BIT 38:1 (1998), 92–111

RUNGE-KUTTA METHODS ON LIE GROUPS *

HANS MUNTHE-KAAS †

Department of Informatics, University of Bergen, Thormøhlensgt. 55
N-5020, Bergen, Norway. email: Hans.Munthe-Kaas@ii.uib.no

Abstract.

We construct generalized Runge-Kutta methods for integration of differential equations evolving on a Lie group. The methods are using intrinsic operations on the group, and we are hence guaranteed that the numerical solution will evolve on the correct manifold. Our methods must satisfy two different criteria to achieve a given order:

- Coefficients $A_{i,j}$ and b_j must satisfy the classical order conditions. This is done by picking the coefficients of any classical RK scheme of the given order.
- We must construct functions to correct for certain non-commutative effects to the given order.

These tasks are completely independent, so once correction functions are found to the given order, we can turn any classical RK scheme into an RK method of the same order on any Lie group.

The theory in this paper shows the tight connections between the algebraic structure of the order conditions of RK methods and the algebraic structure of the so called 'universal enveloping algebra' of Lie algebras. This may give important insight also into the classical RK theory.

1 Introduction.

In many areas of applied mathematics one wants to solve differential equations whose solution is known to evolve on a given manifold. There are two main families of solution techniques for such problems, embedded and intrinsic methods. In the first of these one embeds the manifold in $\mathbb{R}^n$ and employs a classical integration scheme here. The problem of this approach is that, except in very special cases, it is generally impossible to find classical integration schemes which will stay on the correct manifold [5, 9, 11]. The alternative, intrinsic methods, are based on expressing the algorithm via a set of 'basic flows' which in each point span all the directions of the manifold. These methods have the advantage of being guaranteed to sit on the manifold. The price we must pay for this is that we must be able to compute the basic flows exactly, e.g., by computing matrix exponentials or by integrating 'simpler' vector fields defining the basic flows.

*Received August 1996. Revised August 1997. Communicated by Gustaf Söderlind.

†This work is sponsored by NFR under contract no. 111038/410, through the SYNODE project. WWW: **http://www.math.ntnu.no/num/synode**.

The topic of this paper is intrinsic integration algorithms in the case where the domain is a Lie group. This has many nice applications briefly discussed in [12, 14, 15]. In [15] we show that if we know how to integrate equations on general Lie groups, we can also integrate equations on virtually any manifold of practical interest (more precisely on any homogeneous manifold). Another application is the construction of *explicit* orthogonal and unitary integrators. It is known that all classical orthogonal integrators must be *implicit* [17], while these new algorithms can be explicit.

We want to point out some related papers dealing with intrinsic integration algorithms. Iserles [8] presents intrinsic integrators for linear ODEs with variable coefficients. This approach can more generally be understood as a numerical implementation of the method of Lie reduction [2, 22], and can be applied to any equation of Lie type. Zanna [21] proposes generalizations of this approach to general ODEs on Lie groups.

In [13] we presented a class of generalized Runge-Kutta methods for differential equations on Lie groups, which we will refer to as MK methods. The main emphasis in [13] was to show how the classical order theory of Butcher could be understood in terms of operations on commutative Lie groups. A related approach is the method of Crouch and Grossman (C-G) [6], who present a general approach and give explicit formulas for a 3rd order method. The order theory of their algorithms has recently been systematically developed by Owren and Marthinsen in [16], who also device C-G methods of order 4. The main difference between the C-G and MK methods is that whereas C-G does the approximations in the Lie group, MK does them in the Lie algebra. That is, C-G advances by a composed product of exponentials, while MK first combines elements in the Lie algebra, and then advances by a single exponential mapping applied to this combination. It turns out that the order theory of the MK approach is dramatically simpler than that of the C-G approach. The main reason for this is that whereas operations in the Lie group are nonlinear, the Lie algebra is a linear space. However, the original formulation of MK methods cannot achieve order higher than 2 on a non-commutative Lie group, and it needs some modifications.

In the present paper we reformulate the theory of [13] in a more precise language, and introduce the necessary modifications in order to construct higher order methods for general non-commutative groups. The main idea behind this construction is to devise methods whose order theory is as close as possible to the classical Butcher theory. We obtain a family of methods where two different criteria must be satisfied in order to achieve a given order:

- Finding coefficients $A_{i,j}$ and b_j satisfying the classical conditions. This is simply done by picking the coefficients of any classical RK scheme of the given order.
- Finding some correction functions, which correct certain non-commutative effects to the given order.

The two tasks above are completely independent, so once correction functions are found to the given order, we can turn any classical RK scheme into an RK

method of the same order on any Lie group. While Owren and Marthinsen [16] prove that for C-G type methods 5 stages are needed to obtain order 4, the present work give examples of 4 stage 4th order algorithms of MK type[1].

Readers who want to see the structure of the algorithms without diving too deeply into the mathematics may jump to Section 3.2, where 3rd and 4th order algorithms are explicitly written out.

2 Mathematical background and notation.

We have chosen to present this paper in the language of general Lie groups rather than, e.g., in the more special setting of matrix Lie groups. There are several reasons for this choice. First of all a mathematically rigorous presentation will be an important reference for future work, and it emphasizes that the algorithms are not depending on particular representations. The general language is also precise with respect to the type of the mappings involved, and we avoid the 'type errors' which are easily introduced if we employ the language of matrix Lie groups. Thus the abstract language is also an important guide if we want to design object oriented programs for solving differential equations.

The readers who are not familiar with the general notation may find it very useful to interpret the formulas for two concrete examples; first the *Abelian case* (where our methods reduce to the classical Runge-Kutta methods on $\mathbb{R}^n$) and secondly the matrix Lie group case. All concepts in abstract Lie group theory have a concrete counterpart for matrix Lie groups, which gives a good conceptual handle on the essential ideas. The third important interpretation is the case where the Lie group is a group of diffeomorphisms on a manifold and the Lie algebra consists of vector fields tangential to the manifold. The reader will find the details of these three cases in Section 2.4.

The texts [2, 19, 20] are excellent references on Lie group theory.

2.1 Differential equations on manifolds.

Let $f : \mathcal{N} \to \mathcal{M}$ be a smooth mapping between manifolds. We let $f' : T\mathcal{N} \to T\mathcal{M}$ denote the derivative of f, defined as a mapping between the tangent manifolds. The derivative of a composition of mappings is given as $(f \circ g)' = f' \circ g'$. To avoid a cluttered notation we will identify $T\mathbb{R}$ with $\mathbb{R}$ itself, thus, e.g., if $y_t : \mathbb{R} \to \mathcal{M}$ is a curve on $\mathcal{M}$, then $y_t' : \mathbb{R} \to T\mathcal{M}$ is a curve on $T\mathcal{M}$.

A vector field on $\mathcal{M}$ is a smooth mapping $F : \mathcal{M} \to T\mathcal{M}$, such that $\pi \circ F = \mathrm{Id}_{\mathcal{M}}$, where $\pi : T\mathcal{M} \to \mathcal{M}$ is the natural projection. F defines a general initial value problem as:

$$y_t' = F(y_t), \quad y_0 = p. \tag{2.1}$$

The *flow* of F is the solution operator $\Psi_{F,t} : \mathcal{M} \to \mathcal{M}$ defined in such a way that $y_t = \Psi_{F,t}(p)$ is the integral curve starting in p, i.e.,

$$\frac{\partial}{\partial t}\Psi_{F,t}(p) = F(\Psi_{F,t}(p)), \quad \Psi_{F,0}(p) = p. \tag{2.2}$$

[1] In a succeeding paper [14], we show how classical RK methods of *any* order can be modified to a Lie group invariant method of the same order.

Our goal is to study the numerical integration of (2.1) in the case where $\mathcal{M}$ is a Lie group.

2.2 Lie groups and algebras.

A Lie group is defined as a manifold G equipped with a continuous and smooth group product $\cdot : G \times G \to G$. We define left and right multiplications in the group as:

$$\begin{aligned} L_a(b) &= a \cdot b \quad \text{for } a, b \in G \\ R_a(b) &= b \cdot a \quad \text{for } a, b \in G \end{aligned}$$

The Lie algebra is defined as the tangent space of G in the identity, $\mathfrak{g} = TG|_e$, and has the structure of a (real or complex) linear space equipped with a bilinear skew symmetric form $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$. It is called the *Lie bracket*, defined below. If G is a matrix Lie group then $\mathfrak{g}$ is a space of square matrices, and the Lie bracket is the matrix commutator $[u, v] = uv - vu$.

