Chapter 11
Mixed Equations

We have now studied the solution of various types of time-dependent equations: ordinary differential equations (ODEs), parabolic partial differential equations (PDEs) such as the heat equation, and hyperbolic PDEs such as the advection equation. In practice several processes may be happening simultaneously, and the PDE model will not be a pure equation of any of the types already discussed but rather will be a mixture. In this chapter we discuss several approaches to handling more complicated equations. We restrict our attention to time-dependent PDEs of the form

$$u_t = \mathcal{A}_1(u) + \mathcal{A}_2(u) + \cdots + \mathcal{A}_N(u), \tag{11.1}$$

where each of the $\mathcal{A}_j(u)$ are (possibly nonlinear) functions or differential operators involving only spatial derivatives of u. For simplicity, most of our discussion will be further restricted to only two terms, which we will write as

$$u_t = \mathcal{A}(u) + \mathcal{B}(u), \tag{11.2}$$

but more terms often can be handled by extension or combination of the methods described here.

11.1 Some examples

We begin with some examples of PDEs involving more than one term. See Appendix E for more discussion of some of these equations.

- *Multidimensional problems*, such as the diffusion equation in two dimensions,

$$u_t = \kappa(u_{xx} + u_{yy}), \tag{11.3}$$

  or the three-dimensional version. This problem has already been discussed in Section 9.7, where we saw that efficient methods can be developed by splitting (11.3) into two one-dimensional problems. Hyperbolic equations also arise in multidimensional domains, such as the two-dimensional hyperbolic system

$$u_t + Au_x + Bu_y = 0 \tag{11.4}$$

or nonlinear hyperbolic conservation laws

$$u_t + f(u)_x + g(u)_y = 0, \tag{11.5}$$

where $f(u)$ and $g(u)$ are the flux functions in the two directions.

All the problems discussed below also have multidimensional variants, where even more terms arise. For simplicity we display only the one-dimensional case.

- *Reaction-diffusion equations* of the form

$$u_t = \kappa u_{xx} + R(u), \tag{11.6}$$

where κ is a diffusion coefficient (or diagonal matrix of diffusion coefficients if different components in the system diffuse at different rates) and $R(u)$ represents chemical reactions, and is typically nonlinear. The reaction terms might or might not be stiff. If not, then we typically want to handle these terms explicitly (to avoid solving nonlinear systems of equations in each time step), while the diffusion term is stiff and requires appropriate methods.

Even if the reaction terms are stiff, they apply locally at a point in space, unlike the diffusion term that couples different grid points together. Recognizing this fact can lead to more efficient solution techniques, as discussed further below.

- *Advection-diffusion equations* of the form

$$u_t + au_x = \kappa u_{xx}. \tag{11.7}$$

The diffusion term is stiff and requires an appropriate solver, while the advection term can be handled explicitly.

- *Nonlinear hyperbolic equations* with viscous terms,

$$u_t + f(u)_x = \kappa u_{xx}. \tag{11.8}$$

The advection-diffusion equation (11.7) is one example of this form, but more generally the flux function $f(u)$ can be nonlinear, modeling fluid dynamics, for example, in which case the right-hand side represents viscous terms and perhaps heat conduction. The *Navier–Stokes equations* for compressible gas dynamics have this general form, for example. A simpler example is the *viscous Burgers equation*,

$$u_t + uu_x = \epsilon u_{xx}, \tag{11.9}$$

where the flux function is $f(u) = \frac{1}{2}u^2$. This is a simple scalar model for some of the effects seen in compressible flow, and it has been widely studied.

- *Advection-diffusion-reaction equations* or reacting flow problems,

$$u_t + f(u)_x = \kappa u_{xx} + R(u). \tag{11.10}$$

If chemical reactions are occurring in a fluid flow, then equations of this general form are obtained. Combustion problems are particularly challenging problems of

this type, where exothermic chemical reactions directly influence the fluid dynamics. Chemotaxis problems are also of this type, which arise in biology when substances move in response to concentration gradients of other substances, and often give rise to interesting pattern formation [73].

