REPORTS
IN
INFORMATICS

ISSN 0333-3590

Numerical integration of Lie–Poisson systems while preserving coadjoint orbits and energy

Kenth Engø and Stig Faltinsen

REPORT NO 179 **October 1999**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL `http://www.ii.uib.no/publikasjoner/texrap/ps/1999-179.ps`

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Numerical integration of Lie–Poisson systems while preserving coadjoint orbits and energy

Kenth Engø*
Department of Informatics
University of Bergen
N-5020 Bergen, Norway

Stig Faltinsen†
Department of Applied Mathematics
and Theoretical Physics
Silver Street, Cambridge
England CB3 9EW

This version: October 29, 1999

Abstract

In this paper we apply geometric integrators of the RKMK type to the problem of integrating Lie–Poisson systems numerically. By using the coadjoint action of the Lie group G on the dual Lie algebra $\mathfrak{g}^*$ to advance the numerical flow, we devise methods of arbitrary order that automatically stay on the coadjoint orbits. First integrals known as Casimirs are retained to machine accuracy by the numerical algorithm. Within the proposed class of methods we find integrators that also conserve the energy. These schemes are implicit and of second order. Nonlinear iteration in the Lie algebra and linear error growth of the global error are discussed. Numerical experiments with the rigid body, the heavy top and a finite–dimensional truncation of the Euler equations for a 2D incompressible fluid are used to illustrate the properties of the algorithm.

AMS Subject Classification: 65L06, 34A50, 34A26, 70H99

Key Words: Lie–Poisson systems, geometric integration, energy conserving, coadjoint action, coadjoint orbits, Casimir functions, numerical analysis, Lie groups, Lie algebras

1 Introduction

Hamiltonian problems are ubiquitous in modern applied mathematics, physics, and chemistry. The dynamics of a vast number of problems can be phrased in the classical form of Hamilton's equations. The study of the numerical solution of classical Hamiltonian problems are well understood and symplectic and energy preserving algorithms demonstrate good intrinsic properties [23].

However, large classes of problems can be cast in a form that constitutes a natural generalization of Hamiltonian formalism. The dynamical equations are then stated in non–canonical coordinates,

*Email: `Kenth.Engo@ii.uib.no`
†Email: `S.Faltinsen@damtp.cam.ac.uk`

and classical approaches seem insufficient to capture the additional rich geometric structure. This generalization of the Hamiltonian formalism is usually referred to as the Lie–Poisson formalism.

Applications that fit into the Lie–Poisson formalism are numerous: mechanical systems such as the Euler equations and the heavy top equations, problems in optics described by the three-wave equations and Maxwell–Bloch equations. Likewise, the equations of motion for the interaction of stars, the Vlasov–Poisson equations, and also the Euler equations in hydrodynamics describing inviscid, incompressible fluid flow are all Lie–Poisson systems. The list of examples extends beyond the above and includes many more interesting problems formulated both as ordinary differential equations (ODEs) and partial differential equations (PDEs) [1, 15]. This paper aims to present geometric integrators of the Runge–Kutta Munthe-Kaas (RKMK) type [20] that render correctly several of the analytic features of the exact solution of Lie–Poisson systems.

The Lie–Poisson equations are naturally formulated in the dual space of a Lie algebra $\mathfrak{g}$. The solutions are constrained to nonlinear submanifolds of $\mathfrak{g}^*$ better known as group coadjoint orbits. By using the coadjoint action of the Lie group G on the dual Lie algebra $\mathfrak{g}^*$ to advance the numerical flow, we devise methods of arbitrary order that preserve the coadjoint orbits. Moreover, first integrals known as Casimirs are retained to machine accuracy by the numerical flow. When it comes to energy and symplectic (or Lie–Poisson) structure, it seems clear that a choice has to be made. A corollary of the result of Ge and Marsden in [9] yields that in general (subject to certain conditions that are made clear in the text) the proposed numerical algorithm cannot simultaneously preserve both the energy and the Lie–Poisson structure.

Several authors have successfully constructed integrators that are Lie–Poisson. Generating functions were used in [9], McLachlan constructed first-order explicit integrators in [17] based on splitting, and Reich considered numerical methods for the unreduced equations in [22]. In this paper we show how to make energy preserving schemes using discrete gradients as introduced by Gonzalez [10]. The proposed algorithm preserve coadjoint orbits, Casimirs, and energy, and is applicable to any Lie–Poisson system. The proposed algorithm is implicit and of second order.

The paper is organized as follows. Section 2 discusses in some more detail the Lie–Poisson equations on the dual of a Lie algebra. This is followed by a discussion on conservation laws present in Lie–Poisson systems in Section 3. The algorithm preserving coadjoint orbits, Casimirs and energy is the topic of Sections 4 and 5. Numerical experiments are included in Section 6, and the paper is concluded in Section 7.

2 The Lie–Poisson equations

All time-dependent physical phenomena where dissipation is absent or can be ignored can be modelled as a Hamiltonian system. The classical form of Hamilton's equations in canonical coordinates on the phase space $\mathbb{R}^{2n}$ is

$$\begin{aligned} \frac{dp}{dt} &= -\frac{\partial H}{\partial q} \\ \frac{dq}{dt} &= \frac{\partial H}{\partial p} \end{aligned}$$

where $H = H(p, q)$ and $p, q \in \mathbb{R}^n$. Alternatively, if we set $y = (p; q) \in \mathbb{R}^{2n}$ then (2) can be rephrased as

$$\frac{dy}{dt} = J\nabla H(y), \tag{2.1}$$

where J is the constant, skew-symmetric matrix

$$\begin{bmatrix} 0 & -\mathrm{I}_n \\ \mathrm{I}_n & 0 \end{bmatrix}.$$

There exists a natural symplectic structure on the phase space $\mathbb{R}^{2n}$, given in terms of the canonical two-form

$$\omega(\xi,\eta) = \xi^T\, J^{-1}\eta, \qquad \text{for all } \xi,\ \eta\ \in \mathbb{R}^{2n}. \tag{2.2}$$

In the following, y is to be considered an element in the dual space of a Lie algebra $\mathfrak{g}$. Starting from (2.1) we arrive at the Lie–Poisson equations simply by letting the matrix J depend linearly on y.

$$\frac{dy}{dt} = J(y)\nabla H(y), \qquad y(t_0) = y_0, \tag{2.3}$$

where $J(y)$ is a skew-symmetric matrix defined as

$$\big(J(y)\big)_{i,j} = \sum_{k=1}^{d} C_{i\ j}^{k} y_k,$$

where $y = \sum_{k=1}^{d} y_k e_k$ and $\{e_k\}_{k=1}^{d}$ form a basis of the dual Lie algebra $\mathfrak{g}^*$. The constants $C_{i\ j}^{k}$ are called *structure constants* and they are unique for a given Lie algebra $\mathfrak{g}$ and a choice of basis [25]. The structure constants are defined as $[f_i, f_j] = \sum_{k=1}^{d} C_{i\ j}^{k} f_k$ where $\{f_k\}_{k=1}^{d}$ form a basis for $\mathfrak{g}$ and $[\cdot,\cdot]$ is the *commutator* [25].

The rigid body equation as a Lie–Poisson system. Consider a rigid body with the centre of mass fixed at the origin. Let $y \in \mathbb{R}^3$ describe the angular momenta of the rigid body. The equations of motion are

$$\frac{dy}{dt} = y \times \nabla H(y), \tag{2.4}$$

where $\times$ is the cross product of vectors in $\mathbb{R}^3$. A choice of Hamiltonian function is the kinetic energy of the rigid body:

$$H(y) = \frac{1}{2}\left(\frac{y_1^2}{I_1} + \frac{y_2^2}{I_2} + \frac{y_3^2}{I_3}\right), \quad \text{yielding} \quad \nabla H(y) = \begin{bmatrix} I_1^{-1}y_1 \\ I_2^{-1}y_2 \\ I_3^{-1}y_3 \end{bmatrix} = \mathcal{I}^{-1}y,$$

where I_1, I_2 and I_3 are the principal moments of inertia, and $\mathcal{I}^{-1} = \mathrm{diag}(I_1^{-1}, I_2^{-1}, I_3^{-1})$. The equations of motion for the rigid body can be cast in the Lie–Poisson form (2.3) by choosing

$$J(y) = \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix}.$$

The Hamiltonian equations (2.4) for the rigid body are formulated on the dual Lie algebra $\mathfrak{so}(3)^*$, where $\mathfrak{so}(3)$ is the set of all skew-symmetric three-by-three matrices. A basis for $\mathfrak{so}(3)$ consists of the three matrices

$$f_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \quad f_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \text{ and } f_3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The commutator is the standard matrix commutator, and the non-zero structure constants are $C_{12}^{3} = -C_{21}^{3} = -1$, $C_{13}^{2} = -C_{31}^{2} = 1$ and $C_{23}^{1} = -C_{32}^{1} = -1$. We can identify the space $(\mathfrak{so}(3), [\cdot,\cdot])$ with $(\mathbb{R}^3, \times)$ through the hat map, defined as

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \mapsto \quad \hat{y} = \begin{bmatrix} 0 & -y_3 & y_2 \\ y_3 & 0 & -y_1 \\ -y_2 & y_1 & 0 \end{bmatrix}. \tag{2.5}$$

The isomorphism is characterized by $\hat{y}v = y \times v$, and it is easy to verify that $[\hat{v}, \hat{u}] = \widehat{v \times u}$. For more information about the rigid body, see [15].

The Kirchhoff equations as a Lie–Poisson system. Next, we consider a heavy top modeled on SE(3), the special Euclidean group. The difference from the rigid body case is the presence of a gravitational field that breaks the rotational SO(3) symmetry.

A pair of 3-vectors, $y = (u, v)$, represents the state of the heavy top in local coordinates, where u describes the angular momentum and v is the gravity vector as seen intrinsically on the heavy top.

