On the Numerical Solution of Isospectral Flows

by Antonella Zanna
of Newnham College

A dissertation presented in the fulfillment of the requirements for the degree of Doctor in Philosophy at the University of Cambridge.
February 1998

On the Numerical Solution of Isospectral Flows

Antonella Zanna

In this thesis we address the problem of approximating numerically the solution of a class of matrix ordinary differential equations (ODEs) of the form

$$L' = [B(L), L], \qquad L(0) = L_0, \quad L \in \mathbb{R}^{d\times d},$$

where $B(L)$ is a matrix function of L. The above ODE is known as an *isospectral flow,* since the spectrum of the initial matrix L_0 is retained by the solution L. Such flows arise in a number of applications, ranging from Hamiltonian systems to well-known matrix algorithms in numerical analysis.

The first part of this dissertation is devoted to proving that most classical ODE methods cannot be isospectral, so that their error on the eigenvalues is generally of the same order of magnitude as the local truncation error of the method itself. Since isospectrality is the main feature of these flows, it is of substantial interest to devise numerical methods that share the same qualitative information. To this end, we exploit the fact that isospectrality can be obtained by means of the group action of the set of orthogonal matrices on the set of all matrices with prescribed eigenvalues and present a number of numerical schemes that preserve orthogonality and hence can be employed to devise isospectral numerical approximations. Inter alia we mention those based on Runge–Kutta Gauss–Legendre methods, which are weakly orthogonal, as well as semi-explicit schemes, which, instead of employing the orthogonal group action, use the action of $\mathrm{GL}(d)$.

Since isospectrality is equivalent to a group action, we also address the more general task of devising numerical methods for matrix Lie groups, although the schemes can be applied to general Lie groups, provided they are appropriately modified. Departing from two Lie-group and Lie-algebra expansions of the exact solution, the Fer and the Magnus series, and extending the classical approach of collocation to a Lie-algebra setting, we introduce a class of new schemes, whose coefficients are obtained by integrating the Lagrangian cardinal polynomials along the trees that feature in the underlying expansion. The collocation approach yields implicit numerical methods, therefore, in order to reduce the implicitness of the schemes, we relax the collocation conditions and present a general recipe for explicit third-order Lie-group schemes, based on classical Runge–Kutta methods.

Declaration

The work presented in this thesis is result of my own research, except for part of the material presented in Chapter 2 and Chapter 3, which is fruit of a collaboration with Arieh Iserles (DAMTP, University of Cambridge) and Mari Paz Calvo (Universidad de Valladolid, Spain).

Acknowledgements

To my Father

I decided to embark upon this Ph.D. experience after having met Dr Arieh Iserles first at the International Summer School in Mathematics in Perugia (1993) and then at the Advanced Course in Numerical Analysis in Cortona (1994), Italy. Since that time, I considered his way of teaching and discussing numerical analysis very enjoyable, challenging and stimulating, and that motivated me to continue my studies in this field. His mathematical erudition and enthusiasm, his jokes, as well as his continuous and unconditional support have been of great value to me during my stay in Cambridge. Many thanks also to the Numerical Analysis Group, to the old students and the new ones, and to all the visitors in the period between 1995 and 1997, for numerous discussions and interesting debates, as well for enjoyable free time spent together.

My second thanks go to my family in Italy. To my father, Giacomo, who died shortly before I started my Ph.D., and who strongly supported me throughout all my life, my choices and my studies. He would have been very proud of me now and I am most grateful to him for everything — he will always be present in my heart. Thanks to my mother, brother and sister, who have always been so near and supportive.

I would also like to express my gratitude to Newnham College, first for accepting me as a graduate student, and then for offering me a Research Studentship, a unique opportunity to continue my research here in Cambridge. I have made many friends in Newnham College. The delightful afternoons in the college garden will be always vivid in my memory.

My gratitude goes also to a number of people with whom I have discussed the material presented in this thesis and to those who have offered interesting comments or useful references.

A very special thank you goes to my husband, Hans, to Sondre and Njål and to my 'Norwegian family'. Hans and I met at a conference in Park City, Utah, in 1995 and since then we have shared life, interests and even mathematics! I owe very much to him, for his love, his support and affection. ;-)

Contents

1 Isospectral Flows **1**
- 1.1 Isospectral flows . . . 2
- 1.2 Some recurring examples: the Toda lattice . . . 3
- 1.3 Generalized QR flows . . . 8
- 1.4 Eigenvalue problems, list sorting and SVD . . . 9
- 1.5 Inverse eigenvalue problems for symmetric Toeplitz matrices . . . 11

2 Negative results for classical ODE methods **14**
- 2.1 Conserved integrals . . . 14
- 2.2 Numerical methods for ODEs . . . 16
- 2.3 Linear conservation laws . . . 17
- 2.4 Numerical methods and quadratic conservation laws . . . 18
  - 2.4.1 Analysis for LMMs . . . 19
  - 2.4.2 RK and quadratic invariants . . . 21
- 2.5 Cubic integrals under discretization . . . 24
  - 2.5.1 RK schemes and cubic invariants . . . 24
  - 2.5.2 OLMs and cubic integrals . . . 29
- 2.6 RK and higher order conservation laws . . . 33

3 Isospectral methods **41**
- 3.1 Isospectral solutions via the Lax formalism . . . 41
- 3.2 Numerical methods for Orthogonal Flows . . . 42
  - 3.2.1 Projection methods . . . 43
  - 3.2.2 Runge–Kutta schemes . . . 44
  - 3.2.3 Cayley transforms . . . 47
  - 3.2.4 Lie-group methods . . . 48
- 3.3 The early isospectral methods . . . 49
  - 3.3.1 The Lie-bracket method . . . 49
  - 3.3.2 Modified Gauss–Legendre schemes . . . 50
- 3.4 Semi-explicit methods for isospectral flows . . . 54

4 Lie groups and Lie-group methods **55**
- 4.1 Lie groups and their general properties . . . 55
  - 4.1.1 ODEs on Lie groups . . . 57

4.1.2 Some recurring examples . . . 57
4.2 Lie-group expansions and the exponential function . . . 61
4.2.1 The Fer expansion . . . 62
4.2.2 The Magnus expansion . . . 65
4.3 Order trees and exponential expansions . . . 66
4.3.1 Order trees for the Magnus expansion . . . 67
4.3.2 Order trees for the Fer expansion . . . 69
4.4 Collocation methods for the Fer and Magnus expansions . . . 73
4.4.1 Collocation . . . 73
4.4.2 Algebraic formulae for the coefficients and the weights . . . 82
4.4.3 Relaxed collocation . . . 84
4.4.4 Explicit order-three methods . . . 86
4.5 Other classes of Lie-group methods . . . 90
4.5.1 The RK-type Lie-group methods of Crouch and Grossman . . . 90
4.5.2 The RK-type Lie-group methods of Munthe-Kaas . . . 92

5 Numerical examples **96**
5.1 Toda lattices . . . 96
5.1.1 Three-particle periodic lattice . . . 97
5.1.2 The nonperiodic Toda lattice . . . 108
5.2 Matrix orbits visualization . . . 110
5.3 The inverse Toeplitz eigenvalue problem (ITEP) . . . 113
5.4 Tables and comments . . . 117
5.4.1 Problem 1: Three-particle periodic Toda lattice . . . 117
5.4.2 Problem 2: Three-particle nonperiodic Toda Lattice . . . 121
5.4.3 Problem 3: The inverse Toeplitz eigenvalue problem . . . 121

6 Conclusions **128**

Bibligraphy **130**

1

Isospectral Flows

In this thesis we are concerned with the problem of devising numerical methods for time integration of a class of Ordinary Differential Equations (ODEs) known under the name of *isospectral flows*. The course of evolution of the aforementioned dynamical systems is such that the matrix in question undergoes an isospectral deformation, that leaves the eigenvalue unchanged. In other words, the eigenvalues are integrals of the motion.

Some isospectral flows are completely integrable systems, so that in principle the corresponding equation of motions can be reduced to quadrature, i.e. the solution can be obtained by finitely many algebraic operations and by evaluating integrals of known functions (Perelomov 1990). In spite of this, explicit integration is a more challenging problem, and often numerical integration constitutes the last resort. Vice versa, in many cases numerical integration is itself employed to investigate and understand the dynamics of the underlying system.

The numerical methods for finite-dimensional matrix problems, that we present in this thesis, are isospectral, hence they exactly (i.e. to machine accuracy) preserve the integrals of the motion of the underlying isospectral flow. Our original motivation was the empirical observation that classical *off-the-shelf* methods of numerical analysis do not generally preserve all the integrals of motion. This is rigourously proved in this dissertation, thereby concluding that if we want isospectral numerical methods, we need to follow a different approach.

The thesis is organized as follows. In the first chapter, we discuss some theory behind isospectral flows, and we illustrate some applications in which isospectral flows arise. These examples range from classical mechanics, like Toda flows and N-body systems, to linear algebra, like direct and inverse eigenvalue problems, list sorting, Singular Value Decompositions (SVD), not to mention a variety of flows that arise in conjunction with unconstrained optimization.

In the second chapter, we discuss classical numerical methods for ODEs (i.e. Linear Multistep Methods, one-leg methods and Runge–Kutta schemes) and study their retention of isospectrality. We discuss how well these methods can retain conservation laws, since this indicates whether or not the schemes can preserve the integrals of the motion. The results of our analysis indicate that these classical schemes behave very poorly in this respect.

In the third chapter, we introduce isospectral methods via the Lax formalism. The isospectral problem is reduced to finding the solution of an *orthogonal flow* and orthogonal methods are thereby discussed. Orthogonality is essentially equivalent to a quadratic conservation law, and numerical methods that preserve quadratic invariants can therefore

be employed.

In Chapter 4 we find very useful to consider orthogonal flows as flows on *Lie groups.* We hence introduce a class of numerical methods suitable for solving ODEs on Lie groups. These new methods, based on Lie-group and Lie-algebra operations, take advantage of two expansions of the exact solution known as the *Fer* and the *Magnus* expansions. This allows us to construct isospectral methods of arbitrarily high order, and, in particular, to construct explicit numerical schemes.

To conclude, in Chapter 5 we present some numerical examples with application of the proposed methods, whereby we experiment with schemes and different approaches and eventually compare with some classical methods.

1.1 Isospectral flows

Isospectral flows are characterized by the matrix differential equation

$$L' = [B, L], \qquad L(0) = L_0, \tag{1.1}$$

where $L, B \in \mathbb{F}^{d\times d}$, the set of $d \times d$ matrices with entries in $\mathbb{F}$, and L_0 is a given $d \times d$ initial matrix. Here, and in the sequel, the $\mathbb{F}$ stands for either $\mathbb{R}$ or $\mathbb{C}$. The matrix function $B \equiv B(t, L)$ depends on L and, possibly, on the time t. The square brackets denote the commutator *(Lie bracket)* on matrices, $[B, L] = BL - LB$. The functions B and L that obey the differential equation (1.1) are usually called a *Lax pair* (Lax 1968).

It is easy to verify that the solution $L(t)$ of (1.1) evolves according to an isospectral deformation: First of all, we associate to (1.1) the differential equation

$$U' = B(t, L)U, \qquad U(0) = I, \tag{1.2}$$

where I is the identity matrix. Then, we construct the matrix function $U(t)^{-1}L(t)U(t)$ and observe that, by virtue of (1.1) and (1.2), together with

$$\frac{\mathrm{d}}{\mathrm{d}t}U^{-1} = -U^{-1}U'U^{-1},$$

one has

$$\frac{\mathrm{d}}{\mathrm{d}t}U(t)^{-1}L(t)U(t) = O,$$

where O denotes the zero matrix. Thus, the matrix function $U(t)^{-1}L(t)U(t)$ is time independent, hence it must equal at all times its initial condition, obtained by substituting $t = 0$. It follows that

$$L(t) = U(t)L_0U(t)^{-1} \tag{1.3}$$

is a similarity transformation that leaves the eigenvalues unchanged.

As we shall see in the sequel, in many important examples the matrix function B is skew-symmetric. Then the equation (1.2) is an *orthogonal flow,* since its solution $U(t)$ is an orthogonal matrix, i.e. $UU^T = U^TU = I$. In that case, $U^{-1} = U^T$, hence (1.3) becomes

$$L(t) = U(t)L_0U(t)^T, \tag{1.4}$$

and, provided that the matrix L_0 is symmetric, also $L(t)$ remains symmetric for all time t. Unless otherwise mentioned, we will assume that the solution of (1.1) is of the form (1.4) throughout this thesis.

1.2 Some recurring examples: the Toda lattice

In the present and in the following sections we introduce some problems that can be modelled in order to obey the isospectral equation (1.1). As we shall see, isospectral flows arise in a number of traditionally rather disparate areas of mathematics. To cite Bloch (1994), "one of the key interactions is that between dynamical systems and algorithms."

The first example, the Toda lattice and the completely integrable N-body problem, arise in classical mechanics and are obtained by means of a transformation of variables in the Hamiltonian equations that leads to the Lax form (1.1).

Returning to Toda lattices, we recall that those are one-dimensional lattices of particles whose motion is described by a nearest-neighbour interaction of an exponential type and can be used to model a continuum of flows, ranging form the hard-sphere limit to the atomic case (Toda 1981, Toda 1989).

Assume that the particles all have the same mass m (lattice without impurities) and that y_j denotes the displacement of the j-th particle. Then the equations of motion are given by

$$m\frac{\mathrm{d}^2 y_j}{\mathrm{d}t^2} = \frac{\mathrm{d}}{\mathrm{d}r}\phi(y_{j+1}-y_j) - \frac{\mathrm{d}}{\mathrm{d}r}\phi(y_j - y_{j-1}), \qquad j = 1, 2, \ldots, \tag{1.5}$$

where $\phi(r)$ is the interaction potential. Introducing the momenta $p_j = my_j'$ and the generalized coordinates $q_j = y_j$, $j = 1, 2, \ldots$, and letting ϕ be the exponential potential

$$\phi(r) = \mathrm{e}^{-r} + r,$$

so that $\phi'(r) = -\mathrm{e}^{-r} + 1$, the equations of motion (1.5) become the Hamiltonian system

$$\begin{aligned} q_j' &= \frac{1}{m}p_j, \\ p_j' &= \mathrm{e}^{-(q_j - q_{j-1})} - \mathrm{e}^{-(q_{j+1}-q_j)}, \end{aligned} \tag{1.6}$$

and, for simplicity sake, we have assume $m = 1$. The corresponding Hamiltonian function is

$$H(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2}\sum_j p_j^2 + \sum_j (\mathrm{e}^{-(q_{j+1}-q_j)} - 1),$$

where $\boldsymbol{p} = \{p_j\}$ and $\boldsymbol{q} = \{q_j\}$.

To write (1.6) in the isospectral form (1.1), we present the procedure of Toda (1981) following the approach of Flaschka (1974). Introducing the change of variables

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

so that

$$\begin{aligned} \alpha_j' &= \alpha_j(\beta_j - \beta_{j+1}), \\ \beta_j' &= 2(\alpha_{j-1}^2 - \alpha_j^2), \end{aligned}$$

we obtain the Lax form (1.1), where

$$L = \begin{bmatrix} \beta_1 & \alpha_1 & 0 & \dots & 0 \\ \alpha_1 & \beta_2 & \alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \alpha_{d-1} \\ 0 & \dots & 0 & \alpha_{d-1} & \beta_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -\alpha_1 & 0 & \dots & 0 \\ \alpha_1 & 0 & -\alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha_{d-1} \\ 0 & \dots & 0 & \alpha_{d-1} & 0 \end{bmatrix}. \quad (1.7)$$

Such matrices L and B correspond to the nonperiodic lattice, when the first and the last particle do not interact with each other. This can be obtained by introducing two auxiliary particles such that $q_0 = -\infty$ and $q_{d+1} = +\infty$.

In the periodic case,

$$\alpha_{j+d} = \alpha_j, \qquad \beta_{j+d} = \beta_j,$$

and

$$L = \begin{bmatrix} \beta_1 & \alpha_1 & 0 & \dots & \alpha_d \\ \alpha_1 & \beta_2 & \alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \alpha_{d-1} \\ \alpha_d & \dots & 0 & \alpha_{d-1} & \beta_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -\alpha_1 & 0 & \dots & \alpha_d \\ \alpha_1 & 0 & -\alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha_{d-1} \\ -\alpha_d & \dots & 0 & \alpha_{d-1} & 0 \end{bmatrix}. \quad (1.8)$$

The solution of the Toda lattice equations has been discussed in a number of papers. In the nonperiodic case (1.7), the motion of the particles is *asymptotically free,* in the sense that the particles are scattered to an infinite distance from each other and asymptotically behave as free particles (see Figure 1.1).

Moser (1975*b*) and Kac & van Moerbeke (1975) showed that the positions q_j are related to some rational functions of $\exp(\lambda_k t)$, for $k = 1, 2, \dots, d$. Here λ_k is the k-th eigenvalue *(asymptotic momentum)* and it appears as the k-th diagonal element of L as $t \to \infty$, when L tends to a diagonal matrix.

In Figure 1.1 we have plotted the positions and the momenta of the first six particles of the symmetric nonperiodic lattice described by the differential equations (1.6) and corresponding to the initial conditions

$$\boldsymbol{q}(0) = [-5, -4, \dots, 0, \dots, 4, 5]^T,$$

and

$$p_j(0) = \begin{cases} 4, & 1 \le j \le 5, \\ 0, & j = 6, \\ -4, & 7 \le j \le 11. \end{cases}$$

Unlike the nonperiodic system, the periodic Toda lattice (1.8) is not asymptotically free and its motion is by far more complicated. In some special cases, solution of the equations of motion can be written in terms of theta functions by using some algebro-geometric methods. We refer the reader to (Toda 1981, Toda 1989) and (Perelomov 1990) for further reading.

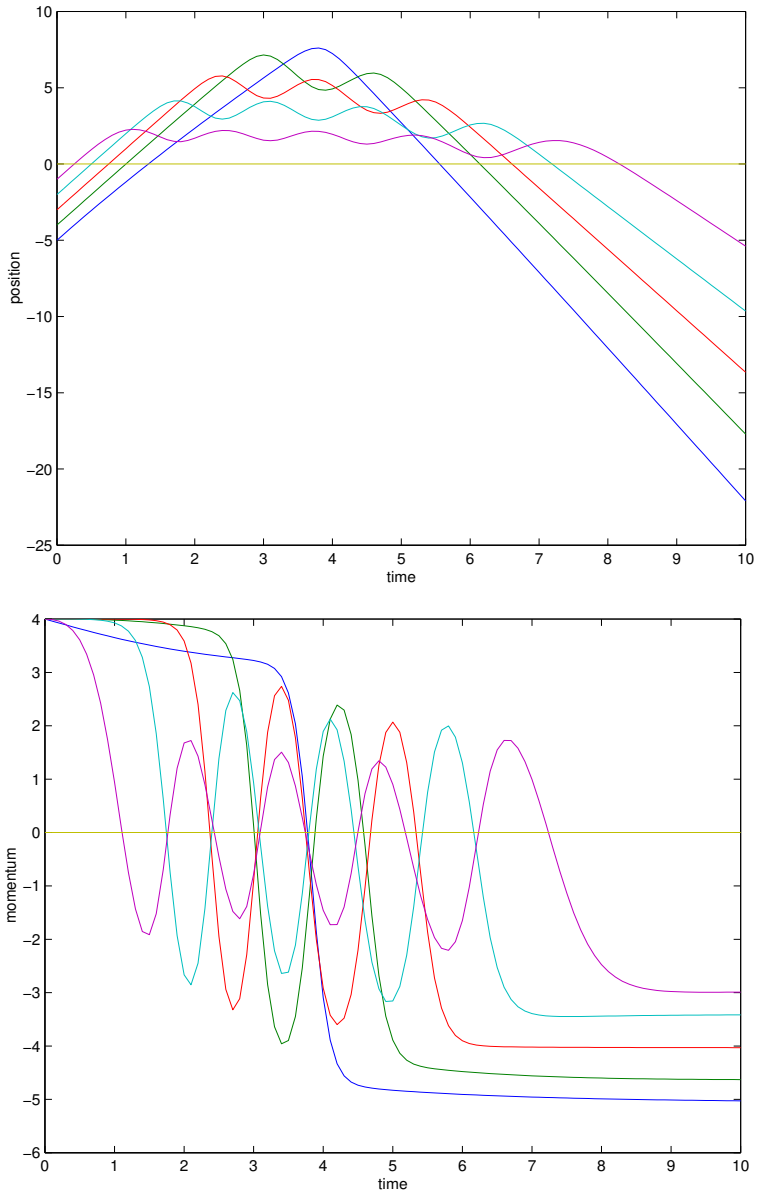


Figure 1.1: Positions and momenta of the first six particles of a symmetric eleven-particles nonperiodic Toda lattice.

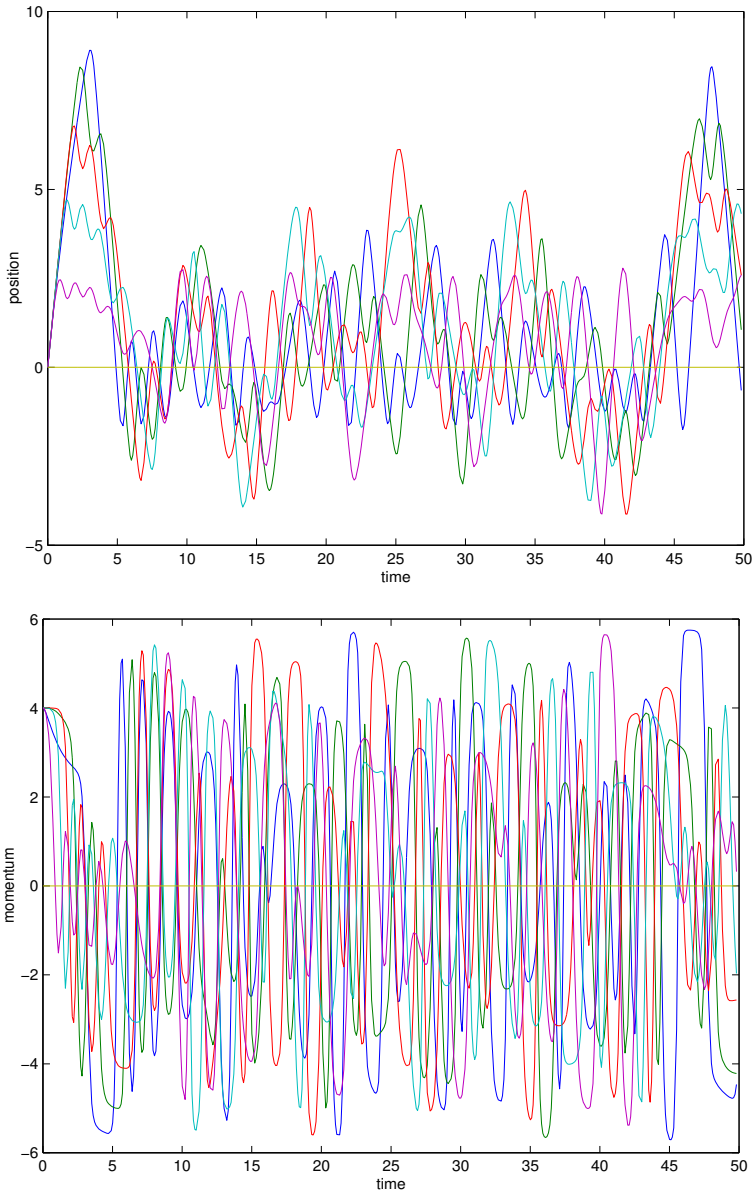


Figure 1.2: Positions and momenta for the first six particles of a symmetric eleven-particles periodic Toda lattice.

In Figure 1.2 we have plotted the positions and the momenta of the first six particles of the symmetric periodic lattice described by the differential equations (1.6) and corresponding to the initial conditions

$$q_j(0) = 0, \qquad j = 1, \dots, 11,$$

and

$$p_j(0) = \begin{cases} 4, & 1 \le j \le 5, \\ 0, & j = 6, \\ -4, & 7 \le j \le 1. \end{cases}$$

Toda flows are a classical example of an *integrable system* (Perelomov 1990). Integrable systems are of particular interest because when a system is integrable, then it is possible to write down its solution in a finite number of algebraic operations. Furthermore, the trajectories relative to an initial condition live on a manifold which is either a d-torus, T^d, or the cartesian product of an euclidean space and a torus, $\mathbb{R}^k \times T^{d-k}$, for some $k \le d$, so that each trajectory of the system lives on a manifold that is determined by its initial condition. Whenever a given dynamical can be put in the Lax form (1.1), and provided that it has a sufficient number of independent integrals, then it is integrable and its solution possesses the nice properties described above (Perelomov 1990). When an Hamiltonian system is non integrable, some of the torii degenerate and chaotic trajectories appear instead (KAM theory, see the review article of Berry (1987) or (Moser 1973)). Toda lattices arose in this context, when Morikazu Toda constructed an integrable system as an approximation to the Hénon–Heiles system, used to study the motion of stars in a galaxy with cylindrical symmetry, which is instead non-integrable (Toda 1981).

Other systems of N given bodies that are integrable have also been studied and classified. It turns out that the most interesting systems are also one-dimensional, like the Toda lattice, and have the function

$$H = \frac{1}{2} \sum_{j=1}^{N} p_j^2 + g^2 \sum_{\substack{j,k=1 \\ j<k}}^{N} \phi(q_j - q_k), \qquad g = \text{const}, \tag{1.9}$$

as their Hamiltonian, so that the equations of motion are given by

$$\begin{aligned} p_j' &= -\frac{\partial H}{\partial q_j}, \\ q_j' &= p_j. \end{aligned} \tag{1.10}$$

Not all possible potentials ϕ, though, yield integrable systems. In order to be integrable, the potential $\phi(\xi)$ must be one of the following ones,

$$\phi(\xi) = \begin{cases} -\xi^2, & \text{I} \\ \dfrac{a^2}{\sinh^2(a\xi)}, & \text{II} \\ \dfrac{a^2}{\sin^2(a\xi)}, & \text{III} \\ a^2 \mathcal{P}(a\xi), & \text{IV} \end{cases} \tag{1.11}$$

where a is a parameter and $\mathcal{P} = \mathcal{P}(\xi, \omega_1, \omega_2)$ is the Weierstrass function of complex variable ξ and periods $2\omega_1$ and $2\omega_2$ (Abramowitz & Stegun 1965, Rainville 1967). The formulae that yield the Lax pair L, B, were introduced by Calogero (1975). Their entries of L and B are

$$\begin{aligned} L_{j,k} &= p_j \delta_{j,k} + \mathrm{i}g(1-\delta_{j,k})x(q_j - q_k), \\ B_{j,k} &= -\mathrm{i}g\Big[\delta_{j,k}\Big(\sum_{\substack{\ell,j=1 \\ \ell \neq j}}^{N} z(q_j - q_k)\Big) - (1-\delta_{j,k})y(q_j - q_\ell)\Big], \qquad j,k = 1,2,\ldots,N. \end{aligned} \tag{1.12}$$

The values of the unknown functions $x(\xi), y(\xi)$ and $z(\xi)$ appearing above in (1.12) are given by

$$x(\xi) = \begin{cases} \xi^{-1}, & \text{I} \\ a\coth(a\xi), \dfrac{a}{\sinh(a\xi)}, & \text{II} \\ a\operatorname{ctg}(a\xi), \dfrac{a}{\sin(a\xi)}, & \text{III} \\ a\dfrac{\operatorname{cn}(a\xi)}{\operatorname{sn}(a\xi)}, a\dfrac{\operatorname{dn}(a\xi)}{\operatorname{sn}(a\xi)}, \dfrac{a}{\operatorname{sn}(a\xi)}, & \text{IV} \end{cases} \tag{1.13}$$

$\operatorname{cn}, \operatorname{sn}, \operatorname{dn}$ being Jacobi elliptic functions (Abramowitz & Stegun 1965, Rainville 1967),

$$y(\xi) = -x'(\xi), \tag{1.14}$$

and

$$z(\xi) = \frac{x''(\xi)}{2x(\xi)}. \tag{1.15}$$

We refer the reader to Perelomov (1990) for a description of the solutions corresponding to each potential.

1.3 The QR flow and its generalizations

A *QR flow* can be viewed as a generalization of a nonperiodic Toda flow. Assume that f is a function which is analytic on an open set containing the spectrum of L_0, the initial condition of (1.1). Then a (generalized) QR flow is obtained by letting

$$B(L) = f(L)_- - f(L)_+, \tag{1.16}$$

the subscripts $-$ and $+$ denoting the lower and upper triangular parts of $f(L)$ respectively. Thus nonperiodic Toda flows are obtained when $f(x) = x$ and L_0 is symmetric and tridiagonal.

Assume that L is symmetric and let $Q\Lambda Q^T$ be its diagonalization, Q being an orthogonal matrix and Λ a diagonal matrix. Then also $f(L)$ is symmetric, since it is a sum of terms of the form $Q\Lambda^k Q^T$. Thus it follows that $B(L)$ is skew-symmetric and the solution $L(t)$ evolves on the set of symmetric matrices. Furthermore, it is also possible to show that the bandwidth (tridiagonality) of L is retained (Deift, Nanda & Tomei 1983).

The name 'QR flow' originates from the fact that for a symmetric positive definite initial condition L_0 there exists a connection between the isospectral flow (1.1) and the iterates of the familiar QR method for finding the eigenvalues of a matrix. We briefly recall that the QR method for eigenvalues consists in finding the decomposition

$$L_k = Q_k R_k, \qquad k = 0, 1, 2, \ldots$$

where Q_k is an orthogonal matrix and R_k is an upper triangular matrix, with positive diagonal entries (Golub & van Loan 1989). The next matrix in the sequence is then obtained as

$$L_{k+1} = R_k Q_k$$

and, when L_0 is symmetric and positive definite, the method converges at an exponential speed to a diagonal matrix whose entries are the eigenvalues of L_0. It is easy to see that

$$L_{k+1} = Q_k^T L_k Q_k,$$

rendering the connection with the isospectral flows is less obscure. If L_0 is symmetric and positive definite, choosing $f(x) = \log x$ in (1.16), the matrices $L(1), L(2), \ldots$, interpolate the iterates $L_1, L_2, \ldots$, obtained this time with the QR method for eigenvalues.

Such a connection was discovered by Symes (1982) and then widely investigated by Nanda (1982), Deift et al. (1983), Deift, Rivera, Tomei & Watkins (1990), Chu (1984), Chu & Driessel (1989), Lagarias (1990) and many others. Among some of its relevant features, (1.16) can be generalized in such a way to include the theory of a number of other matrix flows that are related to linear algebra methods for finding eigenvalues. Among these, we mention the *LU flow,* the *Cholesky flow,* the *power method,* and so on. See the review articles (Chu 1988, Chu 1994, Watkins 1984) and (Calvo, Iserles & Zanna 1995) for a more comprehensive list of references to related work.

1.4 Eigenvalue problems, list sorting algorithms and Singular Value Decompositions

Motivated by the relations between the Toda flow and the QR method for finding the eigenvalues of a matrix, in the last few years a number of researchers devoted themselves to devising isospectral problems, having in mind alternative methods for finding i) the eigenvalues of a given matrix; ii) the singular values[1] of an assigned rectangular matrix; and iii) continuous version of a number of iterative processes of linear algebra. Besides the Toda and the QR flows, described above in Section 1.2 and in Section 1.3 respectively, one of the most popular tools for i), ii) and iii), is the *double-bracket flow,*

$$L' = [[N, L], L], \qquad L(0) = L_0, \tag{1.17}$$

[1]Let L_0 be a $d \times k$ matrix. Then, there exist an orthogonal $d \times d$ matrix U, a $d \times k$ diagonal matrix Σ and a $k \times k$ orthogonal matrix V such that $L_0 = U\Sigma V^T$. Such decomposition is called *singular value decomposition* (SVD) and the diagonal entries of Σ are called *singular values of* L_0. See (Golub & van Loan 1989).

introduced by Brockett (1991). The matrix N is usually a preassigned symmetric matrix, that without loss of generality, we might assume to be diagonal. Note that, letting $B(L) = [N, L]$, we recover the general equation (1.1). Moreover, if we choose $N = \mathrm{diag}(1, 2, \ldots, d)$, then the resulting equations coincide with those of the nonperiodic Toda lattice (1.7).

Double-bracket flows, besides being isospectral, possess another important feature: they are *gradient flows.* If we let U be the solution of the orthogonal flow $U' = [N, L]U$ with initial condition $U(0) = I$ and introduce the scalar function

$$\psi(U) = \mathrm{tr}(UL_0U^TN),$$

then, along the solution of (1.17), one has

$$\psi' = -\langle [U, \psi_U], [U, \psi_U] \rangle_K$$

(Brockett 1993), where $\langle \cdot\,, \cdot \rangle_K$ denotes the *Killing* form

$$\langle x, y \rangle_K = \mathrm{tr}([\mathrm{ad}x \circ \mathrm{ad}y]),$$

which, in the case of skew-symmetric matrices, reduces to $\langle A, B \rangle_K = -\mathrm{tr}(AB^T)$ (Gallot, Hulin & Lafontaine 1990). In practice, the advantage of having gradient flows is that their dynamic is relatively simple. As $t \to \infty$, the function ψ, that plays the role of a *Lyapunov* function, i.e. it is monotonically decreasing along the trajectory of the system, approaches a constant and so does the solution of (1.17).

We have mentioned that the solution of the Toda flow (1.7) tends to a diagonal matrix whose entries (the eigenvalues) are sorted from the smallest to the largest (Moser 1975*a*). In the double-bracket case (1.17) it is possible to choose the ordering of the eigenvalues by choosing different matrices N, the eigenvalues of L being asymptotically ordered in the same fashion as the entries of N (Brockett 1991). Because of this feature, double-bracket flows have been suggested for solving optimization problems, and especially for sorting lists and rearranging functions. We refer to (Brockett 1991) for more details of such applications.

In the SVD context, isospectral flows were first considered by Chu (1986) who analysed the Toda case. Deift, Demmel, Li & Tomei (1991), proposed a more general flow of Toda type,

$$B(L) = f(L)^T_- - f(L)_-, \tag{1.18}$$

suitable for invertible $d \times d$ matrices. Deift et al. (1991) chose the function f as $f(x) = \log x$. The flow proposed by Chu can be recovered letting $f(x) = x$ in (1.18).

A flow of a double-bracket type for SVD was proposed by Smith (1991) and similar results were presented by Helmke & Moore (1990). We follow Smith (1991) and let L_0 and N be two given $d \times k$ matrices. Hence we consider the flow

$$\Sigma' = \Sigma[\![\Sigma^T, N^T]\!] - [\![\Sigma, N]\!]\Sigma, \qquad \Sigma(0) = L_0, \tag{1.19}$$

where, given two $m \times \ell$ matrices, the bracket $[\![\cdot\,, \cdot]\!] : \mathbb{R}^{m\times\ell} \times \mathbb{R}^{m\times\ell} \to \mathfrak{so}(m)$ is the bilinear operator

$$[\![A, B]\!] = \frac{1}{m-2}(AB^T - BA^T), \qquad m \geq 3.$$

The symbol $\mathfrak{so}(m)$ denotes the set of $m \times m$ skew-symmetric matrices in Lie-group formalism.

With regards to the asymptotic state, we summarize here the results proved in the aforementioned papers. Both (1.18) and (1.19) converge to unique attractive equilibrium point of the system, and, provided that the singular values of L_0 are all distinct, the convergence is exponential. In general, this system has $k!$ equilibria, all but one being saddle points. When $t \to \infty$, the singular values will appear along the main diagonal of Σ, and will be arranged in increasing order for the flow of type (1.18) if f is monotonically increasing, while they will be ordered similarly to the entries of N in the case (1.19).

1.5 Inverse eigenvalue problems for symmetric Toeplitz matrices

To introduce the inverse eigenvalue problem, we follow the formulation of Friedland, Nocedal & Overton (1987). Thus, let $A(\boldsymbol{c})$ be the affine family

$$A(\boldsymbol{c}) = A_0 + \sum_{k=1}^{d} c_k A_k,$$

where $\boldsymbol{c} \in \mathbb{R}^d$ and $\{A_k\}$ are real symmetric $d \times d$ matrices. Denote the eigenvalues of $A(\boldsymbol{c})$ by $\{\lambda(\boldsymbol{c})\}_1^d$, where

$$\lambda_1(\boldsymbol{c}) \leq \lambda_2(\boldsymbol{c}) \leq \cdots \leq \lambda_d(\boldsymbol{c}).$$

Then the following problem is called *an inverse eigenvalue problem (IEP).*
IEP: Given d real numbers $\lambda_1^* \leq \cdots \leq \lambda_d^*$, find $\boldsymbol{c} \in \mathbb{R}^d$ such that $\lambda_i(\boldsymbol{c}) = \lambda_i^*$, for $i = 1, 2, \ldots, d$.

Inverse eigenvalue problems arise in various areas of application. A classical example is the *discrete* inverse Sturm–Liouville problem,

$$\frac{-u_{k+1} + 2u_k - u_{k-1}}{h^2} + \phi_k u_k = \lambda_j^* u_k, \qquad k = 1, \ldots, n, \quad u_0 = u_{n+1} = 0,$$

where the spectrum $\lambda_1^*, \ldots, \lambda_n^*$ is assigned and one wishes to find the discrete potential ϕ_k. Similar applications arise in nuclear spectroscopy, in which case the λ_j^* are obtained from experimental measurements (Friedland et al. 1987).

Although not much is known with regards to the existence and uniqueness of the solution of the IEP in the most general case, there exists a large literature on such conditions in many special cases. With regards to numerical methods, some (locally) quadratically convergent schemes for the IEP are reviewed in (Friedland et al. 1987).

However, if we are interested only in the final stage $A \equiv A(\boldsymbol{c})$, it makes perfect sense to construct an isospectral flow that starts from the diagonal matrix with entries $\lambda_1^*, \ldots, \lambda_d^*$ and converges (hopefully) to a target matrix A. In this case, since the flow is isospectral, A will have eigenvalues $\lambda_1^*, \ldots, \lambda_d^*$, and hence will be a solution of the inverse eigenvalue problem.

Needless to say, the corresponding isospectral flow has to be solved numerically. Thus, it is crucial to employ a numerical method that is isospectral. Otherwise, the numerical

error due to our ODE solver will influence so much the numerical solution A that we will eventually converge to an approximate solution $\tilde{A}$ that possesses the wrong eigenvalues!

Among IEPs, the most fascinating is the inverse problem for Toeplitz matrices (ITEP). Toeplitz matrices are constant along their diagonals. Until a few years ago, it was an open problem (posed by P. Delsarte and Y. Genin) whether or not real symmetric Toeplitz matrices can have arbitrary real eigenvalues. The question was first solved by Delsarte & Genin (1984) for $d \leq 4$, and then, in the general case, by Landau (1994), subject to some regularity assumptions. Unfortunately, to cite Landau's words, "The argument—based on topological degree—is non constructive", and also: "It is tempting to conjecture that the above correspondence is one-to-one. Of course, It would also be very interesting to find algorithms for carrying it out".

With regards to numerical methods for the ITEP, a discrete isospectral deformation method was proposed by Laurie in 1991. The algorithm employs plane matrix rotations to update the eigenvectors' matrix (Laurie 1991). However, we are here more interested to an ODE approach, therefore we follow Chu & Driessel (1989), who suggested the isospectral flow

$$L' = [B(L), L], \qquad L_0 = \operatorname{diag}(\lambda_1^*, \ldots, \lambda_d^*),$$

where $B(L)$ is a Toeplitz annihilator, in other words, a skew-symmetric matrix function B such that

$$B(L) = O \iff L \in \mathcal{T}, \tag{1.20}$$

where $\mathcal{T}$ denotes the set of $d \times d$ symmetric Toeplitz matrices. Although there are many ways to construct a skew-symmetric function B that also satisfies the condition (1.20), we consider the function proposed in (Chu & Driessel 1989),

$$B_{i,j} = \begin{cases} L_{i+1,j} - L_{i,j-1}, & \text{if } 1 \leq i < j \leq d, \\ 0, & \text{if } i = j, \quad i,j = 1, \ldots, d, \\ L_{i-1,j} - L_{i,j+1}, & \text{otherwise.} \end{cases} \tag{1.21}$$

Unfortunately, not much is known about the convergence of such flow, although numerical experiments seem to indicate that the algorithm converges globally (Chu 1994). We will use the isospectral flow (1.1) together with B of the form (1.21) for our numerical experiments in Chapter 5.

Comments

The isospectral deformation method was introduced by P. Lax in connection with the Korteg–de Vries equation (KdV)

$$u_t + uu_x + u_{xxx} = 0$$

which describes waves in shallow water (Lax 1968). The method consists of associating to a differential equation (partial or ordinary) a Lax pair (1.1), where, however, L and B are two operators (selfadjoint and antisymmetric respectively) over some Hilbert space. The first application of the method to a system with a finite number of degrees of freedom

is due to Flaschka (1974), who applied the method to the Toda lattice. Other pioneering investigations were being made, independently and at the same time, by Manakov and Moser. In particular, Moser (1975*b*) suggested first the term *isospectral deformation method.* We refer to (Perelomov 1990) for more historical details as well as more background and theory on integrable systems.

Given a differential equation with a sufficient number of integrals of motion, the hard job is to find, whenever it exists, a Lax representation. This is usually done by the method of *projection,* which is described at grater length in (Perelomov 1990). Nowadays, Lax's representations are known for a wide class of dynamical systems. All the systems that we consider in this thesis are already given in their (matrix) Lax representation.

2

Negative results for classical ODE methods

In this chapter we translate the isospectrality of the system (1.1) into d algebraic conditions *(conserved integrals).* To be isospectral, a numerical method for ODEs has to obey all of them simultaneously.

We introduce the two classes of methods for ODEs that are most commonly used, i.e. *Linear Multistep Methods* (LMMs), one-leg methods (OLMs) and *Runge–Kutta* schemes (RK). It follows from our analysis that, although RK schemes can do as well as preserving quadratic conservation laws (and hence be isospectral up to $d = 2$), none of these schemes can be isospectral when $d \geq 3$, in the sense that, for each given numerical method, it is possible to formulate an isospectral flow for which the method fails to be isospectral.

2.1 Conserved integrals

Before analysing numerical methods for ODEs and whether they retain isospectrality, it is necessary to translate isospectrality in terms of algebraic relations that can be verified for each numerical schemes.

