ELSEVIER

Applied Numerical Mathematics 29 (1999) 115–127

APPLIED NUMERICAL MATHEMATICS

High order Runge–Kutta methods on manifolds ☆

Hans Munthe-Kaas *

Department of Informatics, University of Bergen, N-5020 Bergen, Norway

Abstract

We present 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. These are defined in a general abstract framework, based on a Lie algebra acting on the manifold. The general framework gives rise to a wide range of different concrete applications; we present some examples.

1. Introduction

Numerical integration of ordinary differential equations (ODEs) on manifolds has received significant attention recently. A major goal 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, [4–6,14–16,18,21,22,25,26]. Although many basic results on which these methods rely were developed already in the late 19th and first half of the 20th century, only recently have the implications for practical numerical algorithms been understood.

In [15], we established the connection between the Butcher order theory for numerical integration of ODEs on $\mathbb{R}^n$ and a special form of Lie series on Lie groups, and we proposed a class of integration schemes later named RKMK methods. In their original formulation, these methods could achieve only order two on a general Lie group. In [16], correction functions for the basic RKMK scheme were introduced, and the order conditions derived for such methods of arbitrarily high order. Third- and fourth-order methods were explicitly derived, but it was unclear how to construct higher order methods within this framework. In [18], the methods were generalized to homogeneous manifolds.

Here, we formulate the correction process slightly differently, and construct RKMK methods of arbitrarily 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. This result

☆ Partly sponsored by NFR through the SYNODE project, contract no. 111038/410.
* E-mail: hans.munthe-kaas@ii.uib.no.

0168-9274/99/$19.00
PII: S0168-9274(98)00030-0

was first presented at the Conference on Foundations of Computational Mathematics, Rio de Janeiro, January 1997, but did not appear in the paper published in the conference proceedings [18].

The goal of this paper is to give a compact and concise presentation of high order RKMK methods. Much of the background motivation and implications of these algorithms is not discussed. For further information, extensive reference lists, and related and forthcoming papers on 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 ODEs on manifolds is also available from this web-site.

2. Background theory and notation

First, we define manifolds, Lie algebras and Lie algebra actions. Manifolds provide us with the abstract definition of the domains where the ODEs are evolving. Lie algebras and Lie algebra actions give us the structures for defining movements on the manifold. Our numerical schemes advance by following flows defined by actions on a time interval h. From an action $\lambda(v,\cdot)$, we can compute the infinitesimal generator of the action, $\lambda_* v$, which defines directions tangent to the manifold. Using this, we write ODEs on the manifold in a generic form suitable for theoretical studies and practical computer implementations.

2.1. Manifolds and tangent mappings

A d-dimensional 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, see [1]. If $\phi : \mathcal{M} \to \mathcal{N}$ is a mapping between manifolds, 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 $y'(t) = Ty(t,1)$. Here, we employ the usual identification of $\mathbb{R}$ and $T\mathbb{R} = \mathbb{R} \times \mathbb{R}$ given by $t \mapsto (t,1)$.

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 $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ (the *Lie bracket* on $\mathfrak{g}$) satisfying the Jacobi identity:

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

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

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

As an example of an infinite dimensional Lie algebra, 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

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

for $X, Y \in \mathfrak{X}(\mathcal{M})$, $p \in \mathcal{M}$, $r \in \mathbb{R}$. The $(-)$*Lie–Jacobi bracket*, $[\cdot,\cdot]^-_{\mathrm{LJ}} : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M})$, 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 vector fields with components X^i, Y^i, Z^i, and if $Z = [X, Y]^-_{\mathrm{LJ}}$, then

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

Note, many authors use the $(+)$Lie–Jacobi bracket $[\cdot,\cdot]^+_{\mathrm{LJ}} = -[\cdot,\cdot]^-_{\mathrm{LJ}}$.

