Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

# Topics on geometric integration

**Ph.D. candidate**
Giulia Ortolan

**Advisor**
prof. Alessandro Beghi

Ph.D. School
in Information Engineering
2011

*To myself*

"Bazinga!"
"I am not insane, my mother had me tested."
Jim Parsons as Sheldon Cooper, *The Big Bang Theory*, tv series (2007-)

"Ah, mi spiace, ma io so' io, e voi nun siete un cazzo!"
Alberto Sordi as Onofrio del Grillo, *Il Marchese del Grillo*, 1981

"Il mio sistema è in crash."
Immanuel Casto, *Crash*, 2010

# Abstract

In many different engineering branches, computer-based simulations for analyzing systems' behavior hold a growing importance for the development of new products and new technical solutions. Pursuing this aim requires two stages: building a mathematical model of the system and solving it employing a computer. Algorithms for calculating the numerical solution of differential equations, called *numerical integrators*, become thus extremely decisive as they influence the reliability of the simulation results.

Many dynamical systems exhibit properties that are preserved by the flow, e.g., energy conservation, symmetry, momentum, symplecticity, configuration manifold. A conventional numerical integrator approximates the flow of the continuous-time equations using only the information about the vector field, ignoring the physical laws and the properties of the original trajectory. In this way, small inaccuracies accumulated over long times will significantly diminish the operational lifespan of such discrete solutions. *Geometric integrators*, on the other hand, are built in a way that preserve the structure of continuous dynamics, so maintaining the qualitative behavior of the exact flow even for long-time integrations.

In this thesis, two different issues related to geometric integration are investigated. The first one describes the design of a numerical test which can be employed to assess in a easy way the long-time behavior of a rigid body integrator, in terms of energy preservation. The second one presents a straightforward approach to use off-the-shelf Lie methods for the integration of Hamiltonian systems evolving on a product of two spheres.

## A numerical test of long-time stability for rigid body integrators

The continuous-time flow of a Hamiltonian system (like a rigid body immersed in a static potential field) is symplectic, that is, it preserves the symplectic form. If the chosen integration method is symplectic or conjugate-symplectic then backward error analysis can be used to prove that (under some technical conditions) the method has an excellent long-time behavior. Moreover, symplecticity is often regarded as a key property for the preservation of the structure and the properties of the continuous-time flow.

A large number of geometric algorithms for time integration of rigid body rotational dynamics have been proposed in the last 30 years. Some of these algorithms preserve by construction the canonical symplectic form, and therefore their good long-time behavior is assured from backward error analysis. Some other algorithms have been obtained approximating the continuous-time dynamics using *ad hoc* methodologies, with the declared aim of obtaining computationally fast algorithms with small error constant. In this latter case, the long-time behavior has been assessed using numerical experiments on a series of test cases, monitoring the behavior of the energy and known first integrals, but without any guarantee of good performance over long-time in a generic situation.

Our contribution in this context is to propose a simple numerical test that has shown to be able to spot an energy drift in many rigid body algorithms that were reported in the literature to possess good energy behavior over long-time. The test consists in integrating the rotation dynamics of a rigid body in a suitable static field which is the sum of a bounded and an unbounded attractive terms. The presence of an energy drift implies that the energy error is not bounded over exponentially long times, and this allows us to conclude that these methods are not symplectic neither conjugate-symplectic. Among them, we cite the Lie-Newmark family of methods, whose flat space counterpart is instead known to be conjugate-symplectic.

**Keywords: geometric integration, long-term stability, Hamiltonian mechanics, variational integrators**

## Lie methods for integration of dynamical systems on two-spheres

The two-sphere $\mathbb{S}^2$ is defined as the set of all points in $\mathbb{R}^3$ which have a unit length from the origin. Many classical and important dynamical systems evolve on two-sphere or on a product of two-spheres. In these cases, the configuration is usually described using 2 angles or a constraint enforcing unit length (on $\left(\mathbb{S}^2\right)^n$, $2n$ angles or $n$ constraints); these representations should be however avoided, since they yield additional complexity in the computation.

The geometric approach to this problem exploits the fact that the special orthogonal Lie group of rotation matrices SO(3) acts transitively on the two-sphere; in this way, it is assured that the discrete trajectory belongs to the configuration space at any time without enforcing any constraints. One can therefore move the problem to the group space, searching for the trajectory in SO(3) that, through the group action, generates the flow in $\mathbb{S}^2$.

Our contribution to this issue consists in a simple and straightforward method to adapt off-the-shelf Lie methods to solve for the dynamics of a system whose configuration space is the product of two-spheres. This approach is based on the Euler-Lagrange equations written on two-sphere, and has been tested on several significative numerical examples, proving to be accurate and computationally efficient. The biggest advantage of the proposed approach is represented by the possibility of effortlessly employing well-known Lie group methods, thus easily obtaining a high-order accuracy in the integration.

**Keywords: geometric integration, integration on two-spheres, Lie methods**

# Sintesi

In molti campi di ricerca, l'importanza di simulare al calcolatore il comportamento di sistemi dinamici diventa cruciale nello sviluppo e nel test di nuovi prodotti o soluzioni tecniche. A questo proposito si rende necessario avere un modello matematico del sistema di interesse e risolvere le equazioni della dinamica utilizzando un computer. La scelta dei metodi che consentono di calcolare la soluzione numerica di equazioni differenziali, detti *integratori numerici*, diventa quindi determinante, influenzando l'affidabilità dei risultati delle simulazioni.

Una gran parte dei sistemi dinamici possiede proprietà che vengono preservate dal flusso, come ad esempio la conservazione dell'energia, la simmetria, le mappe dei momenti, la simpletticità, lo spazio delle configurazioni. Un metodo numerico generico genera un'approssimazione del flusso a tempo continuo usando solo l'informazione contenuta nelle equazioni della dinamica, tralasciando le leggi fisiche e le proprietà della traiettoria originale. In questo modo, l'accumularsi degli errori rende inaffidabile la soluzione numerica discreta generata da una simulazione di lungo periodo. Al contrario gli *integratori geometrici* sono pensati in modo da preservare la struttura della dinamica a tempo continuo, mantenendo così le proprietà del flusso esatto anche per simulazioni molto lunghe.

Questa tesi affronta due differenti problemi, entrambi annessi al campo dell'integrazione geometrica. Nella prima parte viene descritto un test numerico che consente di verificare facilmente il comportamento energetico a lungo termine degli integratori del corpo rigido. Nella seconda parte viene presentato un approccio che consente di utilizzare metodi numerici per gruppi di Lie per l'integrazione di sistemi dinamici il cui spazio delle configurazioni è un prodotto di sfere unitarie.

## Stabilità a lungo termine per gli integratori del corpo rigido

Il flusso a tempo continuo di un sistema hamiltoniano (quale, ad esempio, un corpo rigido immerso in un campo potenziale statico) è simplettico, ovvero preserva la forma simplettica. Se anche il metodo numerico impiegato per l'integrazione della dinamica è simplettico o coniugato-simplettico, si può dimostrare che (dimostrato che valgono alcune condizioni tecniche) l'algoritmo dimostra una performance eccellente per simulazioni a lungo termine; più precisamente, l'errore sull'energia totale rimane limitato per tempi esponenzialmente lunghi. Inoltre, una buona parte della letteratura vede nella simpletticità una proprietà chiave per la conservazione della struttura e delle caratteristiche del flusso a tempo continuo.

Negli ultimi 30 anni, un gran numero di metodi numerici per l'integrazione della dinamica rotazionale del corpo rigido è stato proposto in letteratura. Di essi, una parte è simplettica per costruzione, e ciò ne garantisce un ottimo comportamento per simulazioni a lungo termine. Altri algoritmi sono invece stati costruiti usando metodologie *ad hoc*, con lo scopo di ottenere un metodo computazionalmente veloce e accurato. Per questi ultimi, l'analisi della soluzione per tempi lunghi è stata condotta mediante esperimenti numerici, monitorando il comportamento dell'energia e degli integrali del moto; in questo caso, tuttavia, non si ha nessuna garanzia sul buon comportamento del metodo per simulazioni a lungo termine in un caso generale.

In questo contesto, abbiamo costruito un test numerico che ha dimostrato di evidenziare drift energetici in algoritmi che, in esperimenti precedenti, avevano mostrato un buon comportamento per simulazioni di lungo periodo. Il test consiste nell'integrare la dinamica di un corpo rigido immerso in un potenziale statico somma di due termini attrattivi, uno limitato e uno illimitato. La presenza di un drift nell'energia assicura che l'errore non rimanga limitato per tempi esponenzialmente lunghi, e ciò ci consente di concludere che gli algoritmi da noi testati non sono simplettici nè coniugato-simplettici. È degno di nota il fatto che il nostro test ha permesso di escludere la natura simplettica (e anche coniugato-simplettica) dei ben noti algoritmi di Lie-Newmark, apparsi in letteratura più di 20 anni fa.

**Parole chiave: sistemi hamiltoniani, integrazione geometrica, stabilità a lungo termine, integratori variazionali**

## Integrazione con metodi di Lie della dinamica di sistemi su sfere unitarie

La sfera unitaria $\mathbb{S}^2$ è definita come l'insieme di tutti i punti in $\mathbb{R}^3$ che hanno distanza unitaria dall'origine. Molti sistemi dinamici classici evolvono sulla sfera unitaria o sul prodotto di sfere unitarie. In questi casi, la configurazione è solitamente descritta usando 2 angoli o un vincolo che impone lunghezza unitaria (più in generale, $2n$ angoli e $n$ vincoli); rappresentazioni di questo tipo aggiungono però complessità ai calcoli, ed andrebbero pertanto evitate.

L'approccio geometrico, che garantisce senza l'imposizione di vincoli che ogni punto della traiettoria discreta appartenga alla sfera unitaria, sfrutta il fatto che il gruppo speciale ortogonale di Lie delle matrici di rotazione SO(3) agisce transitivamente sulla sfera unitaria. Ciò suggerisce di spostare il problema nello spazio dell'azione, cercando in SO(3) la traiettoria che genera la soluzione del sistema in $\mathbb{S}^2$.

Il nostro contributo in questo ambito consiste in un metodo semplice e diretto per adattare metodi di Lie alla risoluzione della dinamica di sistemi il cui spazio delle configurazioni è un prodotto di sfere unitarie. Tale approccio si basa sulle equazioni del moto di Eulero-Lagrange scritte sulla sfera di raggio unitario, ed è stato ampiamente testato in molti esempi numerici di interesse pratico e scientifico, mostrandosi accurato e computazionalmente efficiente. Il principale vantaggio offerto da questa soluzione è rappresentato dal poter sfruttare senza sforzi addizionali metodi numerici già studiati in letteratura per l'integrazione su gruppi di Lie, arrivando così ad ottenere facilmente un ordine di accuratezza anche molto alto.

**Parole chiave: integrazione geometrica, integrazione sulla sfera unitaria, metodi di Lie**

# Contents

# List of Figures

# 1 Numerical integration

In this Chapter we briefly introduce numerical integration, focusing on geometric integration and the numerical methods arising from a variational approach (Section 1.1); in order to make things easier for the reader, in Section 1.2 we also recall the basic definitions and properties for numerical integrators. The main contributions of this Thesis in the framework of geometric integration are discussed in Section 1.3.

## 1.1 Introduction to geometric integration

Solving an autonomous ordinary differential equation (ODE) means seeking for a function $x(t)$ whose derivative satisfies

$$\frac{d}{dt}x(t) = f(x(t)),$$

where $f(x(t))$ is called *vector field*. If we specify an initial condition $x(t_0) = x_0$, we are dealing with an initial value problem (IVP), and under some Lipschitz condition on $f$ there exists a unique solution $x(t)$. The function $\varphi_t(x_0) = x(t)$ is called *flow* of the differential equation, and the uniqueness of solutions yields $\varphi_0 = id$ and $\varphi_s \circ \varphi_t = \varphi_{s+t}$. Without loss of generality, from now on we will suppose $t_0 = 0$.

A *numerical method* is a discrete approximation $\Phi_h(x_0)$ of $\varphi_h(x_0) = x(h)$, where $h$ is called timestep. Applying iteratively the discrete flow we obtain a discrete trajectory

$$\tilde{x}_0, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k, \ldots \overset{\text{def}}{=} \tilde{x}_0, \tilde{x}_h, \tilde{x}_{2h}, \ldots, \tilde{x}_{kh}, \ldots$$

which is ideally a sampling of the exact trajectory $x(t)$ at times $kh$. In reality, though, the discrete trajectory only approximates the original solution; the effectiveness of the approximation can be evaluated, e.g., in terms of accuracy or preservation of the properties of the continuous trajectory.

General purpose numerical integrators construct an approximation of the exact flow using only information from the vector field, thus ignoring the physical laws or the structures which mark the dynamics. *Geometric integration* refers to numerical methods which preserve inherent geometric structures of the continuous-time dynamics, such as configuration manifold, Hamiltonian or Poisson structure, invariants, symmetries. A geometric integrator is therefore able to capture the long-term behavior of the dynamics, which is a crucial aspect for long-time simulations.

Hamiltonian systems constitute a relevant class of problems for numerical integration; among other geometric properties of their flow, preservation of the symplectic form deserves a special mention. In fact, if a numerical integrator is symplectic, it also exactly preserves the momentum maps arising from eventual symmetries, and keeps bounded the error on the total energy. More than energy conservation, for Hamiltonian systems symplecticity reveals itself as a crucial property to capture essential properties of the system dynamics [BR07].

As shown in [MW01], discrete variational mechanics allows the derivation of numerical methods using a discrete version of the Hamiltonian principle of least action, and all such methods are symplectic. Actually, a stronger result holds: a method is symplectic if and only if it is a variational method [MW01]. Numerical simulations also show that variational integrators exhibit a good performance for forces and dissipative systems; variational integrators represent therefore one of the most important class among geometric integrators, and they assure an excellent performance on long-time simulations.

## 1.2 Properties of numerical integrators

In this Section we recall the most important definitions and properties related to numerical integration, which will be used all throughout this work. The basic references used are [CS72, MW01] for subsection 1.2.1, [HLW06] for subsection 1.2.2, [MW01, MR99, BG94] for subsection 1.2.3 and [HLW06] for subsection 1.2.4. To our knowledge, Proposition 1.2.15 is a novel contribution to geometric integration theory, even if its statement was a predictable fact [Hai08, HMS09]. We will often omit proof of propositions and lemmas, inviting the reader to make reference to the bibliography to gain some technical insight.

### 1.2.1 Order of a method

Let us preliminarly introduce the $\mathcal{O}$ notation. Be $x, y : \mathbb{R} \to \mathbb{R}$ two smooth functions such that $x(t), y(t) \to 0$ when $t \to 0$. We write $x(t) = \mathcal{O}(y(t))$ meaning that

$$\|x(t)\| \leq C\|y(t)\|, \; C > 0$$

when $t$ is close enough to 0. More specifically, we usually mean that $x(t)$ behaves exactly like $y(t)$ as $t$ approaches 0.

**Definition 1.2.1** (Consistency). *An integrator $\Phi_h$ is said to be consistent of order $r$ if there exist an open set $U$ and constant $\overline{h} > 0$ such that*

$$\|\Phi_h(x_0) - \varphi_h(x_0)\| = \mathcal{O}(h^{r+1}) \quad \textit{for } h \leq \overline{h}, \; x_0 \in U,$$

*with $r \geq 1$.*

The expression on the left-hand side of the inequality is known as *local error*, and measures how much error is introduced in the discrete trajectory after a single timestep.

**Definition 1.2.2** (Convergency). *An integrator $\Phi_h$ is said to be convergent of order $r$ if there exist an open set $U$ and constant $\tilde{h} > 0$ such that*

$$\|\Phi_h^N(x_0) - \varphi_{Nh}(x_0)\| = \mathcal{O}(h^r) \quad \textit{for } h \leq \tilde{h},\ x_0 \in U.$$

Convergence is related to the global accuracy, and bounds the global error after $N$ timesteps $e_{h,N} \overset{\text{def}}{=} \|\Phi_h^N(x_0) - \varphi_{Nh}(x_0)\|$. The convergence order can be assessed by expanding in a Taylor series in $h$ both the true flow and its discrete approximation and then comparing terms: if they agree up to order $r$, then the integrator is of order $r$.

It is reasonable to assume that some perturbation acts on the method, e.g., computational errors. Be $p_0, p_1, \ldots, p_N$ the perturbation sequence acting on a $N$-steps discrete trajectory, such that $x_{k+1} = \Phi_h(x_k + p_k)$, and be $\mu(p) \overset{\text{def}}{=} \mu(\{p_i\}_{i=0}^N)$ a measure for the perturbation.

**Definition 1.2.3** (Orderstability). *An integrator is orderstable if the distance between the non-perturbed trajectory and a perturbed one is a $\mathcal{O}(\mu(p))$ as $h \to 0$ and $\mu(p) \to 0$.*

The orderstability property is related to the continuity of the method with respect to the perturbation when the timestep gets smaller and smaller.

**Theorem 1.2.4.** *An orderstable integrator is consistent of order $r$ if and only if it is also convergent of order $r$.*

*Proof.* See [MNS74] or [CS72]. □

Theorem 1.2.4 is a powerful tool, since it relates global and local error of an integrator and unifies the concept of order of a numerical method. We can therefore have information about the global behavior of an integrator analyzing only its one-step iteration rule.

Since expanding in a Taylor series in $h$ the discrete flow can be complicated, one can instead compute the global error of the method at a certain time $T$, for different timesteps $h = T/N$, and assess the relationship between the timestep and the global error. Suppose we are working with a $r-$th order method, $e_{h,N} = \mathcal{O}(h^r)$, that is, $e_{h,N} \approx Ch^r$, and then plot couples $(h, e_{h,N})$ in a log-log diagram. Basic logarithm property leads to

$$(\log h, \log Ch^r) = (\log h, \log C + r \log h),$$

which corresponds to a line with a slope of $r$. Such diagrams are called *time-precision diagrams*. In virtue of Theorem 1.2.4, the order of a method can then be assessed just estimating the slope of a line. Note that this procedure requires to know $\varphi_{Nh}$, in order to compute $e_{h,N}$. Usually, a discrete approximation obtained with a high-order method and a tiny timestep (or with a fixed error tolerance and a variable timestep) is considered as the exact solution.

### 1.2.2 Symmetry and reversibility

The flow $\varphi_t$ of an autonomous ODE is time-reversible, that is,

$$\varphi_{-t} = \varphi_t^{-1}.$$

The corresponding property in the discrete framework is discussed in the following.

**Definition 1.2.5** (Adjoint method). *The adjoint method $\Phi_h^*$ of a method $\Phi_h$ is the inverse map of the original method with reverse timestep h:*

$$\Phi_h^* \overset{def}{=} \Phi_{-h}^{-1}.$$

**Theorem 1.2.6.** *If a method $\Phi_h$ has order p, its adjoint has the same order p.*

*Proof.* See [HLW06, Theorem II.3.2]. □

**Definition 1.2.7.** *The numerical method $\Phi_h$ is called symmetric or time-reversible if it satisfies*

$$\Phi_{-h} \circ \Phi_h = \Phi_h \circ \Phi_{-h} = id.$$

Symmetry assures that the same discrete flow can be generated backwards, starting from the endpoint, just changing the timestep sign. Recalling definition 1.2.5, one can express the symmetry condition as

$$\Phi_h = \Phi_h^*,$$

that is, a symmetric method is equal to its adjoint. A simple method to verify if a method is symmetric is to exchange $h \longleftrightarrow -h$ and $x_0 \longleftrightarrow x_1$ and checking if the algorithm remains unchanged.

**Proposition 1.2.8.** *The composition of a method with its adjoint yields a symmetric method.*

*Proof.* Denote $\Psi_h = \Phi_h \circ \Phi_h^* = \Phi_h \circ \Phi_{-h}^{-1}$. Then $\Psi_{-h} = \Phi_{-h} \circ \Phi_h^{-1}$. Composing $\Psi_h$ and $\Psi_{-h}$ yields

$$\Psi_h \circ \Psi_{-h} = \Phi_h \circ \Phi_{-h}^{-1} \circ \Phi_{-h} \circ \Phi_h^{-1} = id,$$
$$\Psi_{-h} \circ \Psi_h = \Phi_{-h} \circ \Phi_h^{-1} \circ \Phi_h \circ \Phi_{-h}^{-1} = id,$$

which proves the symmetry of $\Psi_h$. □

**Theorem 1.2.9.** *If a method is symmetric, its order is even.*

*Proof.* See [HLW06, Theorem II.3.2]. □

In a more general sense, if $\rho$ is a linear invertible transformation, the vector field $f$ is said to be *$\rho$-reversible* if

$$\rho \circ f = -f \circ \rho.$$

The flow of a $\rho$-reversible autonomous differential equation satisfies

$$\rho \circ \varphi_t = \varphi_{-t} \circ \rho = \varphi_t^{-1} \circ \rho,$$

as shown, e.g., in [HLW06, Chapter V].

**Definition 1.2.10** ($\rho$-reversibility). *A map $\Phi_h$ is said $\rho-$reversible if*

$$\rho \circ \Phi_h = \Phi_h^{-1}.$$

**Lemma 1.2.11** ($\rho-$compatibility condition). *Let $\Phi_h$, when applied to a $\rho$-reversible dynamics, satisfy*

$$\rho \circ \Phi_h = \Phi_{-h} \circ \rho.$$

*Then $\Phi_h$ is $\rho$-reversible if and only if it is symmetric.*

Considering a partitioned system $x = (u, v)$, the most important $\rho$ transformation is given by

$$\rho(u, v) = (u, -v). \tag{1.1}$$

From now on, an ODE will be called reversible meaning it is $\rho$-reversible with respect to (1.1). Since reversible numerical methods often exhibit a good long-time behavior, especially in energy preservation, one usually employs a reversible map to solve for the dynamics of a reversible system. Actually, Chapter XI of [HLW06] provides a theoretical explanation of this good behavior and discusses the necessary conditions for it to happen.

### 1.2.3 Symplecticity

A (strong) *symplectic manifold* is a pair $(P, \Omega)$ where $P$ is a manifold and $\Omega$ is a closed, strongly nondegenerate two-form on $P$. Recall that a two-form $\Omega$ on a manifold $P$ is a function $\Omega(p) : T_pP \times T_pP \to \mathbb{R}$ that assigns to each point $p \in P$ a skew-symmetric bilinear form on the tangent space $T_pP$. A closed form has a null exterior derivative; a form is said to be nondegenerate if $\Omega(z_1, z_2) = 0$ for all $z_2$ implies $z_1 = 0$, while a form is said to be strongly nondegenerate if, for a fixed $z_1$, and $\Omega(z_1, z_2)$ is an isomorphism. A corollary of the Darboux Theorem [MR99, Theorem 5.1.2] assures that, if a strong symplectic manifold is even dimensional, then there exist local coordinates $\{q_1, \dots, q_n, p_1, \dots, p_n\}$ such that the two-form $\Omega$ can be represented as

$$\Omega = \sum_{i=1}^{n} dq_i \wedge dp_i, \tag{1.2}$$

the *canonical symplectic form*. In matrix representation, $\Omega$ becomes

$$\Omega = \begin{bmatrix} \mathbf{0}_n & I_n \\ -I_n & \mathbf{0}_n \end{bmatrix}, \tag{1.3}$$

where $I_n$ is the $n \times n$ identity matrix and $\mathbf{0}_n$ is the null $n \times n$ matrix. We call (1.3) *symplectic matrix*. Unless otherwise specified, from now on we will use symplectic form referring to (1.2).

A vector field $f$ is called *Hamiltonian* if there exists a function $H : P \to \mathbb{R}$ such that

$$\Omega_z(f(z), v) = \mathrm{D}H(z) \cdot v, \;\; \forall z \in P, \;\; v \in T_zP;$$

the flow of a Hamiltonian system preserves the Hamiltonian:

$$H \circ \varphi_t = H.$$

Now consider a $C^\infty$ transformation $\psi : P_1 \to P_2$ between two symplectic manifolds $P_1$ and $P_2$. The mapping $\psi$ is called *symplectic* if for all $z \in P_1,\ v,\ w \in T_z P_2$ we have

$$\Omega_1(z)(v,w) = \Omega_2(\psi(z))(D\psi(z)\cdot v, D\psi(z)\cdot w),$$

that is, it preserves the symplectic form.

As it is shown in [MR99, Chapter 5], the flow $\varphi_t$ of a vector field $f$ is a symplectic transformation if and only if $f$ is (locally) Hamiltonian. Therefore, the symplectic form is preserved along the solution of a Hamiltonian system. Moreover, it can be proved that a symplectic flow also preserves some quantities, called *momentum maps*, which derive from eventual invariance of the system to certain action.

A really relevant class of ODE is represented by Hamiltonian systems, e.g., conservative mechanical systems, whose Hamiltonian is usually the total energy of the system. Numerical methods which guarantee a good performance in solving for Hamiltonian dynamics are therefore of great interest.

**Definition 1.2.12** (Symplectic method). *A method $\Phi_h$ is a symplectic map if the one-step iteration $\Phi_h(x)$ is symplectic when applied to a Hamiltonian system.*

If a method $\Phi_h$ is symplectic, it preserves exactly the momentum maps of the system [MR99]. Benettin and Giorgilli [BG94] add a fundamental characterization of symplectic mappings. Suppose that $\Phi_h$ is an analytic transformation. They proved that there exists a modified analytic Hamiltonian $\widetilde{H}$ and a corresponding modified vector field $\widetilde{f}$ such that

$$\|\widetilde{\varphi}_h - \Phi_h\| = \mathcal{O}(e^{-\frac{1}{\beta h}}), \tag{1.4}$$

where $\beta$ is a constant positive number which depends on the method[1]. This fact has crucial consequences on the preservation of the original Hamiltonian.

**Theorem 1.2.13.** *Consider a Hamiltonian system with analytic H and apply a symplectic method $\Phi_h$ of order $r$. Be $\widetilde{H}$ the modified analytic Hamiltonian which satisfies* (1.4). *Then there exists $h_0 > 0$ such that*

$$\begin{aligned}\widetilde{H}(\Phi_h^n(x_0)) &= \widetilde{H}(x_0) + \mathcal{O}(e^{-\frac{h_0}{2h}}),\\ H(\Phi_h^n(x_0)) &= H(x_0) + \mathcal{O}(h^s)\end{aligned}$$

*over exponentially long time intervals $nh \leq \mathrm{e}^{\frac{h_0}{2h}}$.*

[1] For the sake of simplicity, we will suppose the global existence of the modified Hamiltonian. This is always true if the domain in simply connected, otherwise some additional assumptions would be required. We refer the reader to [BG94, Remark on page 1123] for further information.

*Proof.* See [BG94]. □

This Theorem proves that a symplectic method nearly preserves energy over exponentially long time intervals; this is a crucial fact for long-time simulations.

### 1.2.4 Conjugate symplecticity

Symplecticity is a strong requirement for a numerical method. In this subsection we will show how to keep good long-time behavior properties of an integrator relaxing that condition.

