Lie–Butcher series and geometric numerical integration on manifolds

PhD Thesis

Alexander Lundervold

Department of Mathematics
University of Bergen

June, 2011

Acknowledgements

This dissertation is submitted as a partial fulfillment of the requirements for the degree Doctor of Philosophy (PhD) at the Faculty of Mathematics and Natural Sciences, University of Bergen, Norway.

I would like to thank my advisors, Prof. Hans Z. Munthe-Kaas and Dr. Kurusch Ebrahimi-Fard, for their support and guidance throughout my period as a doctoral candidate. Their breadth of mathematical interests and their extensive network of collaborators has helped broaden my mathematical horizon considerably.

A special thanks also to the students and staff of the Department of Mathematics for making my time there well spent.

Contents

Outline of the thesis **1**

I Background **3**

1 Geometric numerical integration on vector spaces **5**

1.1 Numerical methods and structure-preservation 6

1.2 Trees and Butcher series 8

1.3 Hopf algebras and the composition of Butcher series 13

1.4 Substitution and backward error analysis for Butcher series 16

1.5 Pre-Lie Butcher series 18

2 Geometric numerical integration on manifolds **23**

2.1 Setting the stage: homogeneous manifolds and differential equations 24

2.2 Trees, D-algebras and Lie–Butcher series 26

2.3 Composition of Lie–Butcher series 31

2.4 Substitution and backward error analysis for Lie–Butcher series . 33

3 Summaries of papers **35**

Bibliography **41**

II Included Papers **47**

A Hopf algebras of formal diffeomorphisms and numerical integration on manifolds

B Backward error analysis and the substitution law for Lie group integrators

C On pre-Lie-type algebras with torsion

Outline of the thesis

The thesis belongs to the field of “geometric numerical integration” (GNI), whose aim it is to construct and study numerical integration methods for differential equations that preserve some geometric structure of the underlying system. Many systems have conserved quantities, e.g. the energy in a conservative mechanical system or the symplectic structures of Hamiltonian systems, and numerical methods that take this into account are often superior to those constructed with the more classical goal of achieving high order.

An important tool in the study of numerical methods is the *Butcher series* (B-series) invented by John Butcher in the 1960s. These are formal series expansions indexed by rooted trees and have been used extensively for order theory and the study of structure preservation. The thesis puts particular emphasis on B-series and their generalization to methods for equations evolving on manifolds, called *Lie–Butcher series* (LB-series).

It has become apparent that algebra and combinatorics can bring a lot of insight into this study. Many of the methods and concepts are inherently algebraic or combinatoric, and the tools developed in these fields can often be used to great effect. Several examples of this will be discussed throughout.

The thesis is structured as follows: background material on geometric numerical integration is collected in **Part I**. It consists of several chapters: in **Chapter 1** we look at some of the main ideas of geometric numerical integration. The emphasis is put on B-series, and the analysis of these. **Chapter 2** is devoted to differential equations evolving on manifolds, and the series corresponding to B-series in this setting. **Chapter 3** consists of short summaries of the papers included in Part II. **Part II** is the main scientific contribution of the thesis, consisting of reproductions of three papers on material related to geometric numerical integration.

Part I

Background

Chapter 1

Geometric numerical integration on vector spaces

In numerical analysis the main objects of study are flows of vector fields, given by initial value problems of the type[*]:

$$y'(t) = F(y(t)), \quad y(t_0) = y_0. \tag{1.1}$$

The function y can be real-valued or vector-valued (giving rise to a system of coupled differential equations). The flow of the differential equation is the map $\Psi_{t,F} : \mathbb{R}^n \to \mathbb{R}^n$ defined by $y(t) = \Psi_{t,F}(y_0)$.[†] Note that $F(y) = d/dt|_{t=0}\Psi_{t,F}(y_0)$. In many practical settings, for instance many mechanical systems modeling physical processes, the vector field is Hamiltonian, and such flows have several interesting geometric properties. We seek to construct good approximations to the exact flow, where 'good' can mean several different things, depending on the context. Sometimes what we want are integrators of high order, other times we need approximations that preserve some qualitative or geometric structure of the underlying dynamical system. Preserving geometric structure is particularly important when studying systems over long time intervals. An early illustration of this fact was made by Wisdom and Holman in [75], where they computed the evolution of the solar system over a billion-year time period using a symplectic method, making an energy-error of only 2×10^{-11}. Section 1.1 of this thesis focuses on structure preservation for numerical methods.

As there are several excellent introductions to geometric numerical integration on $\mathbb{R}^n$ we will not go into a detailed study here, but merely describe some of the main ideas. The book [35] is the standard reference; other introductions can be found in [54, 45, 5, 53, 64, 69, 71].

The focus of this thesis will be on some of the algebraic and combinatorial tools of geometric numerical integration, with particular emphasis on the tools we

[*] Non-autonomous differential equations can also be written on this form by adding a component to the y vector

[†] Here we assume Lipschitz continuity of F for the flow to exist and be unique.

will utilize when studying flows on more general manifolds in the next chapter. Lately, there has been quite a lot of interest in these algebraic aspects of geometric integration, and this has resulted in both an increased understanding of the field, and also of its relations to other areas of mathematics.

1.1 Numerical methods and structure-preservation

Consider an initial value problem of the form (1.1):

$$y'(t) = F(y(t)), \quad y(0) = y_0$$

representing the flow of the (sufficiently smooth) vector field F. A numerical method for (1.1) generates approximations $y_1, y_2, y_3, \dots$ to the solution $y(t)$ at various values of t. One of the simplest methods is the (explicit) **Euler method**. It computes approximations y_n to the values $y(nh)$, where $n \in \mathbb{N}$ and h is the step size, using the rule:

$$y_{n+1} = y_n + hF(y_n). \tag{1.2}$$

This generates a numerical flow Φ_h approximating the exact flow Ψ of F. The accuracy of the method can be measured by its **order**: we say that a one-step method $y_{n+1} = \Phi_h(y_n)$ has order n if $|\Phi_h(y) - \Psi_h(y)| = O(h^{n+1})$ as $h \to 0$. Another way to put this is in terms of the curve traced out by the numerical flow: by comparing its Taylor series to the Taylor series for the curve of the exact flow term by term, we can read off the order of the method. The Taylor series for the solution y has the form

$$y(h) = y_0 + hF(y_0) + \frac{1}{2}h^2 F'(y_0)F(y_0) + O(h^3),$$

and we note that the Euler method is of order 1.

Runge–Kutta methods. The Euler method is an example of a **Runge–Kutta method**, a class of methods that are very common in applications [36, 8]. A Runge–Kutta method is a one-step method computing an approximation y_1 to $y(h)$ with y_0 as input, as follows:

Definition 1.1. An s-stage Runge–Kutta method for solving the initial value problem (1.1) is a one-step method given by

$$\begin{aligned} Y_i =& y_0 + h\sum_{j=1}^{s} a_{ij}F(Y_j), \quad i = 1, \dots s \\ y_1 =& y_0 + h\sum_{i=1}^{s} b_i F(Y_i), \end{aligned} \tag{1.3}$$

where $b_i, a_{ij} \in \mathbb{R}$, h is the step size and $s \in \mathbb{N}$ denotes the number of *stages*.

A Runge–Kutta method can be presented as a *Butcher tableau*, which characterizes the method completely:

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array}$$

Here $c_i = \sum_{j=1}^{s} a_{ij}$.

Example 1.2. We note that the Euler method is the Runge–Kutta method with Butcher tableau:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Another well-known example is the **explicit midpoint method**:

$$y_{n+1} = y_n + hF\left(y_n + \frac{1}{2}hF(y_n)\right),$$

given by:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

Given any number m, there is a Runge–Kutta method of order m [8]. Verifying this involves expanding the methods into series involving the derivatives of F, and already at low orders the expressions get quite complicated. However, in Section 1.2 we shall see that the Runge–Kutta methods are special cases of *Butcher series methods*, and that one can find nice descriptions of the order theory and also structure preservation properties for numerical methods within this framework.

Differential equations and geometric structures. When presented with a system modeled by a differential equation one will often first try to determine its qualitative properties: are there any invariants? What kind of geometric structure does the system have? Structures of interest can be energy and volume preservation, symplectic structure, first integrals, restriction to a particular manifold (as studied in Chapter 2), etc. Then, when choosing (or designing) a numerical method for approximating the solution of the differential equation, it might make sense for the method to share these qualitative features. In that way one has control over what kind of errors the method introduces, obtaining a method tailor-made to the problem at hand.

A rich source of problems with geometric structures are the **Hamiltonian systems**. Let $H : \mathbb{R}^{2n} \to \mathbb{R}$ be a smooth function. A **Hamiltonian vector field** is a vector field on $\mathbb{R}^{2n}$ of the form $X_H = \Omega^{-1}\nabla H$, where Ω is an antisymmetric, invertible $2n \times 2n$ matrix.[‡] The flow of X_H is given by

$$\frac{d}{dt}z = \Omega\nabla_z H(z).$$

The function H represents the total energy of the system. Two important properties of the flow of a Hamiltonian vector field X_H is that it is constant along the Hamiltonian function H (conservation of energy) and that it preserves a symplectic form ω on R^{2n}. Using numerical integrators constructed to preserve these properties has been shown to lead to dramatic improvements in accuracy. For examples of this phenomenon see e.g. [35, 34, 45] and references therein.

1.2 Trees and Butcher series

Starting with the work of John Butcher in the 1960s and 70s [6, 7] the study of methods for solving ordinary differential equations has been closely connected to the combinatorics of rooted trees. Many numerical methods $y_{n+1} = \Phi_h(y_n)$ (including all Runge–Kutta methods) can be expressed as certain formal series, named **Butcher series** by Hairer and Wanner in [37]. By a clever representation of the terms, the series can be indexed over the set of rooted trees.

Consider the differential equation

$$y'(x) = F(y(x)). \tag{1.4}$$

Denote the components of $F : \mathbf{R}^n \to \mathbf{R}^n$ by f^i and write

$$f^i_{j_1 j_2 \cdots j_k} = \frac{\partial^k f^i}{\partial x_{j_1} \partial x_{j_2} \cdots \partial x_{j_k}}. \tag{1.5}$$

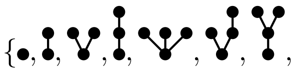
Summing over repeated indices, the first few derivatives of y can be written as:

$$\begin{aligned}
\frac{dy^i}{dx} &= f^i \\
\frac{d^2y^i}{dx^2} &= f^i_j f^j \\
\frac{d^3y^i}{dx^3} &= f^i_{jk} f^j f^k + f^i_j f^j_k f^k \\
\frac{d^4y^i}{dx^4} &= f^i_j f^j_k f^k_l f^l + f^i_j f^j_{kl} f^k f^l + 3 f^i_{jk} f^j_l f^k f^l + f^i_{jkl} f^j f^k f^l.
\end{aligned} \tag{1.6}$$

These expressions soon get very complicated, but the structure can be made much more transparent by observing that the derivatives of F can be associated in a bijective way with rooted trees, an observation already made by Cayley in 1857 [14]. Before giving the exact correspondence between differential equations, rooted trees and Butcher series, we will take a closer look at trees.

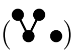
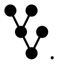
[‡] Hamiltonian vector fields can be defined on any symplectic manifold [3].

Rooted trees. A **tree** is a connected graph with no cycles

$T = \{\bullet,$  $, \ldots\}.$

A **rooted tree** is a tree with one vertex designated as the root. In the pictorial representation of trees, the root will always be drawn as the bottom vertex, and the trees will be ordered from the root to the top. More precisely, a tree τ is a graph consisting of a set of vertices $V(\tau)$ and edges $E(\tau) \subset V(\tau) \times V(\tau)$ so that there is exactly one path connecting any two vertices. A **path** between v_i and v_j is a set of edges $\{v_{s_l}, v_{t_l}\}$ so that $l = 1, 2, \ldots, r$, $s_1 = i$, $t_l = s_{l+1}$ and $t_r = j$. This gives a partial ordering of the tree in terms of paths from the root to the vertices of the tree. A vertex v_i is smaller than another distinct vertex v_j, e.g. $v_i \prec v_j$, if the unique path from from the root to v_j goes via v_i. A vertex v_i is called a **leaf** if there is no vertex v_j with $v_i \prec v_j$. A **child** of a vertex v_i is a vertex v_j with $v_i \prec v_j$ so that there is no vertex v_k with $v_i \prec v_k \prec v_j$. The **order** $|\tau|$ of a tree τ is the number of vertices of the tree. We define a symmetry group on a tree τ as all automorphisms on the vertices. The order of this group, $\sigma(\tau)$, is called the **symmetry** of the tree τ.

A **forest** of rooted trees is a graph whose connected components are rooted trees, e.g. $\omega = \tau_1 \ldots \tau_n$. We include the *empty tree* $\mathbb{I}$, i.e. the graph with no vertices, in the set F of forests. F can be put in bijection to the set of trees via the operator $B^+ : F \to T$, defined on a forest $\omega = \tau_1 \ldots \tau_n$ by connecting the trees to a new root by addition of edges. For example,

$B^+($  $) =$  $.$

This operator can be used to generate all trees recursively from the tree $\bullet$ by the following procedure:

(i) The graph $\bullet$ belongs to T

(ii) If $\tau_1, \ldots, \tau_n \in$ T then $\tau = B^+(\tau_1 \ldots \tau_n)$ is in T.

The **tree factorial** $\tau!$ is given recursively by:

(i) $\bullet! = 1$

(ii) $B^+(\tau_1 \ldots \tau_n)! = |B^+(\tau_1 \ldots \tau_n)| \tau_1! \ldots \tau_n!$.

An important operation on trees is the Butcher product, defined in terms of *grafting*.

Definition 1.3. The **Butcher product** $\tau \diamond \omega$ of a tree $\tau = \mathrm{B}^+(\tau_1 \ldots \tau_n)$ and a forest $\omega = \omega_1 \cdots \omega_m$ is given by grafting ω onto the root of τ:

$$\tau \diamond \omega = \mathrm{B}^+(\tau_1 \ldots \tau_n \, \omega_1 \ldots \omega_m) \qquad (1.7)$$

Butcher series. The calculations of the derivatives of $y'(t) = F(y(t))$ performed at the beginning of the section can be written in terms of the elementary differentials of F.

Definition 1.4. Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a vector field. The **elementary differential** $\mathcal{F}$ of F is

$$\begin{aligned} &\mathcal{F}(\bullet)(t) = F(y) \\ &\mathcal{F}(\tau)(t) = F^{(m)}(y)(\mathcal{F}(\tau_1)(y), \ldots, \mathcal{F}(\tau_m)(y)), \end{aligned} \tag{1.8}$$

where $F^{(m)}$ is the m-th derivative of the vector field F and $\tau = B^+(\tau_1, \ldots, \tau_m)$ is a rooted tree.

We will discuss another way to write elementary differentials in Section 1.5. With the notation from Equation (1.5), the first few elementary differentials are shown in Table (1.1). The vector field F corresponds to the leaves of the tree, the first derivative F' corresponds to a vertex with an edge with one child, the second derivative F'' corresponds to a vertex with two children, etc.

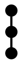
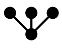
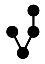
τ	$\mathcal{F}(\tau)(y)^i$
●	f^i
	$f^i_j f^j$
	$f^i_{jk} f^j f^k$
	$f^i_j f^j_k f^k$
	$f^i_{jkl} f^j f^k f^l$
	$f^i_{jk} f^j f^k_l f^l$

Table 1.1: Elementary differentials associated to a vector field F with components f^i.

Butcher series are (formal) Taylor expansions of elementary differentials indexed over trees:

Definition 1.5. A **Butcher series** (B-series) is a (formal) series expansion in a parameter h:

$$\begin{aligned} \mathcal{B}_{h,F}(\alpha) &= \alpha(\mathbb{I})\mathcal{F}(\mathbb{I}) + \sum_{\tau \in \mathrm{T}} h^{|\tau|} \frac{\alpha(\tau)}{\sigma(\tau)} \mathcal{F}(\tau) \\ &= \sum_{\tau \in \tilde{\mathrm{T}}} h^{|\tau|} \frac{\alpha(\tau)}{\sigma(\tau)} \mathcal{F}(\tau), \end{aligned} \tag{1.9}$$

where $\tilde{\mathrm{T}} = \mathrm{T} \cup \{\mathbb{I}\}$, F is a vector field, α is a function $\alpha : \tilde{\mathrm{T}} \to \mathbb{R}$, $\sigma(\tau)$ is the symmetry of τ, h is a real number (representing the step size), and $\mathcal{F}$ is the elementary differential of F, extended to the empty tree $\mathbb{I}$ by $\mathcal{F}(\mathbb{I})(y) = y$.

We shall see that these series can be used to represent numerical methods $y_{n+1} = \Phi_h(y_n)$ approximating the flow of a vector field F, in the sense that the Taylor series for Φ_h can be expanded into a B-series: $\Phi_h = \mathcal{B}_{h,F}(\alpha)$. [§]

By computing the Taylor expansion of the solution to the initial value problem (1.1) one obtains the following result:

Proposition 1.6 ([35])**.** The Taylor series for the solution of the differential equation (1.1) can be written as a B-series:

$$B_{h,F}(\gamma) = \sum_{\tau \in \tilde{\mathrm{T}}} h^{|\tau|} \frac{\gamma(\tau)}{\sigma(\tau)} \mathcal{F}(\tau), \tag{1.10}$$

where $\gamma(\tau) = 1/\tau!$. That is, $y(t+h) = \mathcal{B}_{h,F}(\gamma)(y(t))$.

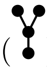
Runge–Kutta methods can also be written as B-series expansions, with coefficients given by the *elementary weights* of the method [6].

Definition 1.7 (Elementary weights)**.** Let b_i and a_{ij} be coefficients of a RK-method as in Definition 1.1, where $i \in \mathbb{N}$. The **elementary weight function** Φ is defined on trees as follows:

$$\begin{aligned} \Phi_i(\bullet) &= c_i \\ \Phi(\bullet) &= \sum_{j=1}^{s} b_j \\ \Phi_i(B^+(\tau_1, \ldots, \tau_k)) &= \sum_{j=1}^{s} a_{ij} \Phi_j(\tau_1) \Phi_j(\tau_2) \ldots \Phi_j(\tau_k) \\ \Phi(B^+(\tau_1, \ldots, \tau_k)) &= \sum_{j=1}^{s} b_j \Phi_j(\tau_1) \Phi_j(\tau_2) \ldots \Phi_j(\tau_k) \end{aligned} \tag{1.11}$$

Here $i = 1, \ldots, s$.

For example,

$\Phi($ $) = \sum_{j=1}^{s} b_j c_j,$ $\quad \Phi($ $) = \sum_{j=1}^{s} b_j c_j^2,$ $\quad \Phi($$) = \sum_{j,k=1}^{s} b_j a_{jk} c_k^2$

Theorem 1.8 ([6])**.** *The B-series for a RK-method given by the elementary weights* $\Phi(\tau)$ *is*

$$\mathcal{B}_{h,F}(\Phi) = \sum_{\tau \in \tilde{\mathrm{T}}} h^{|\tau|} \frac{\Phi(\tau)}{\sigma(\tau)} \mathcal{F}(\tau) \tag{1.12}$$

[§] A numerical method for solving a differential equation is called a *B-series method* if it can be written as a B-series.

Order theory for B-series methods. Once we have the B-series of the exact solution and the B-series of a numerical method, it is straightforward to compare the coefficients and read off the order of the method. For Runge–Kutta methods, we obtain the following result:

Proposition 1.9 ([6])**.** A Runge–Kutta method given by a B-series with coefficients $\Phi(\tau)$ has order n if and only if

$$\Phi(\tau) = \gamma(\tau), \qquad \text{for all } \tau \in T \text{ such that } |\tau| < n.$$

B-series methods and structure preservation. The class of B-series methods includes all Taylor series methods and Runge–Kutta methods. It does not, however, include all numerical methods, an example being the class of *splitting methods*.

It is important to point out that focusing only on B-series methods has its drawbacks. Besides the fact that the class does not contain all methods, it is also known that there are certain geometric structures that cannot be preserved by B-series methods. For example, no B-series method can preserve the volume for *all* systems [41]. However, we will be content with this loss of generality and focus exclusively on methods based on B-series in this chapter, and on their generalization – Lie–Butcher series – in the next.

A case which is particularly well-studied is Hamiltonian vector fields. The following two theorems serve as prime examples:

Theorem 1.10 ([33])**.** *Let $G = \mathcal{B}_{h,F}(\alpha)$ be a vector field with $\alpha(\mathbb{I}) = 0, \alpha(\bullet) \neq 0$. Then G is Hamiltonian for all Hamiltonian vector fields $F(y) = \Omega^{-1}\nabla H(y)$ if and only if*

$$\alpha(\tau_1 \diamond \tau_2) + \alpha(\tau_2 \diamond \tau_1) = 0 \tag{1.13}$$

for all $\tau_1, \tau_2 \in \mathrm{T}$. Here $\diamond$ denotes the Butcher product of Definition 1.3.

Theorem 1.11 ([12])**.** *Consider a numerical method given by a B-series $\mathcal{B}_{h,F}(\alpha)$. The method is symplectic if and only if*

$$\alpha(\tau_1 \diamond \tau_2) + \alpha(\tau_2 \diamond \tau_1) = \alpha(\tau_1)\alpha(\tau_2) \tag{1.14}$$

for all $\tau_1, \tau_2 \in \mathrm{T}$, where $\alpha(\mathbb{I}) = 0$.

The paper [16] gives an overview of what is known about structure preservation for B-series, including characterizations of the various subsets of trees corresponding to energy-preserving, Hamiltonian and symplectic B-series.

1.3 Hopf algebras and the composition of Butcher series

Consider two numerical methods given by Φ^1 and Φ^2. Using the method Φ^1 to advance a point y_0 to a point y_1, and then applying the method Φ^2 using y_1 as initial point, results in a point y_2:

$$y_1 = \Phi^1(y_0), \qquad y_2 = \Phi^2(y_1).$$

This is the idea behind **composition** of numerical methods. In the case where both methods are given by B-series, $\Phi^1(y_1) = \mathcal{B}^1_{h,F}(\alpha)(y_0)$, $\Phi^2(\tilde{y}_1) = \mathcal{B}^2_{h,F}(\beta)(\tilde{y}_0)$, the composition method $\Phi^2 \circ \Phi^1$ is again a B-series: $\Phi^2 \circ \Phi^1(y_0) = \mathcal{B}_{h,F}(\gamma)(y_0)$. This is the Hairer–Wanner theorem from [37]. The coefficient function γ of this B-series was first studied by John Butcher in [7], where he found that composition of B-series is a group operation (giving rise to the *Butcher group*) on the coefficient functions, and gave expressions for the product, identity and inverse in this group.

In [43, 21] Connes and Kreimer introduced a Hopf algebra of rooted trees connected to the renormalization procedure in quantum field theory. Later [4] it was pointed out that a variant of this Hopf algebra is closely related to the Butcher group. More precisely, the Butcher group is the group of *characters* in a Hopf algebra $\mathrm{H}_{\mathrm{BCK}}$ defined by Connes and Kreimer.

We will describe the Butcher group indirectly by describing the Hopf algebra $\mathrm{H}_{\mathrm{BCK}}$. But first we will present some basic definitions from the theory of Hopf algebras. For a comprehensive introduction, see [68, 1]. Other excellent references include [13, 51]. A short introduction can also be found in Paper A, reprinted in Part II below.

Hopf algebras. Let k be a field of characteristic zero. An **algebra** A over k is a k-vector space equipped with a multiplication map $\mu : A \otimes A \to A$ and a unit $u : \mathrm{k} \to A$ so that

- $\mu \circ (id \otimes \mu) = \mu \circ (\mu \otimes id) : A \otimes A \otimes A \to A$ (associativity)
- $\mu \circ (u \otimes id) = \mu \circ (id \otimes u) : k \otimes A \cong A \to A$ (unitality)

A **coalgebra** C over k is the dual notion. It consists of a comultiplication map $\Delta : C \to C \otimes C$ and a counit $\epsilon : C \to \mathrm{k}$ so that

- $(\Delta \otimes id) \circ \Delta = (id \otimes \Delta) \circ \Delta : C \to C \otimes C \otimes C$ (coassociativity)
- $(\epsilon \otimes id) \circ \Delta = (id \otimes \epsilon) \circ \Delta : C \to C \otimes \mathrm{k} \cong C$ (counitality)

A **Hopf algebra** is at once an algebra and a coalgebra, and it comes equipped with an antipode $S : H \to H$. These structures have to satisfy certain compatibility conditions, written as the following diagrams, where τ denotes the flip operation $\tau(h_1, h_2) = (h_2, h_1)$:

$$\begin{tikzcd}
H^{\otimes 4} \arrow[rr, "I\otimes\tau\otimes I"] & & H^{\otimes 4} \arrow[d, "\mu\otimes\mu"] \\
H\otimes H \arrow[u, "\Delta\otimes\Delta"] \arrow[r, "\mu"'] & H \arrow[r, "\Delta"'] & H\otimes H
\end{tikzcd}
\qquad
\begin{tikzcd}
H\otimes H \arrow[r, "\epsilon\otimes\epsilon"] \arrow[d, "\mu"'] & k\otimes k \arrow[d, "\cong"] \\
H \arrow[r, "\epsilon"'] & k
\end{tikzcd}$$

$$\begin{tikzcd}
 & H\otimes H \arrow[rr, "S\otimes 1"] & & H\otimes H \arrow[dr, "\mu"] & \\
H \arrow[ur, "\Delta"] \arrow[rr, "\varepsilon"] \arrow[dr, "\Delta"'] & & k \arrow[rr, "u"] & & H \\
 & H\otimes H \arrow[rr, "1\otimes S"'] & & H\otimes H \arrow[ur, "\mu"'] &
\end{tikzcd}$$

The first two diagrams ensure that the coproduct and the counit are both algebra homomorphisms. The last diagram is best interpreted in terms of the characters in a Hopf algebra. Let A be a commutative k-algebra, and let $\mathcal{L}(H, A)$ denote the set of linear maps from H to A. An element $\alpha \in \mathcal{L}(H, A)$ is called a **character** if $\alpha(x \cdot y) = \alpha(x) \cdot \alpha(y)$ for all $x, y \in H$, where the product on the left-hand side is in H, and on the right-hand side in A. The set of characters in $\mathcal{L}(H, A)$ form a group under the **convolution product**:

$$\phi * \psi = \mu \circ (\phi \otimes \psi) \circ \Delta. \tag{1.15}$$

The unit is the composition of the unit and the counit in H, e.g. $\eta := u \circ \epsilon$. The bottom diagram above corresponds to the antipode being the inverse of the identity under this product, and we have $\alpha^{*-1} = \alpha \circ S$.

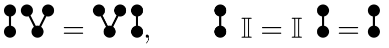
We will also need the concept of **infinitesimal characters**, which are maps α in $\mathcal{L}(H, A)$ satisfying

$$\alpha(x \cdot y) = \eta(x) \cdot \alpha(y) + \alpha(x) \cdot \eta(y).$$

The Butcher–Connes–Kreimer Hopf algebra. Composition of B-series is governed by a certain Hopf algebra $\mathrm{H}_{\mathrm{BCK}}$ based on the set T of rooted trees, called the *Butcher-Connes-Kreimer Hopf algebra.* In the next chapter we will see that a generalization of this Hopf algebra governs the composition of Lie-Butcher series (Section 2.2.3).

To describe the BCK Hopf algebra we need to define its structure as a vector space, an algebra, a coalgebra, and define the antipode. As a $\mathbb{R}$-vector space $\mathrm{H}_{\mathrm{BCK}}$ is generated by the set T of rooted trees, and graded by the order (i.e. number of vertices) of the trees. The algebra structure is that of the symmetric algebra

$S(\mathbb{R}\{T\})$. The product is written as (commutative) concatenation of trees (i.e. disjoint union), giving rise to forests of trees. The unit is the empty tree $\mathbb{I}$.



The coproduct of $\mathrm{H}_{\mathrm{BCK}}$ is the map $\Delta_{\mathrm{BCK}} : \mathrm{H}_{\mathrm{BCK}} \to \mathrm{H}_{\mathrm{BCK}} \otimes \mathrm{H}_{\mathrm{BCK}}$ determined recursively by:

$$\Delta_{\mathrm{BCK}} \circ B^{+}(\omega) = B^{+}(\omega) \otimes \mathbb{I} + (Id \otimes B^{+}) \circ \Delta_{\mathrm{BCK}}(\omega), \tag{1.16}$$

where ω is a forest¶. The counit is the map $\epsilon : \mathrm{H}_{\mathrm{BCK}} \to \mathbb{R}$ given by $\epsilon(\mathbb{I}) = 1$ and $\epsilon(\tau) = 0$ if $\tau \neq \mathbb{I}$. The coproduct can also be written in a non-recursive manner using cuttings of trees.

Cutting trees. An **admissible cut** of a tree τ is a set $c \subset E(\tau)$ of edges of τ such that c contains at most one edge from any path from the root to a leaf. The case $c = \emptyset$ is called the empty cut. Let ω denote the forest with vertices $V(\tau)$ and edges $E(\tau) \setminus c$. We write $R^{c}(\tau)$ for the component of ω containing the root of τ, and $P^{c}(\tau)$ for the forest consisting of the remaining components. The cut resulting in $P^{c}(\tau) = \tau$ and $R^{c}(\tau) = \mathbb{I}$ is also admissible, and called the *full cut* (f.c.).

Theorem 1.12 ([21]). *The coproduct in* $\mathrm{H}_{\mathrm{BCK}}$ *can be written as*

$$\Delta_{\mathrm{BCK}}(\tau) = \sum_{c \in Adm(\tau)} P^{c}(\tau) \otimes R^{c}(\tau) \tag{1.17}$$

Examples of the coproduct can be found in Table 1.2. The antipode can be defined recursively as $S(\mathbb{I}) = \mathbb{I}$ and::

$$S(\tau) = -\tau - \sum_{c \in Adm(\tau) \backslash\backslash \{\emptyset \cup f.c.\}} S(P^{c}(\tau)) R^{c}(\tau) \tag{1.18}$$

The Hairer–Wanner theorem gives the exact correspondence between $\mathrm{H}_{\mathrm{BCK}}$ and composition of B-series:

Theorem 1.13 ([37]). *Let* $\mathcal{B}^{1}_{h,F}(\alpha)$ *and* $\mathcal{B}^{2}_{F}(\beta)$ *be two B-series, with coefficients* $\alpha, \beta : T \to \mathbb{R}$*. The composition* $\mathcal{B}^{2}_{h,F}(\beta) \circ \mathcal{B}^{1}_{h,F}(\alpha)$ *is again a B-series, and we have*

$$\mathcal{B}^{2}_{h,F}(\beta) \circ \mathcal{B}^{1}_{h,F}(\alpha) = \mathcal{B}_{h,F}(\alpha \star \beta), \tag{1.19}$$

where $\star$ *denotes convolution in the Hopf algebra* $\mathrm{H}_{\mathrm{BCK}}$.

¶ Recall that Δ_{BCK} is an algebra morphism and is therefore defined on forests as well as trees, since $\Delta_{\mathrm{BCK}}(\tau_1\tau_2) = \Delta_{\mathrm{BCK}}(\tau_1)\Delta_{\mathrm{BCK}}(\tau_2)$.

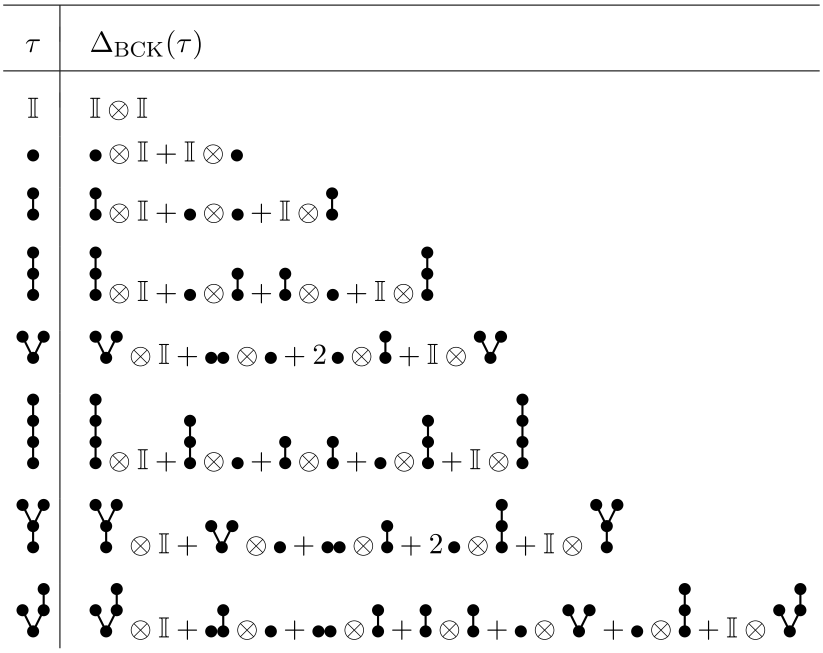


Table 1.2: Examples of the coproduct Δ_{BCK} in the Hopf algebra $\mathrm{H}_{\mathrm{BCK}}$

1.4 Substitution and backward error analysis for Butcher series

Consider a numerical method Φ_h used to solve a differential equation of the form

$$y' = F(y). \tag{1.20}$$

The basic idea of **backward error analysis** of the method Φ_h is to interpret it as giving the exact solution of a modified equation:

$$\tilde{y}' = \tilde{F}_h(\tilde{y}). \tag{1.21}$$

If we can find such an equation, we can use it to study the properties of the numerical method. In other words, the numerical method Φ_h will be represented by a modified vector field $\tilde{F}$, which then can be used to study the method. The idea is based on work by Wilkinson in the context of algorithms for solving equations given by matrices [74], and has been explored in several papers [73, 33, 11, 35, 20]. Recurrence formulas for the modified equation was first obtained in [33, 11].

A related notion is the **modifying integrators** of [20]. The idea is to look for a vector field $\tilde{F}_h$ so that the numerical method Φ_h applied to the flow equation of $\tilde{F}_h$ (Equation 1.21) is the exact solution of Equation 1.20.

It turns out that the case where Φ_h is a B-series method is particularly nice [19, 20, 9]. The vector fields $\tilde{F}_h$ can then be written as B-series whose coefficients are derived from the coefficients of Φ_h, and these coefficients can be expressed by the **substitution law** for B-series methods.

The substitution law. Let $\mathcal{B}_{h,F}(\alpha)$ and $\mathcal{B}_{h,G}(\beta)$ be two B-series, where $\alpha(\mathbb{I}) = 0$. Then $\mathcal{B}_{h,F}(\alpha)$ is a vector field, and we can consider the B-series obtained by using this as the vector field G in the B-series $\mathcal{B}_{h,G}(\beta)$. This is called *substitution* of B-series. The result is given in terms of a bialgebra $\mathrm{H}_{\mathrm{CEFM}}$ by the following theorem:

Theorem 1.14 ([9]). *Let F be a vector field, α, β linear maps $\alpha, \beta : \mathrm{T} \to \mathbb{R}$ where β is an infinitesimal character of $\mathrm{H}_{\mathrm{BCK}}$, and $\alpha(\mathbb{I}) = 0$. Then the vector field $(1/h)\mathcal{B}_{h,F}(\alpha)$ inserted into the B-series $\mathcal{B}_{h,\cdot}(\beta)$ is again a B-series, given by*

$$\mathcal{B}_{h,(1/h)\mathcal{B}_{h,F}(\alpha)}(\beta) = \mathcal{B}_{h,F}(\alpha * \beta), \tag{1.22}$$

where $*$ denotes convolution of characters in the bialgebra H_{CEFM}.

The bialgebra $\mathrm{H}_{\mathrm{CEFM}}$ is the symmetric algebra over rooted trees $S(\mathrm{T})$, with $\bullet$ as unit, equipped with a coproduct given by contracting subforests in trees:

$$\Delta(\tau) = \sum_{\omega \subseteq \tau} \omega \otimes \tau/\omega. \tag{1.23}$$

If τ is a tree then the notation $\omega \subset \tau$ means that ω is a spanning subforest of τ, i.e. that ω is a collection of subtrees of τ so that each vertex of τ belongs to exactly one tree in ω. Then τ/ω denotes the tree obtained by contracting each subtree (with at least two vertices) of τ contained in ω onto a vertex. Some examples of the coproduct can be found in Table 1.3.

There is a Hopf algebra related to $\mathrm{H}_{\mathrm{CEFM}}$, obtained by considering the symmetric algebra over the set of rooted trees T' with at least one edge (e.g. $\bullet$ is not included), and then adding $\bullet$ back as the *unit* for the product. The coproduct is defined as in Equation (1.23). This makes the associated bialgebra connected, and it is therefore a Hopf algebra [51].

For details on these constructions, consult [9].

Backward error analysis and modifying integrators. Once Theorem 1.14 is established one can obtain expressions for backward error analysis and modifying integrators.

Corollary 1.15 (Backward error analysis). *Let $\mathcal{B}_G(\gamma)$ denote the B-series for the exact flow of the vector field G, and let $\mathcal{B}_F(\alpha)$ be a B-series giving a numerical flow for F. The modified vector field $\tilde{F}$ given by $\mathcal{B}_{\tilde{F}}(\gamma) = \mathcal{B}_F(\alpha)$ is a B-series $\mathcal{B}_F(\beta)$ with coefficients given by*

$$\beta * \gamma = \alpha$$

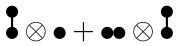
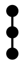
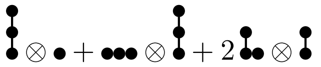
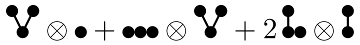
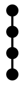
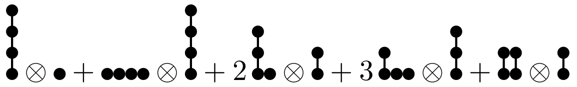
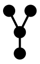
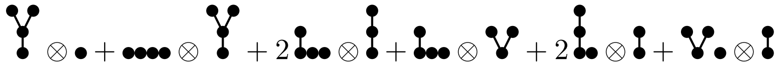
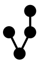
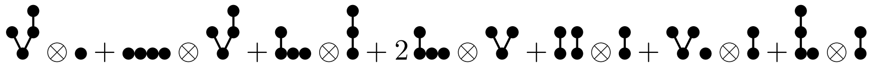
τ	$\Delta_{CEFM}(\tau)$
●	● ⊗ ●
	
	
	
	
	
	

Table 1.3: Examples of the coproduct Δ_{CEFM} in the substitution bialgebra

Corollary 1.16 (Modifying integrators). *Let $\mathcal{B}_G(\gamma)$ denote the B-series for the exact flow of the vector field G, and let $\mathcal{B}_F(\alpha)$ be a B-series giving a numerical flow for F. The modified vector field $\tilde{F}$ so that $\mathcal{B}_{\tilde{F}}(\alpha) = \mathcal{B}_F(\gamma)$ is a B-series $\mathcal{B}_F(\beta)$ whose coefficients are given by*

$$\beta * \alpha = \gamma$$

1.5 Pre-Lie Butcher series

The space of vector fields has the structure of a **pre-Lie algebra**, and in this section we will see that B-series can be formulated purely in terms of this pre-Lie structure. This allows us to lift the concept of B-series to the free pre-Lie algebra, giving rise to **pre-Lie B-series** [26]. Viewing B-series as objects in the free pre-Lie algebra gives a clearer focus on the core algebraic structures at play, and it also enables the application of tools and results from other fields where pre-Lie algebras appear. Two examples of this phenomenon can be found in [25] (see Remark 1.23) and [9]. We give the basic constructions here because formulating Butcher series in terms of pre-Lie algebras will find an analogue in the next chapter, where Lie–Butcher series will be constructed from the so-called D-algebras.

Pre-Lie algebras. The concept of pre-Lie algebras is a relaxation of associative algebras that still preserve their *Lie admissible* property. In other words, for an associative algebra $(A, *)$ antisymmetrization of the product $*$ gives a Lie bracket, making it a Lie algebra: $[a, b] = a*b-b*a$, and this property also holds for pre-Lie algebras. Note, however, that not all pre-Lie algebras are associative. They were first introduced and studied by Vinberg [72], Gerstenhaber [31], and Agrachev and Gamkrelidze [2], under various names. A nice introduction to pre-Lie algebras can be found in [52].

Definition 1.17. A (left) **pre-Lie algebra**[‖] $(A, \rhd)$ is a k-vector space A equipped with an operation $\rhd : A \otimes A \to A$ subject to the following relation:

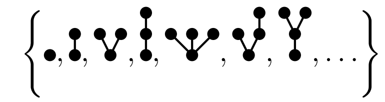
$$(a \rhd b) \rhd c - a \rhd (b \rhd c) \quad = \quad (b \rhd a) \rhd c - b \rhd (a \rhd c) \tag{1.24}$$

Example 1.18 (The pre-Lie algebra of vector fields)**.** The space of vector fields $\mathcal{X}(M)$ on a differentiable manifold M equipped with a flat, torsion-free connection ∇ can be given the structure of a pre-Lie algebra by defining $\rhd$ as $F \rhd G = \nabla_F G$. In the case $M = \mathbb{R}^n$ with the standard flat and torsion-free connection we have that for $F = \sum_{i=1}^n F_i \partial_i$ and $G = \sum_{j=1}^n G_j \partial_j$,

$$F \rhd G = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} F_j (\partial_j G_i) \right) \partial_i. \tag{1.25}$$

In the next chapter we will see that allowing for torsion leads to the concept of **D-algebras**. See also [48], included as Paper C in Part II of the thesis.

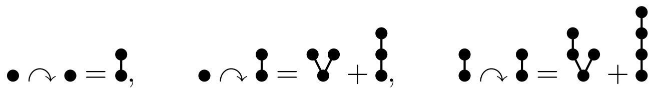
The free pre-Lie algebra. The free pre-Lie algebra has been studied in several papers, most notably by Chapoton and Livernet in [18], Segal in [65], Agrachev and Gramkrelidze in [2], Dzhumadil'daev and Löfwall in [23]. These papers give different bases for the free pre-Lie algebra, and one can choose to work in the basis most beneficial for the problem at hand. A basis for the free pre-Lie algebra $PL(V)$ over a vector space V was described by Chapoton and Livernet in terms of nonplanar rooted trees [18, 17]:



decorated by elements of V. The pre-Lie product $\tau_1 \curvearrowright \tau_2$ of two rooted trees is given by grafting: $\tau_1 \curvearrowright \tau_2$ is the sum of all the trees resulting from the addition of an edge from the root of τ_1 to one of the vertices of τ_2:

$$\tau_1 \curvearrowright \tau_2 := \sum_{v \in V(\tau_2)} \tau_1 \circ_v \tau_2 \tag{1.26}$$

Here $\tau_1 \circ_v \tau_2$ denotes grafting at the vertex v of τ_2.



