BIT
2002, Vol. 42, No. 2, pp. 262–284

0006-3835/02/4202-0262 $16.00

HIGH ORDER OPTIMIZED GEOMETRIC INTEGRATORS FOR LINEAR DIFFERENTIAL EQUATIONS *

S. BLANES[1] F. CASAS[2], and J. ROS[3] †

[1] *Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England. email: S.Blanes@damtp.cam.ac.uk*

[2] *Departament de Matemàtiques, Universitat Jaume I, 12071-Castellón, Spain. email: casas@mat.uji.es*

[3] *Departament de Física Teòrica and IFIC, Universitat de València, 46100-Burjassot, Valencia, Spain. email: Jose.Ros@uv.es.*

Abstract.

In this paper new integration algorithms based on the Magnus expansion for linear differential equations up to eighth order are obtained. These methods are optimal with respect to the number of commutators required. Starting from Magnus series, integration schemes based on the Cayley transform and the Fer factorization are also built in terms of univariate integrals. The structure of the exact solution is retained while the computational cost is reduced compared to similar methods. Their relative performance is tested on some illustrative examples.

AMS subject classification: 65L05, 65L70, 65D30.

Key words: Geometric integrators, linear differential equations, initial value problems, Lie groups.

1 Introduction.

In recent years there has been a renewed interest in designing numerical schemes for solving the linear matrix differential equation

$$\frac{dX}{dt} = A(t)X, \quad X(t_0) = X_0, \tag{1.1}$$

particularly in the context of geometric integration. The main goal in this field is to discretize equation (1.1) in such a way that important geometric and qualitative properties of the exact solution are retained by the numerical approximation. Here $A(t)$ stands for a sufficiently smooth matrix to ensure existence and uniqueness of an n-by-n matrix solution $X(t)$.

*Received July 2000. Revised May 2001. Communicated by Syvert Nørsett.

†S. Blanes acknowledges the Ministerio de Educación y Cultura (Spain) for a post-doctoral fellowship and the University of Cambridge for its hospitality. F. Casas is supported by the Collaboration Program UJI–Fundació Caixa Castelló 1999 under project 0I039.01/1 and by a grant of the MEC (Spain). J. Ros is supported by DGICyT, Spain (Grant no. PB97/1139).

It is generally recognised that geometric integrators provide a better description of the original system than standard integration algorithms, both in the preservation of invariant quantities and in the accumulation of numerical errors along the evolution [4]. For this reason there has been a systematic search of efficient geometric integrators for equation (1.1), especially when it evolves in Lie groups or a homogeneous space (see [12] for a review). In particular, the classical Magnus [14] and Fer [8] analytic expansions have been turned into very effective numerical methods, even for nonlinear matrix differential equations in Lie groups and homogeneous spaces [10, 20, 21], whereas other schemes based on the use of the Cayley transform [6, 11, 15] and (conveniently modified) Runge–Kutta methods [16] have also been proposed.

Although some preliminary analysis of complexity issues related to this class of integration algorithms are already available [5], it is clear that much work along this direction is still to be done. In particular, a detailed study is needed in order to clarify under which circumstances a particular method is preferable to others. This requires, as a first step, some optimization strategy to bring to a minimum the computational cost of the different geometric integrators, in particular by reducing the number of function evaluations and matrix operations (products and/or commutators) involved.

In a recent publication [2], the authors presented a technique that reduced the number of commutators required for methods based on the Magnus expansion up to order eight. Here that technique is improved and put on a sound basis. At the same time we extend the treatment to geometric integrators based on the Fer expansion and the Cayley transform. We show how this class of integration methods can be easily constructed from the Magnus expansion, both in terms of univariate analytic integrals and symmetric quadrature rules. We also discuss the use of diagonal Padé approximants to evaluate the matrix exponential and report on our experience when the new methods are applied to some illustrative examples.

Given the initial value problem (1.1), it is well known that the matrix solution X can, in a neighborhood of t_0, be written in the following forms:

$$
\begin{aligned}
(1.2)\qquad X(t_0+h) &= e^{\Omega(h)} X_0 && \text{Magnus}\\
(1.3)\qquad &= e^{F_1(h)} e^{F_2(h)} \cdots X_0 && \text{Fer}\\
(1.4)\qquad &= e^{S_1(h)} e^{S_2(h)} \cdots e^{S_2(h)} e^{S_1(h)} X_0 && \text{Symmetric Fer}\\
(1.5)\qquad &= \left(I - \tfrac{1}{2}C(h)\right)^{-1} \left(I + \tfrac{1}{2}C(h)\right) X_0 && \text{Cayley}
\end{aligned}
$$

and these expansions are convergent for sufficiently small values of h [1, 11, 21]. Observe that if $A(t)$ belongs to a Lie algebra $\mathfrak{g}$ then schemes (1.2)–(1.4) and the numerical integrators based on them provide approximate solutions staying in the corresponding Lie group G if $X_0 \in G$, whereas this is true for the Cayley transform (1.5) only when G is a J-orthogonal (also called quadratic) Lie group [18]. The choice of the representation (1.2)–(1.5) is dictated by the requirement that the solution stays in G. If on the other hand this requirement is abandoned one has a much larger class of representations to choose from.

The plan of the paper is the following: in Section 2 we tackle the problem of minimising the number of commutators involved in the methods by use of techniques of graded free Lie algebras. In Section 3 we collect some numerical integration schemes of order 4, 6 and 8 based on the Magnus expansion. These methods are used in Section 4 in combination with the Fer expansion and the Cayley transform to get new and optimized (or at least improved) geometric integrators for the differential equation (1.1) up to order 8. The idea is to write the functions F_i, S_i and C in terms of the successive approximations to Ω and, by using the technique of Section 2, to obtain methods which involve considerably fewer matrix commutators than other previously available. Also some considerations about the computational cost of the algorithms are incorporated in this section. In Section 5 we apply these methods to a couple of illustrative examples in order to compare their relative performance and elucidate when it is preferable to use one particular scheme to the others. Finally, Section 6 contains some conclusions.

2 Optimization in a graded free Lie algebra.

The usual approach to obtain a numerical method from the expansions (1.2)–(1.4) is to choose s distinct quadrature nodes $c_1, c_2, \ldots, c_s \in [0,1]$, evaluate $A_k = hA(t_0 + c_k h)$, $k = 1, \ldots, s$, and form the corresponding interpolating polynomial $\tilde{A}(t)$. If the points $c_1 < c_2 < \cdots < c_s$ are symmetric with respect to $\frac{1}{2}$ (as it is the case with Gauss–Legendre points) and the function values $A_1, A_2, \ldots, A_s$ are replaced by the solution of the Vandermonde system

$$\sum_{j=1}^{s} (c_k - \tfrac{1}{2})^{j-1} b_j = A_k, \quad k = 1, 2, \ldots, s, \tag{2.1}$$

then

$$b_i = \frac{h^i}{(i-1)!} A^{(i-1)}(t_0 + \tfrac{h}{2}) \tag{2.2}$$

where $A^{(i-1)}(t_0 + \frac{h}{2})$ is the $(i-1)$th derivative of A at the midpoint $t_0 + \frac{h}{2}$. We can consider then the graded free Lie algebra generated by $\mathcal{B} = \{b_1, \ldots, b_s\}$ with grades $1, 2, \ldots, s$ respectively, write expansions (1.2)–(1.4) in this algebra and finally use (2.1) to obtain the corresponding numerical method in terms of A_k [12].

As is well known, with this procedure it is possible to construct methods of order $2s$ with only s symmetric collocation points and obtain an upper bound on the number of linearly independent terms required, in particular on the number of commutators involved [17]. The available theory, however, does not fix the least number of commutators required for a method of a given order.

