Lie group integrators with nonautonomous frozen vector fields

Borislav V. Minchev

Department of Computer Science, University of Bergen,
Thormhlensgate 55, N-5020 Bergen, Norway

Abstract

Lie group methods for nonautonomous semi-discretized in space, partial differential equations are considered. The choice of frozen vector field and its corresponding algebra action on the manifold for such problems is discussed. A new exponential integrator for semilinear problems, based on commutator free Lie group methods with algebra action arising from the solutions of differential equations with nonautonomous frozen vector fields is derived. The proposed new scheme is then compared with some existing methods in several numerical experiments.

Key words: Lie group methods, algebra action, exponential integrators, stiff systems
1991 MSC: 65L06, 65M29, 35G10, 58F39

1 Introduction

Recently, a lot of Lie group integrators for solving semi-discretized partial differential equations (PDEs) has been derived in the literature. The original idea was first introduced in [16] and then further investigated for the heat equation in [4,12,17], stiff PDEs in [10,11,14], convection diffusion problems in [3] and for the Schrödinger equation in [1]. What is common between all this methods is that to advance from one point to another they all use an algebra action arising from the solution of an *autonomous* differential equation, which does not depend explicitly on the time t.

In this paper we propose a way how to construct Lie group integrators for nonautonomous problems based on an algebra action arising from the solution

Email address: `Borko.Minchev@ii.uib.no` (Borislav V. Minchev).
URL: `http://www.ii.uib.no/∼borko` (Borislav V. Minchev).

of a differential equation which can depend explicitly on t. The idea is a natural extension of the autonomous case. The approach is to rewrite the differential equation in its equivalent autonomous form and then to apply the *affine* action [16] to the transformed equation. Thus, what we obtain are time dependent *frozen* vector fields. This provides us some extra freedom in the construction process, which can be used to choose the algebra action to be a better approximation of the flow of the original vector field.

The paper is organized as follows: We start in Section 2 with introducing some notation and the basic theory involved. Next we consider the framework for nonautonomous problems and discuss how it is related with the choice of the algebra action. In Section 3 we propose a new time dependent frozen vector field and its corresponding algebra action. In addition, we discuss some further generalizations. In Section 4 we derive a new exponential integrator for semilinear problems based on the fourth order commutator free Lie group method [4]. Finally in Section 5 we compare the proposed new exponential integrator with some existing methods and discuss the advantages of the new approach.

2 Background theory and notations

The framework which we use in this paper is given by Lie groups and their action upon a homogeneous manifold [8,15,16]. We take advantage of the fact that, in order to construct a Lie group integrator, we do not really need to know what the structure of the Lie group is and how it acts on the manifold. It is enough to specify the *generic presentation* of the differential equation and the algebra action on the manifold (see [14]). For simplicity, we do not include a discussion on the structure of the underlying Lie group and how it acts on a manifold.

Let us first consider the following differential equation defined on a $d + p$ dimensional manifold $\mathcal{M} \equiv \mathbb{R}^{d+p}$.

$$y' = \mathsf{f}(y(t)), \qquad y(t_0) = y_0. \tag{2.1}$$

The very first question in the construction of a Lie group integrator is how to define the basic motions on $\mathcal{M}$. Since $\mathbb{R}^{d+p}$ is a linear space, it is easy to construct integrator which stays on the manifold. The challenge in this case is how to choose the basic motions in such a way that they provide a good approximation to the flow of the original vector field. In this paper we define the basic movements on $\mathcal{M}$ to be given by the solution of a simpler differential equation

$$y' = \mathcal{F}_\Theta(y), \qquad y(t_0) = y_0, \tag{2.2}$$

which locally approximates (2.1). Thus, the Lie algebra $\mathfrak{g}$ is generated from the set of all coefficients Θ of the *frozen* vector fields $\mathcal{F}_\Theta$ and the Lie algebra action $* : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$ on the manifold is simply given by the solution of (2.2). In other words, if $\Theta \in \mathfrak{g}$, its action upon the point $y_0 \in \mathcal{M}$, which we denote by $h\Theta * y_0$, is given by the solution of (2.2) at time $t_0 + h$. Every frozen vector field can be represented in the form $\mathcal{F}_\Theta(y) = \Theta \circledast y$, where $\circledast : \mathfrak{g} \times \mathcal{M} \to T\mathcal{M}$ and according to (2.2), it satisfies

$$\Theta \circledast y = \left.\frac{d}{dt}\right|_{t=0} t\Theta * y.$$

Note that the map $\Theta \to \mathcal{F}_\Theta$ is an algebra homomorphism between $\mathfrak{g}$ and the set of all vector fields on $\mathcal{M}$. If the algebra action $*$ is *transitive* i.e. starting from a point $y_0 \in \mathcal{M}$ we can reach any other point $y_1 \in \mathcal{M}$ by letting some element $\Theta \in \mathfrak{g}$ act on y_0, the differential equation (2.1) can be rewritten in the form

$$y' = F(y) \circledast y, \qquad y(t_0) = y_0, \tag{2.3}$$

where $F : \mathcal{M} \to \mathfrak{g}$. The above formulation is called the *generic presentation* of the differential equation on the manifold and plays an important role in the theory of the Lie group integrators (see [16]).

