

---

# On the solution of linear differential equations in Lie groups

A. Iserles and S.P. Nørsett

*Phil. Trans. R. Soc. Lond. A* 1999 **357**, 983-1019

doi: 10.1098/rsta.1999.0362

---

## Email alerting service

Receive free email alerts when new articles cite this article - sign up in the box at the top right-hand corner of the article or click [here](#)

---

To subscribe to *Phil. Trans. R. Soc. Lond. A* go to: <http://rsta.royalsocietypublishing.org/subscriptions>

---

# On the solution of linear differential equations in Lie groups

BY A. ISERLES<sup>1</sup> AND S. P. NØRSETT<sup>2</sup>

<sup>1</sup>*Department of Applied Mathematics and Theoretical Physics,  
University of Cambridge, Silver Street, Cambridge CB3 9EW, UK*

<sup>2</sup>*Institute of Mathematical Sciences, Norwegian University of Science and  
Technology, Trondheim, Norway*

The subject matter of this paper is the solution of the linear differential equation  $y' = a(t)y$ ,  $y(0) = y_0$ , where  $y_0 \in G$ ,  $a(\cdot) : \mathbb{R}^+ \rightarrow \mathfrak{g}$  and  $\mathfrak{g}$  is a Lie algebra of the Lie group  $G$ . By building upon an earlier work of Wilhelm Magnus, we represent the solution as an infinite series whose terms are indexed by binary trees. This relationship between the infinite series and binary trees leads both to a convergence proof and to a constructive computational algorithm. This numerical method requires the evaluation of a large number of multivariate integrals, but this can be accomplished in a tractable manner by using quadrature schemes in a novel manner and by exploiting the structure of the Lie algebra.

**Keywords:** Lie groups; differential equation; rooted trees; quadrature

## 1. Introduction

The theme underlying this paper is the solution of the differential equation

$$y' = a(t)y, \quad t \geq 0, \quad y(0) = y_0 \in G, \quad (1.1)$$

where  $G$  is a Lie group,  $a : \mathbb{R}^+ \rightarrow \mathfrak{g}$  is Lipschitz continuous and  $\mathfrak{g}$  is the Lie algebra of  $G$ . It is well known that the solution of (1.1) stays on  $G$  for all  $t \geq 0$ . The retention of this important structural feature of the differential equation under discretization is at the centre of our discussion.

We briefly recall that a *Lie group*  $G$  is a differentiable manifold, equipped with a group structure that is continuous with respect to the underlying topology of the manifold. The *Lie algebra*  $\mathfrak{g}$  is the tangent space of  $G$ : the set of all possible values of  $\gamma'(0)$ , where  $\gamma(t) \in G$  is a smooth curve and  $\gamma(0) = \text{Id}$ , the identity of  $G$ . It is a linear space closed under a binary operation  $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$ . This binary operation (often termed ‘commutation’ or ‘a Lie bracket’) is linear in each component, antisymmetric:

$$[a, b] = -[b, a], \quad a, b \in \mathfrak{g},$$

and subject to the *Jacobi identity*

$$[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0. \quad (1.2)$$

A crucial feature of  $\mathfrak{g}$  is that it can be mapped to  $G$  by the *exponential map*: given  $b \in \mathfrak{g}$  and  $t \in \mathbb{R}^+$ ,  $\exp(tb) \in G$  is the flow generated by the *infinitesimal generator*  $b$  (Olver 1995).

An important special case is when  $G$  is a subgroup of either  $\text{GL}_n(\mathbb{R})$  or  $\text{GL}_n(\mathbb{C})$ , the *general linear group* of all real or complex non-singular  $n \times n$  matrices. It is

then called a *matrix Lie group* and important special cases are:  $SL_n(\mathbb{R})$ , the *special linear group* of real matrices with a unit determinant;  $U_n(\mathbb{C})$ , the *unitary group* of complex unitary matrices;  $O_n(\mathbb{R})$ , the *orthogonal group* of real orthogonal matrices; and the *special unitary* and *special orthogonal groups*  $SU_n(\mathbb{C}) = U_n(\mathbb{C}) \cap SL_n(\mathbb{C})$  and  $SO_n(\mathbb{R}) = O_n(\mathbb{R}) \cap SL_n(\mathbb{R})$ , respectively. The corresponding Lie algebras are well known (Olver 1995). In the case of matrix Lie groups, the exponential operator is just

$$\exp(tb) = \sum_{k=0}^{\infty} \frac{1}{k!} (tb)^k,$$

while  $[a, b] = ab - ba$  is the familiar commutator of two matrices.

We assume in the sequel that  $G$  is a matrix Lie group, whereby (1.1) becomes a matricial *ordinary differential system*. This restriction is more in the nature of conceptualization than essence, since much of our theory (but not necessarily our computational algorithms) remains intact for general Lie groups. An important case when computation leads to additional challenges in a non-matricial case is when  $\mathfrak{g}$  consists of differential operators. This issue is outside the realm of this paper.

The retention of a Lie-group structure is often an imperative in numerical discretization, since it represents invariants that are satisfied by the original differential equations. Important examples, ubiquitous in applications, include the conservation of orthogonality, volume and symplecticity (Iserles & Zanna 1996). Arguably, even more important are invariants that correspond to differential systems on *homogeneous manifolds* (manifolds that are invariant when subjected to a transitive group action). As long as it is known how to discretize in a Lie group, it is possible also to discretize flows on homogeneous manifolds (Munthe-Kaas 1997; Munthe-Kaas & Zanna 1997).

There are two points of departure for this paper. Firstly, in an important paper, Wilhelm Magnus (1954) derived ‘the continuous analogue of the Baker–Hausdorff formula’, an expansion,

$$\begin{aligned} \sigma(t) = & \int_0^t a(\kappa) \, d\kappa - \frac{1}{2} \int_0^t \left[ \int_0^\kappa a(\xi) \, d\xi, a(\kappa) \right] \, d\kappa \\ & + \frac{1}{4} \int_0^t \left[ \int_0^\kappa \left[ \int_0^\xi a(\eta) \, d\eta, a(\xi) \right] \, d\xi, a(\kappa) \right] \, d\kappa \\ & + \frac{1}{12} \int_0^t \left[ \int_0^\kappa a(\eta) \, d\eta, \left[ \int_0^\kappa a(\xi) \, d\xi, a(\kappa) \right] \right] \, d\kappa + \dots, \end{aligned} \quad (1.3)$$

such that

$$y(t) = \exp[\sigma(t)]y_0, \quad t \geq 0. \quad (1.4)$$

Magnus neither proves convergence (to quote from theorem III of Magnus (1954), ‘if certain unspecified conditions of convergence are satisfied, . . .’), nor derives the general form of the expansion (1.3). *Magnus series* have received considerable attention in mathematical physics (Wilcox 1972), control theory (Brockett 1976; Sussman 1986) and the theory of differential equations (Chen 1957). Yet, their coefficients have been derived either by brute force or by a Picard-type iteration, the question of convergence has been moot and, as far as we are aware, no effort has been made

to fashion Magnus series as an effective computational tool. We address ourselves to these issues in this paper, noting in passing that all the operations in (1.3) (integration, commutation and linear combination) respect the Lie algebra. Therefore, by virtue of (1.4), the solution  $y(t)$  lies in the Lie group for every  $t \geq 0$ . This has already been observed by Magnus.

Our second point of departure is the method of *Fer expansions* (Fer 1958), which has been considered by the first author in a numerical setting (Iserles 1984). We let  $y^{[0]} = y_0$  and  $a^{[0]} = a$  and set

$$a^{[m+1]}(t) := \sum_{k=1}^{\infty} (-1)^k \frac{k}{(k+1)!} \operatorname{ad} \left[ \int_0^t a^{[m]}(\kappa) \, d\kappa \right]^k [a^{[m]}(t)], \quad m \in \mathbb{Z}^+, \quad (1.5)$$

where

$$\operatorname{ad}[p]^k[q] := \begin{cases} q, & k = 0, \\ [p, \operatorname{ad}[p]^{k-1}[q]], & k \in \mathbb{N} \end{cases} \quad (1.6)$$

is the *adjoint operator* in  $\mathfrak{g}$ . It is proved in (Iserles 1984) that

$$a^{[m]}(t) = \mathcal{O}(t^{2^{m+1}-2}), \quad t \rightarrow 0;$$

therefore

$$y^{[m]}(t) := \exp \left[ \int_0^t a^{[0]}(\kappa) \, d\kappa \right] \exp \left[ \int_0^t a^{[1]}(\kappa) \, d\kappa \right] \cdots \exp \left[ \int_0^t a^{[m]}(\kappa) \, d\kappa \right] y_0$$

approximates the solution  $y$  of (1.1) to order  $2^{m+1} - 2$ . Judiciously truncating the expansion (1.5) and replacing integrals by quadrature formulae, this becomes the basis for an effective numerical method.

Note that, according to (1.5), if  $a^{[m]}$  lies in a Lie algebra, then so does  $a^{[m+1]}$ . Therefore  $y^{[m+1]}$  remains in the Lie group. It is fair to mention that the purpose of Iserles (1984) was the numerical calculation of the fundamental solution of a linear system of ordinary differential equations, and Lie groups are not even mentioned in that publication. The connection between iterated commutators and Lie-group theory was noted by Casas (1996) and Zanna (1996). It is interesting to mention the origins of the method of iterated commutators in the classical technique of *Lie reduction* (Zanna & Munthe-Kaas 1997).

The plan of this paper is as follows. In §2 we introduce the method of Magnus series and demonstrate that, by indexing the terms in the expansion with a subset of binary trees, it is possible to derive explicit recurrence relations, as well as a convergence proof.

An implementation of Magnus series to a computational end entails an approximate quadrature of a possibly large number of multivariate integrals. This is already demonstrated by the expansion (1.3). As is well known, multivariate quadrature is computationally intensive (Cools 1997), hence presenting a major obstacle toward any numerical application of our theoretical construct. Fortunately, it is possible to reduce the cost of computation very significantly indeed by paying attention to the special form of integrands and domains of integration. As a matter of fact, the cost in function evaluations reduces to that of a single univariate quadrature! Section 3 is concerned with numerical quadrature of multivariate integrals within the context of Magnus series.

Another important saving in a numerical implementation of Magnus series is the reduction in the number of commutators, which, in tandem with function evaluations and exponentiation, is the main contributor to the computational cost of the algorithm. The number of commutators can be overwhelmingly reduced by two devices. Firstly, subjected to our quadrature formulae, different integrals require the evaluation of the same commutators. This can be completely described by exploiting our graph-theoretical interpretation of expansion coefficients. Secondly, the Jacobi identity (1.2) allows the replacement of certain terms within quadrature formulae by linear combinations of other terms. This procedure can be quantified exactly by combinatorial theory (Onischik 1993). A helpful way to rephrase our results is by observing that our quadrature formulae project the solution into a relatively low-dimensional subspace of the Lie algebra. The reduction in the number of commutators is the theme of § 4.

We have already mentioned the method of iterated commutators, which provides an alternative representation of the solution of (1.1) in a Lie group. In a sense, Magnus series and iterated commutators stand for diametrically opposed approaches: the first represents the solution as a single exponential with a ‘complicated’ argument and the second yields  $y(t)$  as an infinite product of exponentials with simple arguments. In § 5 we demonstrate that these approaches can be combined.

Finally, we devote § 6 to a number of numerical examples that demonstrate that our approach works and that, indeed, it offers an avenue toward competitive computational methods. Detailed complexity analysis of different combinations of Magnus series and iterated commutators will be published elsewhere by the present authors.

Section 6 is also concerned with concluding remarks and a brief discussion of the ramifications of Magnus series and of the advantages and obstacles in their numerical implementation.

## 2. Magnus series and binary trees

We wish to represent the solution of (1.1) in the form (1.4). To this end we note the following classical theorems of Hausdorff. Although their proofs are neither new nor particularly complicated, we include them for the sake of completeness and to assist numerical analysts who might lack familiarity with this set of ideas. For simplicity’s sake, the proof of theorem 2.1 is given for the case of a matrix Lie group, whereby  $\exp \sigma$  becomes the standard exponential function.

**Theorem 2.1 (Hausdorff 1906).** *The function  $\sigma$  is the solution of the differential equation*

$$\sum_{l=0}^{\infty} \frac{1}{(l+1)!} \text{ad}[\sigma]^l[\sigma'] = a, \quad t \geq 0, \quad \sigma(0) = 0. \quad (2.1)$$

*Proof.* Letting (1.4) in (1.1), we deduce at once that

$$\left( \frac{d}{dt} e^{\sigma(t)} \right) e^{-\sigma(t)} = a, \quad t \geq 0,$$

and  $\sigma(0) = 0$ . But, expanding into series and changing the order of summation,

$$\begin{aligned} \left(\frac{d}{dt}e^{\sigma(t)}\right)e^{-\sigma(t)} &= \left(\sum_{k=1}^{\infty} \frac{1}{k!} \sum_{j=1}^k \sigma^{j-1} \sigma' \sigma^{k-j}\right) \left[\sum_{l=0}^{\infty} \frac{(-1)^l}{l!} \sigma^l\right] \\ &= \sum_{l=1}^{\infty} \frac{(-1)^l}{l!} \sum_{j=1}^l \left[\sum_{k=j}^l (-1)^k \binom{l}{k}\right] \sigma^{j-1} \sigma' \sigma^{l-j}. \end{aligned}$$

It is trivial to prove by induction that

$$\sum_{k=j}^l (-1)^k \binom{l}{k} = (-1)^j \binom{l-1}{j-1}, \quad 1 \leq j \leq l,$$

and therefore

$$\left(\frac{d}{dt}e^{\sigma(t)}\right)e^{-\sigma(t)} = \sum_{l=0}^{\infty} \frac{(-1)^l}{(l+1)!} \sum_{j=0}^l (-1)^j \binom{l}{j} \sigma^j \sigma' \sigma^{l-j}.$$

The proof is complete since an easy induction affirms the identity

$$\text{ad}[p]^l[q] = \sum_{j=0}^l (-1)^{l-j} \binom{l}{j} p^j q p^{l-j}, \quad p, q \in \mathfrak{g}, \quad l \in \mathbb{Z}^+.$$

■

**Theorem 2.2 (Hausdorff 1906).** Let  $d_0 \neq 0$  and

$$d(z) = \sum_{l=0}^{\infty} d_l z^l, \quad z \in \mathbb{C},$$

be a given function, analytic in a proper neighbourhood of the origin. Then there exists  $t^* > 0$  such that the solution of the equation

$$\sum_{l=0}^{\infty} d_l \text{ad}[\sigma]^l[\sigma'] = a, \quad 0 \leq t \leq t^*, \quad \sigma(0) = 0, \quad (2.2)$$

is the same as that of

$$\sigma' = \sum_{m=0}^{\infty} f_m \text{ad}[\sigma]^m[a], \quad 0 \leq t \leq t^*, \quad \sigma(0) = 0, \quad (2.3)$$

where

$$f(z) := \sum_{m=0}^{\infty} f_m z^m = \frac{1}{d(z)}, \quad z \in \mathbb{C}.$$

*Proof.* Let  $\sigma$  be the solution of (2.3). We wish to demonstrate that it is also the solution of (2.2). It is a trivial consequence of (1.6) that

$$\text{ad}[p]^{l-m}[\text{ad}[p]^m[q]] = \text{ad}[p]^l[q], \quad 0 \leq m \leq l.$$

Hence it follows by substituting  $\sigma'$  on the left-hand side of (2.2) that

$$\begin{aligned} \sum_{l=0}^{\infty} d_l \operatorname{ad}[\sigma]^l[\sigma'] &= \sum_{l=0}^{\infty} d_l \operatorname{ad}[\sigma]^l \left[ \sum_{m=0}^{\infty} f_m \operatorname{ad}[\sigma]^m[a] \right] \\ &= \sum_{m=0}^{\infty} \sum_{l=m}^{\infty} f_m d_{l-m} \operatorname{ad}[\sigma]^{l-m}[\operatorname{ad}[\sigma]^m[a]] \\ &= \sum_{l=0}^{\infty} \left( \sum_{m=0}^l f_m d_{l-m} \right) \operatorname{ad}[\sigma]^l[a]. \end{aligned}$$

Since  $f$  is the reciprocal of  $d$ , it is true that

$$\sum_{m=0}^l f_m d_{l-m} = \begin{cases} 1, & l = 0, \\ 0, & l \in \mathbb{N}, \end{cases}$$

and consequently

$$\sum_{l=0}^{\infty} d_l \operatorname{ad}[\sigma]^l[\sigma'] = a$$

and we recover (2.2). ■

In the case of (2.1), we have

$$d(z) = \frac{e^z - 1}{z},$$

and therefore

$$f(z) = \sum_{m=0}^{\infty} \frac{B_m}{m!} z^m + z = 1 + \frac{1}{2}z + \frac{1}{12}z^2 - \frac{1}{720}z^4 + \frac{1}{30240}z^6 - \frac{1}{1209600}z^8 + \dots, \quad (2.4)$$

where  $\{B_m\}_{m \in \mathbb{Z}^+}$  are *Bernoulli numbers* (Abramowitz & Stegun 1965, p. 804). Note that  $B_{2m+1} = 0$  for  $m \in \mathbb{N}$ .

