Adjoint and selfadjoint Lie-group methods

Antonella Zanna[*], Kenth Engø[†] & Hans Z. Munthe-Kaas[‡]

May 27, 1999

Abstract

In this paper we discuss Lie-group methods and their dependence on centering coordinate charts. The definition of adjoint of a numerical method is thus subordinate to the method itself and the choice of the coordinate map. We study Lie-group numerical methods and their adjoint, and define selfadjoint numerical methods. The latter are defined in terms of classical selfadjoint Runge–Kutta schemes and symmetric coordinates, based on a geodesic midpoint or on a flow midpoint. As a result, the proposed selfadjoint Lie-group numerical schemes obey time-symmetry both for linear and nonlinear problems, a property that is illustrated with three numerical experiments.

1 Introduction

In a series or recent papers [16, 11, 23, 19, 3] a number of Lie-group methods have been introduced. Lie-group methods are numerical methods that solve the Lie-group differential equation

$$y' = \gamma(y)y, \qquad y(0) = y_0, \qquad t \in [0, T], \tag{1.1}$$

whereby $y(t) \in G$, a Lie-group, and $\gamma : G \to \mathfrak{g}$ is a sufficiently smooth function, $\mathfrak{g}$ being the Lie algebra of G, so that the numerical approximation $y_n \approx y(t_n)$, $t_0 = 0 < t_1 < \cdots < T = t_N$, also obeys $y_n \in G$ for all $0 < n \le N$, provided that the initial condition y_0 is also in G.

The majority of such methods is based on the following approach: assuming $y_n \in G$ given, we look for a solution y of (1.1) in the interval $[t_n, t_{n+1}]$ of the form

$$y(t) = \exp(\sigma(t))y_n, \tag{1.2}$$

whereby, for a continuous solution, we require that

$$\sigma(t_n) = 0.$$

A standard differentiation procedure (see for instance [16, 11, 23]) yields the differential equation for the unknown σ,

$$\sigma' = \mathrm{dexp}_\sigma^{-1}(\gamma(\exp(\sigma)y_n)), \qquad \sigma(t_n) = 0, \quad t \in [t_n, t_{n+1}]. \tag{1.3}$$

Both Lie-group Magnus-type methods [11, 23] and Munthe-Kaas-type schemes [16] are based on the solution of (1.3), the difference being that, for the first class of methods (1.3) is first integrated,

[*]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Silver Street, Cambridge, CB3 9EW, England. E-mail: `A.Zanna@damtp.cam.ac.uk`

[†]Department of Computer Science, University of Bergen, Postboks 7800, N-5020 Bergen, Norway. E-mail: `kenth@ii.uib.no`

[‡]Department of Computer Science, University of Bergen, Postboks 7800, N-5020 Bergen, Norway. E-mail: `hans@ii.uib.no`

truncated to appropriate order, and then the integrals are replaced with quadrature formulas, while, for the second class of schemes (1.3) is solved numerically (with Runge–Kutta schemes), provided that the function dexp^{-1} is truncated to appropriate order. Once the approximation

$$\sigma_{n+1} \approx \sigma(t_{n+1}),$$

is obtained, then the approximation of the solution of (1.1) is set

$$y_{n+1} = \mathrm{e}^{\sigma_{n+1}} y_n.$$

Before proceeding further, we shall note that the theory presented in this paper is, for the sake of simplicity, written in the notation of differential equations on matrix Lie groups. However, by straightforward substitutions, all results and algorithms hold also for equations on general homogeneous spaces [18]. Let $\mathcal{M}$ be a homogeneous space endowed with the transitive action $\Lambda : G \times \mathcal{M} \to \mathcal{M}$ of the Lie group G.

- Whenever $g, y \in G$, the product gy, $g \in G$, $y \in \mathcal{M}$, should be replaced by $\Lambda(g, y)$.
- Whenever $v \in \mathfrak{g}, y \in G$, the product vy should be replaced by $v_y = \frac{\partial}{\partial t}\big|_{t=0} \Lambda(\exp(tv), y)$ (note that $v_y \in \mathrm{T}\mathcal{M}_y$).

Having said this, we should explain what we mean by adjoint and selfadjoint numerical methods [9]. By *selfadjointness* we intend the following property: assume that, given $y_n \approx y(t_n)$, we integrate to t_{n+1} with a given numerical method and a given stepsize $h = t_{n+1} - t_n$ to obtain a numerical approximation y_{n+1}. If integrating between t_{n+1} and t_n, with the same numerical method and stepsize $-h$, it happens that the output of the method $\tilde{y}_n$ is always such that $\tilde{y}_n = y_n$, then the numerical method in question is said to be *selfadjoint*. Otherwise, the adjoint of the numerical method is the method that we need to employ to step from t_{n+1} with stepsize $-h$ and initial condition y_{n+1} in order to obtain y_n. Selfadjointness is sometimes known in literature as *time-symmetry* of a numerical scheme [9], not to be confused with *time-reversibility,* which is instead a property of some dynamical systems [15].

In a recent investigation on Lie-group numerical methods [12], it was discovered that, for *linear problems*, i.e. when $\gamma \equiv \gamma(t)$, Magnus-type methods based on symmetric collocation points, *á la* Gauss–Legendre, were selfadjoint. Similar results were obtained for the Munthe-Kaas-type methods [17], again when the RK methods employed in the numerical integration of (1.3) were time-symmetric (like in the classical theory of RK methods [9]).

However, for genuine nonlinear problems, when $\gamma \equiv \gamma(y)$, such schemes (with due exceptions, for instance the implicit midpoint rule) are not generally selfadjoint.

This paper is organized as follows: in the remainder of this section we introduce a numerical experiment that motivated our investigation. In Section 2 we discuss the dependence of Lie-group schemes on the choice of coordinate maps, an argument that is fundamental in the understanding of adjointness and selfadjointness of numerical methods. We introduce the adjointness condition for the center of a coordinate map, and present various results which allow us to construct the adjoint of a given numerical method, and selfadjoint schemes. In Section 3, the procedure is extended to methods based on the Magnus expansion. Finally, in Section 4, we present three numerical experiments and discuss the results obtained in this paper. The appendix is devoted to some results that are used in Section 2 and 3.

1.1 A numerical experiment: the Euler equations

Let $\boldsymbol{y}$ be a vector in $\mathbb{R}^3$. Then, denoted by $\times$ the classical vector product and introduced M, a diagonal matrix with entries m_1, m_2, m_3, the differential system

$$\boldsymbol{y}' = \boldsymbol{y} \times M\boldsymbol{y}, \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0, \tag{1.4}$$

is known as *Euler equations* of the rigid body, and has $H(\boldsymbol{y}) = \frac{1}{2}(m_1 y_1^2 + m_2 y_2^2 + m_3 y_3^2)$ as Hamiltonian function. In each interval $[t_n, t_{n+1}]$ we set $\boldsymbol{y}(t) = \Omega(t)\boldsymbol{y}_n$, whereby $\Omega \in \mathrm{SO}(3)$ obeys the differential equation

$$\Omega'(t) = \gamma(\boldsymbol{y}(t))\Omega, \quad \Omega(t_n) = I,$$

where

$$\gamma : \mathbb{R}^3 \to \mathfrak{g}, \quad \gamma(\boldsymbol{y}) = \widehat{-M\boldsymbol{y}},$$

the *hat-map* being defined as

$$\boldsymbol{v} \in \mathbb{R}^3 \to \hat{\boldsymbol{v}} = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}.$$

We refer the reader to [13] for further details.

Let us introduce the Lie-group version of the implicit midpoint rule for the problem (1.1).[1] Given y_n, we have

$$\begin{aligned} \sigma_{\frac{1}{2}} &= \tfrac{h}{2}\gamma_1, \\ \gamma_1 &= \gamma(\exp(\sigma_{\frac{1}{2}})y_n), \\ \sigma_{n+1} &= h\gamma_1, \\ y_{n+1} &= \exp(\sigma_{n+1})y_n. \end{aligned} \tag{1.5}$$

The midpoint method (1.5) is applied to the Euler equations (1.4), with initial condition $\boldsymbol{y}_0$, a random vector in $\mathbb{R}^3$ with unit norm, $m_1 = 1, m_2 = \frac{1}{3}, m_3 = \frac{1}{5}$ and stepsize $h = \frac{1}{10}$. The Lie-group implicit midpoint retains the norm of $\boldsymbol{y}$ to machine precision (the SO(3) structure is automatically preserved by a Lie-group method), while the energy error $H(\boldsymbol{y}_n) - H(\boldsymbol{y}_0)$, where H is the Hamiltonian function

$$H(\boldsymbol{y}) = \frac{1}{2}(m_1 y_1^2 + m_2 y_2^2 + m_3 y_3^3), \tag{1.6}$$

is plotted below in Figure 1.

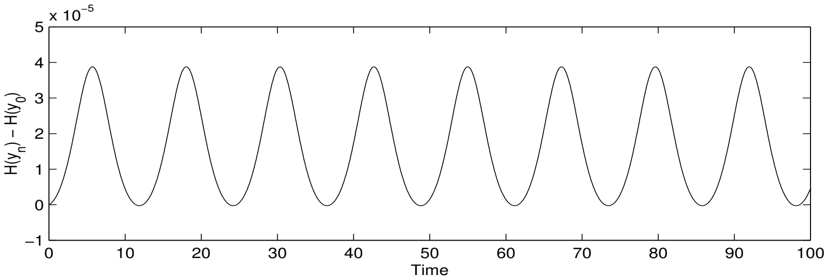

Figure 1: Energy error of the Lie-group version of the implicit midpoint method, when applied to the Euler equations (1.4). The interval of integration is $[0, 100]$. The energy error is uniformly bounded by a constant that depends on the stepsize and on the order of the method, i.e. 2.

The energy error is uniformly bounded by a constant of the order $\mathcal{O}(h^2)$ whereby h is the stepsize of integration. A similar behaviour is observed for various initial conditions and different choices of the stepsize and very large intervals of integration.

[1] Note that, in the case of the implicit midpoint, both the Magnus-type method and the Munthe-Kaas-type method coincide, because for order two it is not necessary to introduce any dexp^{-1} correction.

Let us consider next the Lie-group method of the Munthe-Kaas-type, based on the classical Gauss–Legendre RK method of order four. Once again, the norm of the solution is retained to machine accuracy, while the energy is oscillating but linearly decreasing (Figure 2, top). A similar behaviour is observed with the Magnus-type collocation method based on two Gaussian points [23], except that now the energy is increasing (Figure 2, bottom). The energy error behaviour *is not* significantly improved by evaluating the dexp^{-1} function more accurately.

