ARIZONA STATE UNIVERSITY

Interpolation in Lie Groups and Homogeneous Spaces

by

Arne Marthinsen

Technical Report No. 139

May 1998

Department of Mathematics

Tempe, Arizona 85287-1804, USA

Interpolation in Lie Groups and Homogeneous Spaces*

Arne Marthinsen†

May 12, 1998

Abstract

We consider interpolation in Lie groups and homogeneous spaces. 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 on a Lie group or a homogeneous space, 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 a differential equation in such a way that the configuration space 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.

AMS Subject Classification: 65L06, 34A50

Key Words: geometric integration, numerical integration of ordinary differential equations on manifolds, interpolation, approximation, numerical analysis, Lie groups, homogeneous spaces

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 [3], Munthe-Kaas [14], Leimkuhler and Patrick [10], Dieci et al. [4], Eich et al. [5], Iserles and Nørsett [9], Zanna [26] and Sanz-Serna with coworkers [22, 21]. Some of these are implicit methods, and the nonlinear equations may be solved with the Newton iteration schemes proposed by Owren and Welfert [18].

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,

*This work was in part sponsored by British Petroleum under grant no. B94060.00 and by The Norwegian Research Council under contract no. 123655/410 as well as no. 111038/410, through the SYNODE project. WWW: `http://www.math.ntnu.no/num/synode/`

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

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 numerical methods are directly applicable only to linear problems (e.g. the Magnus series methods [9]), but they may be extended to the nonlinear case by means of dynamic iteration or as in [27]. This again introduces the need for an approximation to the solution at intermediate points in the integration interval.

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 [27] developed collocation and relaxed collocation methods for the Fer and the Magnus expansions. While Zanna's motivation was to construct new integration methods, the motivation of this of this paper is to understand how to construct interpolants in the generalized setting of Lie groups and homogeneous manifolds. 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 for integration of differential equations on manifolds with continuous extensions. Finally, in Section 4 we construct an interpolation procedure to be used with discrete integrators in the setting of homogeneous manifolds.

2 Hermite Interpolation in Lie Groups

In this section we construct an interpolation polynomial based on 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 it is not a requirement; we do not use information about vector fields on G or integration schemes, we use only points in G and tangent vectors at given points.

The setting is given by the following commutative diagram, where 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. R' is the tangent mapping of R.

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

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

$$\begin{aligned}
\mathrm{T}\lambda_y(u,v) &= \mathrm{R}'_y \circ \left(\mathrm{R}'_{\exp(u)} \circ \mathrm{dexp}_u(v)\right) \\
&= \mathrm{R}'_{\exp(u)\cdot y} \circ \mathrm{dexp}_u(v) \\
&= \mathrm{R}'_z \circ \mathrm{dexp}_u(v).
\end{aligned}$$

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

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

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

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

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

Assume now that the we have available $k+1$ points, $\left(t_i, y_i, \dot{y}_i\right) \in G \times \mathfrak{g}$, $i = n, \ldots, n+k$. To simplify notation, we denote them by $(y_i, \dot{y}_i)$, $i = 1, \ldots, k+1$. We also choose a reference element among the data, $(y_1, \dot{y}_1)$, say. Relative to this reference element we compute the "transformed" quantities $u_i = \log(y_i y_1^{-1})$ and $v_i = \mathrm{dexp}_{u_i}^{-1}(\dot{y}_i)$, $i = 1, \ldots, k+1$.

A Hermite interpolation polynomial, $p(t) \in \mathfrak{g}$, may now be computed based on the $2k+2$ elements

$$(t_1, 0), \quad (t_i, u_i),\ i = 2, \ldots, k+1, \qquad \text{and} \qquad (t_1, \dot{y}_1), \quad (t_i, v_i),\ i = 2, \ldots, k+1.$$

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 *derivatives* at the point.

