Application of Geometric Integration to some Mechanical Problems[*]

Kenth Engø[†] and Arne Marthinsen[‡]

Department of Informatics, University of Bergen, N-5020 Bergen, Norway
Department of Mathematical Sciences, NTNU, N-7034 Trondheim, Norway

Version March 22, 1997

Abstract

We apply Munthe-Kaas and Crouch-Grossman methods in the solution of some mechanical problems. These methods are quite new, and they exploit intrinsic properties of the manifolds defined by the mechanical problems, thus ensuring that the numerical solution obey underlying constraints. A brief introduction to the methods are presented, and numerical simulations show some of the properties they possess. We also discuss error estimation and stepsize selection for some of these methods.

AMS Subject Classification: 65L06, 34A50

Key Words: geometric integration, ordinary differential equations, manifolds, numerical analysis, initial value problems, mechanical engineering

1 Introduction

During the past decade there has been an increasing interest in studying numerical methods that preserve certain properties of some differential equations. The reason is that some physical systems possess conserved quantities, and that the solutions of the governing equations of the systems also should contain these invariants. Typical examples are the symplectic structure in a Hamiltonian system, the energy in a conservative mechanical system and the angular momentum of a rotating rigid body in space. Classical numerical methods normally fail to preserve such as the above mentioned quantities (see e.g. [7]).

We view the constraints or invariants as a manifold embedded in some Euclidean space, on which the numerical solution should evolve. The numerical methods we use in this work are constructed so that the computed solution stays on the manifold, and the numerical error will only result in error *on* the manifold. Hence, the numerical solution will obey the underlying constraint. Due to physical understanding of many problems, this manifold is often known in advance. The integration methods are constructed to use this knowledge to integrate exactly with respect to the constraints.

[*] 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/

[†] Email: Kenth.Engo@ii.uib.no, WWW: http://www.ii.uib.no/~kenth

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

In computational mechanics, numerical integration of ordinary differential equations on Lie groups (continuous groups that are also manifolds, e.g. the configuration space of a rigid body) is of interest and has been investigated by e.g. Simo and Wong [15]. In this work we consider two essentially different approaches as introduced by Crouch and Grossman in [3] and Munthe-Kaas in [9, 11]. These methods may in particular be implemented for Lie groups. In Euclidean space, both types of methods coincide with the traditional Runge-Kutta formulation.

We present some numerical experiments showing the qualitative behavior of the new methods. To demonstrate how they work, we compare with the well known classical integrators of Newmark and Runge-Kutta type.

2 The Numerical Methods

The numerical methods used in this paper include classical Runge-Kutta, Newmark, Crouch-Grossman, and Munthe-Kaas methods. In this section we briefly explain how these methods integrate a system of ordinary differential equations. A geometric interpretation of the methods is given in [8].

It will be seen that the numerical methods are fundamentally different. Classical methods are constructed to work in vector spaces, e.g. $\mathbb{R}^n$. The two other classes of methods work on more general geometric objects called *manifolds*. In particular they work on Lie groups. The problems solved in this paper are formulated on Lie groups.

In general we may view a manifold as embedded in a higher dimensional Euclidean space, $\mathbb{R}^N$. We may therefore apply the classical methods to a rewritten version of the Lie group formulation.

2.1 Runge-Kutta Methods

Classical, s-stage, Runge-Kutta methods for solving the first order initial value problem

$$y'(t) = f(t, y), \qquad y(t_0) = y_0, \tag{1}$$

on the interval $[t_n, t_n + h]$, are traditionally written as

$$\begin{aligned} k_i &= f\left(t_n + c_i h, y_n + h\sum_{j=1}^{s} a_{ij}k_j\right), \qquad i = 1, \ldots, s, \\ y_{n+1} &= y_n + h\sum_{i=1}^{s} b_i k_i, \end{aligned}$$

where y_n and y_{n+1} are approximations to $y(t_n)$ and $y(t_{n+1})$, respectively. The coefficients of a method are normally specified in a Butcher tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{s,s-1} & a_{ss} \\ \hline & b_1 & \cdots & b_{s-1} & b_s \end{array} \quad = \quad \begin{array}{c|c} c & A \\ \hline & b^T \end{array}.$$

In this paper we only consider explicit and diagonal implicit methods, i.e. methods for which $a_{ij} = 0$ when $i \leq j$ and $i < j$, respectively. In the explicit case, each stage, k_i, depends only on previously computed stages $k_j, j < i$. With diagonal implicit methods we need to solve a system of equations in each stage. Runge-Kutta methods are extensively studied in e.g. [2, 5].

2.2 Crouch-Grossman Methods

The Crouch-Grossman methods were introduced in [3]. The methods are based upon the existence of a *frame* on a differentiable manifold, $\mathcal{M}$, with tangent space $\mathrm{T}\mathcal{M}_p$, $p \in \mathcal{M}$, and tangent bundle $\mathrm{T}\mathcal{M}$ defined as $\mathrm{T}\mathcal{M} = \bigcup_{p\in\mathcal{M}} \mathrm{T}\mathcal{M}_p$. A vector field on $\mathcal{M}$ is a section of $\mathrm{T}\mathcal{M}$,

$$\begin{array}{rcccl} F & : & \mathcal{M} & \to & \mathrm{T}\mathcal{M} \\ & & p & \mapsto & v(p), \quad v(p) \in \mathrm{T}\mathcal{M}_p. \end{array}$$

The frame is a set of smooth vector fields $E_1, \ldots, E_d$, which at each point $p \in \mathcal{M}$ span the tangent space $\mathrm{T}\mathcal{M}|_p$. A vector field relative to this frame may be written as

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

where $f_i : \mathcal{M} \to \mathbb{R}$ are smooth functions. Ordinary differential equations on $\mathcal{M}$ may be written as

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

The initial value problem is given by (2) together with $y(0) = p_0 \in \mathcal{M}$.

