Commutator-free Lie group methods[*]

Elena Celledoni[†] Arne Marthinsen[‡] Brynjulf Owren[§]

May 22, 2002

Abstract

RKMK methods and Crouch-Grossman methods are two classes of Lie group methods. The former is using flows and commutators of a Lie algebra of vector fields as a part of the method definition. The latter uses only compositions of flows of such vector fields, but the number of flows which needs to be computed is much higher than in the RKMK methods. We present a new type of methods which avoids the use of commutators, but which has a much lower number of flow computations than the Crouch-Grossman methods. We argue that the new methods may be particularly useful when applied to problems on homogeneous manifolds with large isotropy groups, or when used for stiff problems. Numerical experiments verify these claims when applied to a problem on the orthogonal Stiefel manifold, and to an example arising from the semidiscretisation of a linear inhomogeneous heat conduction problem.

AMS Subject Classification: 65L05

Key Words: Lie group methods, geometric integration, numerical analysis, commutators.

1 Introduction

We consider ordinary differential equations on manifolds

$$\dot{y} = F(y), \quad y(0) = y_0, \quad F \in \mathfrak{X}(\mathcal{M}), \tag{1}$$

where $\mathfrak{X}(\mathcal{M})$ is the space of smooth vector fields on the differentiable manifold $\mathcal{M}$. In many situations, it is natural to think of $\mathcal{M}$ as a subset of some linear space V of higher dimension, as for instance when $\mathcal{M}$ consists of the orthogonal $n \times n$ matrices and V is the set of all real $n \times n$ matrices. The vector field F in (1) will then often have a natural extension to all of V or at least to some neighborhood of $\mathcal{M}$. Standard numerical integrators which use vector space operations as primitives can then be applied in this enlarged space, and if it is important that the numerical approximation remains on the manifold $\mathcal{M}$, one can use for instance projection after each step.

If it is also important to avoid using the (extended) vector field F outside the manifold, there are now available various methods as well. One type is based on parametrizing the

[*]This work was in part sponsored by The Norwegian Research Council under contract no. 111038/410, through the SYNODE project. WWW: `http://www.math.ntnu.no/num/synode/`

[†]SINTEF Appl Math, Email: `elenac@math.ntnu.no`, WWW: `http://www.math.ntnu.no/~elenac/`

[‡]Email: `Arne.Marthinsen@math.ntnu.no`, WWW: `http://www.math.ntnu.no/~arnema/`

[§]Email: `Brynjulf.Owren@math.ntnu.no`, WWW: `http://www.math.ntnu.no/~bryn/`

manifold using for instance the tangent space as parameter space, this has been discussed in general terms in [14, 13]. One then needs to find a local diffeomorphism $\phi_p : T_p\mathcal{M} \to U \subset \mathcal{M}$ where U is some open set containing the point p. Setting $p = y_0$ one may represent the solution of (1) locally as $y(t) = \phi_p(\sigma(t))$, where $\sigma(t) \in T_p\mathcal{M}$ and $\sigma(0) = p$. The resulting differential equation for σ is solved by a numerical method and the result is mapped back to $\mathcal{M}$ by ϕ_p.

The Lie group methods represent a similar approach. At the heart of these methods is the way the vector field F is represented. Crouch and Grossman [3] used the notion of *frames*, that is, a set of d smooth vector fields $E_1, \dots, E_d$ which at every point $y \in \mathcal{M}$ span the tangent space, i.e.

$$T_y\mathcal{M} = \mathrm{span}\{E_1(y), \dots, E_d(y)\},\ \forall\, y \in \mathcal{M}.$$

So we must require that $d \geq m = \dim(\mathcal{M})$. Consequently, one can write smooth vector fields on $\mathcal{M}$ in the form

$$F(y) = \sum_{i=1}^{d} f_i(y) E_i(y), \tag{2}$$

where each f_i is a real or complex-valued function on the manifold. A prototypical example frequently seen in the literature is when the manifold is a Lie group of $n \times n$ matrices, and thus the vector field is of the form

$$\dot{y} = A(y) \cdot y, \quad A(y) \in \mathfrak{g}, \tag{3}$$

and $\mathfrak{g}$ is the linear subspace of the $n \times n$ matrices known as the *Lie algebra* of the Lie group. For instance, if the Lie group is the set of orthogonal matrices, the corresponding Lie algebra will be the linear space of skew-symmetric $n \times n$ matrices, see for instance the monograph [4]. It is easily seen that (3) is a special case of (2) by setting $E_i(y) = e_i(y) \cdot y$ for some basis $e_1, \dots, e_d$ for $\mathfrak{g}$, and by letting $A(y) = \sum_i f_i(y) e_i$. The advantage of considering the form (2) rather than (3) is that the former is more general and presents us the possibility in using a general method format that can be adapted to many different situations. An alternative and just as flexible framework for ODEs on manifolds is the Lie group actions, see for instance [10, 8]. Using frames, the notion of *frozen vector field* plays an important role. Given a vector field F and a fixed $p \in \mathcal{M}$ there is a special vector field defined relative to the frame as

$$F_p(y) = \sum_i f_i(p) E_i(y). \tag{4}$$

Vector fields of this form all belong to the linear span of the frame vector fields. Many Lie group methods are designed by using flows of vector fields in this linear span as building blocks. In the particular case (3), the frozen vector fields are of the form

$$F_p(y) = A(p) \cdot y,$$

and the flow is obtained via the matrix exponential `expm`, $\exp(t\, F_p) y_0 = \mathtt{expm}(t\, A(p)) \cdot y_0$. In the case of the Munthe-Kaas methods [7, 8], one also needs to include commutators between frozen vector fields in the method format. In the general formulation (2) this is understood as the Lie-Poisson bracket between vector fields, see e.g. [11, p. 33], but in the example (3) it is nothing else than the matrix commutator

$$[A, B] = A \cdot B - B \cdot A.$$

We now present explicit versions of the methods of Crouch and Grossman [3] and Munthe-Kaas [8] in terms of frames.