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 $p'(t_i) = v_i$, $i = 1, \ldots, k+1$. If

$$\begin{aligned}
f[t_j] &= u_j, \qquad j = 1, \ldots, k+1, \\
f[t_0, t_1, \ldots, t_s] &= \sum_{i=0}^{s} \frac{u_i}{\prod_{\substack{j=0 \\ j \neq i}} (x_i - x_j)} = \frac{f[t_1, \ldots, t_s] - f[t_0, \ldots, t_{s-1}]}{(t_s - t_0)}
\end{aligned}$$

and

$$f[t_i, \ldots, t_{i+k}] = \frac{f[t_{i+1}, \ldots, t_{i+k}] - f[t_i, \ldots, t_{i+k-1}]}{t_{i+1} - t_i},$$

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

$$p(t) = (t - t_0) f[t_0, t_1] + \cdots + \left\{ \prod_{i=0}^{2k} (t - t_i) \right\} f[t_0, t_1, \ldots, t_{2k+1}],$$

since $u_1 = 0$.

As in classical interpolation theory, we define $f[t_j, t_{j+1}] = v_i$ when $t_j = t_{j+1}$, hence Hermite and Hermite-Birkhoff 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

$$y(\tilde{t}) = \lambda_{y(t_n)}(p(\tilde{t})).$$

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

$$\left(y(\tilde{t}), \dot{y}(\tilde{t})\right) = \left(\lambda_{y(t_n)}(p(\tilde{t})), \mathrm{T}\lambda_{y(t_n)}(p(\tilde{t}), p'(\tilde{t}))\right). \tag{1}$$

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

Proof. The interpolation polynomial $p(t)$ belongs to $\mathfrak{g}$ and $(p(t), \dot{p}(t)) \in \mathrm{T}\mathfrak{g}$. Since $\lambda_y : \mathfrak{g} \to G$ and $\mathrm{T}\lambda_y : \mathrm{T}\mathfrak{g} \to G \times \mathfrak{g}$, 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 $\mathrm{T}\mathfrak{g}$. $\square$

Example 2.2. *When solving the nonlinear problem* $\dot{y}(t) = A(t, y(t))y(t)$ *by the Magnus series method developed in [9], we may employ waveform relaxation. This gives us the iteration scheme*

$$y^{[k+1]}(t_{n+1}) = A(t, y^{[k]}(t_n))y^{[k+1]}(t_{n+1}),$$

which can be integrated by the linear Magnus series method of order four, say. But we then need to approximate $y(t_n + c_i h)$, $i = 1, \dots, s$, *with sufficiently high order. Given* $\big(y(t_n), \dot{y}(t_n)\big)$ *and* $\big(y(t_{n+1}), \dot{y}(t_{n+1})\big)$ *we can construct a third order approximation to the quadrature points by the above described procedure. By computing the transformed quantities* u_i *and* v_i, $i = 1, 2$, *we get the interpolation points:*

$$(t_1, 0), \qquad (t_2, u_2), \qquad (t_1, v_1), \qquad \textit{and} \qquad (t_2, v_2).$$

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

$$p(t) = (t-t_1)v_1 + (t-t_1)^2 \frac{u_2 - (t_2-t_1)v_1}{(t_2-t_1)^2} + (t-t_1)^2(t-t_2)\frac{(t_2-t_1)(v_1+v_2) - 2u_2}{(t_2-t_1)^3}.$$

The cost of the above described Hermite interpolation procedure is approximately as follows. The transformation of k elements consists roughly of 1 inverse and k multiplications (however, the actual implementation may use $\sim n^3/3$ flops (LU-decomposition), $\sim n^3/2$ flops (n forward eliminations) plus $\sim n^3/2$ flops (n back substitutions) to accomplish this task), k logarithms ($\sim 25kn^3$ flops (depending on the method used)) and k dexp^{-1}. The computation of $p(t)$ is relatively cheap, while applying the action λ requires 1 exponentiation ($\sim 25n^3$ flops (depending on the method used)), 1 multiplication and possibly 1 dexp computation. Note that the functions dexp and dexp^{-1} are infinite sums. They need, however, only be evaluated to the order of the interpolation polynomial. For order 4, this leads to roughly $8n^3$ 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 in the setting of Lie groups has recently been addressed by Celledoni and Iserles [2].

