To appear in SIAM J. Sci. Comp. Revised, November 1996

EXPONENTIAL INTEGRATORS FOR LARGE SYSTEMS OF DIFFERENTIAL EQUATIONS

MARLIS HOCHBRUCK, CHRISTIAN LUBICH, AND HUBERT SELHOFER*

Abstract. We study the numerical integration of large stiff systems of differential equations by methods that use matrix-vector products with the exponential or a related function of the Jacobian. For large problems, these can be approximated by Krylov subspace methods, which typically converge faster than those for the solution of the linear systems arising in standard stiff integrators. The exponential methods also offer favorable properties in the integration of differential equations whose Jacobian has large imaginary eigenvalues. We derive methods up to order 4 which are exact for linear constant-coefficient equations. The implementation of the methods is discussed. Numerical experiments with reaction-diffusion problems and a time-dependent Schrödinger equation are included.

Key words. Numerical integrator, high-dimensional differential equations, matrix exponential, Krylov subspace methods.

AMS(MOS) subject classifications. 65L05, 65M15, 65F10.

1. Introduction. The idea to use the exponential function of the Jacobian in a numerical integrator is by no means new, but it has mostly been regarded as rather impractical. Since the mid-eighties, Krylov subspace approximations to the action of the matrix exponential operator have, however, been found to be useful in Chemical Physics [16, 20, 22] and subsequently also in other fields [6, 8, 9, 21, 24, 29]. On the numerical analysis side, the convergence of such Krylov approximations was studied in [4, 5, 13, 26]. It was shown in [13], and previously in [4] for the symmetric case, that Krylov approximations to $\exp(\tau A)v$ converge substantially faster than those for the solution of linear systems $(I - \tau A)x = v$, at least unless a good preconditioner is available. Such linear systems arise in the numerical integration of stiff differential equations by standard integrators. For large problems, their solution often dominates the computational work.

For nonlinear differential equations, the exponential of the Jacobian combined with Krylov approximations has previously been used in generalizations of Adams-type multistep methods in [8]. On the other hand, the use of matrix exponentials has for a long time been prominent in the exponential fitting literature, see e.g. [7, 17], and [1, 2] as recent examples.

In this paper, we study new numerical methods for the integration of large stiff systems of nonlinear initial value problems

$$y' = f(y), \qquad y(t_0) = y_0 \ . \tag{1.1}$$

The methods proposed here use matrix-vector multiplications $\varphi(\tau A)v$, where A is the Jacobian of f, τ is related to the step size, and $\varphi(z) = (e^z - 1)/z$. This choice allows us to obtain methods that are exact for constant-coefficient linear problems

$$y' = Ay + b \ . \tag{1.2}$$

[*] Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, D–72076 Tübingen, Germany. E-mail: marlis@na.uni-tuebingen.de, lubich@na.uni-tuebingen.de, hubert@na.uni-tuebingen.de

We remark that Krylov subspace approximations to $\varphi(\tau A)v$ converge about as fast as those for $\exp(\tau A)v$, see [13].

Potential advantages for exponential integrators can thus originate from two different sources: Computing $\varphi(\tau A)v$ can be less expensive than solving $(I - \tau A)x = v$, and the exponential integration method itself may behave more favorably than standard integrators. The latter case occurs in particular for mildly nonlinear differential equations whose Jacobian has large imaginary eigenvalues, e.g., wave equations, Schrödinger equations, flexible mechanical systems, and oscillatory electric circuits. Standard stiff integrators either damp high frequencies or map them to one and the same frequency (or nearly so) in the discretization, neither of which may be desirable.

In Section 2 we give some simple methods of order 2 that are exact for (1.2) or for linear second-order differential equations. They include new symmetric methods, which appear useful for long-time integration of conservative, time-reversible problems.

In Section 3 we consider a class of methods that would reduce to explicit Runge-Kutta methods if $\varphi(z) = (e^z - 1)/z$ were replaced by $\varphi(z) \equiv 1$, and to Rosenbrock-Wanner methods for $\varphi(z) = 1/(1-z)$. We give order conditions, both for exact and inexact Jacobian, and derive sufficient and necessary conditions to ensure that (1.2) is solved exactly.

In Section 4 we extend the methods to differential-algebraic systems. We derive order conditions up to order 3 for such problems and for singularly perturbed systems.

In Section 5 we construct methods of classical order 4 which are exact for (1.2) and have further favorable properties when applied to stiff problems. In particular, we use a reformulation that reduces the computational work for the Krylov processes.

Section 6 deals with implementation issues. Important topics are how to take into account the computational work and storage requirements of the Krylov process in the step size control, and when to stop the Krylov process.

Based on the considerations of Sections 5 and 6, we have written a code `exp4`, which can be obtained via anonymous ftp from na.uni-tuebingen.de in the directory pub/codes/exp4.

In Section 7 we describe numerical experiments with this code for reaction-diffusion problems and for a Schrödinger equation with time-dependent Hamiltonian, which show both the scope and the limitations of using Krylov approximations in exponential integrators.

In a final section we discuss conclusions and perspectives for the methods proposed in this article.

We will describe the methods only for autonomous differential equations (1.1). For non-autonomous problems $y' = f(t, y)$, the methods should be applied to the extended formally autonomous system obtained by adding the trivial differential equation $t' = 1$. The methods are then exact for linear differential equations whose inhomogeneity is linear in t.

2. Simple methods of order 2.

2.1. The exponentially fitted Euler method. The prototype exponential method, which seems to have appeared repeatedly in the literature under various disguises, is

$$y_1 = y_0 + h\varphi(hA)f(y_0) \;, \tag{2.1}$$

where h is the step size, $A = f'(y_0)$, and

$$\varphi(z) = \frac{e^z - 1}{z} \ . \tag{2.2}$$

The method is of order 2, and exact for linear differential equations (1.2).

2.2. A symmetric exponential method. For long-time integration of conservative problems, time reversibility is an important property. A symmetric method of order 2 is given by the two-step formula

$$y_{n+1} - y_n = e^{hA}(y_{n-1} - y_n) + 2h\varphi(hA)f(y_n), \tag{2.3}$$

with $A = f'(y_n)$ and φ given by (2.2). The method is exact for linear problems (1.2) provided that the starting values y_0 and y_1 are exact. The method can be viewed as a generalization of the explicit midpoint rule, to which it reduces for $A = 0$. The characteristic roots of the method applied to $y' = \lambda y$ are $e^{h\lambda}$ and -1 which shows that the method is A-stable. The oscillatory error component $(-1)^n$ can be eliminated by taking the average of two successive values, $(y_n + y_{n+1})/2$, as an approximation to $y(t_n + h/2)$.

2.3. A cosine method for second-order differential equations. We now consider

$$y'' = f(y), \qquad y(0) = y_0, \quad y'(0) = y_0' \ . \tag{2.4}$$

For the linear problem

$$y'' = Ay + b \tag{2.5}$$

the exact solution satisfies

$$y(t+h) - 2y(t) + y(t-h) = h^2\psi(h^2A)(Ay(t) + b) \ , \tag{2.6}$$

with the entire function

$$\psi(z) = \frac{\cos\sqrt{-z} - 1}{z/2}.$$

This motivates the scheme

$$y_{n+1} - 2y_n + y_{n-1} = h^2\psi(h^2A)f(y_n) \ , \tag{2.7}$$

with $A = f'(y_n)$, which is a symmetric method of order 2 for (2.4). Because of (2.6), the scheme is exact for linear problems (2.5).

Derivative approximations that are exact for (2.5) are obtained via

$$y_{n+1}' - y_{n-1}' = 2h\sigma(h^2A)f(y_n), \tag{2.8}$$

where

$$\sigma(z) = \frac{\sin\sqrt{-z}}{\sqrt{-z}} \ .$$

3. Higher-order exponential one-step methods: Order conditions and stability. In this section we study a general class of exponential integration methods introduced in [13]. Starting with y_0 as an approximation to $y(t_0)$, an approximation to $y(t_0+h)$ is computed via

$$k_i = \varphi(\gamma hA)\left(f(u_i) + hA\sum_{j=1}^{i-1}\gamma_{ij}k_j\right), \qquad i = 1,\ldots,s \tag{3.1}$$

$$u_i = y_0 + h\sum_{j=1}^{i-1}\alpha_{ij}k_j \tag{3.2}$$

$$y_1 = y_0 + h\sum_{i=1}^{s}b_ik_i\ . \tag{3.3}$$

Here $A = f'(y_0)$, and $\gamma, \gamma_{ij}, \alpha_{ij}, b_i$, with $\gamma_{ij} = \alpha_{ij} = 0$ for $i \le j$, are the coefficients that determine the method. The internal stages $u_1,\ldots,u_s$ can be computed one after the other, with one multiplication by $\varphi(\gamma hA)$ and a function evaluation at each stage. The scheme would become an explicit Runge-Kutta method for $\varphi(z) \equiv 1$ (and $\gamma_{ij} \equiv 0$), and a Rosenbrock-Wanner method for the choice $\varphi(z) = 1/(1-z)$. As in the exponential Euler method (2.1), we choose instead the function (2.2).

