Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 203 (2005) 72–88

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

# Generalized integrating factor methods for stiff PDEs ☆

S. Krogstad

*Department of Computer Science, University of Bergen, N-5020, Norway*

Received 5 August 2003; received in revised form 9 August 2004; accepted 9 August 2004
Available online 15 September 2004

**Abstract**

The integrating factor (IF) method for numerical integration of stiff nonlinear PDEs has the disadvantage of producing large error coefficients when the linear term has large norm. We propose a generalization of the IF method, and in particular construct multistep-type methods with several orders of magnitude improved accuracy. We also consider exponential time differencing (ETD) methods, and point out connections with a particular application of the commutator-free Lie group methods. We present a new fourth order ETDRK method with improved accuracy. The methods considered are compared in several numerical examples.
© 2004 Elsevier Inc. All rights reserved.

*Keywords:* Stiff systems; Integrating factor methods; Lie group methods; Exponential time differencing

## 1. Introduction

In this paper we consider high order numerical methods for partial differential equations of the form

$$u_t = \mathscr{L}u + \mathscr{N}(u,t), \tag{1}$$

where $\mathscr{L}$ and $\mathscr{N}$ are linear and nonlinear operators, respectively. Many interesting equations can be put in this form, where typically $\mathscr{L}$ represents the stiff part of the equation. When discretizing the spatial part of (1), one obtains an ODE on $\mathbb{R}^n$ of the form

$$u'(t) = Lu + N(u,t), \tag{2}$$

☆ This work was sponsored by by the Norwegian Research Council under Contract Nos. 127582/432 and 142955/431, through the SYNODE project.

*E-mail addresses:* stein@ii.uib.no, Stein.Krogstad@sintef.no.

where $N : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, and $L$ can be represented by an $n \times n$ matrix. Explicit methods for solving ODEs typically match the first few terms of a Taylor series of the true solution, but for ODEs of the form (2) with large $\|L\|$, such methods lead to prohibitive small time steps or instabilities. Numerous high-order approaches (both explicit and implicit) for solving equations of the form (2) exists, and in a recent paper Kassam and Trefethen [14] compare various fourth order methods, including linear implicit, splitting methods, integrating factor (IF) and exponential time differencing (ETD). They conclude that the best by a clear margin is a modification of the so-called ETDRK4 (exponential time differencing fourth-order Runge–Kutta) method. The ETDRK4 method was developed by Cox and Matthews [4], among general formulas for ETD-schemes of arbitrary order (also called ELP-schemes [1]). In this paper we will discuss the construction of ETDRK-schemes and point out connections with commutator-free Lie group methods. In particular, we construct a fourth order ETDRK-scheme with improved accuracy compared to the one proposed in [4]. We will also study IF related methods in a general setting, and obtain schemes with huge improvements in accuracy compared to the standard approach.

Consider the ODE (2) with initial conditions $u(t_0) = u_0$. A key idea we use for the ETD methods treated in this paper is that if we replace the nonlinear term by an approximating finite polynomial in time, then the resulting ODE can be solved exactly. For starters, assume the nonlinear term is a finite polynomial, that is

$$N(u(t_0+h), t_0+h) = \sum_{k=0}^{m-1} \frac{h^k}{k!} N_k, \tag{3}$$

where $N_k \in \mathbb{R}^n,\ k = 0, 1, \ldots, m-1$. Let exp denote the matrix exponential. By the variation-of-constants formula, the solution of the ODE is given as

$$u(t_0+h) = \exp(hL)u_0 + \int_0^h \exp((h-\tau)L) \sum_{k=0}^{m-1} \frac{\tau^k}{k!} N_k \, \mathrm{d}\tau = \exp(hL)u_0 + \sum_{k=1}^{m} h^k \phi_k(hL) N_{k-1}, \tag{4}$$

where for $k = 1,2,\ldots,m$, we have

$$\phi_k(hL) = \frac{1}{h^k} \int_0^h \frac{\tau^{k-1}}{(k-1)!} \exp((h-\tau)L) \, \mathrm{d}\tau. \tag{5}$$

Integration by parts on (5) gives the the recurrence relation $(hL)\phi_k(hL) = \phi_{k-1}(hL) - \frac{1}{(k-1)!}$ where $\phi_0 := \exp$. Thus the $\phi_k$ are analytic functions which for $x \in \mathbb{C}$ are given

$$\phi_k(x) = \frac{\phi_{k-1}(x) - \frac{1}{(k-1)!}}{x} = \frac{1}{k!} + \frac{x}{(k+1)!} + \frac{x^2}{(k+2)!} + \cdots. \tag{6}$$

As a motivation for using the series (4) rather than an ordinary Taylor series, we make the following observation for the ODE (2) when the nonlinear term only depends on time.

**Lemma 1.1.** *Consider the following ODE on* $\mathbb{R}^n$,

$$u'(t) = Lu(t) + N(t), \quad u(t_0) = u_0, \tag{7}$$

*where* $N : \mathbb{R} \to \mathbb{R}^n$ *is m times differentiable in some sufficiently large neighborhood around* $t = t_0$. *Denote* $N^{(k)}(t) = (\mathrm{d}^k/\mathrm{d}t^k)N(t)$. *Then* $u(t_0 + h)$ *admits the expansion*

$$u(t_0+h) = \exp(hL)u_0 + \sum_{k=1}^{m} h^k \phi_k(hL) N^{(k-1)}(t_0) + R_m(h), \tag{8}$$

where

$$R_m(h) = \int_0^h (h-\tau)^m \phi_m((h-\tau)L) N^{(m)}(t_0+\tau) \, \mathrm{d}\tau. \tag{9}$$

**Proof.** We prove the lemma by integration of the expression for $R_m(h)$. To make this straightforward, we first observe that for $\phi_0 := \exp$ and $k = 1,2,\ldots,m$, we have

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\left(\tau^k\phi_k(\tau L)\right) = \tau^{k-1}\phi_{k-1}(\tau L). \tag{10}$$

Integration by parts on $R_k(h)$ for $k = 1,2,\ldots,m$, then gives the relation

$$R_k(h) = -h^k\phi_k(hL)N^{(k-1)}(t_0) + R_{k-1}(h), \tag{11}$$

where by the variation-of-constants formula

$$R_0(h) = \int_0^h \exp((h-\tau)L)N(t_0+\tau)\,\mathrm{d}\tau = -\exp(hL)u_0 + u(t_0+h). \tag{12}$$

Thus we have

$$R_m(h) = -\sum_{k=1}^{m} h^k\phi_k(hL)N^{(k-1)}(t_0) - \exp(hL)u_0 + u(t_0+h), \tag{13}$$

which proves the lemma. □

We see that the the remainder term $R_m$ differs from a standard Taylor series of the nonlinear term by the appearance of the $\phi_m$ function inside the integral rather than $1/m!$ in front. Since we are assuming that the stiffness lies in the linear part, we give a simple bound on $\max_{0\leqslant\tau\leqslant h}\|\phi_m(\tau L)\|$. A standard bound for the matrix exponential is given by the logarithmic norm: Let $\|\cdot\|$ denote the Euclidean vector norm and also the induced norm on $\mathbb{R}^{n\times n}$. Recall that the *logarithmic norm* $\mu(A)$ (with respect to the Euclidean norm) of a matrix $A \in \mathbb{R}^{n\times n}$ is given

$$\mu(A) = \sup_{y\in\mathbb{R}^n}\frac{y^{\mathrm{T}}Ay}{y^{\mathrm{T}}y} = \lambda_{\max}\left(\frac{A+A^{\mathrm{T}}}{2}\right), \tag{14}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of the given symmetric matrix. Note that the logarithmic norm can take negative values, and is thus not a norm in the traditional sense. Consider the curve $y(t) = \exp(tL)y_0$. For the derivative of its norm we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\|y(t)\| = \frac{y(t)^{\mathrm{T}}Ly(t)}{\|y(t)\|} \leqslant \mu(L)\|y(t)\|, \tag{15}$$

and thus it follows that $\|y(t)\| \leqslant \mathrm{e}^{t\mu(L)}\|y_0\|$. Since this holds for every $y_0$ we have

$$\|\exp(tL)\| \leqslant \mathrm{e}^{t\mu(L)} \quad \text{for } t \geqslant 0. \tag{16}$$

