Journal of Computational Physics **176,** 430–455 (2002)
doi:10.1006/jcph.2002.6995, available online at http://www.idealibrary.com on IDEAL®

Exponential Time Differencing for Stiff Systems

S. M. Cox and P. C. Matthews

School of Mathematical Sciences, University of Nottingham, University Park, Nottingham NG7 2RD, UK
E-mail: stephen.cox@nottingham.ac.uk, paul.matthews@nottingham.ac.uk

Received December 1, 2000; revised August 6, 2001

We develop a class of numerical methods for stiff systems, based on the method of exponential time differencing. We describe schemes with second- and higher-order accuracy, introduce new Runge–Kutta versions of these schemes, and extend the method to show how it may be applied to systems whose linear part is nondiagonal. We test the method against other common schemes, including integrating factor and linearly implicit methods, and show how it is more accurate in a number of applications. We apply the method to both dissipative and dispersive partial differential equations, after illustrating its behavior using forced ordinary differential equations with stiff linear parts. © 2002 Elsevier Science (USA)

Key Words: stiff systems; exponential time differencing; integrating factor methods.

1. INTRODUCTION

Stiff systems of ordinary differential equations (ODEs) arise commonly when solving partial differential equations (PDEs) by spectral methods, and their numerical solution requires special treatment if accurate solutions are to be found efficiently. In this paper, we develop and test a class of numerical methods for integrating stiff systems, based on exact integration of the linear part of the equations. These methods differ from the usual integrating factor methods in their treatment of the nonlinear terms and are considerably more accurate than those methods, as we demonstrate in this paper.

When solving a PDE subject to spatially periodic boundary conditions, it is natural to write the solution as a sum of Fourier modes with time-dependent coefficients. The corresponding spectral and pseudo-spectral methods have been shown to be remarkably successful for a wide range of applications [3–5, 21]. However, the set of ODEs for the mode amplitudes is stiff, because the time scale associated with the nth mode scales as $O(n^{-m})$ for large n, where m is the order of the highest spatial derivative, so that the highest modes evolve on short time scales. It is generally the case that the linear terms are primarily responsible for the stiffness of a system of ODEs for modes in a spectral simulation. Several important PDEs, such as the Cahn–Hilliard equation [22] or the Kuramoto–Sivashinsky equation [12],

0021-9991/02 $35.00

involve four spatial derivatives ($m = 4$). In fact, the methods discussed in this paper were derived while considering model PDEs with six spatial derivatives [14, 15], so that $m = 6$, and particularly careful consideration of the time-stepping method is required.

In problems where the boundary conditions are not periodic, a basis other than Fourier modes may be appropriate (e.g., Chebyshev polynomials) and the linearized system may no longer be diagonal; in this case, the stiffness problem is exacerbated [20].

The numerical method treated in this paper is the so-called "exponential time differencing" (ETD) scheme [9, 17, 18], which involves exact integration of the governing equations followed by an approximation of an integral involving the nonlinear terms. It arose originally in the field of computational electrodynamics [19], where the problem of computing the electric and magnetic fields in a box with absorbing boundaries is stiff (essentially because of the large value of c, the speed of light). For that problem, standard, explicit time-stepping techniques require an extremely small time step in order to be stable, and although implicit schemes with less stringent constraints on the time step are available, they are costly (or infeasible) to implement in three dimensions [9]. Explicit "exponential time differencing" [9, 17, 18] with first-order accuracy has been used widely in computational electrodynamics. This original first-order explicit scheme has since been extended to implicit and explicit schemes of arbitrary order, and the stability of such schemes has been discussed in some detail [1]. To develop ETD methods further, in this paper we derive new, more accurate (Runge–Kutta) ETD methods and provide a more succinct derivation of ETD methods than previously given. We also apply ETD methods to various PDEs and compare them with alternative time-stepping methods to illustrate the superior performance of ETD schemes, and show how ETD methods can be applied to systems of ODEs whose linear part is not diagonal. All ETD schemes discussed in this paper are explicit.

The structure of this paper is as follows. In Section 2, we describe the first- and second-order-accurate exponential time differencing scheme and give a derivation of ETD schemes of arbitrary order. We then describe the new Runge–Kutta ETD methods. Other schemes for stiff systems are also described for comparison, and in Section 3 we discuss the stability of the various schemes. In Section 4, we compare the second- and fourth-order ETD schemes with other methods, for several stiff problems. Our primary interest lies in solving PDEs, and we give comparisons for both dissipative and dispersive PDEs. However, to illustrate the behavior of the method, we also provide simpler and more detailed examples in the form of a single, linear, inhomogeneous ODE with either a real or an imaginary linear part. A final set of examples concerns problems in which the linearized system is nondiagonal, and shows how the method can readily be adapted to this important case. We summarize our results in Section 5.

2. DERIVATION OF METHODS

When a PDE is discretized using a Fourier spectral method, a stiff system of coupled ODEs for the Fourier coefficients is obtained. The linear part of this system is diagnonal, while the nonlinear terms are usually evaluated by transforming to physical space, evaluating the nonlinear terms at grid points and then transforming back to spectral space.

In stiff systems, solutions evolve on two time scales. If the stiffness is due to rapid exponential decay of some modes (as with a dissipative PDE), then there is a rapid approach to a "slow manifold," followed by slower evolution along the slow manifold itself [2]. If, by contrast, the stiffness is due to rapid oscillations of some modes (as with a dispersive PDE), then the solution rapidly oscillates about its projection on the slow manifold; it is this projection which evolves slowly. In general, stiffness may have features of both rapid decay and rapid oscillation.

Although our primary interest lies in solving PDEs, it is clearer and more instructive first to describe ETD methods in the context of a simple model ODE for the evolution of a single Fourier mode. Since the linear operator in a Fourier basis is diagonal, the extension of the method to the system of ODEs for the mode amplitudes is then immediate. The model ODE is

$$\dot{u} = cu + F(u, t), \tag{1}$$

where c is a constant and $F(u, t)$ represents nonlinear and forcing terms. For the high-order Fourier modes, c is large and negative (for dissipative PDEs) or large and imaginary (for dispersive PDEs). A suitable time-stepping method for (1) should be able to handle the stiffness caused by the large values of $|c|$ without requiring time steps of order $|1/c|$. However, since the coefficients c span a wide range of values when all Fourier modes are considered, the time-stepping method should also be applicable to small values of $|c|$. Finally, we require that the term $F(u, t)$ be handled explicitly, since fully implicit methods are too costly for large-scale PDE simulations.

When $|c| \gg 1$, solutions of (1) generally consist of two elements: a fast phase in which $u = O(1)$ and $\mathrm{d}/\mathrm{d}t = O(c)$, and a "slow manifold" on which $u = O(1/c)$ and $\mathrm{d}/\mathrm{d}t = O(1)$; if $Re(c) < 0$, solutions are attracted to the slow manifold. The slow manifold can be expressed as an asymptotic series in powers of $1/c$ as

$$u \sim -\frac{F}{c} - \frac{1}{c^2}\frac{\mathrm{d}F}{\mathrm{d}t} - \frac{1}{c^3}\frac{\mathrm{d}^2F}{\mathrm{d}t^2}\cdots, \tag{2}$$

and forms the basis of "nonlinear Galerkin" methods [3, 13]. Since initial conditions do not generally lie on the slow manifold, a numerical method should ideally give stable, highly accurate solutions during both the fast and slow phases.

2.1. Exponential Time Differencing

To derive the exponential time differencing (ETD) methods [1], we begin by multiplying (1) through by the integrating factor e^{-ct}, then integrating the equation over a single time step from $t = t_n$ to $t = t_{n+1} = t_n + h$ to give

$$u(t_{n+1}) = u(t_n)e^{ch} + e^{ch}\int_0^h e^{-c\tau} F(u(t_n + \tau), t_n + \tau)\,\mathrm{d}\tau. \tag{3}$$

This formula is *exact*, and the essence of the ETD methods is in deriving approximations to the integral in this expression.

We denote the numerical approximation to $u(t_n)$ by u_n and write $F(u_n, t_n)$ as F_n. The simplest approximation to the integral in (3) is that F is constant, $F = F_n + O(h)$, between $t = t_n$ and $t = t_{n+1}$, so that (3) becomes the scheme **ETD1**, given by

$$u_{n+1} = u_n e^{ch} + F_n(e^{ch} - 1)/c, \tag{4}$$

which has a local truncation error $h^2\dot{F}/2$. This version of the exponential time differencing method has been applied in computational electrodynamics [9, 17, 19], but (4) is rarely mentioned outside of this field in the numerical analysis literature (the notable exception being [1]). Note that for small $|c|$, (4) approaches the forward Euler method, while for large $|c|$ the first term in the series (2) is recovered.

If instead of assuming that F is constant over the interval $t_n \leq t \leq t_{n+1}$, we use the higher-order approximation that

$$F = F_n + \tau(F_n - F_{n-1})/h + O(h^2), \tag{5}$$

we arrive at the numerical scheme **ETD2** (cf. [1]) given by

$$u_{n+1} = u_n e^{ch} + F_n((1+hc)e^{ch} - 1 - 2hc)/hc^2 + F_{n-1}(-e^{ch} + 1 + hc)/hc^2, \tag{6}$$

which has a local truncation error of $5h^3\ddot{F}/12$. Note that the apparent divergence of the coefficients in (6) as $c \to 0$ is illusory; in fact (6) becomes the second-order Adams–Bashforth method in this limit. For large $|c|$, (6) gives the first two terms of (2).

For completeness we derive concisely in the next section ETD schemes of arbitrary order (cf. [1]). The availability of high-order ETD schemes represents an important advantage of these methods over standard linearly implicit methods, for which the order is limited by stability constraints.

We now note two practical points that must be borne in mind when applying the ETD methods to PDEs. First, there is often some mode or modes with $c = 0$ (as is the case with (58) and (59) later). In that event, the explicit formulae for the coefficients of F_n, F_{n-1}, etc., in the equivalent of (4) or (6) cannot be used directly, since they involve division by zero. Instead, the limiting form of the coefficients as $c \to 0$ must be used for such a mode. Another practical consideration is that care must be taken in the evaluation of the coefficients for modes where $|ch|$ is small, to avoid rounding errors arising from the large amount of cancellation in the coefficients. This becomes increasingly important as the order of the method is raised; in some cases the Taylor series for the coefficients should be used rather than the explicit formulae themselves, when $|ch| \ll 1$.

2.1.1. *Exponential Time Differencing for Arbitrary Order*

Explicit and implicit ETD schemes of arbitrary order have been derived elsewhere [1], but the derivation is rather involved and does not give explicit formulae for the coefficients. Here we give a more straightforward derivation of the explicit methods, based on a polynomial approximation of the integrand in (3).