The tangent space at any point $a \in G$, $TG|_a$ can be identified with $\mathfrak{g}$ using either left or right multiplication, i.e.,

$$TG|_a = \{ L'_a(v) \mid v \in \mathfrak{g} \} = \{ R'_a(w) \mid w \in \mathfrak{g} \}.$$

Thus any vector field $F : G \to TG$ can be written in either of the forms:

$$F(a) = L'_a(f(a)) = R'_a(\tilde{f}(a)), \tag{2.3}$$

where $f, \tilde{f} : G \to \mathfrak{g}$. The *adjoint representation* of G is defined as the mapping[2] $\mathrm{Ad} : G \to \mathrm{Aut}(\mathfrak{g})$ given as:

$$\mathrm{Ad}(a)(v) = (L_a \circ R_{a^{-1}})'(v) \ , \quad v \in \mathfrak{g}. \tag{2.4}$$

We have $\tilde{f}(a) = \mathrm{Ad}(a)(f(a))$, thus the adjoint mapping is a way of relating left and right representations of vector fields.

NOTE 2.1. All the algorithms of this paper exist in left and right versions. We have based most of the presentation here on the left versions. In some situations it is important to know the right version, and therefore we give the most important formulas and definitions in both forms.

The mapping $\mathrm{ad} : \mathfrak{g} \to \mathrm{End}(\mathfrak{g})$ is defined as:

$$\mathrm{ad}(u) = \mathrm{Ad}'(e), \tag{2.5}$$

and the Lie bracket $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ is a bilinear skew symmetric form on $\mathfrak{g}$ defined as

$$[u, v] = \mathrm{ad}(u)(v). \tag{2.6}$$

Since $\mathrm{ad}(u) : \mathfrak{g} \to \mathfrak{g}$, we can define $\mathrm{ad}^n(u)$ as the n-times iterated map, i.e.,

$$\begin{aligned} \mathrm{ad}^0(u)(v) &= v \\ \mathrm{ad}^n(u)(v) &= \mathrm{ad}(u)(\mathrm{ad}^{n-1}(u)(v)) = [u, [u, [\ldots, [u, v]]]], \text{ for } n \geq 1. \end{aligned}$$

[2] $\mathrm{End}(\mathfrak{g})$ is the set of all linear maps from $\mathfrak{g}$ to itself, and $\mathrm{Aut}(\mathfrak{g})$ those which are invertible.

2.3 The exponential mapping and its differential.

The exponential mapping is a mapping $\exp : \mathfrak{g} \to G$. In the case of matrix Lie groups it is given as the matrix exponential (2.11), and on $\mathrm{Diff}(\mathcal{M})$ it returns the flow of a given vector field (2.12).

The differential of the exponential turns out to be of major importance in the order theory for Runge-Kutta methods on Lie groups. Since $\exp : \mathfrak{g} \to G$, we must have $\exp' : T\mathfrak{g} \to TG$. However, since $\mathfrak{g}$ is a vector space we can identify $T\mathfrak{g}$ with $\mathfrak{g}$ itself, and we have seen that also the tangent space $TG|_a$ can be identified with $\mathfrak{g}$. Thus $\exp'$ can be expressed via a function $d\exp : \mathfrak{g} \to \mathfrak{g}$. There are two ways of doing this, depending on whether $TG|_a$ is related to $\mathfrak{g}$ via left of right multiplication. For $u, v \in \mathfrak{g}$ we define:

$$d\exp_u(v) = L'_{\exp(-u)} \circ \exp'(u, v) \tag{2.7}$$

$$d^r \exp_u(v) = R'_{\exp(-u)} \circ \exp'(u, v), \tag{2.8}$$

where $\exp'(u, v) = \frac{\partial}{\partial t} \exp(u + tv)\big|_{t=0}$. There are various formulas for expressing $d\exp_u(v)$ and $d^r \exp_u(v)$:

$$d\exp_u = \sum_{j=0}^{\infty} \frac{(-1)^j}{(j+1)!} \mathrm{ad}^j(u) = \int_0^1 \exp(\mathrm{ad}(-su))\, ds \tag{2.9}$$

$$d^r \exp_u = \sum_{j=0}^{\infty} \frac{1}{(j+1)!} \mathrm{ad}^j(u) = \int_0^1 \exp(\mathrm{ad}(su))\, ds \tag{2.10}$$

The sum form of (2.9) is derived in Varadarajan [19] p. 108, while a form similar to the integral form in (2.10) is implicitly derived in the paper of Crouch and Grossman [6] (their Lemma 3). Iserles [8] derives the sum form of (2.10) (his Theorem 1).

2.4 Some important examples of Lie groups.

2.4.1 Abelian groups and $\mathbb{R}^n$.

Abelian groups are in general groups where $a \cdot b = b \cdot a$ for all $a, b \in G$. This implies that $[u, v] = 0$ and hence that $d\exp_u(v) = v$ for all $u, v \in \mathfrak{g}$. Abelian groups are always locally vector spaces, though they may be globally different (e.g., tori).

The real vector spaces $\mathbb{R}^n$ are special Abelian Lie groups where:

$$\begin{aligned} (G, \cdot) &= (\mathbb{R}^n, +) \\ (\mathfrak{g}, +, [\cdot, \cdot]) &= (\mathbb{R}^n, +, [u, v] = 0) \\ \exp(u) &= u \\ TG &= \mathbb{R}^n \\ R'_a(b) &= L'_a(b) = b. \end{aligned}$$

The initial value problem (2.35) takes the form

$$y'_t = f(y_t), \quad \text{for } f : \mathbb{R}^n \to \mathbb{R}^n.$$

2.4.2 Matrix Lie groups.

The general linear group, $G = GL(n, \mathbb{R})$, is the group of all real invertible $n \times n$ matrices, where the product is the matrix product. The Lie algebra $\mathfrak{g} = \mathfrak{gl}(n, \mathbb{R})$ consists of all real $n \times n$ matrices, $+$ in $\mathfrak{g}$ is the sum of matrices. We have:

$$
\begin{aligned}
[u, v] &= uv - vu \\
\exp(v) &= \sum_{j=0}^{\infty} \frac{v^j}{j!} \\
TG &= \{ bv \mid b \in G, v \in \mathfrak{g} \} = \{ vb \mid b \in G, v \in \mathfrak{g} \} \\
R'_a(vb) &= vba \\
L'_a(bv) &= abv.
\end{aligned} \tag{2.11}
$$

In this case the initial value problem (2.35) takes the form (3.12).

Other matrix Lie groups are continuous subgroups of $GL(n)$, and the corresponding algebras are subalgebras of $\mathfrak{gl}(n)$. Some important cases are:

- The orthogonal group $O(n) = \{ a \in GL(n, \mathbb{R}) \mid a^T a = I \}$. The corresponding algebra consists of all skew symmetric matrices.
- The special linear group $SL(n) = \{ a \in GL(n, \mathbb{R}) \mid \det a = 1 \}$, where the algebra consists of all matrices with trace 0.

$O(n)$ is important in orthogonal flows and $SL(n)$ is essential for the understanding of volume preserving flows.

2.4.3 The group Diff($\mathcal{M}$).

For a given manifold $\mathcal{M}$, $\mathrm{Diff}(\mathcal{M})$ consists of all diffeomorphisms ϕ on $\mathcal{M}$, i.e., smooth invertible mappings $\phi : \mathcal{M} \to \mathcal{M}$. Its Lie algebra is $\mathfrak{X}(\mathcal{M})$, i.e., the set of all tangent vector fields on $\mathcal{M}$, and the bracket $[\cdot, \cdot]$ is the Lie-Jacobi bracket of vector fields, see [1] for its definition. The product in $\mathrm{Diff}(\mathcal{M})$ is the composition of diffeomorphisms and the sum in $\mathfrak{X}(\mathcal{M})$ is the pointwise sum of vector fields. The exponential mapping $\exp : \mathfrak{X}(\mathcal{M}) \to \mathrm{Diff}(\mathcal{M})$ is given as

$$
\exp(tF) = \Psi_{F,t}, \tag{2.12}
$$

where $\Psi_{F,t}$ is given in (2.2).