Using this bound it now follows from (5) that

$$\begin{aligned}\max_{0\leqslant\tau\leqslant h}\|\phi_m(\tau L)\| &\leqslant \frac{1}{m!}\max_{0\leqslant\tau\leqslant h}\|\exp(\tau L)\| \\ &\leqslant \frac{1}{m!}\max\left(1, \mathrm{e}^{h\mu(L)}\right).\end{aligned} \tag{17}$$

Although this bound is not optimal with respect to sharpness, it follows that if the spectrum of the symmetric part of $L$ is close to the left half of the complex plane, then the rate of convergence of the series in Lemma 1.1 is comparable to the standard Taylor series expansion of the nonlinear term, and in any case independent of $\|L\|$. The main idea we use for ETD schemes in this paper is to replace the nonlinear term in (2) by a finite polynomial approximation, and then solve the resulting ODE exactly. It follows from Lemma

1.1 that the expression for the error of such a method will include, in addition to derivatives of the nonlinear term, only functions of $L$ with small norm, and thus one would anticipate accurate methods.

As an illustration of the $\phi_k$ functions, we have in Fig. 1 to the left plotted the image of $\mathbb{C}^-$ under the maps exp, $\phi_1$, $\phi_2$ and $\phi_3$. On the right, the equivalent images of the region

$$D_2 = \{z \in \mathbb{C} | \mathrm{Re}(z) \leqslant 1\} \tag{18}$$

is shown and indicates that the bounds $|\phi_k(z)| \leqslant (1/k!)\mathrm{e}^{\mathrm{Re}(z)}$ are not optimal with respect to sharpness.

## 2. IF, ETD and connections with commutator-free Lie group methods

In this section we introduce IF methods, ETD methods, and point out connections with a particular application of the commutator-free Lie group methods.

### 2.1. Integrating factor

Consider the ODE (2) with $u(t_0) = u_0$. The integrating factor method [2,4,7,22] uses a smooth change of variables in attempt to ameliorate the stiff part of the original equation. It can be described as follows: Search for a $v : \mathbb{R} \to \mathbb{R}^n$ such that for $u(t)$ around $t = t_0$ we have

$$u(t_0 + \tau) = \exp(\tau L)v(\tau). \tag{19}$$

Differentiating this expression and substituting (2) we obtain the following ODE for $v$:

$$v'(\tau) = \exp(-\tau L)N(\exp(\tau L)v, t_0 + \tau), \quad v(0) = u_0. \tag{20}$$

The idea is now to solve Eq. (20) with some explicit numerical method to obtain $v_1 \approx v(h)$, and then obtain the approximate solution $u_1$ from (19). At first glance this transformation may not seem as good idea as the ODE (20) seen isolated can be even more stiff than the original equation due to the minus in the exponential. However, all minuses cancel when the approximate solution is composed with (19) and thus this is not a problem (see [22] for examples on implementation).

On the downside, there are mainly two problems with the IF methods described in the literature [2,4]: it has large error coefficients for $\|L\| \gg 1$, and it does not preserve fixed points of the original ODE. The latter means that if there exists an $u_0$ such that

$$Lu_0 + N(u_0) = 0, \tag{21}$$

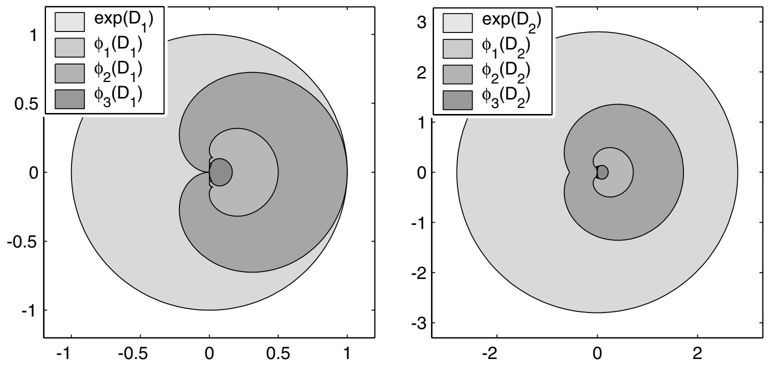

Fig. 1. The image of the functions exp, $\phi_1$, $\phi_2$ and $\phi_3$ of $D_1 = \mathbb{C}^-$ (left) and $D_2 = \{z \in \mathbb{C} | \mathrm{Re}(z) \leqslant 1\}$ (right).

then IF methods will in general move away from $u_0$. Although we have not run into problems due to lack of preserving the fixed points of the true solution, it seems desirable for a numerical method to fulfill this property with respect to capturing as much of the dynamics of the system as possible. In this paper we suggest generalizations of the IF approach, and construct methods where both of these problems are taken care of.

## 2.2. Explicit ETD-methods

Consider the ODE (2). Let $t_j = t_0 + jh$ denote time discretization points and suppose we have numerical approximations $u_n, u_{n-1}, \ldots, u_{n-k+1}$ at the corresponding times. Moreover let $p(\tau) = \sum_{j=0}^{k-1} (\tau^j/j!) p_j$ be the interpolation polynomial through the set of points $\{(t_j, N(u_j, t_j)) : j = n-k+1, \ldots, n\}$ with $p(0) = N(u_n, t_n)$. Given that the approximations $u_j$ are of order $\geqslant k$, the polynomial $p(\tau)$ is an approximation to $N(u(t_n + \tau), t_n + \tau)$ of order $\geqslant k-1$. Using the series of Lemma 1.1 it then follows that

$$u_{n+1} = \exp(hL)u_n + \sum_{j=1}^{k} h^j \phi_j(hL) p_{j-1} \tag{22}$$

defines a $k$th order multistep method for Eq. (2). These are the so-called explicit ETD methods ([1,4,14] and references therein). Similarly to the explicit Adams methods [8] (note that these ETD methods become the Adams methods when the linear part is zero), the $k$th order ETD method (22) can be formulated explicitly as done by Cox and Matthews [4], and earlier by Nørsett [20]:

$$u_{n+1} = \exp(hL)u_n + h \sum_{j=0}^{k-1} g_j(hL) \nabla^j N_n, \tag{23}$$

where $\nabla^j N_n$ denotes the backward differences

$$\nabla^0 N_i = N(u_i, t_i), \quad \nabla^{j+1} N_i = \nabla^j N_i - \nabla^j N_{i-1}, \tag{24}$$

and the functions $g_j$ are given by the recurrence formula

$$x g_0(x) = e^x - 1, \tag{25}$$

$$x g_{j+1}(x) + 1 = g_j(x) + \frac{1}{2} g_{j-1} + \frac{1}{3} g_{j-1} + \cdots + \frac{1}{j+1} g_0(x). \tag{26}$$

In addition to this formula the authors of [4] also derive ETD methods with Runge–Kutta time stepping, the so-called ETDRK methods. Before we discuss these methods we introduce a somewhat related class of methods developed in [3], the *commutator-free Lie group methods*.