3 Generalized Integration Methods with Continuous Extensions

We now 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 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 [13, 6] 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. According to Sophus Lie's third fundamental theorem [1], 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 mapping $\lambda_* : \mathfrak{g} \to \Xi(\mathcal{M})$, where $\Xi(\mathcal{M})$ is the set of all vector fields on $\mathcal{M}$, is defined as

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

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

$$y' = \mathcal{F}_\lambda(t, y) = \big(\lambda_* f(t, y)\big)(y), \qquad y(0) = p. \tag{2}$$

This is the generic form of an ordinary differential equation on a manifold, and if we assume that $y_0 \in \mathcal{M}$, then 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 [13] proved that the following diagram commutes

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

The solution of (2) 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

$$u'(t) = \widetilde{f}(u(t)) = \mathrm{dexp}_{u(t)}^{-1}\big(f(t, \lambda(u(t), p))\big), \qquad u(0) = 0. \tag{3}$$

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. [23, 15]).

3.1 Crouch-Grossman Methods

The Crouch-Grossman methods have been described in a number of texts, e.g. [3, 17, 11]. Much of the discussion in this section is based on notation and results from [17]. 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

$$\dot{y} = F(y) = \sum_{i=1}^{n} f_i(y) E_i, \qquad y(0) = y_0 \in \mathcal{M}, \tag{4}$$

where $f_i : \mathcal{M} \to \mathbb{R}$ are smooth functions. The solution of (4) 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 frozen coefficients relative to the frame. Let $\mathfrak{g}$ be the Lie algebra generated by the frame $E_1, \ldots, E_n$ and let $G \in \mathrm{Diff}(\mathcal{M})$ 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) 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:

$$\begin{aligned} Y_r &= \widetilde{\lambda}_{ha_{r,s}F_{Y_s}} \circ \widetilde{\lambda}_{ha_{r,s-1}F_{Y_{s-1}}} \circ \cdots \circ \widetilde{\lambda}_{ha_{r,1}F_{Y_1}}(y_k), \quad r = 1, \ldots, s & (5) \\ y_{k+1} &= \widetilde{\lambda}_{hb_s F_{Y_s}} \circ \widetilde{\lambda}_{hb_{s-1}F_{Y_{s-1}}} \circ \cdots \circ \widetilde{\lambda}_{hb_1 F_{Y_1}}(y_k). & (6) \end{aligned}$$

As for classical, continuous Runge-Kutta methods (see e.g. [7, 19, 20, 24, 25]) we consider interpolants defined by continuous weights, $b(\theta)$. The methods then take the form (5) together with

$$u(y_k; \theta) = \widetilde{\lambda}_{hb_s(\theta)F_{Y_s}} \circ \widetilde{\lambda}_{hb_{s-1}(\theta)F_{Y_{s-1}}} \circ \cdots \circ \widetilde{\lambda}_{hb_1(\theta)F_{Y_1}}(y_k). \tag{7}$$

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 $e^{\theta h F} p$. 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

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

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

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

where $F^{r+1} = F\big[F^r[\psi]\big]$, $r = 1, 2, \ldots$. By identifying elementary differentials, $\mathbb{F}$, with ordered (rooted) trees (see [17]), we can write the Lie series of the exact solution as

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

It follows that

$$\left. \frac{\mathrm{d}^q}{\mathrm{d}h^q} \left(\psi \circ e^{\theta h F} p\right) \right|_{h=0} = \theta^q \sum_{t \in T_O^{q+1}} \alpha(t) \mathbb{F}_p(t)[\psi](p), \qquad q = 0, 1, \ldots, \tag{8}$$

