UNIVERSITY OF CAMBRIDGE

Numerical Analysis Reports

High Order Runge–Kutta Methods on Manifolds

Hans Munthe-Kaas

DAMTP 1997/NA14
August 1997

Department of Applied Mathematics and Theoretical Physics
Silver Street
Cambridge
England CB3 9EW

High Order Runge–Kutta Methods on Manifolds*

Hans Munthe-Kaas †
Department of Informatics
University of Bergen
N-5020 Norway

email: hans@ii.uib.no

August 11, 1997

Abstract

This paper presents a family of Runge–Kutta type integration schemes of arbitrarily high order for differential equations evolving on manifolds. We prove that *any* classical Runge–Kutta method can be turned into an invariant method of the same order on a general homogeneous manifold, and present a family of algorithms that are relatively simple to implement.

1 Introduction

The numerical integration of ordinary differential equations on manifolds is a subject that has received significant attention the last years. A major goal of this research has been to establish integration methods where the numerical solution is guaranteed to evolve on the same manifold as the analytical solution. This goal has been pursued by several authors, [4, 5, 7, 18, 19, 20, 21, 23, 25, 28, 29]. Although many of the basic results on which these methods rely were developed already in the late 19th and first half of the 20th century, it is just recently that the implications for practical numerical algorithms have been understood.

In [19] we established the connection between the Butcher theory for numerical integration of differential equations on $\mathbb{R}^n$ and a special form of Lie series on Lie groups, and we proposed a class of integration schemes that was later named RKMK methods. In its original formulation these methods could only achieve order two on a general Lie group. In [20] correction functions for the basic RKMK scheme was introduced, and we derived the order conditions for such methods of arbitrarily high order. A third and a fourth order method was explicitly derived, but it was unclear how to construct corrections of arbitrary order within this framework. In the paper [21], the RKMK methods were generalized to homogeneous manifolds.

*Presented at the NSF/CBMS Regional Conference on the Numerical Analysis of Hamiltonian Differential Equations, Golden Colorado, June 2-6 1997.
†Visiting scholar at DAMTP, Cambridge, Jan.–Dec. 1997

In the present paper, we formulate the correction process slightly differently, and show how to construct RKMK methods of arbitrary high order on homogeneous manifolds. The basic result is that any classical Runge–Kutta method can be turned into a method of the same order on a general homogeneous manifold, which stays exactly on the manifold. This result was first presented at the conference Foundations of Computational Mathematics, Rio, January 1997, although it did not appear in the published conference proceedings [21].

The goal of the present paper is to give a compact and concise presentation of the high order RKMK methods. A lot of the background motivation and consequences of these algorithms are not discussed. For further information, extensive reference lists, related and forthcoming papers within this topic, we recommend the homepage of the SYNODE project:

http://www.math.ntnu.no/num/synode/

An object oriented Matlab toolbox, *DiffMan*, for solving differential equations on manifolds is also found at this web-site.

2 Background theory and notation

2.1 Manifolds and tangent mappings

A manifold is a topological space $\mathcal{M}$ equipped with continuous local coordinate charts $\phi_i : U_i \subset \mathcal{M} \to \mathbb{R}^d$ such that all the overlap charts $\phi_{i,j} : \mathbb{R}^d \to \mathbb{R}^d$ are diffeomorphisms. For more details, see [1]. If $\phi : \mathcal{M} \to \mathcal{N}$ is a mapping between manifolds, we let $\phi' \equiv T\phi : T\mathcal{M} \to T\mathcal{N}$ denote the tangent mapping between the tangent manifolds. To avoid a cluttered notation, we make an exception for curves: if $y : \mathbb{R} \to \mathcal{M}$ then $y' : \mathbb{R} \to T\mathcal{M}$ denotes the curve $y'(t) = Ty(t, 1)$.

2.2 Lie algebras and algebra actions on a manifold

Definition 1 *A* Lie algebra *is a vector space* $\mathfrak{g}$ *equipped with a bilinear skew-symmetric bracket* $[\cdot,\cdot] : \mathfrak{g}\times\mathfrak{g} \to \mathfrak{g}$ *satisfying the Jacobi identity:*

$$[a,[b,c]] + [b,[c,a]] + [c,[a,b]] = 0 \ .$$

We call $[\cdot,\cdot]$ *the* Lie bracket *on* $\mathfrak{g}$.

We define $\mathrm{ad}(u) : \mathfrak{g} \to \mathfrak{g}$ as the linear map $\mathrm{ad}(u)(v) = [u, v]$, and $\mathrm{ad}^n(u)$ as the n–times iterated map, i.e.