[‖] Also called a *Vinberg*, *left-symmetric* or *chronological* algebra

Theorem 1.19 ([18]). *$PL(V)$ is the free pre-Lie algebra on the vector space V: for any pre-Lie algebra P equipped with a morphism $V \to P$, there is a unique pre-Lie morphism $PL(V) \to P$ making the following diagram commute:*

$$\begin{tikzcd}
V \arrow[r] \arrow[dr] & PL(V) \arrow[d, "\exists!"] \\
 & P
\end{tikzcd}$$

We write PL for the free pre-Lie algebra on a space with only one element.

The free pre-Lie algebra is related to the Hopf algebra $\mathrm{H}_{\mathrm{BCK}}$ defined in Section 1.3:

Theorem 1.20 ([18]). *The universal enveloping algebra $U(PL)$ of the free pre-Lie algebra on the one-vertex tree, viewed as a Lie algebra, is isomorphic to the dual of the Butcher–Connes–Kreimer Hopf algebra* $\mathrm{H}_{\mathrm{BCK}}$.

In fact, the dual of the Butcher–Connes–Kreimer Hopf algebra is isomorphic to the *Grossman-Larson Hopf algebra* defined [32]. The isomorphism was proven in [38].

Pre-Lie Butcher series. Now we can formulate the pre-Lie Butcher series

Definition 1.21. A **pre-Lie Butcher series** is a formal series in $\mathbb{R}\langle \mathrm{PL} \rangle$:

$$X(\alpha) = \sum_{t \in \mathrm{PL}} h^{|t|} \alpha(t) t. \tag{1.27}$$

The classical B-series are recovered by applying the unique pre-Lie morphism associated to a vector field F:

$$\mathcal{F} : \mathrm{PL} \to \mathcal{X}(\mathbb{R}^n) \quad \text{such that} \quad \mathcal{F}(\bullet) = F.$$

This is the elementary differential function of F as defined in 1.4. It is given recursively by $\mathcal{F}(\bullet) = F$ and

$$\mathcal{F}(t) = F^{(n)}(\mathcal{F}(\tau_1), \ldots, \mathcal{F}(t_n)), \tag{1.28}$$

if $t = \mathrm{B}^+(\tau_1, \ldots, t_n)$.

B-series in any other pre-Lie algebra $(A, \rhd)$ can be defined in the same way: by applying the unique pre-Lie algebra morphism $F : \mathrm{PL} \to A$ to the series (1.27).

Remark 1.22. Since $\mathcal{F} : \mathrm{PL} \to \mathcal{X}(\mathbb{R}^n)$ is a pre-Lie morphism, the trees associated to the derivatives of $y'(t) = F(y(t))$ can be generated by iterated grafting onto the one-vertex tree:

$$\bullet \curvearrowright (\bullet \curvearrowright (\bullet \curvearrowright \ldots (\bullet \curvearrowright \bullet) \ldots)) \quad \text{corresponds to} \quad \frac{d^n y}{dt^n}.$$

This way of looking at elementary differentials will reappear in a different setting in Chapter 2.

Remark 1.23. [Pre-Lie algebras and the Magnus expansion] The formulation of differential equations in terms of pre-Lie algebras has seen some use in numerical analysis. In [25] K. Ebrahimi-Fard and D. Manchon rephrased differential equations of the type $X'(t) = A(t)X(t)$, where X, A are linear operators in a vector space, as combinatorial equations in pre-Lie algebras. In this context they obtained an analogue of the Magnus expansion [50], a series expansion of the solution to the equation in the magma generated by monomials of pre-Lie elements. In this setting it becomes apparent that one can use the pre-Lie relation to cancel out some of the terms in the expansion, leading to a thitherto unknown reduction of the number of terms in the Magnus expansion

Chapter 2

Geometric numerical integration on manifolds

Our main objects of study in this chapter are dynamical systems evolving on *manifolds*:

$$y' = F(y), \qquad y_0 \in M, \quad F \in \mathcal{X}(M), \tag{2.1}$$

where M is a smooth manifold and $\mathcal{X}(M)$ denotes the vector fields on M. As in the previous chapter, the aim is to find good numerical approximations to the flow $\exp(tF) := \Psi_{t,F}$ of (2.1). The study of such systems comprises several different approaches: One simple way to attack the problem is to embed the manifold in $\mathbb{R}^N$, for some N, and use methods developed for $\mathbb{R}^N$ to solve the equation. But then the numerical flow of the method may drift off the manifold, and this can in some cases cause problems [28, 39, 10, 42].

A more satisfying and often better way is to use methods that are intrinsic to the manifold, and not rely on any embedding. Consider for instance a system evolving on the manifold S^3. By embedding S^3 in $\mathbb{R}^4$ one can use numerical methods that approximate the flow of the system using the basic motions of translations in $\mathbb{R}^4$. Another approach is to use *rotations* to move around S^3: $y_{n+1} = Q_n y_n$ where Q_n are orthogonal matrices, i.e. to use the action of the Lie group $SO(3)$ on S^4. This illustrates the intrinsic approach, where we are guaranteed not to drift off S^3. Methods developed for manifolds include the Crouch–Grossman and RKMK-methods (and variants thereof) [56, 57, 22, 61, 27].

In this chapter we will study a generalization of B-series called *Lie–Butcher series*. In analogy to the previous chapter we will look at the composition and substitution of Lie–Butcher series. The papers reproduced in Part II contains most of the theory and results in Lie–Butcher theory that is of interest to us here, and therefore this chapter will mainly consist of sketches of the main results, with references to the relevant papers in Part II.

2.1 Setting the stage: homogeneous manifolds and differential equations

The flows we would like to approximate evolve on smooth manifolds, and so the tools of differential geometry play an important role. We will not review the general theory of smooth manifolds here, but assume a basic knowledge of differential geometry; for excellent introductions see e.g. [67, 66]. For a viewpoint oriented toward geometric numerical integration, see [40]. More precisely, we will be working with smooth manifolds equipped with transitive actions by Lie groups, so called *homogenous manifolds*, where the Lie group provides a way to move around on the manifold.* Because the action is not in general free, the differential equation expressed on the Lie group is not in general unique. Our presentation of differential equations on homogeneous manifolds is based on the papers [59, 57, 27].

Definition 2.1. An **action of a Lie group** G on a smooth manifold M is a group homomorphism $\lambda : G \to \mathrm{Diff}(M)$, $g \mapsto \lambda_g$, where $\mathrm{Diff}(M)$ is the group of diffeomorphisms on M. We will mostly write such an action as a map $\Lambda : G \times M \to M$.

For convenience of notation we write g for the diffeomorphism λ_g, and also $g \cdot m$ for $\lambda_g(m)$. The **orbit** through a point $p \in M$ is the set $G \cdot p = \lambda_G(p)$. The action is called **transitive** if the manifold M is a single G-orbit. That is, if for all $p, q \in M$ there is a $g \in G$ so that $p = g \cdot q$. A manifold equipped with a transitive action by a Lie group G is called a **homogeneous manifold**. A consequence of this is that M is diffeomorphic to the right cosets G/G_x of G, where G_x is the closed Lie subgroup of isotropies, $G_x = \{g \in G \mid gx = x\}$ (the point stabilizer): the smooth manifold structure of G/G_x comes from the quotient map, and the diffeomorphism $F : G/G_x \to M$ is given by $F(gG_x) = g \cdot x$. The group G_x is called *the* subgroup of isotropies because if x' is another point in G, then G_x and $G_{x'}$ are conjugate, and therefore isomorphic.

Important examples of homogeneous manifolds are the spheres $S^n = SO(n+1)/SO(n)$. A (somewhat degenerate) example is the homogeneous manifold $(\mathbb{R}^n, (\mathbb{R}^n, +))$. Here the action of $\mathbb{R}^n$ on itself is given by translations. The theory developed for homogeneous manifolds in this chapter will reduce to the theory developed in the previous chapter when applied to this particular case.

Actions by Lie groups on manifolds can be associated to actions by Lie algebras. Let $\Lambda : G \times M \to M$ be an action of G on M. The associated Lie algebra action $\lambda_* : \mathfrak{g} \to \mathcal{X}(M)$ of $\mathfrak{g}$ on M is the homomorphism defined by:

$$\lambda_*(v)(p) = \frac{d}{dt}\bigg|_{t=0} \Lambda(\exp(tV), p). \tag{2.2}$$

* Note that other manifolds with *local* actions could also be considered, but to to avoid unnecessary complications we elect to only consider homogeneous manifolds.

We sometimes write $v{\cdot}y$ for the element $\lambda_*(v)(y) \in T_yM$. The *Lie–Palais theorem* [62] ensures us that as long as the Lie group G is simply connected, then every action by $\mathfrak{g}$ comes from an action by G. However, if the Lie group is not simply connected, then we can only lift the $\mathfrak{g}$-action to the universal covering group of G. If $F \in \mathcal{X}(M)$ is a vector field, then an element v so that $\lambda_*(v) = F$ is called an *infinitesimal generator* for F.

Remark 2.2. In some cases it makes sense to use other maps $\phi : \mathfrak{g} \to G$ (satisfying $\phi(0) = e$ and $\phi'(0) = V$) besides the exponential map to construct maps $\mathfrak{g} \to \mathcal{X}(M)$ as in Equation (2.2). An overview of various maps of this kind, and their usefulness, can be found in [27].

Differential equations on homogeneous manifolds. Consider the differential equation on a homogeneous manifold (M, G, λ):

$$y' = F(y), \qquad y_0 \in M, \quad F : M \to TM. \tag{2.3}$$

The solution is the flow $\Psi_{t,F} = \exp(tF)$ of the vector field F. The vector field can be written in terms of its infinitesimal generator as $F = \lambda_*(v) : M \to TM$ for an element $v \in \mathfrak{g}$, and the transitivity of the action also allows us to construct a map $f : M \to \mathfrak{g}$ so that

$$F(y) = \lambda_*(f(y))(y) = f(y) \cdot y \tag{2.4}$$

Note that as long as the action is not free, this f is not unique: if $f : M \to \mathfrak{g}$ is such a map, then $f + i : M \to \mathfrak{g}$, where $i(p)$ is in the isotropy subalgebra $\mathfrak{g}_p$ of $\mathfrak{g}$, is another map of the same type. This choice of isotropy class can be helpful when constructing numerical integrators [46].

The differential equation (2.3) can be written as:

$$y' = f(y) \cdot y, \quad \text{where} \quad f : M \to \mathfrak{g}, \tag{2.5}$$

and this is the type of differential equation we will consider in this chapter. Note that in the classical case of $(\mathbb{R}^n, (\mathbb{R}^n, +))$, this equation reduces to the ordinary differential equation (2.3). We also note that the class contains the equations formulated in terms of *frames*:

Remark 2.3 (Frames and differential equations)**.** In the literature for numerical integration of differential equations on manifolds the equations are often simplified by using a *frame* on the manifold [61, 60, 15]. A frame is a set of vector fields $\{E_i\}$ that at each point on the manifold spans the tangent space at that point, so that any vector field F can be written as $F = \sum_i f_i E_i$. The flow equation (2.3) for F can then be written as

$$y' = \sum_i f_i(y) E_i(y), \quad \text{where} \quad f_i : M \to \mathbb{R} \quad \text{are smooth.} \tag{2.6}$$

If we write $\mathfrak{g} \subset \mathcal{X}(M)$ for the Lie subalgebra generated by the vector fields $\{E_i\}$, and let $\lambda_* : \mathfrak{g} \to \mathrm{Diff}(M)$ be as in (2.2), we see that Equation (2.6) is a special case of Equation (2.5), with $f : M \to \mathfrak{g}$ defined by $f(y) = \sum_i f_i(y) E_i$.

Remark 2.4. In [27], K. Engø formulated the general operation of 'moving' differential equations between manifolds using equivariance of actions and relatedness of vector fields. In particular, every differential equation of the form (2.5) was shown to be equivalent to a differential equation on $\mathfrak{g}$. The following diagram from [27] summarizes this:

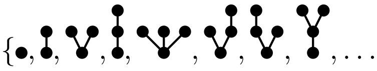
$$\begin{tikzcd}
T\mathfrak{g} \arrow[r, "T(\exp)"] & TG \arrow[r, "T(\lambda.(p))"] & TM \\
\mathfrak{g} \arrow[u] \arrow[r, "\exp"'] & G \arrow[u] \arrow[r, "\lambda.(p)"'] & M \arrow[u, "\lambda_*(v)(p)"']
\end{tikzcd}$$

In other words, the differential equation on a homogeneous manifold (M, G) is moved to the Lie group G (the middle vertical arrow) and then to the Lie algebra $\mathfrak{g}$ (the first vertical arrow). As before, the exponential map $\exp : \mathfrak{g} \to G$ can in some cases be replaced by other maps. The construction of the vertical arrows can be found in [27]. This is the result exploited in the so-called RKMK methods [55, 56, 57].

2.2 Trees, D-algebras and Lie–Butcher series

In Chapter 1 we observed that ordinary differential equations in $\mathbb{R}^n$ are related to rooted trees, and that the formal series indexed over trees we used in our study are related to pre-Lie algebras. In the more general case of differential equations on manifolds, we will see that forests of *ordered* rooted trees and D-algebras play these roles. We will sketch the construction of ordered rooted trees, D-algebras and Lie–Butcher series. Details can be found in [58] or [47] (Paper A in Part II below).

Ordered trees and D-algebras. The set

OT = {  , . . . }.

of ordered rooted trees consists of *all* rooted trees (Section 1.2). Unlike the set $\mathrm{T} \subset \mathrm{OT}$ of rooted trees, we do not identify trees who differ in the order of their branches. In other words, an ordered rooted tree is a tree τ together with a chosen order of the branches connected to each vertex of τ. Write OF for the set of ordered words (including the empty word) of elements from OT, called the set of **ordered forests**. Let $\mathrm{N} = \mathbb{R}\langle \mathrm{OT} \rangle$ be the noncommutative polynomials over OT. The linear dual $\mathrm{N}^* := \mathrm{Hom}(\mathrm{N}, \mathbb{R})$ is identified with the infinite combinations of words, and we write $\langle \cdot, \cdot \rangle$ for the pairing making words in OT orthogonal. That is, $\langle \omega_1, \omega_2 \rangle = \delta_{\omega_1, \omega_2}$, for all $\omega_1, \omega_2 \in \mathrm{OF}$.

It is sometimes convenient to allow the trees to be *decorated* by a set $\mathcal{C}$, often called the set of colors. This is done via a map from the vertices of the tree to the set $\mathcal{C}$. We write $\mathrm{OT}_{\mathcal{C}}$ and $\mathrm{OF}_{\mathcal{C}}$ for the set of trees and forests colored by $\mathcal{C}$.

A basic operation on N is the **left grafting product** $\cdot \curvearrowright \cdot : \mathrm{N} \otimes \mathrm{N} \to \mathrm{N}$ of [58]. It is defined recursively by

$$\begin{aligned}
&\mathbb{I} \curvearrowright \omega = \omega \\
&\omega \curvearrowright \mathbb{I} = 0 \\
&\omega \curvearrowright \bullet = B^{+}(\omega), \\
&\tau \curvearrowright \omega_1\omega_2 = (\tau \curvearrowright \omega_1)\omega_2 + \omega_1(\tau \curvearrowright \omega_2) \\
&(\tau\omega) \curvearrowright \omega_1 = \tau \curvearrowright (\omega \curvearrowright \omega_1) - (\tau \curvearrowright \omega) \curvearrowright \omega_1,
\end{aligned} \tag{2.7}$$

where τ is a tree and ω_1, ω_2 are forests. If we write $(\cdot)[\cdot]$ for $\curvearrowright$, then concatenation and grafting gives N the structure of a D-algebra, as defined in [58] (see also [47, 49, 48]):

Definition 2.5. Let A be a unital associative algebra with product $f, g \mapsto fg$, unit $\mathbb{I}$ and equipped with a non-associative composition $(.)[.] : A \otimes A \to A$ such that $\mathbb{I}[g] = g$ for all $g \in A$. Write $\mathcal{D}(A)$ for the set of all $f \in A$ such that $f[\cdot]$ is a derivation:

$$\mathcal{D}(A) = \{f \in A \mid f[gh] = (f[g])h + g(f[h]) \text{ for all } g, h \in A\}.$$

Then A is called a **D-algebra** if for any derivation $f \in \mathcal{D}(A)$ and any $g \in A$ we have

(i) $g[f] \in \mathcal{D}(A)$
(ii) $f[g[h]] = (fg)[h] + (f[g])[h]$.

In [58] it was also shown that the D-algebra N is the *free* D-algebra:

Theorem 2.6 ([58]). *The vector space $\mathrm{N} = \mathrm{k}\langle \mathrm{OT}_{\mathcal{C}} \rangle$ is the free D-algebra over $\mathcal{C}$. That is, for any D-algebra $\mathcal{A}$ and any map $\nu\colon \mathcal{C} \to D(\mathcal{A})$ there exists a unique D-algebra homomorphism $\mathcal{F}_\nu\colon N \to \mathcal{A}$ such that $\mathcal{F}_\nu(c) = \nu(c)$ for all $c \in \mathcal{C}$.*

$$\begin{tikzcd}
\mathcal{C} \arrow[r, hook] \arrow[d, "\nu"'] & \mathrm{N} \arrow[d, "\exists!\,\mathcal{F}_\nu"] \\
D(\mathcal{A}) \arrow[r, hook] & \mathcal{A}
\end{tikzcd}$$

A D-algebra homomorphism between two D-algebras A and B is an algebra morphism $F : A \to B$ such that $F(\mathcal{D}(A)) \subset \mathcal{D}(B)$, and $F(a[b]) = F(a)[F(b)]$.

This theorem enables us to define elementary differentials and Lie–Butcher series by applying it to the case where $\mathcal{A}$ is the D-algebra $U(\mathfrak{g})$ of differential operators. Recall that a vector field (or, in other words, a first-order differential operator) F on a homogeneous manifold (M, G) can be represented as a function $f : M \to \mathfrak{g}$. Similarly, all higher order differential operators on M can be represented as functions from M to the universal enveloping algebra $U(\mathfrak{g})$ of $\mathfrak{g}$.

Theorem 2.7 ([58]). *Let (M, G) be a homogeneous manifold and let $\mathfrak{g}$ denote the Lie algebra of G. Let $U(\mathfrak{g})$ denote the universal enveloping algebra of $\mathfrak{g}$, consisting of all higher order differential operators on M, and extend its structure to $C^\infty(M, U(\mathfrak{g})) =: U(\mathfrak{g})^M$ via*

$$F[G](p) := (F(p)[G])(p), \quad FG(p) := F(p)G(p). \tag{2.8}$$

These two operations give $U(\mathfrak{g})^M$ the structure of a D-algebra.

Remark: post-Lie algebras. In [48] (reproduced as Paper C in Part II) the author and H. Munthe-Kaas developed a more refined view of D-algebras, where the D-algebras are enveloping algebras of *post-Lie algebras* (post-Lie algebras were also introduced independently by Vallette in [70]). This point of view is currently being studied further in an ongoing project [24], where the *operad* behind post-Lie and D-algebras (also called **post associative algebras**) is explored.

Definition 2.8. A **post-Lie algebra** is a Lie algebra $(A, [\cdot,\cdot])$ equipped with a non-commutative, non-associative product $\rhd : A \otimes A \to A$ satisfying:

$$x \rhd [y, z] = [x \rhd y, z] + [y, x \rhd z] \qquad \text{(derivation property)} \tag{2.9}$$

$$[x, y] \rhd z = a_\rhd(x, y, z) - a_\rhd(y, x, z), \tag{2.10}$$

where $a_\rhd(x, y, z)$ is the associator $a_\rhd(x, y, z) = x \rhd (y \rhd z) - (x \rhd y) \rhd z$.

In [48] it is shown that the free Lie algebra over rooted trees colored by a set $\mathcal{C}$ is the free post-Lie algebra, and that its universal enveloping algebra is the free D-algebra defined above. Notice that relation (2.10) implies that a pre-Lie algebra (Section 1.5) is a post-Lie algebra with vanishing bracket.

Lie–Butcher series

Analogous to the B-series of Chapter 1, the Lie–Butcher series can be used to represent flows – numerical or exact – on homogeneous manifolds. To achieve this one combines the concept of *Lie series* in free Lie algebras with ideas from the theory of B-series. An exposition of free Lie algebras and Lie series can be found in the book [63] by Reutenauer.

The **free Lie algebra** $\mathrm{FLA}(A)$ over a set A of generators is the closure of the generators under commutation and linear combination. In particular, we have the free Lie algebra $\mathrm{FLA}(\mathrm{OT})$ over the set of ordered rooted trees. A **Lie series** is a series expansion:

$$S = \sum_{n \geq 0} S_n, \tag{2.11}$$

where each homogeneous component is an element of $\mathrm{FLA}(\mathrm{OT})$, i.e. the S_n's are *Lie polynomials*.

A Lie series of particular interest to us appears when computing the pullback of functions along flows of vector fields on homogeneous manifolds. Let $F \in \mathcal{X}(M)$ be a vector field with flow $\Phi_{t,F}$, and $\psi : M \to \mathfrak{g}$ a function. Then

$$\left.\frac{d}{dt}\right|_{t=0} \Phi^*_{t,F}\psi = F[\psi]. \tag{2.12}$$

The Taylor expansion of $\Phi^*_{t,F}\psi$ around 0 therefore takes the form of a Lie series

$$\begin{aligned}\Phi^*_{t,F}\psi &= \sum_{n=0}^{\infty} \frac{t^n}{n!}\left(\left.\frac{\partial^n}{\partial t^n}\right|_{t=0} \Phi^*_{t,F}\psi \right) \\ &= \psi + tF[\psi] + \frac{t^2}{2!}F[F[\psi]] + \frac{t^3}{3!}F[F[F[\psi]]] + \cdots .\end{aligned} \tag{2.13}$$

Bell polynomials. The higher order derivatives of the pullbacks can be written in terms of noncommutative Bell polynomials [47]:

Definition 2.9. Let $D = \mathbb{R}\langle\mathcal{I}\rangle$ be the free associative algebra over an alphabet $\mathcal{I} = \{d_i\}$, and let $\partial : D \to D$ denote the derivation given by $\partial(d_i) = d_{i+1}$. The **noncommutative Bell polynomials** $B_n = B_n(d_1, \ldots, d_n) \in \mathbb{R}\langle\mathcal{I}\rangle$ are defined by the recursion

$$\begin{aligned} B_0 &= \mathbb{I} \\ B_n &= (d_1 + \partial)B_{n-1}, \quad n > 0. \end{aligned} \tag{2.14}$$

Theorem 2.10 ([55, 47]). *The derivatives of the pullback of a function ψ along the time-dependent flow $\Phi_{t,F}$ is:*

$$\frac{d^n}{dt^n}\Phi^*_{t,F}\psi = B_n(F)[\psi], \tag{2.15}$$

where $B_n(F_t)$ is the image of the Bell polynomials B_n under the homomorphism given by $d_i \mapsto F^{(i-1)}$ ($(i-1)$th derivative). In particular

$$\left.\frac{d^n}{dt^n}\right|_{t=0} \Phi^*_{t,F_t}\psi = B_n(F_1, \ldots, F_n)[\psi] =: B_n(F_i)[\psi], \tag{2.16}$$

where $F_{n+1} = d^n/dt^n|_{t=0}F$.

This result allows us to rewrite the Lie series (2.13) as the following expression [55]:

$$\Phi^*_{t,F}\psi = \sum_{n=0}^{\infty} F^n[\psi]\frac{t^n}{n!} = \sum_{n=0}^{\infty} B_n(F_i)[\psi]\frac{t^n}{n!}, \tag{2.17}$$

where F^n iterated application of F, as in Equation (2.13).

Remark 2.11. It is well known that the classical Bell polynomials can be defined in terms of determinants, and it seems like the non-commutative Bell polynomials can be defined in the same way, only now in terms of a non-commutative analog of the determinant: the **quasi-determinants** of Gelfand and Retakh ([30], see also [29]). For example, we have

$$\begin{aligned}\det\begin{bmatrix} x_1 & -1 & 0 \\ \binom{3-1}{1}x_2 & x_1 & -1 \\ \binom{3-1}{2}x_3 & \binom{3-2}{1}x_2 & x_1 \end{bmatrix} &= \det\begin{bmatrix} x_1 & -1 & 0 \\ 2x_2 & x_1 & -1 \\ x_3 & x_2 & x_1 \end{bmatrix} \\ &= x_1^3 + 2x_1x_2 + x_2x_1 + x_3 \\ &= B_3,\end{aligned}$$

where $\det$ denotes the quasi-determinant. The significance of this result is at the present time unexplored.

The Lie–series (2.13) can also be written as the *Lie–Butcher series* for the exact flow.

Lie–Butcher series. The general Lie–Butcher series $\mathcal{B}_f(\alpha)$ are constructed to represent flows given by $y_0 \mapsto y_t = \Psi_t(y_0)$:

$$\Psi_t(y(t)) = \mathcal{B}_f(\alpha)[\Psi_t](y_0). \tag{2.18}$$

Before giving the definition of Lie–Butcher series we need to define the elementary differentials of a vector field F:

Definition 2.12. Let $\mathcal{F}_f : \mathrm{N} \to U(\mathfrak{g})^M$ be the unique D-algebra morphism given by Theorem 2.6 by associating $\bullet$ to a vector field $f : M \to \mathfrak{g}$. This is called the **elementary differentials** of the vector field f.

Note that $\mathcal{F}_f : \mathrm{N} \to U(\mathfrak{g})^M$ is given recursively by

(i) $\mathcal{F}_f(\mathbb{I}) = \mathbb{I}$

(ii) $\mathcal{F}_f(\mathrm{B}^+(\omega)) = \mathcal{F}_f(\omega)[f]$

(iii) $\mathcal{F}_f(\omega_1\omega_2) = \mathcal{F}_f(\omega_1)\mathcal{F}_f(\omega_2)$

The general Lie–Butcher series are expansions of elementary differentials indexed over ordered rooted forests.

Definition 2.13. A **Lie–Butcher series** (LB-series) is a formal series expansion in $U(\mathfrak{g})^M$:

$$\mathcal{B}_f(\alpha) = \sum_{\omega \in \mathrm{OF}} h^{|\omega|}\alpha(\omega)\mathcal{F}_f(\omega), \tag{2.19}$$

where $\alpha : \mathrm{N} \to \mathbb{R}$.

It turns out [47] that the Lie series (2.13) can be written as

$$\Phi^*_{t,f}\psi = \sum_{\omega \in \mathrm{OT}} \gamma(\omega)\mathcal{F}_f(\omega), \tag{2.20}$$

where γ are the coefficients appearing when iteratively (left) grafting • onto •. This is the Lie–Butcher series for the exact flow.

See [55, 56, 61, 60, 58], Paper A [47] and Paper B [49] in Part II for examples of and details about LB-series and numerical flows.

2.3 Composition of Lie–Butcher series

We would like to understand the result of *composing* LB-series methods in a similar way as we did for B-series methods in Section 1.3. The basic problem is to determine whether the method Φ resulting from composing two methods $\Phi^2 \circ \Phi^1$– both given by LB-series–is another LB-series, and in that case, what its coefficients are. Just as there is a Hopf algebra governing composition of B-series (the BCK Hopf algebra discussed in Section 1.3), there is a Hopf algebra $\mathrm{H}_{\mathrm{MKW}}$ behind the composition of LB-series. This Hopf algebra was first studied in [58], where its properties and its relation to the BCK Hopf algebra was explored. An introduction can also be found in [47], reproduced as Paper A in Part II.

The Hopf algebra of composition. As a vector space $\mathrm{H}_{\mathrm{MKW}}$ is spanned by the set of ordered forests: $\mathrm{H}_{\mathrm{MKW}} = \mathbb{R}\langle \mathrm{OT}\rangle$. The product is given by *shuffling*:

$$\begin{aligned} &\mathbb{I} \shuffle \omega = \omega = \omega \shuffle \mathbb{I} \\ &(\tau_1\omega_1) \shuffle (\tau_2\omega_2) = \tau_1(\omega_1 \shuffle \tau_2\omega_2) + \tau_2(\tau_1\omega_1 \shuffle \omega_2) \end{aligned} \tag{2.21}$$

where $\tau_1, \tau_2 \in \mathrm{OT}$ and $\omega_1, \omega_2 \in \mathrm{OF}$. The coproduct is given recursively by $\Delta_N(\mathbb{I}) = \mathbb{I} \otimes \mathbb{I}$ and

$$\Delta_N(\omega\tau) = \omega\tau \otimes \mathbb{I} + \Delta_N(\omega) \shuffle \cdot (I \otimes B^+_i)\Delta_N(B^-(\tau)), \tag{2.22}$$

where $\tau \in \mathrm{OT}$, $\omega \in \mathrm{OF}$. Here $\shuffle\cdot : \mathrm{N}^{\otimes 4} \to \mathrm{N} \otimes \mathrm{N}$ denotes shuffle on the left and concatenation on the right: $(\omega_1 \otimes \omega_2) \shuffle \cdot (\omega_3 \otimes \omega_4) = (\omega_1 \shuffle \omega_3) \otimes (\omega_2\omega_4)$.

The coproduct can also be written in terms of *left admissible cuts*, analogous to the coproduct in $\mathrm{H}_{\mathrm{BCK}}$ (Theorem 1.12):

Theorem 2.14 ([58]). *The coproduct in* $\mathrm{H}_{\mathrm{MKW}}$ *can be written as*

$$\Delta_{MKW}(\omega) = \sum_{c \in FLAC(\omega)} P^c(\omega) \otimes R^c(\omega), \tag{2.23}$$

where ω *is a forest in* OT.

A left admissible cut differs from the admissible cuts defined in Section 1.3 (see [58]): an *elementary cut* c of a tree τ is a selection of edges to be removed from τ, chosen in such a way that if an edge e is removed, then all the branches on the same level and to the left of e must also be removed. A cut results in a collection of trees concatenated together to form a forest $P^c_{el}(\tau)$ (the *pruned part*), and a remaining tree $R^c_{el}(\tau)$, containing the root. A left admissible cut $c = \{c_1, \ldots, c_n\}$ on τ is a collection of such elementary cuts, with the property that any path from the root to any vertex crosses at most one cut c_i. The pruned parts from each cut together form the pruned part $P^c(\tau)$ of the left admissible cut, where the parts coming from different cuts are shuffled together. We also include the *full cut* and the *empty cut*, which results in $P^c(\tau) = \tau$ and $P^c(\tau) = \mathbb{I}$, respectively. The cutting operation is extended to forests ω as follows: apply the B^+ operation to ω to get a tree, cut this without using the cut removing *all* the edges coming out of the root, and, finally, remove the added root from $R^c(\omega)$.

See Table 2.1 for some examples of the coproduct Δ_{MKW}, and see [58] or [47] (reproduced as Paper A in Part II below) for further examples and other properties of $\mathrm{H}_{\mathrm{MKW}}$.

ω	$\Delta_{MKW}(\omega)$
$\mathbb{I}$	$\mathbb{I} \otimes \mathbb{I}$
●	● $\otimes \mathbb{I} +$ $\mathbb{I} \otimes$ ●
●●	●● $\otimes \mathbb{I} +$ ● $\otimes$ ● $+ \mathbb{I} \otimes$ ●●
⁞	⁞ $\otimes \mathbb{I} +$ ● $\otimes$ ● $+ \mathbb{I} \otimes$ ⁞
●⁞	●⁞ $\otimes \mathbb{I} + 2$ ●● $\otimes$ ● $+$ ● $\otimes$ ⁞ $+$ ● $\otimes$ ●● $+ \mathbb{I} \otimes$ ●⁞
⁞●	⁞● $\otimes \mathbb{I} +$ ⁞ $\otimes$ ● $+$ ● $\otimes$ ●● $+ \mathbb{I} \otimes$ ⁞●

Table 2.1: Examples of the coproduct Δ_{MKW}

The main result linking $\mathrm{H}_{\mathrm{MKW}}$ to LB-series is the following, which is an analog of the Hairer-Wanner theorem (Theorem 1.13) for B-series:

Theorem 2.15 ([58]). *The composition of two LB-series is again a LB-series:*

$$\mathcal{B}_f(\alpha)[\mathcal{B}_f(\beta)] = \mathcal{B}_f(\alpha * \beta), \tag{2.24}$$

where $*$ *is the convolution product in* $\mathrm{H}_{\mathrm{MKW}}$.

2.4 Substitution and backward error analysis for Lie–Butcher series

In [49] (reproduced as Paper B in Part II) the substitution law for LB-series methods was developed, culminating in a formula that can be used to calculate the modified vector field used in backward error analysis.

The substitution law. The basic idea is as for B-series (Section 1.4): We consider substituting a LB-series into another LB-series, e.g. $\mathcal{B}_{\mathcal{B}_f(\beta)}(\alpha)$, and the question is as before: is this a LB-series, and in that case, which one? The result is given in terms of the *substitution law*:

Theorem 2.16 ([49]). *The substitution law defined in Definition 2.17 corresponds to the substitution of LB-series in the sense that*

$$\mathcal{B}_{\mathcal{B}_f(\beta)}(\alpha) = \mathcal{B}_f(\beta \star \alpha)$$

The substitution law is defined by using the freeness of the D-algebra $\mathrm{N} = \mathbb{R}\langle\mathrm{OT}\rangle$ (Theorem 2.6):

Definition 2.17. For any map $\alpha : \mathcal{C} \to D(\mathrm{N})$ Theorem 2.6 implies that there a unique D-algebra homomorphism $\alpha* : \mathrm{N} \to \mathrm{N}$ such that $\alpha(c) = \alpha * c$ for all $c \in \mathcal{C}$. This homomorphism is called α-substitution.

$$\begin{tikzcd}
\mathcal{C} \arrow[r, hook] \arrow[d, "\alpha"'] & \mathrm{N} \arrow[d, "\alpha*"] \\
D(\mathrm{N}) \arrow[r, hook] & \mathrm{N}
\end{tikzcd}$$

Calculating the substitution law. To obtain a formula for the substitution law, we consider the dual α_*^t of α-substitution:

$$\langle \alpha * \beta, \omega\rangle = \langle \beta, \alpha_*^t(\omega)\rangle, \tag{2.25}$$

and we call it the *substitution character*. The dual pairing $\langle\cdot,\cdot\rangle$ is the one induced by requiring that all forests in OT are orthogonal, and we may write $\langle\alpha,\omega\rangle = \alpha(\omega)$. The map α_*^t is a character for the shuffle product [49, Proposition 3.8]: $\alpha_*^t(\omega_1 ⧢ \omega_2) = \alpha_*^t(\omega_1) ⧢ \alpha_*^t(\omega_2)$.

The formula for the substitution law is based on the cutting of trees as in the coproduct Δ_{MKW}. More specifically, it is based on the dual of grafting, called *pruning*:

$$\mathcal{P}_\nu(\omega) = \sum_{c \in LAC(\omega)} \langle \nu, P^c(\omega)\rangle R^c(\omega). \tag{2.26}$$

Here the sum is over the left admissible cuts, but as opposed to the cuts in the formula (2.23) for Δ_{MKW}, the full cut is not included.

In [49] the following inductive formula for α_*^t was obtained:

Theorem 2.18 ([49]). *We have*

$$\alpha_*^t(\omega) = \sum_{(\omega)\in\Delta_C} \sum_{c\in LAC(\omega_{(2)})} \alpha_*^t(\omega_{(1)})\,\mathrm{B}^+\left(\alpha_*^t(P^c(\omega_{(2)}))\right)\alpha(R^c(\omega_{(2)})),$$

if $\omega \neq 1$ *and* $\alpha_*^t(\mathbb{I}) = \mathbb{I}$. *Here* Δ_C *denotes the deconcatenation coproduct.*

By introducing a magmatic operation $\mu_\times$ on N, given by $\mu_\times(\omega_1,\omega_2) = \omega_1\,\mathrm{B}^+(\omega_2)$[†], this can also be written as a composition of operators:

$$\alpha_*^t = \mu\circ(\mu_\times\otimes I)\circ(\alpha_*^t\otimes\alpha_*^t\otimes a)\circ(I\otimes\Delta'_{MKW})\circ\Delta_C. \tag{2.27}$$

Here Δ_C is deconcatenation, Δ'_{MKW} denotes the coproduct in (2.23) with the full cut removed, and μ denotes concatenation.

Some examples of the substitution character can be found in Table 2.2. Many more examples and details can be found in [49] (Paper B below).

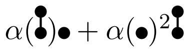
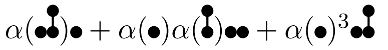
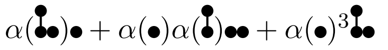
ω	$\alpha_*^t(\omega)$
$\mathbb{I}$	$\mathbb{I}$
$\bullet$	$\alpha(\bullet)\bullet$
$\bullet\bullet$	$\alpha(\bullet)^2\bullet\bullet$
	
	
	

Table 2.2: Examples of the substitution character α_*^t

Remark 2.19. One would like the substitution law $*$ to be a convolution product in a Hopf or bialgebra, analogous to the substitution of B-series (Theorem 1.14). One possible way to achieve this is by obtaining a concrete description of the operations in the post-Lie operad. In that case one can follow the procedure in [9], which, roughly, is the following: The post-Lie operad has a pre-Lie structure (general phenomenon for augmented operads), there is an associated Lie algebra structure, its universal enveloping algebra is a Hopf algebra, and its dual is the Hopf algebra for the substitution law. This is a project currently under investigation [24].

[†] This magmatic operation $\mu_\times$ allows us to rewrite all the basic operations of Lie–Butcher theory in a simpler way, a way which is also convenient for implementation. See Paper B ([49]) for details.

Chapter 3

Summaries of papers

Summary of Paper A

Hopf algebras of formal diffeomorphisms and numerical integration on manifolds

A. Lundervold and H.Z. Munthe-Kaas

Published in *Contemporary Mathematics,* volume 539, 2011

This paper explores several of the algebraic structures appearing in the study of Lie group integrators: Hopf algebras, Lie series, Lie-Butcher series, Lie idempotents, a noncommutative Faà di Bruno algebra and noncommutative Bell polynomials. It serves both as an introduction to relevant algebraic concepts for numerical analysts, and as an introduction to numerical analysis for algebraists. It is partly a review and partly a research paper. Some of the results in the paper can be found elsewhere in the literature; others are original.

Among other things, the paper gives a purely algebraic way to understand Lie–Butcher theory, in the spirit of the paper [58] by H. Munthe-Kaas and W. Wright. The theory is formulated in terms of the ordered rooted trees OT, together with a few basic operations making it a D-algebra (Section 2.2, Part I). Various representation of flows written in terms of Lie–Butcher series are discussed, and we find algebraic methods for converting between the representations. This involves Lie idempotents and the non-commutative Bell polynomials (slightly reformulated to give an operator we call Q):

Flows $y_0 \mapsto y(t) = \Psi_t(y_0)$ on a homogeneous manifold M can be represented by LB-series in several different ways:

1. In terms of pullback series: Find a character α in $\mathrm{H}_{\mathrm{MKW}}$ such that

$$\Psi(y(t)) = \mathcal{B}_t(\alpha)(y_0)[\Psi] \quad \text{for any } \Psi \in U(\mathfrak{g})^M. \tag{3.1}$$

2. In terms of an autonomous differential equation: Find an infinitesimal character β in $\mathrm{H}_{\mathrm{MKW}}$ such that $y(t)$ solves

$$y'(t) = \mathcal{B}_t(\beta)(y(t)). \tag{3.2}$$

3. In terms of a non-autonomous equation of *Lie type*: Find an infinitesimal character γ in H_{sh} such that $y(t)$ solves

$$y'(t) = \left(\frac{\partial}{\partial t}\mathcal{B}_t(\gamma)(y_0)\right) y(t). \tag{3.3}$$

The relationships between the coefficients α, β and γ in the above LB-series can be expressed as follows:

$\beta = \alpha \circ e$	e is the eulerian idempotent in $\mathrm{H}_{\mathrm{MKW}}$.
$\alpha = \exp^{\diamond}(\beta)$	Exponential wrt. GL-product
$\gamma = \alpha \circ Y^{-1} \circ D$	Dynkin idempotent in $\mathrm{H}_{\mathrm{sh}}(\mathrm{OT})$.
$\alpha = Q(\gamma)$	Q-operator in $\mathrm{H}_{\mathrm{sh}}(\mathrm{OT})$.

Here H_{sh} denotes the shuffle Hopf algebra, and Q is constructed from the Bell polynomials.

Summary of Paper B

Backward error analysis and the substitution law for Lie group integrators

A. Lundervold and H.Z. Munthe-Kaas

Submitted to *Foundations of Computational Mathematics,* 2011.

Paper A ends with a short presentation of the substitution law for Lie–Butcher series, which Paper B develops in full detail. We obtain a formula for the substitution law that can be used to calculate the coefficients of the modified vector fields used in backward error analysis.

The paper continues in the tradition of Paper A by explaining how Lie–Butcher theory is purely algebraic. For example, it points out how all the basic definitions follow from the fact that $\mathrm{N} = \mathbb{R}\langle OT \rangle$ (as defined in Section 2.2 in Part I) is the *free* D-algebra. Then elementary differentials F_f, Lie–Butcher series $\mathcal{B}_f$ and also the substitution law $\star$ can be defined in terms of commutative diagrams:

$$\begin{tikzcd}
\{\bullet\} \arrow[r, hook] \arrow[d, "f"'] & \mathrm{N} \arrow[d, "\mathcal{F}_f"] \\
\mathfrak{g}^M \arrow[r, hook] & U(\mathfrak{g})^M
\end{tikzcd}
\qquad
\begin{tikzcd}
\{\bullet\} \arrow[r, hook] \arrow[d, "f"'] & \mathrm{N}^* \arrow[d, "\mathcal{B}_f"] \\
\mathfrak{g}^M \arrow[r, hook] & U(\mathfrak{g})^M
\end{tikzcd}
\qquad
\begin{tikzcd}
\{\bullet\} \arrow[r, hook] \arrow[d, "a"'] & \mathrm{N} \arrow[d, "a\star"] \\
D(\mathrm{N}) \arrow[r, hook] & \mathrm{N}\,.
\end{tikzcd}$$

A future goal will be to describe the Hopf algebra underlying the substitution law, a project currently under investigation [24].