Here we approach this issue in the general framework of a graded free Lie algebra $L_{\mathcal{B}}$ generated by $\mathcal{B}$, with basis (including elements up to grade 7) given in Table 2.1. More specifically, we consider the following problem:

Given an element $Y \in L_{\mathcal{B}}$ of the form

$$Y = \sum_{i=1}^{2s} \sum_{j=1}^{\nu_i} \alpha_{i,j} X_{i,j}, \tag{2.3}$$

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

Table 2.1: Basis of the Lie algebra, $L_{\mathcal{B}}$, generated by $\mathcal{B} = \{b_1, b_2, b_3, b_4\}$ up to grade 7. $[ij \dots kl]$ represents the nested commutator $[b_i, [b_j, [\dots, [b_k, b_l] \dots]]]$.

n	ν_n	$L_{\mathcal{B}}$			
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	3	$X_{4,1} = b_4$;	$X_{4,2} = [13]$;	$X_{4,3} = [112]$;	
5	5	$X_{5,1} = [14]$;	$X_{5,2} = [113]$;	$X_{5,3} = [1112]$;	
		$X_{5,4} = [23]$;	$X_{5,5} = [212]$;		
6	7	$X_{6,1} = [114]$;	$X_{6,2} = [1113]$;	$X_{6,3} = [11112]$;	$X_{6,4} = [123]$;
		$X_{6,5} = [1212]$;	$X_{6,6} = [24]$;	$X_{6,7} = [312]$;	
7	14	$X_{7,1} = [34]$;	$X_{7,2} = [124]$;	$X_{7,3} = [223]$;	$X_{7,4} = [313]$;
		$X_{7,5} = [412]$;	$X_{7,6} = [1114]$;	$X_{7,7} = [1123]$;	$X_{7,8} = [1312]$;
		$X_{7,9} = [2113]$;	$X_{7,10} = [2212]$;	$X_{7,11} = [11113]$;	$X_{7,12} = [11212]$;
		$X_{7,13} = [21112]$;	$X_{7,14} = [111112]$;		

The procedure to solve this problem is in principle very simple, but its technical complexity grows extraordinarily with s.

First, we consider the most general commutator we can build with $\{b_1, \dots, b_s\}$:

$$d_1 = \Big[\sum_{i=1}^{s} x_{1,i}\, b_i, \ \sum_{j=1}^{s} y_{1,j}\, b_j \Big]. \tag{2.4}$$

Next, we write

$$d_2 = \Big[\sum_{i=1}^{s} x_{2,i}\, b_i + x_{2,s+1}\, d_1, \ \sum_{j=1}^{s} y_{2,j}\, b_i + y_{2,s+1}\, d_1 \Big], \tag{2.5}$$

i.e., the most general element one can form with $\{b_1, \dots, b_s, d_1\}$. The procedure can be repeated recursively $2s-2$ times in order to reproduce the term

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

and the problem is to determine the coefficients $x_{i,j}, y_{m,n}$, α_i, β_j such that

$$Y = \sum_{i=1}^{s} \alpha_i b_i + \sum_{i=1}^{2s-2} \beta_i d_i + \Theta(2s+1). \tag{2.6}$$

Here and in the rest of the paper $\Theta(k)$ represents terms in the Lie algebra of grade k or higher. At this point several observations are in order. First, a non-linear system of algebraic equations in $x_{i,j}, y_{m,n}$, α_i, β_j has to be solved in the process, and there is no guarantee at all that this system has real solutions. Second, if there are real solutions then the minimum number of commutators required is precisely $2s-2$. Third, the number of coefficients to be determined increases rapidly with the grade (although some of them are redundant) and so does the technical difficulty of the problem. It is, therefore, very important to take into account whatever additional information one has about it. For example, for some relevant applications $Y(-h) = -Y(h)$, so that $\alpha_{2i,l} = 0$ in (2.3) and the problem simplifies considerably. Let us split the Lie algebra in two subsets

$$\mathcal{S} = \left\{ Z \in L_{\mathcal{B}} \ : \ Z = \sum_{i=1}^{s} \sum_{j=1}^{\nu_i} \alpha_{2i-1,j} X_{2i-1,j} \right\},$$

$$\mathcal{R} = \left\{ Z \in L_{\mathcal{B}} \ : \ Z = \sum_{i=1}^{s} \sum_{j=1}^{\nu_i} \alpha_{2i,j} X_{2i,j} \right\}.$$

Then $L_{\mathcal{B}} = \mathcal{S} \oplus \mathcal{R}$ and $Y \in \mathcal{S}$. In addition $\mathcal{S}$ is a Lie triple system (i.e., it is closed under the double commutator) whereas $\mathcal{R}$ is a subalgebra of $L_{\mathcal{B}}$. Now

$$\begin{aligned} d_1 \equiv s_1 &= [x_{1,1}b_1 + x_{1,3}b_3 + \cdots, \ y_{1,2}b_2 + y_{1,4}b_4 + \cdots] \ \in \mathcal{S}, \\ d_2 \equiv r_1 &= [x_{2,1}b_1 + x_{2,3}b_3 + \cdots + x_{2,s_1}s_1 \ , \ y_{2,1}b_1 + y_{2,3}b_3 + \cdots + y_{2,s_1}s_1] \ \in \mathcal{R} \ , \\ d_3 \equiv s_2 &= [x_{3,1}b_1 + x_{3,3}b_3 + \cdots + x_{3,s_1}s_1, \ y_{3,2}b_2 + y_{3,4}b_4 + \cdots + y_{3,r_1}r_1] \ \in \mathcal{S}, \end{aligned}$$

and so on. Finally

$$Y = \alpha_1 b_1 + \alpha_3 b_3 + \cdots + \beta_1 d_1 + \beta_3 d_3 + \cdots \quad \in \mathcal{S} \tag{2.7}$$

and this is the most general combination which reproduces $Y \in \mathcal{S}$ with the minimum number of commutators. In order to illustrate this technique, we approximate $Y \in \mathcal{S}$ up to grade 6 and 8.

Grade 6. The most general term in $\mathcal{S}$ we can obtain up to grade 6 using $\{b_1, b_2, b_3\}$ is

$$Y = \alpha_1 b_1 + \alpha_2 b_3 + \alpha_3[12] + \alpha_4[23] + \alpha_5[212] + \alpha_6[113] + \alpha_7[1112]. \tag{2.8}$$

Let us consider, for example

$$\begin{aligned} s_1 &= [12], \\ r_1 &= [b_1 \ , \ x_1 b_3 + x_2 s_1] = x_1[13] + x_2[112], \\ s_2 &= [x_3 b_1 + x_4 b_3 + x_5 s_1 \ , \ b_2 + r_1] \\ &= x_3[12] - x_4[23] - x_5[212] + x_1 x_3[113] + x_2 x_3[1112] + \Theta(7) \end{aligned}$$

and choosing

$$x_1 = \frac{\alpha_6}{\alpha_3}, \quad x_2 = \frac{\alpha_7}{\alpha_3}, \quad x_3 = \alpha_3, \quad x_4 = -\alpha_4, \quad x_5 = -\alpha_5, \tag{2.9}$$

for $\alpha_3 \neq 0$ we obtain

$$Y = \alpha_1 b_1 + \alpha_2 b_3 + s_2 + \Theta(7). \tag{2.10}$$

In case $\alpha_3 = 0$ we can choose s_1, r_1, and s_2 in a slightly different way.

Grade 8. Observe that in this case we have $X_{7,14} = [111112]$ so that, if $\alpha_{7,14} \neq 0$, the minimum number of commutators for approximating Y is five. With this number of commutators, the most general element containing the term $X_{7,14}$ and preserving the property $Y(-h) = -Y(h)$ is

$$Z = \alpha_1 b_1 + \alpha_3 b_3 + s_1 + s_2 + s_3 \tag{2.11}$$

with

$$\begin{aligned}
s_1 &= [x_1 b_1 + x_2 b_3 \ , \ x_3 b_2 + x_4 b_4],\\
r_1 &= [x_5 b_1 + x_6 b_3 + x_7 s_1 \ , \ x_8 b_1 + x_9 b_3 + x_{10} s_1],\\
s_2 &= [x_{11} b_1 + x_{12} b_3 + x_{13} s_1 \ , \ x_{14} b_2 + x_{15} b_4 + x_{16} r_1],\\
r_2 &= [x_{17} b_1 + x_{18} b_3 + x_{19} s_1 + x_{20} s_2 \ , \ x_{21} b_1 + x_{22} b_3 + x_{23} s_1 + x_{24} s_2],\\
s_3 &= [x_{25} b_1 + x_{26} b_3 + x_{27} s_1 + x_{28} s_2 \ , \ x_{29} b_2 + x_{30} b_4 + x_{31} r_1 + x_{32} r_2].
\end{aligned}$$