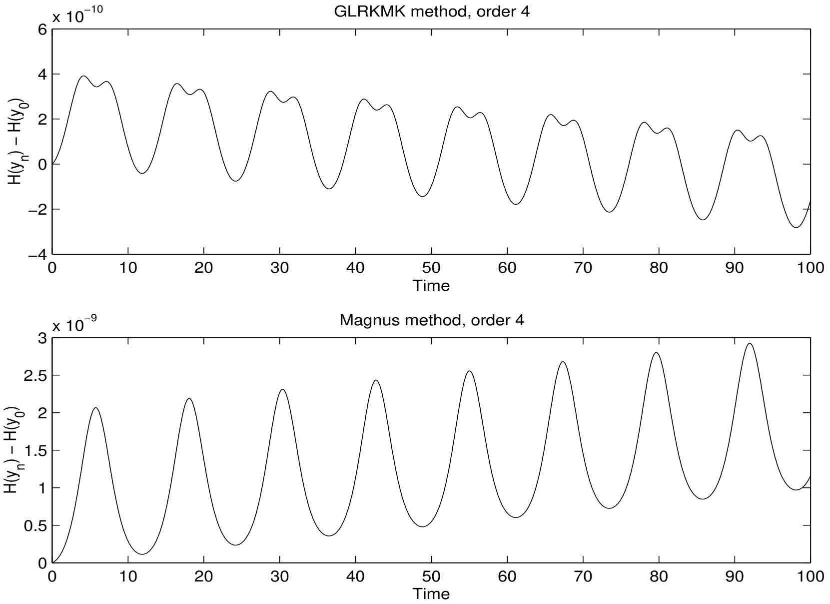

Figure 2: Energy error for the order four Gauss–Legendre RK method of the Munthe-Kaas-type (top) and Gauss–Legendre Magnus-type (bottom) when applied to the Euler equations (1.4). The interval of integration is $[0, 100]$.

Higher-order Gauss–Legendre-type Lie-group method behave more like the order-four ones above, rather than like the implicit midpoint. Thus the fact that the implicit midpoint retains the energy error uniformly in a band that depends on the stepsize, is more an exception than a rule for Gaussian-type methods.

Another consequence of the numerical experiment described above is that Gauss–Legendre-type methods (whether based on a Magnus expansion, or methods of the Munthe-Kaas-type) are not time-symmetric for nonlinear problems, while the implicit midpoint appears to be so. That higher order Gaussian-type methods are not time-symmetric for nonlinear problem is also easily verified by numerical experiments (see for instance [24]).

1.2 Time symmetry of Lie-group implicit midpoint

Let us return to the Lie-group implicit midpoint scheme (1.5). We can write

$$
\begin{aligned}
y_{n+1} &= \exp(\sigma_1) y_n = \exp(h\gamma_1) y_n \\
&= \exp(\frac{h}{2}\gamma_1 + \frac{h}{2}\gamma_1) y_n \\
&= \exp(\frac{h}{2}\gamma_1) y_{n+\frac{1}{2}},
\end{aligned}
$$

whereby we have denoted

$$y_{n+\frac{1}{2}} = \exp(\frac{h}{2}\gamma_1)y_n.$$

The point $y_{n+\frac{1}{2}}$ plays the role of a midpoint between y_n and y_{n+1} where the effective coordinate map is centered. Such $y_{n+\frac{1}{2}}$ is in effect the midpoint on the geodesic connecting y_n and y_{n+1}. Thus,

$$y_{n+1} = \exp(\frac{h}{2}\gamma_1)y_{n+\frac{1}{2}}, \quad y_n = \exp(-\frac{h}{2}\gamma_1)y_{n+\frac{1}{2}},$$

and the scheme is time-symmetric. The time symmetry of the implicit midpoint rule is due to the fact that one might consider the coordinates to be centered around a midpoint $y_{n+\frac{1}{2}}$, which is invariant under time symmetry. In the next section we will generalize this concept and show that most of the known Lie-group methods can be symmetrized in the same fashion.

2 Lie-group methods and midpoint-based coordinates

2.1 Methods centered at arbitrary points

Let us consider a more general form of the map (1.2). Let $\phi : \mathfrak{g} \to G$ be a smooth function such that $\phi(0) = e$ and $\phi'(0) = I$, the identity matrix (or more precisely $\mathrm{T}\phi(0, v) = v$). Thus ϕ is a diffeomorphism mapping a neighbourhood of $0 \in \mathfrak{g}$ to a neighbourhood of $e \in G$. Recurring examples of such maps are:

- Canonical coordinates of 1st kind:

$$\phi(v) = \exp(v).$$

- Canonical coordinates of 2nd kind:

$$\phi(v) = \mathrm{ccsk}(v) = \exp(\beta_1 v_1)\exp(\beta_2 v_2)\cdots\exp(\beta_d v_d),$$

  where $v = \sum_{j=1}^{d} \alpha_j v_j$, $\alpha_j \in \mathbb{R}$, $\{v_j\}_{j=1}^{d}$ is a basis for $\mathfrak{g}$, and the β_i's are real function of $\alpha_1, \ldots, \alpha_d$.

- Cayley coordinates for quadratic Lie groups:

$$\phi(v) = \mathrm{cay}(v) = (I + v/2)(I - v/2)^{-1}.$$

We can obtain coordinates around any point $q \in G$ by (inverting) the map $v \mapsto \phi(v)q$. In that case, we will say that we are employing *coordinates centered at* q. In particular, the solution of (1.2)-(1.3) is advanced in coordinates centered at y_n. For the general case of coordinates centered at some other point q near y_n we will write

$$q = \phi(\theta)^{-1}y_n,$$

$\phi(\theta)^{-1} \in G$ denoting the inverse of $\phi(\theta) \in G$, for some $\theta \in \mathfrak{g}$ to be specified in the sequel. We remark that canonical coordinates of the 1st kind and the Cayley transform both satisfy

$$\phi(v)^{-1} = \phi(-v), \qquad \forall v \in \mathfrak{g},$$

while, for coordinates of the second kind,

$$\mathrm{ccsk}(v)^{-1} = \exp(-\beta_d v_d)\exp(-\beta_{d-1}v_{d-1})\cdots\exp(-\beta_1 v_1).$$

The ultimate goal is to derive time symmetric methods by choosing θ such that we reach the same q when we step forwards from y_n as when step backwards from y_{n+1}.

For $u \in \mathfrak{g}$, let $\mathrm{d}\phi_u : \mathfrak{g} \to \mathfrak{g}$ denote the right trivialized tangent of ϕ, i.e.

$$\mathrm{T}\phi(u, v) = \mathrm{d}\phi_u(v)\phi(v).$$

Since $\phi'(0) = I$, we have $\mathrm{d}\phi_0 = I$. Thus for small u, $\mathrm{d}\phi_u$ is a linear invertible map with inverse $\mathrm{d}\phi_u^{-1}$. For the exponential map [10], as well as for the Cayley transform [4], we have

$$\begin{aligned} \mathrm{dexp}_u^{-1}(v) &= \sum_{k=0}^{\infty} \frac{B_k}{k!}\mathrm{ad}_u^k(v), \quad \text{where } B_k \text{ are Bernoulli numbers,} \\ \mathrm{dcay}_u^{-1}(v) &= v - \frac{1}{2}[u, v] - \frac{1}{4}uvu. \end{aligned}$$

With regards to coordinates of the second kind, it is difficult to give an explicit expression for dccsk_u^{-1}, although in many cases it can be computed very efficiently [19].

Our next step is to look for a solution y of (1.1) in the interval $[t_n, t_{n+1}]$ of the form

$$y(t) = \phi(\sigma(t))\phi(\theta)^{-1}y_n. \tag{2.1}$$

Differentiating $y(t)$ we find

$$y'(t) = \mathrm{d}\phi_{\sigma(t)}(\sigma'(t))\phi(\sigma(t))\phi(\theta)^{-1}y_n.$$

and using (1.1) we obtain the following equation for $\sigma(t)$,

$$\sigma'(t) = \mathrm{d}\phi_\sigma^{-1}(\gamma(y(t))), \tag{2.2}$$

with initial condition $\sigma(t_n) = \theta$. Comparing with (1.3), we observe that changing the center of the coordinate map does not affect the equation for σ, only the initial condition being changed. Solving (2.2) by a Runge–Kutta method results in the following algorithm:

Algorithm 1 (Runge–Kutta based Lie integrator in coordinates centered at $\phi(\theta)^{-1}y_n$) *Denote* $\mathrm{d}\phi^{-1}(\eta, \vartheta, p)$, $\eta, \vartheta \in \mathfrak{g}$, *the truncation of* $\mathrm{d}\phi_\eta^{-1}(\vartheta)$ *to order* p. *For instance*

$$\mathrm{dexp}^{-1}(\eta, \vartheta, p) = \sum_{k=0}^{p-1} \frac{B_k}{k!}\mathrm{ad}_\eta^k(\vartheta),$$

in the case of canonical coordinates of the first kind, the B_k *being Bernoulli numbers [1]. Then, a* ν*-stage order-*p *Lie-group method, based on the order-*p *RK scheme defined by the tableau*

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array},$$

in the interval $[t_n, t_{n+1}]$ *with coordinates centered at a generic point* $\phi(\theta)^{-1}y_n$ *reads*

$$\left.\begin{aligned} \sigma_i &= \theta + h\textstyle\sum_{j=1}^{\nu} a_j^i F_j, \\ \gamma_i &= \gamma(\phi(\sigma_i)\phi(\theta)^{-1}y_n), \\ F_i &= \mathrm{d}\phi^{-1}(\sigma_i, \gamma_i, p), \end{aligned}\right\} \qquad i = 1, \ldots, \nu,$$

$$\begin{aligned} \sigma_{n+1} &= \theta + h\textstyle\sum_{i=1}^{\nu} b_i F_i, \\ y_{n+1} &= \phi(\sigma)\phi(\theta)^{-1}y_n. \end{aligned} \tag{2.3}$$

2.2 Adjoint of Lie-group methods

Let us return to (2.2) and assume that such differential equation is solved numerically in $\mathfrak{g}$ with a one-step method Ψ and positive stepsize h. Ψ might be for instance a Runge–Kutta (RK) method. We will write $\sigma_{n+1} = \Psi(t_n, \sigma(t_n), h)$, where $h = t_{n+1} - t_n$.

It is well known that usually $\Psi(t_{n+1}, \Psi(t_n, \sigma, h), -h) \neq \sigma$, a feature that is instead true for the flow Φ of the exact solution, because of the group properties of the solution operator [20].

Instead, the *adjoint method* Ψ^*, of the numerical method Ψ has the property that

$$\Psi^*(t_{n+1}, \Psi(t_n, \sigma, h), -h) = \sigma,$$

or, in short-hand notation,

$$\Psi^*_{-h} \circ \Psi_h = \mathrm{id}$$

(see [9] for more details). As an example, the adjoint of forward Euler is backward Euler.

To obtain the adjoint of a method in a Lie-group setting, we have to take into account two main elements: that the numerical method Ψ allows us to advance in the algebra, and that the coordinates allow us to advance in the group. For what the first one is concerned, if (2.2) is solved in $[t_n, t_{n+1}]$ with a numerical method Ψ, then the adjoint Ψ^* of Ψ should be used when stepping backward (i.e. with negative h). Secondly, the coordinate map when stepping backward should be centered in the same point as when stepping forward. Therefore, if θ is the initial condition for (2.2) when solved with Ψ, and θ^* when solved with Ψ^*, we shall require that

$$\phi(\theta)^{-1} y_n = \phi(\theta^*)^{-1} y_{n+1}. \tag{2.4}$$

To achieve (2.4), we shall allow either θ, or θ^* (or both of them) to be functions of y_n and h. We write the functional dependence as $\theta_{h,n}$ and $\theta^*_{h,n}$. We define a pair of methods $\tilde{\Psi}$ and $\tilde{\Psi}^*$ on G as

$$\begin{aligned} \tilde{\Psi}(t_n, y_n, h) &= \phi(\sigma_{h,n+1})\phi(\theta_{h,n})^{-1} y_n && (2.5)\\ \tilde{\Psi}^*(t_n, y_n, h) &= \phi(\sigma^*_{h,n+1})\phi(\theta^*_{h,n})^{-1} y_n && (2.6)\\ \theta^*_{-h,n+1} &= \sigma_{h,n+1}, && (2.7) \end{aligned}$$

where $\sigma_{h,n+1} = \Psi(t_n, \theta_{h,n}, h)$ and $\sigma^*_{h,n+1} = \Psi^*(t_n, \theta^*_{h,n}, h)$ are the solutions of (2.2) on $[t_n, t_n + h]$ with initial conditions $\theta_{h,n}$ and $\theta^*_{h,n}$ using the methods Ψ and Ψ^*, respectively.

