

On the Lie Enveloping Algebra of a Post-Lie Algebra

Kurusch Ebrahimi-Fard, Alexander Lundervold,
and Hans Z. Munthe-Kaas*

Communicated by P. Olver

Abstract. We consider pairs of Lie algebras \mathfrak{g} and $\bar{\mathfrak{g}}$, defined over a common vector space, where the Lie brackets of \mathfrak{g} and $\bar{\mathfrak{g}}$ are related via a post-Lie algebra structure. The latter can be extended to the Lie enveloping algebra $\mathcal{U}(\mathfrak{g})$. This permits us to define another associative product on $\mathcal{U}(\mathfrak{g})$, which gives rise to a Hopf algebra isomorphism between $\mathcal{U}(\bar{\mathfrak{g}})$ and a new Hopf algebra assembled from $\mathcal{U}(\mathfrak{g})$ with the new product.

For the free post-Lie algebra these constructions provide a refined understanding of a fundamental Hopf algebra appearing in the theory of numerical integration methods for differential equations on manifolds. In the pre-Lie setting, the algebraic point of view developed here also provides a concise way to develop Butcher's order theory for Runge–Kutta methods.

Mathematics Subject Classification 2010: 65L, 53C, 16T.

Key Words and Phrases: Rooted trees, combinatorial Hopf algebras, post-Lie algebras, universal enveloping algebras, numerical Lie group integration, geometric numerical integration, Butcher's order theory.

1. Introduction

Classical numerical integrators aim at approximating flows of vector fields given by differential equations

$$\dot{y}(t) = F(y(t)), \quad y(0) = y_0,$$

where F is a vector field on \mathbb{R}^n . The so-called *Lie-group integrators* are generalizations to differential equations evolving on manifolds \mathcal{M} . Given a Lie group G acting in a transitive manner on \mathcal{M} , the Lie group integrators approximate differential equations written in the form

$$\dot{y}(t) = F(y(t)) = f(y(t)) \cdot y(t), \quad y(0) = y_0, \quad (1)$$

* We acknowledge support from ICMAT and the Severo Ochoa Excelencia Program, as well as the Nils-Abel project 010-ABEL-CM-2014A, and the SPIRIT project RCN:231632 of the Norwegian Research Council. The first author is supported by a Ramón y Cajal research grant from the Spanish government, and acknowledges support from the Spanish government under project MTM2013-46553-C3-2-P.

where $f: \mathcal{M} \rightarrow \mathfrak{g}$ represents the vector field F via a map to \mathfrak{g} , the Lie algebra of G . The basic problem of numerical Lie group integration is the approximation of the exact solution by computations in \mathfrak{g} , the exponential $\exp: \mathfrak{g} \rightarrow G$ and the action of G on \mathcal{M} . See [14] for a comprehensive review.

Answering questions related to order conditions for Lie group integrators relies on an understanding of the algebraic structure of non-commuting vector fields on \mathcal{M} generated from the vector field f . Our work addresses such algebraic and combinatorial aspects underlying the theory of Lie–Butcher series [18]. More precisely, we describe the setting of an important commutative Hopf algebra of planar rooted trees [24], and the corresponding group and Lie algebra of characters and infinitesimal characters, respectively. In a nutshell, the theory of Lie–Butcher series results from merging Lie theory with the theory of Butcher’s B -series. The latter is well-known, and plays a prominent role in the theory of numerical integration for differential equations on vector spaces [3, 4, 12].

The underlying context is the following: We have a Lie algebra $\mathfrak{g} = (\mathcal{V}, [-, -])$ over a field k of characteristic zero. In addition, the vector space \mathcal{V} is equipped with a binary product $\triangleright: \mathcal{V} \otimes \mathcal{V} \rightarrow \mathcal{V}$, such that the bracket

$$\llbracket x, y \rrbracket := x \triangleright y - y \triangleright x + [x, y]$$

defines a second Lie algebra $\bar{\mathfrak{g}} = (\mathcal{V}, \llbracket -, - \rrbracket)$ on \mathcal{V} . The Lie algebra \mathfrak{g} together with the appropriate relations between its Lie bracket and the product $\triangleright: \mathcal{V} \otimes \mathcal{V} \rightarrow \mathcal{V}$ define what is called a post-Lie algebra. It reduces to a pre-Lie algebra (also called a *Vinberg algebra*) [5, 17] if the Lie algebra \mathfrak{g} is abelian.

Post-Lie algebras appear in both geometry and algebra [22, 31]. From a geometric viewpoint, they can be introduced by means of an archetypal example from differential geometry¹.

Recall that an affine connection (covariant derivative) on the space of smooth vector fields $\mathcal{X}(\mathcal{M})$ on a manifold \mathcal{M} is a map $\nabla: \mathcal{X}(\mathcal{M}) \times \mathcal{X}(\mathcal{M}) \rightarrow \mathcal{X}(\mathcal{M})$ satisfying $\nabla_{fx}y = f\nabla_x y$ and $\nabla_x(fy) = df(x)y + f\nabla_x y$ for any $x, y \in \mathcal{X}(\mathcal{M})$ and scalar field $f: \mathcal{M} \rightarrow \mathbb{R}$. It yields a (non-commutative and non-associative) \mathbb{R} -bilinear product on $\mathcal{X}(\mathcal{M})$, which we denote $x \triangleright y := \nabla_x y$. The torsion of the connection is a skew-symmetric tensor $T: T\mathcal{M} \wedge T\mathcal{M} \rightarrow T\mathcal{M}$ defined as

$$T(x, y) = x \triangleright y - y \triangleright x - \llbracket x, y \rrbracket_J, \quad (2)$$

where $\llbracket \cdot, \cdot \rrbracket_J$ denotes the Jacobi–Lie bracket of vector fields, defined such that $\llbracket x, y \rrbracket_J(\phi) = x(y(\phi)) - y(x(\phi))$ for all vector fields x, y and scalar fields ϕ . The curvature tensor $R: T\mathcal{M} \wedge T\mathcal{M} \rightarrow \text{End}(T\mathcal{M})$ is defined as

$$\begin{aligned} R(x, y)z &= x \triangleright (y \triangleright z) - y \triangleright (x \triangleright z) - \llbracket x, y \rrbracket_J \triangleright z \\ &= a_{\triangleright}(x, y, z) - a_{\triangleright}(y, x, z) + T(x, y) \triangleright z, \end{aligned} \quad (3)$$

where $a_{\triangleright}(x, y, z) := x \triangleright (y \triangleright z) - (x \triangleright y) \triangleright z$ is the associator with respect to the product \triangleright . The relationship between torsion and curvature is given by the Bianchi

¹To keep the discussion as simple as possible, we discuss an affine connection on the tangent bundle. A similar construction is also valid on more general vector bundles such as the Atiyah Lie algebroid [22].

identities

$$\begin{aligned} \sum_{\circlearrowleft} (\mathbb{T}(\mathbb{T}(x, y), z) + (\nabla_x \mathbb{T})(y, z)) &= \sum_{\circlearrowleft} (\mathbb{R}(x, y)z) \\ \sum_{\circlearrowleft} ((\nabla_x \mathbb{R})(y, z) + \mathbb{R}(\mathbb{T}(x, y), z)) &= 0, \end{aligned} \quad (4)$$

where \sum_{\circlearrowleft} denotes the sum over the three cyclic permutations of (x, y, z) . If a connection is flat $\mathbb{R} = 0$ and has constant torsion $\nabla_x \mathbb{T} = 0$, then (4) reduces to the Jacobi identity $\sum_{\circlearrowleft} (\mathbb{T}(\mathbb{T}(x, y), z)) = 0$. Hence the torsion defines a Lie bracket $[x, y] := -\mathbb{T}(x, y)$, which is related to the Jacobi–Lie bracket by (2). The covariant derivation formula $\nabla_x (\mathbb{T}(y, z)) = (\nabla_x \mathbb{T})(y, z) + \mathbb{T}(\nabla_x y, z) + \mathbb{T}(y, \nabla_x z)$ together with $\nabla_x \mathbb{T} = 0$ then imply

$$x \triangleright [y, z] = [x \triangleright y, z] + [y, x \triangleright z]. \quad (5)$$

On the other hand, (3) together with $\mathbb{R} = 0$ imply

$$[x, y] \triangleright z = a_{\triangleright}(x, y, z) - a_{\triangleright}(y, x, z). \quad (6)$$

In Section 2 relations (5) and (6) are formalized to the notion of post-Lie algebras.

Note that for a connection which is both flat, $\mathbb{R} = 0$, and torsion free, $\mathbb{T} = 0$, equation (3) implies that $a_{\triangleright}(x, y, z) - a_{\triangleright}(y, x, z) = 0$, and we obtain a (left) pre-Lie algebra. The free pre-Lie algebra can be described as the space of non-planar rooted trees with product given by grafting of trees [8]. We remark that already in the 1850s Arthur Cayley [6] considered rooted trees as a representation of combinatorial structures related to the free pre-Lie algebra. More than a century later, these structures form the foundation of John Butcher’s theory of *B*-series [2, 3], which has become an indispensable tool in the analysis of numerical integration [12].

Pre-Lie structures on non-planar rooted trees lead to Hopf algebras of combinatorial nature. Combinatorial Lie and Hopf algebras, in particular those defined on rooted trees, have recently attracted a great deal of attention [7, 11, 13, 26].

The basic setting for Lie–Butcher series is provided by a combinatorial Hopf algebra on planar rooted trees, which accompanies a post-Lie structure on the trees [24]. One of the main goals of this work is to provide a precise description of this connection in the context of the free post-Lie algebra. Our approach follows Guin and Oudom [27] by extending the post-Lie structure on a Lie algebra \mathfrak{g} to the corresponding Lie enveloping algebra $\mathcal{U}(\mathfrak{g})$. This permits us to define another associative product on $\mathcal{U}(\mathfrak{g})$, compatible with its usual coalgebra structure. The Hopf algebra assembled from $\mathcal{U}(\mathfrak{g})$ and the new product is isomorphic to the Hopf algebra $\mathcal{U}(\bar{\mathfrak{g}})$, where $\bar{\mathfrak{g}}$ is the Lie algebra defined over the same vector space as \mathfrak{g} , whose Lie bracket is defined in terms of a post-Lie algebra structure.

Once the combinatorial Hopf algebra for Lie–Butcher series has been unfolded we explore some applications, in particular to the order theory of numerical methods on manifolds.

The outline of the paper is as follows. In the next section we recall the definition of pre- and post-Lie algebra. Section 3 generalizes some important result from [27], i.e. the extension of post-Lie structures to universal enveloping Lie algebras, and a Hopf algebra isomorphism between two enveloping algebras. In Section 4 we study numerical integration on pre- and post-Lie algebras.

2. Pre- and post-Lie algebras

We begin by defining pre-Lie algebras, see [1, 5, 17] for further details. A field k of characteristic zero, e.g. $k \in \{\mathbb{R}, \mathbb{C}\}$, is fixed once and for all. Let \mathcal{P} be a vector space equipped with a bilinear product $\triangleright : \mathcal{P} \times \mathcal{P} \rightarrow \mathcal{P}$, satisfying the left pre-Lie relation

$$(x \triangleright y) \triangleright z - x \triangleright (y \triangleright z) = (y \triangleright x) \triangleright z - y \triangleright (x \triangleright z). \quad (7)$$

We call $(\mathcal{P}, \triangleright)$ a left pre-Lie algebra. Note that identity (7) can be written as

$$\ell_{[x, y] \triangleright}(z) = [\ell_{x \triangleright}, \ell_{y \triangleright}](z),$$

where the linear map $\ell_{x \triangleright} : \mathcal{P} \rightarrow \mathcal{P}$ is defined by $\ell_{x \triangleright} y := x \triangleright y$. The bracket on the left-hand side is defined by $[x, y] := x \triangleright y - y \triangleright x$. As a consequence this bracket satisfies the Jacobi identity, turning \mathcal{P} into a Lie algebra.