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

$$(\lambda_* v)(p) = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \lambda(tv, p) \quad \text{for all } v \in \mathfrak{g},\ p \in \mathcal{M}. \tag{1}$$

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

$$\lambda_*[u, v] = [\lambda_* u, \lambda_* v]^-_{\mathrm{LJ}}.$$

Note. If the definition is based on $[\cdot,\cdot]^+_{\mathrm{LJ}}$ then λ_* is an *anti*-homomorphism.

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

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

Ado's theorem [24] states that *any finite dimensional* Lie algebra is isomorphic to a subalgebra of $\mathfrak{gl}(n)$. A useful approach to understanding abstractly defined operations is to write them in concrete matrix representation. However, for the purpose of computer implementations (such as the `DiffMan` package) it is crucial to treat Lie algebras as abstractly defined objects rather than via their matrix representations. This permits us to juggle internal representations (as matrices or as other data structures) according to what is most convenient. The abstract understanding is also necessary to build automatically more complicated algebras from simpler ones. Note, the theory also applies to the infinite dimensional case, and this may have important applications in numerical computations. In this case, brackets must be computed by differentiation and actions by integrating differential equations.

2.3. Lie groups

Strictly speaking, Lie groups and Lie group actions are not needed to define and implement the algorithms of this paper, but are introduced to improve understanding of Lie algebra actions and to simplify some proofs. A *Lie group* is a manifold G equipped with a continuous group product $\cdot : G \times G \to G$. A (left) Lie group action is a mapping $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ which satisfies

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

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

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

where $g(t), h(s) \in G$ are 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 as follows: let $R_y : G \to G$ denote right multiplication, $R_y(g) = g \cdot y$, and

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

Given a fixed $v \in \mathfrak{g}$, $\exp(v) = y(1)$, where $y(t) \in G$ satisfies the ODE

$$y' = R'_y(v), \qquad y(0) = e.$$

If G is a matrix group, then $R'_y(v) = vy$ and

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

Much recent work on numerical algorithms relies on the following result which can be traced to Baker and Hausdorff [2,8] and later to Magnus [12].

Theorem 3. *The differential of the exponential mapping* $\exp' : T\mathfrak{g} = \mathfrak{g} \times \mathfrak{g} \to TG$ *is*

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

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

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

The inverse of dexp_u *is*

$$\operatorname{dexp}_u^{-1} = \sum_{k=0}^{\infty} \frac{B_k}{k!} \operatorname{ad}_u^k,$$

where B_k is the kth Bernoulli number. The first few coefficients are

$$\frac{B_k}{k!} = \begin{cases} 0 & \text{for } k \text{ odd, and } k \neq 1, \\ 1, -\frac{1}{2}, \frac{1}{12}, -\frac{1}{720}, \frac{1}{30240}, -\frac{1}{1209600}, \frac{1}{47900160} & \text{for } k = 0, 1, 2, 4, 6, 8, 10. \end{cases}$$

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

Lemma 4. *If* $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ *is a (left) Lie group action, then*

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

defines a (left) Lie algebra action.

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

Theorem 5 [3]. *Given a Lie group G with algebra $\mathfrak{g}$. For any homomorphism $\lambda_* : \mathfrak{g} \to \mathfrak{X}(\mathcal{M})$, there exist a left group action $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ such that*

$$\lambda_*(u)(p) = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \Lambda(\exp(u), p)$$

for all $p \in \mathcal{M}$ and all u in some neighborhood of 0.

Note. An algebra action λ is not uniquely determined from a group action Λ. Every diffeomorphism $\phi:\mathfrak{g} \to G$ such that $\phi(0) = e$, $\phi'(0) = I$, define Lie algebra actions via $\lambda_\phi(u, p) = \Lambda(\phi(u), p)$. However, they all generate the same homomorphism $\lambda_*(u)(p) = (\mathrm{d}/\mathrm{d}t)|_{t=0}\lambda_\phi(tu, p)$. In this paper we will exclusively work with actions of the form $\lambda(u, p) = \Lambda(\exp(u), p)$, but other choices of ϕ are worth investigating.

