On the discretization of double-bracket flows

Arieh Iserles[1]

Abstract

This paper extends the method of Magnus series to Lie-algebraic equations originating in double-bracket flows. We show that the solution of the isospectral flow $Y' = [[Y,N],Y]$, $Y(0) = Y_0 \in \mathrm{Sym}(n)$, can be represented in the form $Y(t) = \mathrm{e}^{\Omega(t)} Y_0 \mathrm{e}^{-\Omega(t)}$, where the Taylor expansion of Ω can be constructed explicitly, term-by-term, identifying individual expansion terms with certain rooted trees with bicolour leaves. This approach is extended to other Lie-algebraic equations that can be appropriately expressed in terms of a finite 'alphabet'.

1 Introduction

The *double-bracket equations*

$$Y' = [[Y,N],Y], \quad t \geq 0, \qquad Y(0) = Y_0 \in \mathrm{Sym}(n), \tag{1.1}$$

where N is a given matrix in $\mathrm{Sym}(n)$, the set of $n \times n$ symmetric real matrices, have been introduced simultaneously by Brockett (1991) and Chu & Driessel (1990). They share a number of important attributes that make them applicable to a wide range of problems. The most important feature of (1.1) is that they are a special instance of an *isospectral flow*

$$Y' = [A(t,Y),Y], \quad t \geq 0, \qquad Y(0) = Y_0 \in \mathrm{Sym}(n), \tag{1.2}$$

where $A : \mathbb{R}_+ \times \mathrm{Sym}(n) \Rightarrow \mathfrak{so}(n)$, while $\mathfrak{so}(n)$ is the Lie algebra of $n \times n$ real skew-symmetric matrices. Therefore, for every $t \geq 0$ the matrix $Y(t)$ is orthogonally similar to Y_0: there exists a function Q, evolving in the Lie group $\mathrm{SO}(n)$ of $n \times n$ real orthogonal matrices with unit determinant, such that

$$Y(t) = Q(t) Y_0 Q^\top(t). \tag{1.3}$$

Second important characteristic of double-bracket equations is that they form a *gradient system* with respect to the Lyapunov function

$$\phi(X) = \|X - N\|_{\mathrm{F}},$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm and X ranges across all matrices $X \in \mathrm{Sym}(n)$ which are similar to Y_0. An immediate consequence is that, provided that Y_0 itself is not in the commutative Banach algebra generated by N (in which case $Y(t) \equiv Y_0$), a fixed point $\hat{Y} = \lim_{t\to\infty} Y(t)$ is a (local) minimiser of ϕ (Brockett 1991).

This minimality of (1.1) makes it relevant to a wide range of applications. The most obvious is computing eigenvalues of symmetric matrices: If N is a diagonal matrix with distinct diagonal elements then a minimiser of ϕ is itself a diagonal matrix, while

[1]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge CB3 9EW

being similar (hence, sharing eigenvalues) with Y_0. We hasten to add that, needless, to say, solving (1.1) is not a method of choice for eigenvalue calculations. Yet, it might be interesting in a number of specialised situations.

Other applications of (1.1) include sorting lists and solving linear programming problems (Brockett 1991) and, perhaps with greater relevance to practical linear-algebra computations, calculating inverse eigenvalue and least squares problems (Chu & Driessel 1990, Chu 1998). A special choice of N corresponds to a variant of the familiar Toda lattice equations (Bloch 1990).

An obvious means of discretising (1.2) is by standard numerical methods, e.g. Runge–Kutta or multistep. Unfortunately, for $n \geq 3$ this is bound to lead to the loss of the most important structural feature of the system, isospectrality: the numerical solution, in general, changes the eigenvalues (Calvo, Iserles & Zanna 1997). Since the recovery of isospectrality is often essential in applications, it is vital to discretise (1.2) with a method that respects this feature. Such methods, which have been recently introduced in (Calvo et al. 1997, Calvo, Iserles & Zanna 1999, Zanna 1998), share a common denominator: they all regard the *isospectral orbit* $\mathcal{I}_{Y_0} \subset \mathrm{Sym}(n)$ of all symmetric matrices similar to Y_0 as a *homogeneous space,* subjected to the transitive $\mathrm{SO}(n)$ action $\lambda(Q)X = QXQ^\top$. In other words, the solution of (1.2) is done by means of the representation (1.3) and, instead of computing Y at the first place, in each step an orthogonal matrix Q_{N+1} is evaluated and $Y((N+1)h)$ is approximated by $Y_{N+1} = Q_{N+1}(h)Y_0Q_{N+1}^\top(h)$. Clearly, this ensures that $Y_N \in \mathcal{I}_{Y_0}$ for all $N \in \mathbb{Z}_+$.

The evaluation of Q_{N+1} itself can be accomplished by observing that

$$Q'_{N+1} = A(t, Q_{N+1}Y_0Q_{N+1}^\top)Q_{N+1}, \quad t \geq Nh, \qquad Q_{N+1}(Nh) = \mathrm{I}. \tag{1.4}$$

The latter is an example of a *Lie-group flow*

$$X' = A(t, X)X, \quad t \geq 0, \qquad X(t_0) = X_0 \in \mathcal{G}, \tag{1.5}$$

where $A : \mathbb{R}_+ \times \mathcal{G} \to \mathfrak{g}$, $\mathcal{G}$ is a *Lie group* and $\mathfrak{g}$ its *Lie algebra.* This is a differential system that evolves in a Lie group $\mathcal{G}$ (in our particular case, $\mathcal{G} = \mathrm{SO}(n)$ and $\mathfrak{g} = \mathfrak{so}(n)$). We refer the reader to (Varadarajan 1984) for theory and terminology of Lie groups and to (Iserles, Munthe-Kaas, Nørsett & Zanna 2000) for an extensive survey of computational methods that evolve in Lie groups and homogeneous spaces. Observe that the entire purpose of the exercise would be entirely lost unless the numerical solution of (1.4) itself respects orthogonality, otherwise the congruence $Y_{N+1} = Q_{N+1}(h)Y_0Q_{N+1}^\top(h)$ is no longer a similarity transformation. (It is possible to solve (1.4) with any method, while replacing congruence with $Y_{N+1} = Q_{N+1}(h)Y_0Q_{N+1}^{-1}(h)$, but the drawback of this approach is that Y_{N+1} cannot be any longer guaranteed to evolve in $\mathrm{Sym}(n)$ – unless, there is, Q_{N+1} is itself orthogonal.)

No classical numerical methods can respect the Lie-group structure of (1.6) for general $\mathcal{G}$ (Iserles et al. 2000). In the case of orthogonal flows (1.4), exceptionally, symplectic Runge–Kutta methods preserve orthogonality (Dieci, Russell & van Vleck 1994). However, they are implicit, thereby expensive. An alternative is to use methods which respect the Lie-group *by design.* Most (but by no means all) such methods follow a set pattern. The solution of (1.5) is expressed in the form $X = \psi(\Omega)X_0$, where $\psi : \mathfrak{g} \to \mathcal{G}$ and, instead of solving (1.5), one solves the Lie-algebraic equation

$$\Omega' = \mathrm{d}\psi_\Omega A(t, \psi(\Omega)X_0), \quad t \geq t_0, \qquad \Omega(t_0) = O. \tag{1.6}$$

Here $\mathrm{d}\psi_\Omega$ is the *differential* of ψ with respect of Ω (Iserles et al. 2000).

Recall our point of departure, the isospectral flow (1.2). We converted it first to the Lie-group flow (1.4) and, in turn, translated the latter to an Lie-algebraic setting. In general, each time step in the solution of an equation $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$ which evolves in a homogeneous space $\mathcal{M}$ acted upon by a Lie group $\mathcal{G}$, by most Lie-group methods can be described diagrammatically in the form

$$\begin{tikzcd}
\mathcal{M} \arrow[d] & \boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}) & \mathcal{M} \\
\mathcal{G} \arrow[d] & X' = A(t, X)X & \mathcal{G} \arrow[u, "\text{group action}"'] \\
\mathfrak{g} \arrow[rr, "\Omega' = \mathrm{d}\psi_\Omega A(t, \psi(\Omega)X_0)"] & & \mathfrak{g} \arrow[u, "\psi"']
\end{tikzcd}$$

The most natural and universal map from finite-dimensional $\mathfrak{g}$ to $\mathcal{G}$ is the *matrix exponential* $\psi(X) = \mathrm{e}^X$, whereby (1.6) becomes the so-called *dexpinv equation,*

$$\Omega' = \mathrm{dexp}_\Omega^{-1} A(t, \mathrm{e}^\Omega X_0) = \sum_{m=0}^{\infty} \frac{\mathrm{B}_m}{m!} \mathrm{ad}_\Omega^m A(t, \mathrm{e}^\Omega X_0), \quad t \geq 0, \qquad \Omega(t_0) = O \tag{1.7}$$

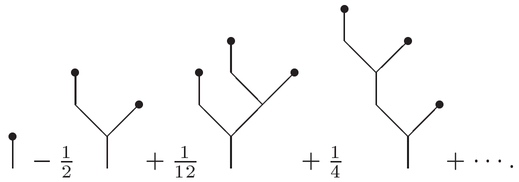
(Iserles et al. 2000). Here B_m is the mth *Bernoulli number,* while the *adjoint operator* ad_Ω is an iterated commutator,

$$\mathrm{ad}_\Omega^0 A = A, \qquad \mathrm{ad}_\Omega^{m+1} A = [\Omega, \mathrm{ad}_\Omega^m A], \quad m \in \mathbb{Z}_+.$$

