UNIVERSITY OF CAMBRIDGE

Numerical Analysis Reports

Collocation and relaxed collocation for the Fer and the Magnus expansions

Antonella Zanna

DAMTP NA1997/17
August 1997

Department of Applied Mathematics and Theoretical Physics
Silver Street
Cambridge
England CB3 9EW

Collocation and relaxed collocation for the Fer and the Magnus expansions

A. Zanna *

Abstract

We consider the Fer and the Magnus expansions for the numerical solution of the *nonlinear* matrix Lie-group ODE

$$y' = \gamma(t, y)y, \qquad y(0) = y_0,$$

where y evolves in a matrix Lie group G and $\gamma(t, y)$ is in the Lie algebra $\mathfrak{g}$. Departing from a geometrical approach, that distinguishes between those operations performed in the group and those performed in the tangent space, we construct Lie-group invariant methods based on collocation. We prove that, as long as the two expansions are correctly truncated, the collocation nodes $c_1, c_2, \ldots, c_\nu$ yield numerical methods whose order is the same as in the classical setting. We also relax the collocation conditions, thereby devising explicit methods of order three. To conclude, we discuss the proposed methods in two numerical experiments that arise in Hamiltonian mechanics and inverse eigenvalue problems, comparing the results with projection techniques.

1 Introduction

The subject matter of this paper is the construction of Lie-group numerical methods for the nonlinear ordinary differential equation

$$y' = \gamma(t, y)y, \quad y(0) = y_0, \tag{1.1}$$

where y lies in a matrix Lie group $G \subseteq \mathrm{GL}(\mathbb{F}, d)$ and $\gamma(t, y) \in \mathfrak{g}$, the Lie algebra of G. The set $\mathbb{F}$ stands for either $\mathbb{C}$ or $\mathbb{R}$. The restriction to matrix Lie groups is not severe, since Ado's Theorem guarantees that any analytical group is locally isomorphic to a matrix Lie group [35]. Furthermore, with opportune modifications our schemes can be extended to general Lie groups.

In particular we are concerned with the case when $\gamma(t, z) \in \mathfrak{g}$ only if $z \in G$, i.e. numerical methods that are *strongly* Lie-group invariant.

Our attention will be focused mainly on two expansions for the solution of (1.1), namely the *Fer* and the *Magnus expansions* [10, 23]. Such expansions were already considered by Zanna [36] and Iserles and Nørsett [17] respectively, for devising Lie-group numerical methods in the linear case, i.e. when the function γ in (1.1) is function of time only. In [36], we also extended the Fer expansion to the nonlinear problem (1.1), however the resulting methods were unsatisfactory and expensive, since lower order methods were recursively employed to obtain numerical approximations of the solution at some intermediate points.

To construct Lie-group methods, namely schemes that produce intrinsically a numerical approximation $y_n \in G$ without employing any projection technique, we depart from a geometrical description of Lie groups, identifying the internal operations in each space: for instance, in G we are allowed to proceed by multiplication, while it is forbidden to multiply elements of the algebra.

*Newnham College, University of Cambridge, Cambridge, England

However, the algebra $\mathfrak{g}$ is a vector space and we can add and commute elements. Moreover, there exists a natural mapping, the *exponential,* that allows us to move from the algebra to the group (see Fig. 1). Hence Lie-group numerical methods are defined in terms of: multiplications (in G); addition, commutation (in $\mathfrak{g}$); exponentiation (from $\mathfrak{g}$ to G). We assume, in the course of this paper, that these operations can be performed to machine accuracy.

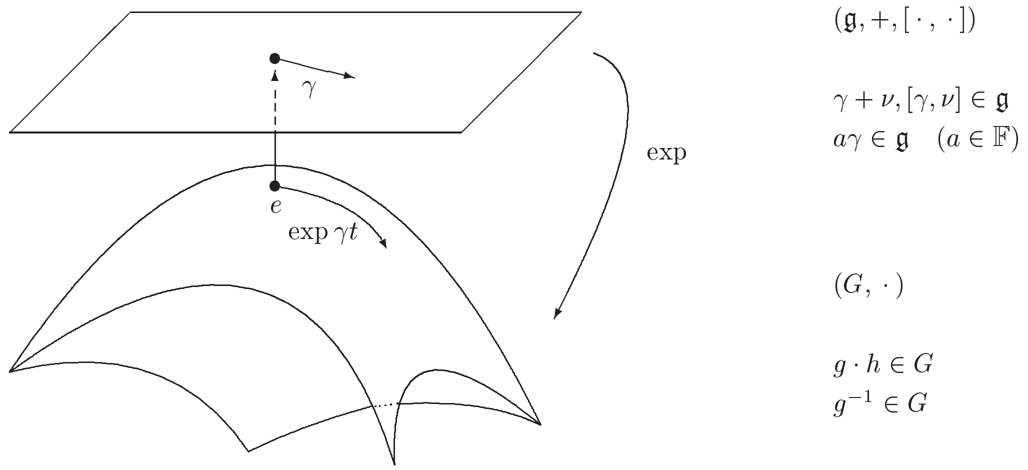


Figure 1: A naive description of a Lie group G and its Lie algebra $\mathfrak{g}$.

The paper is organized as follows. In Section 2 we present the Fer expansion and express the order analysis in terms of order trees, following the construction of Iserles and Nørsett [17].

In Section 3 we recall briefly the features of the Magnus expansion, described at greater length in [17].

In Section 4 we construct the Lie-group collocation methods based on the Fer and the Magnus expansion, and we prove that, provided that the expansions are appropriately truncated, the collocation methods have the same order of the corresponding classical collocation Runge–Kutta schemes.

In Section 5 we relax the collocation conditions. We show that it is possible to choose parameters to annihilate some coefficients in the expansion of the methods. Explicit third-order methods are presented as an example of this approach.

In Section 6 we compare some of the numerical methods proposed with projection schemes that employ QR factorization and polar decomposition. In the first test we consider a Hamiltonian-type problem, while in the second experiment we compare the performances of the schemes for an inverse eigenvalue problem which is instead of a gradient-flow type.

1.1 Lie groups and their general properties

A *Lie group* $(G, \cdot\,)$ is a smooth manifold that also possesses the algebraic structure of a group. In other words, for every $x, y \in G$, the multiplication map

$$m : G \times G \to G, \quad m(x, y) = x \cdot y,$$

and the inversion map,

$$i : G \to G, \quad i(y) = y^{-1},$$

such that $y \cdot y^{-1} = e$, the identity element of G, are smooth maps between manifolds [28].

Given a Lie group G, we denote by T_yG the tangent space to G at the point $y \in G$, while TG denotes the *tangent bundle,* i.e.

$$TG = \bigcup_{y\in G} T_yG.$$

The tangent space at the identity e is of particular relevance and it is called the *Lie algebra* of G. Its support is usually denoted with the corresponding lower case gothic letter,

$$\mathfrak{g} = T_eG.$$

Given a Lie group G, we consider the *right* and *left* multiplications,

$$\begin{aligned} R_y(x) &= x \cdot y, \\ L_y(x) &= y \cdot x, \end{aligned} \qquad x, y \in G, \tag{1.2}$$

which are also called *right* and *left* translations. These maps induce maps on the corresponding tangent spaces; the induced maps, denoted by R'_y, L'_y, are defined as follows:

$$R'_y(\,\cdot\,) : \mathfrak{g} \to T_yG,$$

so that, given $\gamma \in \mathfrak{g}$,

$$R'_y(\gamma) = \frac{\mathrm{d}}{\mathrm{d}t} R_y\big(g(t)\big)\Big|_{t=0}, \tag{1.3}$$

where $g(t) = e+\gamma t+\mathcal{O}\big(t^2\big)$ is a curve in G starting at the identity with tangent γ. The map $Y_\gamma(y) = R'_y(\gamma)$, as a function of $y \in G$, is called *right invariant vector field* since it is invariant under the right translation. The construction is analogous for the left invariant vector field $X_\gamma(y) = L'_y(\gamma)$.

For any Lie group G, the exponential map

$$\exp : \mathfrak{g} \to G,$$

is defined as

$$\exp(\gamma) = y(1), \qquad \gamma \in \mathfrak{g}, \tag{1.4}$$

where y is the integral curve of

$$y' = R'_y(\gamma), \qquad y(0) = e. \tag{1.5}$$

It turns out that the group multiplication of G induces an important structure on the Lie algebra $\mathfrak{g}$. Let $\gamma, \nu \in \mathfrak{g}$ and let $y(t), x(s)$ be two curves in G of the form

$$y(t) = e + \gamma t + \mathcal{O}\big(t^2\big)\,, \qquad x(s) = e + \nu s + \mathcal{O}\big(s^2\big)\,.$$

Then

$$[\gamma, \nu] = \frac{\partial^2}{\partial t \partial s} y(t)x(s)y^{-1}(t)\Big|_{t=s=0}, \tag{1.6}$$

defines a *commutator* or a *Lie-bracket.* The operation (1.6),

$$[\,\cdot\,,\,\cdot\,] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}, \tag{1.7}$$

is bilinear and skew-symmetric, and obeys the *Jacobi identity,*

$$\big[[\gamma, \nu], \eta\big] + \big[[\nu, \eta], \gamma\big] + \big[[\eta, \gamma], \nu\big] = 0, \qquad \forall \gamma, \nu, \eta \in \mathfrak{g}. \tag{1.8}$$

For a fixed $\gamma \in \mathfrak{g}$, we also consider the function $\mathrm{ad}_\gamma : \mathfrak{g} \to \mathfrak{g}$, defined as

$$\begin{aligned} &\mathrm{ad}^0_\gamma(\nu) = \nu, \\ &\mathrm{ad}^i_\gamma(\nu) = \mathrm{ad}_\gamma \circ \mathrm{ad}^{i-1}_\gamma(\nu) = [\gamma, \mathrm{ad}^{i-1}_\gamma(\nu)], \qquad i = 1, 2, \ldots, \end{aligned} \qquad \nu \in \mathfrak{g}, \tag{1.9}$$

the *adjoint representation,* which is linear in the argument ν [35]. Note that $\mathrm{ad}_\gamma(\nu) = [\gamma, \nu]$.

Since we consider matrix Lie groups, the group multiplication is simply the usual matrix multiplication, the Lie-bracket the usual commutator between matrices

$$[\gamma, \nu] = \gamma\nu - \nu\gamma, \tag{1.10}$$

and the exponential the standard matrix exponential,

$$\exp(\gamma) = \sum_{k=0}^{\infty} \frac{\gamma^k}{k!}. \tag{1.11}$$

However, with appropriate modification, the results of this chapter apply to general Lie-groups.

1.2 Ordinary Differential Equations on Lie groups

Since Lie groups are smooth manifolds, it makes sense to consider ODEs on G. We recall that the tangent bundle TG of a Lie group is *trivial* [29], which means that $TG \cong \mathfrak{g} \times G$. Furthermore,

$$\gamma \in T_y G \mapsto R'_{y^{-1}}\gamma \in \mathfrak{g}$$

is an isomorphism between the tangent space at y and at the identity. Thus, every ordinary differential equation on G can be written in the right invariant form

$$\begin{aligned} y' &= R'_y\big(\gamma(t, y)\big), \qquad \gamma(t, y) \in \mathfrak{g}, \\ y(0) &= y_0, \end{aligned} \tag{1.12}$$

(or, equivalently, in a left invariant form by means of L'_y), to which we will continuously refer as a *Lie-group ODE.* In the matrix case, the differential equation (1.12) becomes simply

$$\begin{aligned} y' &= \gamma(t, y)y, \qquad \gamma(t, y) \in \mathfrak{g}, \\ y(0) &= y_0. \end{aligned} \tag{1.13}$$

When solving numerically ODEs on Lie groups, we usually assume that the Lie group G is a submanifold of the Euclidean space. In that case, also $\mathfrak{g}$ is a subspace of the Euclidean space, so that $\gamma(t, \cdot\,)$ may be well defined outside $\mathfrak{g}$. In particular, we are especially interested to the case when

$$\gamma(t, y) \in \mathfrak{g} \Longleftrightarrow y \in G, \tag{1.14}$$

and, unless otherwise specified, we will assume that (1.14) is true.

1.3 Some recurring examples

Example 1.1 [$(\mathbb{R}^d, +)$] The vector space $\mathbb{R}^d$ with the usual $+$ constitutes a Lie group. The identity of the group is the null vector and the left and right translations are $R_y(x) = x+y = y+x = L_y(x)$, since the $+$ is a commutative operation. The Lie algebra of $\mathbb{R}^d$ is $\mathbb{R}^d$ itself and the commutator $[\gamma, \nu] = 0$, for all $\gamma, \nu \in \mathbb{R}^d$, since the $+$ is commutative. Note that

$$R'_y(\gamma) = \frac{\mathrm{d}}{\mathrm{d}t} R_y\big(g(t)\big)\Big|_{t=0} = \frac{\mathrm{d}}{\mathrm{d}t}(0 + \gamma t + \mathcal{O}(t^2) + y)\Big|_{t=0} = \gamma,$$

therefore the ODE formula (1.12) becomes simply

$$y' = \gamma(t, y)$$

(which is the same as $y' = f(t, y)$, the usual ODE formulation in $\mathbb{R}^d$). Thus, to evaluate the exponential function, we consider the ODE

$$y' = \gamma, \qquad y_0 = 0,$$

whose solution is

$$y(t) = \gamma t.$$

Hence, according to the formula (1.5),

$$\exp(\gamma) = y(1) = \gamma,$$

i.e. the exponential coincides with the identity function. $\diamond$

Example 1.2 [The general linear group, $(\mathrm{GL}(\mathbb{F}, d), \cdot)$] Let $\mathbb{F}^{d\times d}$ be the set of all $d \times d$ matrices with entries in $\mathbb{F}$. The set $\mathrm{GL}(\mathbb{F}, d)$ of all invertible $d \times d$ matrices, endowed with the usual matrix multiplication, possesses the structure of a Lie group and is called the *general linear group,*

$$\mathrm{GL}(\mathbb{F}, d) = \{y \in \mathbb{F}^{d\times d} \mid \det y \neq 0\}.$$

The identity element is $e = I$, the $d \times d$ identity matrix. Given two invertible matrices, x, y, the right multiplication is $R_y(x) = xy$. The Lie algebra of $(\mathrm{GL}(\mathbb{F}, d), \cdot)$ is $(\mathfrak{gl}(\mathbb{F}, d), +, [\cdot, \cdot])$, the set of all $d \times d$ matrices,

$$\mathfrak{g}(\mathbb{F}, d) = \mathbb{F}^{d\times d}.$$

The Lie-bracket $[\gamma, \nu] = \gamma\nu - \nu\gamma$ is the usual commutator of matrices. Ordinary differential equations on $\mathrm{GL}(\mathbb{F}, d)$ have the form

$$y' = \gamma(t, y)y, \qquad \gamma(t, y) \in \mathbb{F}^{d\times d}.$$

To find the exponential map, we consider

$$y' = \gamma y, \qquad y_0 = I,$$

whose solution is

$$y(t) = \exp(\gamma t) = \sum_{k=0}^{\infty} \frac{(\gamma t)^k}{k!}.$$

Hence,

$$\exp(\gamma) = \sum_{k=0}^{\infty} \frac{\gamma^k}{k!},$$

the classical matrix exponential. In particular, all the sub-groups (sub-algebras resp.) of $\mathrm{GL}(\mathbb{F}, d)$ ($\mathfrak{gl}(\mathbb{F}, d)$ resp.) will inherit the same multiplication, commutator and exponential map. $\diamond$

Some relevant matrix Lie groups are: $\mathrm{SO}(d)$, the orthogonal group of matrices with determinant equal to one, with algebra $\mathfrak{so}(d)$ of skew-symmetric matrices, $\mathrm{U}(d)$, the unitary group, with algebra $\mathfrak{su}(d)$ of skew-hermitian matrices, and $\mathrm{SL}(d)$, the special linear group of matrices with unit determinant, with algebra $\mathfrak{sl}(d)$ of traceless matrices.

2 The Fer expansion

2.1 The Fer expansion and the method of iterated commutators

The Fer expansion was introduced by Fer [10] to solve the linear matrix differential equation

$$y' = \gamma(t)y, \qquad y(0) = y_0. \tag{2.1}$$

The same construction was essentially obtained by Iserles (the method of *iterated commutators,* [14]) whereby a very similar approach was followed to introduce numerical methods for linear variable-coefficients systems of ODEs.

Neither Fer nor Iserles were aware of the fact that this expansions, exploiting essentially Lie-groups and Lie-algebra operations, produces Lie-group approximation. However, this approach is widely used to find *analytical* (opposed to numerical) solutions of Lie-group differential equations (for instance see [5]). Zanna proposed in [36] to use Iserles' schemes for the numerical solution of Lie-group equations, but the approach that we present in this paper is different, and it is inspired by the tree analysis of Iserles and Nørsett for the Magnus expansion [17].

