Computations in a Free Lie Algebra*

Hans Munthe-Kaas†
Department of Informatics,
University of Bergen, Norway

Brynjulf Owren‡
Department of Mathematical Sciences,
NTNU, Norway.

Version March 11, 1998

Abstract

Many numerical algorithms involve computations in Lie algebras, like composition and splitting methods, methods involving the Baker-Campbell-Hausdorff formula and the recently developed Lie group methods for integration of differential equations on manifolds. This paper is concerned with complexity and optimization of such computations in the general case where the Lie algebra is *free*, i.e. no specific assumptions are made about its structure. It is shown how transformations applied to the original variables of a problem yield elements of a *graded* free Lie algebra whose homogeneous subspaces are of much smaller dimension than the original ungraded one. This can lead to substantial reduction of the number of commutator computations. Witts formula for counting commutators in a free Lie algebra is generalized to the case of a general grading. This provides accurate bounds on the complexity. The interplay between symbolic and numerical computations is also discussed, exemplified by the new Matlab toolbox 'DiffMan'.

1 Motivation and background

The aim of this paper is to discuss practical and theoretical issues related to complexity, optimization and development of numerical algorithms involving computations in a Lie algebra. We will be investigating the general case where no particular algebraic structure is assumed, except for what is common to all Lie algebras. This leads to the concept of a *free Lie algebra*. Most of the applications we discuss come from recently developed methods for the numerical solution of differential equations, in particular methods that are defined for problems in which the exact solution evolve on a manifold. A major cost of these algorithms involves the evaluation of Lie brackets, or commutators. We will discuss ways of reducing the number of commutator computations in various situations. Furthermore, we want to provide good upper bounds on the number of commutators. For this purpose a generalized version of the Witt formula is developed. Through the examples of the paper, we will see the usefulness of having software tools involving both numerical and symbolic computations. We will see

*This work was sponsored in part by The Norwegian Research Council under contract no. 111038/410, through the SYNODE project. WWW: `http://www.math.ntnu.no/num/synode/`

†Email: `hans@ii.uib.no`, WWW: `http://www.ii.uib.no/~hans/`

‡Email: `Brynjulf.Owren@math.ntnu.no`, WWW: `http://www.math.ntnu.no/~bryn/`

that symbolic computations can aid a numerical computation, and vice versa that numerical computations can aid symbolic computations. Many authors have been discussing various computational issues related to Lie algebras. A useful source to the state of the art in this subject is the papers collected in [9].

The paper is organized as follows. In the rest of this section we will give some mathematical background theory and briefly survey various numerical algorithms involving computations in Lie algebras. In Section 2 and 3 we will introduce free Lie algebras and develop new tools for counting commutators, through generalizations of the Witt formula. In Section 4, various applications are discussed.

1.1 Differential equations on manifolds

Consider initial value problems of the form

$$\dot{y} = F(t, y), \quad y(0) = y_0 \in M, \quad F : \mathbb{R} \times M \to TM. \tag{1}$$

where M is a manifold and where F is a time dependent vector field, thus for each t, $F(t, \cdot) \in \mathfrak{X}(M)$, the linear space of smooth vector fields on M. A *Lie algebra* structure can be imposed on $\mathfrak{X}(M)$ by using the *Lie-Jacobi bracket* $[\cdot,\cdot] : \mathfrak{X}(M) \times \mathfrak{X}(M) \to \mathfrak{X}(M)$. We define the bracket in terms of coordinates $x^1, \ldots, x^d$. If $X, Y, Z \in \mathfrak{X}(M)$ are vector fields with components X^i, Y^i, Z^i and $Z = [X, Y]$ then

$$Z^i = \sum_{j=1}^{d} \left(X^j \frac{\partial Y^i}{\partial x^j} - Y^j \frac{\partial X^i}{\partial x^j} \right).$$

In general, a Lie algebra $\mathfrak{g}$ is a linear space equipped with a bilinear bracket such that for $a, b, c \in \mathfrak{g}$ we have

$$\begin{aligned} [a, b] &= -[b, a] && (2)\\ 0 &= [a, [b, c]] + [b, [c, a]] + [c, [a, b]] && (3) \end{aligned}$$

The *flow* of a vector field $X \in \mathfrak{X}(M)$ is a mapping $\exp(X)$ from some open set $D \subseteq M$ into M defined as follows: Denote by $u(t)$ the solution of the differential equation

$$u' = X(u), \qquad u(0) = p,$$

then $\exp(X)(p) = u(1)$. Many numerical methods which are used to solve (1) are based on compositions of maps that are either flows of vector fields or can be well approximated by such flows. This makes it interesting to study compositions of flows

$$\exp(X_1) \circ \exp(X_2) \cdots \circ \exp(X_\mu). \tag{4}$$

In the construction and analysis of such methods, one may proceed by invoking the Baker-Campbell-Hausdorff (BCH) formula. If X and Y are two vector fields in $\mathfrak{X}(M)$ one has

$$\exp(X) \circ \exp(Y) = \exp(Z), \qquad Z \in \mathfrak{X}(M).$$

The formula for Z can be given in terms of iterated Lie-Jacobi brackets as follows (Varadarajan [22, pp. 114–121])

$$
\begin{aligned}
Z &= \sum_{n=1}^{\infty} c_n \\
c_1 &= X + Y \\
(n+1)c_{n+1} &= \frac{1}{2}[X - Y, c_n] + \sum_{p=1}^{[n/2]} \frac{B_{2p}}{(2p)!} \sum [c_{k_1}, [\cdots [c_{k_{2p}}, X + Y] \cdots]], \quad n \geq 1.
\end{aligned}
\tag{5}
$$

where B_j is the jth Bernoulli number. The second sum are over all positive integers $k_1, \ldots, k_{2p}$ such that $k_1 + \cdots + k_{2p} = n$.

By themselves, the formulas (5) are complicated, in general in the applications described above they need to be applied recursively since there are generally more than two elements in the composition. Additional difficulties arise from the fact that the iterated brackets which occur are not generally independent since $\mathfrak{X}(M)$ is a Lie algebra and therefore the brackets obey the identities (2) and (3). Typically, the vector fields X_i, whose flows appear in the composition (4), will depend on a stepsize h in such a way that $X_i = \mathcal{O}(h^{q_i})$ where $q_i \geq 1$ as $h \to 0$. It is clear that if a vector fields $A = \mathcal{O}(h^{q_A})$ and $B = \mathcal{O}(h^{q_B})$ then $[A, B] = \mathcal{O}(h^{q_A + q_B})$. When the composition occurs as a part of an integration method having order of consistency p, we may discard all terms in the BCH formula which are $\mathcal{O}(h^{p+1})$. Obviously, we need a systematic way of identifying these terms, and this problem will be addressed in the sequel.

1.2 Methods based on composition and splitting

When a numerical method is used to approximate the solution of (1) at $t = h$, it is common to represent it by a map $\psi_{h,F}$ called the h-flow of the method. We write $y_1 = \psi_{h,F}(y_0)$. By using backward error analysis [5, 19], one can in some important cases, at least formally, associate the h-flow of a method with the flow of a perturbed vector field $\tilde{F}_h$ such that $\psi_{h,F} = \exp(h\tilde{F}_h)$. A popular way [23, 20] of constructing high order numerical integration methods is to compose h-flows

$$\psi_{h,F} = \psi^1_{h_1,F} \circ \cdots \circ \psi^\mu_{h_\mu,F}$$

of different methods. In particular, this is useful when the flow of F is known to exhibit certain properties, for instance, that it belongs to a known subgroup of the diffeomorphisms of M. Thus the elements of the composition can be chosen accordingly, as members of the same subgroup. For instance, when F is a Hamiltonian vector field, its flow is known to be symplectic. It may therefore be desirable to impose conditions on the coefficients of the methods involved in the composition such that each $\psi_{h_i,F}$ is a symplectic mapping.

A related type of integration methods is the one based on splitting. In this case, it is assumed that the vector field is decomposed into a sum $F = F_1 + \cdots + F_s$, where the flow of each component is, in some sense, easier to compute than that of F. Then, the flow of F can be approximated by composing the (approximate) flows of the F_i's. For instance, for the splitting $F = F_1 + F_2$, the composition $\exp(hF_1/2) \circ \exp(hF_2) \circ \exp(hF_1/2)$ is symmetric and of second order [21].

1.3 Generalized Runge-Kutta methods

Iserles [7] suggested an approach for the solution of linear differential equations based on the Fer expansion [4], in which the approximation is obtained by multiplication of matrix exponentials. Likewise, the integration methods proposed by Crouch and Grossman [3] apply composition of flows of various vector fields. More precisely, it is assumed that there is a set of smooth vector fields, $E_1, \ldots, E_d$ on M, such that (1) can be written in the form

$$\dot{y} = \sum_i (f_i(y)E_i)(y), \quad f_i : M \to \mathbb{R}.$$

The *stages* are defined through the compositions

$$Y_i = \exp(ha_{is}F_s) \circ \cdots \circ \exp(ha_{i1}F_1)(y_0), \quad 1 \le i \le s \tag{6}$$

where the vector fields F_j are defined as *frozen at* Y_j *with respect to the frame vector fields*, i.e.

$$F_j : y \mapsto \sum_i (f_i(Y_j)E_i)(y).$$

The numerical solution is propagated by means of a similar composition

$$y_1 = \exp(hb_sF_s) \circ \cdots \circ \exp(hb_1F_1)(y_0) \tag{7}$$

Recently, Iserles and Nørsett [8] proposed a type of methods to solve the linear matrix equation

$$\dot{y} = a(t)y, \qquad y(t), a(t) \in \mathbb{R}^{n\times n}. \tag{8}$$

The methods make use of the Magnus expansion [11]. In some neighbourhood of a point $y_0 \in \mathbb{R}^{n\times n}$ the solution y of (8) can be represented uniquely by a function $\Omega : [0,T] \to \mathbb{R}^{n\times n}$ by means of the relation $y(t) = \exp(\Omega(t))y_0$. Magnus obtained the following expansion for the function $\Omega(t)$.

$$\begin{aligned} \Omega(t) &= \int_0^t a(\tau)\,d\tau + \frac{1}{2}\int_0^t \left[a(\tau), \int_0^\tau a(\sigma)\,d\sigma\right] d\tau \\ &+\frac{1}{4}\int_0^t \left[a(\tau), \int_0^\tau \left[a(\sigma), \int_0^\sigma a(\rho)\,d\rho\right] d\sigma\right] d\tau \\ &+\frac{1}{12}\int_0^t \left[\left[a(\tau), \int_0^\tau a(\sigma)\,d\sigma\right], \int_0^\tau a(\sigma)\,d\sigma\right] d\tau + \cdots \end{aligned} \tag{9}$$

Here, the bracket denotes the *matrix commutator*, $[A,B] = AB - BA$.

