SIAM J. NUMER. ANAL.
Vol. 37, No. 1, pp. 269–285

INTERPOLATION IN LIE GROUPS*

ARNE MARTHINSEN†

Abstract. We consider interpolation in Lie groups. Based on points on the manifold together with tangent vectors at (some of) these points, we construct Hermite interpolation polynomials. If the points and tangent vectors are produced in the process of integrating an ordinary differential equation in terms of Lie-algebra actions, we use the truncated inverse of the differential of the exponential mapping and the truncated Baker–Campbell–Hausdorff formula to relatively cheaply construct an interpolation polynomial.

Much effort has lately been put into research on geometric integration, i.e., the process of integrating differential equations in such a way that the configuration space of the true solution is respected by the numerical solution. Some of these methods may be viewed as generalizations of classical methods, and we investigate the construction of intrinsic dense output devices as generalizations of the continuous Runge–Kutta methods.

Key words. geometric integration, numerical integration of ordinary differential equations on manifolds, interpolation, approximation, numerical analysis, Lie algebras, Lie groups

AMS subject classifications. 65L06, 34A50

PII. S0036142998338861

1. Introduction.

During the last few years there has been a growing interest in geometric integration, which in particular includes symplectic integration methods and Lie-group methods. Since many differential equations either explicitly or implicitly have solutions that evolve on some manifold (e.g., through constraints or invariants), it is vital to the qualitative behavior of the numerical solution that it is also restricted to the manifold. To accomplish this in a natural way, various approaches have been proposed and a number of numerical methods are constructed. Among these are the methods of Crouch and Grossman [2], Munthe-Kaas [22], Leimkuhler and Patrick [15], Dieci, Russell, and van Vleck [6], Eich et al. [7], Iserles and Nørsett [13], Zanna [36], and Sanz-Serna with coworkers [32, 31]. Some of the Lie-group methods are implicit, and the nonlinear equations may be solved with the Newton iteration schemes proposed by Owren and Welfert [26].

The geometric integration methods, as well as the traditional Runge–Kutta methods, are generally designed to approximate the solution of a system of ordinary differential equations on a mesh. The size of the mesh will typically be determined by some stepsize selection algorithm, and the steps will in general be chosen as large as possible constrained by tolerances to accuracy. Sometimes, however, the solution is required at points not included in the mesh, and an effective device for output of results at randomly chosen points within an interval should be provided. Some of the Lie-group methods are directly applicable only to Lie-type problems (e.g., the Magnus series methods [13]), but they may be extended to the general case by means of dynamic iteration or as in [37]. This again introduces the need for an approximation to the solution at intermediate points in the integration interval.

*Received by the editors May 18, 1998; accepted for publication (in revised form) April 6, 1999; published electronically December 3, 1999. This work was sponsored in part by British Petroleum under grant B94060.00 and by The Norwegian Research Council under contract 123655/410, as well as contract 111038/410, through the SYNODE project.

http://www.siam.org/journals/sinum/37-1/33886.html

†Norwegian University of Science and Technology, Department of Mathematical Sciences, N-7491 Trondheim, Norway (Arne.Marthinsen@math.ntnu.no, http://www.math.ntnu.no/~arnema/).

In this paper we discuss variants of Hermite interpolation in Lie groups. We also consider continuous extensions to some of the new geometric integration methods by equipping them with continuous weights.

It should be mentioned that Zanna [37] developed collocation and relaxed collocation methods for the Fer and the Magnus expansions. While Zanna's motivation apparently was to construct new integration methods, the motivation of this paper is to understand how to construct interpolants in the generalized setting of Lie groups. An analysis of the collocation idea is presented in section 3.3.

The rest of this paper is organized as follows. Section 2 is devoted to the development of a Hermite interpolation procedure. In section 3 we equip some of the recently developed Lie-group methods with continuous extensions. Finally, in section 4 we briefly discuss a procedure to be used with discrete integrators that are based on Lie-algebra actions.

2. Hermite interpolation in Lie groups.

In this section we construct a polynomial that interpolates data given as points on a matrix Lie group and tangent vectors in the tangent space at the points. The values may be obtained from discrete integration of an ordinary differential equation on a Lie group, G, say, but this is not a requirement; we do not use information about vector fields on G or particular integration schemes.

The setting is given by the following commutative diagram, where π denotes projection, π_1 denotes projection onto the first component, and $\mathrm{R}_y : G \to G$ is defined by $\mathrm{R}_y(z) = z \cdot y$, where $\cdot$ is the group multiplication (see [18]).

$$\begin{tikzcd}
\mathrm{T}\mathfrak{g} \simeq \mathfrak{g} \times \mathfrak{g} \arrow[r, "\mathsf{dexp}"] \arrow[d, "\pi_1"] & G \times \mathfrak{g} \arrow[r, "\mathrm{TR}"] \arrow[dr, "\pi_1"'] & \mathrm{T}G \arrow[d, "\pi"] & \\
\mathfrak{g} \arrow[rr, "\mathsf{exp}"] \arrow[rrr, bend right, "\lambda_y"'] & & G \arrow[r, "\mathrm{R}"] & G
\end{tikzcd}$$

Consider the map $\lambda_y : \mathfrak{g} \to G$ defined by

$$\lambda_y = \mathrm{R}_y \circ \exp,$$

where $y \in G$ is fixed. There exists a neighborhood $U \subset G$ around y in which λ_y is invertible. Hence for $z \in U$ we obtain a local coordinate system $\phi_y : G \to \mathfrak{g}$, given by

$$\phi_y(z) = \log(zy^{-1}).$$

The coordinates of this mapping are usually denoted by canonical coordinates of the first kind.

We want to compute the tangent mapping of this coordinate chart. Note that if $z = \lambda_y(u)$ and $v \in \mathfrak{g}$ then

$$\begin{aligned}
\mathrm{T}\lambda_y(u,v) &= \mathrm{T}_{\exp(u)}\mathrm{R}_y\Big(\mathrm{T}_e\mathrm{R}_{\exp(u)}\big(\mathrm{dexp}_u(v)\big)\Big) \\
&= \mathrm{T}_e\mathrm{R}_{\exp(u)\cdot y}\big(\mathrm{dexp}_u(v)\big) \\
&= \mathrm{T}_e\mathrm{R}_z\big(\mathrm{dexp}_u(v)\big).
\end{aligned}$$

Therefore, $\mathrm{T}\lambda_y : \mathrm{T}\mathfrak{g} \simeq \mathfrak{g} \times \mathfrak{g} \to \mathrm{T}G$. Hence, if we right trivialize elements in $\mathrm{T}G$, we get

$$\mathrm{T}\lambda_y(u,v) = \big(z, \mathrm{dexp}_u(v)\big) \in G \times \mathfrak{g}.$$

It readily follows that the mapping $\mathrm{T}\phi_y : G \times \mathfrak{g} \to \mathrm{T}\mathfrak{g}$ is given by

$$\mathrm{T}\phi_y(z, w) = \big(u, \operatorname{dexp}_u^{-1}(w)\big) = (u, v)$$

when $u = \phi_y(z)$ and $\operatorname{dexp}_u(v) = w$.

Assume now that we have available $k+1$ triplets, $\big(t_i, y_i, \widetilde{y}_i\big) \in \mathbb{R} \times G \times \mathfrak{g}$, $i = n, \ldots, n+k$. The tangent vector at the point y_i is given by $\dot{y}_i = \mathrm{T}_e\mathrm{R}_{y_i}(\widetilde{y}_i)$. We choose a reference element among the data, $(t_n, y_n, \widetilde{y}_n)$, say. Relative to this reference element we compute the "transformed" quantities $u_i = \log(y_i y_n^{-1})$ and $v_i = \operatorname{dexp}_{u_i}^{-1}(\widetilde{y}_i)$, $i = n, \ldots, n+k$. A Hermite interpolation polynomial, $p(t) \in \mathfrak{g}$, may now be computed based on the $2k+2$ elements

$$(t_n, 0), \quad (t_i, u_i),\ i = n+1, \ldots, n+k,$$

and

$$(t_n, \widetilde{y}_n), \quad (t_i, v_i),\ i = n+1, \ldots, n+k.$$

Here, $(t_i, u_i) \in \mathbb{R} \times \mathfrak{g}$ correspond to what usually is thought of as *points* and $(t_i, v_i) \in \mathbb{R} \times \mathfrak{g}$ correspond to the (translated) *derivatives* at the points.