2.5 The universal enveloping algebra of $\mathfrak{g}$.

The enveloping algebra allows us to introduce higher order differential operators, and is important in the order theory of RK methods.

An element $v \in \mathfrak{g}$ defines left and right invariant vector fields on G as:

$$
X_v(a) = L'_a(v) \quad \text{(left invariant)} \tag{2.13}
$$

$$
Y_v(a) = R'_a(v) \quad \text{(right invariant)}. \tag{2.14}
$$

The flows of X_v and Y_v are given as:

$$\Psi_{X_v,t} = R_{\exp(tv)} \tag{2.15}$$

$$\Psi_{Y_v,t} = L_{\exp(tv)}. \tag{2.16}$$

Let f be a smooth real valued function on G, $f \in C^\infty(\mathrm{G})$. Then v may act as a derivation of f either from left or right, as the Lie derivative of f with respect to either X_v or Y_v. I.e., $v[\cdot], [\cdot]v : C^\infty(\mathrm{G}) \to C^\infty(\mathrm{G})$ are defined as:

$$(v[f])(a) = \frac{\partial}{\partial t} f(a \cdot \exp(tv))\Big|_{t=0} \tag{2.17}$$

$$([f]v)(a) = \frac{\partial}{\partial t} f(\exp(tv) \cdot a)\Big|_{t=0}. \tag{2.18}$$

We can introduce higher order left and right invariant differential operators by iterating first order operators, i.e., we introduce the second order left and right invariant operators[3] $(u \cdot v)[\cdot] = u[v[\cdot]]$ and $[\cdot](u \cdot v) = [[\cdot]u]v$. It can be shown that

$$(u \cdot v)[f](a) = \frac{\partial^2}{\partial t_1 \partial t_2} f(a \cdot \exp(t_1 u) \cdot \exp(t_2 v))\Big|_{t_1=t_2=0} \tag{2.19}$$

$$[f]\,(u \cdot v)(a) = \frac{\partial^2}{\partial t_1 \partial t_2} f(\exp(t_1 u) \cdot \exp(t_2 v) \cdot a)\Big|_{t_1=t_2=0}. \tag{2.20}$$

The sum of two differential operators is defined the obvious way, $(u + v)[f] = u[f] + v[f]$, and the 0'th order identity operator $\mathbb{I}$ is defined as $\mathbb{I}[f] = f$.

DEFINITION 2.1. *The Universal enveloping algebra $\mathfrak{G}$ of a Lie algebra $\mathfrak{g}$ consists of all invariant differential operators generated by $\{\mathbb{I}, \mathfrak{g}\}$ under the operations $\cdot$ and $+$ defined above.*

For any vector space V we identify TV with V, thus if $\nu_t : \mathbb{R} \to V$ we have $\partial \nu_t / \partial t = \nu_t' : \mathbb{R} \to V$. In this case we will henceforth use the notation:

$$\nu^{(i)} = \frac{\partial^i}{\partial t^i} \nu_t\Big|_{t=0}. \tag{2.21}$$

More specifically, for a curve $\nu_t : \mathbb{R} \to \mathfrak{G}$, we define the curve $\nu_t' : \mathbb{R} \to \mathfrak{G}$ such that

$$\nu_t'[f] = \frac{\partial}{\partial t}(\nu_t[f]) \quad \text{for all } f \in C^\infty(\mathrm{G}).$$

It can be shown that the sum $+$ and the product $\cdot$ on $\mathfrak{G}$ behave in the familiar way under derivations, i.e.,

$$(\nu_t + \mu_t)' = \nu_t' + \mu_t'$$

$$(\nu_t \cdot \mu_t)' = \nu_t' \cdot \mu_t + \nu_t \cdot \mu_t'.$$

[3]The product $u \cdot v$ should not be confused with a matrix product!

NOTE 2.2. We have introduced $\mathfrak{G}$ in terms of invariant derivations on G. An alternative approach is a purely algebraic construction, see [19]. In this construction, the products $u \cdot v$ are tensor products[4]. Thus, e.g., second order homogeneous elements of $\mathfrak{G}$ lie in (a quotient space of) $\mathfrak{g} \otimes \mathfrak{g}$. The detailed algebraic structure of $\mathfrak{G}$ is not needed for this paper. However, to understand the tensorial nature of the order conditions for RK methods, it is very useful to look somewhat closer at the enveloping algebra of the s-fold cartesian product of a Lie group.

2.6 Cartesian products of Lie groups.

Let G_s denote the s-fold cartesian product of a Lie group, $\mathrm{G}_s = \mathrm{G} \times \cdots \times \mathrm{G}$, let $\mathfrak{g}_s$ be its Lie algebra, $\mathfrak{g}_s = \mathfrak{g} \times \cdots \times \mathfrak{g}$ and let $\mathfrak{G}_s$ be the enveloping algebra $\mathfrak{G}_s = \mathfrak{G} \times \cdots \times \mathfrak{G}$. The elements of G_s, $\mathfrak{g}_s$ and $\mathfrak{G}_s$ are s-tuples of G, $\mathfrak{g}$ and $\mathfrak{G}$ respectively. All the basic operations such as product on G_s, $+$ and $[\cdot,\cdot]$ on $\mathfrak{g}_s$, $\exp : \mathfrak{g}_s \to \mathrm{G}_s$ and the product $\cdot$ on $\mathfrak{G}_s$ are defined componentwise on the tuples. We have, e.g.,

$$\begin{aligned} \exp(v) &= (\exp(v_1), \ldots, \exp(v_s)), \text{ where } v = (v_1, v_2, \ldots, v_s) \in \mathfrak{g}_s \\ u + v &= (u_1 + v_1, \ldots, u_s + v_s), \text{ where } u, v \in \mathfrak{g}_s \end{aligned} \tag{2.22}$$

We introduce the following tensor product basis for $\mathfrak{g}_s$:

$$\mathfrak{g}_s = \mathbb{R}^s \otimes \mathfrak{g},$$

i.e., $\mathfrak{g}_s$ is the linear span of vectors such as:

$$q \otimes v = (q_1 v, q_2 v, \ldots, q_s v), \quad \text{where } q \in \mathbb{R}^s, v \in \mathfrak{g}.$$

We will see that this basis can be extended to all of $\mathfrak{G}_s$. We equip $\mathbb{R}^s$ with the Schur product, i.e.,

$$q \cdot r = (q_1 r_1, q_2 r_2, \ldots, q_s r_s)^T \tag{2.23}$$

and the identity in $\mathbb{R}^s$ is given as

$$\mathbb{1} = (1, 1, \ldots, 1)^T.$$

Obviously, the identity in $\mathfrak{G}_s$ is:

$$\mathbb{I}_s = \mathbb{1} \otimes \mathbb{I} = (\mathbb{I}, \mathbb{I}, \ldots, \mathbb{I}),$$

and from (2.23) we see that the product in $\mathfrak{G}_s$ is given as

$$(q \otimes v) \cdot (r \otimes w) = (q \cdot r) \otimes (v \cdot w) \quad \text{for } q, r \in \mathbb{R}^s,\ v, w \in \mathfrak{G}. \tag{2.24}$$

This gives the product on $\mathfrak{G}_s$ in the tensor product basis, and it is extended to all of $\mathfrak{G}_s$ by linearity. For the order theory of Runge-Kutta methods, it is important to note that whereas generally $v \cdot w \neq w \cdot v$ for $v, w \in \mathfrak{G}$, we always have $q \cdot r = r \cdot q$ for $q, r \in \mathbb{R}^s$. *I.e., the non-commutativity sits only in the right part of the product on* $\mathfrak{G}_s$.

[4]Or more precisely a quotient construction on a tensor product space, where elements such as $u \cdot v - v \cdot u - [u, v]$ are divided out.