## 2.3. Lie group methods and ETDRK

Lie group methods [13] were originally designed for differential equations on manifolds in such a manner that also the numerical solution would stay on the manifold. However, in this paper, the manifold is just $\mathbb{R}^n$, so *staying on the manifold* is not an issue. However, one can benefit from the diversity of these methods with respect to choosing an action which in some sense resembles the true flow of the problem. In standard explicit methods the basic action is just to move along the tangent of our current point, i.e. along a straight line. In Lie group methods, however, one can choose actions based on general flows, and for stiff problems this can be utilized to take into account for the rapid changes in the vector field.

For ease of exposition, we will be dealing with matrix Lie groups and matrix Lie algebras only. Let $\mathfrak{g} \subset \mathbb{R}^{n \times n}$ denote a matrix Lie algebra, i.e. a set of matrices closed under linear combinations and matrix commutators,

$$[A,B] = AB - BA. \tag{27}$$

We denote by $G$ the matrix Lie group corresponding to $\mathfrak{g}$, which can be obtained by taking matrix exponentials of elements in $\mathfrak{g}$, and products of these. This group is closed under matrix products and matrix inversion. Let $\mathcal{M}$ be a manifold, and $\cdot : G \times \mathcal{M} \to \mathcal{M}$ an *action* of $G$ on $\mathcal{M}$, i.e.

$$g_2 \cdot (g_1 \cdot y) = (g_2 g_1) \cdot y \quad \text{for all } g_1, g_2 \in G,\ y \in \mathcal{M}. \tag{28}$$

In applications, the action is usually (at least in some representation) just a matrix–matrix or matrix–vector product. A Lie group action on $\mathcal{M}$ induces a product $\cdot : \mathfrak{g} \times \mathcal{M} \to T\mathcal{M}$ ($T\mathcal{M}$ denotes the tangent bundle of $\mathcal{M}$) by

$$A \cdot y = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \exp(tA) \cdot y, \tag{29}$$

again usually matrix–matrix or matrix–vector multiplication in some representation. The main assumption is that we can put our ODE in the form

$$y'(t) = A(y(t), t) \cdot y, \quad y(0) = y_0, \tag{30}$$

where $y(t) \in \mathcal{M},\ A : \mathcal{M} \times \mathbb{R} \to \mathfrak{g}$ and $\mathfrak{g}$ is the Lie algebra corresponding to the Lie group $G$ acting on $\mathcal{M}$.

In the Runge–Kutta Munthe–Kaas (RKMK) methods [18], the idea is to search for a $Z : \mathbb{R} \to \mathfrak{g}$, such that the solution of (30) can be written $y(t) = \exp(Z(t)) \cdot y_0$. Differentiating this expression leads to an ODE for $Z$ of the form

$$Z'(t) = \mathrm{dexp}^{-1}_{Z(t)}(A(\exp(Z(t)) \cdot y_0, t)), \quad Z(0) = 0. \tag{31}$$

The operator $\mathrm{dexp}^{-1}_{Z(t)}(\cdot) : \mathfrak{g} \to \mathfrak{g}$ is an infinite sum of iterated commutators [13], and the usual approach is to truncate it to desired order. Since the Lie algebra is a linear space, a numerical solution $Z_1$ to (31) obtained by a standard Runge–Kutta method, will reside in the Lie algebra. The numerical solution in $\mathfrak{g}$ is then mapped via the Lie group to a numerical solution on $\mathcal{M}$.

A similar approach to solving equations on manifolds is the Crouch–Grossman methods [5,19] and their generalizations, the *commutator-free Lie group methods* suggested by Celledoni et al. [3]. These methods do not transform the ODE to the Lie algebra, but rather compose several exponentials. The general algorithm for an explicit commutator-free Lie group method in the matrix setting can be given as follows:

**Algorithm 2.1.** (*Commutator-free Lie group method*)

$$\left.\begin{aligned} u^{(r)} &= \exp\left(\sum_{k=1}^{r-1} \alpha^k_{rJ} A^{(k)}\right) \cdots \exp\left(\sum_{k=1}^{r-1} \alpha^k_{r1} A^{(k)}\right) \cdot u_n \\ A^{(r)} &= hA(u^{(r)}, t_n + c_r h) \end{aligned}\right\} \quad r = 1, 2, \ldots, s, \tag{32}$$

$$u_{n+1} = \exp\left(\sum_{k=1}^{s} \beta^k_J A^{(k)}\right) \cdots \exp\left(\sum_{k=1}^{s} \beta^k_1 A^{(k)}\right) \cdot u_n \tag{33}$$

In the algorithm the $\alpha^k_{ri}$, the $\beta^k_i$ and the $c_r$ are real coefficients and $s$ is the number of stages. The coefficients of the underlying Runge–Kutta method are given by $a_{rk} = \sum_i \alpha^k_{ri}$ and $b_k = \sum_i \beta^k_i$. Due to the extra freedom in the commutator-free Lie group methods (summing elements in the Lie algebra) compared to the Crouch–Grossman methods, the authors in [3] were able to construct a fourth order method effectively using only five exponentials. In comparison a fourth order Crouch–Grossman method requires 15 [19]. In particular, we consider the fourth order method (CF4) given in [3] with the following *Butcher-like* tableau:

$$\begin{array}{r|llll} c_1=\gamma_{11} & & & & \\ c_2=\gamma_{21} & \alpha_{21}^1 & & & \\ c_3=\gamma_{31} & \alpha_{31}^1 & \alpha_{31}^2 & & \\ c_4\left\{\gamma_{41}\right. & \alpha_{41}^1 & \alpha_{41}^2 & \alpha_{41}^3 & \\ \gamma_{42} & \alpha_{42}^1 & \alpha_{42}^2 & \alpha_{42}^3 & \\ \hline \gamma_1 & \beta_1^1 & \beta_1^2 & \beta_1^3 & \beta_1^4 \\ \gamma_2 & \beta_2^1 & \beta_2^2 & \beta_2^3 & \beta_2^4 \end{array} \quad = \quad \begin{array}{r|rrrr} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1\left\{\frac{1}{2}\right. & \frac{1}{2} & 0 & 0 & \\ \frac{1}{2} & -\frac{1}{2} & 0 & 1 & \\ \hline \frac{1}{2} & \frac{1}{4} & \frac{1}{6} & \frac{1}{6} & -\frac{1}{12} \\ \frac{1}{2} & -\frac{1}{12} & \frac{1}{6} & \frac{1}{6} & \frac{1}{4} \end{array} \tag{34}$$

In addition to the representation of these methods in [3], we have included the $\gamma$ coefficients (the row sums). This is of help in representing general ETDRK methods.