We want to compute the unique polynomial $p : \mathbb{R} \to \mathfrak{g}$ of degree $2k+1$ such that $p(t_i) = u_i$ and $\dot{p}(t_i) = v_i$, $i = n, \ldots, n+k$. By using standard notation from the theory of divided differences (see, e.g., [30]), we have the divided difference

$$f[t_n, t_{n+1}, \ldots, t_{n+k}] = \sum_{\ell=0}^{k} \frac{u_\ell}{\prod_{\substack{j=0 \\ j\neq\ell}} (t_\ell - t_j)}, \tag{2.1}$$

and the polynomial is, by Newton's interpolation method, given by

$$p(t) = (t - t_n) f[t_n, t_{n+1}] + \cdots + \left\{ \prod_{j=n}^{n+k-1} (t - t_j) \right\} f[t_n, t_{n+1}, \ldots, t_{n+k}],$$

since $u_n = 0$. It is well known, however, that it is better to use a recurrence relation to compute $f[\cdots]$ than equation (2.1) directly. Given the divided differences $f[t_{j+1}, \ldots, t_{j+\ell}]$ and $f[t_j, \ldots, t_{j+\ell-1}]$ of order $\ell - 1$, we compute $f[t_j, t_{j+1}, \ldots, t_{j+\ell}]$ of order ℓ as

$$f[t_j, t_{j+1}, \ldots, t_{j+\ell}] = \frac{f[t_{j+1}, \ldots, t_{j+\ell}] - f[t_j, \ldots, t_{j+\ell-1}]}{t_{j+\ell} - t_j}.$$

Initially, we have that $f[t_j] = u_j$, $j = n, \ldots, n+k$. As in classical interpolation theory, we define $f[t_j, t_{j+1}] = v_j$ when $t_j = t_{j+1}$; hence Hermite and Hermite–Birkhoff [16] interpolation readily follow.

Assume now that $p(t)$ is computed. As will be shown in Theorem 2.1, an approximation to an element in G at time $\tilde{t} \in [t_n, t_{n+k}]$ is given as

$$y(\tilde{t}) \approx \lambda_{y_n}(p(\tilde{t})).$$

Accordingly, the element $\big(y(\tilde{t}), \widetilde{y}(\tilde{t})\big) \in G \times \mathfrak{g}$ is approximated by

$$\big(y(\tilde{t}), \widetilde{y}(\tilde{t})\big) \approx \mathrm{T}\lambda_{y_n}\big(p(\tilde{t}), \dot{p}(\tilde{t})\big). \tag{2.2}$$

THEOREM 2.1. *The numerical approximation* (2.2) *to* $\big(y(\tilde{t}), \widetilde{y}(\tilde{t})\big) \in G \times \mathfrak{g}$, $\tilde{t} \in [t_n, t_{n+k}]$, *based on interpolation of* $2k+2$ *points,* $\big(y(t_i), \widetilde{y}(t_i)\big) \in G \times \mathfrak{g}$, *belongs to* $G \times \mathfrak{g}$. *The order of approximation to* $y(\tilde{t})$ *is at least* $2k+1$.

Proof. The interpolation polynomial $p(t)$ belongs to $\mathfrak{g}$ for all t and $(p(t), \dot{p}(t)) \in \mathrm{T}\mathfrak{g}$. Since $\mathrm{T}\lambda_y : \mathrm{T}\mathfrak{g} \to G \times \mathfrak{g}$, the first part of the theorem trivially follows.

By classical theory, the order of the interpolation polynomial in $\mathfrak{g}$ is $2k+1$. The order of approximation is defined through expansion in Lie series around a point, and since λ is a smooth mapping, the order on $G \times \mathfrak{g}$ is at least as high as the order in $\mathrm{T}\mathfrak{g}$. □

The above procedure can roughly be explained as follows. Assume we are given points in a Lie group and tangent vectors in the tangent space at (some of) the points. Through a suitable mapping, $\mathrm{T}\phi$, we change coordinates and express the information in the linear space $\mathrm{T}\mathfrak{g}$. In this space we can apply classical interpolation techniques, and the function λ maps the result back to the Lie group. By requiring that $p(t_i) = u_i$ we make sure that the interpolation conditions in G are satisfied:

$$\lambda_{y(t_n)}\big(p(t_i)\big) = \exp(u_i) y_n = y_i.$$

Furthermore, by requiring that $\dot{p}(t_i) = v_i$ we interpolate the prescribed tangents at the respective points in G:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} y(t)\bigg|_{t=t_i} &= \frac{\mathrm{d}}{\mathrm{d}t} \lambda_{y(t_n)}\big(p(t)\big)\bigg|_{t=t_i} \\ &= \mathrm{dexp}_{p(t_i)}\big(\dot{p}(t_i)\big) \exp\big(p(t_i)\big) y_n \\ &= \mathrm{dexp}_{u_i}(v_i) \exp(u_i) y_n \\ &= \widetilde{y}_i y_i. \end{aligned}$$

Example 2.2. When solving the nonlinear problem $\dot{y}(t) = A(t, y(t))y(t)$, $y(0) = y_0$, with the Magnus series method developed in [13], we may employ waveform relaxation. This gives us the iteration scheme

$$\dot{y}^{[k+1]}(t) = A(t, y^{[k]}(t)) y^{[k+1]}(t), \quad k = 0, 1, 2, \ldots,$$

which can be integrated with the linear Magnus series method of order four, say. But we then need to approximate $y(t_n + c_i h)$, $i = 1, \ldots, s$, with sufficiently high order. Given $\big(y(t_n), \widetilde{y}(t_n)\big)$ and $\big(y(t_{n+1}), \widetilde{y}(t_{n+1})\big)$, we can construct a third order approximation to the solution at the quadrature points by the above described procedure. By computing the transformed quantities u_i and v_i, $i = n, n+1$, we get the interpolation points:

$$(t_n, 0), \qquad (t_{n+1}, u_{n+1}), \qquad (t_n, v_n), \qquad \text{and} \qquad (t_{n+1}, v_{n+1}).$$

The unique polynomial interpolating these elements in $\mathfrak{g}$ is

$$\begin{aligned} p(t) = (t - t_n) v_n + (t - t_n)^2 \frac{u_{n+1} - (t_{n+1} - t_n) v_n}{(t_{n+1} - t_n)^2} \\ + (t - t_n)^2 (t - t_{n+1}) \frac{(t_{n+1} - t_n)(v_n + v_{n+1}) - 2u_{n+1}}{(t_{n+1} - t_n)^3}. \end{aligned}$$

The cost of the above described Hermite interpolation procedure is approximately as follows. Let y_0 be a matrix of dimension $m \times m$. The transformation of k elements consists roughly of one inverse and k multiplications (however, the actual implementation may use $\sim m^3/3$ flops (LU-decomposition), $\sim m^3/2$ flops (m forward eliminations) plus $\sim m^3/2$ flops (m back substitutions) to accomplish this task), k logarithms ($\sim 25km^3$ flops (depending on the method used)), and k dexp^{-1}. The computation of $p(t)$ is relatively cheap, while applying the action λ requires one exponentiation ($\sim 25m^3$ flops (depending on the method used)), one multiplication, and possibly one dexp computation. Note that the functions dexp and dexp^{-1} are infinite sums. They need be evaluated only to the order of the interpolation polynomial, however. For order 4, this leads to roughly $8m^3$ flops for both dexp and dexp^{-1}.

