UNIVERSITY OF CAMBRIDGE

Numerical Analysis Reports

Complexity theory for Lie-group solvers

E. Celledoni, A. Iserles, S.P. Nørsett & B. Orel

DAMTP 1999/NA20
December 1999

Department of Applied Mathematics and Theoretical Physics
Silver Street
Cambridge
England CB3 9EW

Complexity theory for Lie-group solvers

Elena Celledoni [1], ***Arieh Iserles*** [2], ***Syvert P. Nørsett*** [3] ***and Bojan Orel*** [4]

Abstract

Commencing with a brief survey of Lie-group theory and differential equations evolving on Lie groups, we describe a number of numerical algorithms designed to respect Lie-group structure: Runge–Kutta–Munthe-Kaas schemes, Fer and Magnus expansions. This is followed by complexity analysis of Fer and Magnus expansions, whose conclusion is that for order four, six and eight an appropriately discretized Magnus method is always cheaper than a Fer method of the same order.

Each Lie-group method of the kind surveyed in this paper requires the computation of a matrix exponential. Classical methods, e.g. Krylov-subspace and rational approximants, may fail to map elements in a Lie algebra to a Lie group. Therefore we survey a number of approximants based on the splitting approach and demonstrate that their cost is compatible (and often superior) to classical methods.

1 Introduction

A central goal of classical numerical analysis is to design, implement and analyse computational algorithms that cater for a wide range of problems. Thus, for example, even a brief perusal of any standard numerical library will reveal subroutines and software packages intended for 'all' initial-value problems for ordinary differential equations, say. Needless to say, no realistic method can exist in a world devoid of restriction: unless a (local) Lipschitz condition is satisfied, existence and uniqueness of the exact solution cannot be assured. Further conditions need be imposed to derive meaningful criteria about the performance of numerical methods and it is customary to assume that the underlying differential system is sufficiently smooth and that its sensitivity to small perturbations (often phrased in the terminology of Lyapunov exponents) is not too acute. Complexity analysis follows along similar lines and its broad framework is provided by defining a *function class* in terms of smoothness (hence, confining functions to a function space, typically a Sobolev space) and restrictions on the size of Lipschitz constants (Werschultz 1991).

The last few years have witnessed the emergence of an alternative paradigm of scientific computing, emphasizing the imperative to discretise differential equations whilst respecting their known *qualitative* features. The goal of the new approach, often

[1]Institute of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim NTH-7074, Norway, email `elenac@math.ntnu.no`.

[2]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW, England, email `A.Iserles@damtp.cam.ac.uk`.

[3]Institute of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim NTH-7074, Norway, email `norsett@math.ntnu.no`.

[4]Department of Mathematics and Mechanics, University of Ljubljana, Jadranska 19, 1000 Ljubljana, Slovenia, email `bojan@rosika.fri.uni-lj.si`.

termed *geometric integration,* is to fashion a seamless link between pure-mathematical analysis of differential equations and their computation. The benefits of geometric integration range far beyond producing solutions that inherit more 'correct' structure and they include superior immunity to the accumulation of numerical error, recovery of the right dynamical behaviour of the underlying equation and invariance of quantities which are relevant to the mathematical model in science or engineering from which our differential system originates (Budd & Iserles 1999, Iserles 1996).

Generically, we can think about two broad types of 'qualitative' features: asymptotic behaviour and invariants. The importance of asymptotic behaviour under discretization has been recognized since the dawn of computer era. More recently, incorporating ideas from the theory of nonlinear dynamical systems, the retention of correct asymptotics under discretization has been put into a firm and impressive theoretical framework (Stuart & Humphries 1996). Matters are considerably more complicated insofar as the retention of invariants is concerned and the underlying theory is in a considerably more tentative state. Given the differential system

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \quad t \geq 0, \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \tag{1.1}$$

which might be ordinary or partial, we can distinguish among different families of invariants (Budd & Iserles 1999). Firstly, *conservation laws,* invariants that apply to the solution $\boldsymbol{y}$ itself, of the form $I(\boldsymbol{y}(t)) = \text{const}$; secondly, *differential invariants* that act on the derivative (e.g., Lie-group symmetries); and thirdly invariants that apply to linear functionals acting on the derivative or, with greater abstraction, involve the *cotangent bundle* of some manifold, an example being the symplectic form of a Hamiltonian system. In the sequel we focus our attention on the first type of invariants, conservation laws.

It is important to bear in mind that conservation laws are all about the *geometry* of the configuration space of allowed solutions. In the classical numerical scheme of things the discretization evolves in a Euclidean space $\mathbb{R}^m$ but the presence of a conservation law restricts it to the manifold $\mathcal{M} = \{\boldsymbol{x} \in \mathbb{R}^m : I(\boldsymbol{x}) = I(\boldsymbol{y}_0)\}$. Given our understandable reluctance to develop a separate theory for each and every smooth manifold $\mathcal{M}$, we need to classify these constructs into classes that share convenient structural features and lend themselves to a similar type of analytic and numerical treatment. By implication, we classify systems (1.1) into classes, a standard device of complexity theory. Yet, it is essential to emphasise a crucial difference: our classification is primarily *geometrical,* rather than *analytic.* Needless to say, at the end of the day we need to specify also analytic features of the function $\boldsymbol{f}$ but, in the broader scheme of things, this plays a distinctively subsidiary role.

The point of departure for our classification of conservation laws is the introduction of *Lie groups.* In Section 2 we review this construct and describe ordinary differential equations that evolve on Lie groups and several methods for their solution. All such methods share similar basic approach, which is represented pictorially in the following diagram.

$$\begin{array}{c}\textbf{Lie-group equation}\\ \begin{tikzcd} \boldsymbol{y}_N \arrow[d] & & \boldsymbol{y}_{N+1} \\ \boldsymbol{\sigma}_N \arrow[rr, "\text{Lie-algebraic method}"] & & \boldsymbol{\sigma}_{N+1} \arrow[u] \end{tikzcd} \\ \textbf{Lie-algebraic equation}\end{array} \tag{1.2}$$

The Lie-group equation is first lifted into the underlying Lie algebra, solved there and ultimately the solution is mapped back to the Lie group. The main reason for the efficacy of this procedure is that it is considerably easier to respect the structure of a Lie algebra under discretization, since it is a *linear space.*

Many important differential equations evolve on Lie groups. Yet, the importance of Lie-group solvers is significantly greater since they can be generalized to equations evolving in *homogeneous spaces.* This is explained in Section 2.

Having laid the theoretical foundations, we describe in Section 3 a number of Lie-group solvers: *Runge–Kutta–Munthe-Kaas* methods, *Magnus* and *Fer* expansions.

Section 4 is devoted to our preliminary investigation of complexity in the setting of Magnus and Fer expansions. Our purpose in that section is primarily to set the general framework and explain how standard complexity concepts and considerations need be amended consistently with the new approach. The specific issue is the comparison of a number of distinct combinations of Magnus and Fer expansions in terms of their computational cost.

The mapping from a Lie algebra to a Lie group is typically (in finitely-dimensional spaces) the familiar matrix exponential, whose evaluation and complexity are the theme of Section 5. Although the approximation of $\exp A$, where A is a square matrix, is familiar to most numerical analysts, our goal is distinguished in one crucial detail: the approximation *must* map the Lie algebra to the Lie group, otherwise the whole purpose of the computation will be defeated. With important exceptions, this rules out rational approximants.

In Section 6 we conclude our narrative, reiterating how the new desiderata of geometric integration amend the conventional framework of complexity theory and sketching goalposts for future work.

Throughout this paper we are emphasizing the role of the exponential map. It is important to mention that the exponential is not the only useful coordinate map from a Lie algebra to a Lie group. Other significant examples include Cayley transforms and cannonical coordinates of the second kind (Varadarajan 1984).

2 Lie groups and Lie-group equations

2.1 Lie groups and Lie algebras

Although the definitions of a Lie group and a Lie algebra are likely to be familiar, we commence our brief survey of the Lie theory from the very beginning, both to remind the reader and to establish common terminology and notation. Excellent sources for the material of this section are (Carter, Segal & Macdonald 1995, Olver 1995, Onischik 1993, Varadarajan 1984) and, insofar as differential topology is concerned, (Guillemin & Pollack 1974).

We say that the manifold $\mathcal{G}$ is a *Lie group* if it forms a group with respect to a binary operation $\mathcal{G} \times \mathcal{G} \to \mathcal{G}$, so that multiplication and inverse are continuous in the topology of the manifold. Familiar examples of Lie groups are

1. The *general linear group* $\mathrm{GL}(n;\mathbb{F})$ of all nonsingular $n \times n$ matrices over the field F. (Typically, $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$.);
2. The *special linear group* $\mathrm{SL}(n;\mathbb{F})$ of all $X \in \mathrm{GL}(n;\mathbb{F})$ such that $\det X = 1$;
3. The *orthogonal group* $\mathrm{O}(n;\mathbb{F})$ of all $X \in \mathrm{GL}(n;\mathbb{F})$ such that $XX^{\mathrm{T}} = \boldsymbol{I}$;
4. The *special orthogonal group* $\mathrm{SO}(n;\mathbb{F}) = \mathrm{O}(n;\mathbb{F}) \cap \mathrm{SL}(n;\mathbb{F})$;
5. The *symplectic group* $\mathrm{Sp}(n)$ of all matrices $X \in \mathrm{GL}(2n;\mathbb{R})$ such that $XJX^{\mathrm{T}} = J$, where $J = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{O} \end{bmatrix}$;
6. The group $\mathrm{Diff}\,\mathcal{N}$ of all *diffeomorphisms* from the smooth manifold $\mathcal{N}$ onto itself. $\mathrm{Diff}\,\mathcal{N}$ is an example of an infinite-dimensional group, outside the scope of our paper.

We assume in the sequel that $\mathbb{F}$ is of zero characteristic: thus, for example, the *modular group* $\mathrm{SL}(n;\mathbb{Z})$ is banned. The reason is that ultimately our concern is with differential equations that evolve on $\mathcal{G}$. Note for further reference that, except for $\mathrm{Diff}\,\mathcal{N}$, all our examples consist of *matrix groups.*

A critical feature in describing differential equations evolving on a smooth manifold $\mathcal{M}$ is the structure of its *tangent spaces* $\mathrm{T}\mathcal{M}|_X$, $X \in \mathcal{M}$, and of the *tangent bundle* $\mathrm{T}\mathcal{M} = \cup_{X\in\mathcal{M}}\mathrm{T}\mathcal{M}|_X$. A distinguishing feature of a matrix Lie group is that, once the tangent space at the identity, $\mathfrak{g} := \mathrm{T}\mathcal{G}|_{\mathrm{Id}}$, is known, it is true that $\mathrm{T}\mathcal{G}|_X = \mathfrak{g}X$ and $\mathrm{T}\mathcal{G} = \mathfrak{g}\mathcal{G}$. This allows an easy description of vector fields (hence, of differential equations) evolving in $\mathcal{G}$ and underscores the fundamental importance of $\mathfrak{g}$. Fortunately, the set $\mathfrak{g}$ possesses a great deal of structure, since it is a *Lie algebra:* a linear space closed with respect to an anti-symmetric binary operation $[\,\cdot\,,\,\cdot\,] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$:

$$\begin{aligned} &[A,B] \in \mathfrak{g}, \quad [A,B] + [B,A] = \boldsymbol{O}, \qquad A,B \in \mathfrak{g}, \\ &[A,[B,C]] + [B,[C,A]] + [C,[A,B]] = \boldsymbol{O}, \quad A,B,C \in \mathfrak{g} \qquad \text{(the Jacobi identity).} \end{aligned}$$

Examples of Lie algebras include

1. The Lie algebra of $\mathrm{GL}(n;\mathbb{F})$ is $\mathfrak{gl}(n;\mathbb{F})$, the set of all $n \times n$ matrices over $\mathbb{F}$;

2. The Lie algebra of $\mathrm{SL}(n;\mathbb{F})$ is $\mathfrak{sl}(n;\mathbb{F})$, the set of all matrices in $\mathfrak{gl}(n;\mathbb{F})$ with zero trace;

3. The Lie algebra of both $\mathrm{O}(n;\mathbb{F})$ and $\mathrm{SO}(n;\mathbb{F})$ is $\mathfrak{so}(n;\mathbb{F})$, the set of all skew-symmetric matrices in $\mathfrak{gl}(n;\mathbb{F})$;

4. The Lie algebra corresponding to $\mathrm{Sp}(n)$ is $\mathfrak{sp}(n)$, the set of all $A \in \mathfrak{gl}(2n;\mathbb{R})$ such that $AJ + JA^{\mathrm{T}} = O$.

Note that all the above are *matrix Lie algebras* (subalgebras of $\mathfrak{gl}(n;\mathbb{F})$ for some $n \in \mathbb{N}$). This is not an accident since, by the *Ado theorem,* every finite-dimensional Lie algebra is isomorphic to a matrix Lie algebra. Given that we are interested in computation, it is natural to restrict attention to finite dimensions. This is accompanied by a very welcome loss of generality: the anti-symmetric binary operation is now nothing else but the standard additive *commutator* $[X, Y] = XY - YX$.

An important feature of a Lie group is that there exists a natural map exp taking $\mathfrak{g}$ to $\mathcal{G}$. Although called the *exponential map,* its determination for a general group might require the solution of a differential equation. However, whenever the underlying Lie algebra is matricial, the map justifies its name, coinciding with the familiar exponential of a matrix.

It should be borne in mind that $\exp(\mathfrak{g}) \subseteq \mathcal{G}$ and it might well happen that $\exp(\mathfrak{g})$ is a proper subset of $\mathcal{G}$. An example is implicit in the statement that $\mathfrak{so}(n;\mathbb{F})$ is the Lie algebra of both $\mathrm{O}(n;\mathbb{F})$ and $\mathrm{SO}(n;\mathbb{F})$. A more profound example is that not every point in $\mathrm{SL}(2;\mathbb{R})$ can be attained as an exponential of a matrix from $\mathfrak{sl}(2;\mathbb{R})$. This does not represent any significant restriction on the application of the exponential map in the context of Lie-group methods, except that sometimes the step-size of the underlying time-stepping scheme might need to be restricted.

2.2 Lie-group equations

Having described the tangent bundle over $\mathcal{G}$, it is trivial to observe that every differential equation evolving in a Lie group can be written in the form

$$Y' = A(t, Y)Y, \quad t \geq t_0, \qquad Y(t_0) = Y_0 \in \mathcal{G}, \tag{2.1}$$

where $A : [t_0, \infty) \times \mathcal{G} \to \mathfrak{g}$. In addition, A should be locally Lipschitz (to ensure existence and uniqueness) and, insofar as discretization methods are concerned, it makes sense to endow it with more demanding smoothness attributes.