3.1. Order conditions when using the exact Jacobian. Our aim now is to construct higher-order methods. The order conditions for the exponential methods can be derived similarly to Rosenbrock-Wanner methods, see, e.g., [12, Section IV.7]. Therefore, we only state the conditions here. For abbreviation we define

$$\beta_{ij} := \alpha_{ij} + \gamma_{ij}. \tag{3.4}$$

THEOREM 3.1. *An exponential method* (3.1)–(3.3) *with* $A = f'(y_0)$ *is of order* p *iff*

$$\sum_{j=1}^{s} b_j\Phi_j(\tau) = P_\tau(\gamma)$$

for elementary differentials τ *up to order* p. *Here,* $\Phi_j(\tau)$ *and the polynomials* $P_\tau(\gamma)$ *are listed in Table* 3.1 *for* $p \le 5$.

The only difference to the order conditions for Rosenbrock-Wanner methods is in the polynomials $P_\tau(\gamma)$.

THEOREM 3.2. *The method* (3.1)–(3.3) *is exact for linear differential equations* (1.2), *iff for all* $n = 1,2,3,\ldots$

$$\sum b_{j_1}\beta_{j_1,j_2}\beta_{j_2,j_3}\ldots\beta_{j_{n-1},j_n} = \frac{1}{n}\left(\frac{1}{n-1}-\gamma\right)\left(\frac{1}{n-2}-\gamma\right)\cdots\left(\frac{1}{2}-\gamma\right)(1-\gamma). \tag{3.5}$$

These conditions can be fulfilled if γ is the reciprocal of an integer. Then only a finite number of these conditions are needed. The others are satisfied automatically because for sufficiently large n, both sides of (3.5) then vanish.

Proof. For the linear problem (1.2), both the exact and the numerical solution depend analytically on h. Since only the elementary differentials f, $f'f$, $f'f'f$, $f'f'f'f$, ...are nonvanishing for (1.2), it thus suffices to show that their order conditions are given by (3.5). Like for Rosenbrock methods, one obtains that they are of the form

$$\sum b_{j_1}\beta_{j_1,j_2}\beta_{j_2,j_3}\ldots\beta_{j_{n-1},j_n} = P_{n-1}(\gamma), \tag{3.6}$$

Elementary differential τ	$\Phi_j(\tau)$	$P_\tau(\gamma)$
f	1	1
$f'f$	$\sum_k \beta_{jk}$	$1/2(1-\gamma)$
$f''(f,f)$	$\sum_{k,l} \alpha_{jk}\alpha_{jl}$	$1/3$
$f'f'f$	$\sum_{k,l} \beta_{jk}\beta_{kl}$	$1/3(1/2-\gamma)(1-\gamma)$
$f'''(f,f,f)$	$\sum_{k,l,m} \alpha_{jk}\alpha_{jl}\alpha_{jm}$	$1/4$
$f''(f'f,f)$	$\sum_{k,l,m} \alpha_{jk}\beta_{kl}\alpha_{jm}$	$1/8-\gamma/6$
$f'f''(f,f)$	$\sum_{k,l,m} \beta_{jk}\alpha_{kl}\alpha_{km}$	$1/12-\gamma/6$
$f'f'f'f$	$\sum_{k,l,m} \beta_{jk}\beta_{kl}\beta_{lm}$	$1/4(1/3-\gamma)(1/2-\gamma)(1-\gamma)$
$f^{(4)}(f,f,f,f)$	$\sum \alpha_{jk}\alpha_{jl}\alpha_{jm}\alpha_{jp}$	$1/5$
$f'''(f'f,f,f)$	$\sum \alpha_{jk}\beta_{kl}\alpha_{jm}\alpha_{jp}$	$1/10-\gamma/8$
$f''(f,f''(f,f))$	$\sum \alpha_{jk}\alpha_{kl}\alpha_{km}\alpha_{jp}$	$1/15$
$f''(f'f'f,f)$	$\sum \alpha_{jk}\beta_{kl}\beta_{lm}\alpha_{jp}$	$1/30-\gamma/8+\gamma^2/9$
$f''(f'f,f'f)$	$\sum \alpha_{jk}\beta_{kl}\alpha_{jm}\beta_{mp}$	$1/20-\gamma/8+\gamma^2/12$
$f'f'''(f,f,f)$	$\sum \beta_{jk}\alpha_{kl}\alpha_{km}\alpha_{kp}$	$1/20-\gamma/8$
$f'f''(f'f,f)$	$\sum \beta_{jk}\alpha_{kl}\beta_{lm}\alpha_{kp}$	$1/40-5\gamma/48+\gamma^2/12$
$f'f'f''(f,f)$	$\sum \beta_{jk}\beta_{kl}\alpha_{lm}\alpha_{lp}$	$1/60-\gamma/12+\gamma^2/9$
$f'f'f'f'f$	$\sum \beta_{jk}\beta_{kl}\beta_{lm}\beta_{mp}$	$1/5(1/4-\gamma)(1/3-\gamma)(1/2-\gamma)(1-\gamma)$

Table 3.1
Order conditions for exponential methods up to order 5

where P_{n-1} is a polynomial of degree at most $n-1$ which depends on the choice of φ but not on the method coefficients. It remains to show that $P_{n-1}(\gamma)$ is given by the right-hand side of (3.5). If $\gamma = 0$, then the method applied to (1.2) is just a Runge-Kutta method with coefficients β_{jk} and weights b_i. From the order conditions for Runge-Kutta methods, we thus have

$$P_{n-1}(0) = 1/n!\ .$$

The exponential Euler method (2.1) is a one-stage method (3.1)–(3.3) with $b_1 = 1$, $\beta_{11} = 0$, and with $\gamma = 1$. Obviously,

$$\sum b_{j_1}\beta_{j_1,j_2}\beta_{j_2,j_3}\ldots\beta_{j_{n-1},j_n} = 0 \tag{3.7}$$

for $n > 1$ for this method. Since we already know that the exponential Euler method is exact for (1.2) we conclude from (3.6) that

$$P_{n-1}(1) = 0 \qquad \text{for } n > 1\ .$$

Similarly, two consecutive steps of the exponential Euler method with step size $h/2$ can be viewed as one step of a two-stage method (3.1)–(3.3) with $\gamma = 1/2$. For such a method, (3.7) is valid for $n > 2$. As before, we conclude from (3.6) that

$$P_{n-1}(1/2) = 0 \qquad \text{for } n > 2\ .$$

Continuing this argument for $3, 4, \ldots$ steps of the exponential Euler method with step sizes $h/3, h/4, \ldots$, we obtain

$$P_{n-1}(1/j) = 0 \qquad \text{for } j < n\ .$$

Elementary differential τ	$\Phi_j(\tau)$	$P_\tau(\gamma)$
f	1	1
$f'f$ Af	$\sum_k \alpha_{jk}$ $\sum_k \gamma_{jk}$	$1/2$ $-\gamma/2$
$f''(f,f)$ $f'f'f$ $f'Af$ $Af'f$ AAf	$\sum_{k,l} \alpha_{jk}\alpha_{jl}$ $\sum_{k,l} \alpha_{jk}\alpha_{kl}$ $\sum_{k,l} \alpha_{jk}\gamma_{kl}$ $\sum_{k,l} \gamma_{jk}\alpha_{kl}$ $\sum_{k,l} \gamma_{jk}\gamma_{kl}$	$1/3$ $1/6$ $-\gamma/4$ $-\gamma/4$ $\gamma^2/3$

TABLE 3.2
Order conditions for exponential W-methods up to order 3

It follows that $P_{n-1}(\gamma)$ is given by the right-hand side of (3.5). □

3.2. Order conditions for inexact Jacobians. One may also want to use the method with an approximate Jacobian A. This requires further restrictions on the method parameters. For order 3 the conditions are given in Table 3.2. They are the same as for W-methods, see [12, p. 124], except for different polynomials in γ.

If the first five conditions of Table 3.2 are satisfied, then the method is of order 3 when $A - f'(y_0) = O(h)$, cf. [15] for the analogous situation in W-methods.

3.3. Stability. When the method is exact for linear differential equations, it is trivially A-stable. Much more can then in fact be shown about stability, including the practical situation where $\varphi(\gamma hA)v$ is computed only approximately. Consider a perturbed method (3.1)–(3.3) applied to the linear problem (1.2):

$$\begin{aligned} \widetilde{k}_i &= \varphi(\gamma hA)\left(A\widetilde{y}_0 + b + hA\textstyle\sum_{j=1}^{i-1}\beta_{ij}\widetilde{k}_j\right) + \delta_i \\ \widetilde{y}_1 &= \widetilde{y}_0 + h\textstyle\sum_{i=1}^{s} b_i\widetilde{k}_i \ . \end{aligned}$$

Here, δ_i is a perturbation at the ith stage, and $\widetilde{y}_0$ is a perturbed starting value. Subtracting from the unperturbed scheme yields for the error $\varepsilon_1 = \widetilde{y}_1 - y_1$

$$\begin{aligned} \ell_i &= \varphi(\gamma hA)\left(A\varepsilon_0 + hA\textstyle\sum_{j=1}^{i-1}\beta_{ij}\ell_j\right) + \delta_i \\ \varepsilon_1 &= \varepsilon_0 + h\textstyle\sum_{i=1}^{s} b_i\ell_i \ , \end{aligned}$$

where $\ell_j = \widetilde{k}_j - k_j$ and $\varepsilon_0 = \widetilde{y}_0 - y_0$. It is easy to see that

$$\varepsilon_1 = e^{hA}\varepsilon_0 + h\sum_{i=1}^{s} b_i\, p_{s-i}(e^{\gamma hA} - I)\,\delta_i \ ,$$

where $p_k(z)$ is a polynomial of degree k with $p_k(0) = 1$, whose coefficients are products of β_{ij}/γ. In particular, when the numerical range of A is contained in the left half-plane, then we have the stable error recurrence

$$\|\varepsilon_1\| \leq \|\varepsilon_0\| + Ch\sum_{i=1}^{s}\|\delta_i\| \ .$$

The stability analysis could be extended to nonlinear problems $y' = Ay + g(y)$ with Lipschitz-bounded g, to singularly perturbed problems, and to nonlinear parabolic problems in a similar way to what has been done for Rosenbrock methods, cf. [10, 18, 30].