Iserles and Nørsett discretize the integrals in (9) by means of a Gauss quadrature formula to obtain approximations $\Omega_1, \Omega_2, \ldots$ and finally they compute $y_n = \exp(\Omega_n)y_{n-1}$. We note in passing that if $a(t)$ belongs, for all t, to some subspace $\mathfrak{g} \subset \mathbb{R}^{n\times n}$ which is closed under commutation, then there exists a certain submanifold G of $\mathbb{R}^{n\times n}$ on which the solution $y(t)$ evolves. This submanifold can be characterized locally as $\exp(\mathfrak{g}) \cdot y_0$. By construction, the methods proposed by Iserles and Nørsett will also produce approximations which belong to G.

Again, we shall be concerned with the complexity of the expression (9) which apparently becomes even more complicated after discretization with a quadrature rule. Iserles and Nørsett developed a theory involving a certain type of rooted trees for the purpose of analyzing the expressions in (9). It is remarkable that in specific examples, starting from hundreds of commutator terms, after taking into account: the order (in h) of the various terms, the skew-symmetry and Jacobi identity obeyed by the commutator, and the reversibility of the Gauss formula, only a few terms remain. For instance, they show that only 6 commutators are necessary to obtain methods of consistency order 6 [18]. This case will be studied using a different framework in Section 4.2.

Another generalization of Runge-Kutta methods that can be used essentially for the same type of problems as the Crouch-Grossman methods is the methods of Munthe-Kaas [14, 13], subsequently denoted RKMK methods. The methods are based on canonical coordinates of the first kind. To describe these methods in more detail, it is necessary first to introduce a little notation. Let M be a manifold and $\mathfrak{g}$ a Lie algebra, and assume that $\lambda : \mathfrak{g} \times M \to M$ is a left Lie algebra action on M. The methods of Munthe-Kaas [13] can be applied to problems of the form

$$y' = F(y) = \lambda_*(f(y))(y) \tag{10}$$

where $f : M \to \mathfrak{g}$ is assumed to be autonomous, simply for notational convenience. Here λ_* denotes the derivative map

$$\lambda_*(v)(p) = \left.\frac{d}{dt}\right|_{t=0} \lambda(tv, p).$$

Set $\lambda_p(u) = \lambda(p, u)$, $p \in M, u \in \mathfrak{g}$. The main idea behind this type of methods is that there is a vector field on $\mathfrak{g}$, $\tilde{f} \in \mathfrak{X}(\mathfrak{g})$ which is λ_p-related to the original vector field F. Moreover, there is a generic way of constructing $\tilde{f}$, namely

$$\tilde{f}(u) = d\exp_u^{-1}(f \circ \lambda_p(u)). \tag{11}$$

Hence, solving (10) with initial value $y(0) = p$ on M is locally equivalent to solving $u' = \tilde{f}(u)$ with initial value $u(0) = 0$ on $\mathfrak{g}$ in the sense that $y(t) = \lambda_p(u(t)), 0 \leq t \leq t^*$. The RKMK methods work by applying a classical RK method to the problem $u' = \tilde{f}(u)$ and then transforming back to M via the map λ_p, i.e. $y_{n+1} = \lambda_{y_n}(u_{n+1})$ in each step, taking $p = y_n$.

For each $u \in \mathfrak{g}$ the map $d\exp_u$ (as well as $d\exp_u^{-1}$) is a linear map from $\mathfrak{g}$ to $\mathfrak{g}$, one has the expansions

$$d\exp_u(v) = v + \frac{1}{2}[u, v] + \cdots + \frac{1}{(q+1)!}[\underbrace{u, [\ldots, [u,}_{q \text{ times}} v]]] + \cdots \tag{12}$$

$$d\exp_u^{-1}(v) = v + B_1[u, v] + \cdots + \frac{B_q}{q!}[\underbrace{u, [\ldots, [u,}_{q \text{ times}} v]]] + \cdots \tag{13}$$

where the B_q are the Bernoulli numbers. The evaluation of $d\exp_u^{-1}$ in a practical implementation of RKMK methods may seem a little awkward, but thanks to the choice of reference point ($p = y_n$) in each step, it is actually sufficient to retain only the first q terms of (13) in the case that a qth order RK method is applied to the vector field (11). We shall see in Section 4.2 and 4.3 that it is possible to reduce the computational cost even further and in fact substantially so. We conclude the discussion of RKMK methods by reciting the algorithm as given in [13].

$$\left.\begin{array}{rcl} u_i & = & \sum_j a_{ij}\tilde{k}_j \\ k_i & = & hf(\lambda(u_i, y_0)) \\ \tilde{k}_i & = & d\exp_{u_i}^{-1}(k_i) \end{array}\right\} \quad i = 1, \ldots, s$$
$$v = \sum_{i=1}^{s} b_i \tilde{k}_i \tag{14}$$
$$y_1 = \lambda(v, y_0)$$

It is understood in the above algorithm that an approximation is used for $d\exp_{u_i}^{-1}(k_i)$ such that rather than computing this quantity exactly, it suffices to employ an approximation such that the computed $\tilde{k}_i$ satisfies $\tilde{k}_i = d\exp_{u_i}^{-1}(k_i) + \mathcal{O}(h^{q+1})$ where q is the order of the integration method. Note also that the RKMK algorithm is explicit if $a_{ij} = 0$ whenever $i \leq j$.

2 Free Lie algebras

As a tool for simplifying expressions involving commutators, we introduce the concept of a *Free Lie Algebra* (FLA). Given an arbitrary index set I. In the applications of this paper I is either finite or countably infinite. The following definition is equivalent to the one in [22].

Definition 2.1 A Lie algebra $\mathfrak{g}$ is *free* over the set I if:

i) For every $i \in I$ there corresponds an element $X_i \in \mathfrak{g}$.

ii) For any Lie algebra $\mathfrak{h}$ and any function $i \mapsto Y_i \in \mathfrak{h}$, there exists a unique Lie algebra homomorphism $\pi : \mathfrak{g} \rightarrow \mathfrak{h}$ satisfying $\pi(X_i) = Y_i$ for all $i \in I$.

$\diamond$

Let $\mathcal{S} = \{X_i : i \in I\}$, $\mathcal{S} \subset \mathfrak{g}$. The algebra $\mathfrak{g}$ can be thought of as being the set of all (formal) commutators of X_i. To simplify the language, we will say that $\mathfrak{g}$ is the free Lie algebra *generated by* $\mathcal{S}$.

Definition 2.1 is a standard type free construction in category theory. It is well known that this type of constructions are valid for *any* set, and that the object $\mathfrak{g}$ is unique, up to isomorphisms [1]. One can also show that $\mathfrak{g}$ contains no proper subalgebra containing $\mathcal{S}$.

In category theory, $\mathfrak{g}$ is said to be a *universal object*, i.e. it contains a structure that is common to all Lie algebras, but nothing else. Furthermore, computations in $\mathfrak{g}$ can be applied in any concrete Lie algebra $\mathfrak{h}$ via the homomorphism π. More concretely, it is useful to think of $\mathfrak{g}$ as being a 'symbolic computation engine', which can exploit the algebraic manipulations defined in equations (2) and (3). A computation can be done in $\mathfrak{g}$, yielding a (formal) linear combination of brackets. The resulting expression can later be applied in a concrete Lie algebra by replacing each (abstract) X_i with a concrete Y_i, where Y_i could for instance be vector fields or matrices. A free Lie algebra is an invaluable computational tool in any computations on Lie algebras. In Section 4 we will see applications of an FLA module in the Matlab toolbox DiffMan.

Computationally, it is useful to represent a free Lie algebra $\mathfrak{g}$ via a basis, in the sense of linear algebra. There are various ways of constructing a basis for $\mathfrak{g}$. Here we shall consider a procedure based on Hall sets [2, p. 132]. Such a set, which we denote by H, can be given a total ordering defined as follows: First we assume that $\mathcal{S} \subset H$ and define recursively the length $l(\cdot)$ of members of H. Let $l(X) = 1$ if $X \in \mathcal{S}$. If $w \notin \mathcal{S}$ is a member of H then it is of the form $w = [u, v]$ with $u, v \in H$ and we set $l(w) = l(u) + l(v)$. We require for the ordering of H that $u < v$ if $l(u) < l(v)$. Elements of the same length are ordered internally as we please, typically by some lexicographical rule. Elements of length 2 are included in the Hall set if they are of the form $[X, Y]$, $X, Y \in \mathcal{S}$ and $X < Y$. Elements of length ≥ 3 are included if and only if they are of the form $[u, [v, w]]$, $u, v, w \in H$, $[v, w] \in H$, $v \leq u < [v, w]$.

Example 2.1 If $\mathcal{S} = \{X_1, X_2, X_3\}$, we find the following Hall basis consisting of elements with length ≤ 4:

$$X_1 \quad X_2 \quad X_3$$

$$[X_1, X_2] \quad [X_1, X_3] \quad [X_2, X_3]$$

$$\begin{array}{llll} [X_1,[X_1,X_2]] & [X_1,[X_1,X_3]] & [X_2,[X_1,X_2]] & [X_2,[X_1,X_3]] \\ [X_2,[X_2,X_3]] & [X_3,[X_1,X_2]] & [X_3,[X_1,X_3]] & [X_3,[X_2,X_3]] \end{array}$$

$$\begin{array}{lll} [X_1,[X_1,[X_1,X_2]]] & [X_1,[X_1,[X_1,X_3]]] & [X_2,[X_1,[X_1,X_2]]] \\ [X_2,[X_1,[X_1,X_3]]] & [X_2,[X_2,[X_1,X_2]]] & [X_2,[X_2,[X_1,X_3]]] \\ [X_2,[X_2,[X_2,X_3]]] & [X_3,[X_1,[X_1,X_2]]] & [X_3,[X_1,[X_1,X_3]]] \\ [X_3,[X_2,[X_1,X_2]]] & [X_3,[X_2,[X_1,X_3]]] & [X_3,[X_2,[X_2,X_3]]] \\ [X_3,[X_3,[X_1,X_2]]] & [X_3,[X_3,[X_1,X_3]]] & [X_3,[X_3,[X_2,X_3]]] \\ [[X_1,X_2],[X_1,X_3]] & [[X_1,X_2],[X_2,X_3]] & [[X_1,X_3],[X_2,X_3]] \end{array}$$