Equations (2.1), often under disguise, feature in a variety of applications, for example

$\mathrm{SL}(n;\mathbb{R})$:	Volume conservation, Riccati systems, Sturm–Liouville problems;
$\mathrm{O}(n;\mathbb{R})$:	Rigid mechanics, computation of Lyapunov exponents, control theory, numerical linear algebra;
$\mathrm{SO}(3,1;\mathbb{R})$:	Relativity theory;
$\mathrm{SU}(n;\mathbb{C})$:	Quantum mechanics;
$\mathrm{Sp}(n)$:	Hamiltonian mechanics.

To lift the flow $Y(t)$ from the Lie group $\mathcal{G}$ consistently with the diagram (1.2), we let $Y(t) = \exp(\Theta(t))Y_0$, where $\Theta(t)$ evolves in $\mathfrak{g}$. (There are alternatives to the exponential for *some* Lie groups, e.g. a Cayley transform can be used with $\mathrm{O}(n;\mathbb{R})$ or $\mathrm{Sp}(n)$.) This converts the Lie-group equation (2.1) to the Lie-algebra equation

$$\Theta' = \mathrm{dexp}_\Theta^{-1} A(t, \mathrm{e}^\Theta Y_0), \quad t \geq t_0, \qquad \Theta(t_0) = \boldsymbol{O}, \tag{2.2}$$

where the *dexpinv operator* $\mathrm{dexp}_\Theta^{-1}$ is defined as

$$\mathrm{dexp}_\Theta^{-1} C = \frac{\mathrm{ad}_\Theta}{\mathrm{e}^{\mathrm{ad}_\Theta} - \boldsymbol{I}} C = \sum_{l=0}^{\infty} \frac{\mathrm{B}_l}{l!} \mathrm{ad}_\Theta^l C$$

(Hausdorff 1906). Here $\{\mathrm{B}_l\}_{l \in \mathbb{Z}^+}$ are Bernoulli numbers, while the *adjoint operator* ad_Θ is defined by iterated commutation,

$$\mathrm{ad}_\Theta^l C = \begin{cases} C, & l = 0, \\ [\Theta, \mathrm{ad}_\Theta^{l-1} C], & l \in \mathbb{N}. \end{cases}$$

Once (2.2) has been discretized, the value of Θ at the new time step is translated to $\mathcal{G}$ with the exponential map.

2.3 Homogeneous spaces

Suppose that, given a smooth manifold $\mathcal{M}$, there exist a Lie group $\mathcal{G}$ and a map $\lambda : \mathcal{G} \times \mathcal{M} \to \mathcal{M}$ such that $\lambda(X_1, \lambda(X_2, \boldsymbol{y})) = \lambda(X_1 X_2, \boldsymbol{y})$ for every $X_1, X_2 \in \mathcal{G}$, $\boldsymbol{y} \in \mathcal{M}$ and $\lambda(\boldsymbol{I}, \boldsymbol{y}) = \boldsymbol{y}$ for all $\boldsymbol{y} \in \mathcal{M}$. In that case we say that $\mathcal{G}$ *acts* on $\mathcal{M}$ and that λ is a *group action.* A group action is *transitive* if for every $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{M}$ there exists $X \in \mathcal{G}$ such that $\lambda(X, \boldsymbol{y}_1) = \boldsymbol{y}_2$. A manifold that admits a transitive group action is called a *homogeneous space.*

Many important conservation laws can be described as an evolution restricted to a homogeneous space. The simplest example is when $\mathcal{M}$ is itself a Lie group, whereby $G = \mathcal{M}$ acts upon itself and λ is the usual (left or right) multiplication. The reader can affirm easily that both the n-sphere and the n-torus are homogeneous spaces with respect to the action of $\mathrm{O}(n;\mathbb{R})$. Less trivial examples include the *Stiefel manifold* (the set of all $n \times r$ real matrices X, $r \leq n$, such that $X^{\mathrm{T}} X = I$), the *Grassmann manifold* (the set of all r-dimensional linear subspaces of $\mathbb{R}^n$, alternatively all equivalence classes in a Stiefel manifold with respect to a product by a member of $\mathrm{O}(n;\mathbb{R})$), the *projective space* of all lines through the origin in $\mathbb{R}^n$ or $\mathbb{C}$, and the set of all elements of $\mathfrak{gl}(n;\mathbb{R})$ similar to a given matrix. A crucial feature of Lie-group solvers is that they can be extended to homogeneous spaces (Munthe-Kaas 1999).

Differential equations evolving on a homogeneous space $\mathcal{M}$ can be locally written in the form

$$\boldsymbol{y}' = \lambda_{\boldsymbol{y}}(F(t, \boldsymbol{y})), \quad t \geq 0, \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \in \mathcal{M}, \tag{2.3}$$

where $F : \mathbb{R}^+ \times \mathcal{M} \to \mathfrak{g}$ is Lipschitz ($\mathfrak{g}$ being the Lie algebra of $\mathcal{G}$), while the map $\lambda_{\boldsymbol{y}} : \mathfrak{g} \to \mathrm{T}\mathcal{M}$ is defined as

$$\lambda_{\boldsymbol{y}}(X) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \lambda(\exp(\varepsilon X), \boldsymbol{y})|_{\varepsilon = 0}.$$

It is quite easy to extend the diagram (1.2) so as to translate the homogeneous-space equation (2.3) first to the Lie group $\mathcal{G}$, whence to the Lie algebra $\mathfrak{g}$, solve it there and map the solution all the way to $\mathcal{M}$ in accordance with the diagram

$$\begin{tikzcd}
\boldsymbol{y}_N \arrow[d] \arrow[rr, phantom, "\boldsymbol{y}' = \lambda_{\boldsymbol{y}}(F(t,\boldsymbol{y}))"] & & \boldsymbol{y}_{N+1} \\
Y_N \arrow[d] \arrow[rr, phantom, "Y' = F(t,\lambda(Y,\boldsymbol{y}_0))Y"] & & Y_{N+1} \arrow[u] \\
\Theta_N \arrow[rr, "\text{Lie-algebraic method}", "X\Theta' = \mathrm{dexp}_\Theta^{-1} F(t,\lambda(\mathrm{e}^\Theta Y_0,\boldsymbol{y}_0))"'] & & \Theta_{N+1} \arrow[u]
\end{tikzcd}$$

Technical details were given by Munthe-Kaas (1999) but the reader might easily work out explicitly the case of *isospectral* equations, whereby $\lambda(X,Z) = XZX^{-1}$, $X, Z \in \mathrm{GL}(n;\mathbb{R})$ (Calvo, Iserles & Zanna 1997, Zanna 1999).

3 Lie-group solvers

Inasmuch as we wish to focus in this paper on methods that adhere to the general scheme (1.2), it is worthwhile to mention methods that solve Lie-group equations by different means. Perhaps the two most interesting examples are provided by the methods of *rigid frames* (Crouch & Grossman 1993)and *discrete gradients* (McLachlan, Quispel & Robidoux 1999). Another approach which we do not consider here is the recent work of Owren & Marthinsen (1999) on canonical coordinates of the second kind. Having said this, we believe that lifting Lie-group (or homogeneous-space) equations to a Lie algebra is highly useful, since the latter is a linear space: as long as we restrict the numerical solution of (2.2) solely to linear-space operations and to commutators, we are assured that the discretized solution stays (in exact arithmetic) in the algebra!

3.1 Runge–Kutta–Munthe-Kaas schemes

Perhaps the most natural means toward the solution of the Lie-algebraic equation (2.2) is by an *explicit Runge–Kutta* method. This leads to *Runge–Kutta–Munthe-Kaas (RK-MK)* schemes, which have been systematically analysed by Munthe-Kaas (1999). Applied to an arbitrary system (1.1), an explicit Runge–Kutta method has the general form

$$\begin{aligned}
\boldsymbol{k}_1 &= f(t_N, \boldsymbol{y}_N),\\
\boldsymbol{k}_2 &= \boldsymbol{f}(t_N + c_2 h, \boldsymbol{y}_N + h a_{2,1}\boldsymbol{k}_1),\\
\boldsymbol{k}_3 &= \boldsymbol{f}(t_N + c_3 h, \boldsymbol{y}_N + h(a_{3,1}\boldsymbol{k}_1 + a_{3,2}\boldsymbol{k}_2)),\\
&\vdots\\
\boldsymbol{k}_\nu &= \boldsymbol{f}(t_N + c_\nu h, \boldsymbol{y}_N + h(a_{\nu,1}\boldsymbol{k}_1 + a_{\nu,2}\boldsymbol{k}_2 + \cdots + a_{\nu,\nu-1}\boldsymbol{k}_{\nu-1})),\\
\boldsymbol{y}_{N+1} &= \boldsymbol{y}_N + h\sum_{l=1}^{\nu} b_l \boldsymbol{k}_l.
\end{aligned} \tag{3.1}$$

Applying (3.1) to the system (2.2) requires the truncation of infinite series. Note that $\Theta(t_N) = \boldsymbol{O}$ implies that $\mathrm{ad}^l_{\Theta(t_N+h)} Y = \mathcal{O}(h^l)$ for small $h > 0$, $l \in \mathbb{Z}^+$ and that $\mathrm{B}_{2l+1} = 0$ for $l \in \mathbb{N}$. To attain order $p \geq 1$, it is enough to truncate the series so that the function values are within $\mathcal{O}(h^l)$ of their exact values. For example, in the case of the 3-stage order-3 method with the *Butcher tableau*

$$\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
1 & -1 & 2 & \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}$$

we obtain after easy algebra the method

$$\begin{aligned}
K_1 &= A(t_N, Y_N),\\
K_2 &= A(t_{N+\frac{1}{2}}, \mathrm{e}^{\frac{1}{2}hK_1} Y_N),\\
K_3 &= A(t_{N+1}, \mathrm{e}^{-hK_1 + 2hK_2} Y_N),\\
\Phi &= h(\tfrac{1}{6}K_1 + \tfrac{2}{3}K_2 + \tfrac{1}{6}K_3),\\
Y_{N+1} &= \mathrm{e}^{\Phi + \frac{1}{6}[\Phi, K_1]} Y_N.
\end{aligned}$$

Note that we travel between $\mathcal{G}$ and $\mathfrak{g}$ repeatedly in each time step: a ν-stage scheme requires the calculation of ν exponentials. In the next section we compare the cost of different Lie-group methods for *linear* equations $Y' = A(t)Y$. Although RK-MK methods form no part of our analysis, it is pertinent to observe that in the linear case such methods need just a single exponential per time step.

3.2 Fer expansions

For the sake of simplicity, we restrict the exposition in this and next subsections to linear Lie-type equations

$$Y' = A(t)Y, \quad t \geq t_0, \qquad Y(t_0) = Y_0 \in \mathcal{G}, \tag{3.2}$$

where $A : \mathbb{R}^+ \to \mathfrak{g}$. Much of our discussion of this subsection can be generalized to nonlinear equations (2.1) and we refer the reader to (Zanna 1996, Zanna 1999) for details.

The main idea of a *Fer expansion* is to represent the exact solution of (3.2) as an infinite product of the form

$$Y(t) = \mathrm{e}^{B_0(t)}\mathrm{e}^{B_1(t)}\mathrm{e}^{B_2(t)}\cdots Y_N,$$

where the matrix functions $B_0, B_1, B_2, \ldots$ can be obtained recursively (Fer 1958):

$$\left.\begin{aligned} A_0(t) &= A(t_N + t), \qquad B_0(t) = \int_0^t A_0(\xi)\mathrm{d}\xi, \\ A_m(t) &= \sum_{l=1}^{\infty}(-1)^l \frac{l}{(l+1)!}\mathrm{ad}\,^l_{B_{m-1}(t)} A_{m-1}(t), \\ B_m(t) &= \int_0^t A_m(\xi)\mathrm{d}\xi, \end{aligned}\right\} \quad m \in \mathbb{N}. \tag{3.3}$$

Integration is replaced by quadrature, $\tilde{B}_m(h) \approx B_m(h)$, say, in a numerical implementation, while the infinite product is truncated. It has been proved by Iserles (1984) that, provided the quadrature is of a sufficiently high precision, the method

$$Y_{N+1} = \mathrm{e}^{\tilde{B}_0(h)}\mathrm{e}^{\tilde{B}_1(h)}\mathrm{e}^{\tilde{B}_2(h)}\cdots\mathrm{e}^{\tilde{B}_r(h)} Y_N \tag{3.4}$$

is of order $2^{r+2} - 2$.

The simplest Fer method is second-order,

$$Y_{N+1} = \mathrm{e}^{hA^*} Y_N, \qquad \text{where} \qquad A^* = A(t_{N+\frac{1}{2}}). \tag{3.5}$$

To derive a fourth-order method we let $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$, $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$, the Gauss–Legendre points in $[0, 1]$ and set

$$\tilde{B}_0(t) = \tfrac{1}{2}t[A_0(c_1 t) + A_0(c_2 t)].$$

Since $\mathrm{ad}\,^l_{\tilde{B}_0(t)} A_0(t) = \mathcal{O}\big(t^{l+1}\big)$, $l \in \mathbb{N}$, we may approximate

$$A_1(t) \approx \tilde{A}(t) = -\tfrac{1}{2}[\tilde{B}_0(t), A_0(t)] + \tfrac{1}{3}[\tilde{B}_0(t), [\tilde{B}_0(t), A_0(t)]].$$

It is easy to verify that $A_1(t) = \mathcal{O}\big(t^2\big)$ implies

$$\tilde{B}_1(h) = \tfrac{16}{27} h \tilde{A}(\tfrac{3}{4}h)$$

approximates $B_1(h)$ to order 4. This completes all the ingredients for an order-4 Fer method

$$Y_{N+1} = \mathrm{e}^{\tilde{B}_0(h)}\mathrm{e}^{\tilde{B}_1(h)} Y_N. \tag{3.6}$$

The above derivation follows upon the ideas in (Iserles 1984). In Section 4 we present a considerably better and more efficient approach.

3.3 Magnus expansions

The point of departure for our discussion is the remarkable observation of Magnus (1954), namely that the solution of the Lie-algebraic equation (2.2) (for simplicity, assuming again that $A = A(t)$) can be expanded in the form

$$\begin{aligned}\Theta(t) = & \int_{t_N}^{t} A(\xi)\mathrm{d}\xi - \tfrac{1}{2}\int_{t_N}^{t}\int_{t_N}^{\xi_1}[A(\xi_2), A(\xi_1)]\mathrm{d}\xi_2\mathrm{d}\xi_1 \\ & + \tfrac{1}{4}\int_{t_N}^{t}\int_{t_N}^{\xi_1}\int_{t_N}^{\xi_2}[[A(\xi_3), A(\xi_2)], A(\xi_1)]\mathrm{d}\xi_3\mathrm{d}\xi_2\mathrm{d}\xi_1 \\ & + \tfrac{1}{12}\int_{t_N}^{t}\int_{t_N}^{\xi_1}\int_{t_N}^{\xi_1}[A(\xi_3), [A(\xi_2), A(\xi_1)]]\mathrm{d}\xi_3\mathrm{d}\xi_2\mathrm{d}\xi_1 + \cdots.\end{aligned} \tag{3.7}$$

Although Magnus neither proved convergence nor provided a constructive rule to generate expansion coefficients of arbitrarily high order, the expansion (3.7) has been used in literally hundreds of papers in theoretical physics, quantum chemistry and control theory.