In the sequel we address ourselves to expansions of the general equation (2.3). Whenever necessary, the coefficients  $\{f_m\}_{m \in \mathbb{Z}^+}$  may be substituted in line with (2.4) but, as a rule, the more general treatment leads to a more transparent and uncluttered exposition. We note in passing that, as long as  $a(t) \in \mathfrak{g}$ ,  $t \geq 0$ , it is true that  $\sigma(t) \in \mathfrak{g}$ ,  $t \geq 0$ . Perhaps the easiest proof follows readily from (2.3) by applying the method of proof of the Picard–Lindelöf theorem and noting that the Lie algebra  $\mathfrak{g}$  is closed with respect to the operations of integration, commutation and linear combination.

Let us examine the Magnus series (1.3). Following its inner logic, we search for a general expansion of the *Lie-algebraic differential equation* (2.3) in terms that are formed by multiple integrations and commutations. Let  $\mathcal{E}$  be the set of the derivatives of all the terms in the expansion. We propose the following two *composition rules*, which recursively define  $\mathcal{E}$ :

- (1)  $a(t) \in \mathcal{E}$ ; and
- (2) if  $p(t), q(t) \in \mathcal{E}$ , then  $[\int_0^t q(\kappa) d\kappa, p(t)] \in \mathcal{E}$ .

It is an easy exercise to verify that the first few terms in (1.3) are recovered with our two composition rules. More general analysis, however, is greatly assisted by identifying the terms in  $\mathcal{E}$  with a subset of planar binary rooted trees. We refer to Harary (1969) for basic concepts of graph theory.

We associate the function  $a$  with the trivial tree of order one, consisting of a single vertex. Moreover, given that  $p(t), q(t) \in \mathcal{E}$  are associated with the planar binary trees  $\tau_1, \tau_2$ , respectively (a relationship that we express as  $p \sim \tau_1, q \sim \tau_2$ ), we associate

$$\left[ \int_0^t q(\kappa) d\kappa, p(t) \right] \sim \begin{array}{c} \tau_1 \\ \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \\ \tau_2 \end{array}$$

In other words, we encode integration in the language of graph theory by appending a vertex to the root of a tree, whereby the new vertex becomes the root. Commutation is denoted by joining the roots of the two trees to a new vertex which, again, becomes the root.

Note that the antisymmetry of the commutator implies that

$$\begin{array}{c} \tau_1 \quad \tau_2 \\ \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \tau_2 \quad \tau_1 \\ \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array} = 0,$$

while the Jacobi identity (1.2) can be recast in a graph-theoretic form as

$$\begin{array}{c} \tau_2 \quad \tau_3 \\ \bullet \\ \diagup \quad \diagdown \\ \tau_1 \quad \bullet \end{array} + \begin{array}{c} \tau_3 \quad \tau_1 \\ \bullet \\ \diagup \quad \diagdown \\ \tau_2 \quad \bullet \end{array} + \begin{array}{c} \tau_1 \quad \tau_2 \\ \bullet \\ \diagup \quad \diagdown \\ \tau_3 \quad \bullet \end{array} = 0. \quad (2.5)$$

Although of no immediate concern in this section, the Jacobi identity is critical to successful implementation of the computational procedure of this paper. Its significance will become clear in § 4.

The terms in (1.3) can be associated readily with binary trees and this helps in visualizing this relationship. It is helpful, as a notational convention and in line with the definition of  $\mathcal{E}$ , to consider  $\sigma'$ , i.e. to ‘strip’ the outer integral in each term, since this somewhat simplifies the notation. Thus,

$$a \sim \omega_0 = \bullet, \quad \left[ \int_0^t a, a \right] \sim \omega_1 = \begin{array}{c} \bullet \\ \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array},$$

$$\left[ \int_0^t a, \left[ \int_0^t a, a \right] \right] \sim \omega_2 = \begin{array}{c} \bullet \quad \bullet \\ \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array}, \quad \left[ \int_0^t \left[ \int_0^\kappa a, a \right], a \right] \sim \omega_3 = \begin{array}{c} \bullet \\ \bullet \\ \diagup \quad \diagdown \\ \bullet \quad \bullet \end{array}.$$

Note that, while  $\omega_2$  follows from a combination of  $\omega_0$  with  $\omega_1$ ,  $\omega_3$  is obtained by combining  $\omega_1$  with  $\omega_0$ . Anticipating notation in the sequel, we denote this in the form

$$\omega_2 = \mathcal{R}(\omega_0, \omega_1), \quad \omega_3 = \mathcal{R}(\omega_1, \omega_0).$$



Of course, such combinations are *not* commutative.

In general, we seek an expansion of the form

$$\sigma(t) = \sum_{k=1}^{\infty} \sum_{\tau \in \mathcal{T}_k} \alpha_{\tau} \int_0^t H_{\tau}(\kappa) d\kappa, \quad (2.6)$$

where  $\mathcal{T}_k$  is the set of all binary trees of order  $k$  (that is, with  $k$  vertices) which corresponds to terms  $H_{\tau}$  that can be obtained by means of our two composition rules. Thus, for example,  $\mathcal{T}_1 = \{\omega_0\}$ ,  $\mathcal{T}_4 = \{\omega_1\}$ ,  $\mathcal{T}_7 = \{\omega_2, \omega_3\}$  and  $\mathcal{T}_m = \emptyset$  for  $m = 2, 3, 5, 6$ . The coefficients  $\alpha_{\tau}$  depend solely on the sequence  $\{f_l\}_{l \in \mathbb{Z}^+}$ . As a tribute to Wilhelm Magnus, whose pioneering work (Magnus 1954) inspired our analysis, and consistently with publications that have dealt with this construct from a different point of view (cf., for example, Sussman 1986; Wilcox 1972), we henceforth call (2.6) the *Magnus series* of  $\sigma$ . Note that  $H_{\tau}(t) \in \mathfrak{g}$  for all  $\tau \in \mathcal{T}_k$ ,  $k \in \mathbb{Z}^+$ , implies  $\sigma(t) \in \mathfrak{g}$  and therefore  $\exp[\sigma(t)] \in G$ , provided that (2.6) converges.

Let

$$U_m = \text{ad} \left[ \sum_{k=1}^{\infty} \sum_{\tau \in \mathcal{T}_k} \alpha_{\tau} \int_0^t H_{\tau} \right]^m [a], \quad m \in \mathbb{Z}^+,$$

and therefore, according to (2.3),

$$\sigma' = \sum_{m=0}^{\infty} f_m U_m. \quad (2.7)$$

It follows at once from (1.6) that, for every  $m \in \mathbb{N}$ ,

$$\begin{aligned} U_m &= \left[ \sum_{k=1}^{\infty} \sum_{\tau \in \mathcal{T}_k} \alpha_{\tau} \int_0^t H_{\tau}, \text{ad} \left[ \sum_{k=1}^{\infty} \sum_{\tau \in \mathcal{T}_k} \alpha_{\tau} \int_0^t H_{\tau} \right]^{m-1} [a] \right] \\ &= \left[ \sum_{k=1}^{\infty} \sum_{\tau \in \mathcal{T}_k} \alpha_{\tau} \int_0^t H_{\tau}, U_{m-1} \right] \\ &= \sum_{k=1}^{\infty} \sum_{\tau \in \mathcal{T}_k} \alpha_{\tau} \left[ \int_0^t H_{\tau}, U_{m-1} \right]. \end{aligned} \quad (2.8)$$

Because of the second composition rule, it follows at once by induction that for every  $H_{\tau}$  there exist terms  $H_{\tau_1}, H_{\tau_2}, \dots, H_{\tau_r}$ , say, such that

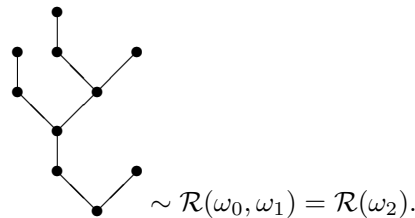
$$H_{\tau} := R(\tau_1, \tau_2, \dots, \tau_r) = \left[ \int H_{\tau_1}, \left[ \int H_{\tau_2}, \dots, \left[ \int H_{\tau_r}, a \right] \dots \right] \right]. \quad (2.9)$$

**Proposition 2.3.** *For every  $m \in \mathbb{N}$  it is true that*

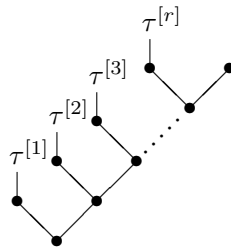
$$U_m = \sum_{k_1=1}^{\infty} \sum_{k_2=1}^{\infty} \dots \sum_{k_m=1}^{\infty} \sum_{\tau_1 \in \mathcal{T}_{k_1}} \sum_{\tau_2 \in \mathcal{T}_{k_2}} \dots \sum_{\tau_m \in \mathcal{T}_{k_m}} \alpha_{\tau_1} \alpha_{\tau_2} \dots \alpha_{\tau_m} R(\tau_1, \tau_2, \dots, \tau_m) \quad (2.10)$$

*Proof.* By induction on (2.8). ■

We denote the binary tree corresponding to  $R(\tau_1, \tau_2, \dots, \tau_r)$  by  $\mathcal{R}(\tau_1, \tau_2, \dots, \tau_r)$ . This representation is not unique, e.g.



To ensure uniqueness we choose  $\tau_1, \tau_2, \dots, \tau_r$  as follows. Every binary tree  $\tau$  that can be obtained from our composition rules can be uniquely written in the form



Then  $\tau = \mathcal{R}(\tau^{[1]}, \tau^{[2]}, \dots, \tau^{[r]})$ , and this is the representation that we adopt in the sequel. In particular, in the last example we obtain  $\mathcal{R}(\omega_2)$ .

**Proposition 2.4.** Given any  $r \in \mathbb{N}$  and  $\tau_k \in \mathcal{T}_{m_k}$ ,  $k = 1, 2, \dots, r$ , it is true that

$$\text{ord } \mathcal{R}(\tau_1, \tau_2, \dots, \tau_r) = \sum_{k=1}^r m_k + 2r + 1, \quad (2.11)$$

where  $\text{ord } \tau$  is the order (i.e. the number of vertices) of the tree  $\tau$ .

*Proof.* It follows from the definition of  $\mathcal{R}$  that

$$\text{ord } \mathcal{R}(\tau_1, \tau_2, \dots, \tau_r) = \text{ord } \mathcal{R}(\tau_1, \tau_2, \dots, \tau_{r-1}) + m_r + 2.$$

The proof of (2.11) follows from this identity by straightforward induction. ■

**Corollary 2.5.**  $\mathcal{T}_k = \emptyset$  whenever  $k \not\equiv 1 \pmod{3}$ .

*Proof.* We prove by induction that all the trees allowed by our composition rules have order  $1 \pmod{3}$ . Clearly,  $\text{ord } \omega_0 = 1$ . Assuming that  $m_k = 3l_k + 1$ ,  $k = 1, 2, \dots, r$ , we deduce from (2.11) that

$$\text{ord } \mathcal{R}(\tau_1, \tau_2, \dots, \tau_r) = 3 \left( \sum_{k=1}^r l_k + 1 \right) + 1.$$

Hence our assertion is true. ■

Substitution of (2.10) in (2.7) and comparison of terms corresponding to order- $k$  trees yields

$$\sum_{\tau \in \mathcal{T}_k} \alpha_\tau H_\tau = \sum_{l=1}^{\lfloor (k-1)/2 \rfloor} f_l \sum_{\substack{n_1, n_2, \dots, n_l \in \mathbb{N} \\ n_1 + \dots + n_l = k - 2l - 1}} \sum_{\substack{\tau_i \in \mathcal{T}_{n_i} \\ i=1, 2, \dots, l}} \alpha_{\tau_1} \alpha_{\tau_2} \cdots \alpha_{\tau_l} R(\tau_1, \tau_2, \dots, \tau_l)$$

*Phil. Trans. R. Soc. Lond. A* (1999)

for every  $k \in \mathbb{N}$ . This identity simplifies by virtue of corollary 2.5 to proposition 2.4, since we need to consider only trees of order 1 mod 3:

$$\sum_{\tau \in \mathcal{T}_{3m+1}} \alpha_{\tau} H_{\tau} = \sum_{l=1}^{\lfloor 3m/2 \rfloor} f_l \sum_{\substack{n_1, n_2, \dots, n_l \in \mathbb{Z}^+ \\ n_1 + \dots + n_l = m-l}} \sum_{\substack{\tau_i \in \mathcal{T}_{3n_i+1} \\ i=1, 2, \dots, l}} \alpha_{\tau_1} \alpha_{\tau_2} \cdots \alpha_{\tau_l} R(\tau_1, \tau_2, \dots, \tau_l). \quad (2.12)$$

Comparing the left- and right-hand sides of (2.12), we deduce that

$$\mathcal{T}_{3m+1} = \{ \mathcal{R}(\tau_1, \tau_2, \dots, \tau_l) : \tau_i \in \mathcal{T}_{3n_i+1}, i = 1, 2, \dots, l, \\ n_1 + n_2 + \dots + n_l + l = m, l = 1, 2, \dots, \lfloor \frac{1}{2}(3m) \rfloor \}. \quad (2.13)$$

Another consequence of (2.12) is that the coefficients  $\alpha_{\tau}$  can be evaluated recursively:

$$\alpha_{\omega_0} = f_0, \\ \alpha_{\mathcal{R}(\tau_1, \dots, \tau_l)} = f_l \prod_{i=1}^l \alpha_{\tau_i}. \quad (2.14)$$

Table 1 displays all the expansion terms, trees and coefficients of order up to ten.




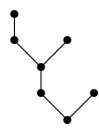
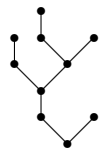
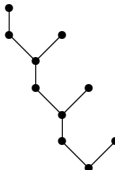
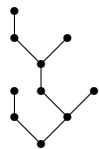
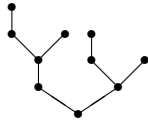
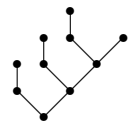
Assisted by table 1, we present the first few terms of the Magnus expansion for equation (2.3):

$$\begin{aligned} \sigma(t) = & f_0 \int_0^t a(\kappa) d\kappa + f_0 f_1 \int_0^t \int_0^{\kappa} [a(\xi), a(\kappa)] d\xi d\kappa \\ & + f_0^2 f_2 \int_0^t \int_0^{\kappa} \int_0^{\kappa} [a(\eta), [a(\xi), a(\kappa)]] d\eta d\xi d\kappa \\ & + f_0 f_1^2 \int_0^t \int_0^{\kappa} \int_0^{\xi} [[a(\eta), a(\xi)], a(\kappa)] d\eta d\xi d\kappa \\ & + f_0^2 f_1 f_2 \int_0^t \int_0^{\kappa} \int_0^{\xi} \int_0^{\xi} [[a(\rho), [a(\eta), a(\xi)]], a(\kappa)] d\rho d\eta d\xi d\kappa \\ & + f_0 f_1^3 \int_0^t \int_0^{\kappa} \int_0^{\xi} \int_0^{\eta} [[[a(\rho), a(\eta)], a(\xi)], a(\kappa)] d\rho d\eta d\xi d\kappa \\ & + f_0^2 f_1 f_2 \int_0^t \int_0^{\kappa} \int_0^{\xi} \int_0^{\kappa} [a(\rho), [[a(\eta), a(\xi)], a(\kappa)]] d\rho d\eta d\xi d\kappa \\ & + f_0^2 f_1 f_2 \int_0^t \int_0^{\kappa} \int_0^{\kappa} \int_0^{\eta} [[a(\rho), a(\eta)], [a(\xi), a(\kappa)]] d\rho d\eta d\xi d\kappa \\ & + f_0^3 f_3 \int_0^t \int_0^{\kappa} \int_0^{\kappa} \int_0^{\kappa} [a(\rho), [a(\eta), [a(\xi), a(\kappa)]]] d\rho d\eta d\xi d\kappa + \dots \quad (2.15) \end{aligned}$$

The original expansion (1.3) of Magnus (1954) is readily recovered by using the values of  $\{f_m\}$  from (2.4).

