

STAGGERED TIME INTEGRATORS FOR WAVE EQUATIONS*

MICHELLE GHRIST[†], BENGT FORNBERG[‡], AND TOBIN A. DRISCOLL[§]

Abstract. We consider variations of the Adams–Bashforth, backward differentiation, and Runge–Kutta families of time integrators to solve systems of linear wave equations on uniform, time-staggered grids. These methods are found to have smaller local truncation errors and to allow larger stable time steps than traditional nonstaggered versions of equivalent orders. We investigate the accuracy and stability of these methods analytically, experimentally, and through the use of a novel root portrait technique.

Key words. finite differences, staggered grid, linear multistep methods, Runge–Kutta methods, stability domain, imaginary stability boundary, root portrait

AMS subject classifications. 65L06, 65L12, 65L20, 65M06, 65M12, 65M20

PII. S0036142999351777

1. Introduction. When wave equations are posed as first-order systems and discretized in space to yield a system of ordinary differential equations (ODEs), the linearization of the resulting system has a purely imaginary spectrum. This corresponds to the fact that only propagation takes place. Many classical methods for ODEs have stability regions that include an interval of the form $[-iS_I, iS_I]$ on the imaginary axis. We call the largest such value of S_I the *imaginary stability boundary* (ISB) of the ODE integrator. In the context of a semidiscrete wave equation, two features are desired for an ODE integrator:

1. small local truncation error and
2. large imaginary stability boundary.

These two properties are typically in opposition to one another.

In [2, 4] it was shown that staggered or interlaced grids in space can increase the accuracy of finite difference and pseudospectral differentiation methods. Similarly, the unknowns of a wave equation can be staggered in time to yield benefits in both accuracy and stability. In this paper we introduce novel families of multistep and multistage staggered ODE integrators. We find that for multistep methods of the same order of accuracy, staggering in time usually improves accuracy by a factor of about 9 and increases the ISB by a factor of 2.4–7.4, with the factor growing as order increases. We also present a fourth-order multistage method which, compared to classical fourth-order Runge–Kutta (RK), has an error constant smaller by a factor of 16 and an ISB larger by a factor of about 2.

Typically the computational cost of using an implicit method is justified only in the presence of stiffness (not an issue for linear wave equations) or when there

*Received by the editors February 5, 1999; accepted for publication (in revised form) December 21, 1999; published electronically August 17, 2000.

<http://www.siam.org/journals/sinum/38-3/35177.html>

[†]Department of Mathematics and Computer Science, Belmont University, Nashville, TN 37212 (ghristm@mail.belmont.edu) The work of the first author was supported by NASA grant NPSC-OCC1035B and NSF grant DMS-9256335.

[‡]Department of Applied Mathematics, Campus Box 526, University of Colorado, Boulder, CO 80309 (fornberg@colorado.edu). The work of the second author was supported by NSF grant DMS-9706919 and AFOSR/DARPA grant F49620-96-1-0426.

[§]Department of Mathematics, Ewing Hall, University of Delaware, Newark, DE 19716. The work of the third author was funded by an NSF Mathematical Sciences Postdoctoral Research Fellowship and NSF VIGRE grant DMS-9810751.

is a relatively small number of equations in the system. We envision our methods being used to solve systems with a very large number of equations, possibly in the millions (as is the case when two-dimensional or three-dimensional wave equations are solved with a method of lines approach). For such situations, it is impractical to generate (and store) an LU decomposition. We thus consider only explicit methods in this analysis.

Although we focus our discussion on linear wave equations, linearity is not a requirement in any of our proposed schemes. Additionally, our time integrators are designed to solve first-order systems. Although systems of wave equations can often be rewritten as second-order systems, first-order formulations are generally preferred in the literature (e.g., Maxwell's equations), partly due to easier implementation of boundary conditions. (It is known that staggered grids are better for approximating odd-order derivatives and nonstaggered grids are better for approximating even-order derivatives [2].)

The rest of the paper is organized as follows:

2. Illustrations of grid staggering for wave equations;
3. Preliminaries;
4. Staggered multistep methods;
5. Theoretical considerations;
6. Staggered RK methods;
7. Root portraits;
8. Numerical experiments;
9. Conclusions.

Because we use a number of acronyms that may be unfamiliar to the reader, a glossary of these abbreviations is included in the appendix.

2. Illustrations of grid staggering for linear wave equations. Staggered grid techniques apply to linear hyperbolic equations which have been written as first-order systems. The variables in the system are staggered in such a way that the locations of values and their derivatives are interlaced.

We give two examples; other linear wave equations, including problems in three dimensions, can be treated similarly. Figure 2.1 gives four different ways to lay out the grid of unknowns u and v for the one-dimensional acoustic wave equation

$$(2.1) \quad \begin{aligned} \frac{\partial u}{\partial t} &= c \frac{\partial v}{\partial x}, \\ \frac{\partial v}{\partial t} &= c \frac{\partial u}{\partial x}. \end{aligned}$$

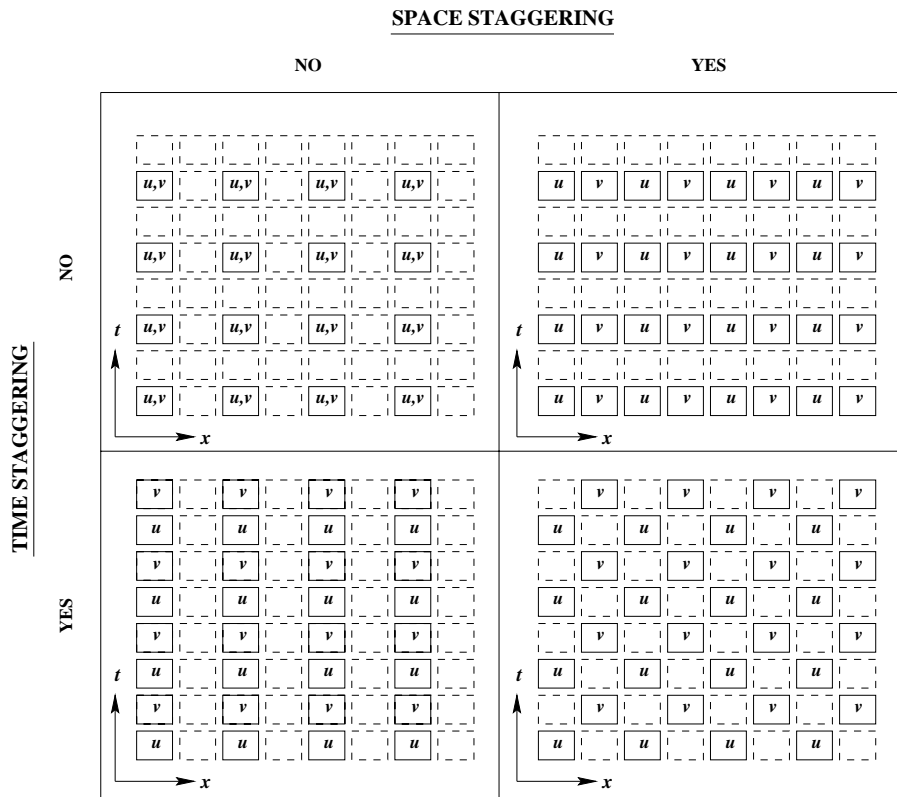


FIG. 2.1. Representative samples of various spatial/time grid layouts for the one-dimensional wave equation (2.1).

One can choose to utilize time staggering, space staggering, both, or neither. In every case the space-time density of data is exactly the same. Note that if one wants to incorporate staggering in time, the variables u and v must exist on interlaced time intervals (e.g., u exists on integer time levels, while v exists on half-integer time levels).

Figure 2.2 shows nonstaggered and staggered space grids for the two-dimensional elastic wave equation

$$\begin{aligned}
 \rho \frac{\partial u}{\partial t} &= \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y}, \\
 \rho \frac{\partial v}{\partial t} &= \frac{\partial g}{\partial x} + \frac{\partial h}{\partial y}, \\
 (2.2) \quad \frac{\partial f}{\partial t} &= (\lambda + 2\mu) \frac{\partial u}{\partial x} + \lambda \frac{\partial v}{\partial y}, \\
 \frac{\partial g}{\partial t} &= \mu \frac{\partial v}{\partial x} + \mu \frac{\partial u}{\partial y}, \\
 \frac{\partial h}{\partial t} &= \lambda \frac{\partial u}{\partial x} + (\lambda + 2\mu) \frac{\partial v}{\partial y}.
 \end{aligned}$$

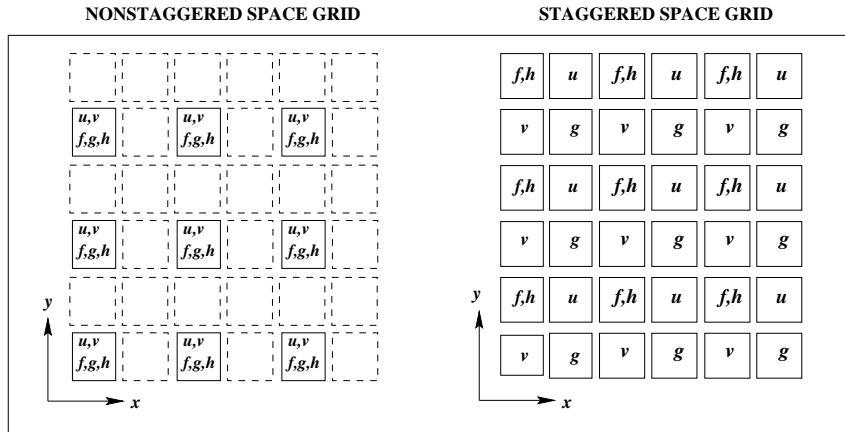


FIG. 2.2. Representative sample spatial grid layouts for the two-dimensional elastic wave equation (2.2).