3. Generic presentation of differential equations on manifolds

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

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

Also, it is tacitly assumed that the numerical integration technique is expressed in terms of basic movements given by the set of all translations on $\mathbb{R}^n$. To discuss numerical integration of ODEs on manifolds, we need more general assumptions. The domains are differentiable manifolds, and the basic movements are Lie algebra actions on the manifold:

Assumption 1 (Generic presentation of ODEs on manifolds). *There exists 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 ODE for $y(t) \in \mathcal{M}$ is*

$$y' = \big(\lambda_* f(t, y)\big)(y), \qquad y(0) = p, \tag{3}$$

where λ_* is defined in (1) [18].

If the algebra action is transitive, then any ODE on $\mathcal{M}$ may 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 $\partial/\partial x^i$ constitutes a commutative and transitive algebra acting on $\mathcal{M}$, via the coordinate flows. Examples of manifolds and group actions are presented in Section 5.

4. RKMK methods of general order

The following algorithm integrates Eq. (3) from $t = 0$ to $t = h$:

Algorithm 1 (RKMK). A truncated approximation for $\mathrm{dexp}_u^{-1}(v)$ is

$$\mathrm{dexpinv}(u, v, q) = \sum_{k=0}^{q-1} \frac{B_k}{k!}\mathrm{ad}_u^k(v).$$

Let $a_{i,j}$ and b_j be the coefficients of an s-stage, qth order classical Runge–Kutta method and let $c_i = \sum_{j=1}^{s} a_{i,j}$. Let $\lambda(u, p) = \Lambda(\exp(u), p)$ for some Lie group action Λ.

$y_0 = p$
for $i = 1, 2, \ldots, s$

$$u_i = h \sum_{j=1}^{s} a_{i,j} \widetilde{k}_j$$
$$k_i = f(hc_i, \lambda(u_i, y_0))$$
$$\widetilde{k}_i = \mathrm{dexpinv}(u_i, k_i, q)$$

end

$$v = h \sum_{j=1}^{s} b_j \widetilde{k}_j$$
$$y_1 = \lambda(v, y_0)$$

Here $p, y_0, y_1 \in \mathcal{M}$ and $u_i, k_i, \widetilde{k}_i, v \in \mathfrak{g}$.

Notes.

(1) If the underlying classical RK method is explicit, i.e., if $a_{i,j} = 0$ for $i \leqslant j$, then the evaluation can be explicit.

(2) The basic operations in this algorithm are in the Lie algebra (sums, scalar products and Lie-brackets), and computing the algebra action λ. Unlike in the formulations in [16,18], we avoid operations in the Lie group G. This simplifies the specification of the operations involved in Runge–Kutta computations, and in many cases also yields a significant computational saving. There are many examples where computing the algebra action λ directly is much less expensive than computing the exponential mapping followed by a computing the group action Λ.

(3) An important special case of (3) arises when $f(t, p) = f(t)$ depends only on time. This is an *equation of Lie type*, or a linear type equation. For these equations, it is not necessary to compute the action λ for the s internal stages. It may be profitable to take several Runge–Kutta steps in $\mathfrak{g}$ before advancing in $\mathcal{M}$, thereby further reducing the number of evaluations of λ. Special methods for solving Lie type equations are presented in [10]. A detailed comparison of these methods is still needed.

(4) For general ODEs, the number of λ evaluations equals the number of function evaluations. By symbolically computing in a *free Lie algebra*, the number of commutators needed can be reduced significantly compared to the basic algorithm; see [17].

Theorem 6. *Algorithm* 1 *stays on the manifold*

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

Proof. All motions on $\mathcal{M}$ are given by λ, defined to evolve on $\mathcal{M}$. □

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 $h = 0$ match the first $q + 1$ terms of the Lie series of the analytical solution of (3) around $t = 0$, see [15,16,21].

Theorem 7. *Algorithm* 1 *has order at least q for any Lie group action Λ on any manifold $\mathcal{M}$.*

Note. 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.