The choice of the frozen vector field $\mathcal{F}_\Theta$ and its corresponding algebra action, very much depends of the actual structure of $\mathsf{f}(y)$. The simplest possible case is when $\mathfrak{g} = \{\mathsf{b} \in \mathbb{R}^{d+p}\}$, $F(y_0) = \mathsf{f}(y_0) = \mathsf{b}$ and $\mathcal{F}_\mathsf{b}(y) = \mathsf{b} \circledast y = \mathsf{b}$ then the algebra action on $\mathcal{M}$ is given by translations and we recover the traditional integration schemes. In the case when $\mathsf{f}(y) = \mathsf{L}(y)y + \mathsf{N}(y)$, one can define the Lie algebra $\mathfrak{g} = \{(\mathsf{A}, \mathsf{b}) \in \mathbb{R}^{(d+p)\times(d+p)} \rtimes \mathbb{R}^{d+p}\}$, the function $F(y_0) = (L(y_0), N(y_0)) = (\mathsf{A}, \mathsf{b})$ and the frozen vector field $\mathcal{F}_{(\mathsf{A},\mathsf{b})}(y) = (\mathsf{A}, \mathsf{b}) \circledast y = \mathsf{A}y + \mathsf{b}$. This is exactly the affine algebra action proposed in [16]. Note that such a representation of $\mathsf{f}(y)$ is always possible, for example by letting $\mathsf{L}(y)$ be the Jacobian of f at the point y and $\mathsf{N}(y) = \mathsf{f}(y) - \mathsf{L}(y)y$. Other choices are also possible see for example [12,17].

In this paper we are interesting in the construction of Lie group methods for the following nonautonomous problem defined on $\mathbb{R}^d$

$$u' = f(u, t), \qquad u(t_0) = u_0. \tag{2.4}$$

Formally it does not fit in the above presented framework, but by adding the trivial differential equation $t' = 1$ to the system (2.4), we can rewrite it in the form (2.1) with $p = 1$ and

$$\mathsf{f} = \begin{bmatrix} f(u,t) \\ 1 \end{bmatrix}, \quad y = \begin{bmatrix} u \\ t \end{bmatrix}.$$

This of course is a very well known idea in the theory of ODEs, however its application to Lie group methods, if done carefully, can lead to some extra freedom which we would like to exploit. Note that now the time variable t goes in the definition of the manifold $\mathcal{M}$ and therefore it also appears like one of the arguments of the generic function F, the frozen vector field $\mathcal{F}$ and its corresponding algebra action.

From a computational point of view it might look some what unreasonable to increase the dimensionality of the problem, but we keep in mind that the solution of (2.4) is given by the first d components of the solution of (2.1). Thus, the approach is to apply a Lie group method to equation (2.1) and then to restate it in $\mathbb{R}^d$.

The simplest nonautonomous case is when the Lie algebra $\mathfrak{g} = \{\mathsf{b} \in \mathbb{R}^{d+1}\}$ or equivalently $\mathfrak{g} = \{(b^{[0]}, \lambda) : b^{[0]} \in \mathbb{R}^d, \lambda \in \mathbb{R}\}$ then the generic function is given by $F\left(\left[\begin{smallmatrix} u_0 \\ t_0 \end{smallmatrix}\right]\right) = (f(u_0, t_0), 1) = (b^{[0]}, 1)$, the frozen vector field is $\mathcal{F}_{(b^{[0]},\lambda)}\left(\left[\begin{smallmatrix} u \\ t \end{smallmatrix}\right]\right) = \left[\begin{smallmatrix} b^{[0]} \\ \lambda \end{smallmatrix}\right]$ and the algebra action is $h(b^{[0]}, \lambda) * \left[\begin{smallmatrix} u_0 \\ t_0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} u_0+hb^{[0]} \\ t_0+h\lambda \end{smallmatrix}\right]$. When $f(u,t) = L(u,t)u + N(u,t)$ then the Lie algebra $\mathfrak{g} = \{(\mathsf{A}, \mathsf{b}) \in \mathbb{R}^{d+1\times d+1} \rtimes \mathbb{R}^{d+1}\}$. It can also be represented like the set of all triples $(A, b^{[0]}, \lambda)$ closed under linear combinations and *commutators* between the elements (see section 3), where $A \in \mathbb{R}^{d\times d}, b^{[0]} \in \mathbb{R}^d, \lambda \in \mathbb{R}$. In this case the generic function is defined like $F\left(\left[\begin{smallmatrix} u_0 \\ t_0 \end{smallmatrix}\right]\right) = (L(u_0, t_0), N(u_0, t_0), 1) = (A, b^{[0]}, 1)$, the frozen vector field is $\mathcal{F}_{(A,b^{[0]},\lambda)}\left(\left[\begin{smallmatrix} u \\ t \end{smallmatrix}\right]\right) = \left[\begin{smallmatrix} Au+b^{[0]} \\ \lambda \end{smallmatrix}\right]$ and its corresponding algebra action is given by $h(A, b^{[0]}, \lambda) * \left[\begin{smallmatrix} u_0 \\ t_0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} e^{hA}u_0+hb^{[0]}\phi^{[1]}(hA) \\ t_0+h\lambda \end{smallmatrix}\right]$, where e^{hA} denotes the matrix exponential and the function $\phi^{[1]}$ is given in Lemma 2 (see section 3).