The spatial staggering layout given in Figure 2.2 is uniquely determined. For example, the first equation requires that u be represented halfway between values of f in the x -direction and halfway between values of g in the y -direction. If $\frac{\partial g}{\partial x}$ also appeared in the first equation, it would not be possible to stagger spatially. We observe that for a large number of linear wave equations (e.g., Maxwell’s equations in any number of dimensions), the structure of the governing equations turns out to be precisely such that a unique and internally conflict-free staggering arrangement is possible, but we are unaware of any discussion of this in the literature. If one also wants to incorporate time staggering for this equation (with or without spatial staggering), we must again split the variables into two groups that exist on interlaced time intervals (e.g., u and v on integer time levels and $f, g,$ and h on half-integer time levels). An illustration of this arrangement is given in Figure 2.3.

3. Preliminaries.

3.1. Definitions. An m -step linear multistep method for solving the ODE

$$(3.1) \quad \frac{dy}{dt} = f(t, y(t))$$

is a difference equation of the form

$$(3.2) \quad \alpha_m y_{n+m} + \alpha_{m-1} y_{n+m-1} + \dots + \alpha_0 y_n = k(\beta_m f_{n+m} + \dots + \beta_0 f_n),$$

where k is the step size, α_i and β_i are* real parameters, $t_i = t_0 + ik$, $y_i = y(t_i)$, and $f_i = f(t_i, y_i)$. The coefficients α_i and β_i can be generated by using a two-line Mathematica or Maple algorithm based on Padé expansions [3]. Another way of representing the above general multistep method is through the use of generating polynomials

$$(3.3) \quad \begin{aligned} \rho(z) &= \alpha_m z^m + \alpha_{m-1} z^{m-1} + \dots + \alpha_0, \\ \sigma(z) &= \beta_m z^m + \beta_{m-1} z^{m-1} + \dots + \beta_0. \end{aligned}$$

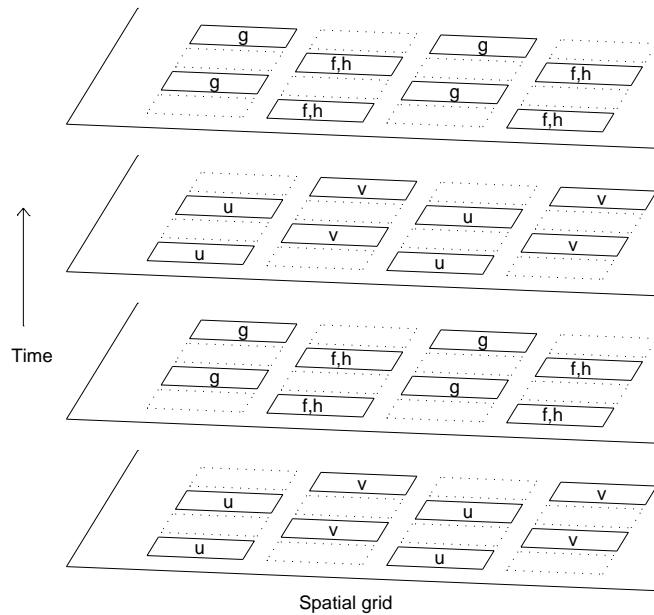


FIG. 2.3. Representative sample of a spatial-staggered, time-staggered grid for the two-dimensional elastic wave equation (2.2).

We consider only explicit methods, in which case $\beta_m = 0$. The local truncation error of a multistep method of order p is usually defined as

$$(3.4) \quad L(y, t, k) = C_{p+1} k^{p+1} y^{(p+1)}(t) + O(k^{p+2}),$$

which results from a simple Taylor expansion. The constant C_{p+1} is given by

$$(3.5) \quad C_{p+1} = \frac{1}{(p+1)!} \left(\sum_{i=0}^m \alpha_i i^{p+1} - (p+1) \sum_{i=0}^m \beta_i i^p \right).$$

However, as noted in [6], this constant does not accurately reflect the global error to be expected when using a method. The proper *error constant* is given by

$$(3.6) \quad C = \frac{C_{p+1}}{\sigma(1)}.$$

It is this coefficient C that we use when comparing the accuracy of methods of the same order.

Similarly, an explicit s -stage RK method can be represented as

$$(3.7) \quad y_{n+1} = y_n + \sum_{j=1}^s b_j d_j,$$

where

$$\begin{aligned}
 d_1 &= kf(t_n, y_n), \\
 d_2 &= kf(t_n + c_2k, y_n + a_{21}d_1), \\
 d_3 &= kf(t_n + c_3k, y_n + a_{31}d_1 + a_{32}d_2), \\
 &\vdots \\
 d_s &= kf\left(t_n + c_s k, y_n + \sum_{i=1}^{s-1} a_{si}d_i\right).
 \end{aligned}
 \tag{3.8}$$

The (linear) error constant for such a method can be found by considering the linear problem

$$y' = f(t, y) = \lambda y \tag{3.9}$$

and Taylor expanding $(y_{n+1} - e^{\lambda k}y_n)$ about $k = 0$ to find C :

$$k \left(\sum_{j=1}^s b_j d_j \right) + (1 - e^{\lambda k})y_n = C(\lambda k)^{p+1} + O((\lambda k)^{p+2}). \tag{3.10}$$

For multistage methods, it is appropriate to normalize the stability domain by dividing by the number of stages s and to normalize the error constant by a factor s^p , where p is the order of the method. This ensures that we are comparing all time-stepping methods on the basis of equal work.

3.2. Maximum imaginary stability boundary. Jeltsch and Nevanlinna [7] have shown that the normalized ISB for a large class of schemes, including multistep and RK methods, cannot exceed 1 in the classical (nonstaggered) case. This limit is achieved by the classical leapfrog scheme

$$y_{n+1} = y_{n-1} + 2kf(t_n, y_n). \tag{3.11}$$

This method has a stability domain $[-i, i]$ on the imaginary axis.

Leapfrog can also be used as a time-staggered method, namely

$$y_{n+1} = y_n + kf\left(t_n + \frac{k}{2}, y_{n+\frac{1}{2}}\right). \tag{3.12}$$

In this context the stability domain is $[-2i, 2i]$; the extra factor of two simply reflects the fact that the time levels are $\{n, n + \frac{1}{2}, n + 1\}$ rather than $\{n - 1, n, n + 1\}$. For staggered multistep and RK methods that we will consider, this implies a maximum normalized ISB of 2.

4. Staggered multistep methods. To utilize the methods that follow, we require only that u and $\frac{\partial u}{\partial t}$ are used on interlaced time levels. However, as noted in section 2, many (if not all) systems of wave equations can be rewritten in the form $u_t = f(t, v(t)), v_t = g(t, u(t))$ (where u and v may be vectors). In this case, by having u on one time level and v on the other interlaced time level, one is effectively able to double the ISB. (Section 3.2 demonstrates this for the leapfrog method.) We envision our methods being used for such systems of wave equations.

4.1. Staggered Adams–Bashforth and backward differentiation methods. We first consider staggered versions of the Adams–Bashforth and backward differentiation time integrators, denoted ABS and BDS, respectively. To illustrate our notation, we show in Figure 4.1 four different ways of representing the third-order ABS method (ABS3): a representative stencil, the stencil coefficients, the polynomials $\rho(z)$ and $\sigma(z)$, and the explicit Taylor formula. Note that all coefficients listed in this paper can be found via Padé expansions [3]. In Table 4.1 we give for stable BDS methods the shape and coefficients of the stencil, the error constant, a picture of the stability domain, and the ISB. (Note that by stable, we mean zero-stable.) Tables 4.2 and 4.3 give the same information for useful ABS and AB methods up to order 8.