It is well known that when exponentiating matrices in $\mathfrak{so}(3)$, the Euler–Rodrigues formula may be applied to reduce the cost of exponentiation. An even simpler formula can be deduced for the logarithm of matrices in SO(3). The topic of efficiently computing the matrix exponential when viewed as a mapping from a Lie algebra to a Lie group has recently been addressed by Celledoni and Iserles [1].

3. Generalized integration methods with continuous extensions. We next consider interpolants for some geometric integrators. All the integrators may use the Hermite interpolation procedure specified in section 2 in the process of producing dense output when integrating on matrix Lie groups. However, as for classical Runge–Kutta methods, we would also like to look for other (intrinsic) ways of estimating the required values. We will use continuous weights for the methods that have coefficients that are based on Butcher tableaus.

For completeness we include brief descriptions of the methods we consider. We adopt some of the notation from [22, 9] and we let $\mathcal{M}$ be a differentiable manifold and G be a Lie group with Lie algebra $\mathfrak{g}$. Let $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ be a (left) Lie-group action. A (left) Lie-algebra action $\lambda : \mathfrak{g} \times \mathcal{M} \to \mathcal{M}$ exists with $\lambda(v, p) = \Lambda(\exp(v), p)$ for all v in a neighborhood of 0. Here, $\exp : \mathfrak{g} \to G$ is the matrix exponential when G is a matrix group. Let the one-parameter family of Lie-algebra actions on a point $p \in \mathcal{M}$ be given as $\lambda_p(v) = \lambda(v, p)$. The Lie-algebra homomorphism $\lambda_* : \mathfrak{g} \to \mathfrak{X}(\mathcal{M})$, where $\mathfrak{X}(\mathcal{M})$ is the set of all vector fields on $\mathcal{M}$, is defined as

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

Marsden and Ratiu [18] define a Lie-algebra action of $\mathfrak{g}$ on $\mathcal{M}$ as a Lie-algebra (anti-)homomorphism from $\mathfrak{g}$ to $\mathfrak{X}(\mathcal{M})$. This corresponds to λ_*. In this paper, however, we use the term Lie-algebra action to denote λ, according to the definition in [22].

We assume that there exists a Lie algebra $\mathfrak{g}$ with a Lie bracket $[\cdot,\cdot]$, a (left) Lie-algebra action defined as above, and a function $f : \mathbb{R} \times \mathcal{M} \to \mathfrak{g}$ such that the ordinary differential equation for $y(t) \in \mathcal{M}$ can be written as

$$y' = \mathcal{F}_\lambda(t, y) = \big(\lambda_* f(t, y)\big)(y), \qquad y(0) = p. \tag{3.1}$$

This is the generic form of an ordinary differential equation on a manifold, and if we assume that $y_0 \in \mathcal{M}$, it follows that $y' \in \mathrm{T}\mathcal{M}_y$, where $\mathrm{T}\mathcal{M}_y$ is the tangent space of $\mathcal{M}$ at $y \in \mathcal{M}$. Munthe-Kaas [22] proved that the following diagram commutes:

$$\begin{tikzcd}
\mathrm{T}\mathfrak{g} \arrow[r, "\mathrm{T}\lambda"] & \mathrm{T}\mathcal{M} \\
\mathfrak{g} \arrow[u, "\tilde{f}"] \arrow[r, "\lambda"] & \mathcal{M} \arrow[u, "\mathcal{F}"']
\end{tikzcd}$$

The solution of (3.1) is given, for sufficiently small t, as $y(t) = \lambda(u(t), p)$, where $y(0) = p$ and $u(t) \in \mathfrak{g}$ satisfies the differential equation

$$u'(t) = \widetilde{f}(u(t)) = \operatorname{dexp}_{u(t)}^{-1}\big(f(t, \lambda(u(t), p))\big), \qquad u(0) = 0. \tag{3.2}$$

When u_1 and u_2, both as elements in $\mathfrak{g}$, are sufficiently small, then

$$\Lambda\big(\exp(u_1), \Lambda(\exp(u_2), p)\big) = \Lambda\big(\exp(u_1) \cdot \exp(u_2), p\big) = \Lambda\big(\exp(\mathcal{B}(u_1, u_2)), p\big),$$

and hence $\lambda(u_1, \lambda(u_2, p)) = \lambda(\mathcal{B}(u_1, u_2), p)$, where $\mathcal{B} : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ is the well-known Baker–Campbell–Hausdorff formula (see, e.g., [33, 23]).

3.1. Crouch–Grossman methods. The Crouch–Grossman methods are described in a number of texts; see, e.g., [2, 25, 19]. Much of the discussion in this section is based on notation and results from [25]. Letting $E_1, \ldots, E_n$ be smooth vector fields (a frame) on the manifold $\mathcal{M}$, we may write a system of differential equations relative to the frame as

$$\dot{y} = F(y) = \sum_{i=1}^{n} f_i(y) E_i, \qquad y(0) = y_0 \in \mathcal{M}, \tag{3.3}$$

where $f_i : \mathcal{M} \to \mathbb{R}$ are smooth functions. The solution of (3.3) is an integral curve $y : [0, T] \to \mathcal{M}$ of the vector field F.

The Crouch–Grossman methods make use of sampled versions of F: for any $p \in \mathcal{M}$ we associate a vector field $F_p = \sum_{i=1}^{n} f_i(p) E_i$, i.e., a vector field with coefficients frozen relative to the frame. Let $\mathfrak{g}$ be the Lie algebra generated by the frame $E_1, \ldots, E_n$ and let $G \in \operatorname{Diff}(\mathcal{M})$ be the collection of flows on $\mathcal{M}$ generated by exponentiation of $\mathfrak{g}$. Since

$$(\lambda_* F_p)(y) = \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \lambda(tF_p, y) = F_p(y),$$

it follows that (3.2) can be written in the form

$$\dot{y} = (\lambda_* F_y)(y), \qquad y(0) = p \in \mathcal{M}.$$

Let $\widetilde{\lambda}_v : \mathcal{M} \to \mathcal{M}$ be defined as $\widetilde{\lambda}_v(p) = \lambda(v, p)$. This is the v-flow through $p \in \mathcal{M}$. The Crouch–Grossman methods may now be written as follows:

$$Y_r = \widetilde{\lambda}_{ha_{r,s}F_{Y_s}} \circ \widetilde{\lambda}_{ha_{r,s-1}F_{Y_{s-1}}} \circ \cdots \circ \widetilde{\lambda}_{ha_{r,1}F_{Y_1}}(y_k), \quad r = 1, \ldots, s, \tag{3.4}$$

$$y_{k+1} = \widetilde{\lambda}_{hb_s F_{Y_s}} \circ \widetilde{\lambda}_{hb_{s-1}F_{Y_{s-1}}} \circ \cdots \circ \widetilde{\lambda}_{hb_1 F_{Y_1}}(y_k). \tag{3.5}$$

Note that these methods only assume that we are able to compute flows of vector fields in the linear span of $E_1, \ldots, E_n$.