Vector fields of this kind with coefficients frozen at a point $p \in \mathcal{M}$ are denoted by

$$F_p = \sum_{i=1}^{d} f_i(p) E_i,$$

and the Crouch-Grossman methods integrate using such frozen vector fields on $\mathcal{M}$.

Algorithm 2.1 *(Crouch-Grossman Methods)*

$$\begin{array}{rcl} p & = & y_k \\ Y_r & = & \exp(h a_{r,s} F_{Y_s}) \circ \exp(h a_{r,s-1} F_{Y_{s-1}}) \circ \cdots \circ \exp(h a_{r,1} F_{Y_1})(p), \quad r = 1, \ldots, s \\ y_{k+1} & = & \exp(h b_s F_{Y_s}) \circ \exp(h b_{s-1} F_{Y_{s-1}}) \circ \cdots \circ \exp(h b_1 F_{Y_1})(p), \end{array}$$

where the exponentiation of a vector field denotes its flow. We picture the coefficients $a_{i,j}$ and b_i in a tableau similar to the Butcher tableau of classical Runge-Kutta methods.

Algorithm 2.1 specifies a general, implicit method. In this paper we consider, as for classical Runge-Kutta methods, only explicit and diagonal implicit methods. When integrating with an implicit Crouch-Grossman method, we need to solve a system of nonlinear algebraic equations. This may, for Lie groups, be accomplished by implementation of the generalized Newton method developed by Owren and Welfert [14].

Because of the special construction of the Crouch-Grossman methods, we need to impose a number of extra order conditions in addition to the classical ones. Therefore, the coefficients of a Crouch-Grossman method of order q also generates an order q method when implemented in the classical Runge-Kutta setting. The converse is of course not true. In [3], Crouch and Grossman develop the algorithm and construct methods of order three. Owren and Marthinsen [12] develop a systematic order theory and present a fourth order method with $s = 5$ stages.

2.3 Munthe-Kaas Methods

Munthe-Kaas methods are described in [10, 9, 11]. Assume that the manifold $\mathcal{M}$ is a matrix Lie group G acting on itself by left multiplication [10]. We define a Lie algebra action $\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$

by

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

where $\exp(u)$ is the matrix exponential and $\cdot$ denotes matrix multiplication. Then problem (2) takes the form

$$y' = f(y) \cdot y, \qquad y \in G, \qquad f : G \to \mathfrak{g}.$$

In this case the Munthe-Kaas method is defined as follows.

Algorithm 2.2 *(Munthe-Kaas Methods)*

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

    $u_i = h \sum_{j=1}^{s} a_{ij} \tilde{k}_j$
    $k_i = f\left(c_i h, \lambda(u_i, y_0)\right)$
    $\tilde{k}_i = \mathrm{dexpinv}(u_i, k_i, q)$

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

where the coefficients are given by the qth order classical Runge-Kutta method's Butcher tableau [5], and the dexpinv *function is defined by*

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

where $[\,,\,]$ *is the matrix commutator and B_k is the kth Bernoulli number.*

The matrix commutator is defined by $[A, B] = AB \Leftrightarrow BA$ when A and B are matrices.

The Munthe-Kaas methods actually implement classical Runge-Kutta methods on a transformed problem. Therefore, as long as the transformation is done with sufficient order of accuracy, the coefficients of classical order q methods also give rise to order q methods of Munthe-Kaas type.

2.4 Newmark Methods

The Hilber-Hughes-Taylor method [6] may be used for solving the linearized version of the initial value problem

$$\widehat{f}(t, y, y', y'') = y'' \Leftrightarrow g(t, y, y') = 0, \qquad y(t_0) = y_0, \qquad y'(t_0) = y_0'. \tag{3}$$

The linearized version of (3) may be written as

$$My'' + Cy' + Ky = F(t), \qquad y(t_0) = y_0, \qquad y'(t_0) = y_0',$$

where

$$M = \frac{\partial \widehat{f}}{\partial y''}, \qquad C = \frac{\partial \widehat{f}}{\partial y'} \qquad \text{and} \qquad K = \frac{\partial \widehat{f}}{\partial y}$$

are $m \times m$ matrices and M is non-singular. The Hilber-Hughes-Taylor method consists of computing

$$\begin{aligned} d_{n+1} &= d_n + h v_n + h^2 \left[\left(\frac{1}{2} \Leftrightarrow \beta \right) a_n + \beta a_{n+1} \right] \\ v_{n+1} &= v_n + h \left[(1 \Leftrightarrow \gamma)\, a_n + \gamma a_{n+1} \right] \end{aligned}$$

where a_{n+1} is found by solving

$$Ma_{n+1} + C\left[(1+\alpha)v_{n+1} \Leftrightarrow \alpha v_n\right] + K\left[(1+\alpha)d_{n+1} \Leftrightarrow \alpha d_n\right] = F(t_{n+1}+\alpha h).$$

Here, d_n and v_n are approximations to $y(t_n)$ and $y'(t_n)$, respectively. The classical Newmark method is obtained with $\alpha = 0$, and with the coefficients $\beta = 1/4$ and $\gamma = 1/2$ the Newmark method is actually the traditional trapezoidal rule. This is the only second order Newmark method. An analysis of certain properties of Newmark methods and a family of Runge-Kutta methods is performed in [13].

Note that (3) may be written as a first order system (1), and the Runge-Kutta method is applicable also for second order initial value problems. Solution of (3) by Runge-Kutta-Nyström methods is discussed in [5].