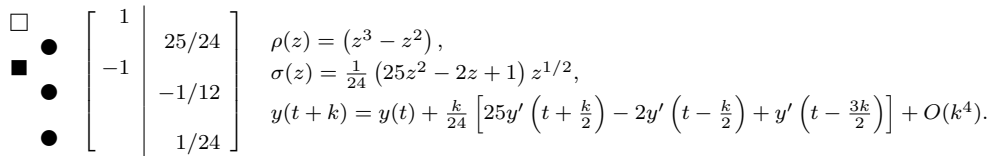


FIG. 4.1. Four representations of ABS3: stencil shape, coefficients, defining polynomials $\rho(z)$ and $\sigma(z)$, and explicit Taylor formula. Here, \square represents an unknown function value, while \blacksquare and \bullet stand for a known function and a known derivative value.

TABLE 4.1

Staggered backward-differentiation time integrators. The normalized local truncation error for BDS_p is $Ck^{p+1}f^{(p+1)}(\eta)$, where C is the error constant. Only stable methods are shown.

Name	Stencil	Coefficients	Error constant	Stability domain	ISB
BDS2 (leapfrog)	$\square \bullet$ $\blacksquare \bullet$	$\left[\begin{array}{c c} 1 & \\ \hline -1 & 1 \end{array} \right]$	$\frac{1}{24}$		2
BDS3	$\square \bullet$ $\blacksquare \bullet$ $\blacksquare \bullet$ $\blacksquare \bullet$	$\left[\begin{array}{c c} 1 & \frac{24}{23} \\ \hline -\frac{21}{23} & \\ -\frac{3}{23} & \\ \frac{1}{23} & \end{array} \right]$	$\frac{1}{24}$		$\frac{5}{3} \approx 1.667$
BDS4	$\square \bullet$ $\blacksquare \bullet$ $\blacksquare \bullet$ $\blacksquare \bullet$ $\blacksquare \bullet$	$\left[\begin{array}{c c} 1 & \frac{12}{11} \\ \hline -\frac{17}{22} & \\ -\frac{9}{22} & \\ \frac{5}{22} & \\ -\frac{1}{22} & \end{array} \right]$	$\frac{71}{1920}$		1

TABLE 4.2

Staggered Adams–Bashforth time integrators. The normalized local truncation error for ABS_p is $Ck^{p+1}f^{(p+1)}(\eta)$, where C is the error constant. Only methods with nonzero ISBs are shown (for orders $p < 10$).

Name	Stencil shape	Coefficients	Error constant	Stability domain	ISB
ABS2 (leapfrog)		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 1 \end{array} \right]$	$\frac{1}{24}$		2
ABS3		$\left[\begin{array}{c c} 1 & 25/24 \\ \hline -1 & -1/12 \\ & 1/24 \end{array} \right]$	$\frac{1}{24}$		$\frac{12}{7}$ ≈ 1.714
ABS4		$\left[\begin{array}{c c} 1 & 13/12 \\ \hline -1 & -5/24 \\ & 1/6 \\ & -1/24 \end{array} \right]$	$\frac{223}{5760}$		$\frac{4}{3}$ ≈ 1.333
ABS7		$\left[\begin{array}{c c} 1 & 1152511/967680 \\ \hline -1 & -7969/10752 \\ & 134881/107520 \\ & -294659/241920 \\ & 76921/107520 \\ & -12629/53760 \\ & 32119/967680 \end{array} \right]$	$\frac{1111}{35840}$		$\frac{30240}{81469}$ ≈ 0.371
ABS8		$\left[\begin{array}{c c} 1 & 295627/241920 \\ \hline -1 & -103021/107520 \\ & 102437/53760 \\ & -2228531/967680 \\ & 24197/13440 \\ & -95251/107520 \\ & 121049/483840 \\ & -1111/35840 \end{array} \right]$	$\frac{13528301}{464486400}$		$\frac{4320}{20209}$ ≈ 0.214

TABLE 4.3

Nonstaggered Adams–Bashforth time integrators. The normalized local truncation error for AB_p is $Ck^{p+1}f^{(p+1)}(\eta)$, where C is the error constant. Other than AB_2 , only methods with nonzero ISBs are shown (for orders $p < 10$).

Name	Stencil Shape	Coefficients	Error Constant	Stability Domain	ISB
AB2		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 3/2 \\ \hline & -1/2 \end{array} \right]$	$\frac{5}{12}$		0
AB3		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 23/12 \\ \hline & -4/3 \\ \hline & 5/12 \end{array} \right]$	$\frac{3}{8}$		$\frac{12}{5\sqrt{11}}$ ≈ 0.724
AB4		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 55/24 \\ \hline & -59/24 \\ \hline & 37/24 \\ \hline & -3/8 \end{array} \right]$	$\frac{251}{720}$		$\frac{52}{15\sqrt{65}}$ ≈ 0.430
AB7		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 198721/60480 \\ \hline & -18637/2520 \\ \hline & 235183/20160 \\ \hline & -10754/945 \\ \hline & 135713/20160 \\ \hline & -5603/2520 \\ \hline & 19087/60480 \end{array} \right]$	$\frac{5257}{17280}$		≈ 0.058
AB8		$\left[\begin{array}{c c} 1 & \\ \hline -1 & 16083/4480 \\ \hline & -1152169/120960 \\ \hline & 242653/13440 \\ \hline & -296053/13440 \\ \hline & 2102243/120960 \\ \hline & -115747/13440 \\ \hline & 32863/13440 \\ \hline & -5257/17280 \end{array} \right]$	$\frac{1070017}{3628800}$		≈ 0.029

The ABS and BDS methods of order 2 are both equivalent to the leapfrog method. ABS and BDS methods are both explicit (whereas nonstaggered BD methods are implicit). AB and ABS methods are always stable; BDS methods are stable for orders up through 4 (while nonstaggered BD methods are stable for orders up through 6).

Stability domains for staggered methods are symmetric with respect to both coordinate axes; one can see this by noting that there is symmetry across the x -axis (true of all stability domains) as well as symmetry about the origin (which comes from the structure of staggered methods). This means that these methods have no real axis coverage; thus, these methods are appropriate only for propagation problems. (However, through exponential time-stepping [9], the schemes can also be applied to problems such as attenuation in Maxwell's equations for lossy media.)

As will be discussed in section 5.1, AB methods have a nonzero ISB only for methods of order 3, 4, 7, 8, 11, 12, etc.; ABS methods additionally include order 2. Note that the error constants for the staggered methods are approximately nine times smaller than those of the nonstaggered methods of equivalent order. In addition, staggering increases the ISB by a factor of 2.4–7.4, with the factor growing as order increases.

We can also compare the staggered methods to Störmer methods [6] in those cases for which the problem can be reformulated as a second-order system, $u_{tt} = F(t, u)$, $v_{tt} = G(t, v)$. With compatible definitions we find ISBs of around 2, 1.73, 1.41, 1.11, 0.84, 0.62, and 0.46 for orders 2–8. The associated error constants are approximately 0.083, 0.083, 0.079, 0.075, 0.071, 0.068, and 0.066. Thus the ABS methods compare favorably for orders of accuracy 4 and less and unfavorably thereafter. However, formulating wave equations using two time derivatives sometimes creates difficulties with boundary conditions.

To implement one of the time-staggered methods, one needs to obtain starting values for several time levels after the initial condition. For nonstaggered multistep methods, this is usually accomplished with an RK method. For staggered time integrators, one should obtain as many (half-integer) levels of u and v as needed using a nonstaggered RK method and then select out those needed to interlace u and v appropriately.

4.2. Free parameter multistep methods. We have developed multistep methods that allow free parameters due to suboptimization of order. We offer an example of such a method as an illustration of opportunities available in this area. The following is a fourth-order staggered multistep scheme with two free parameters, α and β .

$$\begin{aligned} \rho(z) &= z^4 + \left(-\frac{17}{22} - \frac{577}{528}\alpha + \frac{1}{24}\beta\right) z^3 + \left(-\frac{9}{22} + \frac{201}{176}\alpha - \frac{9}{8}\beta\right) z^2 \\ &\quad + \left(\frac{5}{22} - \frac{9}{176}\alpha + \frac{9}{8}\beta\right) z + \left(-\frac{1}{22} + \frac{1}{528}\alpha - \frac{1}{24}\beta\right), \\ \sigma(z) &= \left[\left(\frac{12}{11} - \frac{1}{22}\alpha\right) z^3 - \alpha z^2 - \beta z\right] z^{1/2}. \end{aligned}$$