4. Exponential methods for differential-algebraic and singularly perturbed problems. As with Rosenbrock-Wanner methods [12, Section VI.3], the method (3.1)–(3.3) can be extended to differential-algebraic equations

$$\begin{aligned} y' &= f(y,z) \\ 0 &= g(y,z), \qquad g_z \text{ invertible} \end{aligned} \tag{4.1}$$

by applying it formally to the singularly perturbed differential equation

$$\begin{aligned} y' &= f(y,z) \\ \epsilon z' &= g(y,z) \end{aligned} \tag{4.2}$$

and letting $\epsilon \to 0$ in the scheme. This will give us the following method:

$$\begin{aligned} \begin{pmatrix} k_i \\ l_i \end{pmatrix} = & \begin{bmatrix} I & 0 \\ -g_z^{-1} g_y & I \end{bmatrix} \begin{bmatrix} \varphi(\gamma h B) & 0 \\ 0 & -(\gamma h g_z)^{-1} \end{bmatrix} \begin{bmatrix} I & -f_z g_z^{-1} \\ 0 & I \end{bmatrix} \cdot \\ & \left\{ \begin{pmatrix} f(u_i,v_i) \\ g(u_i,v_i) \end{pmatrix} + h \begin{bmatrix} f_y & f_z \\ g_y & g_z \end{bmatrix} \sum_{j=1}^{i-1} \gamma_{ij} \begin{pmatrix} k_i \\ l_i \end{pmatrix} \right\}, \end{aligned} \tag{4.3}$$

where $B = f_y - f_z g_z^{-1} g_y$, where the Jacobians are evaluated at (y_0, z_0), and where u_i and v_i are defined by

$$\begin{pmatrix} u_i \\ v_i \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + h \sum_{j=1}^{i-1} \alpha_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix} . \tag{4.4}$$

Finally we set

$$\begin{pmatrix} y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + h \sum_{i=1}^{s} b_i \begin{pmatrix} k_i \\ l_i \end{pmatrix} . \tag{4.5}$$

The derivation uses the auxiliary assumption that the eigenvalues of g_z have negative real part. The Jacobian of (4.2) is block diagonalized by

$$X = \begin{bmatrix} I + O(\epsilon) & \epsilon f_z g_z^{-1} + O(\epsilon^2) \\ -g_z^{-1} g_y + O(\epsilon) & I + O(\epsilon) \end{bmatrix} ,$$

viz.,

$$\begin{bmatrix} f_y & f_z \\ \frac{1}{\epsilon} g_y & \frac{1}{\epsilon} g_z \end{bmatrix} = X \begin{bmatrix} B + O(\epsilon) & 0 \\ 0 & \frac{1}{\epsilon} g_z + O(1) \end{bmatrix} X^{-1} .$$

Since

$$\varphi(\tfrac{\gamma h}{\epsilon} g_z) = -\tfrac{\epsilon}{\gamma h} g_z^{-1} + O((\tfrac{\epsilon}{h})^2) \ ,$$

the method (3.1)–(3.3) applied to (4.2) reads

$$\begin{aligned}\begin{pmatrix} k_i \\ l_i \end{pmatrix} = {} & X \begin{bmatrix} \varphi(\gamma h B) + O(h\epsilon) & 0 \\ 0 & -\frac{\epsilon}{\gamma h} g_z^{-1} + O((\frac{\epsilon}{h})^2) \end{bmatrix} X^{-1} \\ & \left\{ \begin{pmatrix} f(u_i, v_i) \\ \frac{1}{\epsilon} g(u_i, v_i) \end{pmatrix} + h \begin{bmatrix} f_y & f_z \\ \frac{1}{\epsilon} g_y & \frac{1}{\epsilon} g_z \end{bmatrix} \sum_{j=1}^{i-1} \gamma_{ij} \begin{pmatrix} k_i \\ l_i \end{pmatrix} \right\} .\end{aligned}$$

We note that

$$\begin{bmatrix} I & 0 \\ 0 & \epsilon I \end{bmatrix} X^{-1} \begin{bmatrix} I & 0 \\ 0 & \frac{1}{\epsilon} I \end{bmatrix} = \begin{bmatrix} I & -f_z g_z^{-1} \\ 0 & I \end{bmatrix} + O(\epsilon) .$$

For $\epsilon \to 0$, these formulas lead to the above method (4.3).

Remark. The matrix B need not be computed when one uses Krylov methods to approximate $\varphi(\gamma h B)u$. Matrix vector multiplications with B are cheap when the action of g_z^{-1} is inexpensive to compute. For example, this is the case in constrained mechanical systems, cf. [12, p. 542],

$$\begin{aligned} q' &= v \\ v' &= a \\ \begin{bmatrix} M & G(q)^T \\ G(q) & 0 \end{bmatrix} \begin{pmatrix} a \\ \lambda \end{pmatrix} &= \begin{pmatrix} \phi(q, v) \\ \psi(q, v) \end{pmatrix} . \end{aligned}$$

Here, q and v are position and velocity variables, respectively, a is acceleration and λ represents the Lagrange multipliers. In this system, g_z corresponds to $\begin{bmatrix} M & G^T \\ G & 0 \end{bmatrix}$. In suitable multibody formulations, linear equations with this matrix can be solved in an amount of work proportional to the dimension. □

When the exponential method is exact for linear differential equations with constant inhomogeneity, then method (4.3)–(4.5) is exact for linear differential-algebraic equations

$$\begin{aligned} y' &= F_y y + F_z z + b \\ 0 &= G_y y + G_z z + c \end{aligned}$$

with constant matrices F_y, F_z, G_y, G_z (G_z invertible) and constant vectors b, c. Apart from a direct calculation, this may be seen as follows: When the eigenvalues of G_z have negative real part, the exactness is again obtained by letting $\epsilon \to 0$ in the singularly perturbed problem, which is solved exactly by the method. From this case, the exactness in the general situation of invertible G_z follows by analytical continuation.

In general, the application of this scheme to differential-algebraic equations results in an order reduction to order 2, unless the method coefficients satisfy additional conditions.

Theorem 4.1. *The method* (4.3)–(4.5) *is convergent of order* 3 *for the differential-algebraic equation* (4.1) *if it satisfies the order conditions of Table* 3.1 *up to order* 3, (3.5)*, and in addition*

$$\sum_{j,k,l,m} b_j \omega_{jk} \alpha_{kl} \alpha_{km} = 1 \qquad (4.6)$$

where $[\omega_{jk}] = (\beta + \gamma I)^{-1}$ *with* $\beta = [\beta_{jk}]$.

The additional order condition is the same as for Rosenbrock methods applied to (4.1) [12, p. 446]. Instead of giving a cumbersome formal proof of the theorem, we make the reappearance of condition (4.6) for exponential methods plausible as follows: Like the order conditions of Section 3, also the differential-algebraic order conditions are of the same form as for Rosenbrock methods, but possibly with different right-hand sides involving γ. We know that the theorem is valid for $\varphi(z) = 1/(1-z)$. The appearance of the ω_{jk} is related only to the term $(\gamma h g_z)^{-1}$ in (4.3), which is independent of φ. The terms α_{jk} are also unrelated to $\varphi(\gamma h B)$. Therefore, the condition remains the same as for Rosenbrock methods.

The differential-algebraic order condition (4.6) is important not only for differential-algebraic systems but also for stiff differential equations. For example, the third-order error bound of Rosenbrock methods for singularly perturbed problems (4.2) in Theorem 1 (case $r = 3$) of [10] can be shown to be valid also for exponential methods.

5. Construction of fourth-order methods.

5.1. Reduced methods. We recall that one step of the exponential method evaluated in the form (3.1)–(3.3) contains s multiplications of $\varphi(\gamma hA)$ with a vector. Since this vector is different in each of these s steps, the approximation with a Krylov subspace method requires the construction of bases of s Krylov spaces with respect to the same matrix A but with different vectors. This turns out to be prohibitively expensive. One may think of exploiting techniques for solving linear systems with multiple right-hand sides [25, 27], but in our experiments the savings achieved were minor. Therefore, we will present an alternative formulation of the method.

A key point for the construction of efficient methods is that one can compute $\varphi(jz)$, $j = 2, 3, \dots$, recursively from $\varphi(z)$:

$$
\begin{aligned}
\varphi(2z) &= \left(\tfrac{1}{2}z\varphi(z) + 1\right)\varphi(z) \\
\varphi(3z) &= \tfrac{2}{3}\left(z\varphi(z) + 1\right)\varphi(2z) + \tfrac{1}{3}\varphi(z) \\
\varphi(4z) &= \dots
\end{aligned}
\tag{5.1}
$$

Once we have computed $\varphi(\gamma hA)$, we can thus compute $\varphi(j\gamma hA)v$ for any integer $j > 1$ with the expense of matrix vector multiplications.