If we consider just the first d components of the algebra action: in the first case we simply obtain translations like basic motions on $\mathbb{R}^d$; in the second case we again recover the affine action. Thus, we conclude that for thes two cases the only difference between autonomous and nonautonomos systems is in the definition of the generic function F, which for nonautonomous systems depends also from the time variable. We remark that in the above two cases the frozen vector field does not really depends of t. This explains the observed similarities between autonomous and nonautonomous systems.

A more interesting situation arises when the function $f(u,t)$ has the form $L(u,t)u + N^{[0]}(u,t) + tN^{[1]}(u,t)$. A natural question now is how to choose the frozen vector field in this case. In the next section, we propose a time dependent frozen vector field and its corresponding algebra action which reflects the structure of f. In addition, we discuss how this idea can be further generalized if second and higher powers of t are included.

3 Nonautonomous frozen vector fields

We first consider the case when the vector field of (2.4) has the form

$$f(u,t) = L(u,t)u + N^{[0]}(u,t) + tN^{[1]}(u,t).$$

If we treat the nonlinear part as a single function $N = N^{[0]} + tN^{[1]}$ then we do not gain anything in comparison with the affine case presented in the previous section. A more demanding task is to allow our frozen vector field to be time dependent. It is desirable in this case to approximate the nonlinear part of $f(u,t)$ by a linear polynomial of t.

The only way to include t in the definition of the frozen vector field is to append it to the dependent variables. Thus, by adding the trivial differential equation $v' = 1$, $v(t_0) = t_0$ to the system (2.4), we obtain

$$y' = \mathsf{L}y + \mathsf{N}, \qquad y(t_0) = y_0, \tag{3.1}$$

where

$$\mathsf{L} = \begin{bmatrix} L(u,t) & N^{[1]}(u,t) \\ 0 & 0 \end{bmatrix}, \quad \mathsf{N} = \begin{bmatrix} N^{[0]}(u,t) \\ 1 \end{bmatrix}, \quad y = \begin{bmatrix} u \\ v \end{bmatrix}, \quad y_0 = \begin{bmatrix} u_0 \\ t_0 \end{bmatrix}.$$

The advantage of rewriting (2.4) in the above form is that now we can easily see how to define the Lie algebra $\mathfrak{g}$, the generic function F, the frozen vector field $\mathcal{F}$ and its corresponding algebra action (see section 2). The Lie algebra in this case is $\mathfrak{g} = \{(\mathsf{A}, \mathsf{b}) \in \mathbb{R}^{d+1\times d+1} \rtimes \mathbb{R}^{d+1}\}$, with

$$\mathsf{A} = \begin{bmatrix} A & b^{[1]} \\ 0 & 0 \end{bmatrix}, \quad \mathsf{b} = \begin{bmatrix} b^{[0]} \\ \lambda \end{bmatrix}, \tag{3.2}$$

where $A \in \mathbb{R}^{d\times d}$, $b^{[1]}, b^{[0]} \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}$. Equivalently we can represent $\mathfrak{g}$ like the quadruplet $(A, b^{[1]}, b^{[0]}, \lambda)$ closed under linear combinations and commutators. The function F which provides the generic presentation is given by

$$F\left(\begin{bmatrix} u_0 \\ t_0 \end{bmatrix}\right) = \big(L(u_0,t_0), N^{[1]}(u_0,t_0), N^{[0]}(u_0,t_0), 1\big) = (A, b^{[1]}, b^{[0]}, 1). \tag{3.3}$$

To obtain an explicit form of the frozen vector field we use the following result.