$\diamond$

It is already indicated by the example that the number of elements in the Hall basis with precisely n iterated brackets increases very fast. The precise result of the dimension ν_n of the corresponding subspaces of a finitely generated FLA is given by Witts formula [2]

$$\nu_n = \frac{1}{n} \sum_{d|n} \mu(d) s^{n/d} \tag{15}$$

where s is the number of generators, and where the sum is over all integers d that divide n. The function $\mu : \mathbb{N} \to \{-1, 0, 1\}$ is defined as follows. If d has a prime factorization:

$$d = p_1^{n_1} p_2^{n_2} \cdots p_q^{n_q}, \quad n_i > 0,$$

then

$$\mu(d) = \begin{cases} 1 & \text{for } d = 1 \\ (-1)^q & \text{if all } n_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

We illustrate the fast growth of the dimensions by giving ν_n for $n \leq 10$ with 3 generators as in Example 2.1.

n	1	2	3	4	5	6	7	8	9	10
ν_n	3	3	8	18	48	116	312	810	2184	5880

3 The dimension of graded free Lie algebras

The formula of Witt (15) counts the number of commutators of a given *length*. For the study of the complexity of numerical computations, it is important to derive similar counting formulas where the commutators are ordered according to some other size measures. For example, if A, B are two matrices depending on a small parameter h as $A = \mathcal{O}(h^{q_A})$ and $B = \mathcal{O}(h^{q_B})$ then $[A, B] = \mathcal{O}(h^{q_A+q_B})$. We might want to count all commutators up to a given order $\mathcal{O}(h^q)$. To model this situation we define a *grading function* w on a free Lie algebra as follows:

- On the generating set $\mathcal{S} = \{X_i : i \in I\}$, the function w is given as:

  $$w(X_i) = w_i, \text{ for all } X_i \in \mathcal{S},$$

  where w_i are arbitrarily chosen positive integer grades.

- On the Hall set, w is extended by additivity

  $$w([u, v]) = w(u) + w(v) \text{ for all } [u, v] \in H.$$

This splits H in a disjoint union $H = \bigcup_{n=1}^{\infty} H_n$, where $H_n = \{h \in H : w(h) = n\}$. Similarly, the free Lie algebra $\mathfrak{g}$ splits in a direct sum of subspaces

$$\mathfrak{g} = \bigoplus_{n=1}^{\infty} \mathfrak{g}_n, \text{ where } \mathfrak{g}_n = \mathrm{span}(H_n).$$

Hence $\mathfrak{g}$ becomes a so called *graded* algebra. Let ν_n denote the number of elements in H_n. Evidently $\nu_n = \dim(\mathfrak{g}_n)$. These numbers are called the *homogeneous dimensions* of the grading. We will prove several important results about ν_n.

Theorem 3.1 *Let $\mathfrak{g}$ be the graded free Lie algebra generated by s elements $X_1, \ldots, X_s$ with a grading w defined by assigning positive integer grades $w_i = w(X_i)$ for all i. Let*

$$p(T) = 1 - \sum_{i=1}^{s} T^{w_i} \tag{17}$$

and let $\{\lambda_i\}_{i=1}^{m}$ be the roots of p, where $m = \max_i w_i$. Then

$$\dim(\mathfrak{g}_n) = \nu_n = \frac{1}{n} \sum_{d|n} \left(\sum_{i=1}^{m} \lambda_i^{-n/d} \right) \mu(d) \tag{18}$$

where the first sum ranges over all integers which divide n, and μ is the Möbius function defined by (16).

Note that when $w_i = 1$, this result reduces to the classical Witt formula.

It might be useful to have an explicit expression for the sum of the inverse powers of the roots. We write the polynomial in (17) as

$$p(T) = 1 - \sum_{i=1}^{s} T^{w_i} = 1 - \sum_{j=1}^{m} r_j T^j$$

and form the inverse of the companion matrix of $p(T)$. This yields:

$$C = \begin{pmatrix} & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \\ r_d & \cdot & \cdot & r_2 & r_1 \end{pmatrix}$$

hence

$$\sum_{i=1}^{m} \lambda_i^{-j} = \mathrm{trace}(C^j).$$

Theorem 3.1 can also be extended to the following important infinite cases:

Theorem 3.2 *Let $\mathfrak{g}$ be the graded free Lie algebra generated by a countable set of elements $X_1, X_2, \ldots$ with a grades $w_i = w(X_i)$ for all i. Suppose that the formal sum $1 - \sum_{i=1}^{\infty} T^{w_i}$ adds up to a rational function $r(T) = p(T)/q(T)$. Let the roots of p and q be denoted $\lambda_1, \ldots, \lambda_m$ and $\gamma_1, \ldots, \gamma_{\tilde{m}}$ respectively. Then ν_n is given as*

$$\nu_n = \frac{1}{n} \sum_{d|n} \left(\sum_{i=1}^{m} \lambda_i^{-n/d} - \sum_{i=1}^{\tilde{m}} \gamma_i^{-n/d} \right) \mu(d). \tag{19}$$

In some cases two different graded FLAs may have homogeneous dimensions ν_n and ν'_n which are identical for all sufficiently large n:

Corollary 3.3 *Let $\mathfrak{g}$ and $\mathfrak{g}'$ be two graded free Lie algebras with corresponding rational functions $r(T)$ and $r'(T)$ as given in Theorem 3.2. Suppose that r and r' are related as follows*

$$r'(T) = \prod_{n=1}^{N} (1 - T^n)^{\alpha_n} \, r(t) \tag{20}$$

where the α_n are arbitrary integers (possibly negative). Then

$$\nu'_n = \nu_n + \alpha_n, \ n \le N, \quad \nu'_n = \nu_n, \ n > N. \tag{21}$$

This Corollary gives an interesting alternative proof of a result by McLachlan [12]: Let $\mathfrak{g}$ be the graded FLA generated by an infinite number of elements $X_1, X_2, \ldots$ such that $w(X_i) = w_i = i$.

Let $\mathfrak{g}'$ be the FLA generated by 2 elements Y_1, Y_2 with grades 1. Then the result by McLachlan says that the corresponding homogeneous dimensions satisfy $\nu_n = \nu'_n$ for $n > 1$. Note that the rational functions of the two cases are

$$r(T) = \frac{1-2T}{1-T} \quad \text{and} \quad r'(T) = 1 - 2T, \quad \text{thus} \quad r'(T) = (1-T)r(T)$$

and McLachlan's result follows from (21).

In the rest of this section we will prove these results. *This may be skipped without loss of continuity.*

For the proof we need the concepts of a free associative algebra and the universal enveloping algebra of a Lie algebra.

Definition 3.4 For any Lie algebra, $\mathfrak{g}$, there is a unique pair $(\mathcal{C}, \pi)$ called its *Universal Enveloping Algebra* (UEA). $\mathcal{C}$ is an associative algebra, and π a linear mapping from $\mathfrak{g}$ to $\mathcal{C}$ such that

i) $\pi[\mathfrak{g}]$ generates $\mathcal{C}$.

ii) $\pi([X,Y]) = \pi(X)\pi(Y) - \pi(Y)\pi(X)$ for all $X, Y \in \mathfrak{g}$.

iii) If $\mathcal{U}$ is *any* associative algebra and ξ a linear map from $\mathfrak{g}$ to $\mathcal{U}$, then there is a homomorphism ξ' from $\mathcal{C}$ to $\mathcal{U}$ such that $\xi(X) = \xi'(\pi(X))$ for all $X \in \mathfrak{g}$.

$\diamond$

We shall not be concerned with the construction of a UEA (see [22], [2]). Here, we shall only need a result concerning the UEA which is normally stated as a corollary to the celebrated Poincaré-Birkhoff-Witt theorem.

Theorem 3.5 *(PBW) Let $\{e_i\}_{i=1}^{\infty}$ be a basis for a Lie algebra $\mathfrak{g}$. Then the UEA of $\mathfrak{g}$ has a basis of the form*

$$\{\pi(e_{i_1}) \cdot \pi(e_{i_2}) \cdots \pi(e_{i_s}), i_1 \le i_2 \le \cdots \le i_s, s \ge 1\}$$

Remark 3.6 The linear map π is injective on $\mathfrak{g}$, hence one may think of $\mathfrak{g}$ as being "contained" in the UEA $(\mathcal{C}, \pi)$ and where the bracket is defined as the commutator in (the associative algebra) $\mathcal{C}$.

In an important construction to be made later, π is actually given as the identity map, and in this case the basis for the UEA is simply

$$\{e_{i_1} \cdot e_{i_2} \cdots e_{i_s}, i_1 \le i_2 \le \cdots \le i_s, s \ge 1\}$$

$\diamond$

Definition 3.7 The *Free Associative Algebra* (FAA) generated by a set $\mathcal{S} = \{X_i : i \in I\}$ is an associative algebra $\mathcal{B} \supset \mathcal{S}$ such that

i) $\mathcal{S}$ generates $\mathcal{B}$.

ii) If $\mathcal{U}$ is *any* associative algebra containing $\mathcal{S}$ then there is a unique homomorphism ξ from $\mathcal{B}$ to $\mathcal{U}$ such that $\xi(X_i) = X_i$ for all $i \in I$.

$\diamond$

Now, let $\mathcal{B}$ be the FAA generated by some set $\mathcal{S} = \{X_i : i \in I\}$. We *define* a bracket on $\mathcal{B}$ by $[u, v] = uv - vu$ for any $u, v \in \mathcal{B}$. This construction yields a Lie algebra $\mathcal{B}_L$ and we denote by $\mathfrak{g}$ the smallest subalgebra of $\mathcal{B}_L$ which contains the set $\mathcal{S}$. We find the following important theorem in Varadarajan [22, Thm 3.2.8, p. 174].

Theorem 3.8 *The Lie algebra* $\mathfrak{g}$ *constructed above is a free Lie algebra. Moreover,* $(\mathcal{B}, \mathrm{Id})$ *is the universal enveloping algebra of* $\mathfrak{g}$.

We shall need to make the FAA $\mathcal{B}$ into a *graded* algebra. That is, for each integer $n \geq 0$ there is a *homogeneous subspace* $\mathcal{B}_n$ of $\mathcal{B}$ with $\mathcal{B}_0 = \mathbb{R}$,

$$\mathcal{B} = \bigoplus_{n=0}^{\infty} \mathcal{B}_n, \quad \text{and} \quad \mathcal{B}_n \cdot \mathcal{B}_m \subseteq \mathcal{B}_{n+m} \quad \text{for all } m, n \geq 0.$$

The elements of $\cup_{n=0}^{\infty} \mathcal{B}_n$ are called *homogeneous* and those of $\mathcal{B}_n$ are called homogeneous of degree n.