Going back to (2.1), both Fer and Iserles showed that the solution $y(t)$ of (2.1) can be expressed as an infinite product of exponentials,

$$y(t) = \lim_{m\to\infty} \mathrm{e}^{\sigma_0(t)} \cdots \mathrm{e}^{\sigma_m(t)} y_0, \tag{2.2}$$

where the matrix functions $\sigma_k(t)$ are obtained in an iterative manner. First of all, we let $\gamma_0(t) = \gamma(t)$, we evaluate

$$\sigma_0(t) = \int_0^t \gamma_0(\tau)\,\mathrm{d}\tau, \tag{2.3}$$

and construct the sequence of commutators

$$\eta_{0,0} = \gamma_0(t), \qquad \eta_{0,i+1} = [\eta_{0,i}, \sigma_0], \quad i \geq 0.$$

Next we form the matrix function

$$\gamma_1(t) = \sum_{i=1}^{\infty} \frac{i}{(i+1)!}\eta_{0,i}(t)$$

and let

$$\sigma_1(t) = \int_0^t \gamma_1(\tau)\,\mathrm{d}\tau.$$

The procedure is hence repeated. Once the functions $\gamma_k(t)$ and $\sigma_k(t)$ are known, we let

$$\gamma_{k+1}(t) = \sum_{i=1}^{\infty} \frac{i}{(i+1)!}\eta_{k,i}(t), \tag{2.4}$$

where

$$\eta_{k,0} = \gamma_k(t), \qquad \eta_{k,i+1} = [\eta_{k,i}, \sigma_k], \quad i \geq 0. \tag{2.5}$$

and finally

$$\sigma_{k+1}(t) = \int_0^t \gamma_{k+1}(\tau)\,\mathrm{d}\tau. \tag{2.6}$$

In passing, we mention that the *iterated commutators* in (2.5) can be conveniently written in a classical Lie-algebra notation as

$$\eta_{k,i} = (-1)^i \mathrm{ad}^i_{\sigma_k}(\gamma_k), \tag{2.7}$$

$\gamma_0(h)=\mathcal{O}(1)$	$\gamma_1(h)=\mathcal{O}(h^2)$	$\gamma_2(h)=\mathcal{O}(h^6)$	$\gamma_3(h)=\mathcal{O}(h^{14})$	$\ldots$
$\sigma_0(h)=\mathcal{O}(h)$	$\sigma_1(h)=\mathcal{O}(h^3)$	$\sigma_2(h)=\mathcal{O}(h^7)$	$\sigma_3(h)=\mathcal{O}(h^{15})$	
$\eta_{0,1}(h)=\mathcal{O}(h^2)$	$\eta_{1,1}(h)=\mathcal{O}(h^6)$	$\eta_{2,1}(h)=\mathcal{O}(h^{14})$	$\vdots$	
$\eta_{0,2}(h)=\mathcal{O}(h^3)$	$\underline{\eta_{1,2}(h)=\mathcal{O}(h^9)}$	$\eta_{2,2}(h)=\mathcal{O}(h^{21})$		
$\eta_{0,3}(h)=\mathcal{O}(h^4)$	$\eta_{1,3}(h)=\mathcal{O}(h^{12})$	$\vdots$		
$\vdots$	$\vdots$			
$\underline{\eta_{0,8}(h)=\mathcal{O}(h^9)}$				
$\vdots$				

Table 1: First functions and their orders for the method of iterated commutators.

where $\mathrm{ad}_u(\,\cdot\,):\mathfrak{g}\to\mathfrak{g}, u\in\mathfrak{g}$, is the *adjoint* operator, whose powers are evaluated as

$$\mathrm{ad}_u^0(v)=u, \qquad \mathrm{ad}_u^{i+1}(v)=\mathrm{ad}_u\circ\mathrm{ad}_u^i(v)=[u,\mathrm{ad}_u^i(v)], \quad i=0,1,2,\ldots, \tag{2.8}$$

where $u,v\in\mathfrak{g}$ [35].

Fer showed that, if we truncate (2.2) to an integer $M\geq 0$, then the order of the approximation is $p=2^{M+2}-2$. Iserles showed that the order p of approximation of the truncated expansion is retained when we replace the integrals in (2.6) and the infinite series in (2.4) by finite sums and quadrature formulae of appropriate order. We remind the reader that, as already mentioned in [36], the Fer expansion (initially devised for linear problems) holds also in the nonlinear case. The same is true for the Magnus expansion.

Although the method of iterated commutators might appear a strange and unconventional way to solve an arbitrary linear differential equation, the algorithm is based on *elementary operations* on Lie groups and Lie algebras: the adjoint map, ad_u, maps $\mathfrak{g}$ onto $\mathfrak{g}$, integrals of matrices in $\mathfrak{g}$ are elements of $\mathfrak{g}$, the exponential is a map from the algebra $\mathfrak{g}$ to the group G and finally the product of exponentials and Lie-group elements is an element of G.

Various numerical methods of different orders can be obtained by truncating the infinite series (2.4) and considering a certain number M of exponentials. For each step with stepsize h, one has $\gamma_k(h)=\mathcal{O}(h^s), s=2^{k+1}-2$ and $\sigma_k(h)=\mathcal{O}(h^{s+1})$, being the integral of γ_k in the interval $[0,h]$. Therefore $\eta_{k,1}(h)=[\gamma_k(h),\sigma_k(h)]$ has order $2s+2$ since the first terms in the Taylor expansions of γ_k and σ_k commute. In general it is possible to show (cf. [14]) that

$$\eta_{k,i}(h)=\mathcal{O}\left(h^{(i+1)(s+1)}\right), \quad s=2^{k+1}-2. \tag{2.9}$$

Hence, for a method of order p, we choose the index M such that $M=\lfloor\log_2(p+1)/2\rfloor$, and, for each $0\leq k\leq M-1$ we evaluate i_k commutators, $\eta_{k,j}, 1\leq j\leq i_k$, where i_k is the first integer such that

$$(i_k+1)(2^{k+1}-1)\geq p-1.$$

Since the commutators are integrated over an interval of length h, multiplication by h ensures that we match powers of h up to order p and the error due to the truncation is subsumed in the $\mathcal{O}(h^{p+1})$ term.

Some of the above functions and their orders for a step with stepsize h are tabulated in Table 1. For instance, for order $p=10$, we have $M=2$ (three exponential factors). To construct $\gamma_{k+1}(h), k=0,1$, we consider in the k-th column all the commutators up to the underlined one (inclusive). Thus i_0 and i_1 equal 8 and 2 respectively.

The definition of the numerical method is completed by specifying the quadrature formulae to integrate the functions $\gamma_k(h)$. One may choose different rules for each value of k: Popular choices, useful especially in a linear contest, are Gaussian or Radau points (cf. [14, 36]). Other variants [36] can be obtained by replacing the product of exponentials with a single exponential, by means of the Baker–Campbell–Hausdorff formula [35]. However, in this paper, we will use one single quadrature rule for all the integrals, since this leads to a saving in function evaluations in the nonlinear case.

To conclude this section, we mention that the method of iterated commutators is related to the *Lie method of reduction* that is used to find solutions of equations of Lie-type by successive analytical quadratures. More about this issue can be found in [38].

2.2 Order trees for iterated commutators

In this section we apply the tree analysis of Iserles and Nørsett [17] to the Fer expansion. Our motivation is that all the terms appearing in the Fer expansion, i.e. commutators and integrals, can be expressed in terms of commutators and integrals of the matrix function $\gamma_0(t) = \gamma(t)$. Also, this approach displays at a glance the integrals and the commutators required without specifying for the time being the quadrature formula used.

The order trees are bi-coloured and are constructed according to the following rules:

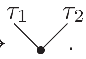
1. The matrix function $\gamma(t)$ is associated with the black vertex $\bullet$,

$$\gamma \quad \to \quad \bullet.$$

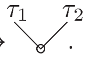
2. Given a tree τ, integration of τ corresponds to appending a white vertex to the tree τ,

$\int \tau \quad \to$  .

3. Given two trees, τ_1 and τ_2, such that $\tau_i \neq \tau_j$, commutation of τ_1 with τ_2 corresponds to joining the two trees with a black vertex,

$[\tau_1, \tau_2] \quad \to$  .

4. Given τ_1 and τ_2, with $\tau_2 = \int \tau_1$, then

$[\tau_1, \tau_2] \quad \to$  .

We remark that the trees are planar, therefore even though two trees might be mirror image of each other, they do represent different conditions.

With these simple rules, it is easy to write the order conditions in terms of trees, since the order of the tree can be evinced counting the number of white vertices. We construct all the order trees up to order $p = 8$. The procedure generalizes in a very similar fashion for the higher order terms.

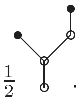
- Order $p = 2$. We construct the matrix $\sigma_0(h) = \int_0^h \gamma(t)\,\mathrm{d}t$ which is associated with the tree

 . (2.10)

- Order $p = 3$. Besides of the matrix σ_0, which is associated with the tree in (2.10), we need to consider the matrix function σ_1, which is appropriately truncated to order $p = 3$,

$$\sigma_1(h) = \tfrac{1}{2}\int_0^h [\gamma(t), \sigma_0(t)]\,\mathrm{d}t = \tfrac{1}{2}\int_0^h\int_0^t [\gamma(t), \gamma(s)]\,\mathrm{d}s\,\mathrm{d}t.$$

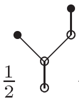
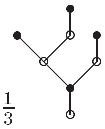
Following our rules, the above matrix is associated to the binary tree

$\frac{1}{2}$  . (2.11)

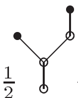
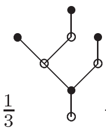
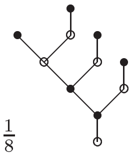
- Order $p = 4$. Besides of σ_0, we have

$$
\begin{aligned}
\sigma_1(h) &= \tfrac{1}{2}\int_0^h [\gamma(t), \sigma_0(t)]\,\mathrm{d}t + \tfrac{1}{3}\int_0^h [[\gamma(t), \sigma_0(t)], \sigma_0(t)]\,\mathrm{d}t \\
&= \tfrac{1}{2}\int_0^h\int_0^t [\gamma(t), \gamma(s)]\,\mathrm{d}s\,\mathrm{d}t + \tfrac{1}{3}\int_0^h\int_0^t\int_0^t [[\gamma(t), \gamma(s)], \gamma(u)]\,\mathrm{d}u\,\mathrm{d}s\,\mathrm{d}t,
\end{aligned}
$$

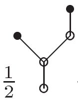
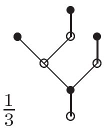
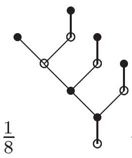
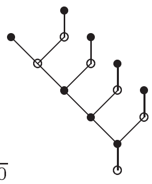
corresponding to the tree

$\frac{1}{2}$  $+\frac{1}{3}$  (2.12)

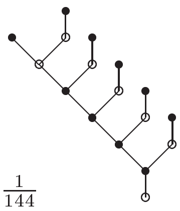
- Order $p = 5$. Besides σ_0, we have the following combination for the function σ_1,

$\frac{1}{2}$  $+\frac{1}{3}$  $+\frac{1}{8}$  (2.13)

- Order $p = 6$. This is the highest order that can be achieved with two single exponentials in the Fer expansion. Besides σ_0, we have the following tree for the function $\sigma_1(h)$,

$\frac{1}{2}$  $+\frac{1}{3}$  $+\frac{1}{8}$  $+\frac{1}{30}$  (2.14)

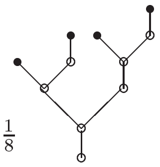
- Order $p = 7$. This is the first case requiring three exponentials. We focus our attention to $\sigma_1(h)$ and $\sigma_2(h)$. For the correct order in $\sigma_1(h)$ we have to add an extra tree to (2.14),

$\frac{1}{144}$  (2.15)

For σ_2, it is sufficient to consider a single commutator. We have

$$\sigma_2(h) = \tfrac{1}{2}\int_0^h [\gamma_1(t), \sigma_1(t)]\,\mathrm{d}t.$$

However, the above expression for $\sigma_2(h)$ can be simplified in order to contain the only tree relevant for order seven. This results in

$\frac{1}{8}$  (2.16)

We remark that, up to order $p = 6$, the order conditions are represented by *tall trees* only and each tree represents a separate order condition, while more order conditions arise when $p \geq 7$, which is also the first case to require three exponentials. See [37] for further order trees.

3 The Magnus expansion

The Fer expansion expresses the solution of (1.1) in terms of the product of infinite 'simple' exponentials. The Magnus expansion, instead, consists of writing the solution $y(t)$ of (1.1) as the exponential of a single and possibly more complicated argument,

$$y(t) = \mathrm{e}^{\sigma(t)} y_0. \tag{3.17}$$

The expansion of the first terms of σ was derived by Magnus in [23]. However, the algorithmic expansion of σ was derived by Iserles and Nørsett, who have also presented the convergence conditions [17].

3.1 Order trees for the Magnus expansion

We consider solutions of (1.1) of the form (3.17), and we wish to identify the relation between σ and the function $\gamma(t, y(t))$.

Differentiating (3.17) with respect to time t, we have

$$y' = \mathrm{dexp}_\sigma(\sigma')\mathrm{e}^{\sigma(t)} y_0 = \mathrm{dexp}_\sigma(\sigma') y, \tag{3.18}$$

where the function dexp is the related to the *derivative* of the exponential function by the formula

$$\mathrm{dexp} : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}, \quad \mathrm{dexp}_\sigma(\sigma') = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} \mathrm{ad}_\sigma^k(\sigma') \tag{3.19}$$

[35].

Comparing (3.18) with (1.1), we find that σ obeys the *implicit* differential equation

$$\mathrm{dexp}_\sigma(\sigma') = \gamma(t, y(t)). \tag{3.20}$$

A general expansion for σ' is obtained by inverting dexp_σ,

$$\sigma' = \mathrm{dexp}_\sigma^{-1}(\gamma(t, y(t))), \tag{3.21}$$

where

$$\mathrm{dexp}_u^{-1}(v) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \mathrm{ad}_u^k(v), \tag{3.22}$$

the B_k's being the Bernoulli numbers (cf. [2], p. 804). The expression for σ comes by integration of (3.22). The terms in σ are obtained by repeated commutation and integration of the function γ, and they are represented by trees in the same fashion already explained for the Fer expansion. For order greater than three, the order conditions of the Magnus expansion are generally represented by a collection of order trees, not all of them independent.

Table 2 illustrates the trees and relative coefficients to order up to five.[1] For higher order conditions, we refer the reader to [17].

4 Collocation for the Fer and the Magnus expansions

4.1 Collocation

Consider the differential equation

$$y' = f(t, y). \tag{4.1}$$

Suppose that the approximation $y_n \approx y(t_n)$ is known. We choose distinct nodes, $c_1, \ldots, c_\nu \in [0, 1]$, and consider a polynomial $u(t) \in \Pi_\nu$, the set of all the polynomials of degree ν, that obeys the differential equation exactly at $t_n + c_1 h, \ldots, t_n + c_\nu h$, the *collocation points.* Then the approximation for y_{n+1} is chosen as the value of the interpolating polynomial at $t_{n+1} = t_n + h$. In other words,

$$\begin{aligned} u(t_n) &= y_n, \\ u'(t_n + c_i h) &= f\big(t_n + c_i h, u(t_n + c_i h)\big), \quad i = 1, \ldots, \nu, \\ y_{n+1} &= u(t_n + h). \end{aligned} \tag{4.2}$$

It is well known [12] that there is an intimate relation between collocation methods and some implicit RK schemes. In effect, a collocation scheme corresponding to the nodes $c_1, \ldots, c_\nu$, is equivalent to the implicit RK

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array}, \tag{4.3}$$

where the coefficients a^i_j of the matrix A and the b_i's of the vector $\boldsymbol{b}$ are evaluated according to the formulae

$$\left.\begin{aligned} a^i_j &= \int_0^{c_i} L_j(t)\,\mathrm{d}t, \\ b_i &= \int_0^1 L_i(t)\,\mathrm{d}t, \end{aligned}\right. \qquad i, j = 1, \ldots, \nu, \tag{4.4}$$

respectively, where $L_i(t)$ is the ith Lagrangian cardinal polynomial,

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

On the other hand, not each implicit RK scheme is equivalent to a collocation method. However, all RK methods, both implicit and explicit, can be obtained by means of *perturbed collocation* [27].

It is instructive to recall the proof of the theorem that links collocation and RK schemes [12]. Without loss of generality, assume that (4.1) is autonomous and suppose that we approximate its exact solution

$$y(t_n + h) = y(t_n) + \int_{t_n}^{t_n + h} f\big(y(t)\big)\,\mathrm{d}t \tag{4.5}$$

[1] The notion of order used in this paper is somewhat different from the notion of order in [17] Here by 'order p' of a tree we mean that this tree is a $\mathcal{O}(h^p)$ term, while in [17] it means that the tree has p vertices.

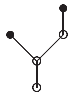
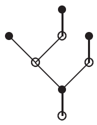
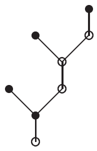
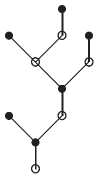
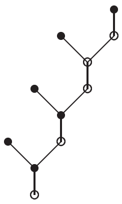
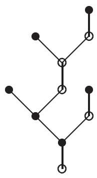
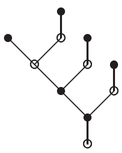
Order	Expression	Tree	Coefficient
1	$\int \gamma$		1
3	$\int [\gamma, \int \gamma]$		$\frac{1}{2}$
4	$\int [[\gamma, \int \gamma], \int \gamma]$		$\frac{1}{12}$
	$\int [\gamma, \int [\gamma, \int \gamma]]$		$\frac{1}{4}$
5	$\int [\gamma, \int [[\gamma, \int \gamma], \int \gamma]]$		$\frac{1}{24}$
	$\int [\gamma, \int [\gamma, \int [\gamma, \int \gamma]]]$		$\frac{1}{8}$
	$\int [[\gamma, \int [\gamma, \int \gamma]], \int \gamma]$		$\frac{1}{24}$
	$\int [[[\gamma, \int \gamma], \int \gamma], \int \gamma]$		0

Table 2: First order trees, their expression and coefficient for the Magnus expansion.

by replacing the integral in (4.5) with a quadrature formula. Since the integration is performed with respect to the variable t, we may fix some nodes of integration, say $c_1, \ldots, c_\nu$, and use the quadrature rule

$$\int_{t_n}^{t_n+h} f\big(y(t)\big)\,\mathrm{d}t \approx h\sum_{i=1}^{\nu} b_i f_i, \tag{4.6}$$

where the f_i are some approximations for the function values $f(y(t_n+c_ih))$. If such approximations are available (as we assume for the time being) then we can use such function values to construct an interpolating polynomial to the function f,

$$f\big(y(t)\big) \approx \sum_{i=1}^{\nu} L_i\left(\frac{t-t_n}{h}\right) f_i,$$

and the weights b_i, which have been defined in (4.4), are easily found by integrating the cardinal polynomials. On the other hand, we employ the same procedure to evaluate the Y_i's, $Y_i \approx y(t_n + c_ih)$, and let $f_i = f(Y_i)$. Note that $f_i \approx f(y(t_n + c_ih))$. Therefore,

$$Y_i = y_n + \int_{t_n}^{t_n+c_ih} \sum_{j=1}^{\nu} L_j\left(\frac{t-t_n}{h}\right) f_j\,\mathrm{d}t = y_n + h\sum_{j=1}^{\nu} a^i_j f_j, \tag{4.7}$$

where the a^i_j are the coefficients in (4.4).