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

$$\left. \frac{\mathrm{d}^q}{\mathrm{d}h^q} \psi \circ \Psi(\theta h, p) \right|_{h=0} = \sum_{t \in T_O^{q+1}} \alpha(t)\gamma(t)\Phi(t;\theta)\mathbb{F}_p(t)[\psi](p), \qquad q = 1, 2, \ldots, \tag{9}$$

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 [17]. By equating (8) and (9) and requiring this condition to hold for each tree of order less that $p+2$, the next theorem follows.

Theorem 3.1. *(Continuous order conditions) The approximation given by (5) and (7) 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 [17]):*

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

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 in 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 [12, 14, 13]. While Crouch-Grossman methods consist of compositions of flows of vector fields on the manifold, Munthe-Kaas methods compose vector fields that are pulled back to the identity element in the group before computing the resulting action on the manifold.

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 $[A, B] = AB - BA$ when A and B are matrices, and B_k is the kth Bernoulli number. Note that when q is an even number, the sum runs from 2

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

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 $i = 1, 2, \dots, s$
  $u_i = h\sum_{j=1}^{s} a_{ij}\tilde{k}_j$
  $k_i = f\left(t_n + c_i h, \lambda(u_i, y_n)\right)$
  $\tilde{k}_i = \mathrm{dexpinv}(u_i, k_i, q)$
end
$v = h\sum_{j=1}^{s} b_j\tilde{k}_j$
$y_{n+1} = \lambda(v, y_n)$.

The coefficients are given by the qth order classical Runge-Kutta method's Butcher tableau [7]. Note that this algorithm describes equation (3) discretized by a classical Runge-Kutta method.

It is obvious that Munthe-Kaas methods, that directly are 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 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

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

with $f : \mathbb{R} \to \mathbb{R}^{n\times n}$. In [27], Zanna proposed a method to extend 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 $f_i \approx f(t_n + c_i h, y(t_n + c_i h))$, $i = 1, \ldots, s$, we can construct an approximation to $f(t, y)$ on the interval $[t_n, t_n + h]$ through its Lagrangian interpolating polynomial:

$$f\big(t, y(t)\big) \approx p_L(t) = \sum_{i=1}^{s} L_i\Big(\frac{t - t_n}{h}\Big) f_i. \tag{10}$$

Zanna considered the general case where $f : \mathbb{R} \times G \to \mathfrak{g}$ and hence the interpolation polynomial belongs to $\mathfrak{g}$. Note that this is approximation 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 $(y_i, f_i) \approx \big(y(t_n + c_i h), f(t_n + c_i h, y(t_n + c_i h))\big)$, $i = 1, \ldots, s$, 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. [8, 9]) inserted the interpolant (10) for f. By truncating the series, as described in [9], 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 [27, 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 $f_i = f(t_n + c_i h, Y_i)$, $i = 1, 2$. 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 $f_i \approx f\big(t_n + c_i h, y(t_n + c_i h)\big)$, $i = 1, 2$, 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

$$\begin{aligned}
\sigma_{11}(\theta) &= -\tfrac{\sqrt{3}}{2}\theta^2 + \big(\tfrac{1}{2} + \tfrac{\sqrt{3}}{2}\big)\theta \\
\sigma_{12}(\theta) &= \tfrac{\sqrt{3}}{2}\theta^2 + \big(\tfrac{1}{2} - \tfrac{\sqrt{3}}{2}\big)\theta \\
\sigma_{21}(\theta) &= -\tfrac{3}{8}\theta^4 + \big(\tfrac{3}{4} - \tfrac{\sqrt{3}}{12}\big)\theta^3 - \tfrac{1}{4}\theta^2 \\
\sigma_{22}(\theta) &= -\tfrac{3}{8}\theta^4 + \big(\tfrac{3}{4} + \tfrac{\sqrt{3}}{12}\big)\theta^3 - \tfrac{1}{4}\theta^2 \\
\sigma^n(\theta; h) &= h\big(\sigma_{11}(\theta) f_1 + \sigma_{12}(\theta) f_2\big) + \tfrac{h^2}{2}\big(\sigma_{21}(\theta) - \sigma_{22}(\theta)\big)[f_1, f_2].
\end{aligned}$$

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 [9] (see also [27]).