The notion of post-Lie algebras was discovered from two distinct directions [24, 31]. We follow the definition given in [22]. The *associator* is defined by $a_{\triangleright}(x, y, z) := x \triangleright (y \triangleright z) - (x \triangleright y) \triangleright z$.

Definition 2.1. [22] A post-Lie algebra $(\mathcal{A}, [-, -], \triangleright)$ is a Lie algebra $\mathfrak{g} = (\mathcal{A}, [-, -])$ together with a bilinear product $\triangleright : \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$ such that for all $x, y, z \in \mathcal{A}$

$$\begin{aligned} x \triangleright [y, z] &= [x \triangleright y, z] + [y, x \triangleright z] \\ [x, y] \triangleright z &= a_{\triangleright}(x, y, z) - a_{\triangleright}(y, x, z). \end{aligned}$$

Proposition 2.2. Let $(\mathcal{A}, [-, -], \triangleright)$ be a post-Lie algebra. The bracket

$$\llbracket x, y \rrbracket = x \triangleright y - y \triangleright x + [x, y] \quad (8)$$

satisfies the Jacobi identity for all $x, y \in \mathcal{A}$.

Remark 2.3. If $(\mathcal{A}, [-, -])$ is an abelian Lie algebra, then $(\mathcal{A}, \triangleright)$ reduces to a left pre-Lie algebra with corresponding Lie bracket (8).

The post-Lie algebra defines relations between two Lie algebras $[-, -]$ and $\llbracket -, - \rrbracket$, which will be explored in the sequel. It should be noted that the role of these two brackets is not at all symmetric in the definition of the post-Lie structure. Whereas the above definition relating $[-, -]$ and \triangleright seems, algebraically speaking, to be the nicest definition, we could instead follow a more geometrically motivated definition, where we equip a general Lie algebra $\{\mathcal{A}, \llbracket -, - \rrbracket\}$ with a product $\triangleright : \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$ such that the ‘curvature’ of \triangleright is zero:

$$R(x, y)z := x \triangleright (y \triangleright z) - y \triangleright (x \triangleright z) - \llbracket x, y \rrbracket \triangleright z = 0. \quad (9)$$

We then define the ‘torsion’ of \triangleright as

$$[x, y] := \llbracket x, y \rrbracket + y \triangleright x - x \triangleright y,$$

and require that this satisfies the condition

$$x \triangleright [y, z] = [x \triangleright y, z] + [y, x \triangleright z] \quad (10)$$

(geometrically, $\nabla T = 0$).

Proposition 2.4. *Let $\{\mathcal{A}, [\cdot, \cdot]\}$ be a Lie algebra with a product $\triangleright: \mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$ satisfying (9)–(10). Then $\{A, [\cdot, \cdot], \triangleright\}$ is post-Lie.*

Proof. A direct computation shows that $[\cdot, \cdot]$ satisfies the Jacobi rule for a Lie bracket. The other conditions of post-Lie are baked into the defining equations. ■

Post-Lie algebras always come in pairs called *adjoint* post-Lie algebras. E.g. in the case of vector fields on Lie groups, the algebras obtained from left- and right trivialization of the vector fields are adjoint.

Proposition 2.5. *Let $(A, [\cdot, \cdot], \triangleright)$ be a post-Lie algebra and define the product \blacktriangleright as*

$$x \blacktriangleright y := x \triangleright y + [x, y]. \quad (11)$$

Then the adjoint $(A, -[\cdot, \cdot], \blacktriangleright)$ of $(A, [\cdot, \cdot], \triangleright)$ is also a post-Lie algebra. The adjoint operation is an involution. The two adjoint post-Lie algebras generate the same ‘double bracket’ $\llbracket \cdot, \cdot \rrbracket$.

3. Universal Lie enveloping algebras of a post-Lie algebra

Recall that we consider a post-Lie algebra as two Lie algebras $\mathfrak{g} := (\mathcal{A}, [\cdot, \cdot])$ and $\bar{\mathfrak{g}} := (\mathcal{A}, \llbracket \cdot, \cdot \rrbracket)$ defined over the vector space \mathcal{A} , related via the post-Lie product (8) in Proposition 2.2

$$\llbracket x, y \rrbracket = x \triangleright y - y \triangleright x + [x, y].$$

It is natural to explore this relation on the level of universal Lie enveloping algebras. This is analogous to the approach in [27], where the universal Lie enveloping algebra of a pre-Lie algebra was studied. In [27] it was shown that the universal Lie enveloping algebra of the Lie algebra obtained by antisymmetrization of a pre-Lie algebra λ is isomorphic as a Hopf algebra to the symmetric algebra $\mathcal{S}(\lambda)$ on λ , equipped with a certain associative product $*$: $\mathcal{S}(\lambda) \otimes \mathcal{S}(\lambda) \rightarrow \mathcal{S}(\lambda)$ defined using the pre-Lie structure.

Let $\mathcal{P} := (\mathcal{A}, [\cdot, \cdot], \triangleright)$ be a post-Lie algebra, and $\mathcal{U}(\mathfrak{g})$ the universal enveloping algebra of the Lie algebra $\mathfrak{g} := (\mathcal{A}, [\cdot, \cdot])$. We generalize the results of [27] to universal Lie enveloping algebras built on Lie algebras related via the post-Lie relation (8).

The post-Lie product can be extended to a product on $\mathcal{U}(\mathfrak{g})$. We first extend it to a product mapping $\mathfrak{g} \otimes \mathcal{U}(\mathfrak{g}) \rightarrow \mathcal{U}(\mathfrak{g})$ and then, in Proposition 3.1, to

all of $\mathcal{U}(\mathfrak{g})$. Let $x, t_1, \dots, t_n \in \mathfrak{g}$, and define

$$x \triangleright \mathbf{1} := 0 \quad x \triangleright (t_1 \cdots t_n) := \sum_{i=1}^n t_1 \cdots t_{i-1} (x \triangleright t_i) t_{i+1} \cdots t_n.$$

Note that this extension defines a natural action of \mathcal{P} on $\mathcal{U}(\mathfrak{g})$, satisfying

$$\begin{aligned} [x, y] \triangleright K &= x \triangleright (y \triangleright K) - (x \triangleright y) \triangleright K - y \triangleright (x \triangleright K) + (y \triangleright x) \triangleright K \\ x \triangleright [K_1, K_2] &= [x \triangleright K_1, K_2] + [K_1, x \triangleright K_2] \end{aligned}$$

for $K, K_1, K_2 \in \mathcal{U}(\mathfrak{g})$ and $x, y \in \mathfrak{g}$, where $[K_1, K_2] = K_1 K_2 - K_2 K_1$.

Recall that $\mathcal{U}(\mathfrak{g})$ with concatenation as product is a non-commutative, cocommutative Hopf algebra. The coshuffle coproduct is defined for $x \in \mathfrak{g} \subset \mathcal{U}(\mathfrak{g})$ by $\Delta(x) := x \otimes \mathbf{1} + \mathbf{1} \otimes x$, and extended multiplicatively to all of $\mathcal{U}(\mathfrak{g})$. We use Sweedler's notation for the coproduct: $\Delta(T) =: T_{(1)} \otimes T_{(2)}$. The counit is denoted by $\epsilon : \mathcal{U}(\mathfrak{g}) \rightarrow k$.

The remainder of this section contains statements and some proofs analogous to the ones in [27, Section 2], slightly modified to match our setting. The omitted proofs can easily be deduced from [27].

Proposition 3.1. *Let $A, B, C \in \mathcal{U}(\mathfrak{g})$ and $x, y \in \mathfrak{g}$. There is a unique extension of the post-Lie product \triangleright from \mathfrak{g} to $\mathcal{U}(\mathfrak{g})$ given by*

$$\mathbf{1} \triangleright A = A \tag{12}$$

$$xA \triangleright y = x \triangleright (A \triangleright y) - (x \triangleright A) \triangleright y$$

$$A \triangleright BC = (A_{(1)} \triangleright B)(A_{(2)} \triangleright C). \tag{13}$$

The result can be proven by induction as in [27, Proposition 3.7]: First, (12) and (13) together lead to the formula

$$x \triangleright (t_1 \cdots t_n) = \sum_{i=1}^n t_1 \cdots t_{i-1} (x \triangleright t_i) t_{i+1} \cdots t_n,$$

and then (13) and induction on the length of B can be used to extend $A \triangleright B$ to monomials A and B .

Proposition 3.2. *Let $A, B, C \in \mathcal{U}(\mathfrak{g})$ and $x \in \mathfrak{g}$. We have*

$$A \triangleright \mathbf{1} = \epsilon(A) \tag{14}$$

$$\epsilon(A \triangleright B) = \epsilon(A)\epsilon(B) \tag{15}$$

$$\Delta(A \triangleright B) = (A_{(1)} \triangleright B_{(1)}) \otimes (A_{(2)} \triangleright B_{(2)}) \tag{16}$$

$$xA \triangleright B = x \triangleright (A \triangleright B) - (x \triangleright A) \triangleright B \tag{17}$$

$$A \triangleright (B \triangleright C) = (A_{(1)}(A_{(2)} \triangleright B)) \triangleright C \tag{18}$$

Proof. The proofs of (14), (15), (16), (17) are straightforward adaptations of the proofs in [27, Proposition 3.9]. We prove Equation (18) by induction on the length

of A .

$$\begin{aligned} xA \triangleright (B \triangleright C) &= x \triangleright (A \triangleright (B \triangleright C)) - (x \triangleright A) \triangleright (B \triangleright C) \\ &= x \triangleright (A_{(1)}(A_{(2)} \triangleright B) \triangleright C) - (x \triangleright A) \triangleright (B \triangleright C) \end{aligned} \tag{19}$$

$$= (xA_{(1)}(A_{(2)} \triangleright B)) \triangleright C + (x \triangleright (A_{(1)}(A_{(2)} \triangleright B))) \triangleright C - (x \triangleright A) \triangleright (B \triangleright C) \tag{20}$$

$$= (xA_{(1)}(A_{(2)} \triangleright B)) \triangleright C + (A_{(1)}(x \triangleright (A_{(2)} \triangleright B))) \triangleright C \tag{21}$$

$$\begin{aligned} &+ (x \triangleright A_{(1)})(A_{(2)} \triangleright B) \triangleright C - (x \triangleright A) \triangleright (B \triangleright C) \\ &= (xA_{(1)}(A_{(2)} \triangleright B)) \triangleright C + (A_{(1)}(xA_{(2)} \triangleright B)) \triangleright C \end{aligned} \tag{22}$$

$$\begin{aligned} &+ ((A_{(1)} \triangleright A_{(2)}) \triangleright B) \triangleright C + (x \triangleright A_{(1)})(A_{(2)} \triangleright B) \triangleright C - (x \triangleright A) \triangleright (B \triangleright C) \\ &= (xA_{(1)}(A_{(2)} \triangleright B)) \triangleright C + (A_{(1)}(xA_{(2)} \triangleright B)) \triangleright C + \end{aligned} \tag{23}$$

$$(A_{(1)}((x \triangleright A_{(2)}) \triangleright B)) \triangleright C + (x \triangleright A_{(1)})(A_{(2)} \triangleright B) \triangleright C - (x \triangleright A)(B \triangleright C)$$