Algorithm 1.1 (RKMK)

for $r = 1 : s$ **do**

$u_r = h \sum_k a_r^k F_k$

$Y_r = \exp(u_r) p$

$\bar{F}_r = F_{Y_r} = \sum_i f_i(Y_r) E_i$

$F_r = \pi(\mathrm{ad}_{u_r})(\bar{F}_r)$

end

$v = h \sum_r b^r F_r$

$y_1 = \exp(v) p$

Algorithm 1.2 (C-G)

for $r = 1 : s$ **do**

$Y_r = \exp(a_r^{r-1} F_{r-1}) \cdots \exp(a_r^1 F_1) p$

$F_r = h F_{Y_r} = h \sum_i f_i(Y_r) E_i$

end

$y_1 = \exp(b^s F_s) \cdots \exp(b^1 F_1) p$

In both of the algorithms, the coefficients a_i^k, b^k are similar to those of a standard explicit Runge-Kutta method. Note also that one could easily generalize the schemes above to implicit ones as for standard Runge-Kutta methods, but then the stage variables $(u_r,\ Y_r,\ F_r)$, $r = 1, \ldots, s$ would have to be solved for simultaneously.

In the RKMK algorithm, π is a polynomial $\pi(z) = \sum_k \pi_k z^k$ with $\pi(0) = 1$, and we let

$$\pi(\mathrm{ad}_u)(v) = v + \pi_1[u, v] + \pi_2[u, [u, v]] + \cdots .$$

In fact, it is well-known [8] that by choosing π to be a sufficiently high order approximation to the function $f(z) = z/(e^z - 1)$, one may obtain order of accuracy q whenever the coefficients a_i^k, b^k are those of a classical qth order Runge-Kutta method. In the methods of Crouch and Grossman it is however necessary to impose extra order conditions on the coefficients when $q \geq 3$, see [12].

In the particular case that the manifold is equal to a linear space, say V, one has $T_y V \cong V$ for each $y \in V$. We can choose the frame vector fields to be simply $E_i(y) = e_i \in V$, where $e_1, \ldots, e_m$ is some basis for V. A frozen vector field is just a constant vector $F_p \in V$, and its flow is $\exp(tF_p)y = y + tF_p$. Since also all commutators vanish in this situation one finds that both the above algorithms become standard Runge-Kutta methods.

It should be noted that there may be situations where it suffices that the vector fields form a frame only locally, the algorithms proposed above make use of the frame only in a neighborhood of the initial point p. One can have in mind the concrete example of S^2, the two-dimensional sphere. It is well known that any smooth vector field on S^2 must vanish at least at one point, thus to obtain a frame, at least three vector fields are needed. In fact, a frame of three vector fields, isomorphic to the Lie algebra of 3×3 skew-symmetric matrices can be found. However, locally we may reduce the number of vector fields from three to two, the integration methods may work always with two out of the three vector fields.

The main difference between algorithms 1.1 and 1.2 is that the former is using commutators as part of the method format as opposed to the latter which instead is using many more exponentials. The number of commutators needed by the RKMK methods depends on the degree of the polynomial π, which in turn depends on the convergence order of the method. As a matter of fact, the issue is even more complicated. A careful analysis of the way various linear combinations of the stages F_r depends on the step size h, show that the number of commutators which are necessary to calculate can be reduced even further, one uses a different polynomial for each stage, see [9].

The complexity in terms of commutators and exponentials for the two methods is shown i Table 1 for $q = 3, 4, 5$. The given number of exponentials for C-G $q = 5$ is just an estimate

order	$q = 3$		$q = 4$		$q = 5$	
	exp's	Com's	exp's	Com's	exp's	Com's
RKMK	3	1	4	2	6	12
C-G	6	0	15	0	45	0

Table 1: Complexity per step of RKMK and C-G methods

obtained by comparing the number of order conditions with the number of free parameters associated with a given number of stages. No such method has actually been constructed as far as we know.

In naive implementations of the two algorithms above where the Lie group in question is realized as a group of $n \times n$ matrices, the cost of each exponential can typically be of size $25\,n^3$ flops for large n. In comparison, the commutator may cost $4\,n^3$ flops when no structure of the underlying Lie algebra is exploited. So from such a crude assessment, it seems obvious that the RKMK methods are significantly less expensive per step than the C-G methods. Note for instance that for $q = 5$ we get $198\,n^3$ and $1125\,n^3$ for the two methods respectively using these naive cost estimates.

In general, the arguments just given may in many cases be misleading. One should note that a property of the methods of Crouch and Grossman is that they only use exponentiation of vector fields belonging to a linear space of frames which is not necessarily a Lie algebra under the Lie-Poisson bracket. An interesting example is that of homogeneous spaces with large isotropy groups. In this case, one can express vector fields locally using only a small subset of the frame vector fields. If this subset is chosen with some care, see e.g. [1], it may also lead to very low complexity algorithms for calculating the exponentials, as opposed to using flow computations of general elements of the Lie algebra acting on $\mathcal{M}$. Another example is related to the idea of Munthe-Kaas [8] to solve stiff problems with Lie group methods. Experiences [5, 15] show that the use of commutator may cause serious limitations on the time step.