Summary of Paper C

On pre-Lie-type algebras with torsion

A. Lundervold and H.Z. Munthe-Kaas

Note: This paper has been updated and will be published under the title *On post-Lie algebras, Lie-Butcher series and moving frames*. See `http://arxiv.org/abs/1203.4738`.

The main motivation for this paper comes from the observation that pre-Lie algebras correspond to algebras of affine connections with vanishing curvature and torsion, which is reflected in their use in classical geometric numerical integration in $\mathbb{R}^n$. As we have seen, the role of pre-Lie algebras are taken over by D-algebras when we look at geometric numerical integration on more general manifolds, which may include both curvature and torsion. In this paper we introduce an algebraic formulation for the case of connections with non-vanishing curvature or torsion.

	flat	const. curvature
torsion-free	**PreLie**	**Lie admissible**
const. torsion	**PostLie**	?*

It turns out that the correct algebraic formulation for flat algebras with constant torsion is **post Lie algebras**. This paper relates these to the D-algebras of numerical integration by showing how the universal enveloping algebra of the free post-Lie algebra is isomorphic to the free D-algebra. This opens up a new way to study Lie–Butcher series, more closely related to their character as "Lie-series". It also gives a cleaner way to understand their geometric features.

* The case corresponding to constant curvature and torsion has not yet been discovered.

Bibliography

[1] E. Abe. *Hopf Algebras*. Cambridge University Press, 1980.

[2] A.A. Agrachev and R.V. Gamkrelidze. Chronological algebras and nonstationary vector fields. *Journal of Mathematical Sciences*, 17(1):1650–1675, 1981.

[3] V.I. Arnold. *Mathematical methods of classical mechanics*. Springer, second edition, 1989.

[4] C. Brouder. Runge-Kutta methods and renormalization. *The European Physical Journal C: Particles and Fields*, 12(3):521–534, 2000.

[5] C.J. Budd and M.D. Piggott. Geometric integration and its applications. In P.G. Ciarlet and F. Cucker, editors, *Handbook of numerical analysis: Foundations of Computational Mathematics*, volume 11, pages 35–139. Elsevier, 2003.

[6] J.C. Butcher. Coefficients for the study of Runge-Kutta integration processes. *Journal of the Australian Mathematical Society*, 3(02):185–201, 1963.

[7] J.C. Butcher. An algebraic theory of integration methods. *Mathematics of Computation*, 26(117):79–106, 1972.

[8] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons Inc, second edition, 2008.

[9] D. Calaque, K. Ebrahimi-Fard, and D. Manchon. Two interacting Hopf algebras of trees: A Hopf-algebraic approach to composition and substitution of B-series. *Advances in Applied Mathematics*, 47(2), 2011.

[10] M. Calvo, A. Iserles, and A. Zanna. Runge–Kutta methods on manifolds. In D.F. Griffiths and G.A. Watson, editors, *Numerical Analysis, A.R. Mitchell 75th Birthday Volume*, pages 57–70. World Scientific, 1996.

[11] M.P. Calvo, A. Murua, and J.M. Sanz-Serna. Modified equations for ODEs. *Contemporary Mathematics*, 173:63–74, 1994.

[12] M.P. Calvo and J.M. Sanz-Serna. Canonical B-series. *Numerische Mathematik*, 67(2):161–175, 1994.

[13] P. Cartier. A primer of Hopf algebras. In P. Cartier, B. Julia, P. Moussa, and P. Vanhove, editors, *Frontiers in number theory, physics, and geometry*, volume II, pages 537–615. Springer, 2007.

[14] A. Cayley. On the theory of the analytical forms called trees. *Philosophical Magazine Series 4*, 13(85), 1857.

[15] E. Celledoni, A. Marthinsen, and B. Owren. Commutator-free Lie group methods. *Future Generation Computer Systems*, 19(3):341–352, 2003.

[16] E. Celledoni, R.I. McLachlan, B. Owren, and G.R.W. Quispel. Energy-preserving integrators and the structure of B-series. *Foundations of Computational Mathematics*, 10:673–693, 2010.

[17] F. Chapoton. A rooted-trees q-series lifting a one-parameter family of Lie idempotents. *Algebra & Number Theory*, 3(6):611–636, 2009.

[18] F. Chapoton and M. Livernet. Pre-Lie algebras and the rooted trees operad. *International Mathematics Research Notices*, 2001(8):395–408, 2001.

[19] P. Chartier, E. Hairer, and G. Vilmart. A substitution law for B-series vector fields. *INRIA report*, (5498), 2005.

[20] P. Chartier, E. Hairer, and G. Vilmart. Numerical integrators based on modified differential equations. *Mathematics of Computation*, 76(260):1941, 2007.

[21] A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Communications in Mathematical Physics*, 199(1):203–242, 1998.

[22] P.E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *Journal of Nonlinear Science*, 3(1):1–33, 1993.

[23] A. Dzhumadil'daev and C. Löfwall. Trees, free right-symmetric algebras, free Novikov algebras and identities. *Homology, Homotopy and applications*, 4(2):165–190, 2002.

[24] K. Ebrahimi-Fard, A. Lundervold, D. Manchon, H. Munthe-Kaas, and J.E. Vatne. On the post-Lie operad. *Preprint*, 2011.

[25] K. Ebrahimi-Fard and D. Manchon. A Magnus-and Fer-type formula in dendriform algebras. *Foundations of Computational Mathematics*, 9:1–22, 2009.

[26] K. Ebrahimi-Fard and D. Manchon. Pre-Lie Butcher series. *Preprint*, 2011.

[27] K. Engø. On the construction of geometric integrators in the RKMK class. *BIT Numerical Mathematics*, 40(1):41–61, 2000.

[28] K. Engø and A. Marthinsen. Modeling and solution of some mechanical problems on Lie groups. *Multibody System Dynamics*, 2(1):71–88, 1998.

[29] I. Gelfand, S. Gelfand, V. Retakh, and R.L. Wilson. Quasideterminants. *Advances in Mathematics*, 193(1):56–141, 2005.

[30] I.M. Gelfand and V.S. Retakh. Determinants of matrices over noncommutative rings. *Functional Analysis and Its Applications*, 25(2):91–102, 1991.

[31] M. Gerstenhaber. The cohomology structure of an associative ring. *Annals of Mathematics*, 78(2):267–288, 1963.

[32] R. Grossman and R.G. Larson. Hopf-algebraic structure of families of trees. *J. Algebra*, 126(1):184–210, 1989.

[33] E. Hairer. Backward analysis of numerical integrators and symplectic methods. *Annals of Numerical Mathematics*, 1(1-4):107–132, 1994.

[34] E. Hairer. Important aspects of geometric numerical integration. *Journal of Scientific Computing*, 25(1):67–81, 2005.

[35] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Springer, second edition, 2006.

[36] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving ordinary differential equations I: Nonstiff problems*. Springer, 1993.

[37] E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing*, 13(1):1–15, 1974.

[38] M. E. Hoffman. Combinatorics of rooted trees and hopf algebras. *Transactions of the American Mathematical Society*, 355(9):3795–3812, 2003.

[39] A. Iserles. Numerical methods on (and off) manifolds. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics*, pages 180–189, 1997.

[40] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, and A. Zanna. Lie-group methods. *Acta numerica*, 9:215–365, 2000.

[41] A. Iserles, G.R.W. Quispel, and P.S.P. Tse. B-series methods cannot be volume-preserving. *BIT Numerical Mathematics*, 47(2):351–378, 2007.

[42] A. Iserles and A. Zanna. Qualitative numerical analysis of ordinary differential equations. In J. Renegar, M. Shub, and S. Smale, editors, *The mathematics of numerical analysis: 1995 AMS-SIAM Summer Seminar in Applied*

Mathematics, Utah, volume 32, page 421. American Mathematical Society, 1996.

[43] D. Kreimer. On the Hopf algebra structure of perturbative quantum field theories. *Advances in Theoretical and Mathematical Physics*, 2(2):303–334, 1998.

[44] D. Kreimer. Chen's iterated integral represents the operator product expansion. *Advances in Theoretical and Mathematical Physics*, 3(3):627–670, 1999.

[45] B. Leimkuhler and S. Reich. *Simulating Hamiltonian dynamics*. Cambridge Univ Press, 2004.

[46] D. Lewis and P.J. Olver. Geometric integration algorithms on homogeneous manifolds. *Foundations of Computational Mathematics*, 2(4):363–392, 2002.

[47] A. Lundervold and H. Z. Munthe-Kaas. Hopf algebras of formal diffeomorphisms and numerical integration on manifolds. *Contemporary Mathematics*, 539:295–324, 2011.

[48] A. Lundervold and H.Z. Munthe-Kaas. On pre-Lie-type algebras with torsion and curvature. *Preprint*, 2010.

[49] A. Lundervold and H.Z. Munthe-Kaas. Backward error analysis and the substitution law for Lie group integrators. *Submitted*, 2011. ArXiv preprint math:1106.1071.

[50] W. Magnus. On the exponential solution of differential equations for a linear operator. *Communications on pure and applied mathematics*, 7(4):649–673, 1954.

[51] D. Manchon. Hopf Algebras in Renormalisation. In M. Hazewinkel, editor, *Handbook of Algebra*, volume 5, pages 365–427. North Holland, 2008.

[52] D. Manchon. A short survey on pre-Lie algebras. Available at `http://math.univ-bpclermont.fr/~manchon/biblio/ESI-prelie2009.pdf`, 2009.

[53] R.I. McLachlan and G.R.W. Quispel. Six lectures on the geometric integration of ODEs. In R.A. DeVore, A. Iserles, and S. Endre, editors, *London Math. Soc. Lecture Note Series*, volume 284, pages 155–210. Cambridge Univ. Press, 2001.

[54] R.I. McLachlan and G.R.W. Quispel. Geometric integrators for ODEs. *Journal of Physics A: Mathematical and General*, 39:5251, 2006.

[55] H. Munthe-Kaas. Lie–Butcher theory for Runge–Kutta methods. *BIT Numerical Mathematics*, 35(4):572–587, 1995.

[56] H. Munthe-Kaas. Runge–Kutta methods on Lie groups. *BIT Numerical Mathematics*, 38(1):92–111, 1998.

[57] H. Munthe-Kaas. High order Runge-Kutta methods on manifolds. *Applied Numerical Mathematics*, 29(1):115–127, 1999.

[58] H. Munthe-Kaas and W. Wright. On the Hopf algebraic structure of Lie group integrators. *Foundations of Computational Mathematics*, 8(2):227–257, 2008.

[59] H. Munthe-Kaas and A. Zanna. Numerical integration of differential equations on homogeneous manifolds. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics*, 1997.

[60] B. Owren. Order conditions for commutator-free Lie group methods. *Journal of Physics A: Mathematical and General*, 39, 2006.

[61] B. Owren and A. Marthinsen. Runge–Kutta methods adapted to manifolds and based on rigid frames. *BIT Numerical Mathematics*, 39(1):116–142, 1999.

[62] R.S. Palais. A global formulation of the Lie theory of transformation groups. *Memoirs of the AMS*, 22, 1957.

[63] C. Reutenauer. *Free Lie algebras*. Oxford University Press, 1993.

[64] J.M. Sanz-Serna and M.P. Calvo. *Numerical Hamiltonian problems*. Chapman & Hall/CRC, 1994.

[65] D. Segal. Free Left-Symmetrical Algebras and an Analogue of the Poincaré-Birkhoff-Witt Theorem. *Journal of Algebra*, 164(3):750–772, 1994.

[66] R.W. Sharpe. *Differential geometry: Cartan's generalization of Klein's Erlangen program*. Springer, 1997.

[67] M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume 1. Publish or Perish, third edition, 2005.

[68] M.E. Sweedler. *Hopf algebras*. W.A. Benjamin, 1969.

[69] P.S.P. Tse. *Geometric Numerical Integration: On the Numerical Preservation of Multiple Geometric Properties for Ordinary Differential Equations*. PhD thesis, La Trobe University, 2007.

[70] B. Vallette. Homology of generalized partition posets. *Journal of Pure and Applied Algebra*, 208(2):699–725, 2007.

[71] G. Vilmart. *Étude d'intégrateurs géométriques pour des équations différentielles*. PhD thesis, Université de Genève, 2008.

[72] E.B. Vinberg. Convex homogeneous cones. *Transactions of the Moscow Mathematical Society*, 12:340–403, 1963.

[73] R.F. Warming and B.J. Hyett. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of Computational Physics*, 14(2):159–179, 1974.

[74] J.H. Wilkinson. Error analysis of floating-point computation. *Numerische Mathematik*, 2(1):319–340, 1960.

[75] J. Wisdom and M. Holman. Symplectic maps for the N-body problem. *The Astronomical Journal*, 102:1528–1538, 1991.

Part II

Included Papers

Paper A

Hopf algebras of formal diffeomorphisms and numerical integration on manifolds [*]

[*] First published in *Contemporary Mathematics*, volume 539, 2011, published by the American Mathematical Society
arXiv: http://arxiv.org/abs/0905.0087

Hopf algebras of formal diffeomorphisms and numerical integration on manifolds

Alexander Lundervold and Hans Munthe-Kaas

Department of Mathematics, University of Bergen,
Johannes Brunsgate 12, N-5008 Bergen, Norway
{alexander.lundervold,hans.munthe-kaas}@math.uib.no

Abstract

B-series originated from the work of John Butcher in the 1960s as a tool to analyze numerical integration of differential equations, in particular Runge–Kutta methods. Connections to renormalization theory in perturbative quantum field theory have been established in recent years. The algebraic structure of classical Runge–Kutta methods is described by the Connes–Kreimer Hopf algebra.

Lie–Butcher series are generalizations of B-series that are aimed at studying Lie-group integrators for differential equations evolving on manifolds. Lie group integrators are based on general Lie group actions on a manifold, and classical Runge–Kutta integrators appear in this setting as the special case of $\mathbb{R}^n$ acting upon itself by translations. Lie–Butcher theory combines classical B-series on $\mathbb{R}^n$ with Lie-series on manifolds, and the underlying Hopf algebra H_N combines the Connes–Kreimer Hopf algebra with the shuffle Hopf algebra of free Lie algebras.

Aimed at a general mathematical audience, we give an introduction to Hopf algebraic structures and their relationship to structures appearing in numerical analysis. In particular, we explore the close connection between Lie series, time-dependent Lie series and Lie–Butcher series for diffeomorphisms on manifolds. The role of the Euler and Dynkin idempotents in numerical analysis is discussed. A non-commutative version of a Faà di Bruno bialgebra is introduced, and the relation to non-commutative Bell polynomials is explored.

Contents

1 Outline 1

2 Introduction to numerical integrators and their analysis 2
2.1 Classical integrators . . . 2
2.2 Lie group integrators . . . 4
2.2.1 Exponential Euler method . . . 4
2.2.2 Choosing a good action . . . 5
2.2.3 Higher order methods . . . 5

3 Hopf algebras 6
3.1 Basic definitions . . . 6
3.2 Examples: The concatenation and shuffle Hopf algebras . . . 8
3.3 Characters and endomorphisms . . . 9
3.3.1 Infinitesimal characters, the exponential and the logarithm . . . 10
3.3.2 Eulerian idempotent . . . 10
3.3.3 The graded Dynkin operator . . . 11

4 Algebras of formal diffeomorphisms on manifolds 12
4.1 Autonomous Lie series . . . 12
4.2 Time-dependent Lie series . . . 13
4.2.1 Non-commutative Bell polynomials and the Dynkin–Faà di Bruno bialgebra 13
4.2.2 Pullback along time-dependent flows . . . 16
4.3 Lie–Butcher theory . . . 17
4.3.1 Differential operators in $U(\mathcal{XM})$ expanded in a non-commuting frame . . . 17
4.3.2 The free D-algebra and elementary differentials . . . 19
4.3.3 A generalized Connes–Kreimer Hopf algebra of planar trees . . . 20
4.3.4 Lie–Butcher series and flows on manifolds . . . 21
4.3.5 Relations to classical B-series . . . 23
4.4 Substitution law for LB-series . . . 24

5 Final remarks and outlook 24

1 Outline

The main objective of this paper is to explore algebraic structures underlying groups of formal diffeomorphisms on manifolds. The focus is on some important mathematical structures appearing in numerical integration on manifolds that are likely to find applications also in other areas of mathematics. The relationship between classical Lie series on manifolds, time-dependent Lie series and Lie–Butcher series is explained in detail. We develop the algebraic structures introduced in [38, 39, 40, 42, 45, 44, 3], and in particular explore connections between Hopf- and Lie algebras, differential geometry and analysis of numerical integration on manifolds. The paper does not go into a detailed study of the many applications of these algebraic structures in numerical analysis, but we give some brief sketches.

The introductory Chapters 2 and 3 contains an overview of well-known results. Chapter 2 contains a concise introduction to numerical integration and algebraic structures appearing in numerical analysis, both classical methods on $\mathbb{R}^n$ and Lie group methods generalizing to manifolds. Chapter 3 presents an introduction to Hopf algebraic structures.

Chapter 4 contains more new and recent material. It details the algebraic structures of Lie–Butcher theory, and discusses the interplay between algebraic and differential geometric points of view. In particular, we want to emphasize the strong connections between the algebraic theory of Lie series, time-dependent Lie series and Lie–Butcher series. Chapter 4 therefore starts with a discussion of classical Lie series and pullback formulas on manifolds, continuing with an exploration of some less known time-dependent pullback formulas. We will explain the relevance of the Euler and Dynkin idempotents, and introduce a non-commutative *Dynkin–Faà di Bruno* bialgebra, related to non-commutative Bell polynomials appearing in various contexts in earlier works: numerical analysis [39], control theory [37] and quantization [31]. This Dynkin–Faà di Bruno bialgebra is related to, but different from, the Hopf algebras explored by Brouder et. al. in [6].

In the final part of Chapter 4, we turn to Lie–Butcher series. We explore backward error analysis and the substitution law in the setting of algebras of non-commuting frames on manifolds. Although we will not give detailed expositions and applications of these subjects we hope that this presentation will systematize the theory, opening it up for further research.

2 Introduction to numerical integrators and their analysis

Let $\mathcal{M}$ be a manifold and $F\colon \mathcal{M} \to T\mathcal{M}$ a vector field. By the flow of an autonomous vector field F we mean the diffeomorphism $\Phi_{t,F}\colon \mathcal{M} \to \mathcal{M}$, defined for $t \in \mathbb{R}$ such that $\Phi_{s,F} \circ \Phi_{t,F} = \Phi_{s+t,F}$, $\Phi_{0,F} = \mathrm{Id}$ and $\partial/\partial t|_{t=0}\, \Phi_{t,F}(p) = F(p)$ for all $p \in \mathcal{M}$.

Numerical integration of ODEs is about constructing good numerical approximations to $\Phi_{t,F}$ for a given vector field F. A numerical integration algorithm yields a diffeomorphism $\Psi_{h,F}$, henceforth called the *numerical integrator*. The real parameter h is called the *step size*. For an initial point $y_0 \in \mathcal{M}$, and a chosen step size $h > 0$, the numerical method produces a discrete sequence of solution points $y_i = \Psi_{h,F}(y_{i-1})$, with the goal of arriving at $y_k \approx \Phi_{kh,F}(y_0)$. Note that, unlike the exact flow, numerical integrators are *not* 1-parameter Lie groups in h. In general we have $\Psi_{h,F} \circ \Psi_{s,F} \neq \Psi_{h+s,F}$, and $\Psi_{-h,F} \neq \Psi_{h,F}^{-1}$. Integrators for which the latter identity holds are called *(time-)symmetric* methods. Most integrators satisfy the consistency conditions $\Psi_{0,F} = \mathrm{Id}$ and $\partial/\partial t|_{t=0}\, \Psi_{t,F}(p) = F(p)$ as well as scaling homogeneity $\Psi_{h,F} = \Psi_{1,hF}$.

Many algebraic aspects of numerical integration are related to the computation of compositions, logarithms and exponentials of numerical integrators. In this introduction we will introduce some basic algebraic structures arising in the analysis of numerical integrators. In particular we will focus on structures that originate from the study of *Lie group integrators*, which are numerical integrators on general manifolds. The resulting theory combines Lie theory with the classical Butcher theory that describes numerical integrators on $\mathbb{R}^n$.

This first section presents a survey of well known results from numerical analysis. A detailed understanding of this introductory section is not necessary for reading the rest of the paper, and readers mainly interested in algebraic structures may jump directly to Section 3.

2.1 Classical integrators

In the early 1960s, John Butcher set out to explore the algebraic territory of numerical algorithms for integrating ODEs evolving on vector spaces

$$y'(t) = F(y), \qquad y \in \mathbb{R}^n, \quad F\colon \mathbb{R}^n \to \mathbb{R}^n. \tag{2.1}$$

In particular he studied the family of Runge–Kutta methods. Given a time step $h \in \mathbb{R}$, these methods advance the solution from $y_0 = y(0)$ to $y_1 \approx y(h)$ as:

```
for  $r = 1 : s$ do
    $Y_r = \sum_{k=1}^{s} a_{rk} F_k + y_0$
    $F_r = hF(Y_r)$
end
$y_1 = \sum_{k=1}^{s} b_k F_k + y_0\,.$
```

This basic step is iterated: $y_0 \mapsto y_1 \mapsto \ldots \mapsto y_n$, with constant or variable step sizes h, until the final solution $y_n \approx y(t_n)$ is reached. The coefficients a_{rk} and b_k for $r, k \in \{1, \ldots, s\}$ define a particular s-stage RK method.

A goal of numerical analysis is to characterize coefficients a_{rk} and b_k that yield 'good' methods (when applied to a given class of differential equations). The view of what a good integration method is has, however, evolved over the last decades. Traditionally, *order theory* and *stability* were the most important properties to consider. A numerical integrator is of order p if the first $p+1$ terms of the Taylor expansion of the analytical solution agrees with the first $p+1$ terms of the numerical method (developed in the parameter h). Requiring a certain order results in algebraic conditions, called *order conditions*, on the coefficients of the method.

Many numerical methods for solving the equation (2.1) can be studied by using *B-series* (see e.g. [25]), introduced by Hairer and Wanner in 1974 [26]. A B-series is a (formal) series indexed over the set of rooted trees T, and can for a vector field F be written as

$$B_h(a)(y) = a(\mathbb{I})y + \sum_{\tau \in T} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau)\mathcal{F}_F(\tau)(y). \tag{2.2}$$

Here a is a map $a : T \to \mathbb{R}$, $\mathbb{I}$ is the empty tree, $|\tau|$ is the number of vertices of τ (the *order* of the tree τ) and σ is a certain symmetry factor. The map $\mathcal{F}_F(\tau) : \mathbb{R}^n \to \mathbb{R}^n$ is the *elementary differential* of the tree τ and is given recursively as follows:

$$\mathcal{F}_F(\tau) = F^{(m)}\left(\mathcal{F}_F(\tau_1)(y), ..., \mathcal{F}_F(\tau_m)(y)\right)(y), \tag{2.3}$$

where $\tau = B^+(\tau_1, ..., \tau_m)$ is the tree constructed by adding a common root to the subtrees $\tau_1 \dots \tau_m$, and $F^{(m)}$ is the mth derivative of the vector field.

One way in which B-series can be applied to the study of numerical methods is to order theory. For example, the order conditions for Runge–Kutta methods can easily be obtained by writing the method as a B-series and then comparing the coefficients of this series with the exact solution written as a B-series (see e.g. [25, Chapter. III.1.2]).

The composition of Runge–Kutta methods is also of great interest, and this leads to the study of the composition of B-series. A series $B_{hF}(a)$ is inserted into another series $B_{hF}(b)$, which gives the B-series $B_{hF}(a)(B_{hF}(y)(b)) = B_{hF}(a \cdot b)(y)$. The resulting product $a \cdot b$ gives rise to a group, called the *Butcher group* [9, 16].

Butcher realized early on that the set of Runge–Kutta methods forms a group, and characterized algebraically the composition and inverse in this group. Much later, this group was identified with the character group of the Connes–Kreimer Hopf algebra [17, 19, 5].

In recent years the importance of preserving various geometric properties of the underlying continuous dynamical system has become better understood. The research topic *Geometric Numerical Integration* [25] emphasizes this view. Geometric integration algorithms have been successfully developed for various classes of differential equations, such as volume preserving flows, Hamiltonian equations, systems with first integrals and equations evolving on manifolds. An important tool for investigating the geometrical properties of a numerical integrator is through *backward error analysis.* For a given numerical method $\Psi_{h,F}$, we seek a series expansion of a modified vector field $(h, F) \mapsto \tilde{F}_h$ such that the numerical solution equals[1] the analytical flow of the modified vector field: $\Psi_{h,F} = \Phi_{t,\tilde{F}_h}\Big|_{t=h}$.

This is computed as a formal logarithm $\tilde{F}_h = \mathrm{Log}(\Psi_{h,F})$, which in Hopf algebraic language is expressed by the *Eulerian idempotent* (Section 3.3.2).

Still another idea, which has been developed in [14], is to ask for a series development of a modified vector field $\overline{F}_h$ such that when the numerical method is applied to $\overline{F}_h$, the exact analytical solution is produced: $\Psi_{h,\overline{F}_h} = \Phi_{h,F}$. This has been taken much further in recent work [15, 10]. The algebraic operation $(h, F) \mapsto \overline{F}_h$ is commonly referred to as a *substitution law.* The Hopf algebra of the substitution law is introduced in [10].

[1] The series for $\tilde{F}_h$ is a formal series which usually does not converge. By truncating the series at an optimal point we find a modified equation which is exponentially close to the numerical solution, see [25]. In this paper we deal only with formal series, and convergence is not considered.

The theory of B-series is often a very important component in a numerical analyst's toolbox, and is used to study all of the above: order theory, backward error analysis, modified vector fields and structure preserving properties of numerical integrators.

2.2 Lie group integrators

Numerical Lie group integrators for ODEs is a generalization of numerical integration of ODEs from the classical setting of equations on $\mathbb{R}^n$ to differential equations on manifolds. See [28] for an extensive survey.

2.2.1 Exponential Euler method

Let $\mathcal{M}$ denote a manifold, $\mathcal{XM}$ its Lie algebra of vector fields with the Jacobi bracket and $\mathrm{Diff}(\mathcal{M})$ the group of diffeomorphisms on $\mathcal{M}$. Let $\exp : \mathcal{XM} \to \mathrm{Diff}(\mathcal{M})$ denote the flow operator. We want to numerically integrate an ODE on $\mathcal{M}$ given as

$$y'(t) = F(y), \quad y(0) = y_0 \quad \text{for } F \in \mathcal{XM}, \tag{2.4}$$

with the analytical solution $y(t) = \exp(tF)\cdot y_0$. Here $\exp(tF)\cdot y_0$ denotes the evaluation of the diffeomorphism $\exp(tF)$ at $y_0 \in \mathcal{M}$.

Assumption 2.1. *The fundamental assumption for numerical Lie group integrators is the existence of a subalgebra $\mathfrak{g} \subset \mathcal{XM}$ such that*

- *All vector fields $V \in \mathfrak{g}$ can be exponentiated exactly.*
- *The Lie algebra $\mathfrak{g}$ defines a* frame *on $T\mathcal{M}$, i.e. $\mathfrak{g}$ spans the tangentspace $T_p\mathcal{M}$ at all points $p \in \mathcal{M}$. In other words, the action generated by $\mathfrak{g}$ is transitive on $\mathcal{M}$.*

The vector fields in $\mathfrak{g}$ are called the *frozen vector fields.* Due to the frame assumption, we can always express the vector field F and the ODE (2.4) in terms of frozen vector fields via a function $f : \mathcal{M} \to \mathfrak{g}$ as $F(y) = f(y)\cdot y$, where $f(y) \in \mathcal{XM}$ and $f(y)\cdot y$ denotes evaluation of this vector field in y. Thus (2.4) can be written in the form

$$y'(t) = f(y)\cdot y, \quad y(0) = y_0 \in \mathcal{M}. \tag{2.5}$$

In the case where $\mathfrak{g}$ forms a basis for $T_y\mathcal{M}$, the function $f(y)$ is uniquely defined. In more general situations, $\mathfrak{g}$ is an overdetermined frame for $T_y\mathcal{M}$, and there is a freedom in the choice of f. This is called a choice of isotropy, and is of major importance for the quality of the numerical integrator.

With the equation written as (2.5), we can present the simplest of all Lie group integrators: the *exponential Euler method.* Given a time step $h \in \mathbb{R}$, the method advances the solution from $y_0 = y(0)$ to $y_1 \approx y(h)$ as:

Algorithm 2.2 (Exponential Euler)**.**

$$y_1 = \exp(hf(y_0))\cdot y_0.$$

In each step the solution is advanced $y_k \mapsto y_{k+1}$ by integrating the frozen vector field equation

$$y'(t) = f(y_k)\cdot y, \quad y(0) = y_k$$

from $t = 0$ to $t = h$. We will in the sequel present methods of higher order and with superior qualities compared to this simple scheme. The main theme of the paper is the algebraic structures arising from the numerical analysis of such integration schemes.

2.2.2 Choosing a good action

In practice it is of importance that $\exp\colon \mathfrak{g} \to \mathrm{Diff}(\mathcal{M})$ can be computed fast, and furthermore that the given vector field $F(y)$ is locally well approximated by $f(y_0)\cdot y$. Exactly what 'well' means depends on what we want to achieve. In many situations we can choose $\mathfrak{g}$ so that certain first integrals of the original system are exactly preserved by the frozen flows. Choosing $\mathfrak{g}$ and $f : \mathcal{M} \to \mathfrak{g}$ is in many ways similar to choosing a *preconditioner* in iterative methods for solving linear equations: we want a good approximation which is easy to compute.

A simple choice of $\mathfrak{g}$ is obtained by embedding $\mathcal{M} \subset \mathbb{R}^N$ and choosing $\mathfrak{g} = \mathbb{R}^N$ as the (commutative) algebra generated by $\{\partial/\partial x_j\}_{j=1}^N$, i.e., the constant vector fields on $\mathbb{R}^N$. Since the vector fields are constant, we have $f(y_0)y = f(y_0)$, so the function f simply becomes $f(y_0) = F(y_0) \in \mathbb{R}^N$, all commutators in $\mathfrak{g}$ vanish and the exponential on $\mathfrak{g}$ is $\exp(V)\cdot p = V + p$ for $V, p \in \mathbb{R}^N$. In this case all Lie group integrators will reduce to classical integrators, e.g. exponential Euler becomes the classical Euler $y_1 = hF(y_0) + y_0$.

The other extreme is $\mathfrak{g} = \mathcal{XM}$ and $f(y) = F$ for all y, in which case exponential Euler yields the analytical solution exactly. However, the computation of the exponential on $\mathfrak{g}$ is just as difficult as solving the original equation. We seek efficient choices in between these two extremes.

In many cases $\mathfrak{g}$ is given as the infinitesimal generators of a (e.g. left) Lie group action on $\mathcal{M}$. For example, consider the sphere $\mathcal{M} = S^2$ acted upon from left by the group $G = \mathrm{SO}(3)$ of orthogonal 3×3 matrices, whose Lie algebra $\mathfrak{so}(3)$ consists of skew 3×3 matrices. Any matrix $V \in \mathfrak{so}(3)$ is uniquely identified with the infinitesimal generator[2] $\xi_V \in \mathcal{XM}$, given by matrix-vector multiplication $\xi_V\cdot y = Vy$ for $y \in S^2$. Therefore (2.5) becomes $y'(t) = V(y)y$, where $V(y)$ is a skew symmetric matrix. The exponentiation is related to the matrix exponential $\mathrm{expm}(V)$ as $\exp(\xi_V)\cdot y = \mathrm{expm}(V)y$.

Another important example of group actions arise in the solution of *isospectral differential equations*, where $\mathrm{GL}(n)$ acts on $\mathfrak{gl}(n)$ by the *adjoint action* (similarity transform) $A\cdot Y = AYA^{-1}$ for $A \in \mathrm{GL}(n)$, $Y \in \mathfrak{gl}(n)$. In these problems $\mathcal{M} \subset \mathfrak{gl}(n)$ is one of the (isospectral) orbits of the action. For this action (2.5) acquires the isospectral form $Y'(t) = [B(Y), Y]$ for some $B(Y) \in \mathfrak{gl}(n)$. Since the action is a similarity transform it is guaranteed that all the eigenvalues of $Y(t)$ are preserved also by the numerical integrator.

Yet another example, which occurs in *Lie–Poisson problems* in computational mechanics, is the *coadjoint action* of a Lie group on the dual of its Lie algebra, $\mathfrak{g}^*$. In this case $\mathcal{M} \subset \mathfrak{g}^*$ is a coadjoint orbit. Using this action we can guarantee that the numerical Lie group integrator exactly preserves the Casimirs of the continuous system.

For other problems it may be advantageous to choose $\mathfrak{g}$ by simplifying the original equation to a family of integrable equations. An example is the computation of the motion of charged particles in a magnetic field. The solution in the case of constant magnetic fields is given by helical motions around the field lines. The corresponding Lie algebra yields fast and accurate Lie group integrators for the full problem of non-constant magnetic fields. Another example is integration of a spinning top, where we obtain simpler equations by considering the direction of gravity as being constant in body coordinates. In both these problems, the action preserves important first integrals of the system. A third example is integration of stiff equations on $\mathbb{R}^n$, where an integrable Lie algebra is obtained by considering all affine linear vector fields. This connects the theory of Lie group integrators with the so-called *exponential integrators.* See [28] for details.

2.2.3 Higher order methods

Most Lie group methods for integrating (2.5) are built from linear operations and commutators in $\mathfrak{g}$ and compution of flows of frozen vector fields (exponentials). Runge–Kutta type methods with basic motions expressed in terms of an exponential of a sum of elements in $\mathfrak{g}$ are commonly referred to as RKMK methods [28], as in the following example:

[2] Recall that the identification of the Lie algebra of a left group action with the infinitesimal generator in $\mathcal{XM}$ is an anti-homomorphism, $[\xi_V, \xi_W] = -\xi_{[V,W]}$. In this paper the brackets are Jacobi brackets on $\mathcal{XM}$, and some signs may differ when compared to cited papers.

Algorithm 2.3 (4th order RKMK from [39]).

$$\begin{array}{ll}
Y_1 = y_0 & F_1 = hf(Y_1) \\
Y_2 = \exp(\frac{1}{2}F_1)\cdot y_0 & F_2 = hf(Y_2) \\
Y_3 = \exp(\frac{1}{2}F_2 + \frac{1}{24}[F_1, F_2])\cdot y_0 & F_3 = hf(Y_3) \\
Y_4 = \exp(F_3 + \frac{1}{6}[F_1, F_3])\cdot y_0 & F_4 = hf(Y_4) \\
V = \frac{1}{6}F_1 + \frac{1}{3}(F_2 + F_3) + \frac{1}{6}F_4 & I = \frac{1}{8}F_1 + \frac{1}{12}(F_2 + F_3) - \frac{1}{24}F_4 \\
y_1 = \exp(V + [I, V])\cdot y_0. &
\end{array}$$

Methods where the basic motions are products of exponentials of simple elements in $\mathfrak{g}$ are called *Crouch–Grossman methods* [18, 45].

Algorithm 2.4 (3rd order Crouch–Grossman method from [45]).

$$\begin{array}{ll}
Y_1 = y_0 & F_1 = hf(Y_1) \\
Y_2 = \exp(\frac{3}{4}F_1)\cdot y_0 & F_2 = hf(Y_2) \\
Y_3 = \exp(\frac{119}{216}F_2)\cdot\exp(\frac{17}{108}F_1)\cdot y_0 & F_3 = hf(Y_3) \\
y_1 = \exp(\frac{13}{51}F_3)\cdot\exp(-\frac{2}{3}F_2)\cdot\exp(\frac{24}{17}F_3)y_0. &
\end{array}$$

More recently methods have been developed which combine exponentials of sums and products of exponentials, as in the *commutator free Lie group methods* [13]. An example is:

Algorithm 2.5 (4th order commutator free method from [13]).

$$\begin{array}{ll}
Y_1 = y_0 & F_1 = hf(y_0) \\
Y_2 = \exp(\frac{1}{2}F_1)\cdot y_0 & F_2 = hf(Y_2) \\
Y_3 = \exp(\frac{1}{2}F_2)\cdot y_0 & F_3 = hf(Y_3) \\
Y_4 = \exp(-\frac{1}{2}F_1 + F_3)\cdot Y_2 & F_4 = hf(Y_4) \\
\multicolumn{2}{l}{y_1 = \exp(\frac{1}{4}F_1 + \frac{1}{6}(F_2 + F_3) - \frac{1}{12}F_4)\cdot\exp(-\frac{1}{12}F_1 + \frac{1}{6}(F_2 + F_3) + \frac{1}{4}F_4)\cdot y_0.}
\end{array}$$

For equations of *Lie type*, $y'(t) = f(t) \cdot y$, numerical methods based on Magnus and Fer expansions have been developed in [29, 27], and the algebraic theory has recently been developed further in [21].

To study order conditions, backward error analysis and structure preservation of such methods, it is important to understand B-series in a general setting of group actions on manifolds. A first attempt at combining Lie and B-series in a common mathematical framework appeared in [38, 39]. Hopf algebraic aspects have been explored further in [40, 3, 42].

3 Hopf algebras

This section gives a short collection of some facts and properties of Hopf algebras that we will use in this work. For a more thorough introduction, see e.g. [1], [35], [48], [30], [11].

3.1 Basic definitions

Let k be a field containing $\mathbb{Q}$.

Definition 3.1. A *k-algebra* A consists of a k-vector space A together with two maps $\mu : A\otimes A \to A$ and $\eta : k \to A$, called the *product* and the *unit* of A, such that:

(i) μ is associative, i.e. $\mu \circ (I \otimes \mu) = \mu \circ (\mu \otimes I)$,

(ii) the composites $A \cong A \otimes k \xrightarrow{I\otimes\eta} A \otimes A \xrightarrow{\mu} A$ and $A \cong A \otimes k \xrightarrow{\eta\otimes I} A \otimes A \xrightarrow{\mu} A$ both equal I.

Here I denotes the identity map.

An algebra A is called commutative if $\mu \circ \tau = \mu$, where τ is the *flip* map $\tau(a_1 \otimes a_2) = a_2 \otimes a_1$.

Definition 3.2. A *k-coalgebra* C is a k-vector space equipped with two maps $\Delta : C \to C \otimes C$ and $\epsilon : C \to k$, the *coproduct* and the *counit*, such that:

(i) Δ is coassociative, i.e. $(\Delta \otimes I) \circ \Delta = (I \otimes \Delta) \circ \Delta$,

(ii) the composites $C \overset{\Delta}{\to} C \otimes C \xrightarrow{I \otimes \epsilon} C \otimes k \cong C$ and $C \overset{\Delta}{\to} C \otimes C \xrightarrow{\epsilon \otimes I} k \otimes C \cong C$ both equal I.

A coalgebra C is called cocommutative if $\tau \circ \Delta = \Delta$.

Definition 3.3. A *bialgebra* H over k is a k-vector space equipped with both an algebra (H, μ, η) and a coalgebra structure (H, Δ, ϵ), such that the coproduct $\Delta : H \to H \otimes H$ and the counit $\epsilon : H \to k$ are algebra morphisms. These compatibility conditions can be expressed in terms of the following commutative diagrams[3], where τ denotes the *flip* operation $\tau(h_1, h_2) = (h_2, h_1)$:

$$\begin{tikzcd}
H^{\otimes 4} \arrow[rr, "I \otimes \tau \otimes I"] & & H^{\otimes 4} \arrow[d, "\mu \otimes \mu"] \\
H \otimes H \arrow[u, "\Delta \otimes \Delta"] \arrow[r, "\mu"'] & H \arrow[r, "\Delta"'] & H \otimes H
\end{tikzcd}
\qquad
\begin{tikzcd}
H \otimes H \arrow[r, "\epsilon \otimes \epsilon"] \arrow[d, "\mu"'] & k \otimes k \arrow[d, "\cong"] \\
H \arrow[r, "\epsilon"'] & k
\end{tikzcd}$$

A bialgebra H is called commutative if it is commutative as an algebra, and cocommutative if it is cocommutative as a coalgebra.

Remark 3.4. There is symmetry in the definition of a bialgebra. Rather than requiring the coalgebra structure to respect the algebra structure in the above sense, we could have switched the role of the two structures. To complete the symmetry, we could in addition reverse the arrows in the two diagrams above. This would result in an equivalent definition.

Grading. Let H be a *graded* k-vector space, i.e. $H = \bigoplus_{n \geq 0} H_n$. There is a notion of a graded bialgebra, obtained by requiring the following of the algebra and coalgebra structure, respectively:

(i) $\mu(H_p, H_q) \subset H_{p+q}$

(ii) $\Delta(H_n) \subset \bigoplus_{p+q=n} H_p \otimes H_q$.

The grading of an algebra H gives rise to the *grading operator* $Y : H \to H$ given by

$$Y : h \mapsto \sum_{k \geq 0} k h_k,$$

where $h = \sum_{n \geq 0} h_n \in \bigoplus_{n \geq 0} H_n$. A graded bialgebra $H = \bigoplus_{n \geq 0} H_n$ is called *connected* if $H_0 = k$.

Proposition 3.5 ([35]). *Let H be a connected, graded bialgebra. Then, for any $x \in H_n$, $n \geq 0$, we have:*

$$\Delta x = 1 \otimes x + x \otimes 1 + \tilde{\Delta} x, \text{ where } \tilde{\Delta} x \in \bigoplus_{p+q=n,\ p,q>0} H_p \otimes H_q.$$

We will often use the *Sweedler notation* for the coproduct:

$$\Delta x = \sum_{(x)} x_{(1)} \otimes x_{(2)} \quad \text{and} \quad \tilde{\Delta} x = \sum_{(x)} x' \otimes x''.$$

[3] All diagrams were created using Paul Taylor's diagram package, available from http://www.paultaylor.eu/diagrams/

Definition 3.6. A *Hopf algebra* is a bialgebra $(H, \mu, \eta, \Delta, \epsilon)$ together with an antihomomorphism S on H, called the *antipode*, with the property given by the commutativity of the following diagram:

$$\begin{tikzcd}
& H \otimes H \arrow[rr, "S\otimes 1"] & & H \otimes H \arrow[dr, "\mu"] & \\
H \arrow[ur, "\Delta"] \arrow[dr, "\Delta"'] \arrow[rr, "\varepsilon"] & & k \arrow[rr, "\eta"] & & H \\
& H \otimes H \arrow[rr, "1\otimes S"'] & & H \otimes H \arrow[ur, "\mu"'] &
\end{tikzcd}$$

A Hopf algebra is graded if it is graded as a bialgebra and the antipode satisifies $S(H_n) \subset H_n$. If a bialgebra is graded and connected, then it is automatically a graded Hopf algebra:

Proposition 3.7 ([35]). *Any connected graded bialgebra is a Hopf algebra. The antipode S is given recursively by $S(1) = 1$ and*

$$S(x) = -x - \sum_{(x)} S(x')x''$$

for $x \in \ker \epsilon$.