The error constant of this scheme is

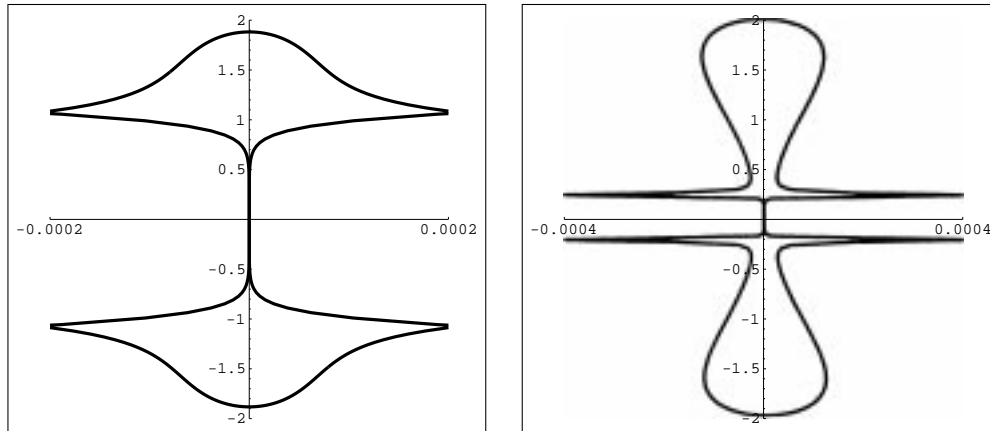


FIG. 4.2. *Stability domains of the fourth-order staggered free parameter scheme: (a) ISB \approx 1.8822, $C \approx$ 0.0353. (b) ISB \approx 1.995, $C \approx$ 32.34.*

$$(4.1) \quad C = \frac{(1704 + 127\alpha + 198\beta)}{1920(24 - 23\alpha - 22\beta)}.$$

By choosing various values of the parameters for which the method is stable, we can change the error constant and the ISB of the method. As the ISB approaches the theoretical limit of 2, the error constant becomes unbounded. We list three examples of interest:

- $\alpha = 1$, $\beta = -1.045$, ISB \approx 1.8822, error constant $C \approx$ 0.03526. While the error constant is comparable to that of ABS4, there is a dramatic improvement in the ISB. We show the stability domain of this method in Figure 4.2(a).
- $\alpha = -0.74$, $\beta = -1.121$, ISB \approx 1.337, $C \approx$ 0.0110. This method improves on the accuracy of ABS4 by about a factor of 3 while maintaining about the same ISB.
- $\alpha = 1.95$, $\beta = -1.00155$, ISB \approx 1.995, $C \approx$ 32.34. This method has an ISB very close to 2, the theoretical limit. See Figure 4.2(b) for the stability domain of this method.

Note that these free parameter methods require no more function evaluations than ABS4; all multistep methods require only one function evaluation per time step. As we have not yet explored the properties of these free parameter schemes in any great detail, we exclude them from further analysis.

5. Theoretical considerations.

5.1. ISB of Adams–Bashforth methods. Adams–Bashforth methods have a nonzero ISB only for orders 3, 4, 7, 8, 11, 12, etc. ABS methods additionally include order 2 (leapfrog). Because we have been unable to locate a proof of either result in the literature, we offer here an outline of our proof; details are given in [5].

The edge of the stability domain is described by the root ξ of the equation

$$\rho(r) - \xi\sigma(r) = 0$$

when r travels around the unit circle ($r = e^{i\theta}$). For an exact method, we would get

$$\xi(\theta) = i\theta.$$

A numerical scheme of order p will instead lead to

$$(5.1) \quad \xi(\theta) = i\theta + c_p(i\theta)^{p+1} + d_p(i\theta)^{p+2} + \dots.$$

The sign of the first *real* term of this expansion will dictate whether the stability domain boundary near the origin swings to the right or to the left of the imaginary axis. For AB and ABS methods, we find that $c_p > 0$ and $d_p < 0$. The pattern for which methods have nonzero ISBs then follows from the powers of the imaginary unit in (5.1).

To find the values of c_p and d_p in the case of nonstaggered AB p methods, we note that these schemes, when applied to $y' = \lambda y$ (with $\lambda = \xi/k$), take the form

$$(5.2) \quad y(k) - y(0) = \frac{\xi}{k} \int_0^k (\text{interpolating polynomial of } y \text{ over } [-(p-1)k, 0]) dt.$$

Substituting $y(t) = e^{i\theta t/k}$ into (5.2) and solving for $\xi(\theta)$ gives (after some algebraic simplifications)

$$(5.3) \quad c_p = \int_0^1 \binom{x+p-1}{p} dx, \quad d_p = - \int_0^1 \binom{x+p-1}{p} \frac{p^2+1-2x}{2(p+1)} dx.$$

The result follows now from the fact that both integrands are nonnegative. In the staggered case, we find similarly

$$(5.4) \quad c_p = \int_{-1/2}^{1/2} \binom{x+p-1}{p} dx, \quad d_p = - \int_{-1/2}^{1/2} \binom{x+p-1}{p} \frac{p(p-1)-2x}{2(p+1)} dx.$$

Although the integrands for both c_p and d_p are no longer of constant sign, one can again establish that $c_p > 0$ and $d_p < 0$ (for $p > 2$) by induction, for example.

5.2. Staggered analogue of Dahlquist's first stability barrier. Dahlquist's first stability barrier for multistep methods states that the order p of an explicit stable m -step method must satisfy $p \leq m$. The analogue of this theorem for staggered multistep methods follows.

The order p of an explicit stable m -step staggered method satisfies

$$(5.5) \quad p \leq \begin{cases} m, & m \text{ an even integer,} \\ m + \frac{1}{2}, & m \text{ a half-integer,} \\ m + 1, & m \text{ an odd integer.} \end{cases}$$

Our proof of this theorem follows those of Jeltsch and Nevanlinna [8] and Dahlquist [1] and can be found in [5].

6. Staggered RK methods. Multistage methods can also be put into a staggering framework. We write the ODE in the form

$$(6.1) \quad \begin{aligned} u' &= f(t, v(t)), \\ v' &= g(t, u(t)). \end{aligned}$$

The splitting into u and v (each could be a vector) is to allow quantities to be given at offset time levels, as suggested in section 2. The splitting into f and g reflects that values of u' (or v') are given at time levels staggered with respect to u (or v).

One form for a staggered RK (RKS) method is

$$(6.2) \quad \begin{aligned} d_1 &= kf(t_{n+1/2}, v_{n+1/2}), \\ d_2 &= kg(t_n + c_2k, u_n + a_{21}d_1), \\ d_3 &= kf(t_{n+1/2} + c_3k, v_{n+1/2} + a_{32}d_2), \\ d_4 &= kg(t_n + c_4k, u_n + a_{41}d_1 + a_{43}d_3), \\ &\vdots \\ d_s &= kf(t_{n+1/2} + c_s k, v_{n+1/2} + a_{s2}d_2 + \cdots + a_{s,s-1}d_{s-1}), \\ u_{n+1} &= u_n + b_1d_1 + b_3d_3 + \cdots + b_sd_s \end{aligned}$$

if s is odd. If s is even, the first stage should be an evaluation of g at time t_n and the stages used to advance from u_n to u_{n+1} are the even-numbered ones. The same formula can be used to advance v , once references to f and g are switched and time levels are shifted forward by $\frac{1}{2}$. Observe that advancing both u and v by one step requires s evaluations each of f and g . The form of the governing equations in (6.1) suggests that an evaluation of both f and g should count as one stage, so (6.2) is an s -stage method.

The coefficients in such a formula can be derived by a straightforward, if laborious, Taylor expansion of both the exact difference $u_{n+1} - u_n$ and the RKS approximation $b_1d_1 + \cdots + b_sd_s$. The expansions must be made so that v (and consequently f) is evaluated only at $t_{n+1/2}$ and u (hence g) is evaluated at t_n . If more than a few stages are desired, a symbolic computational package is useful both for generating these expansions and for solving the system of nonlinear equations that results from equating their coefficients.

Stability analysis follows the usual pattern. The model problem is linear:

$$\begin{bmatrix} u \\ v \end{bmatrix}' = \begin{bmatrix} 0 & \lambda \\ \lambda & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}.$$

(Using different scalars in the off-diagonal entries of the matrix does not change anything essential because the eigenvalues depend only on the product of those entries.) In applying the RKS method, one finds that

$$\begin{aligned} u_{n+1} &= \beta(k\lambda)v_{n+1/2} + \alpha(k\lambda)u_n, \\ v_{n+3/2} &= \beta(k\lambda)u_{n+1} + \alpha(k\lambda)v_{n+1/2}, \end{aligned}$$

or

$$\begin{bmatrix} u_{n+1} \\ v_{n+3/2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\beta & 1 \end{bmatrix}^{-1} \begin{bmatrix} \alpha & \beta \\ 0 & \alpha \end{bmatrix} \begin{bmatrix} u_n \\ v_{n+1/2} \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \alpha\beta & \alpha + \beta^2 \end{bmatrix} \begin{bmatrix} u_n \\ v_{n+1/2} \end{bmatrix} = Q(k\lambda) \begin{bmatrix} u_n \\ v_{n+1/2} \end{bmatrix}.$$

The stability region consists of all values of $k\lambda$ for which both eigenvalues of $Q(k\lambda)$ are inside the unit circle or simple and on the unit circle. After a short calculation, one finds that $[1 \ w]^T$ is an eigenvector if and only if

$$(6.3) \quad w^2 = \beta(k\lambda)w + \alpha(k\lambda)$$

and the corresponding eigenvalue is w^2 . (This is the same equation that arises using the ansatz $u_n = w^{2n}$, $v_{n+1/2} = w^{2n+1}$.) The two roots of (6.3) thus determine the stability region. As was mentioned in section 3.1, we normalize the stability region by the number of stages in order to make a fair comparison to one-stage methods. An error constant can also be defined by looking at the first error term in the approximate solution of the linear model problem. This too should be normalized by a factor s^p for a p th-order method.