Theorem 2.1 (Adjoint of a Lie-group method)
With the same notation as above, $\tilde{\Psi}^*$ is the adjoint algorithm of $\tilde{\Psi}$ and $(\tilde{\Psi}^*)^* = \tilde{\Psi}$.

Proof. In short-hand notation, we have

$$\tilde{\Psi}^*_{-h} \circ \tilde{\Psi}_h(y_n) = \phi(\sigma^*_{-h,n+2})\phi(\theta^*_{-h,n+1})^{-1}\phi(\sigma_{h,n+1})\phi(\theta_{h,n})^{-1} y_n.$$

Equation (2.7) yields (2.4), hence the two middle terms compose into the identity. Now, $\sigma_{h,n+1}$ and $\sigma^*_{-h,n+2}$ are solutions of the same equation (2.2), where the start point of σ^* is the endpoint of σ, and σ^* is obtained by integrating with step $-h$. Since the underlying schemes Ψ_h and Ψ^*_h are adjoint, we get $\sigma^*_{-h,n+2} = \theta_{h,n}$ and hence $\tilde{\Psi}^*_{-h} \circ \tilde{\Psi}_h = \mathrm{Id}|_G$. □

We consider the scheme defined by (2.3) and we let θ be a function of the stage values,

$$\theta_{h,n} = \Theta(h; F_1, F_2, \ldots, F_\nu).$$

Hence Θ depends also on the coefficients of the scheme Ψ. Although now (2.2) becomes a functional differential equation because the initial condition depends functionally on the solution itself, it is possible to prove that its solution exists and is unique provided that the stepsize h is sufficiently small (see the Appendix).

Then, $\theta^*_{-h,n+1} = \Theta^*(-h; F_1^*, \ldots, F_\nu^*)$, hence, because of (2.7), we deduce that Θ and Θ^* obey the fundamental relation

$$\Theta^*(-h; F_1^*, \ldots, F_\nu^*) = \Theta(h; F_1, \ldots, F_\nu) + h \sum_{i=1}^{\nu} b_i F_i, \tag{2.8}$$

that we call *adjointness condition* for the midpoint, i.e. the center of the coordinate map.

Recall that a numerical method is said to be *selfadjoint* if $\Psi^* = \Psi$. Similarly, a RK-type Lie-group method is selfadjoint if $\tilde{\Psi}^* = \tilde{\Psi}$. There are several ways of constructing an adjoint midpoint and these will be discussed in the sequel.

Theorem 2.2 (Selfadjoint Lie-group methods)
Algorithm 1 is selfadjoint if the underlying Runge–Kutta scheme is selfadjoint and Θ *obeys*

$$\Theta(-h; F_\nu, \ldots, F_1) = \Theta(h; F_1, \ldots, F_\nu) + h \sum_{i=1}^{\nu} b_i F_i. \tag{2.9}$$

Proof. When we step backwards with the a selfadjoint RK method, the same stages and stage values are generated, but in opposite order. The result follows immediately from (2.8). □

2.3 Geodesic midpoint

Let us return to the function Θ and to the "midpoint" it generates. Given that we aim at devising selfadjoint Lie-group schemes, some obvious candidates for a suitable midpoint are: the midpoint (in time) of the flow between y_n and y_{n+1} and the midpoint of the geodesic connecting y_n and y_{n+1} (see Figure 3).

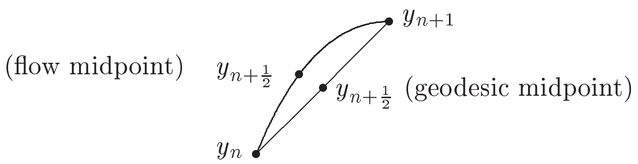

Figure 3: Flow midpoint and geodesic midpoint on the Lie-group G.

Definition 2.1 (Geodesic midpoint)
We say that the function

$$\theta_{h,n} = \Theta(h; F_1, \ldots, F_\nu) = -\frac{h}{2} \sum_{i=1}^{\nu} b_i F_i \tag{2.10}$$

defines a geodesic midpoint $y_{n+\frac{1}{2}} = \phi(\theta_{h,n})^{-1} y_n$.

The above definition is justified because, in the case of the exponential map, $y_{n+\frac{1}{2}}$ it is the midpoint on the geodesic connecting the numerical approximations y_n and y_{n+1}. In the sequel, we will often refer to coordinates that use a geodesic midpoint as *geodesic symmetric* coordinates.

Geodesic symmetric coordinates exist for all RK-type methods, whether explicit or implicit, and it is easily verified that they obey the relation (2.8).

Theorem 2.3 (Adjointness of the geodesic midpoint)
Geodesic midpoint based coordinates obey (2.8).

Proof. We have $\Theta^*(h; F_1^*, \ldots, F_\nu^*) = -h\sum_{i=1}^{\nu} b_i^* F_i^*$, whereby the b_i^*'s are the weights of the adjoint method of Ψ. However, $b_i^* = b_{\nu+1-i}$ (see [9]), hence $\Theta^*(-h; F_\nu, \ldots, F_1) = \frac{1}{2}h\sum_{i=1}^{\nu} b_i F_i = \Theta(h; F_1, \ldots, F_\nu) + h\sum_{i=1}^{\nu} b_i F_i$, and the result follows. □

Corollary 2.3.1 (Selfadjoint Lie-group method with geodesic midpoint)
Selfadjoint RK methods and geodesic midpoint yield selfadjoint Lie-group methods.

Proof. For selfadjoint RK methods, it is true that $b_i = b_{\nu+1-i}$ (see [9]), hence the result follows by virtue of Theorem 2.2 □

In passing, we mention that the Lie-group method $\tilde{\Psi}$ and its adjoint $\tilde{\Psi}^*$ (unless $\tilde{\Psi}$ is selfadjoint), when both stepping forward, do generally employ different geodesic midpoints.

Algorithm 2 (Lie-group RK methods with geodesic midpoint)
Let $\mathrm{d}\phi(\eta, \vartheta, p)$ be a p'th order approximation of $\mathrm{d}\phi_\eta^{-1}(\vartheta)$. A ν-stage order-p RK Lie-group method with geodesic-symmetric coordinates, based on the order-p RK scheme defined by the tableau

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array},$$

in the interval $[t_n, t_{n+1}]$, reads

$$\begin{aligned} \sigma_i &= h\textstyle\sum_{j=1}^{\nu}(a_j^i - \frac{1}{2}b_j)F_j, \\ \gamma_i &= \gamma(\phi(\sigma_i)\phi(\theta_{h,n})^{-1}y_n), \qquad i = 1, \ldots, \nu, \\ F_i &= \mathrm{d}\phi^{-1}(\sigma_i, \gamma_i, p), \\ \\ \sigma_{n+1} &= \tfrac{1}{2}h\textstyle\sum_{i=1}^{\nu} b_i F_i, \\ \theta_{h,n} &= -\sigma_{n+1}, \\ y_{n+1} &= \phi(\sigma_{n+1})\phi(\theta_{h,n})^{-1}y_n. \end{aligned} \tag{2.11}$$

In particular, if the underlying RK method is selfadjoint, also the above scheme is Lie-group selfadjoint.

Note that, both for canonical coordinates of the first kind and for the Cayley map for quadratic Lie-groups, at each step σ_{n+1} and $\theta_{h,n}$ commute; for coordinates of the second kind, y_{n+1} is instead obtained by means of a symmetric composition of exponentials, *a lá* Strang splitting.

2.4 Flow midpoint

In order to define a flow midpoint, we need to evaluate $y_{n+\frac{1}{2}}$ at the midpoint of the flow in the interval $[t_n, t_{n+1}]$. Therefore, we restrict our attention to methods based on collocation, since they naturally give a continuous solution in the interval under consideration. Assume that $c_1, \ldots, c_\nu \in [0, 1]$ are given nodes and denote

$$L_i(\tau) = \prod_{j \neq i}^{\nu} \frac{\tau - c_j}{c_i - c_j},$$

the cardinal Lagrangian polynomials. We set

$$w_i = \int_0^{\frac{1}{2}} L_i(\tau)\,\mathrm{d}\tau, \qquad i = 1, \ldots, \nu. \tag{2.12}$$

Definition 2.2 (Flow midpoint)
We say that the choice

$$\theta_{h,n} = \Theta(h; F_1, \ldots, F_\nu) = -h \sum_{i=1}^{\nu} w_i F_i, \tag{2.13}$$

defines a flow midpoint $y_{n+\frac{1}{2}} = \phi(\theta_{h,n})^{-1} y_n$.

In effect, Θ evaluates the value of the function σ at $t_n + \frac{h}{2}$, from which the name of flow midpoint. Often, coordinates that employ the flow midpoint will be referred to as *flow symmetric* coordinates.

Theorem 2.4 (Adjointness of the flow midpoint)
Flow midpoint based coordinates obey (2.8).

Proof. We have $\Theta^*(h; F_1^*, \ldots, F_\nu^*) = -h \sum_{i=1}^{\nu} w_i^* F_i^*$, where $w_i^* = \int_0^{\frac{1}{2}} L_i^*(\tau)\, \mathrm{d}\tau$. But $L_i^*(\tau) = \prod_{j \neq i} \frac{\tau - c_j^*}{c_i^* - c_j^*}$, whereby the condition for Ψ^* to be the adjoint of Ψ implies that $c_i^* = 1 - c_{\nu+1-i}$ (see [9]). Changing integration variable, we deduce that $w_i^* = \int_{\frac{1}{2}}^{1} L_{\nu+1-i}(\tau)\, \mathrm{d}\tau = b_{\nu+1-i} - w_{\nu+1-i}$, from which the result follows. □

Corollary 2.4.1 (Selfadjoint Lie-group methods with flow midpoint)
Selfadjoint RK methods in tandem with flow symmetric coordinates yield selfadjoint Lie-group methods.

Proof. Follows from the fact that, for selfadjoint collocation methods, it is true that $c_i = 1 - c_{\nu+1-i}$ (see [9]). □

Algorithm 3 (Lie-group collocation RK with flow midpoint)
Let $\mathrm{d}\phi(\eta, \vartheta, p)$ *be a p'th order approximation of* $\mathrm{d}\phi_\eta^{-1}(\vartheta)$*, as above. Consider a* ν*-stage order-*p *collocation RK method based on* $c_1, \ldots, c_\nu$*, whose tableau is*

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array},$$

with $a_j^i = \int_0^{c_i} L(\tau)\, \mathrm{d}\tau$*,* $b_i = \int_0^1 L_i(\tau)\, \mathrm{d}\tau$*. Define* $w_i = \int_0^{\frac{1}{2}} L_i(\tau)\, \mathrm{d}\tau$*. The corresponding Lie-group method based on flow-symmetric coordinates in the interval* $[t_n, t_{n+1}]$ *reads*

$$\begin{aligned}
&\sigma_i = h \textstyle\sum_{j=1}^{\nu} (a_j^i - w_j) F_j, \\
&\gamma_i = \gamma(\phi(\sigma_i)\phi(\theta_{h,n})^{-1} y_n), \qquad i = 1, \ldots, \nu, \\
&F_i = \mathrm{d}\phi^{-1}(\sigma_i, \gamma_i, p), \\
\\
&\sigma_{n+1} = h \textstyle\sum_{i=1}^{\nu} (b_i - w_i) F_i, \\
&\theta_{h,n} = -h \textstyle\sum_{i=1}^{\nu} w_i F_i, \\
&y_{n+1} = \phi(\sigma_{n+1})\phi(\theta_{h,n})^{-1} y_n.
\end{aligned} \tag{2.14}$$

In particular, if the underlying RK method is selfadjoint, also the above scheme is Lie-group selfadjoint.