We depart from the matrix $L(t)$, the exact solution of (1.1) and consider its d-degree characteristic polynomial

$$p(\lambda) = \det(\lambda I - L) = \lambda^d - p_{d-1}\lambda^{d-1} + \cdots + (-1)^d p_0, \tag{2.1}$$

whereby the zeros of $p(\lambda)$ are the eigenvalues of L. The coefficients appearing in (2.1) can be expressed in terms of the matrix L and its principal minors, or as elementary symmetric polynomials in its eigenvalues (Gantmacher 1977, Golub & van Loan 1989). Explicitly we have

$$\begin{aligned} p_{d-1} &= \lambda_1 + \cdots + \lambda_d, & (2.2)\\ p_{d-2} &= \sum_{i=1}^{d}\sum_{j=1}^{d} \lambda_i\lambda_j, & (2.3)\\ &\vdots & \\ p_0 &= \lambda_1\lambda_2\cdots\lambda_d. & (2.4) \end{aligned}$$

Since the flow is isospectral, the coefficients given by (2.2)–(2.4) are independent of time. Therefore they constitute d integrals associated with the flow $L(t)$. Toda (1981) shows

how these constants are related to Hénon's conserved quantities. Here, we observe that the conservation of (2.2)–(2.4) is equivalent to the conservation of

$$\mathcal{I}_r = \sum_{i=1}^{d} \lambda_i^r, \qquad r = 1, \ldots, d. \tag{2.5}$$

Since it is known from classical matrix theory that

$$\sum_{i=1}^{d} \lambda_i^r = \mathrm{tr}(L^r), \qquad r = 1, \ldots, d,$$

(Gantmacher 1977), we prefer the integrals $\mathcal{I}_r$ in (2.5) rather than (2.2)–(2.4) to carry out our theoretical analysis of numerical methods and their retention of isospectrality. In (Perelomov 1990), the integrals of the motion are chosen as

$$I_r = \frac{1}{r}\mathrm{tr}(L^r), \qquad r = 1, \ldots, d, \tag{2.6}$$

which is a scaled form of (2.5), since $I_r = \mathcal{I}_r/r$.

Before proceeding further, it is important to observe that, for every $r = 1, 2, \ldots, d$, the sum $\mathcal{I}_r$ is an r-degree polynomial in the entries of the matrix L. Thus, if we reorder L columnwise and denote by $\boldsymbol{y} \in \mathbb{R}^{d^2}$ the vector which has been obtained in this manner, we reformulate the isospectral flow (1.1) as a vector ODE

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0, \tag{2.7}$$

and the solution of the latter must obey d polynomial conservation laws, each of degree r for $r = 1, \ldots, d$. The vector formulation (2.7) will be used below in the analysis of numerical methods and the retention of quadratic and cubic conservation laws.

We conclude that, if a numerical scheme for the above ODE does recover all r-degree polynomial conservation laws for $r = 1, \ldots, d$, then it is isospectral when applied to the problem (1.1). Vice versa we show that, if a method fails to recover any of these conservation laws, then it is possible to construct an isospectral ODE of the form (1.1) for which the scheme fails to retain isospectrality.

An r-degree conservation law can be written in tensor form, namely

$$S(\underbrace{\boldsymbol{y}, \boldsymbol{y}, \ldots, \boldsymbol{y}}_{r \text{ times}}) \equiv \text{const}, \tag{2.8}$$

where S is a constant[1] symmetric r-tensor, i.e. $S_{\pi(1,\ldots,r)} = S_{1,\ldots,r}$, for every permutation π of the indices $(1, \ldots, r)$. The analysis for a symmetric tensor is not just a simplification but, in fact, covers all possible cases of polynomial conservation laws: assume that the solution of our differential equation obeys

$$T(\underbrace{\boldsymbol{y}, \boldsymbol{y}, \ldots, \boldsymbol{y}}_{r \text{ times}}) \equiv \text{const},$$

[1] Independent of time t and of the particular vector $\boldsymbol{y}(t)$

where the r-tensor T is asymmetric. We introduce the tensor S as

$$S(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_r) = \frac{1}{r!} \sum_{\pi \in \Pi} T(\boldsymbol{y}_{\pi(1)}, \ldots, \boldsymbol{y}_{\pi(r)}),$$

where Π is the set of all the $r!$ permutations π of the indices $(1, 2, \ldots, r)$. Now, S is symmetric by construction, and, moreover,

$$S(\boldsymbol{y}, \ldots, \boldsymbol{y}) = T(\boldsymbol{y}, \ldots, \boldsymbol{y}).$$

Finally, we mention that, if $\boldsymbol{f}$ is defined in $[0, T) \times \tilde{\Omega}$ (eventually $T = +\infty$), where $\tilde{\Omega}$ is an open subset of $\mathbb{R}^q$ for some integer q, then differentiation affirms that (2.8) is equivalent to

$$S(\tilde{\boldsymbol{\omega}}, \ldots, \tilde{\boldsymbol{\omega}}, \boldsymbol{f}(t, \tilde{\boldsymbol{\omega}})) = 0, \qquad \forall\, \tilde{\boldsymbol{\omega}} \in \tilde{\Omega},\ t \in [0, T).$$

along the trajectory of the solutions. However, assume that there exists an analytic extension of $\boldsymbol{f}$, say $\bar{\boldsymbol{f}}$, to a larger open set Ω containing $\tilde{\Omega}$ such that $\bar{\boldsymbol{f}}|_{\tilde{\Omega}} = \boldsymbol{f}$. Then we require that (for simplicity we write $\boldsymbol{f}$ instead of $\bar{\boldsymbol{f}}$)

$$S(\boldsymbol{\omega}, \ldots, \boldsymbol{\omega}, \boldsymbol{f}(t, \boldsymbol{\omega})) = 0, \qquad \forall\, \boldsymbol{\omega} \in \Omega,\ t \in [0, T). \tag{2.9}$$

The relation (2.9) will be a fundamental assumption in this chapter, since, for numerical schemes like Runge–Kutta methods, the internal stages might be in Ω and not necessarily in $\tilde{\Omega}$.

2.2 Numerical methods for ODEs

Given the differential system

$$\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y}), \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0, \qquad t \geq 0, \tag{2.10}$$

with $\boldsymbol{y}_0 \in \mathbb{R}^q$ and $\boldsymbol{f} : [0, T) \times \Omega \to \mathbb{R}^q$, a linear s-step method produces the recurrence

$$\sum_{i=0}^{s} a_i \boldsymbol{y}_{n+i} = h \sum_{i=0}^{s} b_i \boldsymbol{f}(t_{n+i}, \boldsymbol{y}_{n+i}), \qquad n = 1, 2, \ldots \tag{2.11}$$

where, without loss of generality, we set $a_s = 1$. Let us assume that scheme is implemented with constant stepsize h, so that $t_n = nh$. The method can be characterized by the polynomials (ρ, σ), where

$$\rho(z) = \sum_{i=0}^{s} a_i z^i, \qquad \sigma(z) = \sum_{i=0}^{s} b_i z^i, \qquad z \in \mathbb{C}. \tag{2.12}$$

To initiate the numerical time-stepping we need s starting conditions, $\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{s-1}$. We assume that these starting conditions are available consistently with the isospectral problem (1.1).

Similarly, a ν-stage Runge–Kutta (RK) scheme, defined by the *Butcher tableau*

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array}, \tag{2.13}$$

where $A = \{a_{i,j}\} \in \mathbb{R}^{\nu\times\nu}$, $\boldsymbol{b} = \{b_i\}$, $\boldsymbol{c} = \{c_i\} \in \mathbb{R}^{\nu}$, produces the time-stepping formula

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\sum_{i=1}^{\nu} b_i \boldsymbol{k}_i, \tag{2.14}$$

where

$$\boldsymbol{k}_i = \boldsymbol{f}(t_n + c_i h, \boldsymbol{\phi}_i), \qquad i = 1, \ldots, \nu, \tag{2.15}$$

is coupled with

$$\boldsymbol{\phi}_i = \boldsymbol{y}_n + h\sum_{j=1}^{\nu} a_{i,j}\boldsymbol{k}_j, \qquad i = 1, \ldots, \nu. \tag{2.16}$$

The vectors $\boldsymbol{\phi}_i$'s are also called *internal stages* of the RK scheme. The $\boldsymbol{k}_i$'s and $\boldsymbol{\phi}_i$'s depend on n, but, for ease of notation, this is not explicitely stated in (2.15), (2.16). We have assumed that the scheme is implemented with a constant stepsize h and that $t_n = nh$. Generalization to variable stepsize is easier for RK schemes than for LMMs, and follows by introducing a partition $0 = t_0 < t_1 < \ldots$ and letting h be replaced by $h_n = t_{n+1} - t_n$.

Both the LMM (2.11) and the RK scheme (2.13) can be applied to differential equations in a matrix form.

Before proceeding further, we require that these schemes are consistent. Thus, for a LMM we have

$$\rho(1) = 0,$$

and

$$c_i = \sum_{\ell=1}^{\nu} a_{i,\ell}, \quad i = 1, \ldots, \nu, \qquad \text{and} \qquad \sum_{\ell=1}^{\nu} b_\ell = 1$$

for a RK method. For further details, that are outside the scope of this thesis, for example order conditions, stability, and other features of these schemes, we refer the reader to (Hairer, Nørsett & Wanner 1993, Iserles 1996, Lambert 1991).

We have finally all the technical details to commence our analysis. We start from the simplest condition, when $r = 1$, which corresponds to the retention of the trace (linear conservation law) under discretization.

2.3 Linear conservation laws

Recall that the trace is a linear operator on the space of $d \times d$ matrices. Thus, for all real numbers γ_1, γ_2 and for all real matrices Γ_1, Γ_2, it is true that

$$\mathrm{tr}(\gamma_1\Gamma_1 + \gamma_2\Gamma_2) = \gamma_1\mathrm{tr}(\Gamma_1) + \gamma_2\mathrm{tr}(\Gamma_2).$$

Furthermore, for arbitrary $d \times d$ matrices,

$$\mathrm{tr}(\Gamma_1\Gamma_2) = \mathrm{tr}(\Gamma_2\Gamma_1),$$

hence it follows that

$$\mathrm{tr}([\Gamma_1, \Gamma_2]) = 0. \tag{2.17}$$

Theorem 2.1 *Every consistent linear s-step method* (2.11)*, applied to the isospectral problem* (1.1) *with starting values* $L_0, L_1, \ldots, L_{s-1}$ *satisfying* $\mathrm{tr}(L_{s-1}) = \cdots = \mathrm{tr}(L_1) = \mathrm{tr}(L_0)$*, produces an approximation* L_n *which retains the trace of* L_0 *for all* $n = s, s+1, \ldots$.

Proof. Assume that $\mathrm{tr}(L_{n+s-1}) = \mathrm{tr}(L_{n+s-2}) = \cdots = \mathrm{tr}(L_n)$. Because of linearity of the trace and (2.17), then

$$\begin{aligned} \mathrm{tr}(L_{n+s}) &= -\left(\sum_{i=0}^{s-1} a_i\right) \mathrm{tr}(L_n) + h \sum_{i=0}^{s} b_i \mathrm{tr}([B_{n+i}, L_{n+i}]) = -\left(\sum_{i=0}^{s-1} a_i\right) \mathrm{tr}(L_n) \\ &= -(\rho(1) - 1)\mathrm{tr}(L_n), \end{aligned}$$

where $B_{n+i} = B(t_{n+i}, L_{n+i})$. The theorem follows by induction on n since, by virtue of consistency, $\rho(1) = 0$. □

Theorem 2.2 *Every consistent* ν*-stage RK method* (2.13)*, when applied to the isospectral problem* (1.1)*, retains the trace of* L_0.

Proof. Consider a ν-stage RK method defined by the Butcher tableau (2.13). Then, at each step,

$$L_{n+1} = L_n + h \sum_{i=1}^{\nu} b_i K_i.$$

Here, the K_i's are commutators of two matrices (hence, by (2.17), they have zero trace). Moreover, taking into account that the trace is a linear operation on matrices, it follows that

$$\mathrm{tr}(L_{n+1}) = \mathrm{tr}(L_n),$$

and, by induction on n, the trace of L_0 is retained. □

Thus, no particular conditions need be imposed on LMMs and RK schemes when we want just the first (linear) integral to be retained.

2.4 Numerical methods and quadratic conservation laws

For quadratic conservation laws we prefer to derive our conditions, on the coefficients of the numerical methods, using the vector formulation of the problem, already introduced in Section 2.1. In this way we are able to take into account already known results, since quadratic conservation laws have been widely investigated for both LMMs and RK schemes.

We assume throughout this section that the ODE (2.10) obeys the quadratic conservation law

$$\boldsymbol{y}^T S \boldsymbol{y} = \text{const}, \qquad S \text{ symmetric},$$

together with the assumption (2.9), i.e. that

$$\boldsymbol{\omega}^T S \boldsymbol{f}(t, \boldsymbol{\omega}) = 0, \qquad \forall\, \boldsymbol{\omega} \in \Omega,\ \forall\, t \geq 0.$$

2.4.1 Analysis for Linear Multistep Methods

Unlike Runge–Kutta methods, LMMs are characterized by the fact that they are *Taylor-type* methods: in other words, given an s-step method of order p and assuming that $\boldsymbol{y}_n = \boldsymbol{y}(t_n), \ldots, \boldsymbol{y}_{n+1-s} = \boldsymbol{y}(t_{n+1-s})$, the error expansion at time t_{n+1} has the Taylor-like form

$$\boldsymbol{y}_{n+1} - \boldsymbol{y}(t_{n+1}) = c_n h_n^{p+1} \boldsymbol{y}^{(p+1)}(t_n) + \mathcal{O}\left(h^{p+2}\right),$$

where $h_n = t_{n+1} - t_n$ and the constant c_n does not depend on the function $\boldsymbol{f}$. The simplest example of Taylor-type methods is the *truncated Taylor expansion* of order p,

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}(t_n) + h\boldsymbol{y}'(t_n) + \ldots + \frac{h^p}{p!}\boldsymbol{y}^{(p)}(t_n).$$

Being a Taylor-type methods means essentially that the first error term in the numerical expansion of the method depends in a simple way on the true solution $\boldsymbol{y}$, since it is proportional to its $(p+1)$-th derivative. In this respect, RK methods are very different, since the first error term depends in a crucial way on various products of derivatives *(elementary differentials)* of the function $\boldsymbol{f}$ (Hairer et al. 1993).

Taylor-type methods and their retention of arbitrary nonlinear conservation laws have been investigated at a greater length in (Iserles 1997). Here we present the main results. Let us assume that the solution $\boldsymbol{y}$ of (2.10) obeys the nonlinear conservation law

$$u(\boldsymbol{y}(t)) \equiv \text{const},$$

and hence

$$\frac{\mathrm{d}}{\mathrm{d}t} u(\boldsymbol{y}(t)) = 0,$$

which is equivalent to the *skew-gradient system* (Itoh & Abe 1988, Quispel & Chapel 1997)

$$\boldsymbol{y}' = \Psi(t, \boldsymbol{y}) \nabla u(\boldsymbol{y}),$$

where Ψ is a skew-symmetric matrix function. Without loss of generality, we consider the two-dimensional case, when the above skew-gradient system is given by

$$\begin{aligned} y_1' &= \psi(t, \boldsymbol{y}) \frac{\partial u}{\partial y_2}, \\ y_2' &= -\psi(t, \boldsymbol{y}) \frac{\partial u}{\partial y_1}, \end{aligned}$$

where ψ is an arbitrary scalar function. Then, unless u is the solution of the *Bateman equation* (Fairlie & Mulvey 1994),

$$\mathcal{B}(u) = \left(\frac{\partial u}{\partial y_2}\right)^2 \frac{\partial^2 u}{\partial y_1^2} - 2\frac{\partial u}{\partial y_1}\frac{\partial u}{\partial y_2}\frac{\partial^2 u}{\partial y_1 \partial y_2} + \left(\frac{\partial u}{\partial y_1}\right)^2 \frac{\partial^2 u}{\partial y_2^2} = 0,$$

the given Taylor-type method cannot retain the invariant $u(\boldsymbol{y}) = \text{const}$. On the other hand, it is possible to show that if u obeys the Bateman equation, then u must be a linear function of $\boldsymbol{y}$. The conclusion is summarized in the following theorem.

Theorem 2.3 (Iserles (1997)) *A Taylor-type method can retain the invariant* $u(\boldsymbol{y}) = \text{const}$ *if and only if* u *is a linear function of* $\boldsymbol{y}$. □

Hence, for isospectral flows, LMMs are of little merit.

The results seem more promising for a slightly different class of schemes, the *one-leg* methods (OLM). Although these methods are, in a certain sense, not classical, their formalism is very similar both to LMMs and RK methods.

Given the s-step scheme (2.11), the corresponding one-leg method is

$$\rho(E)\boldsymbol{y}_n = h\boldsymbol{f}(\sigma(E)\boldsymbol{y}_n), \quad \rho(z) = \sum_{i=0}^{s} a_i z^i,\ \sigma(z) = \sum_{i=0}^{s} b_i z^i, \tag{2.18}$$

where E stands for the familiar *shift operator* on sequences,

$$E\boldsymbol{y}_n = \boldsymbol{y}_{n+1}.$$

For simplicity, assume that $a_i = b_i = 0$ when $i \geq s+1$.

Theorem 2.4 (Eirola & Sanz-Serna (1992)) *If the differential equation* (2.10) *obeys the quadratic condition* $\boldsymbol{y}^T S\boldsymbol{y} = \text{const}$ *together with* $\boldsymbol{\omega}^T S\boldsymbol{f}(\boldsymbol{\omega}) = 0$, *the OLM* (2.18) *retains the quadratic invariant*

$$[\boldsymbol{y}_n^T, \ldots, \boldsymbol{y}_{n+s-1}^T](\Lambda \otimes S)[\boldsymbol{y}_n, \ldots, \boldsymbol{y}_{n+s-1}] = \text{const},$$

with

$$\lambda_{i,j} = \sum_{m \geq 0} (a_{i+m}b_{j+m} + a_{j+m}b_{i+m}), \qquad i,j = 1, \ldots, s,$$

if and only if the scheme is reversible, i.e. the polynomials ρ *and* σ *obey the condition*

$$\begin{aligned} \rho(z) &= -z^s \rho(z^{-1}), \\ \sigma(z) &= z^s \sigma(z^{-1}), \end{aligned} \tag{2.19}$$

or equivalently,

$$\rho(z)\sigma(z^{-1}) + \rho(z^{-1})\sigma(z) = 0, \tag{2.20}$$

for all $z \in \mathbb{C}$. □

Note that the above theorem does not guarantee that $\boldsymbol{y}^T S\boldsymbol{y}$ is retained by the one-leg method, unless $\Lambda = I$, the identity matrix.

The *implicit midpoint rule*

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\boldsymbol{f}\Big(\frac{\boldsymbol{y}_n + \boldsymbol{y}_{n+1}}{2}\Big),$$

is perhaps the most familiar and important example of one-leg methods, it has order two and is reversible according to the definition of reversibility in (2.19). It also preserves $\boldsymbol{y}^T S\boldsymbol{y}$ since for this method $\Lambda = I$.

2.4.2 Runge–Kutta methods and quadratic invariants

For RK schemes, we introduce the symmetric matrix $M = (m_{i,j})$, whose entries are

$$m_{i,j} = b_i a_{i,j} + b_j a_{j,i} - b_i b_j, \qquad i,j = 1,\ldots,\nu, \tag{2.21}$$

(Butcher 1975). Hence we can quote the following result, essentially due to (Cooper 1987).

Theorem 2.5 *Runge–Kutta schemes for which $M = O$ retain all quadratic conservation laws.*

Proof. The original result follows from algebraic stability type considerations. However, we prefer to follow Sanz-Serna & Calvo (1994). Recalling the time-stepping formula (2.14) for an RK method, it is possible to show, by virtue of (2.9), that

$$\begin{aligned} \boldsymbol{y}_{n+1}^T S \boldsymbol{y}_{n+1} &= \boldsymbol{y}_n^T S \boldsymbol{y}_n + h^2 \sum_{i,j=1}^{\nu} (b_i b_j - b_i a_{i,j} - b_j a_{j,i}) \boldsymbol{k}_i^T S \boldsymbol{k}_j \\ &= \boldsymbol{y}_n^T S \boldsymbol{y}_n - h^2 \sum_{i,j=1}^{\nu} m_{i,j} \boldsymbol{k}_i^T S \boldsymbol{k}_j. \end{aligned} \tag{2.22}$$

If $M = O$ then the method recovers quadratic integrals since

$$\boldsymbol{y}_{n+1}^T S \boldsymbol{y}_{n+1} = \boldsymbol{y}_n^T S \boldsymbol{y}_n,$$

and the proof follows by induction on n. □

It is also possible to prove the converse for a large subset of RK schemes, namely *nonconfluent* schemes. An RK scheme is said to nonconfluent if, for all $i,j = 1,\ldots,\nu$, it is true that $c_i \neq c_j$, for $i \neq j$.

Theorem 2.6 (Calvo et al. (1995)) *For nonconfluent RK schemes, the retention of quadratic invariants implies the condition $M = O$.*

Proof. Because of the symmetry of the matrix M, it is sufficient to prove that $m_{i,j} = 0$ for $1 \leq i \leq j \leq \nu$. Thus, consider the ODE

$$\begin{bmatrix} y^1 \\ y^2 \end{bmatrix}' = \begin{bmatrix} f^1(t,\boldsymbol{y}) \\ f^2(t,\boldsymbol{y}) \end{bmatrix}, \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0,$$

where $\boldsymbol{y}_0$ is given and is such that $\|\boldsymbol{f}(0,\boldsymbol{y}_0)\|_2 \neq 0$. Assume that the solution obeys the quadratic law

$$\boldsymbol{y}^T \boldsymbol{y} = \text{const}, \qquad (S = I),$$

or, equivalently, $y^2 f^2(t,\boldsymbol{y}) = -y^1 f^1(t,\boldsymbol{y})$. This system can be viewed as a 2×2 isospectral flow (1.1), where

$$L = \begin{bmatrix} y^1 & y^2 \\ y^2 & -y^1 \end{bmatrix}, \qquad B = \frac{1}{2}\begin{bmatrix} 0 & -g(t,\boldsymbol{y}) \\ g(t,\boldsymbol{y}) & 0 \end{bmatrix},$$

where $g(t, \boldsymbol{y})$ is, for the time being, a smooth arbitrary function related to $\boldsymbol{f}$ in terms of the formula

$$\boldsymbol{f}(t, \boldsymbol{y}) = g(t, \boldsymbol{y}) \begin{bmatrix} y^2 \\ -y^1 \end{bmatrix}.$$

The underlying quadratic conservation law is $\mathrm{tr}(L^2) = 2[(y^1)^2 + (y^2)^2] = \text{const}$. We write the system in an autonomous form, introducing a third component for the vector $\boldsymbol{y}$, namely $y^3 = t$. Thus we rename $\boldsymbol{y} = [y^1, y^2, y^3]^T$ and it follows that $f^3(\boldsymbol{y}) = (y^3)' \equiv 1$. The quadratic law now in effect is $\boldsymbol{y}^T S \boldsymbol{y} = \text{const}$, with

$$S = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}, \tag{2.23}$$

where $\mathbf{0} = [0, 0]^T$. The differential system can be written in the form

$$\boldsymbol{y}' = \begin{bmatrix} g(\boldsymbol{y}) y^2 \\ -g(\boldsymbol{y}) y^1 \\ 1 \end{bmatrix},$$

where $g : \mathbb{R}^3 \to \mathbb{R}$. We consider just the first step of the method. Let ℓ be a fixed index in $\{1, \ldots, \nu\}$ and

$$\begin{aligned} \phi_i^1 &= \frac{y_0^1 + h y_0^2 a_{i,\ell} + h^2 y_0^1 a_{\ell,\ell}(a_{\ell,\ell} - a_{i,\ell})}{1 + h^2 a_{\ell,\ell}^2}, \\ \phi_i^2 &= \frac{y_0^2 - h y_0^1 a_{i,\ell} + h^2 y_0^2 a_{\ell,\ell}(a_{\ell,\ell} - a_{i,\ell})}{1 + h^2 a_{\ell,\ell}^2}, \qquad i = 1, \ldots, \nu. \\ \phi_i^3 &= c_i h, \end{aligned} \tag{2.24}$$

Since the method is nonconfluent, it follows that $\boldsymbol{\phi}_i \neq \boldsymbol{\phi}_j$ for $i \neq j$, since the c_i's are all distinct. Thus it is possible to find a Lipschitz function g (e.g. by interpolation), such that

$$g(\boldsymbol{\phi}_\ell) = 1, \qquad g(\boldsymbol{\phi}_i) = 0, \quad i = 1, \ldots, \nu, \quad i \neq \ell.$$

Correspondingly to this choice for g, it follows that the RK scheme produces exactly the internal stages $\boldsymbol{\phi}_i$, $i = 1, \ldots, \nu$, as in (2.24). Moreover,

$$\boldsymbol{k}_i = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \qquad i \neq \ell,$$

and, by virtue of (2.23), we obtain

$$\boldsymbol{k}_i^T S \boldsymbol{k}_j = 0, \qquad (i, j) \neq (\ell, \ell),$$

hence

$$\boldsymbol{y}_{n+1}^T S \boldsymbol{y}_{n+1} = \boldsymbol{y}_n^T S \boldsymbol{y}_n - h^2 \sum_{i,j=1}^{\nu} m_{i,j} \boldsymbol{k}_i^T S \boldsymbol{k}_j = \boldsymbol{y}_n^T S \boldsymbol{y}_n - h^2 m_{\ell,\ell} \boldsymbol{k}_\ell^T S \boldsymbol{k}_\ell,$$

Due to the choice of $\boldsymbol{y}_0$, the scalar $\boldsymbol{k}_\ell^T S \boldsymbol{k}_\ell = [f^1(\boldsymbol{\phi}_\ell)]^2 + [f^2(\boldsymbol{\phi}_\ell)]^2$ is nonzero for a suitably small stepsize h. Consequently, the h^2 term disappears if and only if $m_{\ell,\ell} = 0$. From

the arbitrariness of $\ell \in \{1, \ldots, \nu\}$ we deduce that all the diagonal elements of M must necessarily vanish. An analogous procedure can be applied for the super-diagonal of M. We fix $\ell \in \{1, \ldots, \nu - 1\}$ and let the vectors $\boldsymbol{\phi}_\ell, \boldsymbol{\phi}_{\ell+1}$ be the solution (which always exists for sufficiently small h) of the implicit system

$$\boldsymbol{\phi}_i = \boldsymbol{y}_0 + h \sum_{j=1}^{\nu} a_{i,j} \boldsymbol{\psi}_j, \qquad i = 1, \ldots, \nu,$$

where

$$\boldsymbol{\psi}_j = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \qquad \text{for } j = 1, \ldots, \nu, \quad j \neq \ell, \ell + 1$$

while

$$\boldsymbol{\psi}_j = \begin{bmatrix} \phi_j^2 \\ -\phi_j^1 \\ 1 \end{bmatrix}, \qquad \text{for } j = \ell, \ell + 1.$$

Once again, by virtue of nonconfluence, the $\boldsymbol{\phi}_i$'s are all distinct. Hence we can find a function g such that

$$g(\boldsymbol{\phi}_\ell) = g(\boldsymbol{\phi}_{\ell+1}) = 1, \qquad g(\boldsymbol{\phi}_i) = 0, \quad i = 1, \ldots, \nu, \quad i \neq \ell, \ell + 1.$$

With this choice for g, we deduce that the first two components of $\boldsymbol{k}_i$ are zero, for $i = \ell, \ell+1$, and from (2.23) it follows that

$$\boldsymbol{k}_i S \boldsymbol{k}_j = 0 \qquad \text{for } (i, j) \neq (\ell, \ell), (\ell, \ell + 1), (\ell + 1, \ell), (\ell + 1, \ell + 1).$$

Thus the h^2 contribution is given by

$$\begin{aligned} \sum_{i,j=1}^{\nu} m_{i,j} \boldsymbol{k}_i^T S \boldsymbol{k}_j &= m_{\ell,\ell} \boldsymbol{k}_\ell^T S \boldsymbol{k}_\ell + m_{\ell+1,\ell+1} \boldsymbol{k}_{\ell+1}^T S \boldsymbol{k}_{\ell+1} \\ &\quad + m_{\ell,\ell+1} \boldsymbol{k}_\ell^T S \boldsymbol{k}_{\ell+1} + m_{\ell+1,\ell} \boldsymbol{k}_{\ell+1}^T S \boldsymbol{k}_\ell \\ &= 2 m_{\ell,\ell+1} \boldsymbol{k}_\ell^T S \boldsymbol{k}_{\ell+1}, \end{aligned}$$

since we have already proved that the diagonal elements of M are zero. Moreover, $\boldsymbol{k}_\ell^T S \boldsymbol{k}_{\ell+1}$ is nonzero for sufficiently small h, hence we deduce that necessarily

$$m_{\ell,\ell+1} = 0,$$

and, ℓ being arbitrary, the latter must be true for all $\ell = 1, \ldots, \nu - 1$. We continue this procedure by induction to the other super-diagonals of M and the assertion follows. □

In other words, unless $M = O$, it is possible to find an isospectral flow for which the scheme is not isospectral. Therefore the condition $M = O$ is necessary for isospectrality and not just sufficient as already seen in Section 2.4.2. In particular, we deduce the following result from Theorem 2.5.

Corollary 2.6.1 *All RK schemes for which $M = O$ are isospectral for 2×2 systems.* □

Runge–Kutta methods that obey the condition $M = O$ are known to be *symplectic* (Sanz-Serna & Calvo 1994), hence the latter corollary justifies the observation, based on numerical experiments, that symplectic RK schemes retain isospectrality of 2×2 systems.

2.5 Cubic integrals under discretization

The next condition for isospectrality, when $d \geq 3$, is the retention of cubic integrals. We assume that the system (2.10) obeys

$$S(\boldsymbol{y}, \boldsymbol{y}, \boldsymbol{y}) = \text{const},$$

or equivalently

$$S(\boldsymbol{\omega}, \boldsymbol{\omega}, \boldsymbol{f}(t, \boldsymbol{\omega})) = 0 \qquad \forall\, \boldsymbol{\omega} \in \Omega,\ t \in [0, T). \tag{2.25}$$

for some symmetric 3-tensor S.

2.5.1 Runge–Kutta schemes and cubic invariants

We wish to solve (2.10) with an RK method, since in the last paragraph we have identified RK methods as plausible candidates for isospectrality.

We define the *departure* from the cubic manifold as

$$d(\mathcal{I}_3) = S(\boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1}) - S(\boldsymbol{y}_n, \boldsymbol{y}_n, \boldsymbol{y}_n), \tag{2.26}$$

where we have assumed that $S(\boldsymbol{y}_n, \boldsymbol{y}_n, \boldsymbol{y}_n) = S(\boldsymbol{y}_0, \boldsymbol{y}_0, \boldsymbol{y}_0)$.

By virtue of the time-stepping formula (2.14) and the symmetry of the tensor S, we can expand $S(\boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1})$ in powers of h, namely

$$\begin{aligned} S(\boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1}) &= S(\boldsymbol{y}_n, \boldsymbol{y}_n, \boldsymbol{y}_n) + 3h \sum_{i=1}^{\nu} b_i S(\boldsymbol{y}_n, \boldsymbol{y}_n, \boldsymbol{k}_i) \\ &\quad + 3h^2 \sum_{i,j=1}^{\nu} b_i b_j S(\boldsymbol{y}_n, \boldsymbol{k}_i, \boldsymbol{k}_j) + h^3 \sum_{i,j,\ell=1}^{\nu} b_i b_j b_\ell S(\boldsymbol{k}_i, \boldsymbol{k}_j, \boldsymbol{k}_\ell). \end{aligned}$$

We have

$$d(\mathcal{I}_3) = 3h \sum_{i=1}^{\nu} b_i S(\boldsymbol{y}_n, \boldsymbol{y}_n, \boldsymbol{k}_i) + 3h^2 \sum_{i,j=1}^{\nu} b_i b_j S(\boldsymbol{y}_n, \boldsymbol{k}_i, \boldsymbol{k}_j) + h^3 \sum_{i,j,\ell=1}^{\nu} b_i b_j b_\ell S(\boldsymbol{k}_i, \boldsymbol{k}_j, \boldsymbol{k}_\ell).$$

It is convenient to express the vector $\boldsymbol{y}_n$ in terms of the $\boldsymbol{\phi}_i$'s, since, from (2.16), we have

$$\boldsymbol{y}_n = \boldsymbol{\phi}_i - h \sum_{j=1}^{\nu} a_{i,j} \boldsymbol{k}_j.$$

Substitution yields

$$\begin{aligned} S(\boldsymbol{y}_n, \boldsymbol{y}_n, \boldsymbol{k}_i) &= S(\boldsymbol{\phi}_i, \boldsymbol{\phi}_i, \boldsymbol{k}_i) - 2h \sum_{j=1}^{\nu} a_{i,j} S(\boldsymbol{\phi}_i, \boldsymbol{k}_i, \boldsymbol{k}_j) \\ &\quad + h^2 \sum_{j,\ell=1}^{\nu} a_{i,j} a_{i,\ell} S(\boldsymbol{k}_i, \boldsymbol{k}_j, \boldsymbol{k}_\ell) \end{aligned} \tag{2.27}$$

$$= \quad -2h\sum_{j=1}^{\nu} a_{i,j}S(\boldsymbol{\phi}_i,\boldsymbol{k}_i,\boldsymbol{k}_j) + h^2\sum_{j,\ell=1}^{\nu} a_{i,j}a_{i,\ell}S(\boldsymbol{k}_i,\boldsymbol{k}_j,\boldsymbol{k}_\ell),$$

$$S(\boldsymbol{y}_n,\boldsymbol{k}_i,\boldsymbol{k}_j) = S(\boldsymbol{\phi}_i,\boldsymbol{k}_i,\boldsymbol{k}_j) - h\sum_{\ell=1}^{\nu} a_{i,\ell}S(\boldsymbol{k}_i,\boldsymbol{k}_j,\boldsymbol{k}_\ell). \tag{2.28}$$

The first term in (2.27) vanishes because of (2.25), $\boldsymbol{k}_i$ being $\boldsymbol{f}(t_n+c_ih,\boldsymbol{\phi}_i)$, for $i=1,\ldots,\nu$. Hence the departure $d(\mathcal{I}_3)$ reduces to

$$d(\mathcal{I}_3) = 3h^2\left[-2\sum_{i,j=1}^{\nu} b_ia_{i,j}S(\boldsymbol{\phi}_i,\boldsymbol{k}_i,\boldsymbol{k}_j) + \sum_{i,j=1}^{\nu} b_ib_jS(\boldsymbol{\phi}_i,\boldsymbol{k}_i,\boldsymbol{k}_j)\right] \tag{2.29}$$
$$+ h^3\sum_{i,j,\ell=1}^{\nu}(3b_ia_{i,j}a_{i,m} - 3b_ib_ja_{i,\ell} + b_ib_jb_\ell)S(\boldsymbol{k}_i,\boldsymbol{k}_j,\boldsymbol{k}_\ell),$$

whereby, recalling that, for (2.16), $\boldsymbol{\phi}_i = \boldsymbol{\phi}_j + h\sum_{\ell=1}^{\nu}(a_{i,\ell}-a_{j,\ell})\boldsymbol{k}_\ell$, we find

$$S(\boldsymbol{\phi}_i,\boldsymbol{k}_i,\boldsymbol{k}_j) = S(\boldsymbol{\phi}_j,\boldsymbol{k}_i,\boldsymbol{k}_j) + h\sum_{\ell=1}^{\nu}(a_{i,\ell}-a_{j,\ell})S(\boldsymbol{k}_i,\boldsymbol{k}_j,\boldsymbol{k}_\ell),$$

and this allows us to rearrange the indices so that

$$d(\mathcal{I}_3) = h^2\sum_{i,j=1}^{\nu}(b_ib_j - b_ia_{i,j} - b_ja_{j,i})S(\boldsymbol{\phi}_i,\boldsymbol{k}_i,\boldsymbol{k}_j) \tag{2.30}$$
$$+ h^3\sum_{i,j,\ell=1}^{\nu}(3b_ia_{i,j}a_{j,\ell} - 3b_ib_ja_{i,\ell} + b_ib_jb_\ell)S(\boldsymbol{k}_i,\boldsymbol{k}_j,\boldsymbol{k}_\ell).$$

In particular, the coefficients of the h^2 terms are the entries of the matrix M, while the coefficients of the h^3 terms, after rearranging indices, are given by

$$\Upsilon_{i,j,\ell} = (b_ia_{i,j}a_{j,\ell} + b_ja_{j,\ell}a_{\ell,i} + b_\ell a_{\ell,i}a_{i,j}) \tag{2.31}$$
$$- (b_ib_\ell a_{i,j} + b_\ell b_j a_{\ell,i} + b_jb_ia_{j,\ell}) + b_ib_jb_\ell, \quad i,j,\ell = 1,2,\ldots,\nu.$$

Thus, if $M=O$ and $\Upsilon=O$, it follows that the cubic conservation law is retained and the departure $d(\mathcal{I}_3)$ from the cubic manifold is zero.

The 3-tensor Υ looks like a generalization of the matrix M and, for a while, we may have the illusion that the conditions on M and Υ might be both satisfied at the same time. This is not the case, since the following negative result holds.

Theorem 2.7 (Calvo et al. (1995)) *No nonconfluent RK schemes can recover arbitrary cubic conservation laws.*

Proof. We first observe that the conditions $M=O$ and $\Upsilon=O$ are contradictory for schemes of nontrivial order. In fact, since $\mathbf{1}^T\boldsymbol{b}=1$, there exist an index $s\in\{1,\ldots,\nu\}$ such that $b_s\neq 0$. We find

$$m_{s,s}=0 \quad \Longrightarrow \quad a_{s,s}=\frac{1}{2}b_s.$$

Substitution in $\Upsilon_{s,s,s} = 0$ yields

$$b_s^3 = 0,$$

hence a contradiction.

The most technically difficult part of the proof consists in showing that the conditions $m_{s,s} = 0$ and $\Upsilon_{s,s,s} = 0$ are also necessary for the recovery of cubic conservation laws and not just sufficient, as it already follows from our previous discussion.

For a proof of necessity, we follow essentially the analysis of isospectral systems presented in (Calvo et al. 1995).

We let $\boldsymbol{y}$ be a vector in $\mathbb{R}^5$ satisfying the following differential equation

$$\begin{bmatrix} y^1 \\ y^2 \\ y^3 \\ y^4 \\ y^5 \end{bmatrix}' = g(t, \boldsymbol{y}) \begin{bmatrix} 2(y^4)^2 \\ 2[(y^5)^2 - (y^4)^2] \\ -2(y^5)^2 \\ y^4(y^2 - y^1) \\ y^5(y^3 - y^2) \end{bmatrix}, \qquad \boldsymbol{y}(0) = \boldsymbol{y}_0.$$

The function $g(t, \boldsymbol{y})$, for the time being, is a sufficiently smooth function from $[0, T) \times \mathbb{R}^5$ to $\mathbb{R}$, otherwise arbitrary. This system might be viewed as a generalization of a Toda flow written in the Lax form (1.1), the first three entries of $\boldsymbol{y}$ being the diagonal elements of the matrix L, while the remaining two are its subdiagonal elements. By construction this flow is symmetric and retains the bandwidth, so it is sufficient to consider $\boldsymbol{y}$ evolving in $\mathbb{R}^5$ instead of $\mathbb{R}^9$. Furthermore, it retains the cubic conservation law

$$S(\boldsymbol{y}, \boldsymbol{y}, \boldsymbol{y}) = \operatorname{tr}(L^3) = \text{const},$$

where

$$\operatorname{tr}(L^3) = \Big[(y^1)^3 + (y^2)^3 + (y^3)^3\Big] + 3\Big[(y^4)^2(y^1 + y^2) + (y^5)^2(y^2 + y^3)\Big].$$

The above relation allows us to find the $5^3 = 125$ entries of the symmetric 3-tensor S, since comparison with

$$\sum_{i,j,\ell=1}^{5} S_{i,j,\ell} y^i y^j y^\ell = \text{const},$$

yields

$$S_{i,i,i} = 1, \qquad i = 1, 2, 3,$$

and

$$S_{\pi(1,4,4)} = S_{\pi(2,4,4)} = S_{\pi(2,5,5)} = S_{\pi(3,5,5)} = 1$$

for every permutation π of the corresponding indices. All the remaining entries of S are zero. Before proceeding further, it is convenient to write the system in an autonomous form, introducing the sixth variable $y^6 = t$ where $(y^6)' \equiv 1$ and $y_0^6 = 0$. We again denote by $\boldsymbol{y}$ the vector obtained in this manner and by S the corresponding tensor, that now has $6^3 = 216$ components. However, the nonzero components of S are the same as those derived before.