Here (19) follows by induction, (20) from (17), (21) from (13) in Proposition 3.1, (22) from (17), and (23) from (17). We have

$$\begin{aligned} (x \triangleright A) \triangleright (B \triangleright C) &= (x \triangleright A)_{(1)}((x \triangleright A)_{(2)} \triangleright B) \triangleright C \\ &= (x \triangleright A_{(1)})(A_{(2)} \triangleright B) \triangleright C + A_{(1)}((x \triangleright A_{(2)}) \triangleright B) \triangleright C, \end{aligned}$$

so that

$$\begin{aligned} xA \triangleright (B \triangleright C) &= (xA_{(1)}(A_{(2)} \triangleright B)) \triangleright C + (A_{(1)}(xA_{(2)} \triangleright B)) \triangleright C \\ &= (xA)_{(1)}((xA)_{(2)} \triangleright B) \triangleright C. \end{aligned} \quad \blacksquare$$

Proposition 3.3. *On $(\mathcal{U}(\mathfrak{g}), \triangleright)$ the product:*

$$A * B := A_{(1)}(A_{(2)} \triangleright B) \tag{24}$$

*is associative. Moreover, $(\mathcal{U}(\mathfrak{g}), *, \Delta)$ is a Hopf algebra.*

Proof. From 3.2 we have

$$\begin{aligned} A * (B * C) &= A_{(1)} (A_{(2)} \triangleright (B_{(1)}(B_{(2)} \triangleright C))) \\ &= A_{(1)} ((A_{(2)} \triangleright B_{(1)})(A_{(3)} \triangleright (B_{(2)} \triangleright C))) \end{aligned} \tag{25}$$

$$= A_{(1)} ((A_{(2)} \triangleright B_{(1)})((A_{(3)}(A_{(4)} \triangleright B_{(2)})) \triangleright C)) \tag{26}$$

$$= A_{(1)} ((A_{(3)} \triangleright B_{(1)})((A_{(2)}(A_{(4)} \triangleright B_{(2)}))) \triangleright C) \tag{27}$$

$$= (A_{(1)}(A_{(2)} \triangleright B)) * C$$

$$= (A * B) * C,$$

where (25) follows from (13), (26) from (18), and (27) from cocommutativity. The compatibility between (24) and the coproduct Δ follows from (13) \blacksquare

Note that for a and b in \mathfrak{g} , we have $\Delta(a) = a \otimes I + I \otimes a$, so

$$a * b = a \triangleright b + ab.$$

Note also that the product $*$ can equivalently be defined in terms of the adjoint post-Lie product (11), as

$$A * B := (A_{(1)} \blacktriangleright B)A_{(2)}. \tag{28}$$

Recall that $\bar{\mathfrak{g}} := (\mathcal{A}, \llbracket \cdot, \cdot \rrbracket)$ is related to \mathfrak{g} via (8) in Proposition 2.2.

Theorem 3.4. *The Hopf algebra $(\mathcal{U}(\mathfrak{g}), *, \Delta)$ is isomorphic to the enveloping algebra $\mathcal{U}(\bar{\mathfrak{g}})$.*

This is essentially Theorem 3.14 in [27], generalized to our noncommutative setting. One first notes that the product $*$ respects the filtration of $\mathcal{U}(\mathfrak{g})$ by the length of the monomials, and that this graded product coincides with the concatenation product in each degree. The graded isomorphism $gr\mathcal{U}(\bar{\mathfrak{g}}) \rightarrow \mathcal{U}(\mathfrak{g})$ results in an isomorphism $\mathcal{U}(\bar{\mathfrak{g}}) \rightarrow (\mathcal{U}(\mathfrak{g}), *, \Delta)$. See the proof of [27, Theorem 3.14].

3.1. Planar trees and the Grossman–Larson product.

We turn to the free post-Lie algebra, which as a vector space is the free Lie algebra over the set of all planar rooted trees [21, 22, 31]. The product (24) constructed in Section 3 corresponds to the Grossman–Larson product on planar rooted trees [11]. The dual of this product is the coproduct Δ_{MKW} defined in [24], and we remark that the constructions of Section 3 provide a new perspective on the Hopf algebra H_{MKW} underlying Lie–Butcher theory.

We restate a few facts about the free post-Lie algebra. More details can be found in [22]. The free post-Lie algebra

$$\mathfrak{g}_F = \text{PostLie}(\bullet)$$

on one generator is the free Lie algebra over the set OT of planar rooted trees, equipped with a post-Lie product \curvearrowright called *left grafting*. The first few elements of OT are

$$\left\{ \bullet, \begin{array}{c} \bullet \\ | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \quad \bullet \\ | \quad | \quad | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \quad \bullet \quad \bullet \\ | \quad | \quad | \quad | \\ \bullet \end{array}, \dots \right\}$$

The left grafting operation on two trees $\tau_1 \curvearrowright \tau_2$ is the sum of all trees resulting from attaching the root of τ_1 to all the nodes of τ_2 from the left.

$$\begin{array}{c} \bullet \\ | \\ \circ \end{array} \curvearrowright \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \end{array} = \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \circ \end{array} + \begin{array}{c} \bullet \\ | \\ \bullet \end{array} + \begin{array}{c} \bullet \\ | \\ \bullet \end{array}.$$

On brackets the operation acts as a derivation from the left, and as a difference of associators from the right. See [22, Proposition 3.1].

The post-Lie operation \curvearrowright has a unique extension to the universal enveloping algebra $\mathcal{U}(\mathfrak{g}_F)$ by Proposition 3.1. The elements of $\mathcal{U}(\mathfrak{g}_F)$ are the ordered finite words in the alphabet OT, including the empty word \mathbb{I} , and is called the set OF of *ordered forests*.

$$\text{OF} = \left\{ \mathbb{I}, \bullet, \bullet\bullet, \begin{array}{c} \bullet \\ | \\ \bullet \end{array}, \bullet\bullet\bullet, \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \quad \bullet \\ | \quad | \quad | \\ \bullet \end{array}, \begin{array}{c} \bullet \\ | \\ \bullet \end{array}, \dots \right\}.$$

The result of left grafting a forest $\omega = \tau_1\tau_2$ on a forest ν is a sum of words obtained by left grafting τ_2 to all the nodes in ν , then τ_1 to all the nodes not originally from τ_2 .

Using the left grafting operation one can define the important *Grossman-Larson product*:

$$\omega_1 \circ \omega_2 = B^-(\omega_1 \curvearrowright B^+(\omega_2)),$$

where B^+ produces a tree from a forest by adding a root and B^- is the inverse operation of removing a root, producing a forest.

Proposition 3.5. *In $\mathcal{U}(\mathfrak{g}_F)$, the product $*$ defined in Proposition 3.3 is the Grossman-Larson product \circ .*

In [24] the Hopf algebra H_{MKW} underlying Lie-Butcher theory and numerical integration on Lie groups and homogeneous manifolds was introduced. It was shown that it is the dual of the Hopf algebra $(\mathcal{U}(\mathfrak{g}_F), \circ, \Delta)$, and, as mentioned, the discussion in Section 3 provides an alternative point of view for H_{MKW} .

Note that all the above constructions can be generalized to multiple generators by considering *labelled* trees and forests [22]. We refer the reader to [31] for an operadic approach to post-Lie algebras.

Since it is established that \circ and \curvearrowright coincide with the general post-Lie operations $*$ respectively \triangleright , we will in the sequel use the symbols $*$ and \triangleright also on forests.

3.2. Lie-Butcher series.

We consider infinite series in the free post-Lie algebra, and the interpretation of these as vector fields and flows on manifolds. Consider $\mathcal{U}(\mathfrak{g}_F)$ as a vector space graded by the number of nodes in the forests, and let

$$\mathcal{U}(\mathfrak{g}_F)^* = \varprojlim_n (\mathcal{U}(\mathfrak{g}_F))_n$$

denote the graded completion of $\mathcal{U}(\mathfrak{g}_F)$, i.e. $\mathcal{U}(\mathfrak{g}_F)^*$ consists of all infinite sums in $\mathcal{U}(\mathfrak{g}_F)$ with the inverse limit topology, whose open sets are generated by sequences agreeing up to order n . We let the pairing $\langle \cdot, \cdot \rangle: \mathcal{U}(\mathfrak{g}_F)^* \times \mathcal{U}(\mathfrak{g}_F) \rightarrow k$ be defined such that the ordered forests form an orthonormal set

$$\langle \omega, \omega' \rangle = \begin{cases} 0 & \text{if } \omega \neq \omega' \\ 1 & \text{if } \omega = \omega'. \end{cases}$$

Following [29] we may identify $\mathcal{U}(\mathfrak{g}_F)^*$ with the graded dual of $\mathcal{U}(\mathfrak{g}_F)$.

Definition 3.6 (Universal Lie-Butcher series). A universal LB series is an element $\alpha \in \mathcal{U}(\mathfrak{g}_F)^*$. This can be written as an infinite sum

$$\alpha = \sum_{\omega \in \text{OF}} \langle \alpha, \omega \rangle \omega.$$

Of particular importance are two special subspaces of $\mathcal{U}(\mathfrak{g}_F)^*$:

Definition 3.7. The *infinitesimal characters* (primitive elements) $\mathfrak{c}_F \subset \mathcal{U}(\mathfrak{g}_F)^*$ and the *characters* (group-like elements) $\mathfrak{C}_F \subset \mathcal{U}(\mathfrak{g}_F)^*$ are defined as

$$\begin{aligned} \mathfrak{c}_F &:= \{ \alpha \in \mathcal{U}(\mathfrak{g}_F)^* \mid \Delta(\alpha) = I \otimes \alpha + \alpha \otimes I \} \\ \mathfrak{C}_F &:= \{ \alpha \in \mathcal{U}(\mathfrak{g}_F)^* \mid \Delta(\alpha) = \alpha \otimes \alpha \}, \end{aligned}$$

where Δ is the coshuffle coproduct.

Remark 3.8. The name (*infinitesimal*) *characters* originate from the identification of an infinite series $\alpha \in \mathcal{U}(\mathfrak{g}_F)^*$ with an element of the dual space, $\alpha: \mathcal{U}(\mathfrak{g}_F) \rightarrow k, \omega \mapsto \langle \alpha, \omega \rangle$. Since the right part of the pairing is always a finite linear combination, the evaluation of the pairing is always a finite sum. Computations on infinite series is always done by throwing the computation onto the finite right hand side, e.g. for $\alpha, \beta \in \mathcal{U}(\mathfrak{g}_F)^*$ we compute the composition as

$$\langle \alpha * \beta, \omega \rangle = \langle \alpha \otimes \beta, \Delta_*(\omega) \rangle,$$

where $\Delta_*: \mathcal{U}(\mathfrak{g}_F) \rightarrow \mathcal{U}(\mathfrak{g}_F) \otimes \mathcal{U}(\mathfrak{g}_F)$ is the dual of the non-planar Grossman–Larson product $*$. Thus primitive elements (resp. group-like elements) in the completion $\mathcal{U}(\mathfrak{g}_F)^*$ are infinitesimal characters (resp. characters) on the Hopf algebra H_{MKW} , having shuffle as product and Δ_* as coproduct. Computational formulas for H_{MKW} can be found in [24].