The order of magnitude of the elements in (2.4) is of interest, in particular whenever we wish to approximate  $y$  by truncating the series. Let us suppose that  $a(t) = \mathcal{O}(t^{\lambda})$ , where  $\lambda \in \mathbb{Z}^+$ . Insofar as the present discussion is concerned, the reader might just as well assume that  $\lambda = 0$ , but the more general case will be useful in §5. We define the function  $\beta_{\tau}$  as the least integer such that  $H_{\tau}(t) = \mathcal{O}(t^{\beta_{\tau}})$ , where  $H_{\tau}$  is an expansion term corresponding to the tree  $\tau$ . Therefore:

Table 1. Expansion terms  $H_\tau$  corresponding to trees  $\tau$  of order  $\leq 10$ , their representations  $\mathcal{G}$  and coefficients  $\alpha_\tau$ 

order	name	expression	tree	representation	coefficient
1	$\omega_0$	$a$		—	$f_0$
4	$\omega_1$	$[f a, a]$		$\mathcal{R}(\omega_0)$	$f_0 f_1$
7	$\omega_2$	$[f a, [f a, a]]$		$\mathcal{R}(\omega_0, \omega_0)$	$f_0^2 f_2$
	$\omega_3$	$[f [f a, a], a]$		$\mathcal{R}(\omega_1)$	$f_0 f_1^2$
10	$\omega_4$	$[f [f a, [f a, a]], a]$		$\mathcal{R}(\omega_2)$	$f_0^2 f_1 f_2$
	$\omega_5$	$[f [f [f a, a], a], a]$		$\mathcal{R}(\omega_3)$	$f_0 f_1^3$
	$\omega_6$	$[f a, [f [f a, a], a]]$		$\mathcal{R}(\omega_0, \omega_1)$	$f_0^2 f_1 f_2$
	$\omega_7$	$[f [f a, a], [f a, a]]$		$\mathcal{R}(\omega_1, \omega_0)$	$f_0^2 f_1 f_2$
	$\omega_8$	$[f a, [f a, [f a, a]]]$		$\mathcal{R}(\omega_0, \omega_0, \omega_0)$	$f_0^3 f_3$

- (1)  $\beta_{\omega_0} = \lambda$ ;  
 (2)  $\beta_\tau \geq \beta_{\tau_1} + \beta_{\tau_2} + 1$ , where

$$\tau = \begin{array}{c} \tau_1 \\ | \\ \bullet \\ \diagdown \quad \diagup \\ \bullet \end{array} \quad ; \quad (2.16)$$

- (3)  $\beta_\tau \geq 2\beta_{\tau_1} + 2$ , where

$$\tau = \begin{array}{c} \tau_1 \\ | \\ \bullet \\ \diagdown \quad \diagup \\ \bullet \end{array} \quad . \quad (2.17)$$

In other words,  $\tau_1 = \tau_2$  in (2.16) results in a larger exponent.

While (1) and (2) are trivial, the proof of (3) follows at once by letting

$$H_\tau = pt^{\beta_\tau} + \mathcal{O}(t^{\beta_\tau+1}),$$

whereby

$$\left[ \int H_\tau, H_\tau \right] = \left[ pt^{\beta_\tau} + \mathcal{O}(t^{\beta_\tau+1}), \frac{1}{\beta_\tau + 1} pt^{\beta_\tau+1} + \mathcal{O}(t^{\beta_\tau+2}) \right] = \mathcal{O}(t^{2\beta_\tau+2}).$$

**Proposition 2.6.** For every  $\tau \in \mathcal{T}_{3k+1}$ ,  $k \in \mathbb{N}$ , it is true that

$$\beta_\tau \geq (k+1)(\lambda+1). \quad (2.18)$$

*Proof.* By induction. We commence by observing that, according to (3),  $\beta_{\omega_1} \geq 2(\lambda+1)$ , consistently with (2.18). Next, suppose that  $\tau \in \mathcal{T}_{3k+1}$ ,  $k \geq 2$ , is derived from  $\tau_1$  and  $\tau_2$  in line with (2.16). Then  $\tau_1 \in \mathcal{T}_{3l+1}$ ,  $\tau_2 \in \mathcal{T}_{3(k-l-1)+1}$  for some  $l \in \{0, 1, \dots, k-1\}$ . Note that it is impossible for both  $\tau_1 = \omega_0$  and  $\tau_2 = \omega_0$ .

If  $\tau_2 = \omega_0$ , then  $l = k-1$  and, by induction,  $\beta_{\tau_1} \geq (l+1)(\lambda+1)$ . Therefore, according to (1) and (2),

$$\beta_\tau \geq \lambda + k(\lambda+1) + 1 = (k+1)(\lambda+1).$$

An identical argument affirms (2.18) when  $\tau_1 = \omega_0$ . Finally, if  $\tau_1, \tau_2 \neq \omega_0$ , then  $l \in \{1, 2, \dots, k-2\}$ . Hence, by induction and (2),

$$\left. \begin{array}{l} \beta_{\tau_1} \geq (l+1)(\lambda+1), \\ \beta_{\tau_2} \geq (k-l)(\lambda+1), \end{array} \right\} \Rightarrow \beta_\tau \geq (k+1)(\lambda+1) + 1,$$

actually in excess of the lower bound (2.18). ■

It is often possible to obtain better estimates than (2.18). Thus, for example,  $\beta_{\omega_7} \geq 4\lambda + 6$ , because of (3), while  $\tau_1 = \omega_1$ ,  $\tau_2 = \omega_3$ , composed with (2.16), yields  $k = 4$  and  $\beta_\tau \geq 5\lambda + 6$ . However, the lower bound (2.18) is the best possible insofar as the totality of all the non-trivial trees is concerned.

The following theorem, critical to truncating Magnus series as a means to approximate the solution of (1.1), follows at once from our analysis.

**Theorem 2.7.** Let  $a(t) = \mathcal{O}(t^\lambda)$ ,  $p \in \mathbb{N}$  and set

$$\sigma_p(t) := \sum_{k=0}^{p-1} \sum_{\tau \in \mathcal{T}_{3k+1}} \alpha_\tau \int_0^t H_\tau(\kappa) d\kappa. \quad (2.19)$$

Then  $\sigma_p(t) = \sigma(t) + \mathcal{O}(t^{(p+1)(\lambda+1)+1})$ ; therefore  $\exp \sigma_p(t) = y(t) + \mathcal{O}(t^{(p+1)(\lambda+1)+1})$ .

Needless to say, as things stand, (2.15) is a purely formal construct, since we have not proved yet that a Magnus expansion converges. This is our next task.

Let us assume that  $\mathfrak{g}$  is equipped with a norm  $\|\cdot\|$  (as long as  $G$  is a matrix Lie group,  $\|\cdot\|$  might be taken, for example, as the standard matrix Euclidean norm) and that there exist  $\rho, t^* > 0$  such that

$$\|a(t)\| \leq \rho t^\lambda, \quad 0 \leq t \leq t^*.$$

**Proposition 2.8.** For every  $\tau \in \mathcal{T}_{3k+1}$  it is true that

$$\|H_\tau(t)\| \leq \rho t^\lambda \left( \frac{2\rho t^{\lambda+1}}{\lambda+1} \right)^k. \quad (2.20)$$

*Proof.* By induction. The bound (2.20) is certainly true for  $k = 0$ . Suppose that  $k \in \mathbb{N}$  and that  $\tau$  has been obtained from  $\tau_1 \in \mathcal{T}_{3l+1}$  and  $\tau_2 \in \mathcal{T}_{3(k-l-1)+1}$  by means of the composition rule (2.16). Then, by the induction assumption,

$$\begin{aligned} \|H_\tau(t)\| &= \left\| \int_0^t H_{\tau_2}(\kappa) d\kappa, H_{\tau_1}(t) \right\| \leq 2 \|H_{\tau_1}(t)\| \int_0^t \|H_{\tau_2}(\kappa)\| d\kappa \\ &\leq 2\rho t^\lambda \left( \frac{2\rho t^{\lambda+1}}{\lambda+1} \right)^l \frac{\rho t^{\lambda+1}}{(k-l)(\lambda+1)} \left( \frac{2\rho t^{\lambda+1}}{\lambda+1} \right)^{k-l-1} \\ &\leq \rho t^\lambda \left( \frac{2\rho t^{\lambda+1}}{\lambda+1} \right)^k, \end{aligned}$$

because  $l \in \{0, 1, \dots, k-1\}$ . ■

Substituting (2.20) into the Magnus series, we deduce from the triangle inequality that

$$\begin{aligned} \|\sigma(t)\| &\leq \frac{1}{2} \sum_{k=0}^{\infty} \frac{\theta_k}{k+1} \left( \frac{2\rho t^{\lambda+1}}{\lambda+1} \right)^{k+1} \\ &= \frac{1}{2} \int_0^{2\rho t^{\lambda+1}/(\lambda+1)} \Theta(z) dz, \quad 0 \leq t \leq t^*, \end{aligned} \quad (2.21)$$

where  $\theta_k$  is the number of terms in the set  $\mathcal{T}_{3k+1}$ ,  $k \in \mathbb{Z}^+$  and  $\Theta$  is the generating function of the sequence  $\{\theta_k\}_{k \in \mathbb{Z}^+}$ :

$$\Theta(z) = \sum_{k=0}^{\infty} \theta_k z^k.$$

Counting the members of  $\mathcal{T}_{3k+1}$  is a trivial exercise in graph theory. Yet, we include it here for the sake of completeness. Every element of  $\mathcal{T}_{3k+1}$  is obtained uniquely from (2.16), where  $\tau_1 \in \mathcal{T}_{3l+1}$  and  $\tau_2 \in \mathcal{T}_{3(k-l-1)+1}$  for some  $l \in \{0, 1, \dots, k-1\}$ . Likewise, for every such  $l$ ,  $\tau_1$  and  $\tau_2$ , the composition (2.16) produces an element of  $\mathcal{T}_{3k+1}$ . Therefore,

$$\theta_k = \sum_{l=0}^{k-1} \theta_l \theta_{k-l-1}, \quad k \in \mathbb{N}.$$

Since  $\theta_0 = 1$ , we deduce that

$$\Theta(z) = \sum_{k=0}^{\infty} \theta_k z^k = 1 + \sum_{k=1}^{\infty} \sum_{l=0}^{k-1} \theta_l \theta_{k-l-1} z^k = 1 + z[\Theta(z)]^2.$$

Therefore  $\Theta(z)$  is a solution of the quadratic equation

$$z[\Theta(z)]^2 - \Theta(z) + 1 = 0$$

and analyticity at the origin implies that

$$\Theta(z) = \frac{1 - \sqrt{1 - 4z}}{2z} = \sum_{k=0}^{\infty} \frac{(2k)!}{k!(k+1)!} z^k. \quad (2.22)$$

We deduce not just the explicit value of  $\theta_k$ ,  $k \in \mathbb{Z}^+$ , but also that  $\Theta$  is analytic for all  $|z| \leq \frac{1}{4}$ .

**Theorem 2.9.** *The Magnus series converges for all*

$$0 \leq t \leq \min \left\{ t^*, \left( \frac{\lambda + 1}{8\rho} \right)^{1/(\lambda+1)} \right\}.$$

Moreover, within this range of  $t$ , the solution of (1.1) obeys the inequality

$$\|y(t)\| \leq \|y_0\| \exp \left[ \frac{1}{2} \int_0^{8\rho t^{\lambda+1}/(\lambda+1)} \frac{1 - \sqrt{1 - z}}{z} dz \right].$$

*Proof.* This follows at once from our analysis and the upper bound (2.21). ■

### 3. Numerical quadrature of multivariate integrals

The first step in the implementation of a Magnus expansion as a numerical procedure is a truncation of the series. In line with theorem 2.7, if all terms corresponding to  $\mathcal{T}_k$ ,  $k \geq p$  are truncated, the outcome is a  $p$ th-order method (we substitute  $\lambda = 0$  in (2.19)). Next, all the integrals are replaced by quadratures of a sufficiently high order and, finally, we need to evaluate an exponential of an element in the Lie algebra  $\mathfrak{g}$ .

On the face of it, the most computationally expensive step is the replacement of integrals by quadrature formulae†. Multivariate numerical integration is typically very expensive (Cools 1997) and we need to approximate a significant number of integrals, e.g. four integrals for order  $p = 4$  and nine for order  $p = 5$ . Naively, each integral might be approximated by Gaussian quadrature optimized with respect to the (multivariate) domain of integration; for example,

$$\int_0^h a(\kappa) d\kappa = h \sum_{j=1}^{\nu_1} b_j a(c_j h),$$

$$\int_0^h \int_0^{\kappa} [a(\xi), a(\kappa)] d\xi d\kappa = h^2 \sum_{j=1}^{\nu_2} \tilde{b}_j [a(\tilde{c}_{j,1} h), a(\tilde{c}_{j,2} h)],$$

where  $\{c_j : j = 1, 2, \dots, \nu_1\}$  are Gaussian points in  $[0, 1]$ ,  $\{\tilde{c}_{j,1}, \tilde{c}_{j,2} : j = 1, 2, \dots, \nu_2\}$  are Gaussian points in the triangle with vertices at  $\{(0, 0), (1, 0), (0, 1)\}$ , while  $b_j$  and

† We do not address ourselves in this paper to the evaluation of the exponential function.

$\tilde{b}_j$  are appropriate weights. In that case, stipulating that both quadratures are of the same order of accuracy requires  $\nu_2 = \mathcal{O}(\nu_1^2)$ . Further integrals, across polyhedra of increasingly higher dimension, require even more function evaluations. Fortunately, the naive approach is completely unnecessary and, as we demonstrate in this section, the evaluation of *all* the necessary quadratures to order  $p$  can be accomplished in just  $\lfloor \frac{1}{2}(p+1) \rfloor$  function evaluations.

Given  $\tau \in \mathcal{T}_{3m+1}$ , we can write the integral of  $H_\tau$  in the form

$$\begin{aligned} I(h) &:= \int_0^h H_\tau(t) dt \\ &= \int_0^h \int_0^{\kappa_{i_0}} \int_0^{\kappa_{i_1}} \cdots \int_0^{\kappa_{i_{m-1}}} \mathcal{L}(a(\kappa_0), a(\kappa_1), \dots, a(\kappa_m)) d\kappa_m d\kappa_{m-1} \cdots d\kappa_0, \end{aligned} \quad (3.1)$$

where

$$i_j \in \{0, 1, \dots, j\}, \quad j = 0, 1, \dots, m-1$$

and

$$\mathcal{L} : \overbrace{\mathfrak{g} \times \mathfrak{g} \times \cdots \times \mathfrak{g}}^{m+1 \text{ times}} \rightarrow \mathfrak{g}$$

is a multilinear form. For example, (2.15) demonstrates that

$$\begin{aligned} \tau = \omega_0 : \quad \mathcal{L}(x_0) &= x_0, \\ \tau = \omega_1 : \quad \mathcal{L}(x_0, x_1) &= [x_1, x_0], \quad i_0 = 0, \\ \tau = \omega_2 : \quad \mathcal{L}(x_0, x_1, x_2) &= [x_2, [x_1, x_0]], \quad i_0 = 0, \quad i_1 = 0, \\ \tau = \omega_3 : \quad \mathcal{L}(x_0, x_1, x_2) &= [[x_2, x_1], x_0], \quad i_0 = 0, \quad i_1 = 1. \end{aligned}$$

Let  $c_1, c_2, \dots, c_\nu$  be  $\nu$  distinct points in the interval  $[0, 1]$ , where  $\nu \in \mathbb{N}$  is given. We wish to approximate

$$\int_0^h H_\tau(t) dt$$

by the quadrature formula

$$I(h) \approx Q(h) := h^{m+1} \sum_{\mathbf{l} \in \Lambda_m^\nu} b_{\mathbf{l}} \mathcal{L}(a(c_{l_0} h), a(c_{l_1} h), \dots, a(c_{l_m} h)), \quad (3.2)$$

where  $\Lambda_m^\nu$  is the set of all the combinations  $\mathbf{l} = (l_0, l_1, \dots, l_m)$  of  $(m+1)$ -tuples of integers from the set  $\{1, 2, \dots, \nu\}$  and the weights  $b_{\mathbf{l}}$  are given. Note that the implementation of (3.2) requires the evaluation of the function  $a$  at just  $\nu$  points. Moreover, as long as we can use the same nodes  $\mathbf{c} = (c_1, c_2, \dots, c_\nu)$  for *all* the integrals, the entire evaluation of  $\sigma_p$  requires just  $\nu$  function evaluations!