Methodical study of the Magnus expansion and its numerical implementation has been undertaken in (Iserles & Nørsett 1999). This has resulted in a general recursive scheme for the generation of expansion coefficients based on their interpretation as isomorphic to a subset of binary rooted trees, a convergence proof and comprehensive study of numerical issues and multivariate quadrature. Note that naive interpretation of (3.7) is likely to deter most numerical analysts, weaned on the belief that multivariate integration is an intrinsically expensive undertaking. Each term in the Magnus expansion requires integration over a different polytope and, on the face of it, this procedure is unlikely to result in a worthwhile computational procedure. Fortunately, multivariate integration can be performed relatively cheap by paying heed to the special structure of the problem. Being integrals of iterated commutators, all the terms in the Magnus expansion are of the form

$$I(h) = \int_{h\mathcal{S}} \mathcal{L}(A(\xi_1), A(\xi_2), \ldots, A(\xi_s))\mathrm{d}\xi_s\mathrm{d}\xi_{s-1}\cdots\mathrm{d}\xi_1,$$

where $\mathcal{S}$ is a polytope and $\mathcal{L}$ is a multilinear form. It has been proved in (Iserles & Nørsett 1999) that $I(h)$ can be approximated to order 2ν by the quadrature formula

$$h^s \sum_{\boldsymbol{l}\in C_s^\nu} b_{\boldsymbol{l}}\mathcal{L}(A_{l_1}, A_{l_2}, \ldots, A_{l_\nu}),$$

where $C_s^\nu = \{1, 2, \ldots, \nu\}^s$ and $A_k = A(t_N + c_k h)$, $c_1, c_2, \ldots, c_\nu$ being the nodes of Gauss–Legendre quadrature in $[0, 1]$. Observe that it suffices to evaluate the matrix function A at just ν locations to obtain order 2ν in *all* the terms of the Magnus expansion!

The understanding of analytic and computational features of Magnus expansions has been further assisted by a number of recent publications. Blanes, Casas, Oteo & Ros (1998) and Moan (1998) have produced a much enhanced convergence analysis; Iserles, Marthinsen & Nørsett (1999) described error control and practical implementation of Magnus methods, Iserles, Nørsett & Rasmussen (1998) proved time-symmetry of (appropriately truncated) Magnus approximants, Munthe-Kaas & Owren (1999) used the theory of free graded Lie algebras to reduce optimally the number of calculations of commutators and Zanna (1999) extended the method to nonlinear equations.

While the second-order Magnus method coincides with (3.5), the fourth-order scheme reads

$$Y_{N+1} = \mathrm{e}^{\Theta_{N+1}} Y_N, \qquad \text{where} \qquad \Theta_{N+1} = D_1 + \tfrac{1}{12}[D_1, D_2], \tag{3.8}$$

with

$$D_1 = \tfrac{1}{2}h(A_1 + A_2), \qquad D_2 = \sqrt{3}h(A_2 - A_1).$$

The function values A_i are evaluated at the Gauss–Legendre points,

$$A_1 = A(t_N + (\tfrac{1}{2} - \tfrac{\sqrt{3}}{6})h), \qquad A_2 = A(t_N + (\tfrac{1}{2} + \tfrac{\sqrt{3}}{6})h).$$

Note that $D_l \approx h^l A^{(l-1)}(t_N + \frac{1}{2}h)$, $l = 1, 2$. The replacement of function values by their linear combinations which approximate derivatives at the midpoint confers a number of important advantages, not least in the reduction in the number of terms featuring in the expansion. It can be proved that only odd powers of h feature in such expansion and that linear dependencies between expansion terms can be identified and exploited (Munthe-Kaas & Owren 1999).

To obtain a sixth-order Magnus method we let

$$D_1 = hA_2, \qquad D_2 = \tfrac{\sqrt{15}}{3}h(A_3 - A_1), \qquad D_3 = \tfrac{20}{3}h(A_3 - 2A_2 + A_1),$$

where

$$A_1 = A(t_N + (\tfrac{1}{2} - \tfrac{\sqrt{15}}{10})h), \qquad A_2 = A(t_N + \tfrac{1}{2}h), \qquad A_3 = A(t_N + (\tfrac{1}{2} + \tfrac{\sqrt{15}}{10})h),$$

whence $Y_{N+1} = \mathrm{e}^{\Theta_{N+1}} Y_N$ with

$$\begin{aligned}\Theta_{N+1} = D_1 &+ \tfrac{1}{24}D_3 + \tfrac{1}{12}[D_2, D_1] - \tfrac{1}{480}[D_3, D_2] + \tfrac{1}{240}[D_2, [D_2, D_1]] \\ &- \tfrac{1}{720}[D_1, [D_3, D_1]] - \tfrac{1}{720}[D_1, [D_1, [D_2, D_1]]].\end{aligned} \tag{3.9}$$

The derivation of higher-order Magnus expansions whilst minimising the number of commutators requires a great deal of algebra. Fortunately, this can be accomplished relatively easily by means of symbolic computation. The package `DiffMan` contains a large number of routines for the derivation and implementation of Magnus expansions and many other Lie-group solvers (Engø, Marthinsen & Munthe-Kaas 1997).

3.4 Extrapolation

Recently, Orel (2000) has considered the combination of Fer and Magnus expansions with Romberg extrapolation. Although we do not intend to review this topic, we add few preliminary details on the complexity of such methods in the next section.

4 Complexity analysis of Lie-group methods

4.1 The Fer expansion

There are three computational activities that need be compared in the course of comparing the cost of a single time step of different Lie-group solvers of the kind described in this paper: the number of function evaluations, the number of exponentiations and the number of commutators. The precise comparison among these computations is sensitive to the Lie algebra in question and to the structure of the matrix under consideration, as well as to the definition of the function A. Hence, the best we can do in a general case is to measure the complexity of each method by the triplet $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C})$, where $\boldsymbol{F}$ is the number of function evaluations and $\boldsymbol{E}$ the number of exponents, while $\boldsymbol{C}$ is the number of commutators.

We should perhaps emphasise that our figures for the number of commutators are, strictly speaking, upper bounds, although we have made an effort to make them as tight as possible. It is, at least in principle, possible that a representation of function values in a different basis or a different aggregation of terms in commutators might lead to smaller count. Another important remark is that our calculations are performed in a *free* Lie algebra. The number of commutators in a specific finite-dimensional algebra is bounded by the dimension, although it is not clear at all whether this observation can be translated to a computationally-useful strategy. For example, $\mathfrak{so}(4;\mathbb{R})$ is six-dimensional, therefore, every commutator (and every other term) can be expressed as a linear combination of just six basis terms. This can be done by using the *structure constants* of the algebra, but preliminary investigation seems to indicate that, at least for the above purpose, this procedure actually adds to computational cost (Celledoni & Iserles 1999).

We solve the equation (3.2) by means of the (truncated, discretized) Fer expansion (3.4). We recall that the order is $2^{r+2}-2$ (or less if the order of the quadrature is smaller) and that the rule in truncating the infinite expansion (3.3) is that

$$A_m(t) = \mathcal{O}(t^s) \qquad \Rightarrow \qquad \mathrm{ad}^{\,l}_{\int_0^t A_m(\xi)\mathrm{d}\xi} A_m(t) = \mathcal{O}\Big(t^{(l+1)(s+1)}\Big). \tag{4.1}$$

Let us first consider in detail the schemes (3.5) and (3.6), since their derivation offers insight into the general case. Because of (4.1), generically $A_1(t_N+t)=\mathcal{O}(t^2)$, therefore order 2 can be attained with $r=1$. The best choice of an integration scheme consistent with second order is the midpoint rule,

$$\int_0^t A_0(\xi)\mathrm{d}\xi \approx hA_0(\tfrac{1}{2}h) = hA(t_{N+\frac{1}{2}}),$$

and this gives (3.5). We conclude that for order-2 Fer we have $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (1,1,0)$.

To attain order 4 we require $\boldsymbol{E}=2$ and both A_0 and A_1 need be evaluated with (at least) order-4 quadrature. Moreover, consistently with (4.1), we approximate

$$\begin{aligned} A_1(t) \approx \tilde{A}_1(t) &= \sum_{l=1}^{2} (-1)^l \frac{l}{(l+1)!} \mathrm{ad}^{\,l}_{\tilde{B}_0(t)} A_0(t) \\ &= -\tfrac{1}{2}[\tilde{B}_0(t), A_0(t)] + \tfrac{1}{3}[\tilde{B}_0(t), [\tilde{B}_0(t), A_0(t)]], \end{aligned}$$

where $\tilde{B}_0(t) \approx B_0(t)$. Since $A_1(t) = \mathcal{O}(t^2)$, order-4 quadrature can be attained in a single function evaluation by using a Gauss–Jacobi scheme,

$$\int_0^h \tilde{A}_1(\xi)\mathrm{d}\xi \approx h\tilde{B}_1(h) = \tfrac{16}{27}h\tilde{A}_1(\tfrac{3}{4}h).$$

Pursuing this approach, we must evaluate the function $\tilde{B}_0(t)$ at $t = h$ (to be used in (3.4)) and at $t = \frac{3}{4}h$ (to evaluate $\tilde{A}_1(\frac{3}{4}h)$). Using 4th-order Gauss–Legendre quadrature, this procedure results in $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (4, 2, 2)$. This, as a matter of fact, is far from optimal. An alternative approach is to exploit an observation of Zanna (1999), namely that the multivariate quadrature of Iserles & Nørsett (1999) can be used in the present framework. Thus, truncation of the infinite series yields

$$\begin{aligned} B_1(t) = \int_0^t A_1(\xi)\mathrm{d}\xi &\approx -\tfrac{1}{2}\int_0^h\int_0^{\xi_1}[A_0(\xi_2), A_0(\xi_1)]\mathrm{d}\xi_2\mathrm{d}\xi_1 \\ &+ \tfrac{1}{3}\int_0^h\int_0^{\xi_1}\int_0^{\xi_1}[A_0(\xi_3),[A_0(\xi_2),A_0(\xi_1)]]\mathrm{d}\xi_3\mathrm{d}\xi_2\mathrm{d}\xi_1. \end{aligned}$$

Let

$$A_{0,1} = A_0((\tfrac{1}{2} - \tfrac{\sqrt{3}}{6})h), \qquad A_{0,2} = A_0((\tfrac{1}{2} + \tfrac{\sqrt{3}}{6})h)$$

and set

$$D_1 = \tfrac{1}{2}h(A_{0,1} + A_{0,2}), \qquad D_2 = \sqrt{3}h(A_2 - A_1).$$

Then

$$\int_0^h\int_0^{\xi_1}[A_0(\xi_2), A_0(\xi_1)]\mathrm{d}\xi_2\mathrm{d}\xi_1 \approx \tfrac{1}{6}[D_1, D_2],$$

$$\int_0^h\int_0^{\xi_1}\int_0^{\xi_1}[A_0(\xi_3),[A_0(\xi_2),A_0(\xi_1)]]\mathrm{d}\xi_3\mathrm{d}\xi_2\mathrm{d}\xi_1 \approx \tfrac{1}{8}[D_1,[D_1,D_2]].$$

Therefore we take

$$\begin{aligned} &\tilde{B}_0(h) = D_1, \\ &\tilde{B}_1(h) = -\tfrac{1}{12}[D_1, D_2] + \tfrac{1}{48}[D_1,[D_1,D_2]], \\ &Y_{N+1} = \mathrm{e}^{\tilde{B}_0(h)}\mathrm{e}^{\tilde{B}_1(h)}Y_N. \end{aligned}$$

We thus have $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (2, 2, 2)$.

We focus in this paper on even-order methods for two reasons. Firstly, Gauss–Legendre integration and its multivariate extension *á la* (Iserles & Nørsett 1999) lend themselves naturally to even-order methods. Secondly, appropriately discretized Magnus expansions are time-symmetric, hence automatically of an even order (Iserles et al. 1998). Therefore we now consider a sixth-order Fer expansion. We need two calculations of a matrix exponential and, truncating the series consistently with (4.1) and order six, we have

$$B_0(h) = \int_0^h A_0(\xi)\mathrm{d}\xi$$

and

$$
\begin{aligned}
B_1(h) \approx & -\tfrac{1}{2}\int_0^h\int_0^{\xi_1}[A_0(\xi_2),A_0(\xi_1)]\mathrm{d}\xi_2\mathrm{d}\xi_1 \\
& + \tfrac{1}{3}\int_0^h\int_0^{\xi_1}\int_0^{\xi_1}[A_0(\xi_3),[A_0(\xi_2),A_0(\xi_1)]]\mathrm{d}\xi_3\mathrm{d}\xi_2\mathrm{d}\xi_1 \\
& - \tfrac{1}{8}\int_0^h\int_0^{\xi_1}\int_0^{\xi_1}\int_0^{\xi_1}[A_0(\xi_4),[A_0(\xi_3),[A_0(\xi_2),A_0(\xi_1)]]]\mathrm{d}\xi_4\mathrm{d}\xi_3\mathrm{d}\xi_2\mathrm{d}\xi_1 \\
& + \tfrac{1}{30}\int_0^h\int_0^{\xi_1}\int_0^{\xi_1}\int_0^{\xi_1}\int_0^{\xi_1}[A_0(\xi_5),[A_0(\xi_4),[A_0(\xi_3),[A_0(\xi_2),A_0(\xi_1)]]]]\mathrm{d}\xi_5\mathrm{d}\xi_4\mathrm{d}\xi_3\mathrm{d}\xi_2\mathrm{d}\xi_1 .
\end{aligned}
$$

Therefore, we need to evaluate five integrals over different polytopes and do so by evaluating A_0 at Gauss–Legendre points: $c_1 = \frac{1}{2} - \frac{\sqrt{15}}{10}$, $c_2 = \frac{1}{2}$, $c_3 = \frac{1}{2} + \frac{\sqrt{15}}{10}$. Thus, $A_{0,i} = A_0(c_i h) = A(t_N + c_i h)$, $i = 1, 2, 3$, and we let

$$
\begin{aligned}
D_1 &= hA_{0,2} = \mathcal{O}(h) , \\
D_2 &= \tfrac{\sqrt{15}}{3} h(A_{0,3} - A_{0,1}) = \mathcal{O}\big(h^2\big) , \\
D_3 &= \tfrac{20}{3} h(A_{0,3} - 2A_{0,2} + A_{0,1}) = \mathcal{O}\big(h^3\big) .
\end{aligned}
$$

Our quadrature formulae assume the form

$$
\int_{h\mathcal{S}} \mathcal{L}(A_0(\xi_1), \ldots, A_0(\xi_s))\mathrm{d}\xi_s \cdots \mathrm{d}\xi_1 \approx \sum_{\boldsymbol{i}\in C_s^3} \beta_{\boldsymbol{i}} \mathcal{L}(D_{i_1}, D_{i_2}, \ldots, D_{i_s}),
$$

where C_s^3 is the set of all s-tuples with values in $\{1, 2, 3\}$. The weights $\beta_{\boldsymbol{i}}$ can be evaluated explicitly: as a matter of fact,

$$
\beta_{\boldsymbol{i}} = \int_{\mathcal{S}} \mathcal{L}((\xi_1 - \tfrac{1}{2})^{i_1}, (\xi_2 - \tfrac{1}{2})^{i_2}, \ldots, (\xi_s - \tfrac{1}{2})^{i_s})\mathrm{d}\xi_s \cdots \mathrm{d}\xi_1
$$

(Munthe-Kaas & Owren 1999). However, in our context we are not interested in their numerical values but just in the number of commutators. To this end we can aggregate coefficients and exploit symmetries, redundancies and linear dependencies according to the analysis in (Iserles & Nørsett 1999, Iserles et al. 1998, Munthe-Kaas & Owren 1999). To that end we make the following observations:

1. Since $D_i = \mathcal{O}\big(h^i\big)$, it follows that

   $$
   \mathcal{L}(D_{i_1}, D_{i_2}, \ldots, D_{i_s}) = \mathcal{O}\Big(h^{|\boldsymbol{i}|}\Big) ,
   $$

   where $|\boldsymbol{i}| = i_1 + i_2 + \cdots + i_s$. Therefore we may discard all terms such that $|\boldsymbol{i}|$ exceeds 6, the order of the method.