We recognize leapfrog as a 1-stage RKS method of order 2. Computation of the expansions for 2-stage and 3-stage methods reveals that neither has enough additional free parameters to improve upon the order of leapfrog. While 4-stage, third-order methods do exist, they do not improve on their nonstaggered counterparts. The first interesting higher-order method is the five-stage RKS method. Here there are 13 constants to be determined in the formula. To achieve fourth-order accuracy, 21 conditions (most of which are nonlinear) must be satisfied. Remarkably, there is a family of solutions parameterized by b_5 . With $\gamma = (6b_5)^{-1/2}$, the tableau for the general solution is

$$(6.4) \quad \begin{array}{ccc|ccc} & 0 & & & & \\ & \frac{1}{4}(2-\gamma) & & \frac{1}{4}(2-\gamma) & & \\ & -\frac{1}{2}\gamma & & & -\frac{1}{2}\gamma & \\ \frac{1}{4}(2+\gamma) & & & \frac{1}{4}(2+\gamma) & & 0 \\ & \frac{1}{2}\gamma & & & 0 & \frac{1}{2}\gamma \\ \hline & & & 1-2b_5 & b_5 & b_5 \end{array} .$$

(Entries which are blank are zero for structural reasons.) The stability region and error constant are independent of the choice of the free parameter. The most appealing member of the family, which we call RKS4, is given with $b_5 = 1/24$ and hence $\gamma = 2$:

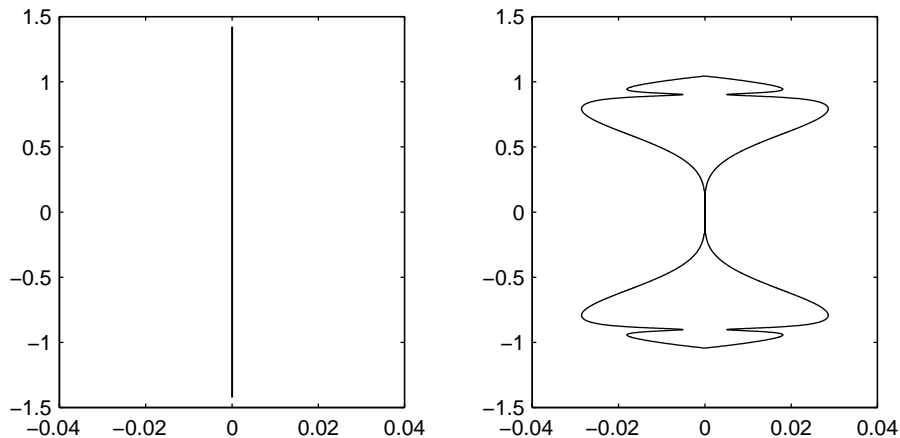


FIG. 6.1. Stability domains for RKS methods: (a) order 4 (from (6.5)); (b) order 3 (from (6.6)).

$$\begin{aligned}
 d_1 &= kf(t_{n+1/2}, v_{n+1/2}), \\
 d_2 &= kg(t_n, u_n), \\
 d_3 &= kf(t_{n+1/2} - k, v_{n+1/2} - d_2), \\
 d_4 &= kg(t_n + k, u_n + d_1), \\
 d_5 &= kf(t_{n+1/2} + k, v_{n+1/2} + d_4), \\
 u_{n+1} &= u_n + \frac{11}{12}d_1 + \frac{1}{24}d_3 + \frac{1}{24}d_5.
 \end{aligned}
 \tag{6.5}$$

Observe that while five stages are required, the stage d_1 is actually equivalent to the future stage d_2 for the advance of v from time level $n + \frac{1}{2}$ to $n + \frac{3}{2}$. Hence only four evaluations each of f and g are needed to advance both u and v one time step, and we consider this to be a four-stage method for purposes of normalization of stability and error constant.

RKS4 has a simple interpretation. Given the original data u_n and $v_{n+1/2}$, leapfrog is used repeatedly to estimate $v_{n-1/2}$, u_{n+1} , and $v_{n+3/2}$ in succession. The three estimates of v values are then combined according to a finite-difference stencil to relate u_n to the new value of u_{n+1} . The method is fourth order due to a symmetry which produces cancellation in the leapfrog errors.

The stability region of RKS4 is a segment of the imaginary axis, and the normalized ISB of this method is about 1.425 (see Figure 6.1). Hence for equivalent amounts of work per step, time steps about twice as large as those of standard RK4 are possible. The error constant is $1/1920$, compared to $1/120$ for RK4. (After normalization for comparison to one-stage methods, these constants become $2/15$ and $32/15$, respectively.) As with multistep methods, the problem may in many cases be written as a second-order system in time. The three-stage, fourth-order Nyström method presented in [6] has an equivalent normalized ISB of about 0.86, far less than that of RKS4.

Notice that stages 4 and 5 are independent of stages 2 and 3. The storage requirements can therefore be kept low. In the following procedure, time dependence and subscripts on u and v are omitted for clarity and z_1 is assumed to start with the value $kg(u)$, obtained from the previous advance of v .

$$\begin{aligned}
 z_1 &\leftarrow v - z_1 \\
 z_1 &\leftarrow kf(z_1) \\
 z_2 &\leftarrow kf(v) \\
 z_3 &\leftarrow u + z_2 \\
 z_3 &\leftarrow kg(z_3) \\
 z_3 &\leftarrow v + z_3 \\
 z_3 &\leftarrow kf(z_3) \\
 u &\leftarrow u + z_1/24 + 11z_2/12 + z_3/24
 \end{aligned}$$

At the end of this procedure, z_2 holds the value that serves as stage 2 of the next advance of v . Only three temporary variables are needed. Each needs to have as many components as the larger of u and v . In the common situation where u and v each hold half of the variables of the system, the additional storage is equivalent to $3/2$ of the total number of unknowns. In standard fourth-order RK, the best temporary storage is twice the number of unknowns.

The RKS method suggested above evaluates a number of stages to advance u , then a new set of independent stages to advance v (although RKS4 can reuse one stage). An alternative is to use a joint set of stages to advance u and v simultaneously. While still using the same number $2s$ of individual f and g evaluations per time step as the other s -stage staggered methods, a potential advantage here is that the number of free constants grows more quickly with s . As an example, we have a third-order, three-stage method:

$$\begin{aligned}
 d_1 &= kf(t_{n+1/2}, v_{n+1/2}), \\
 d_2 &= kg(t_n, u_n), \\
 d_3 &= kf(t_{n+1/2} - \frac{1}{2}k, v_{n+1/2} - \frac{1}{2}d_2), \\
 d_4 &= kg(t_n + \frac{13}{12}k, u_n + \frac{13}{12}d_1), \\
 d_5 &= kf(t_{n+1/2} + \frac{1}{2}k, v_{n+1/2} + \frac{7}{26}d_2 + \frac{3}{13}d_4), \\
 d_6 &= kg(t_n + \frac{13}{12}k, u_n + (\frac{91}{72} - 2\gamma)d_1 + (-\frac{13}{72} + \gamma)d_3 + \gamma d_5), \\
 u_{n+1} &= u_n + \frac{2}{3}d_1 + \frac{1}{6}d_3 + \frac{1}{6}d_5, \\
 v_{n+3/2} &= v_{n+1/2} + \frac{1}{13}d_2 + \frac{6}{13}d_4 + \frac{6}{13}d_6.
 \end{aligned}
 \tag{6.6}$$

Here γ is a constant that affects the accuracy and stability of the method. The ISB is approximately optimized if $\gamma = 104/181$. For this choice, the normalized ISB is ≈ 1.044 and the normalized error constant is $27\gamma/24 \approx 0.6464$. The stability region is displayed in Figure 6.1. For comparison, classical RK3 has a normalized ISB of $1/\sqrt{3} \approx 0.577$ and normalized error constant of $9/8 = 1.125$.

We have by no means exhausted the possibilities for either type of staggering in RK methods; our intent has been to demonstrate that such methods do exist and can improve on their nonstaggered counterparts.