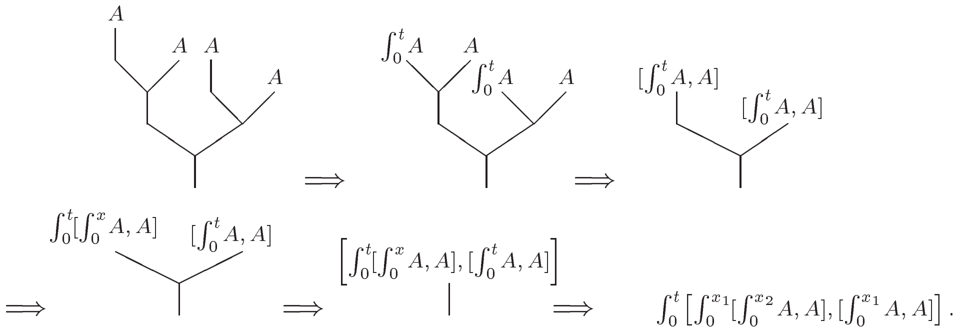
Since a Lie algebra is nothing more than a linear space closed under commutation, any numerical solution of (1.7) (with appropriately truncated series) that employs solely linear combinations and commutators is bound to stay in $\mathfrak{g}$. A particularly attractive option, introduced by Munthe-Kaas (1998), is to solve (1.7) with an explicit Runge–Kutta method. Insofar as nonlinear equations (1.5) are concerned, this is the most natural means of computing the dexpinv equation (1.7). Yet, if $A = A(t)$ and (1.5) is linear, Magnus (1954) showed that Ω can be expanded as an infinite linear combination of multiple integrals over iterated commutators,

$$\begin{aligned}
\Omega(t) = \int_0^t A(x)\mathrm{d}x &- \tfrac{1}{2} \int_0^t \int_0^{x_1} [A(x_2), A(x_1)] \mathrm{d}x_2 \mathrm{d}x_1 \\
&+ \tfrac{1}{12} \int_0^t \int_0^{x_1} \int_0^{x_1} [A(x_3), [A(x_2), A(x_1)]] \mathrm{d}x_3 \mathrm{d}x_2 \mathrm{d}x_1 \\
&+ \tfrac{1}{4} \int_0^t \int_0^{x_1} \int_0^{x_2} [[A(x_3), A(x_2)], A(x_1)] \mathrm{d}x_3 \mathrm{d}x_2 \mathrm{d}x_1 + \cdots .
\end{aligned} \tag{1.8}$$

The *Magnus expansion* (1.8) has received a very great deal of attention in applied mathematics, but only in the last few years a concerted effort has been undertaken to fashion it into an effective numerical tool. In particular, Iserles & Nørsett (1999) associated expansion terms with *binary trees,* thereby presenting a convenient recursive algorithm for the successive derivation of Magnus terms. To elucidate this association, let us examine first the graph-theoretic notation for (1.8),



To obtain a term in $\mathfrak{g}$ from a binary tree, we proceed as follows. First label every leaf (a 'black node' above) with the function $A(t)$. Next, prune the tree according to the following rule: if a leaf is a lone child, integrate its label. If two leaves share a parent, commute their labels. Thus, for example,



The coefficients in (1.8) can be also obtained recursively. We note in passing that (Iserles & Nørsett 1999) and more recent publications contain a wealth of further material, e.g. on efficient numerical approximation of multivariate integrals in the Magnus expansion. The reader is referred to (Iserles et al. 2000) for a review of analytic and numerical features of Magnus expansions which are of lesser relevance to this paper.

Once we contemplate the computation of *nonlinear* Lie-algebraic equations originating in the double-bracket flow (1.1), 'plain' Magnus expansions are no longer relevant. Although some work has been done to extend (1.8) to nonlinear setting (Zanna 1999), ensuing methods are highly implicit, thus not very competitive in comparison with, say explicit Runge–Kutta–Munthe-Kaas schemes. On the face of it, Magnus expansions are irrelevant to the computation of double-bracket flows.

Yet, we can adopt an entirely different point of view. Consider the dexpinv equation originating in the double-bracket flow,

$$\Omega' = \mathrm{dexp}_{\Omega}^{-1}[\mathrm{Ad}_{e^{\Omega}} Y_0, N], \quad t \geq 0, \qquad \Omega(t_0) = O, \tag{1.9}$$

where $\mathrm{Ad}_B A = BAB^{-1}$ is the adjoint operator in a Lie group. *The entire information necessary for the determination of Ω consists of just two matrices, N and Y_0!* In other words, each term in the Taylor expansion of Ω can be obtained from just $N, Y_0 \in \mathrm{Sym}(n)$ in a finite number of linear combinations and commutators. If we can only determine the precise rules underpinning this process, the outcome will be a viable and explicit numerical procedure for the approximation of Ω to arbitrary accuracy. This is the goal of the present paper.

The procedure leading towards a constructive Taylor expansion of Ω is similar to the Magnus expansion and it employs very similar terminology of binary rooted trees. Having said this, it is much more delicate and, unless intermediate steps are highlighted, perhaps counterintuitive. Thus, we proceed in the sequel in three steps. Firstly, in Section 2 we assemble one-by-one the building blocks necessary to derive the Magnus expansion of double-bracket flows. In Section 3 all this information is assembled into a well-defined, convergent numerical algorithm. Finally, in Section 4 we present two generalisations of our approach to other Lie-algebraic equations originating in homogeneous spaces or Lie groups. Firstly, we indicate how our technique can be generalised to a time-dependent matrix $N(t)$. Secondly, we consider the case (1.7) when the matrix function A depends in an appropriate fashion on a finite number of constant matrices. One example originates in the *double double-bracket flow* $X' = [X, [Y, X]]$, $Y' = [Y, [Y, X]]$ (Bloch & Crouch 1996, Bloch, Brockett & Crouch 1997), except that in this case we can write explicitly the exact solution. Another example is the *triple-bracket equation* $Y' = [[[Y, M], [Y, N]], Y]$. We present an extension of our algorithm that can generate, term-by-term, the Taylor expansion of Ω in all these cases.

2 Elements of the Magnus expansion for (1.9)

Since $\mathrm{Ad}_{\mathrm{e}^\Omega} A = \mathrm{e}^{\mathrm{ad}_\Omega} A$, where $\mathrm{e}^{\mathrm{ad}_\Omega} = \sum_{m=0}^{\infty} \frac{1}{m!} \mathrm{ad}_\Omega^m$ (Varadarajan 1984), we rewrite (1.9) in a form more consistent with (1.7),

$$\Omega' = \mathrm{dexp}_\Omega^{-1} [\mathrm{e}^{\mathrm{ad}_\Omega} Y_0, N], \quad t \ge 0, \qquad \Omega(t_0) = O. \tag{2.1}$$

We seek to expand the solution of (2.1) in Taylor series of the form

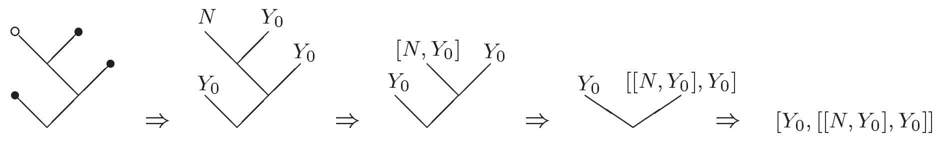
$$\Omega(t) = \sum_{m=1}^{\infty} t^m \sum_{\tau \in \mathbb{F}_m} \alpha(\tau) H_\tau. \tag{2.2}$$

The *index set* $\mathbb{F}_m$ is composed out of suitable rooted *strictly binary* trees, α is a map from the set $\mathbb{F} = \bigcup_{m \in \mathbb{N}} \mathbb{F}_m$ to $\mathbb{R}$ and the term H_τ is constructed from Y_0 and N by nested commutation.

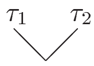
We colour the leaves of $\tau \in \mathbb{F}$ with two colours, black and white:

$$\bullet \rightsquigarrow Y_0, \qquad \circ \rightsquigarrow N.$$

The trees are pruned exactly like 'plain' Magnus trees, for example



Formally, once H_{τ_l} has been computed for $l = 1, 2$, we let



$$\rightsquigarrow \quad [H_{\tau_1}, H_{\tau_2}]. \tag{2.3}$$

The sets $\mathbb{F}_m$ and the coefficient map α will be determined later. For the time being we use (2.2) as a purely formal construct.

We define

$$V_r(t) := \mathrm{ad}_\Omega^r Y_0, \qquad r \in \mathbb{Z}_+,$$
$$U_s(t) := \mathrm{ad}_\Omega^s [\mathrm{e}^{\mathrm{ad}_\Omega} Y_0, N], \qquad s \in \mathbb{Z}_+.$$

Since the Taylor expansions of each V_r and U_s can be also obtained as linear combinations of nested commutators, we formally stipulate that

$$V_r(t) = \sum_{k=r}^{\infty} t^k \sum_{\tau \in \mathbb{V}_{r,k}} \beta(\tau) H_\tau,$$
$$U_s(t) = \sum_{l=s}^{\infty} t^l \sum_{\tau \in \mathbb{U}_{s,l}} \gamma(\tau) H_\tau.$$

Here, again, H_τ can be computed recursively via (2.3), while $\beta(\tau)$ and $\gamma(\tau)$ are scalars. The index sets $\mathbb{V}_{r,k}$ and $\mathbb{U}_{s,l}$ are, for the time being, purely formal collections of rooted binary trees. Note that the summations commence from $k = r$ and $l = s$ respectively since $\Omega(t) = \mathcal{O}(t)$ implies that $\mathrm{ad}_\Omega^m B = \mathcal{O}(t^m)$, $m \in \mathbb{Z}_+$.

We next proceed to enquire into the functions V_r. Our goal is to establish recursive rules to form the sets $\mathbb{V}_{r,k}$, $k \geq r$, and the coefficient map β. Since $V_0 = Y_0$, we deduce at once that

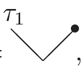
$$\mathbb{V}_{0,0} = \{\bullet\}, \qquad \mathbb{V}_{0,k} = 0, \quad k \in \mathbb{N}, \qquad \beta(\bullet) = 1.$$

Thus, formally substituting (2.2),

$$V_1(t) = [\Omega(t), V_0(t)] = \sum_{m=0}^{\infty} t^m \sum_{\tau \in \mathbb{F}_m} \alpha(\tau)[H_\tau, Y_0].$$

Therefore, for every $k \geq 1$,

$$\tau \in \mathbb{V}_{1,k} \qquad \Leftrightarrow \qquad \tau =$$



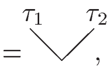
$$, \quad \text{where} \quad \tau_1 \in \mathbb{F}_k.$$

Moreover, in that case $\beta(\tau) = \alpha(\tau_1)$.