In [18] Munthe-Kaas suggested to use Lie group methods to solve stiff equations by using the affine action on $\mathbb{R}^n$,

$$(M,b)\cdot u = Mu + b, \tag{35}$$

where $M \in \mathbb{R}^{n\times n}$ is invertible and $b \in \mathbb{R}^n$. Pairs of the type $(M,b)$ indeed form a Lie group. To make it clear that this group and its action can be represented as a matrix group and matrix–vector multiplication, respectively, consider the following matrix group $G \subset \mathbb{R}^{(n+1)\times(n+1)}$,

$$G = \left\{ \begin{pmatrix} M & b \\ \mathbf{0}^{\mathrm{T}} & 1 \end{pmatrix} : \; M \in \mathbb{R}^{n\times n} \text{ with } \det(M) \neq 0, b \in \mathbb{R}^n \right\}, \tag{36}$$

where $\mathbf{0}$ denotes the $n \times 1$ zero vector. Representing a point $u \in \mathbb{R}^n$ by a point $y \in \mathbb{R}^{n+1}$ such that $y^{\mathrm{T}} = (u^{\mathrm{T}}\ 1)$, it is clear that for $A \in G$, the multiplication $Ay$ gives the desired affine action (35). The matrix Lie algebra $\mathfrak{g}$ corresponding to $G$ is simply given

$$\mathfrak{g} = \left\{ \begin{pmatrix} M & b \\ \mathbf{0}^{\mathrm{T}} & 0 \end{pmatrix} : M \in \mathbb{R}^{n\times n} \text{ (arbitrary)}, \; b \in \mathbb{R}^n \right\}. \tag{37}$$

Similarly to the action, a tangent vector to $u$ may be represented by an $A \in \mathfrak{g}$ by $Ay$, where $y^{\mathrm{T}} = (u^{\mathrm{T}}\ 1)$. For ease of notation we write both elements of the Lie group and Lie algebra as pairs $(M,b)$. In this notation exp: $\mathfrak{g} \to G$ is given

$$\exp(M,b) = (\exp(M), \phi_1(M)b). \tag{38}$$

This can either be seen from the first two terms of the series of Lemma 1.1, or by computing the matrix exponential of a matrix of the form (37).

Motivated by exponential integrators for stiff systems [11], the suggestion in [18] was to represent a stiff ODE $u'(t) = f(u,t)$ as

$$u'(t) = (J, f(u,t) - Ju)\cdot u, \quad u(0) = u_0, \tag{39}$$

and solve it using a Lie group integrator. Here $J$ denotes some approximation to $Df(u)$, where $Df$ is the Jacobian of $f$. In this paper we simply take $J$ to be the linear term, and thus we represent the ODE (2) as

$$u'(t) = (L, N(u,t))\cdot u, \quad u(0) = u_0. \tag{40}$$

In this case a sufficient Lie algebra is the set of pairs $(\alpha L,b)$, where $\alpha \in \mathbb{R}$ and $b \in \mathbb{R}^n$. The corresponding Lie group consists of pairs of the form $(\exp(\alpha L),b)$. The RKMK method based on forward Euler applied to (40) is now given

$$u_{n+1} = \exp(h(L, N(u_n, t_n))) \cdot u_n = \exp(hL)u_n + h\phi_1(hL)N(u_n, t_n), \tag{41}$$

which coincides with ETD1. For RKMK methods of order $\geqslant 3$, one needs to evaluate commutators of elements of the Lie algebra, which in this setting involves products of the type $Lb$. Since $L$ represents the stiff term, this is not desirable since one typically has $\|L\|$ large compared to $\|\phi_k(L)\|$, $k = 0,1$. Numerical experiments [16,21,15] also shows that RKMK methods in this form is not well suited for stiff equations, unless the $\mathrm{dexp}^{-1}$ function is computed with caution. To avoid commutators, one option is thus to use the commutator-free Lie group methods of Celledoni et al. as presented in one of their examples in [3]. To see the resemblance with the ETDRK4 method of Cox and Matthews [4], we explicitly write out the algorithm of CF4 applied to Eq. (40).

**Algorithm 2.2.** (*CF*4)

$$\begin{aligned}
&u^{(1)} = u_n, && N^{(1)} = N(u^{(1)}, t_0),\\
&u^{(2)} = \exp(\tfrac{h}{2}L)u_n + \tfrac{h}{2}\phi_1(\tfrac{h}{2}L)N^{(1)}, && N^{(2)} = N(u^{(2)}, t_0 + \tfrac{h}{2}),\\
&u^{(3)} = \exp(\tfrac{h}{2}L)u_n + \tfrac{h}{2}\phi_1(\tfrac{h}{2}L)N^{(2)}, && N^{(3)} = N(u^{(3)}, t_0 + \tfrac{h}{2}),\\
&u^{(4)} = \exp(\tfrac{h}{2}L)u^{(2)} + h\phi_1(\tfrac{h}{2}L)(-\tfrac{1}{2}N^{(1)} + N^{(3)}), && N^{(4)} = N(u^{(4)}, t_0 + h),
\end{aligned} \tag{42}$$

$$\begin{aligned}
\hat{u} &= \exp(\tfrac{h}{2}L)u_n + h\phi_1(\tfrac{h}{2}L)(\tfrac{1}{4}N^{(1)} + \tfrac{1}{6}N^{(2)} + \tfrac{1}{6}N^{(3)} - \tfrac{1}{12}N^{(4)}),\\
u_{n+1} &= \exp(\tfrac{h}{2}L)\hat{u} + h\phi_1(\tfrac{h}{2}L)(-\tfrac{1}{12}N^{(1)} + \tfrac{1}{6}N^{(2)} + \tfrac{1}{6}N^{(3)} + \tfrac{1}{4}N^{(4)}).
\end{aligned} \tag{43}$$

Note that the exponential in the first substage in the fourth internal stage has already been computed in the second internal stage ($u^{(2)}$), which simplifies the algorithm. It is interesting to see that the internal stages (42) are exactly the same as the internal stages of the method ETDRK4 by Cox and Matthews. We have found that the ETDRK4 method can be obtained from (43) by considering a continuous extension of the contribution from the nonlinear part. We start by introducing the auxiliary differential equation

$$v'(t) = N(u(t), t), \quad v(t_0) = u_0, \tag{44}$$

where $u(t)$ is the solution of the original ODE (2). An approximation $v_1 \approx v(t_0 + h)$ is obtained by setting $L$ to zero in the final stage (43), that is (keeping $N^{(i)}, i = 1, 2, 3, 4$ fixed)

$$v_1 = u_0 + h\left(\frac{1}{6}N^{(1)} + \frac{1}{3}N^{(2)} + \frac{1}{3}N^{(3)} + \frac{1}{6}N^{(4)}\right). \tag{45}$$

Note that the coefficients appearing in this expression are the $b_i$-values of the underlying classical Runge–Kutta method. In order to find a polynomial approximation for $v(t)$ (and thus for $N(u(t),t)$), we use the techniques of *continuous Runge–Kutta methods* [8]. That is, we wish to find a continuous approximate solution $\tilde{v}(t_0 + \theta h)$ to (44), such that $\tilde{v}(t_0 + h) = v_1$ and for $0 < \theta < 1$ we want $\tilde{v}(t_0 + \theta h)$ to be an approximation to $v(t_0 + \theta h)$ of as high order as possible. We represent the approximation as $\tilde{v}(t_0 + \theta h) = u_0 + h\sum_{i=1}^{4} b_i(\theta)N^{(i)}$, where the $N^{(i)}$ are the (fixed) vectors from (42). The polynomials $b_i(\theta)$ can be found to satisfy the Runge–Kutta order conditions of third order:

$$\begin{aligned}
&b_1 + b_2 + b_3 + b_4 = \theta, && \tfrac{1}{2}b_2 + \tfrac{1}{2}b_3 + b_4 = \tfrac{\theta^2}{2},\\
&\tfrac{1}{4}b_2 + \tfrac{1}{4}b_3 + b_4 = \tfrac{\theta^3}{3}, && \tfrac{1}{4}b_3 + \tfrac{1}{2}b_4 = \tfrac{\theta^3}{6}.
\end{aligned} \tag{46}$$

The solution of this system is given by $b_1(\theta) = (2/3)\theta^3 - (3/2)\theta^2 + \theta$, $b_2(\theta) = b_3(\theta) = -(2/3)\theta^3 + \theta^2$ and $b_4(\theta) = (2/3)\theta^3 - (1/2)\theta^2$. This suggest the following approximation for $N(u(t_0 + \tau), t_0 + \tau)$:

$$N(u(t_0+\tau),t_0+\tau) \approx \frac{\mathrm{d}}{\mathrm{d}\tau}\tilde{v}(t_0+\tau) = \sum_{i=1}^{4} b_i'\left(\frac{\tau}{h}\right)N^{(i)}. \tag{47}$$