Note that Δ preserves the grading and the definitions are therefore induced from the same conditions on all finite components. Note also that \mathfrak{c}_F is the same as the graded completion of \mathfrak{g}_F . The concatenation product conc and the Grossman–Larson product $*$ defined on $\mathcal{U}(\mathfrak{g}_F)$ extend to $\mathcal{U}(\mathfrak{g}_F)^*$. We define exponentials with respect to these products for $f \in \mathcal{U}(\mathfrak{g}_F)^*$ as

$$\begin{aligned} \exp^*(f) &= 1 + f + \frac{1}{2}f * f + \frac{1}{6}f * f * f + \dots \\ \exp(f) &= 1 + f + \frac{1}{2}ff + \frac{1}{6}fff + \dots \end{aligned}$$

It can be shown that these exponentials restricted to \mathfrak{c}_F are bijections $\exp^*, \exp: \mathfrak{c}_F \rightarrow \mathfrak{C}_F$. The two Lie brackets $[[\cdot, \cdot]]$ and $[\cdot, \cdot]$ defined on \mathfrak{g}_F extend to \mathfrak{c}_F , turning the infinitesimal characters \mathfrak{c}_F into a post-Lie algebra. The characters \mathfrak{C}_F form a group with both the products $*$ and conc , with inverses $\exp^*(f) \mapsto \exp^*(-f)$ and $\exp(f) \mapsto \exp(-f)$, respectively. More explicit formulas for the inverses are given by the antipode in H_{MKW} and the antipode in the concatenation–deshuffle Hopf algebra [24]. The following characterization of the two group products will later be given concrete interpretation in terms of composition of flows on manifolds.

Proposition 3.9. *The two group products $*$ and conc on \mathfrak{C}_F are related as*

$$\exp(f) * \exp(g) = \exp(f) (\exp(f) \triangleright \exp(g)).$$

Proof. Since $\exp(f)$ is a character, $\Delta(\exp(f)) = \exp(f) \otimes \exp(f)$ and the result follows from (24). ■

We have the following characterization of the two exponentials in terms of differential equations:

Proposition 3.10. *For an infinitesimal character $f \in \mathfrak{c}_F$, the curve $y(t) = \exp^*(tf) \in \mathfrak{C}_F$ solves the differential equation*

$$y'(t) = y(t) (y(t) \triangleright f), \quad y(0) = I, \tag{29}$$

while the curve $z(t) = \exp(tf) \in \mathfrak{C}_F$ solves

$$z'(t) = z(t)f, \quad z(0) = I. \tag{30}$$

Proof. Differentiating $y(t) = \exp^*(tf) = \sum_{j=0}^{\infty} \frac{t^j}{j!} f^{j*}$, we get $y'(t) = y(t) * f$. Using $\Delta(y(t)) = y(t) \otimes y(t)$ and (24), we obtain (29). Differentiation of $z(t) = \exp(tf)$ yields $z'(t) = z(t)f$. ■

Note: For $y(t) \in \mathfrak{C}_F$, the expression $y(t) \triangleright f$ is on a manifold interpreted as the parallel transport (by the connection \triangleright) of the vector field f along the flow of $y(t)$. Thus (29) is an abstract version of (1). The reason for the opposite ordering in the abstract setting versus the concrete manifold formulation is that the abstract group \mathfrak{C}_F represents pullback of functions on a manifold, and hence the product $*$ maps contravariantly to composition of diffeomorphisms. We could alternatively use the other post-Lie product \blacktriangleright defined in Proposition 2.5, which from (28) yields $y'(t) = (y(t) \blacktriangleright f) y(t)$. However, the connection \blacktriangleright on the manifold is obtained from left trivialization, in which case (1) also takes the opposite form $y'(t) = y(t) \cdot f(y(t))$.

Several characterizations of the solution operator $f \mapsto \exp^*(tf)$ are known in the literature. For most purposes it is sufficient to characterize $\exp^*(\bullet)$, since the exponential of a general $tf \in \mathfrak{c}_F$ can be recovered from this by the *substitution law* $\bullet \mapsto tf$, see [16]. The explicit form

$$\begin{aligned} \exp^*(\bullet) = & I + \frac{1}{2!} \bullet + \frac{1}{3!} (\bullet\bullet + \bullet) + \frac{1}{4!} (\bullet\bullet\bullet + \bullet\bullet + 2\bullet\bullet + \bullet\bullet + \bullet) \\ & + \frac{1}{5!} (\bullet\bullet\bullet\bullet + \bullet\bullet\bullet + 2\bullet\bullet\bullet + 3\bullet\bullet\bullet + \bullet\bullet\bullet + \bullet\bullet\bullet + 3\bullet\bullet\bullet + 3\bullet\bullet\bullet + 3\bullet\bullet\bullet) \\ & + \frac{1}{6!} (\bullet\bullet\bullet\bullet\bullet + \dots) + \dots \end{aligned}$$

is derived with recursion formulas for the coefficients in [28] and also discussed in [15], where the coefficients are related to *non-commutative Bell polynomials* [30, 10].

The following result is of fundamental importance in many applications:

Proposition 3.11. *The two exponentials $\exp^*, \exp: \mathfrak{c}_F \rightarrow \mathfrak{C}_F$ are related as*

$$\exp^*(f) = \exp(\theta(1)),$$

where $\theta(t) \in \mathfrak{c}_F$ satisfies the differential equation

$$\theta'(t) = d \exp_{\theta(t)}^{-1} (\exp(\theta(t)) \triangleright f), \quad \theta(0) = 0, \tag{31}$$

and where $d \exp_{\theta}^{-1}$ denotes the inverse left trivialized differential of the exponential map, given as an infinite series in terms of the Bernoulli numbers B_k as

$$d \exp_{\theta}^{-1}(\theta') = \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_{\theta}^{(k)}(\theta') = \theta' + \frac{1}{2}[\theta, \theta'] + \frac{1}{12}[\theta, [\theta, \theta']] + \dots$$

Proof. By differentiating $y(t) = \exp^*(tf) = \exp(\theta(t))$ and using $d/dt \exp(\theta) = \exp(\theta)d \exp_{\theta}(\theta')$, we obtain

$$y'(t) = y(t) (y(t) \triangleright f) = y(t)d \exp_{\theta}(\theta'),$$

from which the result follows by inverting the linear operator $d \exp_{\theta}$. ■

Remark 3.12. The differential equation (31) leads to an interesting expansion of θ . A more precise description of its solution $\theta(t)$ will be given in a forthcoming article. First, we already stated above that

$$d \exp_{\theta(t)}^{-1} := \frac{\text{ad}_{\theta(t)}}{\exp(\text{ad}_{\theta(t)}) - 1} = \sum_{s \geq 0} \frac{B_s}{s!} \text{ad}_{\theta(t)}^{(s)},$$

where $\text{ad}_a^{(s)}(b) := \text{ad}_a^{(s-1)}([a, b])$. Second, $\exp(\theta(t)) \triangleright f = f + \sum_{i>0} \frac{1}{i!} (\theta(t))^i \triangleright f$. Since f is constant, we may use the ansatz $\theta(t) = \sum_{n>0} \theta_n(f)t^n$, which yields

$$d \exp_{\theta(t)}^{-1} = 1 + \sum_{s>0} t^s \sum_{j=1}^s \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=s \\ k_l>0}} \text{ad}_{\theta_{k_1}} \cdots \text{ad}_{\theta_{k_j}}$$

and

$$\exp(\theta(t)) \triangleright f = f + \sum_{i>0} \frac{t^i}{i!} \sum_{u=1}^i \sum_{\substack{k_1+\dots+k_u=i \\ k_l>0}} (\theta_{k_1} \theta_{k_2} \cdots \theta_{k_u}) \triangleright f.$$

The composition of the two leads to a rather intriguing Magnus-type expansion:

$\theta_1(f) = f$, $2\theta_2(f) = f \triangleright f$ and

$$\begin{aligned} n\theta_n(f) &= \sum_{j=1}^{n-1} \frac{1}{j!} \sum_{\substack{k_1+\dots+k_j=n-1 \\ k_i>0}} (\theta_{k_1}\theta_{k_2}\dots\theta_{k_j}) \triangleright f \\ &+ \sum_{j=1}^{n-1} \frac{B_j}{j!} \sum_{\substack{k_1+\dots+k_j=n-1 \\ k_i>0}} \text{ad}_{\theta_{k_1}} \dots \text{ad}_{\theta_{k_j}} f \\ &+ \sum_{j=2}^{n-1} \left(\left(\sum_{q=1}^{j-1} \frac{B_q}{q!} \sum_{\substack{k_1+\dots+k_q=j-1 \\ k_i>0}} \text{ad}_{\theta_{k_1}} \dots \text{ad}_{\theta_{k_s}} \right) \times \right. \\ &\left. \times \left(\sum_{p=1}^{n-j} \frac{1}{p!} \sum_{\substack{k_1+\dots+k_p=n-j \\ k_i>0}} (\theta_{k_1}\theta_{k_2}\dots\theta_{k_p}) \triangleright f \right) \right). \end{aligned}$$

3.3. Vector fields and flows on manifolds.

We briefly discuss post-Lie algebras appearing in differential geometry, providing concrete interpretations of the abstract notions discussed above.

A prime example of a post-Lie algebra in differential geometry is the set of vector fields on a Lie group, with the connection given by the (right or left) Maurer–Cartan form. We will use this example to concretize the abstract operations discussed above. This example generalizes to the concept of *post-Lie algebroids*, defined in [22] as Lie algebroids equipped with a flat connection with constant torsion.

Let $(\mathfrak{g}, [\cdot, \cdot]_{\mathfrak{g}})$ be a finite-dimensional Lie algebra with a corresponding Lie group G and write $\exp_{\mathfrak{g}}: \mathfrak{g} \rightarrow G$ for the classical Lie exponential. For example, if G is a matrix Lie group, then $\exp_{\mathfrak{g}}$ denotes the matrix exponential. Vector fields on G can be trivialized by either left or right multiplication. This gives rise to two adjoint post-Lie algebras. We choose right trivialization here.

Consider the vector bundle $\mathfrak{g} \times G \rightarrow G$, the left Lie algebra action $L: \mathfrak{g} \times G \rightarrow G$ and the *anchor map* $\lambda: \mathfrak{g} \times G \rightarrow TG$ defined as

$$L(V, x) := \exp_{\mathfrak{g}}(V) \cdot x \quad (32)$$

$$\lambda(V, x) \equiv V \cdot x := \left. \frac{\partial}{\partial t} \right|_{t=0} L(tV, x). \quad (33)$$

Let \mathfrak{g}^G denote functions from G to \mathfrak{g} and let $\mathcal{X}(G)$ denote the vector fields on G . A smooth $f \in \mathfrak{g}^G$ is identified with a section of the vector bundle $x \mapsto (f(x), x): G \rightarrow \mathfrak{g} \times G$, and the section maps to a vector field $F \in \mathcal{X}(G)$ by composition with the anchor λ ,

$$F(x) = \lambda(f(x), x) = f(x) \cdot x,$$

which we write compactly as $F = \lambda(f)$. In the case of Lie groups, the identification $f \mapsto F$ is a bijection, since f is recovered from F by right trivialization (composition of f with the right Maurer–Cartan form). For more general post-Lie

algebroids over homogeneous spaces, this identification is not injective. On \mathfrak{g}^G we define a Lie bracket and a connection

$$[f, g](x) := [f(x), g(x)]_{\mathfrak{g}}$$

$$f \triangleright g := \lambda(f)(g) \quad \Rightarrow \quad (f \triangleright g)(x) = \left. \frac{\partial}{\partial t} \right|_{t=0} g(\exp_{\mathfrak{g}}(tf(x)) \cdot x).$$

For the rest of this section we put $\mathcal{L} := (\mathfrak{g}^G, [\cdot, \cdot], \triangleright)$.