We have $\{x_i\}_{i=1}^{32}$ variables to solve a system of 20 non-linear equations but some of them are redundant. For example, we can rewrite r_1 as

$$r_1 = z_1[b_1, b_3] + z_2[b_1, s_1] + z_3[b_3, s_1] + \Theta(8) \tag{2.12}$$

and only three among the six $\{x_i\}_{i=5}^{10}$ are necessary. Similarly, for r_2

$$r_2 = v_1[b_1, b_3] + v_2[b_1, s_1] + v_3[b_1, s_2] + v_4[b_3, s_1] + \Theta(8) \tag{2.13}$$

because $s_2 = \gamma s_1 + \Theta(5)$ with $\gamma = x_{11}x_{14}/(x_1 x_3)$ and then $[s_1, s_2] = \Theta(8)$ and $[b_3, s_2] = \gamma[b_3, s_1] + \Theta(8)$. Thus only 4 among the 8 variables are necessary. On the other hand, if $x_1 \neq 0$ then $s_1 = [b_1 + w_1 b_3 \ , \ w_2 b_2 + w_3 b_4]$ with $w_1 = x_2/x_1$, $w_2 = x_1 x_3$ and $w_3 = x_1 x_4$, and similarly with s_2 and s_3. In summary, when all these simplifications are taken into account we have (provided y_5, $y_{12} \neq 0$)

$$\begin{aligned}
s_1 &= [b_1 + y_1 b_3 \ , \ y_2 b_2 + y_3 b_4],\\
r_1 &= [b_1 + \frac{y_6}{y_5} b_3 \ , \ y_4 b_3 + y_5 s_1],\\
s_2 &= [b_1 + y_7 b_3 + y_8 s_1 \ , \ y_9 b_2 + y_{10} b_4 + y_{11} r_1],\\
r_2 &= [b_1 - \frac{y_{15}}{y_{12}} s_1 \ , \ y_{12} b_3 + y_{13} s_1 + y_{14} s_2],\\
s_3 &= [b_1 + y_{16} b_3 + y_{17} s_1 + y_{18} s_2 \ , \ y_{19} b_2 + y_{20} b_4 + y_{21} r_1 + y_{22} r_2],
\end{aligned} \tag{2.14}$$

and there are still 22 variables to solve 20 equations.

3 Optimal Magnus integration methods.

Several $2n$-th order time-symmetric integration algorithms for Equation (1.1) based on the Magnus expansion (1.2) have been obtained recently in [2]. These methods, for a given time interval $[t_k, t_k + h]$ and step-size h, read

$$X(t_k + h) = \exp(\Omega^{[2n]})\, X(t_k), \tag{3.1}$$

where $\Omega^{[2n]}$ approximates Ω up to order $2n$ and is expressed as a linear combination of the univariate integrals

$$B^{(i)} = \frac{1}{h^i} \int_{t_k}^{t_k+h} \left(t - \left(t_k + \frac{h}{2}\right)\right)^i A(t)\, dt, \quad i = 0, 1, 2, \ldots, \tag{3.2}$$

and their nested commutators. In particular, methods of order 4, 6 and 8 have been reported involving 1, 4 and 10 commutators, respectively. Here we apply the analysis done in the previous section to obtain integration schemes up to order 8 optimal with respect to the number of commutators.

When a symmetric collocation scheme is chosen and the graded basis $\{b_1, b_2, b_3, b_4\}$ of Section 2 is considered, then we obtain

$$\begin{aligned}
\Omega = {} & X_{1,1} + \frac{1}{12}X_{3,1} - \frac{1}{12}X_{3,2} - \frac{1}{80}X_{5,1} + \frac{1}{240}X_{5,4} \\
& - \frac{1}{1344}X_{7,1} + \frac{1}{360}X_{5,2} - \frac{1}{240}X_{5,5} - \frac{1}{2240}X_{7,2} + \frac{1}{6720}X_{7,3} \\
& + \frac{1}{6048}X_{7,4} - \frac{1}{840}X_{7,5} + \frac{1}{720}X_{5,3} + \frac{1}{6720}X_{7,6} - \frac{1}{7560}X_{7,7} \\
& + \frac{1}{4032}X_{7,8} + \frac{11}{60480}X_{7,9} - \frac{1}{6720}X_{7,10} - \frac{1}{15120}X_{7,11} - \frac{1}{30240}X_{7,12} \\
& + \frac{1}{7560}X_{7,13} - \frac{1}{30240}X_{7,14} + O(h^9)
\end{aligned} \tag{3.3}$$

and we have the following approximations.

Order 4.

$$\Omega^{[4]} = b_1 - \frac{1}{12}[b_1, b_2] \tag{3.4}$$

or, in terms of the integrals (3.2),

$$\Omega^{[4]} = B^{(0)} - [B^{(0)}, B^{(1)}] + O(h^5).$$

Order 6. We have an expression similar to (2.8) with

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

so that the scheme reads

$$\Omega^{[6]} = b_1 + \frac{1}{12}b_3 + \frac{1}{240}[-20b_1 - b_3 + s_1\ ,\ b_2 + r_1] \tag{3.5}$$

with

$$s_1 = [b_1, b_2],$$
$$r_1 = -\frac{1}{60}\,[b_1, 2b_3 + s_1], \tag{3.6}$$

and, up to this order,

$$b_1 = \frac{3}{4}\,(3B^{(0)} - 20B^{(2)}), \quad b_2 = 12\,B^{(1)}, \quad b_3 = -15\,(B^{(0)} - 12B^{(2)}). \tag{3.7}$$

Order 8. The equations obtained by expanding (2.11) with (2.14) and equating to (3.3) do not admit real solutions: with (2.14) we cannot solve simultaneously the equations corresponding to $X_{7,10}$, $X_{7,12}$, $X_{7,13}$, $X_{7,14}$. Thus at least one additional commutator is required. In fact, with six commutators we can approximate Ω up to order 8. More specifically,

$$\Omega^{[8]} = b_1 + \frac{1}{12}b_3 - \frac{7}{120}s_2 + \frac{1}{360}s_3, \tag{3.8}$$

s_2, s_3 being determined through the sequence

$$\begin{aligned}
s_1 &= -\frac{1}{28}\Big[b_1 + \frac{1}{28}b_3\ ,\ b_2 + \frac{3}{28}b_4\Big],\\
r_1 &= \frac{1}{3}\Big[b_1, -\frac{1}{14}b_3 + s_1\Big],\\
s_2 &= \Big[b_1 + \frac{1}{28}b_3 + s_1\ ,\ b_2 + \frac{3}{28}b_4 + r_1\Big], \qquad s_2' = [b_2, s_1],\\
r_2 &= \Big[b_1 + \frac{5}{4}s_1\ ,\ 2b_3 + s_2 + \frac{1}{2}s_2'\Big],\\
s_3 &= \Big[b_1 + \frac{1}{12}b_3 - \frac{7}{3}s_1 - \frac{1}{6}s_2\ ,\ -9b_2 - \frac{9}{4}b_4 + 63r_1 + r_2\Big].
\end{aligned} \tag{3.9}$$

Again, this algorithm can be formulated in terms of the integrals $B^{(i)}$ through the following change of basis:

$$\begin{aligned}
b_1 &= \tfrac{3}{4}\,(3B^{(0)} - 20B^{(2)}), \qquad & b_3 &= -15\,(B^{(0)} - 12B^{(2)}),\\
b_2 &= 15\,(5B^{(1)} - 28B^{(3)}), \qquad & b_4 &= -140\,(3B^{(1)} - 20B^{(3)}).
\end{aligned} \tag{3.10}$$

Observe that schemes (3.4), (3.5) and (3.8) are explicitly self-adjoint and involve just 1, 3 and 6 commutators, respectively. This is the *minimum* number needed at each order of approximation.

4 Lie-group solvers obtained from Magnus methods.

4.1 Fer based methods.

We now construct integration methods based on the Fer factorization (1.3) by applying the Baker–Campbell–Hausdorff (BCH) formula to the Magnus expansion

(3.3) and subsequently the optimization technique of Section 2. More specifically, in the domain of convergence of expansions (1.2) and (1.3) we can write

$$e^{\Omega(h)} = e^{F_1(h)}\, e^{\bar{F}(h)},$$

where

$$F_1 = \Omega_1 = \int_{t_0}^{t_0+h} A(t)dt = b_1 + \frac{1}{12} b_3 + \cdots$$

is the first term in the Magnus series and $e^{\bar{F}(h)} = e^{F_2(h)}\, e^{F_3(h)} \cdots$, with $F_2 = O(h^3)$ and $F_3 = O(h^7)$. In this way

$$e^{\bar{F}(h)} = e^{-F_1(h)}\, e^{\Omega(h)}$$

and the use of the BCH formula allows to write

$$\begin{aligned} \bar{F}(h) = & -\frac{1}{12} X_{3,2} + \frac{1}{24} X_{4,3} + \frac{1}{360} X_{5,2} \\ & - \frac{1}{80} X_{5,3} + \frac{1}{240} X_{5,4} - \frac{1}{240} X_{5,5} - \frac{1}{720} X_{6,2} \\ & + \frac{1}{360} X_{6,3} - \frac{1}{480} X_{6,4} + \frac{1}{480} X_{6,5} + \frac{1}{288} X_{6,7} + O(h^7). \end{aligned} \tag{4.1}$$

Observe that now there are terms of even order because the Fer factorization lacks the time-symmetry property. As a consequence, the minimum number of commutators required to attain a given order is higher (in general) than in the Magnus case, as noticed in [5].