Let  $u_j \in \mathbb{P}_{\nu-1}$  be the  $j$ th cardinal polynomial of Lagrange interpolation at the points  $c_1, c_2, \dots, c_\nu$ ,

$$u_j(t) = \prod_{\substack{i=1 \\ i \neq j}}^{\nu} \frac{t - c_i}{c_j - c_i}, \quad j = 1, 2, \dots, \nu.$$



We approximate  $a$  by its *Lagrange interpolation polynomial*

$$a(t) \approx \hat{a}(t) = \sum_{j=1}^{\nu} a(c_j h) u_j \left( \frac{t}{h} \right),$$

hence  $\hat{a}(c_k h) = a(c_k h)$ ,  $k = 1, 2, \dots, \nu$ , and define the weights  $b_l$  through the explicit formula

$$b_l = \int_0^1 \int_0^{\kappa_{i_0}} \int_0^{\kappa_{i_1}} \cdots \int_0^{\kappa_{i_{m-1}}} \prod_{i=1}^{\nu} u_{l_i}(\kappa_i) d\kappa_m d\kappa_{m-1} \cdots d\kappa_0, \quad l \in \Lambda_m^{\nu}. \quad (3.3)$$

Because of the multilinearity of  $\mathcal{L}$ , we have

$$\begin{aligned} & \mathcal{L}(\hat{a}(\kappa_0), \hat{a}(\kappa_1), \dots, \hat{a}(\kappa_m)) \\ &= \mathcal{L} \left( \sum_{l_0=1}^{\nu} u_{l_0} \left( \frac{\kappa_0}{h} \right) a(c_{l_0} h), \sum_{l_1=1}^{\nu} u_{l_1} \left( \frac{\kappa_1}{h} \right) a(c_{l_1} h), \dots, \sum_{l_m=1}^{\nu} u_{l_m} \left( \frac{\kappa_m}{h} \right) a(c_{l_m} h) \right) \\ &= \sum_{l_0=1}^{\nu} \sum_{l_1=1}^{\nu} \cdots \sum_{l_m=1}^{\nu} \prod_{i=0}^m u_{l_i} \left( \frac{\kappa_i}{h} \right) \mathcal{L}(a(c_{l_0} h), a(c_{l_1} h), \dots, a(c_{l_m} h)). \end{aligned}$$

Therefore, with the above definition of  $b_l$ ,

$$Q(h) = \int_0^h \int_0^{\kappa_{i_0}} \int_0^{\kappa_{i_1}} \cdots \int_0^{\kappa_{i_{m-1}}} \mathcal{L}(\hat{a}(\kappa_0), \hat{a}(\kappa_1), \dots, \hat{a}(\kappa_m)) d\kappa_m d\kappa_{m-1} \cdots d\kappa_0. \quad (3.4)$$

The identity (3.4) provides both the *raison d'être* for the definition (3.3) and a convenient means for an order analysis of the quadrature formula  $Q(h)$ .

Let

$$\mathcal{S} = \mathcal{S}_i := \{(\kappa_0, \kappa_1, \dots, \kappa_m) \in \mathbb{R}^{m+1} : 0 \leq \kappa_j \leq \kappa_{i_{j-1}}, j = 1, 2, \dots, m\},$$

so that  $h\mathcal{S}$  is the domain of integration in (3.1), and

$$d\boldsymbol{\kappa} = d\kappa_m d\kappa_{m-1} \cdots d\kappa_0,$$

therefore

$$\begin{aligned} I(h) &= \int_{h\mathcal{S}} \mathcal{L}(a(\kappa_0), a(\kappa_1), \dots, a(\kappa_m)) d\boldsymbol{\kappa}, \\ Q(h) &= \int_{h\mathcal{S}} \mathcal{L}(\hat{a}(\kappa_0), \hat{a}(\kappa_1), \dots, \hat{a}(\kappa_m)) d\boldsymbol{\kappa}. \end{aligned}$$

Given that

$$a(t) = \sum_{k=0}^{\infty} a_k t^{k+\lambda},$$

where  $\lambda \in \mathbb{Z}^+$ , and letting

$$\mathcal{L}_{\mathbf{k}} := \mathcal{L}(a_{k_0}, a_{k_1}, \dots, a_{k_m}),$$

we thus deduce that

$$\begin{aligned} I(h) &= \int_{hS} \sum_{k_0=0}^{\infty} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \mathcal{L}_{\mathbf{k}} \kappa_0^{k_0+\lambda} \kappa_1^{k_1+\lambda} \cdots \kappa_m^{k_m+\lambda} d\mathbf{k} \\ &= h^{(m+1)(\lambda+1)} \sum_{k_0=0}^{\infty} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h^{k_0+k_1+\cdots+k_m} \mathcal{L}_{\mathbf{k}} \int_S \kappa_0^{k_0+\lambda} \kappa_1^{k_1+\lambda} \cdots \kappa_m^{k_m+\lambda} d\mathbf{k} \\ &= h^{(m+1)(\lambda+1)} \sum_{k_0=0}^{\infty} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h^{k_0+k_1+\cdots+k_m} \gamma_{\mathbf{k}} \mathcal{L}_{\mathbf{k}}, \end{aligned}$$

where

$$\gamma_{\mathbf{k}} = \int_S \kappa_0^{k_0+\lambda} \kappa_1^{k_1+\lambda} \cdots \kappa_m^{k_m+\lambda} d\mathbf{k}, \quad \mathbf{k} \in (\mathbb{Z}^+)^{m+1}. \quad (3.5)$$

Likewise, it follows from (3.2) that

$$\begin{aligned} Q(h) &= h^{m+1} \sum_{\mathbf{l} \in \Lambda_m^\nu} b_{\mathbf{l}} \sum_{k_0=0}^{\infty} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} \mathcal{L}_{\mathbf{k}} c_{l_0}^{k_0+\lambda} c_{l_1}^{k_1+\lambda} \cdots c_{l_m}^{k_m+\lambda} h^{k_0+k_1+\cdots+k_m+(m+1)\lambda} \\ &= h^{(m+1)(\lambda+1)} \sum_{k_0=0}^{\infty} \sum_{k_1=0}^{\infty} \cdots \sum_{k_m=0}^{\infty} h^{k_0+k_1+\cdots+k_m} \delta_{\mathbf{k}} \mathcal{L}_{\mathbf{k}}, \end{aligned}$$

where

$$\delta_{\mathbf{k}} = \sum_{\mathbf{l} \in \Lambda_m^\nu} b_{\mathbf{l}} c_{l_0}^{k_0+\lambda} c_{l_1}^{k_1+\lambda} \cdots c_{l_m}^{k_m+\lambda}, \quad \mathbf{k} \in (\mathbb{Z}^+)^{m+1}. \quad (3.6)$$

Comparing (3.5) with (3.6), we deduce the following order condition for the quadrature formula.

**Proposition 3.1.** *The quadrature method (3.2) is of order  $p \in \mathbb{N}$ , i.e.  $Q(h) = I(h) + \mathcal{O}(h^{p+1})$  for all sufficiently smooth functions  $a$ , if and only if*

$$\delta_{\mathbf{k}} = \gamma_{\mathbf{k}}, \quad \mathbf{k} \in (\mathbb{Z}^+)^{m+1}, \quad k_0 + k_1 + \cdots + k_m \leq p - (m+1)(\lambda+1). \quad (3.7)$$

Letting

$$p_j(t) := \sum_{i=1}^{\nu} u_i(t) c_i^j, \quad j \in \mathbb{Z}^+,$$

we can simplify (3.6) by substituting the definition (3.3) of  $b_{\mathbf{l}}$ . The outcome is

$$\delta_{\mathbf{k}} = \int_S p_{k_0+\lambda}(\kappa_0) p_{k_1+\lambda}(\kappa_1) \cdots p_{k_m+\lambda}(\kappa_m) d\mathbf{k}, \quad \mathbf{k} \in (\mathbb{Z}^+)^{m+1}. \quad (3.8)$$

It follows at once from the properties of Lagrange cardinal polynomials,  $u_i$ , that

$$p_j(t) = t^j + \omega(t) e_j(t), \quad j \in \mathbb{Z}^+, \quad (3.9)$$

where

$$\omega(t) := \prod_{k=1}^{\nu} (t - c_k)$$

and  $e_j \in \mathbb{P}_{j-\nu}$ , the set of polynomials of degree  $j - \nu$ . (We let  $\mathbb{P}_{j-\nu} = \{0\}$  for  $j < \nu$ .) Substituting (3.9) into (3.8), we obtain

$$\delta_{\mathbf{k}} = \int_S \prod_{j=0}^m [\kappa_j^{k_j+\lambda} + \omega(\kappa_j) e_{k_j+\lambda}(\kappa_j)] d\mathbf{\kappa}. \quad (3.10)$$

Altogether, we have  $2^{m+1}$  terms in the integrand. One of these, namely

$$\kappa_0^{k_0+\lambda} \kappa_1^{k_1+\lambda} \dots \kappa_m^{k_m+\lambda},$$

is precisely the integrand in (3.5). We wish to explore the conditions that ensure that all the remaining  $2^{m+1} - 1$  terms vanish.

Given such a term,  $F(\kappa_0, \kappa_1, \dots, \kappa_m)$ , say, there exists an index  $r \in \{0, 1, \dots, m\}$  such that

$$F(\kappa_0, \kappa_1, \dots, \kappa_m) = \kappa_0^{k_0+\lambda} \kappa_1^{k_1+\lambda} \dots \kappa_{r-1}^{k_{r-1}+\lambda} \omega(\kappa_r) e_{k_r+\lambda}(\kappa_r) \dots$$

Suppose that  $r \geq 1$ . In that case we execute  $(r - 1)$  exchanges of the order of integration so that  $\kappa_r$  is the last to be integrated. We commence by exchanging the order of  $\kappa_r$  and  $\kappa_{r-1}$ . The outcome is

$$\int_0^1 \int_0^{\kappa_{i_0}} \dots \int_0^{\kappa_{i_{r-3}}} \int_0^{\kappa_{i_{r-2}}} \int_{\psi_r}^{\chi_r} \int_0^{\kappa_{i_r}} \dots \int_0^{\kappa_{i_{m-1}}} F(\kappa_0, \kappa_1, \dots, \kappa_m) \\ \times d\kappa_m \dots d\kappa_{r+1} d\kappa_{r-1} d\kappa_r d\kappa_{r-2} \dots d\kappa_0,$$

where

$$(\psi_r, \chi_r) = \begin{cases} (\kappa_r, \kappa_{i_{r-2}}), & i_{r-1} = r - 1, \\ (0, \kappa_{i_{r-1}}), & i_{r-1} \in \{0, 1, \dots, r - 2\}. \end{cases}$$

Continuing in this manner, we arrive at

$$\int_0^1 \int_{\psi_1}^{\chi_1} \int_{\psi_2}^{\chi_2} \dots \int_{\psi_m}^{\chi_m} F(\kappa_0, \kappa_1, \dots, \kappa_m) d\kappa_m \dots d\kappa_{r+1} d\kappa_{r-1} \dots d\kappa_0 d\kappa_r,$$

where

$$\psi_j, \chi_j \in \begin{cases} \{0, \kappa_r, \kappa_0, \kappa_1, \dots, \kappa_{j-2}\}, & j = 1, 2, \dots, r \\ \{0, \kappa_0, \kappa_1, \dots, \kappa_{j-1}, 1\}, & j = r + 1, \dots, m. \end{cases}$$

Relabelling the variables, we deduce that

$$\int_S F(\kappa_0, \kappa_1, \dots, \kappa_m) d\mathbf{\kappa} = \int_0^1 \omega(\kappa_0) e_{k_r+\lambda}(\kappa_0) \int_{\mathcal{A}} K(\kappa_1, \kappa_2, \dots, \kappa_m) d\mathbf{\kappa}, \quad (3.11)$$

where

$$\mathcal{A} = \{(\kappa_1, \kappa_2, \dots, \kappa_m) \in \mathbb{R}^m : \psi_j \leq \kappa_j \leq \chi_j, j = 1, 2, \dots, m\}, \\ \left. \begin{array}{l} \psi_j \in \{0, \kappa_0, \kappa_1, \dots, \kappa_{j-1}\}, \\ \chi_j \in \{\kappa_0, \kappa_1, \dots, \kappa_{j-1}, 1\}, \end{array} \right\} \quad j = 1, 2, \dots, m,$$

and

$$K(\kappa_1, \kappa_2, \dots, \kappa_m) = \prod_{j=1}^m \tilde{p}_j(\kappa_j).$$

Each  $\tilde{p}_j(t)$  in the above definition is either  $\omega(t)e_{l_j+\lambda}(t)$  or  $t^{l_j+\lambda}$ , where

$$l_j = \begin{cases} k_{j-1}, & j = 1, 2, \dots, r, \\ k_j, & j = r + 1, \dots, m. \end{cases}$$

For example,

$$\begin{aligned} & \int_0^1 \int_0^{\kappa_0} \int_0^{\kappa_1} \int_0^{\kappa_2} \kappa_0^{k_0+\lambda} \kappa_1^{k_1+\lambda} \kappa_2^{k_2+\lambda} \omega(\kappa_3) e_{k_3+\lambda}(\kappa_3) d\kappa_3 d\kappa_2 d\kappa_1 d\kappa_0 \\ &= \int_0^1 \omega(\kappa_0) e_{k_3+\lambda}(\kappa_0) \int_{\kappa_0}^1 \int_{\kappa_0}^{\kappa_1} \int_0^{\kappa_2} \kappa_1^{k_0+\lambda} \kappa_2^{k_1+\lambda} \kappa_3^{k_2+\lambda} d\kappa_3 d\kappa_2 d\kappa_1 d\kappa_0, \end{aligned}$$

and hence

$$(\psi_1, \chi_1) = (\kappa_0, 1), \quad (\psi_2, \chi_2) = (\kappa_0, \kappa_1), \quad (\psi_3, \chi_3) = (0, \kappa_2)$$

and

$$\tilde{p}_j(t) = t^{k_j+\lambda}, \quad j = 1, 2, 3.$$

Each  $\tilde{p}_j$  is a polynomial of degree  $l_j + \lambda$ . Therefore, and bearing in mind the definition of  $(\psi_i, \chi_i)$ ,  $i = 1, 2, \dots, m$ , it follows at once by induction that

$$\begin{aligned} \tilde{p}_m(\kappa_m) &\in \mathbb{P}_{l_m+\lambda}[\kappa_m] \\ &\Rightarrow \int_{\psi_m}^{\chi_m} \tilde{p}_m(\kappa_m) d\kappa_m \in \mathbb{P}_{l_m+\lambda_1}[\kappa_0, \dots, \kappa_{m-1}] \\ &\Rightarrow \int_{\psi_{m-1}}^{\chi_{m-1}} \int_{\psi_m}^{\chi_m} \tilde{p}_{m-1}(\kappa_{m-1}) \tilde{p}_m(\kappa_m) d\kappa_m d\kappa_{m-1} \in \mathbb{P}_{l_{m-1}+l_m+2(\lambda+1)}[\kappa_0, \dots, \kappa_{m-2}] \\ &\Rightarrow \dots \\ &\Rightarrow \int_{\psi_1}^{\chi_1} \dots \int_{\psi_m}^{\chi_m} \tilde{p}_1(\kappa_1) \dots \tilde{p}_m(\kappa_m) d\kappa_m \dots d\kappa_1 \in \mathbb{P}_{l_1+\dots+l_m+m(\lambda+1)}[\kappa_0] \\ &\Rightarrow \int_{\mathcal{A}} K(\kappa_1, \dots, \kappa_m) d\kappa_m \dots d\kappa_1 \in \mathbb{P}_{l_1+\dots+l_m+m(\lambda+1)}[\kappa_0]. \end{aligned}$$

Since  $l_1 + \dots + l_m = k_0 + \dots + k_{r-1} + k_{r+1} + \dots + k_m$  and  $e_{k_r+\lambda} \in \mathbb{P}_{k_r+\lambda-\nu}$ , it follows from (3.11) that

$$\int_S F(\kappa_0, \kappa_1, \dots, \kappa_m) d\kappa = \int_0^1 \omega(\kappa) q(\kappa) d\kappa, \quad (3.12)$$

where

$$q \in \mathbb{P}_{k_0+k_1+\dots+k_m+(m+1)(\lambda+1)-\nu-1}.$$

**Theorem 3.2.** Let us suppose that the polynomial  $\omega$  obeys the orthogonality conditions

$$\int_0^1 \omega(\kappa) \kappa^{j-1} d\kappa = 0, \quad j = 1, 2, \dots, r. \quad (3.13)$$

Then  $Q(h)$  is an order- $(\nu + r)$  approximation to the multivariate integral  $I(h)$ .

*Proof.* Subject to (3.13), it follows from (3.12) that

$$k_0 + k_1 + \cdots + k_m \leq r + \nu - (m + 1)(\lambda + 1)$$

implies

$$\int_S F(\kappa_0, \kappa_1, \dots, \kappa_m) d\boldsymbol{\kappa} = 0.$$

This is true with regard to all the  $2^{m+1} - 1$  terms in the integrand. In other words, just one term contributes to the integral (3.10) and we deduce that  $\gamma_{\mathbf{k}} = \delta_{\mathbf{k}}$ . The theorem now follows from the order condition (3.7). ■

**Corollary 3.3.** *Let  $(c_1, c_2, \dots, c_\nu)$  be the nodes of Gaussian quadrature in  $[0, 1]$ . Then  $Q(h)$  is a quadrature of order  $2\nu$ .*

To rephrase theorem 3.2 and corollary 3.3, the underlying order conditions are nothing else but the usual orthogonality conditions that guarantee the order of a univariate quadrature. All the necessary integrals can thus be evaluated to order  $p$  by evaluating the function  $a$  at just  $\lfloor \frac{1}{2}(p+1) \rfloor$  Gauss–Legendre points in  $[0, 1]$  or, for that matter, at  $\lfloor \frac{1}{2}p \rfloor + 1$  Gauss–Radau points there.