Proposition 3.13. $\mathcal{L} = (\mathfrak{g}^G, [\cdot, \cdot], \triangleright)$ is post-Lie. The anchor map λ maps the bracket $[[\cdot, \cdot]]$ defined in Proposition 2.2 to the Jacobi bracket of vector fields on G , hence $(\mathfrak{g}^G, [[\cdot, \cdot]], \lambda)$ is a Lie algebroid.

Proof. See [22]. ■

Let $\mathcal{U}(\mathcal{L})$ denote the enveloping algebra of \mathcal{L} . We can identify this with $\mathcal{U}(\mathcal{L}) = (\mathcal{U}(\mathfrak{g})^G, \text{conc})$, the set of maps from G to $\mathcal{U}(\mathfrak{g})$ where the product $\text{conc}(f, g) =: fg$ is given as

$$(fg)(x) = f(x)g(x) \quad \text{for } f, g \in \mathcal{L} \text{ and } x \in G.$$

With this product $\mathcal{U}(\mathcal{L})$ is a graded algebra and we can form the graded completion $\mathcal{U}(\mathcal{L})^*$. As in Definition 3.7 we have the infinitesimal characters $\mathfrak{c} \subset \mathcal{U}(\mathcal{L})^*$ and the characters $\mathfrak{C} \subset \mathcal{U}(\mathcal{L})^*$. The infinitesimal characters represent vectorfields on G , while the characters represent diffeomorphisms. Note, however, that these are formal series defined in the inverse topology, and will not necessarily converge in the analytical sense. In applications, the remedy for a lack of convergence is truncation of the infinite series and estimates of exponential closeness [12].

The definition of a free object implies that for any post-Lie algebra \mathcal{L} and any assignment $\mathfrak{g}_F \ni \bullet \mapsto f \in \mathcal{L}$, there exists a unique post-Lie morphism $\mathbb{F}_f: \mathfrak{g}_F \rightarrow \mathcal{L}$ such that $\mathbb{F}_f(\bullet) = f$. This extends to a unique morphism $\mathbb{F}_f: \mathcal{U}(\mathfrak{g}_F) \rightarrow \mathcal{U}(\mathcal{L})$, see [24, 22].

Therefore, for any universal Lie–Butcher series there corresponds an infinite series we call a *Lie–Butcher series in \mathcal{L}*

$$\mathbb{B}_f(\alpha) := \sum_{\omega \in \text{OF}} \langle \alpha, \omega \rangle \mathbb{F}_f(\omega).$$

For $\alpha \in \mathfrak{c}_F$ we have $\mathbb{B}_f(\alpha) \in \mathfrak{c}$ and for $\alpha \in \mathfrak{C}_F$ we have $\mathbb{B}_f(\alpha) \in \mathfrak{C}$.

The post-Lie algebra \mathcal{L} acts on $\mathcal{F}(G)$, the ring of scalar functions on G , via a derivation defined as

$$f \triangleright \phi := \lambda(f)(\phi), \quad \text{for } f \in \mathcal{L}, \phi \in \mathcal{F}(G).$$

The derivation satisfies, for all $f, g \in \mathcal{L}$ and $\phi, \tilde{\phi} \in \mathcal{F}(G)$,

$$f \triangleright (\phi \tilde{\phi}) = (f \triangleright \phi) \tilde{\phi} + \phi (f \triangleright \tilde{\phi})$$

$$[f, g] \triangleright \phi = f \triangleright (g \triangleright \phi) - (f \triangleright g) \triangleright \phi - g \triangleright (f \triangleright \phi) + (g \triangleright f) \triangleright \phi. \tag{34}$$

Equation (34) is equivalent to

$$f \triangleright (g \triangleright \phi) - g \triangleright (f \triangleright \phi) = \llbracket f, g \rrbracket \triangleright \phi = (f * g - g * f) \triangleright \phi.$$

The post-Lie action is extended to $\triangleright: \mathcal{U}(\mathcal{L}) \times \mathcal{F}(G) \rightarrow \mathcal{F}(G)$ as in Proposition 3.1, where the rightmost elements in the equations are taken from $\mathcal{F}(G)$.

An important issue is the identification of the exponentials $\exp^*(f)$ and $\exp(f)$, as diffeomorphisms on the domain G . Since the product $*$ in $\mathcal{U}(\mathcal{L})$ models the composition of differential operators (Lie derivations on G), it follows from elementary Lie theory that exponentials act as pullback on the function ring:

Proposition 3.14. *Let $\psi_{tf}: G \rightarrow G$ denote the time- t flow of the vector field $\lambda(f) \in \mathcal{X}(G)$ and $\psi_{tf}^{\leftarrow} \phi$ the pullback of a scalar function $\phi \in \mathcal{F}(G)$, defined as $(\psi_{tf}^{\leftarrow} \phi)(x) = \phi(\psi_{tf}(x))$. Then*

$$\psi_{tf}^{\leftarrow} \phi = \exp^*(tf) \triangleright \phi.$$

Proof. From $f \triangleright \phi := \left. \frac{\partial}{\partial t} \right|_{t=0} \psi_{tf}^{\leftarrow} \phi$ it follows that $\left. \frac{\partial}{\partial t} \right|_{t=0} \psi_{tf}^{\leftarrow} \phi = \psi_{tf}^{\leftarrow} (f \triangleright \phi)$. Iterating this, we find the Taylor series of the pullback

$$\begin{aligned} \psi_{tf}^{\leftarrow} \phi &= \phi + tf \triangleright \phi + \frac{t^2}{2} f \triangleright (f \triangleright \phi) + \frac{t^3}{3!} f \triangleright (f \triangleright (f \triangleright \phi)) + \dots \\ &= \phi + tf \triangleright \phi + \frac{t^2}{2} (f * f) \triangleright \phi + \frac{t^3}{3!} (f * f * f) \triangleright \phi + \dots = \exp^*(tf) \triangleright \phi. \quad \blacksquare \end{aligned}$$

Note that the pullbacks compose in a contravariant fashion

$$\psi_g^{\leftarrow} \circ \psi_f^{\leftarrow} = (\psi_f \circ \psi_g)^{\leftarrow}.$$

This is why the symbolic differential equation (30) appears in opposite order compared to the manifold equation (1).

The flow $\psi_{tf}: G \rightarrow G$ can be found from $\exp^*(f) \triangleright \phi$, by choosing ϕ as coordinate maps on G , and we obtain a well-defined action of the post-Lie characters $\exp^*(f)$ on G

$$\exp^*(tf) \cdot x := \psi_{tf}(x) \quad \text{for } x \in G. \quad (35)$$

Note that this is a right action: $\exp^*(f) \cdot (\exp^*(g) \cdot x) = (\exp^*(g) * \exp^*(f)) \cdot x$.

To understand the geometric difference between $\exp^*(f)$ and $\exp(f)$ we define *frozen sections*. For any $f \in \mathfrak{g}^G$ and $x_0 \in G$ there exist an $f_{x_0} \in \mathfrak{g}^G$ called *f frozen at x_0* , defined as

$$f_{x_0}(x) = f(x_0) \quad \text{for all } x \in G.$$

The freezing extends to all of $\mathcal{U}(\mathcal{L})$ as

$$(fg)_{x_0} := f_{x_0} g_{x_0} \quad \text{for } f, g \in \mathcal{L}.$$

It is easy to verify that the freezing operation $f \mapsto f_{x_0}$ satisfies the following relations for all $f, g \in \mathcal{L} \simeq \mathfrak{g}^G$:

$$g \triangleright f_{x_0} = 0 \quad \text{for all } g \in \mathfrak{g}^G \tag{36}$$

$$(f_{x_0})_{x_1} = f_{x_0} \tag{37}$$

$$(f \triangleright g)_{x_0} = (f_{x_0} \triangleright g)_{x_0} \tag{38}$$

$$[f, g]_{x_0} = [f_{x_0}, g_{x_0}]. \tag{39}$$

In particular (36) implies that $f_{x_0} \triangleright f_{x_0} = 0$, thus a frozen section is invariant under parallel transport by itself. Equations (37)-(38) show that $f \mapsto f_{x_0}$ is projection onto the frozen sections tangent to f at x_0 , and (39) shows that the torsion bracket is defined fiber-wise on the sections.

Recall that a geodesic of a connection (geodesic of a covariant derivative) is a curve defined such that parallel transport of a tangent to the curve along the curve is invariant, e.g. the flow of f_{x_0} is a geodesic of the connection because it satisfies the infinitesimal condition of invariance under parallel transport $f_{x_0} \triangleright f_{x_0} = 0$. Hence the flow of a frozen section f_{x_0} is a geodesic of the connection, which is tangent to f at the point x_0 . The concatenation exponential $\exp(f)$ models the exact flow of a geodesic tangent to f :

Proposition 3.15. *For all $f \in \mathcal{L}$ we have*

$$\exp(f)_{x_0} = \exp^*(f_{x_0}), \tag{40}$$

the action of the character $\exp(f)$ on G , defined in (35), is given as

$$\exp(f) \cdot x_0 = \exp_{\mathfrak{g}}(f(x_0)) \cdot x_0$$

and parallel transport (pullback) of any section $A \in \mathcal{U}(\mathcal{L}) \simeq \mathcal{U}(\mathfrak{g})^G$ along the tangent geodesic to f at x_0 is given as

$$(\exp(f) \triangleright A)(x_0) = A(\exp_{\mathfrak{g}}(f(x_0)) \cdot x_0).$$

Proof. Since $f_{x_0} * f_{x_0} = f_{x_0} f_{x_0} + f_{x_0} \triangleright f_{x_0} = f_{x_0} f_{x_0}$, we obtain (40). From (32)-(33) it follows that the geodesic tangent to f in x_0 is

$$\exp(tf_{x_0}) \cdot x_0 = \exp_{\mathfrak{g}}(tf(x_0)) \cdot x_0.$$

The pullback formula follows similarly to the proof of Proposition 3.14. ■

3.4. Vector fields and flows on \mathbb{R}^n .

The post-Lie algebra on a Lie group described in the previous section becomes particularly simple in the euclidean case \mathbb{R}^n . As a Lie algebra we have $G = \mathfrak{g} = \mathbb{R}^n$ and $\exp_{\mathfrak{g}}(f) = f$. The canonical connection on \mathbb{R}^n is given as

$$\left(f^i \frac{\partial}{\partial x^i} \right) \triangleright \left(g^j \frac{\partial}{\partial x^j} \right) := f^i \frac{\partial g^j}{\partial x^i} \frac{\partial}{\partial x^j},$$

which is a flat connection with zero torsion $[f, g] = 0$, and hence defines a pre-Lie algebra on the set of vector fields. In this case Proposition 3.15 becomes

Proposition 3.16. *The action of the character $\exp(f)$ on \mathbb{R}^n , defined in (35), is*

$$\exp(f) \cdot x_0 = f(x_0) + x_0,$$

and parallel transport of a section $A \in \mathcal{U}(\mathcal{L}) \simeq \mathcal{U}(\mathfrak{g})^G$ is given as

$$(\exp(f) \triangleright A)(x_0) = A(f(x_0) + x_0).$$

4. Numerical integration on post-Lie algebras

Lie group integrators generalize traditional numerical integrators of differential equations from vector spaces to more general manifolds. Since the initial developments of these methods in the 1990s, it has been clear that two different Lie algebras are important for their formulation and analysis. The standard formulation of Lie group integrators is based on the full Lie algebra of all vector fields on a Lie group and the sub-algebra of (right or left) invariant vector fields. The algorithms are formulated by defining the operation of *freezing* a general vector field at a given point on the group, which means replacing a general vector field by an invariant vector field tangent at the given point. Basic motions used to formulate the methods are obtained by exponentiating this ‘frozen’ vector field. A problem with the standard formulation is that it is difficult to give an abstract algebraic meaning to the operation of freezing a general vector field at a point. It is not an operation that can be defined on an abstract post-Lie algebra, without imposing additional structure. For this reason Lie group methods and classical Runge–Kutta methods have so far always been formulated in a concrete setting of vector fields on a manifold (or vector space).