- The *Korteweg–de Vries (KdV)* equation,

$$u_t + uu_x = vu_{xxx}. \tag{11.11}$$

This is similar to the viscous Burgers equation (11.9), but the term on the right-hand side is dispersive rather than dissipative. This leads to very different behavior and is the simplest example of an equation having soliton solutions. It arises as a simple model of certain kinds of wave phenomena in fluid dynamics and elsewhere. The third derivative term is stiffer than a u_{xx} term and would typically require $k = O(h^3)$ for an explicit method. However, similar to the advective terms considered in Chapter 10, the eigenvalues of a discretization of u_{xxx} will typically lie on or near the imaginary axis over an interval stretching distance $O(1/h^3)$ from the origin, rather than along the negative real axis, influencing the type of time discretization appropriate for these equations.

Many other equations that couple nonlinearity with dispersion are of importance in applications, for example, the *nonlinear Schrödinger equation*

$$i\psi_t(x,t) = -\psi_{xx}(x,t) + V(\psi) \tag{11.12}$$

with a nonlinear potential $V(\psi)$. (With $V = 0$ the equation is linear and dispersive, as shown in Section E.3.8.)

- The *Kuramoto–Sivashinsky* equation,

$$u_t + \frac{1}{2}(u_x)^2 = -u_{xx} - u_{xxxx}. \tag{11.13}$$

The right-hand side gives exponential growth of some low wave numbers, as shown in Section E.3.7. The nonlinear term transfers energy from low wave numbers to higher wave numbers, which are damped by the fourth order diffusion. The result is bounded solutions, but ones that can behave quite chaotically. The fourth order diffusion term is even more stiff than second order diffusion. Eigenvalues of a discretization of this term typically lie on the negative real axis over an interval of length $O(1/h^4)$.

Many approaches can be used for problems that involve two or more different terms, and a huge number of specialized methods have been developed for particular equations. The remainder of this chapter contains a brief overview of a few popular approaches, but it is by no means exhaustive.

11.2 Fully coupled method of lines

One simple approach is to discretize the full right-hand side of (11.1) in space using appropriate spatial discretizations of each term to obtain a semidiscrete method of lines (MOL)

system of the form $U'(t) = F(U(t))$, where F represents the full spatial discretization. This system of ODEs can now be solved using an ODE method in MATLAB or with other ODE software. This may work well for equations where the terms all have similar character. The problem in general, however, is that the same ODE method is being applied to all aspects of the spatial discretization, which can be very wasteful for many of the problems listed above.

Consider a reaction-diffusion equation of the form (11.6), for example. If this represents a system of s equations and we discretize in x on a grid with m points, then we obtain a coupled system of ms ODEs. Typically the reaction terms $R(u)$ are nonlinear and so this will be a nonlinear system. Generally an implicit method is used since the diffusion terms are stiff, so in every time step a nonlinear system of dimension ms must be solved. However, if the reaction terms are not stiff, then there is no need to make these terms implicit and it should be possible to solve only linear systems for the diffusion terms, and s decoupled linear systems of size m each (with tridiagonal matrices) rather than a fully coupled nonlinear system of size ms. Even if the reaction terms are stiff, the reaction terms $u_t = R(u)$ are local at a point and by splitting the reaction from the diffusion (using one of the other approaches discussed below), it is possible to solve decoupled nonlinear equations of dimension s at each grid point to advance the reaction terms in a stable manner.

11.3 Fully coupled Taylor series methods

A first order accurate explicit method for the equation (11.2) can be obtained by using the first order term in the Taylor series,

$$u(x, t_n + k) \approx u(x, t_n) + k(\mathcal{A}(u(x, t_n)) + \mathcal{B}(u(x, t_n))), \tag{11.14}$$

and then replacing the spatial operators $\mathcal{A}$ and $\mathcal{B}$ with discretizations. A second order accurate method can sometimes be obtained by adding the next term in the Taylor series, but this requires determining u_{tt} in terms of spatial derivatives of u. We did this for the advection equation $u_t + au_x = 0$ in Section 10.3 to derive the Lax–Wendroff method, in which case $u_{tt} = a^2 u_{xx}$. Whether we can do this in general for (11.2) depends on how complicated the right-hand side is, but in some cases it can be done. For example, for the two-dimensional hyperbolic equation (11.4) we can compute

$$\begin{aligned} u_{tt} &= -Au_{tx} - Bu_{ty} \\ &= A(Au_x + Bu_y)_x + B(Au_x + Bu_y)_y \\ &= A^2 u_{xx} + (AB + BA)u_{xy} + B^2 u_{yy}. \end{aligned} \tag{11.15}$$

A second order accurate Lax–Wendroff method can then be derived from

$$u(x, y, t_n + k) \approx u - k(Au_x + Bu_y) + \frac{1}{2}k^2(A^2 u_{xx} + (AB + BA)u_{xy} + B^2 u_{yy}) \tag{11.16}$$

(where the terms on the right-hand side are all evaluated at (x, y, t_n)) by discretizing in space using second order accurate centered approximations. This gives the two-dimensional Lax–Wendroff method.