As for classical, continuous Runge–Kutta methods (see, e.g., [11, 27, 28, 34, 35]) we consider interpolants defined by continuous weights, $b(\theta)$. The methods then take the form (3.4) together with

$$u(y_k; \theta) = \widetilde{\lambda}_{hb_s(\theta)F_{Y_s}} \circ \widetilde{\lambda}_{hb_{s-1}(\theta)F_{Y_{s-1}}} \circ \cdots \circ \widetilde{\lambda}_{hb_1(\theta)F_{Y_1}}(y_k). \tag{3.6}$$

It is often natural to require that $b_i(0) = 0$ and $b_i(1) = b_i$, $i = 1, \ldots, s$, where b_i, $i = 1, \ldots, s$, are the discrete weights, so that

$$u(p; 0) = y_k \qquad \text{and} \qquad u(p; 1) = y_{k+1}.$$

One step from $y_k = p$ with the continuous method may be written as $\Psi(\theta h, p)$, and the exact solution of the problem is given by $e^{\theta hF}p$. Therefore, the uniform order of accuracy to which the interpolant approximates the local solution may be defined as the greatest integer $q \geq 0$ for which

$$\max_{\theta\in[0,1]} \left|\left(\psi \circ e^{\theta hF}\right) - \left(\psi \circ \Psi(\theta h, p)\right)\right| = \mathcal{O}\left(h^{q+1}\right) \qquad \text{for all } \psi \in C^{\infty}(\mathcal{M}, \mathbb{R}).$$

The Lie series of $\psi \circ e^{\theta hF}p$ is given by

$$\psi(p) + \theta hF[\psi](p) + \tfrac{1}{2}\theta^2 h^2 F^2[\psi](p) + \cdots + \tfrac{1}{k!}\theta^k h^k F^k[\psi](p) + \cdots,$$

where $F^{r+1} = F\left[F^r[\psi]\right]$, $r = 1, 2, \ldots$. By identifying elementary differentials, $\mathbb{F}$, with ordered (rooted) trees (see [25]), we can write the Lie series of the exact solution as

$$\psi \circ e^{\theta hF}p = \sum_{t\in T_O} \frac{(\theta h)^{\rho(t)-1}}{(\rho(t)-1)!}\alpha(t)\mathbb{F}(t)[\psi](p).$$

It follows that

$$(3.7) \qquad \frac{\mathrm{d}^q}{\mathrm{d}h^q}\left(\psi \circ e^{\theta hF}p\right)\bigg|_{h=0} = \theta^q \sum_{t\in T_O^{q+1}} \alpha(t)\mathbb{F}_p(t)[\psi](p), \qquad q = 0, 1, \ldots,$$

where $\mathbb{F}_p$ is an elementary differential frozen at $p \in \mathcal{M}$.

On the other hand, the qth derivative of the numerical solution at time θh is given by

$$(3.8) \quad \frac{\mathrm{d}^q}{\mathrm{d}h^q}\psi \circ \Psi(\theta h, p)\bigg|_{h=0} = \sum_{t\in T_O^{q+1}} \alpha(t)\gamma(t)\Phi(t;\theta)\mathbb{F}_p(t)[\psi](p), \qquad q = 1, 2, \ldots,$$

where

$$\Phi(t;\theta) = \sum_{j_1=1}^{s}\sum_{j_2=j_1}^{s}\cdots\sum_{j_\mu=j_{\mu-1}}^{s} \frac{1}{j!}b_{j_1}(\theta)\cdots b_{j_\mu}(\theta)\Phi_{j_1}(t_1)\cdots\Phi_{j_\mu}(t_\mu),$$

with $j!$ and $\Phi_k(t)$ defined as in section 4 of [25]. By equating (3.7) and (3.8) and requiring this condition to hold for each tree of order less than $p+2$, the next theorem follows.

THEOREM 3.1 (continuous order conditions). *The approximation given by* (3.4) *and* (3.6) *is of uniform order p if and only if*

$$\gamma(t)\Phi(t;\theta) = \theta^{\rho(t)-1} \qquad \textit{for all } t \in T_O^2 \cup T_O^3 \cup \cdots \cup T_O^{p+1}.$$

Example 3.2. Consider the following Crouch–Grossman method of order three with three stages (left tableau) (see [25]):

$$\begin{array}{c|ccc}
0 & & & \\
\frac{3}{4} & \frac{3}{4} & & \\
\frac{17}{24} & \frac{119}{216} & \frac{17}{108} & \\
\hline
 & \frac{13}{51} & -\frac{2}{3} & \frac{24}{17}
\end{array}
\qquad\qquad
\begin{array}{c|ccccc}
0 & & & & & \\
\frac{3}{4} & \frac{3}{4} & & & & \\
\frac{17}{24} & \frac{119}{216} & \frac{17}{108} & & & \\
\frac{3}{4} & -\frac{1}{4} & 0 & 1 & & \\
1 & -\frac{581}{108} & 0 & 0 & \frac{689}{108} & \\
\hline
 & b_1(\theta) & b_2(\theta) & b_3(\theta) & b_4(\theta) & b_5(\theta)
\end{array}$$

We can equip it with an interpolant by adding two stages, yielding, e.g., the right tableau, where

$$\begin{aligned}
b_1(\theta) &= \tfrac{9076}{14739}\theta^3 - \tfrac{6686}{4913}\theta^2 + \theta + \tfrac{1}{14739}\gamma(\theta),\\
b_2(\theta) &= \tfrac{2442}{289}\theta^3 - \tfrac{7904}{867}\theta^2 + \tfrac{1}{1734}\gamma(\theta),\\
b_3(\theta) &= -\tfrac{363648}{34391}\theta^3 + \tfrac{41220}{34391}\theta^2 - \tfrac{144}{34391}\gamma(\theta),\\
b_4(\theta) &= \tfrac{1234}{867}(\theta^3 - \theta^2) + \tfrac{7}{1734}\gamma(\theta),\\
b_5(\theta) &= \tfrac{172}{2023}(\theta^3 - \theta^2) - \tfrac{1}{2023}\gamma(\theta)
\end{aligned}$$

with

$$\gamma(\theta) = \theta\sqrt{\theta\,(5337936\,\theta^3 - 11339416\,\theta^2 + 5416833\,\theta + 58467)}.$$

This interpolant satisfies $b_i(1) = b_i$, $i = 1, \ldots, 3$, and $b_4(1) = b_5(1) = 0$. In addition, $b_i(0) = 0$, $i = 1, \ldots, 5$.

The cost involved in advancing the numerical solution from t_n to $t_n + h$ with an s-stage Crouch–Grossman method is roughly s function evaluations and $\frac{1}{2}s(s+1)$ flow computations (exponentiation and group multiplication). The cost of approximating the solution on the interval $[t_n, t_n + h]$ with the continuous extension is roughly given by s flow computations.

3.2. Munthe-Kaas Methods. The Munthe-Kaas methods were developed in the trilogy [20, 21, 22]. While Crouch–Grossman methods consist of compositions of flows of vector fields on the manifold, Munthe-Kaas methods compose pulled back vector fields before computing the resulting action on the manifold.

The basic idea is to integrate (3.2) using a classical Runge–Kutta method. Since the action λ is assumed to be a smooth mapping, the order of approximation to the solution of (3.1) is at least the same as the classical order of the Runge–Kutta method. Let us consider the methods in more detail. For simplicity we assume that the manifold $\mathcal{M}$ is a homogeneous manifold given by a matrix Lie group G acting on itself by left multiplication. Let the truncated $\mathrm{dexp}_u^{-1}(v)$ function be defined by

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

where $[\cdot,\cdot]$ is the matrix commutator[1] defined by $[A, B] = AB - BA$, and B_k is the kth Bernoulli number. Note that when q is an even number, the sum runs from 2 to $q - 2$, since the odd Bernoulli numbers are zero. The Munthe-Kaas methods are then defined by the following scheme:

```
Assume that $y_n \approx y(t_n)$ is available
for $i = 1, 2, \ldots, s$
    $u_i = h \sum_{j=1}^{s} a_{ij} \tilde{k}_j$
    $k_i = f\,(t_n + c_i h, \lambda(u_i, y_n))$
    $\tilde{k}_i = \mathrm{dexpinv}(u_i, k_i, q)$
end
$v = h \sum_{j=1}^{s} b_j \tilde{k}_j$
$y_{n+1} = \lambda(v, y_n)$.
```

[1] The method is defined for more general groups than matrix groups, but, for simplicity, we restrict ourselves to matrix groups in this exposition.

