



# Optimization of Lie group methods for differential equations

S. Blanes\*, F. Casas

*Departament de Matemàtiques, Universitat Jaume I, 12071 Castellón, Spain*

## Abstract

In this paper we present a technique for reducing to a minimum the number of commutators required in the practical implementation of Lie group methods for integrating numerically matrix differential equations. This technique is subsequently applied to the linear and nonlinear case for constructing new geometric integrators, optimal with respect to the number of commutators.

© 2002 Elsevier Science B.V. All rights reserved.

*Keywords:* Lie group solvers; Magnus expansion; Geometric integration

## 1. Introduction

The aim of geometric integration is to build numerical methods for discretizing differential equations whilst preserving their known qualitative features, such as invariant quantities and the geometric structure. It is widely recognized that this class of numerical algorithms (the so-called *geometric integrators*) provides a better description of the original system than standard methods, both with respect to the preservation of invariants and in the accumulation of numerical errors along the evolution [5]. By sharing geometric structure and invariants with the exact solution, these methods provide numerical approximations which are more accurate and more stable for a significant class of differential equations, such as those evolving on Lie groups.

It is not difficult to show that every differential equation evolving on a matrix Lie group  $\mathcal{G}$  can be written in the form

$$Y' = A(t, Y)Y, \quad Y(t_0) = Y_0 \in \mathcal{G} \quad (1)$$

\* Corresponding author.

*E-mail addresses:* sblanes@mat.uji.es (S. Blanes),  
casas@mat.uji.es (F. Casas).

with  $A : [t_0, \infty[ \times \mathcal{G} \rightarrow \mathfrak{g}$  sufficiently smooth to ensure: (a) existence and uniqueness of solution and (b) the validity of discretization methods. Here  $\mathfrak{g}$  denotes the corresponding Lie algebra (the tangent space at the identity of  $\mathcal{G}$ ). Eq. (1) appears in relevant physical fields such as rigid mechanics ( $\mathcal{G} \equiv \text{SO}(n)$ ), Hamiltonian dynamics ( $\mathcal{G} \equiv \text{Sp}(n)$ ), Quantum Mechanics ( $\mathcal{G} \equiv \text{SU}(n)$ ), etc. Even more, the analysis of generic differential equations defined in homogeneous spaces can be reduced to the Lie group equation (1) [20], and thus, provided a discretization is done in  $\mathcal{G}$ , the structure of any homogeneous manifold acted upon by the group  $\mathcal{G}$  is also preserved. In other words, the Lie group solver can be extended to homogeneous spaces [18]. It is therefore of the greatest interest to design numerical schemes which preserve the most salient qualitative features of the system and at the same time are computationally as efficient as possible.

In this paper new Lie group solvers for Eq. (1) are presented which are optimal with respect to the number of commutators required. This reduction in the computational cost of the algorithms is achieved by analyzing the Lie algebraic structure of the problem. Both linear and nonlinear differential equations are considered.

## 2. Lie group solvers for linear and nonlinear differential equations

Although there are several exceptions, the technique used by Lie group methods for Eq. (1) is to lift  $Y(t)$  from  $\mathcal{G}$  to the underlying Lie algebra  $\mathfrak{g}$ , then formulate and solve there an associated differential equation and finally map the solution back to  $\mathcal{G}$ . In this way the discretization procedure operates in a linear space rather than in the Lie group. The lift of the flow  $Y(t)$  from the Lie group  $\mathcal{G}$  to the Lie algebra is usually done by the exponential map, although the Cayley transform can also be used with quadratic Lie groups [11,14].

To be more specific, we write

$$Y(t) = \exp(\Lambda(t))Y_0, \quad \Lambda(t) \in \mathfrak{g}, \tag{2}$$

so that Eq. (1) is transformed to the Lie algebra equation [12]

$$\Lambda' = d \exp_{\Lambda}^{-1} A(t, e^{\Lambda} Y_0), \quad \Lambda(t_0) = 0, \tag{3}$$

where

$$d \exp_{\Lambda}^{-1} C = \sum_{j=0}^{\infty} \frac{B_j}{j!} \text{ad}_{\Lambda}^j C. \tag{4}$$

Here  $B_j$  are Bernoulli numbers [1] and the adjoint operator  $\text{ad}_{\Lambda}$  is defined by iterated commutators

$$\text{ad}_{\Lambda}^0 C = C, \quad \text{ad}_{\Lambda}^j C = [\Lambda, \text{ad}_{\Lambda}^{j-1} C], \quad j \geq 1.$$

The Lie algebraic equation (3) can be solved by an explicit classical Runge–Kutta (RK) method. Then a Runge–Kutta–Munthe-Kaas (RKMK) scheme is obtained [18]. The following algorithm describes a single step from  $Y_n \in \mathcal{G}$ , the numerical approximation to  $Y(t = t_n)$ , to  $Y_{n+1} \in \mathcal{G}$  for an RKMK method based on an  $s$ -stage RK integration scheme:

$$\begin{aligned} \Lambda_k &= \sum_{l=1}^s a_{k,l} F_l, & \Lambda_k &= hA(t_n + c_k h, e^{\Lambda_k} Y_n) \\ F_k &= d \exp_{\Lambda_k}^{-1}(A_k), & \text{for } k &= 1, \dots, s, \\ \Lambda &= \sum_{l=1}^s b_l F_l, & Y_{n+1} &= \exp(\Lambda)Y_n. \end{aligned} \tag{5}$$

In this algorithm a practical means for computing  $F_k = d \exp_{\Lambda_k}^{-1}(A_k)$  has to be provided. In general, the expansion (4) may be truncated to the order  $p$  of the

underlying RK method and the resulting scheme will get this same order  $p$ , while evolving on the Lie group  $\mathcal{G}$ . In any case, a significant number of commutators must be evaluated if  $F_k$  is computed with this procedure.