3 Error Estimation and Stepsize Selection Strategies

We would like to construct a stepsize selection strategy for the Crouch-Grossman and Munthe-Kaas methods. As for classical Runge-Kutta methods, we consider embedded methods that provide an extra set of weights, $\widehat{b}$, that advances the solution from t_k to t_{k+1}:

$$y_{k+1} = e^{hb_sF_p^s}\circ\cdots\circ e^{hb_1F_p^1}p \quad\text{and}\quad \widehat{y}_{k+1} = e^{h\widehat{b}_sF_p^s}\circ\cdots\circ e^{h\widehat{b}_1F_p^1}p,$$

in the case of Crouch-Grossman methods, and

$$\begin{array}{rcl} v &=& h\sum_{j=1}^{s} b_j\tilde{k}_j \\ y_{k+1} &=& \lambda(v, y_k) \end{array} \qquad\text{and}\qquad \begin{array}{rcl} \widehat{v} &=& h\sum_{j=1}^{s} \widehat{b}_j\tilde{k}_j \\ \widehat{y}_{k+1} &=& \lambda(\widehat{v}, y_k) \end{array}$$

in the case of Munthe-Kaas methods. The weights b and $\widehat{b}$ are chosen so that y_{k+1} and $\widehat{y}_{k+1}$ correspond to methods of order p and $p+1$, respectively.

Consider first Crouch-Grossman methods. Since we work on general manifolds, we can not talk about a distance in the classical sense, e.g. $||y_{k+1} \Leftrightarrow \widehat{y}_{k+1}||$, since addition is not a defined operation. Instead we now compute $u_{k+1} \in \mathfrak{g}$ such that

$$y_{k+1}^{-1}\widehat{y}_{k+1} = \exp(u_{k+1})$$

and define the distance between y_{k+1} and $\widehat{y}_{k+1}$ as

$$\widehat{e}_{k+1} = ||u_{k+1}||.$$

We have now defined the distance between two elements on the manifold through an element in the Lie algebra (cf. [14]), where we may define a norm.

The Munthe-Kaas methods make linear combinations in the Lie algebra, and we easily obtain an error estimate through

$$\widehat{e}_{k+1} = ||v \Leftrightarrow \widehat{v}||.$$

We assume local extrapolation.

In order to achieve the local error estimate $r_{k+1} = \varepsilon$, the next stepsize h_{k+1} is chosen as follows. Compute first

$$\widehat{h}_{k+1} = \alpha\left(\frac{\varepsilon}{r_k}\right)^{1/(p+1)} h_k,$$

where p is the order of the lower order approximation scheme, $r_k = \widehat{e}_k$ (see e.g. [4]), and α is a "pessimist factor". It may typically be between 0.8 and 0.9, but is heuristically determined. In order to prevent rapid oscillations of the stepsize, it is common to restrict how much the stepsize is allowed to change from one step to another. One typically uses the strategy

$$h_{k+1} = \min\left(h_{\max}, \max\left(\alpha_{\text{small}} h_k, \min\left(\alpha_{\text{large}} h_k, \widehat{h}_{k+1}\right)\right)\right),$$

where $h_{\max}$ is the maximum allowed stepsize, and α_{small} and α_{large} are two constants (typically 0.5 and 2.0, respectively).

If the local error exceeds the tolerance by a factor more than α_{accept}, which again is a constant chosen by the user (typically 1.2), then we should reject the step and retry with a smaller stepsize h_{k+1}. This algorithm should proceed until the local error estimate satisfies

$$\widehat{e}_{k+1} \leq \alpha_{\text{accept}} \varepsilon.$$

4 Numerical Simulations

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{3}{4} & \frac{3}{4} & & \\ \frac{17}{24} & \frac{119}{216} & \frac{17}{108} & \\ \hline & \frac{13}{51} & \Leftrightarrow\frac{2}{3} & \frac{24}{17} \end{array} \qquad \begin{array}{c|ccc} 1 & 1 & & \\ \frac{5}{8} & \Leftrightarrow\frac{3}{8} & 1 & \\ \frac{23}{48} & \Leftrightarrow\frac{125}{432} & \Leftrightarrow\frac{25}{108} & 1 \\ \hline & \frac{31}{75} & \Leftrightarrow\frac{4}{3} & \frac{48}{25} \end{array}$$

RK4 **CG3** **ICG3**

The numerical simulations are performed with methods of the classes described in Section 2. Let the methods be defined as follows:

RK4: a classical Runge-Kutta method defined by the Butcher tableau **RK4**

CG3: a third order Crouch-Grossman method defined by the Butcher tableau **CG3**

ICG3: a third order semi-implicit Crouch-Grossman method defined by the Butcher tableau **ICG3**

MK4: a fourth order Munthe-Kaas method defined by the same Butcher tableau as method **RK4**.

N: a Newmark method with $\alpha = 0$, $\beta = 1/4$ and $\gamma = 1/2$.

We have implemented the stepsize selection strategy presented in the previous section on the humming top test problem. The Crouch-Grossman method defined by the Butcher tableau

$$\begin{array}{c|c} c & A \\ \hline & b^T \\ \hline & \widehat{b}^T \end{array} \qquad\qquad \begin{array}{c|ccc} 0 & 0 & & \\ \frac{3}{4} & \frac{3}{4} & & \\ \frac{17}{24} & \frac{119}{216} & \frac{17}{108} & \\ \hline & \frac{3}{4} & \frac{31}{4} & \Leftrightarrow\frac{15}{2} \\ \hline & \frac{13}{51} & \Leftrightarrow\frac{2}{3} & \frac{24}{17} \end{array} \qquad (4)$$

is of order $2(3)$, and it will in the following be denoted by **CG2(3)**. In accordance with standard notation, we extend the Butcher tableau with an extra row for the sake of error estimation. To the end of this work, the b and $\widehat{b}$ weights give rise to methods of order p and $p+1$, respectively.