As a trivial example, we choose  $\lambda = 0$ ,  $\nu = 2$  and the Gauss–Legendre points

$$c_1 = \frac{1}{2} - \frac{1}{6}\sqrt{3}, \quad c_2 = \frac{1}{2} + \frac{1}{6}\sqrt{3}.$$

Therefore,

$$u_1(t) = \frac{1}{2} - (t - \frac{1}{2})\sqrt{3}, \quad u_2(t) = \frac{1}{2} + (t - \frac{1}{2})\sqrt{3}.$$

Brief calculation affirms that the fourth-order quadrature is

$$\int_0^h \int_0^\kappa \mathcal{L}(a(\kappa), a(\xi)) d\xi d\kappa \approx h^2 \left[ \frac{1}{8} \mathcal{L}_{1,1} + \left( \frac{1}{8} - \frac{1}{12}\sqrt{3} \right) \mathcal{L}_{1,2} + \left( \frac{1}{8} + \frac{1}{12}\sqrt{3} \right) \mathcal{L}_{2,1} + \frac{1}{8} \mathcal{L}_{2,2} \right].$$

In the special case when

$$\mathcal{L}(x, y) = [y, x],$$

which is of concern in this paper, obvious cancellations yield

$$\int_0^h \int_0^\kappa [a(\xi), a(\kappa)] d\xi d\kappa \approx \frac{1}{6}\sqrt{3}h^2 [a(c_1h), a(c_2h)].$$

Likewise,

$$\begin{aligned} \int_0^h H_{\omega_2}(\kappa) d\kappa &= \int_0^h \left[ \int_0^\kappa a(\rho) d\rho, \left[ \int_0^\kappa a(\xi) d\xi, a(\kappa) \right] \right] d\kappa \\ &\approx h^3 \left\{ -\left( \frac{3}{80} + \frac{1}{16}\sqrt{3} \right) [[a(c_1h), a(c_2h)], a(c_1h)] \right. \\ &\quad \left. + \left( \frac{3}{80} - \frac{1}{16}\sqrt{3} \right) [[a(c_1h), a(c_2h)], a(c_2h)] \right\} \\ &= -h^3 [[a(c_1h), a(c_2h)], \left( \frac{3}{80} + \frac{1}{16}\sqrt{3} \right) a(c_1h) - \left( \frac{3}{80} - \frac{1}{16}\sqrt{3} \right) a(c_2h)] \end{aligned}$$

and

$$\begin{aligned} \int_0^h H_{\omega_3}(\kappa) d\kappa &= \int_0^h \left[ \int_0^\kappa \left[ \int_0^\xi a(\rho) d\rho, a(\xi) \right] d\xi, a(\kappa) \right] d\kappa \\ &\approx h^3 \left\{ \left( \frac{3}{80} - \frac{1}{48}\sqrt{3} \right) [a(c_1h), [a(c_1h), a(c_2h)]] \right. \\ &\quad \left. - \left( \frac{3}{80} + \frac{1}{48}\sqrt{3} \right) [a(c_2h), [a(c_1h), a(c_2h)]] \right\} \\ &= h^3 \left[ \left( \frac{3}{80} - \frac{1}{48}\sqrt{3} \right) a(c_1h) - \left( \frac{3}{80} + \frac{1}{48}\sqrt{3} \right) a(c_2h), [a(c_1h), a(c_2h)] \right]. \end{aligned}$$

Recall, however, from (2.15) that  $\int H_{\omega_2}$  and  $\int H_{\omega_3}$  need to be evaluated as part of a linear combination. Taking this into account, we obtain

$$f_0^2 f_2 \int_0^h H_{\omega_2}(\kappa) d\kappa + f_0 f_1^2 \int_0^h H_{\omega_3}(\kappa) d\kappa \approx h^3 [[a(c_1 h), a(c_2 h)], b_1 a(c_1 h) - b_2 a(c_2 h)],$$

where

$$\begin{aligned} b_1 &= \left(\frac{3}{80} + \frac{1}{16}\sqrt{3}\right) f_0^2 f_2 + \left(\frac{3}{80} - \frac{1}{48}\sqrt{3}\right) f_0 f_1^2, \\ b_2 &= \left(\frac{3}{80} - \frac{1}{16}\sqrt{3}\right) f_0^2 f_2 + \left(\frac{3}{80} + \frac{1}{48}\sqrt{3}\right) f_0 f_1^2. \end{aligned}$$

In particular, if  $f_0 = 1$ ,  $f_1 = \frac{1}{2}$  and  $f_2 = \frac{1}{12}$ , in line with (2.4), then  $b_1 = b_2 = \frac{1}{80}$  and

$$f_0^2 f_2 \int_0^h H_{\omega_2}(\kappa) d\kappa + f_0 f_1^2 \int_0^h H_{\omega_3}(\kappa) d\kappa \approx \frac{1}{80} h^3 [[a(c_1 h), a(c_2 h)], a(c_1 h) - a(c_2 h)].$$

We thus deduce from our analysis and theorem 2.7 that a fourth-order truncation of the Magnus series requires just two evaluations of  $a$  and the computation of only two commutators.

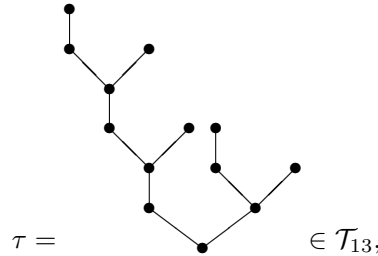
#### 4. Reducing the number of commutators

A careful reader should have observed in the last section that there is a very large scope for reducing the number of commutators that need to be evaluated to attain order 4. The number of function evaluations being determined solely by the order of the underlying method, the computation of commutators is a major expense of our algorithms, hence it makes sense to discuss in a more orderly fashion how to exploit the Lie-algebraic setting to reduce the number of commutators in a general case.

Another pertinent observation from the last section is that, although a definition of the weights by the formula (3.8) makes it amenable to successful order analysis, it is not of a form suitable to a (possibly symbolic) manipulation. Recall that the terms in a Magnus expansion are indexed by binary trees and are derived naturally in terms of the composition rules from § 2. We wish to redefine the weights  $b_l$  in a similar fashion. It is a tribute to the power of graph theory to illuminate the matter at hand that both goals—reduction in the number of commutators and a recursive formula for quadrature weights—can be accomplished by the same mechanism. This is the theme of the present section. Our conclusion is (we believe) striking. Recall that we need to implement the multivariate quadrature (3.2) for all trees of requisite order and that the number of such trees grows exponentially with  $m$ . Yet, as it turns out, far fewer binary trees determine the number of commutators that need be computed to attain order  $p$ . This renders our computational algorithm significantly cheaper and more competitive with alternative numerical approaches.

Let  $\tau$  be an arbitrary binary tree that has been constructed by the composition rules of § 2. We factorize  $\tau$  into two trees, the *I-tree* (integration tree) and the *C-tree* (commutation tree) in the following manner. The I-tree is formed from  $\tau$  by pruning all the vertices with two children. (By ‘pruning’ we mean a progression in a monotone ordering, from the ‘foliage’, the top of the tree, down to the root.) Similarly, the C-tree is formed by pruning all the vertices with a single child. These concepts are made considerably clearer upon an examination of an example. For instance, let us

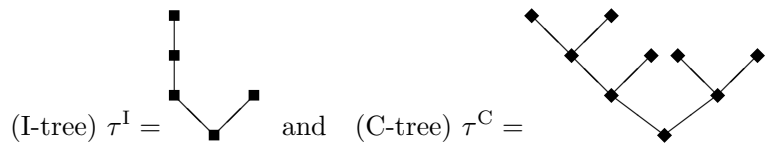
consider the tree



which corresponds to the Magnus expansion term

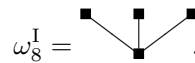
$$\int_0^t \left[ \int_0^{\kappa_0} \left[ \int_0^{\kappa_2} \left[ \int_0^{\kappa_3} a(\kappa_4) d\kappa_4, a(\kappa_3) \right] d\kappa_3, a(\kappa_2) \right] d\kappa_2, \left[ \int_0^{\kappa_0} a(\kappa_1) d\kappa_1, a(\kappa_0) \right] \right] \kappa_0.$$

Our factorization yields



For clarity, we denote the I-tree and the C-tree of  $\tau$  by  $\tau^I$  and  $\tau^C$ , respectively, and use ‘■’ and ‘◆’ to distinguish between their vertices.

Every C-tree is a *strictly* binary tree: each vertex has either two children or none. An I-tree, however, need not be binary at all. For example,



It is easy to verify that the number of vertices in an I-tree  $\tau^I$  is exactly the same as the number of integrals in  $\int H_\tau$ . Moreover, suppose that  $H_\tau$  has been derived from  $H_{\tau_1}$  and  $H_{\tau_2}$  by the second composition rule from §2, i.e. that (2.16) holds. Let us define by  $\mathcal{S}$  and  $\mathcal{S}_j$  the polyhedra of integration in  $\tau$  and in  $\tau_j$ ,  $j = 1, 2$ , respectively. Counting the number of integrals in each term, we deduce at once that

$$\text{ord } \tau^I = \text{ord } \tau_1^I + \text{ord } \tau_2^I.$$

Letting  $m = \text{ord } \tau$ , we set

$$B_{\tau, \mathbf{l}}(t) := u_{l_0}(t) \int_0^t \int_0^{\kappa_{i_1}} \cdots \int_0^{\kappa_{i_{m-1}}} \prod_{i=1}^m u_{l_i}(\kappa_i) d\kappa_m d\kappa_{m-1} \cdots d\kappa_1, \quad \mathbf{l} \in \Lambda_m^\nu,$$

and therefore, by (3.3),

$$b_{\mathbf{l}} = \int_0^1 B_{\tau, \mathbf{l}}(t) dt. \tag{4.1}$$

Similar definitions pertain to  $\tau_1$  and  $\tau_2$ . It is an immediate consequence of (2.16) that

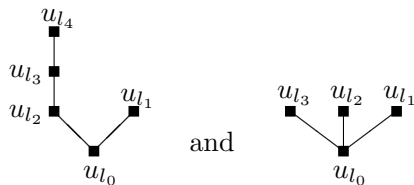
$$B_{\tau, \mathbf{l}}(t) = \int_0^t B_{\tau_1, \mathbf{l}_1}(\kappa) \int_0^\kappa B_{\tau_2, \mathbf{l}_2}(\xi) d\xi d\kappa, \tag{4.2}$$

where

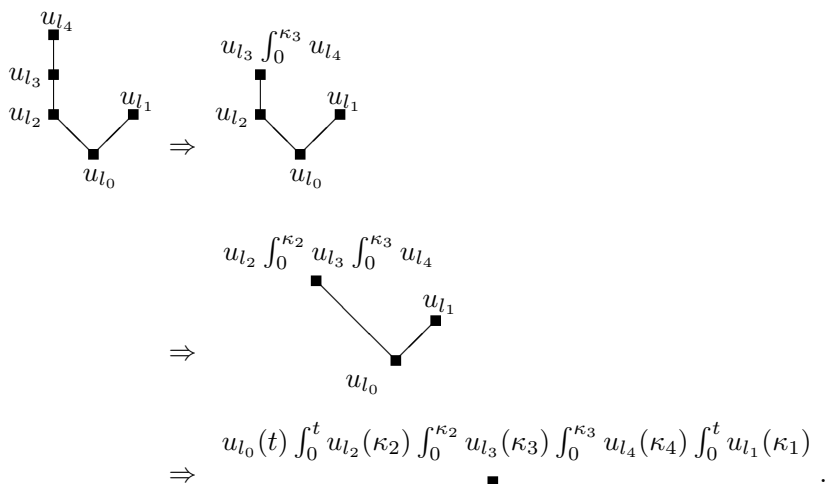
$$l = (l_1, l_2), \quad l_i \in \{1, 2, \dots, \nu\}^{\text{ord } \tau_i}, \quad i = 1, 2.$$

The weight  $b_l$  can be computed recursively by means of the formula (4.2), followed by the integral (4.1).

An alternative approach is, given  $l \in \Lambda_m^\nu$ , to label all the vertices of an I-tree by the cardinal Lagrange polynomials  $u_{l_i}$  in a *reverse natural ordering*: we label the root with  $u_{l_0}$  and then progress to label all the remaining vertices so that each child is always labelled after its parent and a child to the right is always labelled first. Thus, for example,



Next, the I-tree is *pruned*. Whenever a leaf (a childless vertex) is removed, the label of its parent is multiplied by the integral of the label of the vertex in question. Thus, for example,



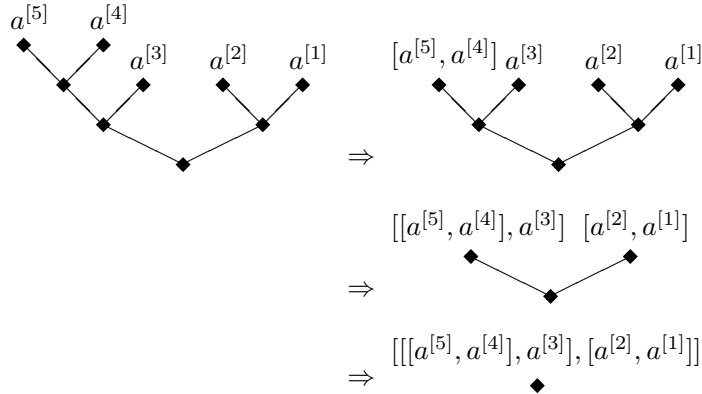
The outcome, as is trivial to prove, is the polynomial  $B_{\tau,l}$ . Finally, we use (4.1) to convert it to the weight  $b_l$ .

We next turn our attention to C-trees. Provided that  $\tau \in \mathcal{T}_{3m+1}$ , we deduce at once that its C-tree has order  $2m + 1$ , since  $\tau$  contains exactly  $m$  vertices with a single child. Moreover, it is trivial to observe that the C-tree has exactly  $m + 1$  leaves (i.e. childless vertices).

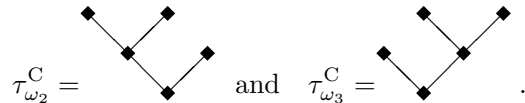
The C-tree determines the integrand similarly to the I-tree yielding the domain of integration. Similar connection extends to quadrature formulae (3.2): while the I-tree has just been exploited in the derivation of  $b_l$ , we can use the C-tree to determine the structure of  $\mathcal{L}(a(c_{l_0}h), a(c_{l_1}h), \dots, a(c_{l_m}h))$ . It follows from our original definition of the connection between trees and expansion terms in the Magnus series that the



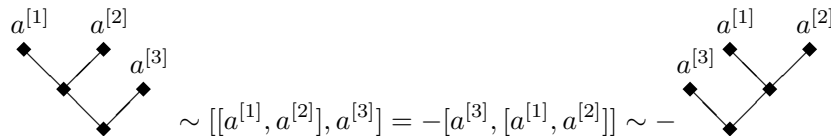
latter term can be obtained as follows. We label the leaves of the C-tree, progressing from the right leftwards, by  $a(c_{l_0}h), a(c_{l_1}h), \dots, a(c_{l_m}h)$ . Next we prune the tree. Each time a pair of children is pruned, their parent receives the label which is the commutator of their labels. For example, letting  $a^{[i]} := a(c_{l_i}h)$ ,  $i = 1, 2, \dots, m$ ,



The C-tree is instrumental not just in regard to the recovery of the exact form of commutators but also in exploiting symmetries to eliminate many of the commutators, thereby decreasing a great deal the computational expense. Recall from § 3 that  $H_{\omega_2}$  and  $H_{\omega_3}$  require the calculation of exactly the same commutators, albeit with a different labelling of the indices. This kind of symmetry is easily revealed from the structure of the C-tree. Thus,



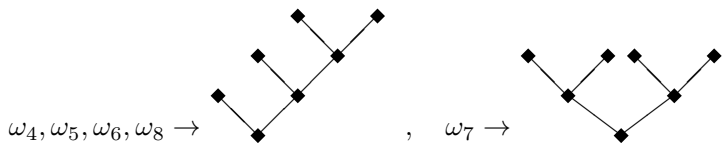
Bearing in mind the relationship between C-trees and commutators, we observe that rotating a C-tree about a vertical axis is equivalent to exchanging the terms in the underlying commutator. In other words,



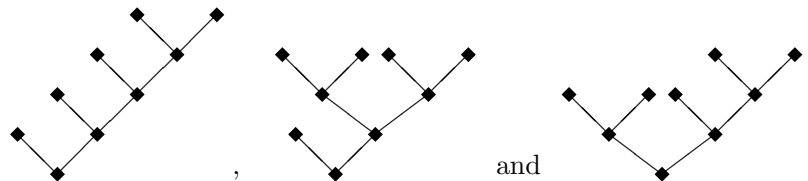
and there is no need to evaluate the second commutator. Of course, we need to keep track of the exchange of indices, but this can be done easily.