When the Lie group equation (1) is linear, i.e.,

$$Y' = A(t)Y, \quad Y(t_0) = Y_0 \in \mathcal{G}, \tag{6}$$

a specially well suited approach for obtaining approximate solutions is to make use of the so-called *Magnus expansion* [16]. In essence, the idea is to write the flow as in (2)

$$Y(t) = \exp(\Omega(t))Y_0, \quad \Omega(t) \in \mathfrak{g},$$

but now  $\Omega(t)$  is expressed as an infinite series

$$\Omega(t) = \sum_{k=1}^{\infty} \Omega_k(t),$$

whose terms are linear combinations of integrals and nested commutators involving the matrix  $A$  at different times. Thus, the first terms read

$$\begin{aligned} \Omega_1(t) &= \int_{t_0}^t A(t_1) dt_1, \\ \Omega_2(t) &= \frac{1}{2} \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 [A_1, A_2], \\ \Omega_3(t) &= \frac{1}{6} \int_{t_0}^t dt_1 \int_{t_0}^{t_1} dt_2 \\ &\quad \times \int_{t_0}^{t_2} dt_3 ([A_1, [A_2, A_3]] + [A_3, [A_2, A_1]]), \end{aligned}$$

where  $A_i \equiv A(t_i)$ . Explicit formulae for  $\Omega_k$  of all orders have been given in [10] by using graph theory, whereas in [13] a recursive procedure for the generation of  $\Omega_k$  was proposed, which allows to establish absolute convergence for  $t$  values which satisfy [2]

$$\int_{t_0}^t \|A(s)\| ds < \xi = 1.086869.$$

The first analysis of the Magnus expansion as a numerical procedure for integrating matrix differential equations in Lie groups was given by Iserles and Nørsett [10]. They showed how  $\Omega_k, k > 1$ , can be approximated in terms of nested commutators of  $A(t_{i_k})$  at different nodes  $t_{i_k} \in [t_0, t_0 + h]$ ,  $h$  being the time step,

$$\Omega_k \simeq h^k \sum_{1 \leq i_1, i_2, \dots, i_k \leq N} \beta_{i_1, i_2, \dots, i_k} [A(t_{i_1}), [A(t_{i_2}), \dots, [A(t_{i_{k-1}}), A(t_{i_k})]]]. \tag{7}$$

Then, in terms of matrix evaluations, the cost of all the multivariate quadratures needed to approximate the Magnus expansion to a given order is the same as the cost of the single quadrature formula for  $\Omega_1$ . Nevertheless, the coefficients  $\beta_{i_1, i_2, \dots, i_k}$  in (7) have to satisfy an extraordinarily large system of linear equations for  $k \geq 3$  and consequently the number of commutators required grows rapidly with the order. On the other hand, time-symmetry of the expansion allows to conclude that for achieving a  $2n$ th ( $n > 1$ ) order integration method only terms up to  $\Omega_{2n-2}$  in the  $\Omega$  series have to be considered [3].

Generally speaking, there are two critical factors in the computational cost of the Lie group solvers we are considering here. First, the evaluation of the exponential map from the Lie algebra  $\mathfrak{g}$  into the Lie group  $\mathcal{G}$ . Second, the computation of the commutators involved in the algorithms.

Concerning the first aspect, several approximation schemes have been designed such that the outcome lies in the correct Lie group  $\mathcal{G}$  and differs from the exact solution in a way that is consistent with the order of integration. For more details, the reader is referred to [6,7,12].

With respect to the second factor, different strategies have been analyzed to reduce the total number of commutators. In particular, the theory of graded free Lie algebras (FLAs) allows to obtain an upper bound on the number of linearly independent terms required for a method of order  $n$ , in particular on the commutators involved [19]. The available theory, however, does not fix the *least* number of commutators required for a method of a given order.

In the next section we present a systematic procedure for reducing to a minimum the number of commutators required by a generic Lie group solver. We should remark that, for  $N \times N$  matrices,  $2N^3$  operations are needed for evaluating one commutator. Thus, reducing to a minimum this number of commutators is of the greatest importance for an efficient study of far-reaching physical problems with this class of algorithms [9,15,23].

### 3. Optimization technique in a graded FLA

In the general analysis of Lie group methods for integrating differential equations it is quite useful the

notion of an FLA [19,22]. This, intuitively, is a Lie algebra with no specific assumptions about its structure, except for what is common to all Lie algebras. In other words, it is a Lie algebra whose terms are generated by commutators of pairwise elements and such that the only reducing mechanisms are skew-symmetry and the Jacobi identity. Thus, the results obtained in this setting can be applied afterwards in any concrete Lie algebra.

Let us consider an FLA  $\mathcal{L}_B$  generated by the set  $B = \{b_1, b_2, \dots, b_s\}$ . We introduce a grading function  $w$  on  $\mathcal{L}_B$  simply by assigning a grade to the generators,  $w(b_l) = l, l = 1, \dots, s$ , and then propagating the grades throughout  $\mathcal{L}_B$  by additivity,  $w([u, v]) = w(u) + w(v)$ . In the context of Lie group solvers, usually  $b_l = O(h^l)$  so that, intuitively, the grade provides information about the size of each term in  $\mathcal{L}_B$ . It is also useful to consider in  $\mathcal{L}_B$  a basis, in the sense of linear algebra. For example, in Table 1 we give a basis of nested commutators (including elements up to grade 6) when  $B = \{b_1, b_2, b_3\}$ . Then we can formulate the following general question.