Let $G_n(\tau) = F(u(t_n + \tau), t_n + \tau)$. We have used above two approximations for G_n, one constant and one linear in τ. In general we seek approximations to G_n that are polynomials in τ, valid on the interval $0 \leq \tau \leq h$, using information about F at the nth and previous time steps. To derive a numerical scheme with local truncation error of order h^{s+1}, we note [8] that the approximating polynomial $\sum_{m=0}^{s-1} G_n^{(m)}\tau^m$ of degree $s-1$ can be written as

$$\left[1 - \binom{-\tau/h}{1}\nabla + \binom{-\tau/h}{2}\nabla^2 + \cdots + (-1)^{s-1}\binom{-\tau/h}{s-1}\nabla^{s-1}\right]G_n(0), \tag{7}$$

where ∇ is the backwards difference operator, such that

$$\nabla G_n(0) = G_n(0) - G_{n-1}(0), \quad \nabla^2 G_n(0) = G_n(0) - 2G_{n-1}(0) + G_{n-2}(0), \tag{8}$$

etc., and

$$m!\binom{-\tau/h}{m} = (-\tau/h)(-\tau/h - 1)\cdots(-\tau/h - m + 1), \tag{9}$$

for $m = 1, \ldots, s-1$.

It then follows from (3) that our approximation to u_{n+1} satisfies

$$\begin{aligned} u_{n+1} - u_n e^{ch} &= e^{ch}\int_0^h e^{-c\tau}\left[1 - \binom{-\tau/h}{1}\nabla + \cdots + (-1)^{s-1}\binom{-\tau/h}{s-1}\nabla^{s-1}\right] G_n(0)\,\mathrm{d}\tau \\ &= e^{ch}\sum_{m=0}^{s-1}\int_0^h (-1)^m e^{-c\tau}\binom{-\tau/h}{m}\,\mathrm{d}\tau\,\nabla^m G_n(0) \\ &= h\sum_{m=0}^{s-1}(-1)^m\int_0^1 e^{ch(1-\lambda)}\binom{-\lambda}{m}\,\mathrm{d}\lambda\,\nabla^m G_n(0) \\ &= h\sum_{m=0}^{s-1} g_m \nabla^m G_n(0), \end{aligned} \tag{10}$$

where

$$g_m = (-1)^m\int_0^1 e^{ch(1-\lambda)}\binom{-\lambda}{m}\,\mathrm{d}\lambda. \tag{11}$$

It remains then to calculate the g_m. This is straightforwardly accomplished by introducing the generating function

$$\Gamma(z) = \sum_{m=0}^{\infty} g_m z^m, \tag{12}$$

which is readily found to be

$$\begin{aligned} \Gamma(z) &= \int_0^1 e^{ch(1-\lambda)}\sum_{m=0}^{\infty}\binom{-\lambda}{m}(-z)^m\,\mathrm{d}\lambda \\ &= \int_0^1 e^{ch(1-\lambda)}(1-z)^{-\lambda}\,\mathrm{d}\lambda \\ &= \frac{e^{ch}(1 - z - e^{-ch})}{(1-z)(ch + \log(1-z))}, \end{aligned} \tag{13}$$

provided the order of the sum and integral may be interchanged.

A recurrence relation for the g_m can be found by rearranging (13) to the form

$$(ch + \log(1-z))\Gamma(z) = e^{ch} - (1-z)^{-1} \tag{14}$$

and expanding as a power series in z to give

$$\left(ch - z - \frac{1}{2}z^2 - \frac{1}{3}z^3 - \cdots\right)(g_0 + g_1 z + g_2 z^2 + \cdots) = e^{ch} - 1 - z - z^2 - z^3 - \cdots. \tag{15}$$

Thus, by equating like powers of z we find that

$$chg_0 = e^{ch} - 1, \tag{16}$$

$$chg_{m+1} + 1 = g_m + \frac{1}{2}g_{m-1} + \frac{1}{3}g_{m-2} + \cdots + \frac{g_0}{m+1} = \sum_{k=0}^{m} \frac{g_k}{m+1-k}, \tag{17}$$

for $m \geq 0$. For example,

$$g_0 = \frac{e^{ch} - 1}{ch} \quad \text{and} \quad g_1 = \frac{g_0 - 1}{ch} = \frac{e^{ch} - 1 - ch}{c^2h^2}, \tag{18}$$

which give rise to the scheme (6).

Having obtained the g_m, the ETD scheme (10) is given explicitly as

$$u_{n+1} = u_n e^{ch} + h \sum_{m=0}^{s-1} g_m \sum_{k=0}^{m} (-1)^k \binom{m}{k} F_{n-k}. \tag{19}$$

2.2. Exponential Time Differencing Method with Runge–Kutta Time Stepping

The ETD methods described above are of multistep type, requiring s previous evaluations of the nonlinear term F. Such methods are often inconvenient to use, since initially only one value is available. This problem can be avoided by the use of Runge–Kutta (RK) methods, which also typically have the advantages of smaller error constants and larger stability regions than multistep methods. In this section we obtain ETD methods of RK type of orders 2, 3, and 4. As usual for RK methods, these are not unique.

2.2.1. *Second-Order Runge–Kutta ETD Method*

A second-order ETD method of RK type, analogous to the "improved Euler" method, is as follows. First, the step (4) is taken to give

$$a_n = u_n e^{ch} + F_n(e^{ch} - 1)/c. \tag{20}$$

Then the approximation

$$F = F(u_n, t_n) + (t - t_n)(F(a_n, t_n + h) - F(u_n, t_n))/h + O(h^2) \tag{21}$$

is applied on the interval $t_n \leq t \leq t_{n+1}$, and is substituted into (3) to yield the scheme **ETD2RK** given by

$$u_{n+1} = a_n + (F(a_n, t_n + h) - F_n)(e^{ch} - 1 - hc)/hc^2. \tag{22}$$

The truncation error per step for this method is $-h^3\ddot{F}/12$; note that this is smaller by a factor of 5 than that of ETD2.

2.2.2. *Third- and Fourth-Order Runge–Kutta ETD Methods*

A third-order ETD RK scheme can be constructed in a similar way, analogous to the classical third-order RK method (see, for example, [10]): **ETD3RK** is given by

$$\begin{aligned} a_n &= u_n e^{ch/2} + \left(e^{ch/2} - 1\right) F(u_n, t_n)/c, && (23)\\ b_n &= u_n e^{ch} + (e^{ch} - 1)(2F(a_n, t_n + h/2) - F(u_n, t_n))/c, && (24)\\ u_{n+1} &= u_n e^{ch} + \{F(u_n, t_n)[-4 - hc + e^{ch}(4 - 3hc + h^2c^2)] \\ &\quad + 4F(a_n, t_n + h/2)[2 + hc + e^{ch}(-2 + hc)] \\ &\quad + F(b_n, t_n + h)[-4 - 3hc - h^2c^2 + e^{ch}(4 - hc)]\}/h^2c^3. && (25) \end{aligned}$$

The terms a_n and b_n approximate the values of u at $t_n + h/2$ and $t_n + h$, respectively. The final formula (25) is the quadrature formula for (3) derived from quadratic interpolation through the points t_n, $t_n + h/2$, and $t_n + h$.

A straightforward extension of the standard fourth-order RK method yields a scheme which is only third order. However, by varying the scheme and introducing further parameters, a fourth-order scheme **ETD4RK** is obtained:

$$\begin{aligned} a_n &= u_n e^{ch/2} + \left(e^{ch/2} - 1\right) F(u_n, t_n)/c, && (26)\\ b_n &= u_n e^{ch/2} + \left(e^{ch/2} - 1\right) F(a_n, t_n + h/2)/c, && (27)\\ c_n &= a_n e^{ch/2} + \left(e^{ch/2} - 1\right)(2F(b_n, t_n + h/2) - F(u_n, t_n))/c, && (28)\\ u_{n+1} &= u_n e^{ch} + \{F(u_n, t_n)[-4 - hc + e^{ch}(4 - 3hc + h^2c^2)] \\ &\quad + 2(F(a_n, t_n + h/2) + F(b_n, t_n + h/2))[2 + hc + e^{ch}(-2 + hc)] \\ &\quad + F(c_n, t_n + h)[-4 - 3hc - h^2c^2 + e^{ch}(4 - hc)]\}/h^2c^3. && (29) \end{aligned}$$

The computer algebra package Maple was used to confirm that this method is indeed fourth order.

2.3. Standard Integrating Factor Methods: IFAB2 and IFRK2

Standard integrating factor (IF) methods [3, 4, 6, 21] are obtained by rewriting (1) as

$$\frac{\mathrm{d}}{\mathrm{d}t} u e^{-ct} = F(u, t) e^{-ct} \tag{30}$$

and then applying a time-stepping scheme to this equation. If we use the second-order Adams–Bashforth method we obtain the IF method

$$\textbf{IFAB2} \quad u_{n+1} = u_n e^{ch} + \frac{3h}{2} F_n e^{ch} - \frac{h}{2} F_{n-1} e^{2ch}. \tag{31}$$

The local truncation error for this method is

$$\frac{5}{12} h^3 \frac{\mathrm{d}^2(F e^{-ct})}{\mathrm{d}t^2} \sim \frac{5}{12} h^3 c^2 F \quad \text{as } |c| \to \infty. \tag{32}$$

Applying instead the second-order Runge–Kutta method to (30), we find

$$\textbf{IFRK2}\quad u_{n+1} = u_n e^{ch} + \frac{h}{2}(F_n e^{ch} + F((u_n + hF_n)e^{ch}, t_n + h)), \tag{33}$$

with a truncation error of the same order as IFAB2.

Although they are commonly used, integrating factor methods have a number of weaknesses. Unlike most standard methods and the ETD methods described above, the fixed points of IF methods are not the same as the fixed points of the original ODE. A second important drawback [3, 7] is the large error constants; the errors in the above IF methods are greater than those of the ETD methods by a factor of order $c^2 \gg 1$.