2.7 Equations of Lie type.

Given a curve $\nu_t \in \mathfrak{g}$, an equation

$$y_t' = X_{\nu_t}(y_t) = L'_{y_t}(\nu_t), \quad y_0 = p, \tag{2.25}$$

where the right-hand side is a time dependent, left- or right-invariant vector field, is called an equation of *Lie type*. Such equations play a central role in the analytical treatment of ODEs. Let s_t be the integral curve of (2.25) starting in $s_0 = e$, where e is the identity in G. This is called the ***fundamental solution.*** The general solution of (2.25) is given as $y_t = p \cdot s_t$. If the right-hand side is instead Y_{ν_t}, we obtain the solution $y_t = s_t \cdot p$. Thus:

$$\Psi_{X_{\nu_t},t} = R_{s_t}, \quad \text{where } s_t' = X_{\nu_t}(s_t), \quad s_0 = e, \tag{2.26}$$

$$\Psi_{Y_{\nu_t},t} = L_{s_t}, \quad \text{where } s_t' = Y_{\nu_t}(s_t), \quad s_0 = e. \tag{2.27}$$

Equations of Lie type provide us with a natural 1–1 correspondence between curves on $\mathfrak{g}$ and curves on G starting in the point p. This will be used in our definition of the order of numerical integration schemes.

LEMMA 2.1. *Given an arbitrary curve $y_t \in G$, $y_0 = p$. If $\nu_t \in \mathfrak{g}$ is given as*

$$\nu_t = L'^{-1}_{y_t}(y_t'),$$

then

$$y_t = R_{s_t}(p), \quad \textit{where } s_t' = X_{\nu_t}(s_t), \quad s_0 = e.$$

2.8 Lie-Butcher series.

The basic tool for series developments on manifolds is *Lie series*, which is essentially Taylor series adapted to manifolds. The Lie-Butcher series, introduced in [13], is a special form of Lie series which generalizes the theory of J. Butcher [3, 4] to manifolds.

Let F be a vector field on a manifold $\mathcal{M}$, let $\Psi_{F,t}$ be the flow of F and let f be a 'geometric object'[5] on $\mathcal{M}$. We now assume that $f \in C^\infty(\mathcal{M})$. Let $\Psi^*_{F,t} f$ denote the ***pullback*** of the object along the flow. The definition of the pullback depends on the object. When $f \in C^\infty(\mathcal{M})$ the pullback is simply defined as

$$\Psi^*_{F,t} f = f \circ \Psi_{F,t}. \tag{2.28}$$

The Lie derivative of the object f with respect to a vector field F is generally defined as:

$$F[f] = \frac{\partial}{\partial t}(\Psi^*_{F,t} f)\Big|_{t=0}, \tag{2.29}$$

i.e., the rate of change of f at time $t = 0$, as it pulled back along the flow. By iterating this formula, we find the basic form of the Lie series:

$$\Psi^*_{F,t} f = f + tF[f] + \frac{t^2}{2}F[F[f]] + \frac{t^3}{3!}F[F[F[f]]] + \cdots = \sum_{i=0}^{\infty} \frac{t^i}{i!} F^i[f].$$

[5] It can, e.g., be a real function, a tensor field, a differential k-form etc.

If both the vector field F and the object f are time dependent, the rate of change at an arbitrary time is given as [1] (p.285)

$$\frac{\partial}{\partial t}(\Psi^*_{F_t,t} f_t) = \Psi^*_{F_t,t}\left(F_t[f_t] + \frac{\partial f_t}{\partial t}\right). \tag{2.30}$$

Now, if f is independent of time and $F_t = X_{\nu_t}$ then (2.30) yields:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\Psi^*_{X_{\nu_t},t} f\right) &= \Psi^*_{X_{\nu_t},t}(\nu_t[f]) \\ \frac{\partial^2}{\partial t^2}\left(\Psi^*_{X_{\nu_t},t} f\right) &= \Psi^*_{X_{\nu_t},t}(\nu_t \cdot \nu_t[f] + \nu_t'[f]) \quad \text{etc.}\end{aligned}$$

To express the nth derivative, we introduce $B^n_t(\nu_t) \in \mathfrak{G}$ recursively as:

$$B^0_t(\nu_t) = \mathbb{I} \tag{2.31}$$

$$B^i_t(\nu_t) = \nu_t \cdot B^{i-1}_t(\nu_t) + \frac{\partial}{\partial t}\left(B^{i-1}_t(\nu_t)\right), \quad i > 0. \tag{2.32}$$

From (2.30) we see that

$$\frac{\partial^n}{\partial t^n}\left(\Psi^*_{X_{\nu_t},t} f\right) = \Psi^*_{X_{\nu_t},t} B^n_t(\nu_t)[f]. \tag{2.33}$$

Note that $B^n_t(\nu_t)$ depends only on the derivatives of ν_t up to order $n-1$. We define

$$B^n\left(\nu^{(0)}, \nu^{(1)}, \ldots, \nu^{(n-1)}\right) = B^n_t\left(\sum_{i=0}^{n-1} \frac{t^i}{i!}\nu^{(i)}\right)\Bigg|_{t=0}. \tag{2.34}$$

B^n are n-variate polynomials in $\mathfrak{G}$. The first of these are:

$$\begin{aligned}B^0() &= \mathbb{I} \\ B^1\left(\nu^{(0)}\right) &= \nu^{(0)} \\ B^2\left(\nu^{(0)}, \nu^{(1)}\right) &= \nu^{(0)} \cdot \nu^{(0)} + \nu^{(1)} \\ B^3\left(\nu^{(0)}, \nu^{(1)}, \nu^{(2)}\right) &= \nu^{(0)} \cdot \nu^{(0)} \cdot \nu^{(0)} + 2\nu^{(0)} \cdot \nu^{(1)} + \nu^{(1)} \cdot \nu^{(0)} + \nu^{(2)}.\end{aligned}$$

We will use these polynomials and Lemma 2.1 to find a special type Lie series characterizing the solution y_t of the general initial value problem (2.1). Using the left form of (2.3) we can write (2.1) as

$$y_t' = L'_{y_t}(f(y_t)), \quad y_0 = p, \tag{2.35}$$

where $f : \mathrm{G} \to \mathfrak{g}$. For vector valued functions (such as this f), the definition of pullback and derivations is identical to the definition for functions in $C^\infty(G)$ given above[6]. Thus from Lemma 2.1 and (2.26), we find that $\nu_t \in \mathfrak{g}$ is given as

$$\nu_t = L'_{y_t^{-1}}(y_t') = f(y_t) = f(p \cdot s_t) = \left(R^*_{s_t} f\right)(p) = \left(\Psi^*_{X_{\nu_t},t} f\right)(p). \tag{2.36}$$

[6]Consider $f : \mathrm{G} \to V$ via its components $f^i \in C^\infty(G)$ given relative to some basis for V.

From (2.33) we obtain:

THEOREM 2.2. *Let y_t be the solution of the initial value problem* (2.35) *and let ν_t be the corresponding curve on $\mathfrak{g}$ defined in Lemma* 2.1. *Then*

$$\nu_t = \sum_{i=0}^{\infty} \frac{t^i}{i!} \nu^{(i)}, \tag{2.37}$$

where

$$\nu^{(i)} = B^{(i)}\left(\nu^{(0)}, \nu^{(1)}, \ldots, \nu^{(i-1)}\right)[f](p) \tag{2.38}$$

and B^i are the polynomials defined in (2.34).

The series defined in (2.38) is what we call the *Lie-Butcher series* characterizing the initial value problem (2.35). To see clearer what it means, we introduce the following shorthand notation: For $\nu \in \mathfrak{G}$, define $\overline{\nu} \in \mathfrak{g}$ as

$$\overline{\nu} = \nu[f](p). \tag{2.39}$$

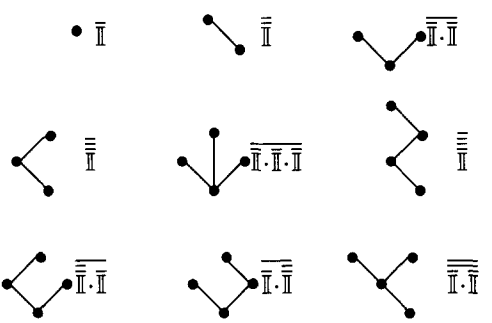

Figure 2.1: Correspondence with the Butcher-tree notation.