**Problem 1.** Given an element  $Z \in \mathcal{L}_B$  of the form

$$Z = \sum_{i=1}^{2s} \sum_{j=1}^{v_i} \alpha_{i,j} X_{i,j}, \tag{8}$$

where  $X_{i,j}$  denotes the  $j$ th element of the basis of the FLA of grade  $i$ , obtain an approximate expression for  $Z$  up to grade  $2s$  involving the minimum number of commutators.

The procedure to solve this problem is in principle very simple, but its technical complexity increases

Table 1  
Basis of the Lie algebra,  $\mathcal{L}_B$ , generated by  $B = \{b_1, b_2, b_3\}$  up to grade 6.  $[i, j, \dots, k, l]$  represents the nested commutator  $[b_i, [b_j, [\dots, [b_k, b_l], \dots]]]$

$n$	$v_n$	$\mathcal{L}_B$
1	1	$X_{1,1} = b_1$
2	1	$X_{2,1} = b_2$
3	2	$X_{3,1} = b_3, X_{3,2} = [12]$
4	2	$X_{4,1} = [13], X_{4,2} = [112]$
5	4	$X_{5,1} = [113], X_{5,2} = [1112],$ $X_{5,3} = [23], X_{5,4} = [212]$
6	5	$X_{6,1} = [1113], X_{6,2} = [11112],$ $X_{6,3} = [123], X_{6,4} = [1212], X_{6,5} = [312]$

dramatically with  $s$ . First, we take the most general commutator we can build with  $\{b_1, \dots, b_s\}$

$$d_1 = \left[ \sum_{i=1}^s x_{1,i} b_i, \sum_{j=1}^s y_{1,j} b_j \right].$$

Next, we write the most general element one can form with  $\{b_1, \dots, b_s, d_1\}$ ,

$$d_2 = \left[ \sum_{i=1}^s x_{2,i} b_i + x_{2,s+1} d_1, \sum_{j=1}^s y_{2,j} b_j + y_{2,s+1} d_1 \right],$$

and we repeat the procedure recursively  $2s - 2$  times to reproduce the term

$$\underbrace{[b_1, b_1, \dots, b_1, b_2]}_{2s-2 \text{ times}}$$

if its corresponding coefficient  $\alpha_{i,j} \neq 0$ . The problem is reduced then to determine the coefficients  $x_{i,j}, y_{m,n}, \alpha_i, \beta_j$  such that

$$Z = \sum_{i=1}^s \alpha_i b_i + \sum_{i=1}^{2s-2} \beta_i d_i + \Theta(2s + 1),$$

where  $\Theta(k)$  represents terms in  $\mathcal{L}_B$  of grade  $k$  or higher. Some remarks are in order. First, a nonlinear system of algebraic equations in  $x_{i,j}, y_{m,n}, \alpha_i, \beta_j$  has to be solved in the process, and there is no guarantee at all that this system has real solutions. In this case, new commutators  $d_l$  have to be introduced. Second, if there are real solutions then the minimum number of commutators required is precisely  $2s - 2$ . Third, although some of the coefficients are redundant, the number of variables to be determined increases rapidly with the grade and so does the difficulty of the problem [4]. It is, therefore, of primary interest to take into account whatever symmetry properties it has. In particular, if  $\alpha_{2i,l} = 0$  in (8), we can split  $\mathcal{L}_B$  as  $\mathcal{L}_B = \mathcal{S} \oplus \mathcal{R}$  with

$$\mathcal{S} = \left\{ U \in \mathcal{L}_B : U = \sum_{i=1}^s \sum_{j=1}^{v_i} \alpha_{2i-1,j} X_{2i-1,j} \right\},$$

$$\mathcal{R} = \left\{ U \in \mathcal{L}_B : U = \sum_{i=1}^s \sum_{j=1}^{v_i} \alpha_{2i,j} X_{2i,j} \right\},$$

and  $Z \in \mathcal{S}$ . Now

$$d_1 = [x_{1,1} b_1 + x_{1,3} b_3 + \dots, y_{1,2} b_2 + y_{1,4} b_4 + \dots] \in \mathcal{S},$$

$$d_2 = [x_{2,1} b_1 + x_{2,3} b_3 + \dots + x_{2,s_1} d_1, y_{2,1} b_1 + y_{2,3} b_3 + \dots + y_{2,s_1} d_1] \in \mathcal{R},$$

$$d_3 = [x_{3,1} b_1 + x_{3,3} b_3 + \dots + x_{3,s_1} d_1, y_{3,2} b_2 + y_{3,4} b_4 + \dots + y_{3,r_1} d_2] \in \mathcal{S},$$

and so on. The most general combination which reproduces  $Z \in \mathcal{S}$  with the minimum number of commutators is then

$$Z = \alpha_1 b_1 + \alpha_3 b_3 + \dots + \beta_1 d_1 + \beta_3 d_3 + \dots \in \mathcal{S}.$$

**Example 1.** Let us illustrate this technique by obtaining an approximation of  $Z \in \mathcal{S}$  up to grade 6 with three generators  $\{b_1, b_2, b_3\}$ . From Table 1

$$Z = \alpha_1 b_1 + \alpha_2 b_3 + \alpha_3 [12] + \alpha_4 [23] + \alpha_5 [212] + \alpha_6 [113] + \alpha_7 [1112]. \tag{9}$$