2.4. Linearly Implicit Schemes

Finally, we consider two “linearly implicit” (LI) or “semi-implicit” schemes—these are generally regarded as being very well suited to systems such as (1) [3, 6, 7, 22]. Treating the linear terms with the second-order Adams–Moulton method (trapezium rule) and the nonlinear terms with the second-order Adams–Bashforth method gives

$$\textbf{AB2AM2}\quad u_{n+1} = u_n + \frac{h}{2}(cu_n + cu_{n+1}) + \frac{3h}{2}F_n - \frac{h}{2}F_{n-1}. \tag{34}$$

Note that in this case the formulae for the two methods can simply be added together. The second-order backward difference formula can also be combined with AB2, although not in such a straightforward way. The resulting method, obtained by the method of undetermined coefficients and Taylor expansion, is

$$\textbf{AB2BD2}\quad u_{n+1} = (4u_n - u_{n-1} + 4hF_n - 2hF_{n-1})/(3 - 2hc). \tag{35}$$

This method has been used previously for a spectral simulation of the Cahn–Hilliard equation [22]. The truncation errors for the linearly implicit methods (34) and (35) have the same scaling. For solutions that lie off the slow manifold, the error per step is of order c^3h^3 for large $|c|$, but for solutions on the slow manifold the error is smaller, of order h^3.

3. STABILITY

In this section we compare the stability of several of the second-order methods described above. The general approach for stability analysis of a numerical method that uses different methods for the linear and nonlinear parts of the equation is as follows (see for example, [1, 6]). For the nonlinear, autonomous ODE

$$\dot{u} = cu + F(u), \tag{36}$$

we suppose that there is a fixed point u_0, so that $cu_0 + F(u_0) = 0$. Linearizing about this fixed point leads to

$$\dot{u} = cu + \lambda u, \tag{37}$$

where u is now the perturbation to u_0 and where $\lambda = F'(u_0)$. If (36) represents a system of ODEs, then λ is a diagonal or block diagonal matrix containing the eigenvalues of F. The fixed point u_0 is stable if $Re(c + \lambda) < 0$ for all eigenvalues λ.

When a second-order numerical method, for example, ETD2, is applied to (36), the linearization of the nonlinear term in the numerical method leads to a recurrence relation involving u_{n+1}, u_n, and u_{n-1}. This is equivalent to applying the method to (37), with the term λu regarded as the nonlinear term. Note that an implicit assumption in this approach is that the fixed points of the numerical method are the same as those of the ODE. This is true for ETD and LI methods, but not for the IF methods; it follows that the meaning of the stability analysis for IF methods is not clear.

In general, both c and λ in (37) are complex, so the stability region for these methods is four dimensional. In order to plot two-dimensional stability regions, previous authors have used the complex λ-plane, assuming c to be fixed and real [1], or have assumed that both c and λ are purely imaginary [6]. An alternative approach, used below, is to concentrate on the case in which c and λ are real.

Consider first the method AB2AM2 applied to (37). This leads to

$$u_{n+1} = u_n + \frac{h}{2}(cu_n + cu_{n+1}) + \frac{3h}{2}\lambda u_n - \frac{h}{2}\lambda u_{n-1}. \tag{38}$$

After defining $r = u_{n+1}/u_n$, $x = \lambda h$, $y = ch$, we find the following quadratic equation for the factor r by which the solution is multiplied after each step:

$$(2-y)r^2 - (2+3x+y)r + x = 0. \tag{39}$$

With the assumption that x and y are real, it can be shown that r is real, so that the stability boundaries correspond to $r = 1$ and $r = -1$ in (39). These correspond to the lines $x + y = 0$ and $x = -1$ in the x, y plane, respectively.

Similarly, for the method AB2BD2, the quadratic for r is

$$(3-2y)r^2 - 4(1+x)r + 2x + 1 = 0 \tag{40}$$

and the stability boundaries are the lines $x + y = 0$ and $y = 4 + 3x$. For ETD2, the equation for r is

$$y^2r^2 - (y^2e^y + x[(1+y)e^y - 1 - 2y])r + x(e^y - 1 - y) = 0 \tag{41}$$

and the stability boundaries are the lines $x + y = 0$ and

$$x = \frac{-y^2(1+e^y)}{ye^y + 2e^y - 2 - 3y}. \tag{42}$$

Finally, for the ETD2RK method, we have

$$r = e^y + \frac{x}{y}(e^y - 1) + \frac{x}{y^3}(x+y)(e^y - 1)(e^y - 1 - y) \tag{43}$$

and the stability region is bounded by two lines on which $r = 1$, which are $x + y = 0$ and $x = -y^2/(e^y - 1 - y)$.

The stability regions for these four methods are shown in Fig. 1. Note that for all four methods the stability region includes the negative y-axis, and the width of this region increases as $|y|$ increases. The right-hand boundary is the same for all four methods, corresponding simply to the true stability boundary of (37). For both ETD2 and AB2BD2 the

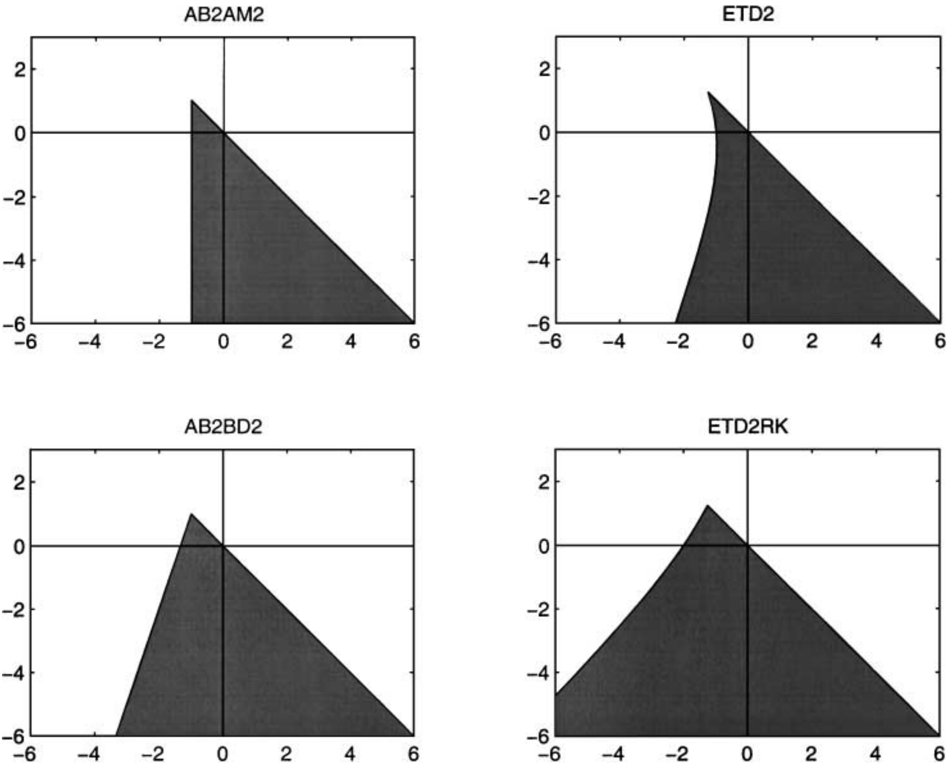

FIG. 1. Stability regions (shaded) in the x, y plane for four methods.

left-hand boundary is parallel to $y = 3x$ as $y \to -\infty$, while for ETD2RK, which has the largest stability region, it is parallel to $y = x$.

An alternative way of presenting the stability regions is in the complex x plane, at a fixed value of y that is real and negative. The boundaries of the stability regions are obtained by substituting $r = e^{i\theta}$ into the formulae (39)–(41) and (43) and then by solving for x. Figure 2 shows the stability regions plotted in this way for the same four methods, with $y = -20$. For each method, the boundary of the stability region passes through the point $x = -y$. As in the purely real case, AB2AM2 has the smallest stability region and ETD2RK has the largest. In the limit $y \to -\infty$, the stability region for ETD2RK simplifies to the disc $|x| < |y|$. In the same limit, the boundaries of the stability regions for ETD2 and AB2BD2 become $x = ye^{2i\theta}/(1 - 2e^{i\theta})$ and that of AB2AM2 becomes $x = -y(e^{2i\theta} + e^{i\theta})/(3e^{i\theta} - 1)$. Note that for AB2AM2 the radius of the stability region does not grow linearly with y at $\theta = \pi$, as is also apparent from Fig. 1.

4. NUMERICAL EXAMPLES AND EXPERIMENTS

In this section the second- and fourth-order ETD methods and the new Runge–Kutta ETD methods described above in Sections 2.1 and 2.2 are compared with the standard methods described in Sections 2.3 and 2.4. Such tests do not seem previously to have been carried out. Our primary goal is to test the ETD methods on dissipative and dispersive partial differential equations, in which the high-wavenumber modes experience rapid decay and rapid oscillation, respectively. As a precursor to these PDE tests, we begin by examining model ordinary differential equations for which the linear part gives either rapid decay or

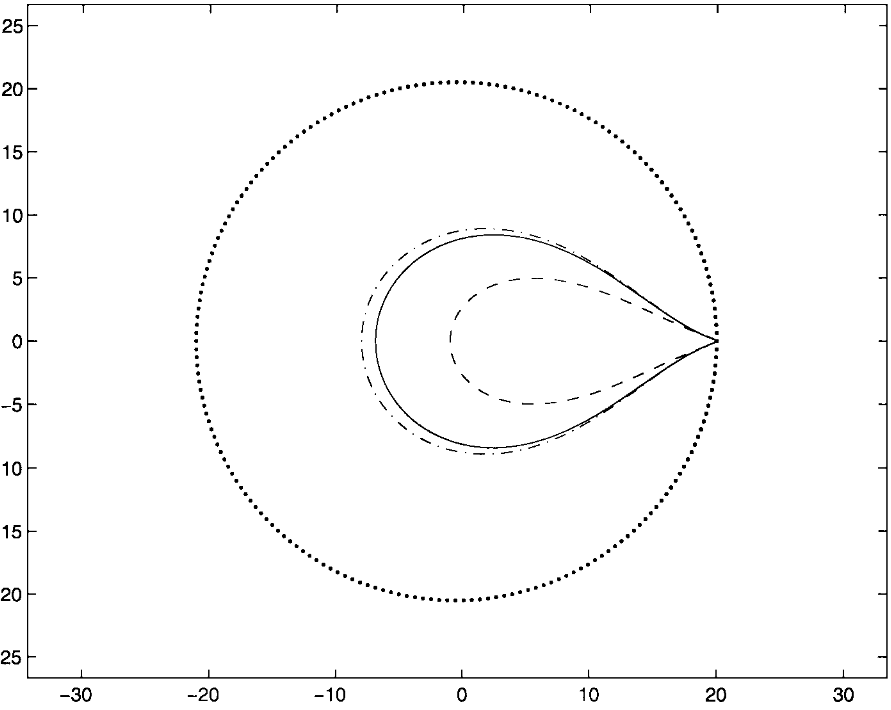

FIG. 2. Stability regions (interior of closed curves) in the complex x plane with $y = -20$. The four methods are AB2AM2 (dashed), ETD2 (solid), AB2BD2 (dash-dot), and ETD2RK (dotted).

oscillation. In these simple models more analytical progress is possible. In order to make a fair comparison between different methods, most examples will focus on the second-order formulae; however, in applications, higher-order schemes may be more appropriate.