The recurrence (5.1) is equally useful for the more interesting case where $\varphi(j\gamma hA)v$ is approximated by Krylov methods. The Krylov subspace approximation is of the form

$$
\varphi(\tau A)v \approx V_m \varphi(\tau H_m)e_1 \cdot \|v\|_2 \tag{5.2}
$$

where $V_m = [v_1, \dots, v_m]$ is the matrix containing the Arnoldi (or Lanczos) basis of the mth Krylov subspace with respect to A and v, and H_m is the orthogonal (oblique) projection of A to the mth Krylov subspace, which is an $m \times m$ upper Hessenberg (block tridiagonal, respectively) matrix. Further, e_1 is the first m-dimensional unit vector.

The iteration number m is typically very small compared to the dimension of the matrix A, so that the matrix $\varphi(\gamma hH_m)$ can be computed quite cheaply (see §6 for details). Then the recurrence (5.1) can be used to compute $\varphi(j\gamma hH_m)e_1$ by performing matrix vector multiplications with the small matrices H_m and $\varphi(\gamma hH_m)$. If we denote the identity matrix of dimension m by I_m, then

$$\begin{aligned}
\varphi(2\tau A)v &\approx V_m\varphi(2\tau H_m)e_1\|v\|_2 \\
&= V_m\left(\tfrac{1}{2}\tau H_m\varphi(\tau H_m)+I_m\right)\varphi(\tau H_m)e_1\|v\|_2 \\
\varphi(3\tau A)v &\approx V_m\left(\tfrac{2}{3}\left(\tau H_m\varphi(\tau H_m)+I_m\right)\varphi(2\tau H_m)+\tfrac{1}{3}\varphi(\tau H_m)\right)e_1\|v\|_2\ .
\end{aligned}$$

We can exploit the recurrences (5.1) by reformulating the method. For this we introduce auxiliary vectors

$$d_i = f(u_i) - f(y_0) - hA\sum_{j=1}^{s}\alpha_{ij}k_j. \tag{5.3}$$

Note that for $A = f'(y_0)$, this corresponds to a first-degree Taylor expansion of f around y_0. Hence the vectors d_i are usually small in norm and would vanish for linear f. With (3.4) and (5.3) we have

$$k_i = k_1 + \varphi(\gamma hA)d_i + \varphi(\gamma hA)hA\sum_{j=1}^{s}\beta_{ij}k_j.$$

Because of (5.1) we can choose β_{kl} such that for $\gamma = 1/n$ and $i = 1,\ldots,n$

$$\begin{aligned}
k_i &= \varphi(i\gamma hA)f(y_0), \\
k_{nj+i} &= k_1 + \varphi(i\gamma hA)d_{nj+i}, \qquad j \geq 1.
\end{aligned} \tag{5.4}$$

All the coefficients β_{kl} are uniquely determined by (5.4). In order to apply the recurrence formulas (5.1) in (5.4) we further choose

$$\alpha_{nj+i,l} = \alpha_{nj+1,l}, \qquad i = 1,\ldots,n, \quad j,l \geq 1$$

which gives

$$\begin{aligned}
u_{nj+i} &= u_{nj+1}, \\
d_{nj+i} &= d_{nj+1},
\end{aligned} \qquad i = 1,\ldots,n, \quad j \geq 1.$$

This reduces the number of f-evaluations and of evaluations of $\varphi(\gamma hA)$ by a factor of n compared to the general scheme (3.1)–(3.3). This is particularly important when this *reduced* method is combined with a Krylov process for approximating $\varphi(\gamma hA)v$ since in this case we need to compute a basis of a new Krylov space only at every nth intermediate step. Moreover, since the vectors d_i are usually small in norm, the Krylov approximation of $\varphi(i\gamma hA)d_{nj+1}$ ($j \geq 1$) typically takes only few iterations to achieve the required accuracy. The cost for building up the Krylov space of A with respect to the vector $f(y_0)$ thus dominates the computational cost.

We note finally that we can reorganize the computations in (5.4) as

$$\begin{aligned}
\tilde{k}_i &= k_i = \varphi(\tfrac{i}{n}hA)f(y_0) \\
\tilde{k}_{nj+i} &= k_{nj+i} - k_1 = \varphi(\tfrac{i}{n}hA)d_{nj+1}
\end{aligned} \tag{5.5}$$

for $i = 1,\ldots,n$ and $j \geq 1$, and we can use the values $\tilde{k}_l$ in (3.2) and (3.3), with appropriately modified weights:

$$\tilde{\alpha}_{k,l} = \begin{cases} \alpha_{k,l} + \sum_{m>n}\alpha_{k,m} & l = 1 \\ \alpha_{k,l} & l > 1 \end{cases}$$

and

$$\tilde{b}_l = \begin{cases} b_l + \sum_{m>n} b_m & l = 1 \\ b_l & l > 1\ . \end{cases} \tag{5.6}$$

5.2. Methods of order 4. Next we show that the reduced scheme proposed above still allows the construction of higher-order methods. Here, we concentrate on $\gamma = 1/2$ and $\gamma = 1/3$ and start with a 3-stage method for $\gamma = 1/2$ that uses 2 function evaluations per step. The parameters β_{kl} satisfying (5.4) are given by

$$\beta = \begin{bmatrix} 0 & 0 & 0 \\ 1/4 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

To fulfill the conditions for order 4, there remain two free parameters $\alpha_{3,1}$, $\alpha_{3,2}$, and the weights b_j, $j = 1, \ldots, 4$. The order conditions from Table 3.1 have a unique solution

$$\alpha = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3/8 & 3/8 & 0 \end{bmatrix}, \qquad b^T = (-16/27,\ 1,\ 16/27)\,.$$

This yields the scheme

$$\begin{aligned} k_1 &= \varphi(\tfrac{1}{2}hA)f(y_0) \\ k_2 &= \varphi(hA)f(y_0) \\ w_3 &= \tfrac{3}{8}(k_1 + k_2) \\ u_3 &= y_0 + hw_3 \\ d_3 &= f(u_3) - f(y_0) - hAw_3 \\ k_3 &= \varphi(\tfrac{1}{2}hA)d_3 \\ y_1 &= y_0 + h(k_2 + \tfrac{16}{27}k_3)\ . \end{aligned} \tag{5.7}$$

On k_3 we have omitted the tilde corresponding to (5.5). This method is of order 4, and exact for linear differential equations (1.2). However, it is only of first order when used with inexact Jacobian and of second order when applied to DAEs. Moreover, it is impossible to construct an embedded method of order 3, which makes it hard to perform a reliable estimation of local errors for step size control. The only cheap variant is to use the exponential Euler method (2.1), which is only of order 2 and thus tends to overestimate the local error.

The method (5.7) with embedded (2.1) is however of interest as a very economical method in situations where the time step is not restricted by accuracy, but only by the convergence of the Krylov process for computing $\varphi(hA)f(y_0)$. We note that k_3 is usually well approximated in a low dimensional Krylov space, because d_3 is often much smaller in norm than $f(y_0)$.

A more sophisticated method can be constructed with $\gamma = 1/3$ and $s = 7$, using

3 function evaluations per step. The parameters for (5.4) are given by

$$\beta = \left[\begin{array}{ccccccc} 0 & & & & & & 0 \\ 1/6 & 0 & & & & & \\ 1/9 & 2/9 & 0 & & & & \\ 0 & 0 & 0 & 0 & & & \\ -1/6 & 0 & 0 & 1/6 & 0 & & \\ -1/3 & 0 & 0 & 1/9 & 2/9 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right].$$

With these parameters β_{kl}, all the order conditions (3.5) for linear problems are satisfied automatically for $n \geq 4$.

For our method we choose to evaluate the function f at both end points and at the middle of the time interval, i.e.,

$$\sum_{j=1}^{3} \alpha_{4,j} := \frac{1}{2}, \qquad \sum_{j=1}^{6} \alpha_{7,j} := 1\ .$$

The solution is obtained by first solving the order condition up to order 4 from Table 3.1. The equations for $f''(f,f)$ and $f'f$ immediately yield $b_3 = 1$ and $b_2 = 0$. Next the conditions for $f'f'f$, $f'''(f,f,f)$, $f'f''(f,f)$ and (4.6) result in a linear system with four equations for the unknowns b_j, $j = 4,\dots,7$. This system has the unique solution $b_4 = b_6 = 1$, $b_5 = -4/3$, $b_7 = 1/6$, which also satisfies the second order W-condition. From the equation for f we obtain $b_1 = -5/6$. It remains to fulfill the equation for $f''(f'f,f)$, and further we satisfy the third-order W-condition for $f'f'f$ in order to obtain order 3 when the approximation to the Jacobian is $O(h)$ close to the true Jacobian. This yields $\alpha_{4,2} = 5/4 - \alpha_{4,3} - 1/2\alpha_{7,2} - \alpha_{7,3}$ and $\alpha_{7,4} = -\alpha_{7,5} - \alpha_{7,6} + 2$. We still have some freedom so that we can solve the fifth-order conditions for $f'''(f'f,f,f)$ and $f''(f'f'f,f)$. This gives $\alpha_{7,2} = 2/5 + 4\alpha_{4,3}$ and $\alpha_{7,3} = -2\alpha_{4,3} + 13/20$. Since no other fifth-order conditions can be satisfied, we now minimize

$$\sum_{j=1}^{3} \alpha_{4,j}^2 + \sum_{j=1}^{6} \alpha_{7,j}^2$$

which yields $\alpha_{4,3} = -37/300$ and $\alpha_{7,5} = \alpha_{7,6} = 2/3$.