In general, we say that two strictly binary trees,  $\tau_1$  and  $\tau_2$ , say, are *equivalent* if  $\tau_2$  can be converted to  $\tau_1$  by a finite number of rotations with respect to vertical axes. Only a single representer from a class of equivalent trees need be evaluated for every index  $l$ . Hence, it is of interest to compare the number of equivalent trees to the overall number of strictly binary trees of any given order  $2m + 1$ . We recall from § 2 that there are exactly  $\theta_m = (2m)!/[m!(m + 1)!]$  members in  $\mathcal{T}_{3m \pm 1}$ , and hence this is also the number of strictly binary trees of order  $2m + 1$ . Let  $\hat{\theta}_m$  denote the number of equivalent strictly binary trees of order  $2m + 1$ . Trivially,  $\hat{\theta}_0 = \hat{\theta}_1 = 1$  and we have already seen also that  $\hat{\theta}_2 = 1$ . It follows easily from table 1 that  $\hat{\theta}_3 = 2$ , with

the representers



Likewise, for  $m = 4$  we may choose the representers



and therefore  $\tilde{\theta}_4 = 3$ .

In general,  $\tilde{\theta}_m$  is known in combinatorics as the  $m$ th *Wedderburn–Etherington number* (Comtet 1970). It is easy to affirm the recursive formula

$$\left. \begin{aligned} \tilde{\theta}_0 &= 1, \\ \tilde{\theta}_{2m-1} &= \frac{1}{2} \sum_{j=0}^{2m-2} \tilde{\theta}_j \tilde{\theta}_{2m-j-2} + \frac{1}{2} \tilde{\theta}_{m-1}, \\ \tilde{\theta}_{2m} &= \frac{1}{2} \sum_{j=0}^{2m-1} \tilde{\theta}_j \tilde{\theta}_{2m-j-1}, \end{aligned} \right\} m \in \mathbb{N},$$

and it follows after an elementary manipulation that the generating function

$$\tilde{\Theta}(z) = \sum_{m=0}^{\infty} \tilde{\theta}_m z^m$$

obeys the functional equation  $\tilde{\Theta}(z) = 1 + \frac{1}{2}z[\tilde{\Theta}(z)]^2 + \frac{1}{2}z\tilde{\Theta}(z^2)$ . The asymptotic behaviour of  $\{\tilde{\theta}_m\}_{m \in \mathbb{Z}^+}$  has been determined by Otter (1948):

$$|\tilde{\theta}_m|^{1/m} \rightarrow 2.483\,253\,536\dots, \quad m \rightarrow \infty.$$

This is clearly much better than the growth in the number of all rooted trees, a phenomenon illustrated in table 2.

Redundancy of strictly binary trees is just one mechanism that allows us to reduce the number of commutators that need be computed. Other devices are the antisymmetry of the commutator and the Jacobi identity. We demonstrate in the sequel that an exploitation of these two devices reduces the extent of computation in the implementation of our quadrature formulae to just a *single tree* for every  $m \geq 0$ .

We denote by  $\mathcal{V}_m$  the set of all the strictly binary trees of order  $2m + 1$  (that is, all the C-trees that correspond to trees in  $\mathcal{T}_{3m+1}$ ). Given  $\tau \in \mathcal{V}_m$  and  $a_1, a_2, \dots, a_m \in \mathfrak{g}$ , we say that  $h_\tau(a_1, a_2, \dots, a_m) \in \mathfrak{g}$  is a *realization* of  $\tau$  if it is obtained by the nested action of  $m$  commutators, consistently with our connection between commutation and C-trees. Thus, for example,

$$h_{\omega_2^C}(a_1, a_2, a_3) = [a_1, [a_2, a_3]] \quad \text{and} \quad h_{\omega_7^C}(a_1, a_2, a_3, a_4) = [[a_1, a_2], [a_3, a_4]].$$

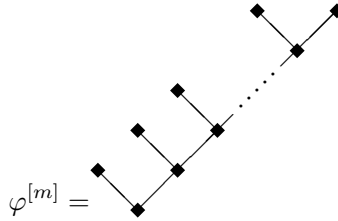
*Phil. Trans. R. Soc. Lond. A* (1999)

Table 2. The number of all C-trees (ALL) and of equivalence classes of C-trees (ECCT) for  $m = 0, 1, \dots, 6$ 

$m$	ALL	ECCT
0	1	1
1	1	1
2	2	1
3	5	2
4	14	3
5	42	6
6	132	11

Let  $\mathcal{F}_m^\nu$  stand for the linear subset of  $\mathfrak{g}$ , which is generated by all realizations  $h_\tau(a^{[l_1]}, a^{[l_2]}, \dots, a^{[l_m]})$ , where  $\tau$  ranges across  $\mathcal{V}_m$  and  $\mathbf{l} \in \Lambda_m^\nu$ . The dimension of  $\mathcal{F}_m^\nu$  is denoted by  $W_{m,\nu}$ . Clearly, it represents the number of commutators that must be calculated in the implementation of the quadrature formulae (3.2) for all  $\int H_\tau$  with  $\tau \in \mathcal{T}_{3m+1}$ . Motivated by an earlier version of this paper, Munthe-Kaas & Owren (1998) proved that in general  $W_{2,\nu} = \frac{1}{3}(\nu^2 - 1)\nu$  and  $W_{3,\nu} = \frac{1}{4}(\nu^2 - 1)\nu^2$ †.

We select a particular tree in  $\mathcal{V}_m$ , namely



Therefore,  $\varphi^{[2]} = \omega_2^C$  and  $\varphi^{[3]} = \omega_4^C$ . Each realization  $\varphi^{[m]}$  is easily identified with the iterated commutator,

$$h_{\varphi^{[m]}}(a_1, a_2, \dots, a_m) = [a_1, [a_2, \dots, [a_{m-1}, a_m] \dots]].$$

We adopt the notation

$$[l_1, l_2, \dots, l_m] = h_{\varphi^{[m]}}(a^{[l_1]}, a^{[l_2]}, \dots, a^{[l_m]}).$$

Let  $\mathcal{K}_m^\nu$  be the linear space spanned by all the realizations of  $\varphi^{[m]}$ . Then, according to theorem 3.1 of Onischik (1993), the Lie algebra  $\mathcal{K}^\nu$ , generated by the free system of generators  $\{a^{[1]}, a^{[2]}, \dots, a^{[\nu]}\}$  can be written in the form

$$\mathcal{K}^\nu = \bigoplus_{r=1}^{\infty} \mathcal{K}_r^\nu.$$

In particular, every term that can be generated from these elements by  $m$  commutators belongs to

$$\bigoplus_{r=1}^m \mathcal{K}_r^\nu.$$

† The dimension might be reduced if there are linear dependencies between the  $a^{[j]}$  and their commutators. In an extreme case, if all the elements  $a^{[j]}$  commute, then  $\mathcal{F}_m^\nu = \emptyset$ . Moreover, needless to say, it cannot exceed the dimension of  $\mathfrak{g}$  when the Lie algebra is finite dimensional. We henceforth disregard these phenomena.

We deduce that  $\mathcal{F}_m^\nu$  is contained in the above set and conclude that

$$\bigoplus_{r=1}^m \mathcal{F}_r^\nu = \bigoplus_{r=1}^m \mathcal{K}_r^\nu. \quad (4.3)$$

Given that  $\mathcal{K}_r^\nu$  for  $r = 1, 2, \dots, m-1$  would have been already evaluated, we conclude that, provided that we have calculated all the commutators corresponding to the realizations of  $\varphi^{[m]}$  in  $\{a^{[0]}, a^{[1]}, \dots, a^{[m]}\}$ , we can compute all of  $\mathcal{F}_m^\nu$ , and hence all the quadrature formulae allowing us to advance from order  $m$  to order  $m+1$ .

This analysis affirms our observation that the realizations of  $\varphi^{[2]}$  and  $\varphi^{[3]}$  can be used to construct all other commutators of their orders, and it extends it to all  $\varphi^{[m]}$ . Moreover, the dimension of  $\mathcal{K}_m^\nu$  is known (Onischik 1993, p. 71) and hence

$$W_{m,\nu} = \dim \mathcal{K}_m^\nu = \frac{1}{m+1} \sum_{d|(m+1)} \mu(d) \nu^{(m+1)/d}, \quad (4.4)$$

where the sum is carried out over all the divisors of  $m+1$  and the *Möbius function*  $\mu(d)$  equals  $(-1)^k$  if  $d$  is a product of  $k$  distinct primes; otherwise (i.e. when  $d$  is not square-free) it is zero (van Lint & Wilson 1992). Thus,

$$\begin{aligned} W_{2,\nu} &= \frac{1}{3}(\nu^2 - 1)\nu, \\ W_{3,\nu} &= \frac{1}{4}(\nu^2 - 1)\nu^2, \\ W_{4,\nu} &= \frac{1}{5}(\nu^4 - 1)\nu, \\ W_{5,\nu} &= \frac{1}{6}(\nu^3 + \nu - 1)(\nu^2 - 1)\nu. \end{aligned}$$

Unfortunately, neither (4.3) nor (4.4) is accompanied by a *constructive* proof that can illuminate a numerical procedure whereby the least possible number of commutators is computed. This has been recently remedied by Munthe-Kaas & Owren (1998), who have presented careful analysis and explicit algorithms that take advantage of the Jacobi identity to produce remarkable computational savings.

## 5. Magnus series and iterated commutators

The method of *Magnus series* and the method of *iterated commutators* (Iserles 1984) represent alternative means to represent and discretize the solution of the system (1.1). Although they share the important attribute of respecting Lie-algebraic structure, the two methods are, in a sense, based on ‘orthogonal’ reasoning. While the Magnus series is based on a single exponential of an infinite (or, subject to truncation, ‘long’) expansion, the method of iterated commutators represents the solution as an infinite product of exponentials, each with a simple argument, and it approximates the solution by a finite product of this type. It is natural to consider an approach that borrows elements from both Magnus series and iterated commutators.

**Lemma 5.1.** *Let  $y$  be the solution of (1.1) and set*

$$x(t) = e^{-\varrho(t)} y(t), \quad t \geq 0,$$

where  $\varrho$  is an arbitrary sufficiently smooth function. Then  $x$  obeys the differential equation

$$x' = bx, \quad t \geq 0, \quad x(0) = \text{Id}, \quad (5.1a)$$

where

$$b(t) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \operatorname{ad}[\varrho]^k[a] - \sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)!} \operatorname{ad}[\varrho]^k[\varrho'] \quad (5.1b)$$

and  $\operatorname{Id}$  is the identity. Moreover, if  $a, \varrho : \mathbb{R}^+ \rightarrow \mathfrak{g}$  then also  $b : \mathbb{R}^+ \rightarrow \mathfrak{g}$ , and consequently  $x(t) \in G$  for every  $t \geq 0$ .

*Proof.* Formally differentiating  $x$  we deduce at once that

$$b(t) = \left[ \frac{d}{dt} e^{-\varrho(t)} \right] e^{\varrho(t)} + e^{-\varrho(t)} a(t) e^{\varrho(t)}, \quad t \geq 0.$$

We next replace  $\sigma$  with  $-\varrho$  in the proof of theorem 2.1, concluding that

$$\sum_{m=0}^k (-1)^{k-m} \binom{k}{m} \varrho^m \varsigma \varrho^{k-m} = \operatorname{ad}[\varrho]^k[\varsigma], \quad k \in \mathbb{Z}^+ \quad (5.2)$$

and

$$\left[ \frac{d}{dt} e^{-\varrho(t)} \right] e^{\varrho(t)} = - \sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)!} \operatorname{ad}[\varrho]^k[\varrho']. \quad (5.3)$$

Employing (5.2), we also conclude that

$$\begin{aligned} e^{-\varrho(t)} a e^{\varrho(t)} &= \sum_{l=0}^{\infty} \sum_{k=0}^{\infty} \frac{(-1)^k}{k!l!} \varrho^l a \varrho^k \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \sum_{l=0}^k (-1)^l \binom{k}{l} \varrho^l a \varrho^{k-l} = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!} \operatorname{ad}[\varrho]^k[a]. \end{aligned} \quad (5.4)$$

Putting (5.3) and (5.4) together, we obtain the explicit form of the function  $b$  in (5.1). The proof of the lemma is concluded by observing that both (5.3) and (5.4) produce terms in  $\mathfrak{g}$  as long as  $a(t), \varrho(t) \in \mathfrak{g}$  for  $t \geq 0$ . ■

Note that letting  $\varrho(t) = \int_0^t a(\kappa) d\kappa$  results in the formula (1.5) with  $a_m \equiv a$ .

We let  $a_0 \equiv a$  and replace the Magnus expansion of  $\sigma$ , where  $y(t) = \exp \sigma(t)$ , with  $\sigma_{0,p_0} \equiv \sigma_{p_0}$ , where  $p_0 \in \mathbb{Z}^+$ . In accordance with theorem 2.7, we note that

$$y_{[0]}(t) := e^{\sigma_{0,p_0}(t)} y_0 = y(t) + \mathcal{O}(t^{p_0+2}). \quad (5.5)$$

Letting  $\varrho \equiv \sigma_{0,p_0}$ , we form  $a_1 \equiv b$  and note that (5.5) implies that

$$x(t) = \operatorname{Id} + \mathcal{O}(t^{p_0+2}) \quad \text{and} \quad a_1(t) = \mathcal{O}(t^{p_0+1}).$$

Next we truncate the Magnus expansion of (5.1) as in (2.19), where  $p = p_1$ . The outcome is  $\sigma_{1,p_1} \equiv \sigma_{p_1}$  which, after exponentiation, yields the approximant

$$y_{[1]}(t) = e^{\sigma_{0,p_0}(t)} e^{\sigma_{1,p_1}(t)} y_0, \quad t \geq 0.$$

To examine the order of  $y_{[1]}$  we note that theorem 2.7 need now be applied with  $\lambda = p_0 + 1$ . Therefore,

$$y_{[1]}(t) = y(t) + \mathcal{O}(t^{(p_0+1)(p_1+1)} + 1). \quad (5.6)$$

The general pattern emerges. Given  $a_m(t) = \mathcal{O}(t^{\gamma_m})$ ,  $m \in \mathbb{N}$ , where

$$\left. \begin{aligned} \gamma_0 &= 0, \\ \gamma_m &= (p_{m-1} + 1)(\gamma_{m-1} + 1), \quad m \in \mathbb{N}, \end{aligned} \right\} \quad (5.7)$$

we approximate the solution of

$$x' = a_m x, \quad t \geq 0, \quad x(0) = \text{Id},$$

by  $\exp \sigma_{x,p_m}$ . The outcome is the approximant

$$y_{[m]}(t) = e^{\sigma_{0,p_0}(t)} e^{\sigma_{1,p_1}(t)} \dots e^{\sigma_{m,p_m}(t)} y_0 = y(t) + \mathcal{O}(t^{\gamma_m+1}), \quad t \geq 0.$$

Unless we decide to terminate, we next generate  $a_{m+1}$  by letting  $a = a_m$ ,  $\varrho = \sigma_{m,p_m}$  in (5.1).

Note that (5.7) is consistent with (5.5) and (5.6). Letting  $p_m \equiv 1$  we recover the standard iterated commutators from Iserles (1984) and Zanna (1996). With greater generality, choosing  $p_m \equiv r \in \mathbb{N}$ , we can verify at once that

$$\gamma_m = \frac{r+1}{r} [(r+1)^m - 1], \quad m \in \mathbb{Z}^+.$$

Therefore the order increases exponentially.

Blending Magnus series with iterated commutators yields an abundant choice of discretization methods for every order  $p \geq 3$ . Each method corresponds to a specific choice of  $m \in \mathbb{N}$  and  $p_0, p_1, \dots, p_{m-1} \in \mathbb{N}$  such that  $\gamma_m \geq p$ . It is natural to search for a method that is, in some well-defined sense, ‘optimal’, the criterion being either the computational cost or ‘nice’ qualitative features. In a future paper we will address ourselves to complexity issues associated with different methods of this kind and demonstrate that distinct criteria of computational cost (e.g. the relative cost of commutators and of function evaluations) often lead to different optimal methods.