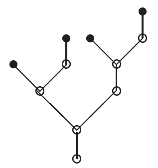
The procedure extends to Lie groups as follows: consider the differential equation

$$y' = \gamma(y)y, \qquad y \in G, \quad \gamma(y) \in \mathfrak{g}, \tag{4.8}$$

and suppose that some approximations $\gamma_i \approx \gamma(y(t_n + c_ih))$ are known. Then we approximate the function $\gamma(y)$ with its Lagrangian interpolating polynomial at the nodes $c_1, \ldots, c_\nu$,

$$\gamma\big(y(t)\big) \approx \sum_{i=1}^{\nu} L_i\left(\frac{t-t_n}{h}\right)\gamma_i. \tag{4.9}$$

At this point, we insert (4.9) into our favourite expansion, say the Fer expansion of the iterated commutators or the Magnus series. This expansion is hence employed to approximate the unknown values $y(t_n+c_ih)$ in the Lie group as for the classical collocation. However, for Lie-group methods, we recover not just the weights and the a^i_j's corresponding to collocation RK, but also some extra weights and coefficients which arise from the order trees for the Fer expansion (as in section 2.2) and for the Magnus series (as in [17]). In more details, recall that the terms featuring in the expansions are associated with order trees. For instance, the tree



represents (after the substitution of $\gamma(y)$ with its the Lagrangian interpolant)

$$\sum_{j,k,\ell,m=1}^{\nu} \Big(\int_0^{\{c_i,1\}} L_j(s)\int_0^s\int_0^s L_k(t)L_\ell(u)\int_0^u L_m(v)\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}t\,\mathrm{d}s\Big)\,[[\gamma_j,\gamma_k],[\gamma_\ell,\gamma_m]], \tag{4.10}$$

where we take c_i for the internal stage Y_i and 1 for y_{n+1} in the first integral. We would like to replace (4.10) by a term of the form

$$\sum_{j,k,\ell,m=1}^{\nu} a^i_{j,k,\ell,m}\,[[\gamma_j,\gamma_k],[\gamma_\ell,\gamma_m]],$$

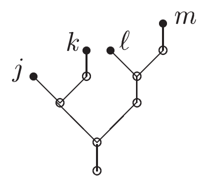
in the evaluation of the intermediate stages Y_i, or by

$$\sum_{j,k,\ell,m=1}^{\nu} b_{j,k,\ell,m} \left[[\gamma_j, \gamma_k], [\gamma_\ell, \gamma_m]\right],$$

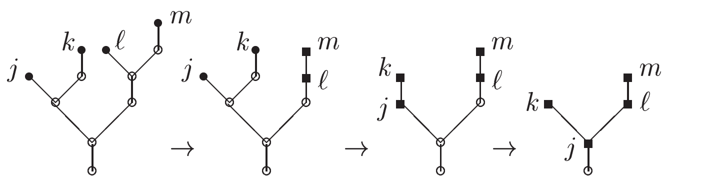
in the evaluation of y_{n+1}. Thus, a comparison with (4.10) yields

$$\begin{aligned} a^i_{j,k,\ell,m} &= \int_0^{c_i} L_j(s) \int_0^s \int_0^s L_k(t) L_\ell(u) \int_0^u L_m(v)\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}t\,\mathrm{d}s, \\ b_{j,k,\ell,m} &= \int_0^1 L_j(s) \int_0^s \int_0^s L_k(t) L_\ell(u) \int_0^u L_m(v)\,\mathrm{d}v\,\mathrm{d}u\sigma\,\mathrm{d}t\,\mathrm{d}s. \end{aligned}$$

The above coefficients can be automatically found by labelling the leaves of an order tree in a lexicographic order from the left to the right



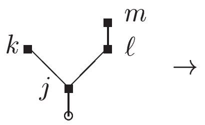
and extracting from the order tree the *integration tree*, that is obtained by removing the commutators. For instance, we obtain



order tree integration tree

for the tree (2.16).

Alternatively, the vertices of the integration tree can be labelled at each level from left to right, and upwards. The corresponding coefficient is easily found by polynomial integration. The upper extreme of integration of a child vertex is the integration variable of its parent. For instance, given the integration tree above, we have



$$\begin{aligned} a^i_{j,k,\ell,m} &= \int_0^{c_i} L_j(s) \int_0^s \int_0^s L_k(t) L_\ell(u) \int_0^u L_m(v)\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}t\,\mathrm{d}s, \\ b_{j,k,\ell,m} &= \int_0^1 L_j(s) \int_0^s \int_0^s L_k(t) L_\ell(u) \int_0^u L_m(v)\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}t\,\mathrm{d}s. \end{aligned}$$

Here, the main difference from [17] is the presence of the $a^i_{j,k,\ell,\ldots}$ coefficients, that originate from the intermediate stages Y_i.

Definition 4.1 Let $c_1, c_2, \ldots, c_\nu$ be given collocation points. For each order tree (of order greater than two) we refer to the corresponding $a^i_{j,k,\ell,\ldots}$ as *higher-order coefficients* and to the corresponding $b_{j,k,\ell,\ldots}$ as the *higher-order weights.* ◇

Lemma 4.1 (Iserles–Nørsett) *Let $c_1, \ldots, c_\nu$ be nodes of quadrature of order p in the univariate case, i.e. suppose that the quadrature formula*

$$\int_0^1 f(t)\,\mathrm{d}t \approx \sum_{i=1}^{\nu} b_i f(c_i), \qquad b_i = \int_0^1 L_i(t)\,\mathrm{d}t,$$

is exact for all polynomials of degree p. Then, provided that $\mathcal{L}$ is an r-linear operator, the quadrature formula

$$\int_0^1 \int_0^{t_1} \int_0^{t_2} \cdots \int_0^{t_{r-1}} \mathcal{L}\big(f(s_1), f(s_2), \ldots, f(s_r)\big)\,\mathrm{d}s_r \cdots \mathrm{d}s_1 \approx \sum_{i_1,\ldots,i_r=1}^{\nu} b_{i_1,\ldots,i_r} \mathcal{L}\big(f(c_{i_1}), \ldots, f(c_{i_r})\big),$$

where $t_k \in \{s_1, \ldots, s_k\}, k = 1, \ldots, r-1,$ with weights

$$b_{i_1,\ldots,i_r} = \int_0^1 \int_0^{t_1} \int_0^{t_2} \cdots \int_0^{t_{r-1}} L_{i_1}(s_1) \cdots L_{i_r}(s_r)\,\mathrm{d}s_r \cdots \mathrm{d}s_1,$$

is also of order p.

For the proof of Lemma 4.1 we refer the reader to [17].

The above lemma guarantees that substituting the integrals of the Lagrangian polynomials with the corresponding weights, we can replace the order trees in the Fer/Magnus series with discrete sums of the right order of approximation.

Before proceeding further, it is useful to recall the Alexeev–Gröbner Lemma, which constitutes the nonlinear equivalent of the *variation of constants* formula (cf. [27]).

Lemma 4.2 (Alekseev–Gröbner) *Let $y(t)$ be the solution of the differential equation* (4.1) *with initial condition $y(t_0) = y_0$ and let $w(t)$ be a C^1 approximation for y such that $w(t_0) = y_0$. Then, provided that f and its first derivative are continuous,*

$$y(t) - w(t) = \int_{t_0}^{t} \Phi\big(t, s, w(s)\big)\Big(f\big(s, w(s)\big) - w'(s)\Big)\,\mathrm{d}s,$$

$\Phi\big(t, s, w(s)\big)$ being the result of partial differentiation of the solution of (4.1) *passing through $s, w(s)$, with respect to the initial values $w(s)$.*

Lemma 4.3 *Let $c_1, \ldots, c_\nu$ be zeros of quadrature of order p, let $y(t)$ be the exact solution of* (4.8) *with the initial condition $y(t_0) = y_0$ and let $w(t)$ be the exact solution of*

$$w' = \left(\sum_{i=1}^{\nu} L_i\left(\frac{t - t_n}{h}\right)\gamma_i\right) w(t), \qquad w(t_n) = y(t_n), \tag{4.11}$$

where $\gamma_i = \gamma(t_n + c_i h, w(t_n + c_i h))$. Then w is an order-p approximation to $y(t)$ for $t_n \le t \le t_n + h$.

Proof. Let $t_n \le t \le t_n + h$. Because of the Alekseev–Gröbner Lemma, we deduce that

$$y(t) - w(t) = \int_{t_n}^{t} \Phi\big(t, s, w(s)\big)\Big(\gamma\big(s, w(s)\big)w(s) - w'(s)\Big)\,\mathrm{d}s. \tag{4.12}$$

Note that at the collocation points, the argument of the integral vanishes, therefore, if we integrate (4.12) with the quadrature formula based on such nodes, the error in the integral will be simply the quadrature error, i.e. $\mathcal{O}\big(h^{p+1}\big)$. Hence w is an approximation of order p to y. □

At this point, we can introduce the main result of this paper.

Theorem 4.4 (Collocation Theorem) *Let $c_1, c_2, \ldots, c_\nu$ be ν collocation points, and let a^i_j and b_j, for $i, j = 1, \ldots, \nu$, be the coefficients and the weights, respectively, of the corresponding Runge–Kutta order-p collocation scheme. Let the Fer or the Magnus expansion be truncated up to the order-p trees for y_{n+1} and order-$(p-1)$ trees for the intermediate stages and let $a^i_{j,k,\ldots}$ and $b_{j,k,\ldots}$ be the higher-order coefficients and weights derived from the corresponding order trees. Then, the resulting schemes have order p.*

Proof. The proof follows in three steps. First of all, let $w(t)$ be as in Lemma 4.3, expanded in terms of the infinite Fer/Magnus expansion. We truncate the infinite expansion to order p, keeping all the integrals exact, and denote the result by $\tilde{w}(t)$. At $t_0 + h$ we have only truncation error, therefore

$$w(t_n + h) - \tilde{w}(t_n + h) = \mathcal{O}\big(h^{p+1}\big) .$$

Next, we replace the integrals in $\tilde{w}$ by quadrature formulae, whereby the weights are evaluated according to the order trees. Let w_{n+1} denote such an approximation. Then, as a consequence of Lemma 4.1,

$$\tilde{w}(t_n + h) - w_{n+1} = \mathcal{O}\big(h^{p+1}\big) ,$$

the multilinear functions in $\tilde{w}$ being combinations of commutators.

Finally we note that the order $\mathcal{O}\big(h^{p+1}\big)$ in the internal stages for the evaluation of w_{n+1} does not contribute to the $\mathcal{O}\big(h^{p+1}\big)$ term at $t_n + h$ but only in the $\mathcal{O}\big(h^{p+2}\big)$ term, therefore the internal stages need be approximated to one order of magnitude less. Let y_{n+1} denote the value obtained in this way. Such u_h is the resulting method, in the statement of the theorem.We have $w_{n+1} - y_{n+1} = \mathcal{O}\big(h^{p+1}\big)$. Hence,

$$y(t_n + h) - y_{n+1} = \Big[y(t_n + h) - \tilde{w}(t_n + h)\Big] + \Big[w_{n+1}(t_n + h) - w_{n+1}\Big] + (w_{n+1} - y_{n+1}) = \mathcal{O}\big(h^{p+1}\big) ,$$

and this proves that y_{n+1} is an approximation of order p to $y(t_n + h)$. □

The meaning of Theorem 4.4 is illustrated in the examples below.

Example 4.1 We consider the collocation points $c_1 = 0, c_2 = \frac{1}{2}, c_3 = 1$ which correspond to the *Lobatto* quadrature points. In the classical RK case, these points correspond to the order-four IRK known as *Ehle III$_A$* [12], with the RK tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & \frac{-1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} . \tag{4.13}$$

We choose the Fer expansion for order 4, i.e. we need to consider the trees (2.10) and (2.12). The tree (2.10) gives rise to the same weights as in (4.13), while the order trees in (2.12) give rise to new higher-order coefficients and weights, according to the integration tree. Suppose that $y_n \in G$

and a stepsize of integration h are given. The scheme becomes

$$
\begin{aligned}
Y_i &= \exp\Big(h\sum_{j=1}^{3} a^i_j\gamma_j\Big)\exp\Big(\frac{h^2}{2}\sum_{j,k=1}^{3} a^i_{j,k}[\gamma_j,\gamma_k] \\
&\qquad + \frac{h^3}{3}\sum_{j,k,\ell=1}^{3} a^i_{j,k,\ell}\big[[\gamma_j,\gamma_k],\gamma_\ell\big]\Big)y_n, \qquad i=1,\dots,3, \\
\gamma_i &= \gamma(Y_i), \\
y_{n+1} &= \exp\Big(h\sum_{j=1}^{3} b_j\gamma_j\Big)\exp\Big(\frac{h^2}{2}\sum_{j,k=1}^{3} b_{j,k}[\gamma_j,\gamma_k] \\
&\qquad + \frac{h^3}{3}\sum_{j,k,\ell=1}^{3} b_{j,k,\ell}\big[[\gamma_j,\gamma_k],\gamma_\ell\big]\Big)y_n.
\end{aligned} \tag{4.14}
$$

However, to have the whole scheme order four, it is sufficient to evaluate the intermediate stages Y_i to order three only. Therefore (4.14) reduces to

$$
\begin{aligned}
Y_i &= \exp\Big(h\sum_{j=1}^{3} a^i_j\gamma_j\Big)\exp\Big(\frac{h^2}{2}\sum_{j,k=1}^{3} a^i_{j,k}[\gamma_j,\gamma_k]\Big)y_n, \qquad i=1,\dots,3, \\
\gamma_i &= \gamma(Y_i), \\
y_{n+1} &= \exp\Big(h\sum_{j=1}^{3} b_j\gamma_j\Big)\exp\Big(\frac{h^2}{2}\sum_{j,k=1}^{3} b_{j,k}[\gamma_j,\gamma_k] \\
&\qquad + \frac{h^3}{3}\sum_{j,k,\ell=1}^{3} b_{j,k,\ell}\big[[\gamma_j,\gamma_k],\gamma_\ell\big]\Big)y_n.
\end{aligned} \tag{4.15}
$$

Here, the coefficients a^i_j and b_j are the same as in the corresponding collocation RK in (4.13), while the higher-order coefficients and weights are evaluated as follows,

$$
\begin{aligned}
a^i_{j,k} &= \int_0^{c_i} L_j(t)\int_0^t L_k(\tau)\,\mathrm{d}\tau\,\mathrm{d}t, \qquad i,j,k=1,\dots,3, \\
b_{j,k} &= \int_0^1 L_j(t)\int_0^t L_k(\tau)\,\mathrm{d}\tau\,\mathrm{d}t, \qquad j,k=1,\dots,3, \\
b_{j,k,\ell} &= \int_0^1 L_j(t)\int_0^t\int_0^t L_k(\tau)L_\ell(\sigma)\,\mathrm{d}\sigma\,\mathrm{d}\tau\,\mathrm{d}t, \qquad j,k,\ell=1,\dots,3.
\end{aligned} \tag{4.16}
$$

We find

$$
a^1 = O, \quad a^2 = \begin{pmatrix} \frac{25}{1152} & \frac{17}{1440} & \frac{-13}{5760} \\ \frac{83}{1440} & \frac{1}{18} & \frac{-13}{1440} \\ \frac{-37}{5760} & \frac{-7}{1440} & \frac{1}{1152} \end{pmatrix}, \quad a^3 = \begin{pmatrix} \frac{1}{72} & \frac{-1}{90} & \frac{-1}{360} \\ \frac{11}{90} & \frac{2}{9} & \frac{-1}{90} \\ \frac{11}{360} & \frac{11}{90} & \frac{1}{72} \end{pmatrix},
$$

for the higher-order coefficients, while the higher-order weights are

$$
b_{i,j} = a^3_{i,j}, \quad i,j=1,\dots,m,
$$

(since $c_3 = 1$), and

$$\begin{array}{lll}
b_{1,2,1} = \frac{-13}{5670}, & b_{2,1,1} = \frac{131}{5670}, & b_{3,1,1} = \frac{25}{4536}, \\
b_{1,2,2} = \frac{-32}{2835}, & b_{2,1,2} = \frac{121}{2835}, & b_{3,1,2} = \frac{61}{2835}, \\
b_{1,2,3} = \frac{-2}{2835}, & b_{2,1,3} = \frac{-13}{5670}, & b_{3,1,3} = \frac{53}{22680}, \\
b_{1,3,1} = \frac{-1}{2268}, & b_{2,3,1} = b_{2,1,3}, & b_{3,2,2} = \frac{44}{567}, \\
b_{1,3,2} = b_{1,2,3}, & b_{2,3,2} = \frac{-1}{567}, & b_{3,2,1} = b_{3,1,2}, \\
b_{1,3,3} = \frac{-1}{22680}, & b_{2,3,3} = \frac{1}{1134}, & b_{3,2,3} = \frac{5}{567}.
\end{array}$$

A note on the implementation of (4.15). Although the matrices a^2, a^3 are full, we need to take into account that the commutator is a skew-symmetric and linear operator. Therefore,

$$Y_i = \exp\Big(h\sum_{j=1}^{3} a^i_j \gamma_j\Big) \exp\Big(\frac{h^2}{2} \sum_{\substack{j,k=1\\ j<k}}^{3} (a^i_{j,k} - a^i_{k,j})[\gamma_j, \gamma_k]\Big) y_n,$$

and, similarly,

$$\begin{aligned}
y_{n+1} \quad = \quad & \exp\Big(h\sum_{j=1}^{3} b_j\gamma_j\Big) \exp\Big(\frac{h^2}{2} \sum_{\substack{j,k=1\\ j<k}}^{3} (b_{j,k} - b_{k,j})[\gamma_j, \gamma_k] \\
& + \frac{h^3}{3} \sum_{\substack{j,k=1\\ j<k}}^{3} \big[[\gamma_j, \gamma_k], \sum_{\ell=1}^{3}(b_{j,k,\ell} - b_{k,j,\ell})\gamma_\ell\big]\Big) y_n.
\end{aligned}$$

Hence, not only the diagonal higher order coefficients and weights need not be calculated, but also there is no need for those high-order coefficients and weights whose first two lower indices are equal. For a further discussion on how to reduce the number of commutators, we refer the reader to [16].