It is sufficient to consider a single step of the method. Then,

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{y}_0 + h\sum_{i=1}^{\nu} b_i \boldsymbol{k}_i, \\ \boldsymbol{\phi}_i &= \boldsymbol{y}_0 + h\sum_{j=1}^{\nu} a_{i,j}\boldsymbol{k}_j, \qquad i = 1,\ldots,\nu, \end{aligned} \tag{2.32}$$

where

$$\boldsymbol{k}_j = \begin{bmatrix} 2g(\boldsymbol{\phi}_j)(\phi_j^4)^2 \\ 2g(\boldsymbol{\phi}_j)[(\phi_j^5)^2 - (\phi_j^4)^2] \\ -2g(\boldsymbol{\phi}_j)(\phi_j^5)^2 \\ g(\boldsymbol{\phi}_j)\phi_j^4(\phi_j^2 - \phi_j^1) \\ g(\boldsymbol{\phi}_j)\phi_j^5(\phi_j^3 - \phi_j^2) \\ 1 \end{bmatrix}, \qquad j = 1,\ldots,\nu.$$

Let s be the index for which the conditions $m_{s,s} = 0$ and $\Upsilon_{s,s,s} = 0$ contradict each other, and let the $\boldsymbol{\phi}_i$'s be the solution of the system

$$\boldsymbol{\phi}_i = \boldsymbol{y}_0 + h\sum_{j=1}^{\nu} a_{i,j}\boldsymbol{\psi}_j, \qquad i = 1,\ldots,\nu,$$

where $\mathbf{0}$ is the null vector in $\mathbb{R}^5$,

$$\boldsymbol{\psi}_j = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \qquad \text{for } j \neq s,$$

and

$$\boldsymbol{\psi}_s = \begin{bmatrix} 2(\phi_s^4)^2 \\ 2[(\phi_s^5)^2 - (\phi_s^4)^2] \\ -2(\phi_s^5)^2 \\ \phi_s^4(\phi_s^2 - \phi_s^1) \\ \phi_s^5(\phi_s^3 - \phi_s^2) \\ 1 \end{bmatrix}.$$

In particular, since the method is nonconfluent, $\phi_i^6 = c_i$ and the $\boldsymbol{\phi}_i$'s are all distinct. Thus, it is possible to find a function g such that

$$\begin{aligned} g(\boldsymbol{\phi}_s) &= 1, \\ g(\boldsymbol{\phi}_i) &= 0, \qquad i = 1,\ldots,\nu, \quad i \neq s. \end{aligned}$$

This choice for g yields stages $\boldsymbol{k}_i = \boldsymbol{\psi}_i$, i.e.

$$\boldsymbol{k}_i = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \qquad i = 1,\ldots,\nu, \quad i \neq s,$$

hence,

$$S(\boldsymbol{\phi}_i, \boldsymbol{k}_j, \boldsymbol{k}_\ell) = S(\boldsymbol{k}_i, \boldsymbol{k}_j, \boldsymbol{k}_\ell) = 0, \quad \text{for } (i,j,\ell) \neq (s,s,s).$$

Furthermore,

$$\begin{aligned} S(\boldsymbol{\phi}_s, \boldsymbol{k}_s, \boldsymbol{k}_s) &= 4\Big[(\phi_s^1+\phi_s^2)(\phi_s^4)^4 - (\phi_s^1+\phi_s^3)(\phi_s^5\phi_s^4)^2 + (\phi_s^2+\phi_s^3)(\phi_s^5)^4\Big] \\ &\quad + \Big[(\phi_s^1)^3 + (\phi_s^2)^3 - \phi_s^1\phi_s^2(\phi_s^2+\phi_s^1)\Big](\phi_s^4)^2 \\ &\quad + \Big[(\phi_s^2)^3 + (\phi_s^3)^2 - \phi_s^2\phi_s^3(\phi_s^3+\phi_s^2)\Big](\phi_s^5)^2, \\ S(\boldsymbol{k}_s, \boldsymbol{k}_s, \boldsymbol{k}_s) &= -6(\phi_s^4\phi_s^5)^2\Big\{4(\phi_s^5)^2 - (\phi_s^4)^2] - (\phi_s^1)^2 + (\phi_s^3)^2 + 2\phi_s^1\phi_s^2 - 2\phi_s^2\phi_s^3\Big\}, \end{aligned}$$

whereby $\boldsymbol{\phi}_s$ is such that, for an appropriate choice of the initial condition $\boldsymbol{y}_0$ and stepsize h, the latter expressions are nonzero.[2] The departure from the cubic manifold reduces to

$$d(\mathcal{I}_3) = -h^2 m_{s,s} S(\boldsymbol{\phi}_s, \boldsymbol{k}_s, \boldsymbol{k}_s) + h^3 \Upsilon_{s,s,s} S(\boldsymbol{k}_s, \boldsymbol{k}_s, \boldsymbol{k}_s)$$

and it vanishes if and only if $m_{s,s} = 0$ and $\Upsilon_{s,s,s} = 0$, which is a contradiction because of the choice of the index s. It follows that for this system the departure from the cubic manifold is nonzero. □

Corollary 2.7.1 *Nonconfluent RK methods cannot be isospectral when $d \geq 3$, i.e. for every nonconfluent RK scheme, it is possible to find an isospectral flow* (1.1) *with $d \geq 3$ and an initial condition L_0 for which the scheme fails to be isospectral.* □

The results in Theorems 2.1–2.7 are illustrated in the following numerical example.

Example 2.1 We consider a periodic Toda lattice of three particles with matrices L, B as in (1.8). Three is the minimal number of particles that we need to consider to observe a difference in the behaviour of the numerical solution between isospectral and methods like RK with $M = O$ that obey linear and quadratic conservation laws. The initial conditions are

$$\boldsymbol{p}_0 = [0, \sqrt{2}, 0]^T, \quad \boldsymbol{q}_0 = [0,0,0]^T,$$

so that the initial matrix L_0 is

$$L_0 = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & \sqrt{2}/2 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix}.$$

We perform the numerical integration with the Implicit Midpoint Rule (IMR) that is a RK method of the Gauss–Legendre family, which obey the condition $M = O$. Its RK tableau is

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array},$$

[2] We can choose $\boldsymbol{y}_0$ such that:

$$\begin{aligned} S(\boldsymbol{y}_0, \boldsymbol{f}(0,\boldsymbol{y}_0), \boldsymbol{f}(0,\boldsymbol{y}_0)) &\neq 0, \\ S(\boldsymbol{f}(0,\boldsymbol{y}_0), \boldsymbol{f}(0,\boldsymbol{y}_0), \boldsymbol{f}(0,\boldsymbol{y}_0)) &\neq 0. \end{aligned}$$

and the numerical integration is performed with stepsize $h = \frac{1}{5}$ in the interval $[0, 100]$. The errors on the linear, quadratic and cubic conservation laws, evaluated as at each step $t_n = nh$,

$$e_n^k = \mathrm{tr}(L_n^k) - \mathrm{tr}(L_0^k), \qquad k = 1, 2, 3,$$

are displayed in Figures 2.1, 2.2 and 2.3 respectively.

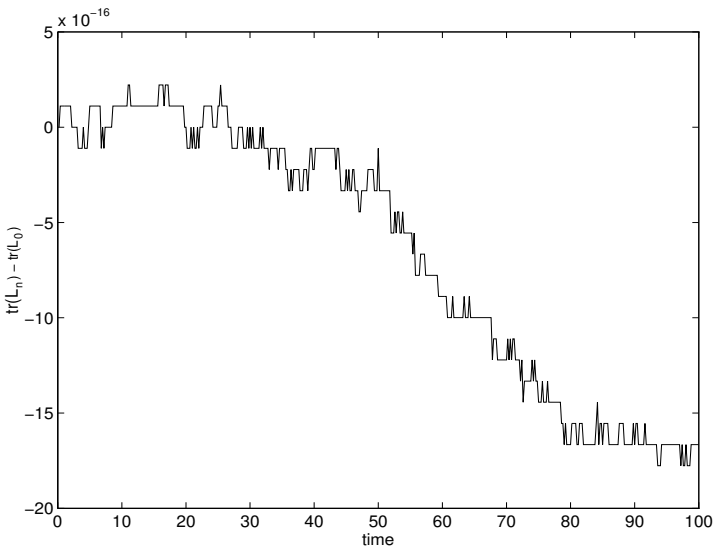


Figure 2.1: Error in the linear conservation law for the IMR applied to a three-particle periodic Toda lattice (Example 2.1).

Note that, for linear and quadratic integrals the error is of machine accuracy (we have used tolerance $\varepsilon = 10^{-15}$ in the fixed point iterations for the implicit solution of the inner stage of the method), while the error in the cubic integral is nonzero (Figure 2.3).

If the method is of order p, common sense indicates that the error in the cubic term will be of order p too. In the case of the IMR, the local error on the cubic integral will be of order $\mathcal{O}(h^3)$ but the global error on the cubic integral will be of order $\mathcal{O}(h^2)$. This is illustrated in Figure 2.4 where we plot in logarithmic scale the error at $T = 10$ against the stepsize $h = \frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{64}$ (solid line). For comparison, we have plotted also (h, h^2), which gives the slope for the $\mathcal{O}(h^2)$ line (dotted line), and (h, h^3), which gives the slope for the $\mathcal{O}(h^3)$ line (dash-dotted line).

For big stepsizes, the global error is $\mathcal{O}(h^3)$ since the interval of integration is not large enough compared to h. For smaller values of h everything is as the theory predicts and the global error becomes parallel to the $\mathcal{O}(h^2)$ line. ◇

2.5.2 One-leg methods and cubic integrals

We consider the one-leg method (2.18)

$$\rho(E)\boldsymbol{y}_n = h\boldsymbol{f}(\sigma(E)\boldsymbol{y}_n)),$$

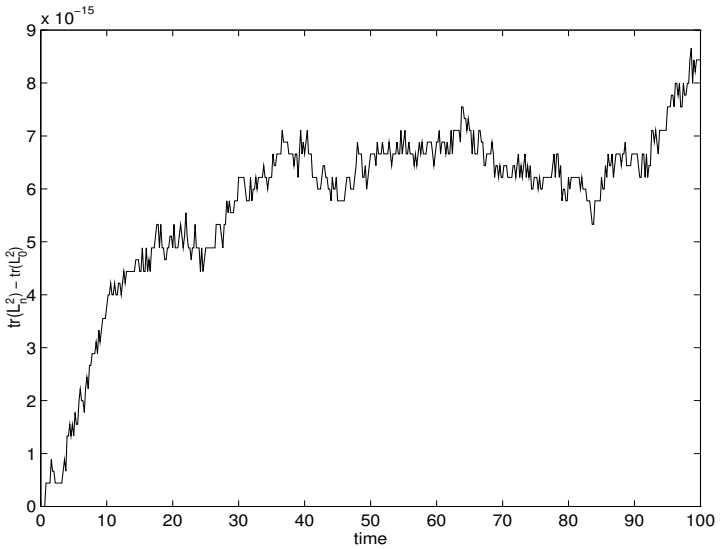


Figure 2.2: Error in the quadratic conservation law for the IMR applied to a three-particle periodic Toda lattice (Example 2.1).

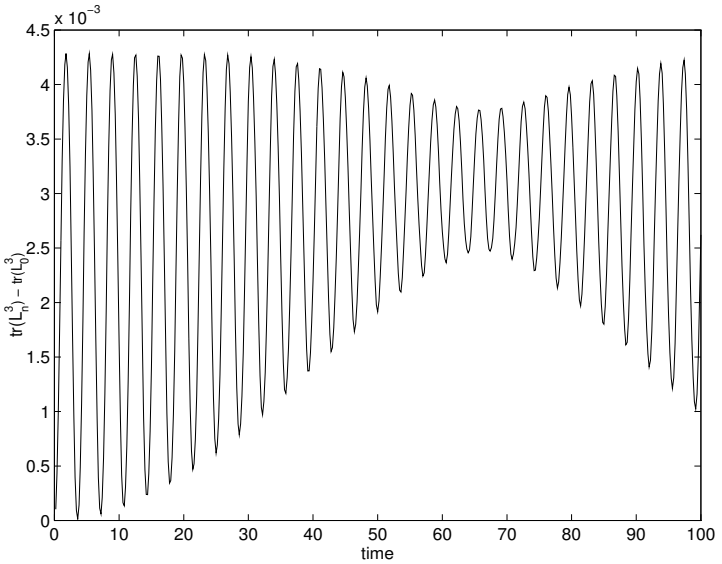


Figure 2.3: Error in the cubic integral for the IMR on the three-particle periodic Toda lattice in Example 2.1.

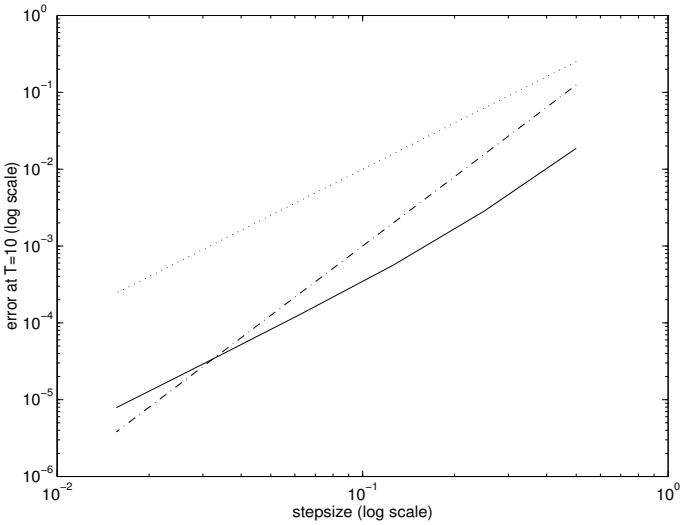


Figure 2.4: Error versus stepsize (solid line) for the cubic integral at $T = 10$ for IMR as in Example 2.1. The dotted and dash-dotted lines indicate the slope for and order-two and order-three error.

and assume that it retains quadratic invariants, i.e. that the reversibility conditions (2.19) or equivalently (2.20) are obeyed by the scheme under consideration.

Following the technique of (Eirola & Sanz-Serna 1992), we introduce the vector

$$\boldsymbol{Y}_n = [\boldsymbol{y}_n, \ldots, \boldsymbol{y}_{n+s-1}]^T$$

and require that

$$\Sigma(\boldsymbol{Y}_{n+1}, \boldsymbol{Y}_{n+1}, \boldsymbol{Y}_{n+1}) = \Sigma(\boldsymbol{Y}_n, \boldsymbol{Y}_n, \boldsymbol{Y}_n), \tag{2.33}$$

for some suitable symmetric three-tensor Σ of the form

$$\Sigma = \Gamma \otimes S,$$

Γ being a symmetric three-tensor that does not depend on $\boldsymbol{f}$ or on S. Note that (2.33) is the same as

$$\sum_{i,j,k=1}^{s} \gamma_{i,j,k} S(\boldsymbol{y}_{n+i}, \boldsymbol{y}_{n+j}, \boldsymbol{y}_{n+k}) - \sum_{i,j,k=1}^{s} \gamma_{i,j,k} S(\boldsymbol{y}_{n-1+i}, \boldsymbol{y}_{n-1+j}, \boldsymbol{y}_{n-1+k}) = 0. \tag{2.34}$$

On the other hand, because of the condition $S(\boldsymbol{\omega}, \boldsymbol{\omega}, f(\boldsymbol{\omega})) = 0$ for all $\boldsymbol{\omega} \in \Omega$, we have

$$S(\sigma(E)\boldsymbol{y}_n, \sigma(E)\boldsymbol{y}_n, \rho(E)\boldsymbol{y}_n) = 0$$

(E denotes the shift operator), or, equivalently,

$$\sum_{i,j,k=1}^{s} (a_i b_j b_k + a_j b_k b_i + a_k b_i b_j) S(\boldsymbol{y}_{n+i}, \boldsymbol{y}_{n+j}, \boldsymbol{y}_{n+k}) = 0.$$

Therefore, in order for (2.33) to be true, we must have

$$\sum_{i,j,k=1}^{s} \gamma_{i,j,k} x_i y_j z_k - \sum_{i,j,k=1}^{s} \gamma_{i,j,k} x_{i-1} y_{j-1} z_{k-1} = \sum_{i,j,k=1}^{s} (a_i b_j b_k + a_j b_k b_i + a_k b_i b_j) x_i y_j z_k$$

for all the values of x_i, y_j, z_k. In terms of generating functions, this is equivalent to requiring that

$$\begin{aligned}(XYZ-1)\sum_{i,j,k=1}^{s} \gamma_{i,j,k} X^{i-1} Y^{j-1} Z^{k-1} = \rho(X)\sigma(Y)\sigma(Z) + \rho(Y)\sigma(Z)\rho(X) \\ + \rho(Z)\sigma(X)\sigma(Y),\end{aligned} \tag{2.35}$$

for all values of X, Y and Z. Note that, when $XYZ = 1$ then the left-hand side of (2.35) vanishes; at the same time, also the right-hand side of (2.35) should equal zero. We take first

$$X = \frac{1}{YZ},$$

so that for the right-hand side of (2.35) we have

$$\text{r.h.s.} = \rho(\tfrac{1}{YZ})\sigma(Y)\sigma(Z) + \rho(Y)\sigma(Z)\sigma(\tfrac{1}{YZ}) + \rho(Z)\sigma(\tfrac{1}{YZ})\sigma(Y). \tag{2.36}$$

Next, we make use of the reversibility conditions (2.19) to find that

$$\text{r.h.s.} = \frac{1}{Y^s Z^s}\Big((\rho(Y)\sigma(Z) + \rho(Z)\sigma(Y))\sigma(YZ) - \rho(YZ)\sigma(Y)\sigma(Z)\Big), \tag{2.37}$$

so that, in order for (2.37) to vanish, we have the condition

$$\frac{\sigma(YZ)}{\rho(YZ)} = \frac{\sigma(Y)\sigma(Z)}{\rho(Y)\sigma(Z) + \rho(Z)\sigma(Y)}. \tag{2.38}$$

The next step consists in substituting $Z \equiv Y$ into (2.38), deriving the relation

$$\frac{\sigma(Y^2)}{\sigma(Y)} = \frac{1}{2}\frac{\rho(Y^2)}{\rho(Y)}, \tag{2.39}$$

that should be satisfied for all values of Y. It is clear that, taking Y sufficiently large, (2.39) implies

$$1 = \frac{1}{2},$$

obviously an inconsistency. This completes the proof of the following result.

Theorem 2.8 *The conditions for the retention of quadratic and cubic integrals are not compatible for OLMs.* □

We should point that, although we have not proved the necessity for the conditions for the retention of cubic conservation laws, it is not even guaranteed that the law obeyed by the numerical method is the same as the one obeyed by the differential equation. Therefore, as far as isospectral flows are concerned, we conclude that also OLMs are of little use.

Example 2.2 We consider the order-two one-leg scheme

$$\boldsymbol{y}_{n+3} = \boldsymbol{y}_n - 2\boldsymbol{y}_{n+1} + 2\boldsymbol{y}_{n+2} + h\boldsymbol{f}\Big(\frac{1}{2}(\boldsymbol{y}_{n+1} + \boldsymbol{y}_{n+2})\Big), \tag{2.40}$$

defined in terms of the polynomials

$$\rho(z) = (z-1)(z^2 - z + 1), \quad \sigma(z) = \frac{1}{2}(z^2 + z).$$

Such method is reversible, since the conditions (2.19) are obeyed. However, for such ρ and σ, one has $\Lambda \neq I$, since

$$\Lambda = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 0 \end{pmatrix},$$

the entries of Λ being evaluated as

$$\lambda_{i,j} = \sum_{m \geq 0} (a_{i+m} b_{j+m} + a_{j+m} b_{i+m}).$$

In other words, the quadratic conservation of the underlying system is not retained, although we know that the numerical method obeys some 'related' quadratic law, namely,

$$\sum_{i,j=1}^{3} \lambda_{i,j} \boldsymbol{y}_{n+i}^T S \boldsymbol{y}_{n+j} = \text{const.} \tag{2.41}$$

To test the numerical scheme, we consider the same periodic Toda lattice already considered in Example 2.1, with the same initial condition, and we perform the integration with the OL method (2.40) in the interval $[0, 10]$ with stepsize $h = \frac{1}{10}$. As expected, the linear conservation law is retained to machine accuracy (see Figure 2.5), while the quadratic law $\mathrm{tr}(L^2) = \text{const}$ is not obeyed (see Figure 2.6). However, the relation (2.41) is obeyed to machine accuracy (see Figure 2.7) while, again, the cubic integral of the system is not retained (Figure 2.8). ◇

2.6 Runge–Kutta methods and higher order conservation laws

The generic case of cubic invariants was treated in Section 2.5.1, where we have proved that for every nonconfluent Runge–Kutta method there exists a symmetric 3-tensor S such that the method in question fails to retain S constant along the numerical trajectory. In this section, we consider the more general approach of (Calvo, Iserles & Zanna 1996) and consider the case when S is an r-tensor with $r \geq 3$. We shall show that no nonconfluent Runge–Kutta method can preserve S along the its numerical trajectory for any $r \geq 3$.

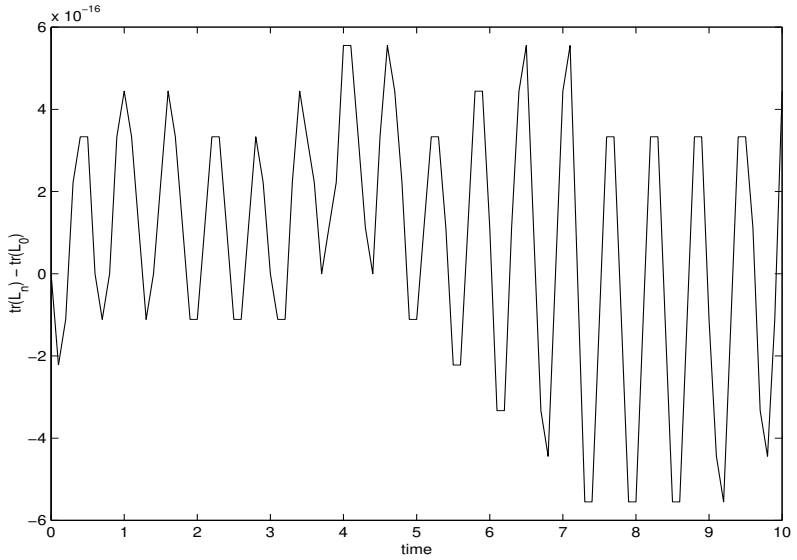


Figure 2.5: Error in the linear integral for the one-leg method (2.40). (See Example 2.2 for details).

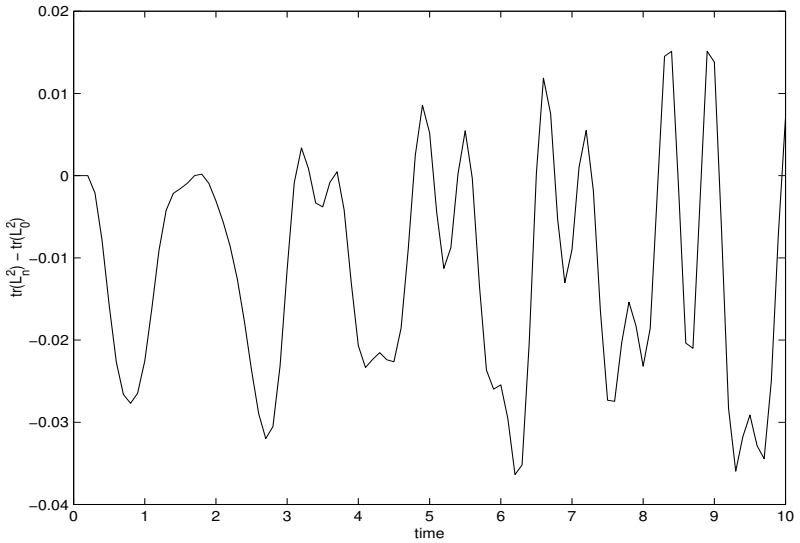


Figure 2.6: Error in the quadratic conservation law for the one-leg method (2.40). See text for details.

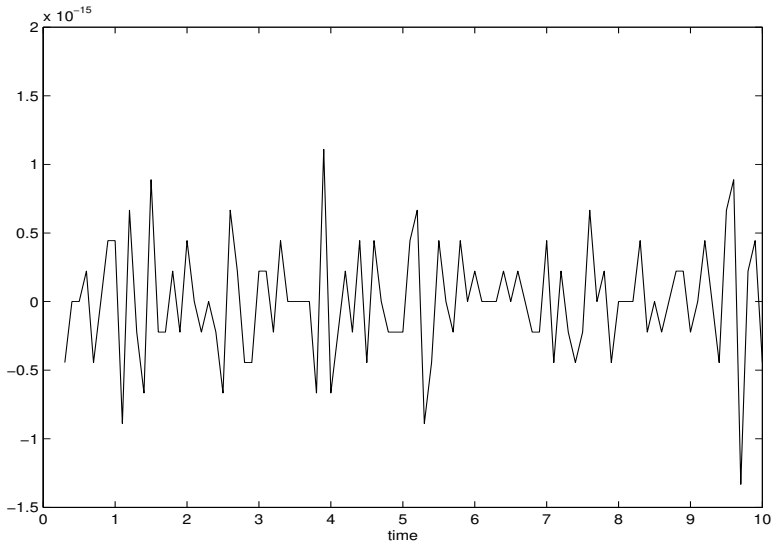


Figure 2.7: The quadratic integral (2.41) is retained to machine accuracy by the one-leg method (2.40).

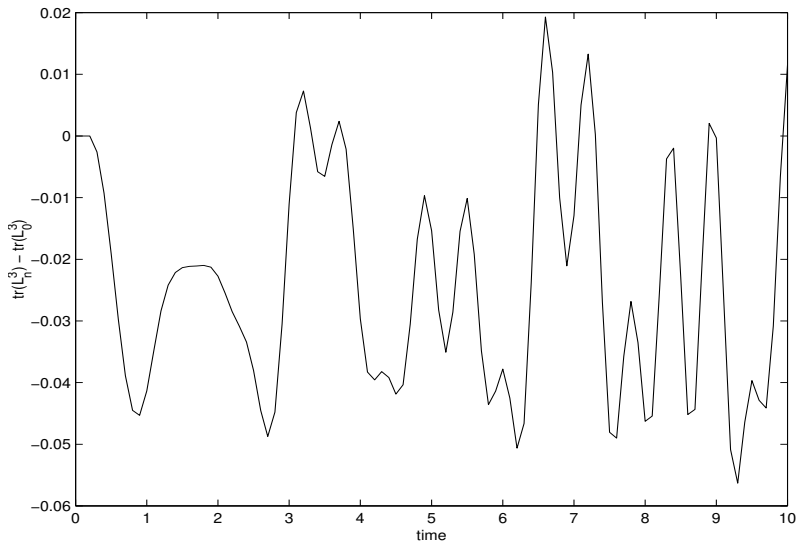


Figure 2.8: Error in the cubic integral for the one-leg method (2.40) in Example 2.2.

We denote by $S_{k_1,k_2,\ldots,k_q}$, where $1 \le k_1 < k_2 < \cdots < k_q \le r$, the value of the r-tensor S when each k_jth coordinate has been replaced by $\boldsymbol{k}_{\ell_j}$, $j = 1, 2, \ldots, q$, while all the remaining coordinates are $\boldsymbol{y}_n$. By virtue of the linearity of S,

$$\begin{aligned} S(\boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1}, \ldots, \boldsymbol{y}_{n+1}) &= S\Big(\boldsymbol{y}_n + h\sum_{\ell=1}^{\nu} b_\ell \boldsymbol{k}_\ell, \boldsymbol{y}_n + h\sum_{\ell=1}^{\nu} b_\ell \boldsymbol{k}_\ell, \ldots, \boldsymbol{y}_n + h\sum_{\ell=1}^{\nu} b_\ell \boldsymbol{k}_\ell\Big) \\ &= S(\boldsymbol{y}_n, \boldsymbol{y}_n, \ldots, \boldsymbol{y}_n) + h\sum_{k_1=1}^{r}\sum_{\ell_1=1}^{\nu} b_{\ell_1} S_{k_1} \\ &\quad + h^2 \sum_{k_1=1}^{r-1}\sum_{k_2=k_1+1}^{r}\sum_{\ell_1,\ell_2=1}^{\nu} b_{\ell_1} b_{\ell_2} S_{k_1,k_2} \\ &\quad + h^3 \sum_{k_1=1}^{r-2}\sum_{k_2=k_1+1}^{r-1}\sum_{k_3=k_2+1}^{r}\sum_{\ell_1,\ell_2,\ell_3=1}^{\nu} b_{\ell_1} b_{\ell_2} b_{\ell_3} S_{k_1,k_2,k_3} + \mathcal{O}\big(h^4\big)\,. \end{aligned}$$

Thus, our goal is to examine the discrepancy

$$d_r = S(\boldsymbol{y}_{n+1}, \boldsymbol{y}_{n+1}, \ldots, \boldsymbol{y}_{n+1}) - S(\boldsymbol{y}_n, \boldsymbol{y}_n, \ldots, \boldsymbol{y}_n).$$

Exploiting the symmetry of S, we have

$$S_{k_1,k_2,\ldots,k_q} = S(\overbrace{\boldsymbol{y}_n, \boldsymbol{y}_n \ldots, \boldsymbol{y}_n}^{r-q \text{ times}}, \boldsymbol{k}_{\ell_1}, \boldsymbol{k}_{\ell_2}, \ldots, \boldsymbol{k}_{\ell_q}),$$

hence we find that

$$d_r = \sum_{q=1}^{r} \binom{r}{q} h^q \sum_{\ell_1,\ell_2,\ldots,\ell_q=1}^{\nu} b_{\ell_1} b_{\ell_2} \cdots b_{\ell_q} S_{k_1,\ldots,k_q}. \tag{2.42}$$

Let us consider first the term corresponding to $q = 1$. According to (2.16), we may substitute

$$\boldsymbol{y}_n = \boldsymbol{\phi}_\ell - h\sum_{j=1}^{\nu} a_{\ell,j} \boldsymbol{k}_j$$

for any $\ell \in \{1, 2, \ldots, \nu\}$. Therefore

$$\begin{aligned} \sum_{\ell=1}^{\nu} b_\ell S_\ell &= \sum_{\ell=1}^{\nu} b_\ell S\Big(\boldsymbol{\phi}_\ell - h\sum_{j_1=1}^{\nu} a_{\ell,j_1}\boldsymbol{k}_{j_1}, \ldots, \boldsymbol{\phi}_\ell - h\sum_{j_{r-1}=1}^{\nu} a_{\ell,j_{r-1}}\boldsymbol{k}_{j_{r-1}}, \boldsymbol{k}_\ell\Big) \\ &= \sum_{p=0}^{r-1} (-1)^p \binom{r-1}{p} h^p \sum_{\ell=1}^{\nu}\sum_{j_1,\ldots,j_p=1}^{\nu} b_\ell a_{\ell,j_1} a_{\ell,j_2} \cdots a_{\ell,j_p} \\ &\qquad \times S(\overbrace{\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell}^{r-p-1 \text{ times}}, \boldsymbol{k}_{j_1}, \boldsymbol{k}_{j_2}, \ldots, \boldsymbol{k}_{j_p}, \boldsymbol{k}_\ell). \end{aligned}$$

Because of (2.9), $S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_\ell) = 0$, and consequently,

$$\begin{aligned}\sum_{\ell=1}^{\nu} b_\ell S_\ell &= -(r-1)h \sum_{\ell,j=1}^{\nu} b_\ell a_{\ell,j} S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_\ell) \\ &\quad + \frac{(r-2)(r-1)}{2} h^2 \sum_{\ell,j_1,j_2=1}^{\nu} b_\ell a_{\ell,j_1} a_{\ell,j_2} S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_{j_1}, \boldsymbol{k}_{j_2}, \boldsymbol{k}_\ell) \qquad (2.43) \\ &\quad + \mathcal{O}\big(h^3\big)\,.\end{aligned}$$

Next, we proceed to the $q = 2$ term. By similar substitution, we readily obtain

$$\begin{aligned}\sum_{\ell_1,\ell_2=1}^{\nu} b_{\ell_1} b_{\ell_2} S_{\ell_1,\ell_2} &= \sum_{\ell_1,\ell_2=1}^{\nu} b_{\ell_1} b_{\ell_2} S\Big(\boldsymbol{\phi}_{\ell_1} - h\sum_{j=1}^{\nu} a_{\ell_1,j}\boldsymbol{k}_j, \ldots, \boldsymbol{\phi}_{\ell_1} - h\sum_{j=1}^{\nu} a_{\ell_1,j}\boldsymbol{k}_j, \boldsymbol{k}_{\ell_1}, \boldsymbol{k}_{\ell_2}\Big) \\ &= \sum_{\ell_1,\ell_2=1}^{\nu} b_{\ell_1} b_{\ell_2} S(\boldsymbol{\phi}_{\ell_1}, \ldots, \boldsymbol{\phi}_{\ell_1}, \boldsymbol{k}_{\ell_1}, \boldsymbol{k}_{\ell_2}) \\ &\quad - (r-2)h \sum_{\ell_1,\ell_2,j=1}^{\nu} b_{\ell_1} b_{\ell_2} a_{\ell_1,j} S(\boldsymbol{\phi}_{\ell_1}, \ldots, \boldsymbol{\phi}_{\ell_1}, \boldsymbol{k}_j, \boldsymbol{k}_{\ell_1}, \boldsymbol{k}_{\ell_2}) \\ &\quad + \mathcal{O}\big(h^2\big)\,. \qquad (2.44)\end{aligned}$$

Finally, the $q = 3$ term is

$$\frac{(r-2)(r-1)r}{6} \sum_{\ell_1,\ell_2,\ell_3=1}^{\nu} b_{\ell_1} b_{\ell_2} b_{\ell_3} S(\boldsymbol{\phi}_{\ell_1}, \boldsymbol{\phi}_{\ell_1}, \ldots, \boldsymbol{\phi}_{\ell_1}, \boldsymbol{k}_{\ell_1}, \boldsymbol{k}_{\ell_2}, \boldsymbol{k}_{\ell_3}) + \mathcal{O}(h)\,. \qquad (2.45)$$

Collecting together the contributions of (2.43)–(2.45) and using time and again the symmetry of S, we expand d_r in powers of h, so that

$$\begin{aligned}d_r &= h^2\Big[-(r-1)r \sum_{\ell,j=1}^{\nu} b_\ell a_{\ell,j} S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_\ell) \\ &\qquad + \frac{(r-1)r}{2} \sum_{\ell,j=1}^{\nu} b_\ell b_j S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_\ell)\Big] \\ &\quad + h^3\Big[\frac{(r-2)(r-1)r}{2} \sum_{\ell,j,i=1}^{\nu} b_\ell a_{\ell,j} a_{\ell,i} S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_i, \boldsymbol{k}_\ell) \\ &\qquad - \frac{(r-2)(r-1)r}{2} \sum_{\ell,j,i=1}^{\nu} b_\ell b_j a_{\ell,i} S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_i, \boldsymbol{k}_\ell) \\ &\qquad + \frac{(r-2)(r-1)r}{6} \sum_{\ell,j,i=1}^{\nu} b_\ell b_j b_i S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_i, \boldsymbol{k}_\ell)\Big] \\ &\quad + \mathcal{O}\big(h^4\big)\,.\end{aligned}$$

In order to obtain $d_r = 0$, we shall require the coefficients of all powers of h to be zero. We start with the $\mathcal{O}\big(h^2\big)$ term. However, since

$$\begin{aligned} S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_\ell) &= S\Big(\boldsymbol{\phi}_j + h\sum_{i=1}^{\nu}(a_{\ell,i} - a_{j,i})\boldsymbol{k}_i, \ldots, \boldsymbol{\phi}_j \\ &\qquad + h\sum_{i=1}^{\nu}(a_{\ell,i} - a_{j,i})\boldsymbol{k}_i, \boldsymbol{k}_j, \boldsymbol{k}_\ell\Big) \qquad (2.46) \\ &= S(\boldsymbol{\phi}_j, \ldots, \boldsymbol{\phi}_j, \boldsymbol{k}_j, \boldsymbol{k}_\ell) + h\sum_{i=1}^{\nu}(a_{\ell,i} - a_{j,i})S(\boldsymbol{\phi}_j, \ldots, \boldsymbol{\phi}_j, \boldsymbol{k}_i, \boldsymbol{k}_j, \boldsymbol{k}_\ell) \\ &\quad + \mathcal{O}\big(h^2\big)\,. \end{aligned}$$

the $\mathcal{O}\big(h^2\big)$ term reduces to

$$\binom{r}{2}\sum_{\ell,j=1}^{\nu}(b_\ell b_j - b_\ell a_{\ell,j} - b_j a_{j,\ell})S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_\ell),$$

while adding a further contribution to the $\mathcal{O}\big(h^3\big)$ term.

With regards to the cubic term,

$$S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_i, \boldsymbol{k}_\ell) = S(\boldsymbol{\phi}_j, \ldots, \boldsymbol{\phi}_j, \boldsymbol{k}_j, \boldsymbol{k}_i, \boldsymbol{k}_\ell) + \mathcal{O}(h)\,,$$

so that, after some calculations, one has

$$\begin{aligned} &\binom{r}{3}\sum_{\ell,j,i=1}^{\nu}[b_\ell b_j b_i - (b_\ell b_j a_{\ell,i} + b_j b_i a_{j,\ell} + b_i b_\ell a_{i,j}) \\ &\qquad + b_j a_{j,\ell} a_{\ell,i} + b_\ell a_{\ell,i} a_{i,j} + b_i a_{i,j} a_{j,\ell})]\, S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{k}_j, \boldsymbol{k}_i, \boldsymbol{k}_\ell). \end{aligned}$$

We thus deduce that the conditions for eliminating the h^2 and h^3 terms are those already encountered in Section 2.4.2 and Section 2.5.1,

$$m_{\ell,j} = b_\ell a_{\ell,j} + b_j a_{j,\ell} - b_\ell b_j = 0, \qquad \ell, j = 1, 2, \ldots, \nu,$$

$(M = O)$, and

$$\begin{aligned} \Upsilon_{j,\ell,i} &= b_j a_{j,\ell} a_{\ell,i} + b_\ell a_{\ell,i} a_{\ell,j} + b_i a_{i,j} a_{i,\ell} \\ &\quad - (b_\ell b_j a_{\ell,i} + b_j b_i a_{j,\ell} + b_i b_\ell a_{i,j}) + b_\ell b_j b_i = 0, \end{aligned} \qquad \ell, j, i = 1, 2, \ldots, \nu,$$

$(\Upsilon = O)$ respectively, which conflict for every RK method of nontrivial order.

On the other hand, they are also necessary for nonconfluent RK schemes. Let us consider a nonconfluent method, and consider the differential equation

$$\begin{aligned} y_1' &= y_2^{r-1} g(\boldsymbol{y}), \\ y_2' &= -y_1^{r-1} g(\boldsymbol{y}), \qquad (2.47) \\ y_3' &\equiv 1, \end{aligned}$$

where $g : \mathbb{R}^3 \to \mathbb{R}$ is for the time being an arbitrary Lipschitz function. It is trivial to verify that

$$y_1^{r-1}(t)y_1'(t) + y_2^{r-1}(t)y_2'(t) \equiv 0, \qquad t \geq 0,$$

hence, for all $t \geq 0$, the ODE possesses the integral

$$y_1^r(t) + y_2^r(t) = y_1^r(0) + y_2^r(0),$$

which is equivalent to

$$S(\boldsymbol{y}, \ldots, \boldsymbol{y}) = S(\boldsymbol{y}_0, \ldots, \boldsymbol{y}_0),$$

where the $(i, i, \ldots, i)$th component of S equals one, for $i = 1, 2$, and zero otherwise. A simple calculation shows that

$$\begin{aligned} S(\boldsymbol{\phi}, \ldots, \boldsymbol{\phi}, \boldsymbol{f}(\tilde{\boldsymbol{\phi}}), \boldsymbol{f}(\boldsymbol{\phi})) &= g(\boldsymbol{\phi})g(\tilde{\boldsymbol{\phi}})(\varphi_1\varphi_2)^{r-2}(\varphi_2\tilde{\varphi}_2^{r-1} + \varphi_1\tilde{\varphi}_1^{r-1}), \\ S(\boldsymbol{\phi}, \ldots, \boldsymbol{\phi}, \boldsymbol{f}(\tilde{\boldsymbol{\phi}}), \boldsymbol{f}(\hat{\boldsymbol{\phi}}), \boldsymbol{f}(\boldsymbol{\phi})) &= g(\boldsymbol{\phi})g(\tilde{\boldsymbol{\phi}})g(\hat{\boldsymbol{\phi}})(\varphi_1\varphi_2)^{r-3}(\varphi_2^2\tilde{\varphi}_2^{r-1}\hat{\varphi}_2^{r-1} - \varphi_1^2\tilde{\varphi}_1^{r-1}\hat{\varphi}_1^{r-1}) \end{aligned}$$

where $\boldsymbol{\phi} = [\varphi_1, \varphi_2, \varphi_3]^T$.

We choose $\boldsymbol{y}_0 \in \mathbb{R}^3$ so that $y_{0,1}, y_{0,2}, y_{0,2}^r - y_{0,1}^r \neq 0$ and $y_{0,3} = 0$. Let $\boldsymbol{\phi}_s$ be a real solution of the polynomial system

$$\begin{aligned} \varphi_{s,1} &= y_{0,1} + ha_{s,s}\varphi_{s,2}^{r-1}, \\ \varphi_{s,2} &= y_{0,2} - ha_{s,s}\varphi_{s,1}^{r-1}, \end{aligned}$$

which, according to the implicit function theorem, exists as long as $h > 0$ is sufficiently small. Setting

$$\begin{aligned} \varphi_{\ell,1} &:= y_{0,1} + ha_{\ell,s}\varphi_{s,2}^{r-1}, \\ \varphi_{\ell,2} &:= y_{0,2} - ha_{\ell,s}\varphi_{s,1}^{r-1}, \end{aligned}$$

for all $\ell = 1, 2, \ldots, \nu$, $\ell \neq s$ and choosing a Lipschitz function g such that

$$g(\boldsymbol{\phi}_\ell) = \begin{cases} 1, & \ell = s, \\ 0, & \ell \in \{1, 2, \ldots, \nu\},\ \ell \neq s, \end{cases}$$

as long as $h > 0$ is sufficiently small, we have

$$S(\boldsymbol{\phi}_s, \ldots, \boldsymbol{\phi}_s, \boldsymbol{f}(\boldsymbol{\phi}_s), \boldsymbol{f}(\boldsymbol{\phi}_s)), S(\boldsymbol{\phi}_s, \ldots, \boldsymbol{\phi}_s, \boldsymbol{f}(\boldsymbol{\phi}_s), \boldsymbol{f}(\boldsymbol{\phi}_s), \boldsymbol{f}(\boldsymbol{\phi}_s)) \neq 0$$

and

$$S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{f}(\boldsymbol{\phi}_j), \boldsymbol{f}(\boldsymbol{\phi}_\ell)) = S(\boldsymbol{\phi}_\ell, \ldots, \boldsymbol{\phi}_\ell, \boldsymbol{f}(\boldsymbol{\phi}_j), \boldsymbol{f}(\boldsymbol{\phi}_i), \boldsymbol{f}(\boldsymbol{\phi}_\ell)) = 0$$

for all $\ell \neq s$, $i, j = 1, 2, \ldots, \nu$. Consequently, both $M = O$ and $\Upsilon = O$ must be true in order for the $\mathcal{O}\big(h^2\big)$ and $\mathcal{O}\big(h^3\big)$ terms to be eliminated.

The result of Theorem 2.7 is readily generalized to r-degree polynomial conservation laws, for $r \geq 3$.

Theorem 2.9 (Calvo et al. (1996)) *Let $r \geq 3$ be a given integer. No nonconfluent Runge–Kutta method can retain r-degree polynomial conservation laws, in the sense that, for every given nonconfluent Runge–Kutta scheme it is is possible to find an ODE which obeys the r-th degree conservation law $S(\boldsymbol{y}, \ldots, \boldsymbol{y}) = \text{const}$, but such relation is not obeyed by the numerical approximation of method in question.*

Thus, even though RK schemes have many features that make them appealing in other contexts, there is a restriction to how well RK schemes retain polynomial conservation laws under discretization.

Comments

Numerical methods and their retention of quadratic conservation laws have been widely investigated in the last few years especially in conjunction with Hamiltonian systems (Sanz-Serna & Calvo 1994). The condition (2.21), $M = O$, for instance, characterizes all those RK schemes which are symplectic. The matrix M was introduced by Butcher (1975) and is ubiquitous in the analysis of RK methods. In 1988, Lasagni, Suris and Sanz-Serna discovered independently that RK methods for which $M = O$ are symplectic (Sanz-Serna & Calvo 1994). The symplecticity condition is essentially the same as the conservation of quadratic laws. But unfortunately, the condition $M = O$ is not sufficient for our purposes, i.e. when the system possesses some other integrals which are nonlinear.