2. Anti-symmetry of the commutator and the Jacobi identity imply a large number of linear dependencies among terms. Such dependencies can be explored very effectively in terms of the theory of graded free algebras (Munthe-Kaas & Owren

1999) and this leads to a determination of the dimension of the underlying linear space and to constructive formation of its basis. Since limitations of space prevent us from reviewing the requisite theory, we will deal with this critical phenomenon on a case-by-case basis.

Therefore, using when necessary the Jacobi identity to eliminate terms,

1. To evaluate $\int_0^h A_0(\xi)\mathrm{d}\xi$, with $s=1$, we form a linear combination of D_1 and D_3. There are no commutators.
2. To evaluate $\int_0^h\int_0^{\xi_1}[A(\xi_2),A(\xi_1)]\mathrm{d}\xi_2\mathrm{d}\xi_1$, with $s=2$, we form a linear combination of $[D_2,D_1]$ and $[D_3,D_2]$, at the cost of two commutators.
3. The triple integral can be approximated by a linear combination of $[D_1,[D_2,D_1]]$, $[D_1,[D_3,D_1]],[D_1,[D_3,D_2]],[D_2,[D_2,D_1]]$ and $[D_2,[D_3,D_1]]$, at the cost of further six commutators.
4. The fourfold integral can be approximated by a linear combination of the terms $[D_2,[D_1,[D_2,D_1]]]$ and $[D_2,[D_1,[D_3,D_1]]]$, hence two further commutators.
5. The fivefold integral can be approximated by $[D_1,[D_1,[D_1,[D_2,D_1]]]]$ at the cost of one additional commutator.

This makes altogether eleven commutators. Some of these can be, however, computed by grouping terms togather, according to the following scheme.

$$\begin{aligned}
\tilde{B}_0(h) &= \frac{h}{18}(5A_{0,1}+8A_{0,2}+5A_{0,3}),\\
\tilde{B}_1(h) &= \tfrac{1}{12}[D_2,D_1]-\tfrac{1}{480}[D_3,D_2]-\tfrac{1}{24}[D_1,[D_2.D_1]]\\
&\quad-\tfrac{1}{720}[D_1,[D_3,D_1]]+[D_2,(\tfrac{1}{240}[D_2,D_1]-\tfrac{1}{576}[D_3,D_1])]\\
&\quad-[D_1,(\tfrac{1}{4320}[D_3,D_2]-\tfrac{1}{1440}[D_1,[D_3,D_1]]+\tfrac{1}{360}[D_1,[D_1,[D_2,D_1]]])]\\
&\quad-\tfrac{1}{80}[D_1,[D_1,[D_2,D_1]]],\\
Y_{N+1} &= \mathrm{e}^{\tilde{B}_0(h)}\mathrm{e}^{\tilde{B}_1(h)}Y_N.
\end{aligned}$$

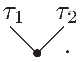
We thus deduce that the complexity of sixth-order Fer method is $(\boldsymbol{F},\boldsymbol{E},\boldsymbol{C})=(3,2,8)$.

Finally, we consider an eighth-order Fer expansion. Note that we now need three exponentials and, using Gauss–Legendre quadrature, four function evaluations. The remaining quantity that we need to evaluate is the number of commutators.

As the number and intricacy of integrals of nested commutators grows, it becomes increasingly more difficult to handle them without recourse to more succinct notation. Thus, we take a leaf out of (Iserles & Nørsett 1999) and use graphs (specifically, binary trees) to denote terms in our expansions. The basic rules are as follows:

a. The function $A(t)$ is denoted by a single-node tree, $\bullet$;

b. Given that the term $\mathcal{C}_\tau(t)$ already corresponds to a binary tree τ, we let the integral $\int_0^t \mathcal{C}_\tau(\xi)\mathrm{d}\xi$ correspond to $\begin{smallmatrix}\tau\\ |\\ \bullet\end{smallmatrix}$;

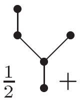
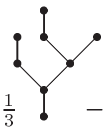
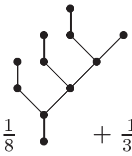
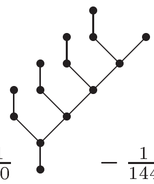
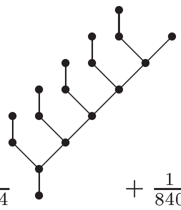
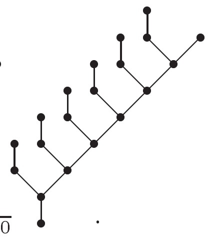
c. Given that $\mathcal{C}_{\tau_1}(t)$ and $\mathcal{C}_{\tau_2}(t)$ correspond to the binary trees τ_1 and τ_2 respectively, the commutator $[\mathcal{C}_{\tau_1}(t), \mathcal{C}_{\tau_2}(t)]$ corresponds to 

.

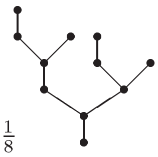
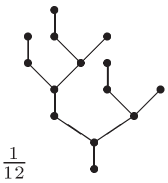
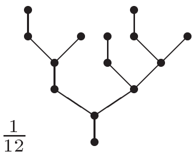
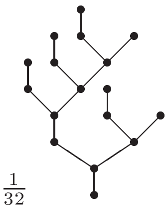
Therefore

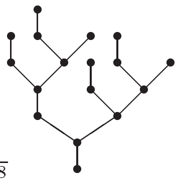
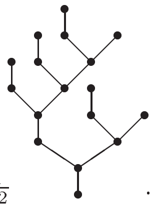
$B_0(h) \rightsquigarrow$,

while

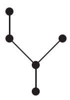
$B_1(h) \rightsquigarrow -\frac{1}{2}$  $+\frac{1}{3}$  $-\frac{1}{8}$  $+\frac{1}{30}$  $-\frac{1}{144}$  $+\frac{1}{840}$  .

To calculate $B_2(h)$ we use (4.1) to argue that $B_2(h) \approx -\frac{1}{2}\int_0^h [B_1(\xi), A_1(\xi)]\mathrm{d}\xi$. From here, after some algebra and having discarded high-order terms, we derive

$B_2(h) \rightsquigarrow -\frac{1}{8}$  $+\frac{1}{12}$  $+\frac{1}{12}$  $-\frac{1}{32}$ 

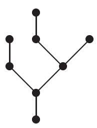
$-\frac{1}{18}$  $-\frac{1}{32}$  .

We again form linear combinations $D_i \approx h^i A_0^{(i-1)}(\frac{1}{2}h)$, $i = 1, 2, 3, 4$. Clearly, we need to evaluate no commutators to approximate $B_0(h)$. Insofar as $B_1(h)$ is concerned, recycling commutators whenever possible, we obtain the following result



We need four commutators, namely

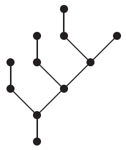
$$[D_2, D_1], [D_4, D_1], [D_3, D_2], [D_4, D_3].$$



We need seven commutators, namely

$[D_3, D_1], [D_4, D_2], [D_1, [D_3, D_1]], [D_2, [D_2, D_1]], [D_2, [D_3, D_2]], [D_1, [D_4, D_2]],$
$[D_2, [D_4, D_1]].$

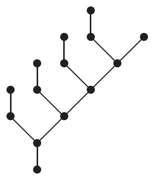
Note that we use the Jacobi identity to obtain $[D_4,[D_2,D_1]]$ from $[D_1,[D_4,D_2]]$ and $[D_4,[D_2,D_1]]$.



Now nine commutators are required, namely

$$[D_1,[D_2,D_1]],[D_1,[D_4,D_1]],[D_2,[D_3,D_1]],[D_1,[D_1,[D_2,D_1]]],$$
$$[D_2,[D_2,[D_2,D_1]]],[D_1,[D_1,[D_4,D_1]]],[D_1,[D_1,[D_3,D_1]]],$$
$$[D_1,[D_2,[D_3,D_1]]],[D_1,[D_3,[D_2,D_1]]],[D_3,[D_1,[D_2,D_1]]].$$

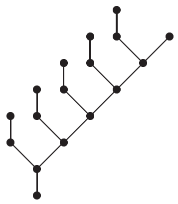
Again, we employ the Jacobi identity to express $[D_3,[D_2,D_1]]$ in terms of $[D_1,[D_3,D_2]]$ and $[D_2,[D_3,D_1]]$.



In this case we need five commutators,

$$[D_1,[D_1,[D_3,D_1]]],[D_1,[D_2,[D_2,D_1]]],[D_1,[D_1,[D_1,[D_3,D_1]]]],$$
$$[D_1,[D_1,[D_2,[D_2,D_1]]]],[D_2,[D_1,[D_1,[D_2,D_1]]]].$$

Other commutators can be eliminated with the Jacobi identity.

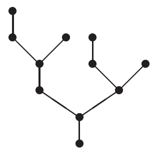


Just a single commutator is needed, namely

$$[D_1,[D_1,[D_1,[D_1,[D_2,D_1]]]]].$$

The last term is $\mathcal{O}(h^8)$ and need not be evaluated. We thus deduce that the computation of $B_1(h)$ requires 45 commutators.

Remarkably enough, the computation of $B_2(h)$ *requires no extra commutators.* The general reason is that, in general, the linear space spanned by all terms with m commutators, say, over elements $x_1,x_2,\ldots,x_r$ is exactly the same as the linear space spanned by the elements $[x_{i_1},[x_{i_2},[\ldots[x_{i_{m-1}},x_{i_m}]\cdots]]]$, where $i_1,i_2,\ldots,i_m \in \{1,2,\ldots,x_r\}$ (Munthe-Kaas & Owren 1999, Onischik 1993). (Our calculus takes place in a *free* algebra: we pay no heed to possible linear dependencies due to the dimension of the underlying Lie algebra.) A basis of the latter linear space has been, however, computed in the course of evaluating $B_1(h)$. For example, the computation of the term corresponding to



involves terms $[[D_i, D_j], [D_k, D_l]]$ for $i, j, k, l \in \{1, 2, 3, 4\}$. Yet, by the Jacobi identity

$$[[D_i, D_j], [D_k, D_l]] = [D_i, [D_j, [D_k, D_l]]] - [D_j, [D_i, [D_k, D_l]]].$$

We thus deduce that the complexity of eighth-order Fer expansion is $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (4, 3, 45)$.

4.2 Magnus methods

Two of our parameters, $\boldsymbol{F}$ and $\boldsymbol{E}$, can be easily computed for all Magnus methods. Clearly, $\boldsymbol{E} = 1$ and, for a method of order $p \geq 1$ the use of Gauss–Legendre quadrature requires $\boldsymbol{F} = \lfloor (p+1)/2 \rfloor$. We recall that in this case an appropriate truncation of the infinite series results in an even-order method. Therefore, we may restrict our attention to even p, whereby $\boldsymbol{F} = p/2$.

Second-order Magnus method coincides with our second-order Fer expansion, with the count $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (1, 1, 0)$, while the fourth-order method (3.8) yields $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (2, 1, 1)$.