With greater generality, suppose that we have already determined (up to the yet-unknown expansion (2.2)) the sets $\mathbb{V}_{m,k}$ and the underlying coefficient map β for $m \leq r - 1$ and $k \geq m$. Since $V_r = \mathrm{ad}_\Omega V_{r-1} = [\Omega, V_{r-1}]$, we deduce that

$$\begin{aligned} V_r(t) &= \left[\sum_{m=0}^{\infty} t^m \sum_{\tau_1 \in \mathbb{F}_m} \alpha(\tau_1) H_{\tau_1}, \sum_{k=r-1}^{\infty} t^k \sum_{\tau_2 \in \mathbb{V}_{r-1.k}} \beta(\tau_2) H_{\tau_2} \right] \\ &= \sum_{m=0}^{\infty} \sum_{k=r-1}^{\infty} t^{m+k} \sum_{\tau_1 \in \mathbb{F}_m} \sum_{\tau_2 \in \mathbb{V}_{r-1.k}} \alpha(\tau_1)\beta(\tau_2)[H_{\tau_1}, H_{\tau_2}]. \end{aligned}$$

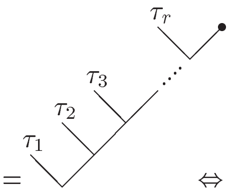
Therefore

$$\tau \in \mathbb{V}_{r,k} \qquad \Leftrightarrow \qquad \tau =$$  $$, \quad \text{where} \quad \tau_1 \in \mathbb{F}_{k_1},\ \tau_2 \in \mathbb{V}_{r-1,k_2},\ k_1 + k_2 = k. \tag{2.4}$$

In that case

$$\beta(\tau) = \alpha(\tau_1)\beta(\tau_2). \tag{2.5}$$

An easy inductive argument, in tandem with the fact that $V_0 = Y_0$, can be now used to express (2.4) and (2.5) exclusively in terms of trees from $\mathbb{F}_1, \mathbb{F}_2, \ldots, \mathbb{F}_{r-1}$:

$$\mathbb{V}_{r,k} \ni \tau =$$  $$\Leftrightarrow \qquad \tau_l \in \mathbb{F}_{k_l},\ l = 1, 2, \ldots, r, \ \sum_{l=1}^{r} k_l = k, \tag{2.6}$$

$$\beta(\tau) = \prod_{l=1}^{r} \alpha(\tau_l). \tag{2.7}$$

Having expressed the V_rs in terms of the set $\mathbb{F}$, we proceed to do the same for the functions U_s. We commence by noting that

$$\mathrm{e}^{\mathrm{ad}_\Omega} Y_0 = \sum_{r=0}^{\infty} \frac{1}{r!} \mathrm{ad}_\Omega^r Y_0 = \sum_{r=0}^{\infty} \frac{1}{r!} V_r.$$

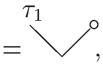
Therefore

$$U_0(t) = [\mathrm{e}^{\mathrm{ad}_{\Omega(t)}} Y_0, N] = \sum_{r=0}^{\infty} \frac{1}{r!} [V_r(t), N] = \sum_{l=0}^{\infty} t^l \sum_{r=0}^{l} \frac{1}{r!} \sum_{\tau \in \mathbb{V}_{r,l}} \beta(\tau)[H_\tau, N].$$

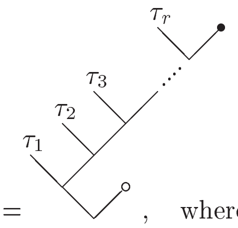
Hence,

$$\sum_{\tau \in \mathbb{U}_{0,l}} \gamma(\tau) H_\tau = \sum_{r=0}^{l} \frac{1}{r!} \sum_{\tau \in \mathbb{V}_{r,l}} \beta(\tau)[H_\tau, N].$$

We thus deduce that for every $l \in \mathbb{Z}_+$

$$\mathbb{U}_{0,l} \ni \tau =$$  $$, \quad \text{where} \quad \tau_1 \in \bigcup_{r=0}^{l} \mathbb{V}_{r,l} \qquad \text{and} \qquad \gamma(\tau) = \frac{1}{r!} \beta(\tau_1).$$

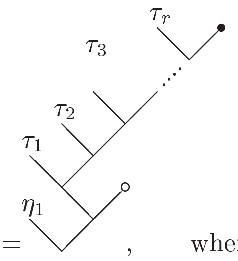
Note for future reference that the above can be reformulated as

$$\mathbb{U}_{0,l} \ni \tau =$$  $$, \quad \text{where} \quad \tau_i \in \mathbb{F}_{k_i},\ \sum_{i=1}^{r} k_i = l,\ \beta(\tau) = \frac{1}{r!}\prod_{i=1}^{r}\alpha(\tau_i). \tag{2.8}$$

We continue as before. Thus,

$$U_1(t) = [\Omega(t), U_0(t)] = \sum_{m=0}^{\infty}\sum_{l=0}^{\infty} t^{m+l} \sum_{\tau_1 \in \mathbb{F}_m}\ \sum_{\tau_2 \in \mathbb{U}_{0,l}} \alpha(\tau_1)\gamma(\tau_2)[H_{\tau_1}, H_{\tau_2}],$$

therefore

$$\mathbb{U}_{1,l} \ni \tau =$$  $$, \qquad \text{where} \qquad \eta_1 \in \mathbb{F}_{q_1},\ \tau_i \in \mathbb{F}_{k_i},\ q_i + \sum_{i=1}^{r} k_i = l$$

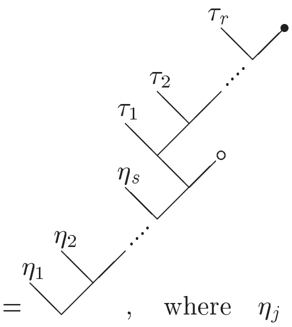
and

$$\gamma(\tau) = \frac{1}{r!}\alpha(\tau_1)\prod_{i=1}^{r}\alpha(\tau_1).$$

Likewise,

$$U_s(t) = [\Omega(t), U_{s-1}(t)], \qquad s \in \mathbb{N},$$

and proceeding like before we deduce that

$$\mathbb{U}_{s,l} \ni \tau =$$  $$, \quad \text{where} \quad \eta_j \in \mathbb{F}_{q_j},\ \tau_i \in \mathbb{F}_{k_i},\ \sum_{j=1}^{s} q_j + \sum_{i=1}^{r} k_i = l,$$
$$\gamma(\tau) = \frac{1}{r!}\prod_{j=1}^{s}\alpha(\eta_j)\prod_{i=1}^{r}\alpha(\tau_i). \tag{2.9}$$

Having derived U_s, $s \in \mathbb{Z}_+$, we note that

$$\mathrm{dexp}_\Omega^{-1}[\mathrm{e}^{\mathrm{ad}_\Omega} Y_0, N] = \sum_{s=0}^{\infty} \frac{\mathrm{B}_s}{s!} U_s.$$

Integrating, we deduce from the dexpinv equation (2.1) that

$$\begin{aligned}\Omega(t) &= \sum_{s=0}^{\infty} \frac{\mathrm{B}_s}{s!} \int_{t_0}^{t} U_s(x)\mathrm{d}x = \sum_{s=0}^{\infty} \frac{\mathrm{B}_s}{s!} \sum_{m=s}^{\infty} \frac{t^{m+1}}{m+1} \sum_{\tau \in \mathbb{U}_{s,m}} \gamma(\tau) H_\tau \\ &= \sum_{m=1}^{\infty} \frac{t^m}{m} \sum_{s=0}^{m-1} \frac{\mathrm{B}_s}{s!} \sum_{\tau \in \mathbb{U}_{s,m-1}} \gamma(\tau) H_\tau. \end{aligned} \tag{2.10}$$

3 The Magnus algorithm for double-bracket flows

3.1 The algorithm

We have two representations of the unknown function Ω, namely the Taylor series (2.2) and (2.10). Although the latter representation has been constructed formally from the first, a minor mismatch in indices allows us to develop a recursive procedure for the construction of Ω. The mismatch in question is that, while in (2.2) the inner summation is in $\mathbb{F}_m$, in (2.10) it extends across $\mathbb{U}_{s,m-1}$.