4.1. A Model Ordinary Differential Equation with Rapid Decay

For our first comparison between the different methods, we consider the model ODE

$$\dot{u} = cu + \sin t, \qquad u(0) = u_0, \tag{44}$$

where we choose $c = -100$ to generate the rapid linear decay characteristic of stiff systems, compared with the $O(1)$ time scale of the forcing. For subsequent evaluation of the different numerical schemes, we note that the exact solution is

$$u(t) = u_0 e^{ct} + \frac{e^{ct} - c\sin t - \cos t}{1 + c^2}. \tag{45}$$

Equation (44) is one of the simplest possible ODEs of the required form (1), with the key properties of a stiff linear part and a forcing term that does not vary rapidly. For the differential equation (44), the methods under consideration are ETD2, ETD2RK, IFAB2, IFRK2, AB2AM2, and AB2BD2. Note that if the forcing term F in (1) is a constant, then the ETD methods are exact, which makes comparisons rather unfair!

When $c < 0$ and $|c| \gg 1$, the behavior of (44) can be split into a fast phase, during which the solution rapidly approaches the slow manifold on which $u \sim -(\sin t)/c - (\cos t)/c^2 + \cdots$, and then a phase in which the solution moves along this slow manifold. It is necessary to seek numerical methods that capture both of these phases accurately, and work well when the time step h is of the order of $1/c$.

In evaluating the different numerical schemes, a useful feature of (44) is that the recurrence relations resulting from the various numerical methods can be solved exactly (see Section 4.1.1 below). However, a disadvantage of this equation from the point of view of distinguishing between the various numerical schemes is that initial conditions tend to get lost as the solution becomes phase-locked to the forcing term. So poor numerical schemes are "helped along" by the forcing, and can recover from a bad start. Below, in Sections 4.2 and 4.5, we treat systems with self-sustained oscillations, which do not have this drawback; these later problems provide a more demanding test of the various numerical schemes.

4.1.1. *Analytical Solution to the Numerical Schemes Used to Solve (44)*

In evaluating the various numerical schemes, it is helpful to write down and solve the recurrence relations corresponding to each scheme. In solving exactly the various numerical schemes, it is useful to adopt a formulation in complex variables. The numerical schemes ETD2, ETD2RK, IFAB2, IFRK2, and AB2AM2 for (44) then all give rise to recurrence relations of the form

$$w_{n+1} = \alpha w_n + i\beta\gamma^n, \tag{46}$$

where $u_n = \mathrm{Re}(w_n)$. Table I gives the values of α, β, and γ for each numerical scheme. The solution to (46), subject to the initial value $w_0 = u_0$, is

$$w_n = \alpha^n w_0 + \frac{i\beta(\gamma^n - \alpha^n)}{\gamma - \alpha}. \tag{47}$$

It is possible to use this formula to evaluate the accuracy of the various numerical schemes (apart from AB2BD2). To do so, we note that the exact solution (45) may be written as $u(t) = \mathrm{Re}(w(t))$, where

$$w(t) = e^{ct} w_0 - \frac{i(e^{-it} - e^{ct})}{i + c}. \tag{48}$$

TABLE I

The Values of α, β, and γ in the Recurrence Relation (46) Corresponding to Various Numerical Schemes

	α	β	γ
ETD2	e^{ch}	$\dfrac{(1+ch)e^{ch} - (1+2ch)}{hc^2} + \dfrac{-e^{ch} + (1+ch)}{hc^2} e^{ih}$	e^{-ih}
ETD2RK	e^{ch}	$\dfrac{(ch-1)e^{ch} + 1}{hc^2} + \dfrac{e^{ch} - (1+ch)}{hc^2} e^{-ih}$	e^{-ih}
IFAB2	e^{ch}	$\dfrac{1}{2} h e^{ch}(3 - e^{ch} e^{ih})$	e^{-ih}
IFRK2	e^{ch}	$\dfrac{1}{2} h(e^{ch} + e^{-ih})$	e^{-ih}
AB2AM2	$\dfrac{2+ch}{2-ch}$	$\dfrac{h(3 - e^{ih})}{2-ch}$	e^{-ih}

For AB2BD2, the recurrence relation (35) may be written as

$$w_{n+1} = \frac{4w_n - w_{n-1}}{3 - 2ch} + i\beta\gamma^n, \tag{49}$$

where as above $u_n = \mathrm{Re}(w_n)$, the parameters in this equation being $\beta = 2h(2 - e^{ih})/(3 - 2ch)$ and $\gamma = e^{-ih}$. The solution to (49) is then

$$w_n = X\xi^n + Y\eta^n + Z\gamma^n, \tag{50}$$

where

$$\xi = \frac{2 + \sqrt{1 + 2ch}}{3 - 2ch} \quad \text{and} \quad \eta = \frac{2 - \sqrt{1 + 2ch}}{3 - 2ch}. \tag{51}$$

The remaining coefficients in (50) satisfy

$$(\gamma - \xi)(\gamma - \eta)Z = i\beta\gamma \tag{52}$$

with

$$X = (\eta(w_0 - Z) + \gamma Z - w_1)/(\eta - \xi) \tag{53}$$

$$Y = (\xi(Z - w_0) - \gamma Z + w_1)/(\eta - \xi). \tag{54}$$

Note that for AB2BD2, two starting values are needed. In situations where an exact solution is known, this may be used to generate the second starting value; otherwise a Runge–Kutta scheme, for instance, may be used at start-up.

Figure 3 shows the magnitude of the relative error $(u_{\mathrm{num}} - u_{\mathrm{exact}})/u_{\mathrm{exact}}$ in the numerical solution to (44), with $u_0 = 1$ (off the slow manifold), at $t = \pi/2$ for the six methods above.

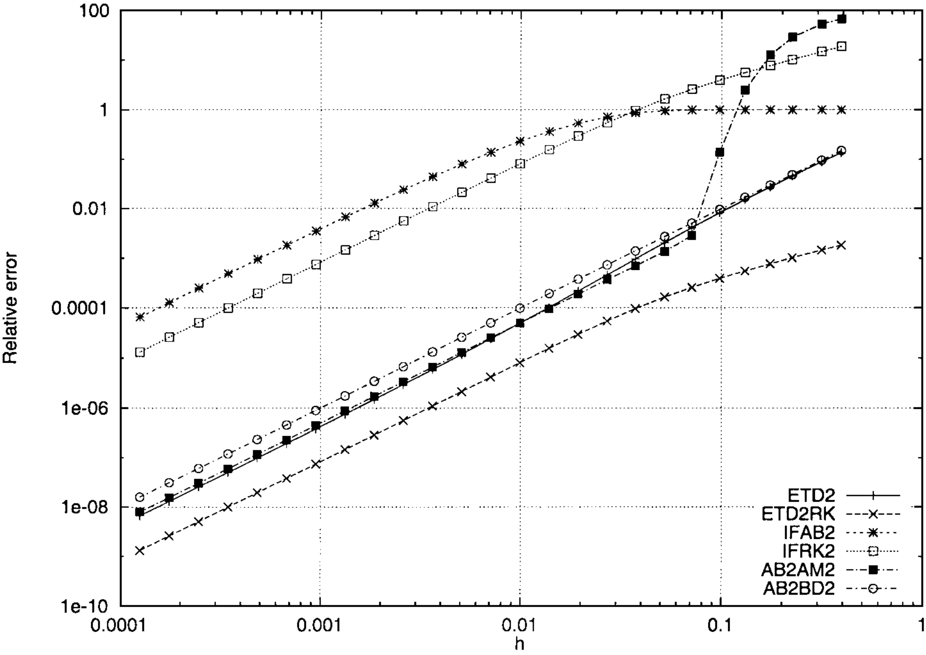

FIG. 3. Magnitude of the relative error at $t = \pi/2$ in (44) with $u_0 = 1$ for six methods.

TABLE II
The Relative Error at $t = \pi/2$ in the Numerical Solution to (44), with $u_0 = 1$ and $c = -100$, is Given by kh^2, Where k is the Constant Given in the Table

	IFAB2	IFRK2	AB2BD2	AB2AM2	ETD2	ETD2RK
k	−4167.08	833.417	1.0000	0.5000	0.4167	−0.0833

It is a straightforward matter to calculate, using the results above, this relative error in the limit as $h \to 0$. In this limit there is a clear ranking of the various schemes—the results are shown in Table II. Note that for each method the relative error is smaller by a factor of c than that expected from the truncation error per step; this is because the errors introduced are exponentially damped as the calculation proceeds.

The most striking feature of Fig. 3 and Table II is the very poor performance of the integrating factor methods IFAB2 and IFRK2; the error in these methods is a factor of approximately 10^3–10^4 greater than that of the other methods. This is because the truncation error per step in these methods is of order c^2h^3, while for the other methods it is of order $h^3 \ll c^2h^3$. Other authors have pointed out the weakness of integrating factor methods [3, 6, 7].