Still, we may conclude that it is difficult to find situations where the commutator-free methods of Crouch and Grossman can compete with the cheapest Lie group methods which are using commutators, like the RKMK methods. At least this seems to be true for high order methods. In this paper we will introduce a new class of Lie group methods which are also commutator-free, but which use substantially less exponentials than the methods of Crouch and Grossman. The general order theory for these methods is easily adapted from [12], and the somewhat complicated conditions can be simplified significantly such that methods of order 3, 4 are easily constructed. In this note, we will not discuss the details of the order theory, but we will present two one-parameter families of third order methods which use only 3 exponentials per step and no commutators. Thus, counting just complexity per step, it is cheaper than any known third order Lie group method based on exponentials. We will also give an example of a fourth order method which uses effectively 5 exponentials per step and no commutators. Numerical experiments will support our claim that these new methods are indeed a worthy alternative to the RKMK methods, especially when the frame does not form a Lie algebra and for stiff problems.

2 A new class of Lie group methods

We propose the following new format of Lie group methods in terms of a frame

Algorithm 2.1

for $r = 1 : s$ do

$\quad Y_r = \exp(\sum_k \alpha_{rJ}^k F_k) \cdots \exp(\sum_k \alpha_{r1}^k F_k)(p)$

$\quad F_r = hF_{Y_r} = h \sum_i f_i(Y_r) E_i$

end

$y_1 = \exp(\sum_k \beta_J^k F_k) \cdots \exp(\sum_k \beta_1^k F_k) p$

The way we present the method as an algorithm suggests that it should be explicit, this means that $\alpha_{rj}^k = 0$ whenever $k \geq r$, if this is not so, one needs to solve for all Y_r and F_r simultaneously. Here we will consider only explicit schemes.

The parameter J counts the number of exponentials (flows) for each stage. It is convenient to allow J to have a different value for each step, in fact we will always try to keep the total number of exponentials in the method as low as possible.

Now we consider the important special case discussed in the previous section where the manifold is a linear space and the frame vector fields are constant fields such that $\exp(tF_p)y = y + tF_p$ for frozen vector fields F_p. Then

$$Y_r = p + h \sum_{k=1}^{s} a_r^k F_k, \ r = 1, \ldots, s, \qquad y_1 = p + h \sum_{k=1}^{s} b^k F_k,$$

where

$$a_r^k = \sum_{j=1}^{J} \alpha_{rj}^k, \quad b^k = \sum_{j=1}^{J} \beta_j^k. \tag{5}$$

Since the method should satisfy the order conditions also for this choice of frame, we may immediately conclude that the coefficients of a qth order method must be such that (a_r^j) and (b^r) in (5) satisfy that usual order conditions for standard RK methods.

One may also observe that the C-G methods appear as a special case of the new methods, where $J = s$ and $\alpha_{rj}^k = \delta_{jk} a_r^k$. The methods of Munthe-Kaas are not included in the above format, although by allowing for a stage correcting function, this could easily be achieved, but is not needed in the present work.

Reusing flow calculations. Suppose for instance that for some $r < \bar{r}$,

$$\alpha^k_{r,1} = \alpha^k_{\bar{r},1}, \quad \text{or} \quad \alpha^k_{r,1} = \beta^k_1, \quad \forall k.$$

Then Y_r and $Y_{\bar{r}}$ (y_1 resp) could be calculated as

$$\begin{aligned}
Y_{r,1} &= \exp\big(\sum_k \alpha^k_{r,1} F_k\big) p, \\
Y_m &= \exp\big(\sum_k \alpha^k_{m,J} F_k\big) \cdots \exp\big(\sum_k \alpha^k_{m,2} F_k\big) Y_{r,1}, \ m = r, \bar{r}, \\
\Big(y_1 &= \exp\big(\sum_k \beta^k_J F_k\big) \cdots \exp\big(\sum_k \beta^k_2 F_k\big) Y_{r,1} \Big),
\end{aligned}$$

and 1 flow computation (exponential) would be saved. Of course this may be further generalized. In the case that the manifold is a matrix Lie group, one may actually store each of the computed exponentials and reuse them several times, keep in mind that the cost of calculating a $n \times n$ matrix exponential is 20–30 n^3 flops, whereas a matrix-matrix multiplication is $2n^3$ flops.

For other manifolds, it may not be feasible to store flows in the form of mappings as the matrix exponential, instead one calculates only the flow applied to a point on the manifold, like $\exp(F)\,p$, where $p \in \mathcal{M}$.

Third order methods It is known that the maximal order one can obtain if $J = 1$ for all the stages is two. Thus, using 3 stages, we need at least one extra exponential. It also turns out to be sufficient to add an exponential in the last stage, in the expression for y_1. The only way one can reuse flow calculations, is to reuse either Y_2 or Y_3 in the calculation of y_1, thus either

$$\beta^1_1 = c_2, \ \beta^2_1 = \beta^3_1 = 0,$$

or

$$\beta^1_1 = c_3 - a_{32}, \ \beta^2_1 = a_{32}, \ \beta^3_1 = 0.$$

In the former case, we obtain the one-parameter family of methods with Butcher tableau

$$\begin{array}{c|ccc}
0 & & & \\
\frac{1}{3} & \frac{1}{3} & & \\
\alpha & \alpha(2-3\alpha) & \alpha(3\alpha-1) & \\
\hline
 & \frac{1}{3} & 0 & 0 \\
 & \frac{3-5\alpha}{6\alpha} & \frac{3(3\alpha-2)}{2(3\alpha-1)} & \frac{1}{2\alpha(3\alpha-1)}
\end{array}$$

where $\alpha \notin \{0, \frac{1}{3}\}$ Here the first row under the horizontal line contains β^k_1 and the second contains β^k_2. Choosing $\alpha = \frac{2}{3}$ leads to the well-known underlying third order classical Runge-