7. Root portraits. Since our goal is to perform time stepping for wave equations, it is illuminating to compare methods based on their performance on the one-dimensional scalar wave equation,

$$\begin{aligned}
 u_t &= v_x, \\
 v_t &= u_x.
 \end{aligned}
 \tag{7.1}$$

We think of the spatial domain as unbounded and spatial derivatives as exact. For a Fourier mode whose spatial dependence is $e^{i\omega x}$, the wave equation becomes the ODE system

$$(7.2) \quad \begin{bmatrix} u \\ v \end{bmatrix}_t = \begin{bmatrix} 0 & i\omega \\ i\omega & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}.$$

These equations allow leftgoing and rightgoing modes. When a mode is advanced in time by an amount k , the solution is multiplied by a factor $e^{\pm ik\omega}$ with the sign determining only direction of travel.

For the classical multistep methods, the analysis reduces to the situation familiar from linear stability. The numerical solution is capable of travel in either direction and to advance a mode by time k the solution is multiplied by a factor $z(ik\omega)$. For linear multistep methods, z is a root of the characteristic polynomial equation

$$(7.3) \quad \rho(z) - ik\omega\sigma(z) = 0.$$

When $k\omega = 0$, a stable method has exactly one root at $z = 1$. As $ik\omega$ travels along the imaginary axis, this root approximates the exact factor $e^{ik\omega}$ (or its conjugate) but eventually becomes noticeably different. The other roots of the characteristic polynomial are physically irrelevant. When $k\omega$ is larger than the ISB of the method, some root is outside the unit disk and the method becomes unstable.

To visualize this process, we draw a “root portrait” that traces the physically relevant root as $k\omega$ takes on all stable values. An example for AB3 is shown in Figure 7.1. A point $ik\omega$ on the imaginary axis should ideally map to $e^{\pm ik\omega}$ on the unit circle, as the tick marks outside the unit circle suggest. The physically relevant root, as determined by the characteristic polynomial, is perfect at the origin and a good approximation nearby, but eventually the path of the root diverges from the circle. When $ik\omega$ encounters the boundary of the stability region, one of the parasitic roots not shown is just crossing the unit circle on its way to creating time instability.

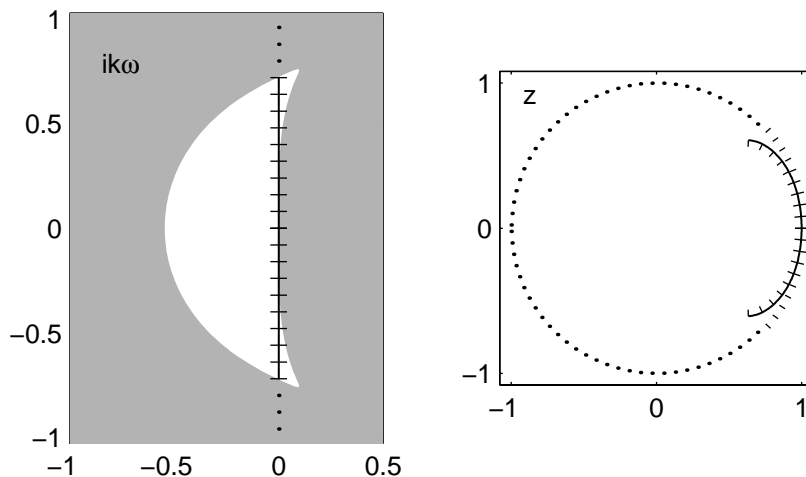


FIG. 7.1. Example of a “root portrait.” The portion of the imaginary axis which lies inside the stability region of AB3 is mapped to the physically relevant roots inside the unit circle. Ideally, the evenly spaced tick marks along the unit circle should line up with the tick marks along the root path, but this is true only near the origin.

A similar analysis can be made for classical RK methods. Here the characteristic polynomial is linear in z , but there is a polynomial dependence on $ik\omega$. (For orders p less than five, this polynomial is just the p th-order Taylor polynomial for $e^{ik\omega}$.) Also, the stability region must be normalized by the number s of stages in the method, and the s th root of z must be taken in accordance. Because there is only one root, the physically relevant root also determines stability.

For staggered multistep schemes, the characteristic equation is again (7.3). This is now a polynomial in $z^{1/2}$, and the roots are easily found. Again only one root per direction of travel is physically relevant. For staggered RK methods, the stability analysis in section 5 applies; in fact, $z^{1/2}$ is just the variable w in the characteristic equation (6.3), and λ is purely imaginary in that formula.

Figure 7.2 displays the root portraits for classical and staggered methods of orders 2, 3, 4, and 7. As the order of a method increases, inner and outer ticks match up more accurately near $z = 1$. The stability restriction is made clear by where the tick marks on the unit circle end. The AB2 and RK2 methods are stable but have zero ISB. The ABS2, BDS2, and RKS2 methods are all equivalent to leapfrog, which has the maximum possible ISB of 2. In every case, staggered schemes are seen to have stability and accuracy properties superior to their nonstaggered counterparts.

Another way to view the root portraits is in terms of numerical dissipation and dispersion. Because we have eliminated the spatial discretization errors, root portraits clearly show the errors solely due to the time stepping schemes. The amount of numerical dissipation in a scheme is shown by how close the path of the root portrait stays to the unit circle, whereas the amount of dispersion is shown by how well the inner ticks on the root portrait path match up to the outer ticks on the unit circle. ABS2/BDS2/RKS2 (leapfrog) and RKS4 have no dissipation because all roots stay on the unit circle but have significant dispersion near the edge of the stability domain because the inner and outer ticks do not match well there.

8. Numerical experiments. The root portrait data can be used to experimentally compare AB, ABS, BDS, RK, and RKS time integrators for wave propagation. As described in the previous section, when solving (7.1), one can model the effect that a particular numerical time integrator has on a particular Fourier mode $e^{i\omega x}$ by solving (7.3) for the physical root $z(ik\omega)$. We choose the physical root so that the solution moves strictly to the right. Then the solution at the n th time step is given by

$$(8.1) \quad \begin{aligned} u(t = kn) &= z^n e^{i\omega x}, \\ v(t = kn) &= -z^n e^{i\omega x}. \end{aligned}$$

We use the initial condition

$$(8.2) \quad u(x, 0) = \begin{cases} (1 + \cos(\frac{x}{0.15}))^2, & |x| < 0.15, \\ 0, & |x| \geq 0.15, \end{cases}$$

and advance the solution to final time $T = 6\pi$, so that the exact final solution is the same as the initial condition. We define N to be the number of function evaluations used to advance the solution from $T = 0$ to $T = 6\pi$, i.e., the number of time steps taken multiplied by the number of stages. This provides a legitimate comparison between one-step methods like AB, ABS, and BDS and multistage methods like RK and RKS.