The coefficients are given by the qth order classical Runge–Kutta method's Butcher tableau [11].

It is obvious that Munthe-Kaas methods, which are directly based on classical Runge–Kutta methods, can use the continuous weights, $b_i(\theta)$, $i = 1, \dots, s$, of the underlying continuous Runge–Kutta methods. The solution at an intermediate point $t_n + \theta h$, $\theta \in (0, 1)$, is then given by

$$\begin{aligned} v(\theta) &= h \sum_{j=1}^{s} b_j(\theta) \tilde{k}_j, \\ y_{n+1}(\theta) &= \lambda\big(v(\theta), y_n\big). \end{aligned}$$

The work needed to find the solution at one intermediate point is therefore given by a linear combination in the Lie algebra and a Lie-algebra action (which is matrix exponentiation and multiplication(s) in the case of matrix Lie groups). A new step with the Munthe-Kaas methods, however, costs roughly s function evaluations, s Lie-algebra actions, and s dexp^{-1}.

Example 3.3. Consider the classical Runge–Kutta method of order four with four stages (left tableau):

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{2}{3}\theta^3 - \frac{3}{2}\theta^2 + \theta & -\frac{2}{3}\theta^3 + \theta^2 & -\frac{2}{3}\theta^3 + \theta^2 & \frac{2}{3}\theta^3 - \frac{1}{2}\theta^2 \end{array}$$

We can equip it with a classical interpolant of order three without adding any stages, yielding the right tableau. This interpolant satisfies $b_i(0) = 0$ and $b_i(1) = b_i$, $i = 1, \dots, 4$.

3.3. Magnus series and Fer expansion methods. The Magnus series and Fer expansion methods were originally developed to analyze and solve matrix differential equations of the form

$$y' = f(t)y, \qquad y(0) = y_0 \in \mathrm{GL}(n), \tag{3.9}$$

with $f : \mathbb{R} \to \mathbb{R}^{n \times n}$. Magnus [17] assumed a solution of the form $y(t) = \exp\big(\sigma(t)\big)y_0$ and derived a differential equation to be satisfied by σ. Fer [10] assumed a solution in the form of compositions of exponentials. In [13], Iserles and Nørsett analyzed problems of the form (3.9) along the same line as Magnus, and they devised very efficient numerical methods based on analysis involving certain rooted trees and multivariate quadrature. The analysis has been extended in [12, 14].

Zanna [37] proposed a way of extending these methods to numerically integrate general initial value problems of the form $y' = f(t, y)y$ in terms of collocation and relaxed collocation. Let $L_i(t)$ be the ith Lagrangian cardinal polynomial defined on a set of s abscissae values, $c_1, \dots, c_s \in [0, 1]$:

$$L_i(t) = \prod_{\substack{k=1 \\ k \neq i}}^{s} \frac{t - c_k}{c_i - c_k}.$$

By letting $f_i \approx f(t_n+c_ih, y(t_n+c_ih))$, $i = 1, \dots, s$, we can construct an approximation to $f(t, y)$ on the interval $[t_n, t_n + h]$ through its Lagrangian interpolating polynomial:

$$f(t, y(t)) \approx p_L(t) = \sum_{i=1}^{s} L_i\Big(\frac{t-t_n}{h}\Big) f_i. \tag{3.10}$$

Instead of letting $f : \mathbb{R} \to \mathbb{R}^{n\times n}$, Zanna considered the general case where $f : \mathbb{R} \times G \to \mathfrak{g}$ and therefore the interpolation polynomial belongs to $\mathfrak{g}$. Note that this is interpolation in a linear space and is perfectly legal for that reason. It does, however, not directly give approximations to the solution in the time-interval of interest, as the interpolants described in this paper do. By inserting the approximation $p_L(t)$ into the Magnus series or the Fer expansion, numerical schemes of collocation type can be derived.

Having completed a step, the pairs $(y_i, f_i) \approx \big(y(t_n+c_ih), f(t_n+c_ih, y(t_n+c_ih))\big)$, $i = 1, \dots, s$, are available. We can now approximate the solution in the interval $[t_n, t_n + h]$ as follows. Let

$$\begin{aligned} \sigma^n(\theta; h) &= \int_0^\theta h\sum_{i=1}^{s} L_i(\kappa) f_i \,\mathrm{d}\kappa + \frac{1}{2}\int_0^\theta \Big[h\sum_{i=1}^{s} L_i(\kappa) f_i, \int_0^\kappa h\sum_{j=1}^{s} L_j(\xi) f_j \,\mathrm{d}\xi\Big] \mathrm{d}\kappa + \cdots \\ &= h\sum_{i=1}^{s} f_i \int_0^\theta L_i(\kappa)\,\mathrm{d}\kappa + \frac{h^2}{2}\sum_{i=1}^{s}\sum_{j=1}^{s} [f_i, f_j] \int_0^\theta L_i(\kappa) \int_0^\kappa L_j(\xi)\,\mathrm{d}\xi\,\mathrm{d}\kappa + \cdots \end{aligned}$$

be the Magnus expansion (see, e.g., [12, 13]) inserted into the interpolant (3.10) for f. By truncating the series, as described in [13], the solution at time $t_n + \theta h$ is given as

$$y(t_n + \theta h) \approx \lambda\big(\sigma^n(\theta; h), y_n\big).$$

Example 3.4. Consider the fourth order collocation method described in [37, Example 4.2]. The method is based on the Gauss–Legendre points $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$ and $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$:

$$\begin{aligned} \sigma_1^n &= h\big(\tfrac{1}{4} f_1 + (\tfrac{1}{4} - \tfrac{\sqrt{3}}{6}) f_2\big) + \tfrac{h^2}{2}\big(\tfrac{5-3\sqrt{3}}{72}\big)[f_1, f_2], \\ Y_1 &= \lambda\big(\sigma_1^n, y_n\big), \\ \sigma_2^n &= h\big((\tfrac{1}{4} + \tfrac{\sqrt{3}}{6}) f_1 + \tfrac{1}{4} f_2\big) - \tfrac{h^2}{2}\big(\tfrac{5+3\sqrt{3}}{72}\big)[f_1, f_2], \\ Y_2 &= \lambda\big(\sigma_2^n, y_n\big) \end{aligned}$$

with $f_i = f(t_n + c_ih, Y_i)$, $i = 1, 2$. The step is advanced with

$$\begin{aligned} \sigma^n &= \tfrac{h}{2}(f_1 + f_2) - \tfrac{\sqrt{3}}{12} h^2 [f_1, f_2], \\ y_{n+1} &= \lambda\big(\sigma^n, y_n\big). \end{aligned}$$

Having completed the step, the values $f_i \approx f\big(t_n + c_ih, y(t_n + c_ih)\big)$, $i = 1, 2$, are available. The following interpolant is based on these values and yields an order two approximation to the solution in the interval $(t_n, t_n + h)$:

$$y(t_n + \theta h) \approx \lambda\big(\sigma^n(\theta; h), y_n\big), \qquad \theta \in [0, 1],$$

with

$$\begin{aligned}\sigma_{11}(\theta) &= -\tfrac{\sqrt{3}}{2}\theta^2 + \left(\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}\right)\theta,\\ \sigma_{12}(\theta) &= \tfrac{\sqrt{3}}{2}\theta^2 + \left(\tfrac{1}{2} - \tfrac{\sqrt{3}}{2}\right)\theta,\\ \sigma_{21}(\theta) &= -\tfrac{3}{8}\theta^4 + \left(\tfrac{3}{4} - \tfrac{\sqrt{3}}{12}\right)\theta^3 - \tfrac{1}{4}\theta^2,\\ \sigma_{22}(\theta) &= -\tfrac{3}{8}\theta^4 + \left(\tfrac{3}{4} + \tfrac{\sqrt{3}}{12}\right)\theta^3 - \tfrac{1}{4}\theta^2,\end{aligned}$$

$$\begin{aligned}\sigma^n(\theta;h) &= h\big(\sigma_{11}(\theta)f_1 + \sigma_{12}(\theta)f_2\big) + \tfrac{h^2}{2}\big(\sigma_{21}(\theta) - \sigma_{22}(\theta)\big)[f_1,f_2],\\ &= h\big(\sigma_{11}(\theta)f_1 + \sigma_{12}(\theta)f_2\big) - h^2\tfrac{\sqrt{3}}{12}\theta^3[f_1,f_2].\end{aligned}$$

Note that the interpolant coincides with the discrete integrator at $\theta = 1$, i.e.,

$$\lambda\big(\sigma^n(1;h), y_n\big) = y_{n+1},$$

and the approximation is hence of order four there. At the midpoint of the interval, $\theta = \frac{1}{2}$, the approximation is of order three. To see this, we need a result from [13] (see also [37]).