Thus the bar operator is an $\mathbb{R}$-linear map from $\mathfrak{G}$ to $\mathfrak{g}$ depending on f and p. Equation (2.38) yields

$$\nu^{(0)} = \overline{B^0()} = \overline{\mathbb{I}} \tag{2.40}$$

$$\nu^{(1)} = \overline{B^1(\nu^{(0)})} = \overline{\overline{\mathbb{I}}} \tag{2.41}$$

$$\nu^{(2)} = \overline{B^2(\nu^{(0)}, \nu^{(1)})} = \overline{\overline{\mathbb{I}} \cdot \overline{\mathbb{I}}} + \overline{\overline{\overline{\mathbb{I}}}} \tag{2.42}$$

$$\nu^{(3)} = \overline{\overline{\mathbb{I}} \cdot \overline{\mathbb{I}} \cdot \overline{\mathbb{I}}} + 2\,\overline{\overline{\mathbb{I}} \cdot \overline{\overline{\mathbb{I}}}} + \overline{\overline{\overline{\mathbb{I}}} \cdot \overline{\mathbb{I}}} + \overline{\overline{\mathbb{I}} \cdot \overline{\mathbb{I}}} + \overline{\overline{\overline{\mathbb{I}}}} \text{ etc.} \tag{2.43}$$

If we turn the bar-patterns upside down, we see the relationship between these operators and the elementary differentials in the Butcher theory, represented as trees. See Figure 2.1. Note that when the group is non-commutative, then permutation of the branches in the trees yields different elementary differentials. This differs from the classical Butcher theory, where trees with permuted branches are considered identical.

3 Runge-Kutta methods.

We will study the numerical solution of (2.35). There exist several ways to generalize classical Runge-Kutta methods to non-commutative Lie groups in such a way that they reduce to the classical methods when we plug in a commutative group. The formulation we study here is deliberately designed such that the order theory becomes as close as possible to the classical order theory. It is close to the formulation in [13], the main difference being the correction functions $\zeta_i()$ and $\zeta()$ which are new. The choice of correction functions is discussed later. Let s be the number of stages of the method and h the step size and let $\hat{y}_i$ be the numerical solution in step i. Our family of generalized RK methods is defined as:

ALGORITHM 3.1. **RK-MK :**

$\hat{y}_0 = p$
for $n = 0, 1, \ldots$
  for $i = 1, 2, \ldots, s$
    $u_i = h \sum_{j=1}^{s} A_{i,j} k_j$
    $\tilde{u}_i = \zeta_i(u_i, k_1, k_2, \ldots, k_s)$
    $k_i = f(\hat{y}_n \cdot \exp(\tilde{u}_i))$
  end
  $v = h \sum_{j=1}^{s} b_j k_j$
  $\tilde{v} = \zeta(v, k_1, k_2, \ldots, k_s)$
  $\hat{y}_{n+1} = \hat{y}_n \cdot \exp(\tilde{v})$
end

where $u_i, \tilde{u}_i, k_i, v, \tilde{v} \in \mathfrak{g}$ *and* $\hat{y}_i \in G$. *The real constants* $A_{i,j}, b_j$ *and the correction functions* $\zeta_i()$ *and* $\zeta()$ *determine a particular scheme. The method is explicit if* $A_{i,j} = 0$ *for* $i \leq j$ *and the functions* $\zeta_i(u_i, k_1, \ldots)$ *depend only on* $k_1, k_2, \ldots, k_{i-1}$. *In this case, the iteration over* i *can be computed explicitly.*

The classical way of defining the order of RK methods on $\mathbb{R}^n$ is to say that the method has order q if $||\hat{y}_1 - y_h|| = \mathcal{O}(h^{q+1})$. Since there is no particular metric on G, we must modify this definition. If we let h vary, $\hat{y}_1$ will trace out a curve $\hat{y}_h \in$ G. To this curve there corresponds a curve $\hat{\nu}_h \in \mathfrak{g}$ via Lemma 2.1. Our definition of order is based on matching the Taylor series of this curve with the Taylor series of the curve $\nu_t \in \mathfrak{g}$ derived from the analytical solution:

DEFINITION 3.1. *Let* $\hat{y}_h = \hat{y}_1$ *be the numerical solution after one step of Algorithm* 3.1, *considered as a function of* h. *Let* $\hat{y}_t$ *be the analytical solution of* (2.35) *and let* $\hat{\nu}_h$ *and* ν_t *be the corresponding curves in* $\mathfrak{g}$ *given by Lemma* 2.1:

$$\begin{aligned} \hat{\nu}_h &= L'^{-1}_{\hat{y}_h}(\hat{y}'_h) \\ \nu_t &= L'^{-1}_{y_t}(y'_t) \end{aligned}$$

then the method has order q *if*

$$\nu^{(i)} = \hat{\nu}^{(i)} \quad \text{for } i = 0, 1, \ldots, q-1 \ . \tag{3.1}$$

A very convenient way of analyzing the order of Algorithm 3.1 is to consider it as a 1-stage method on an s-fold product manifold: Let $\mathrm{G}_s = \mathrm{G} \times \cdots \times \mathrm{G}$, $f_s = f \times \cdots \times f$, let $\mathfrak{g}_s$ be the Lie algebra of G_s and $\mathfrak{S}_s$ its enveloping algebra. We make the identification $\mathfrak{g}_s \equiv \mathbb{R}^s \otimes \mathfrak{g}$ as in Section (2.6) and let $A_s = A \otimes I$, $b_s = b^T \otimes I$. The first step of Algorithm 3.1 can then be written as:

$$
\begin{aligned}
(3.2) \qquad p_s &= (p, p, \ldots, p) \in \mathrm{G}_s \\
(3.3) \qquad u &= hA_s k \\
(3.4) \qquad \tilde{u} &= \zeta(u, k) \\
(3.5) \qquad k &= f_s(p_s \cdot \exp(\tilde{u})) = \left(R^*_{\exp \tilde{u}} f_s\right)(p_s) \\
(3.6) \qquad v &= hb_s k \\
(3.7) \qquad \tilde{v} &= \zeta(v, k) \\
(3.8) \qquad \hat{y}_1 &= p \cdot \exp(\tilde{v}) = R_{\exp \tilde{v}}(p)
\end{aligned}
$$

where $u, \tilde{u}, k \in \mathfrak{g}_s$, $v, \tilde{v} \in \mathfrak{g}$ and $\hat{y}_1 \in \mathrm{G}$.

In the following analysis, we will let derivations such as $\tilde{v}'$ and $\tilde{v}^{(i)}$ be with respect to h.

LEMMA 3.1. *Let $\hat{\nu}_h \in \mathfrak{g}$ be defined in Definition* 3.1, *let $\tilde{z}_h = \exp(\tilde{u}) \in G_s$ and let $\hat{\mu}_h = L'_{\tilde{z}_h^{-1}}(\tilde{z}'_h) \in \mathfrak{g}_s$. Then*

$$
\begin{aligned}
\hat{\nu}_h &= d\exp_{\tilde{v}}(\tilde{v}') \\
\hat{\mu}_h &= d\exp_{\tilde{u}}(\tilde{u}')
\end{aligned}
$$

PROOF:

$$
\hat{\nu}_h = L'_{\exp(\tilde{v})^{-1}} \circ L'_{p^{-1}} \circ (p \cdot \exp(\tilde{v}))' = L'_{\exp(-\tilde{v})} \circ \exp'(\tilde{v}, \tilde{v}') = d\exp_{\tilde{v}}(\tilde{v}').
$$

The equation for $\hat{\mu}_h$ follows by a similar computation. □

We will henceforth make the following assumption about the correction functions:

ASSUMPTION 3.1. *The functions $\zeta(\cdot, \cdot)$ in* (3.4) *and* (3.7) *are chosen such that*

$$
\begin{aligned}
\hat{\mu}^{(i)} &= u^{(i+1)} \text{ for } i = 0, 1, \ldots, q-2, \\
\hat{\nu}^{(i)} &= v^{(i+1)} \text{ for } i = 0, 1, \ldots, q-1.
\end{aligned}
$$

NOTE 3.1. If G is Abelian, i.e., if $\mathrm{ad}(u) = 0$ for all $u \in \mathfrak{g}$, then we see from (2.9) and Lemma 3.1 that $\hat{\mu}^{(i)} = \tilde{u}^{(i+1)}$ and $\hat{\nu}^{(i)} = \tilde{v}^{(i+1)}$. Thus in this case the assumption holds if we choose $\tilde{u} = u$ and $\tilde{v} = v$.