Differently from the geodesic-midpoint case now σ_{n+1} and $\theta_{h,n}$ do not generally commute, even for canonical coordinates of the first kind or for Cayley maps for quadratic groups.

The question we address next is: what are all the possible choices of midpoints so that (2.9) is obeyed? As far as the general case is concerned, the following result holds.

Lemma 2.5
The set of all Θ's that obey (2.9) is convex.

Proof. If Θ_1 and Θ_2 satisfy (2.9), then so does also $\alpha\Theta_1 + (1-\alpha)\Theta_2$ for all α. □

In particular, for selfadjoint RK methods, also a linear combination of geodesic and flow midpoint yields a numerical scheme that is Lie-group selfadjoint. Other nontrivial choices of selfadjoint midpoints are described below.

Lemma 2.6
Let $\bar\sigma_i = h\sum_{j=1}^{\nu} a_j^i F_j$. If the underlying Runge–Kutta method is selfadjoint, then for any $i = 1, 2, \ldots, \nu$ the function

$$\Theta(h; F_1, \ldots, F_\nu) = -\frac{h}{2}\left(\bar\sigma_i + \bar\sigma_{\nu+1-i}\right) \tag{2.15}$$

satisfies (2.9).

Proof. We recall that selfadjoint methods satisfy

$$a_{\nu-j+1}^{\nu-i+1} + a_j^i = b_{\nu-j+1} = b_j$$

(see [9]). This yields:

$$\begin{aligned}
&\Theta(-h; F_\nu, \ldots, F_1) - \Theta(h; F_1, \ldots, F_\nu) = \\
&\quad \frac{h}{2}\sum_{j=1}^{\nu}\left(a_j^i F_{\nu+1-j} + a_j^{\nu+1-i} F_{\nu+1-j} + a_j^i F_j + a_j^{\nu+1-i} F_j\right) = \\
&\quad \frac{h}{2}\sum_{j=1}^{\nu}\left(\left(a_j^i + a_{\nu+1-j}^{\nu+1-i}\right) F_{\nu+1-j} + \left(a_j^i + a_{\nu+1-j}^{\nu+1-i}\right) F_j\right) = \\
&\quad \frac{h}{2}\sum_{j=1}^{\nu}\left(b_{\nu+1-j} F_{\nu+1-j} + b_j F_j\right) = h\sum_{j=1}^{\nu} b_j F_j,
\end{aligned}$$

which completes the proof. □

3 The Magnus expansion with geodesic and flow midpoints

Up to this point, we have focused on symmetric coordinates for RK methods of the Munthe-Kaas type. However, similar results hold also for methods based on the Magnus expansion, mainly constructed by means of collocation. The main difference with the approach above is that the differential equation (2.2) is first integrated, then truncated to appropriate order, and finally the integrals are approximated with suitable quadrature formulas. Furthermore, since the expansion of the solution heavily depends on the choice of ϕ, the procedure is far less treatable in generality, i.e. irrespective of ϕ. We will therefore describe in detail the case of canonical coordinates of the first kind, in order to give to the reader the idea of the typical procedure.

Hence, let us assume $y(t) = \exp(\sigma)\exp(-\theta_{h,n})y_n$ in the interval $[t_n, t_{n+1}]$, which, in the algebra $\mathfrak{g}$ reduces to (2.2). However, it is convenient to replace σ by $\sigma - \sigma_{\frac12}$, $\sigma_{\frac12} = -\theta_{h,n}$, hence to look for a solution $y(t) = \exp(\sigma - \sigma_{\frac12})\exp(\sigma_{\frac12})y_n$, so that σ is solution of the differential equation

$$\sigma' = \mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}}(\gamma(y(t))), \qquad \sigma(t_n) = 0. \tag{3.1}$$

In the classical setting, the differential equation for σ, $\sigma' = \mathrm{dexp}_\sigma^{-1}(\gamma)$, is integrated (by means of Picard iterations) and truncated to appropriate order. A truncated approximation of σ can be represented in terms of order trees [11, 23]. In particular a *canonical* Magnus expansion of order p is obtained by including *only* the necessary terms for order p [12]. Such canonical expansion can be written in terms of the function γ and its integrals, and commutators of those. To obtain collocation methods, with nodes $c_1, c_2, \ldots, c_\nu$, the function γ is then replaced by its Lagrangian interpolant

$$\gamma(y(t)) \approx \sum_{j=1}^{\nu} L\left(\frac{t-t_n}{h}\right)\gamma_i, \quad L_j(t) = \prod_{k\neq j} \frac{t-c_k}{c_j-c_k}, \quad t \in [t_n, t_{n+1}],$$

whereby $\gamma_i = \gamma(\exp(\sigma_i)y_n)$, and the coefficient of the methods are obtained by integrating combinations of the cardinal Lagrangian polynomials [23].

We apply the same procedure to (3.1). However, the resulting Magnus expansions are slightly different, in the case of flow-symmetric and geodesic-symmetric coordinates. Therefore, we shall consider the two cases separately.

3.1 Flow midpoint for the Magnus expansion

Our point of departure is the differential equation (3.1), in tandem with the assumption

$$\sigma_{\frac{1}{2}} = \sigma(t_n + \frac{h}{2}).$$

For ease of notation, let us focus on the interval $[0, h]$. By Picard iterations, we obtain

$$\sigma(t) = \int_0^t \gamma(s)\,\mathrm{d}s - \frac{1}{2}\int_0^t [\int_0^s \gamma(u)\,\mathrm{d}u - \int_0^{\frac{1}{2}h} \gamma(u)\,\mathrm{d}u, \gamma(s)]\,\mathrm{d}s + \text{h.o.t.}, \tag{3.2}$$

where we have omitted the explicit dependence of γ on y. Disregarding the higher order terms in (3.2) yields an order-four approximation to σ. We shall focus here on the procedure for order-four schemes only, although everything can be extended to higher order methods, for which the calculation of the coefficients is slightly more laborious.

From

$$\sigma(t) \approx \int_0^t \gamma(s)\,\mathrm{d}s - \frac{1}{2}\int_0^t [\int_0^s \gamma(u)\,\mathrm{d}u - \int_0^{\frac{1}{2}h} \gamma(u)\,\mathrm{d}u, \gamma(s)]\,\mathrm{d}s,$$

we obtain

$$\begin{aligned} \sigma(t) \quad \approx \quad & h\sum_{j=1}^{\nu} \int_0^t L_j(s)\,\mathrm{d}s\ \gamma_j - \frac{1}{2}h^2 \sum_{j,k=1}^{\nu} \int_0^t \int_0^s L_j(u)L_k(s)\,\mathrm{d}u\,\mathrm{d}s\ [\gamma_j, \gamma_k] \\ & + \frac{1}{2}h^2 \sum_{j,k=1}^{\nu} \int_0^t L_k(s)\,\mathrm{d}s \int_0^{\frac{1}{2}} L_j(u)\,\mathrm{d}u\ [\gamma_j, \gamma_k], \end{aligned}$$

by replacing γ by its interpolating polynomial. Thus, at the collocation nodes c_i, we find

$$\begin{aligned} &\sigma_i = h\textstyle\sum_{j=1}^{\nu} a_j^i \gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu}(a_{j,k}^i + w_j a_k^i)[\gamma_j, \gamma_k], \\ &\gamma_i = \gamma\Big(\exp(\sigma_i - \sigma_{\frac{1}{2}})\exp(\sigma_{\frac{1}{2}})y_0\Big), \end{aligned} \tag{3.3}$$

whereby, as in the classical theory [9],

$$a_j^i = \int_0^{c_i} L_j(s)\,\mathrm{d}s,$$

as in [23], we have

$$a^i_{j,k} = \int_0^{c_i} \int_0^t L_j(t) L_k(u) \, \mathrm{d}u \, \mathrm{d}t,$$

and the w_i's are given by (2.12). A similar rule applies to the calculation of $\sigma_{\frac{1}{2}}$,

$$\sigma_{\frac{1}{2}} = h \sum_{j=1}^{\nu} w_j \gamma_j + \frac{h^2}{2} \sum_{j,k=1}^{\nu} (w_{j,k} + w_j w_k)[\gamma_j, \gamma_k],$$

where

$$w_{j,k} = \int_0^{\frac{1}{2}} \int_0^t L_j(t) L_k(s) \, \mathrm{d}s \, \mathrm{d}t. \tag{3.4}$$

However, due to the skew-symmetry of the commutator operator,

$$\sum_{j,k=1}^{\nu} w_j w_k [\gamma_j, \gamma_k] = 0,$$

hence, $\sigma_{\frac{1}{2}}$ reduces to

$$\sigma_{\frac{1}{2}} = h \sum_{j=1}^{\nu} w_j \gamma_j + \frac{h^2}{2} \sum_{j,k=1}^{\nu} w_{j,k} [\gamma_j, \gamma_k].$$

Once the internal stages are evaluated, we can compute

$$\sigma_h = h \sum_{j=1}^{\nu} b_j \gamma_j + \frac{h^2}{2} \sum_{j,k=1}^{\nu} (b_{j,k} + w_j b_k)[\gamma_j, \gamma_k], \tag{3.5}$$

where

$$b_j = \int_0^1 L_j(t) \, \mathrm{d}t, \quad b_{j,k} = \int_0^1 \int_0^t L_j(t) L_k(s) \, \mathrm{d}s \, \mathrm{d}t.$$

This completes the list of ingredients for the evaluation of y_1,

$$y_1 = \exp(\sigma_h - \sigma_{\frac{1}{2}}) \exp(\sigma_{\frac{1}{2}}) y_0.$$

The same procedure is applied in each interval $[t_n, t_{n+1}]$.

Similarly to the the procedure presented in the previous section, we have that $\sigma_{\frac{1}{2}}$ is a function of the stage values,

$$\sigma_{\frac{1}{2}} = -\theta_{h,n} = -\Theta(h; \gamma_1, \ldots, \gamma_\nu).$$

Thus, because of (2.7), the adjointness condition (2.8) becomes equivalent to

$$\Theta^*(-h; \gamma_\nu, \ldots, \gamma_1) = \Theta(h; \gamma_1, \ldots, \gamma_\nu) + h \sum_{i=1}^{\nu} b_i \gamma_i + \frac{h^2}{2} \sum_{i,j=1}^{\nu} (b_{i,j} + w_i b_j)[\gamma_i, \gamma_j]. \tag{3.6}$$

Definition 3.1
We say that the choice

$$\Theta(h; \gamma_1, \ldots, \gamma_\nu) = -h \sum_{i=1}^{\nu} w_i \gamma_i - \frac{h^2}{2} \sum_{i,j=1}^{\nu} w_{i,j} [\gamma_i, \gamma_j]$$

defines a flow midpoint $y_{n+\frac{1}{2}} = \exp(-\theta_{h,n}) y_n$ *for the Magnus expansion.*

Theorem 3.1 (Adjointness of the flow midpoint for Magnus expansions)
The flow midpoint for the Magnus expansion obeys (3.6).