Observe that, due to the symmetry, the term  $X_{6,2}$  does not appear in  $Z$  and thus, in principle, three commutators (instead of four) will be needed if  $\alpha_7 \neq 0$ . If we consider, for instance,  $s_1 = [12] \in \mathcal{S}, r_1 = [b_1, x_1 b_3 + x_2 s_1] \in \mathcal{R}, s_2 = [x_3 b_1 + x_4 b_3 + x_5 s_1, b_2 + r_1] \in \mathcal{S}$  and choose  $x_1 = \alpha_6/\alpha_3, x_2 = \alpha_7/\alpha_3, x_3 = \alpha_3, x_4 = -\alpha_4, x_5 = -\alpha_5$  (if  $\alpha_3 \neq 0$ ) then  $Z = \alpha_1 b_1 + \alpha_2 b_3 + s_2 + \Theta(7)$ . If  $\alpha_3 = 0$  we can choose  $s_1, r_1$  and  $s_2$  in a slightly different way.

## 4. Application to linear differential equations

### 4.1. Homogeneous equations

The Lie group equation (6) may be considered to be associated with the linear homogeneous system of the  $N$ th order

$$y' = A(t)y, \quad y(t_0) = y_0 \tag{10}$$

with  $y \in \mathbb{R}^N$ , in the sense that  $y(t) = Y(t)y_0$  is a solution of (10) if  $Y(t)$  satisfies (6) with the identity matrix as initial condition.

A possible strategy to obtain a numerical integrator from the Magnus expansion is to choose  $s$  distinct quadrature nodes  $0 \leq c_1 < c_2 < \dots <$

$c_s \leq 1$  symmetric with respect to  $1/2$ , evaluate  $A_k \equiv hA(t_0 + c_k h)$ ,  $k = 1, \dots, s$ , and substitute the corresponding interpolating polynomial in the recursive scheme for computing  $\Omega_k$ . If the function values  $A_1, A_2, \dots, A_s$  are replaced by their linear combinations  $b_1, b_2, \dots, b_s$  that solve the Vandermonde system

$$\sum_{j=1}^s \left(c_k - \frac{1}{2}\right)^{j-1} b_j = A_k, \quad k = 1, 2, \dots, s \quad (11)$$

then  $b_i = [h^i / (i-1)!] A^{(i-1)}(t_0 + (h/2))$ . We can consider the graded FLA generated by  $B = \{b_1, \dots, b_s\}$  with grades  $1, 2, \dots, s$ , respectively, write the Magnus expansion in this algebra and finally use (11) to obtain the corresponding numerical method in terms of  $A_k$  [12]. With this procedure it is possible to build methods of order  $2s$  with only  $s$  symmetric collocation points. In particular, up to order 6 we obtain in terms of  $\{b_1, b_2, b_3\}$  [4]

$$\Omega = b_1 + \frac{1}{12}b_3 - \frac{1}{12}[12] + \frac{1}{240}[23] + \frac{1}{360}[113] - \frac{1}{240}[212] + \frac{1}{720}[1112] + O(h^7),$$

i.e., an expression similar to Eq. (9) with

$$\alpha_1 = 1, \quad \alpha_2 = -\alpha_3 = \frac{1}{12}, \quad \alpha_4 = -\alpha_5 = \frac{1}{240}, \\ \alpha_6 = \frac{1}{360}, \quad \alpha_7 = \frac{1}{720},$$

so that a sixth-order scheme for Eq. (6) reads as follows for  $t_{n+1} = t_n + h$ :

$$C_1 = [b_1, b_2], \quad C_2 = [b_1, 2b_3 + C_1], \\ C_3 = [-20b_1 - b_3 + C_1, b_2 - \frac{1}{60}C_2], \quad (12)$$

$$\Omega^{[6]}(h) = b_1 + \frac{1}{12}b_3 + \frac{1}{240}C_3, \\ Y(t_{n+1}) \approx \exp(\Omega^{[6]})Y(t_n).$$

In terms of Gauss–Legendre collocation points

$$A_1 = hA(t_n + (\frac{1}{2} - \frac{1}{10}(\sqrt{15}))h), \\ A_2 = hA(t_n + \frac{1}{2}h), \\ A_3 = hA(t_n + (\frac{1}{2} + \frac{1}{10}(\sqrt{15}))h),$$

we have

$$b_1 = A_2, \quad b_2 = \frac{1}{3}(\sqrt{15})(A_3 - A_1), \\ b_3 = \frac{10}{3}(A_3 - 2A_2 + A_1), \quad (13)$$

whereas if the  $A(t)$  matrix is known only at equispaced points,  $A_i = hA(t_n + (ih/4))$ ,  $i = 0, \dots, 4$ , then

$$b_1 = \frac{1}{60}(-7(A_0 + A_4) + 28(A_1 + A_3) + 18A_2), \\ b_2 = \frac{1}{15}(7(A_4 - A_0) + 16(A_3 - A_1)), \\ b_3 = \frac{1}{3}(7(A_0 + A_4) - 4(A_1 + A_3) - 6A_2). \quad (14)$$

This method involves the minimum number of commutators (three) and requires three or four  $A(t)$  matrix evaluations per step. Higher orders can be treated in a similar way. For instance, an eighth-order Magnus method has been obtained with only six commutators [4].

#### 4.2. Nonhomogeneous equations

As a matter of fact, the Magnus expansion can also be applied to solve numerically the nonhomogeneous problem

$$y' = A(t)y + g(t), \quad y(t_0) = y_0, \quad (15)$$

where  $g$  is a continuous vector on the time interval which is not identically zero there. Our objective is to design numerical integrators for the initial value problem (15) based on algorithm (12) requiring the lowest computational effort. The resulting numerical scheme will preserve whatever qualitative properties system (15) has.