A short proof of Theorem 7 is based on the concept of ϕ-relatedness of vector fields [1]. When ϕ is a smooth invertible mapping between manifolds, this is equivalent to 'pull back' of vector fields. The basic idea is to transform the differential equation on $\mathcal{M}$ to an equivalent equation on $\mathfrak{g}$ using pull back 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 invertible, we discuss relatedness rather than pull backs. 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 *ϕ-related* if $\phi' \circ G = F \circ \phi$, written $G \sim_\phi F$. Consider the ODEs for $y(t) \in \mathcal{M}$ and $u(t) \in \mathcal{N}$:

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

It is straightforward to check that if $G \sim_\phi F$ then $y(t) = \phi(u(t))$. Now, let $\lambda_p(u) = \lambda(u, p) = \Lambda(\exp(u), p)$. We use λ_p to pull back Eq. (3) from $\mathcal{M}$ to $\mathfrak{g}$.

Lemma 8. *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 3 and 5,

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

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

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

giving

$$F \circ \lambda_p(u) = \lambda_*(f \circ \lambda_p(u))(\lambda_p(u)) = \Lambda_p' \circ R_{\exp(u)}'(f \circ \lambda_p(u)) = \lambda_p' \circ \tilde{f}(u). \qquad \square$$

Hence, we obtain the following important result.

Corollary 9. *For sufficiently small t, the solution of (3) is*

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

where $u(t) \in \mathfrak{g}$ satisfies the ODE

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

Proof of Theorem 7. Observe that Algorithm 1 is equivalent to a single classical Runge–Kutta step in $\mathfrak{g}$ on the ODE (4) producing a numerical approximation $u_1 \approx u(h)$ followed by 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, the order on $\mathcal{M}$ is at least as high as the order of the corresponding equation on $\mathfrak{g}$. Finally, the approximation error in dexp_u^{-1} introduces an $\mathrm{O}(q)$ modification of $\tilde{f}$ that does not reduce the order. □

5. Examples

We give various examples of ODEs written in the form in Assumption 1.

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

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

$$y' = f(t, y)y, \qquad y(0) = p.$$

Example 3 (Isospectral flows [5]). Let $\mathcal{M} \subset \mathbb{R}^{n\times n}$, $G = \mathrm{SO}(n)$ be the orthogonal group, $\mathfrak{g} = \mathfrak{so}(n)$ its Lie algebra and $\lambda(v, p) = \exp(v)p\exp(-v)$, then $\lambda_* v(p) = vp - pv$ and (3) reduces to the isospectral equation:

$$y' = f(t, y)y - yf(t, y), \qquad y(0) = p.$$

Since the action is isospectral, Algorithm 1 preserves isospectrality exactly.

Example 4 (Exponential integrators for stiff systems). Consider a differential equation on $\mathbb{R}^d$ written in standard form (2). When this system is stiff, it is well known that one must employ implicit classical integration methods. One type of stiffness arises 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. Let $G = \mathrm{GL}(d) \rtimes \mathbb{R}^d$ be the *semidirect product* [24] of the general linear group and $\mathbb{R}^d$. G is the group of all affine linear maps acting on $\mathbb{R}^d$,

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

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

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

and exponential mapping

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

where

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

Let the algebra action be

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

yielding

$$\lambda_*(A,b)(y) = Ay + b.$$

Note, the map $\lambda_*(\cdot)(p):\mathfrak{g} \to T\mathcal{M}|_p$ has a nontrivial kernel, hence there is freedom in choosing the function $f:\mathbb{R}\times\mathbb{R}^d \to \mathfrak{g}$ to change (2) to the form (3). One possible choice is

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

which just recovers the classical methods for (2). Another is

$$f(t,y) = \big(J, F(t,y) - Jy\big),$$

where J is the Jacobian of F at the point y. Also, we may let J be the Jacobian evaluated at some nearby point, and change J whenever necessary. This saves computations of the exponential. These methods solve linear systems exactly, and hence Algorithm 1 becomes A-stable, even in explicit form. In the case where Algorithm 1 is based on forward Euler, the resulting method is

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

which is the explicit second-order A-stable method in [19]. Related methods have been developed in [9]. So, 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, and the stiffness properties of the algorithm depend on the choice of action and of f.