Proof. Note that

$$\Theta^*(-h;\gamma_\nu,\ldots,\gamma_1) = h\sum_{i=1}^{\nu} w_i^*\gamma_{\nu+1-i} - \frac{h^2}{2}\sum_{i,j} w_{i,j}^*[\gamma_{\nu+1-i},\gamma_{\nu+1-j}].$$

Hence, recall that $w_i^* = b_{\nu+1-i} - w_{\nu+1-i}$. Furthermore, by direct computation, we observe that $w_{i,j}^* = b_{\nu+1-i}b_{\nu+1-j} - w_{\nu+1-i}b_{\nu+1-j} - b_{\nu+1-i,\nu+1-j} + w_{\nu+1-i,\nu+1-j}$ from which (3.6) follows, since $\sum_{i,j} b_i b_j[\gamma_i,\gamma_j] = 0$. □

Corollary 3.1.1 (Selfadjoint Magnus methods with flow midpoint)
Flow midpoint coordinates and selfadjoint collocation methods yield selfadjoint Magnus methods.

Proof. It follows from the fact that, for selfadjoint collocation methods, the collocation nodes obey $c_i = 1 - c_{\nu+1-i}$. □

Algorithm 4 (Collocation order-four Magnus methods with flow midpoint)
Assume that $c_1,\ldots c_\nu$ are collocation nodes in $[0,1]$. Furthermore, assume that, in the classical setting, these nodes originate a collocation RK scheme of order p, with $p \geq 4$. Then the following scheme yields a collocation Magnus method on Lie groups of order four. Let $y_n \in G$, $t \in [t_n, t_{n+1}]$.

$$\begin{aligned}
\sigma_i &= h\textstyle\sum_{j=1}^{\nu} a_j^i\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu}(a_{j,k}^i + w_j a_k^i)[\gamma_j,\gamma_k], \\
\sigma_{\frac12} &= h\textstyle\sum_{j=1}^{\nu} w_j\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu} w_{j,k}[\gamma_j,\gamma_k], \qquad\qquad i = 1,\ldots,\nu, \\
\gamma_i &= \gamma\Big(\exp(\sigma_i - \sigma_{\frac12})\exp(\sigma_{\frac12})y_n\Big), \\
\sigma_{n+1} &= h\textstyle\sum_{j=1}^{\nu} b_j\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu}(b_{j,k} + w_j b_k)[\gamma_j,\gamma_k], \\
y_{n+1} &= \exp(\sigma_{n+1} - \sigma_{\frac12})\exp(\sigma_{\frac12})y_n.
\end{aligned} \tag{3.7}$$

In particular, if the underlying RK method is selfadjoint (i.e. the collocation nodes are symmetrically distributed in $[0,1]$), then also the above scheme is selfadjoint.

Example 3.1
Let us construct an order-four Magnus method based on Gaussian quadrature with flow symmetric coordinates. We have $c_1 = \frac12 - \frac{\sqrt3}{6}$ and $c_2 = \frac12 + \frac{\sqrt3}{6}$, hence the method (3.7) results into the numerical scheme

$$\begin{aligned}
\sigma_1 &= h(\tfrac14\gamma_1 + (\tfrac14 - \tfrac{\sqrt3}{6})\gamma_2) + \tfrac{h^2}{2}(\tfrac{1}{144} - \tfrac{\sqrt3}{48})[\gamma_1,\gamma_2], \\
\gamma_1 &= \gamma\Big(\exp(\sigma_1 - \sigma_{\frac12})\exp(\sigma_{\frac12})y_n\Big), \\
\sigma_2 &= h((\tfrac14 + \tfrac{\sqrt3}{6})\gamma_1 + \tfrac14\gamma_2) + \tfrac{h^2}{2}(-\tfrac{1}{144} - \tfrac{\sqrt3}{48})[\gamma_1,\gamma_2], \\
\gamma_2 &= \gamma\Big(\exp(\sigma_2 - \sigma_{\frac12})\exp(\sigma_{\frac12})y_n\Big), \\
\sigma_{\frac12} &= h((\tfrac14 + \tfrac{\sqrt3}{8})\gamma_1 + (\tfrac14 - \tfrac{\sqrt3}{8})\gamma_2) - \tfrac{\sqrt3}{96}h^2[\gamma_1,\gamma_2], \\
\sigma_{n+1} &= \tfrac{h}{2}(\gamma_1 + \gamma_2) - \tfrac{\sqrt3}{48}h^2[\gamma_1,\gamma_2], \\
y_{n+1} &= \exp(\sigma_{n+1} - \sigma_{\frac12})\exp(\sigma_{\frac12})y_n.
\end{aligned}$$

3.2 Collocation methods based on the geodesic midpoint

When using the geodesic midpoint we have $\sigma_{\frac{1}{2}} = \frac{1}{2}\sigma_h$, hence integrating (3.1) by Picard iteration, we find

$$\sigma(t) = \int_0^t \gamma(s)\,\mathrm{d}s - \frac{1}{2}\int_0^t [\int_0^s \gamma(u)\,\mathrm{d}u - \frac{1}{2}\int_0^h \gamma(u)\,\mathrm{d}u, \gamma(s)]\,\mathrm{d}s + \text{h.o.t.}, \tag{3.8}$$

For order-four schemes, we disregard the higher order terms. We replace γ by its interpolating polynomial, to find

$$\begin{aligned} \sigma(t) \quad &\approx \quad h\sum_{j=1}^{\nu}\int_0^t L_j(s)\,\mathrm{d}s\;\gamma_j - \frac{1}{2}h^2\sum_{j,k=1}^{\nu}\int_0^t\int_0^s L_j(u)L_k(s)\,\mathrm{d}u\,\mathrm{d}s\;[\gamma_j,\gamma_k] \\ &\qquad + \frac{1}{4}h^2\sum_{j,k=1}^{\nu}\int_0^t L_k(s)\,\mathrm{d}s\int_0^1 L_j(u)\,\mathrm{d}u\;[\gamma_j,\gamma_k]. \end{aligned}$$

Thus, at the collocation point c_i,

$$\begin{aligned} \sigma_i &= h\textstyle\sum_{j=1}^{\nu} a^i_j\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu}(a^i_{j,k} + \frac{1}{2}b_j a^i_k)[\gamma_j,\gamma_k], \\ \gamma_i &= \gamma\Big(\exp(\sigma_i - \sigma_{\frac{1}{2}})\exp(\sigma_{\frac{1}{2}})y_n\Big), \end{aligned} \tag{3.9}$$

so that we can finally calculate

$$\sigma_h = h\sum_{j=1}^{\nu} b_j\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu} b_{j,k}[\gamma_j,\gamma_k],$$

and hence

$$y_1 = \exp(\sigma_h)y_0.$$

Analogous procedure in the interval $[t_n, t_{n+1}]$.

We observe that (2.7) is now equivalent to

$$\Theta^*(-h;\gamma_\nu,\ldots,\gamma_1) = \Theta(h;\gamma_1,\ldots,\gamma_\nu) + h\sum_{i=1}^{\nu} b_i\gamma_i + \frac{h^2}{2}\sum_{i,j=1}^{\nu} b_{i,j}[\gamma_i,\gamma_j]. \tag{3.10}$$

Definition 3.2
We say that the choice

$$\Theta(h;\gamma_1,\ldots,\gamma_\nu) = -\frac{h}{2}\sum_{i=1}^{\nu} b_i\gamma_i - \frac{h^2}{4}\sum_{i,j=1}^{\nu} b_{i,j}[\gamma_i,\gamma_j]$$

defines a geodesic midpoint for the Magnus expansion.

Theorem 3.2 (Adjointness of the geodesic midpoint for Magnus expansions)
The geodesic midpoint for the Magnus expansion obeys (3.10).

Proof. Recall that $b^*_i = b_{\nu+1-i}$. Furthermore, by direct computation, $b^*_{i,j} = b_{\nu+1-i}b_{\nu+1-j} - b_{\nu+1-i,\nu+1-j}$ from which (3.10) follows by direct substitution. □

Corollary 3.2.1 (Selfadjoint Magnus methods with geodesic midpoint)
Geodesic midpoint coordinates and selfadjoint collocation methods yield selfadjoint Magnus methods.

Proof. See Corollary 3.1.1. □

Algorithm 5 (Collocation order-four Magnus methods with geodesic midpoint) *With the same assumptions as in Algorithm 4, the corresponding collocation order-four Magnus method based on geodesic midpoint reads*

$$\begin{aligned}
\sigma_i &= h\sum_{j=1}^{\nu} a_j^i\gamma_j + \tfrac{h^2}{2}\sum_{j,k=1}^{\nu}(a_{j,k}^i + \tfrac{1}{2}b_j a_k^i)[\gamma_j,\gamma_k], \\
\sigma_{\frac{1}{2}} &= \tfrac{1}{2}h\sum_{j=1}^{\nu} b_j\gamma_j + \tfrac{h^2}{4}\sum_{j,k=1}^{\nu} b_{j,k}[\gamma_j,\gamma_k], \qquad\qquad i = 1,\dots,\nu, \\
\gamma_i &= \gamma(\exp(\sigma_i - \sigma_{\frac{1}{2}})\exp(\sigma_{\frac{1}{2}})y_n), \\
\sigma_{n+1} &= h\sum_{j=1}^{\nu} b_j\gamma_j + \tfrac{h^2}{2}\sum_{j,k=1}^{\nu} b_{j,k}[\gamma_j,\gamma_k], \\
y_{n+1} &= \exp(\sigma_{n+1})y_n.
\end{aligned} \tag{3.11}$$

In particular, if $c_i = 1 - c_{\nu+1-i}$, *then also the above procedure yields a selfadjoint Lie-group numerical method.*

Example 3.2
For the order-four Magnus method based on Gaussian quadrature with geodesic midpoint, the algorithm (3.11) results into the numerical scheme

$$\begin{aligned}
\sigma_1 &= h(\tfrac{1}{4}\gamma_1 + (\tfrac{1}{4} - \tfrac{\sqrt{3}}{6})\gamma_2) + \tfrac{h^2}{2}(\tfrac{5}{72} - \tfrac{\sqrt{3}}{12})[\gamma_1,\gamma_2], \\
\gamma_1 &= \gamma\Big(\exp(\sigma_1 - \sigma_{\frac{1}{2}})\exp(\sigma_{\frac{1}{2}})y_n\Big), \\
\sigma_2 &= h((\tfrac{1}{4} + \tfrac{\sqrt{3}}{6})\gamma_1 + \tfrac{1}{4}\gamma_2) + \tfrac{h^2}{2}(-\tfrac{5}{72} - \tfrac{\sqrt{3}}{12})[\gamma_1,\gamma_2], \\
\gamma_2 &= \gamma\Big(\exp(\sigma_2 - \sigma_{\frac{1}{2}})\exp(\sigma_{\frac{1}{2}})y_n\Big), \\
\sigma_{\frac{1}{2}} &= \tfrac{h}{4}(\gamma_1 + \gamma_2) - \tfrac{\sqrt{3}}{24}h^2[\gamma_1,\gamma_2], \\
\sigma_{n+1} &= \tfrac{h}{2}(\gamma_1 + \gamma_2) - \tfrac{\sqrt{3}}{12}h^2[\gamma_1,\gamma_2], \\
y_{n+1} &= \exp(\sigma_{n+1})y_n.
\end{aligned}$$

4 Numerical experiments

4.1 The Euler equations

Let us return to the Euler equations for the rigid body that we have already introduced in § 1.1. We observed that the order-four RKMK method based on Gauss–Legendre nodes with coordinates centered at y_n had an energy dissipating behaviour (see Fig 2). The same behaviour was observed for the corresponding Magnus method as derived in [23]. Those methods are not selfadjoint, because, although the corresponding classical methods are selfadjoint, that is not the case for the coordinate map.