The passage from quadratic to cubic conservation laws shows clearly the limitations of classical numerical methods, since the conditions for quadratic and cubic invariance are contradictory.

Classical numerical methods for ODEs are usually obtained advancing the variable y_n, that approximates the unknown solution, by a linear combination of past values and 'tangent vectors' in some points. Multistep methods, Runge–Kutta methods and one-leg methods are constructed according to this criterion—hence it is clear that they retain linear conservation laws. Requiring that the solution stays by design on a generic manifold (embedded in a Euclidean space) is somewhat more complicated. Those methods that by construction share the same attributes as the unknown solution are often called *geometric integrators.*

But why should we use geometric integrators? Do they produce any better numerical approximations? If the exact solution lives on some manifold, can't we simply use an arbitrary numerical method and then project the numerical approximation on the manifold? The reason is that a numerical method that shares the main mathematical attributes of the underlying solution is usually better than a projection method. This has been highlighted already in the case of Hamiltonian systems: for instance, the error of symplectic methods or energy-preserving methods accumulates at a slower pace (Cano & Sanz-Serna 1996, Stuart & Humphries 1996). For geometric integrators, the error has less directions of propagation. When using projection methods, instead, first of all we produce a numerical error due to the ODE integrator (whose numerical approximation departs from the manifold), then we generate another kind of error when projecting back on the manifold. So, although we are back on the manifold, we allow the error to accumulate by two different mechanisms.

3

Isospectral methods

In Chapter 2 we have discussed classical ODEs methods and their retention of conserved integrals to conclude that their behaviour in the preservation of conservation laws is generally quite poor. The question we address in this chapter is therefore the following: Is it possible to devise numerical methods that *are* isospectral, for the matrix Lax pair $L' = [B(L), L]$?

As we shall see, the answer is positive, and these methods will rely on our ability of devising schemes for *orthogonal flows.* For this reason, a significant portion of Chapters 3 and 4 is devoted to the numerical solution of orthogonal flows.

3.1 Isospectral solutions via the Lax formalism

As already seen in Section 1.1, given the Lax pair (1.1),

$$L' = [B(L), L], \qquad L(0) = L_0,$$

where L_0 is a symmetric initial condition and $B(L)$ is a skew-symmetric matrix function of L, then the solution $L(t)$ is a symmetric matrix and is isospectral for all values of t. On the other hand, if we evaluate our numerical approximation to the system (1.1) directly with a standard ODE method, we are bound to lose isospectrality, as a consequence of the analysis presented in Chapter 2.

However, it is useful to recall the proof of isospectrality of (1.1), already presented in section 1.1. We have seen that, if $U(t)$ is the solution of the ODE (1.2),

$$U' = B(L)U, \qquad U(0) = I,$$

which, because of the skew-symmetry of B, is an *orthogonal flow*, then the function $L(t)$ can be obtained by means of the similarity transformation (1.4),

$$L(t) = U(t)L_0U(t)^T.$$

Therefore, the main idea at the basis of isospectral numerical methods is to solve the orthogonal flow (1.2) instead; then, provided that U_n is the numerical approximation for $U(t_n)$, we approximate $L(t_n)$ as $U_nL_0U_n^T$. Numerically, we consider the following procedure:

1. Assume that L_n is an approximation for the exact flow $L(t_n)$ at time $t = t_n$ and that it is isospectrally similar to L_0;

2. Let U_{n+1} be the numerical approximation at time $t = t_{n+1}$ to the solution of the orthogonal flow

$$U' = B(L)U, \qquad U(t_n) = I, \quad t_n \leq t \leq t_{n+1}. \tag{3.1}$$

3. Set

$$L_{n+1} = U_{n+1} L_n U_{n+1}^T. \tag{3.2}$$

as the numerical approximation $L_{n+1} \approx L(t_{n+1})$.

Provided that U_{n+1} is an *orthogonal matrix,* the matrix L_{n+1} in (3.2) is symmetric and isospectrally similar to L_n and, by induction, to L_0.

In the case when U_{n+1} is not orthogonal, the formula (3.2) does not preserve isospectrality. We might consider replacing (3.2) with the *non-symmetric* similarity transformation

$$L_{n+1} = U_{n+1} L_n U_{n+1}^{-1}, \tag{3.3}$$

with the disadvantage that we lose the symmetry of the numerical approximation. Such numerical schemes will be described below in Section 3.4.

3.2 Numerical methods for Orthogonal Flows

In this section, we address the problem of approximating numerically the solution of the differential equation

$$U' = B(U)U, \qquad U(0) = I, \tag{3.4}$$

where B is a skew-symmetric matrix function of U. In such case, the solution of (3.4) obeys the orthogonality condition

$$U^T U = U U^T = I, \tag{3.5}$$

i.e. (3.4) is an orthogonal flow.

Definition 3.1 Let $0 = t_0 < t_1 < \cdots < t_n < \cdots$ be a partition of the time interval $[0, T)$ (eventually $T = +\infty$) and let $U_n \approx U(t_n)$ be an orthogonal numerical approximation to the true solution of (3.4). We say that a one-step method for ODEs is *orthogonal* if it produces a numerical approximation U_{n+1} such that $U_{n+1}^T U_{n+1} = U_{n+1} U_{n+1}^T = I$. ◇

The above definition can be easily extended to *Stiefel manifolds,* i.e. orthogonal rectangular matrices of dimension $d_1 \times d_2$, with $d_2 < d_1$. Also, in a very similar way, we might introduce the concept of orthogonality for multistep methods.

Before proceeding further, we point out that the orthogonality condition (3.5) is essentially a quadratic conservation law (it suffices to order U columnwise and to consider the corresponding quadratic conservation law for vectors). Hence, by virtue of our discussion in Chapter 2 it is possible to identify those numerical schemes that are candidates for orthogonality: such methods will be all those that preserve quadratic invariants, while the remaining methods will generally produce a numerical solution that departs from the orthogonal manifold.

We abstain from considering rectangular orthogonal flows here, and, although most of the approaches described here apply to the rectangular case, we refer the interested reader to the review article of Dieci & van Vleck (1996).

3.2.1 Projection methods

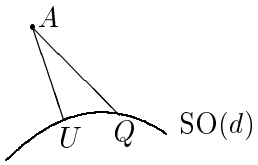
Given that the majority of numerical methods for ODEs is not preserving quadratic conservation laws, an easy way to construct orthogonal methods is by means of *projection methods* (Dieci, Russel & van Vleck 1994). The basic idea for a one-step method is as follows: Assume that U_n is an orthogonal approximation at $t = t_n$ and let $\widehat{U}_{n+1}$ be a numerical approximation for the true value $U(t_{n+1})$ computed with a non-orthogonal method. Then we can evaluate the QR factorization of $\widehat{U}_{n+1} = Q_{n+1}R_{n+1}$ or a Gram-Schmidt orthogonalization (Golub & van Loan 1989) and set $U_{n+1} = Q_{n+1}$, the orthogonal factor. Similarly, the procedure extends to multistep methods. In particular, if the numerical method is of order p, then also the projected method will display order p.

Although this approach sounds very appealing, since in principle we can employ our favourite numerical scheme, there are two main drawbacks:

1. First of all, the QR factorization is not a *best approximation* in any norm. On the other hand, in the *Frobenius* norm,

$$\|A\|_F^2 = \mathrm{tr}(AA^T),$$

   the best approximation is known, and it is the *orthogonal polar factor.* We recall that, given a full rank $d \times d$ matrix A, there exist a unique orthogonal matrix U (the polar factor) and a unique symmetric positive definite matrix H such that $A = UH$. Such decomposition is called *polar decomposition* of the matrix A (Higham 1989).



2. The QR or the polar decomposition produce orthogonal factors whether or not the exact solution is an orthogonal flow! Ideally, we would like to use a numerical method that is 'intelligent', in other words that produces an orthogonal approximation when the underlying solution is orthogonal, and a non-orthogonal approximation otherwise.

The polar factorization applied in the ODE context is considered in (Higham 1997). The polar factorization of a matrix A can be computed by evaluating the singular value decomposition of A or by means of an iterative procedure (Higham 1997). The first one,

$$\begin{aligned} U_0 &= A, \\ U_{i+1} &= \tfrac{1}{2}(U_i + U_i^{-T}), \qquad i = 0, 1, 2, \ldots, \end{aligned} \tag{3.6}$$

is also called *Newton iteration* because it can be obtained by applying the classical Newton method for finding zeros of nonlinear functions to the function $f(U) = U^TU - I$.

The second method is based on the *Schultz* iteration,

$$\begin{aligned} U_0 &= A, \\ U_{i+1} &= U_i\Big(I + \tfrac{1}{2}(I - U_i^TU_i)\Big), \qquad i = 0, 1, 2, \ldots, \end{aligned} \tag{3.7}$$

and a sufficient condition for convergence is that $\|A^TA - I\|_2 < 1$.

Both the Newton and the Schultz iterations are locally quadratically convergent, the cost depending heavily on the nature of the problem, and their application to the solution of orthogonal flows is described at a greater length in (Higham 1997).

3.2.2 Runge–Kutta schemes

In Chapter 2 we have learnt that RK methods that obey the condition $M = O$ retain quadratic invariants. Therefore, we wish to apply such schemes to the numerical solution of orthogonal flows, since the orthogonality condition $U^TU = I$ is essentially a quadratic conservation law. However, the orthogonality of Runge–Kutta schemes whose coefficients obey the condition $M = O$ can be proved directly.

Theorem 3.1 (Calvo et al. (1995)) *Consider the differential equation* (3.4)*, where the matrix function $B(U)$ is skew-symmetric, and let $\{U_n\}_{n=0}^{\infty}$ be the numerical approximation produced by an RK scheme. If coefficients and weights of the scheme obey the condition $M = O$ then U_n is orthogonal for all $n \geq 0$.*

Proof. Recall that

$$U_{n+1} = U_n + h\sum_{i=1}^{\nu} b_i K_i,$$

where, having denoted $B(\Phi_\ell)$ by B_ℓ,

$$\begin{aligned} K_\ell &= B_\ell\Phi_\ell, \qquad\qquad \ell = 1,\ldots,\nu, \\ \Phi_\ell &= U_n + h\sum_{j=1}^{\nu} a_{\ell,j}K_j. \end{aligned}$$

Hence

$$\begin{aligned} U_{n+1}^TU_{n+1} &= U_n^TU_n + h\sum_{\ell=1}^{\nu} b_\ell(U_n^TK_\ell + K_\ell^TU_n) \\ &\quad + h^2\sum_{\ell=1}^{\nu}\sum_{j=1}^{\nu} b_\ell b_j K_\ell^TK_j. \end{aligned} \tag{3.8}$$

We find

$$\begin{aligned} U_n^TK_\ell &= \Phi_\ell^TK_\ell - h\sum_{j=1}^{\nu} a_{\ell,j}K_j^TK_\ell, \\ K_\ell^TU_n &= K_\ell^T\Phi_\ell - h\sum_{j=1}^{\nu} a_{\ell,j}K_\ell^TK_j. \end{aligned}$$

However, one has

$$\Phi_\ell^TK_\ell + K_\ell^T\Phi_\ell = \Phi_\ell^TB_\ell\Phi_\ell + \Phi_\ell^TB_\ell^T\Phi_\ell = \Phi_\ell^T(B_\ell + B_\ell^T)\Phi_\ell = O, \tag{3.9}$$

since B_ℓ is a skew-symmetric matrix. It follows that (3.8) reduces to

$$U_{n+1}^T U_{n+1} = U_n^T U_n + h^2 \sum_{\ell=1}^{\nu} \sum_{j=1}^{\nu} (b_\ell b_j - b_\ell a_{\ell,j} - b_j a_{j,\ell}) K_j^T K_\ell.$$

Assuming that $U_n^T U_n = I$, it is clear that if $M = O$ then

$$U_{n+1}^T U_{n+1} = I.$$

Similarly, $U_{n+1} U_{n+1}^T = I$, and the proof follows by induction on n. □

Corollary 3.1.1 (Dieci et al. (1994)) *Subject to the same assumptions of Theorem 3.1, Gauss–Legendre Runge–Kutta methods are orthogonal.*

However, it was pointed out by Higham (1996) that the above theorems are valid only for a restricted class of orthogonal flows, namely when the function B is skew-symmetric regardless of the orthogonality of its argument. If $B(U) = -B(U)^T$ *if and only if* U is orthogonal, then, even methods that obey $M = O$ will behave like the non-orthogonal schemes: in general, the internal stages, the Φ_ℓ's, do not have to be orthogonal and consequently the error term (3.9) does not vanish since the B_ℓ's are not skew-symmetric. In effect, accordingly to the terminology of Section 2.1, a function that is skew-symmetric only for orthogonal arguments does not obey the relation (2.9). We can reconsider the Definition 3.1 saying that a method that produces orthogonal numerical approximations only when $B(U)$ is skew-symmetric for all U is *weakly orthogonal.* Thus, Runge–Kutta methods with $M = O$ are only weakly orthogonal.

The weak orthogonality of the Implicit Midpoint Rule, whose coefficients obey the condition $M = O$, is displayed in Figure 3.1, where we plot the orthogonality error $\|U_n^T U_n - I\|$ against time. The top plot corresponds to the case when

$$B(U) = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \tag{3.10}$$

i.e. B is always skew-symmetric, while the bottom plot corresponds to

$$B(U) = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} - (U^T U - I), \tag{3.11}$$

which corresponds to a slight perturbation of (3.10), so that $B(U)$ is skew-symmetric for orthogonal arguments only. The initial condition is chosen to be the identity matrix.

Higham (1996) proposed two methods of order one and two respectively that are orthogonal even when B is skew-symmetric for orthogonal arguments only. These method mimic the RK formalism, so that the first method (order one) is

$$U_{n+1} = U_n + \frac{h}{2} B_n (U_{n+1} + U_n), \tag{3.12}$$

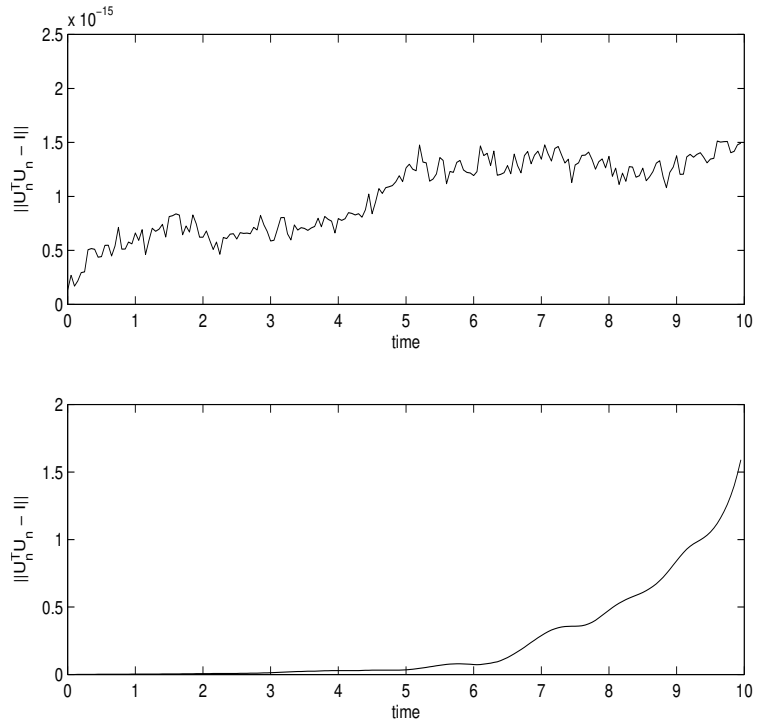


Figure 3.1: Orthogonality error for the IMR (that obeys $M = O$), for the matrix function $B(U)$ as in (3.10) (top) and (3.11) (bottom). The stepsize and interval of integration are $h = \frac{1}{20}$ and $[0, 10]$ respectively.

where $U_n \in \mathbb{R}^{d\times d}$, $U_n^T U_n = I$ and $B_n = B(U_n)$.

The scheme (3.12) is only linearly implicit, since U_{n+1} can be evaluated solving a linear system,

$$U_{n+1} = \Big(I - \frac{h}{2}B_n\Big)^{-1}\Big(I + \frac{h}{2}B_n\Big)U_n,$$

(note the resemblance with the IMR for linear ODEs). Similarly, the order two scheme is

$$\begin{aligned} K_{\frac{1}{2}} &= B_n\Big(U_n + \tfrac{h}{4}K_{\frac{1}{2}}\Big), \\ K_1 &= B(U_n + \tfrac{h}{2}K_{\frac{1}{2}})\Big(U_n + \tfrac{h}{2}K_1\Big), \\ U_{n+1} &= U_n + hK_1. \end{aligned} \tag{3.13}$$

Also this scheme is linearly implicit, since it requires solving a linear system for finding $K_{\frac{1}{2}}$ and K_1. In Figure 3.2 we display the orthogonality error of the schemes (3.12) when B is given by (3.10) and (3.11). The error from orthogonality is of the order of machine accuracy in the both cases, although in the bottom plot there is an evident accumulation of rounding error.

Generalization to higher order schemes is a little more complicated. As we have shown in (Zanna 1996), the numerical schemes (3.12) and (3.13) can be obtained by means

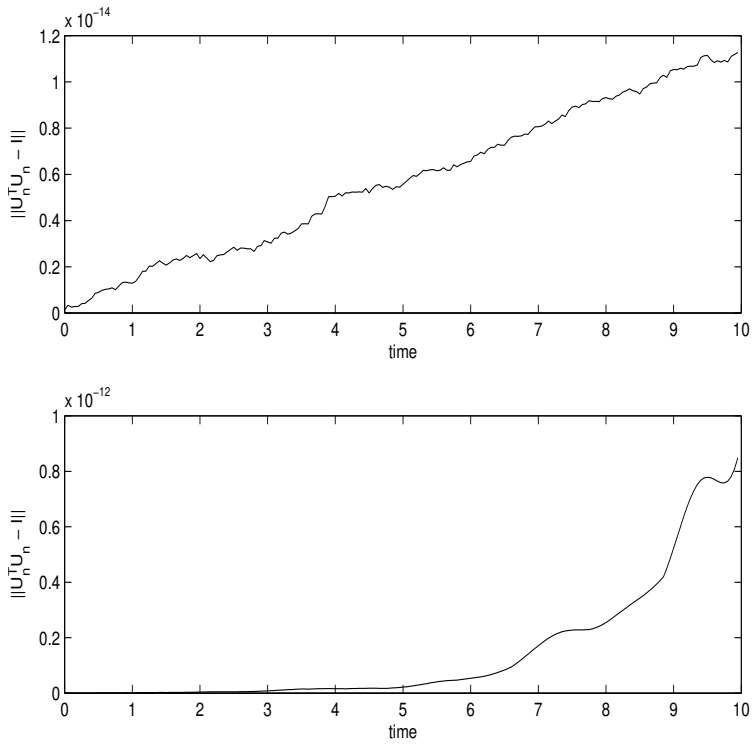


Figure 3.2: Orthogonality error for the schemes (3.12), for the matrix function $B(U)$ as in (3.10) (top) and (3.11) (bottom). The stepsize and interval of integration are $h = \frac{1}{20}$ and $[0, 10]$ respectively.

of *Lie-group* methods, whereby the exponential function is replaced by the $(1, 1)$ Padé approximant.

3.2.3 Cayley transforms

It is well known (see for instance Golub & van Loan (1989)) that, given a skew-symmetric matrix A, its *Cayley transform,*

$$U = (I - A)^{-1}(I + A), \tag{3.14}$$

is an orthogonal matrix. Similarly, provided that $-1 \notin \sigma(U)$, the *inverse* Cayley transform

$$A = (U - I)(U + I)^{-1}, \tag{3.15}$$

is well defined and is a skew-symmetric matrix function. Both the relations (3.14) and (3.15) hold for unitary matrices, whereby the adjective skew-symmetric is replaced by skew-Hermitian.

The idea of using the inverse Cayley transform (3.15) to solve the orthogonal (unitary) flow (3.4) was proposed in (Diele, Lopez & Peluso 1996). Given the ODE (3.4), Diele et

al. find the differential equation for A by differentiating (3.15), so that

$$A' = \frac{1}{2}(I - A)B(t, (I - A)^{-1}(I - A))(I + A) = H(t, A), \qquad A(0) = O, \tag{3.16}$$

where $H(t, A)$ is a skew-symmetric (skew-Hermitian) matrix function. The space of skew-symmetric matrices is a linear space, therefore all linear methods for ODEs (whether Linear Multistep Methods, or Runge–Kutta methods or one-leg methods) will preserve the skew-symmetry of A. Thus, provided that A_n is the numerical approximation at time $t = t_n$ obtained with one of these linear schemes, the approximate orthogonal solution at $t = t_n$ is obtained by means of the Cayley transform (3.14),

$$U_n = (I - A_n)^{-1}(I + A_n).$$

The inverse Cayley transform method is suitable for square systems only and its numerical implementation does require the solution of a number of linear systems. Also, in some cases the matrix A tends to become very large in magnitude although the solution of (3.4) stays bounded, since the Cayley transform is not singularity free. This usually happens when one (or more) of the eigenvalues of U approach -1 and in that case Diele et al. (1996) suggest to implement a restarting procedure.

In Figure 3.3 we have plotted the orthogonality error for the flows with B as in (3.10) and (3.11). The differential equation (3.16) is solved with the explicit RK method known as *Heun's* method,

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array},$$

which possesses order two.

In the examples (3.10) and (3.11), the exact solution is $U(t) = \exp(tB)$, so that the eigenvalues, as functions of time, are $\lambda_1(t) = 1, \lambda_{2,3}(t) = \exp(\pm it\sqrt{3})$. Thus, at $t = \pi/\sqrt{3} \approx 1.81$ two of the eigenvalues of U approach -1, the algorithm becomes very ill conditioned and a restarting procedure is needed. In these very simple examples, the algorithm will require a restarting procedure every time t approaches $t_k = \pi(1 + 2k)/\sqrt{3}$, for $k = 0, 1, 2, \ldots$. In the more general case, when the solution U is less trivial, one should monitor the eigenvalues of U and keep track of the past numerical approximations in case of a restarting procedure. Such implementation might be very cumbersome, although numerical experiments performed by Owren[1] seem to indicate that performing a 'restarting procedure' at each step leads to methods that are quite competitive and accurate.

3.2.4 Lie-group methods

The last approach that we mention here in our review of methods for the ODE (3.4) is based on *Lie-group* methods, which are described in greater detail in Chapter 4.

The basic idea is that orthogonal matrices constitute a Lie group and therefore we want to take advantage of the underlying geometry of Lie groups to construct *ad-hoc* numerical methods.

[1] Personal communication via H. Munthe-Kaas, February 1998.

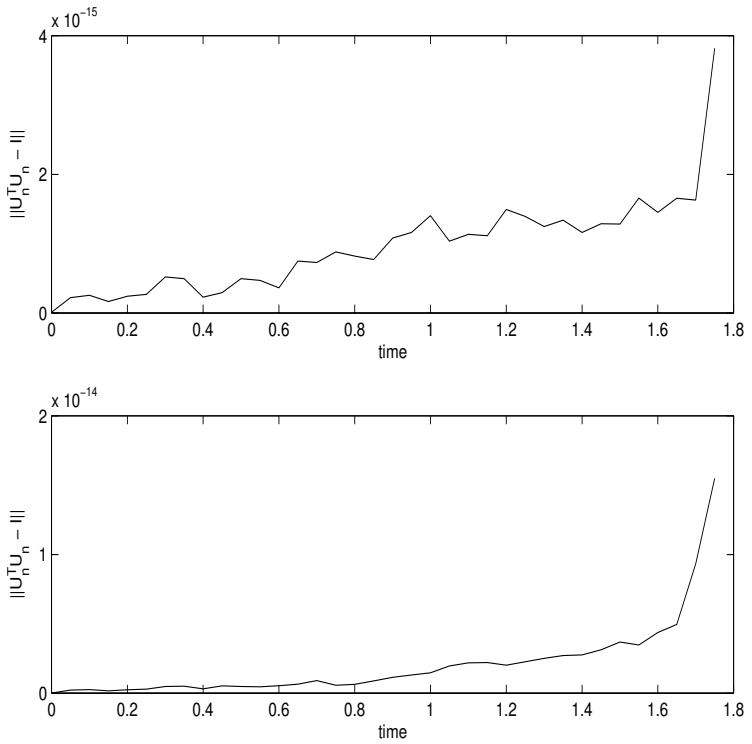


Figure 3.3: Orthogonality error for the inverse Cayley method solved with Heun's method. The matrix function $B(U)$ is given by (3.10) (top) and (3.11) (bottom). The stepsize of integration is $h = \frac{1}{20}$. At time $t \approx 1.81$ two complex conjugate eigenvalues approach -1 and the procedure has to be restarted.

The motivation is twofold: First of all, a number of ODEs are really flows on Lie groups, for instance orthogonal flows, unitary flows, flows on homogeneous manifolds (a.k.a. affine transformations), and some linear volume-preserving ODEs. Hence there exists enough generality to motivate a class of numerical methods that deals well with these problems.

Secondly, it has been shown that geometrical integrators (e.g. symplectic methods (Sanz-Serna & Calvo 1994), energy-preserving methods (Estep & Stuart 1995), reversible schemes (Hairer & Stoffer 1997)) have good properties with regards to error propagation. Therefore we might expect that Lie-group methods will produce a better numerical approximation to the true solution of the problem.

3.3 The early isospectral methods

3.3.1 The Lie-bracket method

The *Lie-bracket method* was introduced by Moore, Mahoney & Helmke (1994) in their studies of the double-bracket flow

$$L' = [[N, L], L], \qquad L(0) = L_0, \quad (N \text{ diagonal}), \tag{3.17}$$

for the singular value decomposition of a given matrix L_0 (see Section 1.4). Moore et al. (1994) considered the orthogonal flow

$$U' = [N, L]U, \qquad U(0) = I, \tag{3.18}$$

associated with (3.17), and solved (3.18) with the one-step scheme

$$U_{n+1} = \mathrm{e}^{h_n[N,L_n]}U_n, \qquad U_n = I, \quad h_n = t_{n+1} - t_n, \tag{3.19}$$

together with the similarity transformation

$$L_{n+1} = \mathrm{e}^{h_n[N,L_n]}L_0\mathrm{e}^{-h_n[N,L_n]}. \tag{3.20}$$

Since $[N, L_n]$ is a skew-symmetric matrix, $\mathrm{e}^{h_n[N,L_n]}$ is an orthogonal matrix, hence (3.20) is an isospectral transformation and the method is isospectral.

Although this scheme was originally proposed for double-bracket flows, its generalization to the isospectral flow (1.1) is straightforward: given the equation $L' = [B(L), L]$, we consider the orthogonal flows $U' = B(L)U$ so that (3.19) and (3.20) become

$$U_{n+1} = \mathrm{e}^{h_n B(L_n)}U_n \quad \text{and} \quad L_{n+1} = \mathrm{e}^{h_n B(L_n)}L_0\mathrm{e}^{-h_n B(L_n)}, \tag{3.21}$$

respectively.

The algorithm (3.21) is explicit and it can be interpreted as an exponentiated form of the *forward Euler* method. Moreover, it is easy to verify that it possesses order one only: assuming $U_n = U(t_n)$, $L_n = L(t_n)$, and letting $B_n = B(L_n)$, we find

$$\begin{aligned} U_{n+1} &= \mathrm{e}^{h_n B_n}U_n = (I + h_n B_n + \frac{h_n^2}{2}B_n^2 + \frac{h^3}{3!}B_n^2 + \cdots)U_n \\ &= U_n + h_n B_n U_n + \frac{h_n^2}{2}B_n^2 U_n + \cdots \end{aligned}$$

which, compared with the true expansion of U,

$$U(t_{n+1}) = U_n + h_n B_n U_n + \frac{h_n^2}{2}(B_n' + B_n^2)U_n + \mathcal{O}\left(h_n^3\right),$$

affirms that $U_{n+1} = U(t_{n+1}) + \mathcal{O}\left(h_n^2\right)$, hence order one. As a consequence, $L_{n+1} = L(t_{n+1}) + \mathcal{O}\left(h_n^2\right)$ is a numerical approximation of order one to the true solution of (1.1).

Moore et al. proposed two variable stepsize strategies for the implementation of the Lie-bracket method and proved that the algorithm converges to an attractive equilibrium point exponentially fast for a generic initial condition. But no generalization to higher order methods was presented. As shown in (Zanna 1996), the Lie-bracket method can be interpreted as an order-one Lie-group method.

3.3.2 Modified Gauss–Legendre schemes

While the Lie-bracket method introduced above in Section 3.3.1 has order one only, modified *Gauss-Legendre* schemes, introduced in (Calvo et al. 1995), can be used to devise

isospectral numerical methods for (1.1) of arbitrarily high order. As the name itself indicates, such schemes are based on Gauss–Legendre Runge–Kutta (GLRK) schemes. We consider the orthogonal flow

$$U' = B(L)U, \qquad U_0 = I,$$

and observe that the function B depends on $L(t)$, which is unknown for $t > t_n$. To apply a GLRK scheme, we would need also the values of L at the Gauss–Legendre nodes $t_n + c_\ell h$, for $\ell = 1, \ldots, \nu$. Alternatively, we might solve

$$U' = B(UL_0U^T)U, \qquad U_0 = I,$$

i.e. considering B as a function of U. But when B is not necessarily skew-symmetric on the internal stages,[2] in that case the GLRK method will fail to produce an orthogonal approximation because they are only weakly orthogonal, as already pointed out in Section 3.2.2.

One way of overcoming this difficulty is by introducing a polynomial $\tilde{L}(t)$ which interpolates the exact flow $L(t)$ at the points t_n and t_{n+1}. Assume that we use a ν-stage GLRK scheme of order 2ν and that the function B is sufficiently smooth with respect to the variable L. Let

$$\tilde{L}^{(j)}(t_{n+i}) = L^{(j)}(t_{n+i}), \qquad i = 0, 1, \quad j = 0, 1, \ldots, \nu - 1,$$

be the Hermite polynomial interpolant of L of degree $2\nu - 1$ at the points $\{t_n, t_{n+1}\}$. The interpolation error is given by

$$\tilde{L}(t) - L(t) = \frac{1}{(2\nu)!}(t - t_n)^\nu (t - t_{n+1})^\nu \Lambda_t,$$

where the (i, j)-th element of the matrix Λ_t is $\ell_{i,j}^{(2\nu)}(\theta_t^{(i,j)})$. If we replace $B(t, L)$ with $B(t, \tilde{L})$, the leading error term is

$$E(t) = \frac{1}{(2\nu)!}(t - t_n)^\nu (t - t_{n+1})^\nu \Lambda_t B_L(t, \tilde{L}). \tag{3.22}$$

Therefore $E(t) = \mathcal{O}\big(h^{2\nu}\big)$. Furthermore, this kind of interpolation does not affect the order of the GLRK method: Consider the two differential equations

$$U' = B(t, L(t))U, \qquad U(t_n) = I, \tag{3.23}$$
$$V' = B(t, \tilde{L}(t))V, \qquad V(t_n) = I. \tag{3.24}$$

Obviously, the latter can be regarded as a perturbation of (3.22). Denoting by $\Psi_B(t)$ the fundamental solution of (3.23), as a consequence of the Alekseev–Gröbner lemma (Hairer et al. 1993), we deduce that

$$V(t) - U(t) = \int_{t_n}^{t} \Psi_B(t - \tau) \left(V'(\tau) - B(\tau, L(\tau))V(\tau)\right) \, \mathrm{d}\tau,$$

[2] For instance if we use UL_0U^{-1} instead of UL_0U^T, which are the same along the theoretical trajectory but not along the numerical one, or solve with Newton's method without iterating to convergence.

and, taking into account that $\Psi_B(t)$ is orthogonal for all t, we obtain the inequality

$$\begin{aligned}\|V(t)-U(t)\|_2 &\leq (t-t_n)\max_{t\in[t_n,t_{n+1}]}\|V'-B(t,L(t))V\|_2\\ &\leq h\,\|B(t,\tilde{L}(t))-B(t,L(t))\|_\infty\times\|V\|_2\\ &= h\,\|B(t,\tilde{L}(t))-B(t,L(t))\|_\infty = h\,\|E(t)\|_\infty. \end{aligned}\tag{3.25}$$

Bearing in mind that $\|E(t)\|_\infty$ is of order $\mathcal{O}\big(h^{2\nu}\big)$, we deduce that $\|V(t)-U(t)\|_2$ is of order $\mathcal{O}\big(h^{2\nu+1}\big)$, provided that we can bound B_L and the higher derivatives of L appearing in the leading error term in (3.22). Hence letting

$$L_{n+1} = V_{n+1}L_nV_{n+1}^T,$$

we are committing at most an error of $\mathcal{O}\big(h^{2\nu+1}\big)$, which is subsumed in the error of a numerical method of order 2ν.

We refer to a *modified* Gauss–Legendre RK method (MGLRK) of order 2ν whenever we apply the classical Gauss–Legendre RK method of order 2ν to (3.24) in tandem with $L_{n+1} = V_{n+1}L_nV_{n+1}^T$. The corresponding methods are implicit, since we need L_{n+1} and, in general, up to the $(\nu-1)$-st derivative of L at the point t_{n+1} to derive the interpolating polynomial $\tilde{L}(t)$. To solve the implicit equations, we could use Newton's method for finding the zeros of nonlinear functions, but in this case we need to iterate to machine accuracy, otherwise the internal stages need not to be orthogonal. Alternatively, one might use more a Lie-group variant of Newton's method (Owren & Welfert 1996). Another choice, the simplest, is a Picard iteration. We set (as an initial guess)

$$L_{n+1}^{(j)\ [0]} = L_n^{(j)}, \qquad j=0,1,\ldots,\nu-1,\tag{3.26}$$

and subsequently use $L_n^{(j)}, L_{n+1}^{(j)\ [k]}$, for $k=0,1,\ldots,$ to construct the Hermite interpolating polynomials $\tilde{L}^{[k]}$, and to solve

$$V^{[k]\prime} = B(t,\tilde{L}^{[k]})V^{[k]}, \qquad V^{[k]}(t_n)=I.\tag{3.27}$$

At each step, $B(t,\tilde{L}^{[k]})$ is a function of t only, so that (3.27) can be solved explicitely inverting a linear system. Then, we advance the numerical approximation of the unknown function L by means of

$$L_{n+1}^{[k+1]} = V_{n+1}^{[k]}L_nV_{n+1}^{[k]\ T}.\tag{3.28}$$

By construction, the matrices $V_{n+1}^{[k]}$ are all orthogonal, for $k=1,2,\ldots$. Hence the Picard iterations need not to be performed to machine accuracy and we might stop when the iteration error, $\|V_{n+1}^{[k]}-V_{n+1}^{[k-1]}\|$, is sufficiently small.

The simplest method in the family of modified Gauss–Legendre RK is obtained by letting $\nu=1$. In this case,

$$\tilde{L}(t) = \frac{1}{t_{n+1}-t_n}\left[(t_{n+1}-t)L_n+(t-t_n)L_{n+1}\right],$$

therefore, denoting

$$B_{n+\frac{1}{2}} = B\left(t_n + \tfrac{1}{2}h, \frac{L_n + L_{n+1}}{2}\right), \tag{3.29}$$

the scheme yields

$$V_{n+1} = \left(I - \tfrac{1}{2}hB_{n+\frac{1}{2}}\right)^{-1}\left(I + \tfrac{1}{2}hB_{n+\frac{1}{2}}\right). \tag{3.30}$$

Hence we let

$$L_{n+1} = V_{n+1}L_nV_{n+1}^T. \tag{3.31}$$

We refer to (3.29)–(3.31) as the *modified implicit midpoint rule* (MIMR).

Theorem 3.2 *The algorithm* (3.29)–(3.31) *is a second order isospectral modification of the implicit midpoint rule when applied to* (1.1).

Proof. We use (3.30) to eliminate V_{n+1} in (3.31),

$$L_{n+1} = \left(I - \tfrac{1}{2}hB_{n+\frac{1}{2}}\right)^{-1}\left(I + \tfrac{1}{2}hB_{n+\frac{1}{2}}\right)L_n\left(I - \tfrac{1}{2}hB_{n+\frac{1}{2}}\right)\left(I + \tfrac{1}{2}hB_{n+\frac{1}{2}}\right)^{-1},$$

and, multiplying both sides of the latter equation by $\left(I - \frac{1}{2}hB_{n+\frac{1}{2}}\right)$ on the left and by $\left(I + \frac{1}{2}hB_{n+\frac{1}{2}}\right)$ on the right, we obtain

$$\left(I - \tfrac{1}{2}hB_{n+\frac{1}{2}}\right)L_{n+1}\left(I + \tfrac{1}{2}hB_{n+\frac{1}{2}}\right) = \left(I + \tfrac{1}{2}hB_{n+\frac{1}{2}}\right)L_n\left(I - \tfrac{1}{2}hB_{n+\frac{1}{2}}\right).$$

Thus

$$\begin{aligned} L_{n+1} - \tfrac{1}{2}hB_{n+\frac{1}{2}}L_{n+1} + \tfrac{1}{2}hL_{n+1}B_{n+\frac{1}{2}} - \tfrac{1}{4}h^2B_{n+\frac{1}{2}}L_{n+1}B_{n+\frac{1}{2}} \\ = L_n + \tfrac{1}{2}hB_{n+\frac{1}{2}}L_n - \tfrac{1}{2}hL_nB_{n+\frac{1}{2}} - \tfrac{1}{4}h^2B_{n+\frac{1}{2}}L_nB_{n+\frac{1}{2}}, \end{aligned}$$

which, after some manipulations, becomes

$$L_{n+1} = L_n + h\left[B_{n+\frac{1}{2}}, \frac{L_n + L_{n+1}}{2}\right] + \tfrac{1}{4}h^2B_{n+\frac{1}{2}}(L_{n+1} - L_n)B_{n+\frac{1}{2}}. \tag{3.32}$$

Comparing (3.32) with the standard IMR applied to (1.1),

$$L_{n+1} = L_n + h\left[B_{n+\frac{1}{2}}, \frac{L_n + L_{n+1}}{2}\right],$$

we notice an extra term in (3.32),

$$\tfrac{1}{4}h^2B_{n+\frac{1}{2}}(L_{n+1} - L_n)B_{n+\frac{1}{2}},$$

which retains the second order of the method since it is $\mathcal{O}(h^3)$, while rendering the scheme isospectral. □

The above observation is very interesting because, as we have already seen in chapter 2, the implicit midpoint rule is not isospectral, except in the 2×2 case.

3.4 Semi-explicit methods for isospectral flows

We consider again the orthogonal flow (1.2), $U' = B(L)U$, and we recall that the solution of the isospectral flow (1.1) can be recovered by the similarity transformation

$$L = UL_0U^{-1}, \tag{3.33}$$

whereby $U^{-1} = U^T$ when U is orthogonal (cf. section 1.1). Motivated by (3.33), Calvo, Iserles & Zanna (1997) proposed to solve (1.2) with an arbitrary numerical method for ODEs, for example with an explicit one. The numerical approximation obtained in this way, U_{n+1}, need not be orthogonal, however, because of (3.33),

$$L_{n+1} = U_{n+1}L_0U_{n+1}^{-1},$$

we obtain a isospectral approximation for $L(t_{n+1})$. Yet, if L_0 is symmetric, the numerical approximation L_{n+1} is not generally a symmetric matrix. In some cases, like for instance the inverse eigenvalue problem for symmetric Toeplitz matrices, or the Toda lattice (whereby the off-diagonal elements are related to the positions of the particles), losing symmetry might be a problem. On the other hand, we have the possibility of using *explicit* methods, so that the numerical integration is performed much faster.

When the loss of symmetry is at issue, Calvo et al. (1997) suggested the following technique:

i) At odd steps, solve

$$U' = B(L)U, \quad U(t_{2n}) = I, \qquad t \in [t_{2n}, t_{2n+1}],$$

in tandem with

$$L_{2n+1} = U_{2n+1}L_{2n}U_{2n+1}^{-1}.$$

ii) At even steps, solve

$$V' = -B(L)V, \quad V(t_{2n+1}) = I, \qquad t \in [t_{2n+1}, t_{2n+2}],$$

together with the similarity transformation

$$L_{2n+2} = V_{2n+2}^{-1}L_{2n}V_{2n+2}.$$

This procedure reduces the symmetry error by one unit at even steps, while adding nothing to the computational cost. More precisely, assuming that the integration is performed with constant stepsize h and with a method of order p, then at each even step,

$$\|L_{2n} - L_{2n}^T\| = \mathcal{O}\left(h^{p+1}\right),$$

while, without alternating between i) and ii), we would have had

$$\|L_{2n} - L_{2n}^T\| = \mathcal{O}(h^p).$$

The procedure can be generalized to reduce further the loss of symmetry with appropriate combinations of i) and ii). See (Iserles, McLachlan & Zanna 1997) for more details.

4

Lie groups and Lie-group methods

In the present chapter we follow a different approach for devising numerical methods for orthogonal flows. The motivation is that the set $\mathrm{O}(d)$ of $d \times d$ orthogonal matrices is a Lie group. Hence we address the more general problem of devising numerical methods that automatically stay on a Lie group whenever also the exact solution does so.

The chapter is organized as follows. In the first section we will introduce Lie groups and their general properties, referring mainly to (Bryant 1995). Other references will be stated explicitely. We also describe the general form of ordinary differential equations on a Lie group and introduce a few examples of Lie groups. In the second section, we will describe two expansions, based on the exponential function, which will be represented in terms of order trees in Section 4.3. In Section 4.4 we present the main results of this chapter, introducing the collocation methods for Lie groups and we present algebraic formulae for the coefficients of these methods. Finally in Section 4.5 we mention other classes of Lie-group methods.

4.1 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 (Olver 1993).