The relative error for small h is shown in Fig. 4 for different values of c: The poor performance of the integrating factor methods is again illustrated in this figure. The LI

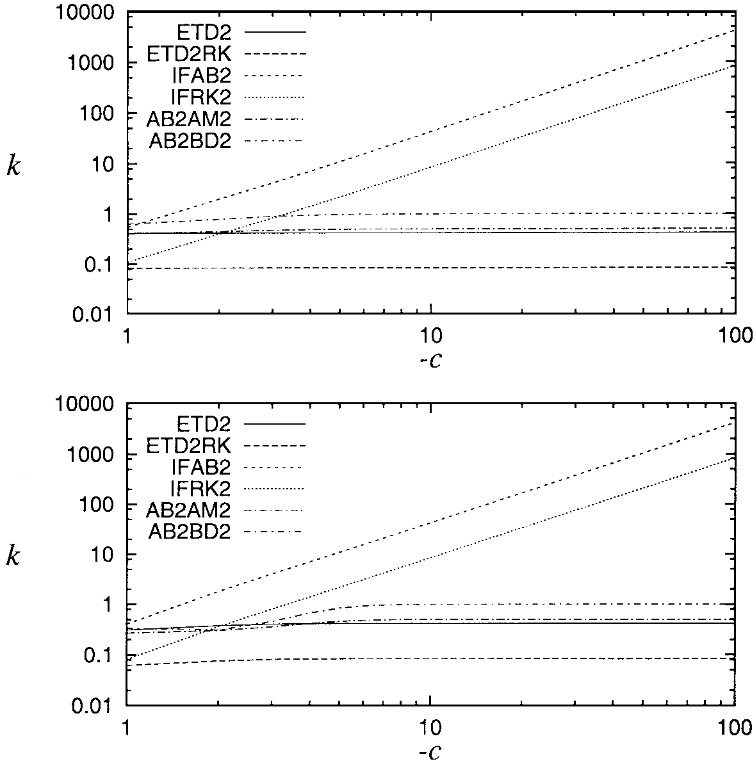

FIG. 4. In the limit as $h \to 0$, the relative error at $t = \pi/2$ in the numerical solution to (44) is given by kh^2. The constant $|k|$ is plotted as a function of c for various numerical schemes. Initial conditions on ($u_0 = 0$) and off ($u_0 = 1$) the slow manifold give the first and second figures, respectively. When $-c \gg 1$, the value of k for the various numerical schemes is, in both cases: IFAB2 ($-5c^2/12$); IFRK2 ($c^2/12$); AB2BD2 (1); AB2AM2 (1/2); ETD2 (5/12); ETD2RK ($-1/12$).

methods are almost as accurate as the ETD methods. However, this is due to fortuitous damping of the errors that arise in the initial fast phase; if the results are compared at an earlier time, the ETD methods show considerably greater accuracy than the LI methods.

The best of the methods considered, for all values of h, is ETD2RK. The error is smaller than that of ETD2 by a factor of 5, which is consistent with the truncation errors for the two methods. However, it should be remembered that this method requires twice as much CPU time, so when this is taken into account the accuracy is improved only by a factor of $5/4$.

The advantages of ETD methods become greater if we consider methods of fourth order. Because of the second Dahlquist stability barrier, LI methods are not A-stable if their order is greater than two (see, for example, [5] or [10]). The stability region for the fourth-order Adams–Moulton method AM4 does not include the entire negative real axis, so this method is not suitable for (44) except for very small h. For the fourth-order backward difference method BD4, the stability region does include the negative real axis, so a fourth-order linearly implicit method AB4BD4 can be constructed for (1);

$$\textbf{AB4BD4} \quad u_{n+1} = (48u_n - 36u_{n-1} + 16u_{n-2} - 3u_{n-3} + 48hF_n \\ -72hF_{n-1} + 48hF_{n-2} - 12hF_{n-3})/(25 - 12hc). \qquad (55)$$

A comparison of four fourth-order methods is shown in Fig. 5. The methods considered are ETD4, obtained from (19), ETD4RK (29), the integrating factor method IFAB4 using the standard fourth-order Adams–Bashforth formula, and AB4BD4. The integrating factor method has errors larger than the other two methods by a factor of c^4. The errors of AB4BD4 are about twice those of ETD4. A further advantage of ETD4 over AB4BD4 is that it requires

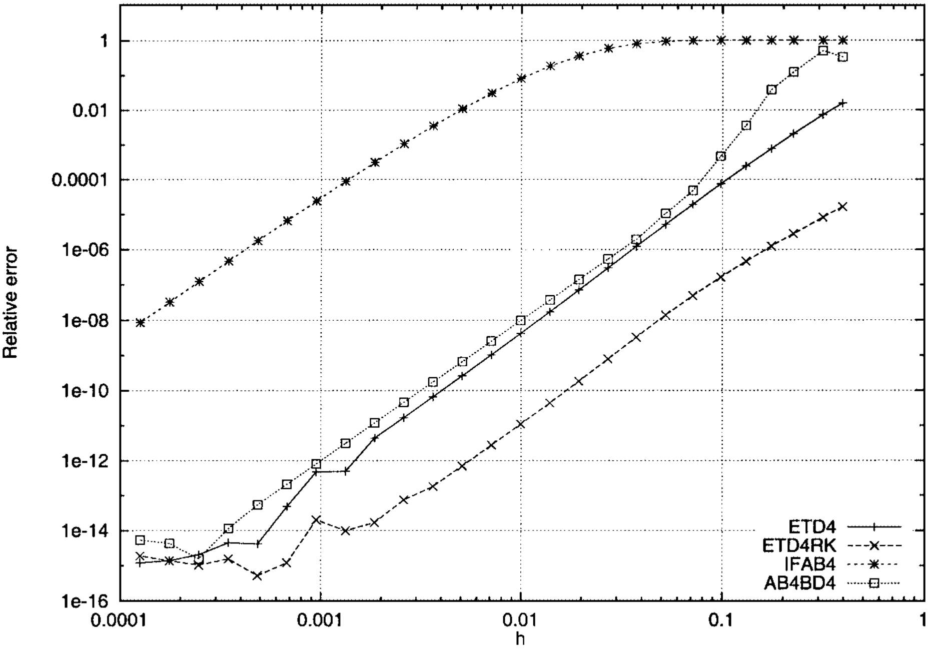

FIG. 5. Magnitude of the relative error at $t = \pi/2$ in (44) with $u_0 = 1$ for four fourth-order methods. Accuracy is ultimately limited by machine precision (here double-precision arithmetic is used).

half the storage (a significant factor for PDE applications) since it uses only previous values of F, whereas AB4BD4 requires previous values of both F and u. But the most accurate method is ETD4RK, with an error smaller than that of ETD4 by a factor of almost 400.

4.2. A Model Ordinary Differential Equation with Rapid Oscillation

Stiffness may also be the result of rapid oscillations generated by the linear terms. As an illustrative example we consider the initial-value problem

$$\dot{u} = icu + e^{it}, \quad u(0) = u_0, \tag{56}$$

which has the exact solution

$$u(t) = u_0 e^{ict} + \frac{e^{it} - e^{ict}}{i(1-c)}, \tag{57}$$

where our interest is in the stiff case for which the real parameter c satisfies $c \gg 1$. Our analysis of the various numerical schemes follows that given above in Section 4.1.

By solving exactly the difference equations that correspond to the various numerical schemes identified above, we are able to calculate the absolute and relative errors in advancing the solution to time $t = T$ using the time step h. For IFAB2, IFRK2, ETD2, and ETD2RK, the results for large c are shown in Table III, for initial conditions on and off the slow manifold (this manifold corresponds to $u_0 = -i/(1-c)$, so that the rapid oscillations in (57) are removed). The absolute error is then $k_1 h^2(e^{iT} - e^{iTc})$, where k_1 is given in the table. The relative error for initial conditions on the slow manifold is $k_2 h^2(e^{iT} - e^{iTc})e^{-iT}$ and off the slow manifold is $k_3 h^2(e^{iT} - e^{iTc})e^{-iTc}$; again k_2 and k_3 are given in Table III. The corresponding large-c results for AB2AM2 and AB2BD2 are shown in Table IV. The absolute error and relative error are denoted by $\epsilon_a h^2$ and $\epsilon_r h^2$, respectively, with superscripts "on" or "off" indicating whether the initial condition lies on or off the slow manifold.

For initial conditions on the slow manifold, the results are very similar to those for (44): The ETD methods are slightly more accurate than the LI methods, which are more accurate than the IF methods by a factor of c^2. However, for initial conditions off the slow manifold, the ETD methods are more accurate by a factor of c^2 than IF methods, which in turn are more accurate than LI methods by a factor c^2 (this strongly situation-dependent performance of LI and IF methods is discussed by Boyd [3], p. 269).

TABLE III
Errors in the Numerical Solution to (56) for Large *c*

	IFAB2	IFRK2	ETD2	ETD2RK
k_1	$\frac{5}{12}ic$	$-\frac{1}{12}ic$	$5i/(12c)$	$-i/(12c)$
k_2	$\frac{5}{12}c^2$	$-\frac{1}{12}c^2$	$\frac{5}{12}$	$-\frac{1}{12}$
k_3	$5ic/(12u_0)$	$-ic/(12u_0)$	$5i/(12u_0c)$	$-i/(12u_0c)$

Note. The absolute error is proportional to k_1. The relative errors on and off the slow manifold are proportional to k_2 and k_3 respectively.

TABLE IV
Errors in the Numerical Solution to (56) for Large *c*: Schemes AB2AM2 and AB2BD2

	AB2AM2	AB2BD2
ϵ_a^{on}	$\frac{1}{2}i(e^{iT}-e^{icT})/c$	$i(e^{iT}-e^{icT})/c$
ϵ_a^{off}	$-\frac{1}{12}iTe^{icT}u_0c^3$	$-\frac{1}{3}iTe^{icT}u_0c^3$
ϵ_r^{on}	$\frac{1}{2}(1-e^{i(c-1)T})$	$1-e^{i(c-1)T}$
ϵ_r^{off}	$-\frac{1}{12}iTc^3$	$-\frac{1}{3}iTc^3$

The relative errors for (56) with $c=100$ and the initial condition $u_0=1$ (off the slow manifold) are shown in Fig. 6. Note that the ETD methods are the most accurate, by a factor of order $c^2=10^4$, for a very wide range of h.

4.3. A Dissipative Partial Differential Equation

In this section we apply exponential time differencing methods to the (dissipative) PDE

$$\frac{\partial u}{\partial t} = -2\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} - u\frac{\partial u}{\partial x}, \tag{58}$$

which is the well-known Kuramoto–Sivashinsky equation [12]. The boundary conditions are periodic, with spatial period 2π; the initial condition chosen is $u(x,0)=0.03\sin x$. The

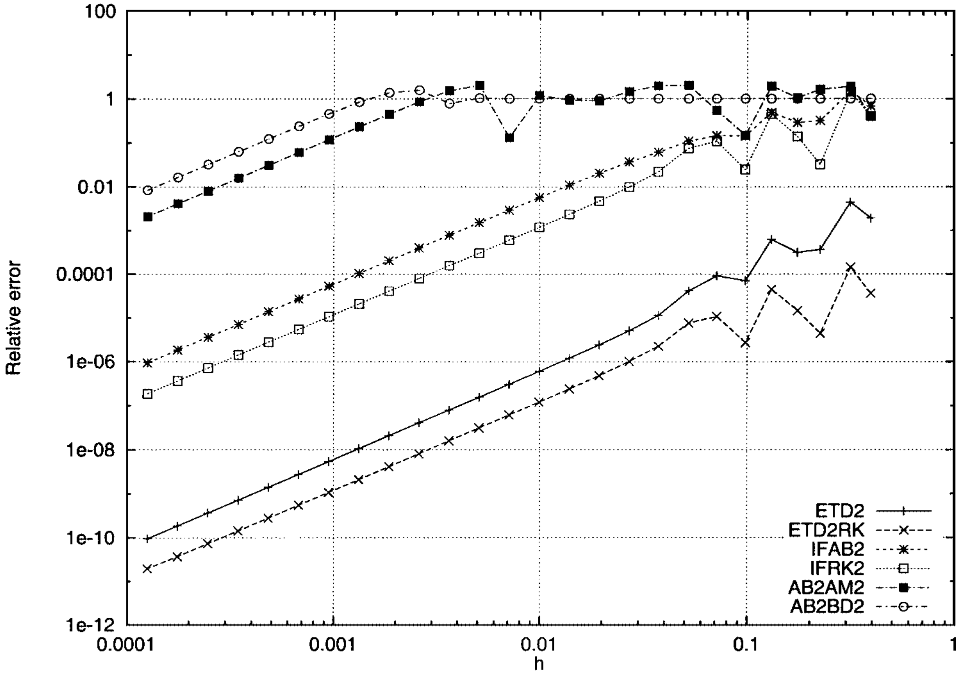

FIG. 6. Magnitude of the relative error at $t=\pi/2$ in (56) with $u_0=1$ for six methods.