For some other problems a similar approach works, e.g., for advection-reaction terms with nonstiff reactions, but this is generally useful only if all terms are nonstiff and can be

advanced with explicit methods. Moreover, it is generally difficult to achieve higher than second order accuracy with this approach.

11.4 Fractional step methods

The idea of a fractional step method (also called a time-split or split-step method, among other things) is to split up the equation into its constituent pieces and alternate between advancing simpler equations in time. The simplest splitting for an equation with two terms of the form (11.2) would be

$$\begin{aligned} U^* &= \mathcal{N}_A(U^n, k), \\ U^{n+1} &= \mathcal{N}_B(U^*, k). \end{aligned} \tag{11.17}$$

Here $\mathcal{N}_A(U^n, k)$ represents some one-step numerical method that solves $u_t = \mathcal{A}(u)$ over a time step of length k starting with data U^n. Similarly, $\mathcal{N}_B(U^*, k)$ solves $u_t = \mathcal{B}(u)$ over a time step of length k starting with the data U^*.

Below we will see that this splitting of the equation does often work—the numerical solution obtained will usually converge to solutions of the original problem as $k \to 0$ provided the numerical methods used in each step are consistent and stable approximations to the separate problems they are designed to solve. The approximation obtained often will be only first order accurate, however, no matter how good each of the constituent numerical methods is. We will see why below and consider some improvements.

First we note that this fractional step approach has several advantages. It allows us to use very different methods for each piece $u_t = \mathcal{A}(u)$ and $u_t = \mathcal{B}(u)$. One can be implicit and the other explicit, for example. For the reaction-diffusion problem (11.6), if $\mathcal{A}(u)$ represents the diffusion terms, then these can be solved with an implicit method, solving tridiagonal linear systems for each component. The reaction terms can be solved with an explicit or implicit method, depending on whether they are stiff. If an implicit method is used, then a nonlinear system is obtained at each grid point, but each is decoupled from the nonlinear system at other grid points, typically leading to a much more efficient solution of these systems.

Another situation in which this type of splitting is often used is in reducing a multidimensional problem to a sequence of one-dimensional problems. In this context the fractional step approach is often called *dimensional splitting*. We saw an example of this in Section 9.8, where the locally one-dimensional (LOD) method for the heat equation was discussed. By decoupling the space dimensions, one obtains a sequence of tridiagonal systems to solve instead of a large sparse matrix with more complicated structure.

Another advantage of the fractional step approach is that existing methods for the simpler subproblems are easily patched together, e.g., ODE methods for the reaction terms can be applied without worrying about the spatial coupling, or a one-dimensional method for a PDE can easily be extended to two or three dimensions by repeatedly applying it on one-dimensional slices.

To see that the fractional step method (11.17) may be only first order accurate, consider a simple linear system of ODEs where the coefficient matrix is split into two matrices as $A + B$, so the system is

$$u_t = Au + Bu. \tag{11.18}$$

Suppose we use the fractional step method (11.17) with the exact solution operator for each step, so

$$\mathcal{N}_A(U,k) = e^{Ak}U, \qquad \mathcal{N}_B(U,k) = e^{Bk}U. \tag{11.19}$$

Then (11.17) gives the numerical method

$$U^{n+1} = e^{Bk}U^* = e^{Bk}e^{Ak}U^n, \tag{11.20}$$

whereas the exact solution satisfies

$$u(t_{n+1}) = e^{(A+B)k}u(t_n). \tag{11.21}$$

By Taylor series expansion of the matrix exponentials (see (D.31)), we find that

$$e^{(A+B)k} = I + k(A+B) + \frac{1}{2}k^2(A+B)^2 + \cdots, \tag{11.22}$$

whereas

$$\begin{aligned} e^{Bk}e^{Ak} &= \left(I + kA + \frac{1}{2}k^2A^2 + \cdots\right)\left(I + kB + \frac{1}{2}k^2B^2 + \cdots\right) \\ &= I + k(A+B) + \frac{1}{2}k^2(A^2 + 2AB + B^2) + \cdots. \end{aligned} \tag{11.23}$$

Note that the quadratic term in (11.22) is

$$(A+B)^2 = A^2 + AB + BA + B^2,$$

which is not the same as the quadratic term in (11.23) if the matrices A and B do not commute.