In the present formulation, we follow a different but equivalent approach. Recall that post-Lie algebras involve two Lie algebras, $\bar{\mathfrak{g}}$ and \mathfrak{g} , where \mathfrak{g} is not a sub-algebra of $\bar{\mathfrak{g}}$. The two are defined over the same set, with two different Lie brackets and two different enveloping algebras. Hence we can define two different exponential maps: with respect to the product in $\mathcal{U}(\bar{\mathfrak{g}})$ and with respect to the product in $\mathcal{U}(\mathfrak{g})$. We have seen that in the case of vector fields on a Lie group, exponentiation of a general vector field using the product in $\mathcal{U}(\mathfrak{g})$ is equivalent to computing the flow of a vector field frozen at a point, whereas the exponential with respect to the product in $\mathcal{U}(\bar{\mathfrak{g}})$ corresponds to following the exact flow of the vector field.

This reformulation provides a fresh and fruitful point of view. The algorithms are formulated without the process of evaluating vector fields at a given point, and without the freezing approximation. As a result the methods can be abstracted to algorithms on pre- and post-Lie algebras. The evaluation at a given point on a manifold reappears only when the abstract algorithm is interpreted in the concrete setting of flows on manifolds. The formulation and analysis of the methods, on the other hand, can be pursued entirely at an abstract algebraic level, without involving the point-evaluation operation.

In the sequel we will formulate Runge–Kutta methods and Lie group integrators abstractly on pre-Lie and post-Lie algebras. As a byproduct we present a surprisingly short derivation of the classical Butcher order theory for Runge–Kutta

methods on pre-Lie algebras. We hope that this will pave the way for a simplified exposition and analysis of general classes of Lie group integrators.

4.1. Integration on post-Lie algebras.

We let $\mathcal{L} = (\mathcal{L}, [\cdot, \cdot], \triangleright)$ denote a post-Lie algebra, with infinitesimal characters $\mathfrak{c} \subset \mathcal{U}(\mathcal{L})^*$, characters $\mathfrak{C} \subset \mathcal{U}(\mathcal{L})^*$ and exponentials $\exp, \exp^*: \mathfrak{c} \rightarrow \mathfrak{C}$. We have seen in the example of vector fields over a Lie group that the map $f \mapsto \exp^*(f)$ can be interpreted as the solution operator of a differential equation with vector field f , whereas $\exp(f)$ is interpreted as the flow of a *frozen* vector field and $\exp(f) \triangleright g$ is the parallel transport of g along a frozen vector field. The flow and parallel transport along frozen vector fields, as well as the computation of commutators $[\cdot, \cdot]$ in \mathfrak{g} , are assumed to be basic operations that can be computed exactly. The basic approximation problem of Lie group integration can be stated in a post-Lie algebra as follows:

Problem 4.1 (The fundamental problem of numerical integration). Given a post-Lie algebra \mathcal{L} , approximate the exponential $\exp^*: \mathfrak{c} \rightarrow \mathfrak{C}$ in terms of linear combinations in \mathcal{L} , the bracket $[\cdot, \cdot]$ on \mathcal{L} , the exponential $\exp: \mathfrak{c} \rightarrow \mathfrak{C}$, products of such exponentials, and parallel transport of the form $\exp(f) \triangleright g$ for $f, g \in \mathcal{L}$.

If \mathcal{L} is a post-Lie algebra, a solution to this approximation problem yields a Lie group integration algorithm. If \mathcal{L} is a pre-Lie algebra, then $[\cdot, \cdot] = 0$ and solutions of the approximation problem are classical integrators.

The simplest reasonable solution to this problem is Euler's method, defined as the approximation

$$\Psi_e(f) := \exp(f).$$

For vector fields on a Lie group, this spells out more explicitly as

$$x_{n+1} = \Psi_e(hf) \cdot x_n = \exp(hf) \cdot x_n = \exp_{\mathfrak{g}}(hf(x_n)) \cdot x_n,$$

where $h \in \mathbb{R}$ is the stepsize, $f \in \mathfrak{g}^G$ a map from the Lie group G to the Lie algebra \mathfrak{g} , and $\exp_{\mathfrak{g}}: \mathfrak{g} \rightarrow G$ denotes the classical Lie exponential. For vector fields on \mathbb{R}^n (pre-Lie case), this is the classical Euler method

$$x_{n+1} = \Psi_e(hf) \cdot x_n = \exp(hf) \cdot x_n = x_n + hf(x_n),$$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Since $f * f = ff + f \triangleright f$, we find $\exp^*(f) - \exp(f) = \frac{1}{2}f \triangleright f + \dots$. Thus, with a stepsize $h \rightarrow 0$, we have $\Psi_e(hf) - \exp^*(hf) = \mathcal{O}(h^2)$, i.e. Euler's method is a *first order approximation*. Runge–Kutta (RK) methods define higher order approximations.

Definition 4.2 (Runge–Kutta method (RK)). An s -stage RK method $\Psi_{RK}(f)$

is defined in terms of $s^2 + s$ real coefficients $\{a_{ij}\}_{i,j=1}^s$, $\{b_i\}_{i=1}^s$ as follows

$$K_i = \exp\left(\sum_{j=1}^s a_{ij}K_j\right) \triangleright f, \quad i = 1, \dots, s,$$

$$\Psi_{RK}(f) = \exp\left(\sum_{j=1}^s b_jK_j\right),$$

where $f, K_i \in \mathcal{L}$ and $\Psi_{RK}(f) \in \mathfrak{C} \subset \mathcal{U}(\mathcal{L})^*$. If a_{ij} is a strictly lower triangular matrix the method is called *explicit*, otherwise K_i are found by solving implicit equations.

In the pre-Lie case of vector fields on \mathbb{R}^n the methods $x_{n+1} = \Psi_{RK}(hf) \cdot x_n$ become classical RK-methods:

$$K_i = hf\left(x_n + \sum_{j=1}^s a_{ij}K_j\right), \quad i = 1, \dots, s,$$

$$x_{n+1} = x_n + \sum_{j=1}^s b_jK_j,$$

Classical Runge–Kutta methods can obtain arbitrary high order, provided the coefficients $\{a_{ij}\}_{i,j=1}^s$, $\{b_i\}_{i=1}^s$ satisfy Butcher's order conditions up to the given order, see below.

In the general post-Lie setting the methods Ψ_{RK} are in general at most second order exact, see [19]. There are several ways of obtaining higher order methods in a general post-Lie algebra [19, 28, 9]. As an example, the RKMK class of methods [19] are of order p on arbitrary post-Lie algebras, provided the coefficients satisfy the classical Butcher order conditions up to order p .

Definition 4.3 (RKMK methods). We define $f \mapsto \Psi_{\text{RKMK}}(f): \mathfrak{c} \rightarrow \mathfrak{C}$ as:

$$\begin{aligned} &\text{for } i = 1, \dots, s \\ &U_i = \sum_j a_{ij}K_j \\ &K_i = d \exp_{U_i}^{-1}(\exp(U_i) \triangleright f) \\ &\text{end} \\ \Psi_{\text{RKMK}}(f) &:= \exp\left(\sum_i b_iK_i\right), \end{aligned}$$

where $U_i, K_i \in \mathfrak{c}$ and where $d \exp^{-1}: \mathfrak{c} \times \mathfrak{c} \rightarrow \mathfrak{c}$ is the inverse differential of \exp , given as an infinite series with Bernoulli numbers B_k as coefficients

$$d \exp_U^{-1}(V) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \text{ad}_U^{(k)}(V) = V - \frac{1}{2}[U, V] + \frac{1}{12}[U, [U, V]] + \dots$$

The expansion of $d \exp^{-1}$ can be truncated to the order of the underlying RK method.

4.2. Pre-Lie algebras and Butcher’s order theory.

In this section \mathcal{L} is a pre-Lie algebra, and $\mathcal{U}(\mathcal{L}) = (\mathcal{S}(\mathfrak{g}), \text{conc}, \triangleright)$, where conc is now the symmetric product, written as a commutative concatenation. We briefly recall the basic definitions of B -series, arising from the general case of post-Lie algebras and Lie–Butcher series discussed above in the special case where $[\cdot, \cdot] \equiv 0$. We then present a short outline of Butcher’s order theory for (classical) Runge–Kutta methods [2, 3] in the setting of pre-Lie algebras. We remark that very recently B -series have been characterized geometrically as a unique and universal Taylor expansion of families of local mappings which preserve all affine symmetries between the affine spaces $\{\mathbb{R}^n\}_{n=1}^\infty$, [25, 23].

Due to the pre-Lie relation (7), the free pre-Lie algebra [8] is spanned by non-planar rooted trees (where the ordering of the branches is neglected). Let T denote the (infinite) alphabet of non-planar rooted trees

$$T = \left\{ \bullet, \begin{array}{c} \bullet \\ | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \quad \bullet \\ | \quad | \quad | \\ \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \quad \bullet \end{array}, \begin{array}{c} \bullet \quad \bullet \quad \bullet \\ | \quad | \quad | \\ \bullet \quad \bullet \quad \bullet \end{array}, \dots \right\},$$

and $\mathcal{T} = \text{Vec}_{\mathbb{R}}(T)$ the \mathbb{R} -vector space spanned by finite linear combinations of elements in T . The pre-Lie product $\triangleright: \mathcal{T} \otimes \mathcal{T} \rightarrow \mathcal{T}$ is defined in terms of the grafting of rooted trees, e.g.

$$\begin{array}{c} \bullet \\ | \\ \bullet \end{array} \triangleright \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \end{array} = \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \bullet \\ | \\ \bullet \quad \bullet \end{array} + \begin{array}{c} \bullet \\ | \\ \bullet \quad \bullet \end{array} = \begin{array}{c} \bullet \quad \bullet \\ | \quad | \\ \bullet \quad \bullet \end{array} + 2 \begin{array}{c} \bullet \\ | \\ \bullet \quad \bullet \end{array}.$$

The free pre-Lie algebra (in one generator) is $\mathfrak{g}_F = \{\mathcal{T}, \triangleright\}$.