Lemma 1 *The solution of the differential equation* $y' = \mathsf{A}y + \mathsf{b}$, $y(t_0) = y_0$, *at the time* $t_0 + h$ *is given by*

$$y(t_0+h) = \begin{bmatrix} u(t_0+h) \\ t_0 + h\lambda \end{bmatrix},$$

where $u(t_0+h)$ is the solution of

$$u' = Au + b^{[0]} + t_0(1-\lambda)b^{[1]} + t\lambda b^{[1]}, \qquad u(t_0) = u_0.$$

Proof: The proof follows directly by substituting (3.2) in the differential equation $y' = \mathsf{A}y + \mathsf{b}$ and then solving it with respect to the last variable. □

Thus, we have obtained the following time dependent frozen vector field

$$\mathcal{F}_{(A,b^{[1]},b^{[0]},\lambda)}\left(\begin{bmatrix} u \\ t \end{bmatrix}\right) = (A, b^{[1]}, b^{[0]}, \lambda) \circledast \begin{bmatrix} u \\ t \end{bmatrix} = \begin{bmatrix} Au + c_0 + tc_1 \\ \lambda \end{bmatrix}, \tag{3.4}$$

where $c_0 = b^{[0]} + t_0(1-\lambda)b^{[1]}$ and $c_1 = \lambda b^{[1]}$. Note that for $\lambda = 1$ we have $c_0 = b^{[0]}$, $c_1 = b^{[1]}$ and therefore the generic presentation $F\left(\left[\begin{smallmatrix} u \\ t \end{smallmatrix}\right]\right) \circledast \left[\begin{smallmatrix} u \\ t \end{smallmatrix}\right] = \left[\begin{smallmatrix} f(u,t) \\ 1 \end{smallmatrix}\right]$ is satisfied.

The last thing which we need to define is the algebra action corresponding to the frozen vector field (3.4). In the next Lemma we give a general formula for the flow of the frozen vector field, which approximates the nonlinear part of f by a polynomial of t of degree p.

Lemma 2 *The solution of the differential equation*

$$u' = Au + \sum_{j=0}^{p} t^j c_j, \qquad u(t_0) = u_0,$$

where $p \in \mathbb{N}$, $A \in \mathbb{R}^{d\times d}$ and $c_j \in \mathbb{R}^d$ at the time $t_0 + h$ is given by

$$u(t_0+h) = e^{hA}u_0 + \sum_{k=0}^{p} h^{k+1}\delta_k \phi^{[k+1]}(hA),$$

where $\delta_k = \sum_{j=k}^{p} \frac{j!}{(j-k)!} t_0^{j-k} c_j$, $\phi^{[1]}(z) = \frac{e^z - 1}{z}$ and $\phi^{[k+1]}(z) = \frac{\phi^{[k]}(z) - \phi^{[k]}(0)}{z}$.

Proof: From the variation of constants formulae it follows that

$$\begin{aligned} u(t_0+h) &= e^{hA}u_0 + e^{hA}\int_0^h e^{-\tau A}\Big(\sum_{j=0}^{p}(t_0+\tau)^j c_j\Big)d\tau \\ &= e^{hA}u_0 + \sum_{j=0}^{p}\Big(\delta_j e^{hA}\frac{1}{j!}\int_0^h e^{-\tau A}\tau^j d\tau\Big). \end{aligned}$$

Multiple applications of integration by parts complete the proof. □

Combining the results of Lemma 1 and Lemma 2 leads to the following explicit form for the algebra action corresponding to the vector field (3.4)

$$h(A, b^{[1]}, b^{[0]}, \lambda) * \begin{bmatrix} u_0 \\ t_0 \end{bmatrix} = \begin{bmatrix} e^{hA}u_0 + h(b^{[0]} + t_0 b^{[1]})\phi^{[1]}(hA) + h^2 \lambda b^{[1]} \phi^{[2]}(hA) \\ t_0 + h\lambda \end{bmatrix} \quad (3.5)$$

Once we have defined the generic presentation, the Lie algebra $\mathfrak{g}$ and its action on $\mathcal{M}$ we can use any Lie group method to find the solution of (2.4). The solution of (3.1) is simply given by its first d components.

In the case when a Runge-Kutta Munthe-Kaas method with exact Exp map is used the format requires the inverse of the dExp map (see [16]). Computationally it might be very expensive to find exactly the dExp^{-1} map and thus the approach proposed in [16] is to replace it with polynomial approximation of order higher than the order of the method. This imposes the necessity of using *commutators* between the elements of $\mathfrak{g}$. In this case if $\Theta_i = (A_i, b_i^{[1]}, b_i^{[0]}, \lambda_i)$ for $i = 1, 2$ are two elements from $\mathfrak{g}$ then their commutator is given by

$$[\Theta_1, \Theta_2] = \left([A_1, A_2], A_1 b_2^{[1]} - A_2 b_1^{[1]}, A_1 b_2^{[0]} - A_2 b_1^{[0]} + \lambda_2 b_1^{[1]} - \lambda_1 b_2^{[1]}, 0\right),$$

where $[A_1, A_2] = A_1 A_2 - A_2 A_1$ is the matrix commutator.