Kutta method of Heun, subsequently refered to as CF3, and we get

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{3} & \frac{1}{3} & & \\
\frac{2}{3} & 0 & \frac{2}{3} & \\
\hline
 & \frac{1}{3} & 0 & 0 \\
 & -\frac{1}{12} & 0 & \frac{3}{4}
\end{array}
\tag{6}
$$

In the second case, we obtain again a one-parameter family of methods where the exponential in Y_3 is reused

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{2}{3}\frac{3\alpha^2-3\alpha+1}{1-\alpha} & \frac{2}{3}\frac{3\alpha^2-3\alpha+1}{1-\alpha} & & \\
\alpha & \frac{1}{4}\frac{12\alpha^3-9\alpha^2+1}{3\alpha^2-3\alpha+1} & \frac{1}{4}\frac{(3\alpha-1)(1-\alpha)}{3\alpha^2-3\alpha+1} & \\
\hline
 & \frac{1}{4}\frac{12\alpha^3-9\alpha^2+1}{3\alpha^2-3\alpha+1} & \frac{1}{4}\frac{(3\alpha-1)(1-\alpha)}{3\alpha^2-3\alpha+1} & 0 \\
 & -\frac{1}{4}\frac{12\alpha^3-21\alpha^2+15\alpha-4}{3\alpha^2-3\alpha+1} & -\frac{1}{4}\frac{(9\,\alpha^2-9\,\alpha+4)(1-\alpha)}{(1+3\,\alpha^2-3\,\alpha)(3\,\alpha-1)} & \frac{1}{(3\alpha-1)}
\end{array}
$$

where it is assumed that $\alpha \notin \{\frac{1}{3}, 1\}$.

An example method is obtained as

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{2}{3} & \frac{2}{3} & & \\
\frac{2}{3} & \frac{5}{12} & \frac{1}{4} & \\
\hline
 & \frac{5}{12} & \frac{1}{4} & 0 \\
 & -\frac{1}{6} & -\frac{1}{2} & 1
\end{array}
$$

Fourth order methods A similar approach can be used to derive methods of order 4. Note that any classical explicit Runge-Kutta method has at least 4 stages. To obtain a 4th order method in the proposed form, it turns out that we need to add two exponentials, and that it is necessary to add one of them to the final stage in the expression for y_1. In addition, one needs to add one other exponential and there is some freedom of choice in which stage to add it. Letting it be included in the expression for Y_4 leaves us freedom to have it coincide with the exponential in either Y_2 or Y_3. The discussion of order conditions is fairly complicated and will therefore appear elsewhere. Here we just present a method which generalizes the classical 4th order Runge-Kutta method and which effectively uses 5 exponentials per step.

$$
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & \frac{1}{2} & 0 & 0 & \\
 & -\frac{1}{2} & 0 & 1 & \\
\hline
 & \frac{1}{4} & \frac{1}{6} & \frac{1}{6} & -\frac{1}{12} \\
 & -\frac{1}{12} & \frac{1}{6} & \frac{1}{6} & \frac{1}{4}
\end{array}
\tag{7}
$$

We subsequently refer to this method as CF4.

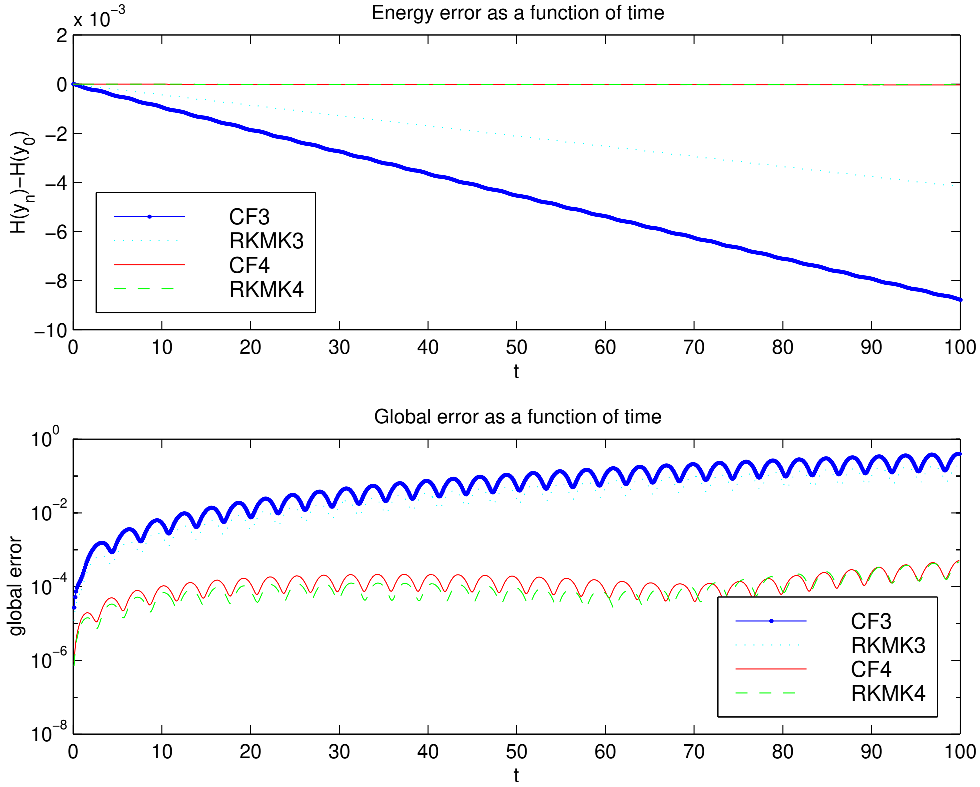


Figure 1: Energy error and global error as a function of time with step size $h = 0.1$.