We have also tested the fourth order scheme presented in [12] together with the scheme **CG3**. This method will be called **CG3(4)**.

As a reference solution (the "correct" solution) we use the output from Matlab routine ode45 with tolerance 10^{-14}. The codes **ODE23** and **ODE45** are Matlab routines with the same name.

4.1 The Humming Top

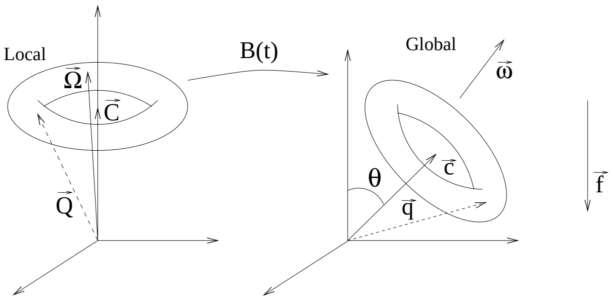

Figure 1: A schematic picture of the humming top in local and global coordinates.

The humming top is modeled on the Lie group $G = \mathrm{SO}(3) \times \mathfrak{so}(3)$, and we therefore denote an element in the group by the pair $(B(t), \omega(t))$ (see Figure 1). The differential equation describing the motion of the top is given by $y' = F(y)y$, where y is the pair $(B(t), \omega(t))$ and $F : G \rightarrow \mathfrak{g}$, where $\mathfrak{g} = \mathfrak{so}(3) \times \mathfrak{so}(3)$ is the Lie algebra of G.

There is a one to one correspondence between the three-vectors in space and the skew-symmetric three by three matrices, in the following manner

$$\vec{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \leftrightarrow \begin{bmatrix} 0 & \Leftrightarrow\omega_3 & \omega_2 \\ \omega_3 & 0 & \Leftrightarrow\omega_1 \\ \Leftrightarrow\omega_2 & \omega_1 & 0 \end{bmatrix} = \omega. \tag{5}$$

In the following derivations of the governing equations we suppress time dependence. We also let prime ($'$) denote derivation with respect to time. First, we find the expression for B': The position of the point $\vec{Q}$ is in global coordinates given by $\vec{q} = B\vec{Q}$. Therefore

$$\vec{q}\,' = B'\vec{Q}.$$

We also have that

$$\vec{q}\,' = \vec{\omega} \times \vec{q} = \omega\vec{q} = \omega B\vec{Q},$$

and thus it follows that

$$B' = \omega B.$$

Second, we find the expression for ω': The angular momentum around $\vec{\omega}$ is given by the well-known relationship

$$\vec{L} = \mathbb{I}\,\vec{\omega} \qquad \Leftrightarrow \qquad \vec{\omega} = \mathbb{I}^{-1}\vec{L},$$

where $\mathbb{I}$ is the inertial tensor of the humming top. The time derivative of $\vec{\omega}$ is

$$\begin{aligned} \vec{\omega}' &= (\mathbb{I}^{-1})'\vec{L} + \mathbb{I}^{-1}\vec{L}' \\ &= (\mathbb{I}^{-1})'\vec{L} + \mathbb{I}^{-1}\vec{M}, \end{aligned}$$

where we have used that the derivative of the angular momentum, $\vec{L}'$, is equal to the torque, $\vec{M}$. In local coordinates,

$$\vec{L}_0 = \mathbb{I}_0\vec{\Omega}.$$

Therefore,

$$\vec{L} = B\vec{L}_0 = B\mathbb{I}_0B^{-1}\vec{\omega},$$

and it follows that $\mathbb{I} = B\mathbb{I}_0 B^{-1} = B\mathbb{I}_0 B^T$ since B is orthogonal. The time derivative of $\mathbb{I}^{-1}$ is given by

$$\begin{aligned}
(\mathbb{I}^{-1})' &= \left(B\mathbb{I}_0^{-1}B^{-1}\right)' \\
&= B'\mathbb{I}_0^{-1}B^T + B\mathbb{I}_0^{-1}B^{T\prime} \\
&= \omega B\mathbb{I}_0^{-1}B^T + B\mathbb{I}_0^{-1}(\omega B)^T \\
&= \omega\mathbb{I}^{-1} + B\mathbb{I}_0^{-1}B^T\omega^T \\
&= \omega\mathbb{I}^{-1} + \mathbb{I}^{-1}(\Leftrightarrow\omega) \\
&= \left[\omega, \mathbb{I}^{-1}\right],
\end{aligned}$$

where we have used the matrix commutator notation $[\,,\,]$.

We finally arrive at the following expression for the derivative of the angular momentum:

$$\vec{\omega}'(t) = [\omega(t), \mathbb{I}^{-1}(t)]\vec{L}(t) + \mathbb{I}^{-1}(t)\vec{M}, \tag{6}$$

where $\mathbb{I}^{-1}(t) = B(t)\mathbb{I}_0^{-1}B^{-1}(t)$ and $\vec{M} = m\vec{c}\times\vec{f}$. Here, $\vec{c}$ is the centroid vector and $\vec{f}$ is the vector of gravity.

This can further be recognized as an obscure version of the Euler equations. Consider the following steps, starting with (6):

$$\begin{aligned}
\mathbb{I}^{-1}\vec{M} &= \vec{\omega}' + [\mathbb{I}^{-1}, \omega]\vec{L} \\
\vec{M} &= \mathbb{I}\,\vec{\omega}' + \mathbb{I}\,[\mathbb{I}^{-1}, \omega]\vec{L} \\
&= \mathbb{I}\,\vec{\omega}' + \omega\vec{L} \Leftrightarrow \mathbb{I}\,\omega\mathbb{I}^{-1}\vec{L} \\
&= \mathbb{I}\,\vec{\omega}' + \vec{\omega}\times\vec{L} \Leftrightarrow \mathbb{I}\,\omega\vec{\omega} \\
&= \mathbb{I}\,\vec{\omega}' + \vec{\omega}\times\vec{L}.
\end{aligned}$$

The last line follows from the previous by recognizing that $\omega\vec{\omega} = \vec{\omega}\times\vec{\omega} = 0$. We have now arrived at the Euler equations in a well-known form, thus verifying the validity of (6).