This construction gives us the following method:

$$\begin{array}{rcl} k_1 &=& \varphi(\tfrac{1}{3}hA)f(y_0) \\ k_2 &=& \varphi(\tfrac{2}{3}hA)f(y_0) \\ k_3 &=& \varphi(hA)f(y_0) \\ w_4 &=& -\tfrac{7}{300}k_1 + \tfrac{97}{150}k_2 - \tfrac{37}{300}k_3 \\ u_4 &=& y_0 + hw_4 \\ d_4 &=& f(u_4) - f(y_0) - hAw_4 \\ k_4 &=& \varphi(\tfrac{1}{3}hA)d_4 \\ k_5 &=& \varphi(\tfrac{2}{3}hA)d_4 \\ k_6 &=& \varphi(hA)d_4 \\ w_7 &=& \tfrac{59}{300}k_1 - \tfrac{7}{75}k_2 + \tfrac{269}{300}k_3 + \tfrac{2}{3}(k_4 + k_5 + k_6) \end{array} \tag{5.8}$$

$$\begin{aligned} u_7 &= y_0 + hw_7 \\ d_7 &= f(u_7) - f(y_0) - hAw_7 \\ k_7 &= \varphi(\tfrac{1}{3}hA)d_7 \\ y_1 &= y_0 + h(k_3 + k_4 - \tfrac{4}{3}k_5 + k_6 + \tfrac{1}{6}k_7) \end{aligned}$$

Again, we have omitted the tilde on $k_4, \ldots, k_7$ as used in (5.5). We summarize the properties of this method in a theorem.

THEOREM 5.1. *The scheme* (5.8) *is of order* 4 *for differential equations* (1.1), *and exact for linear differential equations* (1.2). *It converges of order* 3 *for differential-algebraic equations* (4.1) *and to smooth solutions of singularly perturbed problems* (4.2) *uniformly for $\epsilon \le h^2$. For differential equations* (1.1), *it is of second order when used with inexact Jacobian, and of order* 3 *when the approximation to the Jacobian is $O(h)$ close to the true Jacobian.*

The method satisfies three of the order-5 conditions. The residuals of the other order-5 conditions appear to be rather small, the largest one being 0.1.

Although the scheme (5.8) is a 7-stage method, it requires only three function evaluations. When using Krylov approximations, the computational cost is dominated by computing k_1. As discussed before, the reason is that k_2, k_3, k_5, and k_6 can be computed recursively from (5.1) or the more stable recurrence (6.2) below, and that k_4 to k_7 are typically well approximated in very low dimensional Krylov subspaces, because d_4 and d_7 are usually much smaller in norm than $f(y_0)$. For these reasons, and because of its superior theoretical properties, we prefer (5.8) to a "standard" 3-stage fourth-order scheme of type (3.1)–(3.3).

5.3. Embedded methods. We have constructed two embedded methods with different properties for the scheme (5.8). The first one is of order 3 for differential equations (1.1) and differential-algebraic equations (4.1), and exact for linear equations (1.2). Solving the third-order conditions of Table 3.1 and condition (4.6), and choosing $b_6 = b_7 = 1/2$ gives the embedded scheme

$$\widehat{y}_1 = y_0 + h(k_3 - \tfrac{1}{2}k_4 - \tfrac{2}{3}k_5 + \tfrac{1}{2}k_6 + \tfrac{1}{2}k_7) \ . \tag{5.9}$$

This method does not satisfy the fourth-order conditions, except that for $f'f'f'f$. It is however only of order 1 as a W-method, i.e., when used with inexact Jacobian.

The second embedded method is of order two as a W-method. It is not exact for linear differential equations (1.2), and it does not satisfy the third-order conditions of Table 3.1. It reads

$$\widetilde{y}_1 = y_0 + h(-k_1 + 2k_2 - k_4 + k_7). \tag{5.10}$$

5.4. Dense output. Like for Runge-Kutta and Rosenbrock methods, a continuous numerical solution $y(t_0 + \theta h)$ is defined via

$$y_1(\theta) = y_0 + h\sum_{i=1}^{s} b_i(\theta)k_i$$

with polynomials b_i satisfying $b_i(0) = 0$ and $b_i(1) = b_i$. This approximation is of order p, i. e.,

$$y(t_0 + \theta h) - y_1(\theta) = O(h^{p+1}) \ ,$$

iff

$$\sum_{i=1}^{s} b_i(\theta)\Phi_i(\tau) = \theta^{\rho} P_{\tau}(\gamma)$$

for all elementary differentials τ of order $\rho \leq p$, see [12, p. 452].

For the 3-stage method (5.7) a continuous numerical solution of order 3 is given by

$$\begin{aligned}
b_1(\theta) &= \theta(1-\theta-\tfrac{16}{27}\theta^2), \\
b_2(\theta) &= \theta^2, \\
b_3(\theta) &= \tfrac{16}{27}\theta^3.
\end{aligned}$$

For the 7-stage method (5.8) a continuous numerical solution of order 3, which is also of order 3 for differential-algebraic equations and of order 2 when used with inexact Jacobian, is given by

$$\begin{aligned}
b_1(\theta) &= \theta(1-\tfrac{7}{2}\theta+\tfrac{5}{3}\theta^2), \\
b_2(\theta) &= 2\theta(1-\theta), \\
b_3(\theta) &= \theta^3, \\
b_4(\theta) &= \theta^2(2-\tfrac{4}{3}\theta+\tfrac{1}{3}\theta^2), \\
b_5(\theta) &= -\tfrac{4}{3}\theta^4, \\
b_6(\theta) &= \theta^4, \\
b_7(\theta) &= \theta^2(-\tfrac{1}{2}+\tfrac{2}{3}\theta).
\end{aligned}$$

The actual computation uses

$$y_1(\theta) = y_0 + h\sum_{i=1}^{s} \tilde{b}_i(\theta)\tilde{k}_i$$

where $\tilde{b}_i(\theta)$ are defined as in (5.6) and $\tilde{k}_i$ are the k_i from (5.8).

6. Implementation issues.

6.1. Step size control. The step size control for the scheme (5.8) uses the two embedded methods proposed in Section 5.3. As an estimate for the local error, we choose the minimum of the local error estimates of these two methods. A step size selection strategy due to Gustafsson, see [12, p. 31–35] and the Radau5 code [12, p. 550ff], then yields a new step size proposal h_{err}.

However, if Krylov subspace methods are used to approximate the matrix exponential operator, then in addition to the local error estimate it is necessary to take the work and storage requirements of the Krylov process into account. We propose the following strategy: First choose a suitable "window" $[\mu, M]$ for the number of Krylov steps m required in the approximation of k_1, k_2, k_3 (recall that in reduced methods, the overall work of the Krylov processes is dominated by this first Krylov process). In this window we choose a desirable number of Krylov steps m_{opt}. We preserve the actual step size h of the integration method whenever $m \in [\mu, M]$. If $m > M$ the new

Krylov step size is reduced until the required accuracy is achieved with an $m \in [\mu, M]$. If $m < \mu$ in two consecutive steps, we set

$$h_{kry} = h \left(\frac{m_{opt}}{m} \right)^{\alpha} ,$$

where we have found $\alpha = 1/3$ as a reasonable value in our numerical experiments. It also turned out that a more drastical enlargement of the step size is possible if m is very small for more than two consecutive steps. For example we used

$$h_{kry} = 2^{j-1} h$$

if $m < 4$ in the last j time steps.

Finally we choose the new step size as

$$h_{new} = \min\{h_{err}, h_{kry}\} .$$

6.2. Savings from previous steps. The scheme may reuse the Jacobian of a previous time step as an approximation to the actual Jacobian. This is done if the local error of the embedded method (5.10) is acceptable and in addition $h_{kry} \leq h_{err}$, i.e., the step size is determined by the Krylov process.

Further savings can be achieved if the Jacobian A and the step size h are the same as in the previous time step. We then write

$$k_j(t_n) = k_j(t_{n-1}) + \varphi(j\gamma hA) \left(f(y_n) - f(y_{n-1}) \right) , \qquad j = 1, 2, 3 .$$

If $f(y_n)$ is close to $f(y_{n-1})$, then the initial vector for the Krylov process is small in norm and thus the Krylov process becomes less expensive.