3 Numerical experiments

In this section we illustrate the numerical performance of the new methods when compared to other Lie group methods. We first consider the Euler equations for the rigid body, then ODEs on the Stiefel manifold and finally a stiff problem arising from the semidiscretization of the heat equation.

We compare the commutator-free example methods presented in this paper (CF3 of (6) and CF4 of (7)) with the Runge-Kutta Munthe-Kaas methods of order 3 and 4 (RKMK3 and RKMK4). The RKMK methods are implemented using the optimized versions of the algorithms proposed in [9]. The optimization is obtained by using the free Lie algebra structure to reduce the number of commutators.

In the second experiment we also compare the new methods with a method recently proposed in [2] and based on the use of retractions and the Cayley transformation (RKR-C4). In the last experiment comparison with the classical Runge-Kutta method of order 4 is also included.

3.1 Rigid body equations

In this section we present results from simulations on the rigid body equations. We consider a rigid body with center of mass fixed at the origin. Let $y \in \mathbb{R}^3$ denote the vector of angular momenta of the rigid body and let $\mathcal{I} = \mathrm{diag}(I_1, I_2, I_3)$ be the inertia tensor. The equations

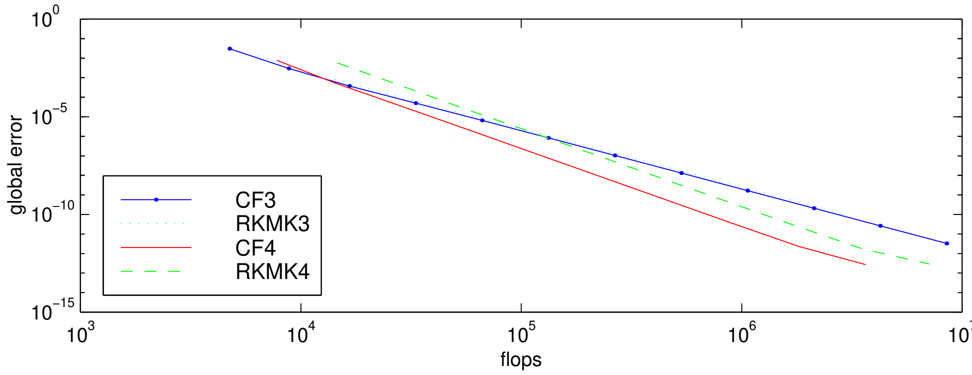


Figure 2: Efficiency measured as global error versus flops.

of motion are given by (see e.g. [6])

$$\frac{\mathrm{d}y}{\mathrm{d}t} = y \times \mathcal{I}^{-1} y.$$

This problem can be written in the form (3) by use of the vector space isomorphism $\widehat{\ }: \mathbb{R}^3 \to \mathfrak{so}(3)$, called the hat map, defined by

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \mapsto \widehat{y} = \begin{pmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & y_1 \\ -y_2 & y_1 & 0 \end{pmatrix}.$$

The function $A(y) \in \mathfrak{so}(3)$ then takes the form

$$A(y) = -\widehat{\mathcal{I}^{-1} y}.$$

In the simulations we have used $y(0) = (\frac{8}{9}, \frac{4}{9}, \frac{1}{9})$ as initial condition and $\mathcal{I} = \mathrm{diag}(\frac{7}{8}, \frac{5}{8}, \frac{1}{4})$ as inertia tensor. We integrate over the time window $[0, 100]$ with a fixed step size of 0.1.

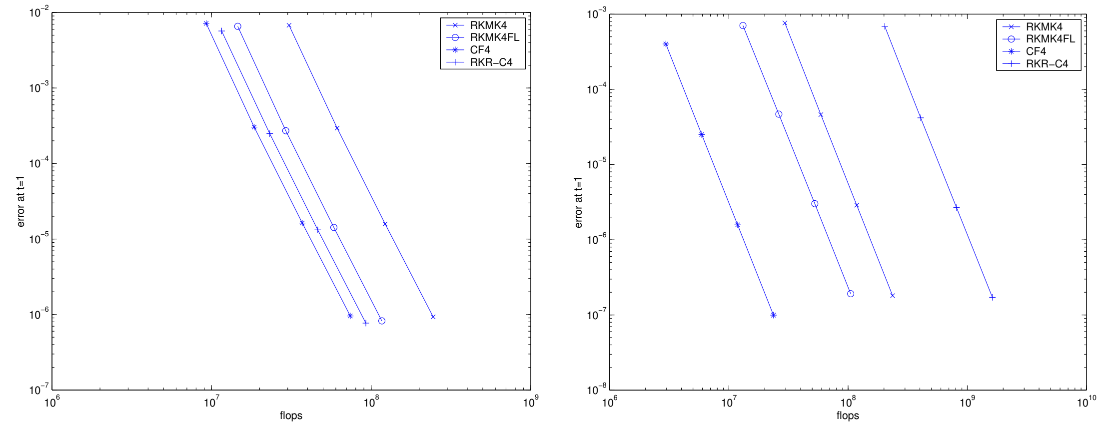
The upper part of Figure 1 shows the energy error as a function of time for the four methods CF3, CF4, RKMK3 and RKMK4. The behavior of CF4 and RKMK4 are essentially the same, but RKMK3 behaves better than CF3. For CF3 and RKMK3 we observe as expected, a more rapid drift in the energy. The lower part of Figure 1 shows global error as a function of time. The best performance is obtained by RKMK4, but the difference between this method and CF4 is negligible. As expected, CF3 and RKMK3, using the same step size as CF4 and RKMK4, generate numerical approximations that contain larger errors than the two former methods.