If they do commute, e.g., in the scalar case, then the splitting is exact and all terms in the Taylor series agree. But in general a one-step error of magnitude $O(k^2)$ is introduced, and so the method is only first order accurate even when the exact solution is used for each piece.

A second order accurate splitting was introduced by Strang [83] in the context of methods for multidimensional hyperbolic equations and is often called the *Strang splitting*:

$$\begin{aligned} U^* &= \mathcal{N}_A(U^n, k/2), \\ U^{**} &= \mathcal{N}_B(U^*, k), \\ U^{n+1} &= \mathcal{N}_A(U^{**}, k/2). \end{aligned} \tag{11.24}$$

Working out the product of the Taylor series expansions in this case for the ODE system (11.18) gives agreement to the quadratic term in (11.22), although there is an error in the $O(k^3)$ term unless A and B commute. A similar result can be shown for general PDEs with smooth solutions split in the form (11.2).

An alternative procedure, which also gives second order accuracy, is to use the splitting (11.17) in every other time step, and in the alternate time steps use a similar splitting but with the order of $\mathcal{N}_A$ and $\mathcal{N}_B$ reversed. Over two time steps this has roughly the same form as the Strang splitting over a time step of length $2k$, although with two applications of $\mathcal{N}_B$ with time step k rather than one application with time step $2k$.

Example 11.1. The LOD method for the heat equation discussed in Section 9.8 uses a splitting of the form (11.17) but is able to achieve second order accuracy because there is no splitting error in this case (except near the boundaries, where appropriate treatment is required). In this case the two-dimensional heat equation is split with $\mathcal{A}(u) = u_{xx}$ and $\mathcal{B}(u) = u_{yy}$, and the operators ∂_x^2 and ∂_y^2 commute. For a more general variable coefficient heat equation with $\mathcal{A}(u) = (\kappa(x, y)u_x)_x$ and $\mathcal{B}(u) = (\kappa(x, y)u_y)_y$, the two operators no longer commute and the LOD method would be only first order accurate.

Another possible way to improve the accuracy of fractional step methods is to combine them with the spectral deferred correction method of [28]. This method improves the accuracy of a time-stepping procedure by a deferred correction process. Application to advection-diffusion-reaction equations in the context of fractional step methods was investigated in [9].

One difficulty with fractional step methods is that boundary conditions may be hard to apply properly when initial boundary value problems are solved. Each application of $\mathcal{N}_A$ or $\mathcal{N}_B$ typically requires boundary conditions, either physical or artificial, and it is not always clear how to properly specify the "intermediate boundary conditions" needed in each stage of the splitting. This has been discussed in relation to the LOD method for the heat equation in Section 9.8. See [65] for a discussion of intermediate boundary conditions for hyperbolic equations.

Another potential difficulty is stability. Even if the methods $\mathcal{N}_A$ and $\mathcal{N}_B$ are each stable methods for the problems they are designed to solve, it is not always clear that alternating between these methods in every time step will lead to a stable procedure. Example D.3 shows the problem that can arise. Suppose $\mathcal{N}_A(U^n, k) = A_0U^n$ and $\mathcal{N}_B(U^*, k) = A_1u^*$, where A_0 and A_1 are given by (D.84). Then each method is stable by itself but the fractional step procedure (11.17) generates exponentially growing solutions. Often stability of fractional step methods can be easily shown, for example, if $\|\mathcal{N}_A(U, k)\| \le \|U\|$ and $\|\mathcal{N}_B(U, k)\| \le \|U\|$ both hold in the same norm, but caution is required.

11.5 Implicit-explicit methods

Suppose we have an equation split as in (11.2), where $\mathcal{A}(u)$ represents stiff terms that we wish to integrate using an implicit method, whereas $\mathcal{B}(u)$ corresponds to nonstiff terms that can be handled explicitly with a reasonable time step. We have seen various examples of this form, such as reaction-diffusion equations with nonstiff reactions, where it may be much more efficient to avoid an implicit solve for the nonlinear reaction terms.