Let us compare the t^m terms in both representations,

$$\sum_{\tau \in \mathbb{F}_m} \alpha(\tau) H_\tau = \frac{1}{m} \sum_{s=0}^{m-1} \frac{\mathrm{B}_s}{s!} \sum_{\tau \in \mathbb{U}_{s,m-1}} \gamma(\tau) H_\tau, \qquad m \in \mathbb{N}. \tag{3.1}$$

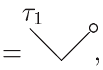
We thus deduce that

$$\mathbb{F}_m = \bigcup_{s=0}^{m-1} \mathbb{U}_{s,m-1}, \qquad m \in \mathbb{N}, \tag{3.2}$$

Let $\tau \in \mathbb{F}_m$, $m \in \mathbb{N}$. Then, according to the above, there exists $s \in \{0, 1, \ldots, m-1\}$ such that $\tau \in \mathbb{U}_{s,m-1}$. We deduce from (3.1) that

$$\alpha(\tau) = \frac{1}{m} \cdot \frac{\mathrm{B}_s}{s!} \gamma(\tau). \tag{3.3}$$

To launch the recursive algorithm, we consider first the case $m = 1$. Since, by (3.2), $\mathbb{F}_1 = \mathbb{U}_{0,0}$, we employ (2.8) to observe that

$\mathbb{U}_{0,0} \ni \tau =$ 

$, \quad \tau_1 \in \mathbb{V}_{0,0}, \qquad \gamma(\tau) = \beta(\tau_1).$

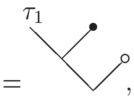
But $\mathbb{V}_{0,0} = \{\bullet\}$ and $\beta(\bullet) = 1$, therefore

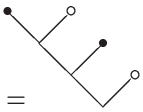
$\mathbb{F}_1 = \mathbb{U}_{0,0} = \Big\{$  $\Big\}, \qquad \alpha\Big($  $\Big) = 1.$

Before we present the general pattern for recursion, we progress in detail one more step, deriving $\mathbb{F}_2$. According to (3.2),

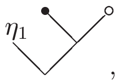
$$\mathbb{F}_2 = \mathbb{U}_{0,1} \cup \mathbb{U}_{1,1}.$$

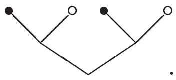
We now use (2.9) and our knowledge of $\mathbb{F}_1$. To obtain $\mathbb{U}_{0,1}$ we note that the unique partition $\sum_{i=1}^{r} k_i = 1$ is $r = 1$, $k_1 = 1$. Hence every term in $\mathbb{U}_{0,1}$ is of the form

$\tau =$ 

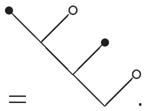
, $\tau_1 \in \mathbb{F}_1$, hence $\tau =$ 

and $\beta(\tau) = 1$. Insofar as $\mathbb{U}_{1,1}$ is concerned, we note that there is just one partition $q_1 + \sum_{i=1}^{r} k_i = 1$, namely $q_1 = 1$ and $r = 0$. Therefore the only element in $\mathbb{U}_{1,1}$ is of the form



, $\eta_1 \in \mathbb{F}_1$, hence  .

It is trivial, though, to observe that this tree corresponds to a commutator of a term with itself, which must be zero. Hence, we need not consider it and deduce that $\mathbb{F}_2$ includes just one element,

$\tau =$  .

Moreover, according to (3.3), $\alpha(\tau) = \frac{1}{2}$.

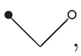
In general, suppose that $\mathbb{F}_1, \mathbb{F}_2, \ldots, \mathbb{F}_{m-1}$ are known, as is the coefficient map α in these sets. To construct $\mathbb{F}_m$ we note that every τ therein is of the form (2.9) for some $s \in \{0, 1, \ldots, m-1\}$ and $r \in \{0, 1, \ldots, m-1\}$. For every such s and r we consider all *partitions*

$$q_1 + q_2 + \cdots + q_s + k_1 + k_2 + \cdots + k_r = m - 1$$

(empty sums are treated in a self-evident manner: if $s = 0$, say, this reduces to $k_1 + k_2 + \cdots + k_r = m - 1$). For every partition we construct the tree $\tau \in \mathbb{U}_{s,m-1}$ in (2.9) and, finally, use (3.3) to determine the coefficient map $\alpha(\tau)$. Along the way, it is a sound policy to eliminate trees that correspond to zero terms (as we have seen for $m = 2$). Moreover, some trees can be replaced by linear combinations of other trees, a subject to which we will return in the sequel.

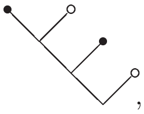
We now proceed through the first few steps of the recursive algorithm, denoting partitions in the form $\langle q_1, \ldots, q_s; k_1, \ldots, k_r \rangle$, with $\langle -; k_1, \ldots, k_r \rangle$ and $\langle q_1, \ldots, q_s; - \rangle$ for $s = 0$ and $r = 0$ respectively. For completeness, we include the derivation of $\mathbb{F}_1$ and $\mathbb{F}_2$.

1. $\mathbb{F}_1$: We have deduced from the partition $\langle -; - \rangle$ that $\mathbb{F}_1 = \{\tau\}$, where

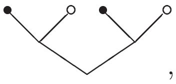
   $\tau_1^1 =$  , $\alpha(\tau_1^1) = 1.$

2. $\mathbb{F}_2$:

(a) $\langle -; 1 \rangle$:

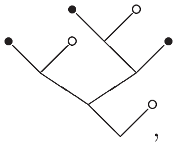
$\tau_1^2 =$  , $\qquad \alpha(\tau_1^2) = \frac{1}{2}$;

(b) $\langle 1; - \rangle$:

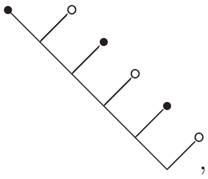
$\tau_2^2 =$  , $\qquad$ zero tree, discard.

3. $\mathbb{F}_3$:

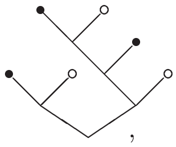
(a) $\langle -; 1, 1 \rangle$:

$\tilde{\tau}_1^3 =$  , $\qquad \alpha(\tilde{\tau}_1^3) = \frac{1}{3} \cdot \frac{1}{2} \cdot 1^2 = \frac{1}{6}$;

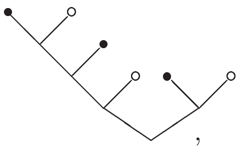
(b) $\langle -; 2 \rangle$:

$\tilde{\tau}_2^3 =$  , $\qquad \alpha(\tilde{\tau}_2^3) = \frac{1}{3} \cdot \frac{1}{1} \cdot \frac{1}{2} = \frac{1}{6}$;

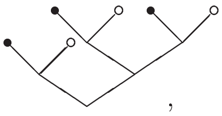
(c) $\langle 1; 1 \rangle$:

$\tilde{\tau}_3^3 =$  , $\qquad \alpha(\tilde{\tau}_3^3) = \frac{1}{3} \cdot \frac{1}{1} \cdot (-\frac{1}{2}) \cdot 1^2 = -\frac{1}{6}$;

(d) $\langle 2; - \rangle$:

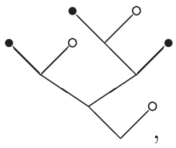
$\tilde{\tau}_4^3 =$  , $\qquad \alpha(\tilde{\tau}_4^3) = \frac{1}{3} \cdot \frac{1}{1} \cdot (-\frac{1}{2}) \cdot \frac{1}{2} = -\frac{1}{12}$;

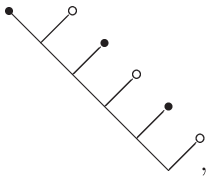
(e) $\langle 1, 1; - \rangle$:

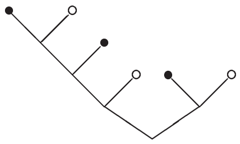
$\tilde{\tau}_5^3 =$  , $\qquad$ zero tree, discard.

Before we continue further, let us pause to tidy up the set $\mathbb{F}_3$. Recalling the interpretation of trees as corresponding to nested commutators, it is clear that rotating any nontrivial sub-tree about its root changes the sign of the underlying commutator.

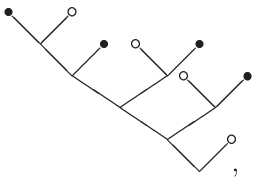
Hence $\tilde{\tau}_3^3$ is nothing but $\tilde{\tau}_4^3$ with the sign of the coefficient map reversed. The two trees can be just aggregated into $\tilde{\tau}_4^3$, say, with the coefficient map replaced by $\alpha(\tilde{\tau}_4^3)-\alpha(\tilde{\tau}_3^3)$. After further trivial rotations, we obtain just three 'generic' trees in $\mathbb{F}_3$,

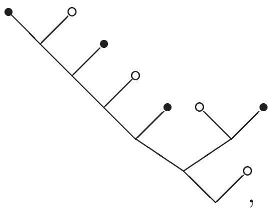
$\tau_1^3 =$  , $\alpha(\tau_1^3) = \frac{1}{6}$,

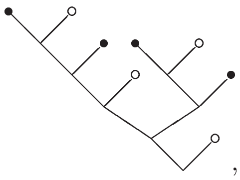
$\tau_2^3 =$  , $\alpha(\tau_2^3) = \frac{1}{6}$;

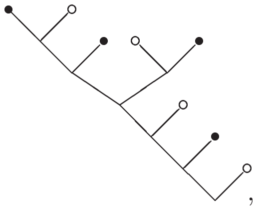
$\tau_3^3 =$  $\alpha(\tau_3^3) = \frac{1}{12}$.

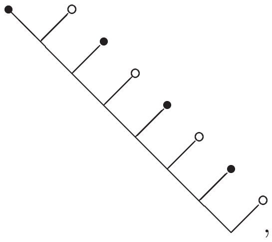
For $\mathbb{F}_4$ we note that there are twelve partitions. Since there are now three elements in $\mathbb{F}_3$, we can expect sixteen elements in $\mathbb{F}_4$, since each of the partitions $\langle -;3\rangle$ and $\langle 3;-\rangle$ corresponds to three possible choices. Altogether, after 'beautifying' rotations, we have

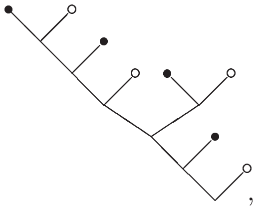
$\langle -;1,1,1\rangle :$ $\tilde{\tau}_1^4 =$  , $\alpha(\tilde{\tau}_1^4) = \frac{1}{24}$,

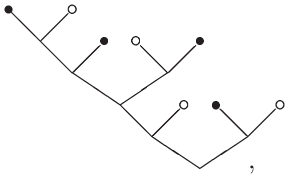
$\langle -;1,2\rangle :$ $\tilde{\tau}_2^4 =$  , $\alpha(\tilde{\tau}_2^4) = -\frac{1}{16}$,

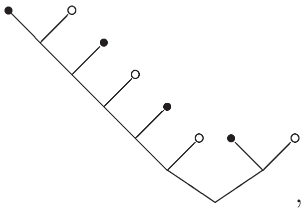
$\langle -;2,1\rangle :$ $\tilde{\tau}_3^4 =$  , $\alpha(\tilde{\tau}_3^4) = \frac{1}{16}$,