The above approach can be easily generalized when the function $f(u,t) = L(u,t)u + \sum_{k=0}^{p} t^k N^{[k]}(u,t)$. In this case we append p trivial differential equations corresponding to $t, t^2, \ldots, t^p$ to the system (2.4) . Thus, the dimension of the manifold is $d+p$, but we keep in mind that we are only interested in its first d components. The Lie algebra is $\mathfrak{g} = \{(A, b^{[p]}, \ldots, b^{[0]}, \lambda) : A \in \mathbb{R}^{d\times d}, \lambda \in \mathbb{R}, b^{[k]} \in \mathbb{R}^d\}$ and its action upon the manifold is given by Lemma 2. The coefficients c_j can be found in the same way as in Lemma 1. For $p = 2$ they are

$$\begin{aligned} c_0 &= b^{[0]} + (1-\lambda)t_0 b^{[1]} + (1-\lambda)^2 t_0^2 b^{[2]}, \\ c_1 &= \lambda b^{[1]} + 2\lambda(1-\lambda)t_0 b^{[2]}, \\ c_3 &= \lambda^2 b^{[2]}. \end{aligned}$$

We conclude this section with the observation that based on the same idea, methods with approximations of the nonlinear part of f by trigonometric polynomials can also be derived. In this case, the exact flow of the frozen vector field can be computed in the similar manner (see [14]).

4 Exponential integrator for semilinear problems

In this section we derive an exponential integrator based on the frozen vector field (3.4) and its corresponding algebra action (3.5) for the semilinear problem

$$
u' = Lu + N(u,t), \qquad u(t_0) = u_0, \tag{4.1}
$$

where L is a constant linear term and N is a nonlinear term. Such systems often arise after the spatial discretization of certain PDEs. Comparisons between the stability regions for different Lie group methods applied to semi-discretized stiff PDEs is given in [11]. There the author suggests that for this type of problem the best methods are likely to be the *commutator free* Lie group methods [4]. This provides our motivation in the choice of the Lie group method.

Next we give an equivalent formulation of the method proposed in [4]. This formulation allows us to construct methods without knowing what the exact structure of the Lie group acting on the manifold is, or how the Exp map between the Lie algebra and the Lie group is defined. The general format of an s stage commutator free Lie group method advancing from point y_n to point y_{n+1} with a time step of size h is given by the following algorithm.

Algorithm 1 *(Commutator-free Lie group method)*

for $i = 1, \ldots, s$ **do**
$\quad U_i = (h \sum_{k=1}^{s} \alpha_{iJ}^{k} F_k) * \cdots * (h \sum_{k=1}^{s} \alpha_{i1}^{k} F_k) * y_n$
$\quad F_i = F(U_i)$
end
$u_{n+1} = (h \sum_{k=1}^{s} \beta_{J}^{k} F_k) * \cdots * (h \sum_{k=1}^{s} \beta_{1}^{k} F_k) * y_n$

Here the function F gives the generic presentation (2.3), the coefficients α_{ij}^k, β_j^k are parameters of the method and the value J counts the number of flow calculations required at each stage. In [4], the following fourth order method based on the classical fourth order method of Kutta is proposed.

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
\frac{1}{2} & \frac{1}{2} & 0 & 0 & \\
\frac{1}{2} & -\frac{1}{2} & 0 & 1 & \\
\hline
 & \frac{1}{4} & \frac{1}{6} & \frac{1}{6} & -\frac{1}{12} \\
 & \frac{1}{12} & \frac{1}{6} & \frac{1}{6} & \frac{1}{4}
\end{array}
\tag{4.2}
$$

We use the symbol } to denote all the substages included in a stage with

$J > 1$. Note that in (4.2) the frozen vector field corresponding to the second stage is the same as for the first substage of the fourth stage. This reduces the cost of the method.