Note that, when $c_1 = 0$, then $a^1_{j,k} = 0$, for $j, k = 1, \ldots, 3$. In general, if $c_1 = 0$ then $a^1_{j,k,\ell,\ldots} = 0$, as for the coefficients of the matrix A in the classical setting. $\diamond$

Example 4.2 Consider the order-four collocation method based on the Gauss–Legendre points, $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$ and $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$, whose classical Runge–Kutta tableau is given by

$$\begin{array}{c|cc}
\frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\
\frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}. \tag{4.17}$$

This time, we consider the Magnus expansion for order four. We evaluate the higher-order coefficients $a^i_{j,k}$ and the higher-order weights $b_{j,k}, b^{(1)}_{j,k,\ell}$, according to the formula (4.16), taking into account that the index range is now $\{1, 2\}$. We also need to evaluate other higher-order weights $b^{(2)}_{j,k,\ell}$, since in the Magnus expansion there are two different trees for order four. We find

$$a^1 = \begin{pmatrix} \frac{1}{32} & \frac{19}{288} - \frac{\sqrt{3}}{24} \\ \frac{-1}{288} & \frac{7}{96} - \frac{\sqrt{3}}{24} \end{pmatrix}, \quad a^2 = \begin{pmatrix} \frac{7}{96} + \frac{\sqrt{3}}{24} & \frac{-1}{288} \\ \frac{19}{288} + \frac{\sqrt{3}}{24} & \frac{1}{32} \end{pmatrix}, \quad (b_{j,k})^2_{j,k=1} = \begin{pmatrix} \frac{1}{8} & \frac{1}{8} - \frac{\sqrt{3}}{12} \\ \frac{1}{8} + \frac{\sqrt{3}}{12} & \frac{1}{8} \end{pmatrix},$$

and finally,

$$\begin{aligned} b^{(1)}_{1,2,1} &= \tfrac{7}{240} - \tfrac{\sqrt{3}}{48}, & b^{(1)}_{2,1,1} &= \tfrac{1}{15} + \tfrac{\sqrt{3}}{24},\\ b^{(1)}_{1,2,2} &= \tfrac{1}{15} - \tfrac{\sqrt{3}}{24}, & b^{(1)}_{2,1,2} &= \tfrac{7}{240} + \tfrac{\sqrt{3}}{48},\\ b^{(2)}_{1,1,2} &= \tfrac{1}{30} - \tfrac{\sqrt{3}}{48}, & b^{(2)}_{1,2,1} &= \tfrac{-1}{240},\\ b^{(2)}_{2,1,2} &= \tfrac{-1}{240}, & b^{(2)}_{2,2,1} &= \tfrac{1}{30} + \tfrac{\sqrt{3}}{48}. \end{aligned}$$

In particular, the weights $b^{(1)}_{j,k,\ell}$ correspond to the tall tree for order four. Having taken into account the skew-symmetry of the bracket, the corresponding order-four, Lie-group method based on the Magnus expansion is:

$$\begin{aligned} Y_1 &= \exp\big(h(a^1_1\gamma_1 + a^1_2\gamma_2) + \tfrac{h^2}{2}(a^1_{1,2} - a^1_{2,1})[\gamma_1,\gamma_2]\big)y_n,\\ \gamma_1 &= \gamma(t_n + c_1 h, Y_1),\\ Y_2 &= \exp\big(h(a^2_1\gamma_1 + a^2_2\gamma_2) + \tfrac{h^2}{2}(a^2_{1,2} - a^2_{2,1})[\gamma_1,\gamma_2]\big)y_n,\\ \gamma_2 &= \gamma(t_n + c_2 h, Y_2),\\ y_{n+1} &= \exp\Big(h(b_1\gamma_1 + b_2\gamma_2) + \tfrac{h^2}{2}(b_{1,2} - b_{2,1})[\gamma_1,\gamma_2],\\ &\quad + h^3\big[[\gamma_1,\gamma_2], \big(\tfrac{1}{12}(b^{(1)}_{1,2,1} - b^{(1)}_{2,1,1}) - \tfrac{1}{4}(b^{(2)}_{1,1,2} - b^{(2)}_{1,2,1})\big)\gamma_1\\ &\quad + \big(\tfrac{1}{12}(b^{(1)}_{1,2,2} - b^{(1)}_{2,1,2}) - \tfrac{1}{4}(b^{(2)}_{2,1,2} - b^{(2)}_{2,2,1})\big)\gamma_2\big]\Big)y_n. \end{aligned} \tag{4.18}$$

Moreover, we observe that the two trees of order four are equivalent, since they posses the same commutator pattern and we can group them in a single term.

In the linear case, it has been pointed out that the Magnus series has order $p = 2q$ even if the expansion of y_{n+1} is truncated to include the order-$(2q-1)$ trees [32]. In our example, we find that

$$\begin{aligned} &h^3\big[[\gamma_1,\gamma_2], \big(\tfrac{1}{12}(b^{(1)}_{1,2,1} - b^{(1)}_{2,1,1}) - \tfrac{1}{4}(b^{(2)}_{1,1,2} - b^{(2)}_{1,2,1})\big)\gamma_1\\ &\qquad + \big(\tfrac{1}{12}(b^{(1)}_{1,2,2} - b^{(1)}_{2,1,2}) - \tfrac{1}{4}(b^{(2)}_{2,1,2} - b^{(2)}_{2,2,1})\big)\gamma_2\big] \quad = \quad -\tfrac{h^3}{80}\big[[\gamma_1,\gamma_2], \gamma_1 - \gamma_2\big]. \end{aligned}$$

But $[\gamma_1,\gamma_2] = \mathcal{O}(h)$ and $\gamma_1 - \gamma_2$ is also $\mathcal{O}(h)$. Hence, the whole term is $\mathcal{O}\big(h^5\big)$ and contributes to the local truncation error only. Thus, the approximation y_{n+1} reduces to

$$\begin{aligned} y_{n+1} &= \exp\Big(h(b_1\gamma_1 + b_2\gamma_2) + \tfrac{h^2}{2}(b_{1,2} - b_{2,1})[\gamma_1,\gamma_2]\Big)y_n\\ &= \exp\Big(\tfrac{h}{2}(\gamma_1 + \gamma_2) - \tfrac{\sqrt{3}}{12}h^2[\gamma_1,\gamma_2]\Big)y_n, \end{aligned} \tag{4.19}$$

which, together with the stages in (4.18), gives a fourth-order scheme for the nonlinear problem (1.1). ◇

Note that, in the same assumption of Theorem 4.4, if the function $\gamma(t,y)$ in (1.1) depends on t only, then the collocation technique applied to the Magnus expansion produces exactly the methods of Iserles and Nørsett in [17].

We also observe that, in the light of §1.3, classical collocation can be viewed as a special case of this more general approach whereby we think about the vector space $\mathbb{F}^d$ as an homogeneous space acting on itself whereby the multiplication rule of the group is the '+' (translation). In that case, since the space is commutative, all the commutator terms disappear and we are left with the terms that require only the coefficients in the classical RK tableau [26].

Once we have chosen our favourite expansion (whether the Fer or the Magnus expansion) the coefficients and the weights depend only on the collocation points and the order trees, most of which are common to both expansions. At a glance, the Fer expansion appears more costly, since it requires a larger number of exponentials per time-step. On the other hand, it generally requires less order trees for achieving a given order. The Magnus series requires less exponential evaluations but its order condition include a larger number of trees which are not all independent. Among these, the bushy trees often display order higher than expected. The independence of the order conditions is at the moment under investigation. For order $p \leq 4$, an analysis of the independent order trees and numerical examples display that the Magnus expansion is generally more convenient than the Fer series. For this reason, we shall employ Magnus-type methods in the numerical tests in Section 6.

4.2 Algebraic formulae for the coefficients and the weights

It is very useful to reduce the evaluation of the higher-order coefficients and higher-order weights to some algebraic conditions. To this purpose, we recall that the interpolating polynomial (4.9) can be written as

$$\gamma(t,y) \approx \sum_{j=1}^{\nu} L_j\left(\frac{t-t_n}{h}\right)\gamma_j = \sum_{j=1}^{\nu}\left(\frac{t-t_0}{h}\right)^{j-1}\alpha_j, \tag{4.20}$$

where $\alpha_j = \sum_{k=1}^{\nu}(V^{-1})_{j,k}\gamma_k$, or, in a more compact form,

$$\boldsymbol{\alpha} = V^{-1}\boldsymbol{\gamma}, \tag{4.21}$$

$\boldsymbol{\gamma}$ being the vector constructed from the stages γ_i. V is the *Vandermonde* matrix

$$V = \begin{pmatrix} 1 & c_1 & \dots & c_1^{\nu-1} \\ 1 & c_2 & \dots & c_2^{\nu-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_\nu & \dots & c_\nu^{\nu-1} \end{pmatrix}, \tag{4.22}$$

(see [27]).

The interpolating polynomial (4.20) is inserted into the Fer or the Magnus expansion. Both expansions are truncated to the relevant order trees.

We consider first the Magnus expansion to find

$$\begin{aligned} y(t) &= \exp\Big(\int_{t_n}^{t}\sum_{j=1}^{\nu}\left(\frac{t-t_0}{h}\right)^{j-1}\alpha_j + \cdots\Big)y_n \\ &= \exp\Big(h\sum_{j=1}^{\nu}\left(\frac{t-t_0}{h}\right)^{j}\frac{\alpha_j}{j} + \cdots\Big)y_n. \end{aligned} \tag{4.23}$$

We obtain the following structure for the coefficients. First of all, in the stages, we recover the matrix A of the collocation RK method corresponding to such nodes. We write such matrix in the form

$$A = CVJV^{-1}, \tag{4.24}$$

where $C = \mathrm{diag}(c_1, c_2, \dots, c_\nu)$, V is the Vandermonde matrix (4.22) and

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & 0 & \ddots & \vdots \\ 0 & 0 & \frac{1}{3} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{\nu} \end{pmatrix}. \tag{4.25}$$

Similarly, the first weights correspond to the weights of the corresponding collocation RK,

$$\boldsymbol{b}^T = (1,1,\ldots,1)JV^{-1}, \tag{4.26}$$

where J is given in (4.25) and V in (4.22). Note that the matrix CVJ in (4.24) and the vector $(1,1,\ldots,1)J$ in (4.26) come from the order-two tree, when we integrate the functions $\left(\frac{t-t_n}{h}\right)^{j-1}$ for $j = 1,\ldots,\nu$. The higher-order coefficients and weights can also be evaluated using these functions instead of the Lagrangian cardinal polynomials according to the order trees. For instance, form the order-three tree we find the higher-order coefficients

$$a^i_{\ell,s} = \sum_{j,k=1}^{\nu} \frac{1}{(j+k)k} c_i^{j+k} (V^{-1})_{j,\ell} (V^{-1})_{k,s}.$$

Let $0 \le \theta \le 1$ and consider the polynomials

$$a_{\ell,s}(\theta) = \sum_{j,k=1}^{\nu} \frac{1}{(j+k)k} \theta^{j+k} (V^{-1})_{j,\ell} (V^{-1})_{k,s}, \quad \ell, s = 1,\ldots,\nu.$$

Then, $a^i_{\ell,s} = a_{\ell,s}(c_i)$. For each value of θ, it is possible to write the matrices $a(\theta)$ corresponding to the tall tree of order three. We have

$$a(\theta) = V^{-T} T(\theta) H J T(\theta) V^{-1}, \qquad V^T a(\theta) V = T(\theta) H J T(\theta), \tag{4.27}$$

where $T(\theta)$ is a diagonal matrix and H is a *Hilbert* matrix,

$$T(\theta) = \begin{pmatrix} \theta & 0 & & & 0 \\ 0 & \theta^2 & 0 & & \\ & 0 & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ 0 & & & 0 & \theta^\nu \end{pmatrix}, \qquad H = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{\nu+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \ddots & \frac{1}{\nu+2} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{2\nu-1} \\ \frac{1}{\nu+1} & \frac{1}{\nu+2} & \cdots & \frac{1}{2\nu-1} & \frac{1}{2\nu} \end{pmatrix}, \tag{4.28}$$

respectively. In particular, the integration weights, $b_{i,j}$, corresponding to the order-three tree, are obtained from $a(\theta)$ letting $\theta = 1$.

A similar procedure extends to higher-order coefficients and weights (which are higher order tensors), except that their representation is more complicated.

Definition 4.2 We say that the an expansion of the form

$$\begin{aligned} Y_i &= \exp\Big(h\sum_{j=1}^{\nu} a^i_j \gamma_j + \frac{h^2}{2} \sum_{j,k=1}^{\nu} a^i_{j,k}[\gamma_j,\gamma_k] + \cdots\Big)\cdots \\ &\quad \exp\Big(\frac{h^q}{r_q} \sum_{i_1,\ldots,i_{q-1}=1}^{\nu} a^i_{i_1,\ldots,i_{q-1}}[\cdots[\gamma_{i_1},\gamma_{i_2}]\cdots,\gamma_{i_{q-1}}]\Big)\cdots y_n, \qquad i = 1,\ldots,\nu, \\ \gamma_i &= \gamma(Y_i), \\ y_{n+1} &= \exp\Big(h\sum_{j=1}^{\nu} b_j\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu} b_{j,k}[\gamma_j,\gamma_k] + \cdots\Big)\cdots \\ &\quad \exp\Big(\frac{h^q}{r_q}\sum_{i_1,\ldots,i_{q-1}=1}^{\nu} b_{i_1,\ldots,i_{q-1}}[\cdots[\gamma_{i_1},\gamma_{i_2}]\cdots,\gamma_{i_{q-1}}] \\ &\quad + \frac{h^{q+1}}{r_{q+1}} \sum_{i_1,\ldots,i_q=1}^{\nu} b_{i_1,\ldots,i_q}[\cdots[\gamma_{i_1},\gamma_{i_2}]\cdots,\gamma_{i_q}]\Big)\cdots y_n, \end{aligned}$$

(a Fer or a Magnus expansion) is *explicit* if the computation of each internal stage Y_i requires only $Y_{i-1}, \ldots, Y_1$. ◇

Note that, if the underlying Runge–Kutta method is implicit, then also the corresponding Fer and Magnus expansions are implicit, since the methods inherit the same RK coefficient matrix A.

Although the schemes that we have presented in this section are *implicit,* as are classical RK collocation schemes, we wish to generate (if possible) *explicit* methods that are suitable for the solution of the nonlinear problem (1.1).

5 Relaxing the collocation conditions for Lie-group expansions

5.1 Relaxed collocation

Collocation is essentially equivalent to an interpolation problem and to inverting the Vandermonde matrix (4.22). However, from our discussion in the previous section, it is clear that the requirement of order p for the internal stages in superfluous, since an order $p-1$ approximation does as well. Algebraically, this can be paraphrased in the following way: Assume that the function $\gamma(t, y)$ is just specified on ν nodes $c_1, \ldots, c_\nu$, i.e. we are given the function values $\gamma_k = \gamma(t_0 + c_k h, y_k)$ for $k = 1, \ldots, \nu$. Furthermore, assume that the polynomial $M(t) = \prod_{k=1}^{\nu}(t - c_k)$ is orthogonal to all polynomials of degree $r-1$, i.e.

$$\int_0^1 M(t) t^q \,\mathrm{d}t = 0, \qquad q = 0, 1, \ldots, r-1.$$

Then, by virtue of Theorem 4.4, the corresponding Lie-group collocation method possesses order $p = \nu + r$.

However, we look for an approximant

$$\tilde{\gamma}(t) = \sum_{k=1}^{\nu} F_k(t) \gamma_k \approx \gamma(t, y),$$

such that *the integral* of $\tilde{\gamma}(t)$ is an approximant of order $\nu \leq \tilde{p} < p$ to *the integral* of γ. In this case, we speak of *relaxed collocation.* Of course, when the $F_k(t)$'s are the Lagrangian polynomials, then $\tilde{p} = p$, and if we express

$$F_k(t) = \sum_{j=1}^{\nu} f_{j,k} t^{j-1},$$

then $f_{j,k} = (V^{-1})_{j,k}$.

We impose the relaxed collocation conditions, thereby looking for a matrix of coefficients $F = (f_{j,k})_{j,k=1}^{\nu}$ such that the integration rule

$$\int_0^1 \gamma(t, y) \,\mathrm{d}t \approx \sum_{k=1}^{\nu} \left(\int_0^1 F_k(t) \,\mathrm{d}t \right) \gamma_k$$

is exact for all polynomials of degree $\tilde{p} - 1$. In other words, F satisfies the algebraic condition

$$FV = \begin{pmatrix} I_{\nu-1} & 0 \\ 0^T & k \end{pmatrix},$$

where $\tilde{p} \geq \nu$ and $k \neq 1$ is an arbitrary parameter. The value $k = 1$ yields collocation methods. Choosing a different k_i for each internal stage corresponds to choosing a different order-$(\tilde{p} - 1)$ approximant at each stage. This is also equivalent to perturbing the nodes $c_1, \ldots, c_\nu$, and usually we find that $\tilde{p}$ equals ν, the number of the collocation points.

However, the free parameters $k_i, i = 1, \ldots, \nu$, may be chosen to annihilate some elements in the coefficient matrices.