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

As an example of an infinite dimensional Lie algebra, we consider a manifold $\mathcal{M}$ and let $\mathfrak{X}(\mathcal{M})$ be the set of all vector fields on $\mathcal{M}$. $\mathfrak{X}(\mathcal{M})$ has the structure of an $\mathbb{R}$-vector space, where

$$\begin{aligned} (X+Y)(p) &= X(p) + Y(p) \\ (r\cdot X)(p) &= r\cdot(X(p)) \end{aligned}$$

for $X, Y \in \mathfrak{X}(\mathcal{M})$, $p \in \mathcal{M}$, $r \in \mathbb{R}$. The *Lie-Jacobi bracket* $[\cdot,\cdot] : \mathfrak{X}(\mathcal{M})\times\mathfrak{X}(\mathcal{M}) \rightarrow \mathfrak{X}(\mathcal{M})$ turns $\mathfrak{X}(\mathcal{M})$ into an infinite dimensional Lie algebra. In local coordinates this bracket is defined as follows: If $X, Y, Z \in \mathfrak{X}(\mathcal{M})$ are three vector fields with components X^i, Y^i, Z^i, and if $Z = [X, Y]$, then

$$Z^i = X^j \frac{\partial Y^i}{\partial x^j} - Y^j \frac{\partial X^i}{\partial x^j} .$$

An important example of a finite dimensional Lie algebra is given by $\mathfrak{gl}(n)$, the linear space of all $n \times n$ matrices equipped with the matrix commutator as the bracket:

$$[a, b] = ab - ba .$$

Ado's theorem [27] states that *any finite dimensional* Lie algebra is isomorphic to a subalgebra of $\mathfrak{gl}(n)$.

We will define the action of a Lie algebra on a manifold. Let $\lambda : \mathfrak{g}\times\mathcal{M} \rightarrow \mathcal{M}$ be a smooth function. Each element $v \in \mathfrak{g}$ generates a vector field on $\mathcal{M}$, let $\lambda_* : \mathfrak{g} \rightarrow \mathfrak{X}(\mathcal{M})$ be defined as:

$$(\lambda_* v)(p) = \left.\frac{d}{dt}\right|_{t=0} \lambda(tv, p) . \tag{1}$$

Definition 2 *We call λ a (left)* Lie algebra action *if the induced map $\lambda_* : \mathfrak{g} \rightarrow \mathfrak{X}(\mathcal{M})$ is a Lie algebra anti-homomorphism, i.e. λ_* is a linear map between Lie algebras such that*

$$\lambda_*[u, v] = -[\lambda_* u, \lambda_* v] .$$

Computationally, the assumption of finite dimensionality of $\mathfrak{g}$ leads to simplifications, since the basic operations of computing brackets and actions then reduce to (finite dimensional) linear algebra. It should, however, be emphasized that the theory also apply to the infinite dimensional case, and that this may have important applications in numerical computations. In this case brackets must be computed by differentiation and actions by solving differential equations.

2.3 Lie groups

A *Lie group*[1] is a manifold G equipped with a continuous group product $\cdot : G\times G \rightarrow G$. A (left) Lie group action on a manifold is a mapping $\Lambda : G\times\mathcal{M} \rightarrow \mathcal{M}$ which satisfy:

$$\begin{aligned} \Lambda(e, p) &= p, \text{ where } e \in G \text{ is the identity,} \\ \Lambda(g_1 \cdot g_2, p) &= \Lambda(g_1, \Lambda(g_2, p)), \text{ for all } g_1, g_2 \in G, p \in \mathcal{M}. \end{aligned}$$

The Lie algebra of a Lie group is defined as the tangent space in the identity $\mathfrak{g} = TG|_e$. Its Lie bracket is given by:

$$[u, v] = \left.\frac{\partial^2}{\partial t \partial s}\right|_{t=s=0} g(t)\cdot h(s)\cdot g(t)^{-1},$$

[1] In the infinite dimensional case, we must replace Lie groups with *Lie pseudo groups*, which is defined in terms of a set of diffeomorphisms acting locally on a manifold [3]. Since we are only concerned with local actions, the replacement of Lie groups with Lie pseudo groups does not lead to serious technical difficulties. We will, however, not pursue this topic here.

where $g(t), h(s) \in G$ are two curves such that $g(0) = h(0) = e$, $g'(0) = u$, $h'(0) = v$. The *exponential mapping* is a function $\exp : \mathfrak{g} \to G$ defined in the following way: Let $R_y : G \to G$ denote right multiplication, $R_y(g) = g \cdot y$, and let

$$R_y{}' = T|_e R_y \ : \ \mathfrak{g} \to TG|_y \, .$$

Given a fixed $v \in \mathfrak{g}$, then $\exp(v)$ is defined as $\exp(v) = y(1)$, where $y(t) \in G$ satisfy the DE:

$$y' = R_y'(v), \ \ y(0) = e.$$

If G is a matrix group, then

$$\exp(v) = \sum_{i=0}^{\infty} \frac{v^i}{i!} \, .$$

Much of the recent work on numerical algorithms rely on the following result which may originally be traced back to Baker and Hausdorff [2, 10] and later to Magnus [16].

Theorem 1 *The differential of the exponential mapping is given as*

$$\exp'(u, v) = R'_{\exp u} \circ \mathrm{dexp}_u(v)$$

where $\mathrm{dexp}_u : \mathfrak{g} \to \mathfrak{g}$ *is the linear map:*

$$\mathrm{dexp}_u = \frac{\exp(\mathrm{ad}(u)) - I}{\mathrm{ad}(u)} = \sum_{j=0}^{\infty} \frac{1}{(j+1)!} \mathrm{ad}^j(u).$$

The inverse of dexp_u *is given as*

$$\mathrm{dexp}_u^{-1} = I - \frac{1}{2}\mathrm{ad}(u) + \sum_{j=2}^{\infty} \frac{B_k}{k!} \mathrm{ad}^k(u),$$

where B_k *is the* k*-th Bernoulli number. The first of the coefficients are given as:*

$$\frac{B_k}{k!} = \begin{cases} 0 & \text{for } k \text{ odd} \\ \frac{1}{12}, \frac{-1}{720}, \frac{1}{30240}, \frac{-1}{1209600} & \text{for } k = 2, 4, 6, 8. \end{cases}$$

There is an intimate connection between group actions and algebra actions:

Lemma 1 *If* $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ *is a (left) Lie group action, then* $\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$ *defined as*

$$\lambda(v, p) = \Lambda(\exp(v), p),$$

is a (left) Lie algebra action.