To solve the above equations, we need to define all the legal operations. These are

Product in G:	$(a, v)\cdot(b, w) = (a\cdot b, v + w)$.
Addition in $\mathfrak{g}$:	$(u, v) + (\widetilde{u}, \widetilde{v}) = (u + \widetilde{u}, v + \widetilde{v})$.
Multiplication by scalar in $\mathfrak{g}$:	$\alpha(u, v) = (\alpha u, \alpha v)$.
Lie bracket in $\mathfrak{g}$:	$[(u, v), (\widetilde{u}, \widetilde{v})] = ([u, \widetilde{u}], 0)$.
Exponential map from $\mathfrak{g}$ to G:	$\exp(u, v) = (\exp(u), v)$.

In the test run we used the following initial values for B_0 and ω_0:

$$B_0 = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & \cos(\phi) & \sin(\phi) \\ 0.0 & \Leftrightarrow\sin(\phi) & \cos(\phi) \end{bmatrix} \quad\text{and}\quad \omega_0 = \begin{bmatrix} 0.0 & 50.0 & 0.0 \\ \Leftrightarrow 50.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \end{bmatrix},$$

with $\phi = \pi/16$. The constant quantities $\vec{C}$, $\vec{f}$ and $\mathbb{I}_0$ are given as

$$\vec{C} = \begin{bmatrix} 0 \\ 0 \\ \sqrt{3}/2 \end{bmatrix}, \qquad \vec{f} = \begin{bmatrix} 0 \\ 0 \\ \Leftrightarrow 9.81 \end{bmatrix} \quad\text{and}\quad \mathbb{I}_0 = \frac{1}{8}\begin{bmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

Figure 2 shows the global error versus stepsize of some of the schemes. The lines in the plot are included for reference only, and they have slopes 2, 3 and 4. We are here talking about order on the

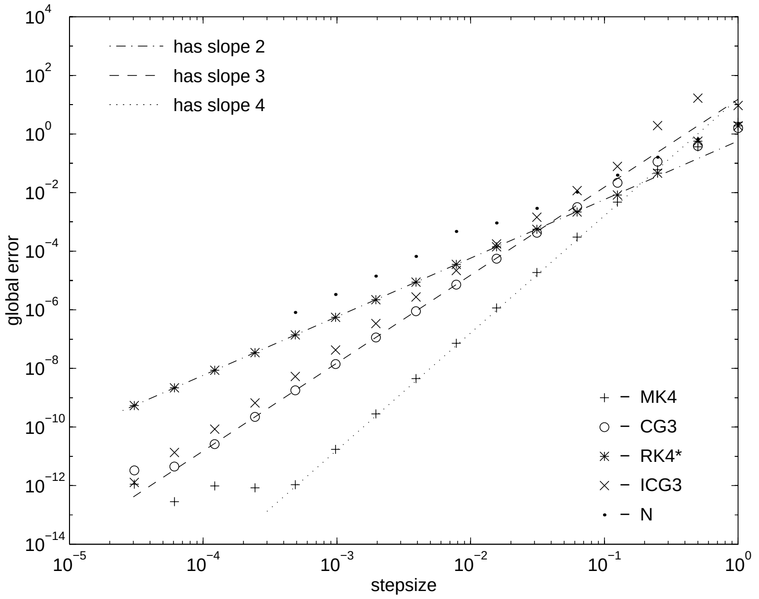

Figure 2: Global error versus stepsize for some of the schemes. **RK4*** is the **RK4** coefficients implemented as a Crouch-Grossman method.

manifold. Note therefore that **RK4**, that has order four in the classical sense, only exhibits order two in in the Crouch-Grossman setting (i.e. implementing using Algorithm 2.1). This is consistent with the theory described in [12].

As may be seen in Figure 3, the orthogonality of the matrices generated by the Munthe-Kaas and Crouch-Grossman methods is preserved to machine accuracy, but for the classical Runge-Kutta method the orthogonality is instantly lost.

It is known that the humming top possesses three first integrals (see [1]). These are the total energy (kinetic plus potential energy), the projection of the angular momentum on the vertical (e_z), L_z, and the projection of the angular momentum on the axis of symmetry (e_3) in the case of a symmetric top, L_3. These are given by $L_z = \vec{L} \cdot \vec{e}_z$, $L_3 = \vec{L} \cdot \vec{e}_3 = \vec{L} \cdot (B\vec{e}_z)$ and $E = K + U = \frac{1}{2}(\mathbb{I}\,\vec{\omega}) \cdot \vec{\omega} + mg\ell\cos(\theta)$, where $\ell = \|\vec{C}\|_2$.

In Figure 4 we have shown these first integrals for a simulation. The new methods are not con-

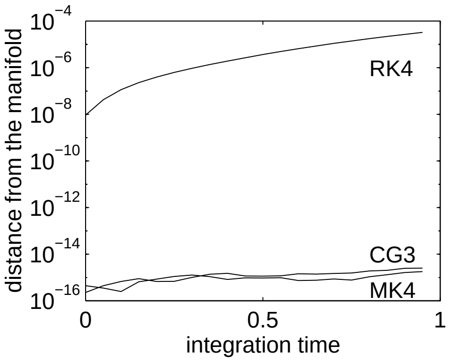

Figure 3: The distance from the manifold for **MK4**, **CG3** and **RK4**. The simulation was done with constant stepsize, $h = 0.01$.