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

$$\int_0^{\theta h} P(\tau)\, \mathrm{d}\tau = h\big(\sigma_{11}(\theta) P(c_1 h) + \sigma_{12}(\theta) P(c_2 h)\big) + \mathcal{O}(h^{q+1}),$$

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 the following:

$$\frac{1}{3}\theta^3 - \frac{1}{2}\theta^2 + \frac{1}{6}\theta = 0 \tag{11}$$

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

The roots of (11) are $\theta \in \left\{\frac{1}{2}, 1\right\}$ while the only solution of (12) in the interval of interest is $\theta = 1$. We conclude by applying [27, 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 [7, 16], and the resulting methods are just the classical collocation methods applied in the setting of Munthe-Kaas.

Choose distinct abscissae values $c_1, \dots, c_s \in [0, 1]$ and construct a polynomial $p(t) \in \Pi_{s-1}$ that obeys equation (3) 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

$$u(t_n + h) = \int_{t_n}^{t_n + h} \operatorname{dexp}^{-1}_{u(t)}\Big(f\big(t, \lambda(u(t), y_n)\big)\Big)\, \mathrm{d}t, \tag{13}$$

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 an interpolation polynomial to the function $\widetilde{f}$:

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

Insertion of this approximation into (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 [27], we do not need any higher-order coefficients or weights.

3.4 Numerical Results

As a test problem we consider an example by Zanna [26]. 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 (2) reduces to

$$y' = f(t,y) \cdot y \qquad \text{where} \qquad f : \mathbb{R} \times G \to \mathfrak{g} \qquad \text{and} \qquad y(0) \in G. \tag{14}$$

The right hand side of (14) is 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 1 and 2 show some results from simulations with the continuous methods described in Examples 3.2 to 3.4.

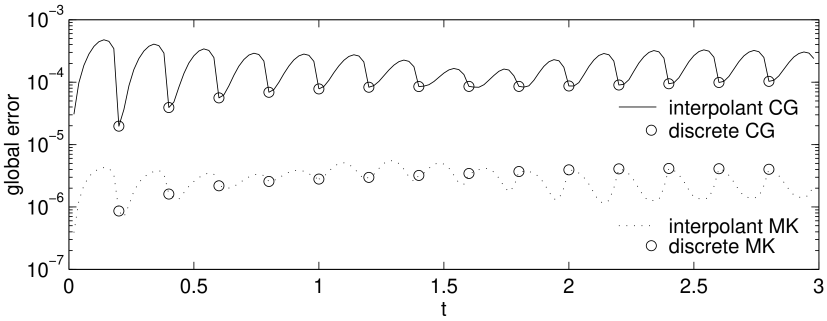

Figure 1: Global error committed by the interpolants (Crouch-Grossman and Munthe-Kaas). The circles denote the approximations of the underlying discrete scheme.

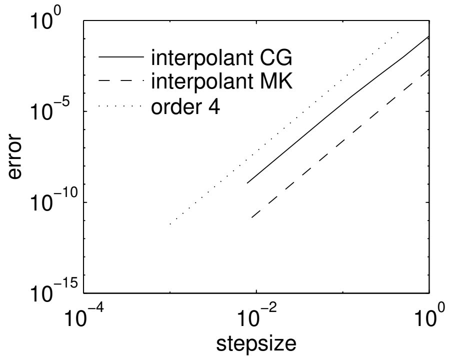

Figure 2: Local error committed by the interpolants as a function of the stepsize (Crouch-Grossman and Munthe-Kaas). The local order of approximation is given by the slope of the graphs.