On the other hand, if we have a Lie algebra action, we may always assume that it locally arises from a group action.

Theorem 2 (Sophus Lie's third fundamental theorem [3, 15]) *Given a left Lie algebra homomorphism* $\lambda : \mathfrak{g} \to \mathcal{M}$*, there exist a Lie group* G *with Lie algebra* $\mathfrak{g}$ *and a left group action* $\Lambda : G \times G \to \mathcal{M}$ *such that*

$$\lambda(u, p) = \Lambda(\exp(u), p)$$

for all u *in some neighborhood of* 0.

3 Generic Presentation of Differential Equations on Manifolds

In the classical theory of numerical ODE integrators, it is generally assumed that the space on which the differential equation evolves is $\mathcal{M} = \mathbb{R}^n$ and that the differential equation is written in the form:

$$y' = F(t, y), \quad y(0) = p, \quad F : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n \ . \tag{2}$$

Furthermore, it is tacitly assumed that the numerical integration technique is expressed in terms of basic movements given by the set of all translations on $\mathbb{R}^n$.

For the purpose of discussing numerical integration of differential equations on manifolds, these assumptions will be replaced by much more general assumptions. The domains are defined as differentiable manifolds, and the basic movements by Lie algebra actions on the manifold:

Assumption 1 Generic Presentation of DEs on Manifolds: *We assume that there exist a Lie algebra $\mathfrak{g}$ with a Lie bracket $[\cdot,\cdot]$, a (left) Lie algebra action $\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$ and a function $f : \mathbb{R} \times \mathcal{M} \to \mathfrak{g}$ such that the equation for $y(t) \in \mathcal{M}$ can be written in the form:*

$$y' = \left(\lambda_* f(t, y)\right)(y) \, , \quad y(0) = p, \tag{3}$$

where λ_* is defined in (1).

This form of writing differential equations on manifolds is discussed in more details in [21]. If the algebra action is transitive then any differential equation on $\mathcal{M}$ can be written in this form. It is always possible to find (locally) a transitive action. E.g. for any local coordinate chart, the linear span of the basic vectorfields $\frac{\partial}{\partial x^i}$ constitute a commutative and transitive algebra acting on $\mathcal{M}$, via the coordinate flows. In this case $\mathfrak{g} \simeq \mathbb{R}^d$ and Algorithm 1 reduces to applying classical Runge–Kutta in local coordinates. There are, however, many other possible choices for the action, and by choosing a suitable action we can design numerical methods that exactly preserve important qualitative features of the equations. Different examples of manifolds and group actions are presented in Section 5.

4 RKMK methods of general order

Let us define the following truncated approximation for $d\exp_u^{-1}$:

$$\mathrm{dexpinv}(u, v, q) = v - \frac{1}{2}[u, v] + \sum_{j=2}^{q-2} \frac{B_k}{k!} \mathrm{ad}^k(u)(v).$$

Algorithm 1 RKMK : *Let $a_{i,j}$ and b_j be the coefficients of an s-stage, q'th order classical Runge–Kutta method and let $c_i = \sum_{j=1}^{s} a_{i,j}$. This algorithm integrates Equation (3) from $t = 0$ to $t = h$:*

$$
\begin{array}{l}
y_0 = p \\
\texttt{for } i = 1, 2, \ldots, s \\
\quad u_i = h \sum_{j=1}^{s} a_{i,j} \tilde{k}_j \\
\quad k_i = f(hc_i, \lambda(u_i, y_0)) \\
\quad \tilde{k}_i = \texttt{dexpinv}(u_i, k_i, q) \\
\texttt{end} \\
v = h \sum_{j=1}^{s} b_j \tilde{k}_j \\
y_1 = \lambda(v, y_0)
\end{array}
$$

Notes 4.1

- *If the underlying classical RK method is explicit, i.e. if $a_{i,j} = 0$ for $i \le j$, then the iteration can be done explicitly.*
- *The basic operations needed in this algorithm are the operations in the Lie algebra (sums, scalar products and Lie-brackets), and the computation of the algebra action λ. Unlike the formulations in the previous papers [20, 21], we avoid any operations in the Lie group G. This is a simplification from the point of view of specification of the necessary operations involved in Runge–Kutta computations, and in many cases it also yields a significant computational saving. There are many examples where the computation of the algebra action λ directly is much cheaper than the computation of the exponential mapping followed by a computation of the group action Λ.*
- *An important special case of (3) is when $f(t,p) = f(t)$ depends only on time. This is called* equations of Lie type, *or linear type equations. For these equations, it is evidently not necessary to compute the action λ for the s internal stages. Special methods for solving Lie type equations are presented in [14]. Algorithm 1 is comparable in efficiency to the more specialized algorithms, although a detailed comparison remains to be done. It might also in this case be profitable to do several Runge–Kutta steps in $\mathfrak{g}$ before advancing in $\mathcal{M}$, and thereby reducing the number of evaluations of λ.*