The equations of motion for the heavy top are given by

$$\frac{du}{dt} = u \times \nabla_u H_1(u) + v \times \nabla_v H_2(v) \tag{2.6}$$

$$\frac{dv}{dt} = v \times \nabla_u H_1(u), \tag{2.7}$$

where $\nabla_u H_1(u), \nabla_v H_2(v) \in \mathbb{R}^3$ are the gradients of the kinetic and potential energy of the Hamiltonian $H(y)$, respectively.

$$H(y) = H(u, v) = \frac{1}{2}\left(\frac{u_1^2}{I_1} + \frac{u_2^2}{I_2} + \frac{u_3^2}{I_3}\right) + Mglv \cdot \boldsymbol{\chi} = H_1(u) + H_2(v), \tag{2.8}$$

where I_1, I_2 and I_3 are the principal moments of inertia, and $\mathcal{I}^{-1} = \mathrm{diag}(I_1^{-1}, I_2^{-1}, I_3^{-1})$. Moreover, $\boldsymbol{\chi}$ is the constant unit vector along the line segment of length l connecting the origin and the center of gravity. M is mass, and g is the gravitational constant. Hence,

$$\nabla_u H_1(u) = \begin{pmatrix} I_1^{-1}u_1 \\ I_2^{-1}u_2 \\ I_3^{-1}u_3 \end{pmatrix} = \mathcal{I}^{-1}u, \qquad \text{and} \quad \nabla_v H_2(v) = Mgl\boldsymbol{\chi}.$$

Now, the equations of motion for the heavy top can be seen to constitute a Lie–Poisson system (2.3), with

$$J(y) = \begin{bmatrix} \hat{u} & \hat{v} \\ \hat{v} & 0_3 \end{bmatrix},$$

and where 0_3 is the 3×3 zero matrix.

The equations of motion for the heavy top are formulated on the dual Lie algebra $\mathfrak{se}(3)^*$, where $\mathfrak{se}(3)$ is the Lie algebra of the special Euclidean group. Given an element $(u, v) \in \mathbb{R}^3 \times \mathbb{R}^3$ we can identify it through the hat map with the element $(\hat{u}, v) \in \mathfrak{se}(3)^*$.

The two dimensional Euler equation for inviscid and incompressible fluid. The equation (2.3) can be viewed either as an ODE or a PDE, and in the latter case the associated Lie algebra is infinite dimensional. In section 6 we describe in detail an example involving the motion of a 2D inviscid and incompressible fluid.

In the case of an infinite dimensional Lie algebra we are from a numerical perspective interested in finding finite-dimensional truncations preserving the original geometric structure. Hence, we would like to approximate the PDE with a system of ODEs retaining the same qualitative behaviour as the original system. More precisely, if $\mathfrak{g}$ is the infinite-dimensional Lie algebra we seek a finite dimensional truncation $\mathfrak{g}^n$ corresponding to a set of ODEs in a Lie–Poisson form. The finite-dimensional Lie algebra $\mathfrak{g}^n$ must converge to $\mathfrak{g}$ as n tends to infinity in an appropriate norm. Examples of such finite dimensional truncations are the Zeitlin truncation of the 2D Euler equations on a torus [27], and the Scovel and Weinstein moment truncation of the Vlasov–Poisson equations of plasma physics [24].

2.1 Coadjoint orbits

Darboux' theorem [25] assures the existence of local canonical coordinates in which we can write the Lie-Poisson equations (2.3) in the form (2.1). Finding these local canonical coordinates is by no means a simple task, and it is preferable to work directly with the equations in a Lie–Poisson form. Another noteworthy point is that even though the dual of a Lie algebra is a linear vector space, the solutions evolve on group coadjoint orbits which are nonlinear submanifolds of $\mathfrak{g}^*$.

As in the canonical setting, there also exists a natural symplectic structure in the Lie–Poisson case. Unfortunately, this structure can be degenerate in the whole space $\mathfrak{g}^*$. It is well known, see [15], that $\mathfrak{g}^*$ is foliated by coadjoint orbits $\mathcal{O}_y$, where

$$\mathcal{O}_y = \{\mathrm{Ad}^*_{g^{-1}} y \mid g \in G\} \subset \mathfrak{g}^*, \tag{2.9}$$

and G is the Lie group associated with $\mathfrak{g}$. The coadjoint action $\mathrm{Ad}^*_{g^{-1}}$ is defined by

$$\langle \mathrm{Ad}^*_{g^{-1}} w, v\rangle = \langle w, \mathrm{Ad}_{g^{-1}} v\rangle, \qquad w \in \mathfrak{g}^*, \text{ and } v \in \mathfrak{g},$$

where $\langle , \rangle$ is a non-degenerate pairing between $\mathfrak{g}^*$ and $\mathfrak{g}$. For matrices the adjoint action is

$$\mathrm{Ad}_{g^{-1}} v = g^{-1} v g. \tag{2.10}$$

Given an initial condition y_0, a solution to the Lie–Poisson equation (2.3) stays on the same coadjoint orbit $\mathcal{O}_{y_0}$ for all time.

Coadjoint orbits for the rigid-body equations. The Lie group SO(3) corresponding to the Lie algebra $\mathfrak{so}(3)$ is the group of all orthogonal matrices with determinant one. We seek the coadjoint action of SO(3) on $\mathbb{R}^3$. Recall that we can identify $\mathfrak{so}(3)$ with $\mathbb{R}^3$ through the isomorphism given by the hat map (2.5). Let $g \in \mathrm{SO}(3)$ and $v, y \in \mathbb{R}^3$, then from (2.10) we have

$$(\mathrm{Ad}_g \hat{y})v = g\hat{y}g^{-1}v = g(y \times g^{-1}v) = gy \times v = \widehat{gy}v.$$

Hence, the adjoint action of SO(3) on $\mathbb{R}^3$ is $\mathrm{Ad}_g y = gy$, and

$$\mathrm{Ad}^*_{g^{-1}} y = gy.$$

Given an initial condition $y_0 \in \mathbb{R}^3$ the coadjoint orbit $\mathcal{O}_{y_0}$ is a sphere of radius $\|y_0\|$.

Coadjoint orbits for the Kirchhoff equations. In the special Euclidean group SE(3), the group multiplication is given by

$$(A_1, a_1) \circ (A_2, a_2) = (A_1A_2, a_1 + A_1a_2).$$

It is worthwhile noticing that the group SE(3) can be realized as a subgroup of SL(4), the special linear group which consists of matrices with determinant one. The identification is given by

$$(A, a) \in \mathrm{SE}(3) \quad \mapsto \quad \begin{bmatrix} A & a \\ 0 & 1 \end{bmatrix} \in \mathrm{SL}(4).$$

Furthermore, if $(u, v) \in \mathbb{R}^3 \times \mathbb{R}^3$ then

$$\begin{bmatrix} \hat{u} & v \\ 0 & 0 \end{bmatrix} \in \mathfrak{sl}(4).$$

We seek the coadjoint action of SE(3) on $\mathbb{R}^3 \times \mathbb{R}^3$. Using (2.10) and the above identification, it is not hard to show that the group adjoint action of SE(3) on $\mathbb{R}^3 \times \mathbb{R}^3$ is

$$\mathrm{Ad}_{(A,a)}(u, v) = (Au, Av + a \times Au).$$

Using these formulae for the Ad operator and the standard pairing of vectors, the group coadjoint action of SE(3) on $\mathbb{R}^3 \times \mathbb{R}^3$ is readily seen to be

$$\mathrm{Ad}^*_{(A,a)^{-1}}(u,v) = (Au + a \times Av, Av).$$

Given an initial condition $y_0 = (u_0, v_0)$ where neither $u_0 = 0$ nor $v_0 = 0$, the norm of v and the dot product $u \cdot v$ are conserved. It can be shown that in this case the coadjoint orbit $\mathcal{O}_{y_0}$ is isomorphic to the tangent bundle of the the sphere of radius $\|v_0\|$. More details can be found in [15, Sect. 14.7].

2.2 The Poisson bracket

By a theorem due to Kirillov, Arnold, Kostant, and Souriau there exists a unique symplectic structure on the coadjoint orbits [15]. In this paper we will not concentrate on the issue of preserving the symplectic structure. For later reference we now define a symplectic diffeomorphism. A mapping $\Phi : \mathfrak{g}^* \to \mathfrak{g}^*$ is *symplectic* or *canonical* if

$$J(\Phi(y)) = D\Phi(y)^T J(y) D\Phi(y), \tag{2.11}$$

where $D\Phi(y)$ is the Jacobian of the map Φ evaluated at the point $y \in \mathfrak{g}^*$. The definition arises naturally from the formulation of the Lie–Poisson equations in terms of Poisson brackets. The Poisson bracket of two smooth functions F and G on the dual Lie algebra $\mathfrak{g}^*$ is defined by

$$\{F, G\}(y) = \nabla F(y)^T J(y) \nabla G(y). \tag{2.12}$$

Using this definition we can write the Lie–Poisson equations (2.3) in the form

$$\frac{dy}{dt} = \{\mathrm{Id}, H\}(y),$$

where Id is the identity function defined by $\mathrm{Id}(y) = y$. In order to keep this exposition as simple as possible we do not enter too far into the Poisson formalism. However, we will exploit it briefly in a later proof.

The Poisson bracket for the rigid body equation. The rigid body bracket is

$$\{f, h\}(y) = y \cdot (\nabla f \times \nabla h). \tag{2.13}$$

The Poisson bracket for the Kirchhoff equation. The equations (2.6) and (2.7) are Hamiltonian in the Lie–Poisson structure on $\mathfrak{se}(3)^*$, where the heavy-top bracket is defined as

$$\{f, h\}(u, v) = u \cdot (\nabla_u f \times \nabla_u h) + v \cdot (\nabla_u f \times \nabla_v h - \nabla_u h \times \nabla_v f). \tag{2.14}$$

To see that this really gives us the same equations we do the following calculations:

$$\begin{aligned}
\frac{d(u,v)}{dt} &= \{\mathrm{Id}, H\}(u,v) \\
&= u \cdot ((\mathrm{Id}, 0) \times \nabla_u H) + v \cdot ((\mathrm{Id}, 0) \times \nabla_v H - \nabla_u H \times (0, \mathrm{Id})) \\
&= (u \times \nabla_u H_1(u), 0) + (v \times \nabla_v H_2(v), 0) - (0, \nabla_u H_1(u) \times v)
\end{aligned}$$

Another way of writing the Lie–Poisson equations is in terms of the coadjoint operator on the dual space of a Lie algebra. The Lie–Poisson equations are the reduced equations of a Hamiltonian problem on a Lie group [15].

$$\frac{dy}{dt} = \pm \mathrm{ad}^*_{\nabla H(y)} y, \tag{2.15}$$

where ad* is the dual of the ad operator, which is defined as the commutator in the Lie algebra. If the Hamiltonian function on the Lie group is left invariant we get the '+', and '−' in case of right invariance. For the rigid body we have $\mathrm{ad}^*_v w = w \times v$, so

$$\frac{dy}{dt} = \mathrm{ad}^*_{\nabla H(y)} y = y \times \nabla H(y).$$

To show that (2.3) and (2.15) are the same in general is not difficult, and will be done later.