LEMMA 3.5 (Iserles and Nørsett). *Let $c_1,\dots,c_m$ be nodes of quadrature of order p. Provided that $\mathcal{L}$ is an s-linear form, the quadrature formula*

$$\int_0^h\int_0^{\tau_1}\cdots\int_0^{\tau_{s-1}} \mathcal{L}\big(f(\xi_1),\dots,f(\xi_s)\big)\,\mathrm{d}\xi_s\cdots\mathrm{d}\xi_1 \approx h^s\sum_{i_1,\dots,i_s=1}^{m} b_{i_1,\dots,i_s}\mathcal{L}\big(f(c_{i_1}h),\dots,f(c_{i_s}h)\big),$$

where $\tau_k \in \{\xi_1,\dots,\xi_k\}$, $k = 1,\dots,s-1$, with weights

$$b_{i_1,\dots,i_s} = \int_0^h\int_0^{\tau_1}\cdots\int_0^{\tau_{s-1}} L_{i_1}(\xi_1)\cdots L_{i_s}(\xi_s)\,\mathrm{d}\xi_s\cdots\mathrm{d}\xi_1,$$

is also of order p.

We proceed by first analyzing the quadrature order q of the scheme

$$\int_0^{\theta h} P(\tau)\,\mathrm{d}\tau = h\big(\sigma_{11}(\theta)P(c_1h) + \sigma_{12}(\theta)P(c_2h)\big) + \mathcal{O}(h^{q+1}),$$

as a function of $\theta \in (0,1]$, when P is any polynomial of degree at least $q+1$. The quadrature is at least $\mathcal{O}(h^3)$ in the interval. The order three and four conditions are

$$\frac{1}{3}\theta^3 - \frac{1}{2}\theta^2 + \frac{1}{6}\theta = 0, \tag{3.11}$$

$$\frac{1}{4}\theta^4 - \frac{5}{12}\theta^2 + \frac{1}{6}\theta = 0. \tag{3.12}$$

The roots of (3.11) are $\theta \in \left\{\frac{1}{2}, 1\right\}$ while the only solution of (3.12) in the interval of interest is $\theta = 1$.

We conclude by applying [37, Lemma 4.2].

An alternative way of solving the generalized problem $y' = f(t,y)y$ using collocation is as follows. The idea is similar to the classical collocation idea [11, 24], and the resulting methods are just the classical collocation methods applied in the setting of Munthe-Kaas (see section 3.2).

Choose distinct abscissae values $c_1, \dots, c_s \in [0,1]$ and construct a polynomial $p(t) \in \Pi_{s-1}$ that obeys (3.2) at the collocation points $c_1 h, \dots, c_s h$:

$$\begin{aligned} p(0) &= 0, \\ p'(c_i h) &= \operatorname{dexp}^{-1}_{p(c_i h)}\Big(f\big(t_n + c_i h, \lambda(p(c_i h), y_n)\big)\Big), \qquad i = 1, \dots, s, \\ y_{n+1} &= \lambda\big(p(h), y_n\big). \end{aligned}$$

The derivation of the collocation schemes follows as for classical collocation. We have that

$$u(t_n + h) = \int_{t_n}^{t_n+h} \operatorname{dexp}^{-1}_{u(t)}\Big(f\big(t, \lambda(u(t), y_n)\big)\Big)\, \mathrm{d}t, \tag{3.13}$$

since $u(t_n) = 0$. By first assuming that approximations $\widetilde{f}_i = \widetilde{f}\big(u(t_n + c_i h)\big)$, $i = 1, \dots, s$, are available, we can construct a polynomial that interpolates $\widetilde{f}$ at the abscissae values:

$$\widetilde{f}\big(u(t)\big) \approx \sum_{i=1}^{s} L_i\Big(\frac{t - t_n}{h}\Big)\widetilde{f}_i.$$

Insertion of this approximation into (3.13) gives, as in the classical case, the method weights $b_i = \int_0^1 L_i(\tau)\,\mathrm{d}\tau$, $i = 1, \dots, s$. The stage values $Y_i \approx u(t_n + c_i h)$ are computed as

$$Y_i = \int_{t_n}^{t_n + c_i h} \sum_{j=1}^{s} L_j\Big(\frac{t - t_n}{h}\Big)\widetilde{f}(Y_j)\,\mathrm{d}t = h \sum_{j=1}^{s} a_{ij}\widetilde{f}(Y_j), \qquad i = 1, \dots, s,$$

with $a_{ij} = \int_0^{c_i} L_j(\tau)\,\mathrm{d}\tau$, $i, j = 1, \dots, s$. As opposed to the collocation schemes by Zanna [37], we do not need any higher-order coefficients or weights.

3.4. Numerical results. As a test problem we consider an example by Zanna [36]. Let $\mathcal{M} = G$ be a matrix Lie group with Lie algebra $\mathfrak{g}$. The action of the Lie algebra on G is given by $\lambda : \mathfrak{g} \times G \to G$, where $\lambda(v, p) = \exp(v) \cdot p$. Now (3.1) reduces to

$$y' = f(y) \cdot y, \qquad \text{where} \qquad f : G \to \mathfrak{g} \qquad \text{and} \qquad y(0) \in G. \tag{3.14}$$

We consider the particular case of $G = \mathrm{SO}(5)$ with the right hand side of (3.14) defined as follows (in MATLAB notation):

```
f(y)=diag(diag(y,1),1)-diag(diag(y,1),-1);
```

and we solve this problem with initial values (a random, orthogonal, 5 × 5 matrix with determinant +1)

```
rand('seed',0); [y0,r]=qr(rand(5)).
```

We integrate from $t_0 = 0$ to $t_{\text{end}} = 3$ with constant stepsize $h = \frac{1}{5}$.

Figures 3.1– 3.3 show some results from simulations with the continuous methods described in Examples 3.2–3.4.