Theorem 3 *Algorithm 1 stays on the manifold $\mathcal{M}_p \subset \mathcal{M}$ given as:*

$$\mathcal{M}_p = \{\, q \in \mathcal{M} \mid q = \lambda(v_k, \ldots, \lambda(v_2, \lambda(v_1, p))) \;, \quad \textit{for some } v_1, \ldots, v_k \in \mathfrak{g} \,\}.$$

Proof: This follows trivially by the observations that all approximations take place in $\mathfrak{g}$. $\diamondsuit$

We say that a numerical method $y_0 = p \mapsto y_1(h)$ has order q if the first $q+1$ terms of the Lie series of $y_1(h)$ around $h = 0$ matches the first $q+1$ terms of the Lie series of the analytical solution of (3) around $t = 0$. The reader is referred to [19, 20, 23] for a thorough discussion of this topic.

Theorem 4 *Algorithm 1 has at least order q for any Lie algebra action λ on any manifold $\mathcal{M}$.*

Note 4.2 *There are important cases where the order is higher than q. Intuitively, this may happen if $y_1(h) = \lambda(h \cdot f(t,p),p)$ approximates the exact flow to an order higher than one. While the proof of Theorem 4 is relatively simple, and does not require the machinery of Lie–Butcher theory [19, 20], the treatment of the precise conditions under which order becomes higher, seems to be best dealt with using the Lie–Butcher theory. This issue will not be discussed in the present paper.*

A short proof of Theorem 4 is based on the concept of ϕ-relatedness of vector fields [1]. In the case when ϕ is a smooth invertible mapping between manifolds, this is equivalent to pullback of vector fields. The basic idea is to transform the differential equation on $\mathcal{M}$ to an equivalent equation on $\mathfrak{g}$ using 'pullback' along λ. Since $\mathfrak{g}$ is a linear space, integration in $\mathfrak{g}$ is simpler than on $\mathcal{M}$. However, since we do not require $\lambda(\cdot,p)$ to be an invertible mapping, we talk about relatedness rather than pullbacks.

Given two manifolds $\mathcal{N}$ and $\mathcal{M}$ and a mapping $\phi : \mathcal{N} \to \mathcal{M}$. Two vector fields $G \in \mathfrak{X}(\mathcal{N})$ and $F \in \mathfrak{X}(\mathcal{M})$ are said to be *ϕ-related* if $\phi' \circ G = F \circ \phi$. We write this as $G \sim_\phi F$. Consider the two differential equations for $y(t) \in \mathcal{M}$ and $u(t) \in \mathcal{N}$:

$$\begin{aligned} y' &= F(y), \;\; y(0) = \phi(u_0) \\ u' &= G(u), \;\; u(0) = u_0 \ . \end{aligned}$$

It is straightforward to check that if $G \sim_\phi F$ then $y(t) = \phi(u(t))$. Now let $\lambda_p(u) = \lambda(u,p)$. We will use λ_p to 'pull back' Equation (3) from $\mathcal{M}$ to $\mathfrak{g}$.

Lemma 2 *If $F \in \mathfrak{X}(\mathcal{M})$ is the vector field $F(p) = (\lambda_* f(p))(p)$ for some $f : \mathcal{M} \to \mathfrak{g}$, and if $\tilde{f} \in \mathfrak{X}(\mathfrak{g})$ is the vector field*

$$\tilde{f}(u) = \mathrm{dexp}_u^{-1}(f \circ \lambda_p(u)),$$

then $\tilde{f} \sim_{\lambda_p} F$.

Proof: We introduce a (local) group action $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ such that $\lambda(u,p) = \Lambda(\exp(u),p)$, and let $\Lambda_p(g) = \Lambda(g,p)$. From Theorems 1,2 we get:

$$\begin{aligned} {\lambda_p}' \circ \tilde{f}(u) &= (\Lambda_p \circ \exp(u))' \circ \tilde{f}(u) \\ &= {\Lambda_p}' \circ R'_{\exp(u)} \circ \mathrm{dexp}_u \circ \mathrm{dexp}_u^{-1}(f \circ \lambda_p(u)) = {\Lambda_p}' \circ R'_{\exp(u)} (f \circ \lambda_p(u)) \ . \end{aligned}$$

For any $u, v \in \mathfrak{g}$ we have

$$\begin{aligned} \lambda_*(v)(\lambda_p(u)) &= \left.\frac{d}{dt}\right|_{t=0} \Lambda\left(\exp(tv), \Lambda(\exp(u),p)\right) \\ &= \left.\frac{d}{dt}\right|_{t=0} \Lambda\left(\exp(tv) \cdot \exp(u), p\right) = {\Lambda_p}' \circ R'_{\exp(u)}(v) \ . \end{aligned}$$

This gives

$$F \circ \lambda_p(u) = \lambda_*\left(f \circ \lambda_p(u)\right)\left(\lambda_p(u)\right) = {\Lambda_p}' \circ R'_{\exp(u)}(f \circ \lambda_p(u)) = {\lambda_p}' \circ \tilde{f}(u). \qquad \diamondsuit$$