Figure 2 shows an efficiency comparison of the four methods. The best performance is obtained by CF4, while CF3 and RKMK3 behave roughly the same on the rigid body problem.

3.2 ODEs on the Stiefel manifold

In this section we consider numerical integration on the Stiefel manifold

$$\mathcal{M} := \{A \in \mathbb{R}^{n\times k} | A^T A = I_p\},$$



(a) First experiment

(b) Second experiment

Figure 3: Global error at $t = 1$ as a function of the number of flops, for CF4, RKMK4, RKMK4FL, and RKR-Cayley for decreasing values of the step size h.

where $p \leq n$ and we denote by I_p the $p \times p$ identity matrix. ODEs on this manifold are of the type

$$Y' = S(Y) \cdot Y, \qquad Y_0 \in \mathcal{M}, \tag{8}$$

with S $n \times n$ skew-symmetric matrix. It has been shown in [2] that any tangent vector V at $Y \in \mathcal{M}$ can be expressed in the following factorized form by means of $n \times p$ matrices

$$V = \begin{bmatrix} g(V) & -Y \end{bmatrix} \begin{bmatrix} Y & g(V) \end{bmatrix}^{\mathrm{T}} \cdot Y, \qquad g(V) = V - Y \frac{Y^{\mathrm{T}} V}{2}. \tag{9}$$

Flows corresponding to vector fields F on the Stiefel manifold frozen at the point Y can be therefore expressed in terms of matrix exponentials as follows

$$\exp\left(t \cdot \begin{bmatrix} g(V) & -Y \end{bmatrix} \begin{bmatrix} Y & g(V) \end{bmatrix}^{\mathrm{T}}\right) \cdot Y, \qquad V = F(Y).$$

The advantage of this formulation arises from the fact that exponentials of such kind of matrices are cheap to compute, see for example [2]. This fact has been exploited in the implementation of all the methods considered in this section.

In the first experiment we compare the new method with the RKMK and the method based on the use of a Cayley retraction map RKR-C proposed in [2]. All the methods have order 4. We report results for RKMK implemented with (RKMK4FL) and without (RKMK4) free Lie algebra optimization. We focus on computational costs presenting a plot that shows the global error as a function of the number of flops. We considered the following ODE on $\mathcal{M}$

$$Y' = (g(Y)Y^T - Yg(Y)^T) \cdot Y,$$

with $g(Y) = DYC$ where $D = \mathrm{diag}([1 : 1/n : 2 - 1/n])$ and $C = \mathrm{diag}([-3 : -1/p : -4 + 1/p])$. The time integration interval is $[0, 1]$.

n	RKMK-4	RKMK4FL	CF4	RKR-C4
100	3325345	1040361	991389	769353
1000	15291745	7289961	4623789	5784113

Table 2: ODEs on the Stiefel manifold $k = 4$, $n = 100, 1000$. Flops count over one time step for the methods. First experiment.

In Table 2 we report the flops count for the methods over one time step, we took $k = 4$ and $n = 100, 1000$.

In Figure 3 (a) we plot the global error as a function of the number of flops for $n = 1000$ and $k = 4$, and various step sizes $h = 1/(2^j)$ with $j = 1, \ldots, 4$.

Compared to the non-optimized RKMK the new method clearly requires a lower number of flops to achieve the same precision in the numerical solution, the improvement is however moderate with respect to the other two methods.

We now consider a problem in which the commutatorfree methods can be exploited with even bigger advantage.

Suppose we are given an ODE on the Stiefel manifold whose vector field can be expressed as

$$S(Y) \cdot Y = \begin{bmatrix} A(Y) & -B(Y)^{\mathrm{T}} \\ B(Y) & O \end{bmatrix} \cdot Y. \tag{10}$$

Here $A(Y)$ is $p \times p$ skew-symmetric $B(Y)$ is $(n-p) \times p$ and O is the $(n-p) \times (n-p)$ zero-matrix. In this example the set of frame vector fields required to represent the given vector field generates a subspace of the whole Lie algebra of vector fields on $\mathcal{M}$. Therefore methods based on linear combinations rather then Lie brackets of vector fields present a much lower computational cost for such kind of problems. In fact the sum of two skew symmetric matrices S_1 and S_2 of the type given in (10) has still the same sparsity pattern while their commutator $[S_1, S_2]$ has in general a larger number of non-zero elements. In this example, instead of using the factorized form of the vector field given in (9) we can use the following

$$\begin{bmatrix} A(Y) & -B(Y)^{\mathrm{T}} \\ B(Y) & O \end{bmatrix} \cdot Y = \begin{bmatrix} \gamma(Y) & -\tilde{I} \end{bmatrix} \cdot \begin{bmatrix} \tilde{I} & \gamma(Y) \end{bmatrix}^{\mathrm{T}} \cdot Y, \tag{11}$$

where $\gamma(Y) = \begin{bmatrix} \mathrm{tril}(A(Y)) & B(Y) \end{bmatrix}^{\mathrm{T}}$, and $\tilde{I}$ has columns equal to the first p canonical vectors in $\mathbb{R}^n$.

