with odd-order methods must always be order $r + 1$, but that cancellation can reduce the amplitude error of even-order schemes beyond order $r + 2$. The leapfrog scheme, Milne's method (Gear 1971), the trapezoidal method, and Young's method A (Young 1968) are examples of even-order schemes with amplitude errors smaller than order $r + 2$.

The phase-speed error may be obtained by observing that, for r odd,

$$\frac{\Im[A]}{\Re[A]} = \tan\omega\Delta t + (C_2 - C_1)(\omega\Delta t)^{r+2} + O[(\omega\Delta t)^{r+3}]; \quad \text{(A9)}$$

and for r even,

$$\frac{\Im[A]}{\Re[A]} = \tan\omega\Delta t + C_1(\omega\Delta t)^{r+1} + O[(\omega\Delta t)^{r+2}]. \quad \text{(A10)}$$

Using the relation

$$\tan\alpha + C\alpha^r = \tan(\alpha + C\alpha^r) + O[\alpha^{r+2}], \quad \text{(A11)}$$

which holds whenever α, $C\alpha^r < 1$, one may show that for r odd

$$R = 1 + (C_2 - C_1)(\omega\Delta t)^{r+1} + O[(\omega\Delta t)^{r+2}]; \quad \text{(A12)}$$

whereas for r even

$$R = 1 + C_1(\omega\Delta t)^r + O[(\omega\Delta t)^{r+1}]. \quad \text{(A13)}$$

This concludes the proof. Note that reductions in the phase-speed error of odd-order schemes are possible through cancellation, but all even rth-order schemes have order-r phase-speed errors. These results notwithstanding, the author is not aware of an odd rth-order scheme that achieves better than order $r + 1$ phase-speed error.

REFERENCES

Asselin, R. A., 1972: Frequency filter for time integrations. *Mon. Wea. Rev.*, **100**, 487–490.

Blum, E. K., 1962: A modification of the Runge–Kutta fourth-order method. *Math. Comput.*, **16**, 176–187.

Durran, D. R., and J. B. Klemp, 1983: A compressible model for the simulation of moist mountain waves. *Mon. Wea. Rev.*, **111**, 2341–2361.

Gear, W. C., 1971: *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall, Inc., 253 pp.

Haltiner, G. J., and R. T. Williams, 1980: *Numerical Prediction and Dynamic Meteorology.* John Wiley and Sons, 477 pp.

Klemp, J. B., and R. B. Wilhelmson, 1978: The simulation of three-dimensional convective storm dynamics. *J. Atmos. Sci.*, **35**, 1070–1096.

Kreiss, H., and J. Oliger, 1973: *Methods for the approximate solution of time-dependent problems,* GARP Publ. Ser. 10. World Meteorological Organization, 107 pp.

Kurihara, Y., 1965: On the use of implicit and iterative methods for the time integration of the wave equation. *Mon. Wea. Rev.*, **93**, 33–46.

Kwizak, M., and A. Robert, 1971: A semi-implicit scheme for grid-point atmospheric models of the primitive equations. *Mon. Wea. Rev.*, **99**, 32–36.

Lilly, D. K., 1965: On the computational stability of numerical solutions of time-dependent nonlinear geophysical fluid dynamics problems. *Mon. Wea. Rev.*, **93**, 11–26.

Lorenz, E. N., 1963: Deterministic nonperiodic flow. *J. Atmos. Sci.*, **20**, 130–141.

——, 1971: An *N*-cycle time differencing scheme for stepwise numerical integration. *Mon. Wea. Rev.*, **99**, 644–648.

Matsuno, T., 1966: Numerical integrations of the primitive equations by a simulated backward difference method. *J. Meteor. Soc. Japan, Ser. 2,* **44**, 76–84.

Magazenkov, L. N., 1980: *Tr. Gl. Geofiz. Observ.*, **410**, 120–129.

Mesinger, F., and A. Arakawa, 1976: *Numerical Methods Used in Atmospheric Models,* Vol. 1, GARP Publ. Ser. 17. World Meteorological Organization, 64 pp.

Moeng, C.-H., 1984: A large-eddy simulation model for the study of planetary boundary-layer turbulence. *J. Atmos. Sci.*, **41**, 2052–2062.

Robert, A. J., 1966: The integration of a low-order spectral form of the primitive meteorological equations. *J. Meteor. Soc. Japan,* **44**, 237–244.

Schlesinger, R. E., L. W. Uccellini and D. R. Johnson, 1983: The effects of the Asselin time filter on numerical solutions to the linearized shallow-water wave equations. *Mon. Wea. Rev.*, **111**, 455–467.

Takacs, L. L., 1985: A two-step scheme for the advection equation with minimized dissipation and dispersion errors. *Mon. Wea. Rev.*, **113**, 1050–1065.

Williamson, D. L., 1983: Description of NCAR Community Climate Model (CCMOB). NCAR Technical Note, NCAR/TN-210 + STR, 88 pp. [Available from Information Services, NCAR, P.O. Box 3000, Boulder, CO 80307.]

Williamson, J. H., 1980: Low-storage Runge–Kutta schemes. *J. Comput. Phys.*, **35**, 48–56.

Wolfram, S., 1988: *Mathematica: A System for Doing Mathematics by Computer.* Addison-Wesley, 749 pp.

Young, J. A., 1968: Comparative properties of some time differencing schemes for linear and nonlinear oscillations. *Mon. Wea. Rev.*, **96**, 357–364.