3 Conservation laws of Lie–Poisson systems

Geometric integration emphasizes the principle that one ought to retain important qualitative features of the flow in the numerical solution of differential equations [2]. For canonical Hamiltonian systems the two main features of interest are conservation of energy in the case of autonomous Hamiltonians $H = H(y)$, and conservation of the symplectic structure (2.2).

In Lie–Poisson systems the energy is still a conserved quantity. The canonical symplectic structure (2.2) is replaced with the non-canonical Lie–Poisson structure (2.11). Moreover, there are yet two other important features of Lie-Poisson systems which we may want to retain in the numerical approximation namely coadjoint orbits (2.9), and Casimirs.

We define and consider examples of Casimirs in Subsection 3.1. In addition to the above conservation laws, we can still have other first integrals. We will show in Subsection 3.2 that from a numerical point of view there is a dichotomy in the design of the geometric integrators: either we conserve the energy or the symplectic form.

3.1 Casimirs and other first integrals

A function C is a Casimir if the Poisson bracket with an arbitrary smooth function is zero

$$\{C, F\}(y) = \nabla C(y)^T J(y) \nabla F(y) = 0, \quad \forall F \in C^\infty(\mathfrak{g}^*).$$

Casimirs can also be recognized as being constant on the coadjoint orbits. Preserving coadjoint orbits imply the conservation of Casimirs, but the converse is in general not true, see [15, Internet Supplement, N14.A].

A *first integral* is a function F, such that

$$\frac{d}{dt} F(y(t)) = 0$$

along an integral curve $y(t)$ of the differential equation. Another way of characterizing a first integral is in terms of the Poisson bracket: $\{F, H\} = 0$, where H is the Hamiltonian. This follows from the chain rule,

$$0 = \frac{d}{dt} F(y) = \nabla F(y)^T\, y' = \nabla F(y)^T J(y) \nabla H(y) = \{F, H\}(y).$$

Note that a Casimir is a first integral. It is worthwhile noting that Casimirs are constant on the coadjoint orbits [15].

Casimirs for the rigid body. Let $C(y) = G_0(\frac{1}{2}\|y\|^2)$ for an arbitrary function $G_0 : \mathbb{R} \to \mathbb{R}$, then by the chain rule

$$\nabla C(y) = G_0'(\frac{1}{2}\|y\|^2) y,$$

so that $\nabla C(y)^T\, J(y) = -G_0'(\frac{1}{2}\|y\|^2) y \times y = 0$. This is the only Casimir for the rigid body [15].

Casimirs and first integrals for the Kirchhoff equation. There exist two Casimirs for the heavy top: $C_1(u, v) = G_1(\frac{1}{2}\|v\|^2)$ and $C_2(u, v) = G_2(u \cdot v)$, where G_1 and G_2 are arbitrary real functions. We get

$$\nabla C_1(y) = G_1'(\frac{1}{2}\|v\|^2) \begin{bmatrix} 0 \\ v \end{bmatrix}, \quad \text{and} \quad \nabla C_2(y) = G_2'(u \cdot v) \begin{bmatrix} v \\ u \end{bmatrix},$$

so that $\nabla C_i(y)^T\, J(y) = 0$ for $i = 1, 2$. These are all the Casimirs for the heavy top [15].

The Hamiltonian energy and the two Casimirs are of course first integrals. Moreover, the projection of the angular momentum u onto the symmetry axis of the top, $u \cdot \boldsymbol{\chi}$, is also a first integral in case of a rotationally symmetric top!

3.2 Symplecticness versus energy conservation

Optimally, we would like a numerical integrator to preserve all the analytic properties of the exact flow. However, Ge and Marsden showed in [9] that if all the first integrals of Hamilton's equations (2.1) are functions of the energy H, and the numerical method preserves the energy and the symplectic structure, then the numerical solution is a reparametrization in time of the exact flow. A slightly imprecise interpretation of this result is that a numerical method *cannot in general* preserve both the energy and the symplectic structure. This result can be tailored to accommodate our setting. Ge and Marsden suggested themselves that if there are other first integrals one can apply reduction to the exact equation, and likewise to the numerical algorithm. The result would then be valid on the reduced space. It is precisely in this sense that the theorem is applicable to integrable systems and other problems where several first integrals are present.

Corollary 1 (Modification of the Ge–Marsden result)
Assume that all the first integrals restricted to a coadjoint orbit are functions of the energy. Assume also that a numerical integrator that preserves the Casimirs conserves both the energy and the Lie–Poisson structure. Then it is a reparametrization in time of the exact flow.

Proof: The idea is to look at the reduced dynamics on the group coadjoint orbits. We can restrict the Lie–Poisson bracket (2.12) to the orbits. Given an initial condition y_0, the solution is constrained to $\mathcal{O}_{y_0}$. Consider therefore $y \in \mathcal{O}_{y_0}$. We define the restricted bracket $\{\ ,\ \}|_{\mathcal{O}_{y_0}}$ by

$$\{F, H\}(y) = \left\{ F|_{\mathcal{O}_{y_0}}, H|_{\mathcal{O}_{y_0}} \right\}\Big|_{\mathcal{O}_{y_0}} (y),$$

where $H|_{\mathcal{O}_{y_0}}$ is the restriction of the function H to the coadjoint orbit.

Backward error analysis tells us that a Lie-group type numerical method is exponentially close in the step size to a nearby perturbed differential equation with solutions on the same coadjoint orbit $\mathcal{O}_{y_0}$ [7]. Denote the solution of the perturbed differential equation by $\tilde{y}(t)$. Since the numerical integrator is assumed to be Lie–Poisson and preserve Casimirs, the nearby differential equation has the form

$$\tilde{y}'(t) = J(\tilde{y}) \nabla \tilde{H}(\tilde{y}), \qquad \tilde{y}(t_0) = y_0, \tag{3.16}$$

where $\tilde{H} : \mathcal{O}_{y_0} \to \mathbb{R}$ is a Hamiltonian function. The numerical method also conserves the energy H. Since H is a first integral of the exact solution it is also preserved by (3.16) [21].

$$\frac{d}{dt} H(\tilde{y}) = \nabla H(\tilde{y})^T \tilde{y}' = \nabla H(\tilde{y})^T J(\tilde{y}) \nabla \tilde{H}(\tilde{y}) = \left\{ H, \tilde{H} \right\}\Big|_{\mathcal{O}_{y_0}} (\tilde{y}) = 0.$$

From this we can conclude that $\tilde{H}$ is a first integral of the Lie–Poisson equations. Assume that $\tilde{H}$ is a Casimir. Then the function restricted to the coadjoint orbit is constant, and the dynamics of the differential equation (3.16) is trivial. By backward error analysis the numerical method must have zero order. We can therefore discard this trivial case. By assumption there exists now a function $F : \mathbb{R} \to \mathbb{R}$ such that $\tilde{H}(\tilde{y}) = F(H(\tilde{y}))$, and equation (3.16) becomes

$$\tilde{y}'(t) = J(\tilde{y})\nabla\tilde{H}(\tilde{y}) = -F'(H(\tilde{y}(t)))J(\tilde{y})\nabla H(\tilde{y}), \qquad \tilde{y}(t_0) = y_0.$$

Consequently, the numerical method must equal the exact solution up to a reparametrization of time. □

This result does not imply that it is impossible for a numerical integrator to preserve both the energy and the symplectic structure. It asserts that for most systems this is not possible, taking into account that a numerical method is rarely a reparametrization of the exact flow. For the particular example of the unreduced rigid body in 3D space coordinates, Lewis and Simo constructed a numerical integrator that conserves energy, symplectic structure, and momentum [13]. To construct algorithms that do this for general problems seems not possible, according to Corollary 1.

The criterion for a good numerical integrator of Lie–Poisson systems becomes the following. The method should produce numerical approximations that stay on the coadjoint orbits, and conserve the Casimirs. Further, the method should *either* conserve the energy *or* preserve the Lie–Poisson structure.

We present methods that automatically retain the two first features. Within this class there exist integrators that also preserve the energy. Conserving the Lie–Poisson structure is a more delicate problem and will not be treated here.

4 Numerical methods that stay on coadjoint orbits

Knowing that the group coadjoint orbits are conserved quantities of the Lie–Poisson system it is natural to look for numerical integrators of the form

$$y_1 = \mathrm{Ad}^*_{g^{-1}} y_0, \tag{4.17}$$

for some element g in the Lie group G. Interpreting this in the RKMK setting, we have chosen the coadjoint orbits in $\mathfrak{g}^*$ as a homogeneous space acted upon by the group coadjoint action. The RKMK method requires the differential equations to be cast in the form of an infinitesimal generator. As it turns out, the form (2.15) of the Lie–Poisson equations is in fact an infinitesimal generator of (4.17), see [15].

4.1 Algorithms using the exponential mapping

We can represent g locally by $g(t) = \exp(\sigma(t))$, where $\sigma(t) \in \mathfrak{g}$ and exp is the exponential mapping from the Lie algebra $\mathfrak{g}$ to the Lie group G. The next step is then to find a differential equation for $\sigma(t)$. Define the operator dexp by

$$\frac{d}{dt}\exp(\sigma(t)) = \mathrm{dexp}_\sigma(\sigma'(t))\exp(\sigma(t)) = \exp(\sigma(t))\,\mathrm{dexp}_{-\sigma}(\sigma'(t)). \tag{4.18}$$

It can be shown that for sufficiently small $\sigma(t)$ there exists an inverse operator given by

$$\mathrm{dexp}_u^{-1}(v) = \sum_{k=0}^{\infty}\frac{B_k}{k!}\mathrm{ad}_u^k(v), \tag{4.19}$$

where B_k are the Bernoulli numbers, $\mathrm{ad}_u(v)$ is the commutator $[u,v]$, and $\mathrm{ad}_u^k(v)$ is defined iteratively as $\mathrm{ad}_u^k(v) = [u, \mathrm{ad}_u^{k-1}(v)]$ [25]. We can now prove the following theorem.