**Definition 1.2.14** (Conjugate symplecticity)**.** *Two numerical methods $\Phi_h$ and $\Psi_h$ are mutually conjugate if there exists a global change of coordinates $\chi_h$ such that*

$$\Psi_h = \chi_h \circ \Phi_h \circ \chi_h^{-1}. \tag{1.5}$$

*We assume that $\chi_h = id + \mathcal{O}(h^s)$, $s \geq 1$, uniformly on the whole domain.*

Note that $\chi_h^{-1} = id + \mathcal{O}(h^s)$. If $\Phi_h$ is a symplectic method and (1.5) holds, then $\Psi_h$ is said to be *conjugate-symplectic*. We now show, in a similar manner to [BG94], that a conjugate-symplectic method nearly preserves exact energy over exponentially long times, when applied to a conservative system.

Be $H : D \to \mathbb{R}$ an analytic Hamiltonian, with $D$ its (possibly infinite) domain, and $\Phi_h$ a $p$-th order symplectic method[2] and $\Psi_h$ its conjugate-symplectic method. Finally, be

$$\widetilde{H}_{N,h}(x) = H(x) + h^p H_{p+1}(x) + ... + h^{N-1} H_N(x)$$

a modified Hamiltonian.

**Proposition 1.2.15.** *There exists a constant $h_0$ and an integer number $N = N(h)$, namely the bigger $N$ such that $Nh \leq h_0$, for which*

$$\widetilde{H}_{N(h),h}(x_n) = \widetilde{H}_{N(h),h}(x_0) + O(h^s) \tag{1.6}$$

$$H(x_n) = H(x_0) + O(h^{\min\{s,p\}}) \tag{1.7}$$

*over exponentially long time intervals $nh \leq e^{h_0/2h}$.*

*Proof.* Consider the difference $\widetilde{H}_{N,h}(\Psi_h^n(x_0)) - \widetilde{H}_{N,h}(x_0)$, where the truncation order $N$ is to be specified. Adding and subtracting $\widetilde{H}_{N,h}(\Phi_h^n(x_0))$, we get

$$\widetilde{H}_{N,h}(\Psi_h^n(y)) - \widetilde{H}_{N,h}(y) = \widetilde{H}_{N,h}(\Psi_h^n(y)) - \widetilde{H}_{N,h}(\Phi_h^n(y)) + \widetilde{H}_{N,h}(\Phi_h^n(y)) - \widetilde{H}_{N,h}(y) \tag{1.8}$$

[2] We also assume that in the power series expansion $\Phi_h(x) = x + h\phi_1(x) + h^2\phi_2(x) + \dots$ the terms $\phi_i$, $i = 1, 2, \dots$ satisfy $\sup_{z \in D_\rho} |\phi_i(z)| \leq \gamma^{i-1}\Gamma$ with $D_\rho \subset \mathbb{C}^{2n}$ is a complex neighborhood of the real domain $D$. See [BG94] for details.

Benettin and Giorgilli showed that there exist a constant $h_0 > 0$ and an optimal truncation $N = N(h)$ (namely the bigger natural such that $Nh \leq h_0$) such that for $h \leq h_0$ and an exponentially long integration interval, i.e., $nh \leq e^{h_0/2h}$, we have[3]

$$|\widetilde{H}_{N(h),h}(\Phi_h^n(x_0)) - \widetilde{H}_{N(h),h}(x_0)| = \mathcal{O}(e^{-h_0/2h}) \tag{1.9}$$

Also, we have

$$\Psi_h^n(x_0) - \Phi_h^n(x_0) = \big(\chi_h \circ \Phi_h^n \circ \chi_h^{-1}\big)(x_0) - \Phi_h^n(x_0) = O(h^s), \tag{1.10}$$

because

$$\begin{aligned}
\big(\Phi_h^n \circ \chi_h^{-1}\big)(x_0) &= \Phi_h^n(x_0 + h^s j(x_0,h)) = \Phi_h^n(x_0) + h^s r(x_0,h)\,,\\
\big(\chi_h \circ \Phi_h^n \circ \chi_h^{-1}\big)(x_0) &= \Phi_h^n(x_0) + h^s r(x_0,h) + h^s g(\Phi_h^n(x_0) + h^s r(x_0,h),h)\\
&= \Phi_h^n(x_0) + h^s r'(x_0,h)\,.
\end{aligned}$$

Using a $h$-independent Lipschitz constant for $\widetilde{H}_{N(h),h}$ (see [BG94, Corollary 2]) and the bound (1.10), we observe that the first difference of the right hand side of (1.8) satisfies

$$\widetilde{H}_{N(h),h}(\Psi_h^n(y)) - \widetilde{H}_{N(h),h}(\Phi_h^n(y)) = O(h^s)\,. \tag{1.11}$$

Combining (1.11) and (1.9) in (1.8), we conclude that, for exponentially long time $nh \leq e^{h_0/h}$,

$$\widetilde{H}_{N(h),h}(\Psi_h^n(y)) - \widetilde{H}_{N(h),h}(y) = O(e^{-h_0/2h} + h^s) = O(h^s)\,. \tag{1.12}$$

as claimed in (1.6).

The bound (1.7) on the preservation of the Hamiltonian $H$ follows from the fact that

$$\widetilde{H}_{N(h),h} - H = O(h^p) \tag{1.13}$$

(see [BG94, Equation (1.7)]). □

## 1.3 Structure of the Thesis

In this Thesis, two different issues related to geometric integration are investigated. The first one describes the design of a numerical test which can be employed to assess in a easy way the long-time behavior of a rigid body integrator, in terms of energy preservation, and it is discussed in Part I. The second one presents a straightforward approach to use off-the-shelf Lie methods for the integration of Hamiltonian systems evolving on a product of two spheres, and it is described in Part II.

[3] We remark that $h_0$ depends on the uniform bounds on $D$ of the terms $\phi_i$, $i = 1,2,\dots$ that defined the integrator $\Phi_h$.

# Part I

# A numerical test for the long-time stability of rigid body integrators

# 2 Rigid body integration

Rigid body dynamics plays a central role in many engineering branches such as civil, mechanics, and control engineering, as well as other important scientific disciplines such as physics and chemistry. Obtaining fast and accurate numerical integration schemes for long-time simulation of rigid-body type mechanical systems (e.g., in celestial mechanics and molecular dynamics) is an active area of research. One of the challenging aspects in designing an integration scheme for rigid body dynamics is that the differential equations are defined on a curved space, a Lie group, which is a smooth manifold possessing a group structure.

This Chapter is structured as follows. Basic definitions and basic tools to work with Lie groups are provided in Section 2.1, while in Section 2.2 rigid body dynamics is described. At the end, a survey of the most important geometric integration methods for rigid-body type dynamics is presented in Section 2.3.

## 2.1 Mathematical preliminaries

In this Section we recall some basic concepts about manifolds and Lie group theory, supposing the reader will find useful this outline. A more detailed description (and all the proof) can be found in [Bak02] and in [BR07] respectively for subsection 2.1.1 and 2.1.2.

### 2.1.1 Lie group theory

A *Lie group* $G$ is a group which is also a differentiable manifold[1] and whose group operations are differentiable. We focus on matrix Lie groups, which are subgroups of the Lie group of linear and invertible transformations from $\mathbb{R}^n$ to $\mathbb{R}^n$, $\mathrm{GL}(n)$, represented by nonsingular $n \times n$ matrices. A *Lie algebra* $\mathfrak{g}$ is a vector space together with a bilinear map $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$,

[1] The definition of *group* and *differentiable manifold* can be found, respectively, in [Hun74] and in [KN63].

called the *Lie bracket* or *commutator*, such that for each $a, b, c \in \mathfrak{g}$

$$[a,b] = -[b,a] \quad \text{(skew-symmetry)}$$
$$[a,[b,c]] + [b,[a,c]] + [c,[a,b]] = 0 \quad \text{(Jacobi identity)}$$

hold. For matrix Lie algebras, Lie bracket is given by the matrix commutator:

$$[A,B] = AB - BA.$$

A *differentiable curve* on a group $G$ is a function $\gamma : (a,b) \to G$ such that

$$\gamma'(t) = \lim_{s \to t} \frac{1}{(s-t)} \left[\gamma(s) - \gamma(t)\right] \in \mathbb{R}^{n \times n}$$

exists $\forall t \in (a,b)$. Be $\gamma$ a differential curve on $G$, $\gamma(0) = g$. The *tangent space* to the group $G$ at $g \in G$ is

$$T_g G = \{\gamma'(0) \in \mathbb{R}^{n \times n}\}.$$

A tangent space is a vector space; the tangent space of a group $G$ at the identity element $e$ is a Lie algebra [Bak02], and we write

$$\mathfrak{g} = T_e G.$$

A *rotation* is an isometry[2] which also preserves handedness of the reference system. Such isometries form the *special orthogonal group* $\mathrm{SO}(n) \subset \mathrm{GL}(n)$, that is, the $n$- dimensional set of all $n \times n$ orthogonal matrix with determinant $+1$. Since we are interested into rigid bodies, we will deal with the Lie group SO(3), the set of $3 \times 3$ rotation matrices together with the binary operation given by the standard matrix multiplication.

In literature, several ways to represent SO(3) have been introduced (see, e.g., [Shu93] for a survey): in this work we will employ matrix representation, exponential coordinates (see subsection 2.1.2), and rotation vector representation, also known as Euler vector representation. In the last case, the set of all rotation matrices is depicted as a $\pi-$radius ball, where each vector corresponds to a rotation around its direction, while the angle of rotation is determined by the module of the vector[3].

**Lemma 2.1.1.** *The Lie algebra of* SO(3) *is* $\mathfrak{so}(3)$*, the vector space of skew-symmetric matrices.*

*Proof.* See [Bak02, Chapter 3]. □

[2] An *isometry* is a geometrical transformation which preserves distances between points.

[3] In order to obtain a one-to-one description, we should actually remove double representations of $\pi$-rotations. For example, we can consider the set

$$B = \{(x,y,z) \in \mathbb{R}^3 | \|(x,y,z)\| \leq 1 \text{ if } z > 0, \text{ or } z = 0, y > 0, \text{ or } z = 0, y = 0, x \neq -1, \|(x,y,z)\| < 1 \text{ if } z < 0\}.$$

An element of $\mathfrak{so}(3)$ can be indifferently represented as a skew-symmetric matrix or as a 3-vector, since $\mathfrak{so}(3)$ is isomorphic to $\mathbb{R}^3$ via the *hat map* $\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$:

$$\hat{\xi} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix}.$$

The inverse of the hat map is the *vee* map $\vee : \mathfrak{so}(3) \to \mathbb{R}^3$:

$$\begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix}^{\vee} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}.$$

We will represent an element of $\mathfrak{so}(3)$ as a vector or as a skew-symmetric matrix according to our convenience. Doing this, we imply that the basis of $\mathbb{R}^3$ is the canonical orthonormal basis

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

while $\mathfrak{so}(3)$ is spanned by $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$. The matrix commutator bracket on $\mathfrak{so}(3)$ corresponds to the cross product in $\mathbb{R}^3$ under the identification $\mathfrak{so}(3) \cong \mathbb{R}^3$.

### 2.1.2 Canonical coordinates of the first kind

Consider a map $F : \mathfrak{so}(3) \to \mathrm{SO}(3)$, analytic and local diffeomorphism, mapping a neighborhood of zero in $\mathfrak{so}(3)$ to one of identity in SO(3). Such a function provides a local chart for SO(3) at the identity, and an atlas can be constructed using left (or right) translation of this map.

**Definition 2.1.2** (Canonical coordinates)**.** *The local coordinates associated with the map $F$ are called canonical coordinates (of the first kind).*

For our purposes, we will use as $F$ the exponential matrix mapping $\exp : \mathfrak{so}(3) \to \mathrm{SO}(3)$ defined as

$$\exp \eta = \sum_{i=0}^{\infty} \frac{\hat{\eta}^i}{i!}. \tag{2.1}$$

or a second-order approximation to the matrix exponential, the Cayley map, which is the Padé (1,1) approximant:

$$\operatorname{cay}(x) = \left(I + \frac{1}{2}\hat{x}\right)\left(I - \frac{1}{2}\hat{x}\right)^{-1}. \tag{2.2}$$

The Cayley map is still a locally diffeomorphic analytic function mapping a neighborhood of zero of $\mathfrak{so}(3)$ to one of the identity in SO(3) [IMKNZ00].

The following Lemmas will provide some useful representations for the map $F$ and its derivatives.

**Lemma 2.1.3.** *Be $x \in \mathfrak{so}(3)$, and $\theta = \|x\|$. Any analytic function $F : \mathfrak{so}(3) \to \mathrm{SO}(3)$ can be written as*

$$F(\hat{x}) = c_0(\theta)I + c_1(\theta)\hat{x} + c_2(\theta)\hat{x}^2, \tag{2.3}$$

*where $I$ is the $3 \times 3$ identity matrix.*

*Proof.* It is easily proved noting that $\hat{x}^3 = -\theta^2\hat{x}$, $\hat{x}^4 = -\theta^2\hat{x}^2$. □

**Definition 2.1.4** (Right-trivialized derivative)**.** *Given a smooth function $F : \mathfrak{so}(3) \to \mathrm{SO}(3)$, the right-trivialized derivative is the function $\mathrm{d}F : \mathfrak{so}(3) \times \mathfrak{so}(3) \to \mathfrak{so}(3)$ defined as*

$$\mathrm{D}F(x) \cdot \delta = \left(\mathrm{d}F_x \cdot \delta\right) F(x).$$

**Lemma 2.1.5.** *If $F$ is given by* (2.3)*, then its right-trivialized tangent is given by*

$$\mathrm{d}F_x = c_1(\theta)I + c_2(\theta)\hat{x} + c_3(\theta)x\,x^T, \tag{2.4}$$

*where*

$$c_3(\theta) = \frac{c_2'(\theta) + c_2^2(\theta)\theta}{c_1(\theta)\theta}. \tag{2.5}$$

*Proof.* See [BR07, Lemma 5.4.2]. We note that there is an error in the last but one passage at page 73: $\dfrac{1}{2\theta}$ should be substituted with $\dfrac{1}{2}$. □

Let us specify the coefficient $c_1, c_2, c_3$ for the exponential and the Cayley mappings.

**Proposition 2.1.6.** *For the exponential map, the following formulas hold:*

$$\exp(x) = I + \frac{\sin\theta}{\theta}\hat{x} + \frac{1-\cos\theta}{\theta^2}\hat{x}^2 \tag{2.6}$$

$$\mathrm{d}\exp_x = \frac{\sin\theta}{\theta}I + \frac{1-\cos\theta}{\theta^2}\hat{x} + \frac{\theta-\sin\theta}{\theta^3}xx^T \tag{2.7}$$

*Proof.* Equation (2.6) is obtained by the definition (2.1) using the Taylor series of sine and cosine. It is well known in literature as the Rodrigues formula [MR99]. The application of Lemma 2.1.5 yields the expression of the right trivialized tangent[4]. □

**Proposition 2.1.7.** *For the Cayley map, the following formulas hold:*

$$\mathrm{cay}(x) = I + \frac{4}{4+\theta^2}\hat{x} + \frac{2}{4+\theta^2}\hat{x}^2 \tag{2.8}$$

$$\mathrm{d}\,\mathrm{cay}_x = \frac{4}{4+\theta^2}I + \frac{2}{4+\theta^2}\hat{x} \tag{2.9}$$

[4] Another equivalent expression for $\mathrm{d}\exp$ is given by [IMKNZ00]

$$\mathrm{d}\exp_x = I + \frac{1-\cos\theta}{\theta^2}\hat{x} + \frac{\theta\sin\theta}{\theta^3}\hat{x}^2.$$

The equivalence between the two formulas can be proved using the formula

$$\hat{x}^2 = xx^T - \|x\|^2 I.$$

*Proof.* The following identity [IMKNZ00] results to be useful:

$$(I-\hat{x})^{-1}=I+\frac{1}{1+\theta^2}\hat{x}+\frac{1}{1+\theta^2}\hat{x}^2.$$

Direct computation from (2.2) yields:

$$\begin{aligned}\operatorname{cay}(x)&=\left(I-\frac{1}{2}\hat{x}\right)\left(I+\frac{2}{4+\theta^2}\hat{x}+\frac{4}{4+\theta^2}\hat{x}^2\right)\\&=I-\frac{16+4(4+\theta^2)-4\theta^2}{8(4+\theta^2)}\hat{x}+\frac{2}{4+\theta^2}\hat{x}^2\\&=I+\frac{4}{4+\theta^2}\hat{x}\frac{2}{4+\theta^2}\hat{x}^2.\end{aligned}$$

The expression of the right-trivialized tangent follows directly from Lemma 2.1.5. □

The fact that exponential and the Cayley mappings take values in SO(3) leads to

$$\left[\exp(x)\right]^{-1}=\left[\exp(x)\right]^T, \tag{2.10}$$

$$\left[\operatorname{cay}(x)\right]^{-1}=\left[\operatorname{cay}(x)\right]^T. \tag{2.11}$$

The following Proposition exploits an important property of exponential and Cayley maps.

**Proposition 2.1.8.** *These formulas hold:*

$$\exp(x)\exp(-x)=\exp(-x)\exp(x)=I, \tag{2.12}$$

$$\operatorname{cay}(x)\operatorname{cay}(-x)=\operatorname{cay}(-x)\operatorname{cay}(x)=I. \tag{2.13}$$

*Proof.* Equation (2.13) can be proved in a straightforward manner using direct computation. In virtue of (2.10), we only need to check that $\exp(-x)=\left[\exp(x)\right]^T$ in order to prove (2.12). This is easily verified remembering that

$$\hat{x}^T=(-x)^\wedge$$

and using the definition of exponential map (2.1). □

## 2.2 Rigid body dynamics

In this Section we recall the rotational dynamics of a rigid body in a static configuration-dependent potential field. Denote by $\mathbf{R}(t)\in\mathrm{SO}(3)$, $\boldsymbol{\omega}(t)=\left[\omega_1(t)\ \omega_2(t)\ \omega_3(t)\right]^T\in\mathfrak{so}(3)$, and $\mathbf{J}=\operatorname{diag}(J_1,J_2,J_3)\in\mathbb{R}^{3\times3}$ the configuration, the body angular velocity and the (time-independent) body-fixed inertia tensor, respectively. Then be $\boldsymbol{\tau}:\mathrm{SO}(3)\to\mathbb{R}^3$ a configuration-dependent torque acting on the body, expressed in body coordinates. In terms of this notation, the governing equations are

$$\begin{cases}\dot{\mathbf{R}} &=\mathbf{R}\,\hat{\boldsymbol{\omega}}, \qquad \text{(reconstruction equation)} & (2.14a)\\ \mathbf{J}\,\dot{\boldsymbol{\omega}} &=\mathbf{J}\,\boldsymbol{\omega}\times\boldsymbol{\omega}+\boldsymbol{\tau}(\mathbf{R}), \qquad \text{(Euler's equation)} & (2.14b)\end{cases}$$

with initial conditions $(\mathbf{R}(0), \boldsymbol{\omega}(0)) = (\mathbf{R}_0, \boldsymbol{\omega}_0) \in \mathrm{SO}(3) \times \mathfrak{so}(3)$.

We assume that the (left-trivialized) rotation dynamics derives from a (left-trivialized) Lagrangian $L : \mathrm{SO}(3) \times \mathfrak{so}(3) \to \mathbb{R}$ of the form

$$L(\mathbf{R}, \boldsymbol{\omega}) = T(\boldsymbol{\omega}) - U(\mathbf{R}),$$

where

$$T(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^T \mathbf{J}\, \boldsymbol{\omega}$$

is the kinetic energy and $U(\mathbf{R})$ is the potential energy. For this to hold, the torque $\boldsymbol{\tau}(\mathbf{R})$ that appears in (2.14b) has to be obtained from the left-trivialized derivative of the potential energy $U$ at $\mathbf{R}$.

**Definition 2.2.1** (Left-trivialized derivative). *Consider the function $U : \mathrm{SO}(3) \to \mathbb{R}$. Its left trivialized derivative is the function $\mathrm{d}\, U : \mathfrak{so}(3) \to \mathbb{R}$ that satisfies*

$$\mathrm{D}\, U(\mathbf{R}) \cdot \delta\mathbf{R} = \mathrm{d}\, U(\mathbf{R}) \cdot \mathbf{R}^T \delta\mathbf{R},$$

*where $\delta\mathbf{R} \in T_{\mathbf{R}}\, \mathrm{SO}(3)$. Note that $\mathrm{d}\, U \in \mathfrak{so}^*(3)$, while $\mathrm{D}\, U \in T^*\, \mathrm{SO}(3)$.*

The torque $\boldsymbol{\tau}(\mathbf{R})$ is then given by

$$\boldsymbol{\tau}(\mathbf{R}) = -\,\mathrm{d}\, U(\mathbf{R}). \tag{2.15}$$

We recall that through the (left-trivialized) Legendre transform

$$\boldsymbol{\mu} = \frac{\partial}{\partial\, \boldsymbol{\omega}} L(\mathbf{R}, \boldsymbol{\omega}) = \mathbf{J}\boldsymbol{\omega},$$

one derives the (left-trivialized) Hamiltonian

$$H(\mathbf{R}, \boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{\mu}^T \mathbf{J}^{-1}\, \boldsymbol{\mu} + U(\mathbf{R})$$

expressed in term of the configuration $\mathbf{R}$ and body angular momentum $\boldsymbol{\mu}$. From the Hamiltonian, one can then derive a set of Hamiltonian equations equivalent to (2.14) (see, e.g., [MR99]).

We remark that $H$ is separable and that this Hamiltonian system is reversible [HLW06], since $H(\mathbf{R}, -\boldsymbol{\mu}) = H(\mathbf{R}, \boldsymbol{\mu})$. As all Hamiltonian systems, the exact continuous-time flow of (2.14) is symplectic and preserves the Hamiltonian $H$. Through the Legendre transform, this also implies that the energy $E(\mathbf{R}, \boldsymbol{\omega}) = T(\boldsymbol{\omega}) + U(\mathbf{R})$ is preserved.

## 2.3 Literature review

In this Section we present a survey of the literature about integrators for conservative rigid body systems. This survey is limited to research papers which address the problem of integrating the dynamics of a rigid body in a generic configuration-dependent potential field.

Since the flow of a Hamiltonian system (such as that of a rigid body in a potential field) preserves the associated Hamiltonian, the canonical symplectic form, and (if a group symmetry is present) the corresponding momentum map, we will emphasize the behavior of the methods related to the preservation of these quantities. We group the algorithms into two categories: those approximating the flow of the equations of motion and those approximating the variational principle from which the Hamiltonian differential equations arise. The algorithms that belong to the second class are symplectic by construction; and therefore their good long-time behavior is guaranteed by backward error analysis. the first class of algorithms is instead more varied, with some algorithms known to be symplectic, other known to be not, and some other claimed to be based on numerical simulations.

**Approximation of Hamiltonian differential equations.** Simo and collaborators have developed a substantial body of work on rigid body integrators [SVQ88, SW91, LS94b]. This work was originally motivated by the need to develop conserving algorithms that efficiently simulate the structural dynamics of rods and shells. In the first paper [SVQ88], the classical Newmark scheme for integration of mechanical systems is extended to the rigid body configuration space, the Lie group SO(3). It was not apparent to these investigators if the proposed Lie-Newmark methods has the necessary structure-preserving properties. In fact, later Simo and Wong [SW91] propose another set of algorithms which preserve momentum enforcing its rate balance at an intermediate time step. Afterwards, Austin *et al.* [AKW93] understand that the midpoint rule applied to Euler's equations with a Cayley reconstruction procedure is, in fact, a simple momentum-preserving method for SO(3) (the method is not a member of the Lie-Newmark family introduced in [SVQ88]). In [LS94b], Lewis and Simo present a symplectic, energy and momentum-preserving integrator for the free rigid body which encompasses, as a particular case, the energy-momentum algorithms of Simo and Wong [SW91] and the midpoint-rule integrator of Austin *et al.* [AKW93], with the further property that the scheme is symplectic. Unfortunately, the energy-momentum algorithm of Simo and Wong loses its preservation properties for a generic potential, although it might be symplectic or energy preserving for some particular choice of the potential energy and in presence of specific group symmetries (e.g., in the case of a heavy top).

In [Kry05], Krysl introduces an integration algorithm which is explicit in the torque evaluation and momentum-preserving (for the free rigid body). The paper proposes a series of numerical tests (namely, free rigid body, fast and slow Lagrangian tops, and rigid body in Coulomb potential with soft wall contact) showing that the accuracy of method outperforms the algorithms by Simo and Wong [SW91], Austin *et al.* [AKW93], and Krysl and Endres [KE05]. Based on numerical evidence, the algorithm is claimed to be symplectic. In a later work [Kry08], Krysl proposes a momentum-preserving form of the trapezoidal rule and of the midpoint rule for the rigid body dynamics. These methods are mutually conjugate and in several numerical tests they exhibit a higher accuracy than the algorithms by Simo and Wong

[SW91] and Austin *et al.* [AKW93].

In [NJ07], Nukala and Shelton propose two schemes, both of which can be thought as splitting methods, based on the ideas of partitioned Runge-Kutta and Crouch-Grossman methods. In the free rigid body case, the algorithms exactly preserve the momentum and are almost Poisson. According to the authors, the numerical results show that those methods exhibit superior performance compared to the algorithms of Simo and Wong [SW91], Lie-Newmark [SVQ88], Moser and Veselov [MV91], Lewis and Simo [LS94b] and Krysl [Kry05]. The tests are conducted on the free rigid body, heavy top, and fast and slow Lagrangian tops.

Recently, Koziara and Bićanić [KB10] proposed a computationally simple explicit method for the rigid body suited for short-term simulations and constrained mechanical systems. For long-time simulations, the authors propose a semi-explicit version which introduces a slight increase in the computational complexity.

**Approximation of Hamiltonian variational principle.** In a famous paper, Moser and Veselov [MV91] derive an integrator for the free rigid body by embedding SO(3) in the linear space of $3 \times 3$ matrices and using Lagrange multipliers to constrain the body configuration to SO(3). The discrete Moser-Veselov is a particular case of the RATTLE algorithm for matrix Lie groups [MS95, MZ05]. The RATTLE scheme is a classical method for the integration of constrained Hamiltonian systems. Its application to matrix Lie group, and in particular to rigid body integration, was proposed independently by Reich [Rei94] and McLachlan and Scovel [MS95]. This method is symplectic and momentum-preserving both for the free body case and the generic potential case, and exactly preserves energy only in the former case.

Based on the new approach to symplectic integration proposed by Veselov [Ves98, MV91], who developed a discrete mechanics using a discretization of Hamilton's principle, Marsden and collaborators [MPS99, MW01] have formalized the concept of discrete Lagrangian and discrete Euler-Lagrange equations. This method leads in a natural way to symplectic and momentum-preserving (symplectic-momentum) integrators. In [BRM09], a Runge-Kutta type discretization of the Hamilton-Pontryagin principle for a mechanical system on Lie groups is proposed. This yields, when applied to a rigid body system, to a class of second-order integrators that resemble the Lie-Newmark method but are symplectic-momentum for a generic potential. In [Sac09], Saccon extends the midpoint rule on a generic finite-dimensional Lie groups, showing, as a particular example, how to build a second-order symmetric variational integrators for the rigid body dynamics.