THEOREM 3.2. *Under Assumption* 3.1, *the Lie-Butcher series for the numerical solution $\hat{y}_h$ is given by the following recursion for $\hat{\nu}^{(i)}$:*

$$
\begin{aligned}
(3.9) \qquad k^{(i)} &= \overline{B^i\left(A_s k^{(0)}, 2A_s k^{(1)}, \ldots, iA_s k^{(i-1)}\right)} \\
(3.10) \qquad \hat{\nu}^{(i)} &= (i+1) b_s k^{(i)},
\end{aligned}
$$

for $i = 0, 1, \ldots, q-1$.

PROOF: From $k = \left(R^*_{\exp \tilde{u}} f_s\right)(p_s)$, Lemma 3.1 and eqns. (2.26), (2.33) we get

$$k^{(i)} = B^i\left(\hat{\mu}^{(0)}, \ldots, \hat{\mu}^{(i-1)}\right)[f_s](p_s) = \overline{B^i\left(\hat{\mu}^{(0)}, \ldots, \hat{\mu}^{(i-1)}\right)},$$

and from the Leibniz rule $\frac{\partial^i}{\partial h^i} h f(h)|_{h=0} = i \frac{\partial^{i-1}}{\partial h^{i-1}} f(h)|_{h=0}$ and eqns. (3.3), (3.6), we get:

$$\begin{aligned} v^{(i)} &= i b_s k^{(i-1)} \\ u^{(i)} &= i A_s k^{(i-1)}. \end{aligned}$$

Thus from Assumption 3.1,

$$\begin{aligned} \hat{\mu}^{(i)} &= (i+1) A_s k^{(i)} \\ \hat{\nu}^{(i)} &= (i+1) b_s k^{(i)} \end{aligned}$$

which yield (3.9), (3.10). □

We will use this recursion to obtain the terms $\hat{\nu}^{(i)}$, and hence also the order conditions on $A_{i,j}$ and b_j. Note the following relation between the bar-operator on $\mathfrak{G}_s$ and on $\mathfrak{G}$:

$$\overline{q \otimes v} = (q_1 v, q_2 v, \ldots, q_s v)\,[f, \ldots, f]\,(p, \ldots, p) = (q_1 \overline{v}, \ldots, q_s \overline{v}) = q \otimes \overline{v}.$$

Let $c = A\mathbb{1}$, where $\mathbb{1} = (1, 1, \ldots, 1)^T$. We get:

$$\begin{aligned} k^{(0)} &= \overline{B^0(\,)} = \bar{\mathbb{I}}_s = \mathbb{1} \otimes \bar{\mathbb{I}} \\ k^{(1)} &= \overline{B^0\left(A_s k^{(0)}\right)} = \overline{(A \otimes I)\left(\mathbb{1} \otimes \bar{\mathbb{I}}\right)} = A\mathbb{1} \otimes \bar{\bar{\mathbb{I}}} = c \otimes \bar{\bar{\mathbb{I}}} \\ k^{(2)} &= \overline{B^2\left(A_s k^{(0)}, 2A_s k^{(1)}\right)} = \overline{\left(A\mathbb{1} \otimes \bar{\mathbb{I}}\right) \cdot \left(A\mathbb{1} \otimes \bar{\mathbb{I}}\right)} + 2Ac \otimes \bar{\bar{\mathbb{I}}} = c \cdot c \otimes \overline{\bar{\mathbb{I}} \cdot \bar{\mathbb{I}}} + 2Ac \otimes \bar{\bar{\mathbb{I}}} \\ k^{(3)} &= c \cdot c \cdot c \otimes \overline{\bar{\mathbb{I}} \cdot \bar{\mathbb{I}} \cdot \bar{\mathbb{I}}} + c \cdot (2Ac) \otimes 2\overline{\bar{\mathbb{I}} \cdot \bar{\bar{\mathbb{I}}}} + (2Ac) \cdot c \otimes \overline{\bar{\bar{\mathbb{I}}} \cdot \bar{\mathbb{I}}} + 3A(c \cdot c) \otimes \overline{\bar{\bar{\mathbb{I}}} \cdot \bar{\bar{\mathbb{I}}}} + 6A^2 c \otimes \bar{\bar{\bar{\mathbb{I}}}} \end{aligned}$$

$$\begin{aligned} \hat{\nu}^{(0)} &= b_s k^{(0)} = b^T \mathbb{1} \otimes \bar{\mathbb{I}} \\ \hat{\nu}^{(1)} &= 2 b_s k^{(1)} = 2 b^T c \otimes \bar{\bar{\mathbb{I}}} \\ \hat{\nu}^{(2)} &= 3 b_s k^{(2)} = 3 b^T c^2 \otimes \overline{\bar{\mathbb{I}} \cdot \bar{\mathbb{I}}} + 6 b^T A c \otimes \bar{\bar{\bar{\mathbb{I}}}} \\ \hat{\nu}^{(3)} &= 4 b^T c^3 \otimes \overline{\bar{\mathbb{I}} \cdot \bar{\mathbb{I}} \cdot \bar{\mathbb{I}}} + 8 b^T (c \cdot Ac) \otimes \left(2\overline{\bar{\mathbb{I}} \cdot \bar{\bar{\mathbb{I}}}} + \overline{\bar{\bar{\mathbb{I}}} \cdot \bar{\mathbb{I}}}\right) + 12 b^T A c^2 \otimes \overline{\bar{\bar{\mathbb{I}}} \cdot \bar{\bar{\mathbb{I}}}} + 24 b^T A^2 c \otimes \bar{\bar{\bar{\mathbb{I}}}} \end{aligned}$$

Comparing this with (2.40)–(2.43), we obtain the order conditions

$$\begin{aligned} & b^T \mathbb{1} = 1 \\ & 2 b^T c = 1 \\ & 3 b^T c^2 = 1, \quad 6 b^T A c = 1 \\ & 4 b^T c^3 = 1, \quad 8 b^T (c \cdot Ac) = 1, \quad 12 b^T A c^2 = 1, \quad 24 b^T A^2 c = 1 \end{aligned}$$

These are the classical order conditions of Runge-Kutta methods. The crucial observation is that since the products on $\mathfrak{G}_s$ are commutative in the left-hand part, the order conditions are the same for elementary differentials which differ only by the ordering of the branches in the Butcher trees. Hence we must have the same order conditions in the commutative as in the non-commutative case.

THEOREM 3.3. *Let $A_{i,j}$ and b_j be coefficients which define a classical Runge-Kutta method of order q. If Assumption* 3.1 *holds, then the Algorithm* 3.1 *has order q on any Lie group G.*

3.1 Computing correction functions.

We want to study the following problem: Given $v_h \in \mathfrak{g}$ such that $v_0 = 0$, let $\tilde{v} = \zeta(v_h)$ and let $\hat{v}_h = d\exp_{\tilde{v}}(\tilde{v}')$. Determine $\zeta(\,)$ such that

$$\hat{v}^{(i)} = v^{(i+1)} \quad \text{for } i = 0, 1, \ldots, q-1. \tag{3.11}$$

We assume for the time being that $v^{(i)}$ can be computed. We will here show how to compute corrections up to order $q = 4$. A similar procedure can be employed to arbitrary order, but the computations become more complicated. From (2.9) we get

$$\hat{v}_h = d\exp_{\tilde{v}}(\tilde{v}') = \tilde{v}' - \frac{1}{2}[\tilde{v}, \tilde{v}'] + \frac{1}{6}[\tilde{v}, [\tilde{v}, \tilde{v}']] - \frac{1}{24}[\tilde{v}, [\tilde{v}, [\tilde{v}, \tilde{v}']]] + \ldots$$

We assume that $\tilde{v} = 0$ for $h = 0$. Using $[u, v]' = [u', v] + [u, v']$, differentiating and setting $h = 0$, we obtain:

$$\begin{aligned}
\hat{v}^{(0)} &= \tilde{v}^{(1)} \\
\hat{v}^{(1)} &= \tilde{v}^{(2)} \\
\hat{v}^{(2)} &= \tilde{v}^{(3)} - \frac{1}{2}\left[\tilde{v}^{(1)}, \tilde{v}^{(2)}\right] \\
\hat{v}^{(3)} &= \tilde{v}^{(4)} - \left[\tilde{v}^{(1)}, \tilde{v}^{(3)}\right] + \frac{1}{2}\left[\tilde{v}^{(1)}, \left[\tilde{v}^{(1)}, \tilde{v}^{(2)}\right]\right].
\end{aligned}$$