Theorem 1
Let the solution to the Lie–Poisson equations (2.15) be given by $y(t) = \mathrm{Ad}^*_{\exp(-\sigma(t))} y_0$, *then*

$$\sigma'(t) = \mathrm{dexp}^{-1}_{\sigma(t)} (\nabla H(y)) . \tag{4.20}$$

Proof: Recall that $\langle , \rangle$ is a pairing between the Lie algebra and its dual. By matrix computations, and using (4.18) we obtain

$$\begin{aligned}
\langle y', \nu \rangle &= \frac{d}{dt} \left\langle \mathrm{Ad}^*_{\exp(-\sigma)} y_0, \nu \right\rangle = \frac{d}{dt} \langle y_0, \exp(-\sigma) \nu \exp(\sigma) \rangle \\
&= \langle y_0, - \exp(-\sigma)\, \mathrm{dexp}_\sigma(\sigma') \nu \exp(\sigma) \\
&\qquad + \exp(-\sigma) \nu\, \mathrm{dexp}_\sigma(\sigma') \exp(\sigma) \rangle \\
&= \left\langle y_0, -\mathrm{Ad}_{\exp(-\sigma)} \mathrm{ad}_{\mathrm{dexp}_\sigma(\sigma')}(\nu) \right\rangle \\
&= \left\langle -\mathrm{ad}^*_{\mathrm{dexp}_\sigma(\sigma')}(y), \nu \right\rangle .
\end{aligned}$$

On the other hand, we know from (2.15) that $y' = -\mathrm{ad}^*_{\nabla H(y)}(y)$, which proves the theorem. See [5, 20] for alternative proofs. □

Note 1
If the equation is $y' = +ad^*_{\nabla H(y)}(y)$, *we obtain the same with* $y_1 = Ad^*_{\exp(\sigma)}(y_0)$.

The idea now is to use any Runge–Kutta method to solve equation (4.20), as proposed by Munthe-Kaas [20]. Indeed, any classical numerical one-step method can be used, and even multistep methods are applicable [8]. The simplest example of a one-step method is forward Euler. This is a first order method and no correction from the dexp^{-1} expansion is needed [20]. Hence, the method reads

$$y_1 = \mathrm{Ad}^*_{\exp(-h\nabla H(y_0))} y_0 .$$

Methods corresponding to backward Euler, the trapezoidal rule, and the midpoint rule are

$$y_1 = \mathrm{Ad}^*_{\exp(-h\nabla H(y_1))} y_0, \qquad \text{(Lie backward Euler)}$$

$$y_1 = \mathrm{Ad}^*_{\exp\left(-\frac{h}{2}(\nabla H(y_1)+\nabla H(y_0))\right)} y_0, \qquad \text{(Lie trapezoidal rule)}$$

$$y_1 = \mathrm{Ad}^*_{\exp(-\sigma)} y_0, \text{ where } \sigma = h\nabla H\left(\mathrm{Ad}^*_{\exp(-\sigma/2)} y_0\right) . \qquad \text{(Lie midpoint rule)}$$

Note that the Lie trapezoidal rule is not the same as the Lie midpoint rule, even if ∇H is linear. We will show later that the two methods perform differently with respect to conservation of energy.

In general, the proposed algorithm is as follows:

Algorithm 2 (Numerical integrator)

1. *Calculate the functional derivative* ∇H. *Determine the operators* $Ad^*_{g^{-1}}$ *and* $ad_u(v)$ *for the given problem.*
2. *Truncate the series (4.19) for* dexp^{-1} *to the order of the method minus one [20], and solve equation (4.20) by a Runge–Kutta method. This yields an approximation* $\sigma_n(h, y_n, y_{n+1})$.
3. *Advance the solution on* $\mathfrak{g}^*$ *by*

$$y_{n+1} = Ad^*_{\exp(-\sigma_n(h,y_n,y_{n+1}))} y_n . \tag{4.21}$$

Note that the approximants automatically stay in $\mathfrak{g}^*$, and moreover they belong to the coadjoint orbit of $\mathcal{O}_{y_0}$. The generality of the approach permits us to apply the method to a large class of problems. Examples include the rigid body, the heavy top and the Sine-Euler equation, as we will see in Section 6.

Lemma 1
By construction Algorithm 2 preserves coadjoint orbits and Casimirs of the Lie–Poisson equation (2.3).

For analytical purposes it is often convenient to express the group adjoint operator as an exponential of the ad, that is $\mathrm{Ad}_{\exp(u)} = \exp(\mathrm{ad}_u) = \exp(\hat{u})$. Using this, we can reformulate (4.21) as

$$y_{n+1} = \exp\left(-\mathrm{ad}^*_{\sigma_n}\right)(y_n). \tag{4.22}$$

To avoid confusion on the part of the reader, we remark here that the exponential mapping from a Lie algebra $\mathfrak{g}$ to a Lie group G can take on different forms/representations, depending on the particular group. For all matrix Lie groups the matrix exponential is defined as

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

For instance, for the Lie group SO(3) the matrix exponential can be given in closed form by the Euler–Rodrigues' formula

$$\exp(\hat{v}) = I + \frac{\sin(\|v\|)}{\|v\|}\hat{v} + 2\frac{\sin^2(\|v\|/2)}{\|v\|^2}\hat{v}^2.$$

This formula is applicable in the rigid body computations.

In case of the Lie group SE(3) and the heavy top, we can exploit the identification to a subgroup of SL(4) and show that the matrix exponential can be expressed as

$$\exp(\hat{x}, y) = (\exp(\hat{x}), \mathrm{dexp}_{\hat{x}}(y)),$$

where dexp is implicitly defined in (4.18), and can be expressed in an explicit form as

$$\mathrm{dexp}_u(v) = \frac{\exp(\mathrm{ad}_u) - I}{\mathrm{ad}_u}(v) = \sum_{k=0}^{\infty} \frac{1}{(k+1)!}\mathrm{ad}_u^k(v).$$

4.2 Algorithms that exploit the Cayley map

For the family of quadratic Lie groups [3] we can use the Cayley map to represent group elements locally. We say that a quadratic Lie group can also have Cayley coordinates. This is in addition to the exponential coordinates[1]. Examples of quadratic Lie groups are $\mathrm{SO}(n)$, the special orthogonal group, $\mathrm{Sp}(2n)$, the symplectic group, and $\mathrm{SO}(3,1)$, the Lorentz group. The Cayley transformation is given as:

$$\mathrm{cay}(u) = (1 - u/2)^{-1}(1 + u/2)$$

Using the Cayley transformation, the dexp^{-1} equation in the Lie algebra is substituted with the following equation for dcay^{-1},

$$\mathrm{dcay}_u^{-1} v = v - \frac{1}{2}[u, v] - \frac{1}{4}uvu.$$

[1] For a more detailed discussion on coordinates in the RKMK method, see [5].

The class of integrators

$$y_{n+1} = \mathrm{Ad}^*_{\mathrm{cay}(-\sigma_n)}(y_n), \tag{4.23}$$

where σ_n approximate $\sigma'(t) = \mathrm{dcay}_\sigma^{-1}(\nabla H(y))$ to order p, is a p'th order method that stays on the coadjoint orbits and preserves the Casimirs. Alternatively, we can write the method in the same form as (4.22),

$$y_{n+1} = \mathrm{cay}\left(-\mathrm{ad}^*_{\sigma_n}\right)(y_n). \tag{4.24}$$

Note that if we only want order two, (4.24) and (4.23) are the same. This is a consequence of the fact that $\mathrm{cay}(hu) = \exp(hu) + \mathcal{O}(h^3)$, so that $\mathrm{Ad}^*_{\mathrm{cay}(hu)} = \mathrm{cay}\left(-\mathrm{ad}^*_{hu}\right) + \mathcal{O}(h^3)$.

The use of the Cayley map in numerical algorithms has already been considered in [4, 5, 12, 13, 16]. Moreover, the use of higher order Padé approximations was discussed in [5]. One of the advantages of the Cayley transform is that it is cheaper to evaluate than the exponential map.

5 Energy conservation

Preserving coadjoint orbits is important, but for better long-time behaviour of the numerical method it is also desirable to conserve the energy of the Hamiltonian system [6].

5.1 Almost conservation of energy

In the last section we introduced a class of numerical integrators of arbitrarily high order that preserves coadjoint orbits and Casimirs. How do these methods behave with respect to retaining the Hamiltonian energy? In a majority of the methods the energy drifts. This is the case for instance for the methods based on forward and backward Euler. However, for the selfadjoint Lie-group methods presented in [26], the error in the Hamiltonian energy is bounded.

Applying the backward error analysis results of [7], the Lie-group methods are solving a nearby differential equation with solutions on the same coadjoint orbit. Moreover, the results by Hairer and Stoffer [11] and Reich in [21], combined with backward error analysis on Lie groups, imply that if the original problem is reversible, and the numerical method is reversible, then the perturbed problem is also reversible. KAM theory for reversible systems applied to periodic problems implies that, if the original differential equation has a periodic orbit, so does the perturbed system. Moreover, by a generalized version of the Alekseev–Gröbner theorem [7], it follows that these two orbits are $\mathcal{O}(h^p)$ close, in the sense of Hausdorff distance. Thus, if the energy depends continuously on the orbit we have for all n

$$H(y_n) - H(y_0) = \mathcal{O}(h^p)\,.$$

The nice long-time behaviour of the energy makes these methods attractive. Incidentally, this is a property that is typical of symplectic methods, however, one can show that the selfadjoint Lie group methods are not symplectic. Not conserving the energy does introduce a phase error for periodic problems. We next show how to construct methods that conserve the energy, from the class of methods introduced in Section 4.