Implicit-explicit (IMEX) methods are fully coupled methods that are designed to handle some terms implicitly and others explicitly. A simple example is obtained by combining forward Euler with backward Euler:

$$U^{n+1} = U^n + k(\mathcal{A}(U^{n+1}) + \mathcal{B}(U^n)). \tag{11.25}$$

Another example is a two-step combination of the second order Adams–Bashforth method for the explicit term with the trapezoidal method for the implicit term:

$$U^{n+1} = U^n + \frac{k}{2}\left(\mathcal{A}(U^n) + \mathcal{A}(U^{n+1}) + 3\mathcal{B}(U^n) - \mathcal{B}(U^{n-1})\right). \tag{11.26}$$

Higher order methods of this type have been derived and widely used. See, for example, [7] for a number of other multistep methods and [6] for some Runge–Kutta methods of this type.

11.6 Exponential time differencing methods

Consider a nonlinear ODE $u' = f(u)$ (possibly an MOL discretization of a PDE) and suppose that over the time interval $[t_n, t_{n+1}]$ we write this as

$$u'(t) = A_n u(t) + \mathcal{B}_n(u(t)), \tag{11.27}$$

where we have split the function $f(u)$ into a linear part and a nonlinear part. The idea of *exponential time differencing* (ETD) methods is to use a form of Duhamel's principle (5.8) to handle the linear part exactly using the matrix exponential and combine this with an appropriate numerical method of the desired order for the $\mathcal{B}_n(u)$ term, typically an explicit method if we assume that the linear term captures the stiff part of the problem.

Two common forms of this type of splitting are as follows:

1. For a general nonlinear function $f(u)$, let $A_n = f'(U^n)$, the Jacobian matrix evaluated at U^n (or perhaps some approximate Jacobian), and then

$$\mathcal{B}_n(u) = f(u) - A_n u. \tag{11.28}$$

2. For problems such as MOL discretizations of reaction-diffusion equations we may take A_n to be the matrix representing the diffusion operator for all n and let $\mathcal{B}_n(u)$ be the reaction terms. In this case A_n is not the full Jacobian of the nonlinear problem, but if the reaction terms are not stiff they might be easily approximated with explicit methods, and there are advantages to having A unchanged from one step to the next—the ETD methods require working with the matrix exponential e^{kA_n}, and if A is constant we may be able to compute this once before beginning the time stepping.

For the system (11.27), Duhamel's principle (5.8) can be generalized to

$$u(t_{n+1}) = e^{A_n k} u(t_n) + \int_{t_n}^{t_{n+1}} e^{A_n(t_{n+1}-\tau)} \mathcal{B}_n(u(\tau))\, d\tau. \tag{11.29}$$

This expression is exact, but the integral must be approximated since we don't know $\mathcal{B}_n(u(\tau))$. Methods of various order can be obtained by different discretizations of this integral. The simplest approximation is obtained by replacing $\mathcal{B}_n(u(\tau))$ with $\mathcal{B}_n(U^n)$. We can then pull this out of the integral and can compute the exact integral of the remaining integrand by integrating the Taylor series for the matrix exponential (D.31) term by term, resulting in (5.12),

$$\begin{aligned} \int_{t_n}^{t_{n+1}} e^{A_n(t_{n+1}-\tau)}\, d\tau &= k + \frac{1}{2}k^2 A_n + \frac{1}{6}k^3 A_n^2 + \cdots \\ &= A_n^{-1}\left(e^{A_n k} - I\right) \quad \text{(if } A_n \text{ is nonsingular).} \end{aligned} \tag{11.30}$$

Using this, we obtain from (11.29) the numerical method

$$U^{n+1} = e^{A_n k} U^n + A_n^{-1} \left(e^{A_n k} - I \right) \mathcal{B}_n(U^n). \tag{11.31}$$

Since $\mathcal{B}_n(U^n) = f(U^n) - A_n U^n$, we can rewrite this as

$$U^{n+1} = U^n + A_n^{-1} \left(e^{A_n k} - I \right) f(U^n). \tag{11.32}$$

Note that if $A_n = 0$, then using the first line of (11.30) we see that (11.32) reduces to Euler's method for $u' = f(u)$ and is only first order accurate. However, we normally assume that A_n is nonsingular and an approximation to the Jacobian matrix. In general we can compute the local truncation error to be