## 6. Numerical examples and concluding remarks

The method of Magnus series is considerably more than just a numerical algorithm. By carrying out the full expansion or, alternatively, truncating (2.6) and subjecting the outcome to perturbation analysis, it should be possible to derive a wealth of qualitative results about linear systems (1.1). This, however, should not obscure the potential of Magnus series as a computational procedure.

The main purpose of this section is to present a number of preliminary numerical results. Our goal is neither to investigate in depth the implementation of Magnus series as a numerical algorithm nor to present a compendium of comparative numerical results that might act as a guide to pros and cons of different Lie-group solvers. More modestly, we intend merely to demonstrate that the method of Magnus series, which at first glance might appear to be exotic and bizarre, can be at a relatively modest expense harnessed to produce results that display advantageous behaviour when compared with classical numerical schemes. To this end we compare three methods, all of order 4. Specializing to the linear system (1.1) and letting  $t_n = nh$ ,  $y_n \approx y(t_n)$ , where  $n \in \mathbb{Z}^+$  and  $h > 0$  is the step-length, these methods are

(1) RK4. Explicit Runge–Kutta:

$$\begin{aligned}k_1 &= a(t_n)y_n, \\k_2 &= a(t_n + \tfrac{1}{2}h)(y_n + \tfrac{1}{2}hk_1), \\k_3 &= a(t_n + \tfrac{1}{2}h)(y_n + \tfrac{1}{2}hk_2), \\k_4 &= a(t_{n+1})(y_n + hk_3), \\y_{n+1} &= y_n + \tfrac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4).\end{aligned}$$

(2) GL4. Gauss–Legendre Runge–Kutta:

$$\begin{aligned}k_1 &= a((\tfrac{1}{2} - \tfrac{1}{6}\sqrt{3})h)[y_n + \tfrac{1}{4}hk_1 + (\tfrac{1}{4} - \tfrac{1}{6}\sqrt{3})hk_2], \\k_2 &= a((\tfrac{1}{2} + \tfrac{1}{6}\sqrt{3})h)[y_n + (\tfrac{1}{4} + \tfrac{1}{6}\sqrt{3})hk_1 + \tfrac{1}{4}hk_2], \\y_{n+1} &= y_n + \tfrac{1}{2}h(k_1 + k_2).\end{aligned}$$

(3) MG4. Truncated Magnus expansion with Gaussian points:

$$\begin{aligned}a_{[1]} &= a(t_n + (\tfrac{1}{2} - \tfrac{1}{6}\sqrt{3})h), \\a_{[2]} &= a(t_n + (\tfrac{1}{2} + \tfrac{1}{6}\sqrt{3})h), \\ \tilde{\sigma} &= \tfrac{1}{2}h(a_{[1]} + a_{[2]}) + \tfrac{1}{12}\sqrt{3}h^2[a_{[2]}, a_{[1]}] + \tfrac{1}{80}h^3[a_{[2]} - a_{[1]}, [a_{[2]}, a_{[1]}]], \\y_{n+1} &= e^{\tilde{\sigma}}y_n,\end{aligned}$$

with the exponential function calculated to machine accuracy by the MATLAB function `expm`.

Note that each method requires exactly two function evaluations per step, provided that  $a(t_{n+1})$  is reused in the next step for RK4.

We have applied these methods to the following three ordinary differential systems.

(A)  $y'' + ty = 0$ ,  $y(0) = 1$ ,  $y'(0) = 0$ . The equation was converted to a two-dimensional vector system with

$$a(t) = \begin{bmatrix} 0 & 1 \\ -t & 0 \end{bmatrix}, \quad t \geq 0,$$

and integrated in the range  $0 \leq t \leq 100$ . Note that the exact solution, a linear combination of *Airy functions*  $\text{Ai}(-t)$  and  $\text{Bi}(-t)$  (Abramowitz & Stegun 1965, p. 446), is asymptotically stable (Bellman 1969) and that it displays oscillations of progressively increasing frequency. Formally, conversion of the equation to a vector system yields a fundamental solution that evolves on the special linear group  $\text{SL}_2(\mathbb{R})$ , but this is of little significance: it is the highly oscillatory behaviour of the solution that presents the most formidable computational challenge.

(B) The  $10 \times 10$  system with

$$[a(t)]_{k,l} = \begin{cases} t \sin \frac{2}{11}k\pi, & k = l, \\ \cos \frac{2}{11}\pi kt, & k = l - 1, \\ -\cos \frac{2}{11}\pi lt, & k = l + 1, \\ 0, & |k - l| \geq 2, \end{cases} \quad k, l = 1, 2, \dots, 10,$$

with  $y(0) = \text{Id}$  and  $0 \leq t \leq 5$ . Note that  $\text{tr} a(t) = 0$ , and hence  $a(t) \in \mathfrak{sl}_{10}(\mathbb{R})$ . Therefore,  $y$  lies in the special linear group, i.e.  $\det y(t) \equiv 1$ .

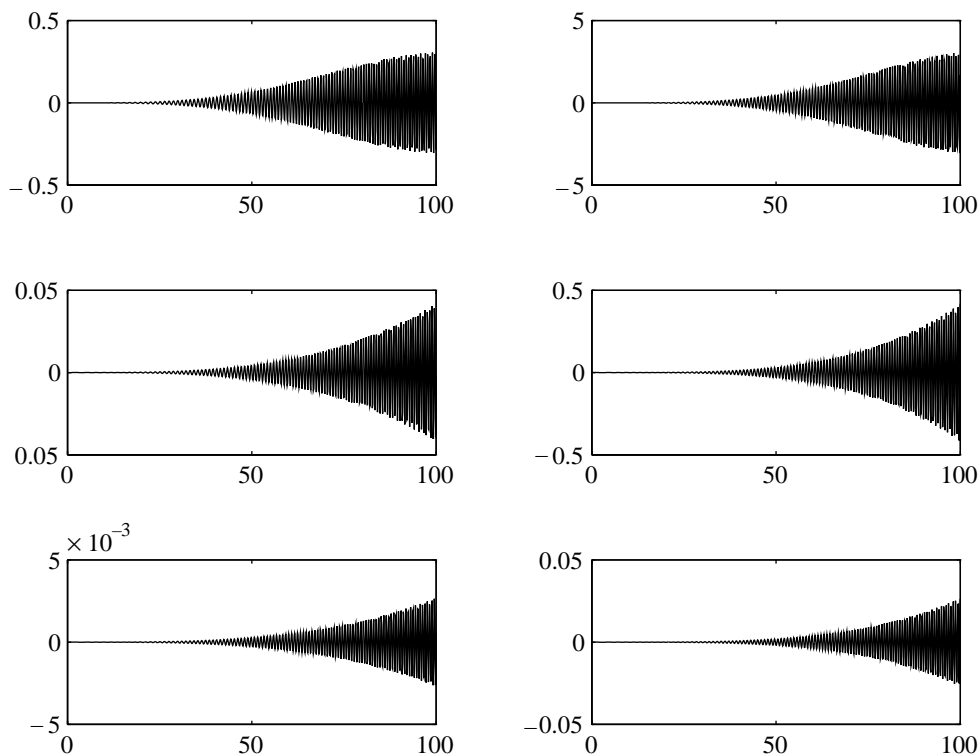


Figure 1. The error in the solution of the problem (A) by RK4 for  $0 \leq t \leq 100$ . The left-hand column displays the error in  $y$ , the right-hand column in  $y'$ , for step sizes  $h = \frac{1}{10}$  (top row),  $h = \frac{1}{20}$  (second row) and  $h = \frac{1}{40}$  (bottom row).

(C) Another  $10 \times 10$  system, namely

$$[a(t)]_{k,l} = \begin{cases} it \sin \frac{2}{11} \pi k, & k = l, \\ \cos \frac{1}{5} \pi kt, & k = l - 1, \\ -\cos \frac{1}{5} \pi lt, & k = l + 1, \\ 0, & |k - l| \geq 2, \end{cases} \quad k, l = 1, 2, \dots, 10,$$

again with  $y(0) = \text{Id}$  and  $0 \leq t \leq 10$ . This time  $a^*(t) + a(t) \equiv 0$  and  $\text{tr } a(t) = 0$ , and therefore  $a(t) \in \mathfrak{su}_{10}(\mathbb{R})$ :  $y(t)$  is unitary and  $\det y(t) \equiv 1$  for all  $t \geq 0$ .

Figure 1 displays the error of RK4 for problem (A). Although the error decays consistently with order 4 when the step size  $h$  is being refined, it starts from a very substantial base. This can be attributed to the unstable and highly oscillatory nature of the exact solution of problem (A) and means that an exceedingly small step size is required for the attainment of reasonable accuracy.

The implicit A-stable method GL4 is only marginally better, as is evident from figure 2. Again, the error is attenuated in line with  $h$  but, as was the case with RK4, a very small step size is required to counteract the deterioration in numerical precision. The performance of GL4 is disappointing because, according to conventional numerical wisdom, we might have expected it to perform much better than RK4 by virtue of its A-stability.



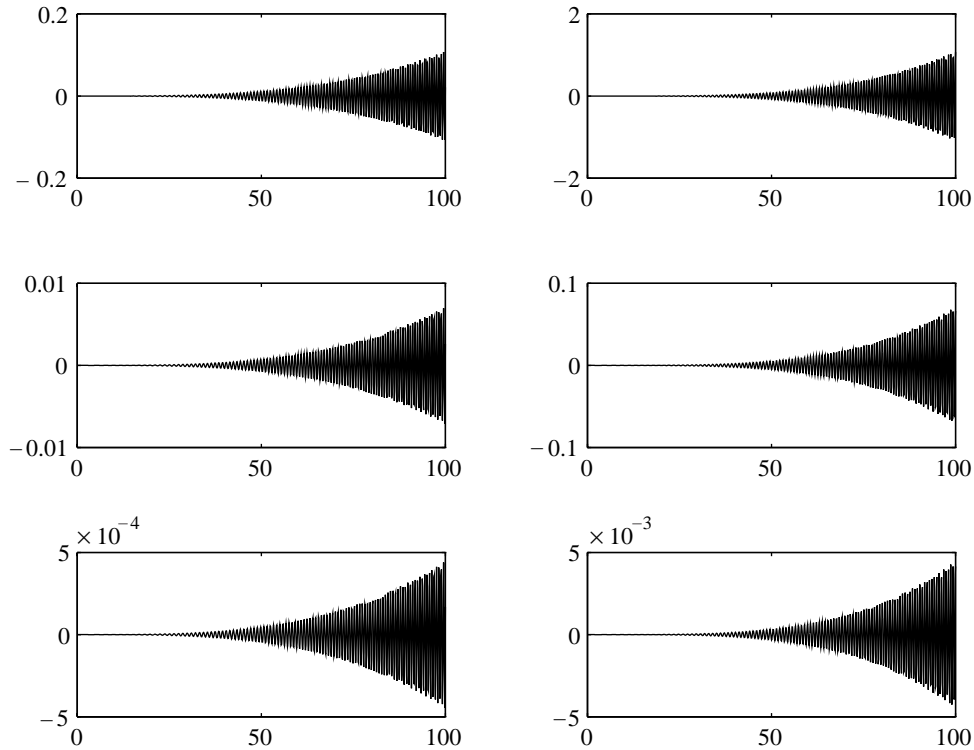


Figure 2. The error in the solution of the problem (A) by GL4. The interpretation of different sub-figures is the same as for figure 1.

The situation is drastically different when problem (A) is discretized with MG4, as can be ascertained from figure 3. The error for the coarsest step size,  $h = \frac{1}{10}$ , is more than five significant digits more accurate than for the other methods! In general, it appears that the Magnus-series approach is superior, in comparison with classical methods, not just with regard to the invariance of a Lie-group structure but also in the general issue of the recovery of qualitative features and robustness with regard to ‘difficult’ phenomena like high-oscillation or complicated asymptotics. Similar pleasing behaviour is probably associated with other Lie-group methods, in particular with iterated commutators (Zanna 1996). This, we believe, is a significant observation, which should act as a spur for further research.

This is perhaps the place to comment that our comparison was solely in terms of accuracy (and, in the sequel, departure from invariants) for identical step-length sequences. An alternative equally legitimate criterion is a comparison with regard to computational expense. We have not done this in general and our programs have been neither optimized for speed nor written with a variable step-size and an error controller. It might be worthwhile to mention, however, that MG4, after a minor optimization to account for the special feature of the problem (A) (in particular, with an explicit evaluation of the  $2 \times 2$  exponential) performed significantly better (counting flops) than the MATLAB `ode45` subroutine (a variable-step implementation of a fourth-order Runge–Kutta scheme).

Figure 4 displays the Euclidean norm of the error in the solution of problem (B) by

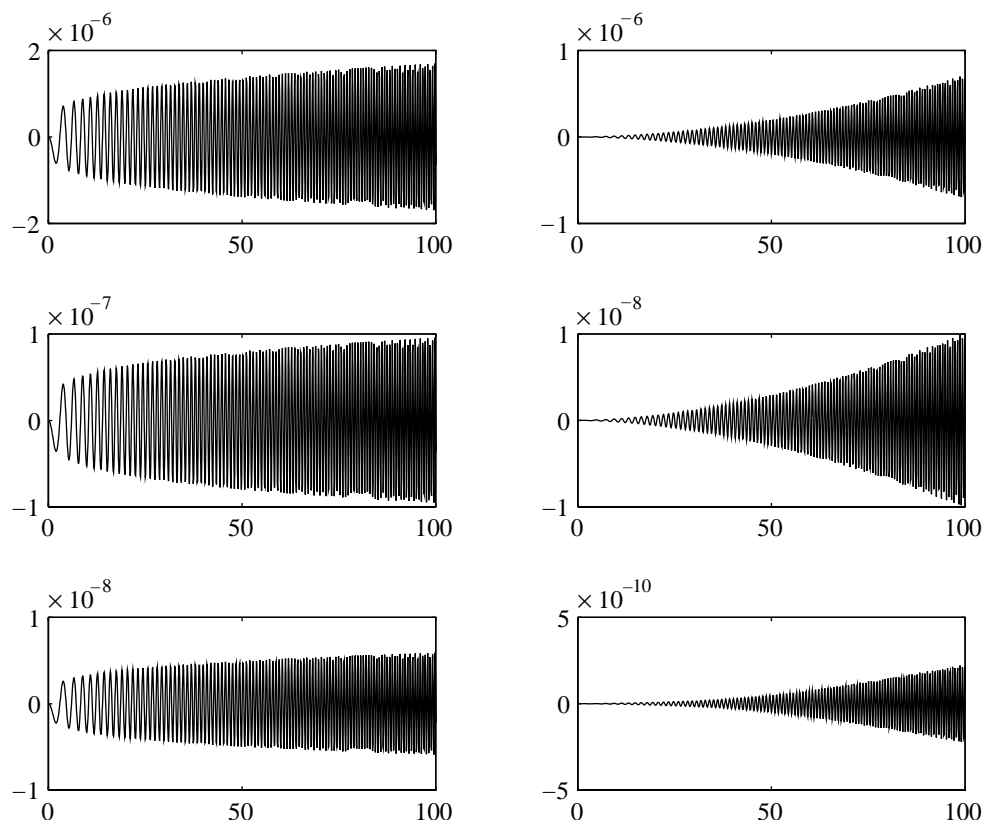


Figure 3. The error in the solution of the problem (A) by MG4. The interpretation of different subfigures is the same as for figures 1 and 2.

the three methods for four step-sizes,  $h = \frac{1}{10}, \frac{1}{20}, \frac{1}{50}, \frac{1}{100}$ . Note that GL4 is marginally better than RK4, while MG4, which starts similarly, accumulates error much slower than the other two methods: at  $t = 5$  it is better roughly by a factor of 100.

The exact solution of problem (B) is unstable and the extent of instability grows with increasing  $t$ . This is reflected by progressive deterioration in the absolute error for all three methods in figure 4.

The main motivation for the use of a Magnus expansion is not to enhance accuracy but to force the solution to respect a Lie-group structure. Insofar as problem (B) is concerned, the Lie group is  $SL_{10}(\mathbb{R})$ , hence  $\det y(t) \equiv 1, t \geq 0$ . This is true by design (to machine accuracy) for MG4, while both RK4 and GL4 depart from the manifold. This departure is illustrated (for the same sequence of step sizes as before) in figure 5.