Having a polynomial approximation for the nonlinear part, we are by Lemma 1.1 able to solve the following ODE exactly:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\tilde{u}(t_0+\tau) = L\tilde{u}(t_0+\tau) + \sum_{i=1}^{4} b_i'(\frac{\tau}{h})N^{(i)}, \quad u(t_0) = u_0. \tag{48}$$

Its solution at $\tau = h$ is given by

$$u_1 = \exp(hL)u_0 + h\big((4\phi_3 - 3\phi_2 + \phi_1)N^{(1)} + (-4\phi_3 + 2\phi_2)(N^{(2)} + N^{(3)}) + (4\phi_3 - \phi_2)N^{(4)}\big), \tag{49}$$

where $\phi_i$ is short for $\phi_i(hL)$. This is exactly the method ETDRK4 given in [4]. We propose to represent this method in the Butcher-like tableau

$$\begin{array}{rc|cccc}
 & 0 & & & & \\
 & \frac{1}{2} & \frac{1}{2}\phi_1 & & & \\
 & \frac{1}{2} & 0 & \frac{1}{2}\phi_1 & & \\
\multirow{2}{*}{$1\left\{\begin{array}{c}\\ \\ \end{array}\right.$} & \frac{1}{2} & \frac{1}{2}\phi_1 & 0 & 0 & \\
 & \frac{1}{2} & -\frac{1}{2}\phi_1 & 0 & \phi_1 & \\
\hline
 & & 4\phi_3 - 3\phi_2 + \phi_1 & -4\phi_3 + 2\phi_2 & -4\phi_3 + 2\phi_2 & 4\phi_3 - \phi_2
\end{array} \tag{50}$$

We do not claim that this method of construction of the ETDRK-schemes from the commutator-free Lie group methods always gives the right order, in fact one would think that since the continuous approximation is only of order three, the resulting method would also be of order three; however, all the third order cases we tested turned out to give the desired order. Cox and Matthews describe that their first attempt to construct a fourth order ETDRK method in the same fashion as their lower order methods failed. This is illustrated by the need for including one extra substage in the internal stages for the CF4-method to obtain a fourth order method. However, we found that by including one extra term in the series of Lemma 1.1 (that is the $\phi_2$ function), we were able to construct a fourth order ETDRK method where none of the internal stages are divided into substages. We represent the method as follows:

$$\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1 & & & \\
\frac{1}{2} & \frac{1}{2}\phi_1 - \phi_2 & \phi_2 & & \\
1 & \phi_1 - 2\phi_2 & 0 & 2\phi_2 & \\
\hline
 & 4\phi_3 - 3\phi_2 + \phi_1 & -4\phi_3 + 2\phi_2 & -4\phi_3 + 2\phi_2 & 4\phi_3 - \phi_2
\end{array} \tag{51}$$

In fact this method is more accurate than ETDRK4, and it also seems to have slightly better stability properties. We denote this method by ETDRK4-B.

## 3. A generalized IF approach

Given the ODE

$$u'(t) = F(u,t) = Lu + N(u,t), \quad u(t_0) = u_0. \tag{52}$$

The standard integrating factor method uses the fact that we can solve a simpler ODE exactly, and that this simpler ODE has more or less the same stiffness properties as the original equation (52). Generalizing this approach the idea is to find some modified vector field $\tilde{F}$ with $\tilde{F}(u_0, 0) = F(u_0, t_0)$, which approximates $F$ around $u_0$ and hopefully captures key features of $F$. In addition we require that we can solve the ODE

$$\tilde{u}'(\tau) = \tilde{F}(\tilde{u}(\tau), \tau), \quad \tilde{u}(0) = \tilde{u}_0, \tag{53}$$

exactly or easily to a given order of accuracy. Let $\phi_{\tau,\tilde{F}} : \mathbb{R}^n \to \mathbb{R}^n$ denote the flow operator, i.e. $\tilde{u}(\tau) = \phi_{\tau,\tilde{F}}(\tilde{u}_0)$ is the solution of (53). Furthermore, let $D\phi_{\tau,\tilde{F}}$ denote its Jacobian. We wish to represent the solution of (52) locally by a $v(\tau)$ with $v(0) = u_0$ such that

$$u(t_0 + \tau) = \phi_{\tau,\tilde{F}}(v(\tau)). \tag{54}$$

Differentiating this relation with respect to $\tau$ we obtain

$$F(u, t_0 + \tau) = \tilde{F}(\phi_{\tau,\tilde{F}}(v), \tau) + D\phi_{\tau,\tilde{F}}(v)v'(\tau), \tag{55}$$

and thus

$$v'(\tau) = \left(D\phi_{\tau,\tilde{F}}(v)\right)^{-1}(F(\phi_{\tau,\tilde{F}}(v), t_0 + \tau) - \tilde{F}(\phi_{\tau,\tilde{F}}(v), \tau)), \quad v(0) = u_0. \tag{56}$$

It is easily seen that by choosing $\tilde{F}$ such that $\tilde{F}(u) = Lu$ one obtains the standard IF-method. More generally one can consider polynomial approximations to the nonlinear term around some point $u_0$. That is to find some finite polynomial $c(\tau)$ such that

$$c(\tau) \approx N(u(t_0 + \tau), t_0 + \tau). \tag{57}$$

The flow of the vector field $\tilde{F}(u, \tau) = Lu + c(\tau)$ is obtained from the series of Lemma 1.1, and by observing that $D\phi_{\tau,\tilde{F}}(v) = \exp(\tau L)$, the equation for $v$ in (56) becomes simply

$$v'(\tau) = \exp(-\tau L)(N(u(t_0 + \tau), t_0 + \tau) - c(\tau)), \quad v(0) = u_0, \tag{58}$$

with $u(t_0 + \tau) = \phi_{\tau,\tilde{F}}(v(\tau))$.

Assume we have approximate solutions $u_1,\ldots,u_n$ at the corresponding times $t_j = t_0 + jh$. The polynomial $c(\tau)$ approximating the nonlinear term around $u_n$ can be chosen as the interpolating polynomial through the points $(N(u_j,t_j),t_j)$, $j = n - k + 1,\ldots,n$ for some $k \leqslant n$. Denoting $N_j = N(u_j,t_j)$, the first few possible choices for $c$ are

$$\begin{aligned} c_0(\tau) &= N_n, \\ c_1(\tau) &= N_n + \tau\left(\frac{N_n - N_{n-1}}{h}\right), \\ c_2(\tau) &= N_n + \tau\left(\frac{\frac{1}{2}N_{n-2} - 2N_{n-1} + \frac{3}{2}N_n}{h}\right) + \frac{\tau^2}{2}\left(\frac{N_{n-2} - 2N_{n-1} + N_n}{h^2}\right). \end{aligned} \tag{59}$$

The approach is now the same as in the standard IF-method. Use an explicit method for the ODE (58) to obtain $v_1 \approx v(h)$, and then set $u_{n+1} = \phi_{h,\tilde{F}}(v_1)$. Also in this case minuses in the exponential cancel, and do not cause problems. As we shall se in the numerical experiments, increasing the order of the interpolating polynomial $c(\tau)$ leads to increasing accuracy in the resulting methods. However, there seems to be a payoff with decreasing stability, and we will conclude that the generalized IF-methods are best suited for equations where the spectrum of the linear operator lies on or close to the real negative axis. It is worth noting that these methods indeed preserve the fixed points of the original equation. The approach we have proposed here is very similar to the one of Maday et al. [17] all though they do not discuss our particular setting. Their approach is more general in the sense that they compute the action of $(D\phi_{\tau,\tilde{F}}(v))^{-1}$ by a numerical method rather than exact. We adopt a similar notation as in [17] for the labeling of our methods, i.e.

the three methods corresponding to $c_0$, $c_1$ and $c_2$ in (59), we call ETD1/RK4, ETD2/RK4 and ETD3/RK4, respectively. This notation is justified by the fact that the flow of the vector field $\tilde{F} = Lu + c_{k-1}(\tau)$ is equal to the ETD$k$-method for $\tau = h$, and that the transformed equation (58) is integratied by RK4.