As is well known, the exact solution of (15) is given by

$$y(t) = Y(t) \left( y_0 + \int_{t_0}^t Y^{-1}(s)g(s) ds \right). \quad (16)$$

If the flow  $Y(t)$  is computed with the (exact or truncated at a sufficiently high order) Magnus expansion, an approximation to (16) at time  $t_{n+1} = t_n + h$  is given by

$$y(t_{n+1}) = e^{\Omega(t_{n+1})} \left( y(t_n) + \int_{t_n}^{t_n+h} F(s) ds \right), \quad (17)$$

where  $F(s) \equiv e^{-\Omega(s)}g(s)$ . To obtain a sixth-order method,  $\Omega(t_{n+1})$  is substituted by  $\Omega^{[6]}(h)$  as given by (12) and the integral appearing in (17) is computed up to this order of approximation. If we use again equispaced points in  $[t_n, t_n + h]$ , then

$$I \equiv \int_{t_n}^{t_n+h} F(s) ds \\ = \frac{h}{90}(7(F_0 + F_4) + 32(F_1 + F_3) + 12F_2) + O(h^7)$$

with  $F_i = F(t_n + (ih/4))$ ,  $i = 0, \dots, 4$  [1]. In more detail,

$$I = \frac{1}{90}h(7g(t_n) + 7e^{-\Omega(t_{n+1})}g(t_{n+1}) + 32e^{-\Omega(t_n+h/4)}g(t_n + \frac{1}{4}h) + 32e^{-\Omega(t_n+3h/4)}g(t_n + \frac{3}{4}h) + 12e^{-\Omega(t_n+h/2)}g(t_n + \frac{1}{2}h)) + O(h^7),$$

so that we only have to obtain approximations for  $\exp(\Omega(t_n + h/4))$ ,  $\exp(\Omega(t_n + h/2))$ ,  $\exp(\Omega(t_n + 3h/4))$ , i.e., for  $\Omega(t_n+h/4)$ ,  $\Omega(t_n+h/2)$ ,  $\Omega(t_n+3h/4)$  up to order 5 (because of the factor  $h$  of the quadrature). We denote them by  $\Omega_{h/4}^{[5]}$ ,  $\Omega_{h/2}^{[5]}$  and  $\Omega_{3h/4}^{[5]}$ , respectively.

This can be done by inserting into the Magnus series  $\Omega(t) = \sum_{i \geq 1} \Omega_i$  the Taylor expansion of  $A(t)$  around  $t = t_n$ ,  $A(t) = \sum_{i \geq 0} a_i(t - t_n)^i$ . In particular, we get for  $t = t_n + (h/4)$ ,

$$\begin{aligned} \Omega_1 &= \frac{1}{4}ha_0 + \frac{1}{32}h^2a_1 + \frac{1}{192}h^3a_2 \\ &\quad + \frac{1}{1024}h^4a_3 + \frac{1}{5120}h^5a_4, \\ \Omega_2 &= -\frac{1}{768}h^3[a_0, a_1] - \frac{1}{3072}h^4[a_0, a_2] \\ &\quad + h^5(-\frac{3}{40960}[a_0, a_3] - \frac{1}{61440}[a_1, a_2]), \\ \Omega_3 &= h^5(\frac{1}{368640}[a_0, [a_0, a_2]] - \frac{1}{245760}[a_1, [a_0, a_1]]), \\ \Omega_4 &= \frac{1}{737280}h^5[a_0, [a_0, [a_0, a_1]]] \end{aligned}$$

with error  $O(h^6)$  ( $\Omega_5 = O(h^7)$ ). Now the expression of  $\Omega_1$  is reproduced with the linear combination  $\omega_{1,h/4} \equiv h \sum_{i=0}^4 \alpha_i A_i$  if

$$\begin{aligned} \alpha_{10} &= \frac{251}{2880}, & \alpha_{11} &= \frac{323}{1440}, \\ \alpha_{12} &= -\frac{11}{120}, & \alpha_{13} &= \frac{53}{1440}, & \alpha_{14} &= -\frac{19}{2880}, \end{aligned}$$

whereas we try to get the relevant terms of  $\Omega_2, \Omega_3, \Omega_4$  with an expression of the form  $\alpha_{21}C_1 + \alpha_{22}C_2 + \alpha_{23}C_3$  ( $C_i$  given by (12)) to minimize the computational effort. The corresponding system of equations does not admit solutions, however, so that at least one additional commutator is required:

$$\begin{aligned} \Omega_2 + \Omega_3 + \Omega_4 &= \alpha_{21}C_1 + \alpha_{22}C_2 + \alpha_{23}C_3 \\ &\quad + h^2[A_0, \alpha_{24}A_1 + \alpha_{25}A_4] + O(h^6). \end{aligned}$$

A possible solution is given by

$$\begin{aligned} \alpha_{21} = \alpha_{22} = 0, & & \alpha_{23} &= \frac{1}{1024}, \\ \alpha_{24} = -\frac{1}{192}, & & \alpha_{25} &= \frac{1}{12288}, \end{aligned}$$

and finally

$$\begin{aligned} \Omega_{h/4}^{[5]} &= \omega_{1,h/4} + \frac{1}{1024}C_3 + C_{4,h/4}, \\ \omega_{1,h/4} &= h(\frac{251}{2880}A_0 + \frac{323}{1440}A_1 - \frac{11}{120}A_2 \\ &\quad + \frac{53}{1440}A_3 - \frac{19}{2880}A_4), \\ C_{4,h/4} &= h^2[A_0, -\frac{1}{192}A_1 + \frac{1}{12288}A_4]. \end{aligned} \tag{18}$$