When we consider the same methods but with symmetric coordinates, both flow symmetric and geodesic symmetric, as described in Algorithms 2-3 and 4-5, it is easily verified by numerical experiments that such methods are selfadjoint or time-symmetric.

Furthermore, numerical experiments reveal that time-symmetry also improves long time behaviour of the numerical approximations. In Figure 4 we display the energy error of the symmetrized methods, the numerical experiments being performed with exactly the same initial condition and stepsize as in § 1.1. Instead of increasing, the energy error is now nicely oscillating

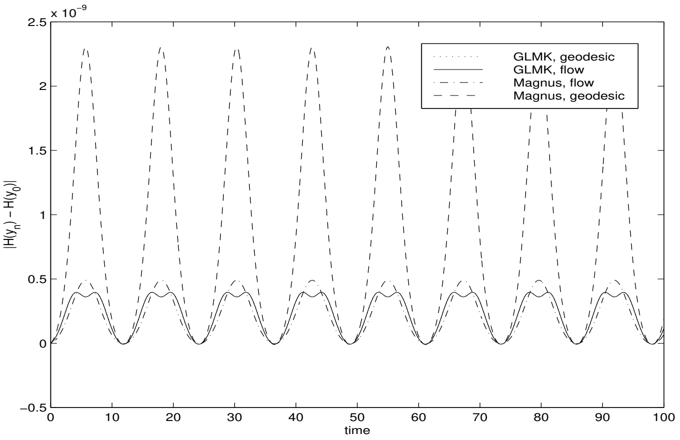

Figure 4: Error in the Hamiltonian of the Euler equations for the rigid body, with Gauss–Legendre methods based on symmetric coordinates. Compare with the non-symmetric coordinates in Fig 2.

in a band whose amplitude is proportional to h^4. This peculiar type error in the Hamiltonian function can be explained by means of the KAM theory [15]. In our case, the Euler equations are reversible, since changing the velocity of the body is equivalent to replacing y with $-y$. This is equivalent to changing the sign of the vector field (1.4), hence to integrating backwards in time. Thus, time-reversibility and selfadjointness reduce to the same condition, from which we deduce that selfadjoint Lie-group methods are time-reversible in this particular example. Therefore, the fact that we are solving a nearby reversible problem, together with the fact that the solution is constrained on a manifold (the sphere) justifies the observed behaviour of the Hamiltonian error.

A more rigourous explanation follows from backward error analysis for Lie-group methods [6].

4.2 The heavy top

We will next consider a symmetric heavy top modelled on $\mathfrak{se}(3)^*$; the dual space of the Lie algebra $\mathfrak{se}(3)$. We recall that $\mathfrak{se}(3)^*$ is isomorphic to the space $\mathbb{R}^3 \times \mathbb{R}^3$, therefore we will use a pair of 3-vectors $(\mathbf{\Pi}, \mathbf{\Gamma})$ to represent the state of the heavy top.

The equations of motion for the heavy top are given by

$$\begin{aligned} \frac{d\mathbf{\Pi}}{dt} &= \mathbf{\Pi} \times \mathbf{\Omega} + Mgl\mathbf{\Gamma} \times \boldsymbol{\chi}, \\ \frac{d\mathbf{\Gamma}}{dt} &= \mathbf{\Gamma} \times \mathbf{\Omega}, \end{aligned}$$

where $\mathbf{\Pi}, \mathbf{\Gamma}, \mathbf{\Omega}, \boldsymbol{\chi} \in \mathbb{R}^3$. $\mathbf{\Pi}$ is the angular momentum, $\mathbf{\Gamma}$ is the gravity vector as seen in local coordinates, $\mathbf{\Omega}$ is the angular velocity, and $\boldsymbol{\chi}$ is the constant unit vector along the line segment of length l connecting origin and the center of gravity. M is mass, and g is the gravitational constant. We refer to [13] for background and notation.

The system (4.1) is Hamiltonian in the Lie-Poisson structure on $\mathfrak{se}(3)^*$, the heavy top bracket being

$$\{f, h\}(\mathbf{\Pi}, \mathbf{\Gamma}) = -\mathbf{\Pi} \cdot (\nabla_\Pi f \times \nabla_\Pi h) - \mathbf{\Gamma} \cdot (\nabla_\Pi f \times \nabla_\Gamma h - \nabla_\Pi h \times \nabla_\Gamma f). \tag{4.1}$$

Furthermore, the heavy top Hamiltonian is given by

$$\mathrm{H}(\mathbf{\Pi}, \mathbf{\Gamma}) = \frac{1}{2} \mathbf{\Pi} \mathrm{I}^{-1} \mathbf{\Pi} + Mgl\mathbf{\Gamma} \cdot \boldsymbol{\chi}, \tag{4.2}$$

where I denotes the inertia tensor.

Because the heavy top equations are Lie-Poisson we use the coadjoint action to advance the solution on $\mathfrak{se}(3)^*$ (see [5]). The coadjoint action of SE(3) on its dual Lie algebra $\mathfrak{se}(3)^*$ is

$$\mathrm{Ad}^*_{(R,\mathbf{\Omega})^{-1}}(\mathbf{\Pi},\mathbf{\Gamma}) = (R\mathbf{\Pi} + \mathbf{\Omega}\times R\mathbf{\Gamma}, R\mathbf{\Gamma}).$$

The symmetric heavy top possesses four conserved quantities: the Hamiltonian (4.2); the projection of the angular momentum on the symmetry axis of the top $\mathbf{\Pi}\cdot\boldsymbol{\chi}$; the projection of the angular momentum on the gravity vector $\mathbf{\Pi}\cdot\mathbf{\Gamma}$; and finally the norm of the gravity vector $||\mathbf{\Gamma}||^2$. The two latter first integrals are Casimirs of the bracket (4.1), and hence they are preserved under the coadjoint action and rendered to machine accuracy in the numerical simulation.

In the numerical test we use the following data:

$$\mathrm{I} = 1/8 \begin{bmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad \boldsymbol{\chi} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (\mathbf{\Pi}_0, \mathbf{\Gamma}_0) = \left(\begin{bmatrix} 0 \\ 0 \\ 0.25 \end{bmatrix}, \begin{bmatrix} 0 \\ -0.195090 \\ 0.980785 \end{bmatrix} \right).$$

Moreover, the gravitational constant g equals 9.81, l equals $\frac{\sqrt{3}}{2}$, and the mass M equals 1. The stepsize of integration is $h = 1/20$, while the interval of integration is $[0, 20]$.

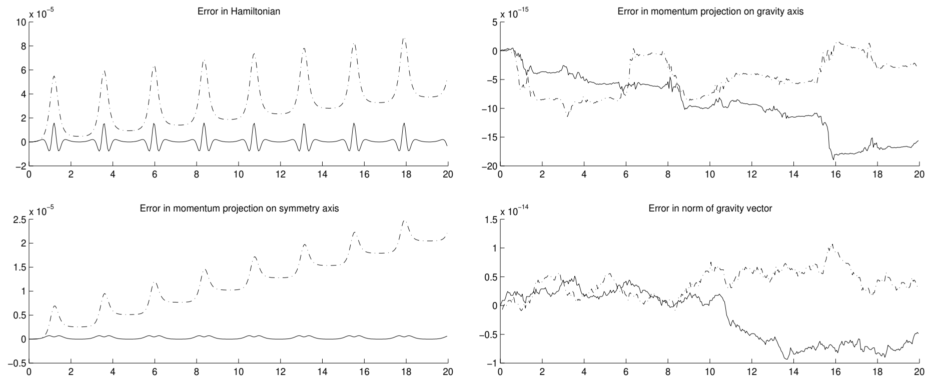

Figure 5: Order-four Gauss–Legendre Magnus method with flow-symmetric coordinates (solid line) compared with coordinates centered at y_n (dash-dotted line).

In Figure 5 the order-four Gauss–Legendre Magnus method with flow-symmetric coordinates, introduced in Example 3.1 is compared with the same scheme using coordinates centered at y_n [23]. As expected, the two Casimirs are rendered to machine accuracy for both the methods (right plots). In the upper plot to the left, the time-symmetric Magnus method shows very good behaviour on the Hamiltonian. The error in the Hamiltonian is confined within a band, whereas the standard choice of coordinates (centered at y_n) gives rise to a linearly increasing absolute error. Thus, the time-symmetric method yield very good behaviour in the retention of the heavy top Hamiltonian, as was also the case for the Euler equations describing a rigid body.

Rather surprisingly, as we can see in the lower left plot, time-symmetry of the numerical method also improves the conservation of the last non-Casimir first integral. The error of this first integral is also confined within a band for the time-symmetric method, just like for the error in the Hamiltonian. The standard method displays a linear increase in the absolute value of the error for the first integral.

In Figure 6 we observe a similar behaviour for the order-six Gauss–Legendre of the Munthe-Kaas type with flow-symmetric coordinates compared with the corresponding method with coordinates

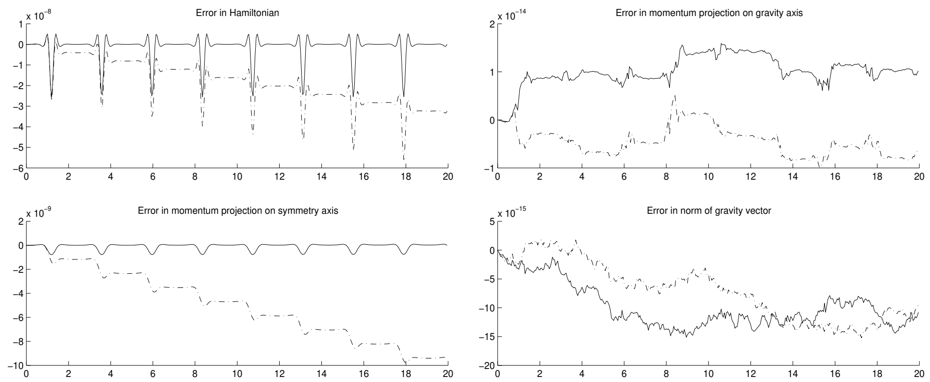

Figure 6: Order-six Gauss–Legendre RKMK method with flow-symmetric coordinates (solid line) compared with coordinates centered at y_n (dash-dotted line).

centered at y_n [16]. Similar improvements are observed with geodesic symmetric coordinates, and in general with all types of symmetric coordinates introduced in this paper.

4.3 A three-particle periodic Toda lattice

As a last example, we consider the three-particle periodic Toda lattice, i.e. three particles on a ring subject to a nearest-neighbour exponential potential, with Hamiltonian function

$$H(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2}(p_1^2 + p_2^2 + p_3^2) + \mathrm{e}^{-(q_2-q_1)} + \mathrm{e}^{-(q_3-q_2)} + \mathrm{e}^{-(q_1-q_3)} - 3, \tag{4.3}$$

an example that has been treated at length already in [23, 25] and references therein.