Hence, we obtain the following important result:

Corollary 1 *The solution of (3) is given, for sufficiently small* t, *as*

$$y(t) = \lambda(u(t), p),$$

where $u(t) \in \mathfrak{g}$ *satisfy the differential equation*

$$u' = \tilde{f}(t, u) = d\mathrm{exp}_u^{-1}(f\,(t, \lambda(u, p)))\ ,\quad u(0) = 0\ . \tag{4}$$

Proof of Theorem 4: This theorem now follows by observing that Algorithm 1 is equivalent to doing a single classical Runge–Kutta step in $\mathfrak{g}$ on the differential equation (4), to produce a numerical approximation $u_1 \approx u(h)$, and then advancing the solution on $\mathcal{M}$ as $y_1 = \lambda(u_1, y_0)$. Since $\mathfrak{g}$ is a vector space, the classical theory of Runge–Kutta methods can be applied to derive the order conditions of $a_{i,j}$ and b_j, and since λ is a smooth mapping, we conclude that the order on $\mathcal{M}$ is at least as high as the order of the corresponding equation on $\mathfrak{g}$. Finally, we observe that the approximation error in $d\mathrm{exp}_u^{-1}$ introduces an $\mathcal{O}(q+1)$ modification of $\tilde{f}$ that does not reduce the order of the scheme. $\diamondsuit$

5 Examples

We start by briefly giving various examples of differential equations written in the form given in Assumption 1.

Example 1 (Classical RK setting) *Let* $\mathfrak{g} = \mathcal{M} = \mathbb{R}^n$, $[u, v] = 0$ *and* $\lambda(v, p) = v + p$. *In this case (3) reduce to the form in (2), and Algorithm 1 reduces to a classical Runge–Kutta scheme.*

Example 2 (Differential equations on Lie groups) *Let* $\mathcal{M} = G$ *be a matrix Lie group,* $(\mathfrak{g}, [\cdot,\cdot])$ *its Lie algebra and* $\lambda(v, p) = \exp(v)\cdot p$. *Then (3) reduces to:*

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

and Algorithm 1 will stay exactly on G.

Example 3 (Isospectral flows) *Let* $\mathcal{M} \subset \mathbb{R}^{n\times n}$, *let* $G = \mathrm{SO}(n)$ *be the orthogonal group,* $\mathfrak{g} = \mathfrak{so}(n)$ *its Lie algebra and let* $\lambda(v, p) = \exp(v)\cdot p\cdot\exp(-v)$. *Then (3) reduces to the isospectral equation:*

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

and Algorithm 1 will exactly preserve isospectrality. This example is treated in [4].

Example 4 (Stiff systems) *Consider a differential equation on* $\mathbb{R}^d$ *written in the standard form (2). When this system is stiff, it is well known that one must employ implicit classical integration methods. One type of stiffness arise when the Jacobian of* F *is ill-conditioned. In this case we may introduce the action on* $\mathcal{M}$ *obtained by exactly integrating all linear equations of the form:* $y' = Ay + b$, *where* A *and* b *are constant. Formally we proceed as follows: We let* $G = \mathrm{GL}(d) \rtimes \mathbb{R}^d$ *be the* semidirect product *[27] of the general linear group and* $\mathbb{R}^d$. G *is the group of all affine linear maps acting on* $\mathbb{R}^d$ *as:*

$$\Lambda\left((A, b), y\right) = Ay + b\ .$$

The Lie algebra of G is given as $\mathfrak{g} = \mathfrak{gl}(d) \rtimes \mathbb{R}^d$ with the Lie bracket

$$\left[(A,b),(\tilde{A},\tilde{b})\right] = \left(\left[A,\tilde{A}\right], A\tilde{b} - \tilde{A}b\right)$$

and exponential mapping

$$\exp(A,b) = (\exp(A), \operatorname{dexp}_A(b)) \ ,$$

where

$$\operatorname{dexp}_A(b) = \frac{\exp(A) - I}{A} \cdot b = \sum_{j=0}^{\infty} \frac{1}{(j+1)!} A^j b \ .$$

We let the algebra action be given as

$$\lambda((A,b),y) = \Lambda\left(\exp(A,b),y\right) = \exp(A)y + \operatorname{dexp}_A(b) \ .$$

In this case the map $\lambda_*(\cdot)(p) : \mathfrak{g} \to T\mathcal{M}|_p$ has a nontrivial kernel, and hence there is no unique way to choose the mapping $f : \mathbb{R} \times \mathbb{R}^d \to \mathfrak{g}$ to turn (2) into the form (3). One choice is:

$$f(t,y) = (0, F(t,y)) \ ,$$

which will just recover the classical methods for solving (2). Another choice is

$$f(t,y) = (J, F(t,y) - Jy) \ , \quad \text{where } J \text{ is the Jacobian of } F.$$

In this case we get methods which solve linear systems exactly, and hence Algorithm 1 becomes A-stable, even in the explicit form. In the case where Algorithm 1 is based on forwards Euler, the resulting method spells out:

$$y_{n+1} = y_n + \operatorname{dexp}_{hJ_n}(hF_n) \ ,$$

which is an explicit second order A-stable method found in Nørsett [22]. Related methods has recently been developed in [11]. This example illustrate several important facts: The order of Algorithm 1 may be higher than the order of the underlying classical scheme if the action tangents the real flow to an order higher than one. We also see that the stiffness properties of the algorithm may depend on the choice of action, and also on the choice of f. This has also been observed in the solution of Riccati type equations [26].