3.2 Examples: The concatenation and shuffle Hopf algebras

Recurring in the sequel are Hopf algebras built from letters in an alphabet. We follow the notation of Reutenauer [47]. Consider a finite or infinite alphabet of letters $\mathcal{A} = \{a, b, c, \ldots\}$. We write $\mathcal{A}^*$ for the collection of all empty or non-empty words over $\mathcal{A}$, where $\mathbb{I}$ is the empty word. Let $k\langle\mathcal{A}\rangle$ be the k-algebra of non-commutative polynomials in $\mathcal{A}$. A polynomial $P \in k\langle\mathcal{A}\rangle$ will be written as a sum

$$P = \sum_{\omega \in A^*} (P, \omega)\omega,$$

where $(P, \omega) \in k$ is non-zero only for a finite number of ω. Let $P, Q \in k\langle\mathcal{A}\rangle$. The product of P and Q, written as PQ, has coefficients

$$(PQ, \omega) = \sum_{\omega = uv} (P, u)(Q, v).$$

The k-linear dual space denoted $k\langle\langle\mathcal{A}\rangle\rangle := \mathrm{Hom}_k(k\langle\mathcal{A}\rangle, k)$ is identified with all infinite k-linear combinations of words. An $\alpha \in k\langle\langle\mathcal{A}\rangle\rangle$ can be written as an infinite series

$$\alpha = \sum_{\omega \in \mathcal{A}^*} (\alpha, \omega)\omega,$$

where $(\alpha, \omega) := \alpha(\omega) \in k$ and $(\cdot, \cdot)$ is the dual pairing defined such that words in $\mathcal{A}^*$ are orthogonal, $(\omega_1, \omega_2) = \delta_{\omega_1, \omega_2}$ for all $\omega_1, \omega_2 \in \mathcal{A}^*$.

We define two different associative products on $k\langle\mathcal{A}\rangle$. The *concatenation product* $\omega_1, \omega_2 \mapsto \omega_1\omega_2$ obtained by concatenation of words and the *shuffle product* $\omega_1, \omega_2 \mapsto \omega_1 \sqcup\!\sqcup\, \omega_2$ obtained by linearly combining all possible *shuffles* of the two words i.e. combinations where the letters within each word are not internally permuted:

$$abc \sqcup\!\sqcup\, de = abcde + abdce + adbce + dabce + abdec + adbec + dabec + adebc + daebc + deabc.$$

The shuffle product can be defined recursively as

$$(a\omega_1) \sqcup\!\sqcup\, (b\omega_2) = a(\omega_1 \sqcup\!\sqcup\, b\omega_2) + b(a\omega_1 \sqcup\!\sqcup\, \omega_2),$$

where $a, b \in \mathcal{A}$ and $\omega_1, \omega_2 \in \mathcal{A}^*$. The unit of both concatenation and shuffle is the empty word $\mathbb{I}$.

By dualization of these products we obtain the *deconcatenation* and the *deshuffle* coproducts. The deconcatenation coproduct $\Delta_d \colon k\langle\mathcal{A}\rangle \to k\langle\mathcal{A}\rangle \otimes k\langle\mathcal{A}\rangle$ is defined for $\omega = a_1 a_2 \cdots a_k \in \mathcal{A}^*$ as:

$$\Delta_d(\omega) = \sum_{i=1}^{k} a_1 \cdots a_i \otimes a_{i+1} \cdots a_k. \tag{3.1}$$

This coproduct is the dual of the concatenation product, so for any $P, Q \in k\langle\mathcal{A}\rangle$

$$(PQ, \omega) = (P \otimes Q, \Delta_d(\omega)) = \sum_{(\omega)_{\Delta_d}} (P, \omega_{(1)})(Q, \omega_{(2)}).$$

The deshuffle product $\Delta_{\sqcup} \colon k\langle\mathcal{A}\rangle \to k\langle\mathcal{A}\rangle \otimes k\langle\mathcal{A}\rangle$ is similarly defined such that

$$(P \sqcup\!\sqcup\, Q, \omega) = (P \otimes Q, \Delta_{\sqcup}(\omega)) = \sum_{(\omega)_{\Delta_{\sqcup}}} (P, \omega_{(1)})(Q, \omega_{(2)}).$$

The two coproducts can also be characterized by requiring that the letters in the alphabet $\mathcal{A}$ are primitive, i.e. that $\Delta(a) = 1 \otimes a + a \otimes 1$ for $a \in \mathcal{A}$, and then extending Δ to be a homomorphism with respect to either of the two products on $k\langle\mathcal{A}\rangle$. We refer to [47] for explicit presentations of the deshuffle coproduct.

We remark that the vector space $k\langle\mathcal{A}\rangle$ can now be turned into Hopf algebras in two different ways. The cocommutative *concatenation Hopf algebra* is obtained by taking the concatenation as product and the deshuffle as coproduct. The commutative *shuffle Hopf algebra* $\mathcal{H}_{\mathrm{Sh}}(\mathcal{A})$ is obtained by taking the shuffle as product and the deconcatenation as coproduct. Both these Hopf algebras share the same antipode:

$$S(a_1 a_2 \ldots a_k) = (-1)^k a_k a_{k-1} \ldots a_1, \tag{3.2}$$

and in both cases the unit and counit is given by $\eta(1) = \mathbb{I}$ and $\epsilon(\mathbb{I}) = 1$, $\epsilon(\omega) = 0$ for all $\omega \in \mathcal{A}^* \backslash \mathbb{I}$. We write $\mathcal{H}_{\mathrm{Sh}}$ when $\mathcal{A}$ is understood.

The vector space $k\langle\mathcal{A}\rangle$ can be identified with the vector space underlying the tensor algebra $T(V)$ on the vector space V generated by the alphabet $\mathcal{A}$. The two algebra structures (concatenation and shuffling) correspond to the usual algebra structures given to the tensor algebra $T(V)$ and the tensor coalgebra $T^c(V)$, respectively.

3.3 Characters and endomorphisms

This section is based on [20]. See also [35], [7] and [46].

Let $(H, \mu, \eta, \Delta, \epsilon)$ be a graded bialgebra, and $(A, \cdot, \eta_A)$ an algebra. The set $\mathrm{Hom}_k(H, A)$ of linear maps from H to A sending $\eta_H(1) =: 1_H$ to $\eta_A(1) =: 1_A$ has an algebra structure given by the *convolution product*:

$$\alpha * \beta = \mu_A \circ (\alpha \otimes \beta) \circ \Delta.$$

The convolutional unit is the composition of the counit of H and the unit of A: $\delta := \eta_A \circ \epsilon$. The convolution can be written using the Sweedler notation:

$$\alpha * \beta = \sum_{(x)} \alpha(x_{(1)}) \cdot \beta(x_{(2)}),$$

from which we find $\alpha * \delta = \delta * \alpha = \alpha$. The *unital algebra morphisms* from H to A consists of all $\alpha \in \mathrm{Hom}_k(H, A)$ such that $\alpha(1_H) = 1_A$ and $\alpha(\mu(h, h')) = \alpha(h) \cdot \alpha(h')$ for all $h, h' \in H$.

Proposition 3.8 ([35])**.** *Let H be a graded Hopf algebra and A a commutative algebra. The set $\mathrm{Hom}_{Alg}(H, A)$ of unital algebra morphisms from H to A equipped with the convolution product, forms a group, $G(H, A)$, called the* group of A-valued characters of H*. The inverse of an element α is given by*

$$\alpha^{*-1} = \alpha \circ S,$$

where S is the antipode of H.

In the special case $A = k$, we get the group of characters of H, written as $G(H) := G(H, k)$. The grading on H splits the group of A-valued characters into graded components:

$$G(H, A) \cong \prod_{n \geq 0} \mathrm{Hom}_{Alg}(H_n, A).$$

This is not a graded vector space, but rather the completion of one (see e.g. [20]), but we will still refer to it as a graded vector space. The restriction of a character $\alpha : H \to A$ to the degree n component H_n of H will be denoted by α_n.

3.3.1 Infinitesimal characters, the exponential and the logarithm

The *infinitesimal A-valued characters*, written $\mathfrak{g}(H, A)$ are the linear maps α from H to A such that:

$$\alpha(\mu(h, h')) = \alpha(h) \cdot \delta(h') + \delta(h) \cdot \alpha(h'),$$

where $\delta = \eta_A \circ \epsilon$. This is a Lie algebra under the bracket induced by the convolution product: $[\alpha, \beta] = \alpha * \beta - \beta * \alpha$. In the special case where $A = k$ we write $\mathfrak{g}(H)$ for $\mathfrak{g}(H, k)$.

The characters and infinitesimal characters are related via the *exponential* and the *logarithmic* map. For $\alpha \in \mathrm{Hom}_k(H, A)$, the exponential and logarithm with respect to convolution are given by the formal series:

$$\begin{aligned} \exp^*(\alpha) &= \sum_{n \geq 0} \frac{1}{n!} \alpha^{*n} \\ \log^*(\delta + \alpha) &= \sum_{n \geq 1} \frac{(-1)^{n-1}}{n} \alpha^{*n}. \end{aligned}$$

If H is graded and connected, and if $\alpha(1) = 0$, where $1 \in k = H_0$, then $\alpha^{*k} = \alpha * \cdots * \alpha = 0$ on H_n for $n < k$, and therefore both these sums are finite when restricted to H_n. The maps $\exp^*$ and $\log^*$ give a bijection between $G(H, A)$ and $\mathfrak{g}(H, A)$.

Example 3.9. Let $\mathcal{H}_{\mathrm{Sh}}$ denote the shuffle algebra over $\mathcal{A}$. Consider the dual space $k\langle\langle\mathcal{A}\rangle\rangle$ equipped with the convolution product

$$(\alpha * \beta, \omega) = \sum_{(\omega)_{\Delta_d}} (\alpha, \omega_{(1)})(\beta, \omega_{(2)}) = \sum_{\omega = \omega_1 \omega_2} (\alpha, \omega_1)(\beta, \omega_2).$$

Note that convolution is just concatenation of series $\alpha * \beta = \alpha\beta$. The characters and infinitesimal characters $\mathfrak{g}(\mathcal{H}_{\mathrm{Sh}}), G(\mathcal{H}_{\mathrm{Sh}}) \subset k\langle\langle\mathcal{A}\rangle\rangle$ are given as

$$\begin{aligned} \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}}) &= \{\, \alpha \in k\langle\langle\mathcal{A}\rangle\rangle \mid \alpha(\mathbb{I}) = 0 \text{ and } \alpha(\omega_1 \sqcup\!\sqcup\, \omega_2) = 0 \text{ for all } \omega_1, \omega_2 \in \mathcal{A}^* \backslash \mathbb{I} \,\} \\ G(\mathcal{H}_{\mathrm{Sh}}) &= \{\, \alpha \in k\langle\langle\mathcal{A}\rangle\rangle \mid \alpha(\mathbb{I}) = 1 \text{ and } \alpha(\omega_1 \sqcup\!\sqcup\, \omega_2) = \alpha(\omega_1)\alpha(\omega_2) \text{ for all } \omega_1, \omega_2 \in \mathcal{A}^* \backslash \mathbb{I} \,\}. \end{aligned}$$

The convolutional unit δ is given as $(\delta, \mathbb{I}) = 1$ and $(\delta, \omega) = 0$ for all $\omega \in \mathcal{A} \backslash \mathbb{I}$. The logarithm of $\alpha \in G(\mathcal{H}_{\mathrm{Sh}})$ can be computed as $\log(\alpha) = \sum_{n>0} \frac{(-1)^{n-1}}{n} (\alpha - \delta)^{*n}$. For any $\omega \in \mathcal{A}^*$ we find that $(\log(\alpha), \omega)$ is given by a finite sum expressed in terms of the Eulerian idempotent.

3.3.2 Eulerian idempotent

Let H be a commutative, connected and graded Hopf algebra. Consider $\mathrm{End}_k(H) = \mathrm{Hom}_k(H, H)$ equipped with the convolution product $*$. Let $\mathrm{Id} \in \mathrm{End}_k(H)$ be the identity endomorphism and $\delta = \eta \circ \epsilon \in \mathrm{End}_k(H)$ the unit of convolution.

Definition 3.10 ([32])**.** The Eulerian idempotent $e \in \mathrm{End}(H)$ is given by the formal power series

$$e := \log^*(\mathrm{Id}) = J - \frac{J^{*2}}{2} + \frac{J^{*3}}{3} + \cdots (-1)^{i+1} \frac{J^{*i}}{i} + \cdots,$$

where $J = \mathrm{Id} - \delta$.

Proposition 3.11 ([32])**.** *For any commutative graded Hopf algebra H, the element $e \in \mathrm{End}_k(H)$ defined above is an idempotent: $e \circ e = e$.*

The practical importance of the Eulerian idempotent in numerical analysis arises in backward error analysis, where the following lemma provides a computational formula for the logarithm:

Proposition 3.12. *For $\alpha \in G(H)$ and $h \in H$, we have*

$$\log^*(\alpha)(h) = \alpha(e(h)).$$

In other words, the logarithm can be written as right composition with the eulerian idempotent:

$$\log^* = \_ \circ e : G(H) \to \mathfrak{g}(H).$$

The result follows from the following computation, which uses that α is a homomorphism:

$$((\alpha - \delta)^{*l}, \omega) = \mu_k^l \circ (\alpha \otimes \cdots \otimes \alpha) \circ \tilde{\Delta}^l \omega = \alpha \circ \mu_H^l \circ \tilde{\Delta}^l \omega = \alpha \circ J^{*l} \omega,$$

where $(-)^l$ denotes l-fold application.

3.3.3 The graded Dynkin operator

There is another bijection between the infinitesimal characters and the characters in any commutative graded Hopf algebra H, described in [20]. The bijection is given in terms of the *Dynkin operator* $D : H \to H$.

Classically, the Dynkin operator is a map $D : k\langle\mathcal{A}\rangle \to Lie(\mathcal{A})$, where $Lie(\mathcal{A}) = \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}}) \cap k\langle\mathcal{A}\rangle$ are the Lie polynomials. The classical Dynkin operator is given by left-to-right bracketing:

$$D(a_1 ... a_n) = [\ldots[[a_1, a_2], a_3], \ldots, a_n], \quad \text{where } [a_i, a_j] = a_i a_j - a_j a_i.$$

Letting $Y(\omega) = \#(\omega)\omega$ denote grading operator, where $\#(\omega)$ is word length, it is known that the *Dynkin idempotent*, given as $Y^{-1}D$, is an idempotent projection on the subspace of Lie polynomials. As in [20], the Dynkin operator can be written as the convolution of the antipode S and the grading operator: $D = S * Y$. This description can be generalized to any graded, connected and commutative Hopf algebra H:

Definition 3.13. Let H be a graded, commutative and connected Hopf algebra with grading operator $Y : H \to H$. The *Dynkin operator* is the map $D : H \to H$ given as

$$D := S * Y.$$

Lemma 3.14 ([20])**.** *The Dynkin operator is a H-valued infinitesimal character of H.*

Theorem 3.15 ([20])**.** *Right composition with the Dynkin operator induces a bijection between $G(H)$ and $\mathfrak{g}(H)$:*

$$\_ \circ D : G(H) \to \mathfrak{g}(H).$$

The inverse is given by $\Gamma : \mathfrak{g}(H) \to G(H)$ as

$$\Gamma(\alpha) = \sum_n \sum_{\substack{k_1 + \cdots + k_l = n \\ k_1, \ldots, k_l > 0}} \frac{\alpha_{k_1} * \cdots * \alpha_{k_l}}{k_1(k_1 + k_2) \cdots (k_1 + \cdots + k_l)}, \tag{3.3}$$

where $\alpha_k = \alpha|_{H_k}$.

Later we will apply the Dynkin operator and its inverse in the setting of a shuffle algebra $\mathcal{H}_{\mathrm{Sh}}(\mathrm{OT})$, where OT is an alphabet of all ordered rooted trees, and the grading $|\tau|$ of $\tau \in \mathrm{OT}$ counts the nodes in the tree.

4 Algebras of formal diffeomorphisms on manifolds

The main goal of this section is to arrive at Lie–Butcher series and the underlying Hopf algebra $\mathcal{H}_N$. This Hopf algebra contains the Connes–Kreimer Hopf algebra as a subalgebra and is also closely related to $\mathcal{H}_{\mathrm{Sh}}$. To emphasize the natural connection between Lie–Butcher series, $\mathcal{H}_N$ and more classical Lie series, we start with a discussions of Lie series (autonomous and non-autonomous).

4.1 Autonomous Lie series

In this section we review the well-known theory of Lie series on manifolds and the corresponding Hopf algebraic structures of the free Lie algebra. The algebraic theory is detailed in [47, 11]. For the analytical theory we refer to [2].

Let F be a vector field on a manifold $\mathcal{M}$ and $\Phi_{t,F}\colon \mathcal{M} \to \mathcal{M}$ its flow. Let $\psi\colon \mathcal{M} \to E$ be a section of a vector bundle over $\mathcal{M}$, and let $\Phi_{t,F}^*\psi$ denote the pullback. For the applications later in this paper we will only consider trivial bundles, in which case we write $\psi\colon \mathcal{M} \to \mathcal{V}$ for some vector space $\mathcal{V}$ and define pullback as composition $\Phi_{t,F}^*\psi = \psi\circ\Phi_{t,F}$. The *Lie derivative* of ψ is defined as

$$F[\psi] = \left.\frac{\partial}{\partial t}\right|_{t=0} \Phi_{t,F}^*\psi. \tag{4.1}$$

Composition of Lie derivatives defines an associative, non-commutative product of vector fields $F, G \mapsto FG$, where vector fields are first order differential operators. The product FG is the second order differential operator $(FG)[\psi] = F[G[\psi]]$ etc. We let $\mathbb{I}$ denote the 0th order identity operator $\mathbb{I}[\psi] = \psi$. The linear span of all differential operators of all orders forms the universal enveloping algebra $U(\mathcal{XM})$.

The basic pullback formula is ([2]):

$$\frac{\partial}{\partial t}\Phi_{t,F}^*\psi = \Phi_{t,F}^*(F[\psi]). \tag{4.2}$$

Iterating this we find $\partial^n/\partial t^n|_{t=0}\Phi_{t,F}^*\psi = F[F[\cdots[\psi]]] := F^n[\psi]$, and hence follows the (Taylor)–Lie form of a *pullback series*:

$$\Phi_{t,F}^*\psi = \sum_{j=0}^{\infty} \frac{t^j}{j!} F^j[\psi] := \mathrm{Exp}(tF)[\psi]. \tag{4.3}$$

Fundamental questions are: Which series in $U(\mathcal{XM})$ represent vector fields and which represent pullback series? How do we algebraically characterize compositions and the inverse of pullback series? How do we understand the Exp map taking vector fields to their pullback series? And what about the inverse Log operation? These questions are elegantly answered in terms of the shuffle Hopf algebra. We will detail these issues. Later we will see the same structures reappear in the discussion of B-series.

An algebraic abstraction of Lie series starts with fixing a (finite or infinite) alphabet $\mathcal{A}$ and a map $\nu\colon \mathcal{A} \to \mathcal{XM}$ assigning each letter to a vector field. As in Example 3.2 we let $\mathbb{R}\langle\mathcal{A}\rangle$ denote all finite $\mathbb{R}$-linear combinations of words built from $\mathcal{A}$ and $\mathcal{H}_{\mathrm{Sh}}$ the shuffle algebra. The map ν can be uniquely extended to a linear $\mathcal{F}_\nu\colon \mathbb{R}\langle\mathcal{A}\rangle \to U(\mathcal{XM})$ as a concatenation homomorphism:

$$\begin{aligned}
\mathcal{F}_\nu(\mathbb{I}) &= \mathbb{I},\\
\mathcal{F}_\nu(a) &= \nu(a) \quad \text{for all letters } a \in \mathcal{A},\\
\mathcal{F}_\nu(\omega_1\omega_2) &= \mathcal{F}_\nu(\omega_1)\mathcal{F}_\nu(\omega_2) \quad \text{for all words } \omega_1, \omega_2 \in \mathcal{A}^*.
\end{aligned}$$

We extend $\mathcal{F}_\nu$ to a map $\mathcal{B}_t$ taking an infinite series $\alpha \in \mathbb{R}\langle\langle\mathcal{A}\rangle\rangle$ to an infinite formal series $\mathcal{B}_t(\alpha) \in U(\mathcal{XM})^*$, defined for $t \in \mathbb{R}$ as follows: Consider the alphabet $\mathcal{A}$ with a grading $|a| \in \mathbb{N}^+$ for all $a \in \mathcal{A}$. This extends to $\mathcal{H}_{\mathrm{Sh}}$ as $|\omega| = |a_1| + \ldots + |a_k|$ for all $\omega = a_1 \ldots a_k \in \mathcal{A}^*$, $|\mathbb{I}| = 0$, thus $\mathcal{H}_{\mathrm{Sh}}$ becomes a graded connected Hopf algebra. Given the grading we define

$$\mathcal{B}_t(\alpha) = \sum_{\omega\in\mathcal{A}^*} t^{|\omega|}\alpha(\omega)\mathcal{F}_\nu(\omega). \tag{4.4}$$

Consider ${\mathcal{H}_{\mathrm{Sh}}}^* = \mathbb{R}\langle\langle\mathcal{A}\rangle\rangle$ with the convolution $\alpha * \beta = \alpha\beta$ as in Example 3.9. By construction $\mathcal{B}_t$ is a convolution homomorphism,

$$\mathcal{B}_t(\alpha * \beta) = \mathcal{B}_t(\alpha)\mathcal{B}_t(\beta).$$

For a real valued infinitesimal character $\alpha \in \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})$, and a fixed $t = h$, $\mathcal{B}_h(\alpha)$ is a formal vector field on $\mathcal{M}$. For a real valued character $\beta \in G(\mathcal{H}_{\mathrm{Sh}})$, $\mathcal{B}_h(\beta)$ represents a formal diffeomorphism Φ_h on $\mathcal{M}$ via the pullback series

$$\mathcal{B}_h(\beta)[\psi] = \psi\circ\Phi_h \quad \text{for } \psi\colon \mathcal{M} \to \mathbb{R}.$$

Note, however, that pullbacks compose contravariantly with respect to composition of diffeomorphisms:

$$\mathcal{B}_h(\beta_1 * \beta_2)[\psi] = \mathcal{B}_h(\beta_1)\mathcal{B}_h(\beta_2)[\psi] = \psi\circ\Phi_2\circ\Phi_1.$$

To summarize: Composition of diffeomorphisms is modelled by convolution in $G(\mathcal{H}_{\mathrm{Sh}})$ (in opposite order), the inverse of a diffeomorphism is computed by right composing with the antipode, the convolutional exponential maps to the exponential of Lie series and the logarithm is computed by composing with the Eulerian idempotent.

$$\begin{aligned}
\mathcal{B}_h(\beta\circ S)\mathcal{B}_h(\beta) &= \mathbb{I} \quad \text{for } \beta \in G(\mathcal{H}_{\mathrm{Sh}})\\
\mathcal{B}_h(\exp^*(\alpha)) &= \mathrm{Exp}(\mathcal{B}_h(\alpha)) \quad \text{for } \alpha \in \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})\\
\mathcal{B}_h(\beta) &= \mathrm{Exp}(\mathcal{B}_h(\beta\circ e)) \quad \text{for } \beta \in G(\mathcal{H}_{\mathrm{Sh}}).
\end{aligned}$$

In the next section we discuss flows of non-autonomous equations, and we will see that right composition with the Dynkin idempotent represents algebraically the operation of finding a non-autonomous vector field corresponding to a diffeomorphism on a manifold.

Remark 4.1. In [41] the Lie algebra of infinitesimal characters $\mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})$ is studied as a graded free Lie algebra. An explicit formula for the dimension of the homogeneous components $\mathfrak{g}_k = \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})|_k$ is derived for general gradings. This is very useful for the study of the complexity of Lie group integrators.

4.2 Time-dependent Lie series

The classical Faà di Bruno Hopf algebra models the composition of formal diffeomorphisms on $\mathbb{R}$ ([23], [22], [24]). We will see that this has a natural generalization to compositions of time-dependent flows on manifolds. We introduce a *Dynkin–Faà di Bruno bialgebra* describing the composition of flows of time-dependent vector fields on a coarse level that considers only the grading of the terms in the t-expansion of the time-dependent vector fields.

4.2.1 Non-commutative Bell polynomials and the Dynkin–Faà di Bruno bialgebra

Let $\mathcal{I} = \{d_j\}_{j=1}^{\infty}$ be an infinite alphabet in 1–1 correspondence with $\mathbb{N}^+$, and consider the free associative algebra $\mathcal{D} = \mathbb{R}\langle\mathcal{I}\rangle$ with the grading given by $|d_j| = j$ and $|d_{j_1}\cdots d_{j_k}| = j_1 + \cdots + j_k$. Let $\partial\colon \mathcal{D} \to \mathcal{D}$ be the derivation given by $\partial(d_i) = d_{i+1}$, linearity and the Leibniz rule $\partial(\omega_1\omega_2) = \partial(\omega_1)\omega_2 + \omega_1\partial(\omega_2)$ for all $\omega_1, \omega_2 \in \mathcal{I}^*$. We let $\#(\omega)$ denote the length of the word ω.

Definition 4.2. The non-commutative Bell polynomials $B_n := B_n(d_1, \ldots, d_n) \in \mathbb{R}\langle\mathcal{I}\rangle$ are defined by the recursion

$$\begin{aligned}
B_0 &= \mathbb{I}\\
B_n &= (d_1 + \partial)B_{n-1} = (d_1 + \partial)^n\mathbb{I} \quad \text{for } n > 0.
\end{aligned}$$

The first of these are given as

$$\begin{aligned}
B_0 &= \mathbb{I} \\
B_1 &= d_1 \\
B_2 &= d_1^2 + d_2 \\
B_3 &= d_1^3 + 2d_1d_2 + d_2d_1 + d_3 \\
B_4 &= d_1^4 + 3d_1^2d_2 + 2d_1d_2d_1 + d_2d_1^2 + 3d_1d_3 + d_3d_1 + 3d_2d_2 + d_4.
\end{aligned}$$

The polynomials B_n are introduced in [38, 39] to explain the Butcher order theory of Runge–Kutta methods in a manifold context, and generalize to certain classes of numerical integrators on manifolds.

Remark 4.3. Additional insight to the Bell polynomials are obtained by considering the free associative algebra generated by two symbols d_1 and ∂, defining

$$d_i := [\partial, d_{i-1}] = \partial d_{i-1} - d_{i-1}\partial \quad \text{for } i > 1.$$

We find by induction that $(d_1 + \partial)^n$ satisfies the binomial relation

$$(d_1 + \partial)^n = \sum_{k=0}^{n} \binom{n}{k} B_k(d_1, \ldots, d_k)\partial^{n-k}, \tag{4.5}$$

which yields the formula

$$\exp(d_1 + \partial) = \sum_{m=0}^{\infty} \frac{B_m(d_1, \ldots, d_m)}{m!} \exp(\partial), \tag{4.6}$$

and also the recursion

$$B_{n+1}(d_1, \ldots, d_{n+1}) = \sum_{k=0}^{n} \binom{n}{k} B_k(d_1, \ldots, d_k) d_{n-k+1} \quad \text{for } n > 0. \tag{4.7}$$

The non-commutative *partial Bell polynomials* $B_{n,k} := B_{n,k}(d_1, \ldots, d_{n-k+1})$ are defined as the part of B_n consisting of the words ω of length $\#(\omega) = k > 0$, e.g. $B_{4,3} = 3d_1^2d_2 + 2d_1d_2d_1 + d_2d_1^2$. Thus

$$B_n = \sum_{k=1}^{n} B_{n,k}.$$

A bit of combinatorics yields an explicit formula:

$$B_{n,k} = \sum_{\substack{\omega \in \mathcal{I}^* \\ |\omega| = n, \#(\omega) = k}} \kappa(\omega) \binom{n}{\omega} \omega, \tag{4.8}$$

where for $\omega = d_{j_1}d_{j_2}\cdots d_{j_k}$

$$\binom{n}{\omega} := \binom{n}{|d_{j_1}|, |d_{j_2}|, \ldots, |d_{j_k}|} := \frac{n!}{j_1!j_2!\cdots j_k!}$$

are the multinomial coefficients and the coefficients $\kappa(\omega)$ are defined as

$$\kappa(\omega) := \kappa(|d_{j_1}|, |d_{j_2}|, \ldots, |d_{j_k}|) := \frac{j_1 j_2 \cdots j_k}{j_1(j_1 + j_2)\cdots(j_1 + j_2 + \cdots + j_k)}. \tag{4.9}$$

The coefficients κ form a partition of unity on the symmetric group S_k,

$$\sum_{\sigma \in S_k} \kappa(\sigma(\omega)) = 1,$$

where $\sigma(\omega)$ denotes a permutation of the letters in ω. E.g. $\kappa(1,2)+\kappa(2,1)=\frac{2}{3}+\frac{1}{3}=1$.

It is often useful to employ polynomials Q_n and $Q_{n,k}$ related to B_n and $B_{n,k}$ by the following rescaling:

$$\begin{aligned} Q_{n,k}(d_1,\dots,d_{n-k+1}) &= \frac{1}{n!}B_{n,k}(1!d_1,\dots,j!d_j,\dots) = \sum_{|\omega|=n,\#(\omega)=k}\kappa(\omega)\omega \\ Q_n(d_1,\dots,d_n) &= \sum_{k=1}^{n} Q_{n,k}(d_1,\dots,d_{n-k+1}) \\ Q_0 &:= \mathbb{I}. \end{aligned} \tag{4.10}$$

Note that B_n and $B_{n,k}$ become the classical Bell- and partial Bell polynomials when the product in $\mathbb{R}\langle\mathcal{I}\rangle$ is commutative, i.e. in the free commutative algebra on $\mathcal{I}$. A non-commutative Faà di Bruno Hopf algebra is studied in [6]. However, their definition differs from the present by defining the polynomials $Q_{n,k}$ without the factor κ that associates different factors to different permutations of a word (adding up to 1 over all permuatations).

These Bell polynomials are closely related to the graded Dynkin operator on a connected graded Hopf algebra H. For $\alpha\in H^*$, define a graded algebra homomorphism $d_i\mapsto d_i(\alpha)\colon \mathcal{D}\to H^*$ as

$$d_i(\alpha)=\alpha_i=\alpha|_{H_i}\,,\qquad d_id_j(\alpha)=\alpha_i*\alpha_j. \tag{4.11}$$

Proposition 4.4. *The operator defined as*

$$Q(\alpha)=\sum_{n=0}^{\infty}Q_n(\alpha), \tag{4.12}$$

is a bijection from infinitesimal characters to characters $Q\colon \mathfrak{g}(H)\to G(H)$ with inverse given by right composition with the Dynkin idempotent $Y^{-1}\circ D$,

$$Q^{-1}(\beta)=\beta\circ Y^{-1}\circ D, \tag{4.13}$$

where Y is the grading operator on H and $D=S*Y$ is the graded Dynkin operator.

Proof. For $\alpha\in\mathfrak{g}(H)$ we have

$$\Gamma(\alpha\circ Y)=\sum_{n=0}^{\infty}\sum_{j_1+\cdots+j_k=n}\frac{j_1j_2\cdots j_k}{j_1(j_1+j_2)\cdots(j_1+\cdots+j_k)}\alpha_{j_1}*\cdots*\alpha_{j_k}=Q(\alpha), \tag{4.14}$$

thus the result follows from Theorem 3.15. □

The non-commutative Dynkin–Faà di Bruno bialgebra $\mathcal{D}$ is obtained by taking the algebra structure of $\mathcal{D}$ and defining the coproduct $\Delta_{\mathcal{D}}$ as

$$\begin{aligned} \Delta_{\mathcal{D}}(\mathbb{I}) &= \mathbb{I}\otimes\mathbb{I} \\ \Delta_{\mathcal{D}}(d_n) &= \sum_{k=1}^{n}B_{n,k}\otimes d_k. \end{aligned} \tag{4.15}$$

This extends to all of $\mathcal{D}$ by the product rule $\Delta_{\mathcal{D}}(d_id_j)=\Delta_{\mathcal{D}}(d_i)\Delta_{\mathcal{D}}(d_j)$. Thus, e.g.

$$\begin{aligned} \Delta_{\mathcal{D}}(d_1) &= d_1\otimes d_1 \\ \Delta_{\mathcal{D}}(d_2) &= d_1^2\otimes d_2+d_2\otimes d_1 \\ \Delta_{\mathcal{D}}(d_1d_2) &= d_1^3\otimes d_1d_2+d_1d_2\otimes d_1^2. \end{aligned}$$

Note that the coproduct is *not* graded by $|\cdot|$, thus Proposition 3.7 does not hold for $\mathcal{D}$. By a lengthy (but not enlightening) induction argument we can prove:

Lemma 4.5. *The coproduct of the partial Bell polynomials are given as*

$$\Delta_{\mathcal{D}}(B_{n,k}) = \sum_{\ell=1}^{n} B_{n,\ell} \otimes B_{\ell,k}. \tag{4.16}$$

Note that $B_{n,1} = d_n$, thus (4.15) is a special case of (4.16). Summing the partial $B_{n,k}$ over k, we find the coproduct of the full Bell polynomials:

$$\Delta_{\mathcal{D}}(B_n) = \sum_{k=1}^{n} B_{n,k} \otimes B_k.$$

Using Lemma 4.5 and the fact that $B_{n,k} = 0$ for $k > n$, one can easily show that $\mathcal{D}$ is a bialgebra.

Proposition 4.6. $\mathcal{D} = \mathbb{R}\langle\mathcal{I}\rangle$ *with the non-commutative concatenation product and the coproduct* $\Delta_{\mathcal{D}}$ *form a bialgebra* $\mathcal{D}$ *which is neither commutative nor cocommutative.*

4.2.2 Pullback along time-dependent flows

Let $F_t = \sum_{j=0}^{\infty} F_{j+1} \frac{t^j}{j!}$ be a time-dependent vector field on $\mathcal{M}$ where $F_j = F_t^{(j-1)}\Big|_{t=0}$. Let Φ_{t,F_t} be the solution operator of the corresponding non-autonomous equation, such that

$$y(t) = \Phi_{t,F_t} y_0 \quad \text{solves} \quad y'(t) = F_t(y(t)), \quad y(0) = y_0.$$

Note that Φ_{t,F_t} is *not* a 1-parameter subgroup of diffeomorphisms in t.

Lemma 4.7 ([38])**.** *The n-th time derivative of the pullback of a (time-independent) function* ψ *along the time-dependent flow* Φ_{t,F_t} *is given as*

$$\frac{\partial^n}{\partial t^n} \Phi_{t,F_t}^* \psi = B_n(F_t)[\psi], \tag{4.17}$$

where $B_n(F_t)$ *is the image of* B_n *under the homomorphism from* $\mathcal{D}$ *to* $U(\mathcal{XM})$ *given by* $d_i \mapsto F_t^{(i-1)}$*. In particular*

$$\frac{\partial^n}{\partial t^n}\Big|_{t=0} \Phi_{t,F_t}^* \psi = B_n(F_1, \ldots, F_n)[\psi]. \tag{4.18}$$

Proof. The non-autonomous vector field F_t on $\mathcal{M}$ corresponds to the autonomous field $F_t + \partial/\partial t$ on $\mathcal{M} \times \mathbb{R}$, thus (4.2) yields

$$\frac{\partial}{\partial t} \Phi_{t,F_t}^* \psi = \Phi_{t,F_t}^* \left((F_t + \partial/\partial t)[\psi] \right) \Rightarrow \frac{\partial^n}{\partial t^n} \Phi_{t,F_t}^* \psi = \Phi_{t,F_t}^* \left((F_t + \partial/\partial t)^n [\psi] \right).$$

Consider the homomorphism induced from $d_1 \mapsto F_t$ and $\partial \mapsto \partial/\partial t$, thus $d_i \mapsto F_t^{(i-1)}$. Equation (4.17) follows directly from Definition 4.2. At $t = 0$ we have $d_i \mapsto F_i$, thus (4.18). □

Remark 4.8. Note that (4.6) yields a space-time split formula for pullback which is valid also for pullback of a time-dependent function ψ_t. The pullback for $t \in [0, h]$ developed at $t = 0$ becomes

$$\Phi_{h,F_t}^* \psi_t = \exp\left(h(F_t + \frac{\partial}{\partial t}) \right)\Bigg|_{t=0} [\psi_t] = \sum_{n=0}^{\infty} \frac{h^n}{n!} B_n(F_t) \exp(h \frac{\partial}{\partial t})\Bigg|_{t=0} [\psi_t] = \sum_{n=0}^{\infty} \frac{h^n}{n!} B_n(F_1, \ldots, F_n)[\psi_h].$$

The Dynkin idempotent relates pullback series with their corresponding time-dependent vector fields. Let $\mathcal{A}$ be an arbitrary alphabet with a grading $|\cdot| \colon \mathcal{A} \to \mathbb{N}^+$, let $\mathcal{H}_{\text{Sh}} = \mathcal{H}_{\text{Sh}}(\mathcal{A})$ be the corresponding graded shuffle algebra and let $\mathcal{B}_t(\alpha)$ be as in (4.4).

Proposition 4.9. *Let $\alpha \in \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})$ and $\beta = Q(\alpha) \in G(\mathcal{H}_{\mathrm{Sh}})$ be related by the graded Dynkin idempotent as in Proposition 4.4. Define the time-dependent vector field*

$$F_t = \frac{\partial}{\partial t}\mathcal{B}_t(\alpha).$$

Then pullback of a time-independent ψ along the time-dependent flow Φ_{t,F_t} is given as

$$\Phi^*_{t,F_t}\psi = \mathcal{B}_t(\beta)[\psi]. \tag{4.19}$$

Proof. We have $F_t = \sum_{j=0}^{\infty} F_{j+1}\frac{t^j}{j!}$ where $F_j = \mathcal{F}_\nu(j!\alpha_j)$. Developing the Taylor series of $\Phi^*_{t,F_t}\psi$ at $t=0$ we get from (4.18)

$$\Phi^*_{t,F_t}\psi = \sum_{n=0}^{\infty}\frac{t^n}{n!}B_n(F_1,\ldots,F_n)[\psi].$$

Thus

$$\frac{1}{n!}B_n(F_1,\ldots,F_n) = \mathcal{F}(\frac{1}{n!}B_n(1!\alpha_1,\ldots,n!\alpha_n)) = \mathcal{F}(Q_n(\alpha_1,\ldots,\alpha_n)).$$

Using (4.14) we obtain the result. □

4.3 Lie–Butcher theory

Pullback formulas such as (4.19) relate the time derivatives of F_t with the spatial derivatives of a function ψ. We have captured the algebraic structure of the temporal derivations through the Dynkin idempotent $Y^{-1}\circ D\colon G(\mathcal{H}_{\mathrm{Sh}}) \to \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})$ and its inverse $\Gamma\circ Y\colon \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}}) \to G(\mathcal{H}_{\mathrm{Sh}})$. However, the spatial Lie derivation $\mathcal{B}_t(\beta)[\psi]$ cannot be algebraically characterized within this structure. In order to do this, we need to refine the Hopf algebra $\mathcal{H}_{\mathrm{Sh}}$. On the manifold M, we obtain a refined version of $U(\mathcal{XM})$ by expanding differential operators in terms of a non-commuting frame on $\mathcal{XM}$. If the manifold is $\mathbb{R}^n$ and the frame is the standard commutative coordinate frame, the construction yields the classical Butcher formulation and the Connes–Kreimer Hopf algebra [4]. More generally we obtain a Hopf algebra $\mathcal{H}_N$, built on forests of planar trees, which contains the Connes–Kreimer algebra as a subalgebra. In $\mathcal{H}_N$ we can represent Lie derivation in terms of tree graftings.

4.3.1 Differential operators in $U(\mathcal{XM})$ expanded in a non-commuting frame

Let $\mathcal{XM}$ denote the Lie algebra of all vector fields on $\mathcal{M}$ and let $\mathfrak{g} \subset \mathcal{XM}$ be a transitive Lie subalgebra, in the sense that $\mathfrak{g}$ everywhere spans $T\mathcal{M}$. This means that $\mathfrak{g}$ defines a frame on the tangent bundle. We do not assume that the frame forms a basis. In general $\dim(\mathfrak{g}) \geq \dim(\mathcal{M})$, and in case of strict inequality we have a non-trivial isotropy subgroup at any point.