Example 5.1 Consider the collocation points $c_1 = 0, c_2 = 1/2, c_3 = 1$, whose corresponding collocation method has been presented in example (4.1). Using the relaxed collocation, we find

$$(a_j^i) = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{8} + \frac{1}{4}k_2 & \frac{1}{2}(1 - k_2) & -\frac{1}{8} + \frac{1}{4}k_2 \\ -\frac{1}{2} + 2k_3 & 2(1 - 2k_3) & -\frac{1}{2} + 2k_3 \end{pmatrix}, \tag{5.1}$$

from which, substituting $k_2 = k_3 = \frac{1}{2}$ in (5.1), we obtain the tableau

$$(a_j^i) = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}, \tag{5.2}$$

while, with $k_2 = \frac{1}{2}, k_3 = \frac{1}{4}$, we find

$$(a_j^i) = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{5.3}$$

Observe that, although the nodes $0, \frac{1}{2}, 1$ are Lobatto points (and the underlying quadrature is of order $p = 2\nu - 2$), relaxing the collocation conditions implies that the order drops to $\tilde{p} = \nu$. Hence, with these points and with relaxed collocation conditions, we can achieve order at most three, therefore the intermediate stages need to be approximated to order two only. Furthermore, we only need the order-three weights $b_{i,j}$, which we have already evaluated in Example 4.1. The numerical method corresponding to (5.3), for the Fer expansion, is:

$$\begin{aligned}
Y_1 &= y_n, \\
\gamma_1 &= \gamma(t_n, Y_1), \\
Y_2 &= \exp\big(\tfrac{h}{4}(\gamma_1 + \gamma_2)\big)y_n, \\
\gamma_2 &= \gamma(t_n + \tfrac{1}{2}h, Y_2), \\
Y_3 &= \exp(h\gamma_2)y_n, \\
\gamma_3 &= \gamma(t_n + h, Y_3), \\
y_n &= \exp\big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3)\big)\exp\big(\tfrac{h^2}{2}(-\tfrac{2}{15}[\gamma_1 - \gamma_3, \gamma_2] - \tfrac{1}{30}[\gamma_1, \gamma_3]\big), \\
&= \exp\big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3)\big)\exp\big(-\tfrac{h^2}{2}([\gamma_1 - \gamma_3, \tfrac{2}{15}\gamma_2 + \tfrac{1}{30}\gamma_3]\big).
\end{aligned} \tag{5.4}$$

A similar procedure applies for the coefficients (5.2). ◇

5.2 Explicit order-three methods

Note that a Fer/Magnus method for Lie groups is explicit when, for instance, fixing an index $1 \leq i \leq \nu$, one has

i) $a^i_{j,k,\ell,m\ldots} = 0$ for $j,k,\ell,\ldots \geq i$,

ii) $a^i_{j,k,\ell,m,\ldots} = a^i_{k,j,\ell,m,\ldots}$ for $j,k,\ell,\ldots \geq i$ (the tensors a^i are symmetric in the first two lower indices).

More simplifying conditions may arise, for higher order tensors a^i, when we take into account the Jacobi identity.

Unfortunately, at this very moment, it is not yet clear how to choose the nodes $c_i, i = 1,\ldots,\nu$, or how to relax the collocation conditions, in order to obey the conditions i) or ii).

However, for order-three schemes, the construction of explicit methods is relatively easy, since the inner stages need be approximated to order two only. In effect, Lie-group methods can be generated along the lines of classical RK, with order-three corrections in the evaluation of y_{n+1}.

Theorem 5.1 *Let c_i, a^i_j, b_i, for $i = 1,\ldots,\nu$, be the nodes, the coefficients and the weights respectively of a classical ν-stage order-three explicit RK scheme. If the order-three weights $b_{i,j}$ obey the condition*

$$\sum_{i,j=1}^{\nu} b_{i,j}(c_i - c_j) = \frac{1}{6}, \tag{5.5}$$

then the explicit scheme

$$\begin{aligned} Y_i &= \exp\Big(h\sum_{i=1}^{\nu} a^i_j \gamma_j\Big) y_n, \qquad\qquad i = 1,\ldots,\nu,\\ \gamma_i &= \gamma(t_n + c_i h, Y_i),\\ y_{n+1} &= \exp\Big(h\sum_{i=1}^{\nu} b_i\gamma_i + \frac{h^2}{2}\sum_{i=1}^{\nu} b_{i,j}[\gamma_i,\gamma_j]\Big) y_n, \end{aligned} \tag{5.6}$$

based on the Magnus expansion, and the explicit scheme

$$\begin{aligned} Y_i &= \exp\Big(h\sum_{i=1}^{\nu} a^i_j \gamma_j\Big) y_n, \qquad\qquad i = 1,\ldots,\nu,\\ \gamma_i &= \gamma(t_n + c_i h, Y_i),\\ y_{n+1} &= \exp\Big(h\sum_{i=1}^{\nu} b_i\gamma_i\Big) \exp\Big(\frac{h^2}{2}\sum_{i,j=1}^{\nu} b_{i,j}[\gamma_i,\gamma_j]\Big) y_n, \end{aligned} \tag{5.7}$$

based on the Fer expansion, possess order three.

Proof. We consider one step of the Magnus scheme (5.6) and, without loss of generality we assume that the given Lie-group differential equation is autonomous and has initial condition $y_0 = I$. Denote $\gamma \equiv \gamma(y)$ and γ_y the derivative of γ with respect to the variable y evaluated at y. Then for the exact solution we have:

$$\begin{aligned} y' &= \gamma y,\\ y'' &= \gamma_y y' y + \gamma y' = \gamma_y \gamma y^2 + \gamma^2 y,\\ y''' &= \gamma_{yy} y' \gamma y^2 + \gamma_y^2 y' y^2 + \gamma_y \gamma y' y + \gamma_y \gamma y y' + \gamma_y y' \gamma y + \gamma\gamma_y y' y + \gamma^2 y', \end{aligned}$$

so that the Taylor expansion of the exact solution is

$$y(h) = I + h\gamma + \frac{h^2}{2}(\gamma_y\gamma + \gamma^2) + \frac{h^3}{6}(\gamma_{yy}\gamma^2 + \gamma_y^2\gamma + 3\gamma_y\gamma^2 + \gamma\gamma_y\gamma + \gamma^3) + \mathcal{O}(h^4)\,,$$

whereby all the relevant functions are now evaluated at $t = 0$.

Let us consider next the expansion of the numerical solution. We have

$$\begin{aligned} Y_i &= \exp(h\sum_{j=1}^{\nu} a_j^i\gamma_j) = I + h\sum_{j=1}^{\nu} a_j^i\gamma_j + \frac{h^2}{2}\Big(\sum_{j=1}^{\nu} a_j^i\gamma_j\Big)^2 + \cdots \\ &= I + h\sum_{j=1}^{\nu} a_j^i\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu} a_j^i a_k^i\gamma_j\gamma_k + \cdots, \end{aligned}$$

so that

$$\begin{aligned} \gamma_i &= \gamma(Y_i) = \gamma(I + h\sum_{j=1}^{\nu} a_j^i\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu} a_j^i a_k^i\gamma_j\gamma_k) + \cdots \\ &= \gamma + \gamma_y\Big(h\sum_{j=1}^{\nu} a_j^i\gamma_j + \frac{h^2}{2}\sum_{j,k=1}^{\nu} a_j^i a_k^i\gamma_j\gamma_k\Big) + \frac{1}{2}\gamma_{yy}\Big(h\sum_{j=1}^{\nu} a_j^i\gamma_j\Big)^2 + \cdots \\ &= \gamma + h\sum_{j=1}^{\nu} a_j^i\gamma_y\gamma + \frac{h^2}{2}\Big(2\sum_{j,k=1}^{\nu} a_j^i a_k^j\gamma_y^2\gamma + \sum_{j,k=1}^{\nu} a_j^i a_k^i\gamma_y\gamma^2 + \sum_{j,k=1}^{\nu} a_j^i a_k^i\gamma_{yy}\gamma_j\gamma_k\Big) + \cdots. \end{aligned}$$

Next we substitute the γ_i's in the expansion of numerical approximation y_1,

$$\begin{aligned} y_1 &= \exp\Big(h\sum_{i=1}^{\nu} b_i\gamma_i + \frac{h^2}{2}\sum_{i,j=1}^{\nu} b_{i,j}[\gamma_i,\gamma_j]\Big) \\ &= I + h\sum_{i=1}^{\nu}\gamma_i + \frac{h^2}{2}\sum_{i,j=1}^{\nu} b_i b_j\gamma_i\gamma_j + \frac{h^2}{2}\sum_{i,j=1}^{\nu} b_{i,j}[\gamma_i,\gamma_j] + \frac{h^3}{6}\sum_{i,j,k=1}^{\nu} b_i b_j b_k\gamma_i\gamma_j\gamma_k + \cdots, \end{aligned}$$

(note that both the last two terms are $\mathcal{O}(h^3)$), to obtain

$$\begin{aligned} y_1 = \; & I + h\sum_{i=1}^{\nu} b_i\gamma + h^2\sum_{i,j=1}^{\nu} b_i a_j^i\gamma_y\gamma + \frac{h^2}{2}\sum_{i,j=1}^{\nu} b_i b_j\gamma^2 + h^3\sum_{i,j,k=1}^{\nu}\gamma_y^2\gamma \\ & + \frac{h^3}{2}\sum_{i,j,k=1}^{\nu} b_i a_j^i a_k^i(\gamma_y\gamma^2 + \gamma_{yy}\gamma^2) + \frac{h^3}{2}\sum_{i,j,k=1}^{\nu} b_i b_j a_k^i(\gamma_y\gamma^2 + \gamma\gamma_y\gamma) \\ & + \frac{h^3}{2}\sum_{i,j,k=1}^{\nu} b_{i,j}(a_k^i - a_k^j)[\gamma_y\gamma,\gamma] + \frac{h^3}{6}\sum_{i,j,k=1}^{\nu} b_i b_j b_k\gamma^3 + \mathcal{O}(h^4)\,. \end{aligned}$$

Taking into account the classical RK conditions for order three,

$$\sum_{i=1}^{\nu} b_i = 1, \quad \sum_{i,j=1}^{\nu} b_i a_j^i = \frac{1}{2}, \quad \sum_{i,j,k=1}^{\nu} b_i a_j^i a_k^i = \frac{1}{3}, \quad \sum_{i,j,k=1}^{\nu} b_i a_j^i a_k^j = \frac{1}{6}$$

[12], after some simple algebraic manipulation we obtain the formula

$$\begin{aligned} y_1 = \; & I + h\gamma + \frac{h^2}{2}(\gamma_y\gamma + \gamma^2) + \frac{h^3}{6}\Big[\gamma_{yy}\gamma^2 + \gamma_y\gamma + \gamma^3 + \Big(\frac{5}{2} + 3\sum_{i,j,k=1}^{\nu} b_{i,j}(a_j^i - a_k^i)\Big)\gamma_y\gamma^2 \\ & + \Big(\frac{3}{2} - 3\sum_{i,j,k=1}^{\nu} b_{i,j}(a_j^i - a_k^i)\Big)\gamma\gamma_y\gamma\Big] + \mathcal{O}(h^4)\,, \end{aligned}$$

which we compare with the Taylor expansion of the exact solution. We find two conditions for order three,

$$\frac{5}{2}+3\sum_{i,j,k=1}^{\nu} b_{i,j}(a_j^i-a_k^i)=3,\quad \frac{3}{2}-3\sum_{i,j,k=1}^{\nu} b_{i,j}(a_j^i-a_k^i)=1,$$

both of which reduce to

$$\sum_{i,j,k=1}^{\nu} b_{i,j}(a_j^i-a_k^i)=\sum_{i,j=1}^{\nu} b_{i,j}(c_i-c_j)=\frac{1}{6}$$

after summation over the index k.

Similarly, the result follows for the Fer expansion, taking into account the first terms of the BCH formula [35]. □

When the collocation points are all distinct, the order-three weights are easily given from the third order trees.

Corollary 5.1.1 *Let the c_i's be distinct nodes and let the order-three weights $b_{i,j}$ be evaluated as*

$$b_{i,j}=\int_0^1\int_0^t L_i(t)L_j(s)\,\mathrm{d}s\,\mathrm{d}t,\qquad i,j=1,\ldots,\nu,$$

or in terms of the formulae (4.27)*, with $\theta=1$. Then, the Lie-group methods* (5.6) *and* (5.7) *have order three.*

Proof. By virtue of Theorem 5.1, it is sufficient to show that the order-three weighs above obey the condition (5.5). We have

$$\begin{aligned}\sum_{i,j=1}^{\nu} b_{i,j}c_i &= \sum_{i,j=1}^{\nu}\int_0^1 L_i(t)c_i\int_0^1 L_j(s)\,\mathrm{d}s\,\mathrm{d}t=\sum_{i=1}^{\nu}\int_0^1 L_i(t)c_it\,\mathrm{d}t=\int_0^1 t^2\,\mathrm{d}t\\ &= \frac{1}{3},\end{aligned}$$

whereby we have taken into account that $\sum_{j=1}^{\nu} L_j(s)=1$ and $\sum_{i=1}^{\nu} L_i(t)c_i=t$. Similarly,

$$\begin{aligned}\sum_{i,j=1}^{\nu} b_{i,j}c_j &= \sum_{i,j=1}^{\nu}\int_0^1 L_i(t)\int_0^1 L_j(s)c_j\,\mathrm{d}s\,\mathrm{d}t=\sum_{i=1}^{\nu}\int_0^1 L_i(t)\frac{t^2}{2}\,\mathrm{d}t=\int_0^1\frac{t^2}{2}\,\mathrm{d}t\\ &= \frac{1}{6}.\end{aligned}$$

Hence we deduce

$$\sum_{i,j=1}^{\nu} b_{i,j}(c_i-c_j)=\frac{1}{3}-\frac{1}{6}=\frac{1}{6},$$

from which the result follows. □

Example 5.2 Consider the order-three explicit RK method due to *Kutta,*

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

(cf. [12]). Then the following Lie-group method based on the Fer expansion has order three:

$$\begin{aligned} Y_1 &= y_n, & \gamma_1 &= \gamma(t_n, Y_1),\\ Y_2 &= \exp(\tfrac{h}{2}\gamma_1)y_n, & \gamma_2 &= \gamma(t_n + \tfrac{h}{2}, Y_2),\\ Y_3 &= \exp\big(h(-\gamma_1 + 2\gamma_2)\big)y_n, & \gamma_3 &= \gamma(t_n + h, Y_3),\end{aligned}$$

$$y_{n+1} = \exp\Big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3)\Big)\exp\Big(-\tfrac{h^2}{2}[\gamma_1 - \gamma_3, \tfrac{2}{15}\gamma_2 + \tfrac{1}{30}\gamma_3]\Big)y_n.$$

Similarly, for the Magnus expansion, one has

$$\begin{aligned} Y_1 &= y_n, & \gamma_1 &= \gamma(t_n, Y_1),\\ Y_2 &= \exp(\tfrac{h}{2}\gamma_1)y_n, & \gamma_2 &= \gamma(t_n + \tfrac{h}{2}, Y_2),\\ Y_3 &= \exp\big(h(-\gamma_1 + 2\gamma_2)\big)y_n, & \gamma_3 &= \gamma(t_n + h, Y_3),\end{aligned}$$

$$y_{n+1} = \exp\Big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3) - \tfrac{h^2}{2}[\gamma_1 - \gamma_3, \tfrac{2}{15}\gamma_2 + \tfrac{1}{30}\gamma_3]\Big)y_n,$$

and Corollary 5.1.1 ensures that also such scheme possesses order three. ◇

6 Some numerical tests

In this section we present the numerical results of the proposed compared with some projection schemes. As a case of study, we consider flows on the Lie group $\mathrm{SO}(d)$, the set of orthogonal matrices with determinant equal to one, which arise in the numerical solution of isospectral flows, i.e. matrix ODEs whose solution always possesses the same set of eigenvalues as the initial condition [37]. In this context it is not clear how to employ DAE techniques, although the problem can be reduced to the solution of an orthogonal flow (which is a flow on a Lie group) that can be treated with Lie-group methods and projection techniques.

The projection schemes are obtained by means of QR factorization (discarding the R factor) and the polar decomposition on the non-orthogonal numerical approximations [13]. The experiments are performed in MATLAB™ . For the QR decomposition we use the MATLAB™ routine `qr`. Since the `qr` routine does not return the canonical QR decomposition (i.e. R does not necessarily possess positive diagonal entries), we correct the nearly-orthogonal factor to an orthogonal matrix

```
[U, R] = qr(U);
U = U * diag(sign(diag(R)));
```

to ensure that U remains $\mathcal{O}(h^p)$-near to the theoretical one.

For the polar decomposition, we employ the Schultz algorithm [13]. Departing from a nearly-orthogonal matrix U_0, we employ the iterations

$$U_{i+1} = U_i\Big(I + \tfrac{1}{2}(I - U_i^T U_i)\Big), \qquad i = 0, 1, 2, \ldots,$$

until U_i is orthogonal to machine accuracy (in reality we use a tolerance of 10^{-12}). We recall that the polar decomposition is a *projection* in the Frobenius norm, i.e. the polar factor U is the best Frobenius-norm orthogonal approximant to U_0. Instead, the orthogonal factor obtained by means of the QR algorithm is not a best approximant in any norm [13].

6.1 The periodic Toda lattice

As a first example, we consider a three-particle periodic Toda lattice, i.e. three particles on a ring subject to a nearest-neighbour exponential potential. The corresponding Hamiltonian function is

$$H(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{2}(p_1^2+p_2^2+p_3^2) + \mathrm{e}^{-(q_2-q_1)} + \mathrm{e}^{-(q_3-q_2)} + \mathrm{e}^{-(q_1-q_3)} - 3. \tag{6.8}$$

Since $p_1+p_2+p_3$ is constant, the centre of mass of the system translates with constant velocity. This appears also from the fact that the Hamiltonian (6.8) does not change if we add an arbitrary constant to the q_i's. This, together with the other conservation laws obeyed by the lattice, means that the trajectories relative to a given initial condition evolve on the submanifold $\mathbb{R}\times T^2 \subset \mathbb{R}^6$, according to the theory of integrable systems [31].