Order 4. It suffices to consider only the first and second terms in (4.1), so that two commutators are needed in this case:

$$F_2^{[4]} = -\frac{1}{12}\Big([b_1, b_2] - \frac{1}{2}[b_1, b_1, b_2]\Big). \tag{4.2}$$

Order 6. We can reproduce the expression (4.1) with just four commutators in the following way:

$$\begin{aligned} d_1 &= 2\Big[b_1, b_2 + \frac{1}{9} b_3\Big], \\ d_2 &= \Big[b_1 + \frac{5}{2} b_3 + \frac{15}{4} d_1 \ , \ b_2 + \frac{1}{9} b_3 + d_1\Big], \\ d_3 &= \Big[b_1, -b_2 - \frac{13}{9} b_3 - 6 d_1 + d_2\Big], \\ F_2^{[6]} &= \frac{1}{720}\Big[b_1 + \frac{1}{20} b_3 - \frac{1}{40} d_1 - \frac{1}{60}(d_2 + d_3) \ , \ -60 b_2 + 15 d_1 + d_2 + d_3\Big]. \end{aligned} \tag{4.3}$$

The resulting n-th order algorithms, $n = 4, 6$, based on the Fer expansion

$$X(t_k + h) = e^{F_1(h)}\, e^{F_2^{[n]}(h)}\, X(t_k) \tag{4.4}$$

are optimal with respect to the number of commutators.

4.2 Symmetric Fer methods.

A drawback of the Lie-group solvers based on the Fer expansion is that, contrarily to Magnus methods, they do not preserve the time-symmetry of the exact solution. For this reason Zanna [21] has proposed recently a self-adjoint version of the Fer factorization in the form (1.4), which can be implemented as time-symmetric Lie-group numerical integrators with essentially the same computational cost as conventional Fer methods. In the following we construct symmetric Fer methods up to order 8 by expressing $S_i(h)$ in terms of Ω and then applying the optimization technique of Section 2. It is shown explicitly (at least up to order 6) that the new methods require exactly the same number of commutators as Magnus based solvers.

We can write (1.4) in the form

$$X(t_0+h) = e^{S_1(h)} e^{V(h)} e^{S_1(h)} X_0$$

with

$$S_1 = \frac{1}{2}\Omega_1 = \frac{1}{2}\int_{t_0}^{t_0+h} A(t)dt$$

and $e^V = e^{S_2} e^{S_3} \cdots e^{S_3} e^{S_2}$. Here, as for the conventional Fer expansion, $S_1 = O(h)$, $S_2 = O(h^3)$ and $S_3 = O(h^7)$.

For sufficiently small h we have

$$e^V = e^{-S_1} e^{\Omega} e^{-S_1},$$

where Ω is given by (3.3). Then, application of the symmetric Baker–Campbell–Hausdorff formula [19] allows to write, after some algebra,

$$V(h) = W + \frac{1}{24}[\Omega_1,[\Omega_1,W]] + \frac{1}{12}[W,[\Omega_1,W]] + \frac{1}{1920}[\Omega_1,[\Omega_1,[\Omega_1,[\Omega_1,W]]]] + O(h^9) \tag{4.5}$$

where $W = \Omega - \Omega_1$. When equation (3.3) is inserted in (4.5) and similar terms are grouped together we get

$$\begin{aligned} V(h) = &-\frac{1}{12}X_{3,2} - \frac{1}{80}X_{5,1} + \frac{1}{240}X_{5,4} \\ &- \frac{1}{1344}X_{7,1} + \frac{1}{360}X_{5,2} - \frac{1}{240}X_{5,5} - \frac{1}{2240}X_{7,2} + \frac{1}{6720}X_{7,3} \\ &+ \frac{1}{6048}X_{7,4} - \frac{1}{840}X_{7,5} - \frac{1}{480}X_{5,3} - \frac{1}{2688}X_{7,6} - \frac{1}{4032}X_{7,7} \\ &- \frac{5}{8064}X_{7,8} + \frac{19}{40320}X_{7,9} - \frac{1}{6720}X_{7,10} + \frac{1}{20160}X_{7,11} + \frac{1}{2688}X_{7,12} \\ &- \frac{1}{2240}X_{7,13} - \frac{1}{53760}X_{7,14} + O(h^9) \end{aligned} \tag{4.6}$$

and the following schemes:

Order 4. It is clear that $V(h) = -\frac{1}{12} X_{3,2} + O(h^5)$ and thus

$$S_2^{[4]}(h) = -\frac{1}{12}[b_1, b_2] = -[B^{(0)}, B^{(1)}] + O(h^5). \tag{4.7}$$

This method has been considered previously in [21].

Order 6. In this case $V(h)$ has an expression similar to (2.8) with

$$\alpha_1 = \alpha_2 = 0, \quad \alpha_3 = -\frac{1}{12}, \quad \alpha_4 = -\alpha_5 = \frac{1}{240}, \quad \alpha_6 = \frac{1}{360}, \quad \alpha_7 = -\frac{1}{480}$$

and thus, by substituting in (2.9) and (2.10) we get

$$V(h) \equiv S_2^{[6]}(h) = \frac{1}{240}\Big[-20b_1 - b_3 + s_1 \ , \ b_2 + r_1 \Big] \tag{4.8}$$

with

$$s_1 = [b_1, b_2], \qquad r_1 = \frac{1}{120}[b_1, -4b_3 + 3s_1]. \tag{4.9}$$

Order 8. Here, similarly to the Magnus case, the algorithm (2.14) involving 5 commutators does not reproduce the expression (4.6) with real coefficients, due exactly to the same reason. With one additional commutator, however, we are not able to solve all the equations (in particular, those corresponding to $X_{3,2}$, $X_{5,1}$, $X_{5,4}$, $X_{7,1}$). Thus, the scheme we propose involves seven commutators and reads

$$S_2^{[8]} = s_1 + s_1' + s_2 + s_3, \tag{4.10}$$

where s_1, s_2 and s_3 are determined by the algorithm (2.14) by considering instead

$$r_2 = \Big[b_1 - \frac{y_{15}}{y_{12}} s_1 \ , \ y_{12} b_3 + y_{13} s_1 + y_{14} s_2 + y_{23} \big[b_2, s_1 \big] \Big]$$

and

$$s_1' = \frac{1229}{162480}\Big[b_1 + \frac{124615}{2787372} b_3 \ , \ \frac{7627140}{1021271} b_2 + b_4 \Big].$$

The numerical values of the coefficients $y_1, \dots, y_{23}$ we have found are

$$\begin{array}{llll}
y_1 = -\frac{1}{35}, & y_2 = -\frac{644615}{113361081}, & y_3 = 0, & y_4 = \frac{1}{360}, \\
y_5 = \frac{37787027}{206276800}, & y_6 = 0, & y_7 = \frac{5}{84}, & y_8 = \frac{1398119999}{162442980}, \\
y_9 = -\frac{21}{148}, & y_{10} = -\frac{1}{48}, & y_{11} = 1, & y_{12} = \frac{53}{13320}, \\
y_{13} = 0, & y_{14} = \frac{1}{56}, & y_{15} = -\frac{13867838909}{1169589456000}, & y_{16} = \frac{2327}{11340}, \\
y_{17} = \frac{21433414477}{162442980}, & y_{18} = -\frac{677}{245}, & y_{19} = \frac{21}{2708}, & y_{20} = \frac{25}{32496}, \\
y_{21} = -\frac{53}{37}, & y_{22} = 1, & y_{23} = -\frac{1095823783}{86636256000}. &
\end{array}$$

The resulting n-th order ($n = 4, 6, 8$) time-symmetric integration algorithms

$$X(t_k + h) = e^{S_1(h)} \, e^{S_2^{[n]}(h)} \, e^{S_1(h)} X(t_k) \tag{4.11}$$

require 1, 3 and 7 commutators, respectively, and can be formulated in terms of the univariate integrals $B^{(i)}$ with the expressions (3.7) and (3.10).