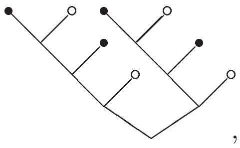
$\langle -;3\rangle:$ $\tilde{\tau}_4^4 =$  , $\alpha(\tilde{\tau}_4^4) = -\frac{1}{24}$,

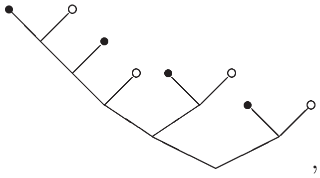
$\tilde{\tau}_5^4 =$  , $\alpha(\tilde{\tau}_5^4) = \frac{1}{24}$,

$\tilde{\tau}_6^4 =$  , $\alpha(\tilde{\tau}_6^4) = \frac{1}{48}$,

$\langle 1;1,1\rangle:$ $\tilde{\tau}_7^4 =$  , $\alpha(\tilde{\tau}_7^4) = \frac{1}{16}$,

$\langle 1;2\rangle:$ $\tilde{\tau}_8^4 =$  , $\alpha(\tilde{\tau}_8^4) = \frac{1}{16}$,

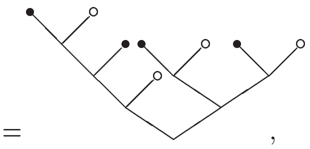
$\langle 2;1\rangle:$ $\tilde{\tau}_9^4 =$  , zero tree, discard,

$\langle 1,1;1\rangle:$ $\tilde{\tau}_{10}^4 =$  , $\alpha(\tilde{\tau}_{10}^4) = \frac{1}{48}$,

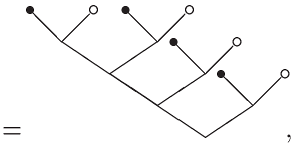
$\langle 3; - \rangle :$ $\tilde{\tau}^4_{11} = \tilde{\tau}^4_7, \quad \alpha(\tilde{\tau}^4_{11}) = 0,$ zero coefficient, discard,

$\tilde{\tau}^4_{12} = \tilde{\tau}^4_8, \quad \alpha(\tilde{\tau}^4_{12}) = 0,$ zero coefficient, discard,

$\tilde{\tau}^4_{13} = \tilde{\tau}^4_{10}, \quad \alpha(\tilde{\tau}^4_{13}) = 0,$ zero coefficient, discard,

$\langle 2,1; - \rangle :$ $\tilde{\tau}^4_{14} =$  , $\alpha(\tilde{\tau}^4_{14}) = 0,$ zero tree, discard,

$\langle 1,2; - \rangle :$ $\tilde{\tau}^4_{15} = \tilde{\tau}^4_{10}, \quad \alpha(\tilde{\tau}^4_{15}) = \frac{1}{96},$

$\langle 1,1,1; - \rangle :$ $\tilde{\tau}^4_{16} =$  , zero tree, discard.

In practice, the Taylor expansion (2.2) is truncated to attain an approximation of given order of accuracy. Rendering everything in terms of commutators, the Taylor expansion of $\Omega(t)$ is

$$
\begin{aligned}
& t[Y_0,N] \ldots\ldots\ldots\ldots \text{order } 1 \\
& + \tfrac{1}{2}t^2[[[Y_0,N],Y_0],N] \ldots\ldots\ldots\ldots \text{order } 2 \\
& + t^3\{-\tfrac{1}{6}[[[[Y_0,N],Y_0],[Y_0,N]],N] + \tfrac{1}{6}[[[[[Y_0,N],Y_0],N],Y_0],N] \\
& + \tfrac{1}{12}[[[[Y_0,N],Y_0],N],[Y_0,N]]\} \ldots\ldots\ldots\ldots \text{order } 3 \\
& + t^4\{\tfrac{1}{24}[[[[[Y_0,N],Y_0],[Y_0,N]],[Y_0,N]],N] - \tfrac{1}{16}[[[[[[Y_0,N],Y_0],N],Y_0],[Y_0,N]],N] \\
& + \tfrac{1}{16}[[[[[Y_0,N],Y_0],N],[[Y_0,N],Y_0]],N] - \tfrac{1}{24}[[[[[[Y_0,N],Y_0],[Y_0,N]],N],Y_0],N] \\
& + \tfrac{1}{24}[[[[[[[Y_0,N],Y_0],N],Y_0],N],Y_0],N] + \tfrac{1}{48}[[[[[[Y_0,N],Y_0],N],[Y_0,N]],Y_0],N] \\
& + \tfrac{1}{16}[[[[[Y_0,N],Y_0],[Y_0,N]],N],[Y_0,N]] + \tfrac{1}{16}[[[[[[Y_0,N],Y_0],N],Y_0],N],[Y_0,N]] \\
& + \tfrac{1}{32}[[[[[Y_0,N],Y_0],N],[Y_0,N]],[Y_0,N]]\} \ldots\ldots\ldots\ldots \text{order } 4 \\
& + \cdots
\end{aligned}
$$

3.2 Graded Lie algebras

Note that two mechanisms are at play, each producing trees that can be disregarded in our recursion: firstly, it might be possible that $H_\tau \equiv 0$ and, secondly, occasionally $\alpha(\tau) = 0$. The latter case occurs quite often because $\mathrm{B}_{2s+1} = 0$ for $s \in \mathbb{N}$. Altogether, discarding both kinds of trees and exploiting $\tilde{\tau}^4_{15} = \tilde{\tau}^4_{10}$, just nine trees in $\mathbb{F}_4$ survive.

Actually, we can reduce the membership of $\mathbb{F}_4$ to just eight trees, since one of the H_τs can be expressed as a linear combination of others. This can be done adopting the technique of *graded free Lie algebras,* which has been pioneered in this context by Munthe-Kaas & Owren (1999). In our case we have a free Lie algebra $\mathscr{F}$ with two generators, $\{Y_0, N\}$. Allocating to each generator a unit grade and propagating grades in the usual way, $\nu([X_1, X_2]) = \nu(X_1) + \nu(X_2)$, where $\nu(X_i)$ is the grade of X_i, the Witt–Birkhoff formula gives the dimension of the linear space spanned by all terms of given grade (Bourbaki 1975). This, however, although immensely useful in reducing the computational cost for standard Magnus expansions (Iserles et al. 2000), falls short of being a sufficiently powerful tool to help in reducing the number of trees that occur in the expansion of Ω. Examine the trees in $\mathbb{F}_m$, $m = 1, 2, 3, 4$. They all seem to share a common feature: the colour of their leaves alternates.

We say that a rooted tree with bicolour leaves is *chequerboard* if it has an even number of leaves, the colours of its leaves alternate and the rightmost leaf is black.

Lemma 1 *All the trees in $\mathbb{F}_m$, $m \in \mathbb{N}$, are chequerboard and possess $2m$ leaves.*

Proof An inductive proof follows at once from the form of $\mathbb{F}_1$, the recursion (2.9) and the formula (3.2). $\square$

Consider now the free Lie algebra $\mathscr{F}$ and denote by $\mathscr{K}_m^\circ$ the linear space of all grade-$2m$ terms that correspond to checkerboard trees. An immediate consequence of Lemma 1 is that $H_\tau \in \mathscr{K}_m^\circ$ for all $\tau \in \mathbb{F}_m$. Thus, $\chi_m := \dim \mathscr{K}_m^\circ$ sets an upper bound on the number of trees that correspond to linearly-independent terms.

Suppose for greater generality that a free Lie algebra is generated by ν 'letters' $\{X_1, X_2, \ldots, X_\nu\}$, each with unit grade. Let $\boldsymbol{s} = (s_1, s_2, \ldots, s_\nu) \in \mathbb{Z}_+^\nu$, $|\boldsymbol{s}| = \sum_{i=1}^{\nu} s_i \geq 1$, and consider the linear space $\mathscr{K}(\boldsymbol{s})$ of all $|\boldsymbol{s}|$-grade terms with exactly s_i occurences of X_i, $i = 1, 2, \ldots, \nu$. Then

$$\dim \mathscr{K}(\boldsymbol{s}) = \frac{1}{|\boldsymbol{s}|} \sum_{d|\boldsymbol{s}} \mu(d) \binom{|\boldsymbol{s}|/d}{\boldsymbol{s}/d}, \tag{3.4}$$

where μ is the *Möbius function* and $d|\boldsymbol{s}$ means that $d \in \mathbb{N}$ is a divisor of all the integers $s_1, s_2, \ldots, s_\nu$ (Bourbaki 1975). Specialising to our situation, for $\mathscr{K}_m^\circ$ we have $\nu = 2$, $s_1 = s_2 = m$, therefore (3.4) yields

$$\chi_m = \frac{1}{2m} \sum_{d|m} \mu(d) \binom{2m/d}{m/d}. \tag{3.5}$$

The first ten χ_ms are

$$1, 1, 3, 8, 25, 75, 245, 800, 2700, 9225.$$

Thus, the first apparent saving occurs in $\mathbb{F}_4$, where we have at least one redundant tree. It is possible to derive a *Hall* or *Lyndon basis* of $\mathscr{K}_m^\circ$ algorithmically (Bourbaki 1975), but the outcome no longer consists of chequerboard trees. Alternatively, we just observe directly that

$$H_{\tilde{\tau}_6^4} = -H_{\tilde{\tau}_2^4} - H_{\tilde{\tau}_3^4}. \tag{3.6}$$

To this end let

$$P = H_{\tau_1^1} = [Y_0, N], \qquad Q = H_{\tau_1^2} = [[[Y_0, N], Y_0], N]$$

and observe that

$$\begin{aligned} H_{\tilde{\tau}_2^4} &= -[[[Q, Y_0], P], N], \\ H_{\tilde{\tau}_3^4} &= [[[Y_0, P], Q], N], \\ H_{\tilde{\tau}_6^4} &= [[[Q, P], Y_0], N]. \end{aligned}$$

We use the *Jacobi identity,*

$$[[\kappa_1, \kappa_2], \kappa_3] + [[\kappa_2, \kappa_3], \kappa_1] + [[\kappa_3, \kappa_1], \kappa_3] = 0,$$

which is valid by definition in every Lie algebra (Varadarajan 1984). We deduce at once that

$$[[Q, P], Y_0] = -[[Q, Y_0], P] - [[Y_0, P], Q].$$

Commuting with N confirms (3.6). This leads to an obvious simplification of the Taylor expansion of Ω presented by the end of the previous subsection.