It is well known that Toda lattices can be written in the Lax form

$$L' = [B(L), L], \qquad L(0) = L_0, \tag{6.9}$$

[21], which is a matrix *isospectral flow,* since it is possible to show that the solution $L(t)$ retains the same eigenvalues as the initial condition. Specifically,

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

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

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

[11, 34]. Since classical numerical methods cannot retain isospectrality (see for instance [4, 37]), isospectrality can be recovered by solving numerically the orthogonal flow

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

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

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

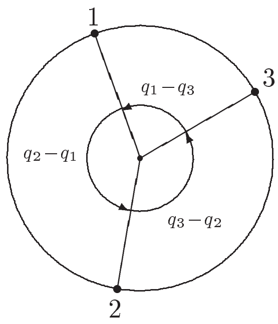


Figure 2: Three particles on a ring with exponential interaction.

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

Besides the Lax form, it is possible to make a (non intuitive) change of variables in the Hamiltonian (6.8),

$$(\boldsymbol{p}, \boldsymbol{q}) \to (\boldsymbol{\eta}, \boldsymbol{\zeta}),$$

where

$$\boldsymbol{p} = A\boldsymbol{\eta}, \qquad \boldsymbol{q} = A\boldsymbol{\zeta}, \tag{6.10}$$

and

$$A = \begin{pmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ -\frac{\sqrt{6}}{3} & 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{pmatrix}, \tag{6.11}$$

which diagonalizes the lattice. We find

$$\begin{array}{ll} q_2 - q_1 = -\frac{\sqrt{6}}{2}\zeta_1 - \frac{\sqrt{2}}{2}\zeta_2, & p_1 = \frac{\sqrt{6}}{6}\eta_1 + \frac{\sqrt{2}}{2}\eta_2 + \frac{\sqrt{3}}{3}\eta_3, \\ q_3 - q_2 = \frac{\sqrt{6}}{2}\zeta_1 - \frac{\sqrt{2}}{2}\zeta_2, & p_2 = -\frac{\sqrt{6}}{3}\eta_1 + \frac{\sqrt{3}}{3}\eta_3, \\ q_1 - q_3 = \sqrt{2}\zeta_2, & p_3 = \frac{\sqrt{6}}{6}\eta_1 - \frac{\sqrt{2}}{2}\eta_2 + \frac{\sqrt{3}}{3}\eta_3, \end{array} \tag{6.12}$$

Note that $p_1^2 + p_2^2 + p_3^2 = \eta_1^2 + \eta_2^2 + \eta_3^2$, and $\eta_3 = \frac{\sqrt{3}}{3}(p_1 + p_2 + p_3)$. Rescaling

$$\zeta_1 = 2\sqrt{2}x, \qquad \zeta_2 = 2\sqrt{2}y, \qquad t \to t/\sqrt{3}$$

and taking into account that $p_1 + p_2 + p_3 = \text{const}$, the problem becomes equivalent to considering a particle moving in a two-dimensional potential, with the Hamiltonian

$$H(p_x, p_y, x, y) = \frac{p_x^2 + p_y^2}{2} + \frac{1}{24}\left(\mathrm{e}^{2y+2\sqrt{3}x} + \mathrm{e}^{2y-2\sqrt{3}x} + \mathrm{e}^{-4y}\right) - \frac{1}{8} \tag{6.13}$$

[34, 3]. Although such Hamiltonian is mathematically equivalent to (6.8), the solution in the new variables evolves on a compact manifold which is a 2-torus embedded in $\mathbb{R}^4$. Hence, if we intersect such a manifold with the (y, p_y) plane (i.e. take a Poincaré section) and integrate for sufficiently long time, we expect to observe smooth closed curves, provided that the stepsize of integration (used in the numerical calculation) and periods of revolution in the independent directions of the torus are incommensurable.

The equations (6.13) were originally considered by Ford and his collaborators in their investigations of the integrability of the cyclic three-particle Toda lattice. If we truncate the Hamiltonian function (6.13) to include all cubic terms, we obtain the well known Hénon–Heiles Hamiltonian,

$$H = \frac{p_x^2 + p_y^2}{2} + \frac{x^2 + y^2}{2} + xy^2 - \frac{1}{3}y^3,$$

which is non integrable.

In order to compare the numerical integrators for the three-particle periodic Toda problem, we plot the Poincaré section of the Hamiltonian (6.13). Such section is obtained taking (y, p_y) when $x = 0$ and $p_x > 0$. Since numerical integration is performed by approximating Flaschka's variables, we employ the α_j's and the β_j's, together with (6.12), to obtain the variables x, y, p_x, p_y that appear in the Hamiltonian function (6.13). Specifically, we follow the algorithm below:

1. First transform the system into the Lax formulation $L' = [B(L), L]$ to obtain an isospectral flow.

2. For $t_n = nh, n = 0, 1, 2, \ldots$, where h is the stepsize of integration, integrate the Lie-group matrix differential equation $y' = B(yL_ny^T)y$, for $t \in [t_n, t_{n+1}]$ and initial condition $y(t_n) = I$, to obtain a numerical approximation y_{n+1}. At this stage, we employ various numerical methods.

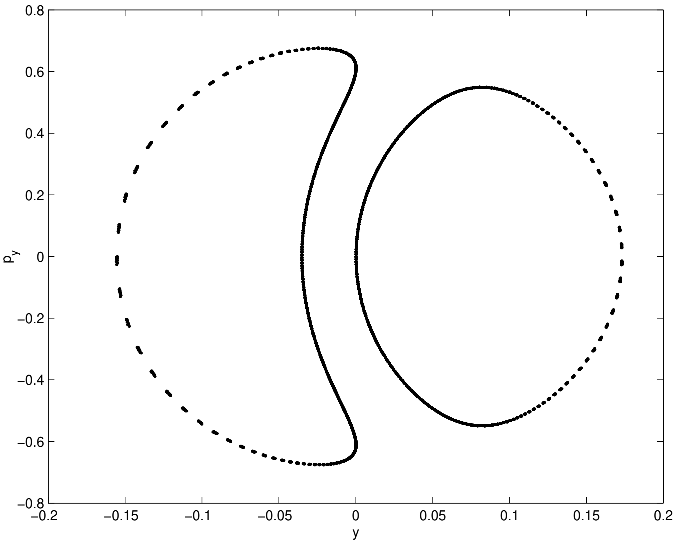


Figure 3: The Poincaré section of the Hamiltonian (6.13) obtained with an isospectral method based on the order-four Magnus collocation method (6.14) for Lie groups (`MagnusZ4`).

3. Update the matrix L by means of the transformation $L_{n+1} = y_{n+1} L_n y_{n+1}^T$, so that the spectrum of L_{n+1} is the same as the spectrum of L_n, provided that y_{n+1} is an orthogonal matrix.

4. Transform the numerical approximation for the Flaschka variables α_j's, β_j's into the scalar variables x, y, p_x, p_y by means of (6.12).

5. Check if the particle, whose state space is now (x, y, p_x, p_y), has crossed the plane (y, p_y).

Since we are interested in a long-term behaviour of the numerical solution, all the numerical integrations in step 2 are performed in the interval $[0, 10000]$, with constant stepsize $h = \frac{1}{10}$ and with the initial condition

$$\boldsymbol{p}_0 = [1, 1, 0]^T, \quad \boldsymbol{q}_0 = [0, 0, 0]^T.$$

In the first case, we solve the associated orthogonal flow $y' = \gamma(y)y$ with the order-four implicit Magnus collocation method based on Gaussian points,

$$\begin{aligned}
Y_1 &= \exp\Big(h\big(\tfrac{1}{4}\gamma_1 + (\tfrac{1}{4} - \tfrac{\sqrt{3}}{6})\gamma_2\big) + \tfrac{h^2}{2}(\tfrac{5}{72} - \tfrac{\sqrt{3}}{24})[\gamma_1, \gamma_2]\Big) y_n, \\
\gamma_1 &= \gamma\big(t_n + (\tfrac{1}{2} - \tfrac{\sqrt{3}}{6})h, Y_1\big), \\
Y_2 &= \exp\Big(h((\tfrac{1}{4} + \tfrac{\sqrt{3}}{6})\gamma_1 + \tfrac{1}{4}\gamma_2) + \tfrac{h^2}{2}(-\tfrac{5}{72} - \tfrac{\sqrt{3}}{24})[\gamma_1, \gamma_2]\Big) y_n, \\
\gamma_2 &= \gamma\Big(t_n + (\tfrac{1}{2} - \tfrac{\sqrt{3}}{6})h, Y_2\Big), \\
y_{n+1} &= \exp\Big(\tfrac{h}{2}(\gamma_1 + \gamma_2) - \tfrac{\sqrt{3}}{12}h^2[\gamma_1, \gamma_2]\Big) y_n.
\end{aligned} \tag{6.14}$$

We label the numerical approximation obtained in this manner as `MagnusZ4` and in Figure 3 the Poincaré section produced by the scheme appears like two smooth closed curves.

In Figure 4 we plot the Poincaré section when the isospectral equations are solved with an isospectral method, based in this case on the explicit Lie-group method

$$\begin{aligned}
Y_1 &= y_n, & \gamma_1 &= \gamma(t_n, Y_1), \\
Y_2 &= \exp(\tfrac{h}{2}\gamma_1) y_n, & \gamma_2 &= \gamma(t_n + \tfrac{h}{2}, Y_2), \\
Y_3 &= \exp\big(h(-\gamma_1 + 2\gamma_2)\big) y_n, & \gamma_3 &= \gamma(t_n + h, Y_3), \\
y_{n+1} &= \exp\Big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3) - \tfrac{h^2}{2}[\gamma_1 - \gamma_3, \tfrac{2}{15}\gamma_2 + \tfrac{1}{30}\gamma_3]\Big) y_n, &&
\end{aligned} \tag{6.15}$$

which we label `RKZ3`.

Next, in Figure 5, we display the sections obtained according to the following procedure, that we call $\mathtt{RK3_U}$: We solve the orthogonal flow $y' = \gamma(y)y$ in step 2 with the classical three-stage, order-three, explicit RK method with tableau

$$\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
1 & -1 & 2 & \\
\hline
 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{array}. \tag{6.16}$$

The matrices y_n are not any longer orthogonal, and the matrix transformation $L_{n+1} = y_{n+1} L_n y_{n+1}^T$ in step 3 is no longer a similarity transformation. Hence the integrals of the motion of the original system are not retained. In order to obtain an orthogonal approximation with $\mathtt{RK3_U}$, at the end of each integration step we perform a QR factorization of y_{n+1}, retaining the orthogonal factor. The corresponding Poincaré section is displayed in Figure 6. The procedure in Figure 7 is essentially the same, except that we orthogonalize also the internal stages Y_i of the Runge–Kutta scheme.

In Figure 8 the orthogonalization of $\mathtt{RK3_U}$ is performed by means of a polar decomposition and in Figure 9 the orthogonalization is also performed at the internal stages. If we simply solve $L' = [B(L), L]$ with the Runge–Kutta scheme with tableau (6.16), a method that we call $\mathtt{RK3_L}$, we obtain the Poincaré section in Figure 10.

A comparison between Figure 3 and those shown next, reveals a fundamental difference in the information delivered by the numerical approximants. The curves in Figures 4–9 spiral inwards even though all the integrals of the original Toda problem are preserved to machine accuracy, while according to the theory of integrable systems, the plots should instead have had the form of smooth closed curves, *á la* Figure 3. In effect, numerical methods applied to the Lax pair (6.9) do not necessarily retain the length of the ring, in which case the system acquires an extra degree of freedom. See [37] for further information.

Numerical results for the various schemes are displayed below in Table 3. The 'exact' solution of the problem is evaluated by means of the Lie-group scheme of the Munthe-Kaas type based on the classical Gauss–Legendre RK of order six, with stepsize $h = \frac{1}{100}$. Table 3 should be understood as follows:

Order Order of the method.

Iso 'y' if the method is isospectral (i.e. the numerical approximation retains all the first integrals), 'n' otherwise.

Method Name of the numerical method. The schemes that we use here are:

`MagnusZ4` Algorithm based on the Magnus order-four collocation method (6.14).

`RKMK4` The RK Munthe-Kaas scheme for Lie-groups based on the classical order-four RK Gauss–Legendre counterpart [25].

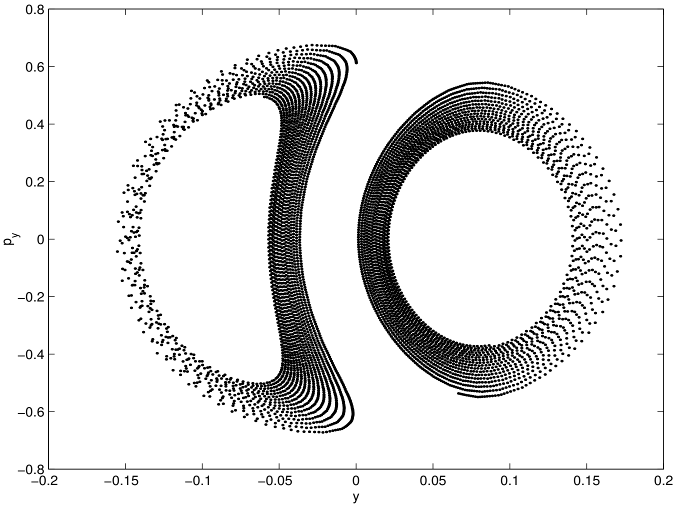


Figure 4: The Poincaré section of the Hamiltonian (6.13) obtained with the isospectral method `RKZ3`, based on the order-three Lie-group scheme (6.15).

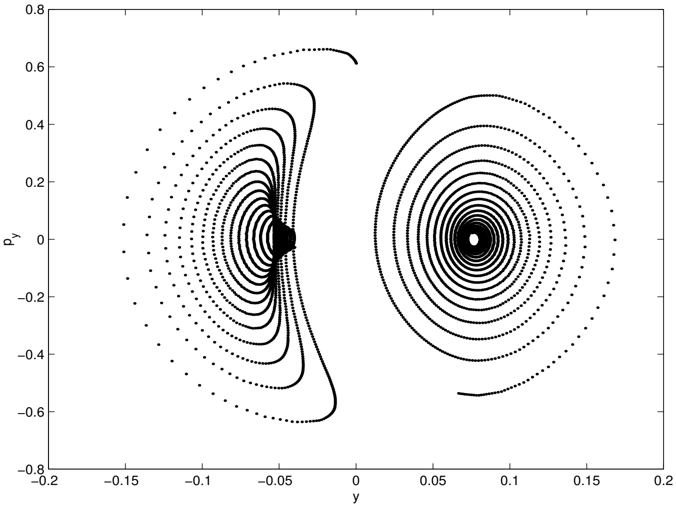


Figure 5: The Poincaré section of the Hamiltonian (6.13) obtained with the non-isospectral method $\mathtt{RK3}_{\mathtt{U}}$, based on the explicit order-three Runge–Kutta scheme with tableau (6.16).

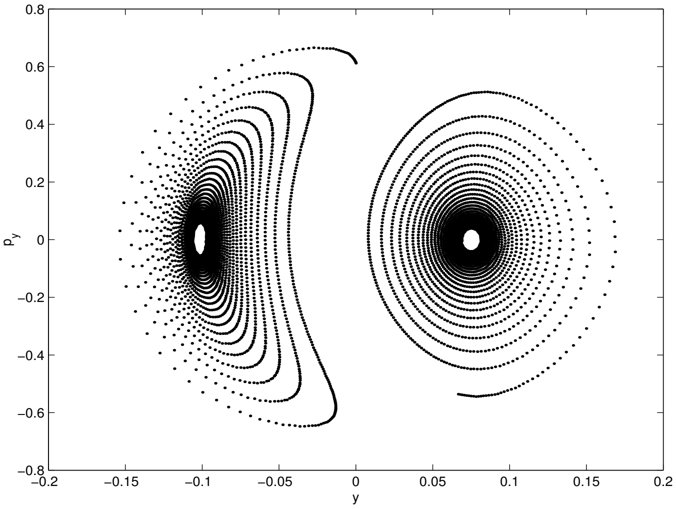


Figure 6: The Poincaré section of the Hamiltonian (6.13) obtained with the method $\mathtt{RK3_U}$, based on the explicit order-three Runge–Kutta scheme with tableau (6.16) and QR factorization at the end of each integration step.

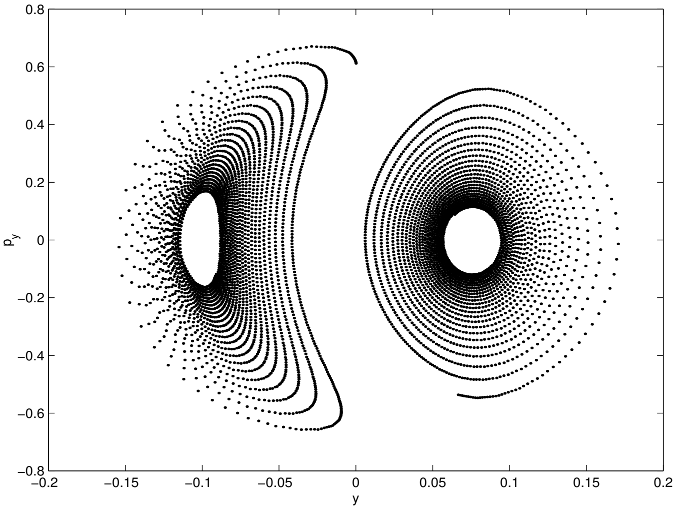


Figure 7: The Poincaré section of the Hamiltonian (6.13) obtained with the method $\mathtt{RK3_U}$, based on the explicit order-three Runge–Kutta scheme with tableau (6.16) and QR factorization at each integration step and at the internal stages.