Let $U(\mathfrak{g})$ denote the universal enveloping algebra of $\mathfrak{g}$. We let $\mathfrak{g}^{\mathcal{M}}$ and $U(\mathfrak{g})^{\mathcal{M}}$ denote maps from $\mathcal{M}$ to $\mathfrak{g}$ and from $\mathcal{M}$ to $U(\mathfrak{g})$. Since $\mathfrak{g}$ is assumed to be transitive, we can represent any vector field $F \in \mathcal{XM}$ with a function $f \in \mathfrak{g}^{\mathcal{M}}$ as in Section 2.2.1. Similarly, any higher order differential operator in $U(\mathcal{XM})$ can be represented as a function in $U(\mathfrak{g})^{\mathcal{M}}$. We have the natural inclusion $\mathfrak{g} \subset \mathfrak{g}^{\mathcal{M}}$ and $U(\mathfrak{g}) \subset U(\mathfrak{g})^{\mathcal{M}}$ as constant maps, called frozen vector fields and higher order differential operators. We identify $U(\mathfrak{g})^{\mathcal{M}}$ with sections of the trivial vector bundle $\mathcal{M}\otimes U(\mathfrak{g}) \to \mathcal{M}$, and for a diffeomorphism $\Phi\colon \mathcal{M} \to \mathcal{M}$ we define pullback of $f \in U(\mathfrak{g})^{\mathcal{M}}$ as $\Phi^* f = f\circ\Phi \in U(\mathfrak{g})^{\mathcal{M}}$. Pullback in this bundle defines a parallel transport which gives rise to a flat connection with torsion. For $f, g \in U(\mathfrak{g})^{\mathcal{M}}$ we define the connection $f[g] \in U(\mathfrak{g})^{\mathcal{M}}$ pointwise from the Lie derivative as

$$f[g](p) = \left(f(p)[g]\right)(p), \qquad p \in \mathcal{M}.$$

Similarly, the concatenation in $U(\mathfrak{g})$ is extended pointwise to a concatenation product $fg \in U(\mathfrak{g})^{\mathcal{M}}$ as

$$(fg)(p) = f(p)g(p), \qquad p \in \mathcal{M}.$$

This is called the *frozen composition* of f and g. We can also compose f and g as non-frozen differential operators $f{\bullet}g \in U(\mathfrak{g})^{\mathcal{M}}$:

$$(f{\bullet}g)[h] = f[g[h]], \qquad \text{for all } h \in U(\mathfrak{g})^{\mathcal{M}} .$$

This is identical to the composition in $U(\mathcal{XM})$, which in Section 4.1 was written as $F, G \mapsto FG$ for $F, G \in \mathcal{XM}$.

It might be illustrative to write out the operations explicitly in terms of a basis $\{\partial_k\}_{k=1}^n$ of (non-commuting) vector fields spanning $\mathfrak{g}$. Writing $f, g \in \mathfrak{g}^{\mathcal{M}}$ in terms of the frame as $f = \sum_k f_k \partial_k$ and $g = \sum_\ell g_\ell \partial_\ell$ for $f_k, g_\ell \in \mathbb{R}^{\mathcal{M}}$, we have

$$\begin{aligned} fg &= \sum_{k,\ell} f_k g_\ell \partial_k \partial_\ell \\ f[g] &= \sum_{k,\ell} f_k \partial_k[g_\ell]\partial_\ell \\ f{\bullet}g &= \sum_{k,\ell} f_k \partial_k[g_\ell]\partial_\ell + \sum_{k,\ell} f_k g_\ell \partial_k \partial_\ell. \end{aligned}$$

The connection $f[g]$, the frozen composition fg and nonfrozen composition $f{\bullet}g$ are related as:

Lemma 4.10. *Let $f \in \mathfrak{g}^{\mathcal{M}}$ and $g, h \in U(\mathfrak{g})^{\mathcal{M}}$. Then we have*

$$\begin{aligned} \mathbb{I}[g] &= g \\ f[gh] &= f[g]h + g(f[h]), \quad \textit{(Leibniz)} \\ (f{\bullet}g)[h] := f[g[h]] &= (fg)[h] + (f[g])[h], \end{aligned}$$

where $\mathbb{I} \in U(\mathfrak{g})^{\mathcal{M}}$ is the constant identity map.

The proof is given in [42]. Note the difference between fg and $f{\bullet}g$. In the concatenation the value of g is frozen to $g(p)$ before the differentiation with f is done, whereas in the latter case the spatial variation of g is seen by the differentiation using f. Interestingly, the work of Cayley from 1857 [12] starts with the same result for vector fields expanded in the commuting frame $\partial/\partial x_i$.

From this lemma we may compute the torsion and curvature of the connection. Let $f, g \in \mathfrak{g}^{\mathcal{M}}$. We henceforth let $[f, g]_\bullet := f{\bullet}g - g{\bullet}f$ denote the Jacobi bracket and $[f, g] = fg - gf$ the *frozen bracket.* The frozen bracket is computed pointwise from the bracket in $\mathfrak{g}$ as $[f, g](p) = [f(p), g(p)]_{\mathfrak{g}}$. Writing the connection as $\nabla_f g := f[g]$, we find

$$\begin{aligned} T(f, g) &= \nabla_f g - \nabla_g f - [f, g]_\bullet = gf - fg = -[f, g] \\ R(f, g)h &= \nabla_f \nabla_g h - \nabla_g \nabla_f h - \nabla_{[f,g]_\bullet} h = 0. \end{aligned}$$

Note that if $\mathfrak{g}$ is commutative, then $[f, g] = 0$ and the connection is both flat and torsion free. In this case $f[g]$ is a pre-Lie product generating the Jacobi bracket: $f[g] - g[f] = [f, g]_\bullet$, but in general $f[g] - g[f] = [f, g]_\bullet - [f, g]$.

The product $f{\bullet}g$ is associative, and thus $U(\mathfrak{g})^{\mathcal{M}}$ with the binary operations $f, g \mapsto f[g]$ and $f, g \mapsto f{\bullet}g$ forms a unital dipterous algebra [33], however, it has more structure than this. Following [42] we define:

Definition 4.11. Let $\mathcal{A} = \mathbb{I} \oplus \overline{\mathcal{A}}$ be a unital associative algebra with product $f, g \mapsto fg$, and also equipped with a non-associative composition $f, g \mapsto f[g] \colon \mathcal{A} \times \mathcal{A} \to \mathcal{A}$. Let $D(\mathcal{A})$ denote all $f \in \mathcal{A}$ such that $f[\cdot]$ is a derivation:

$$D(\mathcal{A}) = \{\, f \in \mathcal{A} \mid f[gh] = (f[g])h + g(f[h]) \,\}.$$

We assume that $D(\mathcal{A})$ generates $\overline{\mathcal{A}}$. We call $\mathcal{A}$ a D-algebra if for any derivation $f \in \mathcal{D}(\mathcal{A})$ and any $g, h \in \mathcal{A}$ we have

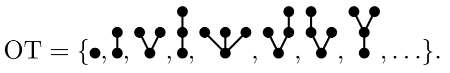
$$\begin{aligned} g[f] &\in D(\mathcal{A}) \\ \mathbb{I}[g] &= g \qquad\qquad (4.20) \\ f[g[h]] &= (fg)[h] + (f[g])[h]. \end{aligned}$$

Definition 4.12. A D-algebra homomorphism is a map $\mathcal{F}\colon \mathcal{A} \to \mathcal{A}'$ between D-algebras such that $\mathcal{F}(D(\mathcal{A})) \subset D(\mathcal{A}')$ and for all $g, h \in \mathcal{A}$ we have

$$\begin{aligned} \mathcal{F}(\mathbb{I}) &= \mathbb{I} \\ \mathcal{F}(gh) &= \mathcal{F}(g)\mathcal{F}(h) \\ \mathcal{F}(g([h]) &= \mathcal{F}(g)[\mathcal{F}(h)]. \end{aligned} \tag{4.21}$$

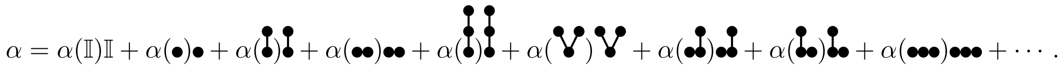
4.3.2 The free D-algebra and elementary differentials

The following definitions are detailed in [42]. Let OT denote the alphabet of all ordered (planar) rooted trees:



More generally, we consider decorated ordered rooted trees, where $\mathcal{C}$ is a (finite or infinite) set of colors. Decorated trees are trees with a color from $\mathcal{C}$ assigned to each node. As above, we let OT^* denote words of trees (forests), let $\mathbb{I}$ be the empty word and let $\omega_1, \omega_2 \mapsto \omega_1\omega_2$ denote concatenation for $\omega_1, \omega_2 \in \mathrm{OT}^*$. Identifying $\mathcal{C} \subset \mathrm{OT}$ with 1-node trees, we can recursively build all words in OT^* from $\mathcal{C}$ by concatenation and adding roots. For $c \in \mathcal{C}$ and $\omega \in \mathrm{OT}^*$, define $B_c^+(\omega) \in \mathrm{OT}$ as the tree with branches ω and root c. Often we will be interested in the case where $\mathcal{C} = \{\bullet\}$, just one color.

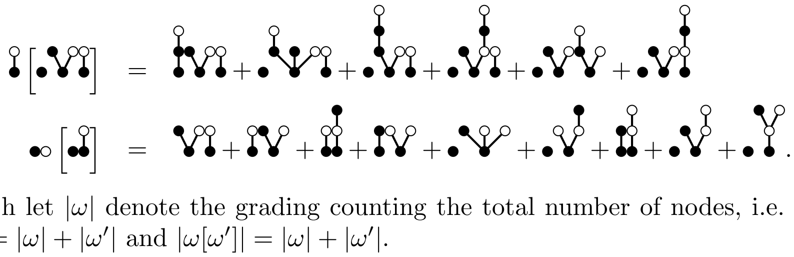
As above, let $\mathbb{R}\langle\mathrm{OT}\rangle$ denote real polynomials (finite $\mathbb{R}$-linear combinations of words) and $\mathbb{R}\langle\langle\mathrm{OT}\rangle\rangle$ the dual space of infinite series, such as



On $\mathbb{R}\langle\mathrm{OT}\rangle$ we define *left grafting* $(\cdot)[\cdot]\colon \mathbb{R}\langle\mathrm{OT}\rangle \times \mathbb{R}\langle\mathrm{OT}\rangle \to \mathbb{R}\langle\mathrm{OT}\rangle$ by extending the following definition for trees by linearity. For all $c \in \mathcal{C}$, all $\tau \in \mathrm{OT}$ and all $\omega, \omega' \in \mathrm{OT}^*$ we define:

$$\begin{aligned} \omega[c] &= B_c^+(\omega) \\ \mathbb{I}[\omega] &= \omega \\ \tau[\omega\omega'] &= \tau[\omega]\omega' + \omega(\tau[\omega']) \\ \tau[\omega[\omega']] &= (\tau\omega)[\omega'] + (\tau[\omega])[\omega']. \end{aligned} \tag{4.22}$$

Compare this with Lemma 4.10. The left grafting $\tau[\omega]$ is obtained by attaching τ in all possible ways from the left to the vertices of ω, and $(\tau\tau')[\omega]$ is obtained by attaching from the left first τ' and then τ on all nodes of ω:



We henceforth let $|\omega|$ denote the grading counting the total number of nodes, i.e. $|c| = 1$ for all $c \in \mathcal{C}$, $|\omega\omega'| = |\omega| + |\omega'|$ and $|\omega[\omega']| = |\omega| + |\omega'|$.

Proposition 4.13. *Let* OT *be planar trees decorated with colors $\mathcal{C}$. Consider $\mathcal{N} = \mathbb{R}\langle\mathrm{OT}\rangle$ with concatenation $\omega, \omega' \mapsto \omega\omega'$, left grafting $\omega, \omega' \mapsto \omega[\omega']$ and unit $\mathbb{I}$ as defined above. $\mathcal{N}$ is a free D-algebra over $\mathcal{C}$, such that for any D-algebra $\mathcal{A}$ and any map $\nu\colon \mathcal{C} \to D(\mathcal{A})$ there exists a unique D-algebra homomorphism map $\mathcal{F}_\nu\colon \mathcal{N} \to \mathcal{A}$ such that $\mathcal{F}_\nu(c) = \nu(c)$ for all $c \in \mathcal{C}$.*

$$\begin{tikzcd} \mathcal{C} \arrow[r, hook] \arrow[d, "\nu"'] & \mathcal{N} \arrow[d, "\exists\,!\ \mathcal{F}_\nu"] \\ D(\mathcal{A}) \arrow[r, hook] & \mathcal{A} \end{tikzcd}$$

Definition 4.14. We define the *ordered Grossman–Larson*[4] product on $\mathcal{N}$ for all $\omega, \omega' \in \mathrm{OT}^*$ as

$$\omega \bullet \omega' = B^-(\omega[B^+(\omega')]).$$

I.e. we add a root to ω', graft on ω and finally remove the root again.

Proposition 4.15. *The GL-product is associative and, for all $n, n', n'' \in \mathcal{N}$, satisfies*

$$n[n'[n'']] = (n \bullet n')[n''] \tag{4.23}$$

$$\mathcal{F}_\nu(n \bullet n') = \mathcal{F}_\nu(n) \bullet \mathcal{F}_\nu(n'). \tag{4.24}$$

Remark 4.16. The classical setting of Cayley, Merson and Butcher is the case where $\mathcal{M} = \mathbb{R}^n$ and $\mathfrak{g} = \{\partial/\partial x_i\} \subset \mathcal{XM}$ is the standard commutative coordinate frame. The construction of Section 4.3.1 produces $U(\mathfrak{g})^{\mathcal{M}}$ as a D-algebra where the concatenation is commutative. The connection is now flat and torsionless, and $f[g]$ becomes a pre-Lie product. The images of the trees $\mathcal{F}(\tau)$, for $\tau \in \mathrm{OT}$, are called the *elementary differentials* in Butcher's theory (see [8]). These are explicitly given in (2.3). The images of the forests $\mathcal{F}(\omega)$, for $\omega \in \mathrm{OT}^*$, are called *elementary differential operators* in Merson's theory (see [36]).

4.3.3 A generalized Connes–Kreimer Hopf algebra of planar trees

We recall from [42] the definition of the Hopf algebra $\mathcal{H}_N$. On the vector space $\mathbb{R}\langle \mathrm{OT} \rangle$ we define the shuffle product $\sqcup\!\sqcup$, and we define the coproduct Δ_N as the dual of the ordered GL product, such that

$$(\alpha \bullet \beta)(\omega) = \sum_{(\omega)_{\Delta_N}} \alpha(\omega_{(1)})\beta(\omega_{(2)}) \quad \text{for all } \alpha, \beta \in \mathbb{R}\langle\langle \mathrm{OT} \rangle\rangle\}. \tag{4.25}$$

The motivation for this construction is the representation of $U(\mathcal{XM})$ in terms of a frame $\mathfrak{g} \subset \mathcal{XM}$ as $U(\mathfrak{g})^{\mathcal{M}}$. The shuffle product is the correct product to characterize which series in $\mathbb{R}\langle\langle \mathrm{OT} \rangle\rangle$ represent vector fields on $\mathcal{M}$ and which represent diffeomorphisms. The composition in $U(\mathcal{XM})$ appears as the product $\bullet$ on $U(\mathfrak{g})^{\mathcal{M}}$, thus with the coproduct Δ_N the convolution on $\mathbb{R}\langle\langle \mathrm{OT} \rangle\rangle$ represents composition in $U(\mathcal{XM})$.

It remains to give a precise characterization of Δ_N and the antipode in $\mathcal{H}_N$. As in the Connes–Kreimer case, both Δ_N and the antipode can be defined directly in terms of admissible cuts or in a recursive fashion. Recursively Δ_N is given as

$$\begin{aligned} \Delta_N(\mathbb{I}) &= \mathbb{I} \otimes \mathbb{I}, \\ \Delta_N(\omega\tau) &= \omega\tau \otimes \mathbb{I} + \Delta_N(\omega) \sqcup\!\sqcup \cdot (I \otimes B_c^+)\Delta_N(\omega_1), \end{aligned} \tag{4.26}$$

where $\tau = B_c^+(\omega_1) \in \mathrm{OT}$, where $\omega, \omega_1 \in \mathrm{OT}^*$ and where $\sqcup\!\sqcup \cdot$ denotes shuffle on the left and concatenation on the right: $(\omega_1 \otimes \tau_1) \sqcup\!\sqcup \cdot (\omega_2 \otimes \tau_2) = (\omega_1 \sqcup\!\sqcup \omega_2) \otimes (\tau_1\tau_2)$. The direct formula is

$$\Delta_N(\omega) = \sum_{\ell \in \mathrm{FALC}(\omega)} P^\ell(\omega) \otimes R^\ell(\omega), \tag{4.27}$$

where FALC denotes *Full Admissible Left Cuts*, $P^\ell(\omega)$ is the shuffle of all the cut off parts, and $R^\ell(\omega)$ is the remaining part containing the root (see [42]). Calculations of the coproduct for forests up to order 4 can be found in Table 1.

Theorem 4.17. *Let $\mathcal{H}_N$ be the vector space $\mathcal{N} = \mathbb{R}\langle \mathrm{OT} \rangle$ with the operations*

$$\begin{aligned} \textit{product}&: \mu_N(a \otimes b) = a \sqcup\!\sqcup b, \\ \textit{coproduct}&: \Delta_N, \\ \textit{unit}&: u_N(1) = \mathbb{I}, \\ \textit{counit}&: e_N(\omega) = \begin{cases} 1, & \textit{if } \omega = \mathbb{I}, \\ 0, & \textit{else.} \end{cases} \end{aligned}$$

[4] The GL product is usually defined in a similar way over non-planar trees.

Then $\mathcal{H}_N$ is a Hopf algebra with an antipode S_N given by the recursion

$$\begin{aligned} S_N(\mathbb{I}) &= \mathbb{I}, \\ S_N(\omega\tau) &= -\mu_N\left((S_N \otimes I)\left(\Delta_N(\omega) \sqcup\!\sqcup \cdot (I \otimes B_i^+)\Delta_N(\omega_1)\right)\right), \end{aligned} \tag{4.28}$$

where $\tau = B_i^+(\omega_1) \in \mathrm{OT}$ and $\omega, \omega_1 \in \mathrm{OT}^*$.

4.3.4 Lie–Butcher series and flows on manifolds

The set of maps $U(\mathfrak{g})^{\mathcal{M}}$ from $\mathcal{M}$ to $U(\mathfrak{g})$ is a D-algebra where the derivations are the vector fields $\mathfrak{g}^{\mathcal{M}}$. Thus, given a set of colors $\mathcal{C}$ and a map $\nu\colon \mathcal{C} \to \mathfrak{g}^{\mathcal{M}}$ there exists a unique map $\mathcal{F}_\nu\colon \mathcal{N} \to U(\mathfrak{g})^{\mathcal{M}}$ such that for all $c \in \mathcal{C}$ and all $g, h \in \mathcal{N}$ we have

$$\begin{aligned} \mathcal{F}_\nu(c) &= \nu(c) \\ \mathcal{F}_\nu(\mathbb{I}) &= \mathbb{I} \\ \mathcal{F}_\nu(gh) &= \mathcal{F}_\nu(g)\mathcal{F}_\nu(h) \\ \mathcal{F}_\nu(g[h]) &= \mathcal{F}_\nu(g)[\mathcal{F}_\nu(h)] \\ \mathcal{F}_\nu(g\bullet h) &= \mathcal{F}_\nu(g)\bullet\mathcal{F}_\nu(h). \end{aligned} \tag{4.29}$$

Definition 4.18. For an infinite series $\alpha \in \mathcal{N}^* = \mathbb{R}\langle\langle \mathrm{OT}\rangle\rangle$ a Lie–Butcher series is a formal series in $U(\mathfrak{g})^{\mathcal{M}}$ defined as

$$\mathcal{B}_t(\alpha) = \sum_{\omega \in \mathrm{OT}^*} t^{|\omega|}\alpha(\omega)\mathcal{F}_\nu(\omega).$$

Note that $\mathcal{N}$ can be turned into a Hopf algebra two different ways: either as $\mathcal{H}_{\mathrm{Sh}}$ with product $\sqcup\!\sqcup$ and deconcatenation coproduct Δ_d, or as $\mathcal{H}_N$ with the same product $\sqcup\!\sqcup$, but where the coproduct Δ_N is the dual of the ordered GL product. This gives rise to two different convolutions on $\mathcal{N}^*$, the frozen composition $\alpha, \beta \mapsto \alpha\beta$ in Example 3.9, and the non-frozen composition $\alpha, \beta \mapsto \alpha\bullet\beta$ as in (4.25). Since the product is the same, we have that the characters and the infinitesimal characters are the same as vector spaces

$$\begin{aligned} \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}}) = \mathfrak{g}(\mathcal{H}_N) &= \{\, \alpha \in \mathcal{N} \mid \alpha(\mathbb{I}) = 0, \alpha(\omega \sqcup\!\sqcup \omega') = 0 \text{ for all } \omega, \omega' \in \mathrm{OT}^* \setminus \mathbb{I} \,\} \\ G(\mathcal{H}_{\mathrm{Sh}}) = G(\mathcal{H}_N) &= \{\, \alpha \in \mathcal{N} \mid \alpha(\mathbb{I}) = 1, \alpha(\omega \sqcup\!\sqcup \omega') = \alpha(\omega)\alpha(\omega') \text{ for all } \omega, \omega' \in \mathrm{OT}^* \,\}. \end{aligned}$$

Hovever, the exponential, logarithm, Dynkin and Eulerian idempotents, as well as the antipode depend on whether they are based on $\mathcal{H}_{\mathrm{Sh}}$ or $\mathcal{H}_N$. Which to use in practice depends on which operation we want to express on the manifold. Recall that *frozen elements* of $U(\mathfrak{g})^{\mathcal{M}}$ are constant functions $g\colon \mathcal{M} \to U(\mathfrak{g})$. If g is frozen then $f[g] = 0$ for all f, and hence $f\bullet g = fg$. The subalgebra of frozen vector fields therefore reduces to $\mathcal{H}_{\mathrm{Sh}}$.

We summarize the basic properties of LB-series: $\mathcal{B}_t$ sends infinitesimal characters to (formal) vector fields on $\mathcal{M}$ and characters to pullback series representing formal diffeomorphisms on $\mathcal{M}$. LB-series preserve both frozen and non-frozen composition and sends left grafting to the connection on $U(\mathfrak{g})^{\mathcal{M}}$.

$$\begin{aligned} \mathcal{B}_t(\alpha\beta) &= \mathcal{B}_t(\alpha)\mathcal{B}_t(\beta) \\ \mathcal{B}_t(\alpha\bullet\beta) &= \mathcal{B}_t(\alpha)\bullet\mathcal{B}_t(\beta) \\ \mathcal{B}_t(\alpha[\beta]) &= \mathcal{B}_t(\alpha)[\mathcal{B}_t(\beta)]. \end{aligned}$$

Note that if $\alpha \in G(\mathcal{H}_N)$, then $\alpha[\beta]$ represents algebraically the pullback (parallel transport) of β along the flow of α. On the manifold

$$\mathcal{B}_h(\alpha[\beta])(y_0) = \mathcal{B}_h(\alpha)[\mathcal{B}_h(\beta)](y_0) = \mathcal{B}_h(\beta)(\Phi(y_0)),$$

where Φ is the diffeomorphism represented by $\alpha \in G(\mathcal{H}_N)$ at $t = h$. Since the connection is flat, the pullback depends only on the endpoint $\Phi(y_0)$ and not on the actual path.

There are (at least) three ways to represent a flow $y_0 \mapsto y_t = \Phi_t(y_0)$ on $\mathcal{M}$, using LB-series:

1. In terms of pullback series. Find $\alpha \in G(\mathcal{H}_N)$ such that
$$\psi(y(t)) = \mathcal{B}_t(\alpha)(y_0)[\psi] \quad \text{for any } \psi \in U(\mathfrak{g})^{\mathcal{M}}. \tag{4.30}$$
This representation is used in the analysis of Crouch–Grossman methods by Owren and Marthinsen [45]. In the classical setting, this is called a S-series [43].

2. In terms of an autonomous differential equation. Find $\beta \in \mathfrak{g}(\mathcal{H}_N)$ such that $y(t)$ solves
$$y'(t) = \mathcal{B}_h(\beta)(y(t)). \tag{4.31}$$
In the classical setting, this is called backward error analysis. In the Lie group setting, this formulation has, however, never been investigated in detail (but it should!).

3. In terms of a non-autonomous equation of *Lie type* (time dependent frozen vector field). Find $\gamma \in \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})$ such that $y(t)$ solves
$$y'(t) = \left(\frac{\partial}{\partial t}\mathcal{B}_t(\gamma)(y_0)\right) y(t). \tag{4.32}$$
This representation is used in [38, 39]. In the classical setting this is (almost) the standard definition of B-series. The connection with the classical B-series is discussed below.

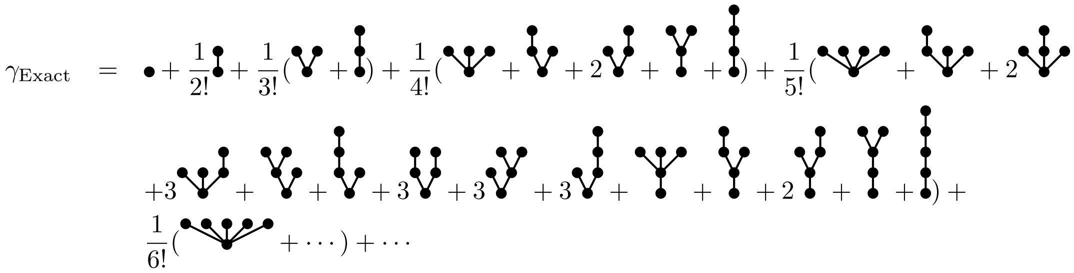
The algebraic relationship between α, β and γ is given as follows:

$\beta = \alpha \circ e$	e is Euler idempotent in $\mathcal{H}_N$.
$\alpha = \exp^{\bullet}(\beta)$	Exponential wrt. GL-product
$\gamma = \alpha \circ Y^{-1} \circ D$	Dynkin idempotent in $\mathcal{H}_{\mathrm{Sh}}(\mathrm{OT})$.
$\alpha = Q(\gamma)$	Q-operator (4.12) in $\mathcal{H}_{\mathrm{Sh}}(\mathrm{OT})$.

Example 4.19. Two examples are of particular interest; the exact solution and exponential Euler method. In both cases we consider $y'(t) = f(y)\cdot y$, where $\mathcal{C} = \{\bullet\}$ and $\nu(\bullet) = f$.

The exponential Euler method is particularly simple. Since each step of the method follows the flow the frozen vector field $f(y_n) \in \mathfrak{g}$, the Type 3 LB-series for Exponential Euler must be given by
$$\gamma_{\mathrm{Euler}} = \bullet$$
just as in the classical setting[5].

Type 3 LB-series for the exact solution can be derived in various ways. Theorem 2.2 in [39] derives the exact solution as the solution of
$$y' = f_t \cdot y, \quad y(0) = y_0,$$
where $f_t = f(y(t)) \in \mathfrak{g}$ is the pullback of f along the time dependent flow of f_t. Letting $f_t = \frac{\partial}{\partial t}\mathcal{B}_t(\gamma)$ we obtain
$$Y \circ \gamma = Q(\gamma)[\bullet] \Rightarrow \gamma = Y^{-1} \circ B^+(Q(\gamma)).$$
Note that this is reminiscent of a so-called combinatorial Dyson–Schwinger equation [24]. Solving by iteration yields



γ_{Exact} =
$\bullet + \frac{1}{2!}$ + $\frac{1}{3!}$(+) + $\frac{1}{4!}$(+ + 2 + +) + $\frac{1}{5!}$(+ + 2
+3 + + + 3 + 3 + 3 + + + 2 + +) +
$\frac{1}{6!}$(+ ⋯) + ⋯

[5] The classical presentation is $\gamma = \mathbb{I} + \bullet$, when the B-series is given in the form (2.2).

Remarkably, the LB-series of the exact solution is just a combination of trees, and not commutators of trees. Thus in Type 3 LB-series developments of numerical integrators, commutators of trees must be zero up to to the order of the method.

Composition and inverse is simplest for pullback series, Type 1. For series of Type 3, we map to Type 1, compose (or invert) and map back again. If $\gamma, \tilde{\gamma}$ are series of Type 3, then the basic operations are done as:

$$\begin{aligned} \text{Composition:} \quad & \gamma, \tilde{\gamma} \mapsto (Q(\gamma) \bullet Q(\tilde{\gamma})) \circ Y^{-1} \circ D & (4.33) \\ \text{Inverse:} \quad & \gamma^{-1} = Q(\gamma) \circ S \circ Y^{-1} \circ D & (4.34) \\ \text{Backward error:} \quad & \operatorname{Log}_3(\gamma) := Q(\gamma) \circ e. & (4.35) \end{aligned}$$

4.3.5 Relations to classical B-series

The relation between classical B-series and LB-series is detailed in [42]. Classical B-series are expressed in terms of linear combinations of non-planar trees T, resulting in the Connes–Kreimer Hopf algebra $\mathcal{H}_C$ built from non-planar trees [4]. In the classical setting the connection is torsion-free, and concatenation is commutative. Therefore $\mathfrak{g}(\mathcal{H}_C) = \operatorname{span}(T)$. That is, $\mathfrak{g}(\mathcal{H}_C)$ is just linear combinations of trees. This fact is the reason why many discussions in the classical setting can avoid series involving forests of trees (words in T^*). Also the difference between series of Type 1 and Type 3 is in not emphasized in many papers. Since the coefficients κ of the Q-polynomials add up to one under symmetrization, we find in the classical setting that

$$Q(\alpha)(\omega) = \alpha(\tau_1)\alpha(\tau_2)\cdots\alpha(\tau_k)\omega,$$

for $\omega = B^+(\tau_1\tau_2\cdots\tau_k)$, so formulas involving pullbacks are often expressed directly from B-series (Type 3) using the Q-polynomials in this form. Our claim that classical B-series fits best into series of Type 3 is based on the trivial observation that the curve $y_t = \mathcal{B}_t(\gamma)(y)$ in (4.32) solves a differential equation with a time dependent frozen vector field given as

$$y(t) = \frac{\partial}{\partial t} \sum_{\tau \in T} \frac{t^{|\tau|}}{\sigma(\tau)} \mathcal{F}(\tau).$$

One can ask why the symmetrization $\sigma(\tau)$ is natural to include in the classical setting, but not in the LB-series setting. To explain the relationship between the two theories we define a symmetrization operator:

Definition 4.20. The symmetrization operator $\Omega : \mathcal{N} \to \mathcal{N}$ is defined for $\omega \in \mathrm{OT}^*$ and $\tau \in \mathrm{OT}$ as

$$\begin{aligned} \Omega(\mathbb{I}) &= \mathbb{I}, \\ \Omega(\omega\tau) &= \Omega(\omega) \sqcup\!\sqcup\, \Omega(\tau), \\ \Omega(B_i^+(\omega)) &= B_i^+(\Omega(\omega)). \end{aligned}$$

The shuffle product permutes the trees in a forest in all possible ways, and the symmetrization of a tree is a recursive splitting in sums over all permutations of the branches. The symmetrization defines an equivalence relation on OT^*, that is

$$\Omega(\omega_1) = \Omega(\omega_2) \quad \Longleftrightarrow \quad \omega_1 \sim \omega_2.$$

Let $\iota : \mathcal{H}_C \to \mathcal{H}_N$ be an inclusion where a tree is identified with one of its equivalent planar trees. In [42] we show that $\tilde{\Omega} = \Omega \circ \iota \colon \mathcal{H}_C \to \mathcal{H}_N$ is a Hopf algebra isomorphism onto its image, i.e. $\mathcal{H}_C$ is a proper subalgebra of $\mathcal{H}_N$. The adjoint map $\tilde{\Omega}^* \colon \mathcal{H}_N^* \to \mathcal{H}_C^*$ is given as

$$\tilde{\Omega}^*(\alpha)(\omega) = \sigma(\omega) \sum_{\omega' \sim \omega} \alpha(\omega').$$

The tree symmetrization $\sigma(\omega)$ enters exactly such that the LB-series as given in (4.18) maps to the classical B-series in (2.2).

4.4 Substitution law for LB-series

The so-called *substitution law* for B-series [14] can without much difficulty be generalized to LB series. Consider $\mathcal{N}$ as a D-algebra where the derivations are the Lie polynomials $D(\mathcal{N}) = \mathfrak{g}(\mathcal{H}_N) \cap \mathcal{N}$. By the universality property of $\mathcal{N}$, we know that for any map $a\colon \mathcal{C} \to D(\mathcal{N})$ there exists a unique D-algebra homomorphism $\mathcal{F}_a : \mathcal{N} \to \mathcal{N}$ such that $\mathcal{F}_a(c) = a(c)$ for all $a \in \mathcal{C}$. This is called the substitution law.

Definition 4.21. For any map $a\colon \mathcal{C} \to D(\mathcal{N})$ there exists a unique D-algebra homomorphism $a\star : \mathcal{N} \to \mathcal{N}$ such that $a(c) = a \star c$ for all $c \in \mathcal{C}$. The map $a\star$ is called a-substitution [6].

$$\begin{tikzcd}
\mathcal{C} \arrow[r, hook] \arrow[d, "a"'] & \mathcal{N} \arrow[d, "a\star"] \\
D(\mathcal{N}) \arrow[r, hook] & \mathcal{N}.
\end{tikzcd}$$

The properties of this substitution law, together with applications of it, will be studied in a forthcoming paper ([34]). We just mention that many of the useful properties of the substitution law follow immediately from the fact that $a\star : \mathcal{N} \to \mathcal{N}$ is a homomorphism. For example, for all $n, n' \in \mathcal{N}$ we have:

$$\begin{aligned}
a \star \mathbb{I} &= \mathbb{I} \\
a \star (nn') &= (a \star n)(a \star n') \\
a \star (n[n']) &= (a \star n)[a \star n'] \\
a \star (n \bullet n') &= (a \star n) \bullet (a \star n')
\end{aligned}$$

5 Final remarks and outlook

Inspired by problems in numerical analysis we have discussed various algebraic structures arising in the study of formal diffeomorphisms on manifolds. We have seen that the Connes–Kreimer Hopf algebra naturally extends from commutative frames on $\mathbb{R}^n$ to non-commutative frames on general manifolds. In particular we have presented the Dynkin and Euler operators and non-commutative Faà di Bruno type bialgebras in this generalized setting.

The formalism in this paper has many applications in numerical analysis, and analysis of Lie group integrators in particular. However, the underlying structures are general constructions with possible applications in other fields, such as geometric control theory and sub-Riemannian geometry. Connections to stochastic differential equations on manifolds is an other topic which is worth investigating further.

Acknowledgements

The authors would like to thank Alessandra Frabetti, Dominique Manchon, Gilles Vilmart and Will Wright for interesting discussions on topics of this paper. In particular we would like to thank Kurusch Ebrahimi-Fard for his support and useful remarks in the writing process. His enthusiasm and inclusive spirit have been of crucial importance for the completion of this paper.

[6] In most applications we want to substitute infinite series and extend $a\star$ to a homomorphism $a\star\colon \mathcal{N}^* \to \mathcal{N}^*$. The extension to infinite substitution is straightforward because of the grading, we omit details.

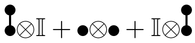
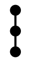
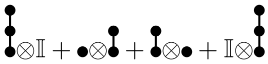
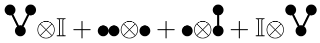
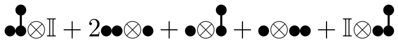
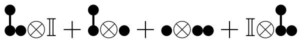
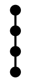
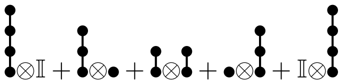
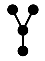
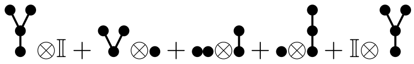
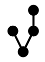
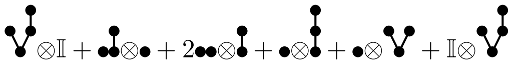
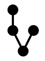
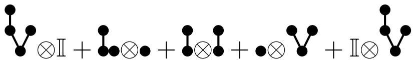
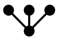
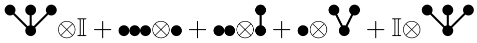
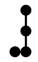
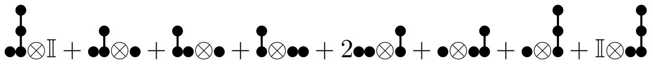
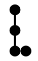
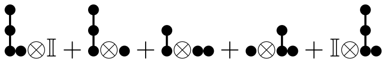
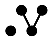
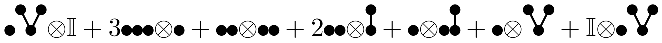
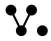
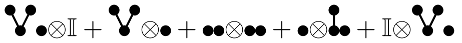
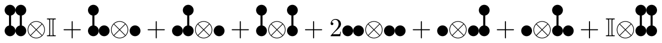
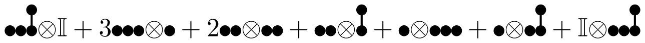
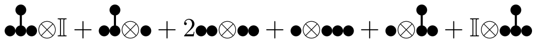
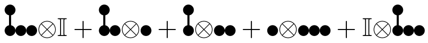
ω	$\Delta_N(\omega)$
$\mathbb{I}$	$\mathbb{I}\otimes\mathbb{I}$
$\bullet$	$\bullet\otimes\mathbb{I}+\mathbb{I}\otimes\bullet$
	
$\bullet\bullet$	$\bullet\bullet\otimes\mathbb{I}+\bullet\otimes\bullet+\mathbb{I}\otimes\bullet\bullet$
	
	
	
	
$\bullet\bullet\bullet$	$\bullet\bullet\bullet\otimes\mathbb{I}+\bullet\bullet\otimes\bullet+\bullet\otimes\bullet\bullet+\mathbb{I}\otimes\bullet\bullet\bullet$
	
	
	
	
	
	
	
	
	
	
	
	
	
$\bullet\bullet\bullet\bullet$	$\bullet\bullet\bullet\bullet\otimes\mathbb{I}+\bullet\bullet\bullet\otimes\bullet+\bullet\bullet\otimes\bullet\bullet+\bullet\otimes\bullet\bullet\bullet+\mathbb{I}\otimes\bullet\bullet\bullet\bullet$

Table 1: Examples of the coproduct Δ_N, defined in (4.26).

References

[1] E. Abe. *Hopf Algebras*. Cambridge University Press, 1980.

[2] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*. AMS 75. Springer-Verlag, Second edition, 1988.

[3] H. Berland and B. Owren. Algebraic structures on ordered rooted trees and their significance to Lie group integrators. *Group theory and numerical analysis*, 39:49–63, 2005.

[4] C. Brouder. Runge-Kutta methods and renormalization. *The European Physical Journal C-Particles and Fields*, 12(3):521–534, 2000.

[5] C. Brouder. Trees, renormalization and differential equations. *BIT*, 44(3):425–438, 2004.

[6] C. Brouder, A. Frabetti, and C. Krattenthaler. Non-commutative Hopf algebra of formal diffeomorphisms. *Advances in Mathematics*, 200(2):479–524, 2006.

[7] E. Burgunder. Eulerian idempotent and Kashiwara-Vergne conjecture. 58(4):1153–1184, 2008.

[8] J. C. Butcher. Coefficients for the study of Runge-Kutta integration processes. *J. Austral. Math. Soc.*, 3:185–201, 1963.

[9] J. C. Butcher. An algebraic theory of integration methods. *Math. Comp.*, 26:79–106, 1972.

[10] D. Calaque, K. Ebrahimi-Fard, and D. Manchon. Two interacting Hopf algebras of trees. *To appear in Adv. Appl. Math*, 2009, math.CO/0806.2238v3.

[11] P. Cartier. A primer of Hopf algebras. In *Frontiers in number theory, physics, and geometry*, volume II, pages 537–615. Springer, Berlin, 2007.

[12] A. Cayley. On the theory of the analytical forms called trees. *Philos. Mag*, 13(19):4–9, 1857.

[13] E. Celledoni, A. Marthinsen, and B. Owren. Commutator-free Lie group methods. *Future Generation Computer Systems*, 19(3):341–352, 2003.

[14] P. Chartier, E. Hairer, and G. Vilmart. A substitution law for B-series vector fields. *INRIA report*, (5498), 2005.

[15] P. Chartier, E. Hairer, and G. Vilmart. Numerical integrators based on modified differential equations. *Mathematics of Computation*, 76(260):1941, 2007.

[16] P. Chartier and A. Murua. An algebraic theory of order. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(4):607–630, 2009.

[17] A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Communications in Mathematical Physics*, 199(1):203–242, 1998.

[18] P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear Sci.*, 3:1–33, 1993.

[19] A. Dür. *Möbius functions, incidence algebras and power series representations*, volume 1202 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1986.

[20] K. Ebrahimi-Fard, J.M. Gracia-Bondía, and F. Patras. A Lie Theoretic Approach to Renormalization. *Communications in Mathematical Physics*, 276(2):519–549, 2007.

[21] K. Ebrahimi-Fard and D. Manchon. A Magnus-and Fer-type formula in dendriform algebras. *Foundations of Computational Mathematics*, 9:1–22, 2009, math.CO/07070607v3.

[22] H. Figueroa and J.M Gracia-Bondia. Combinatorial Hopf algebras in quantum field theory I. *Rev.Math.Phys.*, 17:881, 2005, hep-th/0408145v3.

[23] H. Figueroa, J.M. Gracia-Bondia, and J.C. Varilly. Faa di Bruno Hopf algebras. *Preprint*, 2005, math.CO/0508337.

[24] L. Foissy. Faà di Bruno subalgebras of the Hopf algebra of planar trees from combinatorial Dyson–Schwinger equations. *Advances in Mathematics*, 218(1):136–162, 2008, 0707.1204v2.

[25] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Springer-Verlag, second edition, 2006.

[26] E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing (Arch. Elektron. Rechnen)*, 13(1):1–15, 1974.

[27] A. Iserles, A. Marthinsen, and S.P. Nørsett. On the implementation of the method of Magnus series for linear differential equations. *BIT Numerical Mathematics*, 39(2):281–304, 1999.

[28] A. Iserles, H.Z. Munthe-Kaas, S.P. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.

[29] A. Iserles and S.P. Nørsett. On the solution of linear differential equations in Lie groups. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 357(1754):983–1019, 1999.

[30] C. Kassel. *Quantum groups*. Springer-Verlag, 1995.

[31] R. Lenczewski. A noncommutative limit theorem for homogeneous correlations. *Studia Mathematica*, 129(3), 1998.

[32] J.L. Loday. *Cyclic Homology*. Springer-Verlag, second edition, 1997.

[33] J.L. Loday and M.O. Ronco. Combinatorial Hopf algebras. *Clay Mathematics Proceedings*, 12:347–384, 2010, math.CO/0508337.

[34] A. Lundervold and H.Z. Munthe-Kaas. Backward error analysis and the substitution law for Lie group integrators. *Preprint*, 2010.

[35] D. Manchon. Hopf algebras, from basics to applications to renormalization. *Preprint*, 2006, math.QA/0408405v2.

[36] R. H. Merson. An operational method for the study of integration processes. In *Proc. Conf., Data Processing & Automatic Computing Machines*, pages 110–1–11025, 1957.

[37] S. Monaco, D. Normand-Cyrot, and C. Califano. From chronological calculus to exponential representations of continuous and discrete-time dynamics: a Lie-algebraic approach. *IEEE Transactions on Automatic Control*, 52(12):2227–2241, 2007.