4.3 Cayley-transform methods.

When the differential equation (1.1) evolves on the so-called J-orthogonal Lie group [18]

$$\mathrm{O}_J(n) = \{A \in \mathrm{GL}_n(\mathbb{R}) \,:\, A^T J A = J\}, \tag{4.12}$$

where $\mathrm{GL}_n(\mathbb{R})$ is the group of all $n \times n$ nonsingular real matrices and J is some constant matrix in $\mathrm{GL}_n(\mathbb{R})$, numerical methods based on the Cayley transform constitute a valid option preserving the Lie group structure of the equation [6, 11, 15].

Familiar examples of the J-orthogonal group $\mathrm{O}_J(n)$ are the orthogonal group (when $J = I$, the $n \times n$ identity matrix), the symplectic group $\mathrm{Sp}(n)$ (when J is the basic symplectic matrix) and the Lorentz group (corresponding to $J = \mathrm{diag}(1, -1, -1, -1)$).

As is well known, A is a J-orthogonal matrix if

$$B = -\frac{1}{\alpha}(I + A)^{-1}(I - A)$$

is a J-skew-symmetric matrix, i.e., if $B^T J + JB = 0$ and $\alpha \neq 0$. In fact, the set

$$\mathrm{o}_J(n) = \{B \in \mathfrak{gl}_n(\mathbb{R}) \,:\, B^T J + JB = 0\}, \tag{4.13}$$

where $\mathfrak{gl}_n(\mathbb{R})$ is the Lie algebra of all $n \times n$ real matrices, is the matrix Lie algebra associated with $\mathrm{O}_J(n)$ [18]. Conversely, if $B \in \mathrm{o}_J(n)$, then its Cayley transform

$$A = (I - \alpha B)^{-1}(I + \alpha B) \tag{4.14}$$

is J-orthogonal. Thus, for $\mathrm{O}_J(n)$ the Cayley transform (4.14) provides a useful alternative to the exponential mapping relating the Lie algebra to the Lie group. This fact is particularly important for numerical methods where the evaluation of the exponential matrix is the most computation-intensive stage of the algorithm.

It has been shown that the solution of (1.1) can be written ($\alpha = 1/2$) as

$$X(t) = \Big(I - \frac{1}{2}C(t)\Big)^{-1}\Big(I + \frac{1}{2}C(t)\Big)X_0 \tag{4.15}$$

in a neighborhood of t_0, with $C(t) \in \mathrm{o}_J(n)$ satisfying the so-called *dcayinv equation* [11]

$$\frac{dC}{dt} = A - \frac{1}{2}[C, A] - \frac{1}{4}CAC, \qquad t \geq t_0, \qquad C(t_0) = 0. \tag{4.16}$$

Time-symmetric methods of order 4 and 6 have been obtained based on the Cayley transform (4.15) by expanding the solution of (4.16) in a recursive manner and constructing quadrature formulae for the multivariate integrals that appear in the procedure [11, 15]. Then the recently introduced notion of graded free hierarchical algebra is applied to reduce the number of terms in the quadrature formula [13]. Here we show how efficient Cayley based methods can be built directly from Magnus based integrators.

Equations (1.2) and (1.5) lead to

$$\frac{1}{2}C(h) = \left(e^{\Omega(h)} + I\right)^{-1}\left(e^{\Omega(h)} - I\right)$$

and thus

$$\begin{aligned} C(h) &= \Omega(h) - \frac{1}{12}\Omega(h)^3 + \frac{1}{120}\Omega(h)^5 - \frac{17}{20160}\Omega(h)^7 + \cdots \\ &= \Omega\left(I - \frac{1}{12}\Omega^2\Big(I - \frac{1}{10}\Omega^2\big(I - \frac{17}{168}\Omega^2(I + \cdots)\big)\Big)\right). \end{aligned} \tag{4.17}$$

We get integration methods of order 4, 6 and 8 when the optimal approximation of Ω up to the given order (obtained in Section 3) is inserted in the corresponding truncation of the series (4.17). More specifically, the following schemes are built:

Order 4. In this case

$$C^{[4]} = \Omega^{[4]}\Big(I - \frac{1}{12}(\Omega^{[4]})^2\Big) = C(h) + O(h^5), \tag{4.18}$$

where $\Omega^{[4]}$ is given in (3.4). Then

$$C^{[4]} = b_1 - \frac{1}{12}[b_1, b_2] - \frac{1}{12}b_1^3 + O(h^5) \tag{4.19}$$

or, in terms of the integrals (3.2),

$$C^{[4]} = B^{(0)} - [B^{(0)}, B^{(1)}] - \frac{1}{12}(B^{(0)})^3 + O(h^5).$$

If we consider the Gauss–Legendre quadrature of order 4, given by $c_1 = 1/2 - \sqrt{3}/6$, $c_2 = 1/2 + \sqrt{3}/6$, and define $B_0 = (A_1 + A_2)/2$, $B_1 = \sqrt{3}(A_2 - A_1)$, then

$$C^{[4]} = B_0 + \frac{1}{12}[B_1, B_0] - \frac{1}{12}B_0^3 + O(h^5),$$

which coincides with the result obtained by Iserles [11] and Marthinsen and Owren [15]. Observe that $C^{[4]}$ requires, in general, the computation of four matrix-matrix products. In fact, this number can be reduced to 3 by writing

$$C^{[4]} = B^{(0)} + \Big(B^{(1)} - \frac{1}{12}(B^{(0)})^2\Big)B^{(0)} - B^{(0)}B^{(1)} + O(h^5). \tag{4.20}$$

Order 6. From (4.17) we have

$$C^{[6]} = \Omega^{[6]}\left(I - \frac{1}{12}(\Omega^{[6]})^2\Big(I - \frac{1}{10}(\Omega^{[6]})^2\Big)\right) = C(h) + O(h^7) \tag{4.21}$$

and thus three matrix-matrix products are required in addition to the three commutators involved in the computation of $\Omega^{[6]}$, for a total of nine matrix-matrix products per step. This has to be compared with other 6-th order implementations previously available (18 and 23 matrix-matrix products).

Order 8. Now we have to substitute directly in (4.17) Ω by (3.8) to get an 8-th order approximation $C^{[8]}$ to $C(h)$ requiring 16 matrix-matrix products.

The resulting n-th order ($n = 4, 6, 8$) numerical methods for (1.1) are then given by

$$X(t_k + h) = \Big(I - \frac{1}{2}C^{[n]}\Big)^{-1}\Big(I + \frac{1}{2}C^{[n]}\Big)X(t_k) \tag{4.22}$$

and they preserve time-symmetry.

4.4 Magnus–Padé methods.

As is well known, diagonal Padé approximants map the Lie algebra $\mathrm{o}_J(n)$ to the Lie group $\mathrm{O}_J(n)$ and thus constitute also a valid alternative to the evaluation of the exponential matrix in Magnus based methods for this particular Lie group. More specifically, if $B \in \mathrm{o}_J(n)$, then $\psi_{2m}(tB) \in \mathrm{O}_J(n)$ for sufficiently small $t \in \mathbb{R}$, with

$$\psi_{2m}(\lambda) = \frac{P_m(\lambda)}{P_m(-\lambda)}, \tag{4.23}$$

provided the polynomials P_m are generated according to the recurrence

$$\begin{aligned} P_0(\lambda) = 1, \qquad P_1(\lambda) = 2 + \lambda, \\ P_m(\lambda) = 2(2m-1)P_{m-1}(\lambda) + \lambda^2 P_{m-2}(\lambda). \end{aligned}$$

Moreover, $\psi_{2m}(\lambda) = e^{\lambda} + O(\lambda^{2m+1})$ and ψ_2 corresponds to the Cayley transform, whereas for $m = 2, 3$ we have

$$\begin{aligned} \psi_4(\lambda) &= \Big(1 + \frac{1}{2}\lambda + \frac{1}{12}\lambda^2\Big)\Big/\Big(1 - \frac{1}{2}\lambda + \frac{1}{12}\lambda^2\Big), \\ \psi_6(\lambda) &= \Big(1 + \frac{1}{2}\lambda + \frac{1}{10}\lambda^2 + \frac{1}{120}\lambda^3\Big)\Big/\Big(1 - \frac{1}{2}\lambda + \frac{1}{10}\lambda^2 - \frac{1}{120}\lambda^3\Big). \end{aligned}$$

Thus, we can combine the optimized approximations to Ω obtained in Section 3 for Magnus based methods with diagonal Padé approximants up to the corresponding order to obtain time-symmetric integration schemes preserving the algebraic structure of the problem without computing the matrix exponential.