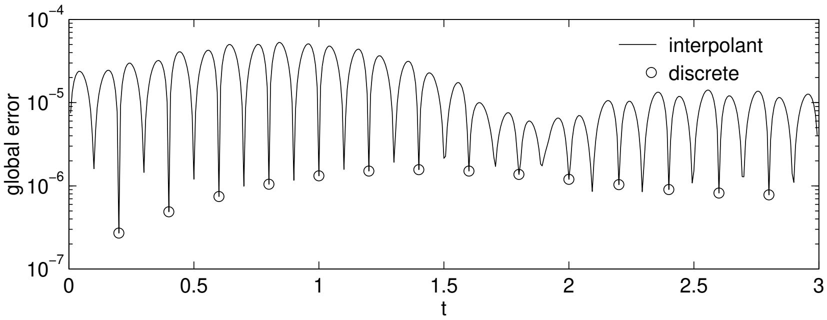

Figure 3: Global error committed by the interpolant to the collocation method. The circles denote the approximations of the underlying discrete collocation scheme.

4 Interpolation in the Setting of Homogeneous Manifolds

Many geometric integrators have been formulated so as to respect homogeneous manifolds. When the methods are applied in this setting, we can not necessarily use the procedure described in Section 2.

We will now compute the quantities to be used in the construction of interpolation polynomials similar to those described in Section 2. We will base the discussion on some ideas from [6].

Assume that $w_i^{(\ell)}$, defined by $\lambda(w_i^{(\ell)}, y_\ell) = y_{\ell,i}$, $i = 0, \dots, s_\ell - 1$ and $\ell = n, \dots, n+\mu$, are available. The point y_ℓ is the approximation to the exact solution at time t_ℓ produced by the discrete integration scheme (e.g. one of those described in Section 3). For notational convenience we let $y_{\ell,0} = y_{\ell+1}$ and thus $\lambda\big(\omega_0^{(\ell)}, y_\ell\big) = y_{\ell+1}$, $\ell = n, \dots, n+\mu-1$. Then $\lambda\big(\omega_i^{(\ell)}, y_\ell\big) = y_{\ell,i}$, $i = 1, \dots, s_\ell - 1$, are possible intermediate approximations at times $t_{\ell,i}$ to be used by the interpolation scheme. There is a total of $k = 1 + \sum_{j=n}^{n+\mu} s_j$ elements. Furthermore, assume for simplicity that the tangent vectors, $f_{\ell,i}$, at all of these points are available. The procedure can easily be changed to handle the case where not all tangent vectors are available.

Choose one element, $\big(y_n, f(t_n, y_n)\big)$, say, and express the available information in a coordinate system around this point. In this case we should not make any assumptions about the Lie algebra action λ, and we will show that transformation of the elements can be done in terms of the Baker-Campbell-Hausdorff formula and dexp^{-1} only. We are thus seeking u_j, $j = 1, \dots, k$, such that $\lambda(u_m, y_n) = \lambda\big(w_i^{(\ell)}, y_\ell\big) = y_{\ell,i}$, $i = 0, \dots, s_\ell - 1$ and $\ell = n, \dots, n+\mu$, where $m = 1 + i + \sum_{j=n}^{\ell-1} s_j$. By construction, $u_1 = 0$. The process is pictured in Figure 4.

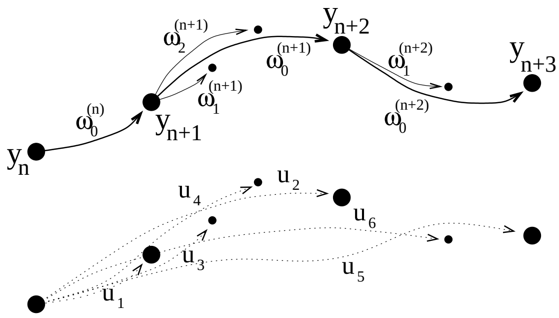

Figure 4: Relation between $\omega_i^{(\ell)}$ and u_j. The large bullets denote discrete approximations produced by the integration scheme, while the small bullets denote intermediate approximations to be used in the interpolation process.