[38] H. Munthe-Kaas. Lie–Butcher theory for Runge–Kutta methods. *BIT*, 35(4):572–587, 1995.

[39] H. Munthe-Kaas. Runge–Kutta methods on Lie groups. *BIT*, 38(1):92–111, 1998.

[40] H. Munthe-Kaas and S. Krogstad. On enumeration problems in Lie–Butcher theory. *Future Generation Computer Systems*, 19(7):1197–1205, 2003.

[41] H. Munthe-Kaas and B. Owren. Computations in a free Lie algebra. *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.*, 357(1754):957–981, 1999.

[42] H.Z. Munthe-Kaas and W. Wright. On the Hopf algebraic structure of Lie group integrators. *Found. Comput. Math*, 8(2):227 – 257, 2008, math/0603023v1.

[43] A. Murua. Formal series and numerical integrators, Part I: Systems of ODEs and symplectic integrators. *Applied numerical mathematics*, 29(2):221–251, 1999.

[44] B. Owren. Order conditions for commutator-free Lie group methods. *Journal of Physics A–Mathematical and General*, 39(19):5585–5600, 2006.

[45] B. Owren and A. Marthinsen. Runge–Kutta methods adapted to manifolds and based on rigid frames. *BIT*, 39(1):116–142, 1999.

[46] F. Patras. On Dynkin and Klyachko idempotents in graded bialgebras. *Advances in Applied Mathematics*, 28(3/4):560–579, 2002.

[47] C. Reutenauer. *Free Lie algebras.* Oxford University Press, 1993.

[48] M. E. Sweedler. *Hopf algebras.* Mathematics Lecture Note Series. W. A. Benjamin, Inc., New York, 1969.

Paper B

Backward error analysis and the substitution law for Lie group integrators [*]

[*] arXiv: `http://arxiv.org/abs/1106.1071`

Backward error analysis and the substitution law for Lie group integrators

Alexander Lundervold [*]   Hans Munthe-Kaas [†]

Keywords: Backward error analysis, Butcher series, Hopf algebras, Lie group integrators, Lie–Butcher series, rooted trees, substitution law
Mathematics Subject Classification (2010): 65L05, 65L06, 37C10

Communicated by Elizabeth Mansfield

Abstract

Butcher series are combinatorial devices used in the study of numerical methods for differential equations evolving on vector spaces. More precisely, they are formal series developments of differential operators indexed over rooted trees, and can be used to represent a large class of numerical methods. The theory of backward error analysis for differential equations has a particularly nice description when applied to methods represented by Butcher series. For the study of differential equations evolving on more general manifolds, a generalization of Butcher series has been introduced, called Lie–Butcher series. This paper presents the theory of backward error analysis for methods based on Lie–Butcher series.

1 Introduction

A fundamental tool in the field of numerical integration of ordinary differential equations on $\mathbb{R}^n$ is the theory of *Butcher series* (B-series). These are formal series expansions of vector fields and flows, expanded over the set of rooted trees. Many numerical methods can be formulated in terms of B-series, and they can be used to, for example, study order theory, structure preserving properties of integrators, backward error analysis and modified vector fields [3, 17, 8, 7, 10, 9, 23]. In the more general setting of differential equations of the form

$$y' = F(y), \;\; y \in M, \;\; F : M \to TM, \tag{1}$$

where M is a (homogeneous) manifold and F a vector field on M, the role of B-series is played by the *Lie–Butcher series* (LB-series) [28, 22, 23]. Considering the importance of classical B-series, LB-series are objects of great interest.

The B-series are based on the *elementary differentials* associated to vector fields, and these can be constructed as homomorphisms from the free *pre-Lie algebra* (or *Vinberg algebra*) into the pre-Lie algebra of vector fields [5]. In the setting of LB-series we get a similar picture, only now the pre-Lie algebras are replaced by the so-called *post-Lie algebras*, defined in [34, 27].

In the present paper we will explore the *substitution law* for Lie–Butcher series, formulated in the language of enveloping algebras of post-Lie algebras: the *D-algebras* of [28]. Once the substitution law is understood, it can be applied to *backward error analysis.* The basic idea of backward error analysis is to interpret the numerical solution of a differential equation as the exact solution of a modified equation, and then use this equation to study the numerical method. Analogous to classical backward error analysis (as developed in [16, 17, 8, 5]), its generalization to the Lie group setting has a particularly nice description for methods based on Lie–Butcher series.

[*] Corresponding author. Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway. alexander.lundervold@gmail.com

[†] Department of Mathematics, University of Bergen, N-5020 Bergen, Norway. hans.munthe-kaas@math.uib.no

Note that the construction of series expansions in the present paper is purely formal: there will be no study of convergence. This separation between the algebraic and the analytic framework for backward error analysis is also present in the setting of B-series, where the main algebraic references are [17, 8, 5, 9] and the analytic references are [1, 32, 17]. An analytic study of backward error analysis for Lie group methods can be found in [14].

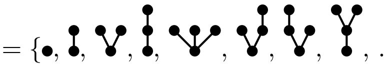
The present study of the backward error and substitution law for Lie group integrators is interesting from a purely algebraic point of view, as this work provides an explicit description of automorphisms of post-Lie algebras. From a numerical point of view, the theory has several applications. The algebraic structures of backward error analysis is important in the analysis of numerical integration algorithms. Additionally, in the case of classical B-series, such algebraic techniques have recently been applied more directly as a computational tool [8]. Similar techniques in the setting of Lie group integrators is a promising approach to structure preserving integration of problems of computational mechanics, such as Lie-Poisson systems.

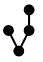
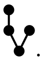
2 Lie–Butcher series

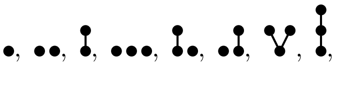
In this section we will define D-algebras, and show how they give rise to Lie–Butcher series. In the next section we will apply them to the study of the substitution law and backward error analysis for Lie group integrators on manifolds.

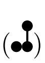
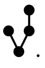
2.1 Trees and D-algebras

Ordered rooted trees and forests. Some basic definitions follow. For a more comprehensive introduction to the combinatorics of trees applied to numerical integration, see [4] or [19]. Let OT denote the alphabet of all ordered (i.e. planar) rooted trees:

OT = {  , . . .}.

The root is the bottom vertex and we consider the trees to grow upwards from the root. The trees being ordered implies that  $\neq$ . This is different from classical B-series theory, where the order of the branches is of no significance. Let OF denote the set of ordered forests, i.e. all possible empty and non-empty words written with letters from the alphabet OT:

OF = { $\mathbb{I}$, , $\cdots$ },

where $\mathbb{I}$ denotes the empty word. On OF we define the *concatenation product* $\omega_1, \omega_2 \mapsto \omega_1\omega_2$, which creates a longer word by joining ω_1 and ω_2 end-to-end. This is an associative, non-commutative product with unit $\mathbb{I}$. Let $B^+\colon$ OF $\rightarrow$ OT denote the operation of adding a root to a word, e.g. $B^+($$)$ = . All of OF is generated from $\mathbb{I}$ by concatenation and adding roots. The *order* of a forest, $|\omega| = |\tau_1 \dots \tau_k|$, is defined by the recursion $|\mathbb{I}| = 0$, $|\tau_1 \dots \tau_k| = |\tau_1| + \cdots + |\tau_k|$, $|B^+\omega| = |\omega| + 1$, i.e. the order counts the number of vertices in a forest. Let k be a field of characteristic 0, e.g. k $= \mathbb{R}$ or k $= \mathbb{C}$. The k-vector space of all finite k-linear combinations of elements in OF is the non-commutative polynomial ring over OT [1], denoted by $\mathcal{N} = \mathrm{k}\langle\mathrm{OT}\rangle$. The k-vector space of infinite linear combinations of OF is $\mathcal{N}^* = \mathrm{k}\langle\langle\mathrm{OT}\rangle\rangle$. $\mathcal{N}^*$ is the dual space of $\mathcal{N}$, with the dual pairing $\langle\cdot,\cdot\rangle\colon \mathcal{N}^* \times \mathcal{N} \rightarrow \mathrm{k}$ defined such that the words in OF form a orthonormal basis: $\langle\omega_1, \omega_2\rangle = 0$ if $\omega_1 \neq \omega_2$, and $\langle\omega, \omega\rangle = 1$. Thus for $a \in \mathcal{N}^*$ we have $a(\omega) = \langle a, \omega\rangle$ and

[1] $\mathcal{N}$ with concatenation product can equivalently be defined as the linear space spanned by trees, $V = k\{\mathrm{OT}\}$, equipped with a tensor product. Hence $\mathcal{N}$ can be defined as the tensor algebra on V. However, because we need other tensor products later we prefer the definition via concatenation of words.

$a = \sum_{\omega\in \mathrm{OF}} a(\omega)\omega$. In the latter sum we understand $\mathcal{N}^*$ as the projective limit $\mathcal{N}^* = \varprojlim \mathcal{N}_k$, where $\mathcal{N}_k = \mathrm{span}\{\omega \in \mathrm{OF}\colon |\omega| \leq k\}$. An infinite $a \in \mathcal{N}^*$ is uniquely defined by its finite projections $a_k \in \mathcal{N}_k$ for all $k \in \mathbb{Z}$, where $a_k = \sum_{|\omega|\leq k} a(\omega)\omega$ is the orthogonal projection of a onto the subspace $\mathcal{N}_k \subset \mathcal{N}$.

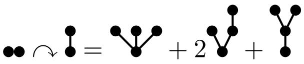
Remark 2.1. In many applications it is necessary to generalize to spaces built from trees with colored vertices. The theory extends from the above presentation with only minor modifications. Let $\mathcal{C}$ be a (finite or infinite) set of colors. A coloring of a tree or a forest is a map from its vertices to $\mathcal{C}$. Let $\mathrm{OT}_{\mathcal{C}}$ and $\mathrm{OF}_{\mathcal{C}}$ denote colored trees and forests. For each $c \in \mathcal{C}$ we have the operation $B_c^+\colon \mathrm{OF}_{\mathcal{C}} \to \mathrm{OT}_{\mathcal{C}}$ creating a tree by adding a root of color c to a word. We identify $\mathcal{C} \subset \mathrm{OT}_{\mathcal{C}} \subset \mathrm{OF}_{\mathcal{C}}$ as the subset of single vertex trees. In the colored context we permit more general gradings $|\cdot|$ on $\mathrm{OT}_{\mathcal{C}}$. We allow the assignment of arbitrary positive integer weights $|c| \in \mathbb{N}$ to the single vertex trees $\mathcal{C} \subset \mathrm{OT}_{\mathcal{C}}$, extended to $\mathrm{OF}_{\mathcal{C}}$ by $|\tau_1 \ldots \tau_k| = |\tau_1| + \cdots + |\tau_k|$ and $|B_c^+\omega| = |\omega| + |c|$. The definitions of finite and infinite linear combinations of forests $\mathcal{N}_{\mathcal{C}} = k\langle \mathrm{OT}_{\mathcal{C}}\rangle$ and $\mathcal{N}_{\mathcal{C}}^* = k\langle\langle \mathrm{OT}_{\mathcal{C}}\rangle\rangle$ are similar to the uni-color case.

Definition 2.2. The *left grafting* product $\cdot \curvearrowright \cdot : \mathcal{N} \otimes \mathcal{N} \to \mathcal{N}$ is defined recursively as follows: let $\tau \in \mathrm{OT}$ and $\omega, \omega_1, \omega_2 \in \mathrm{OF}$. Then

$$\begin{aligned}
\mathbb{I} \curvearrowright \omega &= \omega \\
\tau \curvearrowright \mathbb{I} &= 0 \\
\omega \curvearrowright \bullet &= B^+(\omega), \\
\tau \curvearrowright (\omega_1\omega_2) &= (\tau \curvearrowright \omega_1)\omega_2 + \omega_1(\tau \curvearrowright \omega_2) \\
(\tau\omega) \curvearrowright \omega_1 &= \tau \curvearrowright (\omega \curvearrowright \omega_1) - (\tau \curvearrowright \omega) \curvearrowright \omega_1
\end{aligned}$$

The product is extended to all of $\mathcal{N}$ and $\mathcal{N}^*$ by linearity and projective limits.

For example,



Note that grafting satisfies a Leibniz rule with respect to the concatenation product. If we define $\tau \diamond \omega = \tau\omega + \tau \curvearrowright \omega$, we see that $\tau \curvearrowright (\omega \curvearrowright \omega_1) = (\tau\diamond\omega) \curvearrowright \omega_1$. More generally, $\omega_1 \curvearrowright (\omega_2 \curvearrowright \omega) = (\omega_1 \diamond \omega_2) \curvearrowright \omega$, where $\diamond$ is the associative product defined as follows:

Definition 2.3. The Grossman-Larson product $\diamond : \mathcal{N} \otimes \mathcal{N} \to \mathcal{N}$ of $\omega_1, \omega_2 \in \mathrm{OF}$ is defined in terms of the grafting product as:

$$B^+(\omega_1 \diamond \omega_2) = \omega_1 \curvearrowright B^+(\omega_2),$$

and is extended by linearity.

It is clear that if we write $\omega_1[\omega_2]$ for $\omega_1 \curvearrowright \omega_2$, we have the following structure on $\mathcal{N}$:

Definition 2.4 ([28])**.** Let A be a unital associative algebra with product $f, g \mapsto fg$, unit $\mathbb{I}$ and equipped with a non-associative composition $(.)[.] : A \otimes A \to A$ such that $\mathbb{I}[g] = g$ for all $g \in A$. Write $\mathcal{D}(A)$ for the set of all $f \in A$ such that $f[\cdot]$ is a derivation:

$$\mathcal{D}(A) = \{f \in A \mid f[gh] = (f[g])h + g(f[h]) \text{ for all } g, h \in A\}.$$

Then A is called a *D-algebra* if for any derivation $f \in \mathcal{D}(A)$ and any $g \in A$ we have

$$\begin{aligned}
&\text{(i)} \quad g[f] \in \mathcal{D}(A) \\
&\text{(ii)} \quad f[g[h]] = (fg)[h] + (f[g])[h].
\end{aligned}$$

The free D-algebra. We note that a morphism $\mathcal{F} : A \to A'$ of D-algebras is an algebra morphism satisfying $\mathcal{F}(D(\mathcal{A})) \subset D(\mathcal{A}')$ and $\mathcal{F}(a[b]) = \mathcal{F}(a)[\mathcal{F}(b)]$ for all $a, b \in A$. The D-algebra $\mathcal{N}$ plays a special role: it is a universal object.

Proposition 2.5 ([28]). *Let* OT *be planar trees decorated with colors $\mathcal{C}$. The vector space $\mathcal{N} = \mathbb{R}\langle \mathrm{OT} \rangle$ is a free D-algebra over $\mathcal{C}$. That is, for any D-algebra $\mathcal{A}$ and any map $\nu \colon \mathcal{C} \to D(\mathcal{A})$ there exists a unique D-algebra homomorphism $\mathcal{F}_\nu \colon N \to \mathcal{A}$ such that $\mathcal{F}_\nu(c) = \nu(c)$ for all $c \in \mathcal{C}$.*

$$\begin{tikzcd}
\mathcal{C} \arrow[r, hook] \arrow[d, "\nu"'] & \mathcal{N} \arrow[d, "\exists !\, \mathcal{F}_\nu"] \\
D(\mathcal{A}) \arrow[r, hook] & \mathcal{A}
\end{tikzcd}$$

We will see that based on this result we can construct elementary differentials and Lie–Butcher series for Lie group integrators, and also define the substitution law. To achieve this we utilize D-algebra structure of differential operators on manifolds [28].

The D-algebra of differential operators. There is a D-algebra based on the space of vector fields[2] on the manifold M. Consider the space $\mathcal{C}^\infty(M, \mathfrak{g}) =: \mathfrak{g}^M$, where $\mathfrak{g} \subset \mathcal{X}M$ is a Lie subalgebra of the set of all vector fields on M. For $\Psi \in \mathfrak{g}^M$ and $V \in \mathfrak{g}$, the *Lie derivative* $V[\Psi] \in \mathfrak{g}^M$ of Ψ along V defined by

$$V[\Psi](p) := \frac{d}{dt}\,|_{t=0}\, \Psi(\exp(tV)(p)). \tag{2}$$

$V[\cdot]$ is a first order differential operator on $\mathfrak{g}^M$, satisfying $V[h\Psi] = V[h]\Psi + hV[\Psi]$, where $h \in C^\infty(M, \mathbb{R})$ is a scalar function[3]. The Lie derivative gives rise to differential operators of higher degrees through concatenation: the concatenation of $V, W \in \mathfrak{g}$ is a second-degree differential operator defined by $VW[\Psi] := V[W[\Psi]]$. The $C^\infty(M, \mathbb{R})$-module of all differential operators, including the ones of higher degree, and the degree zero operator spanned by the identity operator $\mathbb{I}$, is called the *universal enveloping algebra* $U(\mathfrak{g})$ of $\mathfrak{g}$. We extend the structure to the space $\mathcal{C}^\infty(M, U(\mathfrak{g})) =: U(\mathfrak{g})^M$ as follows: for $f, g \in U(\mathfrak{g})^M$, $f[g] \in U(\mathfrak{g})^M$ is defined by

$$f[g](p) := (f(p)[g])(p) \tag{3}$$

and $fg \in U(\mathfrak{g})^M$ is defined as

$$fg(p) := f(p)g(p). \tag{4}$$

The latter operation is called the *frozen composition* of f and g. For two vector fields f and g written in terms of the standard coordinate frame $\{\partial/\partial x_i\}$, the operations take the following form:

$$\begin{aligned} f[g] &= \sum_{i,j} f_j \frac{\partial g_i}{\partial x_j} \frac{\partial}{\partial x_i} && (5) \\ fg &= \sum_{i,j} f_i g_j \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_j} && (6) \end{aligned}$$

The operations (3) and (4) endows the space $U(\mathfrak{g})^M$ with the structure of a D-algebra, where the derivations are the vector fields in $\mathfrak{g}^M$:

Lemma 2.6. *Let $f \in \mathfrak{g}^M$ and $g, h \in U(\mathfrak{g})^M$. Then*

$$\begin{aligned} f[gh] &= f[g]h + g(f[h]) \\ f[g[h]] &= (fg)[h] + f[g][h]. \end{aligned}$$

Hence $U(\mathfrak{g})^M$ is a D-algebra.

[2] The vector fields are interpreted as differential operators acting on functions.

[3] This is true when $\mathfrak{g}^M$ is replaced by Ξ^M for any vector space Ξ.

The composition of f and g as differential operators, defined by $(f \diamond g)[h] := f[g[h]]$, is called *non-frozen composition.*[4]

Post-Lie algebras. The theory of Lie–Butcher series can be reformulated in terms of *post-Lie algebras.* These were first studied in the setting of *operads* by Vallette [34], and also by the authors in [27]. Our main motivation for the construction of post-Lie algebras was their relation to the D-algebras defined above, which are universal enveloping algebras of post-Lie algebras.

2.2 Lie–Butcher series

Classical B-series. Recall (see e.g. [17]) that a B-series is a (formal) series indexed over the set NT of *non-planar* rooted trees (i.e. trees without any ordering of the branches) and can for a vector field f be written as

$$B_{h,f}(a)(y) = a(\mathbb{I})y + \sum_{\tau \in \mathrm{NT}} \frac{h^{|\tau|}}{\sigma(\tau)} a(\tau)\mathcal{F}_f(\tau)(y). \tag{7}$$

Here $\sigma(\tau)$ is the symmetry factor for $\tau \in \mathrm{NT}$, and a is a map $a : \mathrm{NT} \to \mathbb{R}$. The map $\mathcal{F}_f(\tau) : \mathbb{R}^n \to \mathbb{R}^n$ is the *elementary differential* of the tree τ, obtained recursively from f and its derivatives:

$$\mathcal{F}_f(\bullet)(y) = f(y), \qquad \mathcal{F}_f(\tau)(y) = f^{(m)}(y)(\mathcal{F}_f(\tau_1)(y), ..., \mathcal{F}_f(\tau_m)(y)), \tag{8}$$

where $\tau = B^+(\tau_1, \ldots \tau_m)$ and $f^{(m)}$ is the mth derivative of the vector field. The parameter h represents the step-size of the numerical method giving rise to the B-series.

LB-series. We will consider a more general setting: that of differential equations evolving on manifolds. Let M be a manifold and $\mathcal{X}M$ the Lie algebra of vector fields $F : M \to TM$ on M. The fundamental assumption for numerical Lie group integrators is the existence of a *frame* on TM, defined as a finite number of vector fields $\{E_1, E_2, \ldots, E_m\}$ spanning the tangent space T_pM at each point $p \in M$. The frame is allowed to be overdetermined. It generates a Lie algebra $\mathfrak{g}$, and it is assumed that flows of vector fields in $\mathfrak{g}$ can be computed exactly [25, 31]. Any vector field $F : M \to TM$ can be written as $F(p) = \sum a_i E_i(p)$. We will study vector fields of the form $F(p) = \sum f_i(p)E_i(p)$ where $f_i : M \to \mathbb{R}$ are smooth functions. Given such a vector field, let $f \in \mathfrak{g}^M$ be defined as $f_p = \sum f_i(p)E_i$. We say that f_p has coefficients frozen relative to the frame. In other words, to each such $F \in \mathcal{X}M$ there is an associated $f \in \mathfrak{g}^M$ so that $F(p) = f_p \cdot p$, where $f_p \cdot p$ denotes evaluation of f_p in p. We will often refer to such $f \in \mathfrak{g}^M$ as vector fields.

The general differential equation (1) can now be written as

$$y' = f_y \cdot y, \qquad \text{where } f \in \mathfrak{g}^M. \tag{9}$$

The Lie–Butcher series are expansions over OT associated to integrators of this equation, just as B-series are associated to differential equations expressing the flow of vector fields in $\mathbb{R}^n$. The non-commutativity of combining vector fields is reflected in the planarity of the trees in OT.

Now we can construct the elementary differentials needed to define Lie–Butcher series. As in the classical case they can be expressed recursively by a function $\mathcal{F}$ based on trees.

Definition 2.7. The elementary differentials associated to a vector field $f : M \to \mathfrak{g}$ is the D-algebra morphism $\mathcal{F}_f : \mathcal{N} \to U(\mathfrak{g})^M$ we get from Proposition 2.5 by associating the tree $\bullet$ to f (i.e. $\mathcal{C} = \{\bullet\}$ and $\nu : \bullet \mapsto f$ in Proposition 2.5). Hence $\mathcal{F}_f$ is defined by

(i) $\mathcal{F}_f(\mathbb{I}) = \mathbb{I}$

(ii) $\mathcal{F}_f(B^+(\omega)) = \mathcal{F}_f(\omega)[f]$

[4] We note that the two operations $f, g \mapsto f[g]$ and $f, g \mapsto f \diamond g$ gives $U(\mathfrak{g})^M$ the structure of a unital dipterous algebra (as defined in [21]).

(iii) $\mathcal{F}_f(\omega_1\omega_2) = \mathcal{F}_f(\omega_1)\mathcal{F}_f(\omega_2)$

When the vector field f is clear from the context we will occasionally write $\mathcal{F}$ instead of $\mathcal{F}_f$.

Definition 2.8. For an infinite series $\alpha \in \mathcal{N}^* = \mathbb{R}\langle\langle \mathrm{OT}\rangle\rangle$ a *Lie–Butcher series* is a formal series in $U(\mathfrak{g})^M$ defined as

$$\mathcal{B}_f(\alpha) = \sum_{\omega \in \mathrm{OF}} \alpha(\omega)\mathcal{F}(\omega).$$

For a vector field f this can also be written as the commutative diagram

$$\begin{tikzcd}
\{\bullet\} \arrow[r, hook] \arrow[d, "f"'] & \mathcal{N}^* \arrow[d, "\mathcal{B}_f"] \\
\mathfrak{g}^M \arrow[r, hook] & U(\mathfrak{g})^M
\end{tikzcd}$$

where $\mathcal{B}_f$ is the unique D-algebra homomorphism given by Proposition 2.5.

Remark 2.9. By coloring the vertices of the trees vi a map ν we can define $\mathcal{F}$ and $\mathcal{B}$ for multiple vector fields. The elementary differentials $\mathcal{F}_\nu$ are still obtained from Proposition 2.5, but the set $\mathcal{C}$ will contain multiple colors.

2.3 Some algebraic constructions

Before we show how LB-series can be used to represent flows of vector fields on manifolds we must conduct a closer study of the space where the coefficients α live. To understand the various ways we can represent such flows it will also be helpful to look at some Lie idempotents, namely the eulerian and Dynkin idempotents (Section 2.3.2), and also certain non-commutative polynomials called Bell polynomials (Section 2.3.3). We will follow the presentation in [28] and [22].

2.3.1 The Hopf algebras $\mathcal{H}_{Sh}$ and $\mathcal{H}_N$

It is well known that inserting a B-series $B_{h,f}(a)$ into another series $B_{h,f}(b)$ results in a B-series $B_{h,f}(a)(B_{h,f}(b)(y)) = B_{h,f}(a \cdot b)(y)$. The product $a \cdot b$ on the set of maps $a : \mathrm{OT} \to \mathbb{R}$ with $a(\mathbb{I}) = 1$ gives rise to a group, called the *Butcher group* [3, 18]. This is the group of characters in a variant of the Connes–Kreimer Hopf algebra of renormalization [11, 2]. A similar result holds for LB-series, where the Hopf algebra of Connes–Kreimer is replaced by a more general Hopf algebraic structure on the set of rooted trees. This Hopf algebra was introduced in [28]. See also [22, 23].

Note first that the vector space $\mathbb{R}\langle \mathrm{OT}\rangle$ spanned by trees can be turned into a Hopf algebra by using concatenation as product and *deshuffling* as coproduct. The deshuffling coproduct Δ_{Sh} is results from requiring the trees to be primitive, and extending by concatenation:

$$\Delta_{Sh}(\tau) = \tau \otimes \mathbb{I} + \mathbb{I} \otimes \tau, \qquad \Delta_{Sh}(\tau_1\tau_2) = \Delta_{Sh}(\tau_1)\Delta_{Sh}(\tau_2),$$

where τ, τ_1, τ_2 are trees. The antipode is defined by

$$S(\tau_1\tau_2\cdots\tau_n) = (-1)^n\tau_n\tau_{n-1}\cdots\tau_1,$$

and the unit η and counit ϵ by $\eta(1) = \mathbb{I}$, and $\epsilon(\mathbb{I}) = 1$, $\epsilon(\omega) = 0$, for all $\omega \in \mathrm{OF}\setminus\mathbb{I}$.

The vector space $\mathcal{N} = \mathbb{R}\langle \mathrm{OT}\rangle$ can also be turned into an algebra using the *shuffle product* $\sqcup\!\sqcup : \mathcal{N} \otimes \mathcal{N} \to \mathcal{N}$ defined recursively by $\mathbb{I} \sqcup\!\sqcup \omega = \omega = \omega \sqcup\!\sqcup \mathbb{I}$ and

$$(\tau_1\omega_1) \sqcup\!\sqcup (\tau_2\omega_2) = \tau_1(\omega_1 \sqcup\!\sqcup \tau_2\omega_2) + \tau_2(\tau_1\omega_1 \sqcup\!\sqcup \omega_2) \tag{10}$$

for $\tau_1, \tau_2 \in \mathrm{OT}$, $\omega_1, \omega_2 \in \mathrm{OF}$.[5] This algebra can be given the structure of a bialgebra in several different ways. We can equip it with the coproduct $\Delta_c : \mathcal{N} \to \mathcal{N} \otimes \mathcal{N}$ given by *deconcatenation* of words:

$$\Delta_c(w) = \mathbb{I} \otimes w + w \otimes \mathbb{I} + \sum_{i=1}^{n-1} \tau_1\cdots\tau_i \otimes \tau_{i+1}\cdots\tau_n, \tag{11}$$

[5] Coproducts will occasionally be written using the *Sweedler* notation $\Delta(\omega) = \sum \omega_{(1)} \otimes \omega_{(2)}$.

where $\omega = \tau_1 \cdots \tau_n$. This results in the shuffle bialgebra, which equipped with the same antipode, unit and counit as the deshuffle Hopf algebra defines the *shuffle Hopf algebra* $\mathcal{H}_{Sh}$.[6]

If we instead equip $\mathcal{N}$ with the coproduct $\Delta_N : \mathcal{N} \to \mathcal{N} \otimes \mathcal{N}$ defined recursively as $\Delta_N(\mathbb{I}) = \mathbb{I} \otimes \mathbb{I}$ and

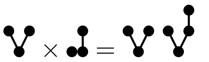
$$\Delta_N(\omega\tau) = \omega\tau \otimes \mathbb{I} + \Delta_N(\omega) \sqcup\!\sqcup \cdot (I \otimes B^+)\Delta_N(B^-(\tau)), \tag{12}$$

where $\tau \in \mathrm{OT}$, $\omega \in \mathrm{OF}$, we get another bialgebra $\mathcal{H}_N$. Here $\sqcup\!\sqcup \cdot : \mathcal{N}^{\otimes 4} \to \mathcal{N} \otimes \mathcal{N}$ denotes shuffle on the left and concatenation on the right: $(\omega_1 \otimes \omega_2) \sqcup\!\sqcup \cdot (\omega_3 \otimes \omega_4) = (\omega_1 \sqcup\!\sqcup \omega_3) \otimes (\omega_2\omega_4)$. An explicit description of the coproduct in terms of tree cuts can be found in Section 3.1 below, and in [28], where it was shown that Δ_N is the dual of the Grossman-Larson product and that $\mathcal{H}_N$ forms a Hopf algebra.[7] This is the Hopf algebra governing composition of LB-series (Theorem 2.10).

To simplify the expressions we introduce a *magmatic* structure (i.e. the structure of a set equipped with a closed binary operation with no further relations) on OF. Additional details and motivation for the introduction of this structure can be found in Section 4. Let ω_1, ω_2 be two elements of OF and define the operation $\times : \mathrm{OF} \times \mathrm{OF} \to \mathrm{OF}$ by

$$\omega_1 \times \omega_2 = \omega_1 B^+(\omega_2). \tag{13}$$

For example,



This operation is magmatic, and the empty tree $\mathbb{I}$ freely generates all of OF. The operation is extended to $\mathcal{N} = \mathbb{R}\langle \mathrm{OT} \rangle$ via linearity. If $\omega = v_1 \times v_2$, $\omega \neq \mathbb{I}$, then we call v_1 the *left part* ω_L of ω and v_2 the *right part* ω_R. The shuffle of two elements of the magma can be defined as

$$v \sqcup\!\sqcup \omega = (v \sqcup\!\sqcup \omega_L) \times v_R + (v_L \sqcup\!\sqcup \omega) \times \omega_R, \tag{14}$$

$\omega \sqcup\!\sqcup \mathbb{I} = \mathbb{I} \sqcup\!\sqcup \omega = \omega$, and we notice that the coproduct in $\mathcal{H}_N$ can be written as

$$\Delta_N(\omega) = \omega \otimes \mathbb{I} + \Delta_N(\omega_L) \sqcup\!\sqcup \times \Delta_N(\omega_R), \tag{15}$$

where $\sqcup\!\sqcup\times$ now denotes shuffle on the left, magma operation on the right.

Characters and the composition of LB-series. Recall that a *character* of a Hopf algebra $(H, \Delta, \cdot)$ over a field k is an algebra morphism $\alpha : H \to \mathrm{k}$, e.g. $\alpha(a \cdot b) = \alpha(a)\alpha(b)$, and $\alpha(1_H) = 1$, where $a, b \in H$, and 1_H denotes the unit. The convolution product $\alpha * \beta$ of two characters is defined by

$$H \xrightarrow{\Delta} H \otimes H \xrightarrow{\alpha\otimes\beta} \mathrm{k} \otimes \mathrm{k} \to \mathrm{k}\,.$$

This gives the set of characters $G(H)$ of H the structure of a group. In fact, the field k can be replaced by any commutative algebra A, giving rise to *A-valued characters*. Another type of character we will need later are the *infinitesimal characters*. An A-valued infinitesimal character is a linear map $\alpha : H \to A$ satisfying

$$\alpha(h \cdot h') = \mu_A(\alpha(h), \delta(h')) + \mu_A(\delta(h), \alpha(h')), \tag{16}$$

where μ_A is the product in A and δ is the composition of the counit of H and the unit of A, $\delta = \eta_A \circ \epsilon$. The characters and infinitesimal characters are connected via the exponential and logarithm, see e.g. [24].

The group structure of the characters in $\mathcal{H}_N$ exactly corresponds to the composition of LB-series.

Theorem 2.10 ([28])**.** *The composition of two LB-series is again a LB-series:*

$$\mathcal{B}_f(\alpha)[\mathcal{B}_f(\beta)] = \mathcal{B}_f(\alpha * \beta),$$

where $*$ is the convolution product in $\mathcal{H}_N$.

[6] Note that the concatenation deshuffling Hopf algebra is dual to the shuffle deconcatenation Hopf algebra.

[7] Being a graded and connected bialgebra $\mathcal{H}_N$ is automatically a Hopf algebra. A more direct argument, and formulas for the antipode, can be found in [28]

2.3.2 Lie idempotents

A *Lie polynomial* over an algebra A is an element of the smallest submodule of $\mathbb{R}\langle A\rangle$ that is closed under the bracket $[P,Q] := PQ - QP$ in $\mathbb{R}\langle A\rangle$. The Lie algebra of these polynomials is the free Lie algebra $\mathrm{Lie}(A)$ on A [33]. There are several important idempotent maps, called *Lie idempotents*, from $\mathbb{R}\langle A\rangle$ to $\mathrm{Lie}(A)$.

Eulerian idempotent Let H be a commutative, connected and graded Hopf algebra. Consider $\mathrm{End}_k(H) = \mathrm{Hom}_k(H,H)$ equipped with the convolution product $*$. Let $\mathrm{Id} \in \mathrm{End}_k(H)$ be the identity endomorphism and $\delta = \eta \circ \epsilon \in \mathrm{End}_k(H)$ the unit of convolution.

Definition 2.11 ([20])**.** The Eulerian idempotent $e \in \mathrm{End}(H)$ is given by the formal power series

$$e := \log^*(\mathrm{Id}) = J - \frac{J^{*2}}{2} + \frac{J^{*3}}{3} + \cdots (-1)^{i+1}\frac{J^{*i}}{i} + \cdots ,$$

where $J = \mathrm{Id} - \delta$.

Proposition 2.12 ([20])**.** *For any commutative graded Hopf algebra H, the element $e \in \mathrm{End}_k(H)$ defined above is a Lie idempotent in H. That is, $e \circ e = e$ and it has image in the free Lie algebra.*

The practical importance of the Eulerian idempotent in numerical analysis arises in the study of backward error analysis, where the following lemma provides a computational formula for the logarithm:

Proposition 2.13 ([22])**.** *For $\alpha \in G(H)$ and $h \in H$, we have*

$$\log^*(\alpha)(h) = \alpha(e(h)).$$

In other words, the logarithm can be written as right composition with the eulerian idempotent:

$$\log^* = \_ \circ e : G(H) \to \mathfrak{g}(H).$$

Dynkin idempotent The classical *Dynkin operator* on the shuffle Hopf algebra is given by left-to-right bracketing:

$$D(a_1 ... a_n) = [\ldots[[a_1,a_2],a_3],\ldots,a_n], \quad \text{where } [a_i,a_j] = a_i a_j - a_j a_i.$$

Letting $Y(\omega) = \#(\omega)\omega$ denote the grading operator, where $\#(\omega)$ is word length, it is known that the *Dynkin idempotent* $Y^{-1} \circ D$ is an idempotent projection on $\mathrm{Lie}(A)$. As in [12], the Dynkin operator can be written as the convolution of the antipode S and the grading operator Y: $D = S * Y$. This description can be generalized to any graded, connected and commutative Hopf algebra H:

Definition 2.14. Let H be a graded, commutative and connected Hopf algebra with grading operator $Y : H \to H$. The *Dynkin operator* is the map $D : H \to H$ given as

$$D := S * Y.$$

2.3.3 The non-commutative Bell polynomials

In [26] some non-commutative polynomials B_n were introduced to express the Butcher order theory of Runge-Kutta methods on manifolds. In [22] it was observed that these polynomials were a non-commutative analogue of Bell polynomials, and that they could be used to study more general flows on manifolds. We recall their definition here.

Let $\mathcal{I} = \{d_j\}_{j=1}^{\infty}$ be an infinite alphabet in 1–1 correspondence with $\mathbb{N}^+$, and consider the free associative algebra $\mathcal{D} = \mathbb{R}\langle\mathcal{I}\rangle$ with the grading given by $|d_j| = j$ and $|d_{j_1}\cdots d_{j_k}| = j_1 + \cdots + j_k$. Let $\partial\colon \mathcal{D} \to \mathcal{D}$ be the derivation given by $\partial(d_i) = d_{i+1}$, linearity and the Leibniz rule $\partial(\omega_1\omega_2) = \partial(\omega_1)\omega_2 + \omega_1\partial(\omega_2)$ for all $\omega_1, \omega_2 \in \mathcal{I}^*$. We let $\#(\omega)$ denote the length of the word ω.

Definition 2.15. The non-commutative Bell polynomials $B_n := B_n(d_1, \ldots, d_n) \in \mathcal{D}$ are defined by the recursion

$$\begin{aligned} B_0 &= \mathbb{I} \\ B_n &= (d_1 + \partial)B_{n-1} = (d_1 + \partial)^n \mathbb{I} \quad \text{for } n > 0. \end{aligned}$$

The first few are

$$\begin{aligned} B_0 &= \mathbb{I} \\ B_1 &= d_1 \\ B_2 &= d_1^2 + d_2 \\ B_3 &= d_1^3 + 2d_1d_2 + d_2d_1 + d_3 \\ B_4 &= d_1^4 + 3d_1^2d_2 + 2d_1d_2d_1 + d_2d_1^2 + 3d_1d_3 + d_3d_1 + 3d_2d_2 + d_4. \end{aligned}$$

We write $B_{n,k} := B_{n,k}(d_1, \ldots, d_{n-k+1})$ for the part of B_n consisting of the words of length k, e.g. $B_{4,3} = 3d_1^2d_2 + 2d_1d_2d_1 + d_2d_1^2$. It is often useful to employ the polynomials Q_n and $Q_{n,k}$ related to B_n and $B_{n,k}$ by the following rescaling:

$$\begin{aligned} Q_{n,k}(d_1, \ldots, d_{n-k+1}) &= \frac{1}{n!}B_{n,k}(1!d_1, \ldots, j!d_j, \ldots) = \sum_{|\omega|=n, \#(\omega)=k} \kappa(\omega)\omega && (17) \\ Q_n(d_1, \ldots, d_n) &= \sum_{k=1}^{n} Q_{n,k}(d_1, \ldots, d_{n-k+1}) && (18) \\ Q_0 &:= \mathbb{I}. && (19) \end{aligned}$$

These polynomials can be used to define an operator Q on any graded Hopf algebra H. Let d_i be defined on H^* by

$$d_i(\alpha) = \alpha_i = \alpha|_{H_i}, \qquad d_id_j(\alpha) = \alpha_i * \alpha_j, \tag{20}$$

where α_i is the degree i component of α and $*$ is the convolution product. The operator Q is a bijection from infinitesimal characters to characters of H (for details, see [22]).

2.4 Lie–Butcher series and flows of vector fields

Flows $y_0 \mapsto y(t) = \Psi_t(y_0)$ on the manifold M can be represented by LB-series in several different ways. Here are three procedures, giving rise to what we will call LB-series of Type 1, 2 and 3:

1. In terms of pullback series: Find $\alpha \in G(\mathcal{H}_N)$ such that

   $$\Psi(y(t)) = \mathcal{B}(\alpha)(y_0)[\Psi] \quad \text{for any } \Psi \in U(\mathfrak{g})^M. \tag{21}$$

   This representation is used in the analysis of Crouch–Grossman methods by Owren and Marthinsen [31]. In the classical setting this is called a S-series [29].

2. In terms of an autonomous differential equation: Find $\beta \in \mathfrak{g}(\mathcal{H}_N)$ such that $y(t)$ solves

   $$y'(t) = \mathcal{B}(\beta)(y(t)). \tag{22}$$

   This is called backward error analysis (confer Section 3.3).

3. In terms of a non-autonomous equation of *Lie type* (time dependent frozen vector field): Find $\gamma \in \mathfrak{g}(\mathcal{H}_{\mathrm{Sh}})$ such that $y(t)$ solves

   $$y'(t) = \left(\frac{\partial}{\partial t}\mathcal{B}(\gamma)(y_0)\right) y(t). \tag{23}$$

   This representation is used in [25, 26]. In the classical setting this is (close to) the standard definition of B-series.

The algebraic relationships between the coefficients α, β and γ in the above LB-series are [22]:

$\beta = \alpha \circ e$	e is Euler idempotent in $\mathcal{H}_N$. (Proposition 2.13)
$\alpha = \exp^{\diamond}(\beta)$	Exponential wrt. GL-product
$\gamma = \alpha \circ Y^{-1} \circ D$	Dynkin idempotent in $\mathcal{H}_{\mathrm{Sh}}$. [22, Proposition 4.4]
$\alpha = Q(\gamma)$	Q-operator (20) in $\mathcal{H}_{\mathrm{Sh}}$. [22, Proposition 4.9]

By using these relationships one can convert between the various representations of flows.

In the notation in the following examples of LB-series we suppress the vector fields and elementary differentials, and phrase the LB-series in terms of the (dual of the) coefficient functions.