5.2 Methods that conserve energy

Discrete gradients were introduced by Gonzalez in [10] to construct energy-conserving schemes for Hamiltonian systems with symmetry. Given a function F, we will in this presentation denote its discrete derivative $\overline{\mathrm{DF}}$ by

$$\overline{\mathrm{DF}}(x,y)\cdot(y-x) = F(y) - F(x),$$

where $\cdot$ is an inner product on $\mathfrak{g}^*$. In [19], McLachlan, Quispel and Robidoux consider a larger class of differential equations, and exploit discrete derivatives to conserve first integrals and Lyapunov functions. It is pointed out in both papers that in the case of Lie–Poisson systems only quadratic coadjoint orbits are conserved. Another drawback of discrete gradients is that we can only construct methods of second order [19]. However, it is possible to increase the order by using composition methods [18]. An important feature of composition methods is that the new higher-order method inherits qualitative features from the base method.

We use discrete derivatives to find a subclass of the methods introduced in Algorithm 2 that are energy conserving, in addition to preserving coadjoint orbits and Casimirs, for both exponential and Cayley coordinates. We seek a solution σ of the differential equation

$$\sigma'(t) = \nabla H(y_n),$$

in the Lie algebra $\mathfrak{g}$. We do not have to correct with higher-order commutator terms because of the order restriction of discrete gradients.

As a matter of fact, there is a whole variety of different actions we can use to stay on coadjoint orbits. Consider an analytic function f from the coadjoint representation of the Lie algebra $\mathfrak{g}$ to the coadjoint representation of the Lie group G. Define the following class of algorithms

$$y_{n+1} = f(-\mathrm{ad}^*_{\sigma_n})y_n, \tag{5.25}$$

where

$$f(x) = \sum_{k=0}^{\infty} c_k\, x^k.$$

For consistency of the method we must have $f(0) = e$ and $f'(0) = I$, where e is the identity element in the coadjoint representation of the Lie group. It follows that $c_0 = 1$. Therefore,

$$y_{n+1} - y_n = \left(\sum_{k=0}^{\infty}(-1)^k c_k\, \mathrm{ad}^{*^k}_{\sigma_n} - I\right) y_n = -\mathrm{ad}^*_{\sigma_n}\left(\sum_{k=1}^{\infty}(-1)^{k-1} c_k\, \mathrm{ad}^{*^{k-1}}_{\sigma_n} y_n\right). \tag{5.26}$$

Algorithms based on exponential and Cayley coordinates alike are clearly in this class.

To prove our main result we need the following lemma.

Lemma 2
For all $\sigma \in \mathfrak{g}$ and $y \in \mathfrak{g}^*$

$$\mathrm{ad}^*_\sigma\, y = -J(y)\,\sigma.$$

Proof: Let $\{e_k\}_{k=1}^d$ and $\{f_k\}_{k=1}^d$ be base of $\mathfrak{g}^*$ and $\mathfrak{g}$, respectively. With $y = \sum_{k=1}^d y_k e_k$ and $\sigma = \sum_{k=1}^d \sigma_k f_k$ we get

$$\big(J(y)\,\sigma\big)_i = \sum_{j=1}^d \left(\sum_{k=1}^d C_{i\,j}^{k} y_k\right)\,\sigma_j.$$

On the other hand,

$$\begin{aligned}
\langle \mathrm{ad}^*_\sigma\, y, f_i\rangle &= \langle y, \mathrm{ad}_\sigma f_i\rangle \\
&= \left\langle \sum_{k=1}^d y_k e_k, \sum_{j=1}^d\sum_{k=1}^d C_{ji}^k \sigma_j\; f_k \right\rangle \\
&= -\sum_{j=1}^d\sum_{k=1}^d C_{ij}^k\,\sigma_j\, y_k.
\end{aligned}$$

□

Theorem 2
Let f be a coordinate map as defined above. The numerical integrator $y_{n+1} = f(-\mathrm{ad}^*_{\sigma_n})y_n$ where $\sigma_n = c(h)\overline{\mathrm{DH}}(y_n, y_{n+1})$ and $c(h)$ is a constant for fixed step size h, preserves energy, coadjoint orbits and Casimirs of the Lie–Poisson equations (2.3).

Proof: Looking at the difference (5.26) of y_{n+1} and y_n, 2 yields

$$y_{n+1} - y_n = J\left(\sum_{k=1}^{\infty}(-1)^{k-1}c_k\, \mathrm{ad}^{*^{k-1}}_{\sigma_n} y_n\right) \sigma_n$$

Thus,

$$\begin{aligned} H(y_{n+1}) - H(y_n) &= \overline{\mathrm{DH}}(y_n, y_{n+1}) \cdot (y_{n+1} - y_n) \\ &= \overline{\mathrm{DH}}(y_n, y_{n+1}) \cdot J\left(\sum_{k=1}^{\infty}(-1)^{k-1}c_k\, \mathrm{ad}^{*^{k-1}}_{\sigma_n} y_n\right) \sigma_n. \end{aligned}$$

Since σ_n is proportional to $\overline{\mathrm{DH}}(y_n, y_{n+1})$ and J is a skew-symmetric matrix, the method is energy conserving. Lemma 1 implies that the coadjoint orbits and the Casimirs are also conserved. □

Energy-conserving schemes for the rigid body. The discrete gradient of the rigid body Hamiltonian is

$$\overline{\mathrm{DH}}(y_n, y_{n+1}) = \frac{1}{2}(\nabla H(y_n) + \nabla H(y_{n+1})) = \frac{1}{2}(I^{-1}y_n + I^{-1}y_{n+1}).$$

Hence, the trapezoidal rule

$$y_{n+1} = f\left(-\mathrm{ad}^*_{\frac{h}{2}(I^{-1}y_n + I^{-1}y_{n+1})}\right) y_n$$

is energy conserving.

A simple calculation shows that the trapezoidal rule is also energy conserving for the heavy top.

5.3 Nonlinear iteration in energy-conserving schemes

Numerical experiments indicate that the energy-conserving property of Algorithm 2 depends crucially on finding the exact solution to the nonlinear equation in the Lie algebra. If the tolerance of the iteration used to find the solution is significantly greater than the machine accuracy the algorithm experiences a linear drift in the energy, and one of the good features of the method is lost. Hence, we are interested in finding efficient ways of speeding up the convergence of the iteration in the Lie algebra. Applying Newton's method is one way of doing this.

We wish to find a root of the equation

$$g(\sigma) = \sigma - \frac{h}{2}\left[\nabla H(y_n) + \nabla H\left(\mathrm{cay}\left(-\mathrm{ad}^*_{\sigma}\right) y_n\right]\right) = 0.$$

The iteration is given by

$$\sigma^{[k+1]} = \sigma^{[k]} - Dg(\sigma^{[k]})^{-1} g(\sigma^{[k]}), \tag{5.27}$$

where it is straight forward to verify that the Jacobian of the function g is

$$Dg(\sigma) \cdot \nu = \left[I - \frac{h}{2}D^2 H\left(\mathrm{cay}\left(-\mathrm{ad}^*_{\sigma}\right) y_n\right)\left(1 + \frac{1}{2}\mathrm{ad}^*_{\sigma}\right)^{-1} J\left(\left(1 + \frac{1}{2}\mathrm{ad}^*_{\sigma}\right)^{-1} y_n\right)\right] \nu. \tag{5.28}$$

A natural choice for initial value of the iteration (5.27) is $\sigma^{[0]} = 0$. Further, exploiting frozen Jacobians instead of evaluating the Jacobian in each iteration, we can replace the Jacobian (5.28) with

$$Dg(\sigma^{[0]}) = I - \frac{h}{2} D^2 H\left(y_n\right) J(y_n),$$

which gives the modified Newton's method:

$$\sigma^{[k+1]} = \sigma^{[k]} - \left(I - \frac{h}{2} D^2 H\left(y_n\right) J(y_n)\right)^{-1} \left(\sigma^{[k]} - \frac{h}{2}\left(\nabla H(y_n) + \nabla H\left(\operatorname{cay}\left(-\operatorname{ad}^*_{\sigma^{[k]}}\right) y_n\right)\right)\right).$$

Similar calculations can also be done for exponential coordinates.

5.4 Linear error growth

Long-time behaviour of the global error is in general better for energy-conserving methods. Estep and Stuart [6] proved that if the period of a periodic solution depends only on the Hamiltonian, the energy-conserving algorithm has a global error that grows linearly in time:

$$\|y_n - y(t_n)\| \leq C t_n h^p,$$

where h is the step size, p is the order of the method and C is a constant. Most other methods display a quadratic growth in the global error.

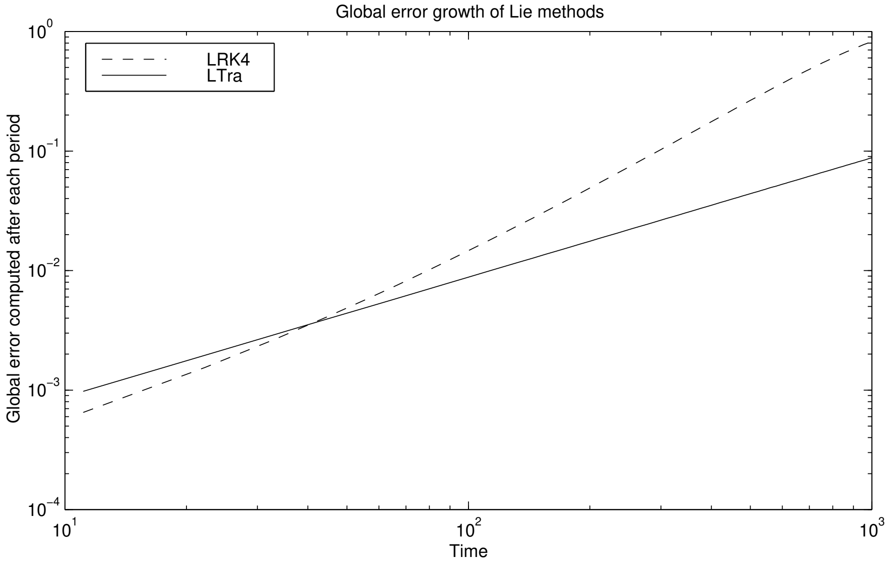

Figure 1: Logarithmic plot of global error versus time for the explicit fourth order Runge–Kutta method (LRK4) and the energy-conserving trapezoidal rule (LTra).