6.3. Stopping criterion for the Krylov method. We need to decide when the Krylov approximation (5.2) is to be considered sufficiently accurate. Since exact errors are inaccessible, the stopping criterion in the iterative solution of linear systems $(\lambda I - \tau A)x = v$ is usually based on the residual

$$r_m(\lambda) = v - (\lambda I - \tau A)x_m$$

instead of the error of the mth iterate

$$e_m(\lambda) = x_m - x .$$

For Galerkin-type methods like FOM and BiCG, the residual vectors can be computed from

$$r_m(\lambda) = \|v\|_2 \cdot \tau h_{m+1,m} \, [(\lambda I - \tau H_m)^{-1}]_{m,1} \cdot v_{m+1} ,$$

where $h_{m+1,m}$ is the $(m+1, m)$ entry of H_{m+1}, and $[\,\cdot\,]_{m,1}$ denotes the $(m, 1)$-entry of a matrix. Using Cauchy's integral formula, the error of the mth Krylov approximation to $\varphi(\tau A)v$ can be written as

$$\epsilon_m = V_m \varphi(\tau H_m) e_1 \cdot \|v\|_2 - \varphi(\tau A)v = \frac{1}{2\pi i} \int_{\Gamma} \varphi(\lambda) \, e_m(\lambda) \, d\lambda ,$$

where Γ is a contour enclosing the eigenvalues of τA and τH_m, cf. [13]. Thus, the error ϵ_m can be interpreted as a linear combination of errors $e_m(\lambda)$ of linear systems. Replacing $e_m(\lambda)$ by $r_m(\lambda)$ in this formula, we get a *generalized residual*

$$\rho_m = \frac{1}{2\pi i} \int_\Gamma \varphi(\lambda)\, r_m(\lambda)\, d\lambda = \|v\|_2 \cdot \tau h_{m+1,m}\, [\varphi(\tau H_m)]_{m,1}\, v_{m+1} \ ,$$

which can be computed at no additional cost. This suggests to use ρ_m instead of the unknown error ϵ_m in the stopping criterion. The use of ρ_m was also proposed by Saad [26], who used a different derivation that is plausible only for small $\|\tau A\|$.

In the scheme (5.8), the Krylov approximations to k_j are multiplied by the step size h. It is therefore reasonable to stop the iteration if

$$h\, \|\rho_m\|_{tol} < 1 \ , \tag{6.1}$$

where $\|\cdot\|_{tol}$ is the weighted norm used in the integrator:

$$\|d\|_{tol} = \left(\frac{1}{N} \sum_{i=1}^{N} (d_i/w_i)^2 \right)^{1/2}$$

with $w_i = \mathrm{atol}_i + \max(|y_{n,i}|, |y_{n-1,i}|) \cdot \mathrm{rtol}$, where atol_i and rtol are the given absolute and relative error tolerances.

In our numerical experiments we found that (6.1) is on the safe side, but sometimes rather pessimistic. Then it may pay off to apply an idea attributed to Shampine in [12, p. 134], which consists in using a smoothed residual $(I - \tau A)^{-1}\rho_m$ instead of the true residual. Since solving a linear system with coefficient matrix $(I - \tau A)$ is prohibitively expensive when A is large, one can perform a smoothing in the m-dimensional subspace and use

$$h \cdot \|v\|_2 \cdot \tau h_{m+1,m}\, |[(I - \tau H_m)^{-1} \varphi(\tau H_m)]_{m,1}| \cdot \|v_{m+1}\|_{tol} < 0.1$$

instead of (6.1) for $m \geq 5$, say. For smaller m, this criterion may be overly optimistic when τA has large norm.

6.4. Computation of $\varphi(\tau H_m)$. To reduce the computational costs, we evaluate $\varphi(\tau H_m)$ only when m figures in an index sequence, e.g., $m \in \{1, 2, 3, 4, 6, 8, 11, 15, 20, 27, 36, 48\}$. This sequence is chosen such that the computation of $\varphi(\tau H_m)$ is about as expensive as the total of the previously computed $\varphi(\tau H_j)$, since the computation of $\varphi(\tau H_m)$ requires $O(m^3)$ arithmetic operations.

If A is Hermitian, then H_m is Hermitian tridiagonal. In this case, one can simply diagonalize H_m.

In the non-Hermitian case, we suggest to use Padé approximation similarly to the third method described in [19] to compute the matrix exponential. Here, the matrix is first scaled by a factor of 2^{-k} such that $\|2^{-k}\tau H_m\| < 1/2$. Then we evaluate the $(6, 6)$ Padé approximation to $\varphi(z)$ for the scaled matrix:

$$\varphi(z) = \frac{1 + \frac{1}{26}z + \frac{5}{156}z^2 + \frac{1}{858}z^3 + \frac{1}{5720}z^4 + \frac{1}{205920}z^5 + \frac{1}{8648640}z^6}{1 - \frac{6}{13}z + \frac{5}{52}z^2 - \frac{5}{429}z^3 + \frac{1}{1144}z^4 - \frac{1}{25740}z^5 + \frac{1}{1235520}z^6} + O(z^{13}) \ .$$

Next, $\varphi(\tau H_m)$ is computed recursively from $\varphi(2^{-k}\tau H_m)$ by applying the following coupled recurrences:

$$\begin{aligned} \varphi(2z) &= \tfrac{1}{2}(e^z+1)\varphi(z) \\ e^{2z} &= e^z e^z \ . \end{aligned} \tag{6.2}$$

This recurrence is stable for all z in the left half-plane, whereas (5.1) becomes unstable for large $|z|$ because of the multiplication with z.

Alternatively, in the non-Hermitian case, one can use a formula due to Saad [26, Section 2.3]:

$$\exp\begin{bmatrix} \tau H_m & e_1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \exp(\tau H_m) & \varphi(\tau H_m)e_1 \\ 0 & 1 \end{bmatrix} .$$

This appears favorable when the dimension m is not too small.

7. Numerical experiments.

We have implemented the method (5.8) with (and without) Krylov approximations in a MATLAB code `exp4`. The program is written in the format used in the MATLAB ODE suite [28], which is available via anonymous ftp on ftp.mathworks.com in the pub/mathworks/toolbox/matlab/funfun directory. The code `exp4` can be obtained from na.uni-tuebingen.de in the pub/codes/exp4 directory. A C version of `exp4` is also available from this ftp site.

7.1. A reaction-diffusion equation with nonstiff chemical reaction: the Brusselator.

To illustrate the behavior of the exponential integrator with Krylov approximations to $\varphi(\gamma hA)v$ in the transition from a nonstiff to a stiff problem, we have chosen the two-dimensional Brusselator [11, pp. 248ff]:

$$\begin{aligned} \frac{\partial u}{\partial t} &= 1 + u^2 v - 4u + \alpha\Delta u \\ \frac{\partial v}{\partial t} &= 3u - u^2 v + \alpha\Delta v \end{aligned}$$

for $0 \le x, y \le 1$ together with Neumann boundary conditions

$$\frac{\partial u}{\partial n} = 0, \qquad \frac{\partial v}{\partial n} = 0$$

and initial conditions

$$u(x,y,0) = 0.5 + y, \qquad v(x,y,0) = 1 + 5x \ .$$

The Laplacian is discretized on a uniform 100 × 100 grid by central differences, so that the dimension of the resulting ODE problem is $20,000$. The eigenvalues of the discretized Laplacian lie between $-80,000$ and zero. We present numerical experiments with three different values of the diffusion coefficient $\alpha = 2\cdot 10^{-4}$, $2\cdot 10^{-3}$, $2\cdot 10^{-2}$, which mark the transition from a nonstiff to a stiff problem. The solution of the problem for $\alpha = 2\cdot 10^{-3}$ is shown in the movie on pp. 250ff in [11].

In Figs. 7.1–7.3 we show work-precision diagrams for our exponential integrator `exp4`, and for the explicit Runge-Kutta integrator `ode45` from the MATLAB ODE suite [28], which is based on a fifth-order method of Dormand and Prince [3]. The vertical axis shows the error at the end point $t = 1$, the horizontal axis gives the