As a matter of fact, (3.5) is just an upper bound on the number of linearly-independent trees in $\mathbb{F}_m$, since, at least in principle, it might happen that a certain term is allowed in $\mathscr{K}_m^\circ$, yet cannot be reached by recursion. For the record, the Hall basis of terms corresponding to trees in $\mathbb{F}_4$ has been constructed using `Maple` and the dimension is indeed eight.

Note that we have not used in our dimension-counting argument the colour-alternation character of chequerboard trees, just the fact that black and white leaves occur in equal numbers.

3.3 Convergence

Our next goal is to ascertain that the Taylor expansion (2.2) converges. This will be accomplished by a tree-counting argument, similar to that in (Iserles & Nørsett 1999).

Each term H_τ for $\tau \in \mathbb{F}_m$ consists of exactly $2m - 1$ nested commutators, with an equal number (m each) of Y_0 and N. Given any norm $\|\cdot\|$ over the Lie algebra $\mathfrak{g}$, it is true that

$$\|[\kappa_1, \kappa_2]\| \le 2\|\kappa_1\| \cdot \|\kappa_2\|, \qquad \kappa_1, \kappa_2 \in \mathfrak{g}. \tag{3.7}$$

It thus follows at once by trivial induction that

$$\tau \in \mathbb{F}_m \qquad \Rightarrow \qquad \|H_\tau\| \le \tfrac{1}{2}(4\|Y\| \cdot \|N\|)^m. \tag{3.8}$$

Theorem 2 *The Lie-algebraic Taylor expansion (2.2) converges for every*

$$0 \le t < \frac{1}{16\|Y_0\| \cdot \|N\|}. \tag{3.9}$$

Proof It follows from (3.8) that

$$\|\Omega(t)\| \le \tfrac{1}{2} \sum_{m=1}^{\infty} \chi_m (4\|Y\| \cdot \|N\|)^m.$$

Let ρ be the radius of convergence about the origin of the analytic function

$$\chi(z) = \sum_{m=1}^{\infty} \chi_m z^m, \qquad z \in \mathbb{C}.$$

Then $\|\Omega(t)\|$ is bounded and (2.2) converges in the underlying norm if

$$|t| < \frac{\rho}{4\|Y\| \cdot \|N\|}.$$

To determine ρ we use (3.5), hence

$$\chi(z) = \sum_{m=1}^{\infty} \frac{1}{2m} \sum_{d|m} \mu(d) \binom{2m/d}{m/d} z^m.$$

Choose any $d \in \mathbb{N}$. It features in the sum whenever m is a multiple of d, therefore its contribution to χ is

$$\frac{\mu(d)}{2d} \sum_{k=1}^{\infty} \frac{1}{k} \binom{2k}{k} z^{dk}.$$

Letting

$$f(z) := \sum_{k=1}^{\infty} \frac{1}{k} \binom{2k}{k} z^k,$$

we obtain

$$\chi(z) = \sum_{d=1}^{\infty} \frac{\mu(d)}{2d} f(z^d).$$

The function f is analytic about the origin with radius of convergence $\frac{1}{4}$, because

$$1 + zf'(z) = \sum_{k=0}^{\infty} \binom{2k}{k} z^k = {}_1F_0 \left[\begin{matrix} \frac{1}{2}; \\ -; \end{matrix} \; 4z \right] = (1-4z)^{-1/2}.$$

We leave the (fairly elementary) manipulations of hypergeometric series above to the reader.

Since the radius of convergence of $f(z^d)$ is $\frac{1}{4^{1/d}}$, we deduce that the radius of convergence of χ is determined by $d = 1$ and it equals $\frac{1}{4}$. The assertion of the theorem follows. $\square$

The bound (3.9) can be increased by a factor of two using a yet-unpublished result of the author, namely that the inequality (3.7) is sometimes much too pessimistic. Specifically, if $\|\cdot\|$ is the *Frobenius norm* and $\mathfrak{g} = \mathfrak{so}(n)$ (as is the case with

Lie-algebraic equations originating in double-bracket equations) then (3.7) might be replaced by

$$\|[\kappa_1, \kappa_2]\| \le \sqrt{2}\|\kappa_1\| \cdot \|\kappa_2\|.$$

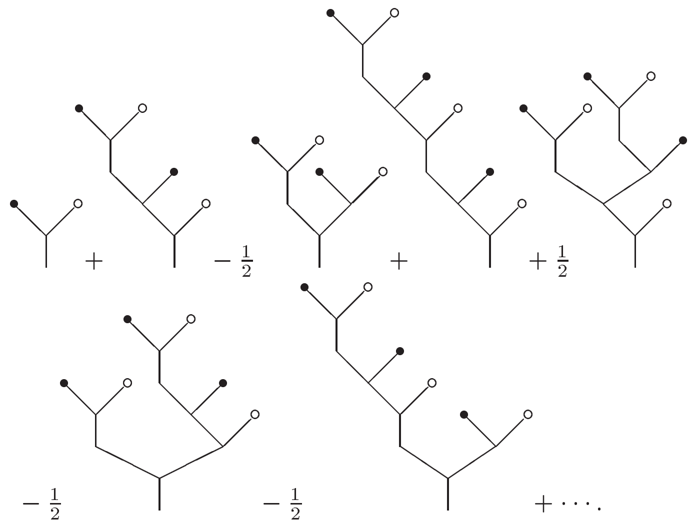
Yet, all this is itself of limited value, since the purpose of Theorem 2 is not to determine the maximal size of t consistent with convergence but merely to prove that convergence takes place. Using technique pioneered in (Blanes, Casas, Oteo & Ros 1998), Fernando Casas has recently (in an unpublished note) showed that the optimal restriction on convergence is

$$0 \le t < \frac{1}{5.8074 \cdots \|Y_0\| \cdot \|N\|}.$$

4 Generalisations

4.1 Variable N

Suppose that N in (2.1) is a function of t. In a sense, this combines the difficulty of a conventional Magnus expansion (nonautonomous vector field in (1.5)) and of the expansion from Section 3 (a nonlinear vector field in (1.5)) and can be approached by combining the technique of (Iserles & Nørsett 1999) with that of this paper. Thus, we again employ binary (but not strictly binary!) rooted trees with bicolour leaves, but (like in 'plain' Magnus expansion) denote integration by 'vertical edges'. It is not the intention of this paper to expand excessive effort on formal and comprehensive treatment of an equation which, on the face of it, is of little independent interest. It suffices to present a third-order expansion of Ω,



4.2 Double double-bracket equations

The *double double-bracket equations*

$$\begin{aligned} X' &= [X,[Y,X]], & & X(0)=X_0,\\ & & t\ge 0, & \\ Y' &= [Y,[Y,X]], & & Y(0)=Y_0, \end{aligned} \tag{4.1}$$

have been introduced by Bloch et al. (1997) as the Hamiltonian form of the geodesic flow on an adjoint orbit of $\mathfrak{g}_u$, a compact real form of a compact, semisimple Lie algebra $\mathfrak{g}$. It is easy to verify that (4.1) evolves by $\mathfrak{gl}(n)$ algebra action,

$$X(t)=\mathrm{e}^{\Omega(t)}X_0\mathrm{e}^{-\Omega(t)}, \qquad Y(t)=\mathrm{e}^{\Omega(t)}Y_0\mathrm{e}^{-\Omega(t)},$$

where

$$\Omega' = \mathrm{dexp}_\Omega^{-1}[\mathrm{e}^{\mathrm{ad}_\Omega}Y_0,\mathrm{e}^{\mathrm{ad}_\Omega}X_0], \quad t\ge 0, \qquad \Omega(0)=O. \tag{4.2}$$

Similarly to (2.1), the solution of (4.2) depends on just two matrices, namely (in the present case) X_0 and Y_0. Thus, we might seek again to express the Taylor expansion of Ω in terms of strictly-binary rooted trees with bicolour leaves. As a matter of record, this is not necessary, since the *exact* solution of (4.2) (and hence of the double double-bracket system (4.1)) can be easily written down: it is simply $\Omega(t)=t[Y_0,X_0]$. This follows at once from

$$[\mathrm{e}^{\mathrm{ad}_{t[Y_0,X_0]}}Y_0,\mathrm{e}^{\mathrm{ad}_{t[Y_0,X_0]}}X_0]=\mathrm{e}^{\mathrm{ad}_{t[Y_0,X_0]}}[Y_0,X_0]=[Y_0,X_0]$$

and (4.2). Therefore,

$$X(t)=\mathrm{e}^{t[Y_0,X_0]}X_0\mathrm{e}^{-t[Y_0,X_0]}, \qquad Y(t)=\mathrm{e}^{t[Y_0,X_0]}Y_0\mathrm{e}^{-t[Y_0,X_0]}.$$

Having said this, it makes sense to consider (4.2) as a generalisation of (2.1), since this leads to a more comprehensive approach, relevant to larger set of equations of this kind. Before we embark on this endeavour, we observe the general pattern. Thus, we have a dexpinv equation, evolving in a Lie algebra $\mathfrak{g}$, whose solution depends on a finite (not necessarily two) known matrices.[2] The challenge, addressed in the next subsection, is the develop a general framework to generate recursively Taylor expansions for the solution of the dexpinv equation. Once this is accomplished, the solution of (4.2) becomes a special case.