In our case, we define a grading in the following way. Set $\mathcal{B}_0 = \mathbb{R}$. Assign to each generator X_i, $i \in I$, a positive integer grade $w_i = w(X_i)$. Then define for any $n \geq 1$

$$\mathcal{S}_n = \bigcup_{\ell \geq 1} \{X_{i_1} \cdot X_{i_2} \cdots X_{i_\ell} : \sum_{j=1}^{\ell} w(X_{i_j}) = n, \; i_j \in I \text{ for all } j\} \tag{22}$$

and set $\mathcal{B}_n = \mathrm{span}(\mathcal{S}_n)$. We have by construction that $\mathfrak{g} \subset \mathcal{B}$, hence it makes sense to define the subspaces $\mathfrak{g}_n = \mathfrak{g} \cap \mathcal{B}_n$ of $\mathfrak{g}$ for all $n \geq 0$. In fact, it is possible to decompose $\mathfrak{g}$ into a direct sum of the $\mathfrak{g}_n$'s. The focus of our interest is the dimensions $\nu_n = \dim(\mathfrak{g}_n)$, $n \geq 1$, of the *homogeneous subspaces* of the free Lie algebra.

Proof of Theorem 3.1: The main idea of the proof is to count the dimension of $\mathcal{B}_n$, the subspace of homogeneous elements of degree n, in two different ways. First, by considering $\mathcal{B}$ as the UEA of $\mathfrak{g}$ and then by considering $\mathcal{B}$ as the FAA generated by $X_1, \ldots, X_s$ (cf. Theorem 3.8). Let $b_n = \dim(\mathcal{B}_n)$ and let

$$g(T) = \sum_{n=0}^{\infty} b_n T^n$$

be the generating function.

i) $\mathcal{B}$ as a UEA. Since $\mathfrak{g}$ is a direct sum of the subspaces $\mathfrak{g}_n$, we can find a basis for $\mathfrak{g}$ consisting of homogeneous elements. We order the basis according to the degree of the elements, i.e. let $e_{1,1}, \ldots, e_{1,\nu_1}$ be the basis elements of degree 1, $e_{2,1}, \ldots, e_{2,\nu_2}$ those of degree 2 etc. In view of the PBW theorem, a basis for the UEA is given as the terms of the expression

$$\sum_{r=0}^{\infty}(e_{1,1})^r \cdot \sum_{r=0}^{\infty}(e_{1,2})^r \cdots \sum_{r=0}^{\infty}(e_{1,\nu_1})^r \cdots\cdots \sum_{r=0}^{\infty}(e_{n,1})^r \cdots \sum_{r=0}^{\infty}(e_{n,\nu_n})^r \cdots$$

Thus, to find the generating function $g(T)$ it suffices to replace the basis elements e_{ij} of $\mathfrak{g}$ with T^i in the above expression. We obtain

$$g(T) = \prod_{n=1}^{\infty}\left(\sum_{r=0}^{\infty} T^{nr}\right)^{\nu_n} = \prod_{n=1}^{\infty}(1-T^n)^{-\nu_n}.$$

ii) $\mathcal{B}$ as an FAA. First define as r_ℓ the number of w_i that are equal to ℓ, and let $m = \max_i w_i$ as before. $\mathcal{B}$ is generated (as an FAA) by the set $\mathcal{S} = \{X_1, \ldots, X_s\}$. To find the dimension of $\mathcal{B}_n$ we use the basis defined by (22). Clearly, any element of $\mathcal{S}_n$ with $n > 0$ can be factored uniquely into $u \cdot v$ where $u \in \mathcal{S}_k$ for some $k < n$ and $v \in \mathcal{S}$ with $w(v) = n - k$. We therefore obtain the following recursion formula for $b_n = \dim(\mathcal{B}_n)$

$$b_n = r_1 b_{n-1} + r_2 b_{n-2} + \cdots + r_m b_{n-m}, \quad n > 0, \quad b_0 = 1,\ b_i = 0 \text{ for } i < 0 \tag{23}$$

Define the polynomial

$$p(T) = 1 - \sum_{\ell=1}^{m} r_\ell T^\ell = 1 - \sum_{i=1}^{s} T^{w_i}.$$

We compute

$$g(T)\cdot p(T) = \left(\sum_{n=0}^{\infty} b_n T^n\right)\left(1 - \sum_{\ell=1}^{m} r_\ell T^\ell\right) = \sum_{n=0}^{\infty}\left(b_n - \sum_{\ell=1}^{m} r_\ell b_{n-\ell}\right)T^n$$

again assuming $b_i = 0$ for $i < 0$. From (23) we therefore obtain $g(T)\cdot p(T) = b_0 = 1$ hence we conclude that

$$g(T) = \frac{1}{p(T)} = \frac{1}{1 - \sum_{i=1}^{s} T^{w_i}}.$$

Now it remains only to equate the expressions for $g(T)$ from i) and ii) and solve for the numbers ν_n, i.e. we must solve the equation

$$g(T) = \prod_{n=1}^{\infty}(1-T^n)^{-\nu_n} = \frac{1}{p(T)} = \frac{1}{1 - \sum_{i=1}^{s} T^{w_i}}. \tag{24}$$

Taking logarithms of both expressions and using the expansion $\log(1-x) = \sum_{j=1}^{\infty} x^j/j$ we obtain

$$\sum_{n=1}^{\infty} \nu_n \sum_{i=1}^{\infty} \frac{1}{i} T^{in} = \sum_{j=1}^{\infty} \frac{1}{j} a_j T^j$$

where the coefficients a_j will be computed later. We equate the terms of power k to obtain

$$\sum_{n|k} \nu_n \frac{n}{k} = \frac{1}{k} a_k, \quad \text{for all } k > 0.$$

It follows from the Möbius inversion formula [2, p. 176] that

$$\nu_n = \frac{1}{n} \sum_{d|n} \mu(d) a_{n/d}.$$

Finally, we compute a_j. Write $p(T) = \prod_{i=1}^{m}(1 - T/\lambda_i)$ where the λ_i are the roots of p. Then

$$\log(p(T)) = \sum_{i=1}^{m} \log\left(1 - \frac{T}{\lambda_i}\right) = \sum_{j=1}^{\infty} \frac{1}{j} \left(\sum_{i=1}^{m} \lambda_i^{-j}\right) T^j$$

thus $a_j = \sum_{i=1}^{m} \lambda_i^{-j}$. This concludes the proof of the theorem. □

Proof of Theorem 3.2: The modifications to the above proof needed to get this result are minor. In (23) the recursion becomes infinite. Let $r(T) = p(T)/q(T)$ be as in Theorem 3.2. Using the recursion it is simple to show that $g(T)p(T)/q(T) = 1$. Hence equation (24) becomes

$$r(T) = p(T)/q(T) = \prod_{n=1}^{\infty} (1 - T^n)^{\nu_n}. \tag{25}$$

The result now follows by taking logarithms and using the Möbius inversion formula. □

Proof of Corollary 3.3: This result follows immediately by applying the relation (20) in equation (25). □

4 Applications

4.1 Computation of the BCH formula

We will let the BCH formula (5) serve as our first example of a computation in a free Lie algebra. DiffMan is a publicly available Matlab toolbox found at:

`http://www.math.ntnu.no/num/diffman/`

In this toolbox most of the integrators described in Section 1 are found. In DiffMan, we find a free Lie algebra module, `lafree`. The commands given below reflect the basic operations of Definition 2.1.

`>> fla = lafree({[p,q],[w1,w2,...,wp]});`
Generate a free Lie algebra from `p` symbols with grades `w1,w2,...,wp`. All terms of total grade greater than `q` are set to 0.

`>> Xi = basis(fla,i);`
Return the `i`'th Hall basis element in `fla`. If $1 \leq$ `i` $\leq$ `p`, return the `i`'th generator X_i.

```
function [z] = bch(q)
   % coefficients of s-stage q'th order Gauss-Legendre RK
   [A, b, s] = glrk(q);
   fla = lafree({[2, q], [1, 1]});
   x = basis(fla, 1);  y = basis(fla, 2);
   u = zeros(fla, s);  k = u;
   % fixed point iteration, q times
   for j = 1 : q,  for i = 1 : s,
      u(i) = y + A(i, 1 : s) * k;
      k(i) = dexpinv(u(i), x, j); % approx up to order j
   end; end;
   z = y + b * k;
return;

>> format rat; z = bch(6)
   z = [1] + [2] + 1/2*[1,2] + 1/12*[1,[1,2]] - 1/12*[2,[1,2]]
      - 1/24*[2,[1,[1,2]]] - 1/720*[1,[1,[1,[1,2]]]]
      - 1/180*[2,[1,[1,[1,2]]]] + 1/180*[2,[2,[1,[1,2]]]]
      + 1/720*[2,[2,[2,[1,2]]]] - 1/120*[[1,2],[1,[1,2]]]
      - 1/360*[[1,2],[2,[1,2]]] + 1/1440*[2,[1,[1,[1,[1,2]]]]]
      + 1/360*[2,[2,[1,[1,[1,2]]]]] + 1/1440*[2,[2,[2,[1,[1,2]]]]]
      + 1/240*[[1,2],[2,[1,[1,2]]]] + 1/720*[[1,2],[2,[2,[1,2]]]]
      - 1/240*[[1,[1,2]],[2,[1,2]]]
```

Figure 1: Computation of BCH to order q.

`>> X+Y; r*X; [X,Y];`
Basic computations in the free Lie algebra.

`>> Z = eval(E,{Y1,Y2,...,Yp});`
If `E` is an element of a free Lie algebra, and `Y1,Y2,...,Yp` are the elements of *any* DiffMan Lie algebra, this will evaluate the expression `E`, using the data set `Y1,Y2,...,Yp` in place of the generating set. This corresponds to the homomorphism $\pi : \mathfrak{g} \to \mathfrak{h}$ in Definition 2.1.

The computation of the BCH formula can in principle be done directly from equation (5). We will instead show a much simpler approach, which illustrate the usefulness of using a numerical computation to produce a symbolic result. By differentiating the expression $\exp(z(t)) = \exp(tx)\exp(y)$ with respect to t, we find

$$z' = d\exp_z^{-1}(x),\ z(0) = y.$$

This equation can be integrated numerically in the free Lie algebra from $t = 0$ to $t = 1$ using for instance a single step of a Runge-Kutta method of sufficiently high order. The implicit Gauss-Legendre methods are useful since they are easy to generate of arbitrary high order, alternatively one can use an extrapolation method. The implicit RK equations can be solved by fixed point iteration in the free Lie algebra, until convergence. The resulting DiffMan code is listed in Figure 1.