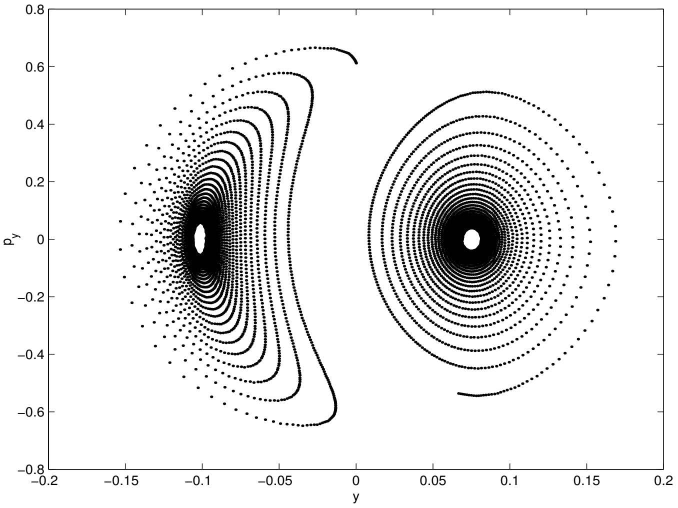


Figure 8: The Poincaré section of the Hamiltonian (6.13) obtained with the method $\mathtt{RK3}_{\mathtt{U}}$, based on the explicit order-three Runge–Kutta scheme (6.16) with polar projection at each integration step.

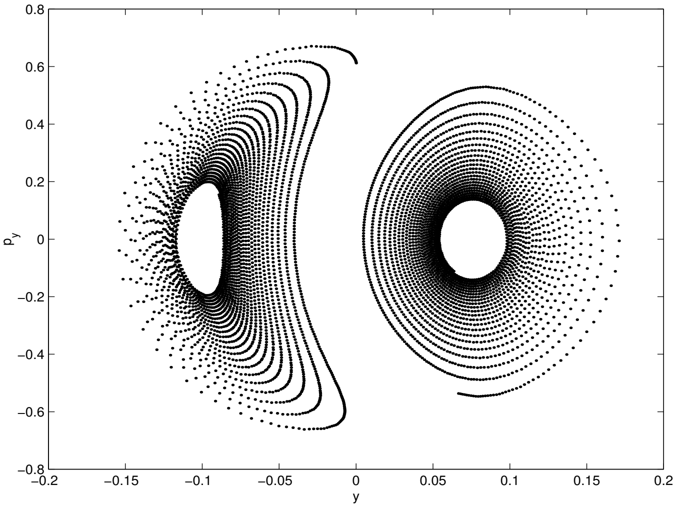


Figure 9: The Poincaré section of the Hamiltonian (6.13) obtained with the method $\mathtt{RK3}_{\mathtt{U}}$, based on the explicit order-three Runge–Kutta scheme (6.16) and polar projection at each integration step and at the internal stages.

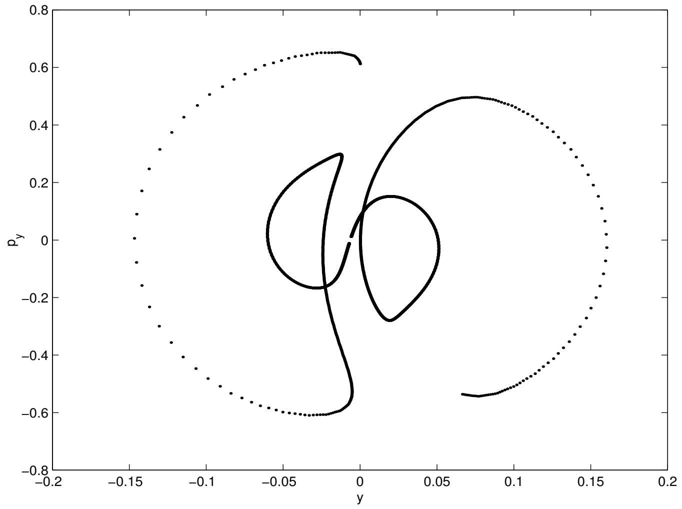


Figure 10: The Poincaré section of the Hamiltonian (6.13) obtained with the non-isospectral method $\mathtt{RK3_L}$, based on the Runge–Kutta scheme with tableau (6.15).

$\mathtt{RK4_L}$, $\mathtt{RK4_U}$ Classical Gauss–Legendre RK of order four.

`RKZ3` Algorithm based on the scheme (6.15), a Lie-group adaption of the classical 'Kutta' RK scheme of order three.

$\mathtt{RK3_L}$, $\mathtt{RK3_U}$ Classical order-three RK method known as *Kutta,* with coefficients in (6.16).

(GL) refers to schemes based on Gauss–Legendre nodes, (Kutta) refers to scheme based on the 'Kutta' scheme (6.16). Since numerical methods are applied in two different ways, we append to a method the subscript `U` when it is used to solve $y' = B(yL_ny^T)y$ in tandem with $L_{n+1} = y_{n+1}L_ny_{n+1}^T$, e.g. $\mathtt{RK3_U}$, while we use the subscript `L` when the method is applied to the flow $L' = [B(L), L]$, e.g. $\mathtt{RK3_L}$. Furthermore, QR refers to a projective technique based on a QR factorization, while P indicates that a polar decomposition is performed.

Global error Global error, evaluated as $\|L_n - L(t_n)\|_2$ at $t_n = T = 10000$.

$|d(\mathcal{I}_2)|$ Error on the quadratic integral at $t_n = T = 10000$. It is evaluated as

$$|d(\mathcal{I}_2)| = \left| \sum_{i=1}^{3} \lambda_i(L_n)^2 - \sum_{i=1}^{3} \lambda_i(L_0)^2 \right|,$$

where $\lambda_i(L_n)$ refers to the i-th eigenvalue of the numerical solution at t_n, while $\lambda_i(L_0)$ refers to the i-th eigenvalue of the initial condition.

$|d(\mathcal{I}_3)|$ Error on the cubic integral at $T = 10000$. It is evaluated as

$$|d(\mathcal{I}_3)| = \left| \sum_{i=1}^{3} \lambda_i(L_n)^3 - \sum_{i=1}^{3} \lambda_i(L_0)^3 \right|,$$

the symbols being as for $d(\mathcal{I}_2)$.

e_{latt} Increment in the lattice length, evaluated as

$$e_{\mathrm{latt}} = \prod_{i=1}^{3} \alpha_i - \frac{1}{8}.$$

The α_i's are the extra-diagonal elements of L_n at $T = 10000$.

Flops Floating point operations, evaluated using MATLAB™ built-in routine `flops`. This includes evaluation of commutators, exponentials, eventual similarity transformations, Picard's fixed point iterations for the implicit schemes, eventual QR and polar factorization, transformation into the (x, y, p_x, p_y) variables of the reduced system and evaluation of the Poincaré section. For methods based on Lie-groups schemes, the results are reported in two columns: 'expm', that refers to the flops used when employing MATLAB™ built-in function `expm`, which evaluates the matrix exponential by scaled Padé approximants, and 'ER_expm', that refers to the flops used when the exponential is evaluated (exactly) using the *Euler–Rodriguez* formula

$$\exp \sigma = I + \frac{\sin \alpha}{\alpha} \sigma + \frac{1}{2} \left[\frac{\sin \frac{\alpha}{2}}{\frac{\alpha}{2}} \right]^2 \sigma^2,$$

where

$$\alpha = \frac{1}{2} \|\sigma\|_F,$$

which is valid for 3×3 skew-symmetric matrices [24, 33].[2]

Taking into account the results in Table 3 and the plots of the corresponding Poincaré section,[3] we see that implicit methods based on Gaussian nodes produce the most satisfactory results. Among these, isospectral methods are better for global error and error on the first integrals of the system (although the error accumulates). Lie-group schemes are quite competitive with regards to the classical `RK4`, which we would have expected to be cheaper because it does not require as many commutators and exponentials as `MagnusZ4` and `RKMK4`. Furthermore the Euler–Rodriguez formula offers significant savings on the implementation cost.

Among explicit methods, although the global error of the schemes is overall very similar, the method `RKZ3` retains better the integrals of motion (and its Poincaré section is nearer to the theoretical one) but at a higher computational cost. The projection methods that employ QR factorization and polar decomposition are not very satisfactory, not even when a projection is also performed in the internal stages of the RK scheme. Moreover, there is no striking difference between the results obtained with QR and a polar projections, the latter being a best orthogonal approximant in the Frobenius norm. A careful comparison reveals that the polar decomposition produces slightly less dissipative results than the QR method, but at a higher price. We remind the reader that also `RKZ3` can be used as a 'projection method' in this context: since $\gamma(y)$ is by construction skew-symmetric regardless of the orthogonality of y, the internal stages could be evaluated in the classical manner (i.e. without performing exponentiation) and then the exponentiation with third-order correction could be performed in the evaluation of y_{n+1} only.

We conclude that in the case when the underlying solution possesses strong conservative attributes, isospectral approximations based on numerical schemes that share the same 'geometric' feature as the underlying flow (for instance $\mathtt{RK4}_\mathtt{U}$ and Lie-group methods) produce the best output. Projection schemes display a more dissipative behaviour, even when a projection is performed in the inner stages of the methods.

[2]The MATLAB™ routine `ER_expm` to evaluate the exponential by means of the Euler–Rodriguez formula, has been written and communicated to me by Arne Marthinsen, NTNU.

[3]The Poincaré sections obtained with `RKMK4`, $\mathtt{RK4}_\mathtt{U}$ and $\mathtt{RK4}_\mathtt{L}$ are, to plotting accuracy, identical to the one displayed in Figure 3.

Order	Iso	Method	Global error $T = 10000, h = 0.1$	$\lvert d(\mathcal{I}_2)\rvert$	$\lvert d(\mathcal{I}_3)\rvert$	e_{latt}	Flops `expm`	Flops `ER_expm`
4	y	`MagnusZ4` (GL)	2.4218e-03	1.9133e-10	4.1180e-10	4.4511e-07	2.0651e+09	1.0588e+09
	y	`RKMK4` (GL)	6.9660e-03	2.7511e-10	5.9217e-10	1.0942e-06	2.9682e+09	1.7320e+09
	y(n)[a]	$\mathtt{RK4}_{\mathtt{U}}$ (GL)	3.6632e-03	6.3204e-10	1.3604e-09	1.7147e-08	2.0349e+09	
	n	$\mathtt{RK4}_{\mathtt{L}}$ (GL)	8.1482e-03	1.6365e-13	3.0964e-07	3.7961e-09	2.2684e+09	
3	y	`RKZ3` (Kutta)	0.2349	1.6999e-10	3.6582e-10	4.3348e-03	3.0930e+08	1.3260e+08
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, QR)	0.3965	8.2401e-13	1.5190e-12	8.3402e-03	1.1580e+08	
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, QR)[b]	0.4030	8.5859e-13	1.5474e-12	7.9999e-03	1.5181e+08	
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, P)	0.3915	3.3890e-14	1.2955e-14	8.3429e-03	2.2654e+08	
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, P)[c]	0.3902	3.4139e-15	4.5908e-14	7.8379e-03	5.8987e+08	
	n	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta)	0.3094	1.5972e-01	1.2069e-01	1.3353e-02	9.7804e+07	
	n	$\mathtt{RK3}_{\mathtt{L}}$ (Kutta)	0.2354	1.2693e-01	6.9215e-02	4.9914e-03	8.9806e+07	

Table 3: Comparison of results for the three-particle periodic Toda lattice. The stage equations for the implicit schemes are solved by Picard iterations, with tolerance $\epsilon = 10^{-12}$.

[a] The Runge–Kutta Gauss–Legendre methods are only weakly orthogonal, see [13, 37].
[b] With QR factorization at each internal stage.
[c] With polar factorization at each internal stage.

6.2 The Toeplitz inverse eigenvalue problem (ITEP)

Isospectral flows are a vast source for application of Lie-group methods. Also in this case, we consider an isospectral flow in order to find a symmetric Toeplitz matrix with a prescribed set of real numbers as its eigenvalues. The flow generally converges to an asymptotic state, so that in this context it is very useful to consider explicit methods. We follow the line of [7, 37] and consider a Lax pair (6.9) with matrix B evaluated as

$$B_{i,j} = \begin{cases} L_{i+1,j} - L_{i,j-1}, & \text{if } 1 \le i < j \le d, \\ 0, & \text{if } i = j, \quad i,j = 1, \dots, d, \\ L_{i-1,j} - L_{i,j+1}, & \text{otherwise.} \end{cases}$$

The asymptotic state is reached when $\|B(L)\| = 0$, hence the real-valued function

$$\psi(L) = \|B(L)\|$$

can be used as an indication of the error committed in the approximation. However, numerical experiments with various norms seem to indicate that the exact value of the function $\psi(L)$ is generally not monotonically decreasing along the orbit of the flow.

It is known that the inverse Toeplitz eigenvalue problem has a solution but not much is known about uniqueness [19]. In the 4×4 case Delsarte and Genin showed that Toeplitz matrices can have either even or odd eigenvectors, and that the number of even and odd eigenvectors can differ by one at most [9]. Recall that a d-component vector $\boldsymbol{v}$ is *even* if

$$v_i = v_{d+1-i}, \qquad i = 1, 2, \dots, d,$$

and *odd* if

$$v_i = -v_{d+1-i}, \qquad i = 1, 2, \dots, d.$$

Furthermore, Delsarte et al. also proved that, in order to have a continuous solution to the ITEP, one must assign opposite parity eigenvectors to adjacent eigenvalues. The general proof of the existence of a solution for the ITEP originated from that observation: Landau showed that, given an arbitrary set of real numbers, it is possible to construct a 'regular' Toeplitz matrix with those numbers as eigenvalues [19]. The regularity condition is equivalent to requiring that the Toeplitz matrix and its principal submatrices have eigenvector of opposite parity associated to adjacent eigenvalues.

Numerical experiments show that this feature is shared by the asymptotic state of the flow proposed in [7, 6]: in our example the initial condition is the diagonal matrix

$$L_0 = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5),$$

where the λ_i's are the eigenvalues of the 5×5 Toeplitz matrix

$$\frac{1}{10} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 2 & 3 \\ 4 & 3 & 2 & 1 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix},$$

which does not obey the regularity conditions of Landau: if we perform a spectral decomposition, we find

$$\begin{array}{lcrcl} \lambda_1 & = & -0.52360679774891 & \to & \boldsymbol{v}_1 \text{ odd}, \\ \lambda_2 & = & -0.16352377300479 & \to & \boldsymbol{v}_2 \text{ even}, \\ \lambda_3 & = & -0.07639320224978 & \to & \boldsymbol{v}_3 \text{ odd}, \\ \lambda_4 & = & -0.05562949153141 & \to & \boldsymbol{v}_4 \text{ even}, \\ \lambda_5 & = & 1.31915326453360 & \to & \boldsymbol{v}_5 \text{ even}, \end{array}$$

which means that the regularity conditions are not satisfied because the bottom eigenvectors are both even. The numerical flow converges to a matrix whose entries (correct to the 10th decimal digit) are

$$L = \begin{pmatrix} 0.0999999999 & 0.3899448328 & 0.1911553690 & 0.3392402861 & 0.2279978954 \\ 0.3899448328 & 0.0999999999 & 0.3899448328 & 0.1911553690 & 0.3392402861 \\ 0.1911553690 & 0.3899448328 & 0.0999999999 & 0.3899448328 & 0.1911553690 \\ 0.3392402861 & 0.1911553690 & 0.3899448328 & 0.0999999999 & 0.3899448328 \\ 0.2279978954 & 0.3392402861 & 0.1911553690 & 0.3899448328 & 0.0999999999 \end{pmatrix},$$

for which one has

$$\begin{array}{rcrcl} \lambda_1 & = & -0.5236067977 & \to & \boldsymbol{v}_1 \text{ even,} \\ \lambda_2 & = & -0.1635237730 & \to & \boldsymbol{v}_2 \text{ odd,} \\ \lambda_3 & = & -0.0763932022 & \to & \boldsymbol{v}_3 \text{ even,} \\ \lambda_4 & = & -0.0556294915 & \to & \boldsymbol{v}_4 \text{ odd,} \\ \lambda_5 & = & 1.3191532645 & \to & \boldsymbol{v}_5 \text{ even,} \end{array}$$

and the same holds also for all the principal submatrices. The numerical convergence sometimes can be quite slow, but it can be accelerated by means of a procedure proposed by [20], which sets the initial condition to a matrix that has the same eigenvalues of the exact solution, is nearly Toeplitz, and, most importantly, has eigenvectors obeying the regularity condition. In our case, the matrix $L_0 = \mathrm{diag}(\lambda_1, \ldots, \lambda_5)$ is mapped into

$$\tilde{L}_0 = \begin{pmatrix} 0.1007934424 & 0.3903137711 & 0.1920535748 & 0.3387502373 & 0.2262286993 \\ 0.3903137711 & 0.0979036491 & 0.3899557789 & 0.1916216567 & 0.3387502373 \\ 0.1920535748 & 0.3899557789 & 0.1026058168 & 0.3899557789 & 0.1920535748 \\ 0.3387502373 & 0.1916216567 & 0.3899557789 & 0.0979036491 & 0.3903137711 \\ 0.2262286993 & 0.3387502373 & 0.1920535748 & 0.3903137711 & 0.1007934424 \end{pmatrix}.$$

Using $\tilde{L}_0$ as the initial, condition the convergence to the target Toeplitz matrix is much faster.

Numerical results for this problem with initial conditions L_0 and $\tilde{L}_0$ are reported below. The algorithm is:

1. First write the system into the Lax formulation $L' = [B(L), L]$.

2. For $t_n = nh, n = 0, 1, 2, \ldots$, where h is the stepsize of integration, integrate the Lie-group matrix differential equation $y' = B(yL_ny^T)y$, for $t \in [t_n, t_{n+1}]$ and initial condition $y(t_n) = I$, to obtain a numerical approximation y_{n+1}. At this stage, we employ various numerical methods.

3. Update the matrix L by means of the transformation $L_{n+1} = y_{n+1}L_ny_{n+1}^T$, so that the spectrum of L_{n+1} is the same as the spectrum of L_n, provided that y_{n+1} is an orthogonal matrix.