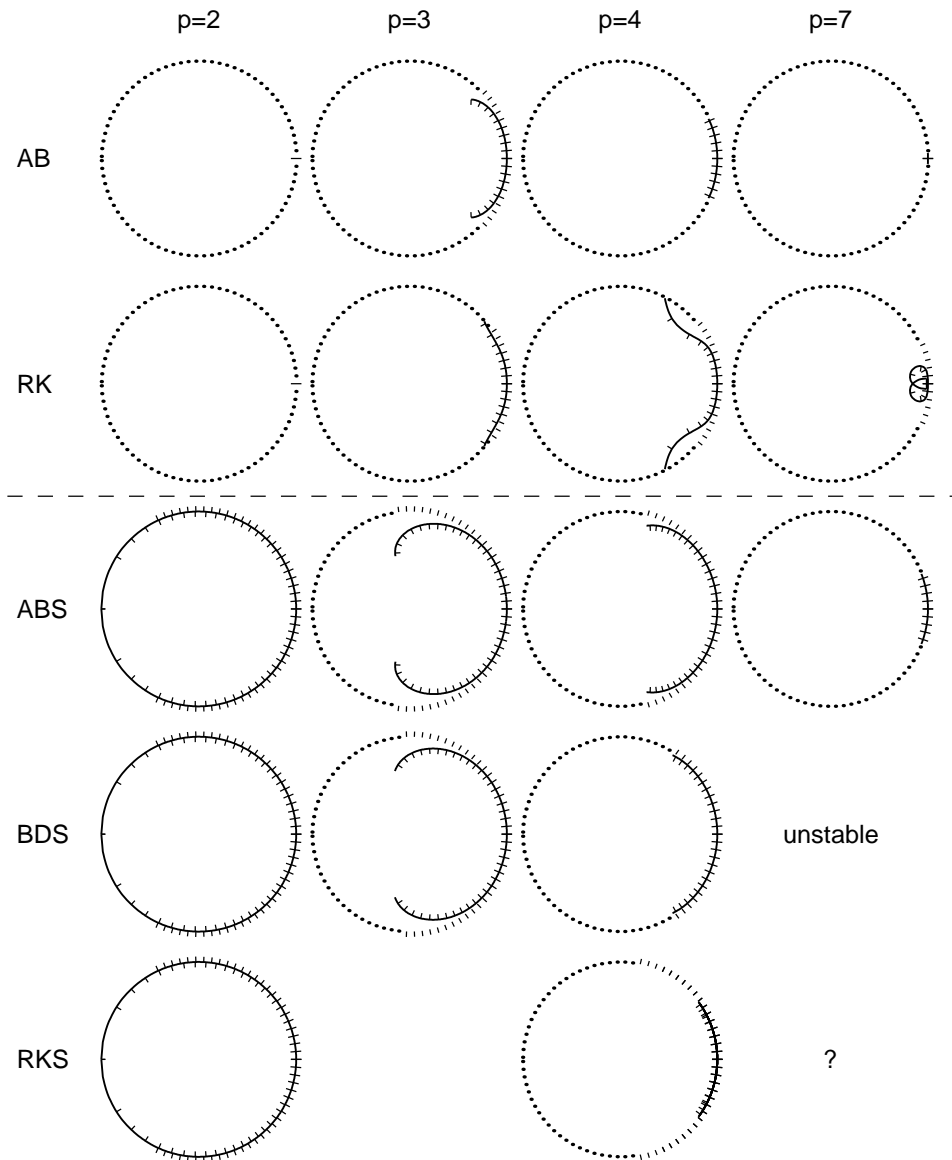


FIG. 7.2. Root portraits for classical and staggered methods of different orders. Stability of the methods for the wave equation is reflected by the length of the arc made by the tick marks on the unit circle. Accuracy is judged by the matching of inner and outer ticks along the root paths (solid lines). In the case of RKS4, the root path doubles back on the unit circle in the wrong direction; those tick marks are omitted for clarity. The existence of an RKS7 formula is unknown, and we do not yet have a useful RKS3 method for which this analysis is appropriate.

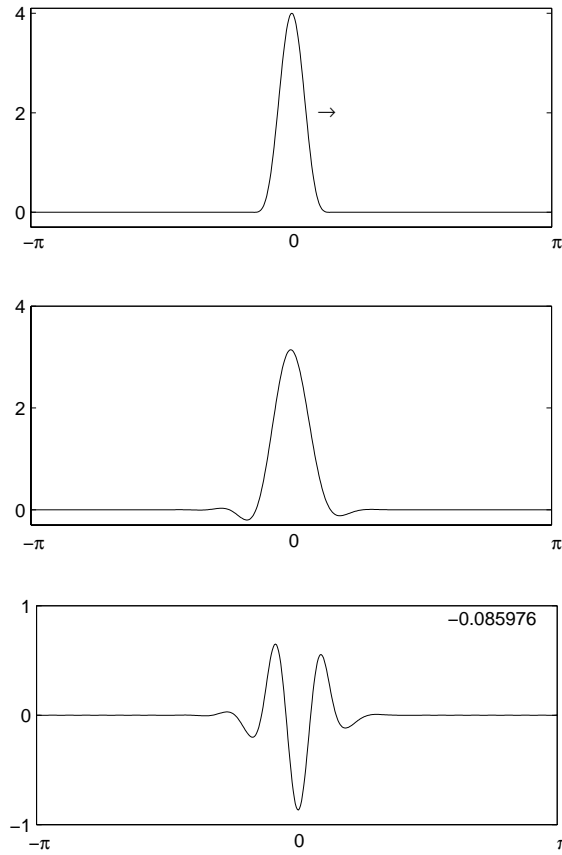


FIG. 8.1. Sample run of ABS3 using the physically relevant root and $N = 375$ function evaluations. (a) Initial condition (and exact final solution). Note that we pick the physical root so that the hump moves to the right. (b) Numerical solution at final time $T = 6\pi$. (c) Error in the numerical solution. The relative loss $(\frac{\|u_{final}\|_2}{\|u_{initial}\|_2} - 1)$ is shown in the upper right-hand corner of the error plot.

Figure 8.1 shows a sample run of this method for third-order ABS using $N = 375$: the initial condition (and exact solution at time $T = 6\pi$), the numerical solution at $T = 6\pi$, and the error in the numerical solution. In our comparison charts, we show only the error in the numerical solution. In order to address the numerical dissipation of the schemes, we have included the relative loss of energy in the discrete L^2 -norm in the upper-right corner of the error plots.

Table 8.1 shows the error in running the second-order leapfrog (ABS2, BDS2, RKS2) method for $N = 500$ and $N = 1000$. Table 8.2 compares the errors obtained by running AB3, ABS3, and BDS3 for $N = 500$ and $N = 1000$, while Table 8.3 shows the errors resulting from running AB4, RK4, ABS4, BDS4, and RKS4 for $N = 800$ and $N = 1600$. Finally, Table 8.4 compares the errors in AB7 and ABS7 for $N = 2000$ and $N = 4000$, while Table 8.5 compares the errors in AB8 and ABS8 for $N = 3000$ and $N = 6000$.

In all cases, staggered methods are superior to nonstaggered methods in terms of accuracy and stability. While the relative accuracy of nonstaggered versus stag-

TABLE 8.1

Error when running the second-order leapfrog method using the root portrait technique. Leapfrog is the only classical second-order multistep method that has a nonzero ISB. Note that vertical scales differ by 10/3. The relative loss $(\frac{\|u_{\text{final}}\|_2}{\|u_{\text{initial}}\|_2} - 1)$ is shown in the upper right-hand corner of the error plots.

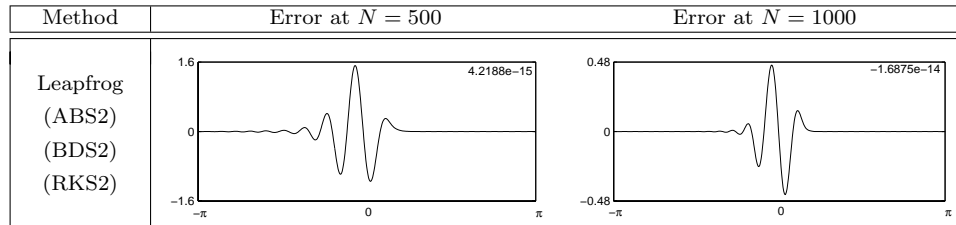
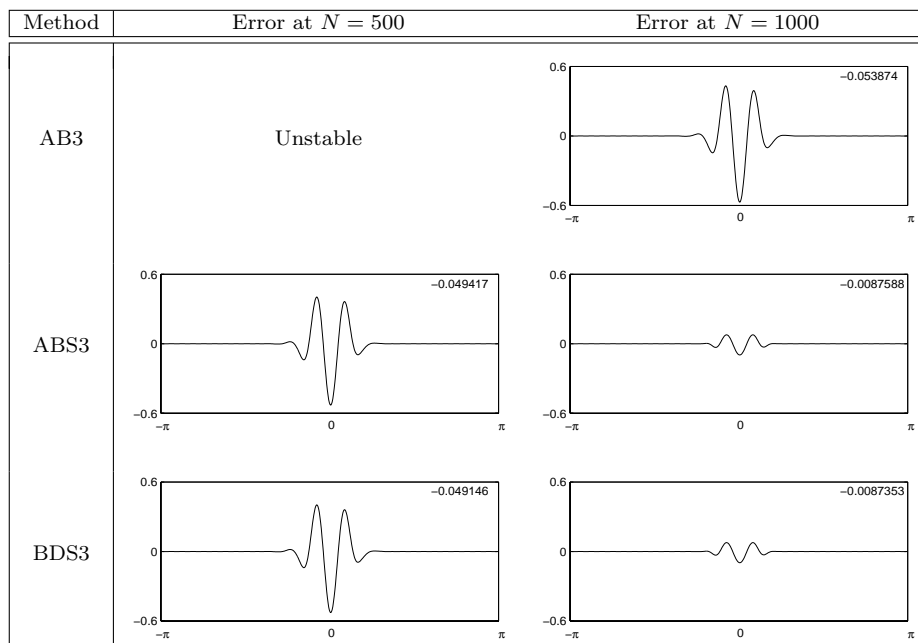


TABLE 8.2

Error when running third-order methods using the root portrait technique. Observe that the vertical scales are the same in all cases and that AB3 is not stable until $N > 834$ for $M = 64$. The relative loss $(\frac{\|u_{\text{final}}\|_2}{\|u_{\text{initial}}\|_2} - 1)$ is shown in the upper right-hand corner of the error plots.