The resulting z may now be applied to data from concrete Lie algebras using the `eval` operator, or if we want to symbolically combine more than two flows, we may apply `eval` on the free Lie algebra itself.

4.2 Implicit RK methods for equations of Lie type

We will in this section consider equations of Lie type (sometimes also called linear equations), given by (8) in matrix Lie group case, or in the general form of (10) as

$$y' = \lambda_*(f(t))(y). \tag{26}$$

This type of equation is the manifold version of a quadrature problem. Sophus Lie showed that it is in theory solvable by quadratures if and only if the Lie algebra is solvable. Numerically, it is harder to solve than the quadrature problem on $\mathbb{R}^n$, since the right hand side depends on y. On the other hand, since f only depends on t it is easier than the general equation (10).

We will consider the solution of (26) by implicit Runge-Kutta methods of the form (14). The fact that f only depends on t makes it possible to solve the implicit RK equations a priori by fixed point iteration in a free Lie algebra, as in Section 4.1. We could let the function values k_i be the generators of the free Lie algebra, but since all k_i are of size $\mathcal{O}(h)$, these free elements have all grade 1, and this approach leads to an explosive growth of commutators. The 'ungraded' line in (27) shows the dimension of the free Lie algebra generated by s elements of grade 1, counting all terms with total grade less than or equal $2s$. We will show that by a change of variables we can introduce a basis with grading $1, 2, 3, \ldots, s$. This reduces the dimensions to those given in the row labeled 'graded'. Finally, it will be shown that if the basis is chosen carefully, only terms with odd total grade will contribute. This reduces the dimensions to the numbers in the last row.

s (stages)	1	2	3	4	5
$2s$ (order of method)	2	4	6	8	10
$\dim(\mathfrak{g})$ (ungraded)	1	8	196	11464	1256567
$\dim(\mathfrak{g})$ (graded)	1	4	15	55	164
$\dim(\mathfrak{g})$ (graded, odd terms)	1	2	7	22	73

(27)

All these numbers are computed from Theorem 3.1.

We will use two crucial observations, introduced within a different framework by Iserles, Nørsett and Rasmussen [18]. Consider a single step of (14) from $t = 0$ to $t = h$, $y_0 \mapsto y_1$. The first observation is that the elements k_i are *not independent*, since they are samples of the same continuous function, $k_i = hf(c_i h)$, where $c_i = \sum_{j=1}^{s} a_{ij}$. We will use this to form a Taylor series type basis for $\mathfrak{g}$. Define the Vandermonde matrix

$$V(c) = (v_{ij})_{ij=1}^{s}, \text{ where } v_{ij} = c_i^{j-1},$$

and we introduce a new basis $Q_1, Q_2, \ldots, Q_s$ as

$$(k_1, k_2, \ldots, k_s)^T = V(c) \cdot (Q_1, Q_2, \ldots, Q_s)^T$$

From standard theory of divided differences, we know that

$$Q_i = \frac{h^i}{(i-1)!} f^{(i-1)}(\xi_i), \text{ for some } \xi_i \in (0, h).$$

Thus, under this change of variables, we get a graded free Lie algebra with grades $1, 2, \ldots, s$.

The second observation is that the problem has a time reversal symmetry. If we change the function f in (26) from $f(t)$ to $-f(h-t)$, we get a flow which between $t = 0$ and $t = h$ will take us back from y_1 to y_0. The underlying scheme, Gauss-Legendre-RK, preserve this symmetry also numerically. To take advantage of this symmetry, we must use a Taylor basis centered around $t = h/2$ instead of around $t = 0$. Thus we let

$$(k_1, k_2, \ldots, k_s)^T = V(c - \frac{1}{2}) \cdot (Q_1, Q_2, \ldots, Q_s)^T.$$

Consider an integration step given for the matrix equation as $y_1 = \exp(v) y_0$ and generally as $y_1 = \lambda(v, y_0)$, where $v = v(Q_1, Q_2, \ldots, Q_s)$. Under the symmetry $f(t) \mapsto -f(h-t)$, the Q_i change as $Q_i \mapsto (-1)^i Q_i$. Since $\exp(v)^{-1} = \exp(-v)$ we arrive at the conclusion that v must have the following symmetry:

$$v\left((-1)^1 Q_1, (-1)^2 Q_2, \ldots, (-1)^s Q_s\right) = -v\left(Q_1, Q_2, \ldots, Q_s\right).$$

Hence, *v depends on the free basis $Q_1, \ldots, Q_s$ only through terms in the Hall set of odd grade.* This reduces the dimension of the free Lie algebra to the numbers given in the final row of (27).

Let a_{ij}, b_j, c_j be the coefficients of the s stage, order $2s$ Gauss-Legendre RK method. For equations of Lie type, the algorithm given in (14) becomes:

$$\begin{aligned} V &= (v_{ij})_{ij=1}^s \text{ where } v_{ij} = (c_i - \tfrac{1}{2})^{j-1} \\ k_i &= hf(hc_i), \quad i = 1, 2, \ldots, s \\ Q_i &= \sum_{j=1}^{s} \left(V^{-1}\right)_{ij} k_j, \quad i = 1, 2, \ldots, s \\ v &= v\left(Q_1, Q_2, \ldots, Q_s\right) \\ y_1 &= \lambda(v, y_0) \end{aligned} \tag{28}$$

The exact form of v and the coefficients of V^{-1} and c can be found for methods of arbitrary order by running the command `glrk_lie_coeff` in DiffMan. For order 2, 4 and 6, the expressions for v become

$$\begin{aligned} v(Q_1) &= Q_1 \\ v(Q_1, Q_2) &= Q_1 - 0.083333[Q_1, Q_2] \\ v(Q_1, Q_2, Q_3) &= Q_1 + 0.083333 Q_3 - 0.083333[Q_1, Q_2] \\ &\quad +0.0041667[Q_2, Q_3] + 0.0027778[Q_1, [Q_1, Q_3]] \\ &\quad -0.0041667[Q_2, [Q_1, Q_2]] + 0.0013889[Q_1, [Q_1, [Q_1, Q_2]]] \end{aligned}$$

For order 8 we get 22 terms. Note that the theory summarized in (27) counts exactly how many terms we get in the expressions for v. The actual number of commutators needed to compute v is, however, more difficult to predict in general. For order 6, we can find v by computing 5 commutators as follows:

$$\begin{aligned}
T_1 &= [Q_1, Q_2],\ T_2 = [Q_2, Q_3 - T_1],\ T_3 = [Q_1, Q_3] \\
T_4 &= [Q_1, T_1],\ T_5 = [Q_1, 2T_3 + T_4] \\
v &= Q_1 + 0.083333(Q_3 - T_1) + 0.0041667T_2 + 0.0013889T_5
\end{aligned}$$

At the moment we have no systematic ways of reducing an expression to the smallest possible number of commutators.

4.3 Explicit RKMK methods for general equations on manifolds

In the general equation (10), where f depends on y, it is not possible to solve implicit RK equations a priori in a free Lie algebra. We will therefore consider explicit RK methods for this problem. In order to optimize the computation of $d\exp_{u_i}^{-1}$ as they occur in (14), we shall find it useful to consider linear combinations of the k_i which are of a high order in the step size h. Such linear combinations are of course method dependent and to find them, we shall begin by considering the *corrected* stages $\tilde{k}_i$ rather than the k_i. Recall that the $\tilde{k}_i$ are the stages of a classical RK method used in the standard way.

Let (A, b) denote an explicit Runge-Kutta (ERK) method. Here A is an $s \times s$ matrix whose elements $a_{ij} = 0,\ j \geq i$, and b is an s-vector. For the differential equation $y' = \tilde{f}(y)$ we apply (A, b) as follows in order to proceed the solution from $y_0 = y(0)$ to $y_1 \approx y(1)$

$$\left.\begin{aligned}
u_i &= y_0 + \sum_{j=1}^{i-1} a_{ij}\tilde{k}_j \\
\tilde{k}_i &= h\tilde{f}(u_i)
\end{aligned}\right\}, \quad i = 1, \ldots, s,$$
$$y_1 = y_0 + \sum_{i=1}^{s} b_i \tilde{k}_i.$$

We now look for linear combinations of $\tilde{k}_1, \ldots, \tilde{k}_m$ of the highest possible order in the stepsize h, i.e. for any fixed m such that $1 \leq m \leq s$, we seek $r = (r_1, \ldots, r_m)$, such that

$$\sum_{i=1}^{m} r_i \tilde{k}_i = \mathcal{O}(h^{q+1}) \tag{29}$$

for q as large as possible. For this purpose we need to use some theory of order conditions for RK methods.

It is well known (see e.g. [6]) that y_1 as well as each of the *stages* $\tilde{k}_i$ has a B-series. This is an expansion which we write in the form

$$B(\mathbf{a}, y) = \sum_{t \in T} \frac{h^{\rho(t)}}{\rho(t)!}\mathbf{a}(t)F(t)(y). \tag{30}$$

Here T is the set of rooted trees, $\rho(t)$ is the number of nodes in the rooted tree t, $\mathbf{a}$ assigns to each $t \in T$ a real number, and $F(t)$ depends on the derivatives of the function $\tilde{f}$ and is called an *elementary differential.* The theory now tells us that there are sequences $\mathbf{u}_i$, $\tilde{\mathbf{k}}_i$, and $\mathbf{y}_1$ such that (formally)

$$u_i = B(\mathbf{u}_i, y_0), \quad \tilde{k}_i = B(\tilde{\mathbf{k}}_i, y_0), \quad y_1 = B(\mathbf{y}_1, y_0), \tag{31}$$

for $i = 1, \ldots, s$. Hairer et al. [6] provide the following recursion formulas for $\mathbf{k}_i$ and $\mathbf{u}_i$

$$\begin{aligned} &\mathbf{u}_i(\emptyset) = 1, && \tilde{\mathbf{k}}_i(\tau) = 1, && \mathbf{y}_1(\emptyset) = 1, \\ &\mathbf{u}_i(t) = \sum_{j=1}^{s} a_{ij}\tilde{\mathbf{k}}_j(t), && \tilde{\mathbf{k}}_i(t) = \rho(t)\mathbf{u}_i(t_1)\cdots\mathbf{u}_i(t_m), && \mathbf{y}_1(t) = \sum_{j=1}^{s} b_j\tilde{\mathbf{k}}_j(t) \end{aligned} \tag{32}$$

where the trees $t_1, \ldots, t_m$ are such that $t = [t_1, \ldots, t_m]$. Given any method (A, b) we can, in view of (32) and (30), derive conditions such that (29) holds for q as large as possible. Before we proceed to specific examples, recall that we really need the k_i rather than the $\tilde{k}_i$ to satisfy (29), hence the following result will be useful to us.