$$\begin{aligned}
\tau^n &= \left(\frac{u(t_{n+1}) - u(t_n)}{k} \right) - k^{-1} A_n^{-1} \left(e^{A_n k} - I \right) u'(t_n) \\
&= \left[u'(t_n) + \frac{1}{2} k u''(t_n) + \frac{1}{6} k^2 u'''(t_n) + \cdots \right] \\
&\quad - \left[I + \frac{1}{2} A_n k + \frac{1}{6} A_n^2 k^2 + \cdots \right] u'(t_n) \\
&= \frac{1}{2} k \left(u''(t_n) - A_n u'(t_n) \right) + \frac{1}{6} k^2 \left(u'''(t_n) - A_n u'(t_n) \right) + \cdots \\
&= \frac{1}{2} k \left(f'(u(t_n)) - A_n \right) u'(t_n) + O(k^2).
\end{aligned} \tag{11.33}$$

We see that the method is second order accurate if $A_n = f'(U^n)$.

Higher order methods can be derived by using better approximations of the integral in (11.29). This can be done either as a multistep method, approximating $\mathcal{B}_n(u)$ by an interpolating polynomial through previous values U^{n-j} as in the derivation of the Adams–Bashforth methods, or as multistage generalizations of the Runge–Kutta methods. See, for example, [8], [19], [48], [53] for more discussion of these methods.

Note that the ETD method is exact on the test problem $u' = \lambda u$ if we take $A_n = \lambda$. So the region of absolute stability for this method is exactly the left half-plane. The method is exact more generally on a linear system of equations, provided of course that we can compute the matrix exponential accurately, as discussed in the next section.

Many mixed equations involve higher order derivative terms that are linear (and often constant coefficient) and ETD methods may be particularly suitable for handling the stiffness of spatial discretizations. Note in particular that for dispersive terms, such as the u_{xxx} term in the KdV equation (11.11), an ETD method that handles this term exactly may be advantageous over an implicit method. This dispersion is nondissipative (eigenvalues are on the imaginary axis), but many implicit methods designed for stiff problems have the imaginary axis in the interior of the stability region, leading to nonphysical dissipation.

11.6.1 Implementing exponential time differencing methods

Computing the matrix exponential is nontrivial—the classic paper [70] presented "19 dubious ways" to do this, and its recent update [71] discusses a 20th way in the appendix,

a more recently developed approach based on Krylov space methods. The latter approach has made the exponential time differencing approach viable for MOL discretizations of parabolic equations and other linear systems of ODEs involving large but sparse coefficient matrices and is discussed further below.

One situation in which ETD methods are relatively easy to implement is when the matrix A_n is diagonal, for then $e^{A_n k}$ is just a diagonal matrix of scalar exponential functions. This arises naturally in some applications, for example, if a problem such as a reaction-diffusion equation is solved with periodic boundary conditions. By Fourier transforming the problem, the diffusion operator is reduced to a diagonal matrix. For this reason ETD methods are often particularly attractive in connection with Fourier spectral methods.

Even in the scalar case, however, the evaluation of the exponential factor

$$\phi(z) = (e^z - 1)/z \tag{11.34}$$

that appears in (11.32) can be susceptible to numerical cancellation effects in floating point arithmetic. For higher order ETD methods such as the fourth order method considered by Cox and Matthews [19], higher order terms of the same nature appear that are even more sensitive to numerical errors. Kassam and Trefethen [53] suggest an approach to evaluating these coefficients in the numerical method using contour integration in the complex plane, numerically approximating the Cauchy integral representation (D.4).

In the nondiagonal case, directly computing the matrix exponential by this sort of approach can still be very effective if the matrix A involved is of modest size, such as may arise from a spectral approximation based on polynomials.

For very large sparse matrices, the Krylov space approach often works best. In this case we do not compute the matrix exponential itself, which is a very large dense matrix, but rather the application of this matrix to a vector. This is all that is needed in (11.32), for example. Actually we need to apply $A^{-1}(e^{Ak} - I) = k\phi(Ak)$ to a vector, which could be done in two steps, first using a Krylov space method for the exponential and then a second Krylov space method to solve the linear system, but the Krylov approach can be applied directly to the function $\phi(Ak)$. This approach has been briefly outlined at the end of Section 4.4. In practice it has been found that in some cases, particularly if a good preconditioner is not available, Krylov space methods may converge faster on the matrix exponential and related functions than it does for a simple linear system with the same coefficient matrix. In such cases the ETD methods may be more efficient than using a traditional implicit method. See, e.g., [32], [48], [77] for more details.