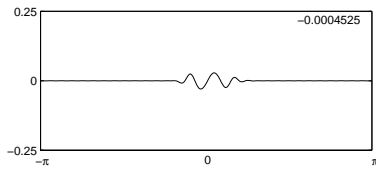
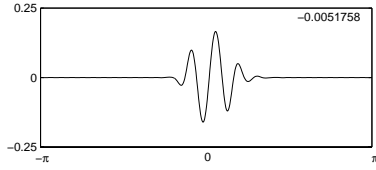
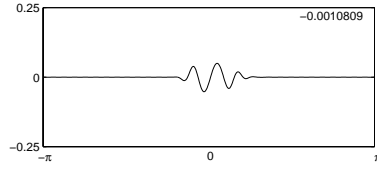
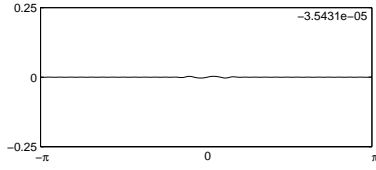
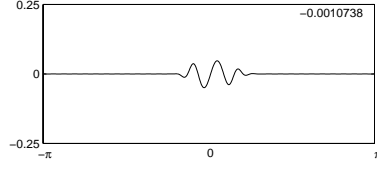
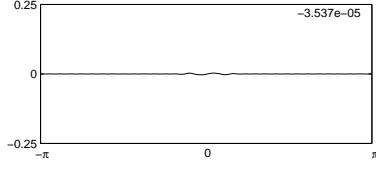
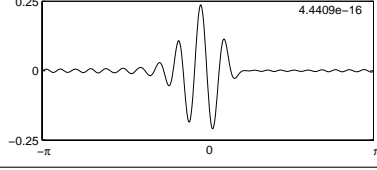
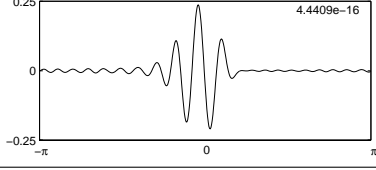


gered methods does not change with order, the improvement in stability from using staggered methods continues to improve as order increases. It is also interesting to note that the RKS4 method is inferior to ABS4 and BDS4 in accuracy but marginally better in stability, whereas it improves on RK4 in both respects.

Notice that there is a different character to the error than is customary. Typically, error trains are one sided due to spatial discretization error. However, as noted in the previous section, we have eliminated spatial discretization errors through the use of the root portrait technique. Thus, the errors shown in these pictures are solely time discretization errors. These error trains are almost symmetric rather than one

TABLE 8.3

Error when running fourth-order methods using the root portrait technique. Note that the vertical scales are the same in all cases and that AB4 is not stable until $N > 1403$ and RK4 is not stable until $N > 854$ for $M = 64$. N is the number of function evaluations used to reach the final time $T = 6\pi$. The relative loss $(\frac{\|u_{\text{final}}\|_2}{\|u_{\text{initial}}\|_2} - 1)$ is shown in the upper right-hand corner of the error plots.

Method	Error at $N = 800$	Error at $N = 1600$
AB4	Unstable	
RK4	Unstable	
ABS4		
BDS4		
RKS4		

sided since the schemes are almost dispersion-free. The amount of dissipation is on the order of machine precision for leapfrog and RKS4 and is reasonably small for the other schemes.

9. Conclusions. We have introduced staggered time integrators for solving systems of wave equations. We find that the staggered versions of Adams–Bashforth and backward differentiation methods have significantly smaller local truncation errors and greater ISBs than their nonstaggered counterparts. In addition, staggered schemes are no more difficult to implement than nonstaggered schemes. We have also

TABLE 8.4

Error when running seventh-order methods using the root portrait technique. *AB7 is not stable until $N > 10384$ for $M = 64$. Note that vertical scales differ by 100. The relative loss $(\frac{\|u_{\text{final}}\|_2}{\|u_{\text{initial}}\|_2} - 1)$ is shown in the upper right-hand corner of the error plots.*

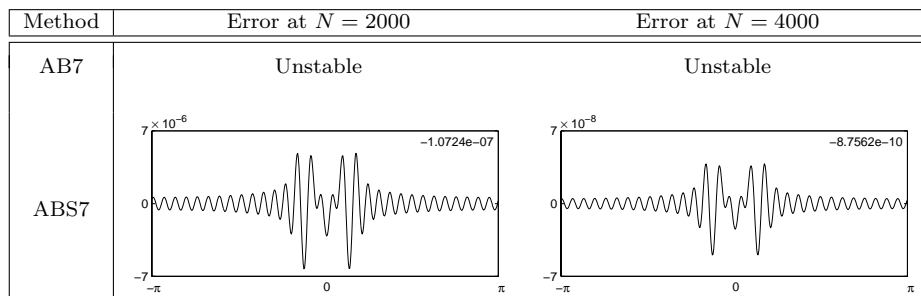
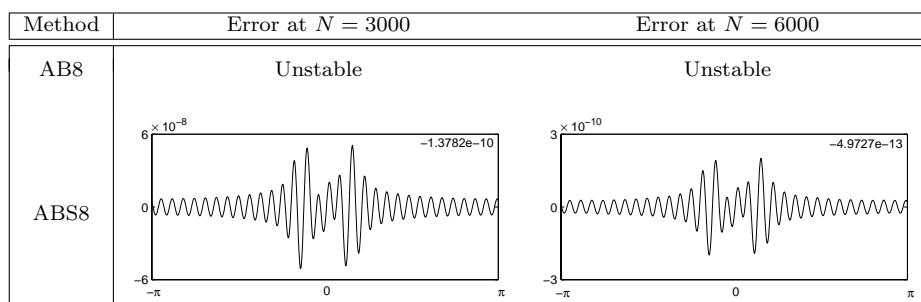


TABLE 8.5

Error given by running eighth-order methods using the root portrait technique. *AB8 is not stable until $N > 20455$ for $M = 64$. Note that vertical scales differ by 200. The relative loss $(\frac{\|u_{\text{final}}\|_2}{\|u_{\text{initial}}\|_2} - 1)$ is shown in the upper right-hand corner of the error plots.*



considered free parameter multistep methods that allow for additional improvement in the ISB. RKS methods also show promise for treating hyperbolic systems. We have introduced a low-storage fourth-order method that has twice the ISB and a much smaller error constant than the classical fourth-order RK method. Table 9.1 summarizes our results concerning staggered fourth-order methods and compares them to other explicit nonstaggered fourth-order methods. Experimental results verify the feasibility of these new methods.

Appendix A. Glossary of abbreviations and terms used in this paper.

AB_p	Adams–Bashforth method of order p
ABS_p	Staggered Adams–Bashforth method of order p
BD_p	Backward differentiation method of order p
BDS_p	Staggered backward differentiation method of order p
RK_p	Runge–Kutta method of order p
RKS_p	Staggered Runge–Kutta method of order p
ISB	Imaginary stability boundary—The largest value of S_I such that the interval $[-iS_I, iS_I]$ is contained in the stability domain of a time-stepping scheme. (For RK methods, normalized by s , the number of stages.)
error constant	Coefficient C that gives an estimate of local truncation error to be expected from a method; the local truncation error is given by $Ck^{p+1}f^{(p+1)}(\xi)$, where p is the order of the method. (To obtain an adequate estimate of global error, normalize by $\sigma(1)$ for multistep methods. To obtain a valid comparison, multiply by s^p for an s -stage method.)

TABLE 9.1

Comparison of fourth-order classical nonstaggered vs. staggered time integrators. The normalized local truncation error is $Ck^5 f^{(5)}(\eta)$, where C is the normalized error constant.

Nonstaggered				Staggered			
Name	Stencil	Normalized ISB	Normalized error constant	Name	Stencil	Normalized ISB	Normalized error constant
						1.0000	≈ 0.0370
	(BD4 implicit)			BDS4			
AB4		≈ 0.430	≈ 0.3486	ABS4		≈ 1.3333	≈ 0.0387
RK4		≈ 0.7071	≈ 2.1333	RKS4		≈ 1.425	≈ 0.1333

REFERENCES

- [1] G. DAHLQUIST, *Convergence and stability in the numerical integration of ordinary differential equations*, Math. Scand., 4 (1956), pp. 33–53.
- [2] B. FORNBERG, *High-order finite differences and the pseudospectral method on staggered grids*, SIAM J. Numer. Anal., 27 (1990), pp. 904–918.
- [3] B. FORNBERG, *Calculation of weights in finite difference formulas*, SIAM Rev., 40, (1998), pp. 685–691.
- [4] B. FORNBERG AND M. GHRIST, *Spatial finite difference approximations for wave-type equations*, SIAM J. Numer. Anal., 37 (1999), pp. 105–130.
- [5] M. GHRIST, *Finite Difference Methods for Wave Equations*, Ph.D. thesis, University of Colorado, Boulder, CO, 2000.
- [6] E. HAIRER, S.P. NØRSETT, AND G. WANNER, *Solving Ordinary Differential Equations I*, Springer-Verlag, Berlin, 1991.
- [7] R. JELTSCH AND O. NEVANLINNA, *Stability of explicit time discretizations for solving initial value problems*, Numer. Math., 37 (1981), pp. 61–91.
- [8] R. JELTSCH AND O. NEVANLINNA, *Dahlquist’s first barrier for multistage multistep formulas*, BIT, 24 (1984), pp. 538–555.
- [9] A. TAFLOVE, *Computational Electrodynamics: The Finite-difference Time-domain Method*, Artech House, Boston, 1995.