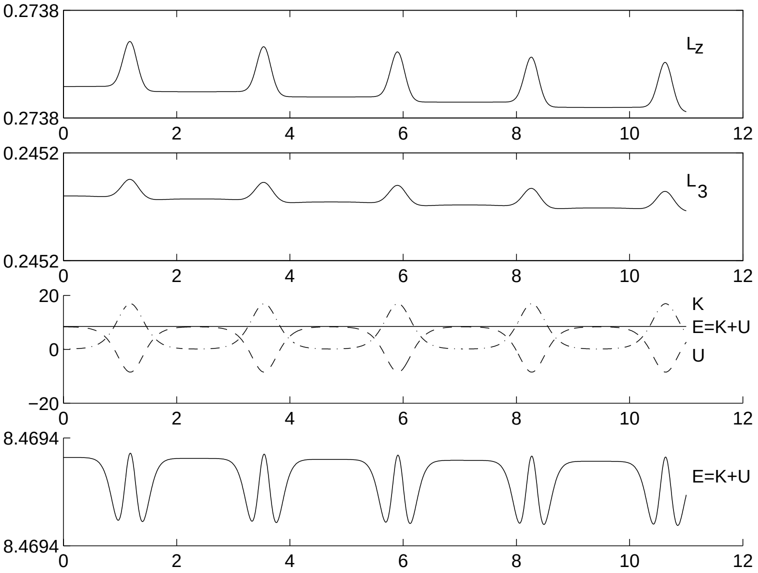

Figure 4: First integrals of the humming top integrated by the **MK4** method: the upper figure shows the projection of the angular momentum on the e_z axis; the second figure shows the projection of the angular momentum on the e_3 axis; the third figure shows the kinetic (K), potential (U) and total (E) energy; and the lower figure shows the total energy alone.

structed to preserve the first integrals, but although it is clear that they are not preserved, the errors are quite small: $|L_z(10) \Leftrightarrow L_z(0)| = 9.6448 \cdot 10^{-8}$, $|L_3(10) \Leftrightarrow L_3(0)| = 2.3164 \cdot 10^{-8}$ and $|E(10) \Leftrightarrow E(0)| = 1.0927 \cdot 10^{-8}$. The constant stepsize used in these simulations was $h = 0.01$. Figure 5 shows the corresponding results from integration with the classical Runge-Kutta method with the **RK4** coefficients. The corresponding errors are here: $|L_z(10) \Leftrightarrow L_z(0)| = 7.1695 \cdot 10^{-7}$, $|L_3(10) \Leftrightarrow L_3(0)| = 2.3598 \cdot 10^{-7}$ and $|E(10) \Leftrightarrow E(0)| = 5.4514 \cdot 10^{-7}$.

The trace of a point on the humming top from time $t_0 = 0$ to $t_{\text{end}} = 11$ is plotted in Figure 6.

Figure 7 shows the number of steps taken by some of the Crouch-Grossman methods when a given tolerance is imposed on the result. Note that the global tolerance obtained is not the same as the local tolerance imposed on the stepsize selection scheme. The relationships between these quantities are shown in Figure 8. We have compared with the global error of the solution computed by the Matlab routines ode23 and ode45, when executed with a range of tolerances.

Note that the only information we can read out of the graphs shown in Figures 7 and 8 is of a relative nature. We can only conclude that the stepsize selection strategy and hence also the error estimation process produce results that behave like the corresponding results for classical Runge-Kutta methods.

It should be emphasized that the Crouch-Grossman methods require more work per step than the classical Runge-Kutta methods. In the classical case, we only perform function evaluations and make linear combinations to compute the numerical approximation. In the Crouch-Grossman case, we have to compute flows of certain vector fields. This is done by evaluating the matrix exponential a number of times for each step. Since this computation is a major time consuming task, it is of great importance to look for more efficient ways to compute flows. It is for instance well known that the Euler-Rodrigues formula may be applied in the case of exponentiating skew-symmetric 3×3 matrices.

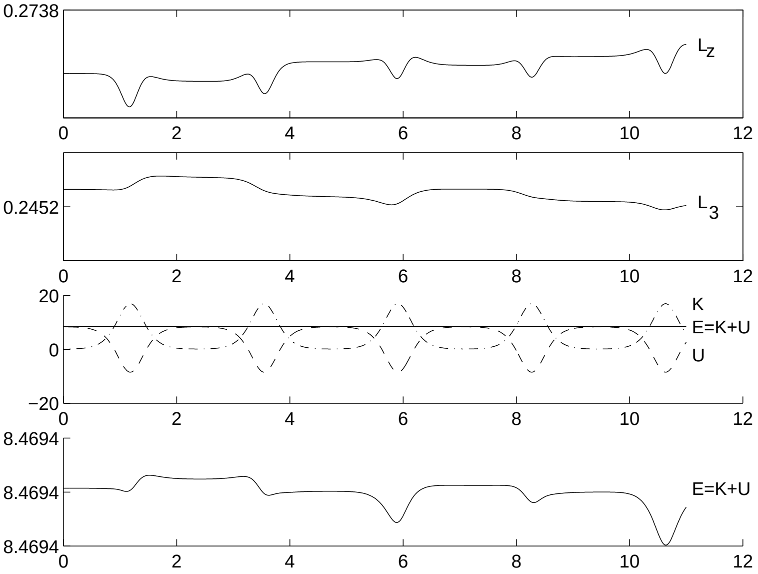

Figure 5: First integrals of the humming top integrated by the classical Runge-Kutta method with the **RK4** coefficients: the upper figure shows the projection of the angular momentum on the e_z axis; the second figure shows the projection of the angular momentum on the e_3 axis; the third figure shows the kinetic (K), potential (U) and total (E) energy; and the lower figure shows the total energy alone.