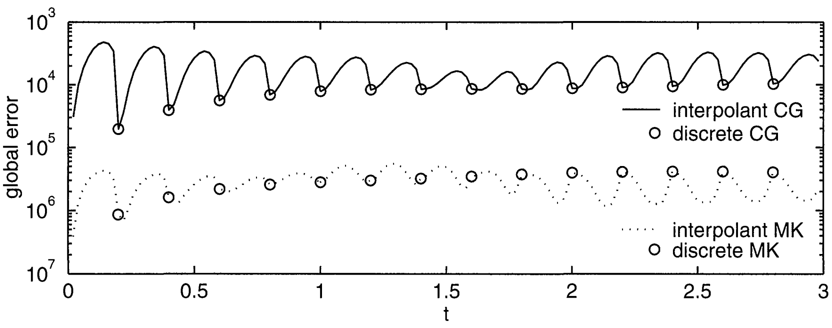

FIG. 3.1. *Global error committed by the interpolants (Crouch–Grossman and Munthe-Kaas). The circles denote the approximations of the underlying discrete scheme.*

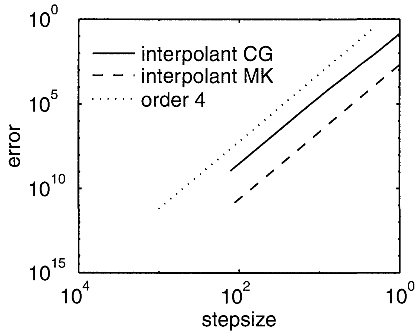

FIG. 3.2. *Local error committed by the interpolants as a function of the stepsize (Crouch–Grossman and Munthe-Kaas). The local order of approximation is given by the slope of the graphs.*

As a second test problem we consider the spinning top. The Crouch–Grossman and Munthe-Kaas methods have already been successfully applied to this problem, but in order to verify the properties of the interpolants, we apply them also to this problem. A complete description of the spinning top problem phrased in the setting necessary for the numerical integrators described in this paper can be found in [8]. However, for completeness we also include a brief description here.

We model the spinning top on the Lie group $G = \mathrm{SO}(3) \times \mathfrak{so}(3)$, and an element in G is hence a pair (B, ω). Both B and ω are time dependent. $B(t)$ describes a rotation and the tangent to $B(t)$ is $\dot{B}(t) \in \mathrm{T}_{B(t)}\mathrm{SO}(3)$, which can be written in the form $\dot{B}(t) = \omega(t) \cdot B(t)$, where $\omega \in \mathfrak{so}(3)$. The Lie algebra of $\mathfrak{g}$ can be identified with $\mathfrak{so}(3) \times \mathfrak{so}(3)$. The differential equation describing the motion of the top is essentially given by the Euler equations. Let $f : G \to \mathfrak{g}$ be the mapping $(B, \omega) \mapsto (\omega, \omega')$. It is shown in [8] that

$$\omega'(t) = \big[\omega(t), \mathsf{I}^{-1}(t)\big]\mathsf{L}(t) + \mathsf{I}^{-1}(t)\mathsf{M},$$

where $\mathsf{I}^{-1}(t) = B(t)\mathsf{I}_0^{-1}B^{-1}(t)$ and $\mathsf{M} = m\mathsf{c} \times \mathsf{f}$. Let subscript 0 refer to quantities in the initial configuration of the top. Then, I_0 is the inertia tensor, c is the centroid vector, the angular momentum $\mathsf{L}(t) = B(t)\mathsf{L}_0 = B(t)\mathsf{I}_0\Omega$ with $\omega(t) = B(t)\Omega$, m is the mass, and f is the vector of gravity. The differential equation describing this system

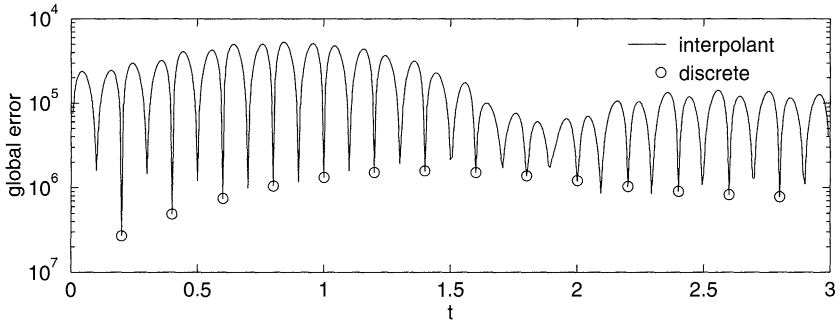

FIG. 3.3. *Global error committed by the interpolant to the collocation method. The circles denote the approximations of the underlying discrete collocation scheme.*

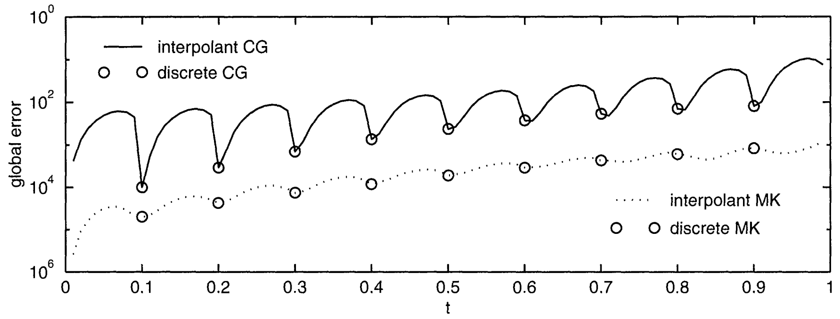

FIG. 3.4. *Global error committed by the interpolants when integrating the spinning top problem. The circles denote the approximations of the underlying discrete scheme.*

can now be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\big(B(t),\omega(t)\big) = \mathrm{T}_e\mathrm{R}_{(B(t),\omega(t))}\big(f\big(B(t),\omega(t)\big)\big).$$

As initial values for the simulations we use

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

with $\phi = \pi/16$. The constants c, f, and I_0 have been taken as

$$\mathsf{c} = \begin{bmatrix} 0 \\ 0 \\ \sqrt{3}/2 \end{bmatrix}, \quad \mathsf{f} = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix}, \quad \text{and} \quad \mathsf{I}_0 = \frac{1}{8}\begin{bmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

Figure 3.4 shows some of the results from the simulations with the continuous methods. Again, the continuous order of approximation of the interpolants are as claimed. The interpolant from Example 3.2 is of order three and the interpolant from Example 3.3 is of order four.

4. Interpolation in the setting of homogeneous manifolds. Many geometric integrators have been formulated so that they respect homogeneous manifolds.

It is therefore of interest to perform Hermite interpolation also on such spaces. However, the procedure introduced in section 2 is not directly applicable. Consider, e.g., SO(3) acting on S^2 (the unit sphere in $\mathbb{R}^3$). Letting $p_1 = (x_1, x_2, x_3)^T, p_2 \in S^2$, there is an infinite number of matrices $v \in \mathfrak{so}(3)$ such that $\lambda(v, p_1) = p_2$. The matrix

$$\widehat{v} = \begin{bmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{bmatrix} \in \mathfrak{so}(3)$$

satisfies $\exp(\widehat{v})p_1 = p_1$; in fact, all matrices $s\widehat{v}$, $s \in \mathbb{R}$, satisfy the same relation. Therefore, all matrices $v, w \in \mathfrak{so}(3)$ related through $v = \mathcal{B}(w, s\widehat{v})$, $s \in \mathbb{R}$, will satisfy $\exp(v)p_1 = \exp(w)p_1$. This fact can make it unclear what elements to use in the interpolation process. Numerical simulations indicate that interpolation results highly depend on choice of elements $v_j \in \mathfrak{g}$ such that $\lambda(v_j, p_1) = p_j \in \mathcal{M}$, $j = 1, \dots, k+1$. If we let Q be a quadratic form on $\mathfrak{g}$, one choice would be to use, for each j, the element $u \in \mathfrak{g}$ that minimizes $u^{\mathrm{T}}Qu$ among all u satisfying $\lambda(u)p_1 = p_j$ [29]. Crouch, Kun, and Leite [3, 4, 5] have constructed splines on Lie groups and spheres, and they have addressed the above problem from a numerical point of view.