Example 5 (Riccati equations) *The Riccati equation*

$$y'(t) = a_0(t) + 2a_1(t)y(t) + a_2(t)\left(y(t)\right)^2 \ , \quad y(t) \in \mathbb{R},$$

is a model example of a differential equation of Lie type. This particular version of the equation can be written as a special form of (3) by where $G = \mathrm{SL}(2,\mathbb{R})$ is the set of real 2×2 matrices with determinant 1 and $\mathfrak{g}$ its Lie algebra. Furthermore, we let $\mathcal{M} = \mathbb{R}$, $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ be the Möbius transformation:

$$\Lambda\left(\begin{pmatrix} a & b \\ c & d \end{pmatrix}, y\right) = \frac{ay+b}{cy+d} \ ,$$

$\lambda(v,y) = \Lambda(\exp(v),y)$, and $f : \mathbb{R} \to G$ be given as:

$$f(t) = \begin{pmatrix} a_1(t) & a_0(t) \\ -a_2(t) & -a_1(t) \end{pmatrix} \ .$$

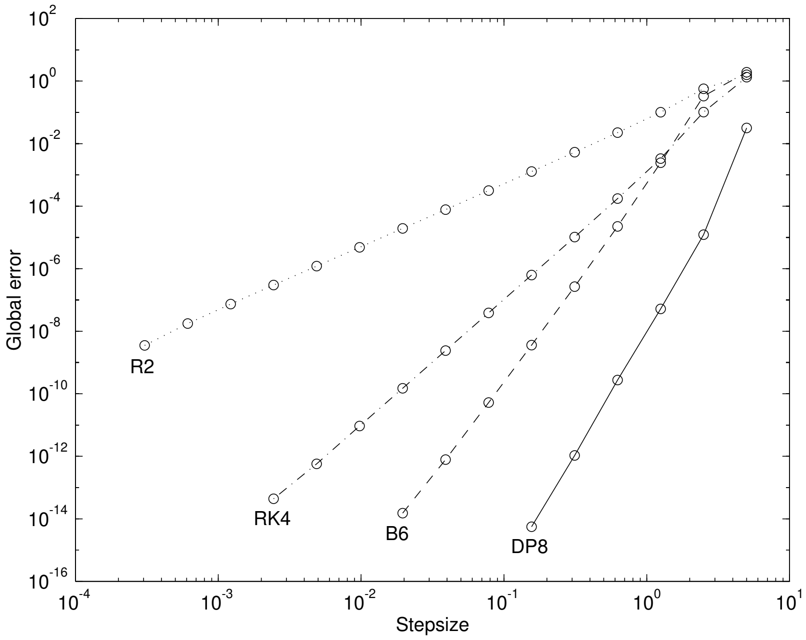

Figure 1: Global error versus stepsize (Example 8).

This example is discussed in [3, 26]. Numerical results show that integration schemes based on this natural action avoid problems with singularities that are seen in classical coordinate based integration schemes.

Example 6 (Rigid frames) *A* frame *on $\mathcal{M}$ is a set of vector fields $E_1, \ldots, E_m \in \mathfrak{X}(\mathcal{M})$ which at each point $p \in \mathcal{M}$ span the tangent space $T\mathcal{M}|_p$. A differential equation on $\mathcal{M}$ can be written in terms of a frame as:*

$$y' = \sum_{i=1}^{m} f_i(y) E_i, \quad \text{where } f_i : \mathcal{M} \to \mathbb{R} \text{ are smooth.} \tag{5}$$

Let $\mathfrak{g} \subset \mathfrak{X}(\mathcal{M})$ denote the Lie sub-algebra of $\mathfrak{X}(\mathcal{M})$ generated by E_i. Let $\mathrm{Diff}(\mathcal{M})$ be the Lie (pseudo) group of diffeomorphisms on $\mathcal{M}$, and let $\exp : \mathfrak{g} \to \mathrm{Diff}(\mathcal{M})$ denote the mapping that maps a vector field to its flow. Let $\Lambda : \mathrm{Diff}(\mathcal{M}) \times \mathcal{M} \to \mathcal{M}$ be the group action of flow evaluation, and let $\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$ be the Lie algebra action

$$\lambda(E, p) = \exp(E)(p).$$

If $f : \mathcal{M} \to \mathfrak{X}(\mathcal{M})$ is given as

$$f(y) = \sum_{i=1}^{m} f_i(y) E_i,$$

then (3) reduces to the special form given in (5). Algorithm 1 then becomes a method for integrating differential equations written in terms of rigid frames, and the basic operation involved in computing the action $\lambda(v, p)$, becomes the