We get

$$\lambda(u_m, y_n) = \lambda(w_i^\ell, y_\ell) = \lambda\Big(w_i^\ell, \lambda(w^\ell, y_{\ell-1})\Big) = \lambda\Big(\mathcal{B}(w_i^\ell, w^\ell), y_{\ell-1}\Big),$$

where $m = i + \sum_{j=n}^{\ell-1} s_j$. Furthermore,

$$\lambda\Big(\mathcal{B}(w_i^\ell, w^\ell), y_{\ell-1}\Big) = \lambda\Big(\mathcal{B}(\mathcal{B}(w_i^\ell, w^\ell), w^{\ell-1}), y_{\ell-2}\Big),$$

and this recursion continues until $y_{\ell-\mu} = y_n$, when we finally get

$$u_m = \mathcal{B}\Big(\cdots \mathcal{B}(\mathcal{B}(w_i^\ell, w^\ell), w^{\ell-1}) \cdots, w^{\ell-\mu}\Big),$$
$$i = 0, \ldots, s_\ell - 1, \quad \ell = n, \ldots, n+\mu \quad \text{and} \quad m = 1 + i + \sum_{j=n}^{\ell-1} s_j.$$

When u_i, $i = 1, \ldots, k$, are computed, the corrected tangent vectors v_i, $i = 1, \ldots, s_v$, may be computed according to

$$v_i = \operatorname{dexp}_{u_i}^{-1}(f_{\ell,i}), \qquad i = 1, \ldots, s_v.$$

A Hermite interpolation polynomial, $p(t) \in \mathfrak{g}$, may now be computed based on the $2k+2$ elements

$$(t_1, 0), \quad (t_i, u_i),\ i = 2, \ldots, k+1, \qquad \text{and} \qquad (t_1, \dot{y}_1), \quad (t_i, v_i),\ i = 2, \ldots, k+1.$$

The polynomial can be computed as described in Section 2, and approximation to elements in the time interval $[t_n, t_{n+\mu, s_{n+\mu}}]$ is given by (1). Theorem 2.1 still applies. And, 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 could be better to choose a point in the middle of the interpolation interval.

If $\mu = 0$ then C^0-continuity is assured if $\lambda_{y_n}(p(0)) = y_n$ and $\lambda_{y_n}(p(1)) = y_{n+1}$. Furthermore, the interpolant is C^1 if

$$\Big(\lambda_{y_n}(p(0)), \mathrm{T}\lambda_{y_n}(p(0), p'(0))\Big) = (y_n, \dot{y}_n) \tag{15}$$

and

$$\Big(\lambda_{y_n}(p(1)), \mathrm{T}\lambda_{y_n}(p(1), p'(1))\Big) = (y_{n+1}, \dot{y}_{n+1}). \tag{16}$$

The case $\mu \neq 0$ depends on how the interpolation polynomials are patched together. But, in general we have to impose conditions similar to (15) and (16) on the interpolation polynomials on each interval. C^0-continuity requires that $\lambda_{y_n}(p_m(0)) = y_n$ and $\lambda_{y_n}(p_m(1)) = y_{n+1}$ for all interpolation polynomials p_m with the numerical approximation at time t_n in the domain of definition. Likewise, the interpolant is C^1 if

$$\Big(\lambda_{y_n}(p_m(0)), \mathrm{T}\lambda_{y_n}(p_m(0), p_m'(0))\Big) = (y_n, \dot{y}_n)$$

and

$$\Big(\lambda_{y_n}(p_m(1)), \mathrm{T}\lambda_{y_n}(p_m(1), p_m'(1))\Big) = (y_{n+1}, \dot{y}_{n+1}).$$

5 Concluding Remarks

We have described how to perform Hermite interpolation on Lie groups. When integration 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 may be equipped with continuous weights. We have also showed how to construct continuous extensions to the Magnus series methods.