Proposition 4.1 *In the RKMK method (14) assume that $r = (r_1, \ldots, r_m)^T$ is such that (29) holds. Then*

$$\sum_{i=1}^{m} r_i k_i = \mathcal{O}(h^{q+1}). \tag{33}$$

Proof: We have from (29), (30) and (31) that

$$\sum_{i=1}^{m} r_i\, \tilde{\mathbf{k}}_i(t) = 0 \quad \text{for all } t \text{ such that } \rho(t) \leq q. \tag{34}$$

We compute

$$k_i = d\exp_{u_i}(\tilde{k}_i) = \tilde{k}_i + \frac{1}{2}[u_i, \tilde{k}_i] + \cdots + \frac{1}{(q+1)!}[\underbrace{u_i, [\ldots, [u_i,}_{q \text{ times}} \tilde{k}_i]]] + \cdots \tag{35}$$

We now substitute the B-series for u_i and $\tilde{k}_i$ in the general term of (35) where $q \geq 1$ to obtain

$$\frac{1}{(q+1)!}\left[\sum_{t_1 \in T} \frac{h^{\rho(t_1)}}{\rho(t_1)!}\mathbf{u}_i(t_1)F(t_1), \left[\ldots, \left[\sum_{t_q \in T} \frac{h^{\rho(t_q)}}{\rho(t_q)!}\mathbf{u}_i(t_q)F(t_q), \sum_{t_{q+1} \in T} \frac{h^{\rho(t_{q+1})}}{\rho(t_{q+1})!}\tilde{\mathbf{k}}_i(t_{q+1})F(t_{q+1})\right]\right]\right].$$

From this expression, we collect all terms of a given order ℓ in the stepsize h, thus, we sum over all ordered $(q+1)$-plets $(t_1, \ldots, t_{q+1})$ such that $\rho(t_1) + \cdots + \rho(t_{q+1}) = \ell$:

$$\sum_{\ell \geq q+1} \frac{h^{\ell}}{(q+1)!} \sum_{t_1, \ldots, t_{q+1}} \frac{\mathbf{u}_i(t_1)\cdots\mathbf{u}_i(t_q)}{\rho_1! \cdots \rho_{q+1}!}\tilde{\mathbf{k}}_i(t_{q+1})[F(t_1), [\ldots, [F(t_q), F(t_{q+1})]]]. \tag{36}$$

Now define the compound tree (of $t_1, \ldots, t_q, t_{q+1}$) as follows:

$$t = \begin{cases} [t_1, \ldots, t_q] & \text{if } t_{q+1} = \tau \\ [t_1, \ldots, t_q, t_{q+1,1}, \ldots, t_{q+1,\mu}] & \text{if } t_{q+1} = [t_{q+1,1}, \ldots, t_{q+1,\mu}]. \end{cases}$$

It follows from (32) that (36) reduces to

$$\sum_{\ell \geq q+1} \frac{h^\ell}{(q+1)!} \sum_{t \in T_\ell} \tilde{\mathbf{k}}_i(t) \sum_{\sigma \in S_{q+1}} K_\sigma \left[F(t_{\sigma(1)}), [\ldots, [F(t_{\sigma(q)}), F(t_{\sigma(q+1)})]]\right]$$

where T_ℓ is the set of rooted trees with precisely ℓ nodes, S_{q+1} is the symmetric group of $q+1$ elements, and K_σ is a constant which depends only on σ and on the ρ_i. The result now follows readily from (34). □

We next consider how to derive conditions on r for which (29) and thereby (33) holds. We enumerate the rooted trees increasingly in terms of their order, i.e. if $\rho(t_i) < \rho(t_j)$ then $i < j$, and for each $q \geq 1$ let N_q be the number of trees t such that $\rho(t) \leq q$. We define $\tilde{\mathbf{K}}_{q,m}$ to be the $N_q \times m$ matrix whose ij-element is $\tilde{\mathbf{k}}_j(t_i)$. We see that if (34) is to hold for some nonzero $r \in \mathbb{R}^m$ it is necessary and sufficient that

$$\operatorname{rank} \tilde{\mathbf{K}}_{q,m} < m.$$

Note that since we are considering only explicit methods, the first $m^* < m$ columns of $\tilde{\mathbf{K}}_{q,m}$ are precisely those of $\tilde{\mathbf{K}}_{q,m^*}$. Therefore, it is clear that highest attainable q in (29) is obtained for $m = s$ the number of stages of the method. From a result by Owren and Zennaro [17, Theorem 2.4] we deduce

Proposition 4.2 *Let (A, b) be an ERK method of order p and assume that the corresponding matrix $\tilde{\boldsymbol{K}}_{p,s}$ is of rank $s^* < s$. Then there exists a method (A^*, b^*) with s^* stages of order p.*

All of the most popular ERK methods have the property that the corresponding matrix $\tilde{\mathbf{K}}_{p,s}$ has full rank, and for such methods, we have the upper bound $q \leq s$ in (29). This means that it suffices to consider the matrix $\tilde{\mathbf{K}}_{p-1,s}$ (and its submatrices). Likewise, we easily see that (29) can be satisfied if $N_q \leq m$, and we thus have also a lower bound for the largest attainable q. One should also note that all stage values $\tilde{k}_i$ are $\mathcal{O}(h)$, hence, with $m = 1$ we obtain $\tilde{k}_1 = \mathcal{O}(h)$, i.e. $q = 0$ in (29).

Examples. We start by considering RK4, "the Runge-Kutta method" with Butcher tableau

$$\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
1/2 & 0 & 1/2 & & \\
1 & 0 & 0 & 1 & \\
\hline
 & 1/6 & 1/3 & 1/3 & 1/6
\end{array}$$

We need only consider $\tilde{\mathbf{K}}_{3,4}$ and its submatrices

$$\tilde{\mathbf{K}}_{3,4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 3/4 & 3/4 & 3 \\ 0 & 0 & 3/2 & 3 \end{bmatrix}, \qquad \begin{array}{c|cccc} q \backslash m & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 2 & 3 \\ 2 & 0 & 0 & 1 & 2 \\ 3 & 0 & 0 & 0 & 0 \end{array}$$

To the right we have listed the dimensions of the kernel of the submatrices $\tilde{\mathbf{K}}_{q,m}$ for the interesting values of q and m. The numbers in the table give, for each m, the number of independent vectors $r \in \mathbb{R}^m$ that can be used to satisfy (29) for each $q = 1, 2, 3$. We may now define quantities Q_1, Q_2, Q_3, Q_4 as follows with the given order in the stepsize h, thanks to Propositon 4.1

$$\begin{aligned} Q_1 &= k_1 = \mathcal{O}(h) \\ Q_2 &= k_2 - k_1 = \mathcal{O}(h^2) \\ Q_3 &= k_3 - k_2 = \mathcal{O}(h^3) \\ Q_4 &= k_1 - 2k_2 + k_4 = \mathcal{O}(h^3). \end{aligned} \tag{37}$$

In order to approximate $d\exp_{u_i}^{-1}(k_i)$ as it appears in (14), we use again the expansion (13), but by rephrasing the algorithm in terms of $Q_1, \ldots, Q_4$ it is now only necessary to retain terms of order $\mathcal{O}(h^q)$, $q \leq 4$. Consider the graded free Lie algebra generated by $Q_1, \ldots, Q_4$ with grades $w(Q_1) = 1$, $w(Q_2) = 1$, $w(Q_3) = 3$, $w(Q_4) = 3$, as suggested by (37). The associated polynomial p of (17) is thus given as $p(T) = 1 - T - T^2 - 2T^3$ whose roots are $T_1 = \frac{1}{2}$, $T_{2,3} = \exp(\pm 2\pi i/3)$. We can apply the formula (18) to find the following dimensions ν_n of the homogeneous components of the free Lie algebra up to $n = 10$.

n	1	2	3	4	5	6	7	8	9	10
ν_n	1	1	3	3	6	9	18	30	56	99

Note also that we can write $p(T) = (1 - 2T)(1 - T^3)(1 - T)^{-1}$ thus, according to (21) the roots $T_{2,3}$ have no effect on the homogeneous dimensions ν_n when $n \geq 4$. For such n the dimensions are the same as for the free Lie algebra generated by two elements, both of grade 1. For the present application, only the first 4 entries of the above table are of significance, we can easily find a corresponding Hall basis of 8 elements namely

$$Q_1,\ Q_2,\ Q_3,\ Q_4,\ [Q_1, Q_2],\ [Q_1, Q_3],\ [Q_1, Q_4],\ [Q_1, [Q_1, Q_2]].$$

There are thus at most four brackets that need to be computed to evaluate (13) to the sufficient order of accuracy. Using $k_1, \ldots, k_4$, one needs instead to compute 6 brackets, taking into account that for the first stage $d\exp_{u_1}^{-1}$ is the identity transformation.

We write out the new version of the RKMK algorithm based on RK4. Note that the u_i needs only to be computed to the order $q - 1$. To see this, consider the original RKMK scheme (14), it suffices to compute the corrected stages $\tilde{k}_i$ to the order of the method. These are obtained by applying $d\exp_{u_i}^{-1}$ to k_i and the k_i are all $\mathcal{O}(h)$ hence the order of the error in u_i is boosted by 1 after applying $d\exp_{u_i}^{-1}$ to k_i. Thus we drop all terms exceeding order 3 in u_i and all terms exceeding 4 in v. This gives:

$$\begin{aligned} u_1 &= 0 && \\ k_1 &= hf(\lambda(u_1, y_0)) && Q_1 := k_1 \\ u_2 &= \tfrac{1}{2}Q_1 && \\ k_2 &= hf(\lambda(u_2, y_0)) && Q_2 := k_2 - k_1 \\ u_3 &= \tfrac{1}{2}Q_1 + \tfrac{1}{2}Q_2 - \tfrac{1}{8}[Q_1, Q_2] && \\ k_3 &= hf(\lambda(u_3, y_0)) && Q_3 := k_3 - k_2 \\ u_4 &= Q_1 + Q_2 + Q_3 && \\ k_4 &= hf(\lambda(u_4, y_0)) && Q_4 := k_4 - 2k_2 + k_1 \\ v &= Q_1 + Q_2 + \tfrac{1}{3}Q_3 + \tfrac{1}{6}Q_4 - \tfrac{1}{6}[Q_1, Q_2] - \tfrac{1}{12}[Q_1, Q_4] && \\ y_1 &= \lambda(v, y_0) && \end{aligned}$$

Thus the number of commutators is reduced from six to two. Interestingly, this scheme is almost identical to the original 4. order algorithm derived in [14] by other means.