Let F denote the set of forests, i.e. all finite words in letters from T , where the ordering of the letters in a word is neglected. The empty word is $\mathbb{I} \in F$. We write $\mathcal{F} = \text{Vec}_{\mathbb{R}}(F)$ for the \mathbb{R} -vector space spanned by elements in F . With the symmetric product of forests we have $\mathcal{F} = \mathcal{S}(\mathcal{T}) = \mathcal{U}(\mathfrak{g}_F)$. The pre-Lie product \triangleright extends uniquely from \mathcal{T} to \mathcal{F} . Note that F and T are in natural 1–1 correspondence via the operations of adding and removing roots, i.e., $B^+: \mathcal{F} \rightarrow \mathcal{T}$ and $B^-: \mathcal{T} \rightarrow \mathcal{F}$.

Let $\mathcal{F}^* = \text{Lin}(\mathcal{F}, \mathbb{R})$ be the graded completion of \mathcal{F} , consisting of all infinite \mathbb{R} -linear combinations of trees and forests. The pairing $\langle \cdot, \cdot \rangle: \mathcal{F}^* \times \mathcal{F} \rightarrow \mathbb{R}$ is defined for $\omega, \omega' \in F$ such that

$$\langle \omega, \omega' \rangle = \begin{cases} \sigma(\omega) & \text{if } \omega = \omega' \\ 0 & \text{else,} \end{cases}$$

where the integer $\sigma(\omega)$ counts the size of the symmetry group of the forest ω . It is defined by

$$\begin{aligned} \sigma(\mathbb{I}) &= 1, \\ \sigma(t_1 t_2 \cdots t_k) &= \sigma(t_1) \cdots \sigma(t_k) \mu_1! \mu_2! \cdots \mu_k!, \quad \text{for all } t_1, \dots, t_k \in T, \\ \sigma(\omega \triangleright \bullet) &= \sigma(\omega), \quad \text{for all } \omega \in F, \end{aligned}$$

where $\mu_1, \mu_2, \dots, \mu_k$ count the number of equal trees for each of the different shapes among t_1, \dots, t_k . Thus the symmetry factor counts the number of planar forests

which are identified by the pre-Lie relation, see [24] for a detailed discussion of this factor.

We define a B -series as an element $\alpha \in \mathcal{F}^*$. It can be represented as an infinite series

$$\alpha = \sum_{\omega \in F} \frac{\langle \alpha, \omega \rangle}{\sigma(\omega)} \omega.$$

In this case the infinitesimal characters and characters are defined as:

Definition 4.4 (Character). A B -series $\alpha \in \mathfrak{C} \subset \mathcal{F}^*$ is a character if and only if $\langle \alpha, \mathbb{I} \rangle = 1$ and for all $\omega, \omega' \in F$

$$\langle \alpha, \omega\omega' \rangle = \langle \alpha, \omega \rangle \langle \alpha, \omega' \rangle.$$

Definition 4.5 (Infinitesimal character). A B -series $\alpha \in \mathfrak{c} \subset \mathcal{F}^*$ is an infinitesimal character if and only if $\langle \alpha, \mathbb{I} \rangle = 0$ and for all $\omega, \omega' \in F \setminus \{\mathbb{I}\}$

$$\langle \alpha, \omega\omega' \rangle = 0.$$

Remark 4.6. Infinitesimal characters are expressible in terms of an infinite series in trees

$$\alpha \in \mathfrak{c} \Leftrightarrow \alpha = \sum_{\tau \in T} \frac{\langle \alpha, \tau \rangle}{\sigma(\tau)} \tau,$$

thus $\mathfrak{c} = \mathcal{T}^*$, the dual of \mathcal{T} . Most authors reserve the term ‘ B -series’ for the infinitesimal characters. However, in that case the exponentials do not map B -series to B -series, and the distinction between the infinitesimal (vector fields) and the finite (flows and numerical methods) becomes obscured. In fact, several results in the literature actually exploit character properties of B -series, often without naming them as such, while still using the term B -series. Consequently, we choose to extend the definition of B -series to all of \mathcal{F}^* .

We will derive the classical order conditions of Runge–Kutta methods due to Butcher [2, 3]. To simplify the discussion, we define operations on s -fold tuples. Let $\mathcal{F}_s := \mathcal{F} \times \mathcal{F} \times \dots \times \mathcal{F}$ and define $\mathfrak{c}_s, \mathfrak{C}_s, \mathcal{T}_s$ and \mathcal{F}_s^* similarly. Define the s -fold exponential $\exp_s: \mathfrak{c}_s \rightarrow \mathcal{F}_s^*$, s -fold pairing $\langle \cdot, \cdot \rangle_s: \mathcal{F}_s^* \times \mathcal{F}_s \rightarrow \mathbb{R}^s$ and s -fold grafting $\triangleright_s: \mathcal{F}_s^* \times \mathcal{F} \rightarrow \mathcal{F}_s^*$ componentwise, that is, for $\mathbf{K} := (K_1, \dots, K_s) \in \mathcal{F}_s^*$

$$\begin{aligned} \exp_s(\mathbf{K}) &= \exp_s((K_1, \dots, K_s)) := (\exp(K_1), \dots, \exp(K_s)) \\ \langle \mathbf{K}, \omega \rangle_s &= \langle (K_1, \dots, K_s), \omega \rangle_s := (\langle K_1, \omega \rangle, \dots, \langle K_s, \omega \rangle) \\ \mathbf{K} \triangleright_s \omega &= (K_1, \dots, K_s) \triangleright_s \omega := (K_1 \triangleright \omega, \dots, K_s \triangleright \omega). \end{aligned}$$

For a square matrix $A = (a_{ij}) \in \mathbb{R}^{s \times s}$ and a line vector $b = (b_i) \in \mathbb{R}^{1 \times s}$ we define linear maps $A: \mathcal{F}_s^* \rightarrow \mathcal{F}_s^*$ and $b: \mathcal{F}_s^* \rightarrow \mathcal{F}^*$ as

$$\begin{aligned} A\mathbf{K} &= A(K_1, \dots, K_s) := \left(\sum_{j=1}^s a_{1j}K_j, \sum_{j=1}^s a_{2j}K_j, \dots, \sum_{j=1}^s a_{sj}K_j \right) \\ b\mathbf{K} &= b(K_1, \dots, K_s) := \sum_{j=1}^s b_j K_j. \end{aligned}$$

We now define Runge–Kutta (RK) methods using this algebraic setting.

Definition 4.7 (Runge–Kutta method). A RK method applied to $\bullet \in \mathfrak{c}$ is

$$\begin{aligned} \mathbf{K} &= \exp_s(A\mathbf{K}) \triangleright_s \bullet \\ \Psi_{RK}(\bullet) &= \exp(b\mathbf{K}), \end{aligned}$$

where $\mathbf{K} \in \mathfrak{c}_s$.

This yields a map $\Psi_{RK}: \mathfrak{c} \rightarrow \mathfrak{C}$, and in particular the *Runge–Kutta character* $\Psi_{RK}(\bullet) \in \mathfrak{C}$. We want to compare the RK-character with the *exact solution character* $\exp^*(\bullet) \in \mathfrak{C}$, explicitly given as

$$\begin{aligned} \exp^*(\bullet) &= \sum_{k=0}^{\infty} \frac{1}{k!} \bullet^{k*} = \mathbb{I} + \bullet + \frac{1}{2} \bullet * \bullet + \frac{1}{6} \bullet * * \bullet + \dots \\ &= \mathbb{I} + \bullet + \frac{1}{2} (\bullet\bullet + \bullet\bullet) + \frac{1}{6} \left(\bullet\bullet\bullet + 3\bullet\bullet\bullet + \bullet\bullet\bullet + \bullet\bullet\bullet \right) + \dots, \end{aligned}$$

Lemma 4.8. The exact solution character $\exp^*(\bullet) \in \mathfrak{C}$ is given as

$$\langle \exp^*(\bullet), \tau \rangle = \frac{1}{\tau!} \quad \text{for all } \tau \in T, \tag{41}$$

where $\tau!$ denotes the tree factorial

$$\tau! = |\tau| \tau_1! \tau_2! \dots \tau_p! \quad \text{for all } \tau = B^+(\tau_1 \tau_2 \dots \tau_p) \in T.$$

Proof. We prove this by induction, using $\bullet * \bullet = \bullet\bullet + \bullet \triangleright \bullet$ and the fact that $\exp^*(\bullet)$ is a character. Assuming (41) holds for all $|\tau'| < k$, we find for $|\tau| = k$, $\tau = B^+(\tau_1 \dots \tau_p)$ that

$$\begin{aligned} \langle \exp^*(\bullet), \tau \rangle &= \frac{1}{k!} \langle \bullet^{k*}, \tau \rangle = \frac{1}{k!} \langle \bullet^{(k-1)*} * \bullet, \tau \rangle = \frac{1}{k!} \langle \bullet^{(k-1)*} \triangleright \bullet, \tau \rangle \\ &= \frac{1}{k} \frac{1}{(k-1)!} \langle \bullet^{(k-1)*}, \tau_1 \dots \tau_p \rangle = \frac{1}{k} \langle \exp^*(\bullet), \tau_1 \dots \tau_p \rangle \\ &= \frac{1}{k} \frac{1}{\tau_1!} \dots \frac{1}{\tau_p!}. \end{aligned}$$

Together with $\langle \exp^*(\bullet), \mathbb{I} \rangle = 1$, the result follows. ■

Lemma 4.9. The Runge–Kutta character is given by the recursions

$$\langle \Psi_{RK}(\bullet), \tau \rangle = b \langle \mathbf{K}, \tau \rangle_s \tag{42}$$

$$\langle \mathbf{K}, \bullet \rangle_s = \mathbf{1} := (1, 1, \dots, 1)^T \in \mathbb{R}^s \tag{43}$$

$$\langle \mathbf{K}, B^+(\tau_1 \tau_2 \dots \tau_p) \rangle_s = A \langle \mathbf{K}, \tau_1 \rangle_s \cdot A \langle \mathbf{K}, \tau_2 \rangle_s \dots A \langle \mathbf{K}, \tau_p \rangle_s, \tag{44}$$

where $b: \mathbb{R}^s \rightarrow \mathbb{R}$, $A: \mathbb{R}^s \rightarrow \mathbb{R}^s$ and $\cdot: \mathbb{R}^s \times \mathbb{R}^s \rightarrow \mathbb{R}^s$ denotes pointwise product of vectors (known as Hadamard or Schur product).