The methods thus obtained involve 3, 8 and 15 matrix-matrix products for order 4, 6 and 8, respectively. In consequence, the computational effort required is similar to that of Cayley methods and constitute a very attractive alternative to them. In fact, these "Magnus–Padé" methods perform better than the Cayley methods in some cases, as we will see in the numerical experiments.

Observe that $\Omega^{[2n]} = O(h)$ and then

$$\psi_{2m}(\Omega^{[2n]}) = \exp(\Omega^{[2n]}) + O(h^{2k+1})$$

where $k = \min\{m, n\}$. With $m = n$ we have a method of order $2n$. However, for some problems this rational approximation to the exponential may be not very accurate depending on the eigenvalues of $\Omega^{[2n]}$. In this case one may take $m > n$, thus giving a better approximation to the exponential and a more accurate result just by increasing slightly the computational cost of the method.

Table 4.1: Computational cost of different Lie-group solvers for the linear differential equation (1.1). P also includes the matrix-matrix products coming from the commutators. The $*$ indicates that the method is optimal with respect to the number of commutators.

Method	Order	F	C	P	E	In
Magnus	4*	2	1	2	1	
	6*	3	3	6		
	8*	4	6	12		
Cayley (Magnus–Padé)	4	2		3		1
	6	3		9(8)		
	8	4		16(15)		
Fer	4*	2	2	4	2	
	6*	3	4	8		
Symmetric Fer	4*	2	1	2	3	
	6*	3	3	6		
	8	4	7	14		

4.5 Computational cost.

In order to compare the efficiency of the different geometric integrators obtained in this paper, we must estimate their computational cost. It is made up of the cost of evaluating the single integrals $B^{(i)}$ (or, instead, the number of A evaluations if quadratures are used), the total number of commutators (or matrix-matrix products) involved and eventually the exponential or the inverse of a matrix. Although the actual cost is highly dependent on the Lie algebra in question and the structure of the matrix differential equation, it is clear, however, that the factors above enumerated could serve as a good indicator of the practical performance of the different methods. These numbers are collected in Table 4.1 in a similar form to reference [5]: F stands for the number of function evaluations, C is the number of commutators, P indicates the total number of matrix-matrix products (commutators included), E is the number of matrix exponentials and In the number of inversions per time step for each class of methods when they are applied to the linear equation (1.1).

5 Numerical examples.

Here we show some of the properties of the Lie-group solvers obtained in Sections 3 and 4 and test their efficiency on some examples, chosen to embrace different kinds of physically relevant behavior.

Example 5.1. First we consider the coefficient matrix

$$A(t) = -i\frac{\omega_0}{2}\sigma_3 - i\beta(\sigma_1 \cos\omega t + \sigma_2 \sin\omega t), \tag{5.1}$$

where σ_j, $j = 1, 2, 3$, denote the Pauli matrices

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and β, ω_0, ω are real parameters. Then the corresponding initial value problem (1.1) with $X(0) = I$ has the exact solution

$$X(t) = \exp\left(-\tfrac{1}{2}i\omega t\sigma_3\right) \exp\left(-it\left(\tfrac{1}{2}(\omega_0 - \omega)\sigma_3 + \beta\sigma_1\right)\right), \tag{5.2}$$

which belongs to SU(2) for all t. This example is well known in the theory of nuclear magnetic resonance [7] and is useful to check the behavior of numerical schemes over very long integration intervals by comparing approximate solutions with the exact result. In particular, we analyze the efficiency of the 6-th and 8-th order algorithms described in Sections 3 and 4, both in terms of analytic integrals $B^{(i)}$ and symmetric quadratures. We compute the global error in the solution as a function of the computational effort measured in CPU time for $\omega_0 = \omega = 1$ and $\beta = 0.8$. The global error is measured by computing the Frobenius norm of the difference between approximate and the exact solution matrices at time

$$t_{\rm f} = 5000\frac{2\pi}{\omega'}, \quad \text{with} \quad \omega' = \sqrt{(\omega_0 - \omega)^2 + 4\beta^2},$$

although similar conclusions can be attained by determining the quantum mechanical transition probability

$$|(X(t))_{21}| = \left(\frac{2\beta}{\omega'} \sin\frac{\omega' t}{2}\right)^2. \tag{5.3}$$

In Figure 5.1(a) we plot (in a log-log scale) the efficiency curves for the 6-th order integration methods based on Magnus with exact univariate integrals (M6) and Gauss–Legendre quadratures (M6c), Magnus–Padé (MP6), and standard and symmetric Fer expansions (F6 and SF6, respectively). For comparison we have also included the results obtained with a standard explicit 6-th order Runge–Kutta method (RK6), which requires six evaluations of $A(t)$ per step [9]. The graph exhibits clearly the order of consistency of the algorithms and the advantages of implementing Magnus with the integrals $B^{(i)}$ instead of quadratures, at least for this example. This conclusion remains also valid for the other schemes. Notice that the improved version (4.21) of C6 obtained from Magnus is the most efficient method, although MP6 proves to be a quantitatively valid alternative to Cayley.

With respect to the 8-th order algorithms of Section 3, their efficiency curves for this problem are shown in Figure 5.1(b), where a similar notation has been used for the methods. Essentially, the same comments apply, but now M8 is the most efficient method and the efficiency does not change if the exponential is computed with a Padé approximant.

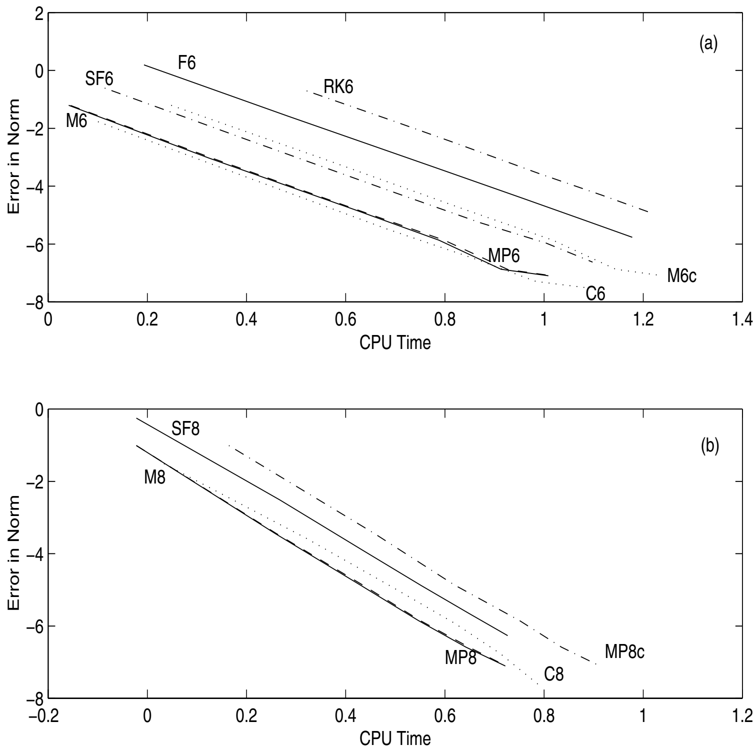

Figure 5.1: Efficiency diagram corresponding to the optimized 6-th (a) and 8-th (b) Lie-group solvers of Section 3 when they are applied to the first example with $\omega_0 = \omega = 1$, $\beta = 0.8$, both with exact integrals $B^{(i)}$ and Gauss–Legendre quadratures. A 6-th order explicit Runge–Kutta method is included for comparison (RK6).

Example 5.2. As a second illustration we consider a skew-symmetric coefficient matrix $A(t)$ and $X_0 = I$, so that the solution $X(t)$ is orthogonal for all t. In particular we take as upper triangular elements of $A(t)$ the following:

$$A_{ij} = \sin\left(t(i^2 - j^2)\right), \qquad 1 \leq i < j \leq N, \tag{5.4}$$

$$A_{ij} = \log\left(1 + t\frac{i-j}{i+j}\right), \tag{5.5}$$

with $N = 10, 20$. In both cases $X(t)$ oscillates with time, mainly due to the time-dependence of $A(t)$ (in (5.4)) or the norm of the coefficient matrix (with (5.5)).

The integration is carried out in the interval $t_{\rm f} \in [0, 10]$ and the approximate solutions are compared with the exact one at the final time $t = 10$, where the corresponding error is computed for different h. The Lie-group solvers presented in this paper are implemented with Gauss–Legendre quadratures and constant step size.