4.3 Finite-alphabet dexpinv equations

Suppose that we are given a Lie algebra $\mathfrak{g}$ and the μ subsets $\mathcal{V}_1,\mathcal{V}_2,\ldots,\mathcal{V}_\mu$ of $\mathrm{M}^{n\times n}[\mathbb{R}]$, the set of $n\times n$ real matrices. (Typically, each $\mathcal{V}_k$ is itself a Lie algebra or a Lie group or a symmetric space.) Let $R_1,R_2,\ldots,R_\mu$ be given $n\times n$ matrices, $R_k\in\mathcal{V}_k$, and $g_1,g_2,\ldots,g_\mu$ given functions, analytic sufficiently near the origin, and set

$$\mathcal{U}_k=\{g_k(\mathrm{e}^{\mathrm{ad}_B}R_k)\,:\,B\in\mathfrak{g}\}, \qquad k=1,2,\ldots,\mu.$$

[2]This should not be confused with a free Lie algebra, since these matrices, in general, do not reside in $\mathfrak{g}$ and cannot be its generators.

Finally, let

$$F : \mathcal{U}_1 \times \mathcal{U}_2 \times \cdots \times \mathcal{U}_\mu \to \mathfrak{g}$$

be a *linear* function, computable in a finite number of commutations. We call

$$\Omega' = \mathrm{dexp}_\Omega^{-1} F(g_1(\mathrm{e}^{\mathrm{ad}_\Omega} R_1), g_2(\mathrm{e}^{\mathrm{ad}_\Omega} R_2), \ldots g_\mu(\mathrm{e}^{\mathrm{ad}_\Omega} R_\mu)), \quad t \ge 0, \qquad \Omega(0) = O, \quad (4.3)$$

a *finite-alphabet dexpinv equation.*

A simple example of (4.3) is provided by (2.1), the dexpinv equation originating in the double-bracket equation. In that case

$$\begin{aligned} &\mathfrak{g} = \mathfrak{so}(n), \qquad \mu = 2, \qquad \mathcal{V}_1 = \mathcal{V}_2 = \mathrm{Sym}(n), \qquad R_1 = Y_0, \quad R_2 = N, \\ &g_1(z) = \mathrm{e}^z, \quad g_2 \equiv 1, \qquad F(B_1, B_2) = [B_1, B_2]. \end{aligned}$$

Another example is the equation (4.2), which we have obtained from the double double-bracket flow, when

$$\begin{aligned} &\mu = 2, \qquad \mathcal{V}_1 = \mathcal{V}_2 = \mathrm{M}^{n \times n}[\mathbb{R}], \qquad R_1 = X_0, \quad R_2 = Y_0, \\ &g_1(z) = g_2(z) = \mathrm{e}^z, \qquad F(B_1, B_2) = [B_1, B_2]. \end{aligned}$$

Another example, without any obvious merit except as a handy demonstration of the case $\mu = 4$, is the *triple-bracket* isospectral flow

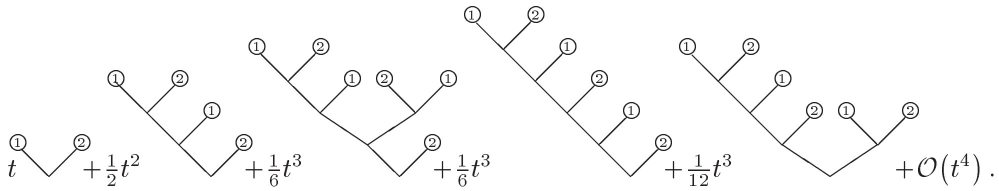
$$Y' = [[[Y, M], [Y.N]], Y], \quad t \ge 0, \qquad Y(0) = Y_0 \in \mathrm{Sym}(n), \qquad (4.4)$$

where $M, N \in \mathrm{Sym}(n)$ are given. In a Lie-algebraic formulation this yields

$$\begin{aligned} &\mathfrak{g} = \mathfrak{so}(n), \quad \mu = 4, \quad \mathcal{V}_1 = \mathcal{V}_2 = \mathcal{V}_3 = \mathrm{Sym}(n), \quad R_1 = Y_0, R_2 = Y_0, R_3 = M, R_4 = N \\ &g_1(z) = g_2(z) = \mathrm{e}^z, \quad g_3, g_4 \equiv 1, \qquad F(B_1, B_2, B_3, B_4) = [[B_1, B_3], [B_2, B_4]]. \end{aligned}$$

Note that we could have used $\mu = 3$, identifying R_2 with R_1, except that in this case F is no longer linear.

We address ourselves to recursive generation of the Taylor expansion for the equation (4.3) by considering strictly-binary rooted trees with *μ-coloured leaves.* Thus, for example, the expansion of (2.1) becomes



We consider the expansion (2.2) of the equation (4.3), except that the index sets $\mathbb{F}_m$ presently contain trees with μ-coloured leaves. The generation of this, more general expansion follows along precisely the same path as that reported in Sections 2 and 3.

We commence by assembling for every $i = 1, 2, \ldots, \mu$ the functions

$$V_r^{[i]} := \mathrm{ad}_\Omega^r R_i, \qquad i = 1, 2, \ldots, \mu, \quad r \in \mathbb{Z}_+.$$

This can be done as follows: Let $V_r^\star := \mathrm{ad}_\Omega^r R^\star$, $r \in \mathbb{Z}_+$, where $R^\star$ is a general $n \times n$ matrix, to which we assign the colour $\odot$. Proceeding like in Section 2, we derive the expansions

$$V_r^\star(t) = \sum_{k=r}^\infty t^k \sum_{\tau \in \mathbb{V}_{r,k}^\star} \beta(\tau) H_\tau, \qquad r \in \mathbb{Z}_+,$$

where the sets $\mathbb{V}_{r,k}^\star$ and the coefficient map β are expressed in terms of the (unknown) quantities from (2.2). It is trivial to observe that

$$V_r^{[i]}(t) = \sum_{k=r}^\infty t^k \sum_{\tau \in \mathbb{V}_{r,k}^{[i]}} \beta(\tau) H_\tau, \qquad r \in \mathbb{Z}_+,$$

where $\mathbb{V}_{r,k}^{[i]}$ is the same as $\mathbb{V}_{r,k}^\star$, except that each occurrence of $\odot$ is replaced by $\textcircled{\scriptsize i}$, $i = 1, 2, \ldots, \mu$, while β is unamended after replacement.

Let

$$g_i(z) = \sum_{r=0}^\infty g_{i,r} z^r$$

be the Taylor expansion of g_i, $i = 1, 2, \ldots, \mu$. Once the $V_r^{[i]}$s have been expanded in the form above, we deduce at once that

$$g_i(\mathrm{e}^{\mathrm{ad}_\Omega}) R_i = \sum_{r=0}^\infty g_{i,r} V_r^{[i]}, \qquad i = 1, 2, \ldots, \mu,$$

and it follows from the linearity of F that

$$\begin{aligned}
&F(g_1(\mathrm{e}^{\mathrm{ad}_\Omega})R_1, g_2(\mathrm{e}^{\mathrm{ad}_\Omega})R_2, \ldots g_\mu(\mathrm{e}^{\mathrm{ad}_\Omega})R_\mu) \\
&= \sum_{r_1=0}^\infty \sum_{r_2=0}^\infty \cdots \sum_{r_\mu=0}^\infty \left(\prod_{i=1}^r g_{i,r_i} \right) F(V_{r_1}^{[1]}, V_{r_2}^{[2]}, \ldots, V_{r_\mu}^{[\mu]}) \\
&= \sum_{r_1=0}^\infty \cdots \sum_{r_\mu=0}^\infty \left(\prod_{i=1}^\mu g_{i,r_i} \right) \sum_{k_1=r_1}^\infty \cdots \sum_{k_\mu=r_\mu}^\infty t^{k_1+\cdots+k_\mu} \\
&\quad \times \sum_{\tau_1 \in \mathbb{V}_{r_1,k_1}^{[1]}} \cdots \sum_{\tau_\mu \in \mathbb{V}_{r_\mu,k_\mu}^{[\mu]}} \left[\prod_{i=1}^\mu \beta(\tau_i) \right] F(H_{\tau_1}, \ldots, H_{\tau_\mu}) \\
&= \sum_{k_1=0}^\infty \cdots \sum_{k_\mu=0}^\infty t^{k_1+\cdots+k_\mu} \sum_{r_1=0}^{k_1} \cdots \sum_{r_\mu=0}^{k_\mu} g^{[\boldsymbol{r}]} \sum_{\tau_1 \in \mathbb{V}_{r_1,k_1}^{[1]}} \cdots \sum_{\tau_\mu \in \mathbb{V}_{r_\mu,k_\mu}^{[\mu]}} \tilde{\beta}(\boldsymbol{\tau}) F(H_{\tau_1}, \ldots, H_{\tau_\mu}),
\end{aligned}$$

where

$$g^{[\boldsymbol{r}]} = \prod_{i=1}^\mu g_{i,r_i}, \qquad \tilde{\beta}(\boldsymbol{\tau}) = \prod_{i=1}^\mu \beta(\tau_i).$$

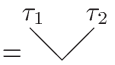
We have just derived the expansion of $U_0 = F(g_1(\mathrm{ad}_\Omega)R_1, \ldots, g_\mu(\mathrm{ad}_\Omega)R_\mu)$, and this is the starting point to recursive derivation of all

$$U_s = \mathrm{ad}_\Omega^s F(g_1(\mathrm{ad}_\Omega)R_1, g_2(\mathrm{ad}_\Omega)R_1, \ldots, g_\mu(\mathrm{ad}_\Omega)R_\mu), \qquad s \in \mathbb{Z}_+.$$

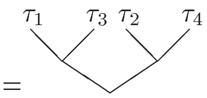
In order to express $\mathbb{U}_{s,l}$, where (as before)

$$U_s(t) = \sum_{l=s}^{\infty} t^s \sum_{\tau \in \mathbb{U}_{s,l}} \gamma(\tau) H_\tau,$$

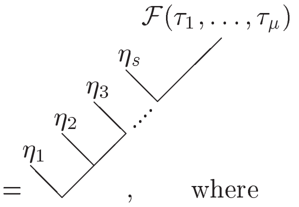
we denote by the *weaving function* $\mathcal{F}(\tau_1, \tau_2, \ldots, \tau_\mu)$ the combination of tree 'branches' $\tau_1, \tau_2, \ldots, \tau_\mu$ that corresponds to the term $F(H_{\tau_1}, H_{\tau_2}, \ldots, H_{\tau_\mu})$. Thus,

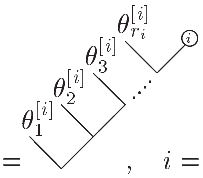
$\mathcal{F}(\tau_1, \tau_2) =$ 

for both double-bracket and double double-bracket equations and

$\mathcal{F}(\tau_1, \tau_2, \tau_3, \tau_4) =$ 

for the triple-bracket equations (4.4). Exactly like in Section 2, we can thus derive a general element in $\mathbb{U}_{s,l}$, namely

$\tau =$ 

, where $\tau_i =$ 

, $i = 1, 2, \ldots, \mu$.