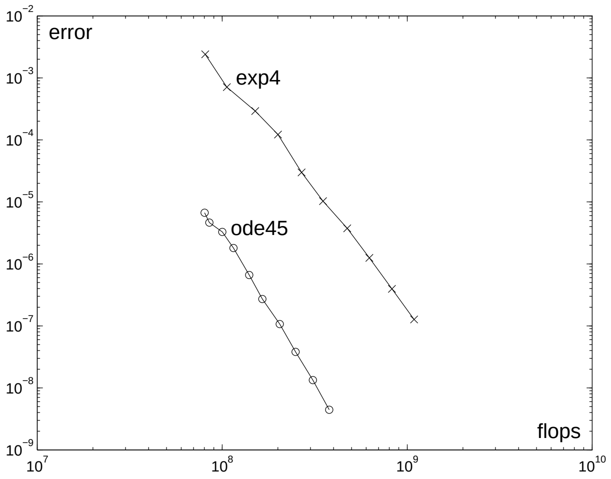

FIG. 7.1. *Brusselator for* $\alpha = 2 \cdot 10^{-4}$

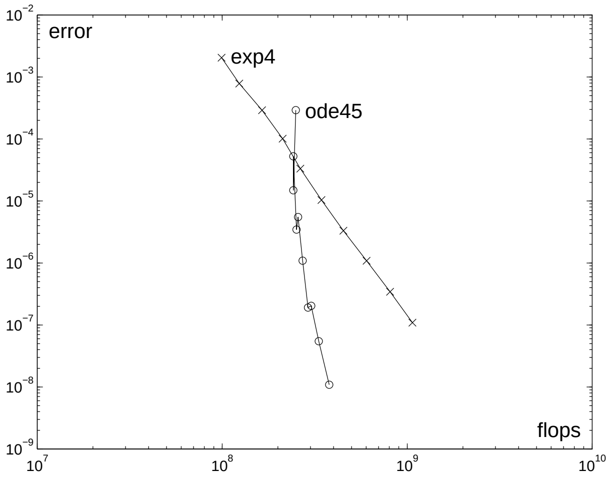

FIG. 7.2. *Brusselator for* $\alpha = 2 \cdot 10^{-3}$

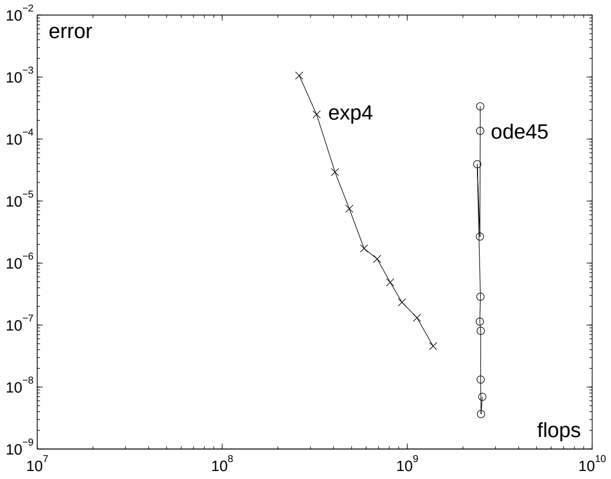

FIG. 7.3. *Brusselator for* $\alpha = 2 \cdot 10^{-2}$

number of flops. The markers × for exp4 and ∘ for ode45 correspond to the error tolerances atol=rtol= 10^{-3}, $10^{-3.5}$, ..., $10^{-7.5}$. While the computational work of the nonstiff integrator increases drastically with growing α, the performance of exp4 is considerably less affected.

7.2. A reaction-diffusion equation with stiff chemistry: the Robertson example. The following example shows the behavior of the exponential integrator for a very stiff problem. We consider the Robertson reaction [12, pp. 3f] with one-dimensional diffusion:

$$\begin{aligned} u_t &= -0.04u + 10^4 vw + \alpha u_{xx} \\ v_t &= 0.04u - 10^4 vw - 3 \cdot 10^7 v^2 + \alpha v_{xx} \\ w_t &= 3 \cdot 10^7 v^2 + \alpha w_{xx} \end{aligned}$$

for $0 \le x \le 1$, $0 \le t \le 400$ together with Neumann boundary conditions $u_x = v_x = w_x = 0$ at $x = 0, 1$, and initial values

$$u(x,0) = 1 + \sin(2\pi x), \quad v(x,0) = w(x,0) = 0 \ .$$

The diffusion coefficient is chosen as $\alpha = 2 \cdot 10^{-2}$. The second spatial derivative is discretized on a uniform grid with 30 grid points. In this problem, the stiffness originates from the reaction terms. We have chosen such a small problem because we intend to illustrate the influence of the Krylov approximation procedure to the performance of the integrator. In Fig. 7.4 we show the step sizes as a function of time in a double logarithmic scale with and without Krylov approximation of $\varphi(\gamma hA)v$. As this example has only dimension 90, $\varphi(\gamma hA)$ can here be computed by diagonalization. For comparison, we also show the step size of the explicit integrator ode45 and the stiff integrator ode15s from the MATLAB ODE suite [28], which uses a variant of a BDF method. All the methods have been run with the same tolerances atol=rtol= 10^{-6}. It is seen that in this example the step size is always limited by the Krylov process. The step size restriction does not appear very severe on the considered time interval. Similar step size sequences are obtained for the Krylov-approximated exponential method for higher-dimensional versions of the problem. However, the limits of the Krylov approach show up when the integration is continued to very long times. There, the step size remains essentially on the level seen at the right-most part of Fig. 7.4. It has been observed that this behavior is largely due to roundoff error effects.

7.3. A Schrödinger equation with time-dependent potential. As an example of a problem whose Jacobian has large imaginary eigenvalues we consider, following [23], the one-dimensional Schrödinger equation for $\psi = \psi(x,t)$

$$i\,\frac{\partial \psi}{\partial t} = H(x,t)\psi$$

with the Hamiltonian

$$H(x,t) = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \kappa\frac{x^2}{2} + \mu \sin^2(t)x \ .$$

This equation models an atom/molecule interacting with a high intensity CW laser. The parameter values used were $\kappa = 10$ and $\mu = 100$. The initial value was $\psi(x,0) =$

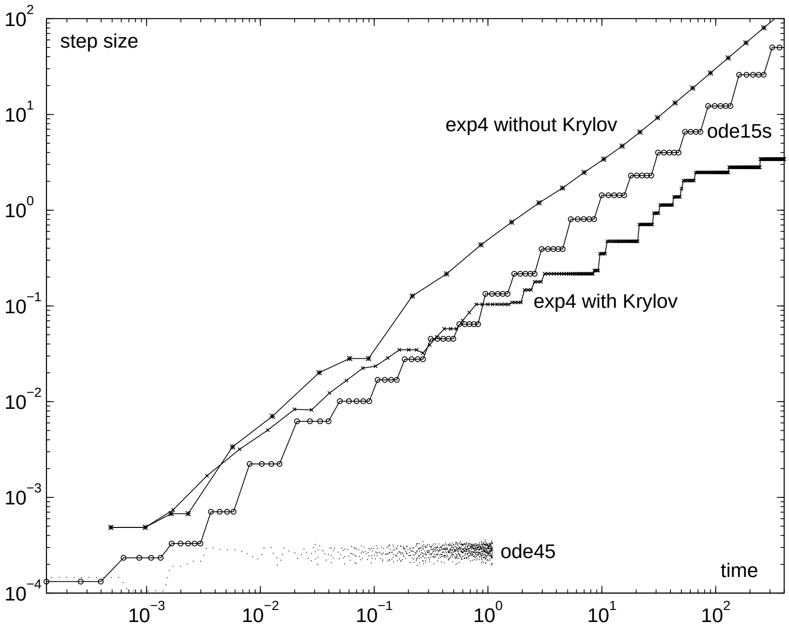

FIG. 7.4. *Step sizes versus time for the Robertson example.*

$e^{-\sqrt{\kappa}x^2/2}$, which corresponds to the eigenstate of the unforced harmonic oscillator to the lowest energy level. Semi-discretization in space is done by a pseudospectral method with $N = 512$ Fourier modes on the space interval $x \in [-a, a]$ for $a = 10$ with periodic boundary conditions. This leads to the non-autonomous linear system of differential equations for $y = (y_1, \dots, y_N)^T$

$$y' = -i\left(-\tfrac{1}{2}\mathcal{F}_N^{-1}D^2\mathcal{F}_N + \operatorname{diag}\left(\kappa\frac{x_j^2}{2} + \mu\sin^2(t)x_j\right)\right)y\ .$$

Here, $y_j(t)$ is an approximation to $\psi(x_j, t)$ at $x_j = -a + j\frac{2a}{N}$, $\mathcal{F}_N$ is the discrete Fourier-transform operator, and

$$D = i\tfrac{\pi}{a}\operatorname{diag}(0, 1, \dots, \tfrac{N}{2} - 1, -\tfrac{N}{2}, -\tfrac{N}{2} + 1, \dots, -1)\ .$$

The Jacobian is full but matrix-vector multiplications are obtained with $O(N \log N)$ flops using FFT. In Fig. 7.5 we show the work-precision diagram at $t = 1$ for `exp4`, for the standard nonstiff and stiff solvers `ode45` and `ode15s` from the MATLAB ODE suite, and for a MATLAB implementation of Hairer and Wanner's [12] Radau5 implicit Runge-Kutta code. The codes were used with tolerances atol=rtol= $10^{-3}, 10^{-4}, \dots, 10^{-8}$. The surprisingly good behavior of the stiff integrators `ode15s` and `radau5` is due to the following matrix-free implementation: In the simplified Newton iterations the Jacobian was approximated by $i/2\mathcal{F}_N^{-1}D^2\mathcal{F}_N$, so that the linear systems could be solved in $O(N \log N)$ operations using FFT. Therefore, the computational cost per time step was essentially the same as for an explicit method. Using the full Jacobian would make the implicit methods completely inefficient for this problem. We note, however, that the performance of the versions with the simplified Jacobian deteriorates when the parameters κ and μ increase.

The exponential code `exp4` is clearly superior to the explicit integrator `ode45`. For accuracy requirements more stringent than 10^{-4}, Fig. 7.5 shows an advantage for

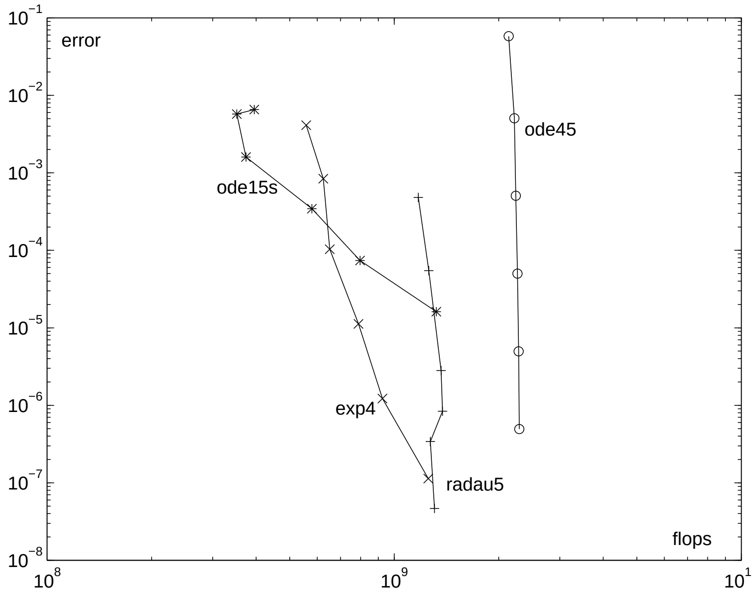

FIG. 7.5. *Work-precision diagram for the Schrödinger equation.*

`exp4` also with respect to the implicit methods in their optimized versions discussed above. This stems from the fact that `exp4` is able to take much larger time steps than the other integrators.