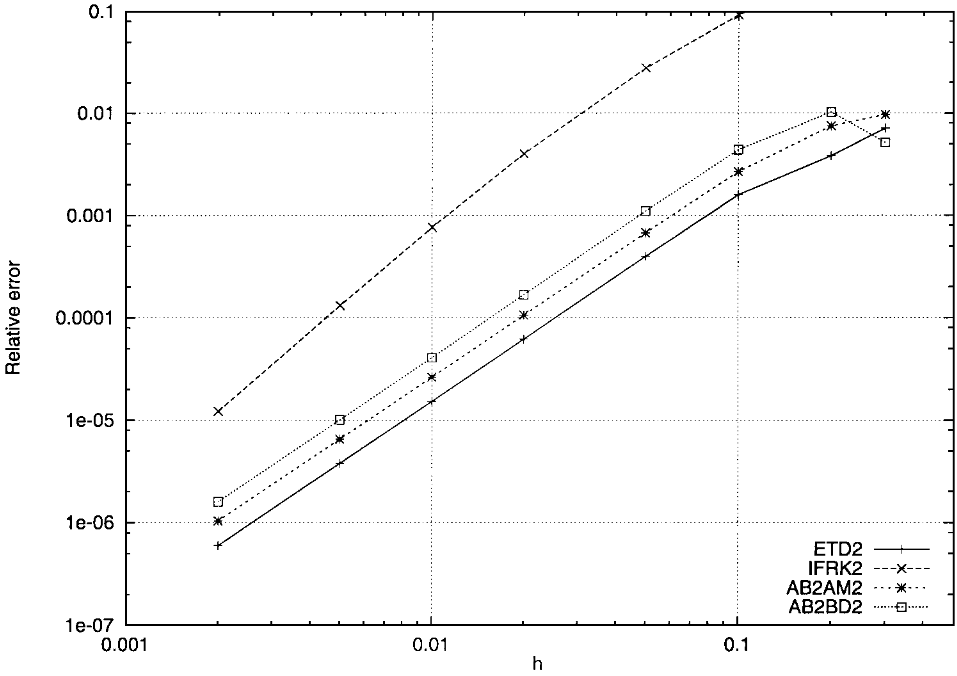

FIG. 7. Magnitude of the relative error at $t = 6$ for the Kuramoto–Sivashinsky equation (58) for four methods.

fourth-derivative term makes the linear part of (58) extremely stiff, with rapid linear decay of the high-wavenumber modes, so that standard explicit methods are impractical. Indeed, the motivation for our interest in ETD methods came from PDEs similar to (58) but with six derivatives in the linear term [14, 15].

The PDE (58) was solved with a pseudospectral method using 32 grid points without dealiasing, and using double precision arithmetic for $0 \leq t \leq 6$. The time stepping was carried out in spectral space, so that the linear parts of the evolution equations for each Fourier mode are uncoupled and the ETD, IF, and LI methods can be straightforwardly applied. The relative error in $\int u^2 \, \mathrm{d}x$ at $t = 6$ is plotted in Fig. 7 for four second-order methods. The error was measured by comparison with the "true" solution determined numerically using a fourth-order method with a very small time step.

In applying the ETD methods to (58), we have used the limiting form of the coefficients for the mode with zero wavenumber, in order to avoid division by zero. However, we did not find it necessary to replace the explicit formulae for the coefficients by their Taylor series for the modes with $|ch| \ll 1$.

The results in Fig. 7 are qualitatively similar to those for (44), showing that (44) is a good model problem for dissipative PDEs. The most accurate method is ETD2, with errors lower than those of AB2AM2 by a factor of 1.7 for a wide range of h. The least accurate method is the standard integrating factor method IFRK2.

The time-dependence of the relative error is shown in Fig. 8, for the same four methods, with a fixed time step of $h = 0.01$. In the initial phase of exponential growth, both of the LI methods perform poorly, compared with the IFRK2 and ETD2 methods. In the later stage of nonlinear equilibration, IFRK2 is the least accurate, while the other three methods have exponentially decaying errors. At all times the ETD2 method is the most accurate.

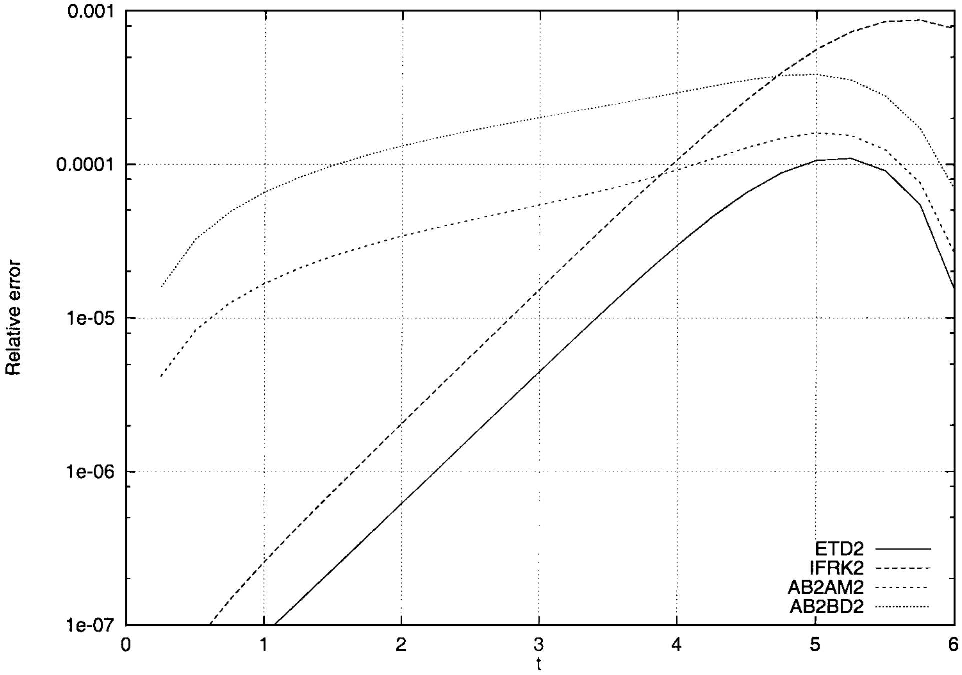

FIG. 8. Magnitude of the relative error as a function of t for the Kuramoto–Sivashinsky equation (58), with $h = 0.01$.

4.4. A Dispersive Partial Differential Equation

We now turn to the KdV equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0, \tag{59}$$

which is becoming a standard test for spectral solvers [6, 16, 21]. The stiffness results from the term u_{xxx} and manifests itself in rapid linear oscillation of the high-wavenumber modes.

The computations are spatially 2π-periodic and follow a soliton solution, $u = f(x - ct)$, where $f(x) = 3c\,\text{sech}^2(c^{1/2}x/2)$ for one period, i.e., up to $t = 2\pi/c$, with $c = 625$. The method uses 256 grid points and is de-aliased using the usual 2/3 rule, since in (59), unlike (58), there are no dissipation terms to remove the aliasing errors. The relative error plotted in Fig. 9 is $(\Sigma(u_j - f_j)^2/\Sigma f_j^2)^{1/2}$, i.e., the scaled 2-norm of the error. The results for the second-order methods are qualitatively similar to those for the previous example, with the ETD2 method giving the most accurate results. To avoid the rounding errors discussed in Section 4.3, the coefficients in (6) were computed using the four-term Taylor series for those Fourier modes with $|ch| < 10^{-4}$.

If higher-order methods are required for a dispersive problem such as (59), the backward difference methods and Adams–Moulton methods cannot be used because their stability regions do not include the entire imaginary axis. This problem can be overcome by more elaborate methods such as using AM2 for the linear terms of high wavenumber, and AB4 for the nonlinear terms and the linear terms of low wavenumber [6]. If fourth-order methods are to be used for all modes, the only possibilities are IF or ETD methods. In view of the initialization problem referred to in Section 2.2, RK methods are most suitable for this problem, so results for ETD4RK (29) and IFRK4 (the integrating factor method combined

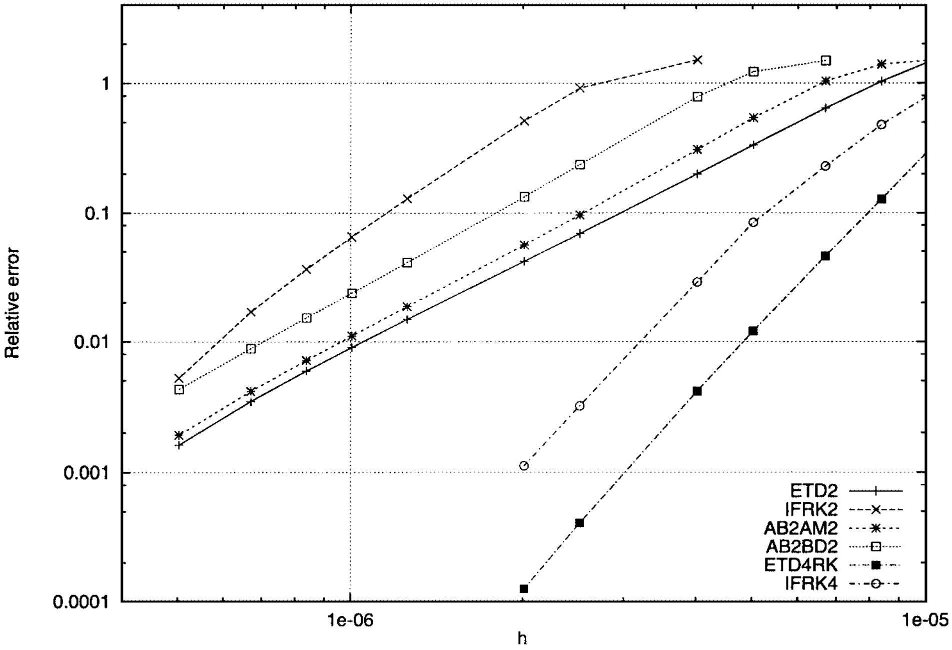

FIG. 9. Error after one soliton period for the KdV equation (59) for six methods.

with the standard fourth-order RK scheme) are shown in Fig. 9. The ETD method shows a tenfold improvement over the IF method.