In order to use the frozen vector field (3.4) from the previous section we rewrite the nonlinear part of (4.1) in the form

$$N(u,t) = N_n + t\,\frac{N(u,t) - N_n}{t} = N^{[0]} + t\,N^{[1]}, \tag{4.3}$$

where $N_n = N(u_n, t_n)$ is the value of the nonlinear part at the beginning of the step number n. Keeping in mind (3.3) and (3.5), based on (4.2), we have found a new fourth order exponential integrator, which written in the original u variable is given by

$$\begin{aligned}
U_1 &= u_n\ ,\\
U_2 &= e^{\frac{hL}{2}} u_n + h\tfrac{1}{2}\phi^{[1]} N_n\ ,\\
U_3 &= e^{\frac{hL}{2}} u_n + h\left[\tfrac{1}{2}\phi^{[1]} N_n + \left(\tfrac{t_n}{2}\phi^{[1]} + \tfrac{h}{4}\phi^{[2]}\right) N_2^{[1]}\right],\\
U_4 &= e^{\frac{hL}{2}} U_2 + h\left[\tfrac{1}{2}\phi^{[1]} N_n + \left(t_n\phi^{[1]} + \tfrac{h}{2}\phi^{[1]} + \tfrac{h}{2}\phi^{[2]}\right) N_3^{[1]}\right],\\
\widehat{U} &= e^{\frac{hL}{2}} u_n + h\left[\tfrac{1}{2}\phi^{[1]} N_n + \left(t_n\phi^{[1]} + \tfrac{h}{2}\phi^{[2]}\right)\left(\tfrac{N_2^{[1]}}{6} + \tfrac{N_3^{[1]}}{6} - \tfrac{N_4^{[1]}}{12}\right)\right],\\
u_{n+1} &= e^{\frac{hL}{2}} \widehat{U} + h\left[\tfrac{1}{2}\phi^{[1]} N_n + \left(t_n\phi^{[1]} + \tfrac{h}{2}\phi^{[1]} + \tfrac{h}{2}\phi^{[2]}\right)\left(\tfrac{N_2^{[1]}}{6} + \tfrac{N_3^{[1]}}{6} + \tfrac{N_4^{[1]}}{4}\right)\right],
\end{aligned} \tag{4.4}$$

where $N_j^{[1]} = \frac{N(U_j,\,t_n+c_jh)-N_n}{t_n+c_jh}$ for $j = 1,\dots,4$ and the arguments of all the $\phi^{[j]}$ functions are $\frac{hL}{2}$.

It is possible to rewrite (4.4) in equivalent form which does not involves splitting of the internal stages (see [1,14]). Such a representation is rather useless, since its implementation is more expensive than the one proposed, but it shows that (4.4) is a method based just on the pure Runge–Kutta idea.

If we represent the nonlinear part of (4.1), as a polynomial of second degree

$$N(u,t) = N_n + t\,\frac{N_n - N_{n-1}}{t} + t^2\,\frac{N(u,t) - 2N_n + N_{n-1}}{t^2},$$

where N_n and N_{n-1} are the values of N at the end of step number n and $n-1$ respectively, we obtain a method which fits into the framework of *general linear methods* [2]. Thus, we see that by just changing the algebra action, any Lie group method based on a pure Runge–Kutta method can result in a more general method. This is a very interesting phenomena which highlights the important role of the algebra action.

5 Numerical experiments

In this section we present results from numerical experiments on the Kuramoto-Sivashinsky and Allen-Cahn equations. For both examples we compare the following four methods:

- **IF4** The fourth order integrating factor method [5,10,18] based on the classical fourth order method of Kutta.
- **CF4** The fourth order commutator free Lie group method (4.2) with affine algebra action [4].
- **CF4A1** The method (4.4) with algebra action given by (3.5).
- **ETDRK4B** The method of Krogstad [10].

Since all of the above methods are based on the nonstiff order conditions, to avoid possible order reduction, we consider examples where the nonlinear term N has sufficient spatial regularity. In general, for applications concerning PDEs, the classical order of convergence is not always obtained. Order reduction, due to the lack of sufficient temporal and spatial smoothness, should be expected. For parabolic problems, full order of convergence can be observerd, if periodic boundary conditions are imposed [6,7].

To avoid problems with numerical instability, the computation of the $\phi^{[i]}$ functions, which suffer from cancelation errors when the eigenvalues of the discretized linear operator are close to zero, we use the approach of Kassam and Trefethen [9]. The idea is to evaluate the $\phi^{[i]}$ functions by Cauchy's integral formula

$$\phi^{[i]}(\gamma hL) = \frac{1}{2\pi i}\int_{\Gamma} \phi^{[i]}(\gamma\lambda)\,(\lambda I - hL)^{-1}\,d\lambda, \tag{5.1}$$

where $\gamma \in \mathbb{R}$. The contour Γ is a closed curve in the complex plane that encloses the eigenvalue of γhL and such that $\gamma\Gamma$ is well separated from zero. The trapezoidal rule is then used to approximate the integral in (5.1). If the discretized linear operator L is diagonal (Kuramoto-Sivashinsky equation) then the integral reduces simply to the mean of $\phi^{[i]}$ over the contour Γ. However, for non-diagonal problems (Allen-Cahn equation), the computations become more expensive and require the computation of several matrix inverses. That is why for such problems it is important for a method to use as few ϕ function evaluations as possible. In addition, if L has a special sparse structure one can apply effective methods to find it inverse [13,14].