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,

$$\begin{aligned} R_y(x) &= x \cdot y, \\ L_y(x) &= y \cdot x, \end{aligned} \qquad x, y \in G, \tag{4.1}$$

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:

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

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

$$R'_y(\gamma) = \frac{\mathrm{d}}{\mathrm{d}t} R_y(g(t))\Big|_{t=0}, \tag{4.2}$$

where $g(t) = e + \gamma t + \mathcal{O}\left(t^2\right)$ 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

$$\exp(\gamma) = y(1), \qquad \gamma \in \mathfrak{g}, \tag{4.3}$$

where y is the integral curve of

$$y' = R'_y(\gamma), \qquad y(0) = e. \tag{4.4}$$

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

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

Then

$$[\gamma, \nu] = \frac{\partial^2}{\partial t \partial s} y(t) x(s) y^{-1}(t)\Big|_{t=s=0}, \tag{4.5}$$

defines a *commutator* or a *Lie-bracket*. The operation (4.5),

$$[\,\cdot\,,\,\cdot\,] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}, \tag{4.6}$$

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

$$[[\gamma, \nu], \eta] + [[\nu, \eta], \gamma] + [[\eta, \gamma], \nu] = 0, \qquad \forall \gamma, \nu, \eta \in \mathfrak{g}. \tag{4.7}$$

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

$$\begin{aligned} &\mathrm{ad}^0_\gamma(\nu) = \nu, \\ &\mathrm{ad}^i_\gamma(\nu) = \mathrm{ad}_\gamma \circ \mathrm{ad}^{i-1}_\gamma(\nu) = [\gamma, \mathrm{ad}^{i-1}_\gamma(\nu)], \qquad i = 1, 2, \ldots, \end{aligned} \qquad \nu \in \mathfrak{g}, \tag{4.8}$$

the *adjoint representation,* which is linear in the argument ν (Varadarajan 1984). Note that $\mathrm{ad}_\gamma(\nu) = [\gamma, \nu]$.

In the sequel we will restrict our attention to matrix Lie groups, whereby the group multiplication is simply the usual matrix multiplication, the Lie-bracket the usual commutator between matrices

$$[\gamma, \nu] = \gamma\nu - \nu\gamma, \tag{4.9}$$

and the exponential the standard matrix exponential,

$$\exp(\gamma) = \sum_{k=0}^{\infty} \frac{\gamma^k}{k!}. \tag{4.10}$$

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

4.1.1 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* (O'Neill 1966), 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

$$\begin{aligned} y' &= R'_y(\gamma(t, y)), \qquad \gamma(t, y) \in \mathfrak{g}, \\ y(0) &= y_0, \end{aligned} \tag{4.11}$$

(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 (4.11) becomes simply

$$\begin{aligned} y' &= \gamma(t, y)y, \qquad \gamma(t, y) \in \mathfrak{g}, \\ y(0) &= y_0. \end{aligned} \tag{4.12}$$

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

$$\gamma(t, y) \in \mathfrak{g} \iff y \in G, \tag{4.13}$$

and, unless otherwise specified, we will assume that (4.13) is true.

4.1.2 Some recurring examples

Example 4.1 $[(\mathbb{R}^d, +)]$ 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

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

therefore the ODE formula (4.11) 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 (4.4),

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

i.e. the exponential coincides with the identity function. ◇

Example 4.2 [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 $\mathrm{GL}(\mathbb{F}, d)$ ($\mathfrak{g}(\mathbb{F}, d)$ resp.) will inherit the same multiplication, commutator and the same exponential map. ◇

Example 4.3 [The special linear group, $(\mathrm{SL}(d), \cdot)$] The sub-group of $\mathrm{GL}(\mathbb{F}, d)$ constituted by all the matrices with determinant equal to one is called the *special linear group* and denoted by $\mathrm{SL}(\mathbb{F}, d)$,

$$\mathrm{SL}(\mathbb{F}, d) = \{y \in \mathrm{GL}(\mathbb{F}, d) \mid \det y = 1\}.$$

The corresponding Lie algebra is the set of matrices with zero trace,

$$\mathfrak{sl}(\mathbb{F}, d) = \{\gamma \in \mathfrak{gl}(\mathbb{F}, d) \mid \mathrm{tr}(\gamma) = 0\}.$$

The bracket and exponential of SL are those of GL. Furthermore, ODEs on SL have the form

$$y' = \gamma(t, y)y, \qquad \mathrm{tr}(\gamma(t, y)) = 0.$$

When γ is function of t only, flows on SL are volume-preserving: Let us consider the action[1] of SL on $\mathbb{R}^d$,

$$y' = \gamma(t)y, \qquad y, y_0 \in \mathbb{R}^d, \quad \gamma \in \mathfrak{sl}(\mathbb{F}, d).$$

Then,

$$\nabla y' = \mathrm{tr}(\gamma) = 0,$$

which is the relation that characterizes volume-preserving flows. The flow, however, is generally not volume-preserving when γ depends also on the variable y. ◇

Example 4.4 [The special orthogonal group, $(\mathrm{SO}(d), \cdot)$] The set of real $d \times d$ orthogonal matrices, constitutes a proper sub-group of $\mathrm{GL}(\mathbb{R}, d)$ and is denoted by $\mathrm{O}(d)$,

$$\mathrm{O}(d) = \{y \in \mathrm{GL}(\mathbb{R}, d) \mid yy^T = y^T y = I\}.$$

The corresponding Lie algebra is the set of all $d \times d$ skew-symmetric matrices and it is denoted by $\mathfrak{o}(d)$,

$$\mathfrak{o}(d) = \{\gamma \in \mathfrak{g}(d) \mid \gamma + \gamma^T = O\}.$$

The bracket is the usual matrix commutator and the exponential is the standard matrix exponential. Furthermore, we can consider the subgroup of $\mathrm{O}(d)$, constituted by all those orthogonal matrices with determinant equal to one. This is the connected component of $\mathrm{O}(d)$ containing the identity and is usually denoted by $\mathrm{SO}(d)$ *(special orthogonal group)*. Its Lie algebra is $\mathfrak{so}(d) = \mathfrak{o}(d)$. We remark that:

- ODEs on $\mathrm{O}(d)$ and $\mathrm{SO}(d)$ are of the form

$$y' = \gamma(t, y)y, \qquad \gamma(t, y) + \gamma(t, y)^T = O.$$

- The product of two orthogonal matrices is an orthogonal matrix;
- The commutator of two skew-symmetric matrices is a skew-symmetric matrix;

[1] See for instance (Bryant 1995, Munthe-Kaas & Zanna 1997, Varadarajan 1984) for the definition of a group action.

- The exponential of a skew-symmetric matrix is an orthogonal matrix.

Such rules will be of great relevance for the construction of orthogonal integrators. ◇

Example 4.5 [The unitary group, $(\mathrm{U}(d), \cdot)$] The set of $d \times d$ *unitary matrices,*

$$\mathrm{U}(d) = \{y \in \mathrm{GL}(\mathbb{C}, d) \mid yy^* = y^*y = I\},$$

is a Lie group and its Lie algebra is $\mathfrak{u}(d)$, the set of $d \times d$ *skew-Hermitian matrices,*

$$\mathfrak{u}(d) = \{\gamma \in \mathfrak{g}(\mathbb{C}, d) \mid \gamma + \gamma^* = O\},$$

where $\gamma^* = \bar{\gamma}^T$. The commutator and the exponential are as for $\mathrm{GL}(\mathbb{C}, d)$. Moreover, if we restrict our attention to unitary matrices with determinant equal to one, we obtain the *special unitary group* that is denoted by $\mathrm{SU}(d)$, with algebra $\mathfrak{su}(d) = \mathfrak{u}(d)$. Also in this case,

- ODEs on $\mathrm{U}(d)$ are of the form

$$y' = \gamma(t, y)y, \qquad \gamma(t, y) + \gamma(t, y)^* = O.$$

- The product of two unitary matrices is a unitary matrix;
- The commutator of two skew-Hermitian matrices is a skew-Hermitian matrix;
- The exponential of a skew-Hermitian matrix is a unitary matrix.

As for orthogonal matrices, the above rules play an important role in devising unitary integrators. ◇

Example 4.6 $[(\mathrm{N}_d, \cdot)]$ Let N_d be the set of all $d \times d$ triangular matrices with diagonal elements equal to one. This set, endowed with the usual matrix multiplication, is a Lie group. It is easy to verify that its Lie algebra, $\mathfrak{n}_d$ is the set of strictly upper triangular matrices. The commutator and the matrix exponential are as in example 4.2. Note that the formula for the exponential consists of a finite number of terms,

$$\exp \gamma = \sum_{k=0}^{d} \frac{\gamma^k}{k!},$$

since $\gamma^d = O$, having γ zero diagonal elements. In particular, the set N_3 is called the *Heisenberg group,* and its Lie algebra, $\mathfrak{n}_3$, is the set of all matrices of the form

$$\mathfrak{n}_3 = \left\{ \begin{pmatrix} 0 & \alpha & \gamma \\ 0 & 0 & \beta \\ 0 & 0 & 0 \end{pmatrix} \mid \alpha, \beta, \gamma \in \mathbb{F} \right\}.$$

In this special case, the exponential is

$$\exp \begin{pmatrix} 0 & \alpha & \gamma \\ 0 & 0 & \beta \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & \alpha & \gamma + \frac{1}{2}\alpha\beta \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{pmatrix},$$

which is evaluated exactly in a very simple fashion. ◇

Example 4.7 [The symplectic group $(\mathrm{Sp}(d), \cdot)$] The set of $2d \times 2d$ symplectic matrices, together with the usual matrix multiplication, is a Lie group and it is denoted by $\mathrm{Sp}(d)$. Recall that a matrix y is *symplectic* if

$$y^T J y = J,$$

J being the $2d \times 2d$ matrix

$$J = \begin{pmatrix} O & I \\ -I & O \end{pmatrix},$$

where O, I denote the $d \times d$ zero and identity matrix respectively. The corresponding Lie algebra $\mathfrak{sp}(d)$ is constituted by all those matrices that obey the condition

$$\gamma^T J + J\gamma = O.$$

Finally, the commutator and the exponential coincide with their classical matrix counterparts. ◇

Example 4.8 [$(G \subset \mathrm{Diff}(\mathcal{M}), \circ)$] The last example is not a matrix group, but it is a subset of the set of diffeomorphisms of a manifold $\mathcal{M}$ to itself. We consider first the linear space $\mathfrak{X}(\mathcal{M})$ of all vector fields on $\mathcal{M}$ together with the *Jacobi bracket* of vector fields, whose i'th component is given as

$$[f, g]_i = \sum_j f_j \frac{\partial g_i}{\partial x_j} - g_j \frac{\partial f_i}{\partial x_j}, \qquad f, g \in \mathfrak{X}(\mathcal{M}),$$

which is bilinear, skew-symmetric and obeys the Jacobi identity (Olver 1993). We further consider a d-dimensional subspace of $\mathfrak{X}(\mathcal{M})$, that we denote by $\mathfrak{X}_d(\mathcal{M})$, together with the Jacobi bracket, thus $(\mathfrak{X}_d(\mathcal{M}), +, [\cdot, \cdot])$ constitutes a Lie algebra. The corresponding group $G = \exp(\mathfrak{X}_d(\mathcal{M}))$, together with the usual composition of functions $\circ$, is a Lie group. The exponential function $\exp(f)$ is now the flow of the vector field f and should be evaluated according to the formula (4.3). ◇

Other relevant examples are $\mathrm{A}(\mathbb{R}, d)$, the group of affine transformations, that can be obtained as a semidirect product of $\mathrm{SL}(d) \rtimes \mathbb{R}^d$, and $\mathrm{E}(\mathbb{R}, d)$, the group of Euclidean transformations, that can be obtained as a semidirect product of $\mathrm{SO}(d) \rtimes \mathbb{R}^d$ (see (Varadarajan 1984, Marsden & Ratiu 1994) and (Munthe-Kaas & Zanna n.d.)).

4.2 Lie-group expansions and the exponential function

In the remainder of this chapter, we will consider exclusively the Lie-group matrix differential equation (4.12),

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

where we assume that the condition (4.13) is true, i.e. that $\gamma(t, y) \in \mathfrak{g}$ if and only if $y \in \mathfrak{g}$.

4.2.1 The Fer expansion

The Fer expansion was introduced by Fer (1958) to solve the linear matrix differential equation

$$y' = \gamma(t)y, \qquad y(0) = y_0. \tag{4.14}$$

The same construction was essentially obtained by Iserles (1984) (the method of *iterated commutators),* whereby a very similar approach was followed to introduce numerical methods for linear variable-coefficients systems of ODEs.

Although neither Fer nor Iserles were aware of the fact that this expansion, exploiting essentially Lie-groups and Lie-algebra operations, produces Lie-group approximation, this approach is widely used to find *analytical* (opposed to numerical) solutions of Lie-group differential equations (for instance see (Casas 1996)). We proposed in (Zanna 1996) to use Iserles' schemes for the numerical solution of Lie-group equations. The approach that we present here is however the same as in (Zanna 1997), and it is inspired by the tree analysis of Iserles & Nørsett (1997) for the Magnus expansion.

Going back to (4.14), both Fer and Iserles independently showed that the solution $y(t)$ of (4.14) can be expressed as an infinite product of exponentials,

$$y(t) = \lim_{m\to\infty} \mathrm{e}^{\sigma_0(t)} \cdots \mathrm{e}^{\sigma_m(t)} y_0, \tag{4.15}$$

where the matrix functions $\sigma_k(t)$ are obtained in an iterative manner. The first step consists of letting $\gamma_0(t) = \gamma(t)$, and evaluating

$$\sigma_0(t) = \int_0^t \gamma_0(s)\,\mathrm{d}s,$$

so that we can construct the sequence of commutators

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

Secondly, we form the matrix function

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

and evaluate

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

Similarly, once the functions $\gamma_k(t)$ and $\sigma_k(t)$ are known, we let

$$\gamma_{k+1}(t) = \sum_{i=1}^{\infty} \frac{i}{(i+1)!} \eta_{k,i}(t), \tag{4.16}$$

where

$$\eta_{k,0} = \gamma_k(t), \qquad \eta_{k,i+1} = [\eta_{k,i}, \sigma_k], \quad i \geq 0. \tag{4.17}$$

and finally set

$$\sigma_{k+1}(t) = \int_0^t \gamma_{k+1}(s)\,\mathrm{d}s. \tag{4.18}$$

Note that the *iterated commutators* in (4.17) can be conveniently written in a classical Lie-algebra notation as

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

where $\mathrm{ad}_\sigma(\,\cdot\,) : \mathfrak{g} \to \mathfrak{g}, \sigma \in \mathfrak{g}$, is the *adjoint* operator defined in (4.8).

The convergence of the series was already proved by Fer, who also showed that if we truncate (4.15) to an integer $M \geq 0$ the order of the approximation is $p = 2^{M+2} - 2$. Furthermore, Iserles showed that the order p of approximation of the truncated expansion is retained when we replace the integrals in (4.18) and the infinite series in (4.16) by finite sums and quadrature formulae of appropriate order, so that the expansion is suitable for numerical calculations.

In passing, we mention that the Fer expansion (to which we will also refer as 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 (Bryant 1995, Zanna & Munthe-Kaas 1997). The basic idea can be summarized as follows: Assume that we wish to find the fundamental solution of

$$y' = \gamma(t)y, \qquad y(0) = I, \quad y \in G, \gamma \in \mathfrak{g}.$$

We decompose such fundamental solution as the product of a known curve y_0 in G, and a residual r_0,

$$y = y_0 r_0,$$

where for instance we can choose $y_0 = \exp\left(\int_0^t \gamma(s)\,\mathrm{d}s\right) = \exp\left(\sigma_0(t)\right)$. Thus, the differential equation for the residual r_0 can be obtained by differentiation. Let dexp denote the function

$$\mathrm{dexp} : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}, \quad \mathrm{dexp}_\sigma(\sigma') = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} \mathrm{ad}^k_\sigma(\sigma') \tag{4.19}$$

(Varadarajan 1984), which is related to the *derivative* of the exponential function by the formula

$$\frac{\mathrm{d}}{\mathrm{d}t}\exp(\sigma) = \mathrm{dexp}_\sigma(\sigma')\exp(\sigma).$$

Then, the relation

$$\begin{aligned} y' &= y_0' r_0 + y_0 r_0' \\ &= \mathrm{dexp}_{\sigma_0}(\sigma_0') y_0 r_0 + y_0 r_0', \end{aligned}$$

implies

$$\gamma(t) y_0 r_0 = \mathrm{dexp}_{\sigma_0}(\sigma_0') y_0 r_0 + y_0 r_0',$$

hence

$$r_0' = \left(y_0^{-1}(\gamma(t) - \mathrm{dexp}_{\sigma_0}(\sigma_0'))y_0\right) r_0,$$

which is again a Lie-group differential equation. After further algebra, we find that

$$y_0^{-1}(\gamma(t) - \mathrm{dexp}_{\sigma_0}(\sigma_0'))y_0 = \gamma_1(t) \in \mathfrak{g},$$

from which we obtain

$$r_0' = \gamma_1(t) r_0.$$

The same decomposition is applied to r_0,

$$r_0(t) = y_1(t) r_1(t),$$

where we let

$$y_1(t) = \exp\Big(\int_0^t \gamma_1(s)\,\mathrm{d}s\Big) = \exp(\sigma_1(t)),$$

and the procedure is repeated iteratively to obtain the final expansion (4.15). In particular, the above procedure highlights that the Fer expansion, initially proposed for linear problems, holds also in the nonlinear case, when all the nonlinear terms that derive from the differentiation of γ_k are subsumed in $\mathrm{dexp}_{\sigma_k}(\sigma_k')$, for $k = 0, 1, \ldots$. It is however important to stress that in the linear case, i.e. when $\gamma = \gamma(t)$, the sequence of differential equations obeyed by the residuals are also all linear.

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_σ, 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.

Coming back to the Fer expansion, various numerical methods of different orders can be obtained by truncating the infinite series (4.16) 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 $\sigma_k(h) = \mathcal{O}\big(h^{s+1}\big)$, 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 (Iserles 1984) that

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

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 $0 \le k \le M - 1$ we evaluate i_k commutators, $\eta_{k,j}, 1 \le j \le i_k$, where i_k is the first integer such that

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

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 $\mathcal{O}\big(h^{p+1}\big)$ term.

Some of the above functions and their orders for a step with stepsize h are tabulated in Table 4.1. 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.

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

Table 4.1: First functions and their orders for the method of iterated commutators.

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, like, for instance, those based on Gaussian or Radau points (Iserles 1984, Zanna 1996). Other variants (Zanna 1996) can be obtained by replacing the product of exponentials with a single exponential, by means of the Baker–Campbell–Hausdorff formula (Varadarajan 1984). However, following Iserles and Nørsett, in the sequel we will use a single quadrature rule for all the integrals, since we expect that this approach yields a smaller number of function evaluations of $\gamma(t, y)$ together with a lesser computational cost.

4.2.2 The Magnus expansion

If the Fer expansion expresses the solution of (4.14) in terms of the product of infinite 'simple' exponentials, the Magnus expansion consists of writing the solution $y(t)$ of (4.14) as the exponential of a single and possibly more complicated argument,

$$y(t) = \mathrm{e}^{\sigma(t)} y_0. \tag{4.20}$$

Thus, we consider solutions of (4.14) of the form (4.20), and we wish to identify the relation between σ and the function $\gamma(t)$. Differentiating (4.20) with respect to time t, we have

$$y' = \mathrm{dexp}_\sigma(\sigma')\mathrm{e}^{\sigma(t)} y_0 = \mathrm{dexp}_\sigma(\sigma') y, \tag{4.21}$$

where the function dexp has been defined in (4.19).

Comparing (4.21) with (4.14), we find that σ obeys the *implicit* differential equation

$$\mathrm{dexp}_\sigma(\sigma') = \gamma(t), \tag{4.22}$$

as was shown by Hausdorff (1906). A general expansion for σ' is obtained by inverting dexp_σ,

$$\sigma' = \mathrm{dexp}_\sigma^{-1}(\gamma(t)), \tag{4.23}$$

where

$$\mathrm{dexp}_\sigma^{-1}(\gamma) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \mathrm{ad}_\sigma^k(\gamma), \tag{4.24}$$

the B_k's being Bernoulli numbers (Abramowitz & Stegun 1965, p 804). The inversion formula (4.23) seems to appear already in a paper of Baker (1905).

The expression for σ comes from the integration of (4.24), its terms being combinations of integrations and commutators of the function γ. The first terms of σ,

$$\begin{aligned}\sigma(t) \;=\; & \int_0^t \gamma(s)\,\mathrm{d}s + \frac{1}{2}\int_0^t\int_0^s [\gamma(s),\gamma(u)]\,\mathrm{d}u\,\mathrm{d}s + \frac{1}{4}\int_0^t\int_0^s\int_0^u [\gamma(s),[\gamma(u),\gamma(v)]]\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}s \\ & + \frac{1}{12}\int_0^t\int_0^s\int_0^s [[\gamma(s),\gamma(u)],\gamma(v)]\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}s + \cdots\end{aligned}$$

were derived by Magnus (1954), who integrated (4.24) using a Picard iteration, in a manner very similar to the modern technique of *waveform relaxation.*

The general formula for the expansion is much more complicated. An algorithmic expansion of σ was formulated much more recently, by Iserles & Nørsett (1997), who also provided a more precise statement of the convergence conditions[2] for the Magnus expansion. We refer the reader to (Iserles & Nørsett 1997) for the general order analysis of the terms in the Magnus expansion.

We remark that, by using the same argument as for the Fer expansion, the Magnus expansion remains valid in the nonlinear case (4.14); convergence and order analysis are just as those for the linear problem (4.14).

4.3 Order trees and exponential expansions

In this section we reconsider a tree analysis similar to the one of Iserles & Nørsett (1997) originally applied to the Magnus expansion and we show that the same tree analysis can be extended to the Fer expansion.

The point of departure for this tree analysis is that, both in the Fer and the Magnus expansions, all the relevant terms, i.e. commutators and integrals, can be expressed in terms of commutators and integrals of the matrix function $\gamma_0(t) = \gamma(t)$.

We construct bi-coloured order trees according to the following rules:

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

$$\gamma \quad \rightarrow \quad \bullet\,.$$

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

$$\int\tau \quad \rightarrow$$

3. Given two trees, τ_1 and τ_2 such that $\tau_i \neq \int \tau_j$, for $i = 1, 2$, commutation of the terms associated with τ_1 with τ_2 corresponds to joining the trees with a black vertex,

$$[\tau_1,\tau_2] \quad \rightarrow$$

[2]From Magnus' paper: "... if certain unspecified conditions of convergence are satisfied ..." (p 660) and later: "In the case where" $\gamma(t)$ "is a finite matrix with bounded elements, it can be shown by standard methods that" the expansion "leads to a function" $\sigma(t)$ "satisfying (4.14) for sufficiently small values of t" (p 661).

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

$$[\tau_1, \tau_2] \quad \rightarrow$$

We remark that, unlike in the classical Runge–Kutta order theory, these trees are planar, therefore even though two trees might be mirror image of each other, they do represent different conditions.

Definition 4.1 Let τ be an order tree, associated to a term in $\mathfrak{g}$ obtained by combination of integrals and commutators of an arbitrary analytic function $\gamma \in \mathfrak{g}$, and let the vertices of the tree τ be coloured according to the rules 1–4 above. Then the *power* of the tree τ is the number q of its white vertices. ◇

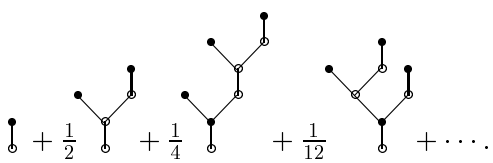
The definition of power of an order tree is given by Iserles & Nørsett (n.d.) as an iterative formula, but it has been implicitly used with a slightly different degree of depth already in a number of papers (see for instance (Iserles & Nørsett 1997, Zanna 1997)). If $\gamma \in \mathfrak{g}$, τ is a tree associated to a Lie-algebra element obtained from γ according to the rules 1–4 and the integration is performed between zero and a $0 < t \leq h$, it is possible to show that this term is an $\mathcal{O}(h^q)$, q being the power of τ: if $\gamma = \mathcal{O}(1)$, by very similar arguments to those employed in the order analysis for the Fer expansion in § 4.2.1, each integration induces a power of h, and commutation of a term with its integral induces two powers of h.

4.3.1 Order trees for the Magnus expansion

We depart from the first terms of the function σ as derived by Magnus,

$$\begin{aligned}\sigma(t) \quad = \quad & \int_0^t \gamma(s)\,\mathrm{d}s + \frac{1}{2}\int_0^t\int_0^s [\gamma(s),\gamma(u)]\,\mathrm{d}u\,\mathrm{d}s + \frac{1}{4}\int_0^t\int_0^s\int_0^u [\gamma(s),[\gamma(u),\gamma(v)]]\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}s \\ & + \frac{1}{12}\int_0^t\int_0^s\int_0^s [[\gamma(s),\gamma(u)],\gamma(v)]\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}s + \cdots\end{aligned}$$

and use the rules 1–4 above. Setting $t = h$, we obtain the following expression for σ in terms of order trees,

$$\sigma(h) =$$ 

Thus, according to Definition 4.1, the first tree has power one, the second tree has power three, the third and the fourth have power four and so on and so forth. Table 4.2 illustrates the trees and relative coefficients up to power five. For trees of higher power, we refer the reader to (Iserles & Nørsett 1997).

A *truncated* Magnus expansion is obtained by truncating the infinite series σ above to a finite number of terms. We define an order-p truncation of σ as that function $\sigma(h,p)$ that matches σ to order p, i.e.

$$\sigma(h,p) - \sigma(h) = \mathcal{O}\big(h^{p+1}\big)\,.$$

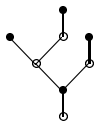
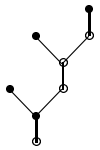
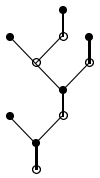
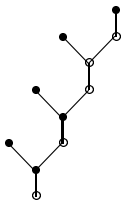
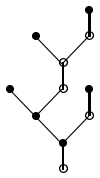
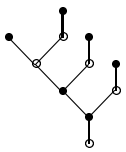
Power	Expression	Tree	Coefficient
1	$\int \gamma$		1
3	$\int [\gamma, \int \gamma]$		$\frac{1}{2}$
4	$\int [[\gamma, \int \gamma], \int \gamma]$		$\frac{1}{12}$
	$\int [\gamma, \int [\gamma, \int \gamma]]$		$\frac{1}{4}$
5	$\int [\gamma, \int [[\gamma, \int \gamma], \int \gamma]]$		$\frac{1}{24}$
	$\int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

Table 4.2: First order trees, their power, expression and coefficient for the Magnus expansion.

Hence $\sigma(h,p)$ is obtained from σ including all those terms corresponding trees with power $q \leq p$. It follows that $\exp(\sigma(h,p))y_0$ is an approximation of order (at least) p to the exact solution $y(h) = \exp(\sigma(h))y_0$.

4.3.2 Order trees for the Fer expansion

With the simple rules introduced in Section 4.3, it is easy to write the order conditions in terms of trees. We construct the σ_i's that yield approximations to the solution $y(t)$ at $t \equiv h$ of order p, by truncating the σ_i's to include all trees of power $q \leq p$. The procedure is presented in detail for $p \leq 9$, and extends in a very similar fashion to higher order approximations.

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

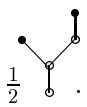
  (4.25)

  (power $q = 1$).

- Order $p = 3$. Besides the matrix σ_0, which is associated with the tree in (4.25), we need to consider the matrix function σ_1, which is appropriately truncated to order $p = 3$,

  $$\sigma_1(h,3) = \tfrac{1}{2}\int_0^h [\gamma(s), \sigma_0(s)]\,\mathrm{d}s = \tfrac{1}{2}\int_0^h \int_0^s [\gamma(s), \gamma(t)]\,\mathrm{d}t\,\mathrm{d}s.$$

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

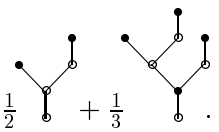
  

  (4.26)

- Order $p = 4$. Besides σ_0, we have

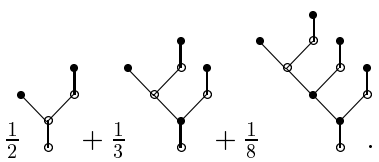
  $$\begin{aligned}\sigma_1(h) &= \tfrac{1}{2}\int_0^h [\gamma(s), \sigma_0(s)]\,\mathrm{d}s + \tfrac{1}{3}\int_0^h [[\gamma(s), \sigma_0(s)], \sigma_0(s)]\,\mathrm{d}s \\ &= \tfrac{1}{2}\int_0^h \int_0^s [\gamma(s), \gamma(t)]\,\mathrm{d}t\,\mathrm{d}s + \tfrac{1}{3}\int_0^h \int_0^s \int_0^s [[\gamma(s),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

  corresponding to the tree

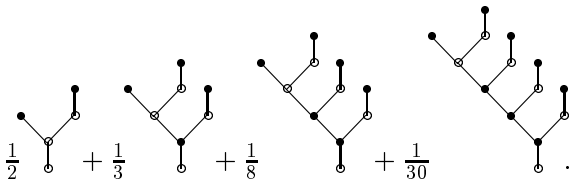
  (4.27)

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

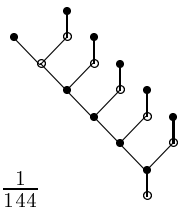
  (4.28)

- 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 $\sigma_1(h)$,



(4.29)

- 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 (4.29),

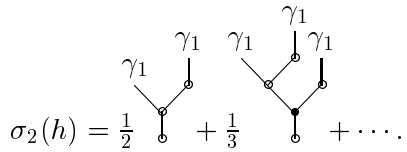


(4.30)

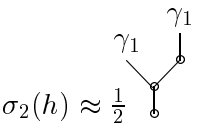
With regards to σ_2, we recall that

$$\sigma_2(h) = \tfrac{1}{2}\int_0^h [\gamma_1(s), \sigma_1(s)]\,\mathrm{d}s + \tfrac{1}{3}\int_0^h [[\gamma_1(s), \int_0^s \gamma_1(t)], \int_0^s \gamma_1(u)]\,\mathrm{d}u\,\mathrm{d}t\,\mathrm{d}s + \cdots,$$

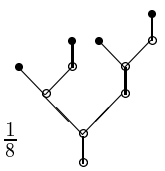
which, in terms of order trees, is represented by



For our purposes, it is sufficient to truncate σ_2 after the first tree,

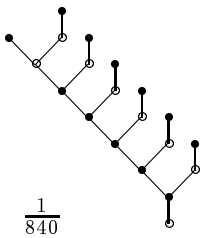


Replacing γ_1 with its corresponding tree expansion and making use of the linearity of the bracket, we find tree relevant for order seven,



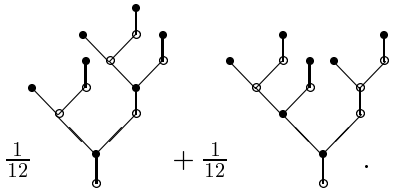
(4.31)

- Order $p = 8$. Besides σ_0 and the trees (4.29)-(4.30) for the function σ_1, we need to consider the trees



(4.32)

and add to (4.31) the trees



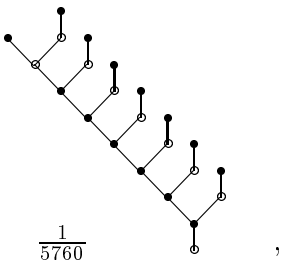
(4.33)

Although the trees (4.33) have power $q = 7$ when considered individually, their combination yields a contribution of $\mathcal{O}(h^8)$: substituting to γ its Taylor expansion, the leading term for the tree on the left is

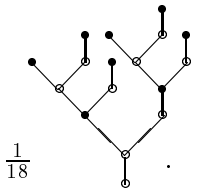
$$h^7[[\gamma', \gamma], [[\gamma', \gamma], \gamma]]$$

(where all the functions are evaluated at $t = 0$), while the leading term for the right tree is exactly the same except for a minus sign. The two $\mathcal{O}(h^7)$ contributions cancel.

- Order $p = 9$. As before, we consider σ_0, add to σ_1 the tall tree



and to σ_2



The procedure generalizes to higher order terms in a very similar fashion.

The order trees of power up to 7 for the Fer expansion are represented in Table 4.3 (we have excluded the trees (4.33) on purpose, since their combination is $\mathcal{O}(h^8)$). Note that, up to order $p = 6$, the order conditions are represented by *tall trees* only and each tree represents a separate order condition.

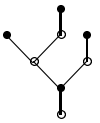
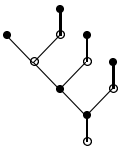
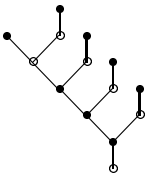
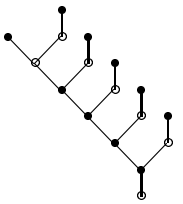
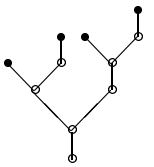
σ	Power	Expression	Tree	Coefficient
σ_0	1	$\int \gamma$		1
σ_1	3	$\int [\gamma, \int \gamma]$		$\frac{1}{2}$
	4	$\int [[\gamma, \int \gamma], \int \gamma]$		$\frac{1}{3}$
	5	$\int [[[\gamma, \int \gamma], \int \gamma], \int \gamma]$		$\frac{1}{8}$
	6	$\int [[[[\gamma, \int \gamma], \int \gamma], \int \gamma], \int \gamma]$		$\frac{1}{30}$
	7	$\int [[[[[\gamma, \int \gamma], \int \gamma], \int \gamma], \int \gamma], \int \gamma]$		$\frac{1}{144}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$
σ_2	7	$\int [[\gamma, \int \gamma], \int [\gamma, \int \gamma]]$		$\frac{1}{8}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 4.3: First order trees, their power, expression and coefficients for the Fer expansion.

4.4 Collocation methods for the Fer and Magnus expansions

4.4.1 Collocation

Let Π_m denote the set of all polynomials of degree less equal to m, and consider the differential equation

$$y' = f(t, y), \qquad y(t_0) = y_0. \tag{4.34}$$

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 $u(t) \in \Pi_\nu$ that obeys the differential equation exactly at the *collocation points*, $t_n + c_1 h, \ldots, t_n + c_\nu h$. The approximation to y_{n+1} is chosen as the value of u at $t_{n+1} = t_n + h$. In other words,

$$\begin{aligned} u(t_n) &= y_n, \\ u'(t_n + c_i h) &= f(t_n + c_i h, u(t_n + c_i h)), \quad i = 1, \ldots, \nu, \\ y_{n+1} &= u(t_n + h). \end{aligned} \tag{4.35}$$

It is well known (Hairer et al. 1993) 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

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array}, \tag{4.36}$$

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

$$\begin{aligned} a^i_j &= \int_0^{c_i} L_j(t)\,\mathrm{d}t, \\ b_i &= \int_0^1 L_i(t)\,\mathrm{d}t, \end{aligned} \qquad i, j = 1, \ldots, \nu, \tag{4.37}$$

respectively,[3] where $L_i(t)$ is the i-th 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 every 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* (Nørsett & Wanner 1981).

It is instructive to recall the proof of the theorem that links collocation and RK schemes (Hairer et al. 1993). Without loss of generality, assume that (4.34) is autonomous and suppose that we approximate its exact solution,

$$y(t_n + h) = y(t_n) + \int_{t_n}^{t_n + h} f(y(t))\,\mathrm{d}t, \tag{4.38}$$

[3] Note that the indices are positioned in a different manner, compared to the classical notation $a_{i,j}$ introduced in Chapter 2. Here, and in the sequel, we wish to highlight that a^i_j arises from the integration of $L_j(t)$ from zero to c_i.

by replacing the integral in (4.38) with a quadrature formula. Since the integration is performed with respect to the variable t, we may fix $c_1, \dots, c_\nu \in [0,1]$ and use the quadrature rule

$$\int_{t_n}^{t_n+h} f(y(t))\,\mathrm{d}t \approx h \sum_{i=1}^{\nu} b_i f_i, \tag{4.39}$$

where each f_i approximates the function value $f(y(t_n + c_i h)), i = 1, \dots, \nu$. Provided that the f_i's are available (as we assume for the time being), we can use them to construct an interpolating polynomial to the function f,

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

and the weights b_i, which have been defined in (4.37), are easily found by integrating the cardinal polynomials. On the other hand, we employ the same procedure to evaluate the Y_i's, $Y_i \approx y(t_n + c_i h)$, and let $f_i = f(Y_i)$. Note that $f_i \approx f(y(t_n + c_i h))$. Therefore,

$$Y_i = y_n + \int_{t_n}^{t_n+c_i h} \sum_{j=1}^{\nu} L_j\left(\frac{t-t_n}{h}\right) f_j\,\mathrm{d}t = y_n + h\sum_{j=1}^{\nu} a_j^i f_j, \tag{4.40}$$

where the a_j^i are the coefficients introduced in (4.37).

The theorem below gives an account of the order of the implicit RK collocation method based on $c_1, \dots, c_\nu$ (Hairer et al. 1993).

Theorem 4.1 *Let $c_1, \dots, c_\nu$ be given collocation points in the interval $[0,1]$ and denote $M(t) = \prod_{k=1}^{\nu}(t - c_k)$. Suppose that*

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

Then the collocation RK with tableau (4.36) and coefficients and weights as in (4.37) has order $p = \nu + \rho$.

We wish to extend the procedure to Lie groups as follows: consider the differential equation

$$y' = \gamma(y)y, \qquad y \in G, \quad \gamma(y) \in \mathfrak{g}, \tag{4.41}$$

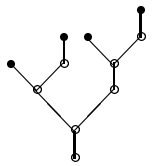
and suppose that $\gamma_i \approx \gamma(y(t_n + c_i h))$ are given. Then we approximate the function $\gamma(y)$ with its Lagrangian interpolating polynomial at the nodes $c_1, \dots, c_\nu$,

$$\gamma(y(t)) \approx \sum_{i=1}^{\nu} L_i\left(\frac{t-t_n}{h}\right)\gamma_i. \tag{4.42}$$

At this point, we insert (4.42) into our favourite expansion, say a truncated Fer or Magnus series. Such truncated expansion is hence employed to approximate the unknown values $y(t_n + c_i h)$ in the Lie group as in the case of the classical collocation. However, for Lie-group methods, we recover not just the weights and the a_j^i'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 4.3.2) and for the Magnus series (as in (Iserles & Nørsett 1997) and Section 4.3.1).

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 γ with its the Lagrangian interpolant)

$$\sum_{j,k,\ell,m=1}^{\nu} \Big(\int_0^{\{c_i,1\}} L_j(s) \int_0^s \int_0^s L_k(t) L_\ell(u) \int_0^u L_m(v)\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}t\,\mathrm{d}s \Big) [[\gamma_j,\gamma_k],[\gamma_\ell,\gamma_m]], \quad (4.43)$$

where we select c_i for the internal stage $Y_i \approx y(t_n + c_i h)$ and 1 for y_{n+1} as the upper extreme of integration in the first integral. We would like to replace (4.43) by a term of the form

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

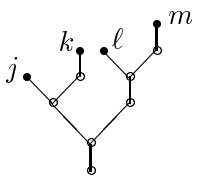
in the evaluation of the intermediate stages Y_i, or by

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

in the evaluation of y_{n+1}. Thus, a comparison with (4.43) 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 (see Figure 4.1). Note that, unlike order trees, integration trees need not be binary. Order trees are instead always binary (Iserles & Nørsett 1997).

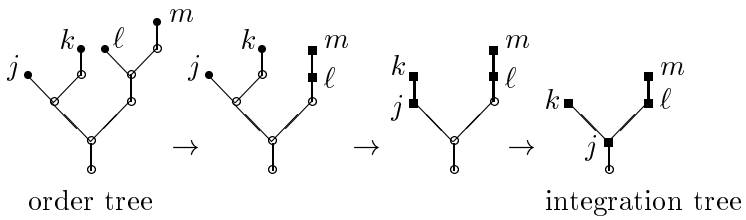
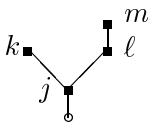


Figure 4.1: Removing the commutators from an order tree yields the corresponding integration tree.

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 tree (4.31), we have



$$a^i_{j,k,\ell,m} = \int_0^{c_i} L_j(s) \int_0^s \int_0^s L_k(t) L_\ell(u) \int_0^u L_m(v)\,\mathrm{d}v\,\mathrm{d}u\,\mathrm{d}t\,\mathrm{d}s,$$

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

Here, the main difference from the work of Iserles & Nørsett (1997) is the presence of the $a^i_{j,k,\ell,\ldots}$ coefficients, that originate from the intermediate stages Y_i, due to the nonlinearity of the problem in question. We remind that Iserles & Nørsett (1997) have addressed the case when γ is function of time only.

Definition 4.2 Let $c_1, c_2, \ldots, c_\nu$ be given collocation points. For each order tree (of power greater than one) 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 *higher-order weights.* ◇

Definition 4.3 We say that $c_1, c_2, \ldots, c_\nu \in [0,1]$ are nodes of quadrature of *order* $p = \nu + \rho$ ($p \le 2\nu$) if the polynomial $M(t) = \prod_{k=1}^{\nu}(t - c_k)$ is orthogonal to all polynomials of degree $\rho - 1$, i.e.

$$\int_0^1 M(t) t^r\,\mathrm{d}t = 0,$$

for $r = 0, 1, \ldots, \rho - 1$. ◇

Lemma 4.2 (Iserles & Nørsett (1997)) *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

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

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.

Before proceeding, it is useful to recall the Alexeev–Gröbner Lemma first, the nonlinear equivalent of the *variation of constants* formula (Nørsett & Wanner 1981).

Lemma 4.3 (Alekseev–Gröbner) *Let $y(t)$ be the solution of the differential equation (4.34) with the 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,*

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

$\Phi(t,s,w(s))$ being the result of partial differentiation of the solution of (4.34) passing through $s, w(s)$, with respect to the initial values $w(s)$.

Lemma 4.4 *Let $c_1,\ldots,c_\nu$ be zeros of quadrature of order p, let $y(t)$ be the exact solution of (4.41) with the initial condition $y(t_0) = y_0$ and let $w(t)$ be the exact solution of*

$$w' = \left(\sum_{i=1}^{\nu} L_i\left(\frac{t-t_n}{h}\right)\gamma_i\right) w(t), \qquad w(t_n) = y(t_n), \tag{4.44}$$

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

$$y(t) - w(t) = \int_{t_n}^{t} \Phi(t,s,w(s))\Big(\gamma(s,w(s))w(s) - w'(s)\Big)\,\mathrm{d}s. \tag{4.45}$$

Note that at the collocation points, the argument of the integral vanishes. Therefore, if we integrate (4.45) with the quadrature formula based on these 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 chapter.

Theorem 4.5 (The Lie-group collocation theorem) *Let $c_1, c_2, \ldots, c_\nu$ be ν collocation points, and let a^i_j and b_j, where $i, j = 1, \ldots, \nu$, be the coefficients and the weights, respectively, of the corresponding ν-stage order-p Runge–Kutta collocation scheme. Let the Fer or the Magnus expansion be truncated to include all the order trees of power $q \leq p$ for y_{n+1} and power $q \leq p-1$ 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 at least equal to p.*

Proof. The proof follows in three steps. First of all let $w(t)$ be as in Lemma 4.4, expanded in terms of the infinite Fer/Magnus series. We truncate the infinite expansion to order p, including all trees of power $q \leq p$, keeping all the integrals exact, and denote the result by $\tilde{w}(t)$. At $t_n + h$ we have only truncation error, therefore

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

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 4.2,

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

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

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

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

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

Note that, subject to the assumption of Theorem 4.5, if the function $\gamma(t, y)$ in (4.12) depends on t only then the collocation technique applied to the Magnus expansion produces the methods of Iserles & Nørsett (1997).