Proceeding in a similar way, we get for  $\Omega_{h/2}^{[5]}$

$$\begin{aligned} \Omega_{h/2}^{[5]} &= \omega_{1,h/2} + \frac{1}{32}C_3 + C_{4,h/2}, \\ \omega_{1,h/2} &= h(\frac{29}{360}A_0 + \frac{31}{90}A_1 + \frac{1}{15}A_2 + \frac{1}{90}A_3 - \frac{1}{360}A_4), \\ C_{4,h/2} &= h^2[A_0, -\frac{1}{48}A_2 + \frac{1}{384}A_4]. \end{aligned} \tag{19}$$

The last approximation needed is

$$\begin{aligned} \Omega_{3h/4}^{[5]} &= \omega_{1,3h/4} + \frac{243}{1024}C_3 + C_{4,3h/4}, \\ \omega_{1,3h/4} &= h(\frac{27}{320}A_0 + \frac{51}{160}A_1 + \frac{9}{40}A_2 + \frac{21}{160}A_3 - \frac{3}{320}A_4), \\ C_{4,3h/4} &= \frac{27}{8}C_{4,h/2} - 9C_{4,h/4}, \end{aligned} \tag{20}$$

so that only two additional commutators are required. The numerical approximation  $y_{n+1}$  to  $y(t_{n+1})$  is finally given by

$$\begin{aligned} y_{n+1} &= \frac{7}{90}h g_4 + e^{\Omega^{[6]}(h)} \\ &\quad \times (y_n + \frac{1}{90}h(7g_0 + 32e^{-\Omega_{h/4}^{[5]}}g_1 \\ &\quad + 12e^{-\Omega_{h/2}^{[5]}}g_2 + 32e^{-\Omega_{3h/4}^{[5]}}g_3)), \end{aligned} \tag{21}$$

where  $g_i \equiv g(t_n + (ih/4))$  and  $\Omega^{[6]}(h)$  is obtained with Eqs. (12) and (14).

The additional cost for solving the nonhomogeneous linear problem with this algorithm is, essentially, two commutators and three matrix exponential–vector products. No additional  $A(t)$  matrix evaluations are required. Although only a Newton–Cotes quadrature formula has been considered here, it is clear that the procedure can also be carried out with other quadrature rules, possibly involving matrix evaluations at different steps along the integration. Algorithm (21) could be useful for the class of problems analyzed in [23].

### 5. Application to nonlinear differential equations in Lie groups

In this section we examine how the optimization technique for reducing the number of commutators can also be applied to the RKMK class of Lie group solvers for the general equation  $Y' = A(t, Y)Y$ .

By combining the stage values  $A_k$  computed by the algorithm (5), linear combinations  $Q_k$  of the highest possible order in the step size  $h$  (i.e., of the highest possible grade) are obtained. In order to build an explicit method it is required that  $Q_k$  are related to  $A_k$  by a triangular matrix. Given a particular RK scheme, optimal combinations are found in [19] by using the theory of order conditions for RK methods. As a result, new versions of the RKMK algorithm (5) based on the classical fourth-order RK scheme and the fifth-order method DOPRI5 are constructed involving two and 12 commutators, respectively, instead of six and 24 in the original implementation [19].

In particular, for the much used fifth-order method DOPRI5, which has a total of seven stages (the seventh stage is used only for error estimation) [8], the modified RKMK algorithm proposed in [19] reads as follows:

```
do  $i = 1, 7$ 
   $u_i$  given by (23)
   $A_i = hA(t_n + c_i h, e^{u_i} Y_n)$ 
   $Q_i$  given by (22)
end do
 $Y_{n+1} = e^{u_7} Y_n$ 
```

with

$$\begin{aligned} Q_1 &= A_1 = O(h), & Q_2 &= A_2 - A_1 = O(h^2), \\ Q_3 &= A_3 - \frac{3}{2}A_2 + \frac{1}{2}A_1 = O(h^3), \\ Q_4 &= A_4 - 6A_3 + 5A_2 = O(h^3), \\ Q_5 &= A_5 - \frac{106}{81}A_4 + \frac{128}{243}A_3 - \frac{53}{243}A_1 = O(h^4), \\ Q_6 &= A_6 - \frac{567}{212}A_5 + \frac{7}{4}A_4 - \frac{4}{53}A_3 = O(h^4), \\ Q_7 &= A_7 - \frac{176}{105}A_6 + \frac{17253}{8480}A_5 - \frac{71}{48}A_4 \\ &\quad + \frac{568}{3339}A_3 - \frac{71}{1440}A_1 = O(h^5), \end{aligned} \tag{22}$$

and

$$\begin{aligned} u_1 &= 0, & u_2 &= \frac{1}{5}Q_1, \\ u_3 &= \frac{3}{10}Q_1 + \frac{9}{40}Q_2 - \frac{9}{400}[Q_1, Q_2] + \frac{3}{4000}[Q_1, [Q_1, Q_2]], \end{aligned}$$