For the reader's convenience, we have organized in Table 2.1 a list of integrators for conservative rigid body systems with a generic potential. In the first three columns, we report the name of the methods as it appeared on the original paper (if no name was given, just the authors' name), corresponding reference, and year of publication. We highlight if the method preserves the canonical symplectic form, energy and spatial momentum, both for the case of a free rigid body and for the case of a rigid body in a generic potential. For this latter case, an integrator which is marked as momentum-preserving conserves the momentum maps arising from a Lie group symmetry. Some of the entries are marked “nearly”, meaning that the error on the preserved quantity is bounded over exponentially long times. A question mark denotes that the conservation of that quantity is not stated nor easily inferred for the corresponding integrator.

| Algorithm | Ref. | Year | *Free rigid body* | | | *Rigid body with generic potential* | | |
|---|---|---|---|---|---|---|---|---|
| | | | Symplectic | Energy | Momentum | Symplectic | Energy | Momentum |
| Lie-Newmark | [SVQ88] | 1988 | ? | | | ? | | |
| Algo_1 | [SW91] | 1991 | | ✔ | | | ? | |
| Algo_C1 | [SW91] | 1991 | | ✔ | ✔ | | ? | ? |
| Austin et al. | [AKW93] | 1993 | | ✔ | ✔ | | ? | ? |
| Lewis & Simo | [LS94b] | 1994 | ✔ | ✔ | ✔ | | | |
| RATTLE | [Rei94, MS95] | 1994 | ✔ | ✔ | ✔ | ✔ | nearly | ✔ |
| Variational | [MPS99] | 1998 | ✔ | nearly | ✔ | ✔ | nearly | ✔ |
| LIEMID(EA) | [Kry05] | 2005 | ? | | | ? | | |
| PRK | [NJ07] | 2007 | ? | ✔ | ✔ | ? | ? | ? |
| MCG | [NJ07] | 2007 | ? | ✔ | | ? | ? | |
| NEW3 | [KB10] | 2010 | ? | | ✔ | ? | | |

Table 2.1: Synoptic table of the rigid body integrators and their preserving properties.

# 3

# Simplecticity test

A large number of geometric algorithms for time integration of rigid body rotational dynamics have been proposed in the last 30 years. As seen in the previous Chapter, some of these algorithms [LS94b, MV91, Rei94, MS95, BRM09, Sac09] are symplectic, and therefore their good long-time behavior is assured from backward error analysis. Some other algorithms [SVQ88, SW91, AKW93, KE05, Kry05, NJ07, KB10] have been constructed approximating the continuous-time dynamics using *ad hoc* methodologies, with the declared aim of obtaining computationally fast algorithms with small error constant. In this latter case, the long-time behavior has been assessed using numerical experiments on a series of test cases, monitoring the behavior of the energy and known first integrals, but without any guarantee of good performance over long-time in a generic situation.

It is not clear which of these latter integrators are symplectic. In addition, they could nearly preserve the energy over long time being conjugate-symplectic like Newmark algorithm on vector spaces (see [KMOW00] and references therein). This motivated us to design a simple test to check numerically if a rigid body integrator is symplectic or conjugate-symplectic.

The test we propose in this Section is strongly inspired by a numerical experiment reported in [FHP04, §4.4] even though, unfortunately, it does not possess a similar simple and clear physical interpretation. In [FHP04], a systematic energy drift of a fourth-order accurate, implicit, and symmetric Lobatto IIIB scheme is shown for the integration of the dynamics of a spring pendulum with exterior forces.

In this Chapter we describe the design of our numerical counterexample. We consider a rigid body in a suitable static potential configuration-dependent field; some insights are provided about the potential energy and the exerted torque.

## 3.1 Numerical test

First of all, let us define the function $m : \mathrm{SO}(3) \times \mathrm{SO}(3) \to \mathbb{R}$ as

$$m(\mathbf{R}_1, \mathbf{R}_2) := \sqrt{2\,\mathrm{tr}(\mathbf{I} - \mathbf{R}_1^T \mathbf{R}_2)}, \tag{3.1}$$

with $\mathbf{I} \in SO(3)$ the $3\times3$ identity matrix. Recalling that the Frobenius matrix norm is defined as $\|A\|_F := \sqrt{\operatorname{tr}(A^T A)}$, for $A \in \mathbb{R}^{n\times n}$, it is straightforward to verify that $m(\cdot,\cdot)$ defines the metric on SO(3) induced by the Frobenius norm (use the identity $\|\mathbf{R}_2 - \mathbf{R}_1\|_F^2 = 2\operatorname{tr}(\mathbf{I} - \mathbf{R}_1^T\mathbf{R}_2) = 2\operatorname{tr}(\mathbf{I} - \mathbf{R}_2^T\mathbf{R}_1)$). With a little abuse of notation, we denote $m_{\overline{\mathbf{R}}} : SO(3) \to \mathbb{R}$ the distance from a fixed point $\overline{\mathbf{R}}$, that is,

$$m_{\overline{\mathbf{R}}}(\mathbf{R}) = m(\mathbf{R}, \overline{\mathbf{R}}).$$

We consider a single rigid body in a static potential configuration-dependent field defined by the potential energy function $U_\alpha : SO(3) \to \mathbb{R}$ given by

$$U_\alpha(\mathbf{R}) = \left(m_{\mathbf{I}}(\mathbf{R}) - 1\right)^2 - \frac{\alpha}{m_{\mathbf{R}_m}(\mathbf{R})}. \tag{3.2}$$

The first term in the right hand side of (3.2) is a bounded potential which attains its minimum value at all $\mathbf{R} \in SO(3)$ satisfying $m(\mathbf{R}, \mathbf{I}) = 1$. The second term is an unbounded Coulomb-like potential that generates an attraction toward the configuration $\mathbf{R}_m \in SO(3)$, whose exact value will be discussed in the following as well as that of the tuning parameter $\alpha$.

For $\alpha = 0$ and referring to the rotation vector representation, the potential energy $U_{\alpha=0}$ level sets are concentric spherical surfaces centered at the origin, as it can be seen in Figure 3.1. $U_{\alpha=0}$ achieves its minimum value on the two-dimensional surface

$$S := \{\mathbf{R} \in SO(3) : m_{\mathbf{I}}(\mathbf{R}) = 1\}.$$

The set $S \times \{\mathbf{0}\} \subset SO(3) \times \mathfrak{so}(3)$ results a (locally) stable set in the sense of Lyapunov for the dynamics of the rigid body, as we know from classical mechanics. One can prove this fact using the energy, that is,

$$E(\mathbf{R}, \boldsymbol{\omega}) = T(\boldsymbol{\omega}) + U_{\alpha=0}(\mathbf{R}),$$

as Lyapunov function and noting that the set $\{(\mathbf{R}, \boldsymbol{\omega}) \in SO(3) \times \mathfrak{so}(3) \,|\, E(\mathbf{R}, \boldsymbol{\omega}) \leq \bar{E}\}$ is a compact neighborhood of $S$ for every $\bar{E} \geq 0$, and it reduces to $S$ for $\bar{E} \to 0$. This means that any trajectory starting close enough to the set $S$ will remain close to it forever.

For $\alpha > 0$, the set $S$ gets perturbed by the unbounded attractive potential. On this perturbed energy landscape, the rigid body experiences an attraction toward the configuration $\mathbf{R}_m$. Yet, if we place the attraction point $\mathbf{R}_m$ sufficiently far from the set $S$ and choose the tuning parameter $\alpha > 0$ sufficiently small, the set $S$ gets only slightly perturbed into a new set, that we label $S_\alpha$. Furthermore, the set $S_\alpha \times \{\mathbf{0}\} \subset SO(3) \times \mathfrak{so}(3)$ is locally Lyapunov stable like the unperturbed set $S \times \{\mathbf{0}\}$.

In summary, we can design $U_\alpha$ so that the true solution is confined to a neighborhood of the set $S_\alpha \times \{\mathbf{0}\}$ of approximately known shape, the neighborhood becoming smaller the closer the initial conditions are to the set of stable equilibria $S_\alpha \times \{\mathbf{0}\}$. Under these conditions, the domain of the dynamics will be simply connected[1], and the conditions for the global

[1] Recall that SO(3) is only connected, not simply connected.

existence of a modified Hamiltonian are verified [BG94]: recall that a symplectic integrator is interpolated by a level set of a modified energy function nearby the true energy [BG94, Rei99, HLW06]. This implies that the trajectory of a symplectic integrator is confined to a neighborhood of $S_\alpha \times \{\mathbf{0}\}$ for the whole duration of the simulation.

The torque exerted on the rigid body by (3.2) is obtained using (2.15). The following lemma results to be useful.

**Lemma 3.1.1.** *A function $f \in \mathfrak{so}^*(3)$ can be represented as a 3-coordinate vector $[f_1, f_2, f_3]^T$, where:*

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} \langle f, e_1 \rangle \\ \langle f, e_2 \rangle \\ \langle f, e_3 \rangle \end{bmatrix}.$$

*Proof.* It follows from direct computation, with respect to the canonical basis $\{e_1^*, e_2^*, e_3^*\}$ of $\mathfrak{so}^*(3)$. □

Be $\delta \mathbf{R} = \mathbf{R}\hat{\eta}$, with $\eta \in \mathfrak{so}(3)$. The left-trivialized tangent of $m_{\overline{\mathbf{R}}}$ is given by

$$\mathrm{d}\, m_{\overline{\mathbf{R}}}(\mathbf{R}) \cdot \mathbf{R}^T \delta \mathbf{R} = \mathrm{d}\, m_{\overline{\mathbf{R}}}(\mathbf{R}) \cdot \eta = -\frac{1}{m_{\overline{\mathbf{R}}}(\mathbf{R})} \operatorname{tr}(\overline{\mathbf{R}}^T \mathbf{R}\eta), \tag{3.3}$$

where the linearity of trace has been used to conclude that

$$\mathrm{D}\operatorname{tr}(AX) \cdot \delta X = \operatorname{tr}(A\,\delta X).$$

In coordinate representation, (3.3) becomes

$$\mathrm{d}\, m_{\overline{\mathbf{R}}}(\mathbf{R}) = -\frac{1}{m_{\overline{\mathbf{R}}}(\mathbf{R})} \begin{bmatrix} \operatorname{tr}\left(\overline{\mathbf{R}}^T \mathbf{R}\hat{e}_1\right) \\ \operatorname{tr}\left(\overline{\mathbf{R}}^T \mathbf{R}\hat{e}_2\right) \\ \operatorname{tr}\left(\overline{\mathbf{R}}^T \mathbf{R}\hat{e}_3\right) \end{bmatrix}. \tag{3.4}$$

Explicit computation of the potential torque then yields

$$\begin{aligned} \tau(\mathbf{R}) &= -\,\mathrm{d}\, U_\alpha(\mathbf{R}) \\ &= -2\left(m_{\mathbf{I}}(\mathbf{R}) - 1\right) \mathrm{d}\, m_{\mathbf{I}}(\mathbf{R}) - \frac{\alpha}{\left(m_{\mathbf{R}_m}(\mathbf{R})\right)^2} \mathrm{d}\, m_{\mathbf{R}_m}(\mathbf{R}) \\ &= \frac{2\left(m_{\mathbf{I}}(\mathbf{R}) - 1\right)}{\left(m_{\mathbf{I}}(\mathbf{R})\right)^2} \begin{bmatrix} \operatorname{tr}\left(\mathbf{R}\hat{e}_1\right) \\ \operatorname{tr}\left(\mathbf{R}\hat{e}_2\right) \\ \operatorname{tr}\left(\mathbf{R}\hat{e}_3\right) \end{bmatrix} + \frac{\alpha}{\left(m_{\mathbf{R}_m}(\mathbf{R})\right)^3} \begin{bmatrix} \operatorname{tr}\left(\mathbf{R}_m^T \mathbf{R}\hat{e}_1\right) \\ \operatorname{tr}\left(\mathbf{R}_m^T \mathbf{R}\hat{e}_2\right) \\ \operatorname{tr}\left(\mathbf{R}_m^T \mathbf{R}\hat{e}_3\right) \end{bmatrix}. \end{aligned} \tag{3.5}$$

As we will show in Chapter 4, this experiment has proven to be able to detect an energy drift in a series of algorithms that, on the contrary, show an excellent long-time behavior in standard tests. Thus, it stands out as an effective necessity test for the underlying symplecticity of a Lie group method in the sense that, if an integrator exhibits a systematic drift in total energy, one can conclude that it is not a symplectic neither a conjugate-symplectic integrator for (2.14).

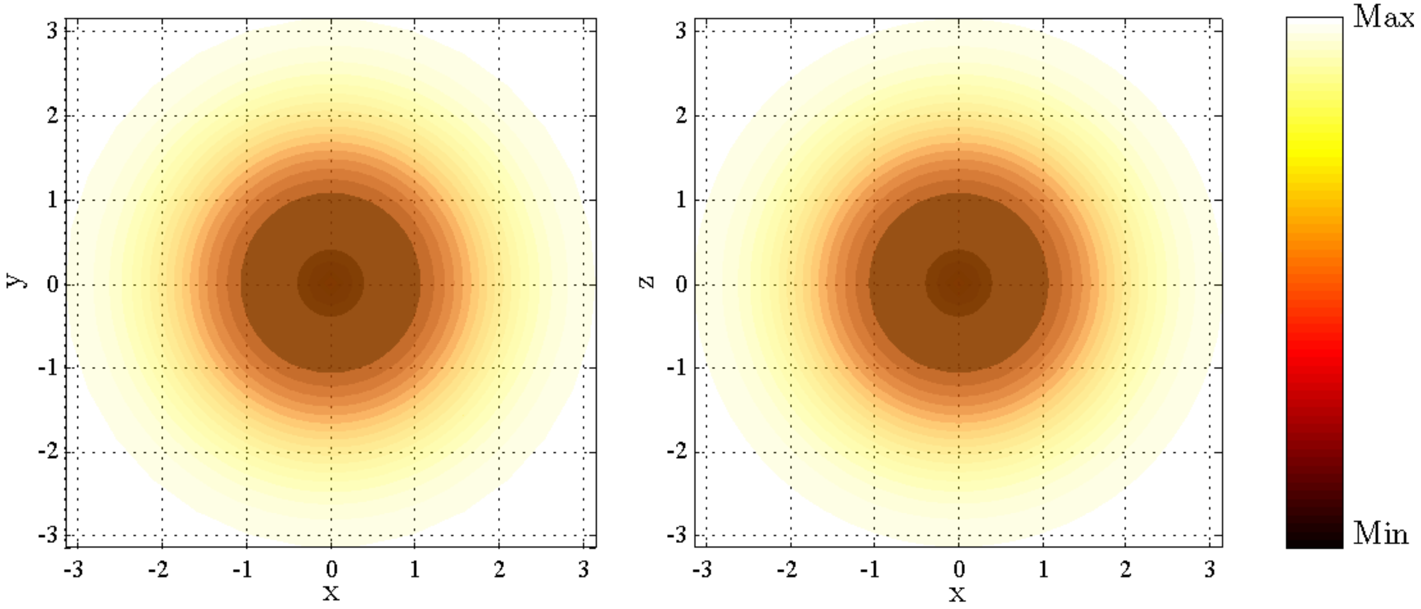


Figure 3.1: Plot of $U_{\alpha=0}(\mathbf{R})$, based on the rotation vector representation of SO(3); the principal axes are named as $x, y, z$, using the conventional notation. The darker the color, the lower the potential value. The position of the viewer is different in the two plots: in terms of cartesian coordinates, the viewpoints in the left and the right figure are respectively $[0,0,1]$ and $[0,1,0]$. The level sets are concentrical spherical surfaces.

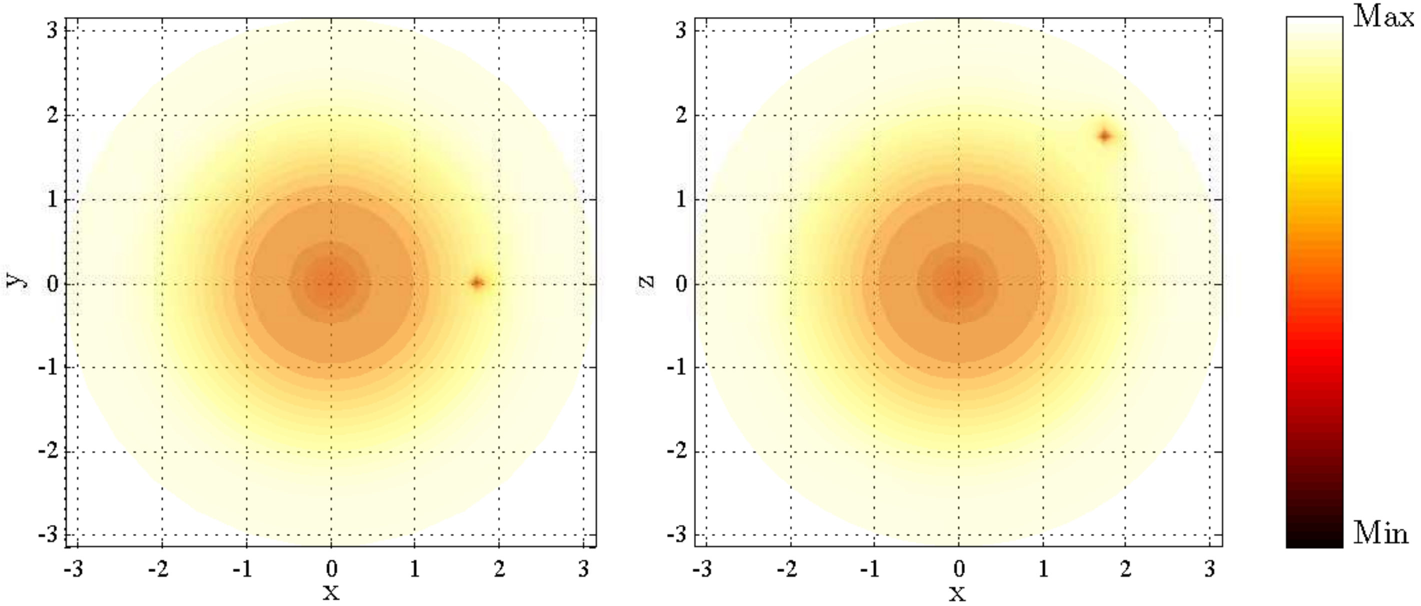


Figure 3.2: Plot of $U_{\alpha=0.3}(\mathbf{R})$, based on the rotation vector representation of SO(3); the principal axes are named as $x, y, z$, using the conventional notation. The darker the color, the lower the potential value. The position of the viewer is different in the two plots: in terms of cartesian coordinates, the viewpoints in the left and the right figure are respectively $[0,0,1]$ and $[0,1,0]$. The value of $\alpha$ and the position of $\mathbf{R}_m$ are discussed in Section 4.2. The minimum value of the potential is attained for $\mathbf{R} = \mathbf{R}_m$.

# 4 Numerical results

In literature, numerical tests which are usually employed to assess the long-time behavior of an integrator are the free rigid body, the slow and the fast Lagrangian top. Nevertheless, we recall that, even if in these tests an algorithm exhibits a good performance in energy or momentum preservation, one has no guarantee about its geometrical properties. As it can be seen from Table 2.3, there are some uncertainties about the symplecticity of a remarkable set of methods; we thus apply the numerical test presented in Chapter 3. A main result of this Thesis is that many rigid body integrators proposed in the literature [SVQ88, Kry05, NJ07, KB10] fail this test, showing that these algorithms can inject or dissipate energy artificially; therefore, they are not symplectic neither conjugate-symplectic.

This Chapter is devoted to the testing of the aforementioned algorithms: in Section 4.1 the methods are described and some details regarding their implementation are given, while the results obtained on our symplecticity test are presented in Section 4.2. Finally, some conclusions are draw in Section 4.3.

## 4.1 Integrators

We test seven different algorithms, most of which appeared in literature in the last ten years. In each subsection a different method is presented, whose identifying acronym is showed in the subsection title. All the algorithms are described using the notation introduced in Chapter 2, and the details about the specific required computations are provided.

### 4.1.1 Explicit Lie-Newmark method (ELN)

A particularly famous class of integrators for the rigid body is that proposed about twenty years ago in [SVQ88]. These methods consist of a Newmark-style discretization [New59] of (2.14b) and a discretization of (2.14a) that ensures that the configuration update remains on SO(3). Following [KMOW00], we will refer to these methods as the Lie-Newmark methods. These algorithms were motivated by the need to develop conserving integrators that can efficiently simulate the structural dynamics of rods and shells. In the paper, Simo and Vu-Quoc consider simulating large deformations of a three-dimensional finite-strain rod model. The rod is discretized using $N$ copies of $\mathbb{R}^3 \times \mathrm{SO}(3)$, where $N$ is the number of discretization points along the line of centroids of the rod. The configuration of each rod segment is specified by an element of SO(3). The dynamical behavior of the rod can then be estimated by simulating the dynamics of $N$ rigid bodies with torques due to elastic coupling between bodies. In [SVQ88], numerical results for the so called trapezoidal scheme ($\gamma = 1/2, \beta = 1/4$) are reported.

For our work, we focus on two specific members of the Lie-Newmark family. The first one is the Lie group analogue of the so called explicit Newmark method on vector spaces (see [HC87, Chapter 9]), known as the Verlet integrator in molecular dynamics [HLW06, Chapter I]. It is obtained choosing $\gamma = 1/2, \beta = 0$ in the general Newmark formulation.

Given the timestep $h$ and the current configuration at the $k$-th instant of time $(\mathbf{R}_k, \boldsymbol{\omega}_k)$, the explicit Lie-Newmark scheme determines the update $(\mathbf{R}_{k+1}, \boldsymbol{\omega}_{k+1})$ by the following iteration rule:

$$\begin{cases} \boldsymbol{\omega}_{k+\frac{1}{2}} = \boldsymbol{\omega}_k + \frac{h}{2}\mathbf{J}^{-1}\left(\mathbf{J}\,\boldsymbol{\omega}_k \times \boldsymbol{\omega}_k + \boldsymbol{\tau}(\mathbf{R}_k)\right), & (4.1a) \\ \mathbf{R}_{k+1} = \mathbf{R}_k \operatorname{cay}\left(h\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right), & (4.1b) \\ \boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_{k+\frac{1}{2}} + \frac{h}{2}\mathbf{J}^{-1}\left(\mathbf{J}\,\boldsymbol{\omega}_{k+1} \times \boldsymbol{\omega}_{k+1} + \boldsymbol{\tau}(\mathbf{R}_{k+1})\right), & (4.1c) \end{cases}$$

where cay is the Cayley map defined in (2.2). As stated before, the Cayley map is a second-order approximation of the exponential map on SO(3). There are other maps one can use in place of the Cayley map in (4.1b) (see, e.g., [BR07, §5.4]) but the Cayley map is known to be computationally less expensive than the exponential map, and it is a usual choice in rigid body integration. As we will discuss in Section 4.2, we have noted no difference in the long-time behavior of the integrator when using the exponential map (as originally proposed in [SVQ88]) instead of the Cayley map, while the overall code running time is minor.

This method is called explicit as it is explicit in the torque evaluation, although the integrator is in fact semi-explicit because the equation (4.1c) is implicit. The implicitness is not severe, however, as the method is only implicit in the angular velocity and not in the attitude. The computation of $\boldsymbol{\omega}_{k+1}$ requires finding the root of the function $f : \mathfrak{so}(3) \cong \mathbb{R}^3 \to \mathfrak{so}(3) \cong \mathbb{R}^3$, given by

$$f(\boldsymbol{\omega}) = -\boldsymbol{\omega} + \boldsymbol{\omega}_{k+\frac{1}{2}} + \frac{h}{2}\mathbf{J}^{-1}\left(\mathbf{J}\,\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau}(\mathbf{R}_{k+1})\right).$$

To this purpose, we use the Newton method [Act90, Chapter 14]; given $\boldsymbol{\omega}^n$ at the $n-$th iteration, next guess $\boldsymbol{\omega}^{n+1}$ is obtained as

$$\boldsymbol{\omega}^{n+1} = \boldsymbol{\omega}^n - J_f^{-1}(\boldsymbol{\omega}^n) f(\boldsymbol{\omega}^n), \tag{4.2}$$

where

$$J_f(\boldsymbol{\omega}) = -\mathbf{I} + \frac{h}{2}\mathbf{J}^{-1}\left[(\mathbf{J}\,\boldsymbol{\omega})^\wedge - \hat{\boldsymbol{\omega}}\mathbf{J}\right]$$

is the Jacobian matrix of $f$. The method is initialized with $\boldsymbol{\omega}^0 = \boldsymbol{\omega}_{k+\frac{1}{2}}$. The procedure is iterated until $\|f(\boldsymbol{\omega}^n)\| < 10^{-12}$, then $\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}^n$.