Corollary 4.5.1 *Subject to the assumption of Theorem 4.5, if $c_1 = 0$, then the higher order coefficients $a^1_{j,k,\ell,\ldots}$ equal to zero, for $j, k, \ell = 1, 2, \ldots$.*

The Corollary 4.5.1 generalizes what happens to the RK matrix A in the classical setting, since when $c_1 = 0$ then the first row of th matrix A consists of zero entries.

The result of Theorem 4.5 is illustrated in the following examples.

Example 4.9 We consider the collocation points $c_1 = 0, c_2 = \frac{1}{2}, c_3 = 1$, which correspond to *Lobatto* quadrature points. In the classical RK case, these points correspond to the

order-four IRK known as *Ehle III_A* (Hairer et al. 1993), with the RK tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & \frac{-1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} . \tag{4.46}$$

We choose the Fer expansion and truncate it to order four, by including all those trees with power $q \leq p$. In other words, we need to consider the trees (4.25) and (4.27). The tree (4.25) gives rise to the same weights as in (4.46), while the order trees in (4.27) give rise to new higher-order coefficients and weights, according to their respective integration tree. Suppose that $y_n \in G$ and a stepsize h are given. The scheme becomes

$$\begin{aligned} Y_i &= \exp\Big(h \sum_{j=1}^{3} a_j^i \gamma_j\Big) \exp\Big(\frac{h^2}{2} \sum_{j,k=1}^{3} a_{j,k}^i [\gamma_j, \gamma_k] \\ &\qquad + \frac{h^3}{3} \sum_{j,k,\ell=1}^{3} a_{j,k,\ell}^i [[\gamma_j, \gamma_k], \gamma_\ell]\Big) y_n, \qquad i = 1, \ldots, 3, \\ \gamma_i &= \gamma(Y_i), \\ w_{n+1} &= \exp\Big(h \sum_{j=1}^{3} b_j \gamma_j\Big) \exp\Big(\frac{h^2}{2} \sum_{j,k=1}^{3} b_{j,k} [\gamma_j, \gamma_k] \\ &\qquad + \frac{h^3}{3} \sum_{j,k,\ell=1}^{3} b_{j,k,\ell} [[\gamma_j, \gamma_k], \gamma_\ell]\Big) y_n. \end{aligned} \tag{4.47}$$

However, for a scheme order four, it is sufficient to evaluate the intermediate stages Y_i to order three only. This can be accomplished by disregarding the power-four tree. Therefore (4.47) reduces to

$$\begin{aligned} Y_i &= \exp\Big(h \sum_{j=1}^{3} a_j^i \gamma_j\Big) \exp\Big(\frac{h^2}{2} \sum_{j,k=1}^{3} a_{j,k}^i [\gamma_j, \gamma_k]\Big) y_n, \qquad i = 1, \ldots, 3, \\ \gamma_i &= \gamma(Y_i), \\ y_{n+1} &= \exp\Big(h \sum_{j=1}^{3} b_j \gamma_j\Big) \exp\Big(\frac{h^2}{2} \sum_{j,k=1}^{3} b_{j,k} [\gamma_j, \gamma_k] \\ &\qquad + \frac{h^3}{3} \sum_{j,k,\ell=1}^{3} b_{j,k,\ell} [[\gamma_j, \gamma_k], \gamma_\ell]\Big) y_n. \end{aligned} \tag{4.48}$$

Here, the coefficients a_j^i and b_j are the same as in the corresponding collocation RK in

(4.46), while the higher-order coefficients and weights are evaluated as follows,

$$\begin{aligned} a^i_{j,k} &= \int_0^{c_i} L_j(t) \int_0^t L_k(s)\,\mathrm{d}s\,\mathrm{d}t, \qquad i,j,k = 1,\ldots,3, \\ b_{j,k} &= \int_0^1 L_j(t) \int_0^t L_k(s)\,\mathrm{d}s\,\mathrm{d}t, \qquad j,k = 1,\ldots,3, \\ b_{j,k,\ell} &= \int_0^1 L_j(t) \int_0^t \int_0^t L_k(s) L_\ell(u)\,\mathrm{d}u\,\mathrm{d}s\,\mathrm{d}t, \qquad j,k,\ell = 1,\ldots,3. \end{aligned} \tag{4.49}$$

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

$$b_{i,j} = a^3_{i,j}, \quad i,j = 1,\ldots,\nu,$$

(since $c_3 = 1$) and

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

A note on the implementation of (4.48). 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,

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

Hence, the diagonal higher-order coefficients and weights need not be calculated and there is no need for those high-order coefficients and weights whose first two lower indices are equal. ◇

Example 4.10 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

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} . \tag{4.50}$$

This time, we consider the Magnus expansion truncated to order four by including all trees of power $q \leq 4$. 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 (4.49), 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 of order four. We find

$$a^1 = \begin{pmatrix} \frac{1}{32} & \frac{19}{288} - \frac{\sqrt{3}}{24} \\ \frac{-1}{288} & \frac{7}{96} - \frac{\sqrt{3}}{24} \end{pmatrix}, \quad a^2 = \begin{pmatrix} \frac{7}{96} + \frac{\sqrt{3}}{24} & \frac{-1}{288} \\ \frac{19}{288} + \frac{\sqrt{3}}{24} & \frac{1}{32} \end{pmatrix}$$

and

$$\{b_{j,k}\}_{j,k=1}^2 = \begin{pmatrix} \frac{1}{8} & \frac{1}{8} - \frac{\sqrt{3}}{12} \\ \frac{1}{8} + \frac{\sqrt{3}}{12} & \frac{1}{8} \end{pmatrix},$$

together with

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

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:

$$\begin{aligned} Y_1 &= \exp\left(h(a^1_1\gamma_1 + a^1_2\gamma_2) + \tfrac{h^2}{2}(a^1_{1,2} - a^1_{2,1})[\gamma_1,\gamma_2]\right) y_n, \\ \gamma_1 &= \gamma(t_n + c_1 h, Y_1), \\ Y_2 &= \exp\left(h(a^2_1\gamma_1 + a^2_2\gamma_2) + \tfrac{h^2}{2}(a^2_{1,2} - a^2_{2,1})[\gamma_1,\gamma_2]\right) y_n, \\ \gamma_2 &= \gamma(t_n + c_2 h, Y_2), \end{aligned} \tag{4.51}$$

$$\begin{aligned} y_{n+1} = \exp\Big(& h(b_1\gamma_1 + b_2\gamma_2) + \tfrac{h^2}{2}(b_{1,2} - b_{2,1})[\gamma_1,\gamma_2], \\ & + h^3[[\gamma_1,\gamma_2], (\tfrac{1}{12}(b^{(1)}_{1,2,1} - b^{(1)}_{2,1,1}) - \tfrac{1}{4}(b^{(2)}_{1,1,2} - b^{(2)}_{1,2,1}))\gamma_1 \\ & + (\tfrac{1}{12}(b^{(1)}_{1,2,2} - b^{(1)}_{2,1,2}) - \tfrac{1}{4}(b^{(2)}_{2,1,2} - b^{(2)}_{2,2,1}))\gamma_2] \Big) y_n. \end{aligned}$$

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

Motivated by the results for the Magnus series in a linear setting reported in (Rasmussen 1997), we see that

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

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 may be subsumed in the local truncation term. Thus, the approximation y_{n+1} can be reduced

$$\begin{aligned} y_{n+1} &= \exp\Big(h(b_1\gamma_1+b_2\gamma_2)+\tfrac{h^2}{2}(b_{1,2}-b_{2,1})[\gamma_1,\gamma_2]\Big)y_n \\ &= \exp\Big(\tfrac{h}{2}(\gamma_1+\gamma_2)-\tfrac{\sqrt{3}}{12}h^2[\gamma_1,\gamma_2]\Big)y_n, \end{aligned} \tag{4.52}$$

which, together with the stages in (4.51), gives a fourth-order scheme for the nonlinear problem (4.12). ◇

Once we have chosen our favourite expansion (Fer or Magnus), coefficients and weights depend only on collocation points and the order trees, most of which are common to both expansions. In particular, 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, the group multiplication rule being the '+' (translation) (Munthe-Kaas & Zanna 1997). In that case, since the group is commutative, all the commutator terms disappear and we are left with the terms that require only the coefficients in the classical RK tableau.

4.4.2 Algebraic formulae for the coefficients and the weights

Recall that in the interval $[t_n,t_n+h]$ the interpolating polynomial (4.42) can be written in the form

$$\gamma(t,y)\approx\sum_{j=1}^{\nu}L_j\left(\frac{t-t_n}{h}\right)\gamma_j=\sum_{j=1}^{\nu}\left(\frac{t-t_n}{h}\right)^{j-1}\alpha_j, \tag{4.53}$$

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

$$\boldsymbol{\alpha}=V^{-1}\boldsymbol{\gamma}, \tag{4.54}$$

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

$$V=\begin{pmatrix} 1 & c_1 & \dots & c_1^{\nu-1} \\ 1 & c_2 & \dots & c_2^{\nu-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_\nu & \dots & c_\nu^{\nu-1} \end{pmatrix}, \tag{4.55}$$

(Nørsett & Wanner 1981).

The interpolating polynomial (4.53) 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

$$\begin{aligned} y(t) &= \exp\Big(\int_{t_n}^{t} \sum_{j=1}^{\nu} \left(\frac{s-t_n}{h}\right)^{j-1} \alpha_j \, \mathrm{d}s + \cdots\Big) y_n \\ &= \exp\Big(h \sum_{j=1}^{\nu} \left(\frac{t-t_n}{h}\right)^{j} \frac{\alpha_j}{j} + \cdots\Big) y_n, \qquad t_n \le t \le t_n + h. \end{aligned} \tag{4.56}$$

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 as mentioned above. We write such matrix in the form

$$A = CVJV^{-1}, \tag{4.57}$$

where $C = \mathrm{diag}(c_1, c_2, \ldots, c_\nu)$, V is the Vandermonde matrix (4.55) and

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{2} & 0 & \ddots & \vdots \\ 0 & 0 & \frac{1}{3} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{\nu} \end{pmatrix}. \tag{4.58}$$

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

$$\boldsymbol{b}^T = (1, 1, \ldots, 1) J V^{-1}, \tag{4.59}$$

where J is given in (4.58) and V in (4.55). Note that the matrix CVJ in (4.57) and the vector $(1, 1, \ldots, 1)J$ in (4.59) come from the power-one tree, when we integrate the polynomials $\left(\frac{t-t_n}{h}\right)^{j-1}$ for $j = 1, \ldots, \nu$. 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, for the power-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 power three. We have

$$a(\theta) = V^{-T} THJ T V^{-1}, \qquad V^T a(\theta) V = THJT, \tag{4.60}$$

where T is a diagonal matrix and H is a *Hilbert* matrix,

$$T = \begin{pmatrix} \theta & 0 & & & 0 \\ 0 & \theta^2 & 0 & & \\ & 0 & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ 0 & & & 0 & \theta^\nu \end{pmatrix}, \qquad H = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{\nu+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \ddots & \frac{1}{\nu+2} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{2\nu-1} \\ \frac{1}{\nu+1} & \frac{1}{\nu+2} & \cdots & \frac{1}{2\nu-1} & \frac{1}{2\nu} \end{pmatrix}, \qquad (4.61)$$

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

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

Definition 4.4 We say that an expansion of the form

$$\begin{aligned} Y_i &= \exp\Big(h \sum_{j=1}^{\nu} a_j^i \gamma_j + \frac{h^2}{2} \sum_{j,k=1}^{\nu} a_{j,k}^i [\gamma_j, \gamma_k] + \cdots\Big) \cdots \\ &\qquad \exp\Big(\frac{h^q}{r_q} \sum_{i_1,\ldots,i_{q-1}=1}^{\nu} a_{i_1,\ldots,i_{q-1}}^i [\cdots[\gamma_{i_1}, \gamma_{i_2}]\cdots, \gamma_{i_{q-1}}]\Big) \cdots y_n, \qquad i = 1, \ldots, \nu, \\ \gamma_i &= \gamma(Y_i), \end{aligned}$$

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

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

4.4.3 Relaxed collocation

The schemes that we have presented above are *implicit,* as are classical RK collocation schemes, since the methods inherit the same RK coefficient matrix A from the power-one tree. However, we wish to generate (if possible) *explicit* methods that are suitable for the solution of the nonlinear problem (4.12). We do this in two steps, first relaxing the collocation condition and then introducing explicit order-three schemes in Section 4.4.4. Our point of departure is the observation that collocation is essentially equivalent to an interpolation problem and to the inversion of the Vandermonde matrix (4.58). However,

from the collocation Theorem 4.5, it is clear that the requirement of order p for the internal stages is superfluous, since an order-$(p-1)$ approximation does just 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 $\rho - 1$,

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

Then, by virtue of Theorem 4.5, the corresponding Lie-group collocation method possesses order $p = \nu + \rho$.

Here, instead, 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 let

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

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

We impose 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 4.11 Consider the collocation points $c_1 = 0, c_2 = 1/2, c_3 = 1$, whose corresponding collocation method has been presented in Example 4.9. Using relaxed collocation, we find

$$A = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{8} + \frac{1}{4}k_2 & \frac{1}{2}(1 - k_2) & -\frac{1}{8} + \frac{1}{4}k_2 \\ -\frac{1}{2} + 2k_3 & 2(1 - 2k_3) & -\frac{1}{2} + 2k_3 \end{pmatrix}, \tag{4.62}$$

from which, substituting $k_2 = k_3 = \frac{1}{2}$ in (4.62), we obtain the tableau

$$A = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}, \tag{4.63}$$

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

$$A = \begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{4.64}$$

However, 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 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 4.9. The numerical method corresponding to (4.64) for the Fer expansion is:

$$\begin{aligned} Y_1 &= y_n, & \gamma_1 &= \gamma(t_n, Y_1), \\ Y_2 &= \exp\left(\tfrac{h}{4}(\gamma_1 + \gamma_2)\right) y_n, & \gamma_2 &= \gamma(t_n + \tfrac{1}{2}h, Y_2), \\ Y_3 &= \exp(h\gamma_2) y_n, & \gamma_3 &= \gamma(t_n + h, Y_3), \end{aligned} \tag{4.65}$$

$$\begin{aligned} y_n &= \exp\left(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3)\right) \exp\left(\tfrac{h^2}{2}(-\tfrac{2}{15}[\gamma_1 - \gamma_3, \gamma_2] - \tfrac{1}{30}[\gamma_1, \gamma_3])\right), \\ &= \exp\left(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3)\right) \exp\left(-\tfrac{h^2}{2}([\gamma_1 - \gamma_3, \tfrac{2}{15}\gamma_2 + \tfrac{1}{30}\gamma_3]\right). \end{aligned}$$

Similar procedure applies to the coefficients (4.63). ◇

4.4.4 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). Instead, 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 4.6 *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*

$$\sum_{i,j=1}^{\nu} b_{i,j}(c_i - c_j) = \frac{1}{6}, \tag{4.66}$$

then the explicit scheme

$$\begin{aligned} Y_i &= \exp\Big(h\sum_{i=1}^{\nu} a^i_j \gamma_j\Big) y_n, \qquad i = 1, \ldots, \nu, \\ \gamma_i &= \gamma(t_n + c_i h, Y_i), \\ y_{n+1} &= \exp\Big(h\sum_{i=1}^{\nu} b_i\gamma_i + \frac{h^2}{2}\sum_{i=1}^{\nu} b_{i,j}[\gamma_i, \gamma_j]\Big) y_n, \end{aligned} \tag{4.67}$$

based on the Magnus expansion, and the explicit scheme

$$\begin{aligned} Y_i &= \exp\Big(h\sum_{i=1}^{\nu} a^i_j \gamma_j\Big) y_n, \qquad i = 1, \ldots, \nu, \\ \gamma_i &= \gamma(t_n + c_i h, Y_i), \\ y_{n+1} &= \exp\Big(h\sum_{i=1}^{\nu} b_i\gamma_i\Big) \exp\Big(\frac{h^2}{2}\sum_{i,j=1}^{\nu} b_{i,j}[\gamma_i, \gamma_j]\Big) y_n, \end{aligned} \tag{4.68}$$

based on the Fer expansion, possess order three.

Proof. We consider one step of the Magnus scheme (4.67) 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

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

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

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

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

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,

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

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

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

Taking into account the classical conditions for order three,

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

(Hairer et al. 1993), after some simple algebraic manipulation we obtain the formula

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

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

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

both of which reduce to

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

after summation over the index k.

Similarly, the result follows for the Fer expansion, taking into account the first terms of the BCH formula (Varadarajan 1984). □

Corollary 4.6.1 *Let the c_i's be distinct nodes and let the order-three weights $b_{i,j}$ be evaluated as*

$$b_{i,j} = \int_0^1 \int_0^t L_i(t)L_j(s)\,\mathrm{d}s\,\mathrm{d}t, \qquad i,j = 1,\ldots,\nu, \tag{4.69}$$

or in terms of the formulae (4.60) *with $\theta = 1$. Then, the Lie-group methods* (4.67) *and* (4.68) *have order three.*

Proof. By virtue of Theorem 4.6, it is sufficient to show that the order-three weighs (4.69) obey the condition (4.66). We have

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

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,

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

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

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

(Hairer et al. 1993). Because the nodes are all distinct, we can use the order-three weights evaluated by means of (4.69). Then the following Lie-group method based on the Fer expansion has order three:

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

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

Similarly, for the Magnus expansion, one has

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

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

and Corollary 4.6.1 ensures that also such scheme possesses order three. ◇

4.5 Other classes of Lie-group methods

In this section we list a few of alternative methods to those introduced above.

4.5.1 The Runge–Kutta type Lie-group methods of Crouch and Grossman

The algorithms presented by Crouch & Grossman (1993) can be used not only in the context of Lie-group differential equations but also in the more general case when the solution (defined in some Euclidean space) of the given differential equation naturally evolves in some embedded submanifold. Thus, we shall consider the general form of the methods and specify them in the context of matrix Lie groups.

Let $\mathcal{M}$ be a submanifold of the Euclidean space $\mathbb{R}^N$ (or, in the case of matrices, $\mathbb{R}^{d\times d}$) and let $\mathfrak{X}(\mathcal{M})$ denote the set of vector fields on $\mathcal{M}$. We consider the differential equation

$$y' = F(t, y), \qquad y \in \mathbb{R}^N, \tag{4.70}$$

and assume that, provided $y_0 \in \mathcal{M}$, it is true that $y(t) \in \mathcal{M}$ for all $t \geq 0$. Assume further that at each point $q \in \mathcal{M}$ there exists a frame $\{E_1, \ldots, E_M\} \in \mathfrak{X}(\mathcal{M})$, so that

$$F(t, q) = \sum_{i=1}^{M} f_i(t, q) E_i(q),$$

where $f_i : \mathbb{R} \times \mathbb{R}^N \to \mathbb{R}$ are analytic functions. Denote by $F_q(z)$ the vector field obtained by fixing the f_i's at q (fixing the coefficients of the frame at the point q),

$$F_q(z) = \sum_{i=1}^{M} f_i(q) E_i(z), \tag{4.71}$$

where, without loss of generality, we have assumed that (4.70) is autonomous. Then, if $y_n \in \mathcal{M}$ is given, and h is the stepsize of integration, an explicit ν-stage Crouch–Grossman (CG) method is given by

$$\begin{aligned}
Y_1 &= y_n, \\
F_{Y_1} &= \textstyle\sum_{i=1}^{M} f_i(y_n) E_i, \\
Y_j &= \exp(h a_{j,j-1} F_{Y_{j-1}}) \circ \cdots \circ \exp(h a_{j,1} F_{Y_1})(y_n), \qquad j = 2, \ldots, \nu, \\
F_{Y_j} &= \textstyle\sum_{i=1}^{M} f_i(Y_j) E_i, \\
y_{n+1} &= \exp(h b_\nu F_{Y_\nu}) \circ \cdots \circ \exp(h b_1 F_{Y_1})(y_n),
\end{aligned} \tag{4.72}$$

where $\exp(F_{Y_j})(q)$ denotes the flow of the vector field F_{Y_j} evaluated at the point q.

In the special case when $\mathcal{M} \equiv G$ is a matrix Lie group, then frames exist globally: for instance, if $\{\eta_1, \ldots \eta_M\}$ is a basis of $\mathfrak{g}$, then

$$E_i(y) = \eta_i y$$

is a global frame. In that case, the ODE (4.70) reduces to

$$y' = \sum_{i=1}^{M} f_i(y) \eta_i y. \tag{4.73}$$

Moreover,

$$\exp\big(\sum_{i=1}^{M} \alpha_i E_i\big)(z) = \exp\big(\sum_{i=1}^{M} \alpha_i \eta_i\big) z,$$

and the methods (4.72) become

$$\begin{aligned}
Y_1 &= y_n, \\
F_{Y_1} &= \textstyle\sum_{i=1}^{M} f_i(y_n) \eta_i, \\
Y_j &= \exp(h a_{j,j-1} F_{Y_{j-1}}) \cdots \exp(h a_{j,1} F_{Y_1}) \cdot y_n, \qquad j = 2, \ldots, \nu, \\
F_{Y_j} &= \textstyle\sum_{i=1}^{M} f_i(Y_j) \eta_i, \\
y_{n+1} &= \exp(h b_\nu F_{Y_\nu}) \cdots \exp(h b_1 F_{Y_1})(y_n).
\end{aligned} \tag{4.74}$$

where the compositions of exponentials in (4.72),(4.74) have been replaced by the group multiplication. Similarly, the procedure can be extended to include implicit methods.

Here, the $a_{i,j}$'s and the b_j's *cannot* be simply extrapolated from the RK tableau of a method of order p (in the classical context) to obtain a method of order p in the Lie-group context. This procedure yields just order two for the scheme (4.72),(4.74). Instead, the $a_{i,j}$'s and the b_j's need to be evaluated so that the Taylor expansion of the exact solution and the expansion of the numerical approximation match to order p,

$$y_{n+1} - y(t_n + h) = \mathcal{O}\left(h^{p+1}\right).$$

The order analysis for the scheme (4.72),(4.74) is considerably more complicated than the classical one for RK methods, and it has been carried out in the general case by Owren & Marthinsen (1997).

The main disadvantage of the CG schemes formulated as in (4.72),(4.74), is that they require a large number of exponentials. For instance, for an explicit ν-stage method, we need to evaluate $\frac{1}{2}\nu(\nu+1)$ exponentials, which is usually more than the number of exponentials required for truncated Fer/Magnus expansions, as well as for the methods of Munthe-Kaas, that we will introduce below in Section 4.5.2. However, such a number of exponential can be reduced by means of the BCH formula, although this will require a certain unclear number of commutators.

On the other hand, CG methods need only to compute flows of elements in the span of $\{E_1, \ldots, E_M\}$, as opposed to the Lie algebra generated by $E_1, \ldots, E_M$.[4] This might be exploited in the case when the exponential is evaluated as the product of simpler exponentials (Celledoni & Iserles 1998), in which case the frame could be chosen to take into account such details.

4.5.2 The Runge–Kutta-type Lie-group methods of Munthe-Kaas

We consider the Lie-group differential equation (4.12),

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

and look for a solution of the form

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

Differentiating in the same fashion as in Section 4.2.2, we find that σ obeys the differential equation

$$\sigma' = \mathrm{dexp}_\sigma^{-1}(\gamma) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \mathrm{ad}_\sigma^k(\gamma) \qquad \sigma(0) = O, \tag{4.75}$$

where the B_k's and the adjoint map are as in section 4.2.2. The idea at the basis of the methods of Munthe-Kaas (1997) consists of solving the differential equation (4.75) with an arbitrary RK scheme (truncating dexp^{-1} to appropriate order) so that, once σ_{n+1} is known, one can approximate

$$y_{n+1} \approx \mathrm{e}^{\sigma_{n+1}} y_0.$$

[4] The algebra generated by $E_1, \ldots, E_M$ and $\mathrm{span}\{\mathrm{E}_1, \ldots, \mathrm{E}_\mathrm{M}\}$ coincide if $E_1, \ldots, E_M$ are in involution.

In practice, assuming that $y_n \in G$ is given, the ODE (4.75) is solved for $t \in [t_n, t_{n+1}]$, with initial condition $\sigma(t_n) = O$, and then $y(t_{n+1})$ is approximated by

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

The difference between this approach and the Magnus expansion in §4.2.2 is as follows: in the latter, we first integrate (4.75) for σ and then discretize the integrals to obtain the numerical methods. Here, a numerical ODE method is applied directly on (4.75). Furthermore, since (4.75) is a differential equation in $\mathfrak{g}$, a vector space, classical RK order analysis holds. Thus, a classical order-p RK scheme will approximate (4.75) to the same order.

Let us describe the methods of Munthe-Kaas in more detail. Consider a ν-stage RK method described by the tableau

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

and define the truncation of dexp^{-1} to order p as

$$\mathrm{dexp}_\sigma^{-1}(\gamma, p) = \sum_{k=0}^{p-2} \frac{B_k}{k!} \mathrm{ad}_\sigma^k(\gamma).$$

Hence,

$$\begin{aligned} \sigma_i &= h \sum_{j=1}^{\nu} a_{i,j} k_j, \\ k_i &= \mathrm{dexp}_{\sigma_i}^{-1}(\gamma_i, p), \qquad\qquad i = 1, \ldots, \nu, \\ \gamma_i &= \gamma(t_n + c_i h, \exp(\sigma_i) y_n), \\ \sigma_{n+1} &= h \sum_{i=1}^{\nu} b_i k_i, \\ y_{n+1} &= \exp(\sigma_{n+1}) y_n \end{aligned} \tag{4.76}$$

(Munthe-Kaas 1997). In particular, if the underlying RK is explicit, then also the method (4.76) is explicit.

Example 4.13 Consider the RK tableau

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

which defines a three-stage, explicit, order-three method. Assume that $y_n \in G$ is known. Then

$$\begin{aligned} &\sigma_1 = O, && \gamma_1 = \gamma(t_n, y_n), && k_1 = \gamma_1, \\ &\sigma_2 = \tfrac{h}{2} k_1, && \gamma_2 = \gamma(t_n + \tfrac{h}{2}, \mathrm{e}^{\sigma_2} y_n), && k_2 = \gamma_2 + \tfrac{1}{2}[\sigma_2, \gamma_2], \\ &\sigma_3 = h(-k_1 + 2k_2), && \gamma_2 = \gamma(t_n + h, \mathrm{e}^{\sigma_3} y_n), && k_3 = \gamma_3 + \tfrac{1}{2}[\sigma_3, \gamma_3], \\ &\multicolumn{3}{c}{\sigma_{n+1} = h\Big(\tfrac{1}{6} k_1 + \tfrac{2}{3} k_2 + \tfrac{1}{6} k_3\Big), \quad y_{n+1} = \exp(\sigma_{n+1}) y_n,} \end{aligned}$$

is the corresponding MK method, which is also three-stage, explicit and possesses order three. ◇

The methods of Munthe-Kaas can also be generalized to homogeneous spaces and to manifolds for which it is possible to find a group or an algebra action. See (Munthe-Kaas 1997, Munthe-Kaas & Zanna 1997) for more details.

Comments

The list of methods that we have presented in this section is not complete by any means but intended to give a flavour on the new techniques for solving ordinary differential equations with special features. Lie-group methods have many applications in mechanics, since SO(3) represents rigid rotations. Indeed, the methods of Crouch and Grossman emerged in this context. We mention the existence of Lie-group methods devised by Lewis & Simo (1995), which are midpoint-like symplectic schemes of order two.

The approach presented in this chapter allows us to generate (in principle) Lie-group methods of arbitrarily high order. The main obstacle is the increasingly high number of trees featuring in the expansions, although some terms, especially those corresponding to 'bushy' trees, often possess order higher than needed. Current studies are under way to identify the minimal number of trees to include in the expansion for a desired order (Munthe-Kaas & Owren 1998).

Lie-group methods are based on two operations, commutation and exponentiation, and, throughout the chapter, we have tacitly assumed we could evaluate commutators and exponentials to machine accuracy. Evaluating a commutator is not usually a demanding problem (in most applications we consider matrix Lie groups so that it suffices to evaluate matrix-matrix products), but the evaluation of the matrix exponential is usually a much difficult and often controversial task, which has inspired a number of papers in the subject (Moler & van Loan 1978).

When solving ODEs, exponentials need not to be evaluated to machine accuracy, because the argument is already affected by the truncation error of the ODE method. Thus, a numerical approximation which possesses the same order as the numerical method usually suffices, as long as this approximation is in the desired Lie group. In some special cases, like when $G = \mathrm{SO}(d), \mathrm{U}(d)$, there exists maps form $\mathfrak{g} \to G$ that can be used to approximate the exponential function. Some of these maps are the *Cayley transform,*

$$\gamma \in \mathfrak{so}(d), \mathfrak{u}(d), \to (I+\gamma)(I-\gamma)^{-1} \in \mathrm{SO}(d), \mathrm{U}(d),$$

and the *diagonal Padé approximant*,

$$\gamma \in \mathfrak{so}(d), \mathfrak{u}(d) \to P_{i,i}(\gamma) \in \mathrm{SO}(d), \mathrm{U}(d), \quad i = 1, 2, \ldots,$$

where

$$P_{1,1}(z) = \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}, \qquad P_{2,2}(z) = \frac{1+\frac{1}{2}z+\frac{1}{12}z^2}{1-\frac{1}{2}z+\frac{1}{12}z^2}, \qquad \ldots,$$

(Baker 1975, Table 1.1 p 11) which is a product of Cayley transforms. The rational function $P_{i,i}$ approximates the exponential to order $2i$, so that $P_{1,1}$ is of order two and so on and so forth. Moreover, in the 3×3 case, there exists the Euler–Rodriguez formula (Lewis & Simo 1995) that allows to evaluate the exponential to machine accuracy.

Unfortunately, in other cases, like $\mathrm{SL}(d)$ and $\mathrm{SU}(d)$, these approximants are not satisfactory if our purpose is to remain on the group: a theorem (Feng & Shang 1995) shows that the only approximation to the exponential function which is analytic at the origin and has derivative equal to one is the exponential itself. However, in these cases, one might approximate the exponential by means of products of exponentials of simpler arguments. At the moment, such alternative techniques are being considered by Celledoni & Iserles (1998). The numerical experiments in Chapter 5 rely mostly on the MATLAB™ routine `expm`, which computer the exponential using a scaling and squaring algorithm with a Padé approximation (*MATLAB User's Guide* 1992).

5

Numerical examples

In this chapter we present a number of numerical experiments to illustrate the use of isospectral methods for solving equations of the type (1.1). Comparison of Lie-group methods is a difficult task. Due to the novel nature of the Lie-group integrators, in general there exists a number of parameters that can be tuned to reduce the cost of implementation, and we shall postpone such task to a latter date. Thus, we do not test all the methods proposed, just some order-three and order-four schemes, together with their classical counterparts. The general procedure that we follow is explained below. We find a numerical approximation to the isospectral flow (1.1),

$$L' = [B(L), L], \qquad L(0) = L_0,$$

by solving numerically the orthogonal flow

$$U' = B(UL_nU^T)U, \qquad U(t_n) = I,$$

with either an orthogonal method *á la* Chapter 3, or with a Lie-group method, *á la* Chapter 4. Thus, if $L_n \approx L(t_n)$ is given and $U_{n+1} \approx U(t_{n+1})$ is an orthogonal matrix, we employ the similarity transformation

$$L_{n+1} = U_{n+1}L_nU_{n+1}^T,$$

to advance from L_n to $L_{n+1} \approx L(t_{n+1})$. In the case when the numerical procedure differs from the one described above, this will be explicitely stated.

5.1 Toda lattices

Our first application is a Toda lattice of d particles with equal mass $m = 1$, whose motion is described in terms of the Hamiltonian equations

$$\begin{aligned} q_j' &= p_j, \\ p_j' &= \mathrm{e}^{-(q_j - q_{j-1})} - \mathrm{e}^{-(q_{j+1} - q_j)}, \end{aligned} \qquad j = 1, 2, \ldots, d,$$

where we assume

$$q_0 = -\infty, \quad q_{d+1} = \infty,$$

for the nonperiodic lattice, while letting

$$q_{j+d} = q_j, \qquad j = 1, 2, \ldots, d,$$

in the equations for the periodic lattice. The Hamiltonian function of the underlying system is

$$H(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{2}\sum_{j=1}^{d} p_j^2 + \sum_{j=1}^{d}(\mathrm{e}^{-(q_{j+1}-q_j)} - 1).$$

We use the change of variable of Flaschka (Flaschka 1974, Toda 1981),

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

so that we obtain the Lax form (1.1). Thus, both in the nonperiodic and periodic case, the equation of the motions are described by the matrices L and B as in (1.8), i.e.

$$L = \begin{bmatrix} \beta_1 & \alpha_1 & 0 & \dots & \alpha_d \\ \alpha_1 & \beta_2 & \alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \alpha_{d-1} \\ \alpha_d & \dots & 0 & \alpha_{d-1} & \beta_d \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -\alpha_1 & 0 & \dots & \alpha_d \\ \alpha_1 & 0 & -\alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha_{d-1} \\ -\alpha_d & \dots & 0 & \alpha_{d-1} & 0 \end{bmatrix}$$

($\alpha_d = 0$ in the nonperiodic case). In terms of the new variables, α_j, β_j, the Hamiltonian function becomes

$$H = 2\sum_{j=1}^{d}(\beta_j^2 + 2\alpha_j^2) - d = 2\mathrm{tr}(L^2) - d.$$

Hence the retention of second integral

$$\mathcal{I}_2 = \mathrm{tr}(L^2),$$

has the physical significance of conserving of the *total energy* H of the system. Similarly, the integral

$$\mathcal{I}_1 = \mathrm{tr}(L) = \sum_{j=1}^{d}\beta_j,$$

corresponds to the *total momentum* of the system. The presence of the other $d-2$ integrals does not appear have a direct physical interpretation.

5.1.1 Three-particle periodic 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

$$H(\boldsymbol{p},\boldsymbol{q}) = \frac{1}{2}(p_1^2 + p_2^2 + p_3^2) + \mathrm{e}^{-(q_2-q_1)} + \mathrm{e}^{-(q_3-q_2)} + \mathrm{e}^{-(q_1-q_3)} - 3. \tag{5.1}$$

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 (5.1) 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 (Perelomov 1990).

Besides the Lax form, it is possible to make a (non intuitive) change of variables in the Hamiltonian (5.1),

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

where

$$\boldsymbol{p} = A\boldsymbol{\eta}, \qquad \boldsymbol{q} = A\boldsymbol{\zeta}, \tag{5.2}$$

and

$$A = \begin{pmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ -\frac{\sqrt{6}}{3} & 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{pmatrix}, \tag{5.3}$$

which diagonalizes the lattice. We find

$$\begin{array}{ll} q_2 - q_1 = -\frac{\sqrt{6}}{2}\zeta_1 - \frac{\sqrt{2}}{2}\zeta_2, & p_1 = \frac{\sqrt{6}}{6}\eta_1 + \frac{\sqrt{2}}{2}\eta_2 + \frac{\sqrt{3}}{3}\eta_3, \\ q_3 - q_2 = \frac{\sqrt{6}}{2}\zeta_1 - \frac{\sqrt{2}}{2}\zeta_2, & p_2 = -\frac{\sqrt{6}}{3}\eta_1 + \frac{\sqrt{3}}{3}\eta_3, \\ q_1 - q_3 = \sqrt{2}\zeta_2, & p_3 = \frac{\sqrt{6}}{6}\eta_1 - \frac{\sqrt{2}}{2}\eta_2 + \frac{\sqrt{3}}{3}\eta_3, \end{array} \tag{5.4}$$

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

$$H(p_x, p_y, x, y) = \frac{p_x^2 + p_y^2}{2} + \frac{1}{24}\left(\mathrm{e}^{2y+2\sqrt{3}x} + \mathrm{e}^{2y-2\sqrt{3}x} + \mathrm{e}^{-4y}\right) - \frac{1}{8} \tag{5.5}$$

(Toda 1981, Berry 1987). Although such Hamiltonian is mathematically equivalent to (5.1), 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

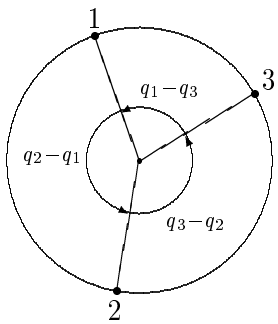


Figure 5.1: Three particles on a ring with exponential interaction.

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 (5.5) were originally considered by Ford and his collaborators in their investigations of the integrability of the cyclic three-particle Toda lattice. As already mentioned in Chapter 1, if we truncate the Hamiltonian function (5.5) 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 (5.5). 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 (5.4), to obtain the variables x, y, p_x, p_y that appear in the Hamiltonian function (5.5). Specifically, we follow the procedure below:

Problem 1 The three-particle periodic Toda lattice. Algorithm:

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 $U' = B(UL_nU^T)U$, for $t \in [t_n, t_{n+1}]$ and initial condition $U(t_n) = I$, to obtain a numerical approximation U_{n+1}. At this stage, we employ various numerical methods.

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

4. Transform the numerical approximation for the Flaschka variables α_j's, β_j's into x, y, p_x, p_y by means of (5.4).

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 were 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 $U' = B(L)U$ with the order-four implicit Magnus collocation method (4.51)-(4.52). When applied to the Lie-group

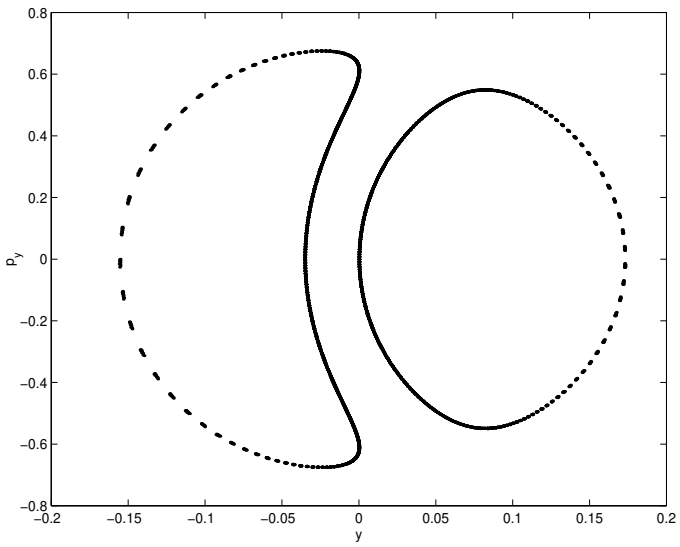


Figure 5.2: The Poincaré section of the Hamiltonian (5.5) obtained with an isospectral method based on the order-four Magnus collocation method (5.6) for Lie groups (`MagnusZ4`).

equation $y' = \gamma(t,y)y$ the method reads

$$\begin{aligned}
Y_1 &= \exp\Big(h(\tfrac{1}{4}\gamma_1 + (\tfrac{1}{4} - \tfrac{\sqrt{3}}{6})\gamma_2) + \tfrac{h^2}{2}(\tfrac{5}{72} - \tfrac{\sqrt{3}}{24})[\gamma_1,\gamma_2]\Big)y_n,\\
\gamma_1 &= \gamma(t_n + (\tfrac{1}{2} - \tfrac{\sqrt{3}}{6})h, Y_1),\\
Y_2 &= \exp\Big(h((\tfrac{1}{4} + \tfrac{\sqrt{3}}{6})\gamma_1 + \tfrac{1}{4}\gamma_2) + \tfrac{h^2}{2}(-\tfrac{5}{72} - \tfrac{\sqrt{3}}{24})[\gamma_1,\gamma_2]\Big)y_n,\\
\gamma_2 &= \gamma\Big(t_n + (\tfrac{1}{2} - \tfrac{\sqrt{3}}{6})h, Y_2\Big),\\
y_{n+1} &= \exp\Big(\tfrac{h}{2}(\gamma_1+\gamma_2) - \tfrac{\sqrt{3}}{12}h^2[\gamma_1,\gamma_2]\Big)y_n.
\end{aligned} \tag{5.6}$$

We label the numerical approximation obtained in this manner as `MagnusZ4` and in Figure 5.2 the Poincaré section produced by the scheme appears like two smooth closed curves.

In Figure 5.3 we plot the Poincaré section when the isospectral equations are solved with an isospectral method, based in this case on an explicit Lie-group method. Thus, we use the Lie-group method in example 4.12, which possesses order three. When applied to the Lie-group equation $y' = \gamma(t,y)y$, the scheme reads

$$\begin{aligned}
Y_1 &= y_n, & \gamma_1 &= \gamma(t_n, Y_1),\\
Y_2 &= \exp(\tfrac{h}{2}\gamma_1)y_n, & \gamma_2 &= \gamma(t_n + \tfrac{h}{2}, Y_2),\\
Y_3 &= \exp\left(h(-\gamma_1 + 2\gamma_2)\right)y_n, & \gamma_3 &= \gamma(t_n + h, Y_3),\\
y_{n+1} &= \exp\Big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3) - \tfrac{h^2}{2}[\gamma_1 - \gamma_3, \tfrac{2}{15}\gamma_2 + \tfrac{1}{30}\gamma_3]\Big)y_n,
\end{aligned} \tag{5.7}$$

and we label it `RKZ3`.

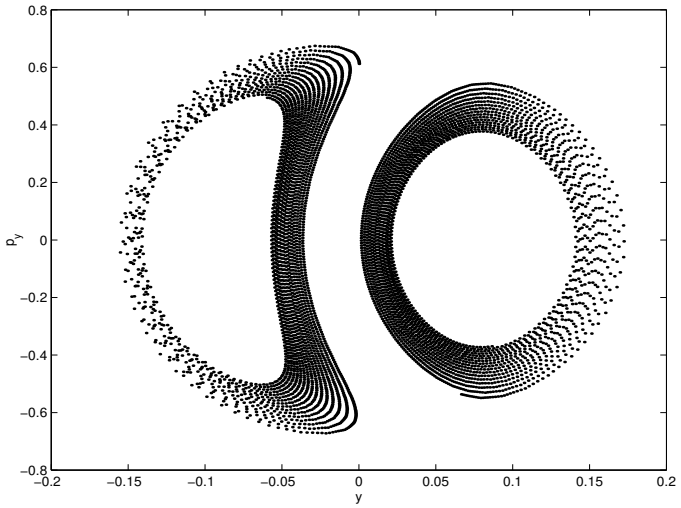


Figure 5.3: The Poincaré section of the Hamiltonian (5.5) obtained with the isospectral method `RKZ3`, based on the order-three Lie-group scheme (5.7).