We proceed to consider the much used fifth order method DOPRI5(4), which has a total of 7 stages, but where the 7th stage is used only for error estimation. The Butcher tableau can be found in [6, p. 178]. We compute $\tilde{\mathbf{K}}_{4,7}$ and the dimensions of the kernel of the submatrices.

$$
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & \frac{2}{5} & \frac{3}{5} & \frac{8}{5} & \frac{16}{9} & 2 & 2 \\
0 & \frac{3}{25} & \frac{27}{100} & \frac{48}{25} & \frac{64}{27} & 3 & 3 \\
0 & 0 & \frac{27}{100} & \frac{48}{25} & \frac{64}{27} & 3 & 3 \\
0 & \frac{4}{125} & \frac{27}{250} & \frac{256}{125} & \frac{2048}{729} & 4 & 4 \\
0 & 0 & \frac{27}{250} & \frac{256}{125} & \frac{2048}{729} & 4 & 4 \\
0 & 0 & \frac{27}{250} & \frac{256}{125} & \frac{2048}{729} & 4 & 4 \\
0 & 0 & 0 & \frac{96}{25} & \frac{3392}{405} & \frac{504}{55} & 4
\end{bmatrix},
\qquad
\begin{array}{c|ccccccc}
q\backslash m & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
2 & 0 & 0 & 1 & 2 & 3 & 4 & 5 \\
3 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}
$$

We find $Q_1, \ldots, Q_7$ as follows

$$
\begin{aligned}
Q_1 &= k_1 = \mathcal{O}(h) \\
Q_2 &= k_2 - k_1 = \mathcal{O}(h^2) \\
Q_3 &= k_3 - \frac{3}{2}\,k_2 + \frac{1}{2}\,k_1 = \mathcal{O}(h^3) \\
Q_4 &= k_4 - 6\,k_3 + 5\,k_2 = \mathcal{O}(h^3) \\
Q_5 &= k_5 - \frac{106}{81}\,k_4 + \frac{128}{243}\,k_3 - \frac{53}{243}\,k_1 = \mathcal{O}(h^4) \\
Q_6 &= k_6 - \frac{567}{212}\,k_5 + \frac{7}{4}\,k_4 - \frac{4}{53}\,k_3 = \mathcal{O}(h^4) \\
Q_7 &= k_7 - \frac{176}{105}\,k_6 + \frac{17253}{8480}\,k_5 - \frac{71}{48}\,k_4 + \frac{568}{3339}\,k_3 - \frac{71}{1440}\,k_1 = \mathcal{O}(h^5).
\end{aligned}
\tag{38}
$$

In this case the polynomial (17) takes the form

$$
p(T) = 1 - T - T^2 - 2T^3 - 2T^4 - T^5 = \left[(1 - T^4)(1 - T^2)^{-1}\right](1 - T - 2T^2 - T^3).
$$

It is interesting to see that for $n \geq 4$ the dimensions ν_n of the graded FLA generated by $Q_1, \ldots, Q_7$ coincide with those of the graded FLA generated by elements $Q'_1, \ldots, Q'_4$ having weights $1, 2, 2, 3$.

A Hall basis for elements up to order 5 based on $Q_1, \ldots, Q_7$ is

$$
\begin{aligned}
&Q_1,\ Q_2,\ Q_3,\ Q_4,\ Q_5,\ Q_6,\ Q_7,\ [Q_1,Q_2],\ [Q_1,Q_3],\ [Q_1,Q_4],\ [Q_1,Q_5],\ [Q_1,Q_6],\\
&[Q_2,Q_3],\ [Q_2,Q_4],\ [Q_1,[Q_1,Q_2]],\ [Q_1,[Q_1,Q_3]],\ [Q_1,[Q_1,Q_4]],\ [Q_2,[Q_1,Q_2]],\\
&[Q_1,[Q_1,[Q_1,Q_2]]].
\end{aligned}
$$

In other words, at most 12 brackets are involved in the algorithm. In comparison, using the expansion (13) one would need 24 brackets.

The modified RKMK algorithm is as follows:

for j=1:7,

u_j is determined from (39)

$k_j = hf(\lambda(u_j, y_0))$

Q_j is determined from (38)

Form all new brackets involving $Q_1, \ldots, Q_j$ from the set above

end for

Set $y_1 = \lambda(v, y_0) = \lambda(u_7, y_0)$

The calculation of $u_1, \ldots, u_7$ is lengthy but straightforward. We give only the final result. As before, we ignore all terms of order exceeding 4 in u_i and exceeding 5 in v.

$$
\begin{aligned}
u_1 &= 0 \\
u_2 &= \frac{1}{5}Q_1 \\
u_3 &= \frac{3}{10}Q_1 + \frac{9}{40}Q_2 - \frac{9}{400}[Q_1, Q_2] + \frac{3}{4000}[Q_1, [Q_1, Q_2]] \\
u_4 &= \frac{4}{5}Q_1 + \frac{8}{5}Q_2 + \frac{32}{9}Q_3 - \frac{2}{75}[Q_1, Q_2] - \frac{8}{15}[Q_1, Q_3] - \frac{73}{2250}[Q_1, [Q_1, Q_2]] \\
u_5 &= \frac{8}{9}Q_1 + \frac{160}{81}Q_2 + \frac{2795}{346}Q_3 - \frac{212}{729}Q_4 + \frac{628}{2187}[Q_1, Q_2] - \frac{1118}{865}[Q_1, Q_3] \\
&\quad + \frac{424}{3645}[Q_1, Q_4] - \frac{157}{1297}[Q_1, [Q_1, Q_2]] \\
u_6 &= Q_1 + \frac{5}{2}Q_2 + \frac{3395}{396}Q_3 - \frac{7}{88}Q_4 - \frac{433}{1583}Q_5 + \frac{4}{33}[Q_1, Q_2] - \frac{455}{264}[Q_1, Q_3] \\
&\quad + \frac{7}{80}[Q_1, Q_4] - \frac{194}{1393}[Q_1, [Q_1, Q_2]] \\
v &= Q_1 + \frac{5}{2}Q_2 + \frac{115}{36}Q_3 + \frac{11}{24}Q_4 + \frac{189}{6784}Q_5 + \frac{11}{84}Q_6 - \frac{5}{12}[Q_1, Q_2] - \frac{55}{72}[Q_1, Q_3] \\
&\quad - \frac{7}{48}[Q_1, Q_4] - \frac{407}{8181}[Q_1, Q_5] - \frac{11}{168}[Q_1, Q_6] - \frac{25}{36}[Q_2, Q_3] - \frac{5}{24}[Q_2, Q_4] \\
&\quad + \frac{5}{216}[Q_1, [Q_1, Q_3]] + \frac{1}{144}[Q_1, [Q_1, Q_4]] - \frac{5}{48}[Q_2, [Q_1, Q_2]] + \frac{1}{144}[Q_1, [Q_1, [Q_1, Q_2]]] \\
u_7 &= v
\end{aligned}
\tag{39}
$$

One may ask if the use of such transformations described above, will generally lead to less expensive approximation of $d\exp_u^{-1}$. This is not necessarily so, because the expansion (13) does not contain brackets from a full Hall basis based on, say, $u_1, \ldots, u_s$, $k_1, \ldots, k_s$, all brackets of n elements involve precisely $n-1$ occurrences of u_i and one of k_i, there are no "mixed terms". To put this discussion to a test, we computed the transformation for the RKF78 method which has 13 stages. We found that the corresponding graded FLA involve 133 brackets of degree not exceeding 8, whereas the direct computation of (13) involves only 72 brackets. It seems that, unless this new approach is combined with some other reduction techniques, we can only expect it to be less expensive for low and moderate order methods.

4.4 Explicit RKCG methods for general equations on manifolds

The methods proposed my Crouch and Grossman in [3] are to a large extent using compositions of vector fields, see (6) and (7). The idea is to approximate these compositions of exponentials, say

$$\exp(v_r hF_r) \circ \cdots \circ \exp(v_1 hF_1)$$

by the exponential of a *single* vector field $\hat{F}$ obtained by truncating the BCH-formula (5).

In a similar way as for the RKMK methods we will consider linear combinations of the vector fields F_i such that for each m, $1 \leq m \leq s$, we find $r = (r_1, \ldots, r_m)$, $1 \leq m \leq s$, such that

$$\sum_{i=1}^{m} r_i \, hF_i = \mathcal{O}(h^{q+1}) \tag{40}$$

for q as large as possible. We need to use the order theory as developed in [16], and there are more conditions to be fulfilled than in the RKMK case. Again, the attainable order q in (29) depends on the particular method which is used. Suppose that we have found an invertible linear transformation of the vector fields $hF_1, \ldots, hF_s$ to yield vector fields $Q_1, \ldots, Q_s$ such that each Q_i depends only on $hF_1, \ldots, hF_i$. To verify the existence of such a transformation it suffices to take the identity transformation. Suppose that the positive integers $q_1, \ldots, q_s$ are such that $Q_i = \mathcal{O}(h^{q_i})$. Let $\mathfrak{g}$ be the graded FLA generated by $Q_1, \ldots, Q_s$ where $w(Q_i) = q_i$. We expand (29) by the BCH formula, substitute for each hF_i the linear combination of the Q_i determined from the inverse of the above transformation and write the resulting expression in terms of the Hall basis. We can discard all terms with degree greater than the order of the method.