Insofar as the sixth-order method (3.9) is concerned, on the face of it we need seven commutators. Yet, this can be reduced to six by aggregating terms:

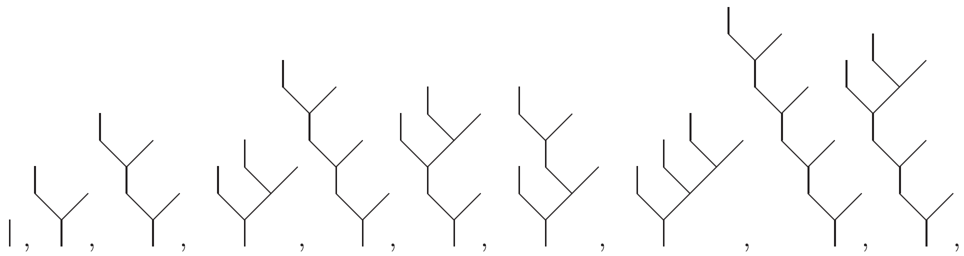
$$\begin{aligned} &C_1 = [D_1, D_2], \quad C_2 = [D_3, D_1], \quad C_3 = [D_3, D_2], \quad C_4 = [D_1, C_1], \\ &C_5 = [D_2, C_1], \quad C_6 = [D_1, C_2 + C_4], \end{aligned}$$

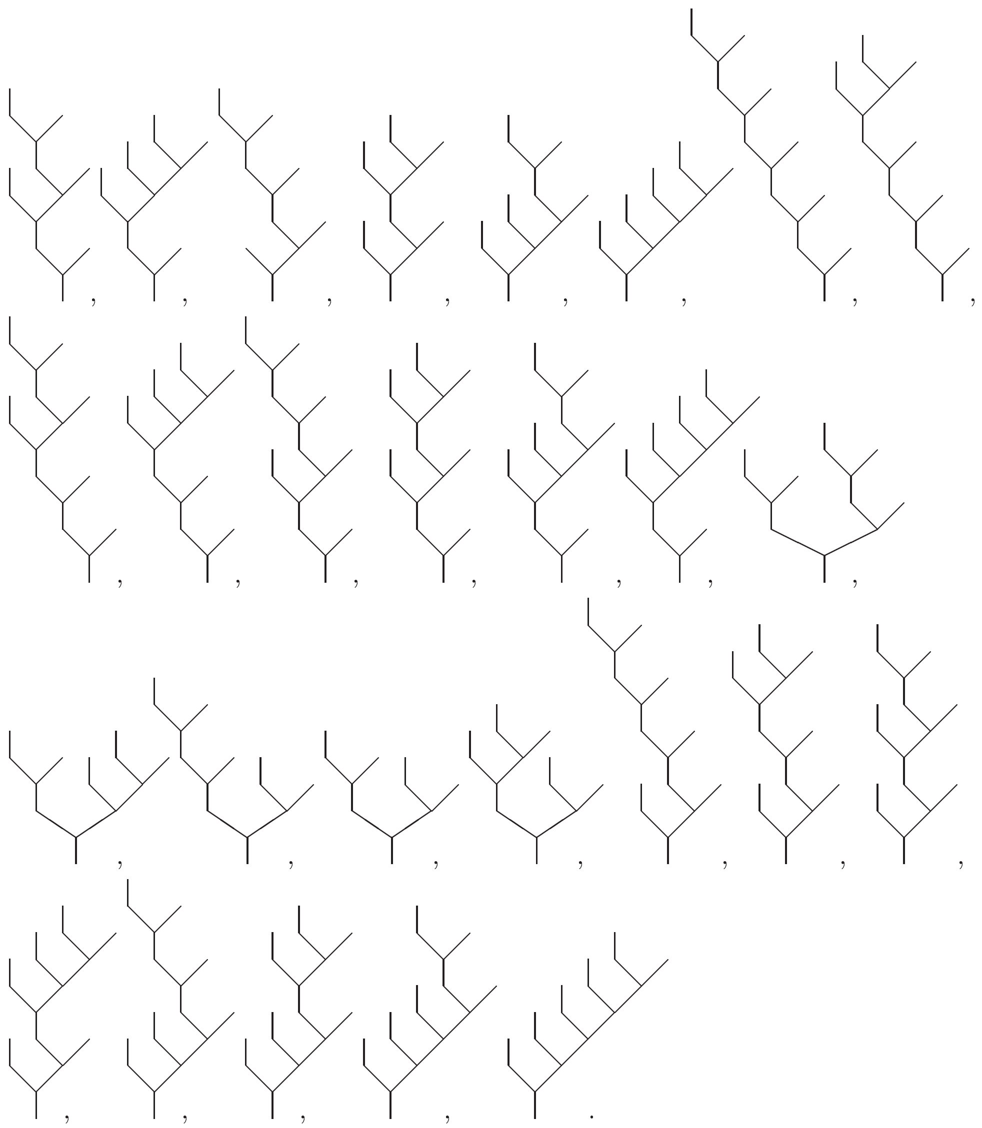
whence

$$\Theta_{N+1} = D_0 + \tfrac{1}{24} D_2 + \tfrac{1}{12} F_1 - \tfrac{1}{480} C_3 + \tfrac{1}{240} C_5 - \tfrac{1}{720} C_6.$$

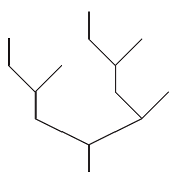
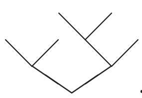
Therefore $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (3, 1, 6)$.

We next consider an eighth-order method. On the face of it, the complexity grows very fast indeed, since the expansion requires 37 terms (each corresponding to an integration across a different polytope) that correspond to the following binary trees:

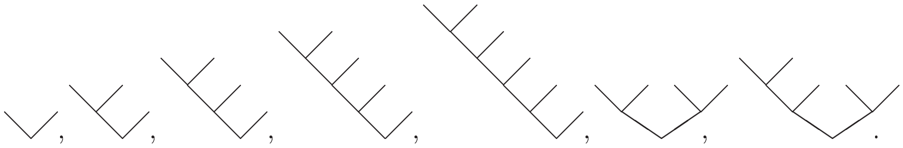




The above list can be compiled according to the theory in (Iserles et al. 1998). Seemingly, complexity is immense, but this initial impression does not reckon with the high level of redundancies inherent in a Lie-algebraic structure. We recall from (Iserles & Nørsett 1999) that a *C-tree* τ^{c} corresponding to a binary tree τ is a strictly binary tree which is obtained by excising all vertical lines, e.g.

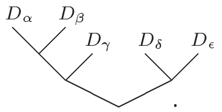
$\tau =$  $\Rightarrow$ $\tau^{\mathrm{c}} =$  .

It is the C-tree that determines the pattern of commutators that need be evaluated. Moreover, if a C-tree τ_1^{c} can be obtained from a C-tree τ_2^{c} by a sequence of vertical rotations then they require the computation of the same set of commutators: such trees are said to be *equivalent* (Iserles & Nørsett 1999). Out of all the 36 trees that include commutators, there are precisely seven equivalence classes of trees,



The commutators corresponding to a C-tree are obtained by labelling the leaves with matrices (function values $A(c_l h)$ or their linear combinations) and pruning according to the convention that each bifurcation becomes a commutator. For example,

$$[[[D_\alpha, D_\beta], D_\gamma], [D_\delta, D_\epsilon]] \quad \rightsquigarrow$$



Moreover, provided that, like in our analysis of the Fer method, $D_k = \mathcal{O}(h^k)$ and that the integral is s-dimensional, we may discard each term for which $\alpha + \beta + \gamma + \delta + \epsilon$ is even or when it exceeds 8 (the order of the method). This is motivated by a result of Munthe-Kaas & Owren (1999), namely that, with quadrature nodes symmetrically distributed with respect to $\frac{1}{2}$, linear combinations of quadratures can be expanded in odd powers of h.

This, however, does not exhaust the range of all possible savings, since some terms can be expressed as a linear combination of other terms. In particular, we claim that terms corresponding to the last two C-trees can be expressed in this fashion. We have already proved this for the sixth C-tree in Subsection 4.2 and the general statement follows from the theory of graded free Lie algebras in (Munthe-Kaas & Owren 1999). We deduce that the time-stepping of an eighth-order Magnus method requires the computation of all commutators originating in the first five C-trees. These, however, are *exactly* the commutators that we need in a single step of an eighth-order Fer method! We just deduce that $(\boldsymbol{F}, \boldsymbol{E}, \boldsymbol{C}) = (4, 1, 45)$.

As a matter of fact, identical reasoning is valid for all Magnus methods of order ≥ 6: we need to valuate exactly the same commutators as for a Fer expansion of the same order! Given that Magnus requires just a single evaluation of the exponential matrix and that the number of function evaluations is $p/2$ in both cases, we deduce that *the Magnus method is always cheaper than the Fer method.*

4.3 Combined Fer–Magnus methods

Iserles & Nørsett (1999) address themselves briefly to the possibility of combining Fer and Magnus expansions. Note from (3.3) that $\mathrm{e}^{B_0(h)} Y_N$ is nothing else but a second-order Magnus method. It is natural to consider its replacement with, say, a fourth-order Magnus scheme. This yields a new function A_1 and the procedure can be repeated. It has been proved in (Iserles & Nørsett 1999) that the use of an order-s truncation of Magnus in each Fer step results in a method of order $s(s^r - 1)/(s - 1)$ in (3.4). Moreover, it is possible to use Magnus expansions of different orders in different parts of the Fer algorithm. We have not computed explicitly the cost of such combinations. However, it is implied in our analysis of the Fer and Magnus expansions that, except for low-order methods where extra cancellations might take place, complexity is always optimised for the 'straight' Magnus expansion.

5 Approximation of the exponential: complexity issues

5.1 Overview: classical methods

All the methods presented in Section 3 require the evaluation (or a suitable approximation) of the matrix exponential once (and often repeatedly) in each time step. This is a major part of the all computation whose impact cannot be neglected in complexity analysis.

The matrix exponential can be computed exactly using high-precision techniques or just exploiting the structure of the matrix B that we want to exponentiate. This is the case for important sets of matrices for which it is possible to compute explicitely and cheaply eigenvalues and eigenvectors, for instance circulant Toeplitz matrices, and matrices in $\mathfrak{sl}(2;\mathbb{R})$ or in $\mathfrak{so}(n;\mathbb{R})$ with $n \leq 4$.

In all other cases, in realistic computational framework we need to use sufficiently accurate approximants of e^{tB}. We thus commence this section with a brief survey of the most popular techniques used for the approximation of the exponential map, commenting why most of them are unsuitable for the task in hand.

We commence with the obvious comment that the replacement of exact exponential functions in an order-p Lie-group integrators with order-p approximants, $R(tB) \approx \mathrm{e}^{tB}$, results in an order-$p$ method. Yet, *this need not mean that the new algorithm is itself a Lie-group integrator,* since it is entirely possible that $R(tB) \notin \mathcal{G}$ for some $B \in \mathfrak{g}$. Thus, the crucial challenge of this section is to identify and analyse methods that are guaranteed to map a Lie algebra into a Lie group. In the sequel we endeavour to classify the approximation methods in terms of their potential to respect this constraint.

Unless we are assured that $R(B) \in \mathcal{G}$ for all $B \in \mathfrak{g}$, we might use one of two strategies.

- We can use R to evaluate the exponential to machine precision by *scaling and squaring* (Moler & Van Loan 1978). Specifically, we evaluate $[R(B/2^s)]^{2^s}$, where $s \in \mathbb{N}$ is large enough, so that the error $\|R(B/2^s) - \mathrm{e}^{B/2^s}\|$ is less than the machine epsilon. Note that, computing powers recursively, we require just s matrix products. This, indeed, is the method of computation of the `expm` function in the MATLAB package.

  The choice of s is governed not just by the accuracy of R and the machine precision, but also by the *condition number* of B, which measures its sensitivity to small perturbations. Moler & Van Loan (1978) have already investigated the theory governing the determination of the condition number for some exponential approximants, while a pioneering work on the impact of conditioning on complexity is (Cucker & Smale 1998).

- The approximant $R(B)$ can be *projected* on the underlying Lie group. Thus, for example, $R(B)/[\det R(B)]^{1/n} \in \mathrm{SL}(n;\mathbb{R})$ while one can always obtain a member of $\mathrm{O}(n;\mathbb{R})$ by subjecting $R(B)$ to the QR (or, even better, polar) decomposition and retaining just the orthogonal part (Higham 1997). This is a perfectly valid approach for many Lie groups, although computer experimentation indicates that sometimes it might degrade stability.

Building upon the discussion in (Moler & Van Loan 1978), we distinguish four different categories of approximants to the matrix exponential: iterative methods, rational approximants, factorization methods, and splitting methods. We assume without loss of generality that tB is in the neighbourhood of the origin. The reason for this is that global results on the asymptotic behaviour of approximate solutions of Lie-group integrators do not rely on global proprieties of exp, the local diffeomorphism from $\mathfrak{g}$ to $\mathcal{G}$. Such results, have been obtained in (Faltinsen 1998) using backward error analysis techniques.

- *Polynomial approximants* are typically based on the construction of approximations of $\mathrm{e}^{tB}\boldsymbol{v}$ for some $\boldsymbol{v} \in \mathbb{R}^n$ that lie in the m-dimensional *Krylov subspace*

  $$K_m(B, \boldsymbol{v}) = \mathrm{Sp}\{\boldsymbol{v}, B\boldsymbol{v}, \ldots, B^{m-1}\boldsymbol{v}\}.$$

  Such methods are based on the idea of projecting the matrix B onto the Krylov subspace and exponentiating the $m \times m$ projected matrix in place of the original one. The construction of $K_m(B, \boldsymbol{v})$ is iterative, each iteration bearing a cost of $\mathcal{O}(n^2)$. The convergence of such methods is superlinear for Krylov subspaces of sufficently large dimension: specifically, for symmetric negative-definite B and $m \geq \sqrt{\|tB\|}$; in general the convergence depends on the location of the eigenvalues of B in the complex plane. Extensive treatement of these issues can be found in (Kosloff, Tal-Ezer & Cerjan 1992, Gallopoulos & Saad 1992, Hochbruck & Lubich 1997) and (Druskin & Knizhnerman 1995). These methods are not suitable for computing e^{tB}, unless it is applied to a vector $\boldsymbol{v}$. However, the most important observation is that, unless the Krylov-subspace approximant is computed to machine precision, there is no reason for a Krylov subspace approximant of e^{tB}, $B \in \mathfrak{g}$, to reside in the Lie group.

- *Rational approximants* $R(tB) = P(tB)/Q(tB)$, where P and Q are polynomials, $Q(0) = 1$, are in general more demanding computationally. The costs are typically of the form qn^3 where q is the maximum of the degrees of the two polynomials. The approximants normally obey $R(tB) = \exp(tB) + \mathcal{O}(t^{p+1})$ for some $p \in \mathbb{N}$ and this allows to match them more easily with the underlying Lie-group method, consistently with its order. We will discuss rational approximants in greater detail in the sequel.

- *Factorization methods* compute a suitable similarity transformation of B, $\Lambda = T^{-1}BT$ where Λ is a matrix with simpler structure, e.g. diagonal or triangular.

In this case $e^{t\Lambda}$ is easy to evaluate and $e^{tB} = Te^{t\Lambda}T^{-1}$. In (Parlett 1976) a recursive formula for deriving the strictly upper triangular part of $e^{t\Lambda}$ for an upper triangular matrix Λ is presented. The main cost of the work for computing e^{tB} with such methods is in the construction of the factorization. There is no truncation error due to the scheme and what counts is the accumulation of roundoff error, that in general grows with the dimension of the matrix and is sensitive to the conditioning of the matrix B.

- *Splitting methods* have been considered by many authors in different contexts and they have proved to be effective in the numerical solution of multivariate partial differential equations and in the geometric integration of Hamiltonian ordinary differential equations (Sanz-Serna & Calvo 1994, Feng & Shang 1995, McLachlan 1995, Schatzman 1997). Their use in the present context has been advocated and analysed in (Stickel 1994, Howland 1983, Celledoni & Iserles 1998, Celledoni & Iserles 1999) and is explained in greater detail in the remainder of this section.

Before we embark on the exposition of the main and most versatile approach, namely splitting, we wish to expand briefly on the use of rational approximants and, with greater generality, analytic approximants defined in the Banach algebra of a matrix $B \in \mathfrak{g}$. *Some* approximants of this form are assured to map *some* Lie algebras to Lie groups.