We will now briefly describe an alternative way of solving this problem. Assume that $\omega_i \in \mathfrak{g}$ defined by $\lambda\big(\omega_i, y_i\big) = y_{i+1}$, $i = n, \dots, n+k$, are available. Given these values, we avoid having to invert the action λ. Such elements are often available from geometric integrators; the Munthe-Kaas method, e.g., generates $\omega = v$ in each step (see section 3.2). The situation is now similar to what we have analyzed in the context of multistep methods on manifolds [9]. We have to express $\omega_n, \dots, \omega_{n+k}$ and the vector field evaluated at $y_n, \dots, y_{n+k}$ in a common coordinate system around one of the points. By using the algorithm from [9], this can be accomplished in terms of $\mathcal{B}$ and dexp^{-1} only. Let, for each $i = n, \dots, n+k$, u_i be ω_i expressed in a coordinate system around ω_n, say, and let v_i be the corresponding transformed samples of the vector field $\big(v_i = \mathrm{dexp}_{u_i}^{-1}(f_i)\big)$. A Hermite interpolation polynomial, $p(t) \in \mathfrak{g}$, can now be computed based on the $2k$ elements

$$(t_n, 0), \quad (t_i, u_i),\ i = n+1, \dots, n+k,$$

and

$$(t_n, f_n), \quad (t_i, v_i),\ i = n+1, \dots, n+k.$$

The polynomial can be computed as described in section 2, and approximation of elements in the time interval $[t_n, t_{n+k}]$ is given by (2.2). Theorem 2.1 still applies. Again, $\mathcal{B}$ and dexp^{-1} can be truncated to the order of the interpolation polynomial.

Note that we could have used any of the available points as the base point and expressed the other points in a coordinate system around this point. To ensure convergence of the exponential mapping, it might be better to choose a point near the middle of the interpolation interval.

5. Concluding remarks. We have described how to perform Hermite interpolation in Lie groups. When integrating in the setting of homogeneous spaces, the procedure changes slightly. Instead of assuming properties of the Lie-algebra action, the interpolation data is provided through use of the truncated Baker–Campbell–Hausdorff and dexp^{-1} formulas. When the interpolation points are produced by discrete integration of ordinary differential equations on Lie groups or homogeneous spaces by methods in the class of Munthe-Kaas or Crouch–Grossman, the integration methods can be equipped with continuous weights. We have also showed how to construct continuous extensions to the Magnus series methods.

Acknowledgments. The author expresses his gratitude to the referees for several valuable criticisms and suggestions. I would also like to thank Hans Munthe-Kaas and Brynjulf Owren for many clarifying discussions and valuable input during the work with this paper. I am grateful to Zdzislaw Jackiewicz and Bruno D. Welfert for their hospitality during my stay at Arizona State University, spring 1998.

REFERENCES

[1] E. Celledoni and A. Iserles, *Approximating the exponential from a Lie algebra to a Lie group*, Math. Comp., to appear.

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

[3] P. E. Crouch, G. Kun, and F. Silva Leite, *The De Casteljau algorithm on Lie groups and spheres*, to appear.

[4] P. E. Crouch, G. Kun, and F. Silva Leite, *Generalization of spline curves on the sphere: A numerical comparison*, to appear.

[5] P. E. Crouch, G. Kun, and F. Silva Leite, *De Casteljau algorithm for cubic polynomials on the rotation group*, in Proceedings of CONTROLO-96, Vol. II, 1996, pp. 547–552.

[6] L. Dieci, R. D. Russell, and E. S. van Vleck, *Unitary integrators and applications to continuous orthonormalization techniques*, SIAM J. Numer. Anal., 31 (1994), pp. 261–281.

[7] E. Eich, C. Führer, B. J. Leimkuhler, and S. Reich, *Stabilization and Projection Methods for Multibody Dynamics*, 1991.

[8] K. Engø and A. Marthinsen, *Modeling and solution of some mechanical problems on Lie groups*, Multibody System Dynamics, 2 (1998), pp. 71–88.

[9] S. Faltinsen, A. Marthinsen, and H. Z. Munthe-Kaas, *Multistep Methods Integrating Ordinary Differential Equations on Manifolds*, Technical Report Numerics 3/1999, The Norwegian University of Science and Technology, Trondheim, Norway, 1999.

[10] F. Fer, *Résolution del l'equation matricielle $\dot{U} = pU$ par produit infini d'exponentielles matricielles*, Bull. Classe des Sci. Acad. Royal Belg., 44 (1958), pp. 818–829.

[11] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I, *Nonstiff Problems.* 2nd ed., Springer-Verlag, Berlin, 1993.

[12] A. Iserles, A. Marthinsen, and S. P. Nørsett, *On the implementation of the method of Magnus series for linear differential equations*, BIT, 39 (1999), pp. 281–304.

[13] A. Iserles and S. P. Nørsett, *On the solution of linear differential equations in Lie groups*, Philos. Trans. Roy. Soc. London Ser. A, 357 (1999), pp. 983–1019.

[14] A. Iserles, S. P. Nørsett, and A. F. Rasmussen, *Reversibility and High-order Magnus Methods*, Technical Report 1998/NA06, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1998.

[15] B. J. Leimkuhler and G. W. Patrick, *A symplectic integrator for Riemannian manifolds*, J. Nonlinear Sci., 6 (1996), pp. 367–384.

[16] G. G. Lorentz, K. Jetter, and S. D. Riemenschneider, *Birkhoff Interpolation*, Addison–Wesley, Reading, MA, 1983.

[17] W. Magnus, *On the exponential solution of differential equations for a linear operator*, Comm. Pure Appl. Math., 7 (1954), pp. 649–673.

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

[19] A. Marthinsen, H. Munthe-Kaas, and B. Owren, *Simulation of ordinary differential equations on manifolds—some numerical experiments and verifications*, Modeling, Identification and Control, 18 (1997), pp. 75–88.

[20] H. Munthe-Kaas, *Lie–Butcher theory for Runge–Kutta methods*, BIT, 35 (1995), pp. 572–587.

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

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

[23] H. Munthe-Kaas and B. Owren, *Computations in a free Lie algebra*, Philos. Trans. Roy. Soc. London Ser. A, 357 (1999), pp. 957–981.

[24] S. P. Nørsett and G. Wanner, *Perturbed collocation and Runge–Kutta methods*, Numer. Math., 38 (1981), pp. 193–208.

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

[26] B. Owren and B. Welfert, *The Newton Iteration on Lie Groups*, Technical Report Numerics 3/1996, Norwegian University of Science and Technology, Trondheim, Norway, 1996.

[27] B. Owren and M. Zennaro, *Order barriers for continuous explicit Runge–Kutta methods*, Math. Comp., 56 (1991), pp. 645–661.

[28] B. Owren and M. Zennaro, *Derivation of efficient continuous explicit Runge–Kutta methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1488–1501.

[29] F. C. Park and B. Ravani, *Bézier curves on Riemannian manifolds and Lie groups with kinematic applications*, ASME J. of Mech. Design, 117 (1995), pp. 36–40.

[30] M. J. D. Powell, *Approximation Theory and Methods*, Cambridge University Press, Cambridge, UK, 1991.

[31] J. M. Sanz-Serna, *Geometric integration*, in The State of the Art in Numerical Analysis, Inst. Math. Appl. Conf. Ser. 63, Oxford Univ. Press, New York, 1997.

[32] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems*, Appl Math. Math. Comput. 7, Chapman & Hall, London, 1994.

[33] V. S. Varadarajan, *Lie Groups, Lie Algebras, and Their Representations*, Grad. Texts in Math. 102, Springer-Verlag, New York, Berlin, 1984.

[34] J. H. Verner, *Differentiable interpolants for high-order Runge–Kutta methods*, SIAM J. Numer. Anal., 30 (1993), pp. 1446–1466.

[35] J. H. Verner and M. Zennaro, *Orders of continuous explicit Runge–Kutta methods*, BIT, 35 (1995), pp. 406–416.

[36] A. Zanna, *The Method of Iterated Commutators for Ordinary Differential Equations on Lie Groups*, Technical Report 1996/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1996.

[37] A. Zanna, *Collocation and relaxed collocation for the Fer and the Magnus expansions*, SIAM J. Numer. Anal., 36 (1999), pp. 1145–1182.