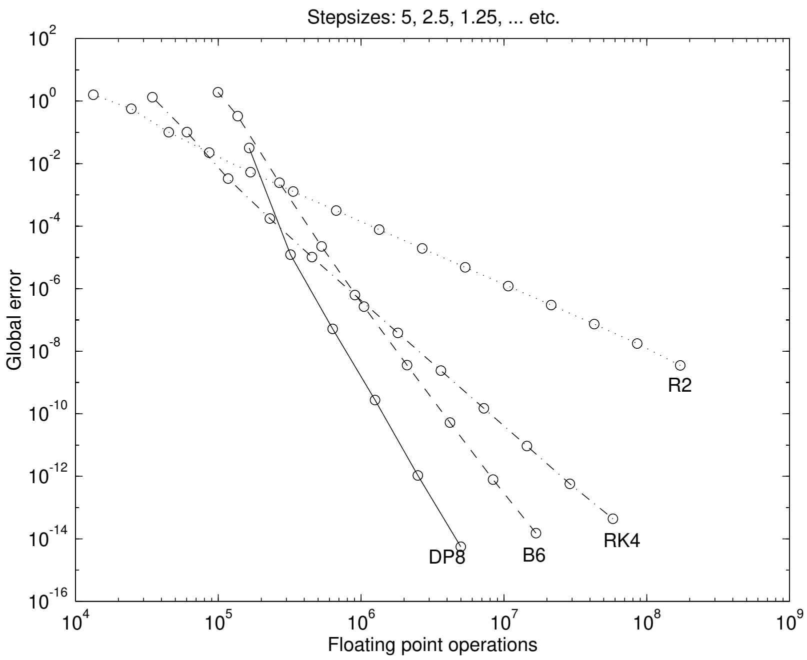

Figure 2: Global error versus floating point operations (Example 8).

task of following a fixed flow in the algebra spanned by $\{E_i\}$. *The Crouch–Grossman family of algorithms discussed in [7, 23] is an alternative approach to integrating equations in this form.*

Example 7 (The heavy top) *The heavy top equations [17] can be nicely formulated as an equation on* $\mathfrak{g}^*$, *the dual of* $\mathfrak{g}$, *where* $\mathfrak{g}$ *is the Lie algebra of* $T\mathrm{SO}(3)$. *In this case, an action that will preserve important components of the angular momentum is given by the* coadjoint action *[17], and in this formulation the equations fits very nicely into the form (3). This approach to numerical integration of the heavy top equations will be discussed in a forthcoming paper. We believe that this example is important for understanding the numerical treatment of several important equations in mathematical physics, which possess a similar structure. Examples are the Euler equations in fluid flow, and the Maxwell-Vlasov equations in plasma theory.*

Example 8 (A numerical example) *As a simple illustration of the algorithms, we choose an example where* $\mathfrak{g}$, λ, G *and* $\mathcal{M}$ *are as in Example 2, and where* $G = \mathrm{SO}(4)$, *the set of orthogonal* 4×4 *matrices. The right hand side* $f(y)$ *is given (in Matlab notation) as:*

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

and the initial condition is the following random orthogonal $4x4$ *matrix:*

$$rand(\text{'}seed\text{'}, 0);\ [y_0, r] = qr(rand(4, 4)) \ .$$

The numerical experiments were performed by integrating from $t = 0$ *to* $t = 10$, *and successively halving the stepsizes:* $h = 5, 2.5, \ldots$. *Figure 1 shows the global* *error at $t = 10$ (measured in 2-norm) versus the stepsize h, while Figure 2 shows error vs. the number of floating point operations performed. R2 is based on Runges 2. order method, RK4 on the classical 4. order Runge–Kutta method, B6 on Butchers 6. order method and DP8 on the 8. order method of Dormand and Prince. The classical versions of these algorithms are found in [9]. They are all modified according to Algorithm 1, and hence preserve orthogonality exactly.*

6 Final remarks

This paper can be seen as a sequel to the three papers [19, 20, 21]. The main open questions of these has been answered, the construction of higher order RKMK type methods has been considerably simplified, and we have gained new insight into the structure of these algorithms. In the new light, they now appear as versions of Runge–Kutta methods employing *canonical coordinates of the first kind* [27], which on a Lie group are coordinates obtained in the neighborhood of a point p by inverting the map: $(x_1, x_2, \ldots, x_d) \mapsto \exp(x_1 v_1 + x_2 v_2 + \cdots + x_d v_d) \cdot p$ for some basis $\{v_i\}$ of $\mathfrak{g}$.

Although the Lie–Butcher theory of [19, 20] is no longer necessary to provide the basic order proof, this theory is still very important for understanding the algorithms in more detail, and establishing the conditions under which the order of the methods are increased.

The theory of this paper allow us to pose new questions, which seems possible to tackle with the current machinery, and which might be the topic of forthcoming papers:

- Crouch–Grossman type methods [7, 23] is related to *canonical coordinates of the second kind* [27]. Can a correction technique be employed to develop such methods of higher order?

- In the present formulation of the methods, we change coordinates for every step. We could consider methods that take several steps in $\mathfrak{g}$ before a step is performed in $\mathcal{M}$. Especially for equations of Lie type this might lead to savings, and one might also consider multistep methods in $\mathfrak{g}$. The main problem of this could be that the conditioning of the coordinate transform becomes worse as we move away from the origin in $\mathfrak{g}$ and eventually the exponential mapping is no longer a diffeomorphism.

- If the manifold $\mathcal{M}$ possesses additional structures such as e.g. a symplectic two-form $\omega : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathbb{R}$, is it possible to devise symplectic methods on $\mathfrak{g}$ which preserve this form? A way to treat this question might be to pull back ω from $\mathcal{M}$ to $\mathfrak{g}$.