$$\begin{aligned} u_4 &= \frac{4}{5}Q_1 + \frac{8}{5}Q_2 + \frac{32}{9}Q_3 - \frac{2}{75}[Q_1, Q_2] \\ &\quad - \frac{8}{15}[Q_1, Q_3] - \frac{73}{2250}[Q_1, [Q_1, Q_2]], \\ u_5 &= \frac{8}{9}Q_1 + \frac{160}{81}Q_2 + \frac{53000}{6561}Q_3 - \frac{212}{729}Q_4 \\ &\quad + \frac{628}{2187}[Q_1, Q_2] - \frac{8480}{6561}[Q_1, Q_3] \\ &\quad + \frac{424}{3645}[Q_1, Q_4] - \frac{3971}{32805}[Q_1, [Q_1, Q_2]], \\ u_6 &= Q_1 + \frac{5}{2}Q_2 + \frac{3395}{396}Q_3 - \frac{7}{88}Q_4 - \frac{5103}{18656}Q_5 \\ &\quad + \frac{4}{33}[Q_1, Q_2] - \frac{455}{264}[Q_1, Q_3] + \frac{7}{80}[Q_1, Q_4] \\ &\quad - \frac{1103}{7920}[Q_1, [Q_1, Q_2]], \\ u_7 &= Q_1 + \frac{5}{2}Q_2 + \frac{115}{36}Q_3 + \frac{11}{24}Q_4 + \frac{189}{6784}Q_5 \\ &\quad + \frac{11}{84}Q_6 - \frac{5}{12}[Q_1, Q_2] - \frac{55}{72}[Q_1, Q_3] \\ &\quad - \frac{7}{48}[Q_1, Q_4] - \frac{675}{13568}[Q_1, Q_5] - \frac{11}{168}[Q_1, Q_6] \\ &\quad - \frac{25}{36}[Q_2, Q_3] - \frac{5}{24}[Q_2, Q_4] \\ &\quad + \frac{5}{216}[Q_1, [Q_1, Q_3]] + \frac{1}{144}[Q_1, [Q_1, Q_4]] \\ &\quad - \frac{5}{48}[Q_2, [Q_1, Q_2]] + \frac{1}{144}[Q_1, [Q_1, [Q_1, Q_2]]]. \end{aligned} \tag{23}$$

Here the coefficients  $c_i$  are [8]  $c_1 = 0, c_2 = 1/5, c_3 = 3/10, c_4 = 4/5, c_5 = 8/9, c_6 = c_7 = 1$ , so that, as can easily be observed, at most 12 commutators are involved. We should mention that in [19] there are some misprints in the coefficients of  $u_5$  and  $u_6$  [21].

Although in this case the grade of the generators  $w(Q_i) \neq l$ , it is possible to generalize the strategy developed in Section 3 to this case and reduce the number of commutators to a minimum. Next we show that, in fact, this minimum number is 5.

We commence by observing that  $u_3$  requires the evaluation of  $[Q_1, Q_2]$  and  $[Q_1, [Q_1, Q_2]]$ . In  $u_4$ , we have the commutator  $[Q_1, Q_3]$ , which cannot be obtained from  $u_3$ . In a similar way, we need to incorporate the commutator  $[Q_1, Q_4]$ , which appears in  $u_5$ , whereas  $u_6$  does not require additional commutators. Thus, four commutators are needed to determine the intermediate steps of the algorithm and the commutator  $[Q_1, [Q_1, [Q_1, Q_2]]]$  (not previously available) is present in  $u_7$ . In consequence, 5 is the minimum number required.

Let us introduce the commutators

$$\begin{aligned} C_1 &= [Q_1, Q_2], & C_2 &= [Q_1, C_1], \\ C_3 &= [Q_1, Q_3], & C_4 &= [Q_1, Q_4], \end{aligned} \tag{24}$$

and, to reproduce  $u_7$  up to order  $O(h^6)$ ,

$$C_5 = [a_1 Q_1 + a_2 Q_2, b_1 Q_3 + b_2 Q_4 + b_3 Q_5 + b_4 Q_6 + b_5 C_1 + b_6 C_2 + b_7 C_3 + b_8 C_4], \quad (25)$$

so that

$$u_7 = Q_1 + \frac{5}{2} Q_2 + \frac{115}{36} Q_3 + \frac{11}{24} Q_4 + \frac{189}{6784} Q_5 + \frac{11}{84} Q_6 + \sum_{i=1}^5 \gamma_i C_i. \quad (26)$$

After expanding  $C_5$ , equating terms with (23) and solving the corresponding equations we get

$$\begin{aligned} a_1 &= 1, & a_2 &= 1, & b_1 &= -\frac{25}{36}, & b_2 &= -\frac{5}{24}, \\ b_3 &= -\frac{675}{13568}, & b_4 &= -\frac{11}{168}, & b_5 &= -\frac{5}{48}, \\ b_6 &= \frac{1}{144}, & b_7 &= \frac{5}{216}, & b_8 &= \frac{1}{144}, \\ \gamma_1 &= -\frac{5}{12}, & \gamma_2 &= \frac{5}{48}, & \gamma_3 &= -\frac{5}{72}, \\ \gamma_4 &= \frac{1}{16}, & \gamma_5 &= 1, \end{aligned} \quad (27)$$

and thus the optimal (with respect to the number of commutators) fifth-order RKMK method based on DOPRI5 replaces  $u_7$  in (23) by (24)–(27).

## 6. Concluding remarks

In this work we have applied the optimization technique in a graded FLA exposed in Section 3 to minimize the number of commutators of Lie group solvers, both for linear and nonlinear differential equations.

In the first case we have considered a numerical scheme built from the Magnus expansion. It requires the evaluation of just one matrix exponential and therefore the optimization of only one element  $Z$  of the form (8). The time-symmetry of the Magnus expansion leads to further simplifications and the optimization is maximal: a sixth-order scheme has been constructed requiring the minimum number of commutators.