The above Toda lattices can be written in the Lax form

$$L' = [B(L), L], \qquad L(0) = L_0, \tag{4.4}$$

a matrix *isospectral flow*.[2] Specifically,

$$L = \begin{bmatrix} \beta_1 & \alpha_1 & \alpha_3 \\ \alpha_1 & \beta_2 & \alpha_2 \\ \alpha_3 & \alpha_2 & \beta_3 \end{bmatrix}, \quad B(L) = \begin{bmatrix} 0 & -\alpha_1 & \alpha_3 \\ \alpha_1 & 0 & -\alpha_2 \\ -\alpha_3 & \alpha_2 & 0 \end{bmatrix},$$

where the new variables are obtained by means of the formulae

$$\begin{aligned} \alpha_j &= \tfrac{1}{2}\mathrm{e}^{-(q_{j+1}-q_j)/2}, \qquad j = 1, 2, 3, \\ \beta_j &= \tfrac{1}{2}p_j, \end{aligned}$$

with the periodicity conditions

$$\alpha_{j+3} = \alpha_j, \quad \beta_{j+3} = \beta_j, \qquad j = 1, 2, 3,$$

[7, 21]. It is well known that classical numerical methods cannot retain isospectrality (see for instance [2, 25]). However, isospectrality can be recovered by solving numerically the orthogonal flow

$$y' = \gamma(y)y, \qquad y(t_n) = I, \quad t \in [t_n, t_n + h], \tag{4.5}$$

[2] I.e., the eigenvalues of the solution $L(t)$ do not change with time.

where $\gamma(y) = B(yL_ny^T)$, with either an orthogonal method or with a Lie-group scheme. Thus, if $L_n \approx L(t_n)$ is given and $y_{n+1} \approx y(t_{n+1})$ is an orthogonal matrix, we employ the similarity transformation

$$L_{n+1} = y_{n+1}L_ny_{n+1}^T,$$

to advance from L_n to $L_{n+1} \approx L(t_{n+1})$.

It was observed in [23, 25] that Lie-group methods with coordinates centered at y_n, applied to (4.5), in tandem with the similarity transformation $L_{n+1} = y_{n+1}L_ny_{n+1}^T$ were preserving isospectrality to machine accuracy. Also the Poincaré section produced by the methods were quite accurate. However, since the α_i's (and not the q_i's) were treated as independent variables, those methods were displaying a error in the lattice length that we define as

$$e_{\mathrm{latt}} = \prod_{j=1}^{3} \alpha_j - \frac{1}{8}$$

(note that e_{latt} should equal zero along the theoretical solution).

In Figure 7 we display the value of e_{latt} for Lie-group methods based on order-four Gauss–Legendre methods. For the numerical experiments, we have used $\boldsymbol{p}_0 = [1, 1, 0]^T$ and $\boldsymbol{q}_0 = [0, 0, 0]^T$ and stepsize $h = \frac{1}{10}$. In particular, the left plot corresponds to the order-four Magnus method in Example 3.1 (flow-symmetric coordinates) while the right plot represents e_{latt} for an order-four Magnus method, this time with coordinates centered at y_n [23, 25]. It is clear that, while the symmetric coordinates produce a result that oscillates periodically around zero, the non-symmetric coordinates produce a numerical approximation with an error that is increasing with time, wrongly representing the qualitative behaviour of the exact solution. Analogous results are obtained comparing flow and geodesic-midpoint based GLRK methods of the Munthe-Kaas type, introduced in Algorithms 2 and 3, with their non-symmetric counterparts [16].

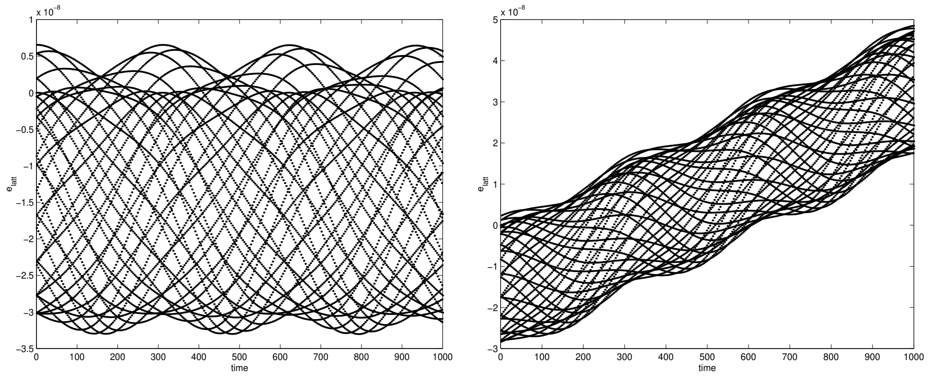

Figure 7: Lattice increment for the order-four Gauss–Legendre Magnus method with flow-symmetric coordinates, as in Example 3.1, (left plot) and non-symmetric coordinates centered at y_n, as in [23, 25], (right plot). The stepsize of integration is $h = 1/10$.

Figure 8 displays e_{latt} for an order-six Gauss–Legendre method of the Munthe-Kaas type with geodesic-symmetric coordinates (flow-symmetric coordinates produce an almost identical result) as in Algorithm 3.

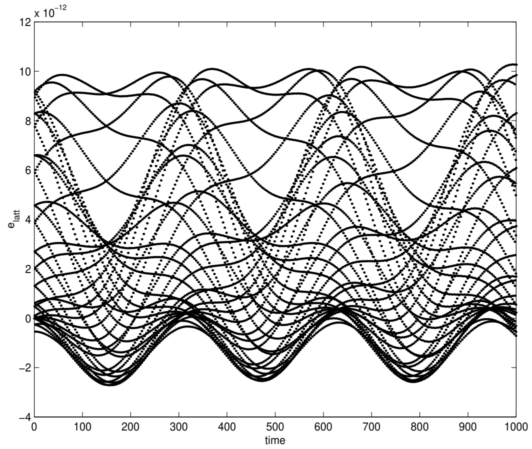

Figure 8: Lattice increment for the order-six Gauss–Legendre method of the Munthe-Kaas type with geodesic midpoint coordinates.

Conclusions

In this paper we have discussed Lie-group schemes and their dependence on the choice of the coordinate chart. It has been shown that the definition of adjoint of a method depends on the numerical method employed in the algebra and the coordinate map. We have introduced new coordinate maps that allow us to generate selfadjoint Lie-group schemes. Special mention deserve geodesic symmetric and flow symmetric coordinates, for their natural geometric interpretation.

The relation between selfadjointness and time-reversibility has been discussed in a number of numerical experiment: although selfadjointness implies time-reversibility for classical methods (note that classical Runge–Kutta methods are a special case of Lie-group methods, when $(\mathbb{R}^d, +)$ is chosen as the Lie group acting on $\mathbb{R}^d$ itself), this is not necessarily the case of generic Lie-group schemes. This suggests that in the general case of methods on nonlinear manifold, relations between global and local properties of numerical methods, like time-reversibility and selfadjointness, need further investigation.

5 Appendix

Theorem 5.1
Let $f : [0,T] \times \Omega \times \Omega \to \mathbb{R}^m$, with $\Omega \subset \mathbb{R}^m$. Assume that Ω is connected and compact. Assume that f is continuous and that there exist two strictly positive constants $\mathcal{L}_1, \mathcal{L}_2$, such that

$$\begin{aligned} \|f(t,y_1,z) - f(t,y_1,z)\| &\leq \mathcal{L}_1 \|y_1 - y_2\|, \\ \|f(t,y,z_1) - f(t,y,z_2)\| &\leq \mathcal{L}_2 \|z_1 - z_2\|, \end{aligned}$$

for all $y_1, y_2, z_1, z_2 \in \Omega$. Then a solution y to the differential equation

$$y' = f(t, y, y(cT)), \quad y(0) = y_0, \quad t \in [0,T], \quad c \in [0,1],$$

exists and is unique when $T < \frac{1}{\mathcal{L}_1} \log(1 + \frac{\mathcal{L}_1}{\mathcal{L}_2})$.

Proof. Let us consider the Picard iteration

$$y^{[k]'} = f(t, y^{[k]}, y^{[k-1]}(cT)), \quad y^{[k]}(0) = y_0.$$

We have

$$\begin{aligned} y^{[k]'} - y^{[k-1]'} &= f(t, y^{[k]}, y^{[k-1]}(cT)) - f(t, y^{[k-1]}, y^{[k-2]}(cT)) \\ &= f(t, y^{[k]}, y^{[k-1]}(cT)) - f(t, y^{[k]}, y^{[k-2]}(cT)) \\ &\quad + f(t, y^{[k]}, y^{[k-2]}(cT)) - f(t, y^{[k-1]}, y^{[k-2]}(cT)), \end{aligned}$$

hence, passing to the norm, using the triangle inequality and the two Lipschitz conditions,

$$\|y^{[k]'} - y^{[k-1]'}\| = \mathcal{L}_1\|y^{[k]} - y^{[k-1]}\| + \mathcal{L}_2\|y^{[k-1]}(cT)) - y^{[k-2]}(cT))\|.$$

Hence,

$$\begin{aligned} \|y^{[k]}(t) - y^{[k-1]}(t)\| &= \|\int_0^t (y^{[k]'}(s) - y^{[k-1]'}(s)\|)\,\mathrm{d}s \\ &\le \int_0^t \|y^{[k]'}(s) - y^{[k-1]'}(s)\|\,\mathrm{d}s \\ &\le \mathcal{L}_1 \int_0^t \|y^{[k]}(s) - y^{[k-1]}(s)\|\,\mathrm{d}s + \mathcal{L}_2 t\|y^{[k-1]}(cT)) - y^{[k-2]}(cT))\|. \end{aligned}$$

Set $e^{[k]} = \|y^{[k]} - y^{[k-1]}\|$. Then,

$$e^{[k]}(t) \le \mathcal{L}_2 t e^{[k-1]}(cT) + \mathcal{L}_1 \int_0^t e^{[k]}(s)\,\mathrm{d}s,$$

hence, by differentiation

$$e^{[k]'} \le \mathcal{L}_2 e^{[k-1]}(cT) + \mathcal{L}_1 e^{[k]}, \quad e^{[k]}(0) = 0,$$

which has solution

$$e^{[k]}(t) \le \frac{\mathcal{L}_1}{\mathcal{L}_2} e^{[k-1]}(cT)(\mathrm{e}^{\mathcal{L}_1 t} - 1) \le \frac{\mathcal{L}_1}{\mathcal{L}_2} \max_{s\in[0,T]} e^{[k-1]}(s)(\mathrm{e}^{\mathcal{L}_1 T} - 1).$$

Thus, if $\frac{\mathcal{L}_1}{\mathcal{L}_2}(\mathrm{e}^{\mathcal{L}_1 T} - 1) < 1$ or, equivalently, $T < \frac{1}{\mathcal{L}_1}\log(1 + \frac{\mathcal{L}_1}{\mathcal{L}_2})$, by the contraction mapping theorem, the Picard iteration converges and the given differential equation has a unique solution in $[0, T]$. □

Example 5.1
Consider the differential equation

$$y' = y + y(cT), \qquad y(0) = y_0.$$

The function $f(t, y, y(cT))$ is $f \equiv y + y(cT)$ with Lipschitz constants $\mathcal{L}_1 = \mathcal{L}_2 \equiv 1$. The solution of the above equation is given by

$$y(t) = \frac{y_0(2\mathrm{e}^t - \mathrm{e}^{cT})}{2 - \mathrm{e}^{cT}},$$

and it is not defined at $T = \frac{1}{c}\log 2$. Thus, when $c = 1$, we have $T < \log 2$, a condition showing that the bound on T obtained in Theorem 5.1 is sharp.