The result of Stuart and Estep is also valid in our setting and just minor changes are needed in order to extend the proof. To illustrate this we have integrated the rigid body problem from $t_0 = 0$ to $t_{\text{end}} = 1000$ with step size $h = 0.1$. In the Lie algebra we used an explicit fourth order Runge–Kutta method (LRK4) and the trapezoidal rule (LTra). The result is plotted in Figure 1. The correlation of $\log(\|y_n - y(t_n)\|)$ with $\log(t)$ is 0.9223 for the trapezoidal rule, and for the fourth order Runge–Kutta method the correlation is 1.6805.

Figure 2 shows the error in the first component of the solution for both numerical methods. The increase in the error for the fourth order method is due to an increase in the phase error.

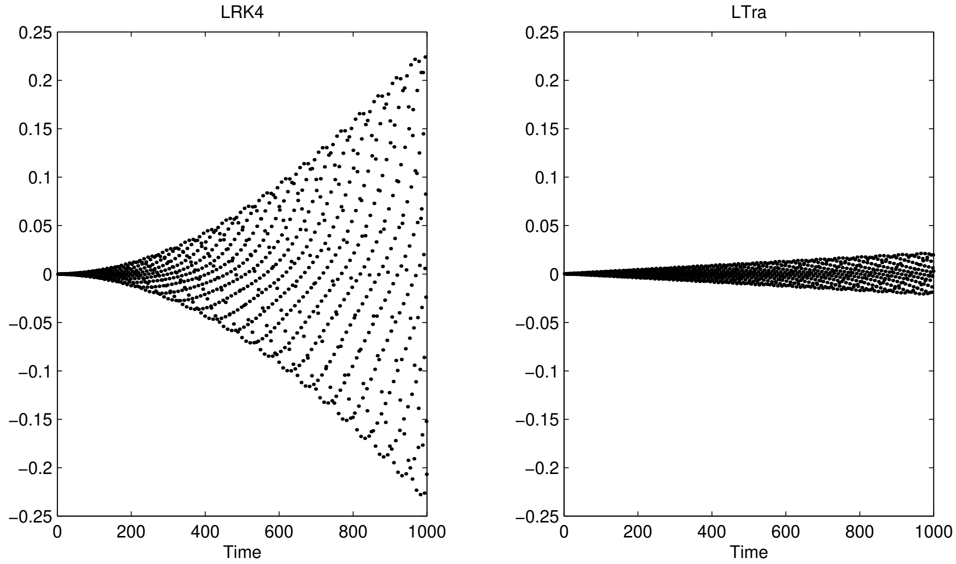

Figure 2: Error in first component of the solution for the explicit fourth order Runge–Kutta method (left) and the trapezoidal rule (right).

6 Numerical examples

We will now illustrate Algorithm 2 with several numerical experiments.

6.1 The rigid body

Recall the Hamiltonian equations for the rigid body with a quadratic Hamiltonian,

$$\frac{dy}{dt} = y \times \mathcal{I}^{-1} y,$$

where $y \in \mathbb{R}^3$ is the momentum vector and $\mathcal{I} = \mathrm{diag}(\frac{7}{8}, \frac{5}{8}, \frac{1}{4})$ is the inertia tensor. We have solved the rigid body equation with initial condition $y(0) = \begin{bmatrix} 0.875 & 0.625 & 0.250 \end{bmatrix}^T$ over the time interval $(0, 25)$ with step size $h = 1/10$. In all the plots the following legend is used: continuous line is the outcome in the exponential coordinates, while the dash-dotted line corresponds to the Cayley coordinates.

Figure 3 shows the results for applying the Gauss–Legendre method (GL2) and Heun's method in the Lie algebra. Both methods are of second order. It is known that for exponential coordinates the GL2 method is a selfadjoint Lie-group method [26]. Thus, the error in the Hamiltonian energy is oscillating within a constant band, and will do so for all time. However, GL2 based on the Cayley coordinates is not selfadjoint, and this is reflected in the numerical experiments through the linear drift in the error of the Hamiltonian. Both methods behave as expected with respect to the retention of the Casimir function – the norm of the angular momentum vector. This is due to the fact that we use coadjoint action in both cases.

Heun's method is not selfadjoint for either coordinates, hence we see linear drift in the error of the Hamiltonian. Again, the properties of the coadjoint action make sure that we preserve the Casimir function.

In Figure 4 we see the results from applying the trapezoidal rule to the ODE in the Lie algebra. In

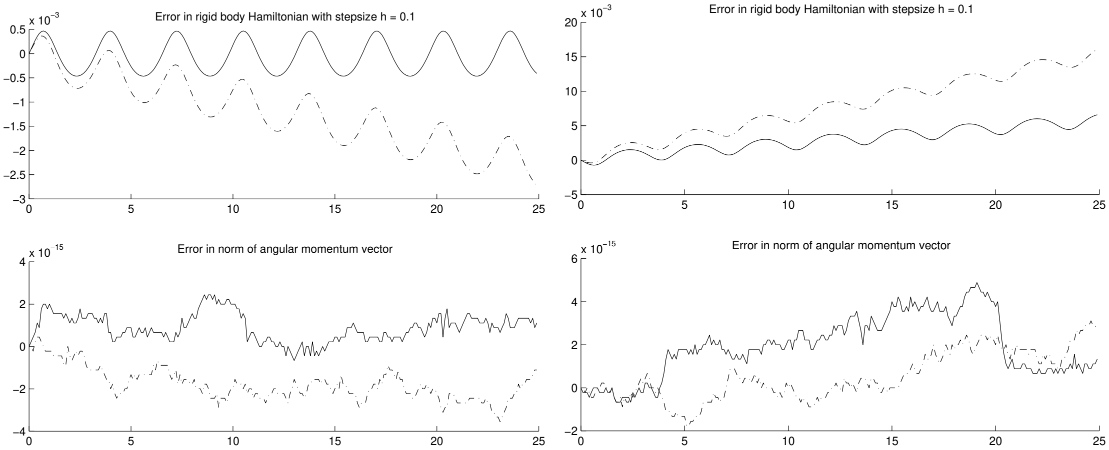

Figure 3: The rigid body integrated with the second-order methods GL2 (left) and Heun (right).

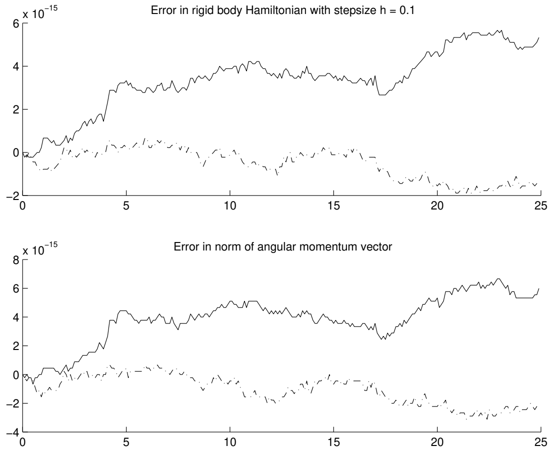

Figure 4: The rigid body integrated with the second-order trapezoidal rule.

this case both the Hamiltonian and the Casimir are preserved to machine accuracy, as expected.

6.2 The Kirchhoff equation

Next, we consider the heavy top modeled in $\mathfrak{se}(3)^*$, where the Hamiltonian equations are given by (2.6–2.7) with the Hamiltonian (2.8). Numerical values used for the constants are the following: mass $M = 1.0$, gravitational constant $g = 9.81$, and length $l = \sqrt{3}/2$. Furthermore, the inertia tensor, center-of-gravity vector, and initial condition have been taken as

$$\mathcal{I} = \frac{1}{8}\begin{bmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad \boldsymbol{\chi} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (u_0, v_0) = \left(\begin{bmatrix} 0 \\ 0 \\ 0.25 \end{bmatrix}, \begin{bmatrix} 0 \\ -0.195090 \\ 0.980785 \end{bmatrix} \right).$$

The step size of integration is $h = 1/10$.

The numerical experiment was done for both exponential and Cayley coordinates. Note that, even though the Lie group SE(3) is not quadratic, the Cayley coordinates are in fact applicable in this

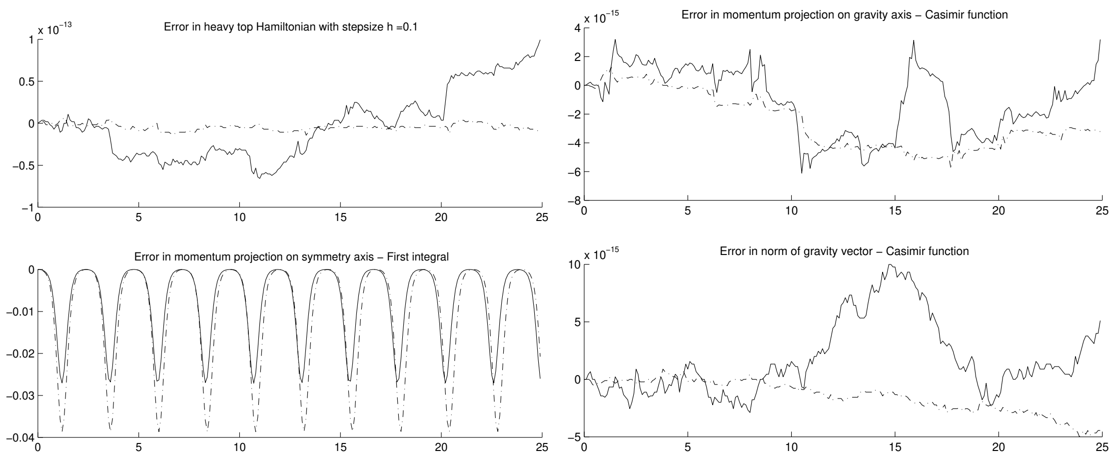

Figure 5: The Kirchhoff equation solved with the second order trapezoidal rule.

example. Figure 5 gives the results. The right plot in Figure 5 shows that both Casimir functions are rendered to machine accuracy. In the left plot we see that the behaviour of the Hamiltonian is as expected, and, rather interestingly, that the error of the last remaining first integral is well behaved. The error of this integral is confined within a constant band. A self-adjoint Lie-group method displays the same behaviour with respect to this first integral.

6.3 The Sine-Euler equation

We will now consider the infinite dimensional Lie–Poisson system given by 2D ideal fluid flow. For additional information on this particular example consult [1, 27].