4.5. Nondiagonal Systems

Exponential time differencing schemes have hitherto been applied only to systems whose linear parts are diagonal (such as arise from solving partial differential equations using a Fourier spectral method). In this case, the expressions for time stepping the amplitude of each mode can be determined independently. However, ETD methods can also readily be generalized to nondiagonal systems (such as arise from solving PDEs using finite differences or Chebyshev methods, for instance). The details of the generalization depend crucially on whether the linear operator has any zero eigenvalues, as demonstrated below in Sections 4.5.1 and 4.5.3.

Since ETD methods require the calculation of the matrix exponential e^{Lh}, and since the calculation of this matrix is nontrivial when L is nondiagonal, the question arises as to whether ETD methods are suitable in the case of nondiagonal L. However, the matrix exponential need be calculated only once, at the start of the integration, and so the computational overhead involved is not large.

We begin our discussion of nondiagonal systems with a model system to illustrate the concepts; in particular we discuss the additional difficulties associated with zero eigenvalues. We then apply ETD to a nondiagonal problem arising in fluid mechanics.

4.5.1. *Nondiagonal System with No Zero Eigenvalues*

To illustrate how ETD can be generalized to a system whose linear part is nondiagonal with no zero eigenvalues, we consider the following autonomous system of two ordinary differential equations

$$\dot{u} = -v(1-\lambda r^2) + cu(1-r^2), \tag{60}$$

$$\dot{v} = u(1-\lambda r^2) + cv(1-r^2), \tag{61}$$

where $r^2 = u^2 + v^2$ and $c > 0$. The behavior of this system is more readily observed by using the amplitude r and phase θ, such that $u = r\cos\theta$ and $v = r\sin\theta$. Then

$$\dot{r} = cr(1-r^2), \tag{62}$$

$$\dot{\theta} = 1 - \lambda r^2, \tag{63}$$

so that (unless r is initially zero) $r \to 1$ and $\dot{\theta} \to 1-\lambda$ at large time. The exact solution to (62) and (63), i.e.,

$$r^2(t) = \frac{r_0^2}{r_0^2(1-e^{-2ct}) + e^{-2ct}}, \tag{64}$$

$$\theta(t) = \theta_0 + (1-\lambda)t - \frac{\lambda}{2c}\log\bigl(r_0^2(1-e^{-2ct}) + e^{-2ct}\bigr), \tag{65}$$

where $r(0) = r_0$ and $\theta(0) = \theta_0$, is useful in calculating the amplitude and phase errors in the numerical solutions.

To implement the nondiagonal ETD methods, we first write (60) and (61) in the vector form

$$\dot{\boldsymbol{u}} = L\boldsymbol{u} + \boldsymbol{F}, \tag{66}$$

where

$$\boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix}, \qquad L = \begin{pmatrix} c & -1 \\ 1 & c \end{pmatrix}, \qquad \boldsymbol{F} = \begin{pmatrix} (\lambda v - cu)r^2 \\ -(\lambda u + cv)r^2 \end{pmatrix}. \tag{67}$$

Note that the matrix L is nondiagonal. Although in this case we could diagonalize L by a suitable change of variables, in applications this may not be a desirable procedure and it may be more convenient to solve the problem in the original variables, for which the linear operator is nondiagonal. By analogy with (3), we then multiply (66) by e^{-Lt} and integrate from $t = t_n$ to t_{n+1} to give the exact result

$$\boldsymbol{u}_{n+1} = e^{Lh}\boldsymbol{u}_n + e^{Lh}\int_0^h e^{-L\tau}\boldsymbol{F}(t_n+\tau)\,\mathrm{d}\tau. \tag{68}$$

Then for ETD1 we approximate the final term in (68) as

$$e^{Lh}\int_0^h e^{-L\tau}\boldsymbol{F}(t_n+\tau)\,\mathrm{d}\tau = e^{Lh}\int_0^h e^{-L\tau}\boldsymbol{F}_n\,\mathrm{d}\tau + O(h^2) = M_1\boldsymbol{F}_n + O(h^2), \tag{69}$$

where

$$M_1 = L^{-1}(e^{Lh} - I), \tag{70}$$

and where I is the 2×2 identity matrix. For ETD2 we adopt the approximation

$$\int_0^h e^{-L\tau} \boldsymbol{F}(t_n + \tau)\, \mathrm{d}\tau = \int_0^h e^{-L\tau} (\boldsymbol{F}_n + (\boldsymbol{F}_n - \boldsymbol{F}_{n-1})\tau/h)\, \mathrm{d}\tau + O(h^3) \tag{71}$$

and note that

$$e^{Lh} \int_0^h \tau e^{-L\tau}\, \mathrm{d}\tau = M_2 \equiv L^{-2}[e^{Lh} - (I + Lh)]. \tag{72}$$

Thus,

$$e^{Lh} \int_0^h e^{-L\tau} \boldsymbol{F}(t_n + \tau)\, \mathrm{d}\tau = M_1 \boldsymbol{F}_n + M_2(\boldsymbol{F}_n - \boldsymbol{F}_{n-1})/h + O(h^3). \tag{73}$$

In this particular example, all the necessary matrices can be computed exactly. The useful matrices are

$$e^{Lh} = \begin{pmatrix} e^{ch}\cos h & -e^{ch}\sin h \\ e^{ch}\sin h & e^{ch}\cos h \end{pmatrix}, \tag{74}$$

$$M_1 = \frac{1}{1+c^2}\begin{pmatrix} -c + e^{ch}(c\cos h + \sin h) & -1 + e^{ch}(\cos h - c\sin h) \\ 1 - e^{ch}(\cos h - c\sin h) & -c + e^{ch}(c\cos h + \sin h) \end{pmatrix}, \tag{75}$$

$$M_2 = \begin{pmatrix} \mu_d & \mu_o \\ -\mu_o & \mu_d \end{pmatrix}, \tag{76}$$

where

$$\mu_d = \frac{1 - c^2 - hc - hc^3 + (2c\sin h - (1-c^2)\cos h)e^{hc}}{(c^2+1)^2}, \tag{77}$$

$$\mu_o = \frac{-2c - h - hc^2 + (2c\cos h + (1-c^2)\sin h)e^{hc}}{(c^2+1)^2}. \tag{78}$$

The scheme ETD1 is thus

$$\boldsymbol{u}_{n+1} = e^{Lh}\boldsymbol{u}_n + M_1\boldsymbol{F}_n \tag{79}$$

and ETD2 is

$$\boldsymbol{u}_{n+1} = e^{Lh}\boldsymbol{u}_n + M_1\boldsymbol{F}_n + M_2(\boldsymbol{F}_n - \boldsymbol{F}_{n-1})/h. \tag{80}$$

The scheme ETD2 may be extended to give a Runge–Kutta scheme ETD2RK by first calculating

$$\boldsymbol{a}_n = e^{Lh}\boldsymbol{u}_n + M_1\boldsymbol{F}_n \tag{81}$$

and then taking the time step using

$$\boldsymbol{u}_{n+1} = \boldsymbol{a}_n + M_2(\boldsymbol{F}(\boldsymbol{a}_n, t_n + h) - \boldsymbol{F}_n)/h. \tag{82}$$

In order to evaluate the schemes ETD1, ETD2, and ETD2RK, we also introduce the more successful competitors from above: AB2AM2 and IFRK2.

The AB2AM2 scheme is derived from the formula

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = \frac{1}{2}hL(\boldsymbol{u}_{n+1} + \boldsymbol{u}_n) + \frac{3}{2}h\boldsymbol{F}_n - \frac{1}{2}h\boldsymbol{F}_{n-1}, \tag{83}$$

from which it follows that

$$\boldsymbol{u}_{n+1} = \left(I - \frac{1}{2}hL\right)^{-1}\left(I + \frac{1}{2}hL\right)\boldsymbol{u}_n + \frac{1}{2}h\left(I - \frac{1}{2}hL\right)^{-1}(3\boldsymbol{F}_n - \boldsymbol{F}_{n-1}). \tag{84}$$

For the standard Runge–Kutta integrating factor method IFRK2 we first calculate

$$\boldsymbol{b} = e^{Lh}(\boldsymbol{u}_n + h\boldsymbol{F}_n) \tag{85}$$

and then take the time step with

$$\boldsymbol{u}_{n+1} = e^{Lh}\left(\boldsymbol{u}_n + \frac{1}{2}h\boldsymbol{F}_n\right) + \frac{1}{2}h\boldsymbol{F}(\boldsymbol{b}, t_n + h). \tag{86}$$

4.5.2. *Evaluation of Numerical Schemes*

In this section we describe numerical solutions to (66) using ETD1, ETD2, ETD2RK, IFRK2, and AB2AM2. For the computations we take the initial condition

$$u(0) = 2, \quad v(0) = 1 \tag{87}$$

and the parameter values $c = 100$, $\lambda = \frac{1}{2}$. We evaluate the various schemes by comparing their predicted amplitude and phase with the exact values of these quantities at time $t = 1$, although other parameter values and end-times give similar results. The results are summarized in Fig. 10, where it is seen that ETD2, ETD2RK, and AB2AM2 are the best schemes. In particular, the integrating factor scheme IFRK2 is particularly poor in its calculation of the amplitude (being out-performed by even the first-order scheme ETD1 in the range of time steps considered).

While the best at capturing the amplitude of the evolving solution, ETD2RK is rather poorer at calculating the phase. An error analysis reveals that there is no simple factor-of-5 difference between the schemes ETD2 and ETD2RK in this system.