From this we find that the corrections hold up to order 4 provided

$$\begin{aligned}
\tilde{v}^{(1)} &= v^{(1)} \\
\tilde{v}^{(2)} &= v^{(2)} \\
\tilde{v}^{(3)} &= v^{(3)} + \frac{1}{2}\left[v^{(1)}, v^{(2)}\right] \\
\tilde{v}^{(4)} &= v^{(4)} + \left[v^{(1)}, v^{(3)}\right].
\end{aligned}$$

There are several ways to construct corrections compatible with these conditions. By differentiation we check that the following holds:

LEMMA 3.4. *The following functions give corrections up to order $q = 4$:*

$$\begin{aligned}
&\textit{Order 2} \quad : \quad \tilde{v} = v_h \\
&\textit{Order 3} \quad : \quad \tilde{v} = v_h + \frac{h}{6}\left[v^{(1)}, v_h\right] \\
&\textit{Order 4} \quad : \quad \tilde{v} = v_h + \frac{h}{4}\left[v^{(1)}, v_h\right] + \frac{h^2}{24}\left[v^{(2)}, v_h\right].
\end{aligned}$$

NOTE 3.2. These equations do not require that $v^{(i)}$ are exactly known, e.g., for order 4, it is sufficient to know an $\mathcal{O}(h^3)$ approximant for $v^{(1)}$ and an $\mathcal{O}(h^2)$ approximant for $v^{(2)}$. Such approximants can be computed from the k_i in the RK scheme.

We have $u^{(n)} = nA_s k^{(n-1)}$ and $v^{(n)} = nb_s k^{(n-1)}$, thus from Theorem 3.2

$$\begin{aligned} u^{(1)} &= A_s k^{(0)} = c \otimes \bar{\mathbb{I}} \\ u^{(2)} &= 2A_s k^{(1)} = 2Ac \otimes \bar{\bar{\mathbb{I}}} \\ v^{(1)} &= b_s k^{(0)} = b^T \mathbb{1} \otimes \bar{\mathbb{I}} = \bar{\mathbb{I}} \\ v^{(2)} &= 2b_s k^{(1)} = 2b^T c \otimes \bar{\bar{\mathbb{I}}} = \bar{\bar{\mathbb{I}}} \end{aligned}$$

Note that the elementary differentials $\bar{\mathbb{I}}, \bar{\bar{\mathbb{I}}}, \overline{\bar{\mathbb{I}} \cdot \bar{\mathbb{I}}}, \ldots \in \mathfrak{g}$, and can be computed to sufficient accuracy from k by linear algebra:

$$\begin{aligned} k &= (k_1, k_2, \ldots, k_s) = k^{(0)} + hk^{(1)} + \frac{h^2}{2} k^{(2)} + \mathcal{O}(h^3) \\ &= \mathbb{1} \otimes \bar{\mathbb{I}} + hc \otimes \bar{\bar{\mathbb{I}}} + \frac{h^2}{2} \left(c^2 \otimes \overline{\bar{\mathbb{I}} \cdot \bar{\mathbb{I}}} + 2Ac \otimes \bar{\bar{\bar{\mathbb{I}}}} \right) + \mathcal{O}(h^3) \end{aligned}$$

Let $d = Ac$. If A defines a 4-stage RK method with $A_{1,j} = 0$, $j = 1, 2, 3, 4$, then we get:

$$\begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & c_2 & c_2^2 & 2d_2 \\ 1 & c_3 & c_3^2 & 2d_3 \\ 1 & c_4 & c_4^2 & 2d_4 \end{pmatrix} \begin{pmatrix} \bar{\mathbb{I}} \\ h\bar{\bar{\mathbb{I}}} \\ \frac{h^2}{2} \overline{\bar{\mathbb{I}} \cdot \bar{\mathbb{I}}} \\ \frac{h^2}{2} \bar{\bar{\bar{\mathbb{I}}}} \end{pmatrix} + \begin{pmatrix} 0 \\ \mathcal{O}(h^3) \\ \mathcal{O}(h^3) \\ \mathcal{O}(h^3) \end{pmatrix}.$$

This equation can be solved for $\bar{\mathbb{I}}$ and $\bar{\bar{\mathbb{I}}}$ to the required accuracy.

3.2 Explicit RK-MK schemes of orders 3 and 4.

The algorithms in this section integrate the general initial value problem (2.35) on a general Lie group. In the matrix Lie group case (2.35) can be written as

$$y_t' = y_t \cdot f(y_t), \; y_0 = p, \quad \text{where } f : \mathrm{G} \to \mathfrak{g}. \tag{3.12}$$

ALGORITHM 3.2. **Explicit 3rd order RK-MK**
Choose the coefficients $A_{i,j}$ and b_j of a classical s-stage, 3rd order explicit RK scheme. Let $c_i = \sum_{j=1}^{s} A_{i,j}$.

$\hat{y}_0 = p$
`for` $n = 0, 1, 2, \ldots$
  $I_1 = k_1 = f(\hat{y}_n)$
  `for` $i = 2, \ldots, s$

$u_i = h\sum_{j=1}^{i-1} A_{i,j}k_j$
$k_i = f(\hat{y}_n \cdot \exp(u_i))$
end
$v = h\sum_{j=1}^{s} b_j k_j$
$\tilde{v} = v + \frac{h}{6}[I_1, v]$
$\hat{y}_{n+1} = \hat{y}_n \cdot \exp(\tilde{v})$
end

ALGORITHM 3.3. **Explicit 4th order RK-MK**
Choose the coefficients $A_{i,j}$ and b_j of a classical s-stage, 4th order explicit RK scheme. Let $c_i = \sum_{j=1}^{s} A_{i,j}$, and $d_i = \sum_{j=1}^{s} A_{i,j}c_j$. Compute the coefficients (m_1, m_2, m_3) by solving the linear system:

$$(m_1\ m_2\ m_3)\begin{pmatrix} c_2 & c_2^2 & 2d_2 \\ c_3 & c_3^2 & 2d_3 \\ c_4 & c_4^2 & 2d_4 \end{pmatrix} = (1\ 0\ 0)\,.$$

Then the 4th order RK-MK scheme is given as

$\hat{y}_0 = p$
for $n = 0, 1, 2, \ldots$
$I_1 = k_1 = f(\hat{y}_n)$
for $i = 2, \ldots, s$
$u_i = h\sum_{j=1}^{i-1} A_{i,j}k_j$
$\tilde{u}_i = u_i + \frac{c_i h}{6}[I_1, u_i]$
$k_i = f(\hat{y}_n \cdot \exp(\tilde{u}_i))$
end
$I_2 = (m_1(k_2 - I_1) + m_2(k_3 - I_1) + m_3(k_4 - I_1))/h$
$v = h\sum_{j=1}^{s} b_j k_j$
$\tilde{v} = v + \frac{h}{4}[I_1, v] + \frac{h^2}{24}[I_2, v]$
$\hat{y}_{n+1} = \hat{y}_n \cdot \exp(\tilde{v})$
end

Right-hand forms of these algorithms are obtained by rewriting (3.12) as

$$y_t' = \tilde{f}(y_t) \cdot y_t,$$

where $\tilde{f}(a) = \mathrm{Ad}(a)(f(a))$, changing the order of multiplication

$$k_i = \tilde{f}(\exp(\tilde{u}_i) \cdot \hat{y}_n), \quad \hat{y}_{n+1} = \exp(\tilde{v}) \cdot \hat{y}_n$$

and changing the sign of some of the correction terms, according to the sign differences in eqns. (2.9) and (2.10):

$$\begin{aligned} &\text{Order 2} \ : \ \tilde{u} = u_h \\ &\text{Order 3} \ : \ \tilde{u} = u_h - \frac{h}{6}\left[u^{(1)}, u_h\right] \\ &\text{Order 4} \ : \ \tilde{u} = u_h - \frac{h}{4}\left[u^{(1)}, u_h\right] - \frac{h^2}{24}\left[u^{(2)}, u_h\right]. \end{aligned}$$