6.3.1 Some background

Starting with the Euler equation in the Helmholtz form over a torus T^2, and expanding the vorticity function in a Fourier series, we arrive at the following infinite system of ordinary differential equations for the Fourier coefficients $\omega_{m_1,m_2}(t) = \omega_{\mathbf{m}}(t)$:

$$\dot{\omega}_{\mathbf{m}}(t) = \sum_{\substack{\mathbf{k}=-\infty \\ (k_1,k_2)\neq(0,0)}}^{\infty} \frac{\mathbf{m} * \mathbf{k}}{|\mathbf{k}|^2} \omega_{\mathbf{m}+\mathbf{k}} \omega_{-\mathbf{k}}, \quad \mathbf{m} = (m_1, m_2) \in \mathbb{Z}^2, \tag{6.29}$$

where $\mathbf{m} * \mathbf{k} = m_1 k_2 - m_2 k_1$. Equation (6.29) can be interpreted as an equation in Lie–Poisson form (2.3), with the Hamiltonian function

$$H(\omega) = \frac{1}{2} \sum_{\mathbf{m}=-\infty}^{\infty} \frac{\omega_{\mathbf{m}} \omega_{-\mathbf{m}}}{|\mathbf{m}|^2}, \quad \text{and} \quad \nabla H(\omega)_{\mathbf{m}} = \frac{\omega_{-\mathbf{m}}}{|\mathbf{m}|^2}.$$

The Lie group is in this case the infinite group consisting of area-preserving diffeomorphisms of the torus, SDiff(T^2), which has the set of divergence-free vector fields, SVect(T^2), as its Lie algebra. This Lie algebra is isomorphic to the set of stream functions on T^2. The dual of the Lie algebra SVect(T^2) is the set of differential one-forms modulo exact one-forms, isomorphic to the set of vorticity functions on T^2 [1]. The structure constants needed to define J in (2.3) are

$$C_{\mathbf{im}}^{\mathbf{k}} = (\mathbf{i} * \mathbf{m}) \delta_{\mathbf{k}-\mathbf{i}-\mathbf{m},0}. \tag{6.30}$$

When integrating (6.29) numerically, a finite-dimensional truncation must be introduced. The finite complex Lie algebra $\mathfrak{sl}(N,\mathbb{C})$ provides a way of doing this while preserving the intrinsic Hamiltonian structure. As N tends to infinity, $\mathfrak{sl}(N,\mathbb{C})$ tends to the Lie algebra $\mathrm{SVect}(T^2)$.

For $\mathfrak{sl}(N,\mathbb{C})$ there exists an (N^2-1)-dimensional basis $\{f_i\}$ with the commutation relations

$$[f_{\mathbf{k}}, f_{\mathbf{m}}] = -2i \sin\left(\frac{2\pi}{N}\mathbf{k} * \mathbf{m}\right) f_{\mathbf{k}+\mathbf{m}}.$$

Hence, instead of the structure constants (6.30), we can choose as structure constants

$$C^{\mathbf{k}}_{\mathbf{im}} = \sin\left(\frac{2\pi}{N}\mathbf{i} * \mathbf{m}\right) \delta_{\mathbf{k}-\mathbf{i}-\mathbf{m},0}, \tag{6.31}$$

while truncating the infinite-dimensional system to a finite one in $\mathfrak{sl}(N,\mathbb{C})$.

We choose odd N and consider a fundamental cell C around the origin of the frequency spectrum. Different choices of N yields different $\mathfrak{sl}(N,\mathbb{C})$ Lie algebras. However, vorticity being a physical quantity imposes a reality condition on the Fourier coefficients,

$$\omega^*_{\mathbf{m}} = \omega_{-\mathbf{m}},$$

where ω^* denotes complex conjugation. Hence, we are really solving a problem in the Lie algebra $\mathfrak{su}(N,\mathbb{C})$ of dimension $\frac{1}{2}(N^2-1)$.

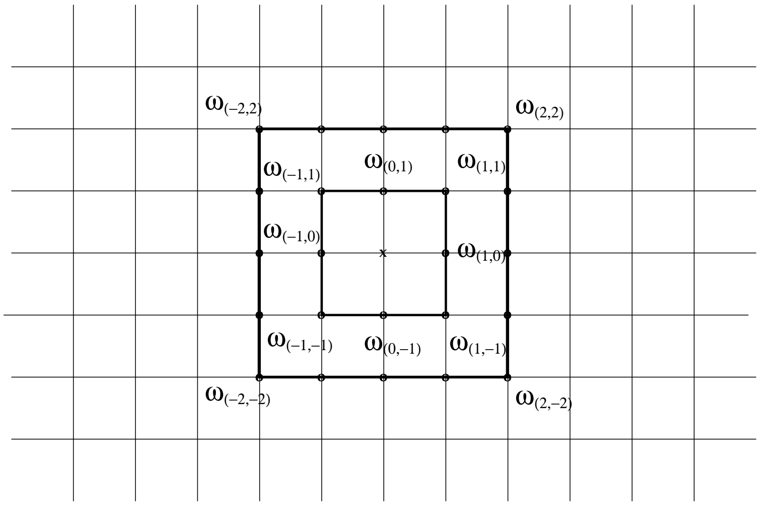

Figure 6: Discretization of the frequency spectrum: A grid with an odd number of nodes N.

In the following, we consider the case $N=3$, which is shown in Figure 6 as the eight innermost nodes (excluding the origin). Then equation (6.29) with structure constants (6.31) takes on the form

$$\dot{\omega}_{\mathbf{m}}(t) = \sum_{\mathbf{k}\in C} \frac{\sin\left(\frac{2\pi}{3}\mathbf{m} * \mathbf{k}\right)}{|\mathbf{k}|^2} \omega_{\mathbf{m}+\mathbf{k}}\omega_{-\mathbf{k}}. \tag{6.32}$$

This is a set of eight differential equations, but because of the reality condition only four of the equations are independent. Numerically, it is easier to work with a $\mathbb{C}^8$ representation of $\mathfrak{sl}(3,\mathbb{C})^*$, obtained through

$$\omega \in \mathbb{C}^8 \quad \Rightarrow \quad (A)_{\mathbf{mk}} = \sum_{\mathbf{i}\in C} C^{\mathbf{k}}_{\mathbf{im}}\omega_{\mathbf{i}} \in \mathfrak{sl}(3,\mathbb{C})^*.$$

Using the above representation, it is possible to show that $J(\omega)$ is of the following form,

$$J(\omega)=\frac{\sqrt{3}}{2}\begin{bmatrix} 0 & \omega_4 & \omega_2 & \omega_3 & 0 & -\omega_3^* & -\omega_4^* & -\omega_2^* \\ -\omega_4 & 0 & \omega_4^* & -\omega_3^* & \omega_3 & 0 & -\omega_1 & \omega_1^* \\ -\omega_2 & -\omega_4^* & 0 & \omega_2^* & \omega_4 & \omega_1^* & 0 & -\omega_1 \\ -\omega_3 & \omega_3^* & -\omega_2^* & 0 & \omega_2 & -\omega_1 & \omega_1^* & 0 \\ 0 & -\omega_3 & -\omega_4 & -\omega_2 & 0 & \omega_4^* & \omega_2^* & \omega_3^* \\ \omega_3^* & 0 & -\omega_1^* & \omega_1 & -\omega_4^* & 0 & \omega_4 & -\omega_3 \\ \omega_4^* & \omega_1 & 0 & -\omega_1^* & -\omega_2^* & -\omega_4 & 0 & \omega_2 \\ \omega_2^* & -\omega_1^* & \omega_1 & 0 & -\omega_3^* & \omega_3 & -\omega_2 & 0 \end{bmatrix}.$$

The inverse inertia operator taking elements in $\mathfrak{sl}(3,\mathbb{C})^*$ to elements in $\mathfrak{sl}(3,\mathbb{C})$ is

$$\mathcal{I}^{-1}=\begin{bmatrix} 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1/2&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1/2\\ 1&0&0&0&0&0&0&0\\ 0&1/2&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&1/2&0&0&0&0 \end{bmatrix}.$$

The functional derivative of the Hamiltonian function is the following vector in $\mathbb{C}^8$,

$$\nabla H(\omega)=\begin{bmatrix}\omega_1^* & \tfrac{1}{2}\omega_2^* & \omega_3^* & \tfrac{1}{2}\omega_4^* & \omega_1 & \tfrac{1}{2}\omega_2 & \omega_3 & \tfrac{1}{2}\omega_4\end{bmatrix}^T,$$

representing an element of $\mathfrak{sl}(3,\mathbb{C})$. Multiplying out

$$\dot{\omega}(t)=J(\omega)\nabla H(\omega),$$

yields the dynamic equations

$$\begin{aligned} \dot{\omega}_1(t) &= \frac{\sqrt{3}}{4}\left(\omega_2\omega_3^*-\omega_3\omega_4^*\right)\\ \dot{\omega}_2(t) &= \frac{\sqrt{3}}{4}\left(\omega_3^*\omega_4^*-\omega_4\omega_1^*\right)\\ \dot{\omega}_3(t) &= \frac{\sqrt{3}}{4}\left(\omega_1\omega_4-\omega_1^*\omega_2\right)\\ \dot{\omega}_4(t) &= \frac{\sqrt{3}}{4}\left(\omega_1\omega_2-\omega_2^*\omega_3^*\right). \end{aligned}$$

The Casimirs are

$$C_1 = \sum_{\mathbf{k}\in C}\omega_{\mathbf{k}}\omega_{-\mathbf{k}} \tag{6.33}$$

$$C_2 = \sum_{\mathbf{k},\mathbf{m}\in C}\cos\left(\frac{2\pi}{3}(\mathbf{k}*\mathbf{m})\right)\omega_{\mathbf{k}}\omega_{\mathbf{m}}\omega_{-\mathbf{k}-\mathbf{m}}. \tag{6.34}$$

6.3.2 Numerical experiments

In the numerical experiments we have solved (6.32) with an initial random complex vorticity distribution. The system has a total of three first integrals; two quadratic – the Hamiltonian and (6.33) – and one cubic, (6.34). We have used the same three second-order methods as in the experiments for the rigid body and added one forth-order method. We compare the present numerical experiments with the ones for the rigid body, because the Euler flow equation is the infinite-dimensional analogue to the finite Euler's rigid-body equation.