Proof. From the definition of the s-fold pairing we find

$$\begin{aligned} \langle b\mathbf{K}, \tau \rangle &= b\langle \mathbf{K}, \tau \rangle_s \\ \langle A\mathbf{K}, \tau \rangle_s &= A\langle \mathbf{K}, \tau \rangle_s. \end{aligned}$$

We also verify that an s-fold character $\mathbf{C} = (C_1, \dots, C_s) \in \mathfrak{C}_s$ satisfies

$$\langle \mathbf{C}, \tau_1\tau_2 \cdots \tau_p \rangle_s = \langle \mathbf{C}, \tau_1 \rangle_s \cdot \langle \mathbf{C}, \tau_2 \rangle_s \cdots \langle \mathbf{C}, \tau_p \rangle_s,$$

where $\cdot : \mathbb{R}^s \times \mathbb{R}^s \rightarrow \mathbb{R}^s$ denotes pointwise product of vectors in \mathbb{R}^s . For an infinitesimal character $K \in \mathfrak{c}$ we have the single exponential $\langle \exp(K), \mathbb{I} \rangle = 1$ and $\langle \exp(K), \tau \rangle = \langle K, \tau \rangle$. Thus for s-fold exponentials we have

$$\begin{aligned} \langle \exp_s(\mathbf{K}), \mathbb{I} \rangle_s &= \mathbf{1} \\ \langle \exp_s(\mathbf{K}), \tau \rangle_s &= \langle K, \tau \rangle_s. \end{aligned}$$

Since $\langle \exp(K) \triangleright \bullet, B^+(\tau_1\tau_2 \cdots \tau_p) \rangle = \langle \exp(K), \tau_1\tau_2 \cdots \tau_p \rangle$, we find

$$\begin{aligned} \langle \mathbf{K}, B^+(\tau_1\tau_2 \cdots \tau_p) \rangle_s &= \langle \exp_s(A\mathbf{K}) \triangleright_s \bullet, B^+(\tau_1\tau_2 \cdots \tau_p) \rangle_s \\ &= \langle \exp_s(A\mathbf{K}), \tau_1\tau_2 \cdots \tau_p \rangle_s \\ &= \langle \exp_s(A\mathbf{K}), \tau_1 \rangle_s \cdots \langle \exp_s(A\mathbf{K}), \tau_p \rangle_s \\ &= A\langle \mathbf{K}, \tau_1 \rangle_s \cdots A\langle \mathbf{K}, \tau_p \rangle_s, \end{aligned}$$

establishing (44). Equations (42)–(43) are verified in a similar way. ■

We have established a classical result in a pre-Lie setting:

Theorem 4.10. *A Runge–Kutta method in a pre-Lie algebra has order p if for all $\tau \in T$ such that $|\tau| \leq p$ we have*

$$\langle \Psi_{RK}(\bullet), \tau \rangle = \frac{1}{\tau!},$$

where $\Psi_{RK}(\bullet)$ is given by (42)–(44).

The conditions up to order 4 are (in compact and componentwise notation):

τ	Compact	Componentwise
\bullet	$b\mathbf{1} = 1$	$\sum_i b_i = 1$
\bullet	$bA\mathbf{1} = \frac{1}{2}$	$\sum_{ij} b_i a_{i,j} = \frac{1}{2}$
\blacktriangledown	$b(A\mathbf{1} \cdot A\mathbf{1}) = \frac{1}{3}$	$\sum_{ijk} b_i a_{ij} a_{ik} = \frac{1}{3}$
\bullet	$bA^2\mathbf{1} = \frac{1}{6}$	$\sum_{ijk} b_i a_{ij} a_{jk} = \frac{1}{6}$
\blacktriangledown	$b(A\mathbf{1} \cdot A\mathbf{1} \cdot A\mathbf{1}) = \frac{1}{4}$	$\sum_{ijkl} b_i a_{ij} a_{ik} a_{il} = \frac{1}{4}$
\blacktriangledown	$b(A\mathbf{1} \cdot A^2\mathbf{1}) = \frac{1}{8}$	$\sum_{ijkl} b_i a_{ij} a_{ik} a_{kl} = \frac{1}{8}$
\blacktriangledown	$bA(A\mathbf{1} \cdot A\mathbf{1}) = \frac{1}{12}$	$\sum_{ijkl} b_i a_{ij} a_{jk} a_{jl} = \frac{1}{12}$
\bullet	$bA^3\mathbf{1} = \frac{1}{24}$	$\sum_{ijkl} b_i a_{ij} a_{jk} a_{kl} = \frac{1}{24}$

4.3. Order conditions for RKMK on post-Lie algebras.

The order conditions for Lie group methods are extensively studied in the literature [19, 28, 9]. The theory can be formulated abstractly in the present post-Lie setting. However, we will not give a detailed treatment of this here, but restrict ourselves to a proof of the order conditions for the RKMK methods in Definition 4.3, following the approach of [20] in a post-Lie setting.

Theorem 4.11. *Let $f \in \mathcal{L}$, where \mathcal{L} is a post-Lie algebra. If $\{a_{ij}\}_{i,j=1}^s$ and $\{b_j\}_{j=1}^s$ satisfy the classical RK order conditions of Theorem 4.10 up to order p , then the RKMK method in Definition 4.3 is also of order p , satisfying*

$$\Psi_{\text{RKMK}}(hf) - \exp^*(hf) = \mathcal{O}(h^{p+1}). \quad (45)$$

Proof. Let \mathfrak{c} and \mathfrak{C} be the infinitesimal characters and characters of the Hopf algebra of \mathcal{L} . Given $f \in \mathfrak{c}$, define a vector field $F: \mathfrak{c} \rightarrow \mathfrak{c}$ as

$$F(\theta) = d\exp_{\theta}^{-1}(\exp(\theta) \triangleright f).$$

According to Proposition 3.11 we have $\exp^*(f) = \exp(\theta(1))$, where $\theta(t) \in \mathfrak{c}$ satisfies the differential equation $\theta(0) = 0$, $\theta'(t) = F(\theta(t))$. Note that \mathfrak{c} is a vector space, and is therefore also naturally a pre-Lie algebra, and we can integrate the differential equation using one step of a classical s -stage RK method of order p , with stepsize $h = 1$, starting at $\theta(0) = 0$. This yields

$$\begin{aligned} U_i &= \sum_{j=1}^s a_{ij} K_j \\ K_i &= F(U_i) = d\exp_{U_i}^{-1}(\exp(U_i) \triangleright f), \quad i = 1, \dots, s, \\ \theta(1) &\approx \sum_{j=1}^s b_j K_j, \end{aligned}$$

which is the RKMK scheme. The verification of the order (45) is now straightforward. ■

Acknowledgments. The authors would like to thank the referee for helpful comments, which led to a substantial improvement of this work.

References

- [1] Burde, D., *Left-symmetric algebras, or pre-Lie algebras in geometry and physics*, Central European J. of Math. **4** (2006), 323–357.
- [2] Butcher, J. C., *Coefficients for the study of Runge-Kutta integration processes*, J. of the Aust. Math. Soc. **3** (1963), 185–201.
- [3] —, *An algebraic theory of integration methods*, Mathematics of Computation, **26** (1972), 79–106.
- [4] —, “Numerical methods for ordinary differential equations,” John Wiley & Sons, 2008.
- [5] Cartier, P., *Vinberg Algebras, Lie Groups and Combinatorics*, Quanta of Maths **11** (2010).
- [6] Cayley, A., *On the theory of the analytical forms called trees*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **13** (1857), 172–176.
- [7] Connes, A., and D. Kreimer, *Hopf Algebras, Renormalization and Noncommutative Geometry*, Commun. Math. Phys. **199** (1998), 203–242.
- [8] Chapoton, F., and M. Livernet, *Pre-Lie algebras and the rooted trees operad*, International Math. Research Notices **2001** (2001), 395–408.
- [9] Celledoni, E., A. Marthinsen, and B. Owren, *Commutator-free Lie group methods*, Future Generation Computer Systems **19** (2003), 341–352.
- [10] Ebrahimi-Fard, K., A. Lundervold, and D. Manchon, *Noncommutative Bell polynomials, quasideterminants and incidence Hopf algebras*, International J. of Algebra and Computation **24** (2014), 671–705.
- [11] Grossman, R., and R. G. Larson, *Hopf-algebraic structure of families of trees*, J. of Algebra **126** (1989), 184–210.
- [12] Hairer, E., C. Lubich, and G. Wanner, “Geometric numerical integration: structure-preserving algorithms for ordinary differential equations,” Springer Series in Computational Mathematics **31**, 2006.
- [13] Hoffman, M., *Combinatorics of rooted trees and Hopf algebras*, Trans. Amer. Math. Soc. **355** (2003), 3795–3811.
- [14] Iserles, A., H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, *Lie-group methods*, Acta Numerica **9** (2000), 215–365.

- [15] Lundervold, A., and H. Z. Munthe-Kaas, *Hopf algebras of formal diffeomorphisms and numerical integration on manifolds*, Contemporary Mathematics **539** (2011), 295–324.
- [16] —, *Backward error analysis and the substitution law for Lie group integrators*, Foundations of Computational Mathematics **13** (2013), 161–186.
- [17] Manchon, D., *A short survey on pre-Lie algebras*, Noncommutative Geometry and Physics: Renormalisation, Motives, Index Theory, (2011), 89–102.
- [18] Munthe-Kaas, H., *Lie-Butcher theory for Runge-Kutta methods*, BIT Numerical Mathematics **35** (1995), 572–587.
- [19] —, *Runge-Kutta methods on Lie groups*, BIT Numerical Mathematics **38** (1998), 92–111.
- [20] —, *High order Runge-Kutta methods on manifolds*, Applied Numerical Mathematics **29** (1999), 115–127.
- [21] Munthe-Kaas, H., and S. Krogstad, *On enumeration problems in Lie-Butcher theory*, Future Generation Computer Systems **19** (2003), 1197–1205.
- [22] Munthe-Kaas, H. Z., and A. Lundervold, *On post-Lie algebras, Lie-Butcher series and moving frames*, Foundations of Computational Mathematics **13** (2013), 583–613.
- [23] Munthe-Kaas, H. Z., and O. Verdier, *Equivariant Series*, arXiv preprint arXiv:1308.5824, (2013).
- [24] Munthe-Kaas, H. Z., and W. M. Wright, *On the Hopf algebraic structure of Lie group integrators*, Foundations of Computational Mathematics **8** (2008), 227–257.
- [25] McLachlan, R. I., K. Modin, H. Z. Munthe-Kaas, and O. Verdier, *B-series methods are exactly the local, affine equivariant methods*, arXiv preprint arXiv:1409.1019, (2014).
- [26] Murua, A., *The Hopf algebra of rooted trees, free Lie algebras, and Lie series*, Foundations of Computational Mathematics **6** (2006), 387–426.
- [27] Oudom, J.-M., and D. Guin, *On the Lie enveloping algebra of a pre-Lie algebra*, J. of K-theory: K-theory and its Applications to Algebra, Geometry, and Topology **2** (2008), 147–167.
- [28] Owren, B., and A. Marthinsen, *Runge-Kutta methods adapted to manifolds and based on rigid frames*, BIT Numerical Mathematics **39** (1999), 116–142.
- [29] Reutenauer, C., *Free Lie algebras*, Handbook of Algebra **3** (2004), 887–903.
- [30] Schimming, R., and S. Z. Rida, *Noncommutative Bell polynomials*, International J. of Algebra and Computation **6** (1996), 635–644.

- [31] Vallette, B., *Homology of generalized partition posets*, J. of Pure and Applied Algebra **208** (2007), 699–725.

Kurusch Ebrahimi-Fard
ICMAT
Calle Nicolás Cabrera 13-15
Campus de Cantoblanco, UAM
Madrid, Spain
kurusch@icmat.es

Alexander Lundervold
Dept. of Computing
Mathematics and Physics
Bergen University College
Postboks 7030
Bergen, N-5020, Norway
alexander.lundervold@hib.no

Hans Z. Munthe-Kaas
Dept. of Mathematics
University of Bergen
Postbox 7800
Bergen, N-5020, Norway
hans.munthe-kaas@math.uib.no

Received November 7, 2013
and in final form April 14, 2015