In computations with Schrödinger equations with time-independent Hamiltonian, the use of Chebyshev approximations to the matrix exponential operator is very popular [16]. We therefore also implemented a version of `exp4` where the Arnoldi process is replaced by a Chebyshev approximation. In our numerical experiments the Chebyshev version needed about twice as many flops as the Arnoldi-based implementation.

[We thank S. Gray for pointing out references [16] and [23].]

8. Conclusions and perspectives.

In this paper we have introduced new integration methods which use matrix-vector multiplications with the exponential of the Jacobian. In particular, we have studied Rosenbrock-like exponential methods. Since a straightforward implementation of these methods is computationally expensive, we have identified a subclass of "reduced" methods which are reformulated such that they allow for an efficient implementation. Two promising fourth-order methods, which are exact for linear constant-coefficient problems, have been given in formulas (5.7) and (5.8). The method (5.8), which offers superior properties at slightly higher cost per time step, has been implemented with Krylov subspace approximations to the matrix exponential operator in a code `exp4`. This implementation requires only function evaluations and matrix-vector multiplications with the Jacobian.

Numerical experiments and theoretical considerations indicate that exponential integrators are highly competitive for the following problem classes of large systems of initial-value problems:

Mildly stiff problems (e.g., reaction-convection-diffusion problems with nonstiff reaction terms): The most efficient traditional methods are explicit integrators which are used despite stability restrictions of the step size. For the special case where the eigenvalues of the Jacobian are on the negative real axis, the high-stage Runge-Kutta-Chebyshev methods of van der Houwen and Sommeijer [14, 31] are known to be very successful. Here, the theory in [5, 13] and [31] tells us that the number of necessary matrix-vector multiplications with the Jacobian in Krylov iterations for exponential methods and the number of function evaluations in Runge-Kutta-Chebyshev methods needed to attain stability are both of the magnitude of $\sqrt{h\|A\|}$. However, the Krylov methods take advantage of clustered eigenvalues and of vectors with small components in some eigendirections. There is no restriction to problems with eigenvalues near the real axis for the exponential methods with Krylov approximations, and much larger time steps than with standard explicit Runge-Kutta methods (such as the Dormand-Prince methods) can be taken.

Stiff problems (e.g., reaction-diffusion problems with stiff reaction terms): For high-dimensional systems, the standard approach is to use implicit methods (such as BDF or Radau) where the linear systems are solved iteratively with the help of a hopefully good and cheap preconditioner. If – and only if – an efficient preconditioner is available, those methods are clearly favorable over the exponential methods proposed here, since it is not known how to precondition the iterative computation of the matrix exponential operator. Due to the superlinear error reduction of the Krylov approximations to the matrix exponential, exponential methods are often competitive even without a preconditioner. We hope that future developments will allow to effectively use ideas of preconditioning in the computation of the exponentials and hence further enlarge the range of stiff problems on which exponential methods are efficiently applicable.

Highly oscillatory problems (e.g., wave equations, Schrödinger equations, elastodynamics, and oscillatory electric circuits): Here, the proposed exponential methods are able to resolve high frequencies to the required error tolerance without the severe time step restrictions of standard schemes. Time step restrictions of an often milder type still occur because of nonlinear effects and because of limitations of the iteration number in the Krylov process. The latter are less severe when the eigenvalues of the Jacobian are clustered. The good resolution of high frequencies with exponential methods is in contrast to usual implicit integrators used with large time steps, which either damp high frequencies or map them to one and the same frequency (or nearly so) in the discretization.

It will be interesting to see how the new methods perform in real-life scientific problems.

REFERENCES

[1] J. Carroll, *Sufficient conditions for uniformly second-order convergent schemes for stiff initial-value problems*, Computers Math. Applic. 24 (1992), 105–116.

[2] G. Denk, *A new efficient numerical integration scheme for highly oscillatory electric circuits*, Mathematical modeling and simulation of electrical circuits and semiconductor devices (Oberwolfach, 1992), Internat. Ser. Numer. Math., 117, Birkhäuser, Basel, 1994, 1–15.

[3] J.R. Dormand and P.J. Prince, *A family of embedded Runge-Kutta formulae*, J. Comp. Appl. Math. 6 (1980), 19–26.

[4] V.L. Druskin and L.A. Knizhnerman, *Error bounds in the simple Lanczos procedure for computing functions of symmetric matrices and eigenvalues*, Comput. Maths. Math. Phys. 7 (1991), 20–30.

[5] V.L. Druskin and L.A. Knizhnerman, *Krylov subspace approximations of eigenpairs and matrix functions in exact and computer arithmetic*, Numer. Lin. Alg. Appl. 2 (1995), 205–217.

[6] W.S. Edwards, L.S. Tuckerman, R.A. Friesner, and D.C. Sorensen, *Krylov methods for the incompressible Navier-Stokes equations*, J. Comp. Phys. 110 (1994), 82–102.

[7] A. Friedli, *Verallgemeinerte Runge-Kutta Verfahren zur Lösung steifer Differentialgleichungen*, in Springer Lecture Notes in Mathematics 631, Springer-Verlag, Berlin, 1978, 35–50.

[8] R.A. Friesner, L.S. Tuckerman, B.C. Dornblaser, and T.V. Russo, *A method of exponential propagation of large systems of stiff nonlinear differential equations*, J. Sci. Comp. 4 (1989), 327–354.

[9] E. Gallopoulos and Y. Saad, *Efficient solution of parabolic equations by Krylov approximation methods*, SIAM J. Sci. Stat. Comp. 13 (1992), 1236–1264.

[10] E. Hairer, Ch. Lubich, and M. Roche, *Error of Rosenbrock methods for stiff problems studied via differential algebraic equations*, BIT 29 (1989), 77–90.

[11] E. Hairer, S.P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*, 2nd edition, Springer-Verlag, Berlin, 1993.

[12] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II*, Springer-Verlag, Berlin, 1991.

[13] M. Hochbruck and Ch. Lubich, *On Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., to appear.

[14] P.J. van der Houwen and B.P. Sommeijer, *On the internal stability of explicit m-stage Runge-Kutta methods for large values of m.* ZAMM 60 (1980), 479–485.

[15] P. Kaps and A. Ostermann, *Rosenbrock methods using few LU-decompositions*, IMA J. Numer. Anal. 9 (1989), 15–27.

[16] R. Kosloff, *Propagation methods for quantum molecular dynamics*, Annu. Rev. Phys. Chem. 45 (1994), 145–178.

[17] J.D. Lawson, *Generalized Runge-Kutta processes for stable systems with large Lipschitz constants*, SIAM J. Numer. Anal. 4 (1967), 372–380.

[18] Ch. Lubich and A. Ostermann, *Linearly implicit time discretization of non-linear parabolic equations.* IMA J. Numer. Anal. 15 (1995), 555–583.

[19] C.B. Moler and C.F. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Review 20 (1979), 801–836.

[20] A. Nauts and R.E. Wyatt, *New approach to many-state quantum dynamics: The recursive-residue-generation method*, Phys. Rev. Lett., 51 (1983), 2238–2241.

[21] B. Nour-Omid, *Applications of the Lanczos algorithm*, Computer Physics Comm. 53 (1989), 157–168.

[22] T.J. Park and J.C. Light, *Unitary quantum time evolution by iterative Lanczos reduction*, J. Chem. Phys. 85 (1986), 5870–5876.

[23] U. Peskin, R. Kosloff, and N. Moiseyev, *The solution of the time dependent Schrödinger equation by the (t,t') method: The use of the global polynomial propagators for time dependent Hamiltonians*, J. Chem. Phys. 100 (1994), 8849–8855.

[24] R.P. Ratowsky and J.A. Fleck Jr., *Accurate numerical solution of the Helmholtz equation by iterative Lanczos reduction*, Optic Letters 16 (1991), 787–789.

[25] Y. Saad, *On the Lanczos method for solving symmetric systems with several right hand sides*, Math. Comp. 48 (1987), 651–662.

[26] Y. Saad, *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal. 29 (1992), 209–228.

[27] B.A. Schmitt and R. Weiner, *Matrix-free W-methods using a multiple Arnoldi iteration*, Appl. Numer. Math. 18 (1995), 307–320.

[28] L.F. Shampine and M.W. Reichelt, *The* Matlab *ODE suite*, SIAM J. Sci. Comp. 18 (1997), 1-22.

[29] R.B. Sidje, *Expokit: Software package for computing matrix exponentials*, Submitted to ACM TOMS, 1996.

[30] K. Strehmel and R. Weiner, *Linear-implizite Runge-Kutta-Methoden und ihre Anwendung.* Teubner, Stuttgart–Leipzig, 1992.

[31] J.G. Verwer, *Explicit Runge-Kutta methods for parabolic partial differential equations.* Appl. Numer. Math., to appear, 1996.