The 'exact' solution of the problem is evaluated by means of the Lie-group scheme of the Munthe-Kaas type based on the classical Gauss–Legendre RK of order six, with stepsize $h = \frac{1}{100}$. Table 4 refers to the 5×5 problem above and should be read as Table 3, except for

$e(\boldsymbol{\lambda})$ Denotes the error on the vectors of eigenvalues at $t_n = T = 1024$ and is evaluated as follows: let $\boldsymbol{\lambda}_0$ be the vector of all the (sorted) eigenvalues of L_0 and let $\boldsymbol{\lambda}_n$ be the vector of (sorted) eigenvalues of the numerical approximation L_n. Then

$$e(\boldsymbol{\lambda}) = \|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}_n\|_2,$$

when $t_n = T$.

Flops Floating point operations, evaluated using MATLAB™ built-in routine `flops`. This includes evaluation of commutators, exponentials, eventual similarity transformations and Picard's fixed point iterations for the implicit schemes and QR and polar factorization for the projective schemes. For Lie-group methods we evaluate the exponential by means of MATLAB™ built-in function `expm`. Because of the dimension of the matrices involved, the Euler–Rodriguez formula cannot be employed in this context.

All the relevant measurements are taken at time $t_n = T = 1024$. Table 4 indicates that isospectral methods, based on orthogonal or Lie-group methods, are generally more accurate than non-isospectral schemes. In particular, those schemes that do not preserve isospectrality, generally converge to an asymptotic state that is an $\mathcal{O}(h^p)$ perturbation of the right one, p being the order of the method. For this value of the stepsize, however, we note that the cost of isospectral methods is generally higher than that of the corresponding standard schemes $\mathtt{RK4_L}$ and $\mathtt{RK3_L}$.

In Table 5 the comparison between the methods is carried in a different manner. We depart from an initial matrix $\tilde{L}_0$ which has the same eigenvalues as L_0 but is nearly Toeplitz and has eigenvalues/eigenvectors already in an alternating order. Such matrix is obtained by means of the routine `toepinit` of Laurie [20]. We run the isospectral flows until the departure form a Toeplitz matrix is less that $\varepsilon = 10^{-10}$. Thus, T_{stop} represents the time at which each of the routines stops, 'Toeplitz error' is the departure from a Toeplitz matrix, evaluated as $\psi(L_n)$ at T_{stop}, '$e(\boldsymbol{\lambda})$' and 'Flops' are the error on the eigenvalues and the number of floating point operations respectively, evaluated as above. It is clear that starting with the initialized condition $\tilde{L}_0$ improves the numerical efficiency of the methods, since we have to integrate for shorter time in order to converge to a Toeplitz matrix. Moreover, the numerical experiments also show that this procedure improves quite a lot the error on the eigenvalues for non-isospectral methods.

To compare the numerical efficiency of the method `RKZ3`, $\mathtt{RK3_U}$ with QR and polar factorization, which yield isospectral but more expensive approximations, and $\mathtt{RK3_L}$ and $\mathtt{RK3_U}$ which are non-isospectral but cheaper, we display a plot of accuracy versus floating point operations. In Figure 11 we plot $e(\boldsymbol{\lambda})$ against the cost of the two methods sampled at $T = 113$ for various stepsizes, $h = \frac{1}{10}, \frac{1}{20}, \ldots, \frac{1}{320}$. The problem under consideration is 10×10 and has the initial value

$$\tilde{L}_0 = \mathtt{toepinit}(\boldsymbol{\lambda}_0),$$

where $\boldsymbol{\lambda}_0$ is the vector of eigenvalues of the Toeplitz matrix

$$\frac{1}{10}\begin{pmatrix} 1 & 2 & 3 & \cdots & 10 \\ 2 & 1 & 2 & \ddots & \vdots \\ 3 & 2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 2 \\ 10 & \cdots & 3 & 2 & 1 \end{pmatrix}.$$

The convergence to a Toeplitz matrix occurs around $T_{\text{stop}} \approx 1130$, so that the number of floating point operations at T_{stop} can be roughly speaking obtained by multiplying the flops at $T = 113$ by ten. The accuracy on the eigenvalues at T_{stop} is instead nearly the same, except to some mild accumulation of rounding error.

We observe that isospectral approximation (based on projective or Lie-group methods) produce far better results especially for larger stepsizes. The Lie-group method `RKZ3` is the most expensive, although it could be used as a 'projection scheme' by exponentiating only in the evaluation of y_{n+1}, while skipping the exponentiation at the internal stages.

We conclude that for this class of problems, which require conservation (i.e. a machine-accuracy retention of the eigenvalues) but at the same time possess a gradient-flow type behaviour since the solution converges to an asymptotic state, projection techniques are more efficient. Yet, the

Order	Iso	Method	Global error $T = 1024, h = 0.1$	$\lvert d(\mathcal{I}_2)\rvert$	$\lvert d(\mathcal{I}_3)\rvert$	$e(\boldsymbol{\lambda})$	Flops
4	y	`MagnusZ4` (GL)	8.8396e-12	7.6730e-12	1.3456e-11	2.9111e-12	3.2257e+08
	y	`RKMK4` (GL)	6.7749e-12	2.1103e-12	2.9887e-12	9.2833e-13	4.9627e+08
	y(n)[a]	$\mathtt{RK4}_{\mathtt{U}}$ (GL)	3.2321e-12	5.9521e-12	1.6911e-11	3.6038e-12	2.7416e+08
	n	$\mathtt{RK4}_{\mathtt{L}}$ (GL)	7.5596e-07	4.4409e-15	2.4323e-08	6.3271e-07	2.3760e+08
3	y	`RKZ3` (Kutta)	3.6613e-12	8.5794e-12	1.3050e-11	3.2886e-12	1.3010e+08
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, QR)	6.1482e-12	1.5269e-14	3.0077e-14	7.3132e-15	5.6235e+07
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, QR)[b]	6.1540e-12	3.9222e-14	6.5225e-14	1.4263e-14	7.1757e+07
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, P)	3.2620e-12	8.5290e-12	1.4449e-11	3.0078e-12	8.3668e+07
	y	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta, P)[c]	3.5650e-12	7.4736e-12	1.3029e-11	2.6624e-12	1.4813e+08
	n	$\mathtt{RK3}_{\mathtt{U}}$ (Kutta)	4.2290e-04	1.1221e-03	2.1980e-03	4.2227e-04	4.8472e+07
	n	$\mathtt{RK3}_{\mathtt{L}}$ (Kutta)	8.2909e-04	2.1957e-03	4.0996e-03	9.7449e-04	4.5333e+07

Table 4: Comparison of results for Problem 3, the 5×5 inverse Toeplitz eigenvalue problem with diagonal initial condition. The stage equations for the implicit schemes are solved by Picard iterations, with tolerance $\epsilon = 10^{-12}$.

[a] Because of weak orthogonality, see [13, 37].
[b] With QR factorization at each internal stage.
[c] With Polar factorization at each internal stage.

Order	Iso	Method	T_{stop}	Toeplitz error	$e(\boldsymbol{\lambda})$	Flops
	y	$\mathtt{MagnusZ4}$ (GL)	104.2	9.9008e-11	8.2242e-14	4.5370e+07
4	y	$\mathtt{RKMK4}$ (GL)	104.2	9.9008e-11	1.6803e-13	6.3683e+07
	y(n)[4]	$\mathtt{RK4_U}$ (GL)	104.2	9.9006e-11	1.3824e-13	3.9613e+07
	n	$\mathtt{RK4_L}$ (GL)	104.2	9.9005e-11	1.4090e-13	3.6122e+07
	y	$\mathtt{RKZ3}$ (Kutta)	104.2	9.9003e-11	6.7646e-14	1.5189e+07
3	y	$\mathtt{RK3_U}$ (Kutta, QR)	104.2	9.9006e-11	1.4863e-13	7.6370e+07
	y	$\mathtt{RK3_U}$ (Kutta, QR)[5]	104.2	9.9006e-11	1.4979e-13	9.1701e+06
	y	$\mathtt{RK3_U}$ (Kutta, P)	104.2	9.9005e-11	8.0003e-15	1.0154e+07
	y	$\mathtt{RK3_U}$ (Kutta, P)[6]	104.2	9.9004e-11	8.7997e-15	1.7127e+07
	n	$\mathtt{RK3_U}$ (Kutta)	104.2	9.9003e-11	3.0478e-09	6.8462e+06
	n	$\mathtt{RK3_L}$ (Kutta)	104.2	9.9004e-11	5.4477e-09	6.3565e+06

Table 5: Comparison of results for Problem 3, the 5×5 inverse Toeplitz eigenvalue problem, with initialized starting condition $\tilde{L}_0$. The stage equations for the implicit schemes are solved by Picard iterations, with tolerance $\epsilon = 10^{-12}$.

numerical results indicate that Lie-group schemes perform quite satisfactorily and are worth being investigated.

6.3 Concluding remarks

The Runge–Kutta formalism has inspired a number of papers which deal with the construction of numerical methods for nonlinear differential equation on Lie groups, on homogeneous manifolds and, with greater generality, on differentiable manifolds.

Among others, we mention the Lie-group RK methods of Munthe-Kaas [25], and methods of Owren and Marthinsen [30], a follow-up on the pioneering work of Crouch and Grossman [8]. In particular, the approach of Munthe-Kaas allows to convert every classical RK method (whether implicit or explicit) into a Lie-group scheme of the same order. Collocation methods can be obtained also by means of Munthe-Kaas' approach and they differ in principal error term from our Magnus-type collocation schemes.

There is a good intuitive explanation why one-step schemes (hence RK) are more successful than linear multistep methods in this context: in general, the manifolds that we want to recover numerically are nonlinear and for this reason, linear multistep methods are bound to fail [15]. Most of the off-the-shelf numerical methods are constructed subject the simplifying assumption that the solution evolves in a linear space. However, in many cases, especially when one wants to recover the underlying qualitative features of the problem, it is useful to describe the problem in a nonlinear setting. At the same time, we feel the need for new numerical methods which should incorporate as much as possible information obtained e.g by mathematical analysis [18]. In this context, it is useful to extract the ideas underpinning classical methods and see if it is possible to extend them to a more general construct. In this paper we have demonstrated that collocation is a very useful tool, as long as it is used properly. When we solve differential equations, we often identify the solution space, in which the solution of the differential equation evolves, with the tangent space, in which the vector field governing the equation varies. We think both of them as linear spaces. In the Lie-group context, however, the two spaces are different, have different structure and different composition rules are allowed. The Lie-group is a nonlinear manifold, the tangent space is linear. Collocation is, in other words, the interpolation of the vector field, defining the given differential equation, in the tangent space. Therefore, as long as the basic composition rules are respected, the resulting methods will preserve the Lie-group structure.

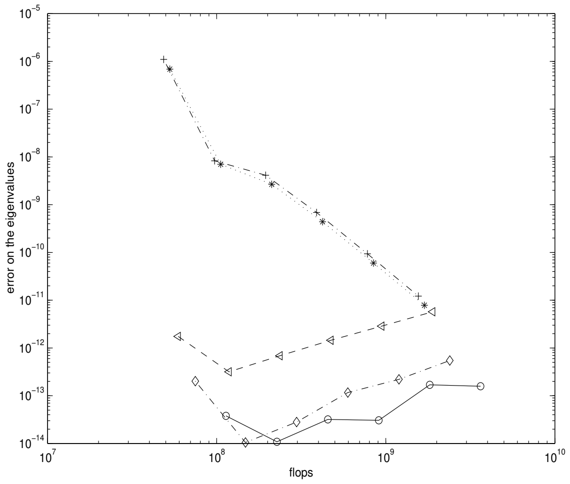


Figure 11: Accuracy on the eigenvalues $e(\boldsymbol{\lambda})$ versus floating point operations for the methods `RKZ3` (ovals and solid line), $\texttt{RK3}_{\texttt{U}}$ (asterisks and dotted line), $\texttt{RK3}_{\texttt{U}}$ with QR factorization (triangles with dashed line) and with polar decomposition (diamonds and dash-dotted line), and $\texttt{RK3}_{\texttt{L}}$ (pluses with dash-dotted line) at $T = 113$ for a 10×10 ITEP with initialized starting condition $\tilde{L}_0$. The stepsizes are $h = \frac{1}{10}, \frac{1}{20}, \ldots, \frac{1}{320}$.

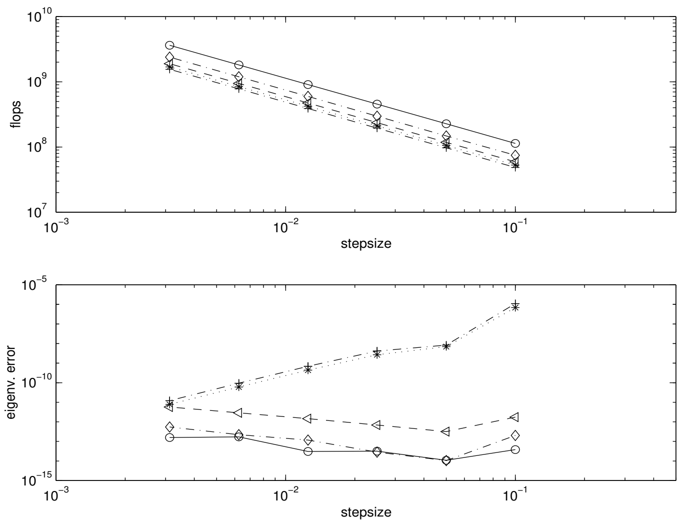


Figure 12: Floating point operations (top) and accuracy on the eigenvalues $e(\boldsymbol{\lambda})$ (bottom) versus stepsize for the methods `RKZ3`, $\texttt{RK3}_{\texttt{U}}$, $\texttt{RK3}_{\texttt{U}}$ with QR projection and polar decomposition, and $\texttt{RK3}_{\texttt{L}}$. Symbols as in Figure 11.

Acknowledgements

Many thanks to Arieh Iserles, Hans Munthe-Kaas, Brynjulf Owren and Walter Oevel for many valuable comments and suggestions on an earlier version of the present work.

References

[1] *MATLAB User's Guide*, 1992.

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

[3] M. V. Berry. Regular and irregular motion. In R.S. MacKay and J.D. Meiss, editors, *Hamiltonian Dynamical Systems*, pages 27–53. Adam Hilger, Bristol, 1987.

[4] M. P. Calvo, A. Iserles, and A. Zanna. Numerical Solution of Isospectral Flows. Technical Report NA1995/03, DAMTP, University of Cambridge, England, 1995. To appear in *Math. of Comp.*

[5] F. Casas. Solution of Linear Partial Differential Equations by Lie Algebraic Methods. *J. Comput. Appl. Math*, 76:159–170, 1996.

[6] M. T. Chu. A List of Matrix Flows with Applications. In A. Bloch, editor, *Hamiltonian ang Gradient Flows, Algorithms and Control*, volume 3 of *Fields Institute Communications*, pages 194–213. American Mathematical Society, Providence, 1994.

[7] M. T. Chu and K. R. Driessel. Can real symmetric Toeplitz matrices have arbitrary real spectra? Technical Report 88-1, Department of Mathematics, Idaho State University, 1989.

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

[9] P. Delsarte and Y. Genin. Spectral properties of finite Toeplitz matrices. In *Lecture Notes in Control and Inform. Sci.*, volume 58, pages 194–213. Springer, New York, 1984.

[10] F. Fer. Résolution de l'équation matricielle $\frac{\mathrm{d}}{\mathrm{d}t}U = pU$ par produit infini d'exponentielles matricielles. *Bull. de l'Acad. Roy. de Belgique, classe des Sciences*, 36:818–829, 1958.

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

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

[13] D. J. Higham. Time-Stepping and Preserving Orthonormality. *BIT*, 37(1):24–36, 1997.

[14] A. Iserles. Solving Linear Ordinary Differential Equations by Exponentials of Iterated Commutators. *Numer. Math.*, 45:183–199, 1984.

[15] A. Iserles. Multistep methods on manifolds. Technical Report NA1997/13, DAMTP, University of Cambridge, 1997.

[16] A. Iserles and S. P. Nørsett. Personal communication, 1997.

[17] A. Iserles and S. P. Nørsett. On the solution of differential equations in Lie groups. Technical Report NA1997/03, DAMTP, University of Cambridge, 1997.

[18] A. Iserles and A. Zanna. Qualitative numerical analysis of ordinary differential equations. In M. Shub J. Renegar and S. Smale, editors, *Lectures in Applied Mathematics*, volume 421–442. American Maths Soc., 1996.

[19] H. J. Landau. The inverse eigenvalue problem for real symmetric Toeplitz matrices. *J. Amer. Math. Society*, 7:749–767, 1994.

[20] D. P. Laurie, 1996. personal communication.

[21] P. Lax. Integrals of nonlinear evolution equations and solitary waves. *Comm. Pure Appl. Math.*, 147:467–490, 1968.

[22] R. S. MacKay. personal communication.

[23] W. Magnus. On the Exponential Solution of Differential Equations for a Linear Operator. *Comm. Pure and Appl. Math.*, VII:649–673, 1954.

[24] 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.

[25] H. Munthe-Kaas. High Order Runge–Kutta Methods on Manifolds. Technical Report NA1997/14, DAMTP, University of Cambridge, 1997. To appear in *Journal of Applied Numer. Math.*

[26] 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.

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

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

[29] B. O'Neill. *Elementary Differential Geometry*. Academic Press, New York, 1966.

[30] 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.

[31] A. M. Perelomov. *Integrable Systems of Classical Mechanics and Lie Algebras*. Birkhäuser, Basel, 1990.

[32] A. F. Rasmussen. Solving Lie-type equations numerically with Magnus series. Master's thesis, NTNU, Trondheim, Norway, 1997.

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

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

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

[36] A. Zanna. The method of iterated commutators for ordinary differential equations on Lie groups. Technical Report NA1996/12, DAMTP, University of Cambridge, 1996.

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

[38] A. Zanna and H. Munthe-Kaas. Iterated commutators, Lie's reduction method and ordinary differential equations on matrix Lie groups. In F. Cucker, editor, *Foundation of Computational Mathematics*, pages 434–441. Springer-Verlag, 1997.