3.3 A numerical example.

To illustrate the algorithms, we choose an example given by Zanna [21]. In this example $G = O(n)$, the space of orthogonal $n \times n$ matrices and $\mathfrak{g}$ is the space of skew-symmetric $n \times n$ matrices. We want to solve (3.12) where $f(y)$ is given (in Matlab notation) as:

$$f(y) = \mathrm{diag}(\mathrm{diag}(y, +1), +1) - \mathrm{diag}(\mathrm{diag}(y, +1), -1),$$

and the initial condition is the following random orthogonal 5×5 matrix:

$$\mathrm{rand}(\text{'seed'}, 0);\ [y_0, r] = \mathrm{qr}(\mathrm{rand}(5)).$$

We base our algorithms on the original 4 stage 4th order RK scheme given by the coefficients

$$\begin{array}{llll} A_{2,1} = 1/2, & A_{3,2} = 1/2, & A_{4,3} = 1, & \text{all other } A_{i,j} = 0, \\ B_1 = 1/6, & B_2 = 1/3, & B_3 = 1/3, & B_4 = 1/6. \end{array}$$

From this we compute

$$\begin{array}{llll} c_1 = 0, & c_2 = 1/2, & c_3 = 1/2, & c_4 = 1, \\ d_1 = 0, & d_2 = 0, & d_3 = 1/4, & d_4 = 1/2, \\ m_1 = 2, & m_2 = 2, & m_3 = -1. & \end{array}$$

The numerical experiments were performed by integrating from $t = 0$ to $t = 1$ using constant stepsizes, $h = \frac{1}{256}, \frac{1}{128}, \frac{1}{64}, \ldots, \frac{1}{2}$. Figure 3.1 shows the global error at $t = 1$ (measured in 2-norm) versus the stepsize h. RK-MK2 is the RK-MK scheme without any correction functions, RK-MK3 and RK-MK4 are corrected to orders 3 and 4. All these methods retain orthogonality perfectly. The line 'Classic RK4' is obtained by embedding (3.12) in $\mathbb{R}^{n^2}$ and use the classical RK scheme here. In this case orthogonality is gradually lost during the integration.

4 Comments added in proof.

In the period between the submission and the final proof reading of this article, there has been a major progress in the area of numerical integration techniques for differential equations on manifolds. In [15], the methods of this paper are generalized to homogeneous manifolds. In [14], we show how classical RK methods of *any* order can be modified to a Lie group invariant method of the same order, thus we solve the problem of finding correction functions of arbitrarily high order. There has also been a significant development of efficient methods specialized at equations of Lie type [10]. It is interesting to note that the new Lie group based methods not only give qualitatively better results than classical integrators (due to exact preservation of the group structure), but in several important examples it is also seen that, since they often have a much smaller error constant, they can beat classical integrators also in terms of accuracy versus floating point operations. We believe that the next few years will bring these new integrators more into the main stream of numerical analysis and software.

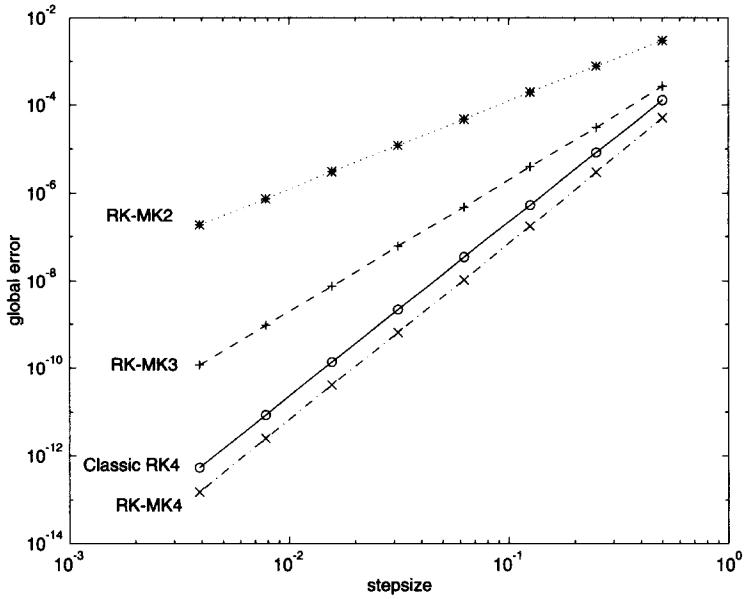

Figure 3.1: Global error versus stepsize for RK-MK schemes based on the classical 4 stage 4th order RK scheme.

Acknowledgments.

I would like to thank Antonella Zanna for many valuable comments and help with the numerical experiments.

REFERENCES

1. R. Abraham, J. E. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications.* AMS 75, Springer-Verlag, second edition, 1988.
2. R. L. Bryant, *An introduction to Lie groups and symplectic geometry*, in Geometry and Quantum Field Theory, D. S. Freed and K. K. Uhlenbeck, editors., Volume 1 of IAS/Park City Mathematics Series, AMS, second edition, 1995.
3. J. C. Butcher, *Coefficients for the study of Runge-Kutta integration processes*, J. Austral. Math. Soc., 3 (1963), pp. 185–201.
4. J. C. Butcher, *An algebraic theory of integration methods*, Math. Comput., 26 (1972), pp. 79–106.
5. M. P. Calvo, A. Iserles, and A. Zanna, *Runge-Kutta methods on manifolds*, in A. R. Mitchell's 75th Birthday Volume, D. F. Griffiths and G. A. Watson, editors, World Scientific, Singapore, pp. 57–70, 1996.
6. P. E. Crouch and R. Grossman, *Numerical integration of ordinay differential equations on manifolds*, J. Nonlinear Sci., 3 (1993), pp. 1–33.
7. E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations I*, Volume 8 of SCM. Springer-Verlag, Berlin, 1987.

8. A. Iserles, *Solving linear ordinary differential equations by exponentials of iterated commutators*, Numer. Math., 45:183–199, 1984.
9. A. Iserles, *Numerical methods on (and off) manifolds*, in F. Cucker, editor, Foundation of Computational Mathematics, pp. 180–189. Springer-Verlag, 1997.
10. A. Iserles and S. P. Nørsett, *On the solution of linear differential equations in Lie groups*, Technical Report 1997/NA3, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997.
11. A. Iserles, *Multistep methods on manifolds*, Technical Report 1997/NA13, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997.
12. A. Marthinsen, H. Munthe-Kaas, and B. Owren, *Simulation of ordinary differential equations on manifolds—some numerical experiments and verifications*, Modeling, Identification and Control, 18 (1997), pp. 75–88.
13. H. Munthe-Kaas. *Lie-Butcher theory for Runge-Kutta methods*, BIT, 35 (1995), pp. 572–587.
14. H. Munthe-Kaas, *High order Runge-Kutta methods on manifolds*, Technical Report 1997/NA14, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997.
15. H. Munthe-Kaas and A. Zanna, *Numerical integration of differential equations on homogeneous manifolds*, in Foundations of Computational Mathematics, F. Cucker, ed., Springer Verlag, pp. 305–315, 1997.
16. B. Owren and A. Marthinsen, *Integration methods based on rigid frames*, Technical Report Numerics No. 1/1997, Department of Mathematical Sciences, The Norwegian University of Science and Technology, 1997.
17. R. D. Russel, L. Dieci, and E. S. van Vleck, *Unitary integrators and applications to continuous orthonormalization techniques*, SIAM J. Numer. Anal., 31:1 (1994), pp. 261–281.
18. J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems*, AMMC 7, Chapman & Hall, 1994.
19. V. S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representation.* GTM 102, Springer-Verlag, 1984.
20. F. W. Warner, *Foundations of Differentiable Manifolds and Lie Groups*, GTM 94, Springer-Verlag, 1983.
21. A. Zanna, *The method of iterated commutators for ordinary differential equations on Lie groups*, Technical Report 1996/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1996.
22. A. Zanna and H. Munthe-Kaas, *Iterated commutators, Lie's reduction method and ordinary differential equations on matrix Lie groups*, in Foundation of Computational Mathematics, F. Cucker, ed., Springer-Verlag, pp. 434–441, 1997.