### 4.1.2 Trapezoidal Lie-Newmark method (TLN)

The second Lie-Newmark method we consider in this paper is the so called Trapezoidal Lie-Newmark algorithm, which is the Lie group analogue of the well-known trapezoidal rule on vector spaces. The second-order accuracy of this algorithm has been proved in [SVQ88]. Given $(\mathbf{R}_k, \boldsymbol{\omega}_k)$ and timestep $h$, the TLN iteration rule is given by

$$\begin{cases} \left(\mathbf{I} + \frac{h}{2}\hat{\boldsymbol{\omega}}_{k+1}\right)\mathbf{J}\,\boldsymbol{\omega}_{k+1} - \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_{k+1}) = \left(\mathbf{I} - \frac{h}{2}\hat{\boldsymbol{\omega}}_k\right)\mathbf{J}\,\boldsymbol{\omega}_k + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_k), & (4.3a) \\ \mathbf{R}_{k+1} = \mathbf{R}_k \operatorname{cay}\left(h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}_{k+1}}{2}\right). & (4.3b) \end{cases}$$

As for the ELN method, we have experienced no differences in the long-time behavior for the choice of the Cayley map instead of the exact exponential map in the reconstruction equation, with a benefit of a minor overall running time. This method is fully implicit and it requires the computation of the first derivative of the torque expression (3.5) when, as in our case, a Newton method is employed to solve the nonlinear set of equations. We think it is worth to show the calculations in detail. First of all, define a function $G_k : \mathfrak{so}(3) \to \mathrm{SO}(3)$ as

$$G_k(\boldsymbol{\omega}) = \mathbf{R}_k \operatorname{cay}\left(h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}}{2}\right), \; \mathbf{R}_k \in \mathrm{SO}(3), \; \boldsymbol{\omega}_k \in \mathfrak{so}(3).$$

Its derivative $\mathrm{D}\,G_k : T\mathfrak{so}(3) \cong \mathfrak{so}(3) \to T\,\mathrm{SO}(3)$ can be written as

$$\mathrm{D}\,G_k(\boldsymbol{\omega}) \cdot \delta = \frac{h}{2}\mathbf{R}_k\left(\mathrm{d}\operatorname{cay}_{h\frac{\omega_k+\omega}{2}}\delta\right)^{\wedge} \operatorname{cay}\left(h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}}{2}\right). \tag{4.4}$$

Thus, if we consider the composite map $m_{\overline{\mathbf{R}}} \circ G_k : \mathfrak{so}(3) \to \mathbb{R}$, its tangent map can be computed as

$$\begin{aligned} \mathrm{D}(m_{\overline{\mathbf{R}}} \circ G_k)(\boldsymbol{\omega}) \cdot \delta &= \mathrm{d}\, m_{\overline{\mathbf{R}}}(G_k(\boldsymbol{\omega})) \cdot \left[G_k^T(\boldsymbol{\omega})\,\mathrm{D}\,G_k(\boldsymbol{\omega}) \cdot \delta\right] \\ &= -\frac{h}{2m_{\overline{\mathbf{R}}}\left(G_k(\boldsymbol{\omega})\right)} \begin{bmatrix} \operatorname{tr}\left(\overline{\mathbf{R}}^T G_k(\boldsymbol{\omega})\hat{e}_1\right) \\ \operatorname{tr}\left(\overline{\mathbf{R}}^T G_k(\boldsymbol{\omega})\hat{e}_2\right) \\ \operatorname{tr}\left(\overline{\mathbf{R}}^T G_k(\boldsymbol{\omega})\hat{e}_3\right) \end{bmatrix}^T \operatorname{cay}\left(-h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}}{2}\right) \mathrm{d}\operatorname{cay}_{h\frac{\omega_k+\omega}{2}} \cdot \delta, \end{aligned}$$

where the property of rotation matrix $\mathbf{R}\hat{\boldsymbol{\omega}}\mathbf{R}^T = (\mathbf{R}\boldsymbol{\omega})^{\wedge}$ has been used. Therefore, we can write $\mathrm{D}(m_{\overline{\mathbf{R}}} \circ G_k)(\boldsymbol{\omega})$ as a linear application

$$\mathrm{D}(m_{\overline{\mathbf{R}}} \circ G_k)(\boldsymbol{\omega}) = -\frac{h}{2m_{\overline{\mathbf{R}}}\left(G_k(\boldsymbol{\omega})\right)} \begin{bmatrix} \operatorname{tr}\left(\overline{\mathbf{R}}^T G_k(\boldsymbol{\omega})\hat{e}_1\right) \\ \operatorname{tr}\left(\overline{\mathbf{R}}^T G_k(\boldsymbol{\omega})\hat{e}_2\right) \\ \operatorname{tr}\left(\overline{\mathbf{R}}^T G_k(\boldsymbol{\omega})\hat{e}_3\right) \end{bmatrix}^T \operatorname{cay}\left(-h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}}{2}\right) \mathrm{d}\operatorname{cay}_{h\frac{\omega_k+\omega}{2}}.$$

Then denote with $y_i : \mathfrak{so}(3) \to \mathbb{R}$ the function

$$y_i(\boldsymbol{\omega}) = \operatorname{tr}\big(G_k(\boldsymbol{\omega})\hat{e}_i\big),$$

whose directional derivative along $\delta = \sum_{j=1}^{3} \delta^j e_j$ can be written as

$$\begin{aligned} \mathrm{D}\, y_i(\boldsymbol{\omega}) \cdot \delta &= \operatorname{tr}\left([\mathrm{D}\, G_k(\boldsymbol{\omega}) \cdot \delta]\hat{e}_i\right) = \operatorname{tr}\left(\left[\mathrm{D}\, G_k(\boldsymbol{\omega}) \cdot \sum_{j=1}^{3} \delta^j e_j\right] \hat{e}_i\right) = \operatorname{tr}\left(\sum_{j=1}^{3} (\mathrm{D}\, G_k(\boldsymbol{\omega}) \cdot \delta^j e_j)\hat{e}_i\right) \\ &= \sum_{j=1}^{3} \delta^j \operatorname{tr}\left((G_k(\boldsymbol{\omega}) \cdot e_j)\hat{e}_i\right). \end{aligned}$$

Lemma 3.1.1 allows us to represent the tangent map of $y_i$ in coordinates as an element of $\mathfrak{so}^*(3)$:

$$\mathrm{D} f_i(\boldsymbol{\omega}) = \begin{bmatrix} \operatorname{tr}\left(\left(\mathrm{D}\, G_k(\boldsymbol{\omega}) \cdot e_1\right) \hat{e}_i\right) \\ \operatorname{tr}\left(\left(\mathrm{D}\, G_k(\boldsymbol{\omega}) \cdot e_2\right) \hat{e}_i\right) \\ \operatorname{tr}\left(\left(\mathrm{D}\, G_k(\boldsymbol{\omega}) \cdot e_3\right) \hat{e}_i\right) \end{bmatrix}. \tag{4.5}$$

The angular velocity update $\boldsymbol{\omega}_{k+1}$ can now be obtained by applying the Newton method to the function $f : \mathfrak{so}(3) \cong \mathbb{R}^3 \to \mathfrak{so}(3) \cong \mathbb{R}^3$:

$$f(\boldsymbol{\omega}) = -\mathbf{J}\, \boldsymbol{\omega} - \frac{h}{2} \hat{\boldsymbol{\omega}} \mathbf{J}\, \boldsymbol{\omega} + \frac{h}{2} \boldsymbol{\tau}\left(G_k(\boldsymbol{\omega})\right) + \left(I - \frac{h}{2}\hat{\boldsymbol{\omega}}_k\right) \mathbf{J}\, \boldsymbol{\omega}_k + \frac{h}{2} \boldsymbol{\tau}(\mathbf{R}_k).$$

Given $\boldsymbol{\omega}^n$ at the $n-$th iteration, next guess $\boldsymbol{\omega}^{n+1}$ is given by (4.2), where the Jacobian matrix $J_f$ follows from some tedious computations:

$$\begin{aligned} J_f(\boldsymbol{\omega}^n) = &-\mathbf{J} - \frac{h}{2}\left[(\mathbf{J}\, \boldsymbol{\omega}^n)^\wedge + \hat{\boldsymbol{\omega}}^n \mathbf{J}\right] + \\ &+ \frac{h^2}{2\big(m_{\mathbf{I}}(G_k(\boldsymbol{\omega}^n))\big)^3} \begin{bmatrix} \operatorname{tr}\left(G_k(\boldsymbol{\omega}^n)\hat{e}_1\right) \\ \operatorname{tr}\left(G_k(\boldsymbol{\omega}^n)\hat{e}_2\right) \\ \operatorname{tr}\left(G_k(\boldsymbol{\omega}^n)\hat{e}_3\right) \end{bmatrix} \begin{bmatrix} \operatorname{tr}\left(G_k(\boldsymbol{\omega}^n)\hat{e}_1\right) \\ \operatorname{tr}\left(G_k(\boldsymbol{\omega}^n)\hat{e}_2\right) \\ \operatorname{tr}\left(G_k(\boldsymbol{\omega}^n)\hat{e}_3\right) \end{bmatrix}^T \operatorname{cay}\left(-h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}^n}{2}\right) \mathrm{d}\operatorname{cay}_{h\frac{\boldsymbol{\omega}_k+\boldsymbol{\omega}^n}{2}} + \\ &\quad + \frac{h^2}{2}\left[1 - \frac{1}{m_{\mathbf{I}}(G_k(\boldsymbol{\omega}^n))}\right] \left[\operatorname{tr}\left(\mathbf{R}_k \left(\mathrm{d}\operatorname{cay}_{h\frac{\boldsymbol{\omega}_k+\boldsymbol{\omega}^n}{2}} \cdot e_j\right)^\wedge \operatorname{cay}\left(h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}^n}{2}\right) \hat{e}_i\right)\right]_{ij} + \\ &+ \frac{3h^2\alpha}{4\big(m_{\mathbf{R}_m}(G_k(\boldsymbol{\omega}^n))\big)^5} \begin{bmatrix} \operatorname{tr}\left(\mathbf{R}_m^T G_k(\boldsymbol{\omega}^n)\hat{e}_1\right) \\ \operatorname{tr}\left(\mathbf{R}_m^T G_k(\boldsymbol{\omega}^n)\hat{e}_2\right) \\ \operatorname{tr}\left(\mathbf{R}_m^T G_k(\boldsymbol{\omega}^n)\hat{e}_3\right) \end{bmatrix} \begin{bmatrix} \operatorname{tr}\left(\mathbf{R}_m^T G_k(\boldsymbol{\omega}^n)\hat{e}_1\right) \\ \operatorname{tr}\left(\mathbf{R}_m^T G_k(\boldsymbol{\omega}^n)\hat{e}_2\right) \\ \operatorname{tr}\left(\mathbf{R}_m^T G_k(\boldsymbol{\omega}^n)\hat{e}_3\right) \end{bmatrix}^T \operatorname{cay}\left(-h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}^n}{2}\right) \mathrm{dcay}_{h\frac{\boldsymbol{\omega}_k+\boldsymbol{\omega}^n}{2}} + \\ &\quad + \frac{h^2\alpha}{4\big(m_{\mathbf{R}_m}(G_k(\boldsymbol{\omega}^n))\big)^3} \left[\operatorname{tr}\left(\mathbf{R}_m^T \mathbf{R}_k \left(\mathrm{d}\operatorname{cay}_{h\frac{\boldsymbol{\omega}_k+\boldsymbol{\omega}^n}{2}} \cdot e_j\right)^\wedge \operatorname{cay}\left(h\frac{\boldsymbol{\omega}_k + \boldsymbol{\omega}^n}{2}\right) \hat{e}_i\right)\right]_{ij}, \end{aligned}$$

where $[\cdot]_{ij}$ denotes the $(i, j)$ element of a $3\times 3$ matrix. The method is initialized with $\boldsymbol{\omega}^0 = \boldsymbol{\omega}_k$. The procedure is iterated until $\|f(\boldsymbol{\omega}^n)\| < 10^{-12}$, then $\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}^n$.

### 4.1.3 Explicit Lie-Midpoint algorithm (LIEMID[EA])

In [Kry05], Krysl introduces this integration algorithm, derived as the composition of a half-step of a first-order midpoint Lie method with its adjoint. This method is therefore symmetric and second-order accurate; besides this, in the free rigid body case, it preserves exactly the spatial angular momentum. Due to its good behavior in preserving the energy, in [Kry05] it is claimed that the method is symplectic.

Given $(\mathbf{R}_k, \boldsymbol{\omega}_k)$ and timestep $h$, the algorithm determines $(\mathbf{R}_{k+1}, \boldsymbol{\omega}_{k+1})$ using the following iteration rule:

$$\left\{\begin{aligned}
\boldsymbol{\Theta}_{k+\frac{1}{2}} &= \frac{h}{2}\mathbf{J}^{-1}\exp\left(-\frac{1}{2}\boldsymbol{\Theta}_{k+\frac{1}{2}}\right)\left(\mathbf{J}\boldsymbol{\omega}_k + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_k)\right), && \text{(4.6a)}\\
\mathbf{R}_{k+\frac{1}{2}} &= \mathbf{R}_k \exp\left(\boldsymbol{\Theta}_{k+\frac{1}{2}}\right), && \text{(4.6b)}\\
\boldsymbol{\omega}_{k+\frac{1}{2}} &= \mathbf{J}^{-1}\exp\left(-\boldsymbol{\Theta}_{k+\frac{1}{2}}\right)\left(\mathbf{J}\boldsymbol{\omega}_k + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_k)\right), && \text{(4.6c)}\\
\boldsymbol{\Theta}_{k+1} &= \frac{h}{2}\mathbf{J}^{-1}\exp\left(-\frac{1}{2}\boldsymbol{\Theta}_{k+1}\right)\mathbf{J}\boldsymbol{\omega}_{k+\frac{1}{2}}, && \text{(4.6d)}\\
\mathbf{R}_{k+1} &= \mathbf{R}_{k+\frac{1}{2}}\exp\left(\boldsymbol{\Theta}_{k+1}\right), && \text{(4.6e)}\\
\boldsymbol{\omega}_{k+1} &= \mathbf{J}^{-1}(\exp\left(-\boldsymbol{\Theta}_{k+1}\right)\mathbf{J}\boldsymbol{\omega}_{k+\frac{1}{2}} + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_{k+1})), && \text{(4.6f)}
\end{aligned}\right.$$

where exp is the matrix exponential map introduced in (2.1). The updates (4.6a) and (4.6d) are both implicit, and therefore the algorithm involves the solution of two nonlinear equations per step, i.e., two Newton solvers are employed. Nevertheless, since they are not implicit in the body attitude, the implicitness is not hard. The four remaining updates are explicit.

Let us show how the Newton method applies to this case. For (4.6a), consider the 3-dimensional function $f : \mathfrak{so}(3) \cong \mathbb{R}^3 \to \mathfrak{so}(3) \cong \mathbb{R}^3$

$$f(\boldsymbol{\Theta}) = -\boldsymbol{\Theta} + \frac{h}{2}\mathbf{J}^{-1}\exp\left(-\frac{1}{2}\boldsymbol{\Theta}\right)\left(\mathbf{J}\boldsymbol{\omega} + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_k)\right);$$

its Jacobian matrix $J_f(\boldsymbol{\Theta})$ can be read from

$$\begin{aligned}
\mathrm{D}f(\boldsymbol{\Theta})\cdot\delta &= -\delta - \frac{h}{4}\mathbf{J}^{-1}\left(\mathrm{d}\exp_{-\frac{\boldsymbol{\Theta}}{2}}\cdot\delta\right)^{\wedge}\exp\left(\frac{\boldsymbol{\Theta}}{2}\right)\left(\mathbf{J}\boldsymbol{\omega} + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R})\right)\\
&= \left[-\mathbf{I} + \frac{h}{4}\mathbf{J}^{-1}\left(\exp\left(-\frac{\boldsymbol{\Theta}}{2}\right)\left(\mathbf{J}\boldsymbol{\omega} + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R})\right)\right)^{\wedge}\mathrm{d}\exp_{-\frac{\boldsymbol{\Theta}}{2}}\right]\cdot\delta.
\end{aligned}$$

For what concern the solution of (4.6d), define the 3-dimensional function $f : \mathfrak{so}(3) \cong \mathbb{R}^3 \to \mathfrak{so}(3) \cong \mathbb{R}^3$

$$f(\boldsymbol{\Theta}) = -\boldsymbol{\Theta} + \frac{h}{2}\mathbf{J}^{-1}\exp\left(-\frac{\boldsymbol{\Theta}}{2}\right)\mathbf{J}\boldsymbol{\omega}_{k+\frac{1}{2}},$$

whose Jacobian matrix is easily obtained from

$$\begin{aligned} \mathrm{D}f(\boldsymbol{\Theta})\cdot\delta &= -\delta - \frac{h}{4}\mathbf{J}^{-1}\left(\mathrm{d}\exp_{-\frac{\boldsymbol{\Theta}}{2}}\cdot\delta\right)\exp\left(-\frac{\boldsymbol{\Theta}}{2}\right)\mathbf{J}\boldsymbol{\omega} \\ &= -\delta + \frac{h}{4}\mathbf{J}^{-1}\left[\exp\left(-\frac{\boldsymbol{\Theta}}{2}\right)\mathbf{J}\boldsymbol{\omega}\right]^{\wedge}\mathrm{d}\exp_{-\frac{\boldsymbol{\Theta}}{2}}\cdot\delta. \end{aligned}$$

### 4.1.4 Partitioned Runge-Kutta Munthe-Kaas method (PRK)

In [NJ07], Nukala and Shelton consider a partitioned Runge-Kutta Munthe-Kaas method for the integration of rigid body dynamics. Runge-Kutta Munthe-Kaas methods are a generalization of Runge-Kutta methods for differential equations evolving on a Lie group. They were introduced by Munthe-Kaas in [MK98, MK99]. The algorithm presented in [NJ07] is *partitioned* [HLW06, Chapter 2] as the rigid body dynamics (2.14) is seen as the partitioned system of equations

$$\begin{cases} \dot{\mathbf{R}} & = f(\mathbf{R},\boldsymbol{\omega}) := \mathbf{R}\,\hat{\boldsymbol{\omega}}, \\ \mathbf{J}\,\dot{\boldsymbol{\omega}} & = g(\mathbf{R},\boldsymbol{\omega}) := \mathbf{J}\,\boldsymbol{\omega}\times\boldsymbol{\omega} + \boldsymbol{\tau}(\mathbf{R}). \end{cases}$$

The idea of a partitioned Runge-Kutta method consists of choosing two different Runge-Kutta methods and integrating the first variable (in our case, **R**) with the first method and the second variable ($\boldsymbol{\omega}$) with the second method. In [NJ07], two Runge-Kutta Munthe-Kaas methods are chosen, corresponding to a 2-stage Lobatto IIIA and a 2-stage Lobatto IIIB. The Butcher tableaus of these two methods are, respectively, given by

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array} \qquad \begin{array}{c|cc} 0 & 1/2 & 0 \\ 1 & 1/2 & 0 \\ \hline & 1/2 & 1/2 \end{array}\ . \tag{4.7}$$

In the free rigid body case, this algorithm exactly preserve the momentum and is almost Poisson [NJ07]. The tests were conducted on the free rigid body, heavy top, and fast and slow Lagrangian tops.

Using the notation introduced in Chapter 2, the partitioned Runge-Kutta Munthe-Kaas method is written as

$$\begin{cases} \boldsymbol{\omega}_{k+\frac{1}{2}} = \boldsymbol{\omega}_k + \dfrac{h}{2}\mathbf{J}^{-1}\left(\boldsymbol{\tau}(\mathbf{R}_k) - \boldsymbol{\omega}_{k+\frac{1}{2}}\times\mathbf{J}\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right), & \text{(4.8a)} \\ \mathbf{R}_{k+1} = \mathbf{R}_k\ \mathrm{cay}\left(h\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right), & \text{(4.8b)} \\ \boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_{k+\frac{1}{2}} + \dfrac{h}{2}\mathbf{J}^{-1}\left(\boldsymbol{\tau}(\mathbf{R}_{k+1}) - \boldsymbol{\omega}_{k+\frac{1}{2}}\times\mathbf{J}\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right). & \text{(4.8c)} \end{cases}$$

This algorithm is explicit in the torque evaluation. The only implicitness is in computing the angular velocity at the midstep $k+1/2$ in (4.8a); this can be solved with the same Newton procedure used for ELN. As a remark, we recall that the corresponding partitioned Runge-Kutta method on vector spaces is symplectic [Jay96] also, we note that the proposed partitioned RKMK method does not follow the ideas presented in [Eng03].

### 4.1.5 Modified Crouch-Grossman method (MCG)

A splitting approach to the integration of (2.14) is described by Nukala and Shelton in [NJ07]. The rigid body dynamics is splitted into

$$\begin{cases} \dot{\mathbf{R}} &= \mathbf{R}\,\hat{\boldsymbol{\omega}} \\ \mathbf{J}\dot{\boldsymbol{\omega}} &= 0 \end{cases}, \quad \begin{cases} \dot{\mathbf{R}} &= 0 \\ \mathbf{J}\dot{\boldsymbol{\omega}} &= (\mathbf{J}\,\boldsymbol{\omega}\times\boldsymbol{\omega}) \end{cases}, \quad \begin{cases} \dot{\mathbf{R}} &= 0 \\ \mathbf{J}\dot{\boldsymbol{\omega}} &= \boldsymbol{\tau}(\mathbf{R}) \end{cases}.$$

The flow of the first and the third vector fields can be computed exactly, while for the second one the authors propose an implicit scheme they refer to as implicit Crouch-Grossman method [CG93, OM99]. The idea is simply to approximate the solution of the quasi-linear system $\dot{Y} = A(Y)Y$ as $Y(h) = \exp(hA(Y(h)))Y(0)$ where, in our case, $Y = \mathbf{J}\boldsymbol{\omega}$ and $A(Y) = -\hat{\boldsymbol{\omega}}$. The three flow maps are then composed (according to splitting principles) with their adjoint methods in order to obtain a second-order symmetric reversible method. In [NJ07], it is also proven that this method exactly preserves the spatial angular momentum for the rigid body and it is almost Poisson.

Given $(\mathbf{R}_k, \boldsymbol{\omega}_k)$, the algorithm is given by

$$\begin{cases} \boldsymbol{\omega}_{k+\frac{1}{2}} = \mathbf{J}^{-1}\left[\exp\left(-\frac{h}{2}\boldsymbol{\omega}_{k+\frac{1}{2}}\right)\left(\mathbf{J}\,\boldsymbol{\omega}_k + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_k)\right)\right], & (4.9a) \\ \mathbf{R}_{k+1} = \mathbf{R}_k \exp\left(h\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right), & (4.9b) \\ \boldsymbol{\omega}_{k+1} = \mathbf{J}^{-1}\left[\exp\left(-h\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right)\left(\mathbf{J}\,\boldsymbol{\omega}_{k+\frac{1}{2}} + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_k)\right) + \frac{h}{2}\boldsymbol{\tau}(\mathbf{R}_{k+1})\right]. & (4.9c) \end{cases}$$

This integrator is semi-explicit, since it is implicit only in the computation of angular velocity (4.9a). Note the similarity between (4.9a) and (4.6a); the same Newton step is implemented for this equation

### 4.1.6 Koziara-Bićanić algorithm (NEW3)

Koziara and Bićanić [KB10] propose a computationally simple explicit method suited for short-term simulations of constrained rigid-body type systems. The method is then modified obtaining a semi-explicit version which shows a significative improvement in the long-time behavior at the cost of a moderate increase in the computational complexity.

The algorithm is described by the following formulas:

$$\begin{cases} \mathbf{R}_{k+\frac{1}{2}} = \mathbf{R}_k \exp\left(\frac{h}{2}\boldsymbol{\omega}_k\right), & (4.10a) \\ \exp\left(\frac{h}{2}\boldsymbol{\omega}_{k+1}\right)\mathbf{J}\,\boldsymbol{\omega}_{k+1} = \exp\left(-\frac{h}{2}\boldsymbol{\omega}_k\right)\mathbf{J}\,\boldsymbol{\omega}_k + h\boldsymbol{\tau}(\mathbf{R}_{k+\frac{1}{2}}), & (4.10b) \\ \mathbf{R}_{k+1} = \mathbf{R}_{k+\frac{1}{2}} \exp\left(\frac{h}{2}\boldsymbol{\omega}_{k+1}\right). & (4.10c) \end{cases}$$

Given $(\mathbf{R}_k, \boldsymbol{\omega}_k)$, in (4.10a) the middle-step rotation $\mathbf{R}_{k+1/2}$ is computed using a forward Lie-Euler method. Then, in (4.10b), the angular velocity $\boldsymbol{\omega}_{k+1}$ is computed using a midpoint approximation of the momentum equation (2.14b). Finally, in (4.10c), a backward Lie-Euler method is used to compute $\mathbf{R}_{k+1}$ from $\boldsymbol{\omega}_{k+1}$. This algorithm is explicit in torque evaluation, and it is implicit only in the second step (4.10b), where the body angular velocity $\boldsymbol{\omega}_{k+1}$ is computed. The Newton method requires considering

$$f(\boldsymbol{\omega}) = -\exp\left(\frac{h}{2}\boldsymbol{\omega}\right)\mathbf{J}\,\boldsymbol{\omega} + \exp\left(-\frac{h}{2}\boldsymbol{\omega}_k\right)\mathbf{J}\,\boldsymbol{\omega}_k + h\boldsymbol{\tau}(\mathbf{R}_{k+\frac{1}{2}}),$$

whose Jacobian matrix can be inferred from

$$\begin{aligned} \mathrm{D}f(\boldsymbol{\omega})\cdot\delta &= -\frac{h}{2}\left(\mathrm{d}\exp_{\frac{h}{2}\boldsymbol{\omega}}\cdot\delta\right)^{\wedge}\exp\left(\frac{h}{2}\boldsymbol{\omega}\right)\mathbf{J}\boldsymbol{\omega} - \exp\left(\frac{h}{2}\boldsymbol{\omega}\right)\mathbf{J}\delta \\ &= \frac{h}{2}\left[\exp\left(\frac{h}{2}\boldsymbol{\omega}\right)\mathbf{J}\boldsymbol{\omega}\right]^{\wedge}\mathrm{d}\exp_{\frac{h}{2}\boldsymbol{\omega}}\cdot\delta - \exp\left(\frac{h}{2}\boldsymbol{\omega}\right)\mathbf{J}\delta. \end{aligned}$$

In the numerical tests reported in [KB10], the algorithm performs well showing no energy drift. The authors therefore suggest that, due to its implementation simplicity, it might be interesting studying its stability properties.

### 4.1.7 Variational Lie-Verlet method (VLV)

The variational Lie-Verlet integrator was proposed in [BRM09] and it is based on the theory of discrete and continuous Euler-Poincaré systems [MPS99, MS93]. The method is closely related to, but different, from the RATTLE method for constrained mechanical systems [HLW06]. Due to its variational nature, the method is symplectic by construction [BRM09]. In the present context, this scheme is used to confirm that no energy drift can be observed using a symplectic integrator when integrating the rigid body dynamics with the static potential introduced in Chapter 3.

Given $(\mathbf{R}_k, \boldsymbol{\omega}_k)$ and timestep $h$, the Lie-Verlet algorithm determines $(\mathbf{R}_{k+1}, \boldsymbol{\omega}_{k+1})$ by the following iteration rule:

$$\begin{cases} \boldsymbol{\omega}_{k+\frac{1}{2}} = \boldsymbol{\omega}_k + \frac{h}{2}\mathbf{J}^{-1}\left[\mathbf{J}\,\boldsymbol{\omega}_{k+\frac{1}{2}}\times\boldsymbol{\omega}_{k+\frac{1}{2}} - \frac{h}{2}\left(\boldsymbol{\omega}_{k+\frac{1}{2}}^T\mathbf{J}\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right)\boldsymbol{\omega}_{k+\frac{1}{2}} + \boldsymbol{\tau}(\mathbf{R}_k)\right], & (4.11a) \\ \mathbf{R}_{k+1} = \mathbf{R}_k\ \mathrm{cay}\left(h\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right), & (4.11b) \\ \boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_{k+\frac{1}{2}} + \frac{h}{2}\mathbf{J}^{-1}\left[\mathbf{J}\,\boldsymbol{\omega}_{k+\frac{1}{2}}\times\boldsymbol{\omega}_{k+\frac{1}{2}} + \frac{h}{2}\left(\boldsymbol{\omega}_{k+\frac{1}{2}}^T\mathbf{J}\,\boldsymbol{\omega}_{k+\frac{1}{2}}\right)\boldsymbol{\omega}_{k+\frac{1}{2}} + \boldsymbol{\tau}(\mathbf{R}_{k+1})\right]. & (4.11c) \end{cases}$$

This algorithm is semi-explicit: the updates (4.11b) and (4.11c) are explicit while (4.11a) is implicit only in the body velocity. Using the Newton method, one needs to define the function $f : \mathfrak{so}(3) \cong \mathbb{R}^3 \to \mathfrak{so}(3) \cong \mathbb{R}^3$

$$f(\boldsymbol{\omega}) = -\boldsymbol{\omega} + \boldsymbol{\omega}_{k+\frac{1}{2}} + \frac{h}{2}\left[(\mathbf{J}\boldsymbol{\omega})\times\boldsymbol{\omega} - \frac{h}{2}(\boldsymbol{\omega}^T\mathbf{J}\boldsymbol{\omega})\boldsymbol{\omega} + \boldsymbol{\tau}(\mathbf{R})\right],$$

whose Jacobian matrix $J_f$ is given by

$$J_f(\boldsymbol{\omega}) = -\mathbf{I} + \frac{h}{2}\mathbf{J}^{-1}\left[(\mathbf{J}\boldsymbol{\omega})^{\wedge} - \hat{\boldsymbol{\omega}}\mathbf{J} - h\xi\boldsymbol{\omega}^T\mathbf{J} - \frac{h}{2}(\boldsymbol{\omega}^T\mathbf{J}\boldsymbol{\omega})\mathbf{I}\right].$$

## 4.2 Numerical simulations

In this Section, we present the numerical results obtained using the methods described in Section 4.1.