Example 2.16 (The exact solution)**.** The exact solution of a differential equation

$$y'(t) = F(y(t))$$

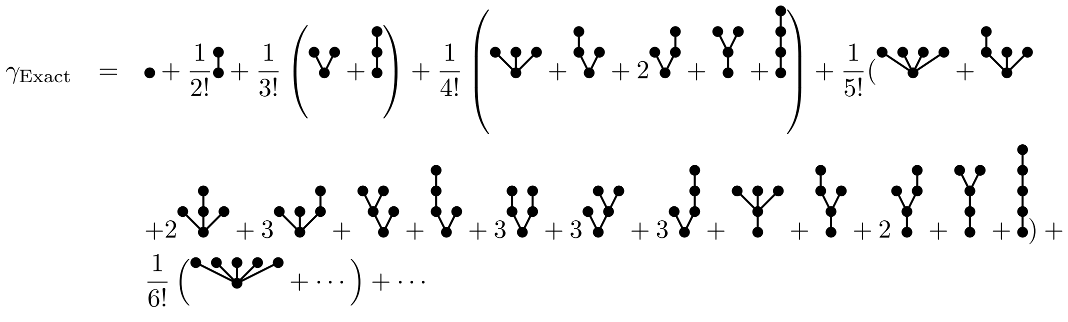
can be written as the solution of

$$y' = F_t \cdot y, \quad y(0) = y_0,$$

where $F_t = F(y(t)) \in \mathfrak{g}$ is the pullback of F along the time dependent flow of F. Let $F_t = \frac{\partial}{\partial t}\mathcal{B}_t(\gamma)$. By [22, Proposition 4.9] the pullback is given by $\mathcal{B}_t(Q(\gamma_{\mathrm{Exact}}))[F]$, so

$$Y \circ \gamma_{\mathrm{Exact}} = Q(\gamma_{\mathrm{Exact}})[\bullet] \Rightarrow \gamma_{\mathrm{Exact}} = Y^{-1} \circ B^{+}(Q(\gamma_{\mathrm{Exact}})).$$

Note that this is reminiscent of a so-called combinatorial Dyson–Schwinger equation [15]. Solving by iteration yields



Note that a formula for the LB-series for the exact solution was given in [31]. We observe that there cannot be any commutators of trees in this expression. Therefore, in LB-series of numerical integrators, commutators of trees must be zero up to the order of the method.

Example 2.17 (The exponential Euler method)**.** The exponential Euler method [19] can be written as follows:

$$y_{n+1} = \exp(hf(y_n))y_n,$$

or, by rescaling the vector field f, as

$$y_{n+1} = \exp(f(y_n))y_n.$$

This equation can be interpreted as a pullback equation of the form $\Phi(y_{n+1}) = \mathcal{B}(\exp(\bullet))[\Phi]y_n$, so

$$\alpha = \exp(\bullet) = \mathbb{I} + \bullet + \frac{1}{2!}\bullet\bullet + \frac{1}{3!}\bullet\bullet\bullet + \cdots .$$

(Here the Grossman-Larson product is the same as concatenation). Note that $\exp(\bullet) = Q(\bullet)$, so the Type 3 LB-series for the Euler method is simply

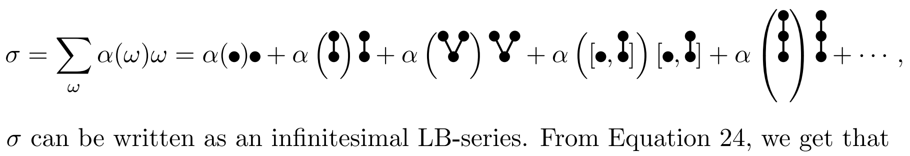
$$\gamma_{\mathrm{Euler}} = \bullet.$$

Example 2.18 (The implicit midpoint method)**.** The implicit midpoint method [19] can be presented as:

$$\sigma = f(\exp(\frac{1}{2}\sigma)y_n) \tag{24}$$

$$y_{n+1} = \exp(\sigma)y_n \tag{25}$$

We make the following ansatz:



σ = Σ α(ω)ω = α(•)• + α ( ) + α ( ) + α ([•,]) [•,] + α ( ) + · · · ,
ω
(26)

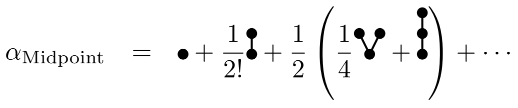
i.e. that σ can be written as an infinitesimal LB-series. From Equation 24, we get that

$$\sigma = \sum_{j=0}^{\infty} \frac{(\sigma)^j}{2^j j!}[\bullet]. \tag{27}$$

Since there are no forests in this expression, we must have $\alpha([\omega, \omega']) = 0$ for all $\omega, \omega' \in \mathrm{OT}$. If we write $\tau = B^+(\tau_1 \cdots \tau_j)$, then by combining Equation 27 with the ansatz, we see that coefficients of the LB-series are given recursively as $\alpha(\bullet) = \frac{1}{2}$,

$$\alpha(\tau) = \frac{1}{2^j j!}\alpha(\tau_1)\cdots\alpha(\tau_j). \tag{28}$$

Hence



3 Substitution law for Lie–Butcher-series

In this section we will generalize the substitution law for B-series [7] to LB-series. Once the substitution law has been established we will apply it to backward error analysis for numerical methods based on LB-series.

3.1 The substitution law

Consider $\mathcal{N}$ as a D-algebra where the derivations are the Lie polynomials $D(\mathcal{N}) = \mathfrak{g}(\mathcal{H}_N) \cap \mathcal{N}$. By the universal property of $\mathcal{N}$, we know that for any map $a \colon \mathcal{C} \to D(\mathcal{N})$ there exists a unique D-algebra homomorphism $\mathcal{F}_a : \mathcal{N} \to \mathcal{N}$ such that $\mathcal{F}_a(c) = a(c)$ for all $a \in \mathcal{C}$. This is called the substitution law.

Definition 3.1. For any map $a : \mathcal{C} \to D(\mathcal{N})$ the unique D-algebra homomorphism $a\star : \mathcal{N} \to \mathcal{N}$ such that $a(c) = a \star c$ for all $c \in \mathcal{C}$ is called a-substitution[8].

$$\begin{tikzcd}
\mathcal{C} \arrow[r, hook] \arrow[d, "a"'] & \mathcal{N} \arrow[d, "a\star"] \\
D(\mathcal{N}) \arrow[r, hook] & \mathcal{N}.
\end{tikzcd}$$

Theorem 3.2. *The substitution law defined in Definition 3.1 corresponds to the substitution of B-series in the sense that*

$$\mathcal{B}_{\mathcal{B}_f(\beta)}(\alpha) = \mathcal{B}_f(\beta \star \alpha)$$

[8] In most applications we want to substitute infinite series and extend $a\star$ to a homomorphism $a\star \colon \mathcal{N}^* \to \mathcal{N}^*$. The extension to infinite substitution is straightforward because of the grading, we omit details. We write $a\star$ also for infinite substitution.

The theorem is easily proven by using the following lemma:

Lemma 3.3. *For all $\beta : \{\bullet\} \to D(\mathcal{N}^*)$ and all B-series $B_f : \mathcal{N}^* \to U(\mathfrak{g})^M$, the composition $B_f \circ \beta$ has image in $\mathfrak{g}^M$. In other words, B-series maps $D(\mathcal{N})$ to derivations on M.*

Proof. It is enough to prove this for Lie polynomials [9]. Since B_f is a D-algebra homomorphism it maps trees to derivations, so the only thing we have to check is that the commutator $[V, W] = VW - WV$ of two derivations V and W is a derivation. This is a straightforward calculation. □

Proof of Theorem 3.2. Except for the use of Lemma 3.3, the proof is purely categorical. Let B_f be a B-series. The composition of B_f with the map $\beta\star$ can be written in diagrammatic form as

$$\begin{tikzcd}
\{\bullet\} \arrow[rr, hook] \arrow[dr, "\beta"] & & \mathcal{N}^* \arrow[dd, "\beta\star"] \\
& D(\mathcal{N}) \arrow[dr, hook] & \\
& & \mathcal{N}^* \arrow[d, "B_f"] \\
& & U(\mathfrak{g})^M
\end{tikzcd}$$

By Lemma 3.3 the composition of the two diagonal arrows and B_f actually has image in $\mathfrak{g}^M$. Therefore the universal property for the diagram obtained by adding the map $\mathcal{B}_f \circ \beta : \{\bullet\} \to \mathfrak{g}^M$ to the above diagram shows that $\mathcal{B}_f \circ \beta\star = \mathcal{B}_{\mathcal{B}_f \circ \beta}$, and hence the theorem. □

Many of the useful properties of the substitution law follow immediately from the fact that $a\star$ is a D-algebra homomorphism. For example, $a\star : \mathcal{N} \to \mathcal{N}$ is a linear map which for any $n, n' \in \mathcal{N}$ satisfies

$$\begin{aligned}
a \star \mathbb{I} &= \mathbb{I} \\
a \star (nn') &= (a \star n)(a \star n') \\
a \star (n \curvearrowright n') &= (a \star n) \curvearrowright (a \star n') \\
a \star (n \diamond n') &= (a \star n) \diamond (a \star n') \\
a \star (n \circ S) &= (a \star n) \circ S \\
a \star (n \circ e) &= (a \star n) \circ e
\end{aligned}$$

where S is the antipode and e is Euler map in $\mathcal{H}_N$.

The free D-algebra $\mathcal{N}$ is the universal enveloping algebra of the free *post-Lie algebra* $\mathfrak{g}$ of rooted trees [27]. By defining a coproduct by requiring that the elements of $\mathfrak{g}$ are primitive (e.g. the deshuffle coproduct of Section 2.3.1), it is a bialgebra. The unique D-algebra morphism $a\star$ is a coalgebra morphism for this coproduct:

Lemma 3.4. *The map $a\star$ is a coalgebra morphism with respect to the coproduct given by deshuffling of words (Section 2.3.1). That is,*

$$(a \star \otimes a\star) \circ \Delta_{Sh} = \Delta_{Sh} \circ a\star,$$

where Δ_{Sh} denotes the deshuffling coproduct.

Proof. The result is easily proven for primitive elements. The general case follows by induction on the length of words. □

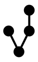
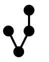
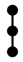
[9] Lie series are formal series whose homogeneous components are Lie polynomials [33]

Remark 3.5 (The Hopf algebra for the substitution law)**.** Based on the results in [5] and the fact that the operad governing post-Lie algebras is known, it is possible to describe the Hopf algebra for the substitution law following the program in [5]. This is a project currently under development [13].

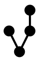
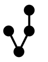
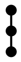
3.2 A formula for the substitution law

The substitution law can be calculated recursively using a formula involving trees. To write down the formula we need to look at cutting operations on trees and forests.

Cutting trees and forest. Let $\tau \in \mathrm{OT}$ be an ordered rooted tree. An *elementary left cut* c of τ is a choice of a set of branches E of τ to be removed from τ. These are chosen in a systematic manner: if an edge e is in E then all the branches on the same level and to the left of e must also be in E. Each cut splits τ into two components: the pruned part $P^c_{el}(\tau)$ consisting of the trees that were cut off concatenated together, and the remaining part $R^c_{el}(\tau)$ consisting of the tree containing the root. We also consider the *empty cut*, i.e. the cut c so that $P^c_{el}(\tau) = \mathbb{I}$ and $R^c_{el}(\tau) = \tau$, to be an elementary cut.

$\tau =$ 	$P^c_{el}(\tau)$	$R^c_{el}(\tau)$
	$\mathbb{I}$	
	$\bullet$	
		$\bullet$
	$\bullet$	

A *left admissible cut* on τ consists of a collection of elementary cuts applied to τ with the property that any path from the root to any vertex of τ crosses at most one elementary cut. The pruned parts corresponding to each elementary cut are shuffled together, with no internal shuffling of the trees resulting from each elementary cut. An admissible cut of a tree results in a collection of shuffles of forests $P^c(\tau)$ and a tree $R^c(\tau)$. The collection of all left admissible cuts for a tree τ is written as $LAC(\tau)$.

$\tau =$ 	$P^c(\tau)$	$R^c(\tau)$
	$\mathbb{I}$	
	$\bullet$	
		$\bullet$
	$\bullet$	
	$\bullet \sqcup \bullet$	

We extend these cutting operations to forests $\omega \in \mathrm{OF}$ by applying the B^+ operator to ω and then cut it as a tree without using cuts of branches growing out of the root, before finally applying the B^- operator to $R^c(\omega)$ to remove the added root.

The coproduct Δ_N of the Hopf algebra $\mathcal{H}_N$ (Section 2.3) can be formulated in terms of these cuts [28]. First one must extend the left admissible cuts to include the *full* cut of a tree, which cuts "below" the root, so that $P^c_{el}(\tau)$ is again τ. The set of all left admissible cuts, including the full cut, is denoted by FLAC, and the coproduct Δ_N can be written as:

$$\Delta_N(\omega) = \sum_{c \in FLAC} P^c(\omega) \otimes R^c(\omega). \tag{29}$$

Table 1 gives the result of this coproduct applied to all forests up to order 4. If we let $\tilde{\Delta}_N(\omega)$ consist only of forests resulting from not using the empty nor the full cut, we get $\Delta_N(\omega) = 1 \otimes \omega + \omega \otimes 1 + \tilde{\Delta}_N(\omega)$. The operation $\tilde{\Delta}_N$ is called the *reduced coproduct.*

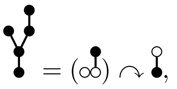
A formula for the substitution law. We will give a formula for the dual of the substitution, i.e. a formula for $a^T_\star$, where $\langle a \star b, \omega\rangle = \langle b, a^T_\star(\omega)\rangle$. The formula is based on the pruning operation $\mathcal{P}$ on forests.

Lemma 3.6 (Pruning)**.** *Let ω and ν be two forests. The dual of grafting, i.e. the operation defined by $\langle \nu \curvearrowright \omega', \omega\rangle = \langle \omega', \mathcal{P}_\nu(\omega)\rangle$, is given by:*

$$\mathcal{P}_\nu(\omega) = \sum_{c \in LAC(\omega)} \langle \nu, P^c(\omega)\rangle R^c(\omega).$$

The operation is called pruning.

Proof. An elementary cut at an edge growing out of a node n of a forest ω is the dual operation of attaching trees via edges to the node n in a certain order, e.g.



where the white nodes indicates where the attachment is done. The shuffling in $P^c(\omega)$ corresponds to the dual of attaching forests in all possible ways to different nodes. Hence the dual of grafting is given by

$$\sum_{c \in LAC(\omega)} P^c(\omega) \otimes R^c(\omega).$$

□

Theorem 3.7. *We have*

$$a^T_\star(\omega) = \sum_{(\omega)\in\Delta_c} \sum_{c \in LAC(\omega_{(2)})} a^T_\star(\omega_{(1)}) B^+\left(a^T_\star(P^c(\omega_{(2)}))\right) a(R^c(\omega_{(2)})),$$

if $\omega \neq \mathbb{I}$, and $a^T_\star(\mathbb{I}) = \mathbb{I}$. Here Δ_c denotes deconcatenation (Section 2.3).

Note that using the magmatic product $\times$ defined in Section 2.3, this can also be written as:

$$a^T_\star = \mu \circ (\mu_\times \otimes I) \circ (a^T_\star \otimes a^T_\star \otimes a) \circ (I \otimes \Delta'_N) \circ \Delta_c, \tag{30}$$

where μ is concatenation, Δ_N is the coproduct in $\mathcal{H}_N$, and $\Delta'_N(\omega) = \Delta_N(\omega) - \omega \otimes \mathbb{I}$ for all forests ω.

Proof. We first prove the formula for ordered trees. Let π_{OT} denote the projection of forests onto trees: $\pi_{\mathrm{OT}}(\omega) = \sum_{\tau\in\mathrm{OT}}\langle\tau,\omega\rangle\omega$. Recall that

$$\langle a \star \omega', \omega\rangle = \langle \omega', a^T_\star(\omega)\rangle, \qquad \langle \nu \curvearrowright \omega', \omega\rangle = \langle \omega', \mathcal{P}_\nu(\omega)\rangle.$$

We have

$$\begin{aligned}
\pi_{\mathrm{OT}}(a_\star^T\omega) = \sum_{\tau\in\mathrm{OT}} \left\langle \tau, a_\star^T\omega\right\rangle \tau &= \sum_{\nu\in\mathrm{OF}} \left\langle \nu \curvearrowright \bullet, a_\star^T(\omega)\right\rangle \nu \curvearrowright \bullet \\
&= \sum_{\nu\in\mathrm{OF}} \langle a\star(\nu\curvearrowright\bullet),\omega\rangle\, \nu\curvearrowright\bullet \\
&= \sum_{\nu\in\mathrm{OF}} \langle (a\star\nu)\curvearrowright a,\omega\rangle\, \nu\curvearrowright\bullet \\
&= \sum_{\nu\in\mathrm{OF}} \langle a, \mathcal{P}_{a\star\nu}\omega\rangle\, \nu\curvearrowright\bullet \\
&= \sum_{c\in LAC(\omega)} \sum_{\nu\in\mathrm{OF}} \langle a\star\nu, P^c(\omega)\rangle \langle a, R^c(\omega)\rangle\, \nu\curvearrowright\bullet.
\end{aligned}$$

Hence,

$$\begin{aligned}
\pi_{\mathrm{OT}}(a_\star^T\omega) &= \sum_{c\in LAC(\omega)} \sum_{\nu\in\mathrm{OF}} \left\langle \nu, a_\star^T P^c(\omega)\right\rangle (\nu\curvearrowright\bullet) a(R^c(\omega)) \\
&= \sum_{c\in LAC(\omega)} \left((a_\star^T(P^c(\omega))\curvearrowright\bullet\right) a(R^c(\omega)) \\
&= \sum_{c\in LAC(\omega)} B^+\left(a_\star^T(P^c(\omega))\right) a(R^c(\omega)).
\end{aligned}$$

The general formula is established by the following calculation, where τ is a tree:

$$\begin{aligned}
a_\star^T(\omega) &= \sum_{\nu,\tau} \left\langle \nu\tau, a_\star^T(\omega)\right\rangle \nu\tau \\
&= \sum_{\nu,\tau} \langle (a_\star\nu)(a_\star\tau),\omega\rangle\, \nu\tau \\
&= \sum_{(\omega)\in\Delta_c} \sum_{\nu,\tau} \left\langle a_\star\nu, \omega_{(1)}\right\rangle \left\langle a_\star\tau, \omega_{(2)}\right\rangle \nu\tau \\
&= \sum_{(\omega)\in\Delta_c} (a_\star^T\omega_{(1)})(\pi_{\mathrm{OT}}(a_\star^T\omega_{(2)})).
\end{aligned}$$

□

As an example, the formula applied to the tree $B^+(\bullet\bullet)$ yields

$$a_\star^T(B^+(\bullet\bullet)) = a(B^+(\bullet\bullet))B^+(\mathbb{I}) + a(\bullet)a(B^+(\bullet))B^+(\bullet) = a(B^+(\bullet\bullet))\bullet + a(\bullet)a(B^+(\bullet))B^+(\bullet) + a(\bullet)^3 B^+(\bullet\bullet).$$

See Table 2 where this formula is computed for all forests up to order 4, under the assumption that a is an infinitesimal character.

Proposition 3.8. *The map $a_\star^T$ is a character for the shuffle product:*

$$a_\star^t(\omega_1 \sqcup\!\sqcup\, \omega_2) = a_\star^t(\omega_1) \sqcup\!\sqcup\, a_\star^t(\omega_2).$$

Proof. The shuffle product is dual to the deshuffle coproduct, so the result follows from Lemma 3.4 by dualization. □

Remark 3.9. There is a similar formula for the substitution of B-series, only with the coproduct Δ_N replaced by the Connes–Kreimer coproduct $\Delta_{CK} = \sum_{c\in AC(\tau)} P_{CK}^c(\tau)\otimes R_{CK}^c(\tau)$:

$$a_\star^T(\tau) = \sum_{c\in AC(\tau)} B^+\left(a_\star^T(P_{CK}^c(\tau))\right) a(R_{CK}^c(\tau)) \tag{31}$$

The proof of this formula is analogous to the proof of Theorem 3.7. This gives a recursive version of the coproduct in the substitution bialgebra H_{CEFM} of [5]

3.3 Backward error analysis and modified vector fields

Recall the results on backward error analysis in [7]: Given a B-series method $B_f(\alpha)$ there is a *modified vector field* $\tilde{f}$ so that the B-series method applied to f generates the exact flow of $\tilde{f}$. Moreover, $\tilde{f}$ can be written as a B-series with coefficients β satisfying $\beta \star \gamma_{\text{Exact}} = \alpha$, where γ_{Exact} is the coefficient function for the B-series of the exact flow, and $\star$ is the substitution law for characters in the Connes–Kreimer Hopf algebra H_{CK}. To generalize to LB-series, consider a numerical solution of the differential equation

$$y' = f(y) \cdot y \tag{32}$$

written in terms of a LB-series $\mathcal{B}_f(\alpha)$. We interpret it as the exact solution of a modified differential equation $y' = \tilde{f}(y) \cdot y$. As in the classical case, it turns out that the modified vector field can be written as a LB-series $\tilde{f} = \mathcal{B}_f(\beta)$. Furthermore, $\tilde{f}$ is such that

$$\mathcal{B}_{\mathcal{B}_f(\beta)}(\gamma_{\text{Exact}}) = \mathcal{B}_f(\alpha), \tag{33}$$

where γ_{Exact} represents the coefficients of the exact solution as described in Section 2.4. This result follows by applying Proposition 3.2.

Theorem 3.10. *Let $\mathcal{B}_f(\alpha)$ be a LB-series method. There is a modified vector field $\tilde{f}$, given by $\tilde{f} = \mathcal{B}_f(\beta)$ such that*

$$\mathcal{B}_{\tilde{f}}(\gamma_{Exact}) = \mathcal{B}_f(\beta).$$

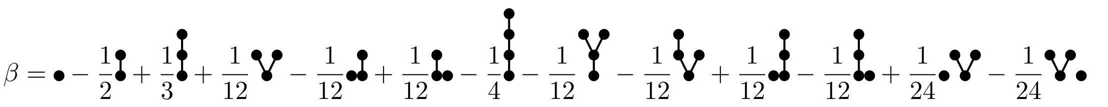
Moreover,

$$\beta * \gamma_{Exact} = \alpha.$$

Example 3.11 (The exponential Euler method)**.** The exponential Euler method is given by

$$y_{n+1} = \exp(hf(y_n))y_n.$$

In Example 2.17 the coefficients of the LB-series for this method was seen to be $\gamma = \bullet$. To get the backward error, we calculate $\beta = Q(\bullet) \circ e$, or $\log^*(Q(\bullet))$ (cf. Section 2.4)



In the classical setting this logarithm has been studied as $\log^*(\delta)$, for a certain character δ [6, 30].

4 Implementation

As pointed out in Section 2.3, the set of forests $\mathcal{F}$ can be generated recursively using a *magmatic product* $\times$ defined on two forests ω_1 and ω_2 by

$$\omega_1 \times \omega_2 = \omega_1 B^+(\omega_2) \tag{34}$$

by starting with the empty tree $\mathbb{I}$. Each forest in $\mathcal{F}$ can uniquely be written as a word in $\mathbb{I}$ and $\times$. Recall that if $\omega = \omega_1 \times \omega_2$, then we call ω_1 the *left part*, ω_L, and ω_2 the *right part*, ω_R, of ω. All the basic algebraic operations used to construct the substitution law can be formulated in terms of this product:

Concatenation: $\omega\,\mathbb{I} = \mathbb{I}\,\omega = \omega$, and $(\omega_1 \times \omega_2)\,\omega_3 = \omega_1 \times (\omega_2\,\omega_3)$.

Shuffle: $\omega \sqcup\!\sqcup \mathbb{I} = \mathbb{I} \sqcup\!\sqcup \omega = \omega$, and $\omega_1 \sqcup\!\sqcup \omega_2 = (\omega_1 \sqcup\!\sqcup \omega_{2L}) \times \omega_{1R} + (\omega_{1L} \sqcup\!\sqcup \omega_2) \times \omega_{1R}$

Coproduct: $\Delta_N(\mathbb{I}) = \mathbb{I} \otimes \mathbb{I}$, and $\Delta_N(\omega) = \omega \otimes \mathbb{I} + \Delta_N(\omega_L) \sqcup\!\sqcup \times \Delta_N(\omega_R)$

The formula (30) for the substitution law in Theorem 3.7 therefore lends itself well to implementation.

Representing the free magma: One way to represent the free magma is by using *well-formed* words of parentheses '(' and ')'. A word w is well-formed if it is made of parentheses coming in pairs of one left and one right bracket, such that the left bracket appears on the left of the corresponding right bracket in w. For example, $(())()$ is a well-formed word. The set of forests equipped with the product $\times$ is then isomorphic to this free magma via the recursion $\mathbb{I} = ()$, $\omega_1 \times \omega_2 = (\omega_1)\omega_2$.

The authors have implemented a variant of the free magma, with elements represented by parentheses, and also the basic operations discussed in this paper. In future work, this implementation will be used to do backward error analysis on interesting test cases, like the dynamics of rigid bodies.

Acknowledgements

We are grateful to Kurusch Ebrahimi-Fard, Dominique Manchon and Jon-Eivind Vatne for interesting and enlightening discussions, and to the anonymous referees for their valuable comments. We would also like to acknowledge support from the Aurora Program, project 205042/V11.

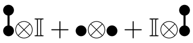
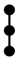
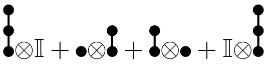
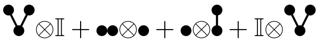
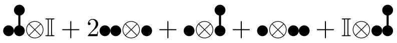
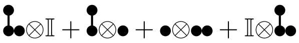
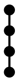
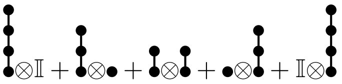
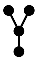
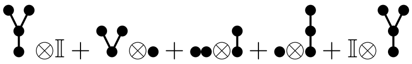
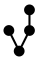
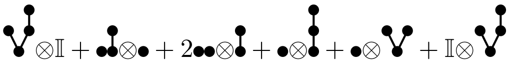
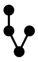
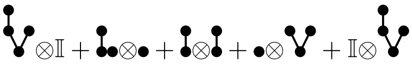
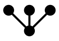
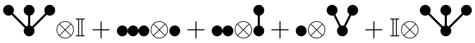
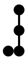
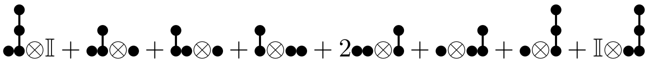
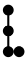
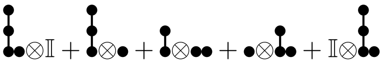
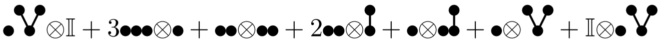
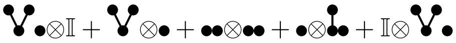
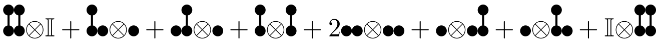
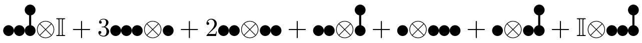
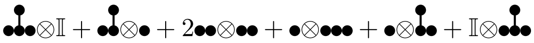
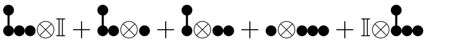
ω	$\Delta_N(\omega)$
$\mathbb{I}$	$\mathbb{I}\otimes\mathbb{I}$
●	●$\otimes\mathbb{I} + \mathbb{I}\otimes$●
	
●●	●●$\otimes\mathbb{I} +$ ●$\otimes$● $+ \mathbb{I}\otimes$●●
	
	
	
	
●●●	●●●$\otimes\mathbb{I} +$ ●●$\otimes$● $+$ ●$\otimes$●● $+ \mathbb{I}\otimes$●●●
	
	
	
	
	
	
	
	
	
	
	
	
	
●●●●	●●●●$\otimes\mathbb{I} +$ ●●●$\otimes$● $+$ ●●$\otimes$●● $+$ ●$\otimes$●●● $+ \mathbb{I}\otimes$●●●●

Table 1: Examples of the coproduct Δ_N

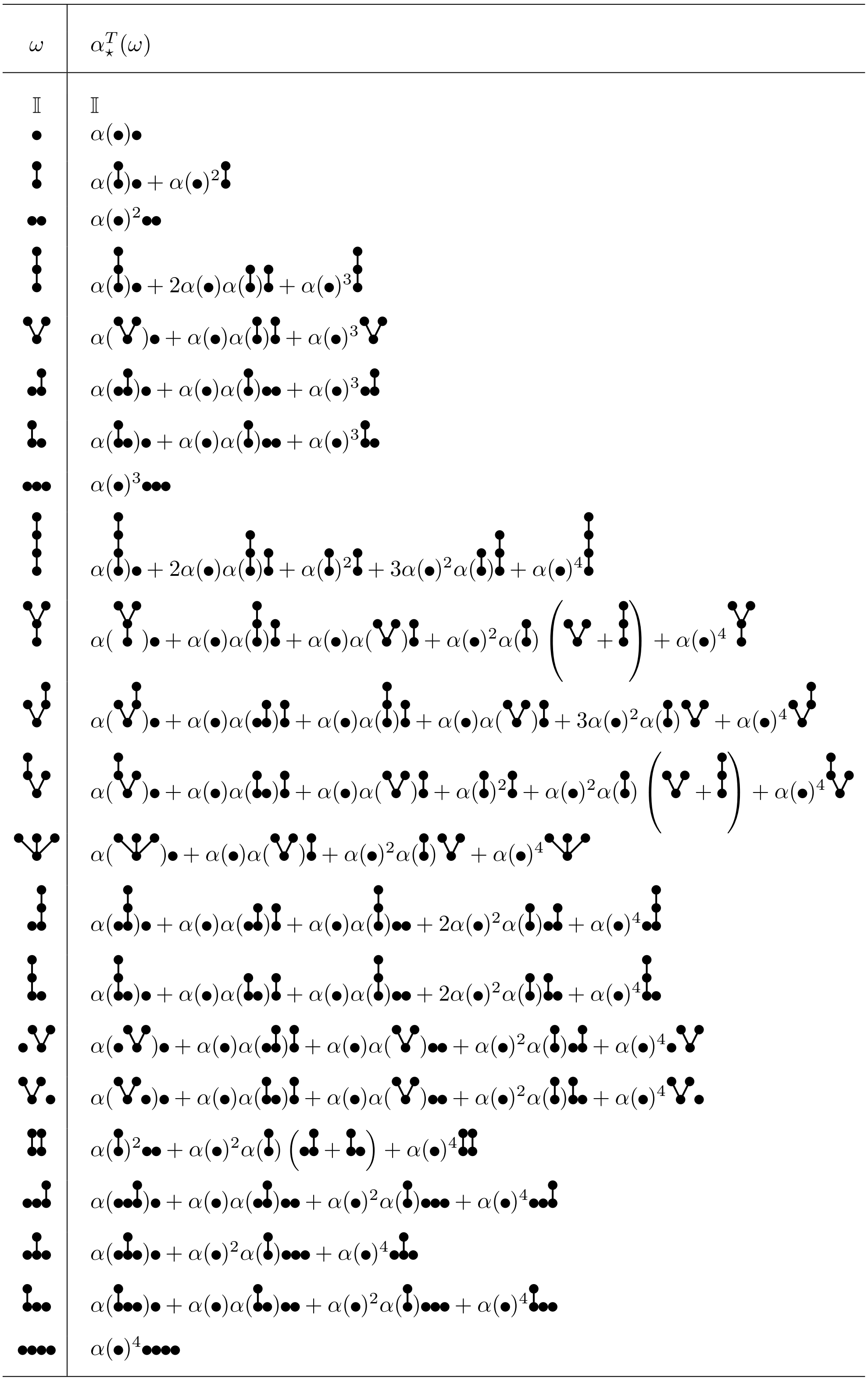

ω
$\alpha_\star^T(\omega)$
$\mathbb{I}$
$\mathbb{I}$

Table 2: Examples of the substitution character $\alpha_\star^T$, where α is an infinitesimal character, for all forests up to and including order four.

References

[1] G. Benettin and A. Giorgilli. On the Hamiltonian interpolation of near-to-the identity symplectic mappings with application to symplectic integration algorithms. *Journal of Statistical Physics*, 74(5):1117–1143, 1994.

[2] C. Brouder. Runge-Kutta methods and renormalization. *The European Physical Journal C: Particles and Fields*, 12(3):521–534, 2000.

[3] J.C. Butcher. An algebraic theory of integration methods. *Mathematics of Computation*, 26(117):79–106, 1972.

[4] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations.* John Wiley & Sons Inc, second edition, 2008.

[5] D. Calaque, K. Ebrahimi-Fard, and D. Manchon. Two interacting Hopf algebras of trees: A Hopf-algebraic approach to composition and substitution of B-series. *Advances in Applied Mathematics*, 47(2), 2011.

[6] F. Chapoton. Rooted trees and an exponential-like series. *ArXiv preprint, 0209104*, 2002.

[7] P. Chartier, E. Hairer, and G. Vilmart. A substitution law for B-series vector fields. Technical Report 5498, INRIA, 2005.

[8] P. Chartier, E. Hairer, and G. Vilmart. Numerical integrators based on modified differential equations. *Mathematics of Computation*, 76(260):1941–1954, 2007.

[9] P. Chartier, E. Hairer, and G. Vilmart. Algebraic structures of B-series. *Foundations of Computational Mathematics*, 10(4):407–427, 2010.

[10] P. Chartier and A. Murua. An algebraic theory of order. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(4):607–630, 2009.

[11] A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Communications in Mathematical Physics*, 199(1):203–242, 1998.

[12] K. Ebrahimi-Fard, J.M. Gracia-Bondía, and F. Patras. A Lie theoretic approach to renormalization. *Communications in Mathematical Physics*, 276(2):519–549, 2007.

[13] K. Ebrahimi-Fard, A. Lundervold, D. Manchon, H. Munthe-Kaas, and J.E. Vatne. On the post-Lie operad. *Preprint*, 2011.

[14] S. Faltinsen. Backward error analysis for Lie-group methods. *BIT Numerical Mathematics*, 40(4):652–670, 2000.

[15] L. Foissy. Faà di Bruno subalgebras of the Hopf algebra of planar trees from combinatorial Dyson–Schwinger equations. *Advances in Mathematics*, 218(1):136–162, 2008.

[16] E. Hairer. Backward analysis of numerical integrators and symplectic methods. *Annals of Numerical Mathematics*, 1(1-4):107–132, 1994.

[17] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration.* Springer, second edition, 2006.

[18] E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing*, 13(1):1–15, 1974.

[19] A. Iserles, H. Munthe-Kaas, S.P. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.

[20] J.L. Loday. *Cyclic Homology.* Springer, second edition, 1997.

[21] J.L. Loday and M.O. Ronco. Combinatorial Hopf algebras. *Quanta of Maths, Clay Mathematics Proceedings*, 11, 2010.

[22] A. Lundervold and H. Munthe-Kaas. Hopf algebras of formal diffeomorphisms and numerical integration on manifolds. *Contemporary Mathematics*, 539:295–324, 2011.

[23] A. Lundervold and H. Munthe-Kaas. On algebraic structures of numerical integration on vector spaces and manifolds. *ArXiv preprint, 1112.4465*, 2011.

[24] D. Manchon. Hopf Algebras in Renormalisation. In M. Hazewinkel, editor, *Handbook of Algebra*, volume 5, pages 365–427. North Holland, 2008.

[25] H. Munthe-Kaas. Lie–Butcher theory for Runge–Kutta methods. *BIT Numerical Mathematics*, 35(4):572–587, 1995.

[26] H. Munthe-Kaas. Runge–Kutta methods on Lie groups. *BIT Numerical Mathematics*, 38(1):92–111, 1998.

[27] H. Munthe-Kaas and A. Lundervold. On post-Lie algebras, Lie–Butcher series and moving frames. *ArXiv preprint, 1203.4738*, 2012.

[28] H. Munthe-Kaas and W. Wright. On the Hopf algebraic structure of Lie group integrators. *Foundations of Computational Mathematics*, 8(2):227–257, 2008.

[29] A. Murua. Formal series and numerical integrators, Part I: Systems of ODEs and symplectic integrators. *Applied Numerical Mathematics*, 29(2):221–251, 1999.

[30] A. Murua. The Hopf algebra of rooted trees, free Lie algebras, and Lie series. *Foundations of Computational Mathematics*, 6(4):387–426, 2006.

[31] B. Owren and A. Marthinsen. Runge–Kutta methods adapted to manifolds and based on rigid frames. *BIT Numerical Mathematics*, 39(1):116–142, 1999.

[32] S. Reich. Backward error analysis for numerical integrators. *SIAM Journal on Numerical Analysis*, 36(5):1549–1570, 1999.

[33] C. Reutenauer. *Free Lie algebras.* Oxford University Press, 1993.

[34] B. Vallette. Homology of generalized partition posets. *Journal of Pure and Applied Algebra*, 208(2):699–725, 2007.

Paper C

On pre-Lie-type algebras with torsion*

[*] This paper has been updated and will be published under the title *On post-Lie algebras, Lie-Butcher series and moving frames*. ArXiv: `http://arxiv.org/abs/1203.4738`

On pre-Lie-type algebras with torsion

Alexander Lundervold [*] Hans Munthe-Kaas[*]

Abstract

Pre-Lie algebras (also called Vinberg algebras) describe the algebra of flat and torsion free connections on a differential manifold. In this paper we will explore algebras of connections which have either non-vanishing torsion or curvature tensors. We will also show how the flat algebras with constant torsion are related to other algebraic structures, some of which appears in the study of numerical integration on homogeneous manifolds. Note that these algebras have also been studied by B. Vallette in [20], under the name *post-Lie algebras*.

1 Introduction

1.1 Pre-Lie, Lie admissible and FCT-algebras

Let $\{A, \rhd\}$ be an algebra where $\rhd\colon A\times A \to A$ is a non-associative, non-commutative product. Define the (negative) *associator* as

$$a_{\rhd}(x,y,z) := x \rhd (y \rhd z) - (x \rhd y) \rhd z. \tag{1}$$

The algebra A is called *pre-Lie* (or *Vinberg* or *left-symmetric*) [21, 5] if the associator is symmetric in the first two arguments

$$a_{\rhd}(x,y,z) - a_{\rhd}(y,x,z) = 0. \tag{2}$$

This implies that the commutator $x \rhd y - y \rhd x$ defines a Lie bracket. Pre-Lie algebras describe algebraic properties of flat and torsion-free connections on manifolds [12]. More generally, an algebra is called *Lie-admissible* if $x \rhd y - y \rhd x$ defines a Lie bracket. It is known that this condition holds if and only if

$$\mathfrak{S}(a_{\rhd}(x,y,z) - a_{\rhd}(y,x,z)) = 0, \tag{3}$$

where $\mathfrak{S}$ denotes the sum over the three cyclic permutations of x, y, z. [1, 6]. Lie admissible algebras model algebraic properties of a torsion-free connection with constant curvature on a manifold.

Motivated by applications related to flows on homogeneous and symmetric spaces, we propose a different generalization of pre-Lie algebras:

Definition 1.1. [FCT-algebra] A *flat algebra with constant torsion*, $\{A, [\cdot,\cdot], \rhd\}$ is a Lie algebra $\{A, [\cdot,\cdot]\}$ equipped with a non-commutative, non-associative product $\rhd\colon A\times A \to A$, called the *connection*, such that the connection act as a derivation of the Lie bracket:

$$x \rhd [y,z] = [x \rhd y, z] + [y, x \rhd z] \tag{4}$$

and the following *flatness condition* holds:

$$[x,y] \rhd z = a_{\rhd}(x,y,z) - a_{\rhd}(y,x,z). \tag{5}$$

The Lie bracket $[\cdot,\cdot]$ is called the *torsion.*

[*] Department of Mathematics, University of Bergen, Norway. Email: {alexander.lundervold, hans.munthe-kaas}@math.uib.no

Note that a pre-Lie algebra is FCT over an abelian Lie algebra, where $[\cdot,\cdot]=0$.

It turns out that FCT algebras have been studied before in a different setting, and under the name *post Lie algebras* [20].

Remark 1.2. In many examples one obtains (5) with opposite sign

$$[x,y]\rhd z = a_{\rhd}(y,x,z) - a_{\rhd}(x,y,z).$$

We could have defined *left* and *right* FCTs according to this sign. However, since the sign in (5) can always be switched by changing the sign in the definition of the torsion, we will not make this distinction.

A morphism $F : A \to B$ of FCT-algebras is a Lie algebra homomorphism that preserves the $\rhd$ operation:

$$\begin{aligned} F([x,y]) &= [F(x),F(y)] \\ F(x\rhd y) &= F(x)\rhd F(y) \end{aligned} \tag{6}$$

for all $x, y \in A$.

1.2 Algebraic structures of vector fields on manifolds

This section will motivate the definition of FCT-algebras through examples of algebras of vector fields on manifolds. Let ∇ be an affine connection on a differential manifold $\mathcal{M}$. The connection defines a non-commutative and non-associative product $x \rhd y := \nabla_x y$ on the set of vector fields such that

$$\begin{aligned} (fx)\rhd y &= f(x\rhd y) \\ x\rhd (fy) &= df(x)y + fx\rhd y \end{aligned}$$

for a scalar field f. The *torsion* of the connection is a skew-symmetric tensor $T\colon T\mathcal{M}\wedge T\mathcal{M}\to T\mathcal{M}$ defined in terms of two vector fields x, y as

$$T(x,y) = x\rhd y - y\rhd x - [\![x,y]\!], \tag{7}$$

where $[\![\cdot,\cdot]\!]$ denotes the Jacobi–Lie bracket of vector fields. The *curvature tensor* $R\colon T\mathcal{M}\wedge T\mathcal{M}\to \mathrm{End}(T\mathcal{M})$ is defined as

$$R(x,y)z = x\rhd(y\rhd z) - y\rhd(x\rhd z) - [\![x,y]\!]\rhd z = a_{\rhd}(x,y,z) - a_{\rhd}(y,x,z) + T(x,y)\rhd z. \tag{8}$$

The relationship between torsion and curvature is given by the Bianchi identities

$$\mathfrak{S}(T(T(x,y),z) + (\nabla_x T)(y,z)) \quad = \quad \mathfrak{S}(R(x,y)z) \tag{9}$$

$$\mathfrak{S}((\nabla_x R)(y,z) + R(T(x,y),z)) \quad = \quad 0. \tag{10}$$

Example 1.3 (Flat, torsion-free)**.** $T=0$ implies $[\![x,y]\!] = x\rhd y - y\rhd x$ and $R(x,y)z = a_{\rhd}(x,y,z) - a_{\rhd}(y,x,z)$. If also $R=0$, we obtain the pre-Lie condition (2).

Example 1.4 (Torsion-free, constant curvature)**.** If $T=0$ and $\nabla R=0$, the Bianchi identities reduce to $\mathfrak{S}(R(x,y)z)=0$, which is equivalent to the Lie-admissible condition (3) [6].