First, we analyze the efficiency of the different 6-th order methods for $N = 10$. The exponentials are approximated using Padé approximants of order 6, XP6, and of order 8, XP68, with X=M,F,SF corresponding to Magnus, Fer and symmetric Fer, respectively. Figure 5.2(a) shows the results obtained for the matrix (5.4). We have not included the XP68 curves because they are very close to the corresponding XP6 ones. The results are similar to those of Figure 5.1(a) and are in accordance with Table 4.4. In Figure 5.2(b) we show the results obtained when (5.5) is considered. For this problem the eigenvalues of $A(t)$ take large values and it is important to consider an accurate approximation to the exponential. This is clearly seen in the picture where Cayley gives a poor approximation to the exponential. In particular, the symmetric-Fer method seems the most efficient since $B^{(0)}$ is the most important term and it is split in two parts.

In contrast with the first example, now there is not a closed formula for the matrix exponentials appearing in the Magnus and Fer based integrators, so that some alternative procedure must be applied. Here the computation of e^C to machine accuracy is done by scaling-Padé-squaring, i.e., we take

$$e^C = \left(e^{C/2^k}\right)^{2^k}$$

for some integer k and estimate $e^{C/2^k}$ by a diagonal Padé approximant of a sufficiently high order m, so that the result is correct up to round-off. Although k and m are clearly h-dependent we have found that, for the integration methods and step sizes h considered in this paper, the pair $k = 2$, $m = 8$ provides successful results if C has the form $C = \alpha_1 h + O(h^2)$, whereas $k = 0$, $m = 4$ lead to the same result as the procedure implemented in Matlab when $C = \alpha_3 h^3 + O(h^4)$, as is the case with Fer methods. Using this approach we obtain for M6 and SF6 curves which are indistinguishable from SFP68 in Figure 5.2(b).

In Figure 5.3 we present, for clarity, only the results from Magnus (solid lines) and Cayley (broken lines) based 6-th and 8-th order methods. For comparison we also include the result obtained with RK6 (dotted lines). We should mention that the relative position of the curves corresponding to the Runge–Kutta method and the geometric integrators considered in this paper does not depend significantly on the dimension of $A(t)$. This is mainly due to the following reasons: (i) when N increases then also does the intricacy of the solution, so that the better preservation of its structure by the geometric integrators compensates their higher computational cost, and (ii) the number of $A(t)$ evaluations is reduced to a minimum for the Lie-group solvers of this paper. We also include for $N = 20$ an estimate of the efficiency of an 8-th order Magnus method involving 45 commutators instead of 6 (circles joined by a solid line in Figures 5.3(c) and (d)). The resulting loss of accuracy is of approximately two orders of magnitude for the same computational effort.

It is also worth noticing that the efficiency achieved by symmetric Fer methods is quite similar to that of Magnus if the matrix exponentials are evaluated accurately up to machine precision. This is so for the matrix (5.4) even if Padé approximants of the corresponding order are used to replace the exponentials. In fact, all the geometric integrators provide essentially the same results at each order (with slightly

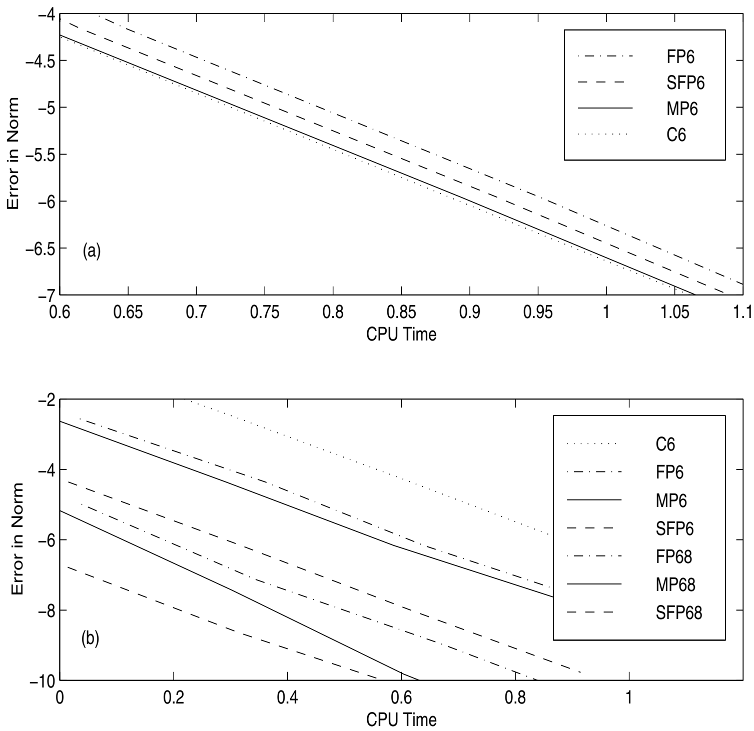

Figure 5.2: Efficiency diagram corresponding to the optimized 6-th order Lie-group solvers when they are applied to the matrix (5.4), (a), and to the matrix (5.5), (b). The labels follow the same order of appearance as the curves from top to bottom.

different CPU time). On the other hand, the efficiency of Magnus–Padé methods (and also of Fer–Padé) is highly deteriorated for the matrix (5.5), although it is always better than the corresponding to Cayley schemes.

To better illustrate all these results, in Figure 5.4 we display the error in the solution corresponding to (5.4) and (5.5) as a function of time in the interval $t \in [0, 100]$ obtained with the 6-th order Lie-group solvers and RK6. We take $N = 10$ and $h = 1/20$, all methods requiring the same number of A evaluations, except RK6, which duplicates this number. Observe the great importance of evaluating the exponential as accurately as possible for the matrix (5.5): by increasing slightly the computational cost per step in the computation of the matrix exponential it is possible to improve dramatically the accuracy of the methods. On the contrary, for matrix (5.4) the meaningful fact seems to be that the integration scheme provides a solution in the corresponding Lie group.

It is well known that the implicit s-stage Runge–Kutta–Gauss–Legendre (RKGL) method of order $2s$ can be considered a geometric integrator if $A \in \mathrm{o}_J(n)$. In fact, for the linear equation (1.1) this method can be written in an explicit form: one

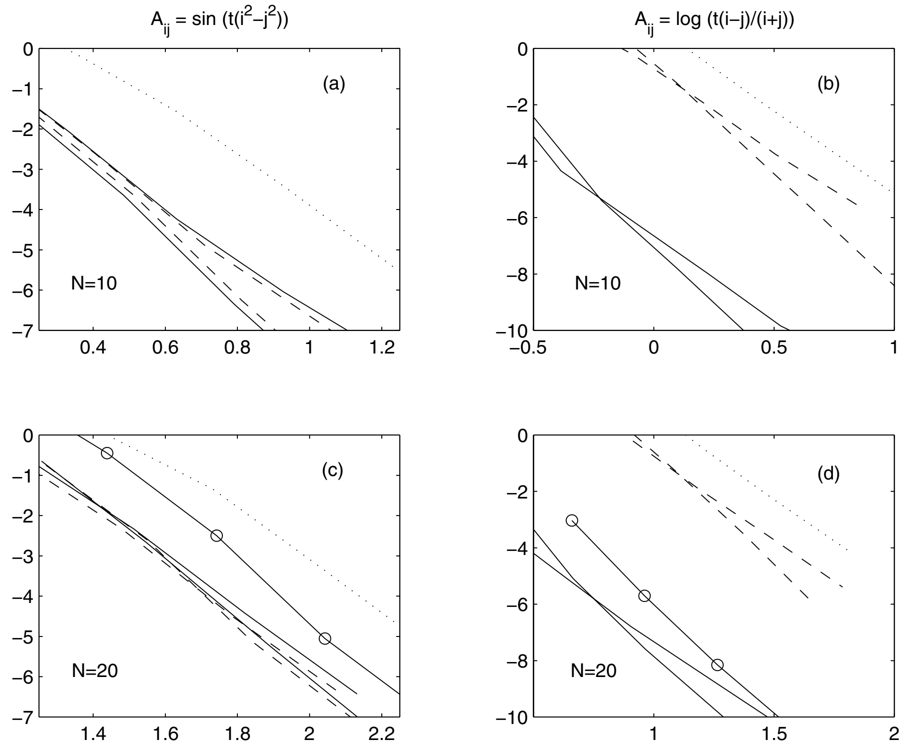

Figure 5.3: Error versus CPU time (in logarithmic scale) obtained with the 6-th and 8-th order integrators based on Magnus (solid lines), Cayley (broken lines) and RK6 (dotted lines). We have also included an 8-th order method based on Magnus with 45 commutators (circles joined by lines).

has to invert an $sn \times sn$ matrix, but in fact its block structure leads to a rational expression similar to Magnus–Padé, involving $n \times n$ matrices and (possibly) more matrix-matrix products. The results achieved by RKGL methods for the matrix (5.4) are slightly less efficient than the other geometric integrators, whereas for the matrix (5.5) are very similar to Magnus–Padé schemes of the same order. This is not very surprising, because in the particular case when A is a constant matrix both methods are equivalent, when neglecting machine accuracy.