In the experiment we consider $\mathrm{tril}(A(Y)) = \mathrm{tril}(\tilde{I}^{\mathrm{T}} G Y)$ where G is skew-symmetric and tridiagonal and $G_{i,i+1} = \left(1 + \sum_{i=1}^{p} Y_{i,i}\right) \cdot i$, for $i = 1, \ldots, n-1$. $B(Y)$ coincides with the first $n - p$ rows of Y.

n	RKMK-4	RKMK4FL	CF4	RKR-C4
100	3272009	967321	165819	1703161
1000	14759609	6565321	1479819	101524077

Table 3: ODEs on the Stiefel manifold $k = 4$, $n = 100, 1000$. Flops count over one time step for the methods. Second experiment.

In this case the advantage of the commutator-free Lie group methods over the others is much more evident, as we can see in Table 3 and Figure 3 (b). The RKR-C4 performs particularly badly in this experiment because unlike the other methods it has not been adapted to the formulation of the problem (11).

We finally remark that even if the problem (10) is of a very special form, it serves to illustrate the more general case in which the ODE vector field can be expressed locally in terms of elements of a (small) subspace of the span of the frame vector fields. It is always possible to obtain such a formulation by taking the dimension of the subspace equal to dimension of the manifold, it may however be challenging to provide such formulations at low computational cost and to switch charts when this becomes necessary.

3.3 Inhomogeneous heat conduction problem

Following an idea first proposed by Munthe-Kaas [8] and later developed for solving inhomogeneous heat conduction problems by Lodden [5] and Suslowicz [15], we consider the linear PDE

$$u_t = (\mu(x)u_x)_x, \quad u(-1,t) = u(1,t) = 0, \; u(x,0) = f(x). \tag{12}$$

The idea is to write (12) in the form

$$u_t = \mu_c u_{xx} + g(x,u), \quad g(x,u) = ((\mu(x) - \mu_c)u_x)_x, \tag{13}$$

where μ_c is some average of $\mu(x)$ over the interval $[-1,1]$. One may discretize this equation in various ways, we here use centered differences on a uniform grid, setting $\Delta x = 2/N$, $x_i = -1 + i\,\Delta x, \; i = 0,\dots,N$, and let $U = (U_1,\dots,U_{N-1})^T$ with $U_i = U_i(t) \approx u(x_i,t)$. We get the ODE system

$$U_t = MU + b(U),$$

where

$$b_i(U) = \frac{(\mu_{i-1/2} - \mu_c)U_{i-1} - (\mu_{i-1/2} + \mu_{i+1/2} - 2\mu_c)U_i + (\mu_{i+1/2} - \mu_c)U_{i+1}}{\Delta x^2},$$

$\mu_{i\pm\frac{1}{2}} = \mu(x_i \pm \Delta x/2)$, and $M = \dfrac{\mu_c}{\Delta x^2}\,\mathrm{tridiag}(1,-2,1)$. We choose the frames to be any set of N vector fields on $\mathbb{R}^{N-1}$ whose linear span is

$$\{F : F(U) = \alpha\, MU + b, \; \alpha \in \mathbb{R}, \, b \in \mathbb{R}^{N-1}\}.$$

Such a vector field, say $F_{\alpha,b}$ has flow $\mathrm{Exp}(tF_{\alpha,b})U = \exp(t\alpha M)U + M^{-1}(\exp(t\alpha M) - I)\,b$ where exp denotes the matrix exponential. The Lie-Poisson bracket is given as $[F_{\alpha,b}, F_{\bar\alpha,\bar b}] = F_{0,c}$ where $c = M(\alpha\bar b - \bar\alpha b)$. Since M is a Toeplitz matrix, its exponential can be calculated efficiently by using the fast Fourier transform, see e.g. [15] for details. It is well-known that the above semidiscretized problem becomes stiff as the number of points N increases. The standard RKMK methods are somewhat sensitive to stiffness as can be seen in the numerical experiments we present here. Various remedies have been proposed in order to damp the fast modes. It seems that the commutator-free methods can better handle stiff problems. We implemented a robust though somewhat expensive step size selection strategy based on Richardson extrapolation, taking two steps with step size $h/2$ to obtain an improved approximation $\tilde U_{n+1}$ to $U(t_{n+1})$ with a resulting local error estimate