Example 1.5 (Flat, constant torsion)**.** If $R=0$ and $\nabla_x T=0$, the Bianchi identities reduce to the Jacobi identity $\mathfrak{S}(T(T(x,y),z))=0$. Thus the torsion defines a Lie bracket

$$[x,y] := -T(x,y).$$

In this case the connection is not Lie-admissible, but we have two distinct Lie algebras: one given by the torsion bracket $[x,y]$ and one by the Jacobi–Lie bracket $[\![x,y]\!]$, related by

$$[\![x,y]\!] = x\rhd y - y\rhd x + [x,y].$$

Lie groups and homogeneous spaces. A slightly different view on torsion and curvature appear in the theory of G-structures and $\mathfrak{g}$-valued forms on a manifold. This is the foundation for Cartan's method of moving frames, which has recently been recognized as an important tool in applied and computational mathematics [16, 11].

Let G be a Lie group with Lie algebra $\mathfrak{g}$, and let $\lambda\colon G \times \mathcal{M} \to \mathcal{M}$ a transitive left action of G on a homogeneous space $\mathcal{M}$, with infinitesimal generator

$$\lambda_\star\colon \mathfrak{g} \times \mathcal{M} \to T\mathcal{M}\colon (V, p) \mapsto \left.\frac{\partial}{\partial t}\right|_{t=0} \lambda(\exp(tV), p).$$

Let $\Omega^k(\mathcal{M}, \mathfrak{g})$ be the space of $\mathfrak{g}$-valued k-forms on $\mathcal{M}$, in particular $\Omega^0(\mathcal{M}, \mathfrak{g})$ is identified with the space of maps from $\mathcal{M}$ to $\mathfrak{g}$. Any $x \in \Omega^0(\mathcal{M}, \mathfrak{g})$ generates a vector field $X\colon \mathcal{M} \to T\mathcal{M}$ as

$$X(p) = \lambda_\star(x(p), p),$$

written in short form as $X = \lambda_\star(x)$.

The space $\Omega^0(\mathcal{M}, \mathfrak{g})$ has the structure of a FCT-algebra:

Proposition 1.6. *Let $\mathcal{M}$ be acted upon from left by a Lie group G with Lie algebra $\{\mathfrak{g}, [\cdot,\cdot]_{\mathfrak{g}}\}$. Let the Lie bracket $[\cdot,\cdot]\colon \Omega^0(\mathcal{M}, \mathfrak{g}) \times \Omega^0(\mathcal{M}, \mathfrak{g}) \to \Omega^0(\mathcal{M}, \mathfrak{g})$ and the product $\rhd\colon \Omega^0(\mathcal{M}, \mathfrak{g}) \times \Omega^0(\mathcal{M}, \mathfrak{g}) \to \Omega^0(\mathcal{M}, \mathfrak{g})$ be defined pointwise at $p \in \mathcal{M}$ as*

$$\begin{aligned}
[x, y](p) &= -[x(p), y(p)]_{\mathfrak{g}} \\
x \rhd y &= \lambda_*(x)(y) \quad \textit{(the Lie derivative of } y \textit{ along } \lambda_*(x)\textit{)}.
\end{aligned}$$

Then $\{\Omega^0(\mathcal{M}, \mathfrak{g}), [\cdot,\cdot], \rhd\}$ is a FCT-algebra.

Proof. This can be verified by a coordinate computation. Let $\{e_j\}$ be a basis for $\mathfrak{g}$ and $\partial_j = \lambda_*(e_j)$ the corresponding right invariant vector fields on $\mathcal{M}$. Note that $\lambda_*(-[e_j, e_k]) = \llbracket \partial_j, \partial_k \rrbracket$, where the right hand side is the Jacobi–Lie bracket of vector fields. Letting $x(p) = \sum_j x^j(p) e_j$ and $y(p) = \sum_k x^k(p) e_k$, where x^j and y^k are scalar functions on $\mathcal{M}$, we obtain

$$\begin{aligned}
[x, y] &= -\sum_{j,k} x^j y^k [e_j, e_k] \\
x \rhd y &= \sum_{j,k} x^j \partial_j (y^k) e_k.
\end{aligned}$$

The FCT conditions follow by a straightforward computation. See [15, Lemma 3] for a slightly different proof of a similar result. □

Example 1.7 (Maurer–Cartan form)**.** A one-form $\omega \in \Omega^1(\mathcal{M}, \mathfrak{g})$ is *compatible* with the group action if $\lambda_\star(\omega(X)) = X$ for all vector fields $X\colon \mathcal{M} \to T\mathcal{M}$. If $\mathcal{M} = G$ and $\lambda(g, p) = g \cdot p$ is the left action of G on itself, then the unique compatible $\omega \in \Omega^1(G, \mathfrak{g})$ is the right Maurer–Cartan form $\omega\colon TG \to \mathfrak{g}$, defined as the map moving $v \in T_gG$ to $\mathfrak{g} = T_eG$ by right translation: $\omega(V) = TR_{g^{-1}}V$.

The Maurer–Cartan form defines a linear isomorphism $\omega_p\colon T_pG \to \mathfrak{g}$ and hence defines an isomorphism between $\Omega^0(G, \mathfrak{g})$ and vector fields on G. Furthermore it satisfies the structural equation

$$d\omega + \frac{1}{2}\omega \wedge \omega = 0. \tag{11}$$

On a general (connected, smooth) manifold $\mathcal{M}$, the existence of a form with these two properties implies that $\mathcal{M}$ can be given the structure of a Lie group (up to a covering) [18, Theorem §8.8.7]. Thus the Maurer–Cartan form is fundamental in a differential geometric characterization of Lie groups.

The curvature of $\omega \in \Omega^1(G, \mathfrak{g})$ is given as $R = d\omega + \frac{1}{2}\omega \wedge \omega \in \Omega^2(G, \mathfrak{g})$, and (11) is a flatness condition equivalent to (5). Taking $\theta = \omega$ as a solder form, we compute the torsion form $\Theta = d\theta + \theta \wedge \omega = \frac{1}{2}\omega \wedge \omega \in \Omega^2(G, \mathfrak{g})$. This yields

$$\Theta(X, Y) = [\omega(X), \omega(Y)]_{\mathfrak{g}}.$$

Therefore, the Maurer–Cartan form has flat curvature and constant torsion.

We conclude that the structure of flat and torsion free connections is naturally occurring in the theory of homogeneous spaces, and in particular in the differential geometry of Lie groups.

2 The free FCT-algebra and universal enveloping algebras

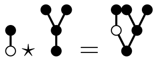
2.1 Free FCT-algebras

In [4] Chapoton and Livernet gave an explicit description of the free pre-Lie algebra in terms of decorated rooted trees and grafting. In this section we will see that there is a similar description of the free FCT-algebra. In fact, we will show that the free FCT-algebra can be described as the free Lie algebra over ordered rooted trees. Furthermore, we will relate FCT-algebras to D-algebras, studied in connection with numerical Lie group integration ([15, 9]). The universal enveloping algebra of an FCT-algebra is a D-algebra, and the FCT-algebra is recovered as the derivations in the D-algebra.

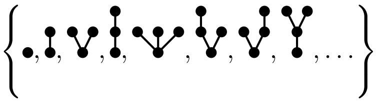
Trees. Let $\mathcal{C}$ be a set, henceforth called *colors*. We define $\mathrm{T}_{\mathcal{C}}$ the set of all ordered (or planar)[1] rooted trees with nodes colored by $\mathcal{C}$. Formally we define this as the free magma

$$\mathrm{T}_{\mathcal{C}} := \mathrm{Magma}(\mathcal{C}).$$

Recall that a *magma* is a set with a binary operation $\star$ without any algebraic relations imposed. The free magma over $\mathcal{C}$ consists of all possible ways to parenthesize binary operations on $\mathcal{C}$. We identify $\mathrm{Magma}(\mathcal{C})$ with planar trees, where the nodes are decorated with colors from $\mathcal{C}$. On trees we interpret $\star$ as the *Butcher-product* [3]: $\tau_1 \star \tau_2 = \tau$ is a tree where the root of the tree τ_1 is attached on the left part of the root of the tree τ_2. For example:

 $= \;(\bullet \star \circ) \star ((\bullet \star (\bullet \star \bullet)) \star \bullet).$

If $\mathcal{C} = \{\bullet\}$ has only one element, we write $\mathrm{T} := \mathrm{T}_{\{\bullet\}}$. The first few elements of T are:

$\mathrm{T} =$  .

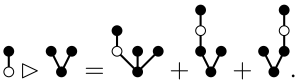
Note that any $\tau \in \mathrm{T}_{\mathcal{C}}$ has a unique maximal right factorization

$$\tau = \tau_1 \star (\tau_2 \star (\cdots (\tau_k \star c))), \quad \text{where } c \in \mathcal{C} \text{ and } \tau_1, \ldots, \tau_k \in \mathrm{T}_{\mathcal{C}}.$$

Here c is the root, k is the *fertility* of the root and $\tau_1, \ldots, \tau_k$ are the branches of the root. Let k be a field of characteristic zero and write $\mathrm{k}\{\mathrm{T}_{\mathcal{C}}\}$ for the free k-vector space over the set $\mathrm{T}_{\mathcal{C}}$, i.e. all k-linear combinations of trees. We define *left grafting*[2] $\rhd\colon \mathrm{T}_{\mathcal{C}} \times \mathrm{T}_{\mathcal{C}} \to \mathrm{k}\{\mathrm{T}_{\mathcal{C}}\}$ by the recursion

$$\begin{aligned}
\tau \rhd c :=&\ \tau \star c \\
\tau \rhd (\tau_1 \star (\tau_2 \star (\cdots (\tau_k \star c)))) :=&\ \tau \star (\tau_1 \star (\tau_2 \star (\cdots (\tau_k \star c)))) \\
&+ (\tau \rhd \tau_1) \star (\tau_2 \star (\cdots (\tau_k \star c))) \\
&+ \tau_1 \star ((\tau \rhd \tau_2) \star (\cdots (\tau_k \star c))) \\
&+ \cdots \\
&+ \tau_1 \star (\tau_2 \star (\cdots ((\tau \rhd \tau_k) \star c))).
\end{aligned} \tag{12}$$

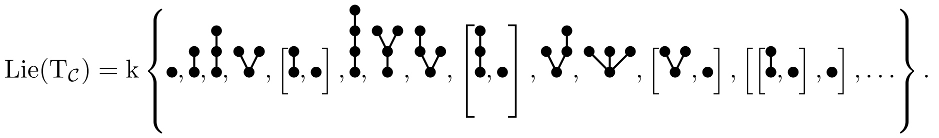
Thus $\tau_1 \rhd \tau_2$ is the sum of all the trees resulting from attaching the root of τ_1 from the left to all the nodes of the tree τ_2. Example:



[1] Trees with different orderings of the branches are considered different, as when pictured in the plane.

[2] Various notations for similar grafting products are found in the literature, e.g. $u \rhd v = u[v] = u \curvearrowright v$.

Free Lie algebras of trees. Let $\mathfrak{g} = \mathrm{Lie}(\mathrm{T}_{\mathcal{C}})$ denote the free Lie algebra over the set $\mathrm{T}_{\mathcal{C}}$ [17]. For $\mathcal{C} = \{\bullet\}$, a Lyndon basis is given up to order four as [13]:



Proposition 2.1. *Let the free Lie algebra* $\mathfrak{g} = \mathrm{Lie}(\mathrm{T}_{\mathcal{C}})$ *be equipped with a product* $\rhd \colon \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, *extended from the left grafting defined on* $\mathrm{T}_{\mathcal{C}}$ *in (12) as*

$$\begin{aligned} u \rhd [v, w] &= [u \rhd v, w] + [v, u \rhd w] && (13)\\ [u, v] \rhd w &= a_{\rhd}(u, v, w) - a_{\rhd}(v, u, w) && (14) \end{aligned}$$

for all $u, v, w \in \mathfrak{g}$. *Then* $\{\mathrm{Lie}(\mathrm{T}_{\mathcal{C}}), [\cdot, \cdot], \rhd\}$ *is an FCT-algebra.*

Proof. Since any $u, v, w \in \mathfrak{g}$ can be written as a sum of trees and commutators of trees, the connection is well-defined on $\mathfrak{g}$. By construction it satisfies the axioms of a FCT-algebra. □

Free FCT-algebras. Proposition 2.1 shows that the free Lie algebra of ordered trees has naturally the structure of an FCT-algebra $\mathrm{FCT}(\mathcal{C}) := \{\mathrm{Lie}(\mathrm{T}_{\mathcal{C}}), [\cdot, \cdot], \rhd\}$. We call this the *free FCT-algebra* over the set $\mathcal{C}$ for the following reason:

Theorem 2.2. *For any FCT-algebra* $\{\mathcal{A}, [\cdot, \cdot], \rhd\}$ *and any function* $f \colon \mathcal{C} \to \mathcal{A}$, *there exists a unique morphism of FCT-algebras* $\mathcal{F} \colon \mathrm{FCT}(\mathcal{C}) \to \mathcal{A}$ *such that* $\mathcal{F}(c) = f(c)$ *for all* $c \in \mathcal{C}$.

Proof. We construct $\mathcal{F}$ in two stages. First we show, using $\rhd$, that f extends uniquely to a function $\mathcal{F}_{\mathrm{T}_{\mathcal{C}}} \colon \mathrm{T}_{\mathcal{C}} \to \mathcal{A}$. Then by universality of the free Lie algebra, there is a unique Lie algebra homomorphism $\mathcal{F} \colon \mathrm{Lie}(\mathrm{T}_{\mathcal{C}}) \to \mathcal{A}$. We show that this is also a homomorphism for the connection product $\rhd$.

To construct the extension to $\mathrm{T}_{\mathcal{C}}$ we first observe that the magmatic product $\tau \star \tau'$ on $\mathrm{T}_{\mathcal{C}}$ (the Butcher product of two trees) can be expressed in terms of left grafting $\rhd$. This is done by induction in the fertility of τ'. For fertility 0, i.e. $\tau' = c \in \mathcal{C}$, we have $\tau \star c = \tau \rhd c$. For fertility k we write $\tau' = \tau_1 \star (\tau_2 \star (\cdots (\tau_k \star c)))$ and find from (12)

$$\tau \star \tau' = \tau \rhd \tau' - (\tau \rhd \tau_1) \star (\tau_2 \star (\cdots (\tau_k \star c))) - \cdots - (\tau_1 \star (\tau_2 \star (\cdots (\tau \rhd \tau_k \star c))).$$

In the right hand side of the equation, the fertility of any term on the right hand side of a $\star$-product is smaller than k, which completes the induction. The fact that $\mathrm{T}_{\mathcal{C}}$ is freely generated from $\mathcal{C}$ by the product $\star$ ensures that $\mathcal{F}_{\mathrm{T}_{\mathcal{C}}}$ is uniquely defined by

$$\begin{gathered} \mathcal{F}_{\mathrm{T}_{\mathcal{C}}}(c) = f(c) \quad \text{for all } c \in \mathcal{C} \\ \mathcal{F}_{\mathrm{T}_{\mathcal{C}}}(\tau \rhd \tau') = \mathcal{F}_{\mathrm{T}_{\mathcal{C}}}(\tau) \rhd \mathcal{F}_{\mathrm{T}_{\mathcal{C}}}(\tau'), \end{gathered}$$

and hence that also $\mathcal{F} \colon \mathrm{Lie}(\mathrm{T}_{\mathcal{C}}) \to \mathcal{A}$ is uniquely defined, as a Lie algebra homomorphism.

Finally, by induction on the length of iterated commutators, we see that $\mathcal{F}(u \rhd v) = \mathcal{F}(u) \rhd \mathcal{F}(v)$ for all $u, v \in \mathrm{Lie}(\mathrm{T}_{\mathcal{C}})$: If $u, v \in \mathrm{T}_{\mathcal{C}}$ this holds by construction. Assuming that $\mathcal{F}(u \rhd v) = \mathcal{F}(u) \rhd \mathcal{F}(v)$ whenever u and v are iterated commutators of length at most k, we find by using (13)–(14) that $\mathcal{F}([u, \tau_1] \rhd [v, \tau_2]) = \mathcal{F}([u, \tau_1]) \rhd \mathcal{F}([v, \tau_2])$ for all $\tau_1, \tau_2 \in \mathrm{T}_{\mathcal{C}}$. □

Proposition 2.3. *Let* $\mathrm{FCT}(\mathcal{C})$ *be graded with the number* n *counting the number of nodes in the trees. Then*

$$dim(\mathrm{FCT}(\mathcal{C})_n) = \frac{1}{2n} \sum_{d|n} \mu(\frac{n}{d}) \binom{2d}{d} n^{|\mathcal{C}|},$$

where μ *is the Möbius function. For* $|\mathcal{C}| = 1$ *the dimensions are* $1, 1, 3, 8, 25, 75, 245, \ldots$. *See also [19, A022553].*

Proof. See [14] and [13]. □

2.2 Universal enveloping algebras

In Section 3 we describe certain algebraic structures that occur naturally in the study of numerical integration methods on manifolds [15]. Central in this work are algebras of derivations, called *D-algebras*. We will see that FCT-algebras relate to D-algebras similarly to the relationship between a Lie algebra and its universal enveloping algebra.

Definition 2.4 (D-algebra)**.** Let B be a unital associative algebra with product $u, v \mapsto uv$, unit $\mathbb{I}$ and equipped with a non-associative product $\cdot \rhd \cdot \colon B \otimes B \to B$ such that $\mathbb{I} \rhd v = v$ for all $v \in B$. Write $\mathrm{Der}(B)$ for the set of all $u \in B$ such that $u \rhd \cdot$ is a derivation:

$$\mathrm{Der}(B) = \{u \in B \mid u \rhd (vw) = (u \rhd v)w + v(u \rhd w) \text{ for all } v, w \in B\}.$$

Then B is called a *D-algebra* if for any $u \in \mathrm{Der}(B)$ and any $v, w \in B$ we have

$$v \rhd u \in \mathrm{Der}(B) \tag{15}$$

$$(uv) \rhd w = u \rhd (v \rhd w) - (u \rhd v) \rhd w. \tag{16}$$

Proposition 2.5. *If B is a D-algebra then the derivations $\mathrm{Der}(B)$ form a FCT-algebra, with torsion $[u, v] = uv - vu$ and connection $\rhd$.*

Proof. If $u, g \in \mathrm{Der}(B)$ we note that

$$(uv - vu) \rhd \cdot = u \rhd (v \rhd \cdot) - v \rhd (u \rhd \cdot) + (u \rhd v) \rhd \cdot - (v \rhd u) \rhd \cdot.$$

The first two terms on the right is a commutator of two derivations and is therefore a derivation. The last two terms are derivations separately. Hence, $[u, v] \in \mathrm{Der}(B)$ and $\{\mathrm{Der}(B), [\cdot, \cdot]\}$ is a Lie algebra. The other axioms of being FCT follows easily from the definition of a D-algebra. □

Universal enveloping algebras. Let $\{A, [\cdot, \cdot], \rhd\}$ be an FCT-algebra, and let $U(A)$ be the universal enveloping algebra of the Lie algebra $\{A, [\cdot, \cdot]\}$. By the Poincaré–Birkhoff–Witt (PBW) theorem we can embed A as a linear subspace of $U(A)$, such that $[u, v] = uv - vu$. The embedding is also denoted by A. The product $\rhd$ on A can be extended to $U(A)$ according to:

$$\mathbb{I} \rhd v = v \tag{17}$$

$$u \rhd (vw) = (u \rhd v)w + v(u \rhd w) \tag{18}$$

$$(uv) \rhd w = u \rhd (v \rhd w) - (u \rhd v) \rhd w, \tag{19}$$

for all $u \in A$ and $v, w \in U(A)$.

Proposition 2.6. *Equations (17)–(19) define a unique extension of $\rhd$ from A to $U(A)$. With the non-associative product $\rhd$, $U(A)$ is a D-algebra with derivations $\mathrm{Der}(U(A)) = A$.*

Proof. See [7, Theorem V.1] for a proof that a derivation on a Lie algebra A extends uniquely to a derivation on $U(A)$. This justifies the extension on the right (18). The extension on the left, given by (17) and (19), is compatible with the the embedding $[u, v] \mapsto uv - vu$ due to the flatness condition (5) for FCTs. From the PBW basis on $U(A)$ it follows that these equations extend $\rhd$ uniquely to all of $U(A)$ also on the left. It is clear that $A \subset \mathrm{Der}(U(A))$. To check that $A = \mathrm{Der}(U(A))$ we verify from (17)–(19) that $\mathbb{I}$ is not a derivation and that $u_1, u_2 \in \mathrm{Der}(U(A)) \Rightarrow u_1u_2 \notin \mathrm{Der}(U(A))$, thus $\mathrm{Der}(U(A))$ cannot be larger than A. □

Definition 2.7 (Universal enveloping algebra of FCT)**.** We call $U(A)$ equipped with this D-algebra structure $\rhd$ the universal enveloping algebra of the FCT algebra A.

Proposition 2.8. *For any D-algebra B and any FCT morphism $f \colon A \to \mathrm{Der}(B)$ there exists a unique D-algebra morphism $\mathcal{F} \colon U(A) \to B$ such that $\mathcal{F}(u) = f(u)$ for all $u \in A$.*

Proof. $\mathcal{F}$ is uniquely defined as a unital associative algebra morphism. It remains to verify that $\mathcal{F}(u \rhd v) = \mathcal{F}(u) \rhd \mathcal{F}(v)$. $U(A)$ has a grading by the length of the monomial basis of PBW. Using (17)–(19) it follows by induction in the grading that $\mathcal{F}(u \rhd v) = \mathcal{F}(u) \rhd \mathcal{F}(v)$. □

Remark 2.9. The preceding results establishes that we have a pair of adjoint functors between the categories of D-algebras and FCT-algebras:

$$U(\cdot) : \text{FCT-alg} : \rightleftarrows \text{D-alg} : \mathrm{Der}(\cdot).$$

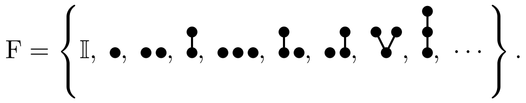
In other words, there is a natural isomorphism

$$\mathrm{Hom}_{\mathrm{FCT}}(\mathrm{Der}(A), B) \to \mathrm{Hom}_{\mathrm{D}}(A, U(B)).$$

Free D-algebras. A direct consequence of Theorem 2.2 and Proposition 2.8 is the following characterization of a free D-algebra:

Corollary 2.10 ([15, Proposition 1])**.** *The algebra* $\mathrm{D}_{\mathcal{C}} := U(\mathrm{FCT}(\mathcal{C}))$ *is the free D-algebra over the set* $\mathcal{C}$*, i.e. for any D-algebra* B *and any function* $f\colon \mathcal{C} \to \mathrm{Der}(B)$ *there exists a unique D-algebra morphism* $\mathcal{F}\colon \mathrm{D}_{\mathcal{C}} \to B$ *such that* $\mathcal{F}(c) = f(c)$ *for all* $c \in \mathcal{C}$*.*

The unital associative algebra of $\mathrm{D}_{\mathcal{C}}$ is $U(\mathrm{Lie}(\mathrm{T}_{\mathcal{C}}))$, which by the Cartier–Milner–Moore theorem is the free associative algebra over $\mathrm{T}_{\mathcal{C}}$. I.e. it is the noncommutative polynomials over rooted trees: $\mathrm{D}_{\mathcal{C}} = \mathrm{k}\langle \mathrm{T}_{\mathcal{C}} \rangle = \mathrm{k}\{\mathrm{F}_{\mathcal{C}}\}$, where $\mathrm{k}\{\mathrm{F}_{\mathcal{C}}\}$ denotes the free vector space over the set of *ordered forests.* $\mathrm{F}_{\mathcal{C}} := \mathrm{T}_{\mathcal{C}}^*$ consist of all words of finite length over the alphabet $\mathrm{T}_{\mathcal{C}}$, including the empty word $\mathbb{I}$. For $\mathcal{C} = \{\bullet\}$ these are



We can create a tree from a forest ω by applying the operator $\mathrm{B}_c^+ : \mathrm{F}_{\mathcal{C}} \to \mathrm{T}_{\mathcal{C}}$, attaching the trees in ω onto a common root labelled by $c \in \mathcal{C}$ and we can create a forest from a tree using the operator $\mathrm{B}^- : \mathrm{T}_{\mathcal{C}} \to \mathrm{F}_{\mathcal{C}}$ removing the root. The concatenation product $\omega_1, \omega_2 \mapsto \omega_1\omega_2$ is the associative operation of sticking shorter words together to create longer words.

Summarizing, the free D-algebra $\mathrm{D}_{\mathcal{C}}$ is the vector space of forests $\mathrm{k}\{F_{\mathcal{C}}\}$ with unit $\mathbb{I}$, concatenation product and the left grafting product $\rhd$ defined on trees in (12) and extended to forests by (17)–(19). This free D-algebra carries a Hopf algebra structure, closely related to the Connes–Kreimer Hopf algebra, to be discussed in the sequel.

3 Related algebraic structures

There are a number of interesting algebraic structures associated with FCT and D-algebras.

3.1 Dipterous, pre-Lie and Lie admissible algebras

The composition product $\circ$ on D-algebras. A *dipterous* algebra [8] is a triple $\{B, \circ, \rhd\}$, where B is a vector space and $\circ$ and $\rhd$ are two binary operations on B satisfying:

$$x\circ(y\circ z) = (x\circ y)\circ z \tag{20}$$

$$x \rhd (y \rhd z) = (x\circ y) \rhd z \tag{21}$$

for all $x, y, z \in B$. Let B be a D-algebra with concatenation $x, y \mapsto xy$ and connection product $x \rhd y$. Define a product $\circ\colon B \times B \to B$ as

$$\begin{aligned} \mathbb{I}\circ y &= y \\ x\circ y &:= xy + x \rhd y \\ (xy)\circ z &:= x\circ(y\circ z) - (x \rhd y)\circ z \quad \text{for all } x \in \mathrm{Der}(B), y, z \in B. \end{aligned} \tag{22}$$

Proposition 3.1. *If B is a D-algebra then $\{B, \circ, \rhd\}$ is a dipterous algebra.*

Proof. Proof by induction in the grading on B provided by the PBW basis. □

The product $x, y \mapsto x \circ y$ will be referred to as the *composition product*, while $x, y \mapsto xy$ is called *frozen composition*, due to the interpretation for differential operators on manifolds. Let $A = \Omega^0(\mathcal{M}, \mathfrak{g})$ be the FCT defined in Proposition 1.6, and let $B = U(A) = \Omega^0(\mathcal{M}, U(\mathfrak{g}))$. For $f, g \in B$ the frozen composition is $(fg)(p) = f(p)g(p)$, where we 'freeze' the value of f and g in a point $p \in \mathcal{M}$ and obtain the product from $U(\mathfrak{g})$. The composition $f, g \mapsto f \circ g$, on the other hand, corresponds to the fundamental operation of composing two differential operators on $\mathcal{M}$. For $f, g \in \mathrm{Der}(B)$ we have $f \circ g = fg + f \rhd g$, splitting the composition in a term fg where g is 'frozen' (constant) and a term $f \rhd g$ where the variation of g along f is taken into account.

On the free D-algebra $D_{\mathcal{C}}$ the composition is computed on two forests $\omega_1, \omega_2 \in \mathrm{F}_{\mathcal{C}}$ as ([15] Definition 2):

$$\omega_1 \circ \omega_2 = B^-(\omega_1 \rhd B^+(\omega_2)). \tag{23}$$

We call this the planar Grossman–Larson product, since it is a planar forest analogue of the Grossman–Larson product of unordered trees appearing in the Connes–Kreimer Hopf algebra.

Jacobi–Lie bracket on FCT.

Proposition 3.2. *If $\{A, [\cdot,\cdot], \rhd\}$ is FCT, then the bracket $[\![x, y]\!]$ defined as*

$$[\![x, y]\!] := x \rhd y - y \rhd x + [x, y]$$

is a Lie bracket, called the Jacobi–Lie bracket.

Proof. Identifying A with $\mathrm{Der}(U(A))$, we get

$$[\![x, y]\!] = x \circ y - y \circ x.$$

Since $\circ$ is associative, this is a Lie bracket. □

In the motivating examples of affine connections on $\mathcal{M}$ and homogeneous spaces in Proposition 1.6, the Lie bracket $[\![\cdot,\cdot]\!]$ corresponds to the Jacobi–Lie bracket of vector fields on $\mathcal{M}$.

Modified connections. By a modification of the product $\rhd$ in A, we obtain another FCT. The two structures can be interpreted as left and right adjoint isomorphic FCTs.

Proposition 3.3. *Let $\{A, [\cdot,\cdot], \rhd\}$ be FCT. Define the product $\blacktriangleright$ as*

$$x \blacktriangleright y = x \rhd y + [x, y].$$

Then $\{A, -[\cdot,\cdot], \blacktriangleright\}$ is FCT.

Proof. Since both $x \rhd \cdot$ and $[x, \cdot]$ are derivations on the torsion bracket, also $x \rhd \cdot + \alpha[x, \cdot]$ is a derivation, for any α. A direct computation shows that (5) holds with opposite sign. We change sign of the torsion and obtain another FCT. □

Proposition 3.4. *Let $\{A, [\cdot,\cdot], \rhd\}$ be FCT. Define the product $\succ$ as*

$$x \succ y = x \rhd y + \frac{1}{2}[x, y].$$

Then $\{A, \succ\}$ is Lie admissible, torsion free with constant curvature

$$R(x, y)z = -\frac{1}{4}[[x, y], z].$$

Proof. Lie admissible follows from $x \succ y - y \succ x = \llbracket x, y \rrbracket$. The curvature is

$$\begin{aligned} R(x,y)z &= x \succ (y \succ z) - x \leftrightarrow y - \llbracket x, y \rrbracket \succ z \\ &= x \rhd (y \rhd z + \frac{1}{2}[y,z]) + \frac{1}{2}[x, y \rhd z + \frac{1}{2}[y,z]] - x \leftrightarrow y - \llbracket x, y \rrbracket \rhd z - \frac{1}{2}[\llbracket x, y \rrbracket, z] \\ &= \frac{1}{2}[x \rhd y, z] + \frac{1}{2}[y, x \rhd z] + \frac{1}{2}[x, y \rhd z] + \frac{1}{4}[x,[y,z]] - x \leftrightarrow y - \frac{1}{2}[\llbracket x, y \rrbracket, z] \\ &= \frac{1}{4}[x,[y,z]] - x \leftrightarrow y - \frac{1}{2}[[x,y],z] = \frac{1}{4}[[x,y],z] - \frac{1}{2}[[x,y],z] \\ &= -\frac{1}{4}[[x,y],z], \end{aligned}$$

where $x \leftrightarrow y$ means swap x and y in everything to the left. □

3.2 Hopf algebras

Hopf algebraic structures related to the free D-algebra $\mathrm{D}_{\mathcal{C}}$ has been studied in [15, 9, 10]. These Hopf algebras can both be seen as generalizations of the shuffle–concatenation Hopf algebras of free Lie algebras as well as of the Connes–Kreimer Hopf algebra, which is closely related to pre-Lie algebras [4].

Shuffle product. From the classical theory of free Lie algebras, it follows that the derivations $\mathrm{Der}(\mathrm{D}_{\mathcal{C}})$ can be characterized in terms of shuffle products. Define the shuffle product $\sqcup\!\sqcup : \mathrm{D}_{\mathcal{C}} \otimes \mathrm{D}_{\mathcal{C}} \to \mathrm{D}_{\mathcal{C}}$ on the free D-algebra $\mathrm{D}_{\mathcal{C}}$ by $\mathbb{I} \sqcup\!\sqcup \omega = \omega = \omega \sqcup\!\sqcup \mathbb{I}$ and

$$(\tau_1\omega_1) \sqcup\!\sqcup (\tau_2\omega_2) = \tau_1(\omega_1 \sqcup\!\sqcup \tau_2\omega_2) + \tau_2(\tau_1\omega_1 \sqcup\!\sqcup \omega_2)$$

for $\tau_1, \tau_2 \in \mathrm{T}$, $\omega_1, \omega_2 \in \mathrm{F}$. Let $(\cdot,\cdot)$ be an inner product on $\mathrm{D}_{\mathcal{C}}$ defined such that the forests form an orthonormal basis, and let the coproduct $\Delta_{\sqcup\!\sqcup} \colon \mathrm{D}_{\mathcal{C}} \to \mathrm{D}_{\mathcal{C}} \otimes \mathrm{D}_{\mathcal{C}}$ be the adjoint of $\sqcup\!\sqcup$.

Proposition 3.5. *The free D-algebra $\mathrm{D}_{\mathcal{C}}$ has the structure of a cocommutative Hopf algebra $\mathcal{H}'_N = \{\mathrm{k}\{\mathrm{F}_{\mathcal{C}}\}, \epsilon, \circ, \eta, \Delta_{\sqcup\!\sqcup}, S\}$ with product being the planar Grossman–Larson product $\circ$ defined in (23), the coproduct $\Delta_{\sqcup\!\sqcup}$ is the adjoint of the shuffle and the unit η and counit ϵ are given as*

$$\begin{aligned} \eta(1) &= \mathbb{I} \\ \epsilon(\mathbb{I}) &= 1, \quad \epsilon(\omega) = 0 \quad \textit{for all } \omega \in \mathrm{F}_{\mathcal{C}} \backslash \{\mathbb{I}\}. \end{aligned}$$

The primitive elements are Prim($\mathcal{H}'_N$) $= \mathrm{Der}(\mathrm{D}_{\mathcal{C}})$. The antipode S is defined in [15].

Proof. The Hopf algebraic structure (for the dual of $\mathcal{H}'_N$) is proven in [15]. Characterization of the primitive elements follows from the free Lie algebra structure [17]. □

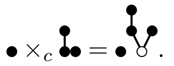
The Hopf algebra $\mathcal{H}_N$ and Lie–Butcher theory. In the study of numerical integration on manifolds it is important to characterize flows and parallel transport on manifolds with connections algebraically. It is convenient to base this on the dual Hopf algebra of $\mathcal{H}'_N$. Let $\mathcal{H}_N = \{\mathrm{k}\{\mathrm{F}_{\mathcal{C}}\}, \epsilon, \sqcup\!\sqcup, \eta, \Delta_\circ, S\}$ be the commutative Hopf algebra of planar forests, where the product is the shuffle product $\sqcup\!\sqcup$ and the coproduct $\Delta_\circ$ the adjoint of the planar Grossman–Larson product. Various expressions for $\Delta_\circ$ and the antipode S are derived in [15]. Our definition of $\mathrm{F}_{\mathcal{C}}$ and $\mathcal{H}_N$ is rather involved, going via trees and enveloping algebras extending $\rhd$ from derivations, introducing the dipterous composition $\circ$ and dualizing to obtain $\Delta_\circ$. However, both $\mathrm{F}_{\mathcal{C}}$ and the Hopf algebra $\mathcal{H}_N$ can alternatively be defined in a compact, recursive manner. We will review this definition, which will be the foundation for the computer implementation of $\mathcal{H}_N$ currently under construction.

Definition 3.6 (Magmatic definition of $\mathrm{F}_{\mathcal{C}}$)**.** Given a set $\mathcal{C}$ we let $\{\times_c\}_{c\in\mathcal{C}}$ be a collection of magmatic products, without any defining relations. Let $\mathbb{I}$ denote the unity and we define $\mathrm{F}_{\mathcal{C}}$ as the free magma generated from $\mathbb{I}$ by the magmatic products.

This definition is related to our previous definition of $\mathrm{F}_{\mathcal{C}}$ by interpreting $\omega_1 \times_c \omega_2$ in terms of forests as

$$\omega_1 \times_c \omega_2 = \omega_1 B_c^+(\omega_2) \tag{24}$$

for all $\omega_1, \omega_2 \in \mathrm{F}_{\mathcal{C}}$, $c \in \mathcal{C}$. Thus, e.g. for $c = \circ$ we have $\mathbb{I} \times_c \mathbb{I} = \circ$, and



Any $\omega \in \mathrm{F}_{\mathcal{C}} \setminus \{\mathbb{I}\}$ can be written uniquely as $\omega = \omega_L \times_c \omega_R$, where $c \in \mathcal{C}$ is the root of the rightmost tree in the forest. We call ω_L and ω_R the left and right parts of ω and c the right root.

Definition 3.7 (Shuffle product.)**.** The shuffle product $⧢\colon\ \mathrm{k}\{\mathrm{F}_{\mathcal{C}}\} \otimes \mathrm{k}\{\mathrm{F}_{\mathcal{C}}\} \to \mathrm{k}\{\mathrm{F}_{\mathcal{C}}\}$ is defined by k-linearity and the recursion

$$\begin{aligned} &\mathbb{I} ⧢ \omega = \omega ⧢ \mathbb{I} = \omega, \quad \text{for all } \omega \in \mathrm{F}_{\mathcal{C}}, \\ &v ⧢ \omega = (v_L ⧢ \omega) \times_c v_R + (v ⧢ \omega_L) \times_d \omega_R, \quad \text{for } v = v_L \times_c v_R,\ \omega = \omega_L \times_d \omega_R. \end{aligned} \tag{25}$$

Definition 3.8 (Coproduct.)**.** The coproduct $\Delta_\circ\colon\ \mathrm{k}\{\mathrm{F}_{\mathcal{C}}\} \to \mathrm{k}\{\mathrm{F}_{\mathcal{C}}\} \otimes \mathrm{k}\{\mathrm{F}_{\mathcal{C}}\}$ is defined by k-linearity and the recursion

$$\begin{aligned} \Delta_\circ(\mathbb{I}) &= \mathbb{I} \otimes \mathbb{I} \\ \Delta_\circ(\omega) &= \omega \otimes \mathbb{I} + \Delta_\circ(\omega_L) ⧢ \times_d \Delta_\circ(\omega_R), \quad \text{for } \omega = \omega_L \times_d \omega_R, \end{aligned} \tag{26}$$

where $⧢ \times_d$ is the shuffle product on the left and the magmatic product $\times_d$ on the right:

$$(u_1 \otimes u_2) ⧢ \times_d (v_1 \otimes v_2) := (u_1 ⧢ v_1) \otimes (u_2 \times_d v_2).$$

Proposition 3.9 ([15])**.** *$\mathcal{H}_N = \{\mathrm{k}\{\mathrm{F}_{\mathcal{C}}\}, \epsilon, ⧢, \eta, \Delta_\circ, S\}$ is a commutative Hopf algebra.*

The Hopf algebra $\mathcal{H}_N$ is the setting for *Lie–Butcher series.*

Definition 3.10 (Lie–Butcher series)**.** Let $\mathcal{H}_N^* = \mathrm{Hom}_{\mathrm{k}}(\mathcal{H}_N, \mathrm{k})$ denote the linear dual space of $\mathcal{H}_N$. An element $\alpha \in \mathcal{H}_N^*$ is called a Lie–Butcher series. We identify α with an infinite series

$$\alpha = \sum_{\omega \in \mathrm{F}_{\mathcal{C}}} \alpha(\omega)\omega,$$

via a dual pairing $(\cdot,\cdot)\colon \mathcal{H}_N^* \times \mathcal{H}_N \to \mathrm{k}$ defined such that

$$\alpha(\omega) = (\alpha, \omega) \quad \text{for all } \omega \in \mathrm{F}_{\mathcal{C}}.$$

The Lie–Butcher series (LB-series) form the basis for *Lie–Butcher theory*, which studies how numerical methods can be represented as LB-series, and how basic operations like composition and substitution of LB-series behaves. Lie–Butcher theory has been studied by several authors, see [2, 9], and references therein. A future project (and one of the main motivations for introducing FCT-algebras) is to reformulate Lie–Butcher theory in the language of FCT algebras. That way, LB-series will be connected closer to their roots as Lie series. We hope that this can lead to new results and insights into their structure and properties.

References

[1] A.A. Albert. Power-associative rings. *Transactions of the American Mathematical Society*, 64(3):552–593, 1948.

[2] H. Berland and B. Owren. Algebraic structures on ordered rooted trees and their significance to Lie group integrators. *Group theory and numerical analysis*, 39:49–63, 2005.

[3] J.C. Butcher. An algebraic theory of integration methods. *Mathematics of Computation*, 26(117):79–106, 1972.

[4] F. Chapoton and M. Livernet. Pre-Lie algebras and the rooted trees operad. *International Mathematics Research Notices*, 2001(8):395–408, 2001.

[5] M. Gerstenhaber. The cohomology structure of an associative ring. *Annals of Mathematics*, 78(2):267–288, 1963.

[6] M. Goze and E. Remm. Lie-admissible algebras and operads. *Journal of algebra*, 273(1):129–152, 2004.

[7] N. Jacobson. *Lie algebras*. Dover, 1979.

[8] J.L. Loday and M.O. Ronco. Combinatorial Hopf algebras. *Quanta of Maths, Clay Mathematics Proceedings*, 11, 2010.

[9] A. Lundervold and H. Z. Munthe-Kaas. Hopf algebras of formal diffeomorphisms and numerical integration on manifolds. *Contemporary Mathematics*, 539:295–324, 2011.

[10] A. Lundervold and H.Z. Munthe-Kaas. Backward error analysis and the substitution law for Lie group integrators. *Submitted*, 2011. ArXiv preprint math:1106.1071.

[11] E.L. Mansfield. *A practical guide to the invariant calculus*. Cambridge Univ. Press, 2010.

[12] Y. Matsushima. Affine Structures on Complex Manifolds. *Osaka J. Math*, 5:215–222, 1968.

[13] H. Munthe-Kaas and S. Krogstad. On enumeration problems in Lie–Butcher theory. *Future Generation Computer Systems*, 19(7):1197–1205, 2003.

[14] H. Munthe-Kaas and B. Owren. Computations in a free Lie algebra. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 357(1754):957, 1999.

[15] H. Munthe-Kaas and W. Wright. On the Hopf algebraic structure of Lie group integrators. *Foundations of Computational Mathematics*, 8(2):227–257, 2008.

[16] P.J. Olver. A survey of moving frames. *Computer Algebra and Geometric Algebra with Applications*, pages 105–138, 2005.

[17] C. Reutenauer. *Free Lie algebras*. Oxford University Press, 1993.

[18] R.W. Sharpe. *Differential geometry: Cartan's generalization of Klein's Erlangen program*. Springer, 1997.

[19] N.J.A. Sloane. The On-Line Encyclopedia of Integer Sequences. 2011.

[20] B. Vallette. Homology of generalized partition posets. *Journal of Pure and Applied Algebra*, 208(2):699–725, 2007.

[21] E.B. Vinberg. Convex homogeneous cones. *Transactions of the Moscow Mathematical Society*, 12:340–403, 1963.