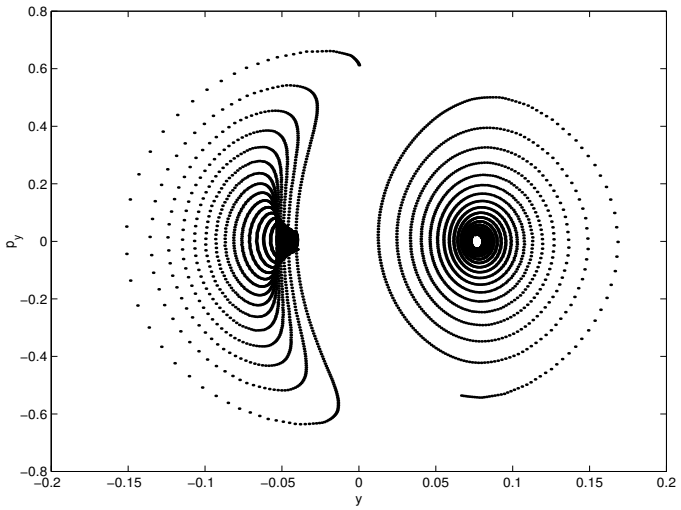


Figure 5.4: The Poincaré section of the Hamiltonian (5.5) obtained with the non-isospectral method $\texttt{RK3}_\texttt{U}$, based on the explicit order-three Runge–Kutta scheme with tableau (5.8).

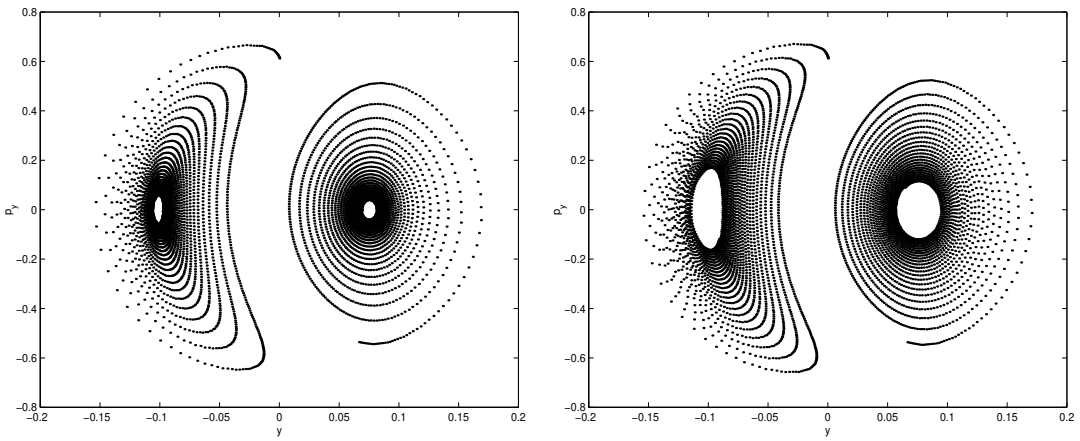


Figure 5.5: The Poincaré section of the Hamiltonian (5.5) obtained with the non-isospectral method $\mathtt{RK3_U}$, based on the explicit order-three Runge–Kutta scheme with tableau (5.8) combined with QR projection (left). In the right plot, a canonical QR projection is applied also to the internal stages of the ODE solver. Both plots reveal much more dissipativity than the Lie-group method (5.7) (compare with Figure 5.3).04

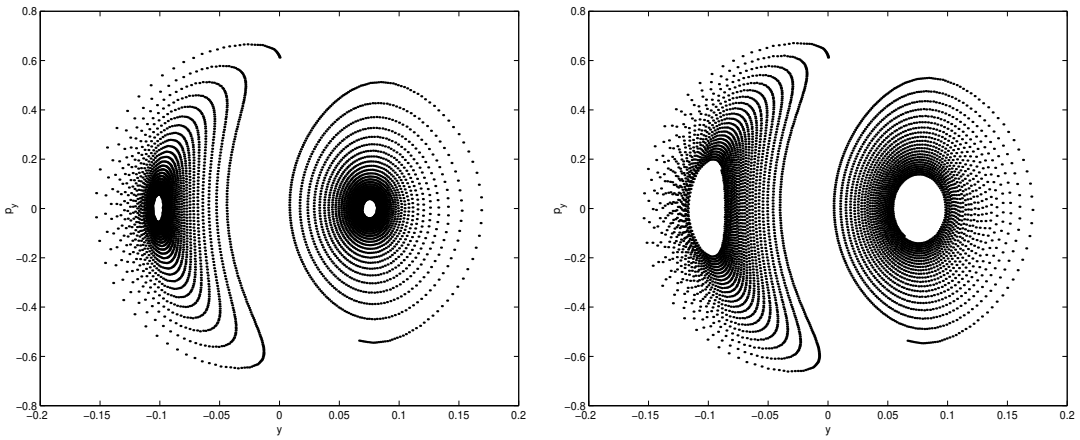


Figure 5.6: The Poincaré section of the Hamiltonian (5.5) obtained with the non-isospectral method $\mathtt{RK3_U}$, based on the explicit order-three Runge–Kutta scheme with tableau (5.8) whereby a *polar* projection is applied to preserve orthogonality (left). In the right plot, the polar projection is applied also to the internal stages of the ODE solver. Also in this case, although we employ orthogonal matrices, the curves spiral inwards much faster than in Figure 5.3.

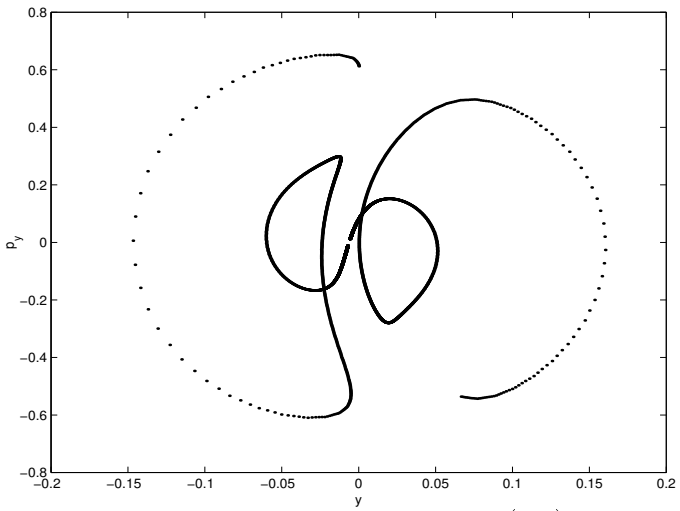


Figure 5.7: The Poincaré section of the Hamiltonian (5.5) obtained with the non-isospectral method $\texttt{RK3}_\texttt{L}$, based on the Runge–Kutta scheme with tableau (5.7).

Next, in Figure 5.4, we display the sections obtained according to the following procedure, that we call $\texttt{RK3}_\texttt{U}$: We solve the orthogonal flow $U' = B(L)U$ in step 2 with the classical three-stage, order-three, explicit RK method with tableau

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \quad . \tag{5.8}$$

The matrices U_n are not any longer orthogonal, and the matrix transformation $L_{n+1} = U_{n+1} L_n U_{n+1}^T$ in step 3 is no longer a similarity transformation. Hence the integrals of the motion of the original system are not retained. In Figures 5.5 and 5.6 a QR projection and a polar projection are applied to orthogonalize U_{n+1} so that isospectrality is retained. If we simply solve $L' = [B(L), L]$ with the Runge–Kutta scheme with tableau (5.8), a method that we call $\texttt{RK3}_\texttt{L}$, we obtain the Poincaré section in Figure 5.7. Also in this case, those curves that should be closed spiral inwards, as a matter of fact much faster than in Figure 5.3.

In order to get an isospectral solution while employing a non-orthogonal scheme in step 2, we should use a similarity transformation in step 3 instead, for instance $L_{n+1} = U_{n+1} L_n U_{n+1}^{-1}$. This procedure is equivalent to a *semi-explicit* method, of the kind introduced in Section 3.4. However, the matrix U may become very ill-conditioned.[1] In the implementation of semi-explicit methods, we have employed the Runge–Kutta method with tableau (5.8) and the numerical experiment can be carried only up to $t = 261.2$. The numerical integration is stopped when the reciprocal of condition number $\mu(U) = \|U\| \cdot \|U^{-1}\|$, is less than 10^{-14}, which means that U is nearly singular: This is very surprising since it is the most obvious approach to solve the isospectral flow $L' = [B(L), L]$

[1] The reason is that, instead of using the SO(3) group action (SO(3) is a compact group), we are using instead the action of GL(3) which is non-compact.

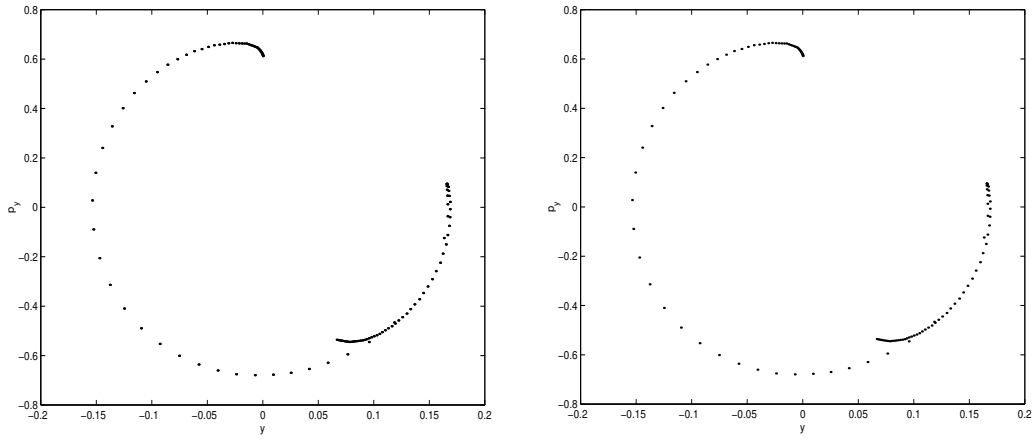


Figure 5.8: The Poincaré section of the Hamiltonian (5.5) obtained with a semi-explicit procedure, without symmetrization (left) and with symmetrization (right). The plots are practically identical, the only difference being the blow-up time, $t = 261.2$ (left) and $t = 266$ (right).

whilst retaining isospectrality. If we use a symmetrising procedure as the one described in §3.4, alternating the integration

$$U' = B(L)U, \qquad U_{2n} = I$$

(in tandem with $L_{2n+1} = U_{2n+1} L_{2n} U_{2n+1}^{-1}$), with

$$V' = -VB(L), \qquad V_{2n+1} = I$$

(coupled with $L_{2n+2} = V_{2n+2}^{-1} L_{2n+1} V_{2n+2}$), we are able to extend the integration time to $t = 266$, but the result obtained is not much better (Figure 5.8).

A comparison between Figure 5.2 and those shown next, reveals a fundamental difference in the information delivered by the numerical approximants. To our surprise, the curves in Figures 5.3 and 5.8 spiral inwards even though all the integrals of the original Toda problem are preserved to machine accuracy, while according to the above mentioned theory of integrable systems, the plots should instead have had the form of smooth closed curves, *á la* Figure 5.2. As explained by MacKay (n.d.), numerical methods applied to the Lax pair (1.1) do not necessarily retain the length of the ring, in which case the system acquires an extra degree of freedom. In the original variables one has

$$(q_2 - q_1) + (q_3 - q_2) + (q_1 - q_3) = 0,$$

which, in the Flaschka variables (see p 97), is equivalent to

$$\prod_{j=1}^{3} \alpha_j = \frac{1}{8}.$$

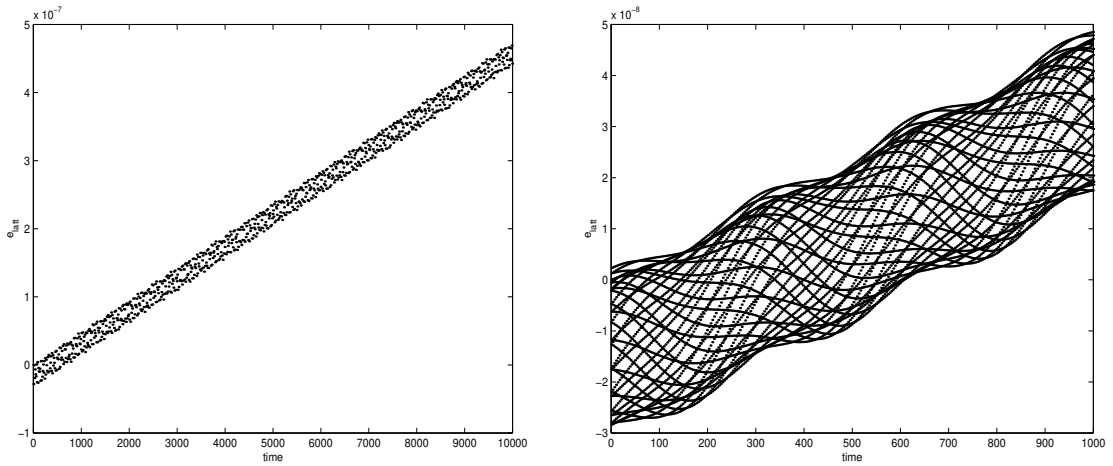


Figure 5.9: Increment in the length of the ring against time for `MagnusZ4` based on (5.6) (left). The right plot closes upon e_{latt} in the interval $[0, 1000]$.

Thus, the *increment in the lattice length*

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

should equal zero along the theoretical solution. Unfortunately, this does not hold for the numerical solution obtained when applying discretization schemes to (1.1): in the Lax formulation of the problem, the α_j's are treated as independent variables so that, under the effect of the numerical error, e_{latt} might differ from zero.

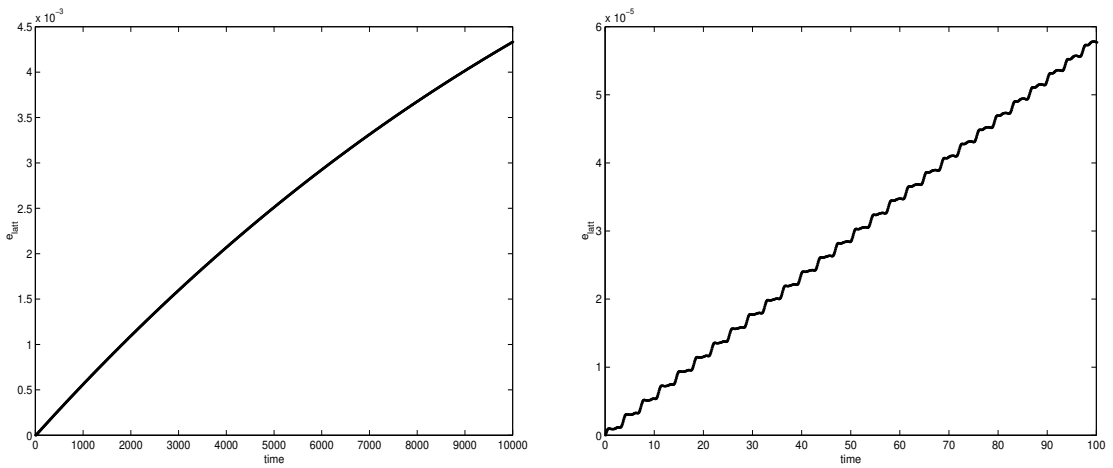


Figure 5.10: Increment in the length of the ring against time for the explicit scheme `RKZ3` based on (5.7) (left). The right plot closes upon the error growth in the interval $[0, 100]$.

In Figures 5.9–5.12 we plot the increment in the lattice length e_{latt} against time for various numerical schemes.

For the isospectral scheme `MagnusZ4`, the increment e_{latt} is linear function of time throughout the whole interval of integration, with slope approximatively equal to $\frac{1}{2}10^{-10}$. Note the curious pattern of e_{latt} in Figure 5.9 which corresponds to a single trajectory.

When we consider the isospectral method `RKZ3`, the error no longer grows linearly, but it is very well approximated by a straight line with slope $\approx \frac{29}{5}10^{-7}$ in the first part of the numerical integration. Moreover, the close-up in Figure 5.10 indicates that, at first, e_{latt} is very similar to a step function.

The increment and its close-up in the interval $[0, 100]$ for the method $\texttt{RK3}_{\texttt{U}}$ are plotted in Figure 5.11. In this case, the error has a nonlinear behaviour. Eventually, both the curves in Figure 5.10 and 5.11 will become parallel to the time-axis, since the schemes display dissipative behaviour, and the particles of the original problem will tend to a resting position as t tends to infinity. Although e_{latt} and the error in the integration by the methods $\texttt{RK3}_{\texttt{U}}$ and `RKZ3` are of the same order of magnitude, at $T = 10000$ the error of the first scheme is three times as large as for `RKZ3`. This is dramatically emphasized in the comparison between Figures 5.3 and 5.4. Note that the three methods `RKZ3`, $\texttt{RK3}_{\texttt{U}}$ and $\texttt{RK3}_{\texttt{L}}$ originate in the same Runge–Kutta scheme, yet `RKZ3` retains the Lie-group structure and takes full advantage of the SO(3) group action, $\texttt{RK3}_{\texttt{U}}$ does not retain the Lie-group structure but employs SO(3)'s action, while $\texttt{RK3}_{\texttt{L}}$ is neither group invariant nor employs the group action.

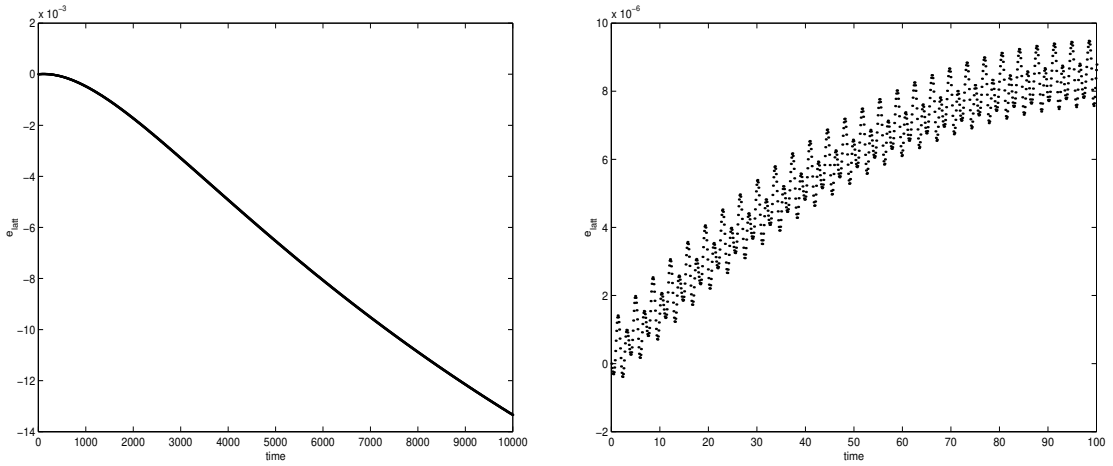


Figure 5.11: Increment in the ring's length (left) for the non-isospectral method $\texttt{RK3}_{\texttt{U}}$ and the close-up on the error in the interval $[0, 100]$ (right).

In Figure 5.12 we have plotted e_{latt} when the isospectral flow is solved with a semi-explicit method (top row) and with a semi-explicit method with symmetrization (bottom row). The schemes produce a linear error growth at first, but both the approaches become quite unstable, so that after $t \approx 261.2$ ($t = 266$) the methods fail. A stepwise increment of the error in the first part of the integration is present also with these schemes.

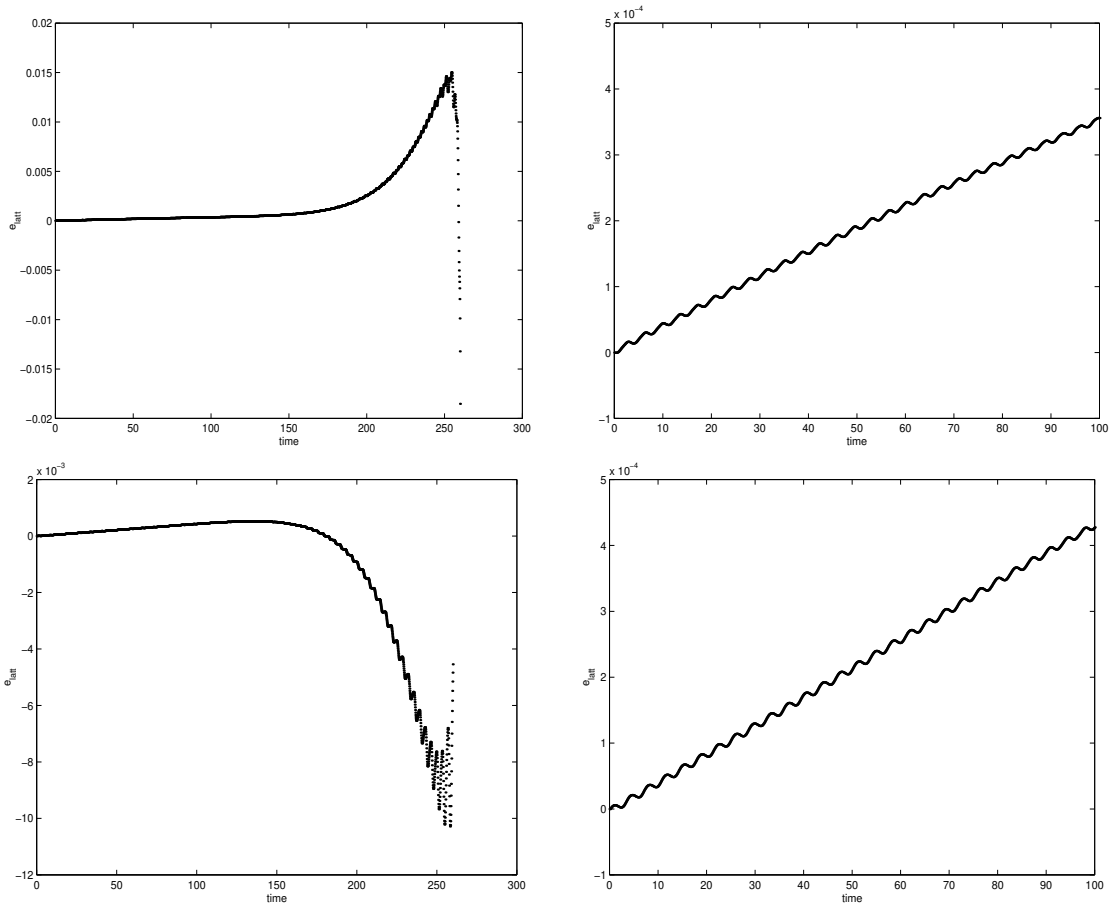


Figure 5.12: Increment in the ring's length for semi-explicit methods without (top) and with symmetrization (bottom). In the right column, we display a close-up of the error produced by both approaches.

Lie-group methods based on the Magnus expansion with symmetric collocation nodes seem to possess a great deal of structure which is yet to be investigated. Figure 5.13 displays the error and its close-up for the order-four isospectral scheme obtained by means of the Lie-group collocation method based on Lobatto points,

$$\begin{aligned}
Y_1 &= y_n,\\
\gamma_1 &= \gamma(t_n, Y_1),\\
Y_2 &= \exp\Big(h(\tfrac{5}{24}\gamma_1 + \tfrac{1}{3}\gamma_2 - \tfrac{1}{24}\gamma_3) - \tfrac{h^2}{2}(\tfrac{11}{240}[\gamma_1,\gamma_2] + \tfrac{5}{576}[\gamma_1,\gamma_3] + \tfrac{1}{72}[\gamma_2,\gamma_3])\Big)\, y_n,\\
\gamma_2 &= \gamma(t_n + \tfrac{1}{2}h, Y_2),\\
Y_3 &= \exp\Big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3) - \tfrac{h^2}{2}(\tfrac{2}{15}[\gamma_1,\gamma_2] + \tfrac{1}{30}[\gamma_1,\gamma_3] + \tfrac{2}{15}[\gamma_2,\gamma_3])\Big)\, y_n,\\
\gamma_3 &= \gamma(t_n + h, Y_3),\\
y_{n+1} &= \exp\Big(h(\tfrac{1}{6}\gamma_1 + \tfrac{2}{3}\gamma_2 + \tfrac{1}{6}\gamma_3) - \tfrac{h^2}{2}(\tfrac{2}{15}[\gamma_1,\gamma_2] + \tfrac{1}{30}[\gamma_1,\gamma_3] + \tfrac{2}{15}[\gamma_2,\gamma_3])\Big)\, y_n,
\end{aligned} \tag{5.9}$$

while its Poincaré section of the Hamiltonian (5.5) is (to plotting accuracy) identical to the one produced by the Magnus order-four collocation method with Gauss–Legendre nodes (see Figure 5.2).

Numerical results for the various schemes are displayed below in Table 5.1.

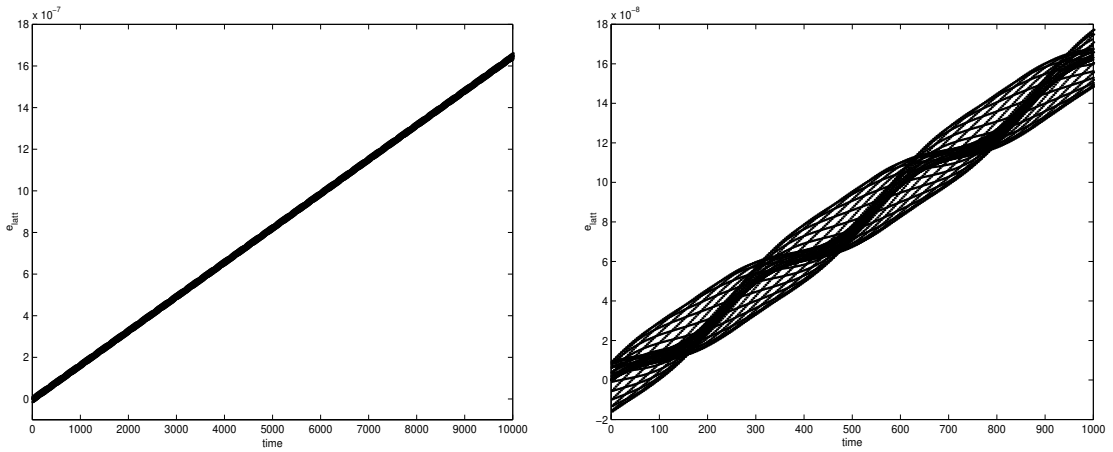


Figure 5.13: Increment in the ring's length (left) for the Lobatto method (5.9) and close-up on the error in the interval $[0, 1000]$ (right).

5.1.2 The nonperiodic Toda lattice

In this second example, we consider a nonperiodic finite lattice. Thus, as already mentioned in the first chapter, the asymptotic behaviour of the system is known, since when $t \to \infty$, the particles act as if they were free. In this case, the matrix L of the Lax pair (1.1) tends to a diagonal matrix, the diagonal entries representing the velocity with which the particles are scattered as t tends to infinity.

The usual procedure for solving this class of problems consists of finding the asymptotic momenta first (the eigenvalues of L) in order to obtain the intermediate position of the particles by means of inverse scattering. However, if we want to recover the intermediate position of the particles without solving first for the eigenvalues, we may employ a numerical method for ODEs directly on the isospectral equation (1.1). The algorithm here is as follows.

Problem 2 Nonperiodic Toda lattice. Algorithm:

1. First write the problem in the isospectral form $L' = [B(L), L]$, where we take $B(L) = L_- - L_+$, i.e. the difference between the lower triangular part of L and its upper triangular part.[2]
2. As step 2 in Problem 1.
3. As step 3 in Problem 1.

Differently from the periodic lattice, we are interested here in short-term integration, since for large values of time there is not much interaction between the particles. We consider a three-particle lattice and choose initial condition $\boldsymbol{p}_0, \boldsymbol{q}_0$ as for the periodic lattice above, i.e.

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

which yield

$$L_0 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \end{pmatrix},$$

as initial condition for the corresponding isospectral flow. The interval of integration is $[0, 1024]$ and the stepsize is $h = \frac{1}{10}$.

Here we consider three order-three explicit schemes: the first is `RKZ3`, the isospectral scheme based on the Lie-group scheme (5.7), the second and third are $\texttt{RK3}_\texttt{U}$ and $\texttt{RK3}_\texttt{L}$ respectively, based on the classical Runge–Kutta scheme with tableau (5.8), and do not retain isospectrality.

In Figure 5.14 we plot the global error $\|L_n - L(t_n)\|_2$ and the magnitude of the extra-diagonal elements of the numerical solution L_n sampled at time $t_n = 2, 4, 8, \ldots, 1024$.

Although the off-diagonal elements are very much the same for the three numerical schemes, when it comes to the global error we observe that the isospectral method `RKZ3` is 'auto-correcting' as $t \to \infty$, while the same cannot be said for the non-isospectral schemes $\texttt{RK3}_\texttt{U}$ and $\texttt{RK3}_\texttt{L}$: in practice, because of the absence of isospectrality, the error committed in the first time steps is propagated throughout the whole interval of integration.

[2] We could also take B as the difference of the subdiagonal of L and its superdiagonal. Along the theoretical trajectory the two choices for B are equivalent. However, since isospectral methods usually introduce some fill-in in the extra-tridiagonal part, the two formulations are not equivalent from the numerical point of view, since the latter introduces spurious fixed points.

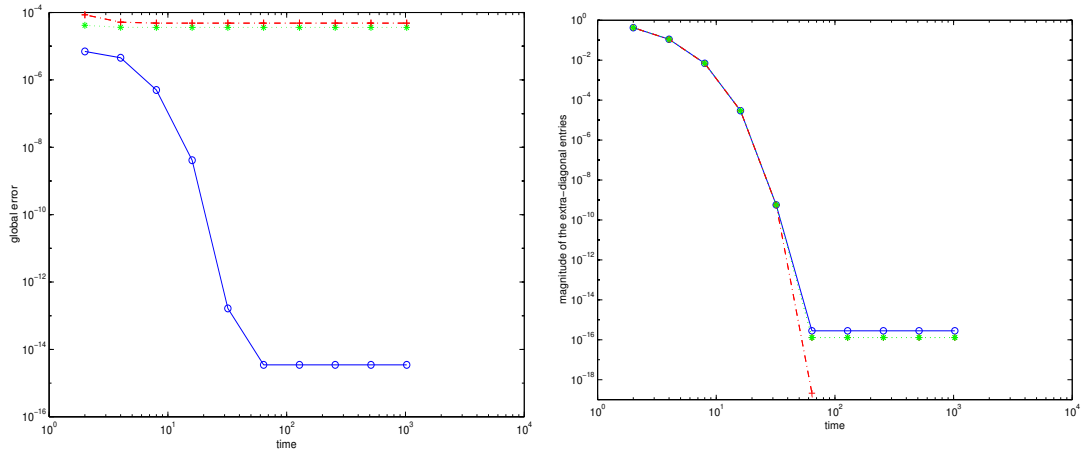


Figure 5.14: Global error, $\|L_n - L(t_n)\|_2$, and extra-diagonal elements of the matrix L_n at $t_n = 2, 4, \ldots, 1024$. The methods are: `RKZ3` (blue ovals joined by a solid line), $\texttt{RK3}_\texttt{U}$ (green asterisks joined by dotted line) and $\texttt{RK3}_\texttt{L}$ (red pluses joined by dash-dotted line).

With regards to banded flows, as for instance the nonperiodic Toda flow, we have to mention that methods that exploit the group action $U' = B(L)U$ usually do not retain the bandwidth (tridiagonality) of the matrix L. This is a consequence of the similarity transformation in step 3 and of numerical error. The fill-in is usually of the order of the method and it is typically very small when compared with the remaining entries of the matrix. Moreover, in the present example, it soon disappears due to the gradient-flow character of the problem. The fill-in in the extra-tridiagonal entries of the matrix L is represented in Figure 5.15 the methods `RKZ3` and $\texttt{RK3}_\texttt{U}$, while it is null for $\texttt{RK3}_\texttt{L}$.

The fill-in is unfortunate in the Toda-lattice case (for which the theoretical solution L is tridiagonal) since it adds to the computational cost of the methods, but it is very natural for other isospectral flows, e.g. generalized QR flows and double-bracket flows, for which the theoretical solution itself L does not retain bandwidth. When tridiagonality is a must, occasional tridiagonalization could help in filtering out the fill-in. Instead, for some linear-algebra type flows, the stepsize could be used to annihilate some targeted entries of the matrix L.

Comparison of results for various methods and approaches for this problem is displayed below in Table 5.2.

5.2 Matrix orbits visualization

As we have already mentioned in Chapter 1, nonperiodic Toda flows, QR flows, double-bracket flows and their generalizations might be used to construct matrix flows to solve eigenvalue problems or to perform various matrix calculations. When applied to these problems, the asymptotic state is often a diagonal matrix so that the results are virtually the same as in the nonperiodic Toda lattice example. In other words, isospectral methods will converge to the correct diagonal matrix, while the others might converge to a wrong

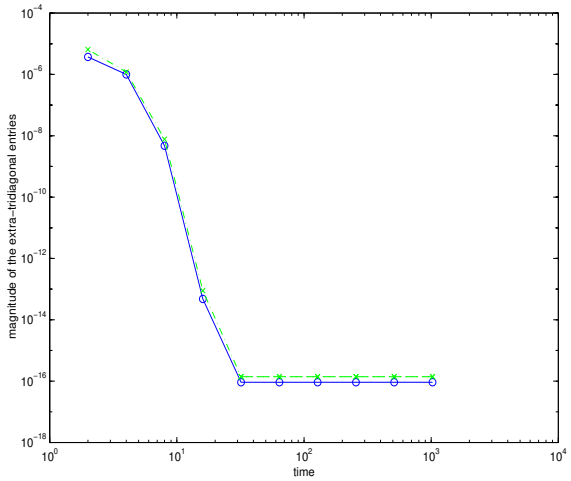


Figure 5.15: Extra-tridiagonal fill-in for the numerical approximations of the matrix L for a three-particle nonperiodic Toda flow. The symbols are like in Figure 5.14. The extra-tridiagonal fill-in is initially of the same order of the truncation error of the method and it converges to zero faster than the subdiagonal elements of L.

diagonal matrix.

However, from a practical point of view, solving eigenvalue and analogous problems with ODE methods is not very competitive when compared with the QR algorithm with shifts (Golub & van Loan 1989). Watkins (1982) claims that the QR method with shifts (which is anyhow isospectral) is superior both for speed and accuracy. Obviously, when solving eigenvalue problems with isospectral flows, accuracy is lost if we do not employ an isospectral method, as already shown in the nonperiodic Toda lattice example. With regards to speed, it is clear that ODE solvers are more expensive than QR iterations because they pay extensive attention to the local accuracy of the solution, while the real interest is in its asymptotic state. Further complications arise from the presence of equilibria other than the asymptotic state, which are saddle points for the flow[3] and slow down the ODE process when the solution happens to pass nearby. Also for SVD and linear programming (e.g. unconstrained optimization) we do not expect isospectral ODE solvers to be competitive with standard SVD and optimization algorithm.

More interestingly, we would like to focus upon the time evolution of nonlinear matrix flows. To this end, we consider matrix orbit visualization, following (Brunet 1994). In the first visualization we consider the flow

$$L' = [B(L), L], \qquad B(L) = L_- - L_+,$$

with an initial condition L_0, a 51×51 symmetric Toeplitz matrix with zeros on the first ten (super- and sub-) diagonals and ones elsewhere. Some snapshots of the flow are plotted in Figure 5.16, for various values of time. The entries of L_n are plotted in a gray scale, black corresponding to zero and white corresponding to one. Entries larger than one in modulus

[3]For instance diagonal matrices where the order of the eigenvalues is exchanged.

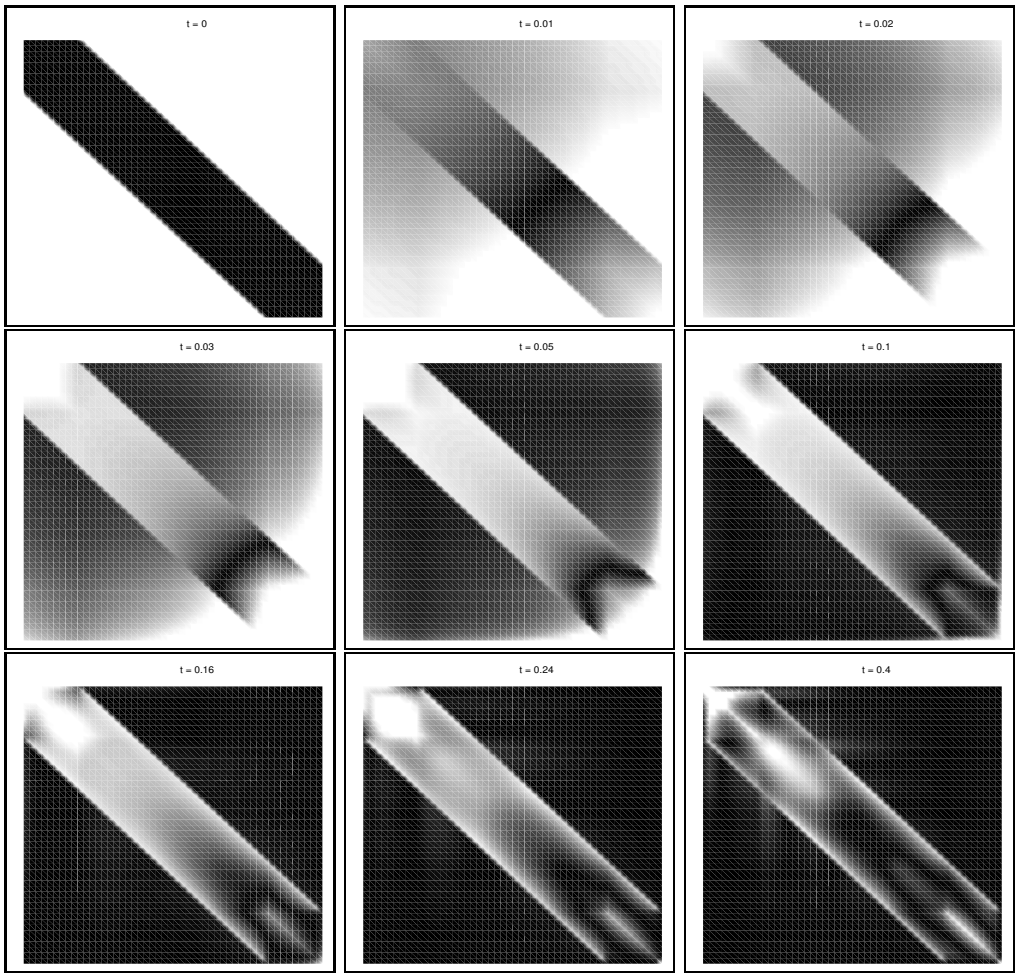


Figure 5.16: Isospectral deformations of a 51×51 Toeplitz matrix (snapshots). The entries of the matrix L (absolute values) are plotted in a gray scale, black corresponding to zero and white corresponding to one.

are also plotted in white. Figure 5.16 is obtained solving with the explicit isospectral method `RKZ3`, with stepsize $h = \frac{1}{100}$. After some time, the real action is concentrated in the banded part of the matrix and the banded structure remains evident throughout the process.

This kind of flows make a useful example to display the symmetrizing strategy for semi-explicit methods. If the stepsize is small, then we would not see much, therefore we employ here a larger stepsize, $h = \frac{1}{10}$, which will not produce any sensible approximation to the problem in terms of accuracy but will magnify the symmetrizing effect. We use again the semi-explicit method, based on the 'Kutta' scheme (5.8) and its symmetric version, alternating even-odd steps as already explained above. Although the approximated solutions are totally wrong and both methods fail (the stepsize is too large), it is interesting how more symmetric is the numerical approximation obtained using the symmetrized scheme (see Figure 5.17).

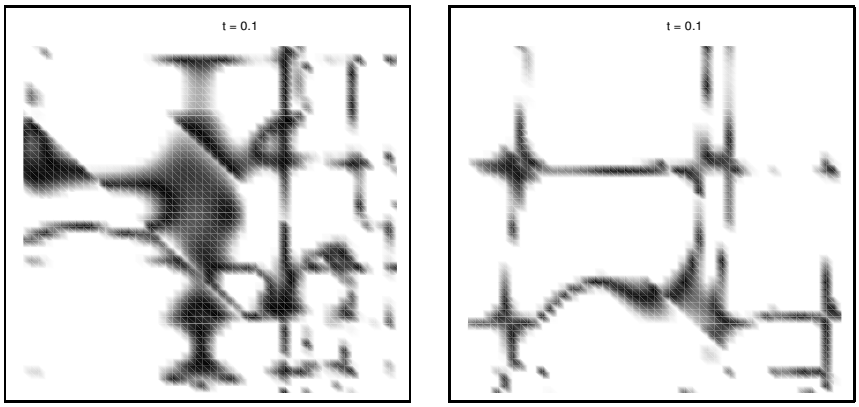


Figure 5.17: Isospectral deformation of a 51×51 Toeplitz matrix with semi-explicit methods without (left) and with symmetrization (right). The stepsize of integration is $h = \frac{1}{10}$.

As already mentioned in the first chapter, a nonperiodic Toda flow is equivalent to a double-bracket flow

$$L' = [[N, L], L],$$

whereby N is the diagonal matrix

$$N = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 2 & 0 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & d \end{pmatrix}.$$

If we modify the matrix N, the orbit becomes totally different, although the initial condition and asymptotic state are the same. In the next experiment, we take N with 51 diagonal *logarithmically* spaced entries between 1 and d. The behaviour of the matrix L is now completely different from that in Figure 5.16, the initial bandwidth becomes smudged already at $t = 0.1$, yet the asymptotic state is the same (see Figure 5.18).

5.3 The inverse Toeplitz eigenvalue problem (ITEP)

In this section we apply isospectral methods to finding a symmetric Toeplitz matrix with a prescribed set of real numbers as its eigenvalues. The flows that we consider in this case generally converge to an asymptotic state, so that in this context it is very useful to consider explicit methods.