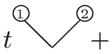
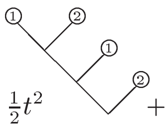
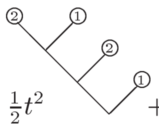
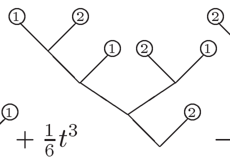
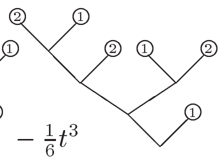
Here

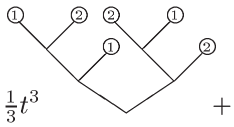
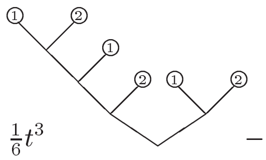
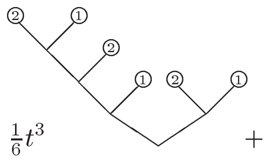
$$\eta_j \in \mathbb{F}_{d_j}, \;\; j = 1, \ldots, s, \qquad \theta_j^{[i]} \in \mathbb{F}_{k_j^{[i]}}, \;\; j = 1, \ldots, r_i, \; i = 1, \ldots, \mu$$

and

$$\sum_{j=1}^{s} d_j + \sum_{i=1}^{\mu} \sum_{j=1}^{r_i} k_j^{[i]} = l, \qquad \gamma(\theta) = g^{[\boldsymbol{r}]} \prod_{j=1}^{s} \alpha(\eta_j) \prod_{i=1}^{\mu} \prod_{j=1}^{r_i} \alpha(\theta_j^{[i]}).$$

Since (2.10) and (3.1–3) all survive intact, this provides us with a well-defined recursive device to generate the expansion (2.2) for general finite-alphabet dexpinv equations (4.3). For example, in the case of the double double-bracket equation (4.2) we obtain after moderately long calculation

t  $+ \frac{1}{2}t^2$  $+ \frac{1}{2}t^2$  $+ \frac{1}{6}t^3$  $- \frac{1}{6}t^3$ 

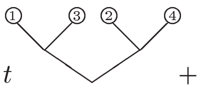
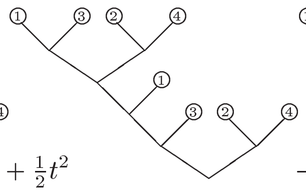
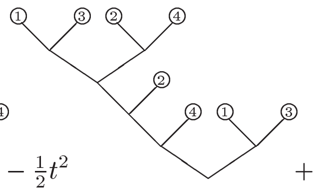
$-\frac{1}{3}t^3$  $+\frac{1}{6}t^3$  $-\frac{1}{6}t^3$  $+\mathcal{O}(t^4)$.

However, it is easy to verify, using the graded Lie-algebra approach of Section 3, that *all* the t^2 and t^3 terms sum to zero: we are left with just $\Omega(t) = t[Y_0, X_0]$, which, as we already know, is the right solution.

Less trivial is the expansion of

$$\Omega' = [[\mathrm{e}^{\mathrm{ad}_\Omega} R_1, R_3], [\mathrm{e}^{\mathrm{ad}_\Omega} R_2, R_3]], \quad t \geq 0, \qquad \Omega(0) = O,$$

the dexpinv equation originating in the triple-bracket system (4.4). The first few terms are

t  $+\frac{1}{2}t^2$  $-\frac{1}{2}t^2$  $+\mathcal{O}(t^3)$.

In acknowledgement of the fact that $R_2 = R_1$, we can relabel the leaves ② as ① once the expansion has been derived.

It is possible to generalise the finite-alphabet equation (4.3) further. Although we do not propose general theory, a single example will suffice to demonstrate the considerably wider range of phenomena that can be modelled by these means. Let $Y_0, N \in \mathrm{M}^{n\times n}[\mathbb{R}]$ and consider the matrix equation

$$Y' = [N, Y^2], \quad t \geq 0, \qquad Y(0) = Y_0. \tag{4.5}$$

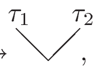
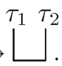
Note firstly that $Y_0 \in \mathfrak{so}(n)$, $N \in \mathrm{Sym}(n)$ implies that Y evolves in $\mathfrak{so}(n)$, while if $Y_0 \in \mathrm{Sym}(n)$ and $N \in \mathfrak{so}(n)$ then the flow lives in $\mathrm{Sym}(n)$. Secondly, even if this is not obvious at first sight, (4.5) is an *isospectral flow,* since it can be rewritten in the form (1.2) (except that Y_0 need not be symmetric) as

$$Y' = [YN + NY, Y], \quad t \geq 0, \qquad Y(0) = Y_0.$$

The corresponding dexpinv equation is

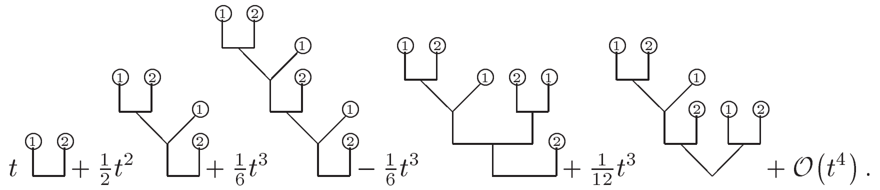
$$\Omega' = \mathrm{dexp}_\Omega^{-1}\{(\mathrm{e}^{\mathrm{ad}_\Omega} Y_0)N + N(\mathrm{e}^{\mathrm{ad}_\Omega} Y_0)\}, \quad t \geq 0, \qquad \Omega(0) = O, \tag{4.6}$$

and it is not anymore in the form (4.3). Having said this, it shares a crucial feature with finite-alphabet equations: the entire information necessary to reconstruct its Taylor coefficients consists of a finite number of matrices: specifically, Y_0 and N. However, we need to consider now *two* kinds of brackets: the standard *Lie bracket* $[A, B] = AB - BA$ and the *Jordan bracket* $[\![A, B]\!] = AB + BA$. Since we wish to use again graph-theoretical formalism, we denote

$[H_{\tau_1}, H_{\tau_2}] \rightsquigarrow$ , $\qquad [\![H_{\tau_1}, H_{\tau_2}]\!] \rightsquigarrow$ .

The latter is not a proper 'graph', but the notation, although informal, is transparent.

As before, we denote Y_0 and N by ① and ② respectively. The first few terms in the Taylor expansion of Ω are



We hasten to confess that the above expansion has not been obtained by a recursive algorithm, unlike the methods reported earlier in this paper. Yet, its very availability and the fact that it can be expressed in a graph-theoretical terminology seem to imply that, with further insight and generalisation, the methods of this paper might be suitable to a wide range of Lie-algebraic equations that can be expressed in terms of a finite 'alphabet'.

Acknowledgements

The author wishes to thank a number of colleagues who have assisted to foster some of the ideas of this paper, in particular Tony Bloch, Fernando Casas, Moody Chu, Fasma Diele, Ken Driessel and Brynjulf Owren.

References

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

Bloch, A. M. (1990), 'Steepest descent, linear programming and Hamiltonian flows', *Contemp. Math. AMS* **114**, 77–88.

Bloch, A. M. & Crouch, P. E. (1996), 'Optimal control and geodesic flows', *Systems & Control Letts* **28**, 65–72.

Bloch, A. M., Brockett, R. W. & Crouch, P. E. (1997), 'Double bracket equations and geodesic flows on symmetric spaces', *Comm. Math. Phys.* **187**, 357–373.

Bourbaki, N. (1975), *Lie Groups and Lie Algebras*, Addison-Wesley, Reading, MA.

Brockett, R. W. (1991), 'Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems', *Lin. Alg. Applics* **146**, 79–91.

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

Calvo, M. P., Iserles, A. & Zanna, A. (1999), 'Conservative methods for the Toda lattice equations', *IMA J. Num. Anal.* **19**, 509–523.

Chu, M. T. (1998), 'Inverse eigenvalue problems', *SIAM Rev.* **40**, 1–39.

Chu, M. T. & Driessel, K. R. (1990), 'The projected gradient method for least squares matrix approximations with spectral constraints', *SIAM J. Numer. Anal.* **27**, 1050–1060.

Dieci, L., Russell, R. D. & van Vleck, E. S. (1994), 'Unitary integrators and applications to continuous orthonormalization techniques', *SIAM J. Num. Anal.* **31**, 261–281.

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

Iserles, A., Munthe-Kaas, H. Z., Nørsett, S. P. & Zanna, A. (2000), 'Lie-group methods', *Acta Numerica* **9**, 215–365.

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

Munthe-Kaas, H. (1998), 'Runge–Kutta methods on Lie groups', *BIT* **38**, 92–111.

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

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

Zanna, A. (1998), On the Numerical Solution of Isospectral Flows, PhD thesis, University of Cambridge, England.

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