As an example we present the overall algorithm for the method resulting from the interpolating polynomial $c_1(\tau)$ in (59) using the classical fourth order Runge–Kutta method on (58). Note that the flow of the vector field $\tilde{F}(v) = Lv + c_1(\tau)$ is given

$$\phi_{\tau,\tilde{F}}(v) = \exp(\tau L)v + \tau\phi_1(\tau L)N_n + \frac{\tau^2}{h}\phi_2(\tau L)(N_n - N_{n-1}), \tag{60}$$

where $N_n = N(u_n,t_n)$ and $N_{n-1} = N(u_{n-1},t_{n-1})$.

**Algorithm 3.1.** (*ETD*2/*RK*4)

$$\begin{aligned} &a = \phi_{\frac{h}{2},\tilde{F}}(u_n), \quad b = \phi_{h,\tilde{F}}(u_n), && N_a = N(a, t_n + \tfrac{h}{2}),\\ &c = a + \tfrac{h}{2}(N_a - \tfrac{3}{2}N_n + \tfrac{1}{2}N_{n-1}), && N_c = N(c, t_n + \tfrac{h}{2}),\\ &d = b + h\exp(\tfrac{h}{2}L)(N_c - \tfrac{3}{2}N_n + \tfrac{1}{2}N_{n-1}), && N_d = N(d, t_n + h), \end{aligned} \tag{61}$$

$$u_{n+1} = b + \frac{h}{3}\exp\left(\frac{h}{2}L\right)(N_a + N_c - 3N_n + N_{n-1}) + \frac{h}{6}(N_d - 2N_n + N_{n-1}) \tag{62}$$

Since the underlying method for solving the transformed equation (58) has order four, we are guaranteed (given sufficient smoothness of the problem) that the resulting method has at least order four. In addition we were able to show that for ETD$k$/RK4, $k = 1, 2, 3$ the local error for a particular scalar test equation is $\mathcal{O}(h^5)$, and thus these methods are indeed of order four.

## 4. Stability

In this section we study the stability regions of the schemes discussed in this paper. In this and the next section we consider the seven methods ETDRK4 (50), ETDRK4-B (51), CF4 (43), IF4 (the standard IF-method based on RK4), and the three methods ETD1/RK4, ETD2/RK4 and ETD3/RK4 described in the previous section. Rather than considering the scalar test equation

$$\dot{u} = \lambda u, \tag{63}$$

we adopt the approach in [1,4] (and references therein), that is to consider the test equation

$$\dot{u} = cu + \lambda u, \tag{64}$$

where $c$ and $\lambda$ are scalars. We note that this approach only can give an indication of stability, since in general one can not linearize both terms simultaneously. When applying our methods to the Eq. (64), we may treat $cu$ as the linear term and $\lambda u$ as the nonlinear. For a fixed value of $c$ (or rather $hc$ where $h$ is the step size), we can obtain a stability region for $h\lambda$ as in the standard case (see [9]). For a fixed $hc$, our approach was simply to apply the method to Eq. (64) (with $u(0) = 1$) for $(h\lambda)_{i,j}$ a set of grid points in $\mathbb{C}$. Taking the absolute value of the resulting set of approximate solutions $(u_1)_{i,j}$, we obtained the boundary of the region using the function `contour` in Matlab.

To obtain any information from the stability regions, we need to consider various values of $ch$. Moreover if $L$ is a matrix representing the linear term of our equation, we would like to plot a region representing the whole spectrum of $L$. We propose to choose some subset $U \in \mathbb{C}$, and to plot the intersection of all regions with $ch \in U$. For applications we consider two particular choices of $U$:

- $ch \in \mathbb{R}^-$: By choosing various values of $ch$ on the negative real axis, we obtained several regions as seen in Fig. 2. The ETD1/RK4 method reduces to RK4 when the linear term is zero, and one recognizes the stability region of RK4 which is also the intersection since the regions increase with increasing $ch$. For the other two methods the regions increase at first, but then split into two and decrease for larger $ch$. As $ch \to -\infty$ the regions converge as seen on the plot. The regions of the methods ETDRK4, ETDRK4-B and CF4 have the same intersection as ETD1/RK4 and are not plotted (they also reduce to RK4 when the linear term is zero).
- $\mathrm{Re}(ch) = 0$: This situation is more demanding, and in Fig. 3 we have plotted regions for values of $ch$ in the interval $[-2\pi i, 2\pi i]$. The method with the largest intersection is ETDRK4-B, while the methods ETDRK4 and CF4 have smaller intersections but still include an interval of the imaginary axis. The intersections of the generalized IF methods touch the imaginary axis only at a point, which suggests that they will not work very well for the KdV-equation (Section 5.3). The reason for this is that the eigen-

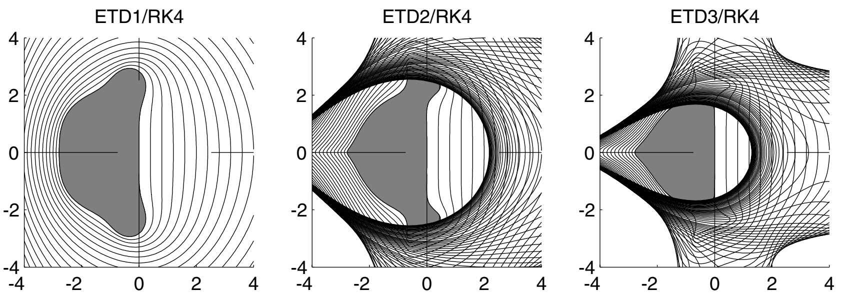

Fig. 2. Stability regions for different $hc \in \mathbb{R}^-$, and their intersection (shaded).

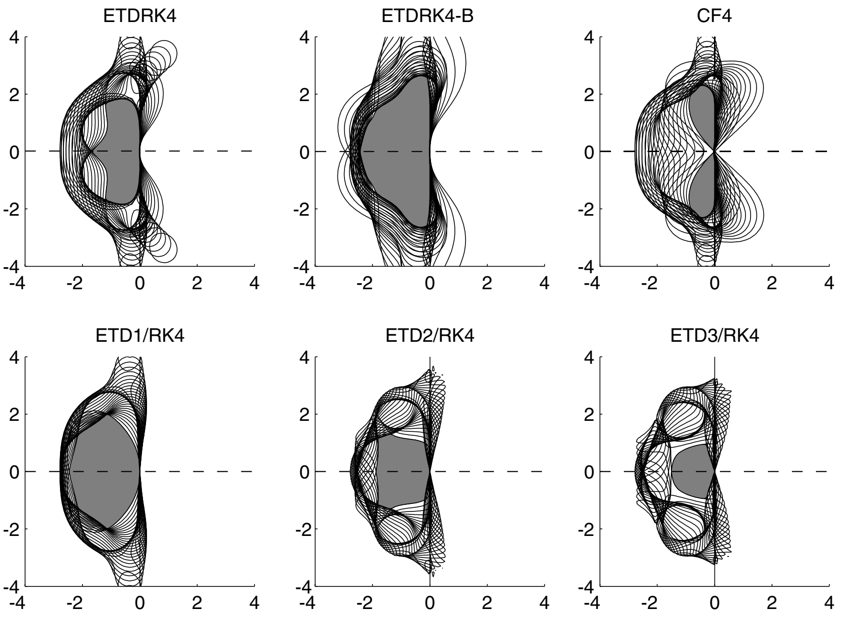

Fig. 3. Stability regions for different $hc \in [-2\pi i, 2\pi i]$, and their intersection.

values of the linearization of the nonlinear part lie on the imaginary axis. We note that these regions only give an indication of the stability of the methods. In the numerical experiments, the spectrum of the linear operators is one to two orders of magnitude greater than the interval we considered here, but still it seems to be giving the right picture.