The Kuramoto-Sivashinsky equation

The first example is the Kuramoto-Sivashinsky equation

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

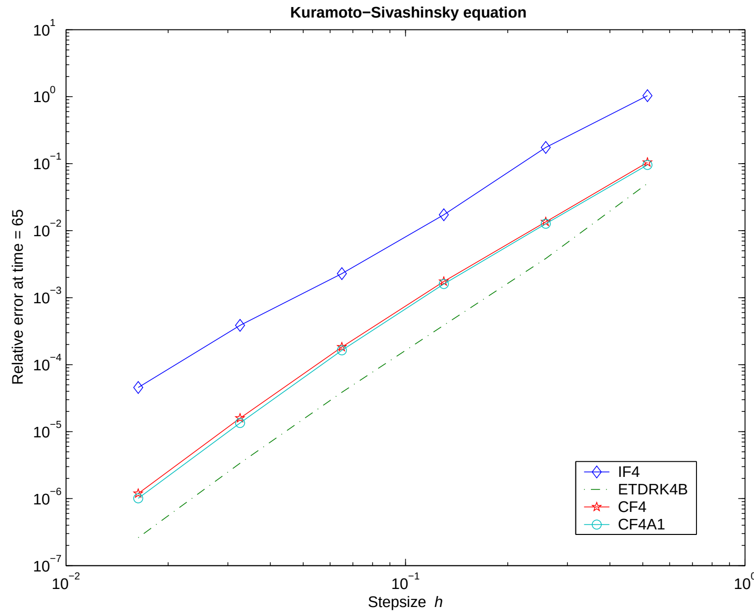

Fig. 1. Step size versus relative error for fourth order methods for the Kuramoto-Sivashinsky equation

with periodic boundary conditions and with the initial condition borrowed from [9]

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

A 128-point Fourier spectral discretization in space is used. Since the boundary conditions are periodic the transformed equation in the Fourier space can be represented in the form (4.1), the linear and nonlinear parts are defined as

$$(L\hat{u})(k) = (k^2 - k^4)\hat{u}(k), \qquad N(\hat{u}) = -\frac{ik}{2}(\mathsf{F}((\mathsf{F}^{-1}(\hat{u}))^2)),$$

where F denotes the discrete Fourier transform. The integration in time is done entirely in the Fourier space until $t = 65$. The results for the four different numerical schemes are plotted in Figure 1.

The Allen-Cahn equation

The second example is the Allen-Cahn equation written in the form

$$u_t = \varepsilon u_{xx} + u - u^3, \qquad x \in [-1,1],$$

where $\varepsilon = 0.01$ and with boundary and initial conditions also borrowed from [9]

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

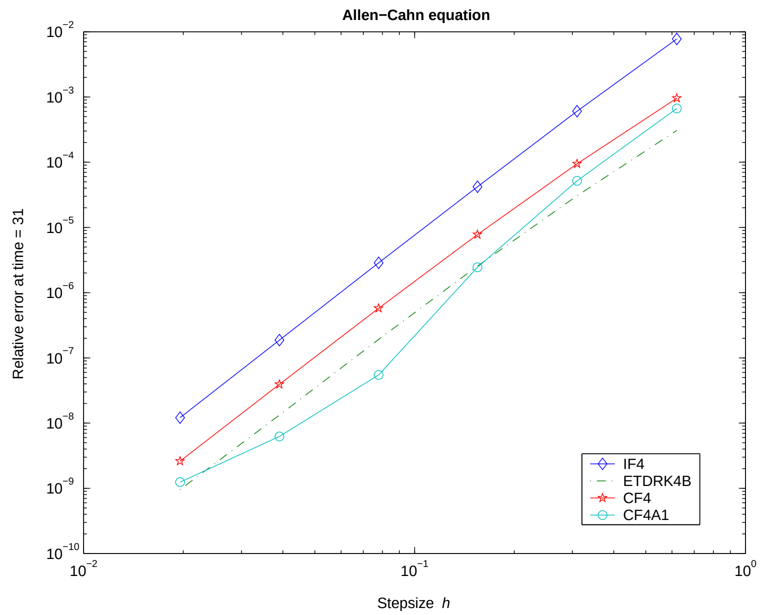

Fig. 2. Step size versus relative error for fourth order methods for the Allen-Cahn equation

After discretization in space based on the Chebyshev grid points we can rewrite the equation in the form (4.1), with

$$L = \varepsilon D^2, \qquad N(u) = u - u^3,$$

where D is the Chebyshev differentiation matrix [18], this means the matrix L is full. The standard build in MATLAB function `inv` was used to find the matrix inverse in (5.1). The integration in time is until $t = 31$. In Figure 2 we have plotted the results for the four different numerical schemes.