Example 5 (Riccati equations [3,23]). The Riccati equation

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

is a model example of an ODE of Lie type. This version of the equation may be written as a special form of (3) where $G = \mathrm{SL}(2,\mathbb{R})$ is the set of real 2×2 matrices with determinant 1 and $\mathfrak{g}$ its Lie algebra, the set of all 2×2 matrices with trace 0. Furthermore, let $\mathcal{M} = \mathbb{R}$, $\Lambda: G\times\mathcal{M}\to\mathcal{M}$ be the Möbius transformation

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

and $\lambda(v,y) = \Lambda(\exp(v),y)$. To compute λ_*, note that $\lambda_*(\cdot)(y):\mathfrak{g}\to T\mathcal{M}|_y$ is always a linear map, and we may compute λ_* in a basis for $\mathfrak{g}$. This yields

$$\lambda_*\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}(y) = 2y, \qquad \lambda_*\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}(y) = 1, \qquad \lambda_*\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}(y) = -y^2.$$

Hence the equation is cast in standard form $f:\mathbb{R}\to\mathfrak{g}$ with

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

Numerical experiments show that integration schemes based on this natural action avoid the problems with singularities that occur for classical coordinate based integration schemes.

Example 6 (Rigid frames [6,21]). 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 $T\mathcal{M}|_p$. The frames represent vector fields that are 'easily' integrated. An ODE 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 $\mathfrak{g} \subset \mathfrak{X}(\mathcal{M})$ be the Lie sub-algebra of $\mathfrak{X}(\mathcal{M})$ generated by E_i, where the bracket on $\mathfrak{g}$ is $[\cdot,\cdot]_{\mathrm{LJ}}^{-}$. Thus $\mathfrak{g}$ is the space of all vector fields spanned by all the original E_i and all their iterated commutators. Let $\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$ be the flow operator; i.e., for all $F \in \mathfrak{g}$, let $y(t) = \lambda(tF, p)$ be the solution of

$$y'(t) = F\big(y(t)\big), \qquad y(0) = p.$$

Hence

$$\lambda_* F(p) = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \lambda(tF, p) = F\big(y(t)\big)\big|_{t=0} = F(p),$$

and (5) is cast in general form if we let $f : \mathcal{M} \to \mathfrak{g}$ be

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

Algorithm 1 becomes a method for integrating ODEs written in terms of rigid frames, and the basic operation involved in computing the action $\lambda(v, p)$ becomes the task of following a fixed flow in the algebra spanned by $\{E_i\}$. The Crouch–Grossman family of algorithms [6,21] provides an alternative approach to integrating such ODEs.

Example 7 (Numerical example). Finally, we present a simple numerical example to show that the algorithms have the correct order. Let $\mathfrak{g}$, λ, G and $\mathcal{M}$ be 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

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

and the initial condition is the random orthogonal 4×4 matrix:

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

The numerical experiments were performed by integrating from $t = 0$ to $t = 10$, and successively halving the stepsizes: $h = 5, 2.5, \ldots$. Fig. 1 shows the global error at $t = 10$ (measured in the 2-norm) versus the stepsize h. R2 is based on Runge's second-order method, RK4 on the classical fourth-order Runge–Kutta method, B6 on Butcher's sixth-order method and DP8 on the eight-order method of Dormand and Prince. The classical versions of these algorithms are given in [7]. We modified them according to Algorithm 1, hence preserving orthogonality exactly.