- The numerical analysis of ODEs have now been enriched by an additional choice, in addition to choosing the basic integration method, we have freedom to choose different $\mathfrak{g}$ and different algebra actions λ. How can we choose actions that preserve properties such as e.g. momentum or components of the momentum in mechanical systems? For which kinds of stiff systems is it possible to find actions that remove the stiffness?

7 Acknowledgments

I would like to thank Brynjulf Owren and Bruno Welfert for pointing out the formula for $d\exp_u^{-1}$ in Theorem 1 and Per Christian Moan for providing the reference [16]. I would also like to thank Arne Marthinsen, Syvert P. Nørsett, Jeremy Schiff and Antonella Zanna for valuable discussions on the topics of this paper. In particular, I would like to thank Arieh Iserles, both for mathematical inputs, and also for his hospitality during my visit to DAMTP.

References

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

[2] H.F. Baker. Alternants and continuous groups. *Proc. of London Math. Soc., Second series*, 3:24–47, 1905.

[3] R.L. Bryant. An Introduction to Lie Groups and Symplectic Geometry. In D.S. Freed and K.K. Uhlenbeck, editors, *Geometry and Quantum Field Theory*, volume 1 of *IAS/Park City Mathematics Series.* AMS, second edition, 1995.

[4] M.P. Calvo, A. Iserles, and A. Zanna. Numerical Solution of Isospectral Flows. Technical Report 1995/NA03, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1995.

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

[6] F. Casas. Solution of Linear Partial Differential Equations by Lie Algebraic Methods. *J. Comput. Appl. Math*, 76:159–170, 1996.

[7] P.E. Crouch and R. Grossman. Numerical Integration of Ordinay Differential Equations on Manifolds. *J. Nonlinear Sci.*, 3:1–33, 1993.

[8] F. Fer. Résolution de l'équation matricielle $\frac{\partial}{\partial t}U = pU$ par produit infini d'exponentielles matricielles. *Bull. de l'Acad. Roy. de Belgique, classe des Sciences*, 36:818–829, 1958.

[9] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*, volume 8 of *SCM.* Springer-Verlag, Berlin, 1987.

[10] F. Hausdorff. Die symbolische exponentialformel in der gruppentheorie. *Berichte der Sächsischen Akademie der Wissenschaften*, 58:19–48, 1906.

[11] Marlis Hochbruck, Christian Lubich, and Hubert Selhofer. Exponential integrators for large systems of differential equations. Technical report, Department of Mathematcs, University of Tuebingen, 1995. To appear in SIAM J. Sci. Comp.

[12] A. Iserles. Solving Linear Ordinary Differential Equations by Exponentials of Iterated Commutators. *Numer. Math.*, 45:183–199, 1984.

[13] A. Iserles. Numerical methods on (and off) manifolds. In F. Cucker, editor, *Foundation of Computational Mathematics*, pages 180–189. Springer-Verlag, 1997.

[14] 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.

[15] Sophus Lie and F. Engel. *Theorie der Transformationsgruppen*. B.G. Teubner, Leipzig, 1888,1890,1893.

[16] W. Magnus. On the Exponential Solution of Differential Equations for a Linear Operator. *Comm. Pure and Appl. Math.*, VII:649–673, 1954.

[17] J.E. Marsden and T.S. Ratiu. *Introduction to Mechanics and Symmetry*. Springer-Verlag, 1994.

[18] A. Marthinsen, H. Munthe-Kaas, and B. Owren. Simulation of Ordinary Differential Equations on Manifolds — Some Numerical Experiments and Verifications. *Modeling, Identification and Control*, 18:75–88, 1997.

[19] H. Munthe-Kaas. Lie-Butcher Theory for Runge-Kutta Methods. *BIT*, 35:572–587, 1995.

[20] H. Munthe-Kaas. Runge-Kutta Methods on Lie Groups. Technical Report 122, Department of computer science, University of Bergen, Norway, 1997. To appear in *BIT*.

[21] H. Munthe-Kaas and A. Zanna. Numerical integration of differential equations on homogeneous manifolds. In F. Cucker, editor, *Foundations of Computational Mathematics*, pages 305–315. Springer Verlag, 1997.

[22] Syvert P. Nørsett. The numerical solution of stiff systems. Master's thesis, University of Oslo, Norway, 1969.

[23] 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.

[24] B. Owren and B. Welfert. The Newton Iteration on Lie Groups. Technical Report Numerics No. 3/1996, Norwegian University of Science and Technology, Trondheim, Norway, 1996.

[25] J.M. Sanz-Serna and M.P. Calvo. *Numerical Hamiltonian Problems*. AMMC 7. Chapman & Hall, 1994.

[26] Jeremy Schiff and S. Shnider. A Natural Approach to the Numerical Integration of Riccati Differential Equations. Technical report, Department of Mathematics and Computer Science, Bar Ilan University, Israel, 1996.

[27] V.S. Varadarajan. *Lie Groups, Lie Algebras, and Their Representation*. GTM 102. Springer-Verlag, 1984.

[28] 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.

[29] A. Zanna and H. Munthe-Kaas. Iterated commutators, Lie's reduction method and ordinary differential equations on matrix Lie groups. In F. Cucker, editor, *Foundation of Computational Mathematics*, pages 434–441. Springer-Verlag, 1997.