For both examples we see that all the methods exhibit the expected fourth order, but the best with respect to the accuracy is the ETDRK4B method. For the Kuramoto-Sivashinsky equation the improvement of using the algebra action (3.5) in CF4A1 is small in comparison with the affine action in CF4. However, for the non-diagonal example, the CF4A1 performs significantly better than CF4 and it is competitive with the ETDRK4B method. This together with the fact that it uses only 1 exponential and 2 ϕ function evaluations per step (for comparison ETDRK4B uses 2 exponentials and 5 ϕ function evaluations per step) suggests that CF4A1 is the best method in this case.

6 Concluding remarks

In this paper we have introduced the use of time dependent frozen vector fields in the construction of Lie group integrators for nonautonomous problems. This approach, provides extra freedom in the choice of the algebra action and allows us to choose the basic motions on the manifold to be given by the solutions of differential equations, which better approximate the flow of the original vector field. Based on this idea we have derived a new fourth order exponential integrator for semilinear problems with constant linear part. We do not claim that the proposed representation (4.3) of the nonlinear part is optimal. Other choices are worth investigating. However, the results from the numerical experiments suggest that the new method based on (4.3) is efficient in the case when the discretized linear operator is non-diagonal and a variable step size strategy is used. The content of this paper poses many questions which need to be answered. For example what are the stability regions of such methods and to what extend the spatial regularity of the problem effects the overall order of the method. An other important question is, for a given problem, how do we find a good algebra action? The goal is to choose a differential equation which is easier to solve, but still captures the key features of the original one. This is a very challenging task and it is likely to be problem dependent. The option presented here is to approximate the vector field by a higher order polynomial with constant coefficients. However, we should keep in mind that there is a certain balance between the benefit provided by increasing the order of the approximation and the computational cost of its corresponding algebra action.

Acknowledgments

The author would like to thank Hans Munthe-Kaas and William Wright for useful discussions throughout the work on this paper. This work has been partially supported by Norwegian Research Council through the GI4PDE project, contract number 142955/431.

References

[1] H. Berland and B. Owren, *Fourth order exponential time integrators for the nonlinear Scrödinger equation*, private conversation, April, 2004.

[2] J. C. Butcher, *Numerical methods for ordinary differential equations*, John Wiley & Sons, 2003.

[3] E. Celledoni, *Eulerian and Semi-Lagrangian schemes based on commutator free*

exponential integrators, Preprint submitted to GI Preprint Server, 2003.

[4] E. Celledoni, A. Marthinsen, and B. Owren, *Commutator-free Lie group methods*, FGCS **19(3)** (2003), 341–352.

[5] B. Fornberg and T.A. Driscoll, *A fast spectral algorithm for nonlinear wave equations with linear dispersion*, J.Comp. Phys. **155** (1999), 456–467.

[6] M. Hochbruck and A. Osterman, *Explicit exponential Runge-Kutta methods for semilinear parabolic problems*, Submitted to *SIAM J. Numer. Anal.*, 2004.

[7] ———, *Exponential Runge-Kutta methods for parabolic problems*, To appear in *Appl. Numer. Math.*, 2004.

[8] A. Iserles, H. Munthe-Kaas, S.P. Nørsett, and A. Zanna, *Lie-group methods*, Acta Numerica **9** (2000), 215–365.

[9] A.-K. Kassam and L.N. Trefethen, *Fourth order time stepping for stiff PDEs*, SIAM J. Sci. Comp., to appear, 2003.

[10] S. Krogstad, *Generalized integrating factor methods for stiff PDEs*, Preprint submitted to J. Comput. Phys., 2003.

[11] ———, *Topics in numerical Lie group integration*, Ph.D. thesis, University of Bergen, 2003.

[12] E. Lodden, *Geometric integration of the heat equation*, Master's thesis, University of Bergen, 2000.

[13] B. Minchev, *Some algorithms for solving special tridiagonal block teoplitz linear systems*, J. Comp. and Appl. Math. **156(1)** (2003), 179–200.

[14] ———, *Exponential integrators for semilinear problems*, Ph.D. thesis, University of Bergen, 2004.

[15] H. Munthe-Kaas, *Runge-Kutta methods on Lie groups*, BIT **38** (1998), 92–111.

[16] ———, *High order Runge-Kutta methods on manifolds*, Appl. Num. Math. **29** (1999), 115–127.

[17] A. Suslowicz, *Application of numerical Lie group integrators to parabolic PDEs*, Technical report **219**, University of Bergen, 2001.

[18] L. N. Trefethen, *Spectral methods in* MATLAB, SIAM, 2000.