For the more general equation  $A' = A(t, Y)Y$ , the existing Lie group solvers (the RKMK or some variants of the Crouch–Grossman class of algorithms [17]) involve the computation of the exponential of several elements of  $\mathfrak{g}$  at the intermediate stages. Accordingly,

one has a sequence of elements  $Z_i$ ,  $i = 1, \dots, l$  in  $\mathcal{L}_B$  to approximate with the minimum number of commutators and, in addition, the evaluation of  $Z_1, \dots, Z_{i-1}$  is required for computing  $Z_i$ . This can impose severe restrictions on the optimization procedure. As a rule, the higher the total number of elements  $Z_i$  involved in the algorithm, the less significant is the reduction in the number of commutators in the corresponding optimal expressions.

## Acknowledgements

This work has been partially supported by Fundació Caixa Castelló-Bancaixa under project PIB99-09 (1999).

## References

- [1] M. Abramowitz, I. Stegun, Handbook of Mathematical Functions, Dover, New York, 1965.
- [2] S. Blanes, F. Casas, J.A. Oteo, J. Ros, Magnus and Fer expansions for matrix differential equations: the convergence problem, *J. Phys. A* 31 (1998) 259–268.
- [3] S. Blanes, F. Casas, J. Ros, Improved high order integrators based on the Magnus expansion, *BIT* 40 (2000) 434–450.
- [4] S. Blanes, F. Casas, J. Ros, High order optimized geometric integrators for linear differential equations, *BIT* 42 (2002) 262–284.
- [5] C.J. Budd, A. Iserles, Geometric integration: numerical solution of differential equations on manifolds, *Philos. Trans. R. Soc. London Ser. A* 357 (1999) 945–956.
- [6] E. Celledoni, A. Iserles, Approximating the exponential from a Lie algebra to a Lie group, *Math. Comp.* 69 (2000) 1457–1480.
- [7] E. Celledoni, A. Iserles, Methods for the approximation of the matrix exponential in a Lie-algebraic setting, *IMA J. Numer. Anal.* 21 (2001) 463–488.
- [8] J.R. Dormand, Numerical Methods for Differential Equations: A Computational Approach, CRC Press, Boca Raton, FL, 1996.
- [9] M. Hochbruck, C. Lubich, On Magnus integrators for time-dependent Schrödinger equations, Technical Report GIPS 2002-004, *SIAM J. Numer. Anal.*, submitted for publication.
- [10] A. Iserles, S.P. Nørsett, On the solution of linear differential equations in Lie groups, *Philos. Trans. R. Soc. London Ser. A* 357 (1999) 983–1019.
- [11] A. Iserles, On Cayley-transform methods for the discretization of Lie group equations, *Found. Comput. Math.* 1 (2001) 129–160.

- [12] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, A. Zanna, Lie group methods, *Acta Numer.* 9 (2000) 215–365.
- [13] S. Klarsfeld, J.A. Oteo, Recursive generation of higher-order terms in the Magnus expansion, *Phys. Rev. A* 39 (1989) 3270–3273.
- [14] L. Lopez, T. Politi, Applications of the Cayley approach in the numerical solution of matrix differential systems on quadratic groups, *Appl. Numer. Math.* 36 (2001) 35–55.
- [15] A. López Ariste, On the asymmetry of Stokes profiles, *Astrophys. J.* 564 (2002) 379–384.
- [16] W. Magnus, On the exponential solution of differential equations for a linear operator, *Commun. Pure Appl. Math.* 7 (1954) 649–673.
- [17] A. Marthinsen, B. Owren, A note on the construction of Crouch–Grossman methods, *BIT* 41 (2001) 207–214.
- [18] H. Munthe-Kaas, High order Runge–Kutta methods on manifolds, *Appl. Numer. Math.* 29 (1999) 115–127.
- [19] H. Munthe-Kaas, B. Owren, Computations in a free Lie algebra, *Philos. Trans. R. Soc. London Ser. A* 357 (1999) 957–981.
- [20] H. Munthe-Kaas, A. Zanna, Numerical integration of differential equations on homogeneous manifolds, in: F. Cucker, M. Shub (Eds.), *Foundations of Computational Mathematics*, Springer, Berlin, 1997, pp. 305–315.
- [21] B. Owren, Private communication.
- [22] M. Postnikov, Lie groups and Lie algebras, Semester V of Lectures in Geometry, URSS Publishers, Moscow, 1994.
- [23] D. Viswanath, The Lindstedt–Poincaré technique as an algorithm for computing periodic orbits, *SIAM Rev.* 43 (2001) 478–495.



**S. Blanes** holds a Research position from the program *Ramón y Cajal* at the University Jaume I in Castellón (Spain). He was born on 11 April 1969, and he got his degree in theoretical physics in 1992 at the University of Valencia (Spain). His research interest is in geometric integration for differential equations, where his main contributions are in the search of more efficient splitting and composition

methods (most of them using a processor) and in the search for new methods based on the Magnus series for nonautonomous DEs. In this subject he got his PhD in 1998. He spent 6 months during 1995/1996 as a PhD student at the Dynamical Systems and Accelerator Theory Group of the University of Maryland (USA), and between 1999 and 2002 he spent, as a post-doc, 2 years at DAMTP (University of Cambridge, UK) and 1 year at the University of Bath (UK).



**F. Casas** is a Lecturer in Applied Mathematics at the University Jaume I in Castellón (Spain). He got his PhD in 1992 in theoretical physics. He has been a visiting research scholar in the Maryland Chaos Group at the University of Maryland (USA) in 1995 and the Numerical Analysis Group at the University of Cambridge (UK) in 2000. His main field of interest is geometric integration, particularly

the design and optimization of Lie group methods for the numerical integration of differential equations and the analysis of composition methods.