Example. We consider the 4th order 5 stage RKCG method given in [16]. With the coefficients of that particular method, we find that we can define $Q_1, \ldots, Q_5$ as follows

$$\begin{aligned}
Q_1 &= hF_1 = \mathcal{O}(h) \\
Q_2 &= hF_2 - hF_1 = \mathcal{O}(h^2) \\
Q_3 &= hF_3 - \frac{1}{9}(4 + 2\,\kappa + \kappa^2)\, hF_2 + \frac{1}{9}(-5 + 2\,\kappa + \kappa^2)\, hF_1 = \mathcal{O}(h^3) \\
Q_4 &= hF_4 + \frac{1}{9}(-2 + 2\kappa + \kappa^2)\, hF_2 - \frac{1}{9}(7 + 2\kappa + \kappa^2)\, hF_1 = \mathcal{O}(h^3) \\
Q_5 &= hF_5 + (\kappa^2 - 1)\, hF_4 - (\kappa^2 - 1)\, hF_3 - hF_1 = \mathcal{O}(h^4)
\end{aligned}$$

where $\kappa = 2^{1/3}$. We invert these equations to find

$$\begin{aligned}
hF_1 &= Q_1 \\
hF_2 &= Q_1 + Q_2 \\
hF_3 &= Q_1 + \frac{1}{9}(4 + 2\kappa + \kappa^2)\, Q_2 + Q_3 \\
hF_4 &= Q_1 - \frac{1}{9}(-2 + 2\kappa + \kappa^2)\, Q_2 + Q_4 \\
hF_5 &= Q_1 + \frac{2}{3} Q_2 + (\kappa^2 - 1)\, Q_3 - (\kappa^2 - 1)\, Q_4 + Q_5.
\end{aligned}$$

We may thus consider the graded free Lie algebra based on the elements $Q_1, \ldots, Q_5$ with weights

element	Q_1	Q_2	Q_3	Q_4	Q_5
weight	1	2	3	3	4

We now compute the Hall basis and exclude terms whose degree exceeds 4 (the order of the method) and we find that the set

$$\mathcal{S} = \{Q_1,\ Q_2,\ Q_3,\ Q_4,\ Q_5,\ [Q_1,Q_2],\ [Q_1,Q_3],\ [Q_1,Q_4],\ [Q_1,[Q_1,Q_2]]\}$$

constitute all terms of order ≤ 4.

To take a step with this Crouch-Grossman method, we begin as before by setting $Y_1 = y_n$ and computing

$$F_1 = F_{Y_1} = \sum_i f_i(Y_1)E_i.$$

Compute $Y_2 = \exp(ha_{21}F_i)\, y_0$ and then

$$F_2 = F_{Y_2} = \sum_i f_i(Y_2)E_i.$$

For each $Y_i,\ i \geq 3$ we must compute

$$Y_i = \exp(ha_{i,i-1}F_{i-1}) \circ \cdots \circ \exp(ha_{i1}F_1)\, y_0 =: \exp(h\hat{F}_{i-1})\, y_0$$

and finally

$$y_1 = \exp(hb_sF_s) \circ \cdots \circ \exp(hb_1F_1)\, y_0 =: \exp(h\hat{F})\, y_0$$

The vector fields $h\hat{F}_{i-1}$, $h\hat{F}$ can be approximated to the order of the method, by means of the BCH formula, the vector fields Q_i are substituted into the resulting expressions and we retain only the terms belonging to the set $\mathcal{S}$. As an example we compute here $h\hat{F}_2$:

$$\begin{aligned}
h\hat{F}_2 &= a_{32}hF_1 + a_{32}hF_2 + \frac{1}{2}a_{31}a_{32}\,[hF_2, hF_1] + \frac{1}{12}a_{31}^2a_{32}\,[hF_1,[hF_1,hF_2]] \\
&\quad + \frac{1}{12}a_{31}a_{32}^2\,[hF_2,[hF_2,hF_1]] + \mathcal{O}(h^5) \\
&= a_{31}Q_1 + a_{32}(Q_1+Q_2) + \frac{1}{2}a_{31}a_{32}[Q_1+Q_2, Q_1] + \frac{1}{12}a_{31}^2a_{32}[Q_1,[Q_1,Q_1+Q_2]] \\
&\quad + \frac{1}{12}a_{31}a_{32}^2[Q_1+Q_2,[Q_1+Q_2,Q_1]] + \mathcal{O}(h^5) \\
&= c_3Q_1 + a_{32}Q_2 - \frac{1}{2}a_{31}a_{32}[Q_1,Q_2] + \frac{1}{12}a_{31}a_{32}(a_{31}-a_{32})[Q_1,[Q_1,Q_2]] + \mathcal{O}(h^5).
\end{aligned}$$

The other compositions are computed in a similar way and we obtain vector fields of the form

$$h\hat{F}_i = \sum_{S\in\mathcal{S}} \alpha_{iS}S + \mathcal{O}(h^5), \quad i = 2,\ldots,s-1, \quad h\hat{F} = \sum_{S\in\mathcal{S}} \beta_S S + \mathcal{O}(h^5).$$

The exact expressions for the coefficients α_{iS} and β_S are fairly complicated, we prefer to give the nonzero ones below as decimal numbers.

$S \backslash i$	α_{2S}	α_{3S}
Q_1	0.13512071919596576e+01	-0.35120719195965763e+00
Q_2	0.60858695853450102e+00	0.41115497228062601e-01
Q_3		-0.44926882389532706e+01
$[Q_1, Q_2]$	-0.22597449460319862e+00	-0.91204491514594451e+00
$[Q_1, Q_3]$		0.93031915958380413e+01
$[Q_1, [Q_1, Q_2]]$	0.50480169255712481e-02	0.13271581092769965e+01

$S \backslash i$	α_{4S}	β_S
Q_1	0.10000000000000000e+01	0.10000000000000000e+01
Q_2	0.33333333333333333e+00	0.33333333333333333e+00
Q_3	-0.10764581018542984e+01	0.22124666701222105e+00
Q_4	-0.35120719195965763e+00	-0.57245385897187868e+00
Q_5		0.67560359597982881e+00
$[Q_1, Q_2]$	-0.31635294839677244e+00	-0.55555555555555555e-01
$[Q_1, Q_3]$	0.14956699006655000e+01	-0.48950087664016622e-01
$[Q_1, Q_4]$	-0.23727684182192271e+00	0.48950087664016622e-01
$[Q_1, [Q_1, Q_2]]$	0.15744757362188735e+00	

5 Open problems

We have been investigating reductions in commutator computations by changing from ungraded to graded bases in the free Lie algebra. There are, however, other optimization issues related to changing bases that have not been addressed. Consider the BCH computation, $Z = \mathrm{bch}(X, Y)$, in the non-graded case. The Witt formula (15) for $s = 2$ gives an *upper bound* on the number of commutators. The actual number of commutators becomes lower than this, due to the symmetry $\mathrm{bch}(X, Y) = -\mathrm{bch}(-Y, -X)$, and possibly also due to other symmetries. However, the actual savings due to symmetries depends on the choice of basis, and for this particular computation the *Lyndon* basis gives slightly fewer commutators than the Hall basis. Further symmetry reductions are possible by doing other transformations of the basis. This is addressed in [10, 15]. Reduction of commutators by systematic use of symmetries seems to be an open but important question, both for ungraded and graded free Lie algebras.

Another important problem for future research is the optimization in the case of special (non-free) Lie algebras, that is when more information about the structure constants of the algebra is available. Such optimization problems may turn out to be computationally hard to solve.

Acknowledgment. Most of this work was done at DAMTP in Cambridge where both authors spent the Michaelmass term 1997. We would like to thank Dr. Arieh Iserles for making the stay in Cambridge possible. We would also like to thank the following people for valuable input: Kenth Engø, Joseph Keller, Arne Marthinsen, and Antonella Zanna.

References

[1] M. Barr and C. Wells. *Category Theory for Computing Science.* Prentice Hall, 1990.

[2] N. Bourbaki. *Lie groups and Lie algebras.* Part I, Chapters 1–3, Addison-Wesley, 1975.

[3] P. E. Crouch and R. Grossman. Numerical Integration of Ordinary Differential Equations on Manifolds. *J. Nonlinear Sci.*, 3:1–33, 1993.

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

[5] E. Hairer. Backward analysis of numerical integrators and symplectic methods. In K. Burrage, C. Baker, P. v.d. Houwen, Z. Jackiewicz, and P. Sharp, editors, *Scientific Computation and Differential Equations*, volume 1 of *Annals of Numer. Math.*, pages 107–132, Amsterdam, 1994. J.C. Baltzer. Proceedings of the SCADE'93 conference, Auckland, New-Zealand, January 1993.

[6] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I, Nonstiff Problems.* Springer-Verlag, Second revised edition, 1993.

[7] A. Iserles. Solving Linear Ordinary Differential Equations by Exponentials of Iterated Commutators. *Numer. Math.*, 45:183–199, 1984.

[8] A. Iserles and S. P. Nørsett. On the Solution of Linear Differential Equations in Lie Groups. Technical Report 1997/NA3, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997.

[9] G. Jacob and P.-V. Koseleff (Eds.). Special Issue on Lie Computations. *Discrete Mathematics and Theoretical Computer Science*, 1(1):99–266, 1997. Electronic journal: **http://dmtcs.thomsonscience.com**.

[10] M. Kolsrud. Maximal reductions in the Baker-Hausdorff formula. *J. Math. Phys.*, 34(1):270–285, 1993.

[11] W. Magnus. On the Exponential Solution of Differential Equations for a Linear Operator. *Comm. Pure and Appl. Math.*, VII:649–673, 1954.

[12] R. I. McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM J. Sci. Comput.*, 16:151–168, 1995.

[13] H. Munthe-Kaas. High Order Runge-Kutta Methods on Manifolds. Technical Report 1997/NA14, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997. To appear in Journal of Applied Numerical Mathematics, Elsevier.

[14] H. Munthe-Kaas. Runge-Kutta Methods on Lie Groups. *BIT*, 38(1):92–111, 1998.

[15] J.A. Oteo. The Baker-Campbell-Hausdorff formula and nested commutator identities. *J. Math. Phys.*, 32(2):419–424, 1991.

[16] B. Owren and A. Marthinsen. Integration Methods Based on Rigid Frames. Technical Report Numerics No. 1/1997, Department of Mathematical Sciences, The Norwegian University of Science and Technology, 1997.

[17] B. Owren and M. Zennaro. Order Barriers for Continuous Explicit Runge-Kutta Methods. *Math. Comp.*, 56:645–661, 1991.

[18] A. Flø Rasmussen. Solving Lie-type equations numerically with Magnus series. Master's thesis, NTNU, Trondheim, Norway, 1997.

[19] S. Reich. Backward error analysis for numerical integrators. Technical Report SC 96-21, Konrad-Zuse Zentrum für Informationstechnik Berlin, 1996.

[20] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian Problems.* AMMC 7. Chapman & Hall, 1994.

[21] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.

[22] V. S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representations.* GTM 102. Springer-Verlag, 1984.

[23] H. Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150:262–268, 1990.