It could be of interest to perform Hermite interpolation on homogeneous 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 are 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 $k\widehat{v}$, $k \in \mathbb{R}$, satisfy the same relation. Therefore, all matrices $v, w \in \mathfrak{so}(3)$ related through $v = \mathcal{B}(w, k\widehat{v})$, $k \in \mathbb{R}$, 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 experiments indicate that interpolation results depend, e.g., on choice of elements $v_j \in \mathfrak{g}$ such that $\lambda(v_j, p_1) = p_j \in \mathcal{M}$, $j = 1, \ldots, s$. This will be the topic of future research.

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

References

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

[2] E. Celledoni and A. Iserles. Approximating the exponential from a Lie algebra to a Lie group. Technical Report 1998/NA3, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1998.

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

[4] L. Dieci, R. D. Russel, and E. S. van Vleck. Unitary Integrators and Applications to Continuous Orthonormalization Techniques. *SIAM J. Numer. Anal.*, 31, No. 1:261–281, 1994.

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

[6] S. Faltinsen, A. Marthinsen, and H. Munthe-Kaas. Multistep Methods Integrating Ordinary Differential Equations on Manifolds. Manuscript, 1998.

[7] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I, Nonstiff Problems*. Springer-Verlag, Second revised edition, 1993.

[8] A. Iserles, A. Marthinsen, and S. P. Nørsett. On the implementation of the method of Magnus series for linear differential equations. Technical Report 1998/NA2, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1998.

[9] A. Iserles and S. P. Nørsett. On the Solution of Linear Differential Equations in Lie Groups. Technical Report 1997/NA3, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997.

[10] B. Leimkuhler and G. W. Patrick. A Symplectic Integrator for Riemannian Manifolds, 1995.

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

[12] H. Munthe-Kaas. Lie-Butcher Theory for Runge-Kutta Methods. *BIT*, 35(4):572–587, 1995.

[13] H. Munthe-Kaas. High Order Runge-Kutta Methods on Manifolds. Technical Report 1997/NA14, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997.

[14] H. Munthe-Kaas. Runge-Kutta Methods on Lie Groups. *BIT*, 38(1):92–111, 1998.

[15] H. Munthe-Kaas and B. Owren. Computations in a Free Lie Algebra. Manuscript, 1998.

[16] S. P. Nørsett and G. Wanner. Perturbed Collocation and Runge-Kutta Methods. *Numer. Math.*, 38:193–208, 1981.

[17] 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.

[18] 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.

[19] B. Owren and M. Zennaro. Order Barriers for Continuous Explicit Runge-Kutta Methods. *Math. Comp.*, 56:645–661, 1991.

[20] B. Owren and M. Zennaro. Derivation of Efficient Continuous Explicit Runge-Kutta Methods. *SIAM J. Sci. Stat. Comput.*, 13:1488–1501, 1992.

[21] J. M. Sanz-Serna. Geometric Integration. Technical Report 1996/2, Departemento de Matemática Aplicada y Computación, Universidad de Valladolid, Spain, 1996.

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

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

[24] J. H. Verner. Differentiable Interpolants for High-Order Runge-Kutta Methods. *SIAM J. Numer. Anal.*, 30:1446–1466, 1993.

[25] J. H. Verner and M. Zennaro. Orders of Continuous Explicit Runge-Kutta Methods. *BIT*, 35(3):406–416, 1995.

[26] 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.

[27] A. Zanna. Collocation and relaxed collocation for the Fer and the Magnus expansions. Technical Report 1997/NA17, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1997.