## 5. Numerical experiments

In this section we consider three numerical examples for comparisons of the methods discussed in this paper. We will also see that the properties indicated by the stability analysis agree with the results in the experiments.

For stable computations of the $\phi_i$ functions (6) (they suffer from cancellation errors for values close to zero), we use the approach of Kassam and Trefethen [14]. For $z$ close to zero they propose to compute $f(z)$ by approximating the integral

$$f(z) = \frac{1}{2\pi i}\int_{\Gamma}\frac{f(t)}{t-z}\,dt, \tag{65}$$

over a contour $\Gamma$ well separated from 0. This approach generalizes to a matrix $L$ by

$$f(L) = \frac{1}{2\pi i}\int_{\Gamma} f(t)(tI-L)^{-1}\,dt. \tag{66}$$

The approach is to approximate these integrals along the contour by the trapezoidal rule. This method is easy to implement and gives stable results. However, if the matrix $L$ is full and large, the computations become expensive, and one might have to consider other methods for computing the matrix functions. One approach including transforming $L$ to upper triangular and making use of a block form of *Parlett's algorithm* is described in [6]. If the matrix or operator $L$ is still to big one may need to consider Krylov methods [10,11], but in this case one needs to approximate the function applied to a vector in every step. Still this can be effective if one is able to construct methods with a minimum number of function evaluations as in the exponential integrators of [11].

### 5.1. The Kuramoto–Sivashinsky equation

We consider the following PDE with initial data borrowed from [14]:

$$u_t = -uu_x - u_{xx} - u_{xxxx}, \quad x \in [0, 32\pi], \tag{67}$$

$$u(x,0) = \cos\left(\frac{x}{16}\right)\left(1+\sin\left(\frac{x}{16}\right)\right). \tag{68}$$

We use a 128-point Fourier spectral discretization and integrate from $t = 0$ to $t = 65$.

Since the eigenvalues of the linear operator of this equation are distributed on the real axis to the left of 0.25, we expect all methods to work reasonably well considering the stability analysis. With periodic boundary conditions the system becomes diagonal in Fourier space, and the $\phi_i$ functions are just stored as vectors. In Fig. 4 we have plotted the results using the seven different methods. We see that the generalized IF method ETD3/RK4 is the most efficient all though it fails to produce a result at the largest step size. The improvement over the standard IF method is about $10^4$. Note also that this method together with ETD2/RK4 appears to have higher order than the other methods, all though they are only fourth order in general. We also observe that the ETDRK4-B performs better than ETDRK4.

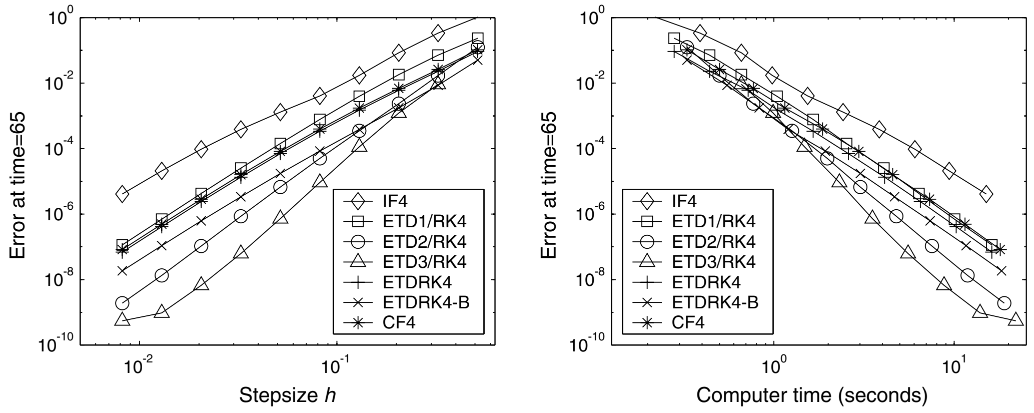

Fig. 4. Step size versus error (left), and cpu time versus error (right), for the Kuramoto–Sivashinsky equation.

## 5.2. The Allen–Cahn equation

We consider an other example from [14] which uses a 50-point Chebyshev spectral method.

$$u_t = \epsilon u_{xx} + u - u^3, \quad x \in [-1, 1], \tag{69}$$

with $\epsilon = 0.01$ and initial conditions

$$u(x, 0) = 0.53x + 0.47\sin(-1.5\pi x), \quad u(-1, t) = -1, \quad u(1, t) = 1. \tag{70}$$

We apply the Matlab function cheb from [22] for generation of the grid and the differentiation matrix. Note that the differentiation matrix in this example is full, and thus the function evaluations involved require much more work than in the previous example. Keeping in mind that these matrix functions need only be computed once the computational cost per step approaches ($N^2$) as the number of steps increase. This is seen in Fig. 5 and for few steps the standard IF method is the most efficient. For higher accuracy the ETD2/IF4 and ETD3/IF4 perform best.

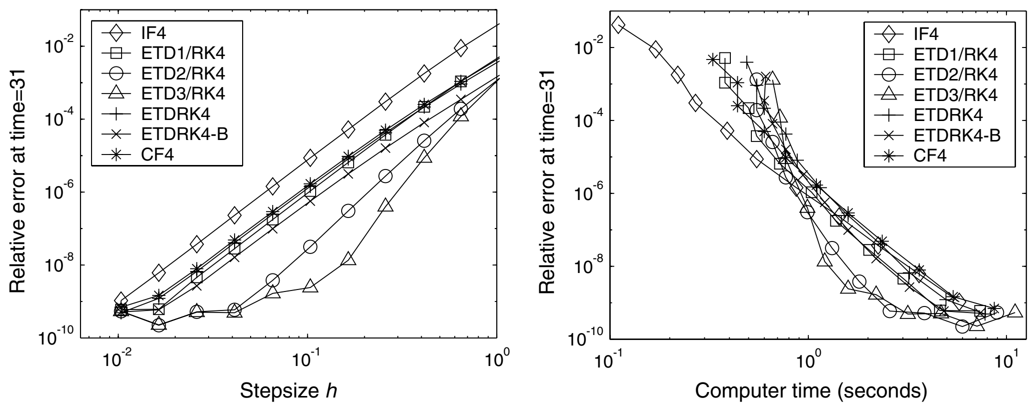

Fig. 5. Step size versus error (left), and CPU time for the overall algorithm versus error for the Allen–Cahn equation.