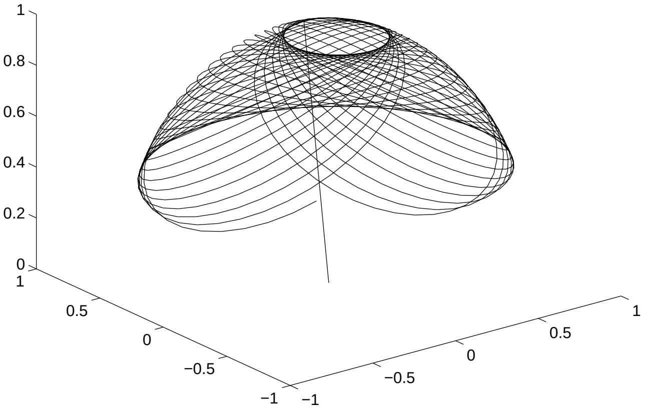

Figure 6: The trace of a point on the humming top.

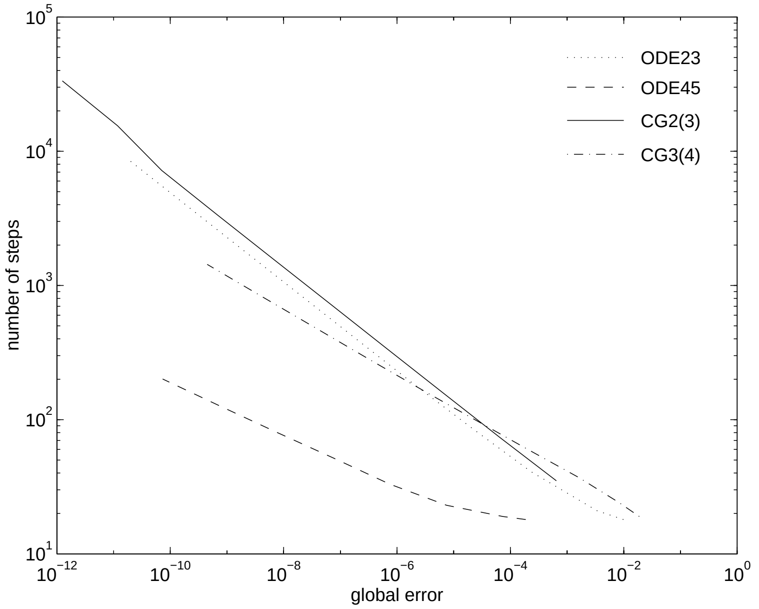

Figure 7: Number of steps as a function of the global error

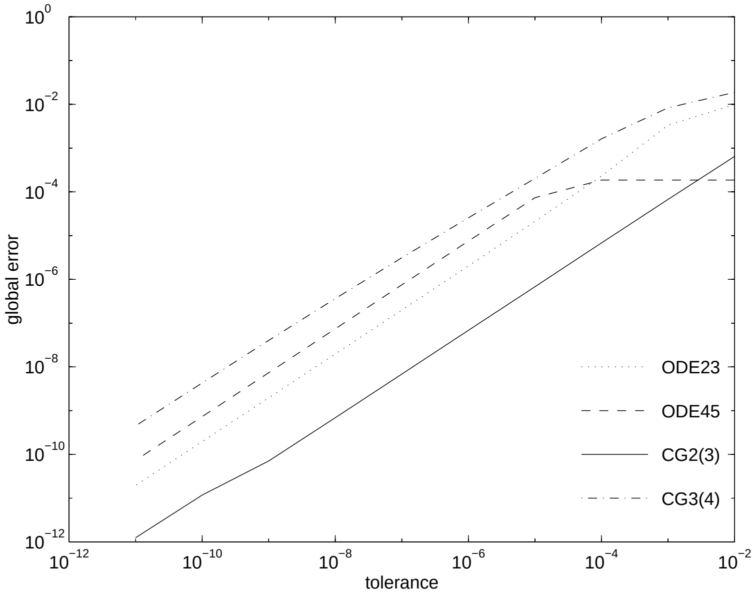

Figure 8: Connection between tolerance imposed on the stepsize selection scheme and the global error

4.2 The Pendulum

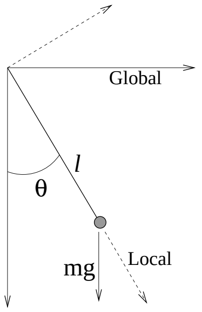

Figure 9: A schematic picture of the pendulum.

We now consider the traditional pendulum problem. The simple pendulum is modeled on the Lie group SO$(2) \times \mathfrak{so}(2)$, and an element in the group is given by the pair $(B(t), \omega(t))$. The differential equation describing the motion of the pendulum is given by $y' = F(y)y$, where y is the pair $(B(t), \omega(t))$ and $F : G \rightarrow \mathfrak{g}$, where $\mathfrak{g} = \mathfrak{so}(2) \times \mathfrak{so}(2)$ is the Lie algebra of G.

Traditionally, the pendulum is described by the equation

$$\ddot{\theta}(t) + \frac{g}{\ell} \sin(\theta(t)) = 0, \tag{7}$$

where g is the magnitude of the gravitational force and ℓ is the length of the string supporting the bob. To solve the problem with the new methods, we need to formulate the problem in terms of the Lie group G and the Lie algebra $\mathfrak{g}$.