In our experiments, we consider the isospectral flow proposed by Chu & Driessel (1989) already introduced in Chapter 1, whereby the matrix B is given by the formula (1.21),

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

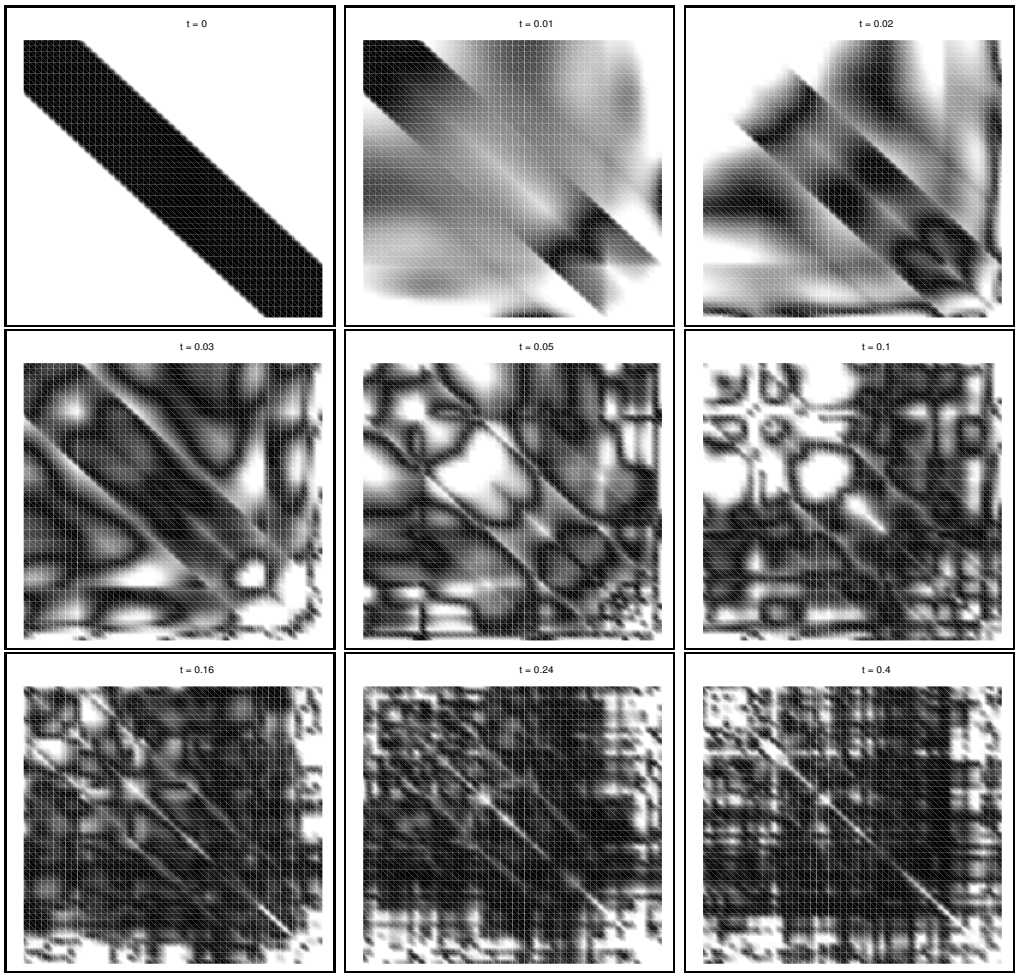


Figure 5.18: Isospectral deformations of a 51×51 Toeplitz matrix (snapshots) for a double-bracket flow whose entries for the matrix N are logarithmically spaced. The elements of the matrix L (absolute values) are plotted in a gray scale, black corresponding to zero and white corresponding to one.

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 not monotonically decreasing along the orbit of the flow, as displayed in Figure 5.19.

It is known that the inverse Toeplitz eigenvalue problem has a solution but not much is known about uniqueness (Landau 1994). In the 4×4 case Delsarte & Genin (1984) 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. Recall that a d-component vector $\boldsymbol{v}$

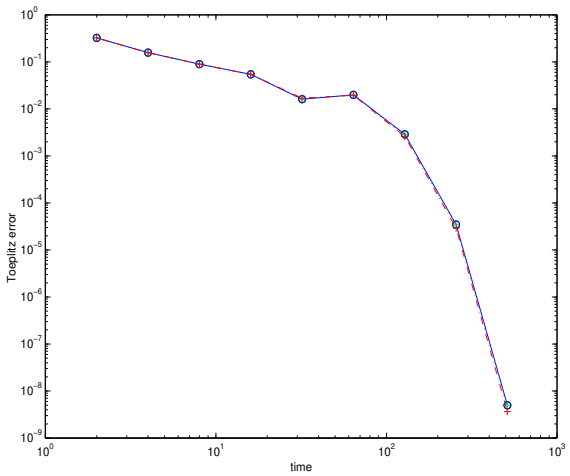


Figure 5.19: Samples of $\|B(L_n)\|$, in 2-norm, at $t_n = 2, 4, \ldots, 512$ (logarithmic scale). The methods are: `RKZ3` (blue ovals with solid line), $\texttt{RK3}_\texttt{U}$ (green asterisks with dotted line) and $\texttt{RK3}_\texttt{L}$ (red plus with dash-dotted line), yield virtually identical outputs.

is *even* if

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

and *odd* if

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

Furthermore, Delsarte & Genin (1984) 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 (1994) showed that, given an arbitrary set of real numbers, it is possible to construct a 'regular' Toeplitz matrix with those numbers as eigenvalues. 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 by Chu & Driessel (1989): 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 decom-

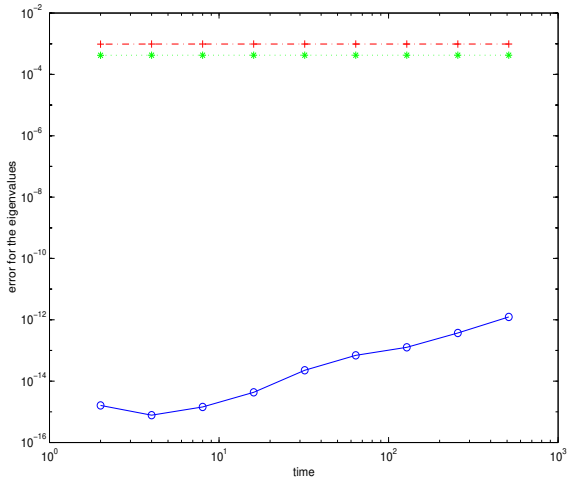


Figure 5.20: Error for the eigenvalues of the numerical solution at $t_n = 2, 4, \ldots, 512$ (logarithmic scale). The symbols are as in Figure 5.19.

position, we find

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

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

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

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 Laurie (1996), 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

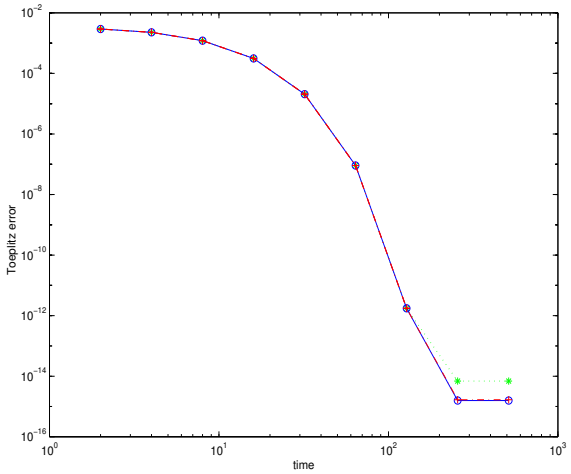


Figure 5.21: Toeplitz error of the flow with initial condition $\tilde{L}_0$ at $t_n = 2, 4, \ldots, 512$. Symbols are as in Figure 5.19. The convergence to a Toeplitz matrix is now faster than in (5.19).

the regularity condition. For our example, 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, as displayed in Figure 5.21. Numerical results for this problem with initial conditions L_0 and $\tilde{L}_0$ are reported below in §5.4.3.

5.4 Tables and comments

5.4.1 Problem 1: Three-particle periodic Toda lattice

Here we reconsider the three-particle periodic Toda lattice example already introduced in §5.1.1 and report some numerical results. 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 5.1 should be understood as follows:

Order Order of the method.

Iso 'y' if the method is isospectral, 'n' otherwise.

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

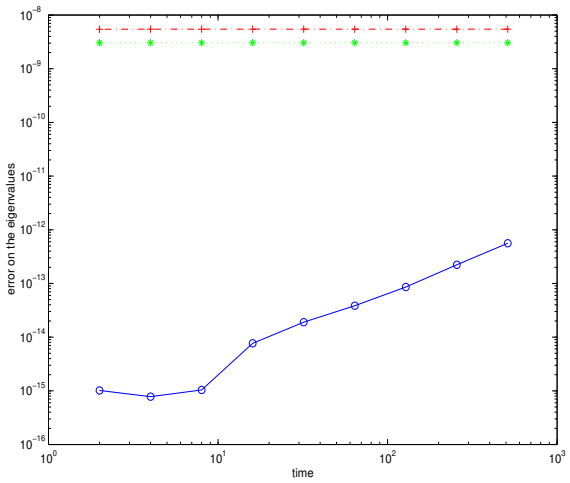


Figure 5.22: Error on the eigenvalues of the numerical solution with initial condition $\tilde{L}_0$ at $t_n = 2, 4, \ldots, 512$ (logarithmic scale). The symbols are as in Figure 5.19.

`MagnusZ4` Algorithm based on the Magnus order-four collocation method (5.6).

`RKMK4` The RK Munthe-Kaas scheme for Lie-groups based on the classical order-four RK Gauss–Legendre counterpart (cf. §4.5.2).

`RK4` Classical Gauss–Legendre RK of order four.

`RKZ3` Algorithm based on the scheme (5.7), a Lie-group adaption of the classical 'Kutta' RK scheme of order three.

`semiexpl3` Semi-explicit method based on the 'Kutta' Runge–Kutta method (5.8) for $U' = B(L)U$. The method is applied also with a symmetrized version. See §3.4.

$\texttt{RK3}_{\texttt{L}}$ Classical order-three RK method known as *Kutta,* with coefficients in (5.8).

(GL) refers to schemes based on Gauss–Legendre nodes, (Kutta) refers to scheme based on the 'Kutta' scheme (5.8). Since non-isospectral methods are applied in two different ways, we append to a method the subscript `U` when it is used to solve $U' = B(UL_nU^T)U$ in tandem with $L_{n+1} = U_{n+1}L_nU_{n+1}^T$, e.g. $\texttt{RK3}_{\texttt{U}}$, while we use the subscript `L` when the method is applied to the flow $L' = [B(L), L]$, e.g. $\texttt{RK3}_{\texttt{L}}$.

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

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

(cf. §5.1.1). 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, 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 (Marthinsen, Munthe-Kaas & Owren 1997, Simo & Wong 1991).[4]

Taking into account the results in Table 5.1 and the plots of the corresponding Poincaré section, 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. Semi-explicit methods instead become very unstable when applied to this problem and should not be used in similar contexts.

[4] 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.

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`	`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]	$\texttt{RK4}_{\texttt{U}}$ (GL)	3.6632e-03	6.3204e-10	1.3604e-09	1.7147e-08	2.0349e+09	
	n	$\texttt{RK4}_{\texttt{L}}$ (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	`semiexpl3` (Kutta)	fail	—	—	—	—	
	y	`semiexpl3`[b] (Kutta)	fail	—	—	—	—	
	n	$\texttt{RK3}_{\texttt{U}}$ (Kutta)	0.3094	1.5972e-01	1.2069e-01	1.3353e-02	9.7804e+07	
	n	$\texttt{RK3}_{\texttt{L}}$ (Kutta)	0.2354	1.2693e-01	6.9215e-02	4.9914e-03	8.9806e+07	

Table 5.1: Comparison of results for Problem 1, 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 §3.2.2.

[b] Symmetric, with even-odd steps alternating solvers, see text for details.

5.4.2 Problem 2: Three-particle nonperiodic Toda Lattice

Also in this case, 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 5.2 should be read as for Table 5.1, except that the meaning of 'Flops' is amended below. All the relevant measurements are taken at time $t_n = T = 1024$.

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. Also in this example, the results are reported in two columns for methods based on Lie-groups schemes: 'expm', that refers to the flops used when employing MATLAB™ built-in function `expm`, and 'ER_expm', that refers to the flops used when the exponential is evaluated (exactly) using the *Euler–Rodriguez* formula (see above).

In this application, the advantage of employing an isospectral method becomes evident by comparing the global errors of the various methods at $T = 1024$. For isospectral schemes both the global error and the error in the first integrals remains at a machine-accuracy magnitude. With regards to implicit order-four methods, we observe that the Euler–Rodriguez formula reduces significantly the implementation costs. Yet in this case, where the solution has a gradient-flow type behaviour, isospectral explicit methods offer an advantageous deal, taking into account implementation costs and error. Semi-explicit methods, especially of the 'symmetric' variety, are very fast and accurate.

5.4.3 Problem 3: The inverse Toeplitz eigenvalue problem

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 5.3 refers to the 5×5 problem discussed in section §5.3 and should be read as for Table 5.1, except for the items below. All the relevant measurements are taken at time $t_n = T = 1024$.

$e(\boldsymbol{\lambda})$ 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. 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.

Order	Iso	Method	Global error $T = 1024, h = 0.1$	$\lvert d(\mathcal{I}_2)\rvert$	$\lvert d(\mathcal{I}_3)\rvert$	Flops	
						`expm`	`ER_expm`
4	y	`MagnusZ4` (GL)	1.2171e-15	5.3291e-15	9.1038e-15	3.9265e+07	1.6206e+07
	y	`RKMK4` (GL)	4.8905e-15	5.1070e-15	7.9936e-15	4.9817e+07	2.8675e+07
	y(n)[a]	$\texttt{RK4}_\texttt{U}$ (GL)	4.2920e-15	0[b]	1.9984e-15	2.3721e+07	
	n	$\texttt{RK4}_\texttt{L}$ (GL)	2.5264e-07	1.1102e-15	4.3503e-07	1.9943e+07	
3	y	`RKZ3` (Kutta)	3.3323e-15	1.9516e-15	2.5220e-15	3.1670e+07	1.2832e+07
	y	`semiexpl3` (Kutta)	6.0026e-15	7.5495e-15	1.3101e-14	1.2083e+07	
	y	`semiexpl3`[c] (Kutta)	2.4649e-15	8.8818e-16	4.4409e-16	1.2083e+07	
	n	$\texttt{RK3}_\texttt{U}$ (Kutta)	3.6071e-05	8.1654e-05	1.3668e-04	1.0282e+07	
	n	$\texttt{RK3}_\texttt{L}$ (Kutta)	4.8405e-05	1.0036e-04	1.6869e-04	9.4628e+06	

Table 5.2: Comparison of results for Problem 2, the three-particle nonperiodic Toda lattice. The stage equations for the implicit schemes are resolved by Picard iterations, with tolerance $\epsilon = 10^{-12}$.

[a] Because of weak orthogonality, see §3.2.2.

[b] Should be read as 'machine accuracy'.

[c] Symmetric, with even-odd steps alternating solvers, see text for details.

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]	$\texttt{RK4}_\texttt{U}$ (GL)	3.2321e-12	5.9521e-12	1.6911e-11	3.6038e-12	2.7416e+08
	n	$\texttt{RK4}_\texttt{L}$ (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	`semiexpl3` (Kutta)	3.2704e-05	3.7659e-13	5.7687e-13	1.3256e-13	5.5261e+07
	y	`semiexpl3`[b] (Kutta)	1.2568e-05	3.4728e-13	5.0049e-13	1.2440e-13	5.5101e+07
	n	$\texttt{RK3}_\texttt{U}$ (Kutta)	4.2290e-04	1.1221e-03	2.1980e-03	4.2227e-04	4.8472e+07
	n	$\texttt{RK3}_\texttt{L}$ (Kutta)	8.2909e-04	2.1957e-03	4.0996e-03	9.7449e-04	4.5333e+07

Table 5.3: 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 §3.2.2.

[b] Symmetric, with even-odd steps alternating solvers, see text for details.

Table 5.3 indicates that isospectral methods are generally more accurate than nonisospectral schemes, except for the semi-explicit schemes, that display a larger global error instead. This is due to the loss of symmetry associated with semi-explicit methods and the global error can be reduced by means of a symmetrising procedure. 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 $\texttt{RK4}_\texttt{L}$ and $\texttt{RK3}_\texttt{L}$.

In Table 5.4 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 (1996). 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`, which is isospectral but more expensive, and of the scheme $\texttt{RK3}_\texttt{L}$, which is non-isospectral but cheaper, we display a plot of accuracy versus floating point operations. In Figure 5.23 we plot $e(\boldsymbol{\lambda})$ against the cost of the two methods sampled at $T = 1024$ 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

$$L_0 = \operatorname{diag}(\boldsymbol{\lambda}_0),$$

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

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

In Figure 5.25, the starting condition is initialized by means of the procedure `toepinit`. In both cases, it is true that `RKZ3` is more expensive than $\texttt{RK3}_\texttt{L}$ when we compare the cost for a fixed value of the stepsize (see Figures 5.24 and 5.26) but `RKZ3` produces more precise results even with larger stepsizes: even with initialized starting value $\tilde{L}_0$, which is already nearly-Toeplitz, in order to get an accuracy of 10^{-10} on the eigenvalues, we should employ $h = \frac{1}{320}$ for $\texttt{RK3}_\texttt{L}$ while $h = \frac{1}{10}$ for `RKZ3` does just as well.

Order	Iso	Method	T_{stop}	Toeplitz error	$e(\boldsymbol{\lambda})$	Flops
4	y	`MagnusZ4` (GL)	104.2	9.9008e-11	8.2242e-14	4.5370e+07
	y	`RKMK4` (GL)	104.2	9.9008e-11	1.6803e-13	6.3683e+07
	y(n)[a]	$\texttt{RK4}_{\texttt{U}}$ (GL)	104.2	9.9006e-11	1.3824e-13	3.9613e+07
	n	$\texttt{RK4}_{\texttt{L}}$ (GL)	104.2	9.9005e-11	1.4090e-13	3.6122e+07
3	y	`RKZ3` (Kutta)	104.2	9.9003e-11	6.7646e-14	1.5189e+07
	y	`semiexpl3` (Kutta)	104.2	9.9312e-11	2.4614e-14	7.5018e+06
	y	`semiexpl3`[b] (Kutta)	104.2	9.9124e-11	2.2800e-14	7.4772e+06
	n	$\texttt{RK3}_{\texttt{U}}$ (Kutta)	104.2	9.9003e-11	3.0478e-09	6.8462e+06
	n	$\texttt{RK3}_{\texttt{L}}$ (Kutta)	104.2	9.9004e-11	5.4477e-09	6.3565e+06

Table 5.4: 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 $\epsilon = 10^{-12}$.

[a] Because of weak orthogonality, see §3.2.2.

[b] Symmetric, with even-odd steps alternating solvers. See text for details.

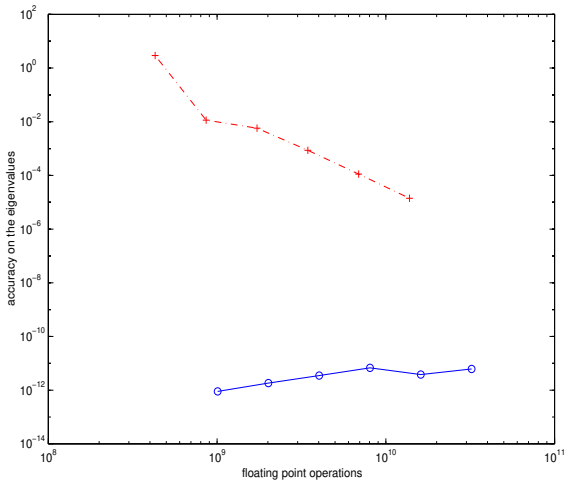


Figure 5.23: Accuracy on the eigenvalues $e(\boldsymbol{\lambda})$ versus floating point operations for the methods `RKZ3` (blue solid line) and $\texttt{RK3}_\texttt{L}$ (red dash-dotted line) at $T = 1024$ for a 10×10 ITEP with diagonal L_0. The ovals and the plus mark the stepsize $h = \frac{1}{10}, \frac{1}{20}, \ldots, \frac{1}{320}$.

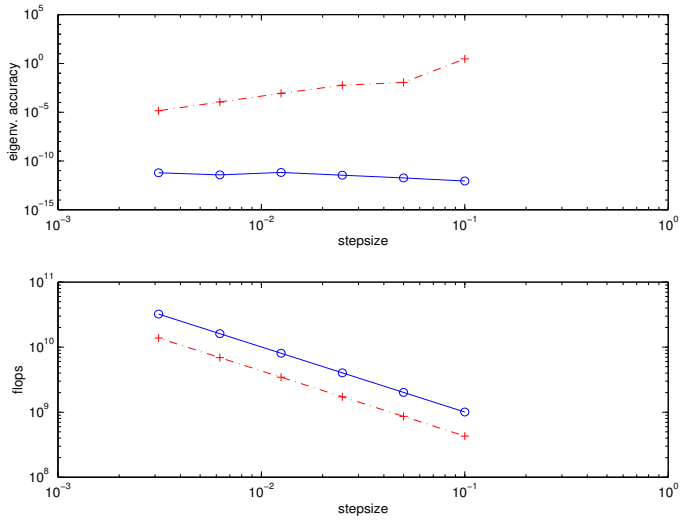


Figure 5.24: Accuracy on the eigenvalues $e(\boldsymbol{\lambda})$ (top) and floating point operations (bottom) versus stepsize for the methods `RKZ3` (blue solid line) and $\texttt{RK3}_\texttt{L}$ (red dash-dotted line) for a 10×10 ITEP problem with diagonal L_0. The ovals and the plus mark the stepsize $h = \frac{1}{10}, \frac{1}{20}, \ldots, \frac{1}{320}$.

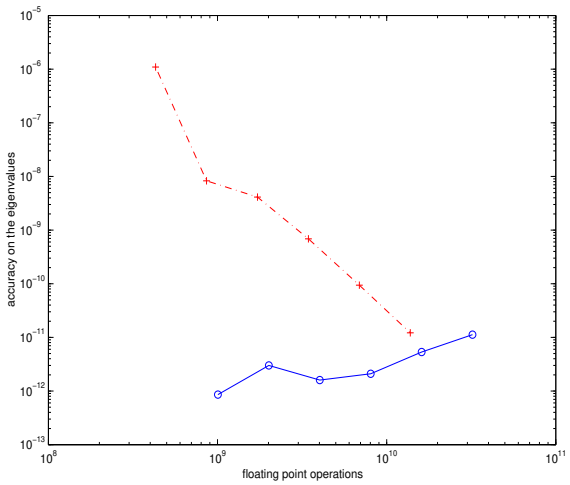


Figure 5.25: Accuracy on the eigenvalues $e(\boldsymbol{\lambda})$ versus floating point operations for the methods `RKZ3` (blue solid line) and $\texttt{RK3}_\texttt{L}$ (red dash-dotted line) at $T = 1024$ for a 10×10 ITEP with initialized starting condition $\tilde{L}_0$. The ovals and the plus mark the stepsize $h = \frac{1}{10}, \frac{1}{20}, \ldots, \frac{1}{320}$.

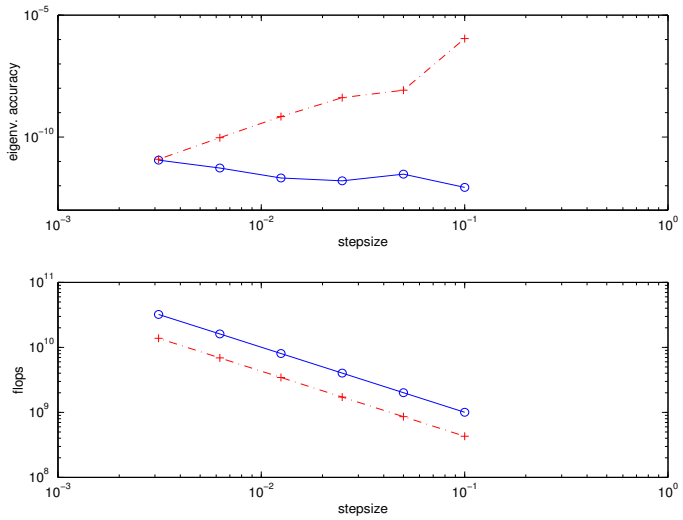


Figure 5.26: Accuracy on the eigenvalues $e(\boldsymbol{\lambda})$ (top) and floating point operations (bottom) versus stepsize for the methods `RKZ3` (blue solid line) and $\texttt{RK3}_\texttt{L}$ (red dash-dotted line) for a 10×10 ITEP problem with initialized staring condition $\tilde{L}_0$. The ovals and the plus mark the stepsize $h = \frac{1}{10}, \frac{1}{20}, \ldots, \frac{1}{320}$.

6

Conclusions

In this thesis we have considered the problem of solving numerically a class of matrix ordinary differential equations of the form

$$L' = [B(L), L], \qquad L(0) = L_0, \quad L \in \mathbb{R}^{d\times d},$$

known as *isospectral flows,* since the spectrum of the initial matrix L_0 is retained by the solution L. Such flows arise in a number of applications, ranging from Hamiltonian dynamics to well-known matrix algorithms in numerical analysis. In Chapter 2 we have proved that most classical ODE methods cannot be isospectral and their error on the eigenvalues is generally of the same order of magnitude as the local truncation error of the method itself. Isospectrality is the main feature of these flows: for Hamiltonian problems it is related to integrability, for flows that arise in a linear algebra context it is related to eigenvalues or given data whose accuracy one would like to preserve. Therefore it is of substantial importance to be able to devise numerical methods that share this feature. To this end, in Chapter 3 we have taken advantage of the fact that isospectrality can be obtained by means of the group action of the set of orthogonal matrices on the set of all matrices with prescribed eigenvalues. In the third chapter we have also presented a number of numerical schemes that preserve isospectrality. Among others, those based on Runge–Kutta Gauss–Legendre methods, which are weakly orthogonal, and semi-explicit schemes, which, instead of employing the orthogonal group action, use the action of $\mathrm{GL}(d)$.

Motivated by the fact that isospectrality is equivalent to a group action, in Chapter 4 we have considered the more general task of devising numerical methods for Lie groups. Though introduced in a matrix setting, the methods can be applied to general Lie groups, provided they are appropriately modified. Departing from two Lie-group and Lie-algebra expansions of the exact solution, namely the Fer and the Magnus series, we have introduced a class of collocation schemes whose coefficients are obtained by integrating the Lagrangian cardinal polynomials along the trees that feature in the underlying expansion. The drawback of the collocation approach is that it yields implicit numerical methods. In order to reduce the implicitness of the schemes, we have relaxed the collocation conditions and presented the general recipe to create explicit order-three schemes, our point of departure being the coefficients of a classical order-three Runge–Kutta method.

Lie-group methods are geometric integrators since they fully exploit the geometry of the underlying differential equations. The interest in Lie-group method is twofold: an interest in the methods *per sé* and because Lie groups arise in a number of applications such as *homogeneous manifolds,* whereby the 'action' is dictated by the underlying Lie group (Munthe-Kaas & Zanna 1997). Because of this, we believe that Lie-group methods will play

an important role in future years. Yet, there are still a number of issues to be addressed and we sketch here a few ideas for future development of the methods presented. First of all, there are a number of technical details to pursue, like for instance, understanding the least number of trees for a Fer or Magnus expansion of a given order. We know that the relations imposed by the order trees are not all independent (Iserles & Nørsett n.d.), and reducing the number of trees means reducing the cost of the methods, since one has to evaluate less commutators. This problem has been attacked by Munthe-Kaas & Owren (1998), while Iserles (1998) has presented some asymptotic results on the number of trees.

Another relevant issue is related to the numerical implementation of the proposed Lie-group methods. Error-control algorithms have been proposed for linear problems only (Iserles, Marthinsen & Nørsett 1998) and not much in this direction has been done for the nonlinear case. Furthermore, in the numerical experiments proposed in this thesis, we have used 'general purpose' algorithms to evaluate exponentials and commutators, which actually do not exploit any of the features of the underlying Lie groups. We believe that specialized algorithms could yield substantial savings — and better performances — in the numerical implementation of the schemes. Yet, the schemes would have enough generality to be applied to a variety of problems in the same class. Some steps in this direction have already been taken by Celledoni & Iserles (1998) and by the *DiffMan* project,[1] which is an object-oriented MATLAB™ toolbox for solving ordinary differential equations on manifolds.

[1]Written by Kenth Engø, Arne Marthinsen and Hans Munthe-Kaas. Visit the home-page of the project at the URL `http://www.math.ntnu.no/num/diffman/`

Bibliography

Abramowitz, M. & Stegun, I. A. (1965), *Handbook of Mathematical Functions*, Dover, New York.

Baker, G. A. (1975), *Essentials of Padé Approximants*, Academic Press, New York.

Baker, H. F. (1905), 'Alternants and continuous groups', *Proceedings of the London Mathematical Society* **3**, 24–47.

Berry, M. V. (1987), Regular and irregular motion, *in* R. MacKay & J. Meiss, eds, 'Hamiltonian Dynamical Systems', Adam Hilger, Bristol, pp. 27–53.

Bloch, A., ed. (1994), *Hamiltonian and Gradient Flows, Algorithms and Control*, Vol. 3 of *Fields Institute Communications*, Amarican Mathematical Society, Providence.

Brockett, R. W. (1991), 'Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems', *Lin. Alg. Applic.* **146**, 79–91.

Brockett, R. W. (1993), Differential geometry and the design of gradient algorithm, *in* 'Proceedings of Symposia in Pure Mathematics', Vol. 54, AMS, pp. 69–92.

Brunet, J.-P. (1994), Integration and visualization of matrix orbits on the connection machine, *in* A. Bloch, ed., 'Hamiltonian and Gradient Flows, Algorithms and Control', Vol. 3 of *Fields Institute Communications*, American Mathematical Society, Providence, pp. 77–86.

Bryant, R. L. (1995), An Introduction to Lie Groups and Symplectic Geometry, *in* D. S. Freed & K. K. Uhlenbeck, eds, 'Geometry and Quantum Field Theory', second edn, Vol. 1 of *IAS/Park City Mathematics Series*, American Mathematical Society.

Butcher, J. C. (1975), 'A-stability properties of Runge–Kutta methods', *BIT* **45**, 358–361.

Calogero, F. (1975), 'Exactly solvable one-dimensional many-body problems', *Lett. Nuovo Cim.* **13**, 411–416.

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

Calvo, M. P., Iserles, A. & Zanna, A. (1996), Runge–Kutta methods on manifolds, *in* D. F. Griffiths & G. A. Watson, eds, 'A. R. Mitchell's 75th Birthday Volume', World Scientific, Singapore, pp. 57–70.

Calvo, M. P., Iserles, A. & Zanna, A. (1997), Semi-explicit methods for isospectral flows, Technical Report NA1997/16, DAMTP, University of Cambridge.

Cano, B. & Sanz-Serna, J. M. (1996), Error Growth in the Numerical Integration of Periodic Orbits by Multistep Methods, with Application to Reversible Systems, Technical Report 1996/5, Departamento de Matemática Aplicada y Computación, Universidad de Valladolid, Spain.

Casas, F. (1996), 'Solution of Linear Partial Differential Equations by Lie Algebraic Methods', *J. Comput. Appl. Math* **76**, 159–170.

Celledoni, E. & Iserles, A. (1998), Approximating the exponential from a Lie algebra to a Lie group, Technical Report NA1998/03, DAMTP, University of Cambridge.

Chu, M. T. (1984), 'The generalized Toda flow, the QR algorithm and the center manifold theory', *SIAM J. Alg. Discr. Meth.* **5**, 187–201.

Chu, M. T. (1986), 'A differential equation approach to the singular value decomposition of bidiagonal matrices', *Lin. Alg. Appl.* **80**, 71–80.

Chu, M. T. (1988), 'On the continuous realization problem of interative processes', *SIAM Rev.* **30**(3), 375–389.

Chu, M. T. (1994), A List of Matrix Flows with Applications, *in* A. Bloch, ed., 'Hamiltonian ang Gradient Flows, Algorithms and Control', Vol. 3 of *Fields Institute Communications*, American Mathematical Society, Providence, pp. 194–213.

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

Cooper, G. J. (1987), 'Stability of Runge–Kutta methods for trajectory problems', *IMA J. Numer. Anal.* **7**, 1–13.

Crouch, P. E. & Grossman, R. (1993), 'Numerical Integration of Ordinay Differential Equations on Manifolds', *J. Nonlinear Sci.* **3**, 1–33.

Deift, P. A., Rivera, S., Tomei, C. & Watkins, D. S. (1990), 'A monotonicity property for Toda-type flows', *SIAM J. Num. Anal.* **27**, 463–468.

Deift, P., Demmel, J., Li, L.-C. & Tomei, C. (1991), 'The bidiagonal singular value decomposition and Hamiltonian mechanics', *SIAM J. Num. Anal.* **28**(5), 1463–1516.

Deift, P., Nanda, T. & Tomei, C. (1983), 'Ordinary differential equations and the symmetric eigenvalue problem', *SIAM J. Num. Anal.* **20**, 1–22.

Delsarte, P. & Genin, Y. (1984), Spectral properties of finite Toeplitz matrices, *in* 'Lecture Notes in Control and Inform. Sci.', Vol. 58, Springer, New York, pp. 194–213.

Dieci, L. & van Vleck, E. S. (1996), Computation of orthonormal factors for fundamental solution matrices, Technical report, Georgia Tech.

Dieci, L., Russel, R. D. & van Vleck, E. S. (1994), 'Unitary Integrators and Applications to Continuous Orthonormalization Techniques', *SIAM J. Numer. Anal.* **31, No. 1**, 261–281.

Diele, F., Lopez, L. & Peluso, R. (1996), The Cayley transform in the numerical solution of unitary differential systems, Technical report, Dipartimento di Matematica, Universitá di Bari.

Eirola, T. & Sanz-Serna, J. M. (1992), 'Conservation of integrals and symplectic structure in the integration of differential equations by multistep methods', *Numer. Math.* **61**, 281–290.

Estep, D. J. & Stuart, A. M. (1995), 'The rate of error growth in Hamiltonian-conserving integrators', *Z. Angew. Math. Phys.*

Fairlie, D. B. & Mulvey, J. A. (1994), 'Integrable generalizations of the 2-dimensional Born–Infeld equation', *J. Phys. A* **27**, 1317–1324.

Feng, K. & Shang, Z.-J. (1995), 'Volume-preserving algorithms or source-free dynamical systems', *Numer. Math.* **71**, 451–463.

Fer, F. (1958), '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.

Flaschka, H. (1974), 'The Toda lattice I', *Phys. Rev.* **B9**, 1924–1925.

Friedland, S., Nocedal, J. & Overton, M. L. (1987), 'The formulationa and analysis of numerical methods for inverse eigenvalue problems', *SIAM J. Numer. Anal.* **24**, 634–667.

Gallot, S., Hulin, D. & Lafontaine, J. (1990), *Riemaniann Geometry*, second edn, Springer-Verlag, Berlin.

Gantmacher, F. R. (1977), *The Theory of Matrices*, Chelsea, New York.

Golub, G. H. & van Loan, C. F. (1989), *Matrix Computations*, 2nd edn, John Hopkins, Baltimore.

Hairer, E. & Stoffer, D. (1997), 'Reversible long-time integration with variable step sizes', *SIAM J. Sci. Comput.* **18**(1), 257–269.

Hairer, E., Nørsett, S. P. & Wanner, G. (1993), *Solving Ordinary Differential Equations I. Nonstiff Problems*, 2nd revised edn, Springer-Verlag, Berlin.

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

Helmke, U. & Moore, J. B. (1990), 'Singular value decomposition via gradient flows', *Systems & Control Lett.* **14**, 313–326.

Higham, D. J. (1996), 'Runge–Kutta Type Methods for Orthogonal Integration', *Appl. Num. Math.* **22**, 217–223.

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

Higham, N. J. (1989), Matrix nearness problems and applications, *in* M. J. C. Gover & S. Barnett, eds, 'Application of Matrix Theory', Oxford University Press, pp. 1–27.

Iserles, A. (1984), 'Solving Linear Ordinary Differential Equations by Exponentials of Iterated Commutators', *Numer. Math.* **45**, 183–199.

Iserles, A. (1996), *A First Course in Numerical Analysis of Differential Equations*, Cambridge University Press, Cambridge.

Iserles, A. (1997), Multistep methods on manifolds, Technical Report NA1997/13, DAMTP, University of Cambridge.

Iserles, A. (1998), In preparation.

Iserles, A. & Nørsett, S. P. (1997), On the solution of differential equations in Lie groups, Technical Report NA1997/03, DAMTP, University of Cambridge.

Iserles, A. & Nørsett, S. P. (n.d.). personal communication.

Iserles, A., Marthinsen, A. & Nørsett, S. P. (1998), On the implementation of the method of Magnus series for linear differential equations, Technical Report NA1998/02, DAMTP, University of Cambridge.

Iserles, A., McLachlan, R. & Zanna, A. (1997), Approximately preserving symmetries in numerical integration, Technical Report NA1997/22, DAMTP, University of Cambridge.

Itoh, T. & Abe, K. (1988), 'Hamiltonian-conserving discrete canonical equations based on variational difference quotients', *J. Comput. Phys.* **76**, 85–102.

Kac, M. & van Moerbeke, P. (1975), 'A complete solution of the periodic Toda problem', *Proc. Nat. Acad. Sci. USA* **72**, 2878–2880.

Lagarias, J. (1990), 'Monotonicity properties of the Toda flow, the QR flow and subspace iterations', *SIAM J. Num. Anal.* **27**, 449–462.

Lambert, J. D. (1991), *Numerical Methods for Ordinary Differential Systems*, John Wiley & sons, Chichester.

Landau, H. J. (1994), 'The inverse eigenvalue problem for real symmetric Toeplitz matrices', *J. Amer. Math. Society* **7**, 749–767.

Laurie, D. P. (1991), 'Solving the inverse eigenvalue problem via the eigenvector matrix', *J. Comput. Appl. Math.* **35**, 277–289.

Laurie, D. P. (1996). personal communication.

Lax, P. (1968), 'Integrals of nonlinear evolution equations and solitary waves', *Comm. Pure Appl. Math.* **147**, 467–490.

Lewis, D. & Simo, J. C. (1995), 'Conserving Algorithms for the Dynamics of Hamiltonian Systems of Lie Groups', *J. Nonlinear Sci.* **4**, 253–299.

MacKay, R. S. (n.d.). personal communication.

Magnus, W. (1954), 'On the Exponential Solution of Differential Equations for a Linear Operator', *Comm. Pure and Appl. Math.* **VII**, 649–673.

Marsden, J. E. & Ratiu, T. S. (1994), *Introduction to Mechanics and Symmetry*, Springer-Verlag.

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

MATLAB User's Guide (1992).

Moler, C. B. & van Loan, C. F. (1978), 'Nineteen dubious ways to compute the exponential of a matrix', *SIAM Rev.* pp. 801–836.

Moore, J. B., Mahoney, R. E. & Helmke, U. (1994), 'Numerical gradient algorithms for eigenvalues and singular value calculation', *SIAM J. Matrix Anal. Applic.* **17**, 881–902.

Moser, J. (1973), *Stable and Random Motion in Dynamical Systems*, Princeton University Press.

Moser, J. (1975*a*), Finitely many mass points on the line under the influence of an exponential potential – An integrable system, *in* J. Moser, ed., 'Dynamical Systems Theory and Applications', Springer-Verlag, New York, pp. 467–497.

Moser, J. (1975*b*), 'Three integrable Hamiltonian systems connected with isospectral deformations', *Adv. Math.* **16**, 197–220.

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

Munthe-Kaas, H. (1998), 'Runge–Kutta Methods on Lie Groups', *BIT* **38**, 92–111.

Munthe-Kaas, H. & Owren, B. (1998), Computations in a free Lie algebra, in preparation.

Munthe-Kaas, H. & Zanna, A. (1997), Numerical integration of differential equations on homogeneous manifolds, *in* F. Cucker, ed., 'Foundations of Computational Mathematics', Springer Verlag, pp. 305–315.

Munthe-Kaas, H. & Zanna, A. (n.d.), Notes on semidirect products of Lie groups and applications, In preparation.

Nanda, T. (1982), PhD thesis, New York University, New York.

Nørsett, S. P. & Wanner, G. (1981), 'Perturbed Collocation and Runge–Kutta Methods', *Numer. Math.* **38**, 193–208.

Olver, P. J. (1993), *Applications of Lie Groups to Differential Equations*, GTM 107, Second edn, Springer-Verlag.

O'Neill, B. (1966), *Elementary Differential Geometry*, Academic Press, New York.

Owren, B. & Marthinsen, A. (1997), Integration Methods Based on Rigid Frames, Technical Report Numerics No. 1/1997, Department of Mathematical Sciences, The Norwegian University of Science and Technology.

Owren, B. & Welfert, B. (1996), The Newton Iteration on Lie Groups, Technical Report Numerics No. 3/1996, NTNU, Trondheim, Norway.

Perelomov, A. M. (1990), *Integrable Systems of Classical Mechanics and Lie Algebras*, Birkhäuser, Basel.

Quispel, G. R. W. & Chapel, H. W. (1997), Solving ODE's numerically while preserving all first integrals, Technical report, La Trobe University.

Rainville, D. (1967), *Special Functions*, Macmillan, New York.

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

Sanz-Serna, J. M. & Calvo, M. P. (1994), *Numerical Hamiltonian Problems*, AMMC 7, Chapman & Hall.

Simo, J. C. & Wong, K. K. (1991), 'Unconditionally Stable Algorithms for Rigid Body Dynamics that Exactly Preserve Energy and Momentum', *Internat. J. Numer. Methods Engrg.* **31**, 19–52.

Smith, S. T. (1991), 'Dynamical systems that perform the singular value decomposition', *Systems & Control Letters* **16**, 319–327.

Stuart, A. M. & Humphries, A. R. (1996), *Dynamical Systems and Numerical Analysis*, Cambridge University Press, Cambridge.

Symes, W. W. (1982), 'The QR algorithm and scattering for the finite nonperiodic Toda lattice', *Phys. D* **4**, 275–280.

Toda, M. (1981), *Theory of Nonlinear Lattices*, Springer-Verlag, Berlin.

Toda, M. (1989), *Nonlinear Waves and Solitons*, Mathematics and Its Applications (Japanese Series), Kluwer Academic Publisher, Dordrecht.

Varadarajan, V. S. (1984), *Lie Groups, Lie Algebras, and Their Representation*, GTM 102, Springer-Verlag.

Watkins, D. S. (1982), Experience with the Toda flow method of calculating eigenvalues, Technical report, Washinton State University.

Watkins, D. S. (1984), 'Isospectral flows', *SIAM Review* **26**, 379–391.

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

Zanna, A. (1997), Collocation and relaxed collocation for the Fer and the Magnus expansions, Technical Report NA1997/17, DAMTP, University of Cambridge.

Zanna, A. & Munthe-Kaas, H. (1997), Iterated commutators, Lie's reduction method and ordinary differential equations on matrix Lie groups, *in* F. Cucker, ed., 'Foundation of Computational Mathematics', Springer-Verlag, pp. 434–441.