$$\mathrm{est}_{n+1} = \frac{1}{2^p - 1}(U_{n+1} - \tilde U_{n+1}),$$

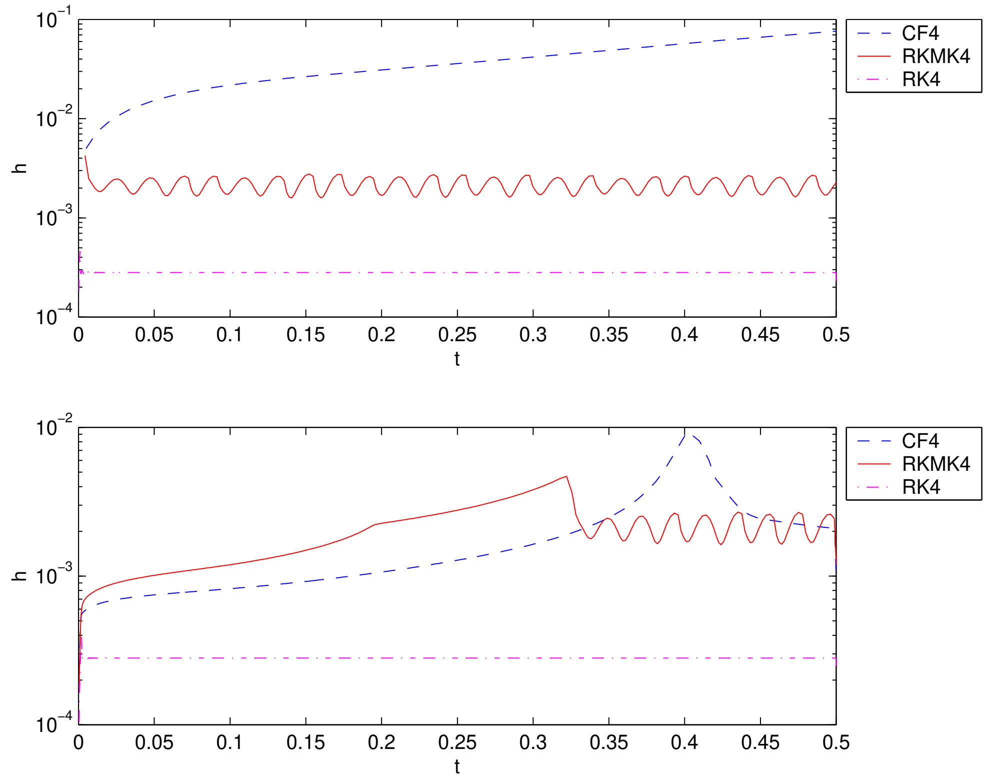


Figure 4: Step size sequences for fourth order methods CF4, RKMK4, RK4 applied to heat conduction problem with $N = 100$ and tolerances 10^{-3} (upper) and 10^{-6} (lower) $h = 0.1$.

where p is the order of the method. A step size selection strategy is then given as $h_{n+1} = \theta(\|\mathrm{est}_{n+1}\|/\mathtt{tol})^{1/(p+1)} \cdot h_n$ where θ is a safety factor, e.g. $\theta = 0.9$, and `tol` is a user provided tolerance.

In all the experiments we used $\mu(x) = 1 - x^2$ and we took $\mu_c = \frac{1}{N}\sum_{i=1}^{N} \mu_{i-\frac{1}{2}}$. As initial function we used $u(x, 0) = 1 - |x|$.

In Figure 4 we see how the step size varies with time using the described step size selection strategy. In the upper part we have used $\mathtt{tol} = 10^{-3}$ and it is seen that the step size choices for RKMK4 and for RK4 (the classical RK4 method) resemble that of a stiff situation whereas the commutator-free method is handling it well. For the stricter tolerance of 10^{-6} in the lower picture, we see that the RKMK4 method is doing better, it does not experience the problem as being stiff anymore, however the classical RK4 method is still encountering stiffness.

In fact, considering Table 4, we see that RK4 is stiff for all tolerances in the range 10^{-2} to 10^{-7} whereas the RKMK4 method is stiff for tolerances down to about 10^{-5}. The commutator-free method CF4 does not appear to have any stability limitations at all for this problem.

We have presented a new class of Lie group methods whose main feature compared to other such methods is that they do not make use of commutators in the method format, and they use as few flow calculations (exponentials) as possible. We have found third order methods which use 3 exponentials per step. This is the same as for the cheapest methods

tol	CF4			RKMK4			RK4		
	GE	STPS	REJ	GE	STPS	REJ	GE	STPS	REJ
10^{-2}	6.1e-2	8	0	2.0e-1	235	19	1.3e-1	1776	3
10^{-3}	2.8e-2	17	0	2.1e-2	237	17	1.4e-2	1777	2
10^{-4}	4.5e-3	57	0	2.9e-3	237	26	1.4e-3	1778	2
10^{-5}	7.5e-4	168	8	3.6e-4	236	23	1.2e-4	1779	2
10^{-6}	8.8e-5	399	8	9.4e-5	296	10	1.3e-5	1781	2
10^{-7}	1.4e-5	784	6	1.4e-5	623	4	2.0e-6	1785	3

Table 4: Statistics for heat conduction problem for various tolerances `tol` and $N = 100$. GE is the obtained global error, STPS is the number of attempted steps, and REJ is the number of rejected steps.

which use commutators. The method presented of order 4 use 5 exponentials per step, this one more than the cheapest methods which use commutators. In particular, we believe that the new methods are promising for solving ODEs on homogeneous manifolds with large stabilizer subgroups and for solving stiff problems.

References

[1] E. Celledoni and A. Iserles. Approximating the matrix exponential from a Lie algebra to a Lie group. *Math. Comp.*, 69:1457–1480, 2000.

[2] E. Celledoni and B. Owren. On the implementation of Lie group methods on the Stiefel manifold. Technical Report Numerics No. 9/2001, The Norwegian University of Science and Technology, Trondheim, Norway, 2001.

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

[4] M. L. Curtis. *Matrix Groups.* Springer-Verlag, Second edition, 1984.

[5] E. Lodden. Geometric integration of the heat equation. Master's thesis, University of Bergen, 2000.

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

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

[8] H. Munthe-Kaas. High order Runge–Kutta methods on manifolds. *Appl. Numer. Math.*, 29:115–127, 1999.

[9] H. Munthe-Kaas and B. Owren. Computations in a free Lie algebra. *Phil. Trans. Royal Soc. A*, 357:957–981, 1999.

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

[11] P. J. Olver. *Applications of Lie Groups to Differential Equations.* GTM 107. Springer-Verlag, Second edition, 1993.

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

[13] F. A. Potra and W. C. Rheinboldt. On the numerical solution of Euler-Lagrange equations. *Mech. Structures Mach.*, 19:1–18, 1991.

[14] F. A. Potra and J. Yen. Implicit numerical integration for Euler-Lagrange equations via tangent space parametrization. *Mech. Structures Mach.*, 19:77–98, 1991.

[15] A. Suslowicz. Application of numerical Lie group integrators to parabolic PDEs. Technical Report 013–2001, University of Bergen, 2001. Submitted to BIT.