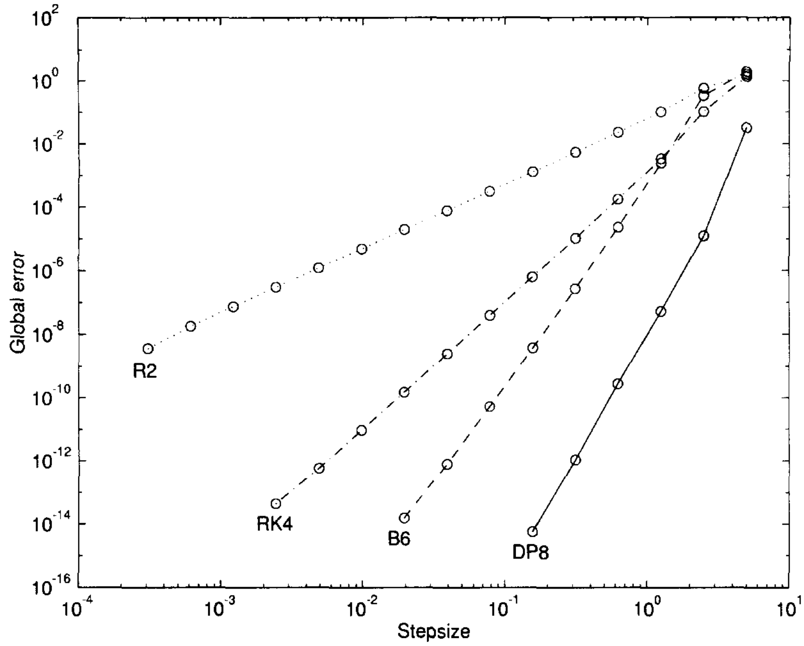

Fig. 1. Global error versus stepsize (Example 7).

6. Final remarks

This paper is a sequel to the papers [15,16,18]. The main open questions of these papers have 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 this new light, they appear as versions of Runge–Kutta methods employing *canonical coordinates of the first kind* [24] in a Lie group. For some basis $\{v_i\}$ of $\mathfrak{g}$, the coordinates are obtained by inverting the map $(x_1, x_2, \ldots, x_d) \mapsto \exp(x_1 v_1 + x_2 v_2 + \cdots + x_d v_d) \cdot p$ in the neighborhood of a point p. Although the Lie–Butcher theory [15,16] is no longer necessary to provide the basic order proof, it remains very important for understanding the algorithms in detail.

The theory of this paper allows us to pose new questions, which can be analyzed with the machinery presented here:

- Crouch–Grossman type methods [6,21] are related to *canonical coordinates of the second kind* [24]. Can a correction technique be employed to develop such methods of higher order?
- In the present formulation of the methods, we change coordinates at every step. We could consider taking several steps in $\mathfrak{g}$ before a step is performed in $\mathcal{M}$, and we might also consider multistep methods in $\mathfrak{g}$. Especially for ODEs of Lie type, this might lead to savings. The main difficulty with this approach could be the conditioning of the coordinate transform which worsens as we move away from the origin in $\mathfrak{g}$, so that eventually the exponential mapping is no longer a diffeomorphism.
- If the manifold $\mathcal{M}$ possesses additional structures such as, for example, a symplectic two-form $\omega : T\mathcal{M} \times T\mathcal{M} \to \mathbb{R}$, is it possible to devise symplectic methods on $\mathfrak{g}$ which preserve this form? One way to treat this question might be to pull back ω from $\mathcal{M}$ to $\mathfrak{g}$.

- Numerical integration of ODEs has now been enriched by a new choice. In addition to selecting the basic integration method, we have freedom to choose different $\mathfrak{g}$ and different algebra actions λ. One may consider the task of finding a good action to be similar to the problem of designing a good preconditioner for iterative solution of linear systems. In both cases, the goal is to find a system that is easier to solve than the original, but which captures some of the original's essential features. We have shown examples of actions but there are many other possible choices. In mechanical problems such as the heavy spinning top, a very nice formulation involves writing the equations in our generic form, using the co-adjoint action of $T\mathrm{SO}(3)$ on $T^*\mathfrak{so}(3)$. The spinning top is a simple model example of a wide class of mechanical problems discussed in [13]. Another interesting example is the use of Toeplitz systems as actions for time integration of parabolic PDEs. Toeplitz systems represent constant coefficient PDEs that can be integrated efficiently using FFTs.