This is perhaps the place to comment on the retention of invariants by classical numerical methods. *Multistep* methods score the worst and it is proved in Iserles (1997) that they retain only linear invariants. Therefore, except for the trivial case of *general linear groups*  $GL_d(\mathbb{R})$  and  $GL_d(\mathbb{C})$ , they should not be used when the retention of a Lie-group structure is at issue. Certain Runge–Kutta methods—GL4, but not RK4—are somewhat better, since they can recover quadratic invariants (Calvo *et al.* 1996). Hence they can be safely used with the *orthogonal group*  $O_d(\mathbb{R})$ , the

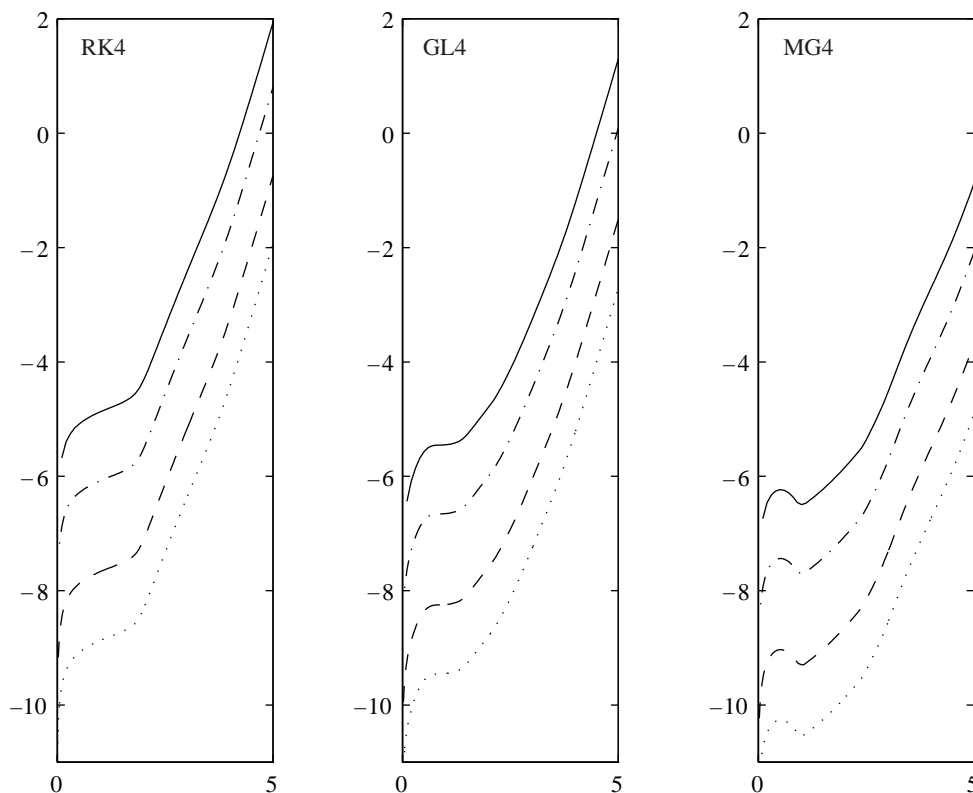


Figure 4. The error  $\log_{10} \|y_n - y(t_n)\|_2$  for the three numerical methods for problem (B) and  $0 \leq t \leq 5$ . The solid, dot-dash, dash-dash and dotted lines correspond to  $h = \frac{1}{10}, \frac{1}{20}, \frac{1}{50}$  and  $\frac{1}{100}$ , respectively.

*unitary group*  $U_d(\mathbb{C})$  and the *symplectic group*  $Sp_{2d}(\mathbb{R})$ . Virtually no non-quadratic manifolds can be retained by Runge–Kutta methods (Iserles & Zanna 1999) and, in particular, no Runge–Kutta may respect manifolds that are level sets of multivariate polynomials of (total) degree greater than 3 (Iserles & Zanna 1999). Hence, no classical method is assured of retaining  $SL_d(\mathbb{R})$  for  $d \geq 3$ , since the condition  $\det y \equiv 1$  for a  $d \times d$  matrix  $y$  is expressible as a level set of a  $d$ -degree polynomial. Seen in this light, figure 5 is hardly surprising.

Our final numerical example is problem (C) and the relevant results are displayed in figure 6. The absolute numerical error follows the same pattern as for problem (B), namely it accumulates slower with GL4 than with RK4, and yet slower with MG4.

Recall that, for the present example,  $y(t) \in SU_{10}(\mathbb{C}) = U_{10}(\mathbb{C}) \cap SL_{10}(\mathbb{C})$ . In other words,  $\bar{y}(t)^T y(t) \equiv \text{Id}$  and  $\det y(t) \equiv 1$ . The Magnus expansion retains, by design, both invariants, and we know that the Gauss–Legendre Runge–Kutta method GL4 retains the first and violates the second. The classical Runge–Kutta method RK4 respects neither invariant. All this is demonstrated in figure 6.

We hasten to reiterate that our numerical results are not intended to present a comprehensive study, merely to provide evidence of computational potential. A future paper will address itself to detailed complexity analysis of Magnus series and of their different combinations, *à la* § 5, with iterated commutators.

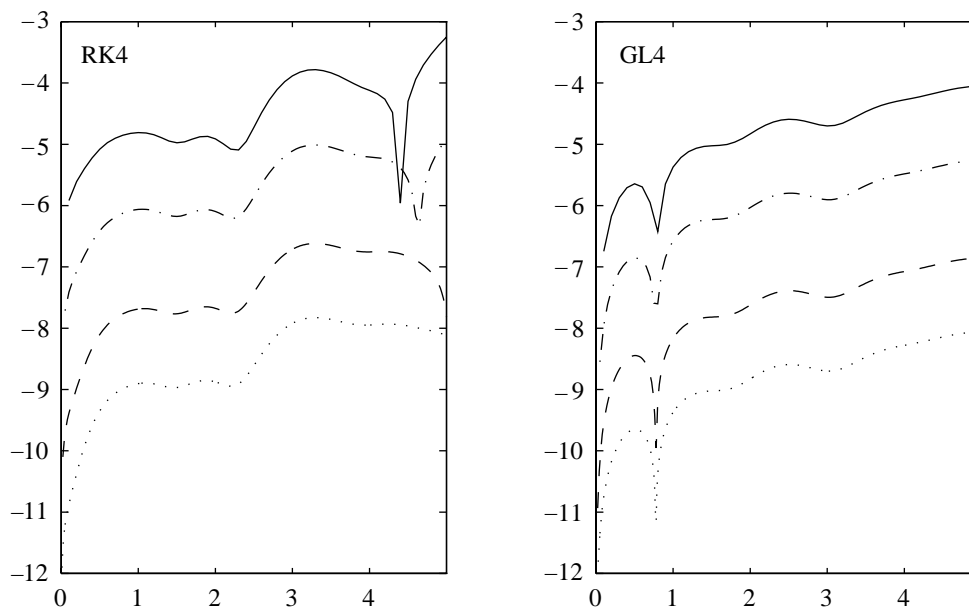


Figure 5. The error  $\log_{10} |\det y_n - 1|$  for RK4 (left) and GL4 (right), applied to problem (B). The interpretation of different lines is the same as in figure 4.

This is the place to mention that the technique of Magnus series is not the only numerical method that is assured to stay on a Lie group. A number of alternative approaches have emerged in the past few years: the methods of *rigid frames* (Crouch & Grossman 1993; Owren & Marthinsen 1997), *discrete gradients* (Quispel & Turner 1996) and, perhaps most promisingly, Runge–Kutta methods corrected for a Lie group (Munthe-Kaas 1997). It is not the purpose of this paper to weigh the pros and cons of alternative techniques, except for observing that, in the present, nascent state-of-the-art in numerical methods on differentiable manifolds, it is perhaps premature to draw definite conclusions.

To conclude this paper, we summarize its three main results. Firstly, we have formally introduced Magnus series for Lie-group linear equations (1.1) and established a relationship between their terms and a subset of binary trees. This connection is the key to the understanding and use of Magnus series, since it allows for their recursive evaluation and leads to a convergence proof. Secondly, we have demonstrated that all the integrals that are used in a discretized version of Magnus series can be approximated by using a single set of univariate quadrature points. Hence, an order- $p$  numerical method requires just  $\lfloor \frac{1}{2}(p+1) \rfloor$  function evaluations. Thirdly, using combinatorial arguments and comparing the dimension of subspaces in the Lie algebra  $\mathfrak{g}$ , we have proved that it is possible to reduce the number of commutators that are necessary to obtain an order- $p$  discretization. Specifically, just  $p-2$  C-trees determine the commutators that need be evaluated, while the number of all C-trees increases exponentially with  $p$ .

We believe that the approach of Magnus series is of interest both as an analytic construct and as a numerical tool. Needless to say, much remains to be done. In particular, we mention the implementation of Magnus series when  $a(t)$  is a differential operator, issues of computational complexity, generalization of the technique

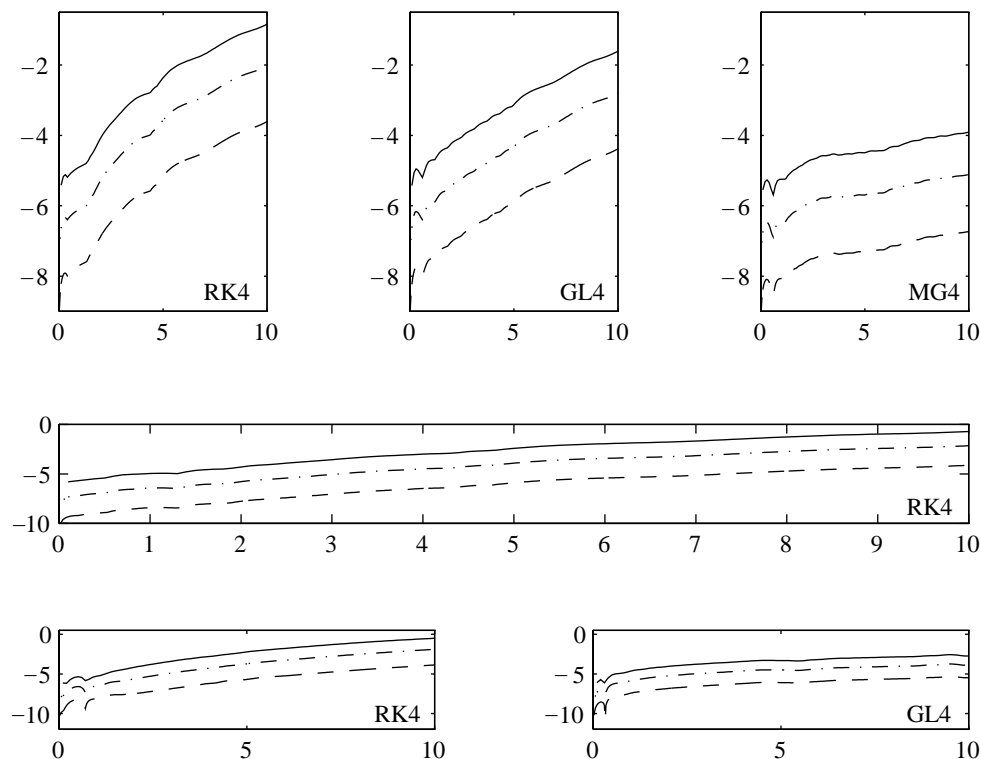


Figure 6. Computational results for problem (C),  $0 \leq t \leq 10$ . The top row displays the absolute error  $\log_{10} \|y_n - y(t_n)\|_2$ : the solid, dot-dash and dash-dash lines correspond to  $h = \frac{1}{10}$ ,  $\frac{1}{20}$  and  $h = \frac{1}{50}$ , respectively. The middle row displays the orthogonality error  $\log_{10} \|y_n^T y_n - \text{Id}\|_2$  for RK4. Finally, the bottom row shows  $\log_{10} |\det y_n - 1|$  for RK4 and GL4.

to nonlinear equations, and an effective approximation of the exponential which, respecting Lie groups, does not negate the most important feature of Magnus series, their retention of Lie-group structure. We hope to return to these themes in future papers.

We thank Brad Baxter, Hermann Brunner, Joe Keller, Yunkang Liu, Per-Christian Moan, Robert McLachlan, Hans Munthe-Kaas, Malcolm Sabin, John Todd, Ralph Wilcox and Antonella Zanna for helpful discussions. S.P.N. was supported by the UK Engineering and Physical Sciences Research Council and by the Norwegian Research Council.

## References

- Abramowitz, M. & Stegun, I. 1965 *Handbook of mathematical functions*. New York: Dover.
- Bellman, R. 1969 *Stability theory of differential equations*. New York: Dover.
- Brockett, R. W. 1976 Volterra series and geometric control theory. *Automatica* **12**, 167–176.
- Calvo, M. P., Iserles, A. & Zanna, A. 1996 Runge–Kutta methods on manifolds. In *Numerical analysis: A.R. Mitchell's 75th birthday volume* (ed. G. A. Watson & D. F. Griffiths), pp. 57–70. Singapore: World Scientific.
- Casas, F. 1996 Solution of linear partial differential equations by Lie algebraic methods. Technical report, Universitat Jaume I, Spain.

- Chen, K. T. 1957 Integration of paths, geometric invariants, and a generalized Baker–Hausdorff formula. *Annals Math.* **67**, 164–178.
- Comtet, L. 1970 *Analyse Combinatoire*. Presses Universitaires de France.
- Cools, R. 1997 Constructing cubature formulas: the science behind the art. *Acta Numerica* **6**, 1–54.
- Crouch, P. & Grossman, R. 1993 Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear Sci.* **3**, 1–33.
- Fer, F. 1958 Résolution de l'équation matricielle  $\dot{U} = pU$  par produit infini d'exponentielles matricielles. *Bull. Class. Sci. Acad. R. Belg.* **44**, 818–829.
- Harary, F. 1969 *Graph theory*. Reading, MA: Addison-Wesley.
- Hausdorff, F. 1906 Die symbolische Exponentialformel in der Gruppentheorie. *Berichte der Sächsischen Akademie der Wissenschaften (Math. Phys. Klasse)* **58**, 19–48.
- Iserles, A. 1984 Solving linear ordinary differential equations by exponentials of iterated commutators. *Numer. Math.* **45**, 183–199.
- Iserles, A. 1997 Numerical methods on (and off) manifolds. In *Foundations of computational mathematics* (ed. F. Cucker & M. Shub), pp. 180–189. New York: Springer.
- Iserles, A. & Zanna, A. 1996 Qualitative numerical analysis of ordinary differential equations. In *The mathematics of numerical analysis* (ed. J. Renegar, M. Shub & S. Smale), vol. 32 of *Lectures in Applied Mathematics*, pp. 421–442. Providence, RI: American Mathematical Society.
- Iserles, A. & Zanna, A. 1999 Numerical methods on differentiable manifolds. (In preparation.)
- Magnus, W. 1954 On the exponential solution of differential equations for a linear operator. *Commun. Pure Appl. Math.* **7**, 649–673.
- Munthe-Kaas, H. 1997 Runge–Kutta methods on Lie groups. Technical report 122, Department of Informatics, University of Bergen.
- Munthe-Kaas, H. & Owren, B. 1998 Computations in a free Lie algebra. Technical report, Norwegian University of Science & Technology, Trondheim.
- Munthe-Kaas, H. & Zanna, A. 1997 Numerical integration of differential equations on homogeneous manifolds. In *Foundations of computational mathematics* (ed. F. Cucker & M. Shub), pp. 305–315. Berlin: Springer.
- Olver, P. 1995 *Equivalence, invariants and symmetry*. Cambridge University Press.
- Onischik, A. L. (ed.) 1993 Lie groups and Lie algebras. In *Encyclopaedia of mathematical sciences*, vol. 20. Berlin: Springer.
- Otter, R. 1948 The number of trees. *Annals Math.* **49**, 583–599.
- Owren, B. & Marthinsen, A. 1997 Integration methods based on rigid frames. Technical report, Norwegian University of Science and Technology, Trondheim.
- Quispel, G. R. W. & Turner, G. S. 1996 Discrete gradient methods for solving ODEs numerically while preserving a first integral. Technical report, La Trobe University, Melbourne.
- Sussman, H. 1986 A product expansion of the Chen series. In *Theory and applications of nonlinear control systems* (ed. C. I. Byrnes & A. Lyndquist), pp. 323–335. New York: Elsevier.
- van Lint, J. H. & Wilson, R. M. 1992 *A course in combinatorics*. Cambridge University Press.
- Wilcox, R. 1972 Bounds for approximate solutions to the operator differential equation  $\dot{Y}(t) = M(t)Y(t)$ ; applications to Magnus expansions and to  $\ddot{u} + [1 + f(t)]u = 0$ . *J. Math. Anal. Appl.* **39**, 92–111.
- 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, UK.
- Zanna, A. & Munthe-Kaas, H. 1997 Iterated commutators, Lie's reduction method and ordinary differential equations on matrix Lie groups. In *Foundations of computational mathematics* (ed. F. Cucker & M. Shub). New York: Springer.

MATHEMATICAL,  
PHYSICAL  
& ENGINEERING  
SCIENCES

THE ROYAL  
SOCIETY

PHILOSOPHICAL  
TRANSACTIONS  
OF

MATHEMATICAL,  
PHYSICAL  
& ENGINEERING  
SCIENCES

THE ROYAL  
SOCIETY

PHILOSOPHICAL  
TRANSACTIONS  
OF