Figure 7 shows the results for GL2 and Heun's method. As expected, Algorithm 2 based on these two methods behaves very well with respect to the Casimirs, regardless of coordinate maps. Both Casimirs are preserved to machine accuracy, the same as for the rigid body. The self-adjointness of GL2 shows improved qualitative behaviour in rendering the Sine-Euler Hamiltonian in the case of exponential coordinates. However, the exact quantitative behaviour is somewhat different from the rigid body case. For the rigid body the Hamiltonian is clearly confined to within a band and the oscillations have a constant amplitude and period. In the case of truncated Euler flow equation, the Hamiltonian is bounded for very long time, but the oscillations seem to have an irregular amplitude and period, see Figure 8.

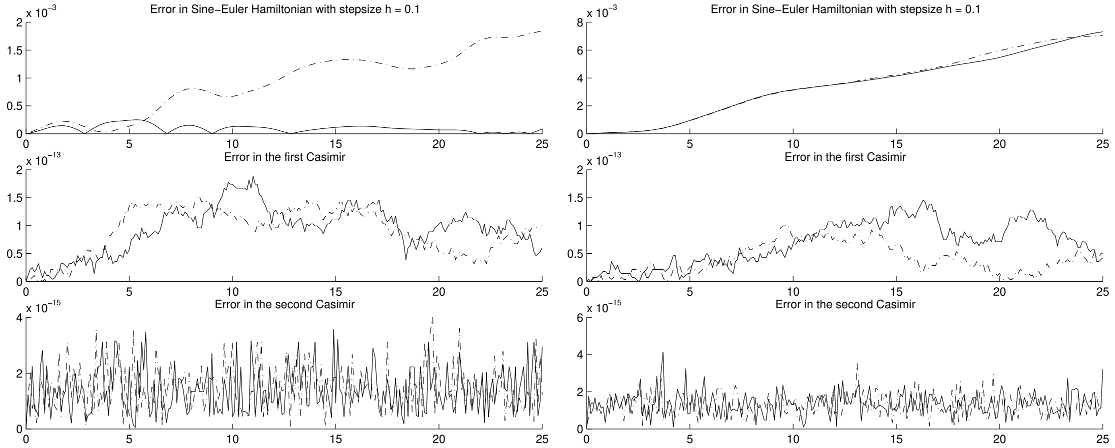

Figure 7: The Sine-Euler equation integrated with the second order methods GL2 (left), Heun (right).

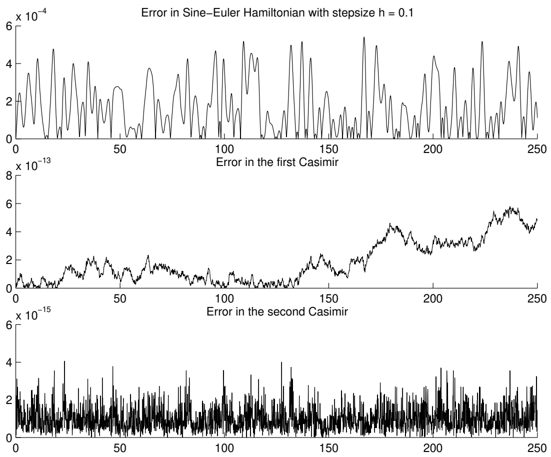

Figure 8: The Sine-Euler equation integrated with GL2 and exponential coordinates over the interval $(0, 250)$.

In Figure 9 we present the results for the second-order trapezoidal rule and the forth-order Runge–Kutta method. The trapezoidal rule behaves very well and preserves all three first integrals – just as in the case of the rigid body. Going up to higher order, we see that in the case of the Runge–Kutta method, the coadjoint action still takes care of Casimir conservation for both exponential and Cayley coordinates. But the error in the Hamiltonian shows a linear drift, just as for the second-order Heun's method.

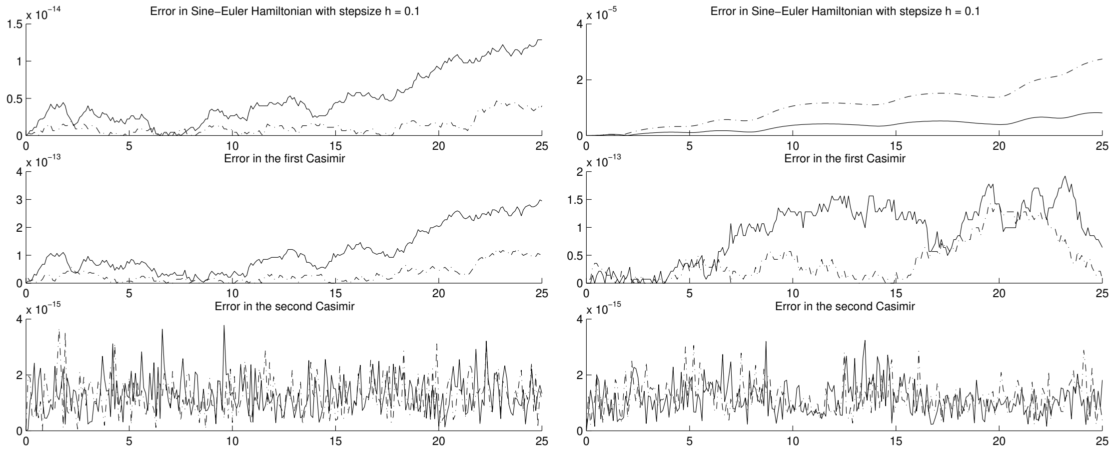

Figure 9: The Sine-Euler equation integrated with the second order trapezoidal rule (left) and the forth order RK4 (right).

7 Concluding remarks

We have shown how to integrate Hamiltonian systems evolving on group coadjoint orbits, which are submanifolds in the dual space of a Lie algebra, in such a way that the numerical algorithm preserves the group coadjoint orbit, Casimir functions and the Hamiltonian energy of the system. Numerical experiments also indicate that the algorithm shows good behaviour on non-Casimir first integrals.

In the present paper we have focused on energy conservation, as opposed – in the sense of the modified Ge–Marsden result – to constructing methods that preserve non-canonical symplectic structure. We believe that the RKMK setting is not sufficient for implementing symplectic methods, and incorporation of ideas similar to [14] should be looked into. This, however, is a topic for future research.

Acknowledgement

We want to thank Arieh Iserles and Hans Munthe-Kaas for valuable input in the preparation of this paper.

References

[1] V. I. Arnold and B. A. Khesin, *Topological Methods in Hydrodynamics*, Springer-Verlag, 1998.

[2] C. Budd and A. Iserles, *Geometric integration: numerical solution of differential equations on manifolds*, Phil. Trans. Royal Soc. A, 357 (1999), pp. 945–956.

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

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

[5] K. Engø, *On the construction of geometric integrators in the RKMK class*, BIT, (2000). To appear.

[6] D. J. Estep and A. M. Stuart, *The rate of error growth in Hamiltonian-conserving integrators*, Z. Angew. Math. Phys., 46 (1995), pp. 407–418.

[7] S. Faltinsen, *Backward Error Analysis for Lie-Group Methods*, Tech. Rep. 1998/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1998.

[8] S. Faltinsen, A. Marthinsen, and H. Z. Munthe-Kaas, *Multistep methods integrating ordinary differential equations on manifolds*, Tech. Rep. Numerics No. 3/1999, Norwegian University of Science and Technology, Trondheim, Norway, 1999.

[9] Z. Ge and J. E. Marsden, *Lie–Poisson Hamiltonian–Jacobi Theory and Lie–Poisson Integrators*, Physics Letters A, 133 (1988), pp. 134–139.

[10] O. Gonzalez, *Time integration and discrete Hamiltonian systems*, J. Nonlinear Sci., 6 (1996), pp. 449–467.

[11] E. Hairer and D. Stoffer, *Reversible long-term integration with variable step sizes*, SIAM J. Sci. Comput., 18(1) (1997), pp. 257–269.

[12] A. Iserles, *On Cayley-transform methods for the discretization of Lie-group equations*, Tech. Rep. 1999/NA04, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1999.

[13] D. Lewis and J. C. Simo, *Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups*, J. Nonlinear Sci., 4 (1994), pp. 253–299.

[14] J. E. Marsden, S. Pekarsky, and S. Shkoller, *Discrete Euler–Poincare and Lie–Poisson equations*. Submitted to Nonlinearity.

[15] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry*, no. 17 in Texts in Applied Mathematics, Springer-Verlag, second ed., 1999.

[16] A. Marthinsen and B. Owren, *Quadrature methods based on the Cayley transform*, Tech. Rep. Numerics No. 1/1999, The Norwegian University of Science and Technology, Trondheim, Norway, 1999.

[17] R. I. McLachlan, *Explicit Lie–Poisson integration and the Euler equations*, Physical Review Letters, 71 (1993), pp. 3043–3046.

[18] R. I. McLachlan, *On the numerical integration of ODE's by symmetric composition methods*, SIAM J. Numer. Anal., 16 (1995), pp. 151–168.

[19] R. I. McLachlan, G. R. W. Quispel, and N. Robidoux, *Geometric integration using discrete gradients*, Phil. Trans. Royal Soc. A, 357 (1999), pp. 1021–1046.

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

[21] S. Reich, *Backward error analysis for numerical integrators*, SIAM J. Numer. Anal., 36 (1999), pp. 1549–1570.

[22] ——, *Symplectic integration of constrained Hamiltonian systems by composition methods*, SIAM J. Numer. Anal., 33 (1999), pp. 475–491.

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

[24] C. Scovel and A. Weinstein, *Finite dimensional Lie–Poisson approximations to Vlasov–Poisson equations*, Comm. Pure and Appl. Math., 47 (1994), pp. 683–709.

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

[26] A. Zanna, K. Engø, and H. Z. Munthe-Kaas, *Adjoint and selfadjoint Lie-group methods*, Tech. Rep. 1999/NA02, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1999.

[27] V. Zeitlin, *Finite-mode analogs of 2D ideal hydrodynamics: Coadjoint orbits and local canonical structure*, Physica D, 49 (1991), pp. 353–362.