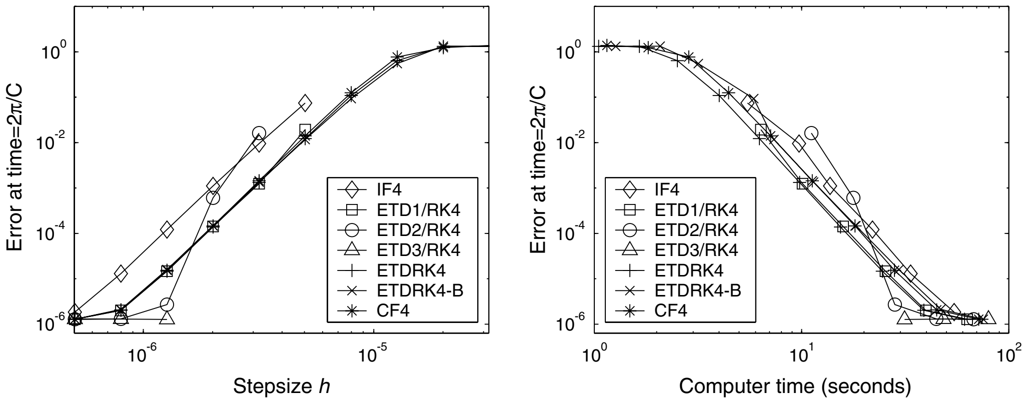

Fig. 6. Step size versus error (left), and CPU time versus error (right), for the KdV equation.

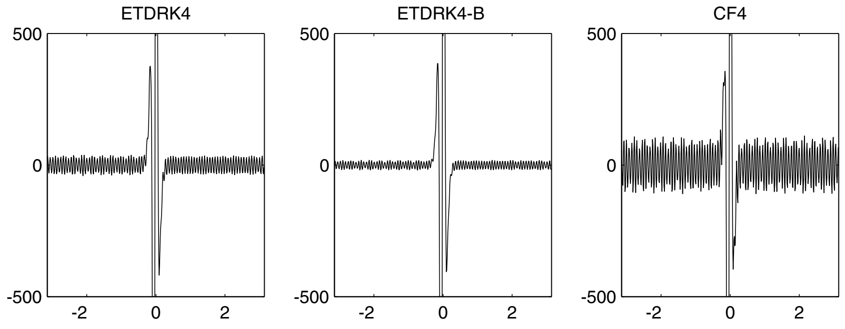

Fig. 7. The third order time derivative of the numerical solutions ($-Lu$) with $h = 10^{-5}$ using ETDRK4, ETDRK4-B and CF4.

## 5.3. The Korteweg de Vries equation

We consider the equation

$$u_t = -u_{xxx} - uu_x, \quad x \in [-\pi, \pi], \tag{71}$$

with initial conditions [4]

$$u(x,0) = 3C/\cosh^2(\sqrt{C}x/2), \tag{72}$$

with $C = 625$. The solution of this system is periodic with period $2\pi/C$ and is given $u(x,t) = u(x - Ct,0)$. We use a 256-point Fourier spectral discretization and run the methods for one period. As seen in Fig. 6 the IF method and ETD1/RK4 perform quite well, but the other generalized IF methods break down at quite small step sizes. This is in good agreement with the stability regions in Fig. 3. The methods ETDRK, ETDRK-B and CF4 all perform very well and produce smooth solutions. As a comparison of these methods, in Fig. 7, we plotted the third space derivative of the numerical solutions after one period obtained with time step $h = 10^{-5}$. Although the difference is not big, the plot agrees well with the regions in Fig. 3.

## 6. Concluding remarks

We have treated various numerical methods for solving semi-discretized PDEs, all having in common that they treat the linear term in an *exact* manner. Such methods include IF, ETD and a special case of the commutator-free Lie group methods. In this paper we have suggested generalizations to the IF approach, and in particular obtained new methods with several orders in magnitude improvement in accuracy over the standard IF method. We have pointed out a connection between the ETDRK4-method proposed by Cox and Matthews with the CF4-method of Celledoni et al. We also constructed a new fourth order ETDRK method where the numerical experiments indicated that this method was more accurate than the one given in [4]. To our knowledge there does not exist any order theory for the ETDRK methods, which would be useful if one wants to construct methods for special purposes. Accuracy, stability and minimizing the number of function evaluations are all properties one would like to take into account both for ETDRK and IF related methods. However, after this paper first was submitted, a convergence analysis of collocation type ETD methods has appeared in [12]. We are not aware of much work done on general integrating factor methods besides [17]. In this paper we have pointed out one way of constructing schemes, and it is an interesting question whether this methodology applies also to other applications.

## Acknowledgments

I thank Hans Munthe-Kaas and Brynjulf Owren for useful discussions, and Arieh Iserles for pointing out Ref. [20]. Also I thank the anonymous referees for valuable comments and suggestions improving the readability of the paper.

## References

[1] G. Beyklin, J.M. Keiser, L. Vozovoi, A new class of time discretization schemes for the solution of nonlinear PDEs, J. Comput. Phys. 147 (1998) 362–387.
[2] J.P. Boyd, Chebyshev and Fourier Spectral Methods, Dover, New York, 2001, Available from: <http://www-personal.engin.umich.edu/~jpboyd> .
[3] E. Celledoni, A. Martinsen, B. Owren, Commutator-free Lie group methods, FGCS 19 (3) (2003) 341–352.
[4] S.M. Cox, P.C. Matthews, Exponential time differencing for stiff systems, J. Comput. Phys. 176 (2002) 430–455.
[5] P.E. Crouch, R. Grossman, Numerical integration of ordinary differential equations on manifolds, J. Nonlinear Sci. 3 (1993) 1–33.
[6] P. Davies, N. Higham, A Schur–Parlett algorithm for computing matrix functions, SIAM J. Matrix Anal. Appl. 25 (2) (2003) 464–485.
[7] B. Fornberg, T.A. Driscoll, A fast spectral algorithm for nonlinear wave equations with linear dispersion, J. Comput. Phys. 155 (1999) 456–467.
[8] E. Hairer, S.P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Nonstiff Problems, second ed.Springer Series in Computational Mathematics, vol. 8, Springer, Berlin, 1993.
[9] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic ProblemsSpringer Series in Computational Mathematics, vol. 14, Springer, Berlin, 1996.
[10] M. Hochbruck, C.h. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM J. Sci. Comput. 34 (5) (1997) 1911–1925.
[11] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, SIAM J. Numer. Anal. 19 (5) (1998) 1552–1574.
[12] M. Hochbruck, A. Ostermann, Exponential Runge–Kutta methods for parabolic problems, Appl. Numer. Math. (to appear).
[13] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, A. Zanna, Lie-group methods, Acta Numer. (2000) 215–236.
[14] A.K. Kassam, L.N. Trefethen, Fourth-order time stepping for stiff PDEs, SIAM J. Sci. Comput. (to appear).
[15] S. Krogstad, RKMK-related methods for stiff nonlinear PDEs, Report, University of Bergen, 2003.
[16] E. Lodden, Geometric integration of the heat equation, Masters Thesis, University of Bergen, 2000.
[17] Y. Maday, A.T. Patera, E.M. Rønquist, An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow, J. Sci. Comp. 5 (4) (1990) 263–292.
[18] H. Munthe-Kaas, High order Runge–Kutta methods on manifolds, J. Appl. Numer. Math. 29 (1999) 115–127.
[19] B. Owren, A. Marthinsen, Runge–Kutta methods adapted to manifolds and based on rigid frames, BIT 39 (1) (1999) 116–142.

[20] S. Nørsett, An A-stable modification of the Adams–Bashforth methodsLecture Notes in Mathematics, vol. 109, Springer, Berlin, 1969, pp. 214–219.

[21] A. Suslowicz, Application of numerical Lie group integrators to parabolic PDE's, Technical Report No. 013, University of Bergen, 2001.

[22] L.N. Trefethen, Spectral methods in MATLAB, SIAM, Philadelphia, 2000.