Theorem 1 *(Celledoni & Iserles 1998) Consider the Lie group $\mathcal{G} := \{y \in \mathrm{GL}(n;\mathbb{R}) : YPY^{\mathrm{T}} = P\}$ where $P \in \mathrm{GL}(n;\mathbb{R})$. The corresponding Lie algebra is $\mathfrak{g} = \{B \in \mathfrak{gl}(n;\mathbb{R}) : BP + PB^{\mathrm{T}} = \boldsymbol{O}\}$. Let φ be a function, analytic in a neighbourhood $\mathcal{U}_0$ of 0, with $\varphi(0) = 1$ and $\varphi'(0) = 1$. If*

$$\varphi(z)\varphi(-z) \equiv 1, \qquad z \in \mathcal{U}_0, \tag{5.1}$$

then

$$\varphi(tB) \in \mathcal{G} \qquad B \in \mathfrak{g} \tag{5.2}$$

for sufficiently small $t \in \mathbb{R}^+$. Moreover if there exists $B \in \mathfrak{g}$ such that $\sigma(B) \neq \{0\}$, where $\sigma(B)$ denotes the spectrum of B, then (5.2) implies (5.1).

An important example of functions φ which are consistent with the condition (5.1) is provided by *diagonal* Padé approximants $R_m(z) = P_m(z)/P_m(-z)$, where

$$P_m(z) = \sum_{l=0}^{m} \binom{m}{l} \frac{(2m-l)!}{(2m)!} z^l.$$

Moreover $R_m(z) = e^z + \mathcal{O}\left(z^{2m+1}\right)$ and the approximants R_m are A-stable. The cost of evaluating $P_m(tB)$ and $P_m(-tB)$ is mn^3 flops, while the inversion of the latter matrix adds $2n^3$ flops.

Examples of Lie groups that conform with the conditions of Theorem 1 are $\mathrm{O}(n;\mathbb{R})$ and $\mathrm{Sp}(n)$. Moreover, replacing transpose by Hermitian transpose, it is easy to extend the theorem to $\mathrm{U}(n;\mathbb{C})$. Insofar as such groups are concerned, diagonal Padé approximants are an attractive computational option. Having said this, they do not offer a universal panacea, as is clear from the next result.

Theorem 2 *(Feng & Shang 1995) The only function φ analytic in a neighbourhood of the origin with $\varphi(0) = 1$ and $\varphi'(0) = 1$ and such that $\varphi(tB) \in \mathrm{SL}(n;\mathbb{R})$ for all $B \in \mathfrak{sl}(n;\mathbb{R})$, $n \geq 3$, is the (exact) exponential map.*

This theorem provides the motivation behind splitting methods which so far constitute the only truly universal algorithm, capable of modelling an exponential map from every matrix Lie algebra to its Lie group.

5.2 Overview: Splitting methods

The material of this subsection is based upon (Celledoni & Iserles 1998) and (Celledoni & Iserles 1999). The main idea is, given an $n \times n$ matrix $B \in \mathfrak{g}$, to split it in the form

$$B = \sum_{i=1}^{k} B_i, \tag{5.3}$$

subject to the following conditions:

1. Each matrix B_i resides in $\mathfrak{g}$;
2. It is easy to compute e^{tB_i}, $i = 1, 2, \ldots, k$, explicitly; and
3. The product
$$R(tB) = \mathrm{e}^{tB_1}\mathrm{e}^{tB_2}\cdots\mathrm{e}^{tB_k} \tag{5.4}$$
approximates e^{tB} to sufficiently high order.

To which we may add

4. Each product $\mathrm{e}^{tB_i}\mathrm{e}^{tB_{i+1}}$, $i = 1, 2, \ldots, k-1$, is easy to compute.

Noting that $\mathrm{e}^{tB_i} \in \mathcal{G}$, $i = 1, 2, \ldots, k$, implies that $R(tB) \in \mathcal{G}$, we take (5.4) as our approximation to the exponential of tB.

In theory, it is possible to eliminate the truncation error in splitting methods by constructing a *commutative splitting* (i.e. such that $[B_i, B_j] = 0$, $1 \leq i, j \leq k$). An example can be found in (Stickel 1994), which describes an interesting algorithm based on a recursive construction of a commutative splitting by suitable projections via the sign function of B. The computation of the sign function of B is obtained iteratively and the cost of each iteration is of order $\mathcal{O}(n^3)$. Once the sign function is calculated to high accuracy, the matrix B is split into two rank-deficient matrices B_1 (of rank r) and B_2 (of rank $n-r$). The cost of evaluating e^{tB_1} and e^{tB_2} is reduced to the cost of evaluating the exponentials of $r \times r$ and $(n-r) \times (n-r)$ matrices respectively, plus the cost of reordering linearly independent rows or columns of B_1 and B_2 that are obtained via QR factorization. Although the possibility of constructing a commutative splitting sounds very attractive, it seems that the computational costs are comparable to those of finding eigenvalues and eigenvectors of the given matrix. A technique based on the use of fourth roots of matrices is described by the same author in (Stickel 1990).

In the next subsection we now construct splittings (5.3) into a sum of low-rank matrices B_i, based on the ideas described in (Celledoni & Iserles 1998). The computational cost of these procedures is cn^3, where c is a constant depending on the order of approximation. This is similar to diagonal Padé approximants. Yet, in the case of splitting, if the task in hand requires to approximate $\mathrm{e}^{tB}\boldsymbol{v}$, where $\boldsymbol{v} \in \mathbb{R}^n$, the cost drops to cn^2, while Padé approximants remain an $\mathcal{O}(n^3)$ process.

5.3 Low-rank splittings

Given a general splitting (5.3), we can expect $R(tB) = \mathrm{e}^{tB} + \mathcal{O}(t^2)$, hence order 1. Special choices of the B_is may lead to higher-order methods, an issue that has been discussed extensively in (Sheng 1989) and (Schatzman 1997). In our case, though, we have the further desiderata, namely that all B_is lie in $\mathfrak{g}$ and their exponentials can be computed with ease. One effective means to this end is to choose the B_is as low-rank matrices. Specifically, we choose each B_i of the form

$$B_i = \sum_{l=1}^{p} \boldsymbol{\alpha}_{i,l}\boldsymbol{\beta}_{i,l}^{\mathrm{T}},$$

where $\boldsymbol{\alpha}_{i,l}, \boldsymbol{\beta}_{i,l} \in \mathbb{R}^n$, $i = 1, 2, \ldots, p$, $p \geq 1$ is small and, needless to say, we take care to ascertain that $B_i \in \mathfrak{g}$. Let $\boldsymbol{\alpha}_i = [\boldsymbol{\alpha}_{i,1}, \boldsymbol{\alpha}_{i,2}, \ldots, \boldsymbol{\alpha}_{i,p}]$, $\boldsymbol{\beta}_i = [\boldsymbol{\beta}_{i,1}, \boldsymbol{\beta}_{i,2}, \ldots, \boldsymbol{\beta}_{i,p}]$, and define $D_i = \boldsymbol{\beta}_i^{\mathrm{T}}\boldsymbol{\alpha}_i \in \mathfrak{gl}(p; \mathbb{R})$. Then

$$\mathrm{e}^{tB_i} = \boldsymbol{I} + t\boldsymbol{\alpha}_i\psi(tD_i)\boldsymbol{\beta}_i^{\mathrm{T}}, \tag{5.5}$$

where $\psi(z) = (\mathrm{e}^z - 1)/z$ is an analytic function. Note that $\psi(tD_i)$ must be evaluated exactly, but this is typically easy since it is an $p \times p$ matrix.

We illustrate this construct by a number of examples, commencing from $B \in \mathfrak{so}(n; \mathbb{R})$. The simplest choice is $p = 2$, $k \leq n-1$. We let $B^{[0]} := B$ and set

$$B_1 = \boldsymbol{b}_1^{[0]}\boldsymbol{e}_1^{\mathrm{T}} - \boldsymbol{e}_1\boldsymbol{b}_1^{[0]\mathrm{T}} \in \mathfrak{so}(n; \mathbb{R}),$$

where $\boldsymbol{b}_1^{[0]}$ is the first column of $B^{[0]}$ and $\boldsymbol{e}_l$ is the lth vector of the canonical basis of $\mathbb{R}^n$. We continue this procedure by letting for $i = 1, 2, \ldots$

$$B^{[i]} := B^{[i-1]} - B_i \qquad \text{and} \qquad B_{i+1} = \boldsymbol{b}_{i+1}^{[i]}\boldsymbol{e}_{i+1}{}^{\mathrm{T}} - \boldsymbol{e}_{i+1}\boldsymbol{b}_{i+1}^{[i]}{}^{\mathrm{T}} \in \mathfrak{so}(n; \mathbb{R}),$$

where $\boldsymbol{b}_{i+1}^{[i]}$ is the first nonzero column of $B^{[i]}$.

Recalling that D is a 2×2 matrix with zeros on the diagonal, we observe that $\psi(tD)$ can be computed explicitely at a cost of ten flops plus the cost of a single sine and cosine evaluation, and the cost for computing n such functions is $\mathcal{O}(n)$. The substantive part of the computation is the formation of products of all the terms (5.5). Careful analysis, based on the fact that inner products with respect to vectors of the canonical basis are very inexpensive and that the vectors $\boldsymbol{b}_{i+1}^{[i]}$ have $n - i$ non-zeros, results in

$$\sum_{i=1}^{n-1}[2n(n-i) + 2n + n(n-i) + n(n-i)] = 2n^3 - 2n$$

flops. We will see in the sequel that the cost of higher-order approximants of similar form is just a scalar multiple of the cost of the above first-order method.

Similar strategy can be used for decomposing matrices $B \in \mathfrak{sl}(n;\mathbb{R})$ (Celledoni & Iserles 1998).

In the case of differential equations that evolve on an homogeneous space $\mathcal{M} \subset \mathbb{R}^n$, acted upon by the matricial Lie group $\mathcal{G} \subset \mathrm{GL}(n;\mathbb{R})$, the approximation of the exponential map is cheaper, because the computational task is to evaluate $R(tB)\boldsymbol{v} \approx \mathrm{e}^{tB}\boldsymbol{v}$ where $\boldsymbol{v} \in \mathcal{M}$ is an n-dimensional vector. In this case the total cost amounts to just $2n^2$ flops. This cost-reduction by an order of magnitude is universal to splitting methods and we will not mention it in the sequel in reference to similar algorithms.

The computational costs of this approach are comparable to polynomial approximants (e.g. Krylov subspace methods (Kosloff et al. 1992, Hochbruck & Lubich 1997, Gallopoulos & Saad 1992)), yet, unlike the latter, they are guaranteed to map a Lie-algebraic element to a Lie group.

5.4 Higher-order schemes

Techniques for improving the order of low-rank methods are described in (Celledoni & Iserles 1998). The common denominator to all such techniques is that they compose suitably scaled approximants to e^{tB}.

Given an arbitrary splitting (5.3), we may replace the scheme (5.4) by a straightforward generalisation of the well-known *Strang splitting,*

$$R(tB) = \mathrm{e}^{\frac{1}{2}tB_1}\mathrm{e}^{\frac{1}{2}tB_2}\cdots\mathrm{e}^{\frac{1}{2}tB_{k-1}}\mathrm{e}^{tB_k}\mathrm{e}^{\frac{1}{2}tB_{k-1}}\cdots\mathrm{e}^{\frac{1}{2}tB_2}\mathrm{e}^{\frac{1}{2}tB_1}. \tag{5.6}$$

The outcome is an order-two method. As long as the B_i are low-rank matrices, similar to the examples of the previous subsection, the cost of computing (5.6) is roughly double that of (5.4), hence it amounts to $4n^3 + \mathcal{O}(n^2)$ flops.

The order of the Strang splitting can be increased by using techniques due to Yoshida (1990). Since $R(tB) = \mathrm{e}^{tB} + \mathcal{O}(t^3)$, where $R(tB)$ has been given in (5.6), it follows that

$$R(\gamma_1 tB)R(\gamma_2 tB)R(\gamma_3 tB), \tag{5.7}$$

where $\gamma_1 = \gamma_3 = 2/3 + \sqrt[3]{2}/3 + \sqrt[3]{4}/6$, $\gamma_2 = -(1+\sqrt[3]{2})^2/3$, is of order 4. This technique was employed by Yoshida (1990) to enhance the order of symplectic integrators for Hamiltonian systems and it can be used in a similar manner in the present framework. The order can be increased further in increments of two by composing Yoshida approximants (5.7).

5.5 Approximations of the exponential matrix based on canonical coordinates of the second kind

Let B be an element in $\mathfrak{g} \subseteq \mathfrak{gl}(n;\mathbb{R})$ be a d-dimensional Lie algebra, $d \le n^2$. We expand

$$B = \sum_{i=1}^{d} \beta_i F_i$$

where $\{F_1, F_2, \ldots, F_d\}$ is a basis of $\mathfrak{g}$. It is then possible to find functions $g_1, g_2, \ldots, g_d$, analytic in the neighbourhood of the origin, so that for sufficiently small $t \ge 0$

$$\mathrm{e}^{tB} = \mathrm{e}^{g_1(t)F_1}\mathrm{e}^{g_2(t)F_1}\cdots\mathrm{e}^{g_d(t)F_d}$$

(Wei & Norman 1963). This is known as an expansion in *canonical coordinates of the second kind.* However, the explicit form of such functions is unknown and, even in the simple case of $\mathfrak{sl}(2;\mathbb{R})$ they are defined only implicitly, as a solution of a nonlinear differential system. Instead, we seek to *approximate* the functions g_i. Specifically, we replace each g_1 by a p-degree polynomial α_i such that

$$F(tB) = \mathrm{e}^{\alpha_1(t)F_1}\mathrm{e}^{\alpha_2(t)F_2}\cdots\mathrm{e}^{\alpha_d(t)F_d} = \mathrm{e}^{tB} + \mathcal{O}\big(t^{p+1}\big)\,. \tag{5.8}$$

Differentiating (5.8) leads to the order conditions

$$\begin{aligned}
p \geq 1: &\quad g_i'(0) = \beta_i, \qquad i = 1,2,\ldots,d,\\
p \geq 2: &\quad \sum_{i=1}^{d}\alpha_i''(0)F_i = -\sum_{i=1}^{d}\beta_i\sum_{j=1}^{i-1}[F_j,F_i]\beta_j,\\
p \geq 3: &\quad \sum_{i=1}^{d}\alpha_i'''(0)F_i = -\sum_{i=1}^{d}\left(2\alpha_i''(0)\sum_{j=1}^{i-1}[F_j,F_i]\beta_j + \beta_i\left\{\sum_{j=1}^{i-1}\sum_{l=1}^{j-1}2[F_l,[F_j,F_i]]\beta_j\beta_l\right.\right.\\
&\qquad\qquad \left.\left. +\ [F_j,[F_j,F_i]]\beta_j^2 + [F_j,F_i]\alpha_j''(0)\right\}\right).
\end{aligned}$$