6 Final comments.

In this paper we have developed a technique for minimising the number of commutators involved in the implementation of Lie group numerical integrators. This technique is applied subsequently to obtain different families of 6-th and 8-th order geometric integrators for linear differential equations based on the Cayley transform and the Magnus and Fer analytical expansions. Some of the methods are optimized in the sense that the number of commutators and matrix-matrix

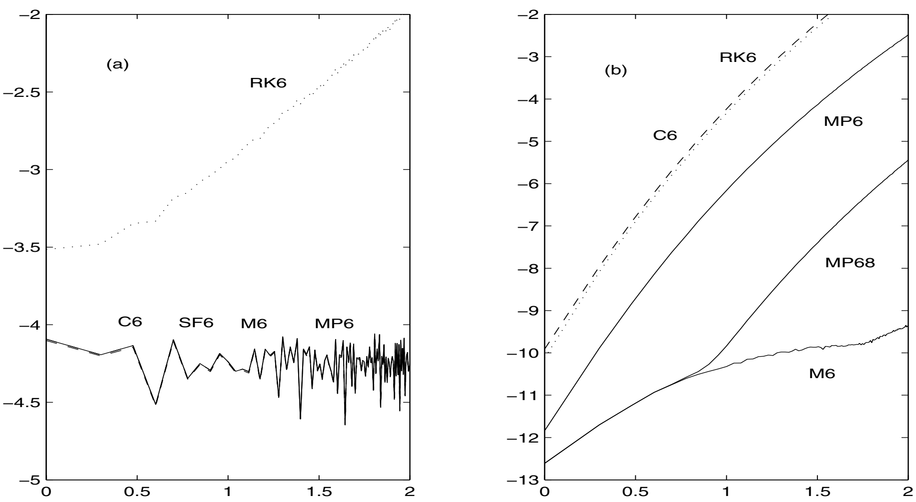

Figure 5.4: Error as a function of time (in logarithmic scale) obtained with different 6-th order integrators: (a) problem (5.4); (b) problem (5.5).

products is reduced to a minimum, so that they are far more efficient than other schemes of similar type found in the literature.

Although only J-orthogonal matrices are considered in this paper, the integration methods based on Magnus and Fer we present here can be applied to more general Lie groups than $O_J(n)$ if the matrix exponentials are evaluated up to machine precision.

We have also studied the applicability of these integration algorithms to several examples involving different algebraic structure. Our purpose was not to compare them exhaustively with other integrators but to establish when the use of one particular Lie-group solver is recommended, based on criteria such as their relative efficiency, the preservation of additional qualitative properties of the exact solution, etc.

From the previous analysis of the examples considered we can draw some tentative conclusions:

- There are problems where the preservation of the Lie group by the numerical solution seems to be the only crucial feature. Then no particular geometric integrator is preferred to the others, because all of them provide the correct qualitative and also quantitative behavior at approximately the same computational cost. In this case Magnus–Padé methods can be considered as an interesting alternative to both Magnus and Cayley solvers.

- On the contrary, for other problems the presence of the exponential in the numerical scheme seems to be essential. It is precisely the matrix exponential which allows to recover accurately the main features of the exact solution, and not only the Lie algebraic structure of the problem. Indeed, a more accurate approximation to the matrix exponential leads directly to a better overall description. In this case it is worthwhile to calculate accurately the matrix exponential even if the computational cost increases. In this respect we should remark that the combined use of the Magnus expansion and high order Padé approximants leads to integration schemes clearly superior to those based on the Cayley transform.

- If the elements of the matrix $A(t)$ are simple enough so as to render feasible the exact evaluation of the integrals $B^{(i)}$, then it is convenient to incorporate them into the algorithms instead of numerical quadratures because the corresponding integration schemes are more efficient.

It has been argued [12] that one of the reasons which explains the superior performance of Lie-group solvers might be that standard methods invariably employ the ansatz that locally the solution behaves like a polynomial in t, whereas the former are based on a representation of the solution as an exponential of a matrix with polynomial entries (or a product of such exponentials). Unlike polynomials, exponentials of matrices with polynomial entries can describe adequately high oscillations, exponential changes, etc. From our analysis we could say that, at least in some situations, these exponentials provide also a better qualitative and quantitative description than rational functions of matrices with polynomials entries.

Finally, it is important to keep in mind that the technique presented in Section 2 for minimising the number of commutators is based solely on the structure of the graded free Lie algebra. Therefore it can be used in any procedure involving such a structure. In particular, it can be applied to reduce the computational cost of certain Lie-group integrators for non-linear differential equations [17] or any composition involving BCH-type expansions [3].

Acknowledgments.

The authors thank A. Iserles, P. C. Moan, and A. Murua for useful remarks.

REFERENCES

1. S. Blanes, F. Casas, J. A. Oteo, and J. Ros, *Magnus and Fer expansions for matrix differential equations: The convergence problem*, J. Phys. A. Math. Gen., 31 (1998), pp. 259–268.
2. S. Blanes, F. Casas, and J. Ros, *Improved high order integrators based on the Magnus expansion*, BIT, 40 (2000), pp. 434–450.
3. S. Blanes, F. Casas, and J. Ros, *Optimization of BCH-type expansions*, to be published.
4. C. J. Budd and A. Iserles, *Geometric integration: Numerical solution of differential equations on manifolds*, Philos. Trans. Royal Soc. London Ser. A, 357 (1999), pp. 945–956.
5. E. Celledoni, A. Iserles, S. P. Nørsett, and B. Orel, *Complexity theory for Lie-group solvers*, J. Complexity, 18 (2002), pp. 242–286.

6. F. Diele, L. Lopez, and R. Peluso, *The Cayley transform in the numerical solution of unitary differential systems*, Adv. Comput. Math., 8 (1998), pp. 317–334.
7. R. R. Ernst, G. Bodenhausen, and A. Wokaun, *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*, Clarendon Press, Oxford, 1986.
8. F. Fer, *Résolution de l'équation matricielle $U' = pU$ par produit infini d'exponentielles matricielles*, Bull. Classe Sci. Acad. Roy. Bel., 44 (1958), pp. 818–829.
9. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations*, 2nd ed., Springer-Verlag, Berlin, 1993.
10. A. Iserles and S. P. Nørsett, *On the solution of linear differential equations in Lie groups*, Philos. Trans. Royal Soc. London Ser. A, 357 (1999), pp. 983–1019.
11. A. Iserles, *On Cayley-transform methods for the discretization of Lie-group equations*, Found. Comput. Math., 1 (2001), pp. 129–160.
12. A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, *Lie group methods*, Acta Numerica, 9 (2000), pp. 215–365.
13. A. Iserles and A. Zanna, *On the dimension of certain graded Lie algebras arising in geometric integration of differential equations*, LMS J. Comput. Math., 3 (2000), pp. 44–75.
14. W. Magnus, *On the exponential solution of differential equations for a linear operator*, Comm. Pure Appl. Math., 7 (1954), pp. 649–673.
15. A. Marthinsen and B. Owren, *Quadrature methods based on the Cayley transform*, Appl. Numer. Math., 39 (2001), pp. 403–413.
16. H. Munthe-Kaas, *High order Runge–Kutta methods on manifolds*, Appl. Numer. Math., 29 (1999), pp. 115–127.
17. H. Munthe-Kaas and B. Owren, *Computations in a free Lie algebra*, Philos. Trans. Royal Soc. London Ser. A, 357 (1999), pp. 957–981.
18. M. Postnikov, *Lie Groups and Lie Algebras. Semester V of Lectures in Geometry*, URSS Publishers, Moscow, 1994.
19. H. Yoshida, *Construction of higher order symplectic integrators*, Phys. Lett. A, 150 (1990), pp. 262–268.
20. A. Zanna, *Collocation and relaxed collocation for the Fer and the Magnus expansions*, SIAM J. Numer. Anal., 36 (1999), pp. 1145–1182.
21. A. Zanna, *The Fer expansion and time-symmetry: A Strang-type approach*, Appl. Numer. Math., 39 (2001), pp. 435–459.