Theorem 5.2
For sufficiently small h, the differential equation

$$\sigma' = \mathrm{dexp}_\sigma^{-1}(\gamma(\exp(\sigma)\exp(-\theta_{h,n})y_n)), \qquad \sigma(t_n) = \theta_{h,n},$$

has a unique solution in $[t_n, t_n + h]$ when $\theta_{h,n}$ defines either a geodesic or a flow midpoint.

Proof. Without loss of generality, we can consider the differential equation

$$\sigma' = \mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}}(\gamma(\exp(\sigma-\sigma_{\frac12})\exp(\sigma_{\frac12})y_n)), \qquad \sigma(t_n)=0,$$

under the transformation $\sigma \to \sigma - \sigma_{\frac12}$, whereby $\sigma_{\frac12} = -\theta_{h,n}$, the latter denoting either a geodesic or a flow midpoint. In order to employ Theorem 5.1, we need to show that, denoting $f(t,\sigma,\sigma_{\frac12}) = \mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}}\gamma(C(\sigma,\sigma_{\frac12},y_n))$, the function f is Lipschitz with respect to its second and third argument, where we have denoted $C(\sigma,\sigma_{\frac12},y_n) = \exp(\sigma-\sigma_{\frac12})\exp(\sigma_{\frac12})y_n$. We have

$$\begin{aligned} f(t,\sigma,\sigma_{\frac12}) - f(t,\pi,\sigma_{\frac12}) &= \mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}}\gamma(C(\sigma,\sigma_{\frac12},y_n)) - \mathrm{dexp}^{-1}_{\pi-\sigma_{\frac12}}\gamma(C(\pi,\sigma_{\frac12},y_n)) \\ &= \mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}}(\gamma(C(\sigma,\sigma_{\frac12},y_n)) - \gamma(C(\pi,\sigma_{\frac12},y_n)) \\ &\quad + \mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}}\gamma(C(\pi,\sigma_{\frac12},y_n)) - \mathrm{dexp}^{-1}_{\pi-\sigma_{\frac12}}\gamma(C(\pi,\sigma_{\frac12},y_n)). \end{aligned}$$

Passing to the norm and making use of the triangle inequality,

$$\begin{aligned} \|f(t,\sigma,\sigma_{\frac12}) - f(t,\pi,\sigma_{\frac12})\| &\le \frac{\mu\|\sigma-\sigma_{\frac12}\|}{\mathrm{e}^{\mu\|\sigma-\sigma_{\frac12}\|}-1}\|\gamma(C(\sigma,\sigma_{\frac12},y_n)) - \gamma(C(\pi,\sigma_{\frac12},y_n))\| \\ &\quad + \|(\mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}} - \mathrm{dexp}^{-1}_{\pi-\sigma_{\frac12}})\gamma(C(\pi,\sigma_{\frac12},y_n))\|, \end{aligned}$$

whereby we have used a standard bound for the function dexp^{-1} (see, for instance [14]) and where μ is the *radius* of the algebra [12], such that $||[\eta,\theta]|| \le \mu\|\eta\|\|\theta\|$ for all $\eta,\theta \in \mathfrak{g}$. Note that, making use of the Baker–Campbell–Hausdorff formula [22], we have

$$\begin{aligned} \|\gamma(C(\sigma,\sigma_{\frac12},y_n)) - \gamma(C(\pi,\sigma_{\frac12},y_n))\| &= \|\gamma(\mathrm{e}^{\sigma-\sigma_{\frac12}}\mathrm{e}^{\sigma_{\frac12}}y_n) - \gamma(\mathrm{e}^{\pi-\sigma_{\frac12}}\mathrm{e}^{\sigma_{\frac12}}y_n\| \\ &\le \mathcal{L}_\gamma\|y_n\|\|\sigma-\pi\| + \text{h.o.t.}, \end{aligned}$$

where $\mathcal{L}_\gamma$ is the Lipschitz constant of the function γ (the higher order terms being the rest of a converging series), therefore we can bound

$$\|\gamma(C(\sigma,\sigma_{\frac12},y_n)) - \gamma(C(\pi,\sigma_{\frac12},y_n))\| \le \tilde{\mathcal{L}}_\gamma\|y_n\|\|\sigma-\pi\|.$$

However, for simplicity sake, we will write $\mathcal{L}_\gamma$ for $\tilde{\mathcal{L}}_\gamma$, hoping that this abuse of notation does not cause confusion to the reader. Moreover,

$$\begin{aligned} \|(\mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}} - \mathrm{dexp}^{-1}_{\pi-\sigma_{\frac12}})\gamma(C(\pi,\sigma_{\frac12},y_n))\| &\le \|\gamma(C(\pi,\sigma_{\frac12},y_n))\|\left|\frac{\mu\|\sigma-\sigma_{\frac12}\|}{\mathrm{e}^{\mu\|\sigma-\sigma_{\frac12}\|}-1} - \frac{\mu\|\pi-\sigma_{\frac12}\|}{\mathrm{e}^{\mu\|\pi-\sigma_{\frac12}\|}-1}\right| \\ &= \frac{\mu}{2}\|\gamma(C(\pi,\sigma_{\frac12},y_n))\|\|\sigma-\pi\| + \text{h.o.t.}, \end{aligned}$$

where we have made use of the inequality $||a|-|b|| \le |a-b|$ and again the higher order terms are rest of a converging series. Abusing notation, we write

$$\|(\mathrm{dexp}^{-1}_{\sigma-\sigma_{\frac12}} - \mathrm{dexp}^{-1}_{\pi-\sigma_{\frac12}})\gamma(C(\pi,\sigma_{\frac12},y_n))\| \le \frac{\mu}{2}\|\gamma\|\|\sigma-\pi\|,$$

hence we can set

$$\mathcal{L}_1 = \frac{\mu\|\sigma-\sigma_{\frac12}\|}{\mathrm{e}^{\mu\|\sigma-\sigma_{\frac12}\|}-1}\mathcal{L}_\gamma\|y_n\| + \frac{\mu}{2}\|\gamma\|.$$

A similar calculation yields

$$\|f(t,\sigma,\sigma_{\frac{1}{2}}) - f(t,\sigma,\pi_{\frac{1}{2}})\| \leq \left(\frac{\mu\|\sigma-\sigma_{\frac{1}{2}}\|}{\mathrm{e}^{\mu\|\sigma-\sigma_{\frac{1}{2}}\|}-1}\mathcal{L}_\gamma\|y_n\| + \frac{\mu}{2}\|\gamma\| \right) \|\sigma_{\frac{1}{2}} - \pi_{\frac{1}{2}}\|,$$

from which we can set

$$\mathcal{L}_2 = \frac{\mu\|\sigma-\sigma_{\frac{1}{2}}\|}{\mathrm{e}^{\mu\|\sigma-\sigma_{\frac{1}{2}}\|}-1}\mathcal{L}_\gamma\|y_n\| + \frac{\mu}{2}\|\gamma\| \equiv \mathcal{L}_1.$$

We can next apply Theorem 5.1 and deduce existence and uniqueness of the given differential equation provided that

$$h < \frac{1}{\mathcal{L}_1}\log 2.$$

□

Note that the largest stepsize h for convergence of the fixed point iterations is determined as

$$h_{\max} = \min\{\frac{1}{\mathcal{L}_1}\log 2, h_{\mathrm{BCH}}, h_{\mathrm{dexp}^{-1}}\},$$

whereby h_{BCH} denotes the radius of convergence of the BCH formula and $h_{\mathrm{dexp}^{-1}}$ denotes the radius of convergence of the function dexp^{-1}.

A similar result holds when we consider the Cayley transform and canonical coordinates of the second kind.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, 1965.

[2] M. P. Calvo, A. Iserles, and A. Zanna. Numerical Solution of Isospectral Flows. *Math. of Comp.*, 66:1461–1486, 1997.

[3] F. Diele, L. Lopez, and R. Peluso. The Cayley transform in the numerical solution of unitary differential systems. *Adv. Comput. Math.*, 8(4):317–334, 1998.

[4] K. Engø. On the construction of geometric integrators in the RKMK class. Technical Report No. 158, Department of Informatics, University of Bergen, Norway, 1998.

[5] K. Engø and S. Faltinsen. Numerical integration of Lie-Poisson systems while preserving coadjoint orbits, Casimirs and the energy. In preparation. 1999.

[6] S. Faltinsen. Backward error analysis for Lie-group methods. Technical Report NA1998/12, DAMTP, University of Cambridge, 1998.

[7] H. Flaschka. The Toda lattice I. *Phys. Rev.*, B9:1924–1925, 1974.

[8] E. Hairer. Backward analysis of numerical integrators and symplectic methods. *Ann. Numer. Math.*, 1:107–132, 1994.

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

[10] F. Hausdorff. Die symbolische Exponentialformel in der Gruppentheorie. *Beriche der Sächsischen Akademie der Wissenschaften (Math. Phys. Klasse)*, 58:19–48, 1906.

[11] A. Iserles and S. P. Nørsett. On the solution of differential equations in Lie groups. *Phil. Trans. R. Soc. Lond. A*, 357:983–1019, 1999.

[12] A. Iserles, S. P. Nørsett, and A. Rasmussen. On the reversibility of the canonical Magnus expansion. Technical Report NA1998/06, DAMTP, University of Cambridge, 1998.

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

[14] P. C. Moan. Efficient approximantion of Sturm–Liouville problems using Lie-group methods. Technical Report NA1998/11, DAMTP, 1998.

[15] J. Moser. *Stable and Random Motion in Dynamical Systems.* Princeton University Press, 1973.

[16] H. Munthe-Kaas. High Order Runge–Kutta Methods on Manifolds. *Journal of Applied Numer. Math.*, 29:115–127, 1999.

[17] H. Munthe-Kaas and B. Owren. Computations in a free Lie algebra. *Phil. Trans. R. Soc. Lond. A*, 357:957–982, 1999.

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

[19] B. Owren and A. Marthinsen. Integration methods based on canonical coordinates of the second kind. Technical Report Numerics No. 5/1999, Department of Mathematical Sciences, The Norwegian University of Science and Technology, Trondheim, Norway, 1999.

[20] A. M. Stuart and A. R. Humphries. *Dynamical Systems and Numerical Analysis.* Cambridge University Press, Cambridge, 1996.

[21] M. Toda. *Theory of Nonlinear Lattices.* Springer-Verlag, Berlin, 1981.

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

[23] A. Zanna. Collocation and relaxed collocation for the Fer and the Magnus expansions. Technical Report NA1997/17, DAMTP, University of Cambridge, 1997. To appear in *SIAM J. Num. Anal.*

[24] A. Zanna. The Fer expansion and time symmetry: a Strang-type approach. Technical Report NA1998/14, DAMTP, 1998.

[25] A. Zanna. *On the Numerical Solution of Isospectral Flows.* PhD thesis, Newnham College, University of Cambridge, 1998.