4.5.3. *Nondiagonal System with Zero Eigenvalues*

When a nondiagonal linearized system has one or more zero eigenvalues, the methods ETD1, ETD2, and ETD2RK cannot be used as presented above, because the matrix L has no inverse. However, they may be readily generalized as follows. When L has one or more zero eigenvalues, we define a pseudo-inverse of L to be $L^\dagger = V^{-1}\Lambda^\dagger(U^T)^{-1}$, where $L = U^T\Lambda V$ is the singular value decomposition of L. Here $\Lambda^\dagger$ is the diagonal matrix obtained from Λ by taking the reciprocal of all the nonzero diagonal elements (and leaving all the zero diagonal elements as they stand).

The expression $e^{Lh}\int_0^h e^{-L\tau}\,\mathrm{d}\tau$ is then given by

$$M_1 \equiv L^\dagger(e^{Lh} - I) + he^{Lh}(I - L^\dagger L), \tag{88}$$

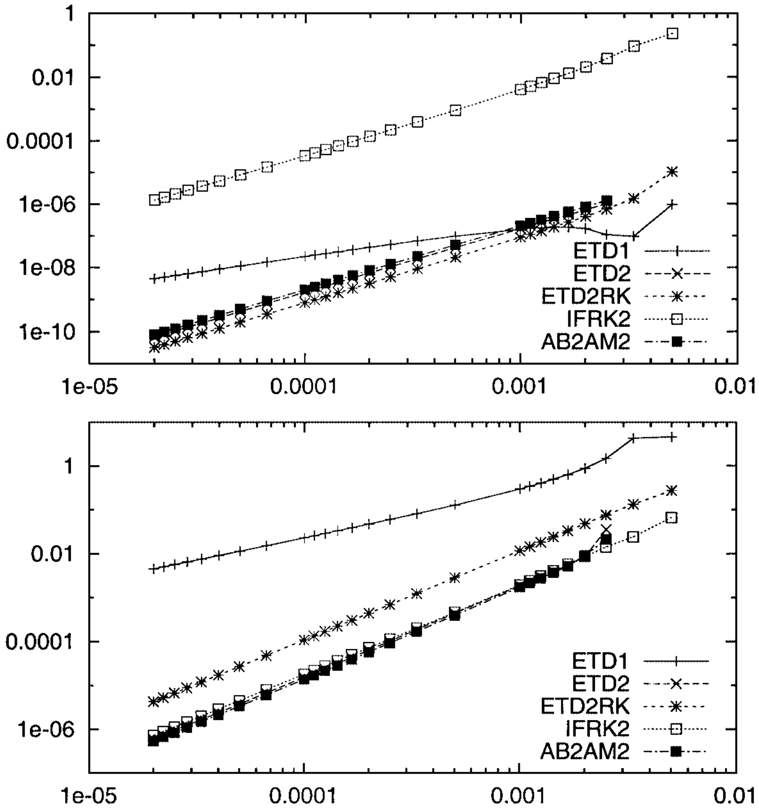

FIG. 10. The magnitude of the relative error at $t = 1$ in the numerical solution to (66), with $u_0 = 2$ and $v_0 = 1$, and with $c = 100$ and $\lambda = \frac{1}{2}$. The first figure shows the amplitude error (i.e., the error in r), and the second the corresponding phase error (in θ).

while $e^{Lh} \int_0^h \tau e^{-L\tau}\, \mathrm{d}\tau$ is given by

$$M_2 \equiv L^{\dagger 2}[e^{Lh} - (I + Lh)] + \frac{1}{2}h^2 e^{Lh}(I - L^{\dagger}L). \tag{89}$$

The numerical schemes ETD1, ETD2, and ETD2RK are then given by (79), (80), and (82), with M_1 and M_2 as in (88) and (89), respectively.

4.5.4. *Application to a Partial Differential Equation*

A good example of a physical partial differential equation that gives rise to a nondiagonal matrix L is the Berman equation

$$f_{yyt} = R^{-1} f_{yyyy} + f_y f_{yy} - f f_{yyy}, \tag{90}$$

subject to the boundary conditions

$$f(-1, t) = f_y(-1, t) = f_y(1, t) = 0, \qquad f(1, t) = 1 \quad (t \geq 0) \tag{91}$$

and the initial condition $f(y, 0) = f_0(y)$. This initial boundary-value problem arises in calculating the flow of a viscous fluid in the channel $-1 \leq y \leq 1$, driven by uniform withdrawal of fluid through the porous walls of the channel [11]; the Reynolds number R is a dimensionless measure of the withdrawal speed. The solution to (90) and (91), after transients have decayed, can exhibit self-sustained oscillations with intricate spatial and temporal behavior and a rich analytical structure. Our numerical scheme to solve (90) and (91) proceeds by first rendering the boundary conditions homogeneous by writing $f(y,t) = p(y) + g(y,t)$, where $p(y) = -(y-2)(y+1)^2/4$ satisfies $p(-1) = p'(-1) = p'(1) = p(1) - 1 = 0$ and $p''''(y) = 0$. We then solve the resulting forced PDE for g by Chebyshev collocation (although clearly a variety of other methods could be used). With this method, the vector $\boldsymbol{g} = (g_1, \ldots, g_{N-1})^t$ of values of g at the interior collocation points satisfies an evolution equation of the form

$$\dot{\boldsymbol{g}} = L\boldsymbol{g} + \boldsymbol{n}, \tag{92}$$

where L is a nondiagonal matrix, readily found in terms of Chebyshev differentiation matrices, and $\boldsymbol{n}$ represents terms generated by $(p' + g_y)(p'' + g_{yy}) - (p + g)(p''' + g_{yyy})$. The matrix L is full, which makes computation of e^{Lh}, required for the implementation of ETD, time consuming. However, e^{Lh} need be computed only *once*, prior to the time-stepping loop, and in our numerical simulations of (90) we have found the computational expense of calculating e^{Lh} to be trivial compared with that of generating the time evolution of the solution.

5. CONCLUSIONS

We have developed and tested a class of numerical methods for systems with stiff linear parts, based on combining exponential time differencing for the linear terms with a method similar to Adams–Bashforth for the nonlinear terms. The ETD method is straightforward to apply and can be extended to arbitrary order (cf. [1]). As the stiffness parameter tends to zero, the ETD method approaches the Adams–Bashforth method of the same order; as the stiffness parameter tends to infinity, the nonlinear Galerkin method [13] is recovered.

In addition to these multistep ETD methods, we have derived new Runge–Kutta forms of the ETD method, of second, third, and fourth order. These are easier to use than the high-order multistep forms, since they do not require initialization, and are more accurate.

These ETD methods have good stability properties and are widely applicable to dissipative PDEs and nonlinear wave equations. They are particularly well suited to Fourier spectral methods, which have diagonal linear part.

We have carried out extensive tests of ETD methods, comparing them with linearly implicit and integrating factor methods. For all the examples tested, the ETD methods are more accurate than either LI or IF methods. For solutions which follow a slow manifold, the second-order ETD method is slightly more accurate than the LI methods, and has the advantage that it readily generalizes to higher order, whereas LI methods do not. But for solutions off the slow manifold, the results of Section 4.2 show that second-order ETD methods are more accurate than LI methods by the fourth power of the stiffness parameter $c \gg 1$, and more accurate than integrating factor methods by a factor c^2. Like integrating factor methods, the ETD methods solve the linear parts exactly. However, ETD methods avoid the major drawback of IF methods, which is the introduction of the fast time scale into the nonlinear terms, leading to large error constants.

REFERENCES

1. G. Beylkin, J. M. Keiser, and L. Vozovoi, A new class of time discretization schemes for the solution of nonlinear PDEs, *J. Comput. Phys*. **147**, 362 (1998).
2. J. P. Boyd, Eight definitions of the slow manifold: Seiches, pseudoseiches and exponential smallness, *Dyn. Atmos. Oceans* **22**, 49 (1995).
3. J. P. Boyd, *Chebyshev and Fourier Spectral Methods* (Dover, New York, 2001).
4. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer Series in Computational Physics (Springer-Verlag, Berlin, 1988).
5. B. Fornberg, *A Practical Guide to Pseudospectral Methods* (Cambridge Univ. Press, Cambridge, UK, 1995).
6. B. Fornberg and T. A. Driscoll, A fast spectral algorithm for nonlinear wave equations with linear dispersion, *J. Comput. Phys*. **155**, 456 (1999).
7. B. García-Archilla, Some practical experience with the time integration of dissipative equations, *J. Comput. Phys*. **122**, 25 (1995).
8. P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations* (Wiley, New York, 1962).
9. R. Holland, Finite-difference time-domain (FDTD) analysis of magnetic diffusion, *IEEE Trans. Electromagn. Compat*. **36**, 32 (1994).
10. A. Iserles, *A First Course in the Numerical Analysis of Differential Equations* (Cambridge Univ. Press, Cambridge, UK, 1996).
11. J. R. King and S. M. Cox, Asymptotic analysis of the steady-state and time-dependent Berman problem, *J. Eng. Math*. **39**, 87 (2001).
12. Y. Kuramoto and T. Tsuzuki, Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, *Prog. Theor. Phys*. **55**, 356 (1976).
13. M. Marion and R. Temam, Nonlinear Galerkin methods, *SIAM J. Numer. Anal*. **26**, 1139 (1989).
14. P. C. Matthews and S. M. Cox, Pattern formation with a conservation law, *Nonlinearity* **13**, 1293 (2000).
15. P. C. Matthews and S. M. Cox, One-dimensional pattern formation with Galilean invariance near a stationary bifurcation, *Phys. Rev. E* **62**, R1473 (2000).
16. P. A. Milewski and E. Tabak, A pseudospectral procedure for the solution of nonlinear wave equations with examples from free-surface flows, *SIAM J. Sci. Comp*. **21**, 1102 (1999).
17. P. G. Petropoulos, Analysis of exponential time-differencing for FDTD in lossy dielectrics, *IEEE Trans. Antennas Propagation* **45**, 1054 (1997).
18. C. Schuster, A. Christ, and W. Fichtner, Review of FDTD time-stepping for efficient simulation of electric conductive media. *Microwave Optical Technol. Lett*. **25**, 16 (2000).
19. A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, Artech House Antenna Library (Artech House, London, 1995).
20. L. N. Trefethen, Lax-stability vs. eigenvalue stability of spectral methods, in *Numerical Methods for Fluid Dynamics III*, edited by K. W. Morton and M. J. Baines (Clarendon Press, Oxford, 1988), pp. 237–253.
21. L. N. Trefethen, *Spectral Methods in Matlab* (Soc. for Industr. & Appl. Math., Philadelphia, 2000).
22. J. Z. Zhu, L.-Q. Chen, J. Shen, and V. Tikare, Coarsening kinetics from a variable-mobility Cahn–Hilliard equation: Application of a semi-implicit Fourier spectral method, *Phys. Rev. E* **60**, 3564 (1999).