The following set of parameters has been chosen after a series of preliminary tuning experiments. The inertia matrix is equal to

$$\mathbf{J} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}.$$

Note that each principal moment of inertia does not exceed the sum of the other two, as it follows from the definition of inertia tensor. The initial attitude $\mathbf{R}_0 \in \mathrm{SO}(3)$ has been selected so that $m(\mathbf{R}_0, \mathbf{I})$ is nearly one while the initial velocity $\boldsymbol{\omega}_0 \in \mathfrak{so}(3)$ has been chosen relatively small. As described in Section 3.1, this choice is motivated by the willing of maintain the system trajectories in a neighborhood of the set $S \times \{\mathbf{0}\} \subset \mathrm{SO}(3) \times \mathfrak{so}(3)$, when the tuning parameter $\alpha$ is zero. Specifically, the initial condition $(\mathbf{R}_0, \boldsymbol{\omega}_0)$ is

$$\mathbf{R}_0 = \exp(v_0),\ v_0 = \begin{bmatrix} 0 \\ 0.7227 \\ 0 \end{bmatrix},\ \boldsymbol{\omega}_0 = \begin{bmatrix} 0 \\ 0 \\ 0.625 \end{bmatrix}.$$

The tuning parameter $\alpha$ and the attraction point $\mathbf{R}_m$ has been set to provide a weak perturbation of the original trajectory. The chosen values are $\alpha = 0.3$ and

$$\mathbf{R}_m = \exp(v_m),\ v_m = \frac{1}{\sqrt{2}}\begin{bmatrix} 2.5 \\ 0 \\ 2.5 \end{bmatrix}.$$

The simulation has been carried on a long-time interval $[0, 10000]$, with timesteps $h = 0.125$ and $h = 0.25$; results are shown in Figure 4.1. A systematic drift for the all the methods but the VLV can be observed. Comparing the figures of the left side with those on the right side, we see that the drift is linear with respect to time. A finer analysis also shows that the drift is quadratic in the timestep $h$. We abbreviate this fact by saying that the total energy error behaves like $\mathcal{O}(Th^2)$. Positive (MCG, NEW3, LIEMID[EA]) and negative energy drifts (ELN, TLN, PRK) of different slopes are observed. Further numerical tests, not reported, conducted using the exponential map in place of the Cayley map in the reconstruction equation of ELN and TLN also show a very similar drift in the energy. Finally, it is of interest to note that LIEMID[EA] exhibits the smallest slope constant (in absolute value).

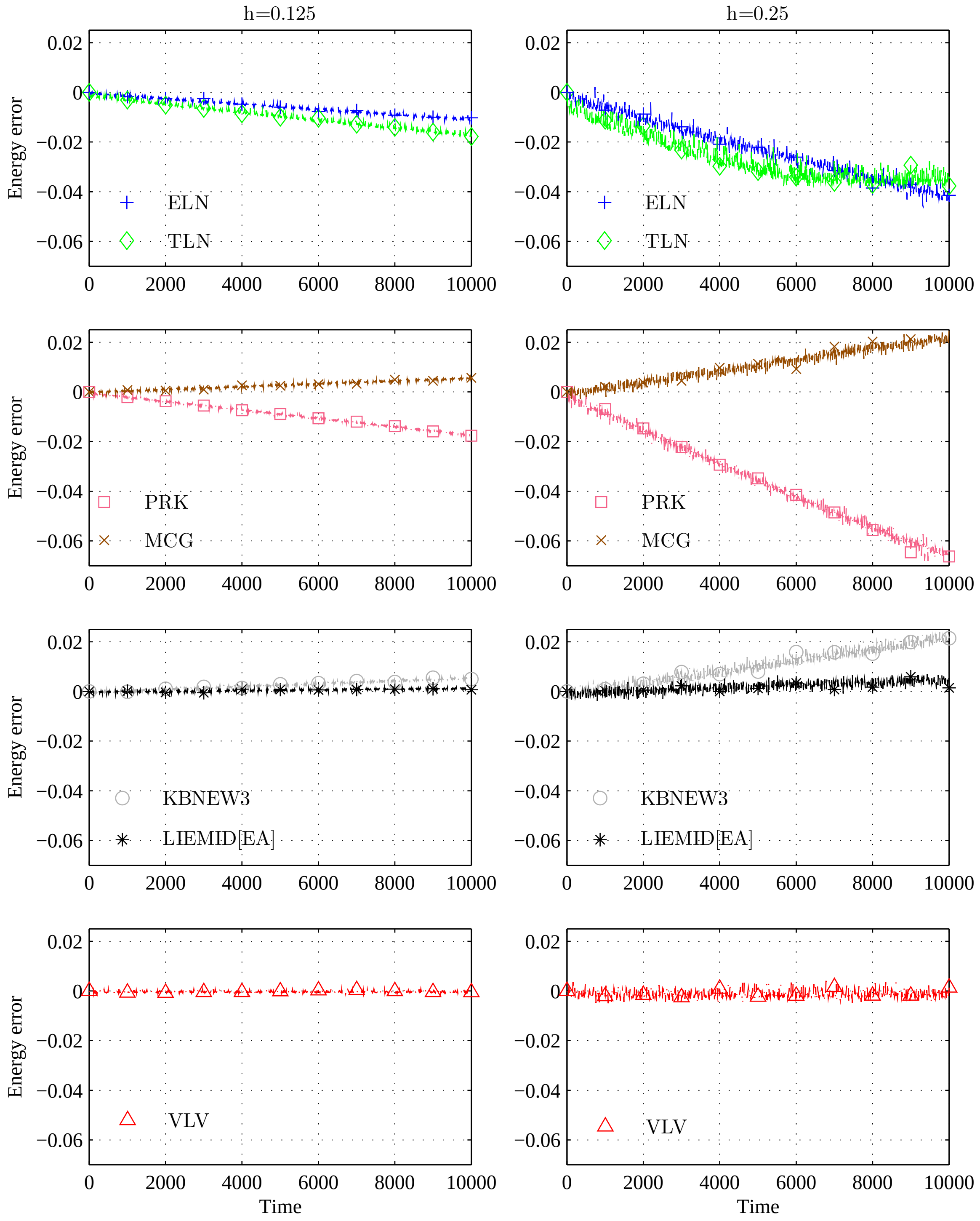


Figure 4.1: Energy error of the ELN, TLN, PRK, MCG, NEW3, LIEMID[EA], VLV algorithms. Plots on the left are obtained with a timestep $h = 0.125$, plots on the right with a doubled timestep $h = 0.25$. All the algorithms but VLV exhibit a systematic energy drift. On the other hand, the energy error of VLV method remains bounded as predicted by theory. The initial conditions and parameters used are provided in the text.

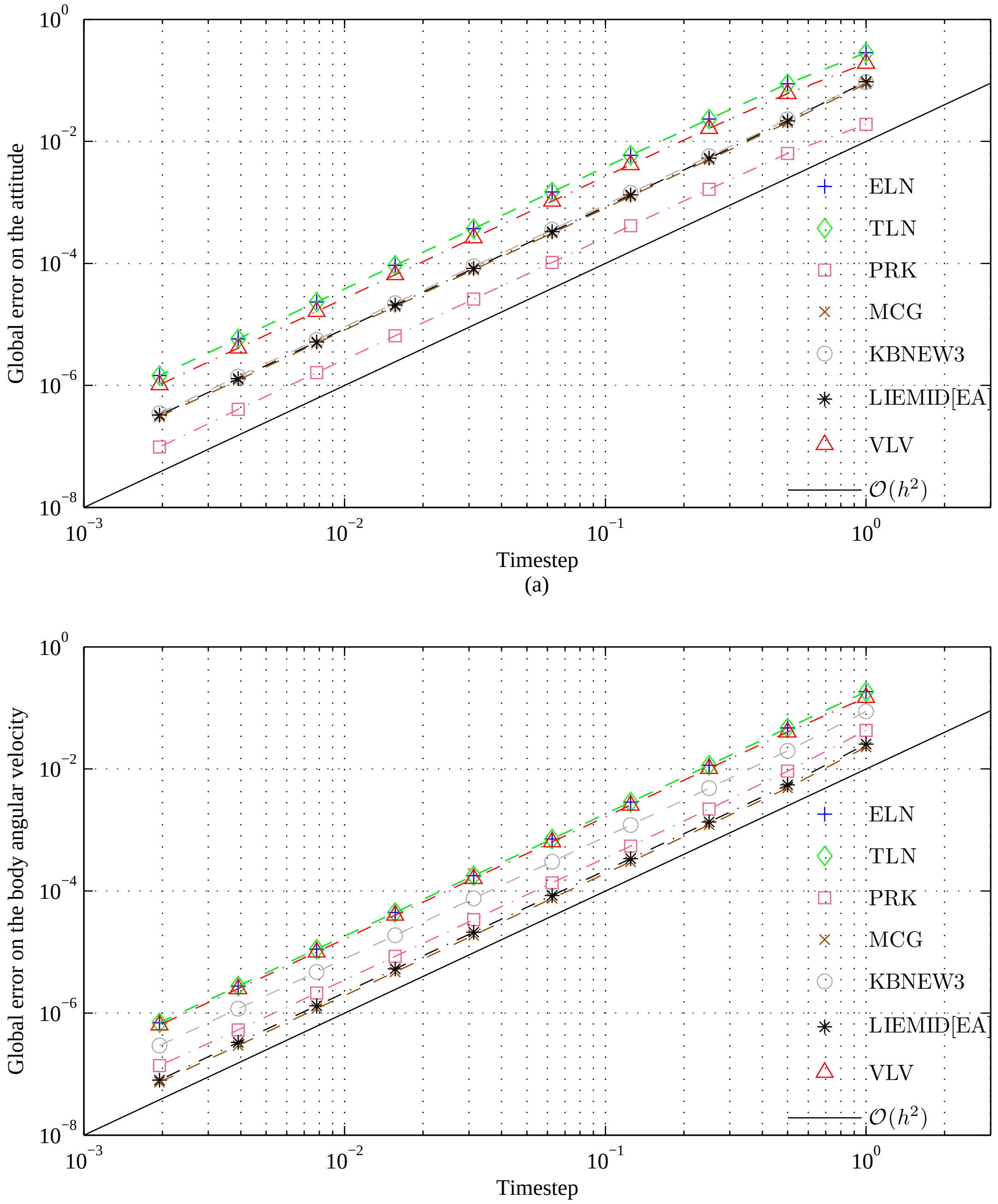
The time-precision diagrams, shown in Figure 4.2, confirm that all the methods are second-order accurate. Figure 4.2(a) shows the global error in the attitude matrix at time $T = 5$ for different timesteps. Figure 4.2(b) shows, similarly, the global error in body angular velocity. The reference solution was computed using the function `ode45` [DP80] in Matlab, with an absolute tolerance $10^{-14}$ and relative tolerance $2{\cdot}10^{-14}$. We recall that `ode45` function is based on an explicit Runge-Kutta formula, the Dormand-Prince pair; it dynamically computes the integration timestep in order to achieve the desired accuracy on the solution. It is interesting that, despite its implicitness, TLN shows the same accuracy than ELN in this numerical test.

An interesting remark is in order at this point. Since the methods we have tested are symmetric and the test defines a reversible Hamiltonian system, this numerical experiment is also an interesting counterexample in the context of Lie groups to the belief that symmetric methods applied to reversible Hamiltonian systems nearly preserve the energy over long times.

Besides accuracy, run-time cost is another important measure to have a global picture of the performance of a numerical method. In Figure 4.3 some details are provided about the required running time of the methods for this numerical test. All the algorithms have been implemented in Matlab using a standard Newton method to solve for the implicit steps, as detailed in Section 4.1. The code has been executed on a P8600 Dual Core x64 pc (2.40 GHz), 4 GB RAM, running Windows 7 Professional and Matlab R2009b. Figure 4.3(a) displays the absolute CPU time for each method, while in Figure 4.3(b) the relationship between the running times is highlighted; since ELN turns out to be the fastest method, the plotted ratio is

$$\frac{\text{running time of the method}}{\text{ELN running time}}.$$

As we were expecting, implicit algorithms are slower than semi-explicit ones; however, TLN requires up to twelve (!) times the running time of the remaining algorithms, despite its accuracy is even worse than other methods'.



(a)

(b)

Figure 4.2: Global error of the ELN, TLN, PRK, MCG, NEW3, LIEMID[EA], VLV algorithms. The global error is evaluated in (a) configuration and (b) body angular velocity at a physical time of $T = 5$ for timesteps $h = \{1, 2^{-1}, \dots, 2^{-9}\}$. We use as reference solution the integration of (2.14) using the MATLAB function ode45 with low tolerance. Observe that all the integrators are second-order accurate, and PRK, MCG and LIEMID[EA] show a better accuracy than other methods.

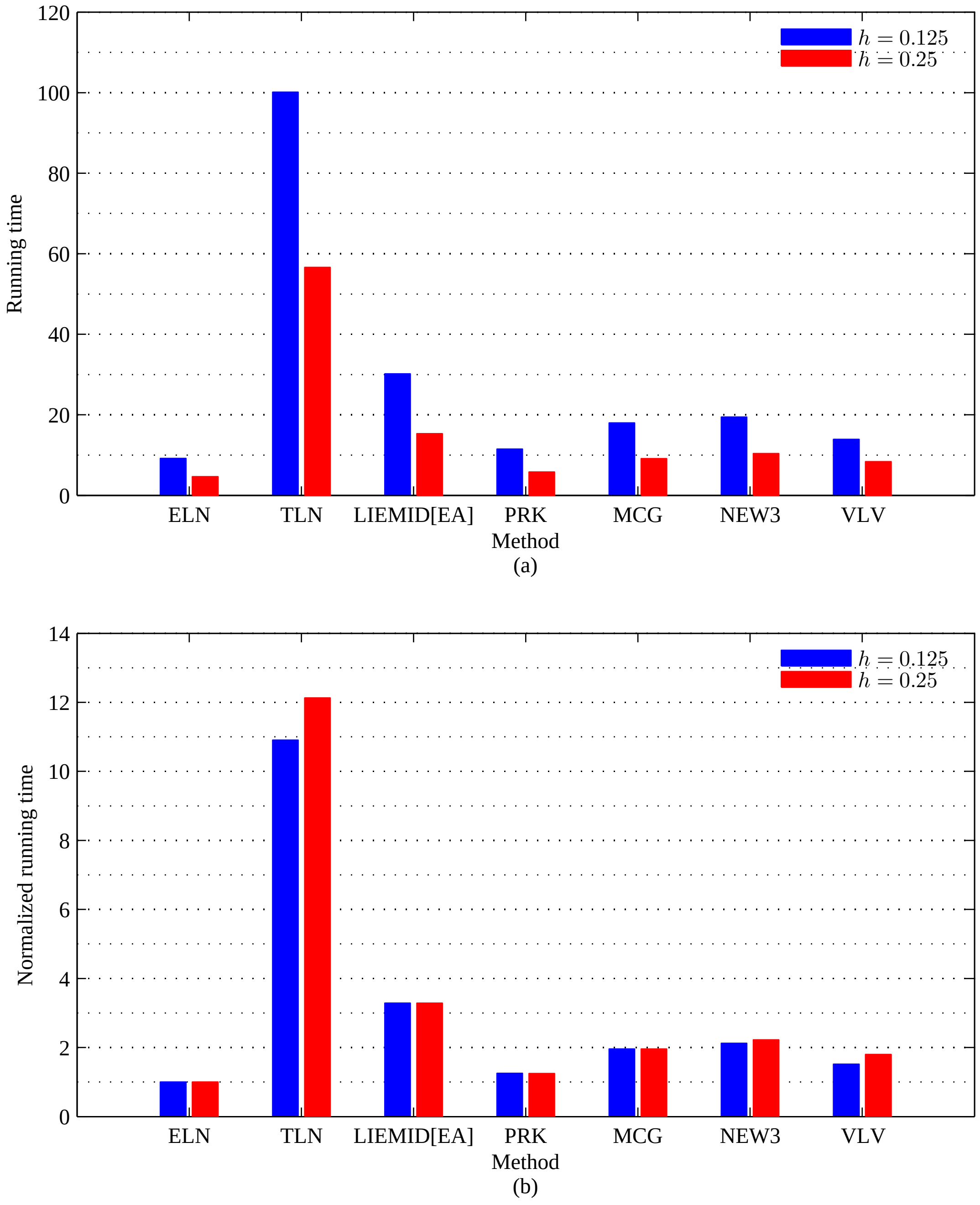


Figure 4.3: Running time of the methods. Absolute CPU time is showed in (a) while the ratio between running time and fastest running time (ELN method) is reported in (b). Note the huge amount of computation time required by TLN, despite its low accuracy. Initial conditions and parameters used are provided in the text.

## 4.3 Conclusions and future works

In this Thesis, we propose an easy-to-implement numerical experiment that has proven effective in detecting the possible energy drift of a conservative rigid body integrator. The test consists in observing the evolution of the dynamics of a rigid body in an *ad hoc* static configuration-dependent potential field, obtained perturbing a stable equilibrium set through the introduction of a potential that defines a strong attractive point. We tested various rigid body methods taken from the literature. All these schemes show an energy drift, thus disproving the conjecture on their symplecticity or conjugate symplecticity. This test remarks the importance in long-time integration of symplecticity for a numerical method, as the simulations of the VLV method confirm. Further theoretical investigations are required to understand why this test is effective in highlighting a drift and to explore the possibility of extending the test on a generic Lie group other than SO(3).

# Part II

# Geometric integration on two-spheres

# 5

# Integration on homogeneous manifolds

A crucial challenge in numerical integration is the preservation of the configuration space for differential equations evolving on manifolds. For years, this aim has been achieved using an *extrinsic* or *embedded* approach, that is, the manifold is embedded in a vector space and the flow is reprojected on the manifold, or position constraints are enforced. Alternatively, one can reformulate the dynamics using a set of coordinates which do not require any constraints. Nevertheless, these solutions are not really effective. Reprojection artificially injects or dissipates energy into the system, and enforcing constraints can be computationally expensive, even for very simple cases. Again, writing the system dynamics in a non-natural set of coordinates can result in a very difficult operation involving complicated trigonometric expression and additional complexity in the computations.

On the other hand, the geometric or *intrinsic* approach is based on natural movement on the manifold: the discrete iteration rule assures that the flow naturally remains on the configuration space. A large amount of literature is devoted to geometric integration on Lie groups, especially on SO(3); integration on homogeneous manifolds represents therefore a natural setting and extension for geometric integration on Lie groups.

This Chapter stands as a brief introduction to the problem of numerical integration on homogeneous spaces. In Section 5.1 some useful mathematical definitions are recalled, while a survey of the existing literature is presented in Section 5.2.

## 5.1 Mathematical preliminaries

In this Section the reader will find the mathematical concepts needed in order to comprehend the following Chapters. Basic reference for subsection 5.1.1 is [MR99], while definitions and results of subsection 5.1.2 are taken from [Hun74]; concepts presented in subsection 5.1.3 are described in high detail in [KN63], even if we advise the shorter and simplified presentation of [Leo04].

### 5.1.1 Homogeneous spaces

Let $M$ be a manifold and $G$ be a Lie group. A *(left) action* of $G$ on $M$ is a differentiable mapping $L_g : M \to M$ such that

i. $L_e(m) = m$ if $e \in G$ is the identity element;

ii. $(L_g \circ L_h)(m) = L_{gh}m,\ \forall\, g, h \in G.$

The set of all elements of $M$ to which $m$ can be moved to is the *orbit* of $m$. The *isotropy group* for $m$ is defined as

$$G_m = \{g \in G | L_g m = m\} \subseteq G.$$

It is straightforward to see that $G_m$ is a group. The *isotropy subalgebra* $\mathfrak{g}_m$ is the Lie algebra of the isotropy group $G_m$.

The *orbit* of a point $m \in M$ is the set of elements in $M$ to which $m$ can be moved via the action by the elements of $G$. The set of orbits forms a partition of $M$; the set of all orbits of $M$ under the action of $G$ is denoted as $M/G$ and is called the *quotient* of the action, or *orbit space*.

Be $L : G \times M \to M$ is an action, and be $\xi \in \mathfrak{g}$. Then $L_{\exp(t\xi)} : M \to M$ defines a flow on $M$. The corresponding vector field on $M$, given by

$$\xi_M(m) \overset{\text{def}}{=} \frac{d}{dt}\bigg|_{t=0} L_{\exp(t\xi)}(m)$$

is called the *infinitesimal generator* of the action corresponding to $\xi$.

If for every $m_1, m_2 \in M$ there exists a $g \in G$ such that $m_2 = L_g m_1$, the action is called *transitive*; in this case all the orbits are equal and coincide with the manifold. If for each $g \in G$ there exists at least an element $m$ such that $L_g m \neq m$, then the action is *effective* (or *faithful*). If instead $L_g m \neq m$ for all $g \in G,\ g \neq e,\ m \in M$, the Lie group is said to act *freely* on the manifold. With a free action all isotropy groups are equal and contains the only identity element. It can be easily seen that freeness is a stronger property than effectiveness (freeness $\Rightarrow$ effectiveness, but not vice versa), since in the latter case there might be $g \in G$, $m \in M$ such that $L_g m = m$, but not in the former case.

A *homogeneous space* is a manifold with a transitive group action by a Lie group.

### 5.1.2 Review of homological algebra

Be $f : A \to B$ a homomorphism. The *kernel* of $f$ (Ker$f$) is the subset of elements of $A$ which are mapped in the neutre element of $B$. A finite sequence of homomorphisms $f_i$

$$A_0 \xrightarrow{f_1} A_1 \xrightarrow{f_2} A_2 \xrightarrow{f_3} \dots \xrightarrow{f_{n-1}} A_{n-1} \xrightarrow{f_n} A_n$$

is called *exact* if $\operatorname{Im} f_i = \operatorname{Ker} f_{i+1}$, for $i = 1, \dots, n-1$. A sequence $0 \xrightarrow{\quad} A \xrightarrow{f} B$, (where the first homomorphism maps the zero element to the neutre element of $A$) is exact if and only if $f$ is a one-to-one mapping. A sequence $B \xrightarrow{g} C \longrightarrow 0$ is exact if and only if $g$ is onto. An exact sequence of the form

$$\begin{tikzcd}
0 \arrow[r] & A \arrow[r, "f"] & B \arrow[r, "g"] & C \arrow[r] & 0
\end{tikzcd}$$

is called a *short exact sequence*; note that $f$ is one-to-one, $g$ is onto.

**Lemma 5.1.1.** *Consider a commutative diagram,*

$$\begin{tikzcd}
0 \arrow[r] & A \arrow[r, "f"] \arrow[d, "\alpha"] & B \arrow[r, "g"] \arrow[d, "\beta"] & C \arrow[r] \arrow[d, "\gamma"] & 0 \\
0 \arrow[r] & A' \arrow[r, "f'"] & B' \arrow[r, "g'"] & C' \arrow[r] & 0,
\end{tikzcd}$$

*where each row is a short exact sequence. Then,*

- *$\alpha, \gamma$ are one-to-one $\Rightarrow$ $\beta$ is one-to-one;*
- *$\alpha, \gamma$ are onto $\Rightarrow$ $\beta$ is onto;*
- *$\alpha, \gamma$ are isomorphism $\Rightarrow$ $\beta$ is an isomorphism and the two sequences are isomorphic.*

*Proof.* See the proof of [Hun74, Lemma 1.17]. □

**Theorem 5.1.2.** *Given a short exact sequence*

$$\begin{tikzcd}
0 \arrow[r] & A_1 \arrow[r, "f"] & B \arrow[r, "g"] & A_2 \arrow[r] & 0,
\end{tikzcd}$$

*the following conditions are equivalent:*

1. *there is a homomorphism $h : A_2 \to B$ with $g \circ h = 1_{A_2}$ (identity map on $A_2$);*
2. *there is a homomorphism $k : B \to A_1$ with $k \circ f = 1_{A_1}$;*
3. *these two sequence are isomorphic:*

$$\begin{tikzcd}
0 \arrow[r] & A_1 \arrow[r, "f"] \arrow[d, "1_{A_1}"] & B \arrow[r, "g"] \arrow[d, "\psi"] & A_2 \arrow[r] \arrow[d, "1_{A_2}"] & 0 \\
0 \arrow[r] & A_1 \arrow[r, "i_1"] & A_1 \oplus A_2 \arrow[r, "\pi_2"] & A_2 \arrow[r] & 0,
\end{tikzcd}$$

*where $i$ is an inclusion, $\pi$ is a projection, $\psi(b) = (k(b), g(b))$. In particular, $B \cong A_1 \oplus A_2$ and the short exact sequence is called split exact sequence.*

*Proof.* See [Hun74, Theorem 1.18]. □

### 5.1.3 Theory of bundles and connection

A *bundle* consists of a triple $(Q, S, \pi)$ where $Q$ and $S$ are topological spaces, respectively the *bundle space* and the *base space*, and $\pi : Q \to S$ is a continuous map. If there exists a group $G$ which acts freely on $Q$, then $(Q, S, \pi)$ is a *principal bundle* and it is isomorphic to $(Q, Q/G, \pi_{Q/G})$. Since the action is free each orbit is homeomorphic to $G$, and $G$ is called *fiber of the bundle*.

Consider a principal bundle with a group $G$. A *connection* is a way of decomposing the tangent space $T_qQ$ into the direct sum $T_qQ = V_qQ \oplus H_qQ$, where $V_qQ$, the *vertical subspace*, is the tangent space to the group orbit, and $H_qQ$ is called *horizontal subspace*. More specifically, a *connection 1-form* is defined as $\mathcal{A} : TQ \to \mathfrak{g}$ such that

1. $\mathcal{A}$ is equivariant: $\mathcal{A} \circ TL_g = g \cdot \mathcal{A} \cdot g^{-1}$, $g \in G$;

2. $\mathcal{A}(v_q)$ is the unique element $\xi$ of the Lie algebra which generates the vertical component of $v_q$.

A connection 1-form is in a one-to-one relationship with a connection [KN63].

The *horizontal lift* of a vector field $X$ on the orbit space $Q/G$ is the unique vector field $X^h : Q \to TQ$ which is horizontal (that is, it takes values only in the horizontal subspace) and projects into $X$, that is,

$$\pi_*(X^h(q)) = X(\pi(q)).$$

As proven in [KN63], this horizontal lift is unique and therefore can be thought as in a one-to-one relationship with the choice of a connection.