Acknowledgements

Thanks to Bruno Welfert and Brynjulf Owren for providing the formula for dexp_u^{-1} in Theorem 3 and to Owren for also pointing out a significant error in an early version of the manuscript. I would also like to thank Per Christian Moan, Arne Marthinsen, Syvert P. Nørsett, Jeremy Schiff and Antonella Zanna for very 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, University of Cambridge, UK, January–December 1997.

References

[1] R. Abraham, J.E. Marsden and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*, 2nd ed. (Springer, New York, 1988).

[2] H.F. Baker, Alternants and continuous groups, *Proc. London Math. Soc., Second Series* 3 (1905) 24–47.

[3] R.L. Bryant, An introduction to Lie groups and symplectic geometry, in: D.S. Freed and K.K. Uhlenbeck, eds., *Geometry and Quantum Field Theory*, 2nd ed., IAS/Park City Mathematics Series, Vol. 1 (Amer. Math. Soc., 1995).

[4] M.P. Calvo, A. Iserles and A. Zanna, Runge–Kutta methods on manifolds, in: D.F. Griffiths and G.A. Watson, eds., *Numerical Analysis, A.R. Mitchell 75th Birthday Volume* (World Scientific, Singapore, 1996) 57–70.

[5] M.P. Calvo, A. Iserles and A. Zanna, Numerical solution of isospectral flows, *Math. Comp.* 66 (1997) 1461–1486.

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

[7] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations, I. Nonstiff Problems*, 2nd ed. (Springer, New York, 1993).

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

[9] M. Hochbruck, C. Lubich and H. Selhofer, Exponential integrators for large systems of differential equations, *SIAM J. Sci. Comput.*, to appear.

[10] A. Iserles and S.P. Nørsett, On the solution of linear differential equations in Lie groups, Technical Report 1997/NA3, DAMTP, University of Cambridge, England (1997).

[11] S. Lie and F. Engel, *Theorie der Transformationsgruppen* (Teubner, Leipzig, 1888, 1890, 1893).

[12] W. Magnus, On the exponential solution of differential equations for a linear operator, *Comm. Pure Appl. Math.* 7 (1954) 649–673.
[13] J.E. Marsden and T.S. Ratiu, *Introduction to Mechanics and Symmetry* (Springer, New York, 1994).
[14] 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) 75–88.
[15] H. Munthe-Kaas, Lie–Butcher theory for Runge–Kutta methods, *BIT* 35 (1995) 572–587.
[16] H. Munthe-Kaas, Runge–Kutta methods on Lie groups, *BIT* 38 (1998) 92–111.
[17] H. Munthe-Kaas and B. Owren, Computations in a free Lie algebra, Technical Report 148/1998, Department of Informatics, University of Bergen, Norway (1998).
[18] H. Munthe-Kaas and A. Zanna, Numerical integration of differential equations on homogeneous manifolds, in: F. Cucker, ed., *Proc. of Conf. on Foundations of Computational Mathematics* (Springer, New York, 1997) 305–315.
[19] S.P. Nørsett, The numerical solution of stiff systems, Master's Thesis, University of Oslo, Norway (1969).
[20] P.J. Olver, *Equivalence, Invariants, and Symmetry* (Cambridge University Press, Cambridge, 1995).
[21] 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, Trondheim, Norway (1997).
[22] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, London, 1994).
[23] J. 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).
[24] V.S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representation* (Springer, New York, 1984).
[25] A. Zanna, The method of iterated commutators for ordinary differential equations on Lie groups, Technical Report 1996/NA12, DAMTP, University of Cambridge, England (1996).
[26] A. Zanna and H. Munthe-Kaas, Iterated commutators, Lie's reduction method and ordinary differential equations on matrix Lie groups, in: F. Cucker, ed., *Proc. of Conf. on Foundations of Computational Mathematics* (Springer, New York, 1997) 434–441.