The above can be simplified by using the *structure constants* of $\mathfrak{g}$: real numbers $c_{j,l}^i$ such that

$$[F_j,F_l] = \sum_{i=1}^{d}c_{j,l}^iF_i, \qquad i,j,l = 1,2,\ldots,d.$$

Thus, for example, the $p \geq 2$ condition becomes

$$\alpha_i''(0) = -\sum_{j=2}^{d}\sum_{l=1}^{j-1}\beta_j c_{l,j}^i\beta_l, \qquad i = 1,2,\ldots,d.$$

Higher-order methods of this form, as well as an alternative approach, implementing coordinates of the second kind in tandem with the Strang splitting (thereby rendering the approximant amenable to Yoshida-type techniques) can be found in (Celledoni & Iserles 1999)

Correct implementation of the algorithm (5.8) is essential and the crucial step is the right choice of the basis of $\mathfrak{g}$, so as to reduce computational complexity. We illustrate this in the case of the algebra $\mathfrak{so}(n;\mathbb{R})$, whereby $d = \frac{1}{2}(n-1)n$. We take as our basis the matrices

$$F_{i,j} = \boldsymbol{e}_i\boldsymbol{e}_j^{\mathrm{T}} - \boldsymbol{e}_j\boldsymbol{e}_i^{\mathrm{T}}, \qquad i = 1,2,\ldots,n, \quad j = i+1,i+2,\ldots,n,$$

where $\boldsymbol{e}_i \in \mathbb{R}^n$ is the ith unit vector. In other words, $B = \sum_{i=1}^n\sum_{j=i+1}^n b_{i,j}F_{i,j}$. Note that $U(t) := \mathrm{e}^{tF_{i,j}}$ is simply an *Euler rotation* in the (i,j) plane: it departs from the identity matrix just at the elements

$$\begin{bmatrix} U_{i,i} & U_{i,j}\\ U_{j,i} & U_{j,j}\end{bmatrix} = \begin{bmatrix} \cos(tb_{i,j}) & \sin(tb_{i,j})\\ -\sin(tb_{i,j}) & \cos(tb_{i,j})\end{bmatrix}.$$

The structure constants are readily available, since

$$[F_{i,j}, F_{l,k}] = \begin{cases} -F_{j,k}, & i = l,\ j \neq k, \\ -F_{i,l}, & i \neq l,\ j = k, \\ F_{i,k}, & i \neq k,\ j = l, \\ F_{j,l}, & i = k,\ j \neq l, \\ O, & \text{otherwise.} \end{cases}$$

Note the very high sparsity of structure constants. It is precisely this phenomenon that allows to reduce complexity. For example, second-order conditions become

$$\begin{aligned} p \geq 1: &\quad \alpha'_{i,j}(0) = b_{i,j}, \\ p \geq 2: &\quad \alpha''_{i,j}(0) = \sum_{s=j+1}^{n} b_{j,s} b_{i,s} - \sum_{r=i+1}^{j-1} b_{r,j} b_{i,r} + \sum_{r=1}^{i-1} b_{r,j} b_{r,i}. \end{aligned}$$

The cost of obtaining the coefficients for the second-order method is thus just $\frac{1}{2}n^3 + \mathcal{O}(n^2)$ flops.

Similar observations about the *Strang splitting,* as well as other Lie algebras, can be found in (Celledoni & Iserles 1999). In general, the idea is to choose a basis of $\mathfrak{g}$ that yields extensive sparsity in the structure constants. A universally good such choice is provided by the *root space decomposition* of the (non-nilpotent) Lie algebra $\mathfrak{g}$ (Varadarajan 1984).

5.6 Using low-rank splittings in Lie-group integrators

In this section we present some numerical results concerning the efficiency of the different methods of approximation of the exponential matrix when used in tandem with a Lie-group integrator. We consider the Magnus method of order 2 and 4, as well as an RK-MK scheme (3.1) of order 4, as applied to two test problems. The first is a linear ordinary differential equation evolving on $\mathbb{R}^n$ and the problem is formulated in terms of an action of $\mathrm{SL}(n;\mathbb{R})$ on the configuration space, while the second test problem is a nonlinear matricial equation describing an *isospectral flow* evolving on $\mathrm{Sym}(n;\mathbb{R})$, the manifold of $n \times n$ symmetric matrices.

In the first example, we combine the Magnus method with three different techniques for the computation of the exponential matrix: the built-in MATLAB function `expm` the low-rank splitting methods of order 4 (Strang splitting enhanced with the Yoshida technique) and a Krylov subspace projection approximant.

The Krylov subspace approach is suitable since in the present case we approximate $\mathrm{e}^{tB}\boldsymbol{v}$ for some $\boldsymbol{v} \in \mathbb{R}^n$. The specific Krylov approximant that we use is obtained by constructing an orthogonal basis $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$, for the Krylov subspace $\mathrm{Sp}\{\boldsymbol{v}, A\boldsymbol{v}, \ldots, A^{k-1}\boldsymbol{v}\}$, via the Arnoldi algorithm and by computing $H_k = V_k^{\mathrm{T}} B V_k$, where $V_k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k]$. The approximant of $\mathrm{e}^{tB}\boldsymbol{v}$ is $\boldsymbol{w}_k = V_k \mathrm{e}^{tH_k}\boldsymbol{e}_1\|\boldsymbol{v}\|$ and we choose k so that $h_{k+1,k}\mu_k\|\boldsymbol{v}\| \leq 10^{-16}$, where μ_k is the absolute value of the kth component of $\mathrm{e}^{H_k}\boldsymbol{e}_1$. Small magnitude of this quantity corresponds in our experiments to analogous value of $\|\mathrm{e}^{tB}\boldsymbol{v} - \boldsymbol{w}_k\|$.

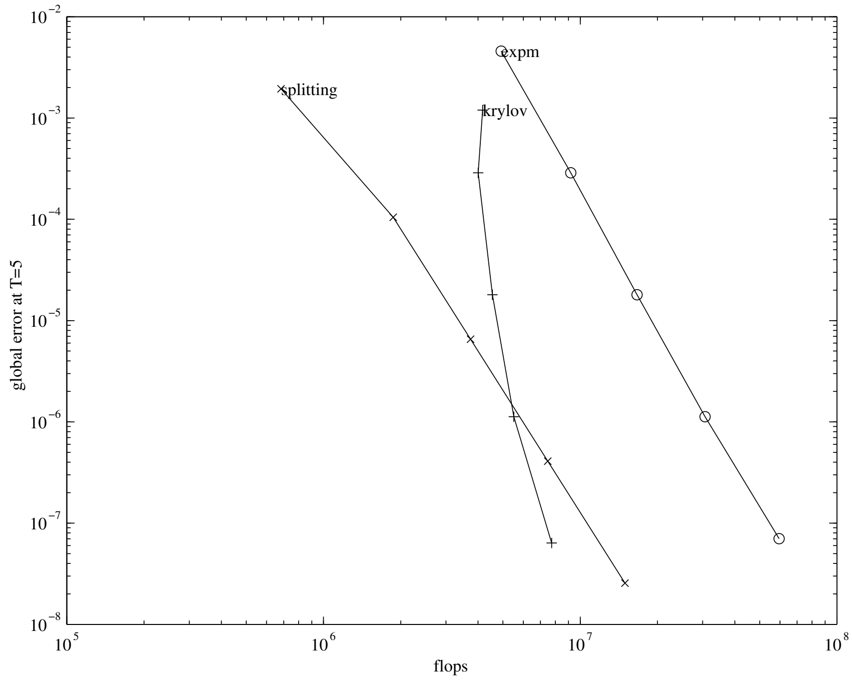


Figure 1: Magnus method of order 4, action of $\mathrm{SL}(30;\mathbb{R})$ on $\mathbb{R}^{30}$: Three different algorithms to approximate the exponential.

We have integrated the system $\boldsymbol{y}' = A(t)\boldsymbol{y}$, $\boldsymbol{e}_1 = \boldsymbol{y}_0 \in \mathbb{R}^{30}$, where $a_{i,i+1}(t) = 1$, $a_{30,1}(t) = -t$ and A is zero otherwise. In Figure 1 we compare the number of flops corresponding to the same global error for the different implementations.

The experiments have been performed as follows: we have first computed the global error $\boldsymbol{g}_{\text{err}}$ at $t = 5$ for the implementation of the Magnus method with `expm`, with constant stepsize, for an increasing number N of integration steps ($N = 5 \cdot 2^j$ for $j = 1, \ldots, 5$). As a reference solution we computed the approximation given by the MATLAB function `ode45` with absolute tolerance 10^{-14}. We then used $\boldsymbol{g}_{\text{err}}$ as a tolerance for the global error given by the implementations with splitting and Krylov techniques; whenever the global error for the given stepsize $h = 2^{-j}$ was greater in norm than $\boldsymbol{g}_{\text{err}}(j)$, h has been replaced with a smaller stepsize.

In Figure 1 we display the results of this experiment. As can be seen, low-rank splitting techniques and Krylov subspace methods give comparable results, both substantially better than `expm`. In Table 1 we report the ratios of the number of flops obtained with `expm` to the corresponding numbers for the other two implementations.

In the next table the same values are reported for the Magnus method of order 2. The splitting method is 4th-order. In the first column the global error is reported for the implementation with `expm` and for $h = 2^{-j}$ with $j = 2, \ldots, 5$.

Low-rank splitting methods seem to perform best when the accuracy required is

global error	`expm`/Krylov	`expm`/splitting
$4.5834 \cdot 10^{-3}$	1.18	7.20
$2.8749 \cdot 10^{-4}$	2.29	4.91
$1.7984 \cdot 10^{-5}$	3.65	4.45
$1.1243 \cdot 10^{-6}$	5.55	4.10
$7.0271 \cdot 10^{-8}$	7.69	3.98

Table 1: Different implementations of exponential approximants: 4th-order Magnus, $n = 30$.

global error	`expm`/Krylov	`expm`/splitting
$2.7609 \cdot 10^{-1}$	2.25	13.22
$6.9066 \cdot 10^{-2}$	3.67	17.97
$1.7269 \cdot 10^{-2}$	4.05	16.56
$4.3175 \cdot 10^{-3}$	7.87	16.08

Table 2: Different implementations of exponential approximants: 2nd-order Magnus, $n = 30$.

moderate and it seems a good strategy to combine a Lie-group integrator of order p with a low-rank splitting method of higher order. These results are only indicative of the performance of different implementations and further experimentation would be beneficial. Yet, we believe that they reflect a general trend.

Note that Krylov subspace methods have proved themselves very effective in a wide range of applications (Hochbruck, Lubich & Selhofer 1998). However, they do not guarantee that the mapping takes an element of $\mathfrak{g}$ to $\mathcal{G}$ unless implemented with very high – practically, machine – accuracy. This problem does not arise for low-rank splittings. It is worthwhile to note that both methods perform well when the dimension n of the problem is large. For example, the same experiment with $n = 100$ on the interval $[0, 1]$ results in the operations count recorded in Table 3.

global error	`expm`/Krylov	`expm`/splitting
$9.8109 \cdot 10^{-5}$	48	25
$6.1357 \cdot 10^{-6}$	86	42
$3.8355 \cdot 10^{-7}$	111	40
$2.3973 \cdot 10^{-8}$	132	31

Table 3: Different implementations of exponential approximants: 2nd-order Magnus, $n = 100$.

In our next numerical experiment we consider the *isospectral flow*

$$L' = [B(L), L], \quad t \geq 0, \qquad L(0) = L_0 \in \mathrm{Sym}(n; \mathbb{R}), \tag{5.9}$$

where $L \in \mathrm{Sym}(n; \mathbb{R})$ and $B(L) : \mathrm{Sym}(n; \mathbb{R}) \to \mathfrak{so}(n; \mathbb{R})$. The most important structural feature of (5.9) and the reason for the name 'isospectral flow' is that the eigenvalues of $L(t)$ remain constant for all $t \geq 0$. An intriguing source for equations (5.9) is Hamiltonian mechanics of Toda lattices. Thus, consider the Hamiltonian function

$$H(\boldsymbol{q},\boldsymbol{p}) = \tfrac{1}{2}\sum_{j=1}^{n} p_j^2 + \sum_{j=1}^{n} \mathrm{e}^{q_j - q_{j+1}}.$$

Changing variables,

$$\begin{aligned} a_j &= \mathrm{e}^{q_j - q_{j+1}}, \qquad j = 1,\dots,n-1, \\ b_j &= \tfrac{1}{2}p_j, \qquad j = 1,2,\dots,n, \end{aligned}$$

we can reformulate the underlying Hamiltonian equations in the *Lax formulation* (5.9), where

$$L = \begin{bmatrix} b_1 & a_1 & 0 & \dots & 0 \\ a_1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & a_{n-1} \\ 0 & \dots & 0 & a_{n-1} & b_n \end{bmatrix} \qquad P = \begin{bmatrix} 0 & a_1 & 0 & \dots & 0 \\ -a_1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & a_{n-1} \\ 0 & \dots & 0 & -a_{n-1} & 0 \end{bmatrix}.$$

Other sources of isospectral flows are numerical linear algebra and control theory (Calvo et al. 1997).

The manifold of all matrices similar to $L_0 \in \mathrm{Sym}(n;\mathbb{R})$ is a homogeneous space, subject to the action of the orthogonal group $\mathrm{O}(n;\mathbb{R})$, namely $\lambda(X,L) = XLX^{\mathrm{T}}$, $X \in \mathrm{O}(n;\mathbb{R})$, $L \in \mathrm{Sym}(n;\mathbb{R})$. We solve (5.9) with $n = 30$ using a fourth-order RK-MK method and compare `expm` with an approximant based on order-4 canonical coordinates of the second kind, as introduced in Subsection 5.5 (but without exploiting sparsity). Our initial conditions are $a_i \equiv \frac{1}{2}\mathrm{e}^{\sin \pi/(2n)}$, $b_i \equiv \sqrt{2}\sin\frac{\pi}{2n}$, which model a system of n equidistant particles on a semi-circle. Table 4 reports the rate of improvement of the CCSK approach, in comparison with the direct use of `expm`, for a range of global errors. The rate of improvement is self-evident and it is compelling to conjecture that it approaches a fixed ratio as the overall precision tends to zero.

global error	`expm`/SKC
$3.3029 \cdot 10^{-5}$	5.02
$2.2366 \cdot 10^{-6}$	7.60
$1.4361 \cdot 10^{-7}$	7.08
$9.0754 \cdot 10^{-9}$	7.09
$5.7005 \cdot 10^{-10}$	7.09

Table 4: A ratio between the number of flops in the solution of the isospectral flow (5.9) by a 4th-order RK-MK method.