## 5.2 Literature review

Not many papers are devoted to the problem of geometric integration on homogeneous spaces. As mentioned in the introduction of this Chapter, this problem is strictly related to numerical integration on Lie group: in fact, the matrix Lie group SO(3) is a homogeneous space under the transitive and free action of itself. The pioneer work about this topic was written by Munthe-Kaas [MKZ97]. In that paper Runge-Kutta Munthe-Kaas Lie group methods [MK98, MK99] are expressed in an abstract coordinate-free manner on a variety of structures including homogeneous spaces. Moreover, it is shown that this method can be employed for the integration on unit spheres $\mathbb{S}^n = \{y \in \mathbb{R}^{n+1} | y^T y = 1\}$.

The Stiefel manifold, whose elements are represented by $n \times k$ matrices with orthonormal columns, is a generalization of the matrix Lie group of orthogonal matrix, and is an important example of homogeneous space. There are several problems that can be modeled as ODE on a Stiefel manifold, e.g., optimization problems or computation of the Lyapunov exponent of a dynamical system. Intrinsic approach for the integration on Stiefel manifolds based on retraction maps is considered in a bunch of papers by Celledoni and Owren [CO02, CO03]:

the Lie group action is exploited in order to obtain a suitable retraction map. We also remark that in those paper some important Lie group algorithms like Runge-Kutta Munthe-Kaas methods [MK98, MK99] and Crouch-Grossman methods [CG93] are adapted for the geometric integration on Stiefel manifolds.

In the paper [LO02], Lewis and Olver address the issue of numerical integration on homogeneous spaces whose points have continuous isotropy, that is, a given flow on the manifold corresponds to continuous families of flows on the acting Lie group. A particular emphasis is dedicated to integration of dynamical systems on unit sphere $\mathbb{S}^2$. The authors show that adding a proper isotropy term in the numerical iteration rule can increase the order of the method, and also improve the accuracy in the preservation of integrals or conservation laws of the original flow. In [LN03] some more insight is gained into the choice of the isotropy generator, using a generalization of the connection form on a principal bundle; moreover, the introduced method is applied to micromagnetic systems.

The dynamics of separable Hamiltonian systems whose configuration space is $\mathbb{S}^2$ or a product of $\mathbb{S}^2$ is considered in [LLM09]. Lee, Leok and McClamroch use the variational principle to write global expression of Euler-Lagrange equations on two-spheres, both in continuous-time and discrete-time settings; this description completely avoids singularities and numerical ill-conditionings. In the paper a second-order variational method is also provided and tested on several examples like the double spherical pendulum, interconnection of spherical pendulums, an array of magnetic dipoles and molecular dynamics on a sphere.

# 6

# Lie methods on two-spheres

The two-sphere $\mathbb{S}^2$ is defined as the set of all points in $\mathbb{R}^3$ which have a unit length from the origin. Many classical and important dynamical systems evolve on two-sphere or on a product of two-spheres. In these cases, the configuration is usually described using 2 angles or a constraint enforcing unit length (on $\left(\mathbb{S}^2\right)^n$, $2n$ angles or $n$ constraints); these representations should be avoided however, since they yield additional complexity in the computation.

The geometric approach to this problem exploits the fact that the special orthogonal Lie group of rotation matrices SO(3) acts transitively on the two-sphere, that is, $\mathbb{S}^2$ is a homogeneous space; by updating the trajectory using a rotation, it is assured that the discrete trajectory belongs to the configuration space at any time without enforcing any constraints.

In Section 6.1 we will describe our approach how to solve the problem of integration of the dynamics of a system whose configuration space is the two-sphere (or the product of two-spheres) using Lie group methods; the generalization to the geometric approach in terms of connection forms on a principal bundle is described in Section 6.2.

## 6.1 Dynamics on two-spheres

The *unit sphere* $\mathbb{S}^2$ is the set of the point in $\mathbb{R}^3$ which have a unit length from the origin,

$$\mathbb{S}^2 = \{\mathbf{q} \in \mathbb{R}^3 | \|\mathbf{q}\| = 1\}. \tag{6.1}$$

Since the tangent space $T_{\mathbf{q}}\mathbb{S}^2$ of $\mathbf{q} \in \mathbb{S}^2$ is the plane tangent to the sphere in $\mathbf{q}$ (see Figure 6.1), the time derivative $\dot{\mathbf{q}}(t) \subset \mathbb{R}^3$ of a curve $\mathbf{q}(t) \subset \mathbb{S}^2$ is pointwise orthogonal to the curve, that is, $\dot{\mathbf{q}}(t) \cdot \mathbf{q}(t) = 0,\ \forall\, t$. Adopting a “left trivialization”, $\dot{\mathbf{q}}$ can be always written as

$$\dot{\mathbf{q}} = \boldsymbol{\omega} \times \mathbf{q}; \tag{6.2}$$

where the *(body) angular velocity* $\boldsymbol{\omega} \in \mathbb{R}^3$ satisfies

$$\mathbf{q} \cdot \boldsymbol{\omega} = 0. \tag{6.3}$$

The tangent bundle $T\mathbb{S}^2$ can be then represented as

$$T\mathbb{S}^2 = \{(\mathbf{q}, \boldsymbol{\omega}) \in \mathbb{R}^3 \times \mathbb{R}^3 | \|\mathbf{q}\| = 1,\ \mathbf{q} \cdot \boldsymbol{\omega} = 0\};$$

this representation can be extended in a straightforward manner for $T(\mathbb{S}^2)^n$.

Note that $\boldsymbol{\omega}$ is uniquely defined if and only if (6.3) holds; otherwise, for a fixed $\mathbf{q}$, there would exist a continuous family of body angular velocities which generates the same dynamics on $\mathbf{q}$, specifically $\boldsymbol{\omega} + \alpha\mathbf{q}$. Differentiating (6.3) with respect to time, we find that

$$\begin{aligned} 0 = \frac{d}{dt}\mathbf{q} \cdot \boldsymbol{\omega} &= \dot{\boldsymbol{\omega}} \cdot \mathbf{q} + \boldsymbol{\omega} \cdot \dot{\mathbf{q}} \\ &= \dot{\boldsymbol{\omega}} \cdot \mathbf{q} + \underbrace{\boldsymbol{\omega} \cdot (\boldsymbol{\omega} \times \mathbf{q})}_{=0} \\ &= \dot{\boldsymbol{\omega}} \cdot \mathbf{q}, \end{aligned} \tag{6.4}$$

that is, the angular acceleration is orthogonal to the position.

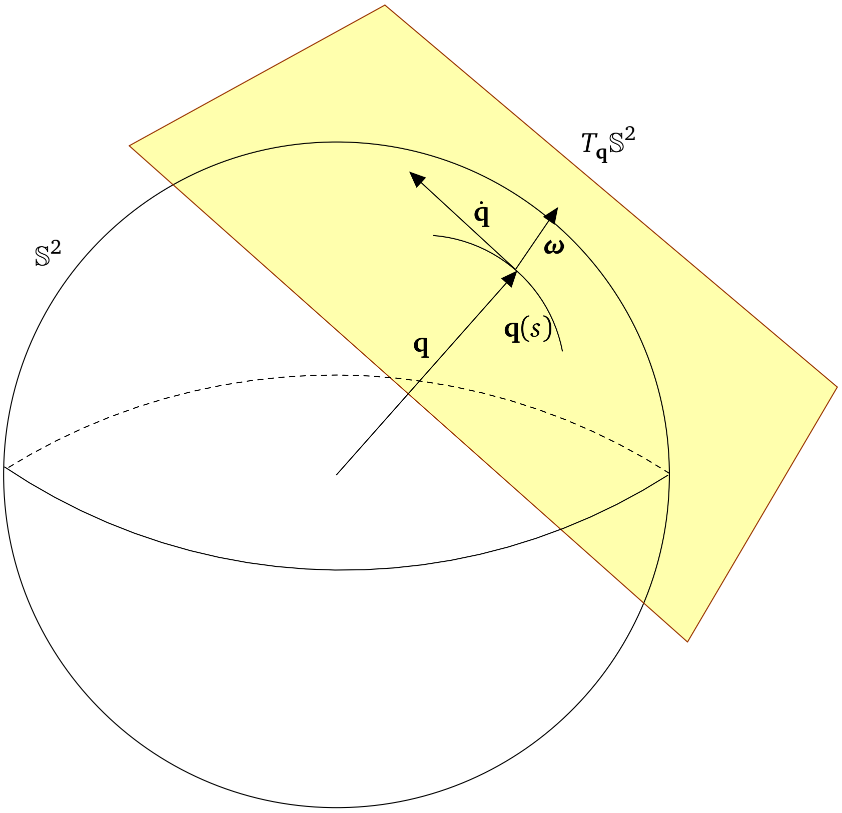


Figure 6.1: The unit sphere $\mathbb{S}^2$ with the tangent space $T_\mathbf{q}\mathbb{S}^2$, that is, the plane tangent to the sphere in $\mathbf{q}$. In every point of a curve $\mathbf{q}(s) \subset \mathbb{S}^2$, the linear velocity $\dot{\mathbf{q}}$ satisfies $\dot{\mathbf{q}} \cdot \mathbf{q} = 0$, therefore we can write $\dot{\mathbf{q}} = \boldsymbol{\omega} \times \mathbf{q}$. If we impose $\mathbf{q} \cdot \boldsymbol{\omega} = 0$, $\boldsymbol{\omega}$ is unique and we can left-trivialize $T_\mathbf{q}\mathbb{S}^2 = \{(\mathbf{q}, \boldsymbol{\omega}) \,|\, \|\mathbf{q}\| = 1,\ \mathbf{q} \times \boldsymbol{\omega} = 0\}$.

SO(3) acts transitively but not freely on $\mathbb{S}^2$, since rotating a vector $\mathbf{q}$ along the $\mathbf{q}$-axis leaves the vector unchanged. Therefore, a nontrivial isotropy subgroup $H_{\mathbf{q}}$ corresponds to each point $\mathbf{q} \in \mathbb{S}^2$, defined as

$$H_{\mathbf{q}} = \{\mathbf{R} \in \mathrm{SO}(3) \,|\, \mathbf{R}\mathbf{q} = \mathbf{q}\}.$$

The isotropy Lie subalgebra $\mathfrak{h}_{\mathbf{q}}$ is diffeomorphic to the vector space spanned by $\mathbf{q}$; note that $H_{\mathbf{q}}$ and $\mathfrak{h}_{\mathbf{q}}$ change wih time. The effect of the isotropy in terms ofthe subalgebra can be described as follows. Be $\mathbf{q}_1$, $\mathbf{q}_2 \in \mathbb{S}^2$, and be $\xi \in \mathfrak{so}(3)$ such that $\mathbf{q}_2 = \exp(\xi)\mathbf{q}_1$; the isotropy implies that

$$\mathbf{q}_2 = \exp(\xi + \sigma)\mathbf{q}_1, \quad \forall\, \sigma \in \mathfrak{h}_{\mathbf{q}_1}.$$

We begin considering a mechanical system whose Lagrangian $L : T(\mathbb{S}^2)^n \to \mathbb{R}$ is the difference between a quadratic kinetic energy and a configuration-dependent static potential energy $V$, that is

$$L(\mathbf{q}_1, \dots, \mathbf{q}_n, \dot{\mathbf{q}}_1, \dots, \mathbf{q}_n) = \frac{1}{2} \sum_{i,j=1}^{n} M_{ij} \dot{\mathbf{q}}_i \cdot \dot{\mathbf{q}}_j - V(\mathbf{q}_1, \dots, \mathbf{q}_n),$$

where $\mathbf{M}$ is a symmetric positive definite matrix. In [LLM09], the Hamilton variational principle is employed to obtain the Euler-Lagrange equations on $T(\mathbb{S}^2)^n$: the continuous equations of motion can be written as a second-order ODE on $\mathbf{q}$ or equivalently, using the left-trivialized representation, as an ODE on $(\mathbf{q}_1, \dots, \mathbf{q}_n, \boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_n)$:

$$\begin{cases} \dot{\mathbf{q}}_i = \boldsymbol{\omega}_i \times \mathbf{q}_i, & (6.5a) \\ \dot{\boldsymbol{\omega}}_i = \dfrac{1}{M_{ii}} \displaystyle\sum_{\substack{j=1 \\ j \neq i}}^{n} \left[ M_{ij} \mathbf{q}_i \times (\mathbf{q}_j \times \dot{\boldsymbol{\omega}}_j) + M_{ij} (\boldsymbol{\omega}_j \cdot \boldsymbol{\omega}_j) \mathbf{q}_i \times \mathbf{q}_j \right] - \dfrac{1}{M_{ii}} \mathbf{q}_i \times \dfrac{\partial V}{\partial \mathbf{q}_i}, & (6.5b) \end{cases}$$

for $i = 1, \dots, n$. We remark that, given proper initial conditions, the flow

$$(\mathbf{q}_1(t), \dots, \mathbf{q}_n(t), \boldsymbol{\omega}_1(t), \dots, \boldsymbol{\omega}_n(t)) \subset T(\mathbb{S}^2)^n$$

generated by (6.5) naturally satisfies conditions (6.3) and (6.4), without enforcing any constraints.

Given a flow $\mathbf{q}(t) \subset \mathbb{S}^2$, we want to lift it to a flow $\mathbf{R}(t) \subset \mathrm{SO}(3)$ such that

$$\begin{tikzcd} \mathbf{R}(t) \subset \mathrm{SO}(3) \arrow[d, "\pi"] \\ \mathbf{q}(t) \subset \mathbb{S}^2 \end{tikzcd}$$

where $\pi$ is the projection map

$$\pi : \mathrm{SO}(3) \to \mathbb{S}^2$$
$$\pi(\mathbf{R}(t)) = \mathbf{R}(t)\mathbf{q}(0).$$

Since $(\mathrm{SO}(3))^n$ acts transitively on $(\mathbb{S}^2)^n$, for each $t$ we can find a rotation matrix such that

$$\mathbf{q}_i(t) = \mathbf{R}_i(t)\mathbf{q}_i(0), \quad i = 1, \ldots, n. \tag{6.6}$$

If we differentiate (6.6) with respect to time and compare the result with (6.5a), we obtain

$$\begin{aligned} \dot{\mathbf{q}}_i(t) &= \dot{\mathbf{R}}_i(t)\mathbf{q}_i(0) \\ &= \boldsymbol{\omega}_i(t) \times \mathbf{q}_i(t) \\ &= \boldsymbol{\omega}_i(t) \times \big(\mathbf{R}_i(t)\mathbf{q}_i(0)\big) \\ &= \hat{\boldsymbol{\omega}}_i(t)\mathbf{R}_i(t)\mathbf{q}_i(0), \end{aligned}$$

which suggest that the dynamics of the flow $\mathbf{R}_i(t) \subset \mathrm{SO}(3)$ acting on $\mathbf{q}_i(0)$ to generate $\mathbf{q}_i(t)$ can be expressed as

$$\begin{cases} \dot{\mathbf{R}}_i(t) = \hat{\boldsymbol{\Omega}}_i(t)\mathbf{R}_i(t), \\ \mathbf{q}_i(t) = \mathbf{R}(t)\mathbf{q}_i(0), \\ \dot{\boldsymbol{\Omega}}_i(t) = \dfrac{1}{M_{ii}} \displaystyle\sum_{\substack{j=1 \\ j \neq i}}^{n} \Big\{ M_{ij} \left(\mathbf{R}_i(t)\mathbf{q}_i(0)\right) \times \Big[ \Big(\mathbf{R}_j(t)\mathbf{q}_j(0)\Big) \times \dot{\boldsymbol{\Omega}}_j(t) \Big] + \\ \qquad\qquad + M_{ij}(\boldsymbol{\Omega}_j(t) \cdot \boldsymbol{\Omega}_j(t)) \left(\mathbf{R}_i(t)\mathbf{q}_i(0)\right) \times \Big(\mathbf{R}_j(t)\mathbf{q}_j(0)\Big) \Big\} + \\ \qquad - \dfrac{1}{M_{ii}} \left(\mathbf{R}_i(t)\mathbf{q}_i(0)\right) \times \dfrac{\partial V}{\partial \mathbf{q}_i}, \end{cases} \tag{6.7}$$

with $\mathbf{R}(0) = I,\ \boldsymbol{\Omega}(0) = \boldsymbol{\omega}(0)$. Equation (6.7) can be interpreted as a standard ODE on the Lie group $\mathrm{SO}(3) \times \mathfrak{so}(3)$, with a right-trivialized representation of the rotation matrices' dynamics, and can therefore be solved using a Lie group method. This approach exploits the fact that $T_{\mathbf{q}}\mathbb{S}^2$ and $\mathfrak{so}(3)$ are vector spaces $T_{\mathbf{q}}\mathbb{S}^2 \subset \mathbb{R}^3$, $\mathfrak{so}(3) \cong \mathbb{R}^3$; we thus identify the body angular velocity of the dynamics on $T\mathbb{S}^2$ with a spatial angular velocity of a right-trivialized dynamics on $\mathrm{SO}(3) \times \mathfrak{so}(3)$.

**Remark** A curve in SO(3) can be written as

$$\dot{\mathbf{R}} = \hat{\xi}\mathbf{R}, \ \xi \in \mathfrak{so}(3);$$

via a left action, this curve generates a curve in the homogeneous space

$$\dot{\mathbf{q}} = \dot{\mathbf{R}}\mathbf{q}(0) = \hat{\xi}\mathbf{R}\mathbf{q}(0) = \xi \times \mathbf{q}. \tag{6.8}$$

Due to the effect of the isotropy, the same curve in $\mathbb{S}^2$ is generated by $\xi + \sigma\mathbf{q}$, $\sigma \in \mathbb{R}$; that is, the same solution in $\mathbb{S}^2$ can be obtained from a family of curves in SO(3). In Section 6.2 we will show that, in the continuous-time, the approach (6.7) naturally avoids the effect of the isotropy.

**Remark** The case we are considering is different from the one addressed in, e.g., [LO02]. Lewis and Olver were in fact referring to a system dynamics on two-spheres expressed as

$$\dot{\mathbf{q}} = X(\mathbf{q}) \times \mathbf{q}, \tag{6.9}$$

where the curve derivative depends only on the position. The framework we are working in, on the other hand, refers to systems whose actual phase space is $T\mathbb{S}^2$, while exploiting the fact that $\mathbb{S}^2$ is a homogeneous space under the action of SO(3). As we describe in Section 7.3, we are currently investigating if our approach also works for systems' dynamics like (6.9).

## 6.2 Choice of a connection

In the following Section we will use the theory of connection forms to show the properties of the lifted flow described by (6.7).

Be $\mathbf{q} \in \mathbb{S}^2$ and consider the corresponding isotropy subgroup $H_\mathbf{q}$ and isotropy subalgebra $\mathfrak{h}_\mathbf{q}$. Since $H_\mathbf{q}$ acts freely on SO(3), $(\mathrm{SO}(3), \mathrm{SO}(3)/H_\mathbf{q}, \pi)$ is a principal bundle; $\pi$ associates to each element in SO(3) the corresponding orbit generated by the $H_\mathbf{q}$ action, and $H_\mathbf{q}$ is the fiber of the principal bundle. Consider the short exact sequence

$$\begin{tikzcd} 0 \arrow[r] & H_\mathbf{q} \arrow[r, "i"] & \mathrm{SO}(3) \arrow[r, "\pi"] & \mathrm{SO}(3)/H_\mathbf{q} \arrow[r] & 0. \end{tikzcd} \tag{6.10}$$

Note that $H_\mathbf{q}$ changes along the solution curve $\mathbf{q} \subset \mathbb{S}^2$. Le us restate the problem in terms of the Lie algebra and subalgebra, in order to work in a linear space. Consider then the short exact sequence

$$\begin{tikzcd} 0 \arrow[r] & \mathfrak{h}_\mathbf{q} \arrow[r] & \mathfrak{so}(3) \arrow[r, "\perp"] & \mathfrak{so}(3)/\mathfrak{h}_\mathbf{q} \arrow[r] & 0, \end{tikzcd} \tag{6.11}$$

where $\perp$ maps an element to the Lie algebra to its projection ontp the orthogonal complement to $\mathfrak{h}_\mathbf{q}$; trivially, $\mathfrak{h}_\mathbf{q} \ni \eta \mapsto 0$. The sequence (6.11) is isomorphic to the sequence (6.10) via the exponential map. If we represent the tangent space of SO(3) using a right-trivialization, $\mathfrak{h}_\mathbf{q}$ represents the vertical subspace, and the $\perp$ mapping corresponds then to the choice of a connection on the principal bundle $(\mathrm{SO}(3), \mathrm{SO}(3)/H_\mathbf{q}, \pi)$.

The effect of isotropy in (6.8) could be avoided by considering only horizontal vector fields on SO(3). If we consider an initial condition for $\boldsymbol{\omega}$ which respects the orthogonality condition (6.3), we have that (6.5b) preserve that structure for the flow. Since we identify $T\mathbb{S}^2$ with $\mathfrak{so}(3)$, this is equivalent to ensuring that the infinitesimal generator of the flow on SO(3) remains in $\mathfrak{so}(3)/\mathfrak{h}_{\mathbf{q}(t)}$, that is, (6.5b) provides a horizontal vector field on SO(3) and providing that $\mathbf{R}(0) = I$ we have $\mathbf{R}(t) \subset \mathrm{SO}(3)/H_{\mathbf{q}(t)}$.

**Remark** In a discrete framework, the iteration rule might not preserve the orthogonality between $\boldsymbol{\omega}$ and $\mathbf{q}$, unless it is explicitly imposed; that is, in order to preserve its geometric properties this approach should be suitably modified for numerical integration. In this Thesis we just ignore this fact and apply without any modifications Lie methods to the integration of (6.7). As suggested in [LLM09], considering a discrete connection [Leo04] might lead to a proper horizontal lift for the discrete flow. Further investigations will be devoted to this issue.

# 7 Numerical examples

In this Chapter some numerical experiments are presented in order to validate the integration method for systems on two-spheres described in Chapter 6. The basic idea of this approach is lift the dynamics from two-spheres to the Lie group of rotation matrices SO(3); doing so, one can employ off-the-shelf Lie group methods for numerical integration of the original problem. The simulations will confirm that our approach preserves the accuracy order of the method also for the flow on $\mathbb{S}^2$; this implies that arbitrarily high-order numerical algorithms can be obtained for integration on two-spheres without any additional effort.

In Section 7.1 the Lie methods we use for our numerical experiments are described; all the simulation results are detailed and commented in 7.2. Finally, Section 7.3 concludes our work and gives some indications for future investigations.

## 7.1 Integrators

According to the approach we are introducing, the problem of integration of the system dynamics of systems evolving on two-spheres is transformed into the problem of solving for the right-trivialized dynamics of a certain system on a Lie group. For convenience we rewrite the dynamics (6.7) as

$$\begin{cases} \dot{\mathbf{R}}_i = \hat{\boldsymbol{\Omega}}_i \mathbf{R}_i, \\ \mathbf{q}_i = \mathbf{R}_i \mathbf{q}(0), \\ \dot{\boldsymbol{\Omega}}_i = \alpha_i(\mathbf{q}, \boldsymbol{\Omega}), \end{cases}$$

for $i = 1, \dots, n$, where $\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_n \end{bmatrix}^T$, $\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\Omega}_1 & \cdots & \boldsymbol{\Omega}_n \end{bmatrix}^T$, and $\alpha_i$ represents the spatial angular velocity.

We choose classical Lie group methods to solve for this dynamics: the Explicit Lie-Newmark method (subsection 7.1.1), a second-order symmetric reversible integrator, and two fourth-order methods from the class of Runge-Kutta Munthe-Kaas integrators (subsection 7.1.2). Each method is different from the other: our intention is not to compare the algorithms' performances, but to provide a numerical evidence of the corresponding properties of the generated flow on two-spheres.

The numerical results on $\mathbb{S}^2$ are compared with the discrete flow obtained with the classical RATTLE integrator, a second-order symplectic method which is the reference method for constrained systems. According to our notation, the RATTLE method is recalled in subsection 7.1.3.

### 7.1.1 Explicit Lie-Newmark (ELN)

The explicit Lie-Newmark method has been previously described in 4.1.1. In this framework we will use a spatial version of the method [SVQ88], which results as

$$\begin{cases} \mathbf{\Omega}_{i,k+\frac{1}{2}} = \mathbf{\Omega}_{i,k} + \frac{h}{2}\alpha(\mathbf{q}_k, \mathbf{\Omega}_k) & (7.1a) \\ \mathbf{R}_{i,k+1} = \exp(h\mathbf{\Omega}_{i,k+\frac{1}{2}})\mathbf{R}_{i,k} & (7.1b) \\ \mathbf{q}_{i,k+1} = \mathbf{R}_{i,k+1}\mathbf{q}_{i,k} & (7.1c) \\ \mathbf{\Omega}_{i,k+1} = \mathbf{\Omega}_{i,k+\frac{1}{2}} + \frac{h}{2}\alpha_i(\mathbf{q}_{k+1}, \mathbf{\Omega}_{k+1}). & (7.1d) \end{cases}$$

This method is fully explicit if the spatial angular acceleration does not depend on the velocity; otherwise, (7.1d) is implicit in the velocity and, e.g., a Newton method can be employed to solve fir the velocity update. We recall that this method is symmetric, reversible and second-order accurate.

### 7.1.2 Runge-Kutta Munthe-Kaas methods

These methods have been introduced by Munthe-Kaas in several papers [MK95, MK98, MK99], as the natural extension to Lie groups of classical Runge-Kutta methods on vector spaces [HLW06]. Differently from Crouch-Grossman approach [CG93], which compute the approximations of the flow in the Lie group, RKMK methods perform the computations in the algebra; in particular, as we will see, this implies that the order condition on RKMK discrete flow is simpler that for CG methods.

Be $G$ a Lie group and $\mathfrak{g}$ its Lie algebra. As explained in [MKZ97], these methods consider Lie-type equations like

$$\dot{g} = y(g) \cdot g, \; g \in G, \; y : G \to \mathfrak{g}. \tag{7.2}$$

First of all, define the (left) Lie algebra action $\lambda : \mathfrak{g} \times G \to G$ as

$$\lambda(\xi, g) = \exp(\xi)g;$$

and recall the expression of $\mathrm{d}\exp_\xi^{-1} : \mathfrak{g} \to \mathfrak{g}$

$$\mathrm{d}\exp_\xi^{-1}(\eta) = \sum_{k=0}^{\infty} \frac{B_k}{k!} \underbrace{[\xi, [\xi, [\dots [}_{k}\xi, \eta]]]],$$

where $B_k$ is the $k$-th Bernoulli number. A $q$th-order approximation of $\mathrm{d}\exp^{-1}$ can be obtained truncating the infinite sum

$$\mathrm{d}\exp^{-1}(\xi,\eta,q) = \sum_{k=0}^{q-1} \frac{B_k}{k!} \underbrace{[\xi,[\xi,[\ldots[}_{k}\xi,\eta]]]].$$

Given $y_0 \in G$ and a timestep $h$, a $s-$stages Runge-Kutta Munthe-Kaas iteration rule is then defined as

```
for  $i = 1,2,\ldots,s$
     $u_i = h\sum_{j=1}^{s} a_{ij}\tilde{k}_j$
     $k_i = y(c_i h, \lambda(u_i, y_0))$
     $\tilde{k}_i = \mathrm{d}\exp^{-1}(u_i, k_i, q)$
end
$v = h\sum_{j=1}^{s} b_j\tilde{k}_j$
$y_1 = \lambda(v, y_0),$
```

where the coefficients $a, b, c$ which characterize the method are included in a Butcher tableau

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} . \tag{7.3}$$

A fundamental theorem stated in [MK98] assures that, if the Butcher tableau (7.3) satisfies the classical Runge-Kutta order conditions up to order $q$, then RKMK method defines at least a $q-$th order method if a $q-$th order approximation of $\mathrm{d}\exp^{-1}$ is used. This result implies that all the order conditions which hold in linear cases also hold for Munthe-Kaas methods; high-order method can thus be obtained using classical theory [HLW06].