Consider two coordinate systems, one local and one global, as shown in Figure 9. The correspondence between the two coordinate systems is given by a plane rotation. Every rotation θ in the plane, is described by the matrix

$$B(\theta) = \begin{bmatrix} \cos(\theta) & \Leftrightarrow\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix},$$

and by differentiating B with respect to time, we obtain

$$B' = \omega B, \qquad \text{where} \qquad \omega = \begin{bmatrix} 0 & \Leftrightarrow\theta' \\ \theta' & 0 \end{bmatrix}.$$

Again we establish a correspondence like (5), but now it is between a scalar and a 2×2 matrix:

$$\bar{\omega} = \theta' \leftrightarrow \begin{bmatrix} 0 & \Leftrightarrow\theta' \\ \theta' & 0 \end{bmatrix} = \omega.$$

By considering (7) and noting that $\sin(\theta) = \vec{e}_y B \vec{e}_x$, where $\vec{e}_p$ is the unit vector in the p direction, we get

$$\bar{\omega}' = \Leftrightarrow\frac{g}{\ell}\, \vec{e}_y B \vec{e}_x = \Leftrightarrow\frac{g}{\ell}\, B_{21}.$$

It may be shown that for commutative groups, the Crouch-Grossman and Munthe-Kaas methods reduce to the classical Runge-Kutta methods. The Lie group SO(2) is a commutative group, and it follows that the configuration group G is also commutative.

The operations used in the implementation of the pendulum problem are shown below:

Product in G:	$(a,v)\cdot(b,w) = (a\cdot b, v+w)$.
Addition in $\mathfrak{g}$:	$(u,v)+(\widetilde{u},\widetilde{v}) = (u+\widetilde{u}, v+\widetilde{v})$.
Multiplication by scalar in $\mathfrak{g}$:	$\alpha(u,v) = (\alpha u, \alpha v)$.
Lie bracket in $\mathfrak{g}$:	$[(u,v),(\widetilde{u},\widetilde{v})] = (0,0)$.
Exponential map from $\mathfrak{g}$ to G:	$\exp(u,v) = (\exp(u), v)$.

When solving (7) with a classical Runge-Kutta method, the length of the string supporting the bob is preserved, and hence the solution stays on the constraint manifold. As for the humming top, this is also the case when integrating with Crouch-Grossman methods.

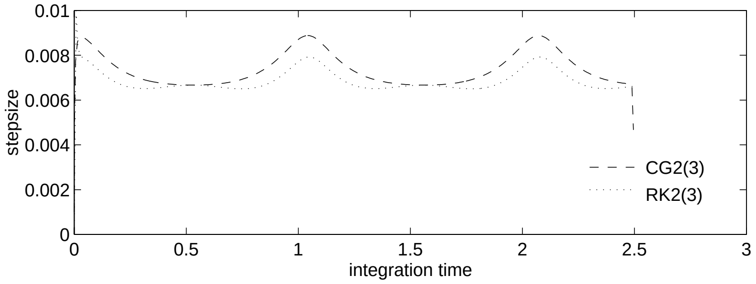

Figure 10: Stepsizes on the pendulum problem

Figure 10 shows what stepsizes the two methods **CG2(3)** and **RK2(3)** selected when integrating the pendulum problem from $t_0 = 0$ to $t_{\text{end}} = 2.5$ with tolerance 10^{-4}. **RK2(3)** is a classical Runge-Kutta method executed with the coefficients of **CG2(3)**. Since we have two different implementations, we may expect some differences. But the overall behavior stepsize selection and the magnitude of the stepsizes are similar for both the classical and the Crouch-Grossman methods.

In the case of a double pendulum one have to consider the configuration space $(\mathrm{SO}(2)\times\mathfrak{so}(2))\times(\mathrm{SO}(2)\times\mathfrak{so}(2))$. This is also a commutative group, and it does not add new insight of interest to this paper. The description of the double pendulum equations is just more intricate than in the simple pendulum case.

5 Concluding Remarks

We have presented some numerical simulations demonstrating some properties of Crouch-Grossman and Munthe-Kaas methods on some mechanical problems, and compared them with the classical Runge-Kutta and Newmark methods.

We have derived the equations describing the humming top and the pendulum in a setting suitable for Crouch-Grossman and Munthe-Kaas methods, and shown the connection between the humming top equations and the Euler equations. Although we have derived the equations in these two cases, it is not known how to do this for general problems. This will, however, be the topic of future work.

Acknowledgement. We would like to thank Fridtjov Irgens, Brynjulf Owren, Anne Kværnø and Hans Munthe-Kaas for enlightening discussions and valuable comments during the work with this paper.

References

[1] V. I. Arnold. *Mathematical Methods of Classical Mechanics.* Springer-Verlag, GTM 60, Second edition, 1989.

[2] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations.* Wiley, 1987.

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

[4] K. Gustafsson. *Control of Error and Convergence in ODE Solvers.* PhD thesis, Department of Automatic Control, Lund Institute of Technology, 1992.

[5] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I.* Springer Verlag, 1993.

[6] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor. Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics. *Earthquake Eng. and Struct. Dynamics*, 5:283–292, 1977.

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

[8] A. Marthinsen, H. Munthe-Kaas, and B. Owren. Simulation of Ordinary Differential Equations on Manifolds — Some Numerical Experiments and Verifications. *Modeling, Identification and Control*, 18:75–80, 1997.

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

[10] H. Munthe-Kaas. Geometric Integration of Differential Equations on Manifolds; a general perspective. Manuscript, 1997.

[11] H. Munthe-Kaas. Runge-Kutta Methods on Lie Groups. Technical Report 122, Department of Informatics, University of Bergen, Norway, 1997.

[12] 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.

[13] B. Owren and H. H. Simonsen. Alternative Integration Methods for Problems in Structural Dynamics. *Comput. Meth. in Appl. Mech. and Eng.*, 122:1–10, 1995.

[14] 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.

[15] J. C. Simo and K. K. Wong. Unconditionally Stable Algorithms for Rigid Body Dynamics that Exactly Preserve Energy and Momentum. *Internat. J. Numer. Methods Engrg.*, 31:19–52, 1991.