6 Conclusions

In Section 4 we have derived complexity results for Fer and Magnus methods of orders up to eight. These results are summed up in Table 5, where we have also included initial and tentative results for the extrapolation methods of Orel (2000). We recall that Magnus methods for order ≥ 4 require lower cost than equivalent Fer methods. This, needless to say, does not mean that the latter are of little virtue in practical computation, since they might well possess other desirable properties. Moreover, we recall that our results are upper bounds. It is entirely possible, exploiting specific values of coefficients and aggregating terms in an astute manner, to improve upon the estimates of Table 5. This has been recently accomplished by Blanes, Casas & Ros (1999) for Magnus expansions of orders six and eight.

Method	order	$\boldsymbol{F}$	$\boldsymbol{E}$	$\boldsymbol{C}$
Fer	2	1	1	0
	4	2	2	2
	6	3	2	8
	8	4	3	45
Magnus	4	2	1	1
	6	3	1	6
	8	4	1	45
Extrapolation	6	3	3	3
	8	9	3	21

Table 5: The complexity of different Lie-group methods for linear equations.

Another shortcoming in our analysis is that we have addressed ourselves solely to the linear case. Both Fer and Magnus methods can be generalized to a nonlinear setting (Zanna 1999) but detailed complexity analysis of this construct has not yet taken place.

In Figure 2 we have compared the efficiency of a number of numerical methods, as applied to the *Airy equation*

$$Y' = \begin{bmatrix} 0 & 1 \\ -t & 0 \end{bmatrix} Y, \quad 0 \leq t \leq 10, \qquad Y(0) = I, \tag{6.1}$$

whose exact solution is a linear combination of Airy functions $\mathrm{Ai}(t)$ and $\mathrm{Bi}(t)$.

The methods are represented in the following manner:

Fer — solid line,

Magnus — dashed line,

Runge-Kutta — dot-dashed line;

employing the following mark types

$\circ$ — 2nd order methods,

$\times$ — 4th order methods,

$+$ — 6th order methods,

$\square$ — 8th order method.

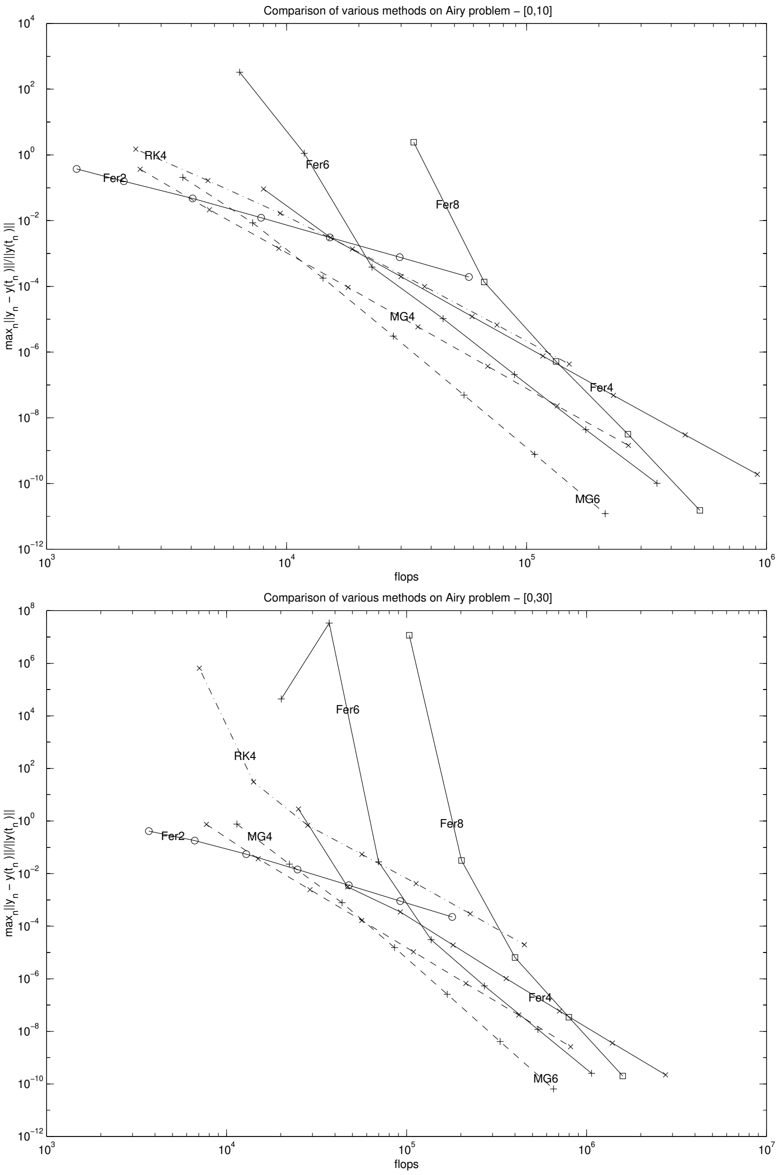


Figure 2: The efficiency of various methods applied to the Airy problem.

The solution of (6.1) is an oscillating trajectory with increasing frequency: $Y(t) \approx t^{-1/4}\cos(\frac{2}{3}t^{3/2})\tilde{Y},\ t \gg 1$, where $\tilde{Y}$ is constant. Therefore the equation represents a formidable challenge in long-term integration. The poor performance of standard methods and the excellent performance of Lie-group solvers has been already noted in (Iserles & Nørsett 1999). In the present example we have compared the behaviour of different Lie-group methods and the Runge–Kutta method. First t was restricted to a relatively small interval $[0, 10]$, hence the dice were loaded in favour of classical methods. Yet, our comparison demonstrates that even in this situation some Lie-group solvers (all Magnus methods and lower-order Fer methods) possess a definite edge over classical methods. Increasing the interval length, the superiority of Magnus methods become even more pronounced (compare with Figure 2).

Insofar as the approximation of the exponential function is concerned, our results are of a more tentative nature. Yet, they are reassuring in one important aspect: methods that respect the Lie-group structure need not be more expensive than classical approximants! As splitting methods are increasingly designed to take advantage of sparsity, their efficiency will be even further enhanced.

The single most important lesson from this initial foray into complexity analysis of Lie-group methods is that the formidable advantages accruing by respecting the geometry of a differential equation need not lead to a prohibitive computational cost. To the contrary, better understanding and modelling of the underlying mathematical structure help computation and render it more efficient.

Acknowledgements

This paper was partially written during the stay of EC, AI and SPN at Mathematical Sciences Research Institute, Berkeley. We wish to acknowledge the many efforts of the institute staff to make our stay enjoyable and the numerous discussions with fellow participants in the “Foundations of computational mathematics” and “Symbolic algebra and analysis” programmes.

References

Blanes, S., Casas, F. & Ros, J. (1999), Improved high order integrators based on Magnus expansion, Technical Report NA1999/08, DAMTP, University of Cambridge.

Blanes, S., Casas, F., Oteo, J. A. & Ros, J. (1998), ‘Magnus and Fer expansions for matrix differential equations: the convergence problem’, *J. Phys. A: Math. Gen.* **31**, 259–268.

Budd, C. J. & Iserles, A. (1999), ‘Geometric integration: Numerical solution of differential equations on manifolds’, *Phil. Trans Royal Society A* **357**, 945–956.

Calvo, M. P., Iserles, A. & Zanna, A. (1997), ‘Numerical solution of isospectral flows’, *Maths Comp.* **66**, 1461–1486.

Carter, R., Segal, G. & Macdonald, I. (1995), *Lectures on Lie Groups and Lie Algebras*, LMS Student Texts, Cambridge University Press, Cambridge.

Celledoni, E. & Iserles, A. (1998), Approximating the exponential from a Lie algebra to a Lie group, Technical Report 1998/NA3, DAMTP, University of Cambridge. To appear in *Maths Comp.*

Celledoni, E. & Iserles, A. (1999), Methods for the approximation of the matrix exponential in a Lie-algebraic setting, Technical Report 015, Mathematical Sciences Research Institute, Berkeley, California.

Crouch, P. & Grossman, R. (1993), 'Numerical integration of ordinary differential equations on manifolds', *J. Nonlinear Sc.* **3**, 1–33.

Cucker, F. & Smale, S. (1998), Complexity estimates depending on condition and round-off error, Technical report, Department of Mathematics City University of Hong Kong.

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

Engø, K., Marthinsen, A. & Munthe-Kaas, H. (1997), `DiffMan:` *An object oriented* `MATLAB` *toolbox for solving differential equations on manifolds*, University of Bergen, Bergen, Norway.

Faltinsen, S. (1998), Backward error analysis for Lie-group methods, Technical Report 1998/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

Feng, K. & Shang, Z.-J. (1995), 'Volume-preserving algorithms for source-free dynamical systems', *Numer. Math.* **71**, 451–463.

Fer, F. (1958), 'Résolution del l'equation matricielle $\dot{U} = pU$ par produit infini d'exponentielles matricielles', *Bull. Classe des Sci. Acad. Royal Belg.* **44**, 818–829.

Gallopoulos, E. & Saad, Y. (1992), 'Efficient solution of parabolic equations by Krylov approximation methods', *SIAM J. Sci. Comput.* **13**, 1236–1264.

Guillemin, V. & Pollack, A. (1974), *Differential Topology*, Prentice-Hall, Englewood Cliffs, NJ.

Hausdorff, F. (1906), 'Die symbolische Exponentialformel in der Gruppentheorie', *Berichte der Sächsischen Akademie der Wissenschaften (Math. Phys. Klasse)* **58**, 19–48.

Higham, D. S. (1997), 'Time-stepping and preserving orthogonality', *BIT* **37**, 24–36.

Hochbruck, M. & Lubich, C. (1997), 'On Krylov subspace approximations to the matrix exponential operator', *SIAM J. Num. Anal.* **34**, 1911–1925.

Hochbruck, M., Lubich, C. & Selhofer, H. (1998), 'Exponential integrators for large systems of differential equations', *SIAM J. Sci. Comput.* **19**, 1552–1574.

Howland, J. L. (1983), 'Further ways to approximate the exponential of a matrix', *Linear and Multilinear Algebra* **14**, 121–129.

Iserles, A. (1984), 'Solving linear ordinary differential equations by exponentials of iterated commutators', *Numerische Mathematik* **45**, 183–199.

Iserles, A. (1996), Beyond the classical theory of computational ordinary differential equations, *in* I. S. Duff & G. A. Watson, eds, 'State of the Art in Numerical Analysis', Oxford University Press, Oxford, pp. 171–192.

Iserles, A. & Nørsett, S. P. (1999), 'On the solution of linear differential equations in Lie groups', *Phil. Trans Royal Society A* **357**, 983–1020.

Iserles, A., Marthinsen, A. & Nørsett, S. P. (1999), 'On the implementation of the method of Magnus series for linear differential equations', *BIT* **39**, 281–304.

Iserles, A., Nørsett, S. P. & Rasmussen, A. F. (1998), Time-symmetry and high-order Magnus methods, Technical Report 1998/NA6, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

Kosloff, R., Tal-Ezer, H. & Cerjan, C. (1992), 'Low order polynomial approximation of propagators of the time-dependent Shroedinger equation', *J. of Comp. Phys.* **100**, 179–187.

Magnus, W. (1954), 'On the exponential solution of differential equations for a linear operator', *Comm. Pure Appl. Maths* **VII**, 649–673.

McLachlan, R. I. (1995), 'On the numerical integration of ordinary differential equations by symmetric composition methods', *SIAM J. Sci. Comput.* **16**, 151–168.

McLachlan, R. I., Quispel, G. R. W. & Robidoux, N. (1999), 'Geometric integration using discrete gradients', *Phil. Trans. Roy. Soc. A* **357**, 1021–1045.

Moan, P. C. (1998), Efficient approximation of Sturm–Liouville problems using Lie-group methods, Technical report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

Moler, C. & Van Loan, C. (1978), 'Nineteen dubious ways to compute the exponential of a matrix', *SIAM Review* **20**, 801–836.

Munthe-Kaas, H. (1999), 'High order Runge–Kutta methods on manifolds', *Appld Num. Maths* **29**, 115–127.

Munthe-Kaas, H. & Owren, B. (1999), 'Computations in a free Lie algebra', *Phil. Trans Royal Society A* **357**, 957–982.

Olver, P. J. (1995), *Equivalence, Invariants, and Symmetry*, Cambridge University Press.

Onischik, A. L., ed. (1993), *Lie Groups and Lie Algebras*, number 20 *in* 'Encyclopaedia of Mathematical Sciences', Springer-Verlag, Berlin.

Orel, B. (2000), Extrapolated Magnus methods, Dept Maths & Mech., Univ. Ljubljana (to appear).

Owren, B. & Marthinsen, A. (1999), Integration methods based on canonical coordinates of the second kind, Technical Report Numerics No. 5/1999, Norwegian University of Science and Technology, Trondheim, Norway.

Parlett, B. (1976), 'A recurrence among the elements of functions of triangular matrices', *Lin. Alg. & Applics* **14**, 117–121.

Sanz-Serna, J. M. & Calvo, M. P. (1994), *Numerical Hamiltonian Problems*, AMMC 7, Chapman & Hall, London.

Schatzman, M. (1997), Higher order alternate directions methods, Technical report, Université de Lyon, Laboratoire d'analyse numérique.

Sheng, Q. (1989), Solving partial differential equations by exponential splitting, PhD thesis, Cambridge University.

Stickel, E. U. (1990), 'An algorithm for fast high precision computation of matrix exponential and logarithm', *Analysis* **10**, 85–95.

Stickel, E. U. (1994), 'A splitting method for the calculation of the matrix exponential', *Analysis* **14**, 103–112.

Stuart, A. M. & Humphries, A. R. (1996), *Dynamical Systems and Numerical Analysis*, number 2 *in* 'Cambridge Monographs on Applied and Computational Mathematics', Cambridge University Press, Cambridge.

Varadarajan, V. S. (1984), *Lie Groups, Lie Algebras, and Their Representations*, number 102 *in* 'Graduate Texts in Mathematics', Springer-Verlag.

Wei, J. & Norman, E. (1963), 'On global representations of the solutions of linear differential equations as a product of exponentials', *Proc. Am. Math. Soc.* pp. 327–334.

Werschultz, A. G. (1991), *The Computational Complexity of Differential and Integral Equations. An Information-Based Approach*, Oxford University Press, Oxford.

Yoshida, H. (1990), 'Construction of higher order symplectic integrators', *Physics Letters A* **150**, 262–268.

Zanna, A. (1996), The method of iterated commutators for ordinary differential equations on Lie groups, Technical Report 1996/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England.

Zanna, A. (1999), 'Collocation and relaxed collocation for the Fer and the Magnus expansions', *SIAM J. Num. Anal.* **36**, 1145–1182.