The dynamics (6.7) evolve on the Lie group $G = \mathrm{SO}(3) \times \mathfrak{so}(3)$, whose Lie algebra can be represented as $\mathfrak{g} = \mathfrak{so}(3) \times \mathfrak{so}(3)$. As shown in [EM98], the operations on this group are:

| | |
|---|---|
| Group operation (product) of $G$ | $(g,\xi)\cdot(h,\eta) = (a\cdot b, \xi+\eta)$ |
| Addition in $\mathfrak{g}$ | $(\xi,\alpha)+(\eta,\beta) = (\xi+\eta, \alpha+\beta)$ |
| Lie bracket in $\mathfrak{g}$ | $[(\xi,\alpha),(\eta,\beta)] = ([\xi,\eta],0)$ |
| Multiplication by scalar in $\mathfrak{g}$ | $a(\xi,\alpha) = (a\xi, a\alpha)$ |
| Exponential map from $\mathfrak{g}$ to $G$ | $\exp((\xi,\alpha)) = (\exp(\xi),\alpha).$ |

In the case we are considering, the vector field in (7.2) becomes

$$y(\mathbf{R},\boldsymbol{\Omega}) = (\boldsymbol{\Omega}, \alpha(\mathbf{q},\boldsymbol{\Omega}));$$

since we work with 4th-order methods, we will use a 4th-order approximation of the $\mathrm{d}\exp^{-1}$ map, given by

$$\mathrm{d}\exp^{-1}((\xi,\alpha),(\eta,\beta),4) = \begin{bmatrix} \left(I - \frac{1}{2}\xi + \frac{1}{12}\xi^2\right)\eta \\ \beta \end{bmatrix}.$$

**Explicit 4th-order Runge-Kutta Munthe-Kaas (RKMK4)**

The most famous explicit Runge-Kutta methods is sure given by

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array}.$$

This method has no properties, but the fact that it is fully explicit makes its implementation simple.

**Implicit 4th-order Runge-Kutta Munthe-Kaas Gauss-Legendre method (RKMKGL4)**

*Collocation methods* are based on the idea of approximating ODE solution choosing a class of candidate solutions and a number of points *(collocation points)* where the solution must satisfy the given differential equation [HLW06]. The choice of two collocation points as Gaussian quadrature nodes, as did by Hammer and Hollingsworth in 1995 [HH55], leads to a fourth-order implicit Runge-Kutta method, described by the following Butcher tableau:

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

It can be proved that, in the linear case, this method is symplectic, and preserves total linear and angular momentum for Hamiltonian systems with pairwise distance-dependent interactions [LR04].

This method is fully implicit, so an algorithm to solve nonlinear equations is required. In this case, instead of the Newton method which requires the computation of a Jacobian matrix, we use the fixed point method [LR04]; in our notation, the fixed point iteration can be written as follows:

```
$u_i = 0$ for $i = 1, \dots, s$
$k_i = y(c_i h, \lambda(0, y_0))$
$\tilde{k}_i = \mathrm{d}\exp^{-1}(0, k_i, 4)$
$u_i^{\text{new}} = h \sum_{j=1}^{s} a_{ij} \tilde{k}_j$
while  $\exists i : \|u_i^{\text{new}} - u_i\| >$threshold
          $u_i = u_i^{\text{new}}$ for $i = 1, \dots, s$
           for  $i = 1, \dots, s$
                $u_i = u_i^{\text{new}}$
                $k_i = y(c_i h, \lambda(u_i, y_0))$
                $\tilde{k}_i = \mathrm{d}\exp^{-1}(u_i, k_i, 4)$
                $u_i^{\text{new}} = h \sum_{j=1}^{s} a_{ij} \tilde{k}_j$
           end
end
```

Differently from the Newton method, whose convergence has a quadratic speed, the fixed point iteration has a linear convergence speed, so it typically requires $\sim 10-20$ iterations to get the result with a good precision.

### 7.1.3 RATTLE

RATTLE method [And83] was proposed for the first time in 1983 as a reformulation of SHAKE algorithm for dynamical systems with constraints. Ten years later, Leimkuhler and Steel proved that RATTLE and SHAKE are equivalent and they are both symplectic [LS94a]. In this Thesis we consider the second-order version of RATTLE; higher-order schemes are discussed in [LR04], where we refer the reader also for more insights about RATTLE method. Be $f(\mathbf{q}) = \mathbf{0}$ the holonomic constraint[1], and denote with

$$F(\mathbf{q}) \overset{\text{def}}{=} \frac{\partial f}{\partial \mathbf{q}}$$

the Jacobian of $f$. Given $(\mathbf{q}_k, \mathbf{v}_k)$ and a timestep $h$, the iteration rule of RATTLE method is

$$\begin{cases}
\mathbf{q}_{k+1} = \mathbf{q}_k + h\mathbf{v}_{k+\frac{1}{2}} & \text{(7.4a)} \\
m\mathbf{v}_{k+\frac{1}{2}} = m\mathbf{v}_k - \frac{h}{2}\left[\frac{\partial V}{\partial \mathbf{q}}(\mathbf{q}_k)\right]^T - \frac{h}{2}F^T(\mathbf{q}_k)\lambda_{k+1}^{\mathbf{q}} & \text{(7.4b)} \\
f(\mathbf{q}_{k+1}) = 0 & \text{(7.4c)} \\
m\mathbf{v}_{k+1} = m\mathbf{v}_{k+\frac{1}{2}} - \frac{h}{2}\left[\frac{\partial V}{\partial \mathbf{q}}(\mathbf{q}_{k+1})\right]^T - \frac{h}{2}F^T(\mathbf{q}_{k+1})\lambda_{k+1}^{\mathbf{v}} & \text{(7.4d)} \\
F(\mathbf{q}_{k+1})\mathbf{v}_{k+1} = 0, & \text{(7.4e)}
\end{cases}$$

[1] A constraint is *holonomic* when it restricts the trajectory of the system to a smooth hypersurface in the configuration space, which can be represented as the level set of a constraint function $f = \left[f_1, \dots, f_k\right]$. Since the contraints are assumed to be linearly independent, the Jacobian $J_f$ is full row rank.

where (7.4c) and (7.4d) are respectively position and velocity constraints, while $\lambda^{\mathbf{q}}$ and $\lambda^{\mathbf{v}}$ are multipliers chosen in order to enforce those constraints. Note that this method works with linear velocity $\mathbf{v}$.

## 7.2 Numerical results

In this Section the numerical results of several relevant dynamical systems solved with the algorithms described in Section 7.1 are presented; each subsection is dedicated to a different example, specified in the subsection title. All the simulations have been performed on a P8600 Dual Core x64 pc (2.40 GHz), 4 GB RAM, running Windows 7 Professional and MATLAB R2009b.

### 7.2.1 Spherical pendulum

The spherical pendulum consists of a mass $m$ attached to a joint via a massless rod of length $l$, moving without friction in a uniform gravitational field. It is a 3-D generalization of the plane pendulum, and has two degree-of-freedom. We denote with $g \in \mathbb{R}$ the constant of gravitational acceleration, $\mathbf{k} = [0\ 0\ 1]^T \in \mathbb{R}^3$ the direction of gravity; the vector $\mathbf{q} \in \mathbb{S}^2$ represents the direction from the pivot to the mass: the configuration space is therefore $\mathbb{S}^2$. Equivalently, we can say that the configuration space is $\mathbb{R}^3$ and the system is subject to the holonomic constraint [MLS94]

$$f(\mathbf{q}) = \mathbf{q}^T\mathbf{q} - l^2 = 0.$$

The kinetic energy is given by

$$\begin{aligned} T(\dot{\mathbf{q}}) &= \frac{1}{2}m\|l\dot{\mathbf{q}}\|^2 \\ &= \frac{1}{2}M\|\dot{\mathbf{q}}\|^2, \end{aligned}$$

where $M = ml^2$; the potential energy is given by

$$\begin{aligned} V(\mathbf{q}) &= mgl\mathbf{k}\cdot\mathbf{q} \\ &= Mg\mathbf{k}\cdot\mathbf{q}. \end{aligned}$$

The Lagrangian is then

$$L(\mathbf{q},\dot{\mathbf{q}}) = \frac{1}{2}M\|\dot{\mathbf{q}}\|^2 - Mg\mathbf{k}\cdot\mathbf{q}$$

and the dynamics is guided by the constrained Euler-Lagrange equations [MLS94]:

$$\begin{cases} \dot{\mathbf{q}} = \mathbf{v}, \\ \dot{\mathbf{v}} = -M\mathbf{k} - \mathbf{J}_f(\mathbf{q})^T\lambda, \\ f(\mathbf{q}) = 0, \end{cases}$$

where $\mathbf{J}_f$ is the Jacobian matrix of the constraint function and $\lambda$ is a Lagrange multiplier.

An alternative way to derive the dynamics of the spherical pendulum is considering the variational approach [LLM09] and writing the Euler-Lagrange equation on two-sphere:

$$\begin{cases} \dot{\mathbf{q}} = \boldsymbol{\omega} \times \mathbf{q}, \\ \dot{\boldsymbol{\omega}} = \frac{g}{l}\mathbf{k} \cdot \mathbf{q}, \end{cases} \tag{7.5}$$

For our numerical simulations, the mass and the rod length are respectively $m = 2$, $r = 2$, the gravity acceleration $g = 9.81$. The initial conditions are

$$\mathbf{q}_0 = \begin{bmatrix} 0.577350269189626 \\ 0.577350269189626 \\ -0.577350269189626 \end{bmatrix}, \quad \dot{\mathbf{q}}_0 = \begin{bmatrix} -3.46410161513776 \\ 3.46410161513776 \\ 0 \end{bmatrix}, \quad \boldsymbol{\omega}_0 = \mathbf{q}_0 \times \dot{\mathbf{q}}_0 = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

(see Appendix A.1); we simulate the system with a timestep $h = 0.01$ up to $T = 200$. In Figure 7.1 we show the total energy, while in Figure 7.2 we show the errors on the preservation of $T\mathbb{S}^2$, that is, the norm $\|\mathbf{q}\|$ and the orthogonality $\mathbf{q}\cdot\boldsymbol{\omega}$. We remark that in this particular case ELN and RKMK4 are fully explicit, while RKMKGL4 and RATTLE require the use of non-linear solver methods to complete an iteration step.

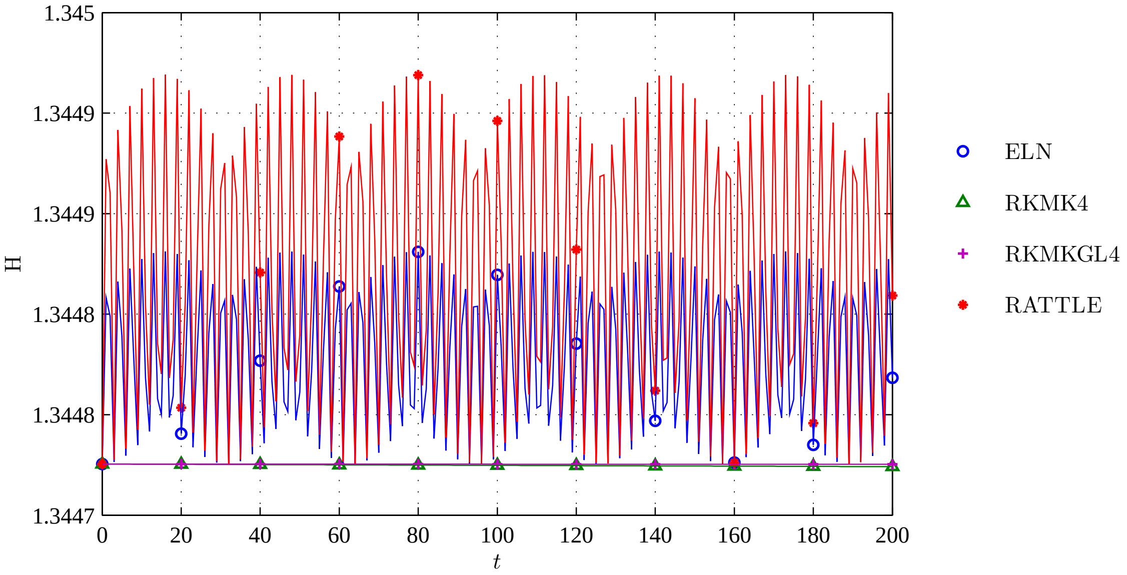


Figure 7.1: Total energy of the single spherical pendulum computed with ELN, RKMK4, RKMKGL4, RATTLE integrators. ELN and RATTLE show a bounded oscillatory behavior, while RKMK methods exhibit a linear drift. Note that ELN absolute value is lower than RATTLE. Initial conditions and parameters used are provided in the text.

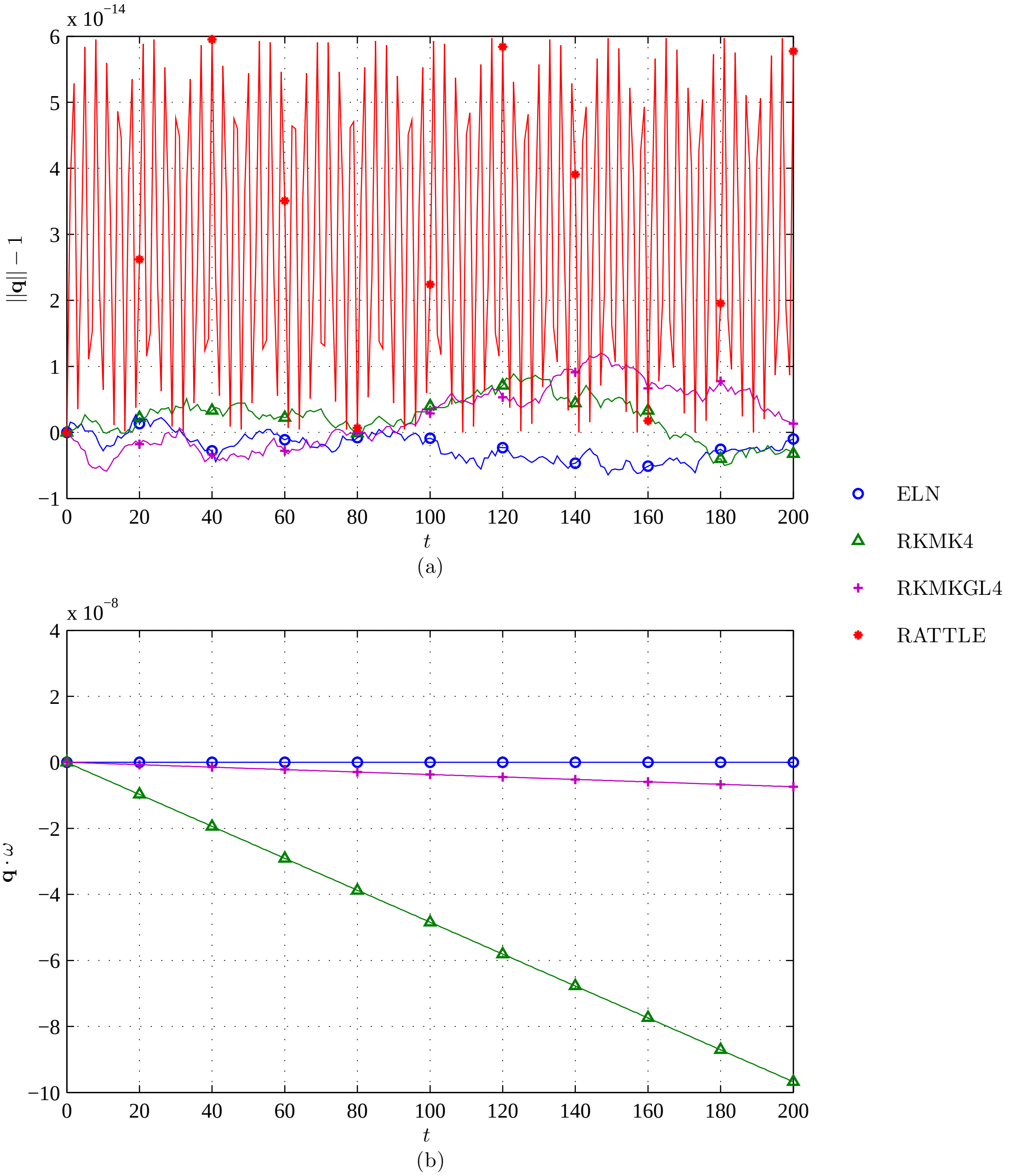


Figure 7.2: Preservation of the configuration space properties by ELN, RKMK4, RKMKGL4, RATTLE integrators: (a) position norm; (b) orthogonality between the position and the angular velocity. In (b) RATTLE is not plotted, since it works with the linear velocity. The unitary norm of $\mathbf{q}$ is preserved up to machine precision by Lie group methods, with a better performance than the one exhibited by RATTLE. Notably, ELN seems to conserve also the orthogonality between $\mathbf{q}$ and $\boldsymbol{\omega}$, while RKMK methods show an evident drift. Initial conditions and parameters used are provided in the text.

ELN energy behavior appears very similar to RATTLE (recall that RATTLE is symplectic, thus Theorem 1.2.13 states it nearly preserves the total energy over exponentially long times), while preserving up to machine precision the structure of $T\mathbb{S}^2$. On the contrary, despite their fourth order, RKMK methods show a linear drift in the energy, and do not preserve the orthogonality $\mathbf{q} \perp \boldsymbol{\omega}$; these facts suggest that, according with general belief on non-symplectic Runge-Kutta methods, they might not be employed for long-time simulations. We recall that the preservation of condition (6.3) is not necessarily required for the methods to be meaningful, as these numerical results also display.

The time-precision diagrams on the position and on the angular velocity are plotted in Figure 7.3. This is a numerical proof of the fact that each Lie algorithm maintains the same original accuracy order also for the flow on $T\mathbb{S}^2$ and, therefore, we can obtain an arbitrarily small global error using high-order RKMK methods (whose coefficients are relatively easy to compute). As we were expecting, implicit RKMKGL4 exhibits a smaller error than RKMK4; remarkably, RATTLE shows a higher error than ELN.

To complete the analysis, the running time of all the methods is shown in Table 7.1; RKMKGL4 requires five times and 227 times the run-time of RKMK4 and ELN, respectively. Therefore, considering the high time amount required by RKMKGL4, one should employ such integrator for short simulation and when a high accuracy is required.

| Algorithm | Run-time [s] |
|---|---|
| ELN | 0.34 |
| RKMK4 | 17.41 |
| RKMKGL4 | 83.17 |
| RATTLE | 2.72 |

Table 7.1: Mean running time over three simulations for the single spherical pendulum described in the text.

### 7.2.2 Double spherical pendulum

The double spherical pendulum consists on two serially connected spherical pendulums, moving without friction in a uniform gravitational field [MS93]. We denote with the real positive constants $m_1, m_2, l_1, l_2$ the masses and the lengths of the two pendulums, $g \in \mathbb{R}$ the constant of gravitational acceleration, $\mathbf{k} = [0\ 0\ 1]^T \in \mathbb{R}^3$ the direction of gravity; the vector $\mathbf{q}_1 \in \mathbb{S}^2$ represents the direction from the pivot to the first mass, and the vector $\mathbf{q}_2 \in \mathbb{S}^2$ represents the direction from the first to the second mass. Thus, the configuration space is $Q = \mathbb{S}^2 \times \mathbb{S}^2$; in an equivalent way, we can say that the configuration space is $\mathbb{R}^3 \times \mathbb{R}^3$ and the

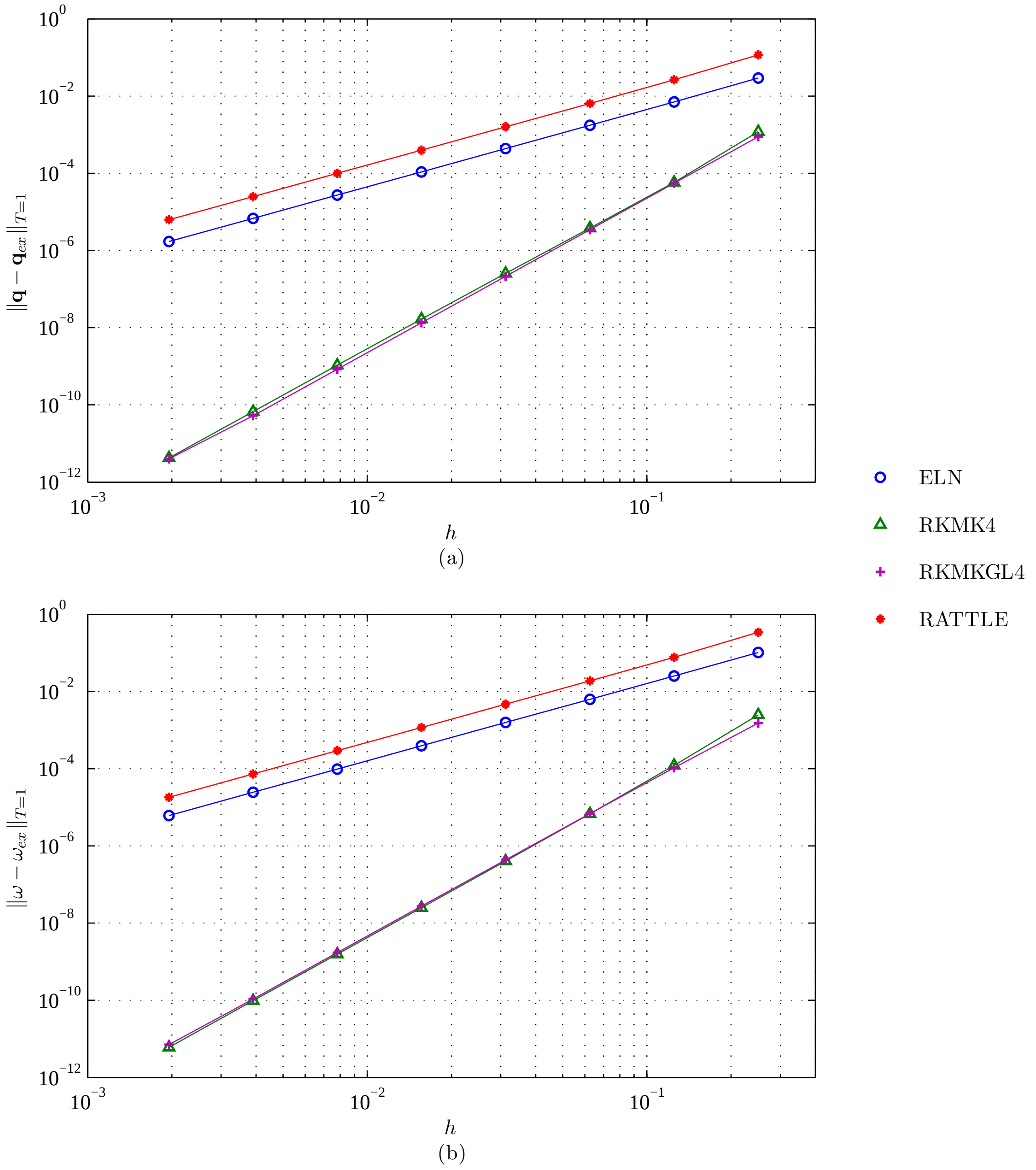


Figure 7.3: Global error after $T = 1$ on (a) the position $\mathbf{q}$ and (b) the angular velocity $\boldsymbol{\omega}$ of the single spherical pendulum. The error is evaluated for $h = 2^i$, $i = -9, \ldots, -2$. The reference solution is obtained integrating (7.5) using MATLAB function `ode45` with low tolerance. Note that the Lie group methods' order is preserved also for the flow on $T\mathbb{S}^2$, and RATTLE exhibits the maximum error norm. Initial conditions and parameters used are provided in the text.

system is subject to the holonomic constraint [MLS94]

$$f(\mathbf{q}_1, \mathbf{q}_2) = \begin{bmatrix} \mathbf{q}_1^T \mathbf{q}_1 - l_1 \\ \\ \mathbf{q}_2^T \mathbf{q}_2 - l_2 \end{bmatrix} = \mathbf{0}.$$

The kinetic energy is given by

$$\begin{aligned}T(\dot{\mathbf{q}}_1,\dot{\mathbf{q}}_2) &= \frac{1}{2}m_1\|l_1\dot{\mathbf{q}}_1\|^2+\frac{1}{2}m_2\|l_1\dot{\mathbf{q}}_1+l_2\dot{\mathbf{q}}_2\|^2\\ &= \frac{(m_1+m_2)l_1^2}{2}\|\dot{\mathbf{q}}_1\|^2+m_2l_1l_2\dot{\mathbf{q}}_1^T\dot{\mathbf{q}}_2+\frac{m_2l_2^2}{2}\|\dot{\mathbf{q}}_2\|^2\\ &= \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{M}\dot{\mathbf{q}},\end{aligned}$$

where $\dot{\mathbf{q}}=\begin{bmatrix}\dot{\mathbf{q}}_1^T & \dot{\mathbf{q}}_2^T\end{bmatrix}^T$ and $\mathbf{M}$ is the symmetric positive definite inertia matrix defined as

$$\mathbf{M}=\begin{bmatrix}(m_1+m_2)l_1^2\mathbf{I}_3 & m_2l_1l_2\mathbf{I}_3\\ m_2l_1l_2\mathbf{I}_3 & m_2l_2^2\mathbf{I}_3\end{bmatrix}\overset{\text{def}}{=}\begin{bmatrix}M_{11}\mathbf{I}_3 & M_{12}\mathbf{I}_3\\ M_{21}\mathbf{I}_3 & M_{22}\mathbf{I}_3\end{bmatrix}.$$

The potential energy is given by

$$\begin{aligned}V(\mathbf{q}_1,\mathbf{q}_2) &= -(m_1+m_2)l_1g\,\mathbf{k}\cdot\mathbf{q}_1-m_2l_2g\,\mathbf{k}\cdot\mathbf{q}_2\\ &= -\frac{M_{11}}{l_1}g\,\mathbf{k}\cdot\mathbf{q}_1-\frac{M_{22}}{l_2}g\,\mathbf{k}\cdot\mathbf{q}_2.\end{aligned}$$

The Lagrangian is then

$$L(\mathbf{q}_1,\mathbf{q}_2,\dot{\mathbf{q}}_1,\dot{\mathbf{q}}_2)=\frac{M_{11}}{2}\|\dot{\mathbf{q}}_1\|^2+M_{12}\dot{\mathbf{q}}_1^T\dot{\mathbf{q}}_2+\frac{M_{22}}{2}\|\dot{\mathbf{q}}_2\|^2+\frac{M_{11}}{l_1}g\,\mathbf{k}\cdot\mathbf{q}_1+\frac{M_{22}}{l_2}g\,\mathbf{k}\cdot\mathbf{q}_2,$$

and the constrained Euler-Lagrange equations of motion are given by

$$\begin{cases}\dot{\mathbf{q}}=\mathbf{v},\\ \dot{\mathbf{v}}=\mathbf{M}^{-1}\begin{bmatrix}\dfrac{M_{11}}{l_1}g\,\mathbf{k}\\ \dfrac{M_{22}}{l_2}g\,\mathbf{k}\end{bmatrix}-\mathbf{J}_f(\mathbf{q})^T\boldsymbol{\lambda},\\ f(\mathbf{q})=0,\end{cases}$$

where $\mathbf{J}_f$ is the Jacobian of the constraint function and $\boldsymbol{\lambda}=\begin{bmatrix}\lambda_1 & \lambda_2\end{bmatrix}^T$ is the vector of Lagrange multipliers.

The conjugate momenta are

$$\begin{aligned}\mathbf{p}_1 &= \frac{\partial L}{\partial\dot{\mathbf{q}}_1}=M_{11}\dot{\mathbf{q}}_1+M_{12}\dot{\mathbf{q}}_2\\ \mathbf{p}_2 &= \frac{\partial L}{\partial\dot{\mathbf{q}}_2}=M_{12}\dot{\mathbf{q}}_1+M_{22}\dot{\mathbf{q}}_2,\end{aligned}$$

therefore the Hamiltonian is

$$H(\mathbf{q}_1,\mathbf{q}_2,\mathbf{p}_q,\mathbf{p}_2)=\frac{1}{2\left(M_{11}M_{22}-M_{12}^2\right)},\left(\mathbf{p}_1^TM_{22}\mathbf{p}_1-2\mathbf{p}_1^TM_{12}\mathbf{p}_2+\mathbf{p}_2^TM_{11}\mathbf{p}_2\right).$$

Be $G$ the group of the rotations about the direction of gravity. $G$ acts on the configuration space $Q$ by

$$\Phi_g(\mathbf{q}_1, \mathbf{q}_2) = (g\mathbf{q}_1, g\mathbf{q}_2);$$

the lifted action to the phase space $T^*Q$ is given by

$$T^*_{(g\mathbf{q}_1, g\mathbf{q}_2)}\Phi_{g^{-1}}(\mathbf{q}_1, \mathbf{q}_2, \mathbf{p}_1, \mathbf{p}_2) = g \cdot (\mathbf{q}_1, \mathbf{q}_2, \mathbf{p}_1, \mathbf{p}_2) = (g\mathbf{q}_1, g\mathbf{q}_2, g\mathbf{p}_1, g\mathbf{p}_2),$$

and the corresponding momentum map is

$$J(\mathbf{q}_1, \mathbf{q}_2, \mathbf{p}_1, \mathbf{p}_2) = (\mathbf{q}_1 \times \mathbf{p}_1 + \mathbf{q}_2 \times \mathbf{p}_2) \cdot \mathbf{k}, \tag{7.6}$$

which corresponds to the total angular momentum about the $\mathbf{k}$ direction. It is easy to verify that the Hamiltonian is invariant under $\Phi_g$: therefore, the corresponding momentum map (7.6) is preserved along the flow of the system [MR99, Theorem 11.4.1].

As shown in [LLM09], one can instead consider the Euler-Lagrange equations on the unit sphere and obtain

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \\ \dot{\mathbf{q}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_1 \times \mathbf{q}_1 \\ \\ \boldsymbol{\omega}_2 \times \mathbf{q}_2 \end{bmatrix}, \\ \begin{bmatrix} \dot{\boldsymbol{\omega}}_1 \\ \\ \dot{\boldsymbol{\omega}}_2 \end{bmatrix} = \begin{bmatrix} M_{11}\mathbf{I}_3 & -M_{12}\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2 \\ \\ -M_{12}\hat{\mathbf{q}}_2\hat{\mathbf{q}}_1 & M_{22}\mathbf{I}_3 \end{bmatrix}^{-1} \begin{bmatrix} M_{12}\|\boldsymbol{\omega}_2\|^2\hat{\mathbf{q}}_1\mathbf{q}_2 + \dfrac{M_{11}}{l_1}g\hat{\mathbf{q}}_1\mathbf{k} \\ M_{12}\|\boldsymbol{\omega}_1\|^2\hat{\mathbf{q}}_2\mathbf{q}_1 + \dfrac{M_{22}}{l_2}g\hat{\mathbf{q}}_2\mathbf{k} \end{bmatrix} \\ \quad =: \overline{\mathbf{M}}^{-1}(\mathbf{q}_1, \mathbf{q}_2) \begin{bmatrix} M_{12}\|\boldsymbol{\omega}_2\|^2\hat{\mathbf{q}}_1\mathbf{q}_2 + \dfrac{M_{11}}{l_1}g\hat{\mathbf{q}}_1\mathbf{k} \\ M_{12}\|\boldsymbol{\omega}_1\|^2\hat{\mathbf{q}}_2\mathbf{q}_1 + \dfrac{M_{22}}{l_2}g\hat{\mathbf{q}}_2\mathbf{k} \end{bmatrix} = \alpha(\mathbf{q}_1, \mathbf{q}_2). \end{cases} \tag{7.7}$$

We show that $\overline{\mathbf{M}}$ is an invertible matrix. After some tedious but straightforward calculations we find:

$$\det(\overline{\mathbf{M}}) = M_{11}M_{22}(M_{11}M_{22} - M_{12}(\mathbf{q}_1 \cdot \mathbf{q}_2)^2)(M_{11}M_{22} - M_{12}^2) > 0$$

since $M_{11}, M_{22} > 0$, $M_{11}M_{22} > M_{12}^2$ (the inertia matrix is positive definite) and $(\mathbf{q}_1 \cdot \mathbf{q}_2)^2 \leq \|\mathbf{q}_1\|^2\|\mathbf{q}_2\|^2 = 1$ (Cauchy-Schwartz inequality).

Note that in the expression of the angular acceleration we need to compute the inverse of a 6 × 6 matrix. The formula of the matrix inversion in block form comes really useful here in order to reduce the computational cost. Since $\overline{\mathbf{M}}$, its upper-left block and its lower-right block are invertible, we can write

$$\overline{\mathbf{M}}^{-1} = \begin{bmatrix} \left(M_{11}\mathbf{I}_3 - \dfrac{M_{12}^2}{M_{22}}\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2\hat{\mathbf{q}}_2\hat{\mathbf{q}}_1\right)^{-1} & \dfrac{M_{12}}{M_{11}}\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2\left(M_{22}\mathbf{I}_3 - \dfrac{M_{12}^2}{M_{11}}\hat{\mathbf{q}}_2\hat{\mathbf{q}}_1\hat{\mathbf{q}}_1\hat{q}_2\right)^{-1} \\ \dfrac{M_{12}}{M_{22}}\hat{\mathbf{q}}_2\hat{\mathbf{q}}_1\left(M_{11}\mathbf{I}_3 - \dfrac{M_{12}^2}{M_{22}}\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2\hat{\mathbf{q}}_2\hat{\mathbf{q}}_1\right)^{-1} & \left(M_{22}\mathbf{I}_3 - \dfrac{M_{12}^2}{M_{11}}\hat{\mathbf{q}}_2\hat{\mathbf{q}}_1\hat{\mathbf{q}}_1\hat{\mathbf{q}}_2\right)^{-1} \end{bmatrix}.$$

More technical details about the inversion of a block matrix can be found in Appendix A.2.

For our numerical simulation we choose $m_1 = 2$, $m_2 = 1$, $l_1 = l_2 = 1$ and $g = 9.81$. The initial conditions are

$$\mathbf{q}_{1,0} = \begin{bmatrix} 0.866019052628739 \\ 0 \\ 0.500011000363013 \end{bmatrix}, \quad \mathbf{q}_{2,0} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix},$$

$$\dot{\mathbf{q}}_{1,0} = \begin{bmatrix} 0 \\ 0.865980947790423 \\ 0 \end{bmatrix}, \quad \dot{\mathbf{q}}_{2,0} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix},$$

which correspond to

$$\boldsymbol{\omega}_{1,0} = \mathbf{q}_{1,0} \times \dot{\mathbf{q}}_{1,0} = \begin{bmatrix} -0.433 \\ 0 \\ 0.749956 \end{bmatrix}, \quad \boldsymbol{\omega}_{2,0} = \mathbf{q}_{2,0} \times \dot{\mathbf{q}}_{2,0} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Simulation horizon is 200 [s], the timestep is $h = 0.01$ [s]. Figure 7.4 shows the total energy and the total angular momentum about the $\mathbf{k}$ direction. The ELN method exhibits an energy behavior comparable to RATTLE (recall that RATTLE is symplectic, so its energy error is bounded over exponentially long times and it exactly preserves the momentum), but ELN does not (nearly) preserve the angular momentum. RKMK methods both display an energy drift, while no overall drift is exhibited by RKMKGL4 in the total angular momentum.

Figure 7.5 shows the errors in the preservation of the $T\mathbb{S}^2$ structure by the discrete flows, and makes us gain some insight about the geometric properties of these numerical methods. The use of Lie methods guarantees a good preservation of the unitary norm of $\mathbf{q}$, up to numerical issues due to the computation of the matrix exponential: a finer zoom shows that the error on $||\mathbf{q}||$ is $\sim 10^{-14}$ at the end of this 200-seconds simulation. On the other hand, RKMK4 fails to preserve the orthogonality between the angular velocity and the position; notably, ELN error on the orthogonality is much smaller than RKMKGL4.

The time-precision diagrams for our integrators are presented in Figure 7.6. As in the single spherical pendulum case, all Lie methods maintain the same accuracy order also on the discrete flow on $T\mathbb{S}^2$. The implicit RKMKGL4 gives the smallest error both on position and velocity; all the integrators are more accurate than RATTLE method.

Some consideration about the run-time of the methods are in order at this point. Total running time needed for a 200 seconds simulation for each method is reported in Table 7.2. RKMK fourth-order methods are predictably more expensive than second-order methods. Let us focus on the comparison between second-order methods ELN and RATTLE: differently from the single spherical pendulum example, ELN is more expensive than RATTLE in this case. In fact, the inertia matrix $\mathbf{M}$ is not a block-diagonal matrix, and this implies that ELN iteration rule becomes implicit in the angular velocity; therefore ELN requires the use of a Newton

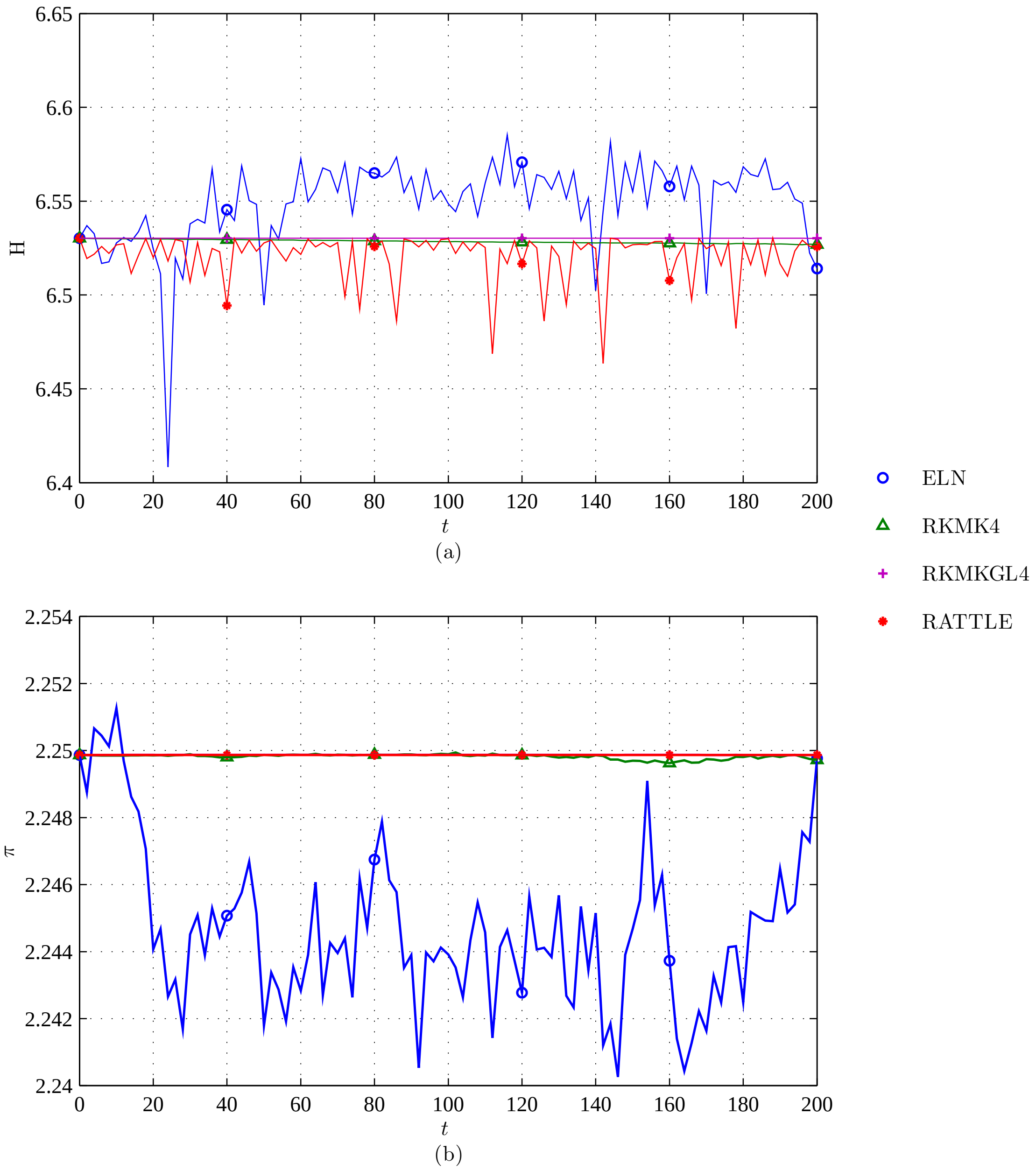


Figure 7.4: Invariants of the flow of the double spherical pendulum computed with ELN, RKMK4, RKMKGL4, RATTLE methods: (a) total energy; (b) total angular momentum about the **k** direction (7.6). ELN do not preserve neither the energy nor the momentum; RKMK methods exhibit a regular energy drift, while they render correctly the angular momentum. The symplectic RATTLE exactly preserves the momentum. Initial conditions and parameters used are provided in the text.

solver. On the other hand, RATTLE method needs to evaluate only two constraints per step. This explains the difference in the computational cost for ELN and RATTLE.

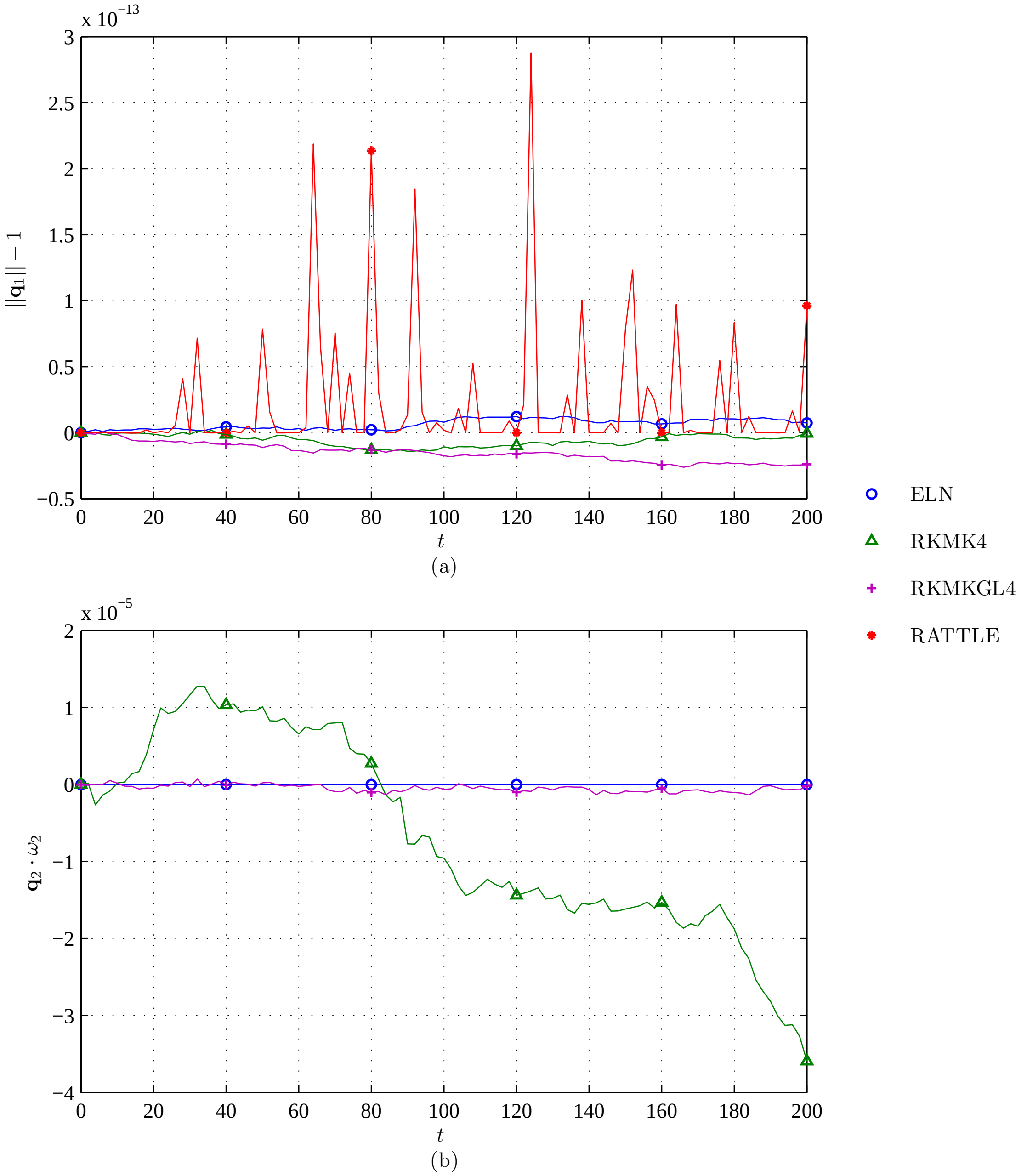


Figure 7.5: Errors in the preservation of the configuration space by ELN, RKMK4, RKMKGL4 and RATTLE methods: (a) position norm for $\mathbf{q}_1$; (b) orthogonality between the position and the angular velocity of the second mass. In (b) RATTLE is not plotted, since it works with the linear velocity. Due to their geometric approach, ELN, RKMK4 and RKMKGL4 preserve the unitary norm of the position up to machine precision; a drift in the orthogonality condition is instead exhibited by RKMK methods. Initial conditions and parameters are provided in the text.

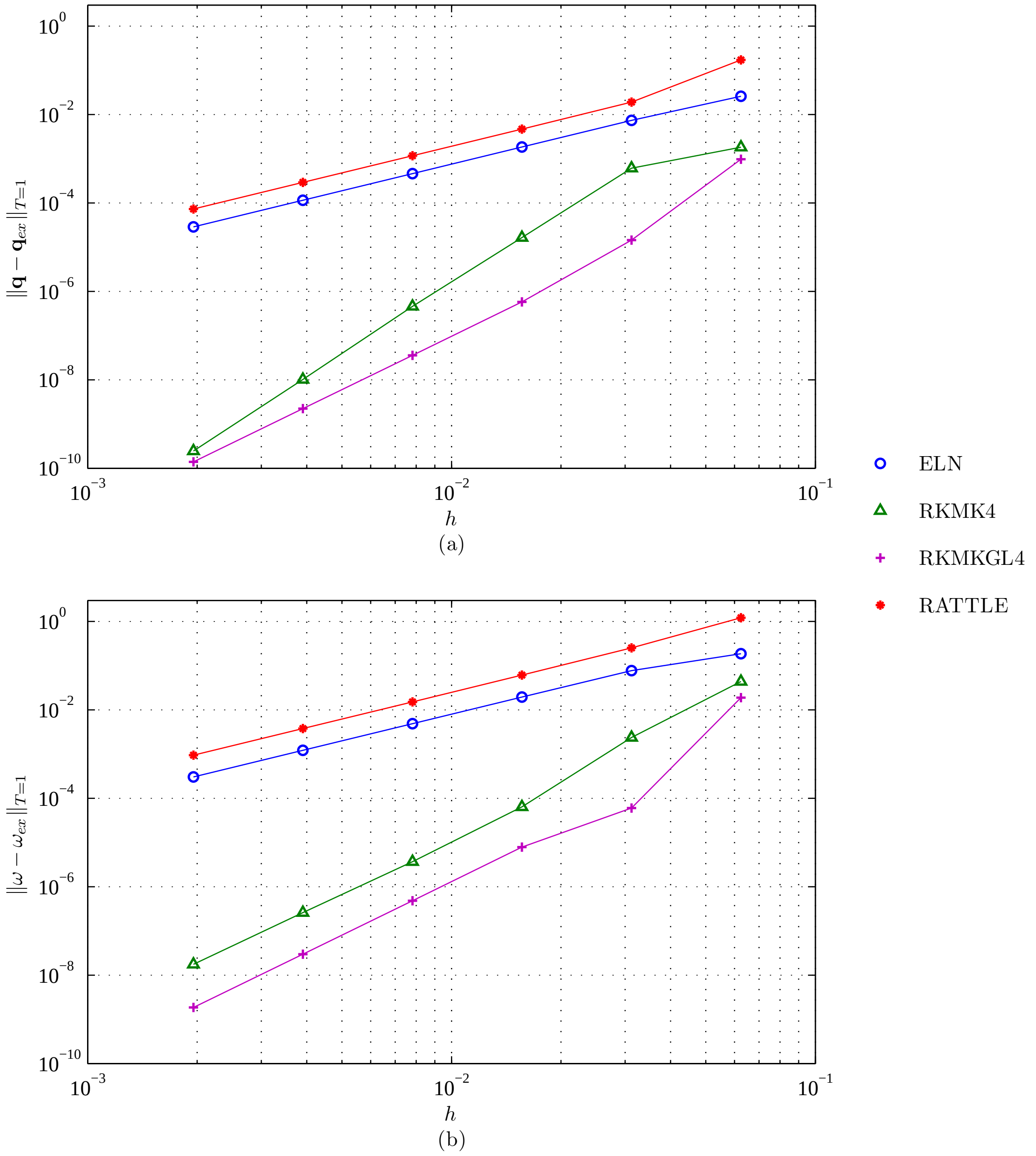


Figure 7.6: Global error after $T = 1$ on (a) the position $\mathbf{q}$ and (b) the corresponding angular velocity $\boldsymbol{\omega}$ for the double spherical pendulum. The error is evaluated for $h = 2^i$, $i = -9, \dots, -4$. The reference solution is obtained integrating (7.7) using MATLAB function `ode45` with low tolerance. ELN and RKMK methods' discrete flows show, respectively, second and fourth order of accuracy, and they exhibit a smallest error norm than RATTLE. Initial conditions and parameters used are provided in the text.

| Algorithm | Run-time [s] |
|---|---|
| ELN | 19.71 |
| RKMK4 | 44.67 |
| RKMKGL4 | 166.57 |
| RATTLE | 12.64 |

Table 7.2: Mean running time over three simulations for the double spherical pendulum described in the text.

### 7.2.3 Array of magnetic dipoles

Magnetic phenomena at a microscopic scale represent an active research field, especially due to their industrial application like microelectronics and data storage [DdA08]. Since the micromagnetism dynamics can be analytically solved only in special cases, numerical simulations play an important role in the understanding of the dynamical behavior of micromagnetic systems. A simple model for the dynamics of micromagnetic particles is constituted by a spatial array of magnetic dipoles, where each dipole is represented as a magnetic needle pinned at its middle point and left free to rotate under the action of the mutual magnetic field. For the sake of simplicity, we consider a uniform $\sqrt{n} \times \sqrt{n}$ grid, but the procedure can be easily extended to a more general case.

If we denote with $\mathbf{q}_i$ the direction from the south pole to the north pole of the $i$-th magnetic dipole, its momentum becomes $\nu_i \mathbf{q}_i$, where $\nu_i$ is the constant momentum magnitude, measured in $[\mathrm{A\,m}^2]$. The configuration space of this system is therefore $Q = \left(\mathbb{S}^2\right)^n$ or, equivalently, $Q \in \left(\mathbb{R}^3\right)^n$ and the system is subject to the holonomic constraint

$$f(\mathbf{q}_1, \dots, \mathbf{q}_n) = \begin{bmatrix} \mathbf{q}_1^T \mathbf{q}_1 - 1 \\ \vdots \\ \mathbf{q}_n^T \mathbf{q}_n - 1 \end{bmatrix} = \mathbf{0}.$$

The potential energy of two magnetic dipoles $\mathbf{q}_i$ and $\mathbf{q}_j$ separated by the vector $\mathbf{r}_{ij}$ is given by [Get08]

$$V(\mathbf{q}_i, \mathbf{q}_j) = \frac{\mu_0 \nu_i \nu_j}{4\pi \|\mathbf{r}_{ij}\|^3} \left( \mathbf{q}_i \cdot \mathbf{q}_j - \frac{3}{\|r_{ij}\|^2} (\mathbf{q}_i \cdot \mathbf{r}_{ij})(\mathbf{q}_j \cdot \mathbf{r}_{ij}) \right),$$

where $\mu_0 = 4\pi \cdot 1^{-7}$ $[\mathrm{N A}^2]$ is the magnetic permeability of the free space. The potential depends on the distance and relative orientation of the dipoles. The case of an array of $n$ dipoles is simply obtained applying the superposition principle:

$$V(\mathbf{q}_1, \dots, \mathbf{q}_n) = \frac{1}{2} \sum_{\substack{i,j=1 \\ j \neq i}}^{n} \frac{\mu_0 \nu_i \nu_j}{4\pi \|\mathbf{r}_{ij}\|^3} \left( \mathbf{q}_i \cdot \mathbf{q}_j - \frac{3}{\|r_{ij}\|^2} (\mathbf{q}_i \cdot \mathbf{r}_{ij})(\mathbf{q}_j \cdot \mathbf{r}_{ij}) \right).$$

Since we suppose that $m_i$ and $l_i$ are respectively the mass and the length of the $i-$th needle, the kinetic energy is given by

$$\begin{aligned} K(\dot{\mathbf{q}}_1, \dots, \dot{\mathbf{q}}_n) &= \frac{1}{2} \sum_{i=1}^{n} \dot{\mathbf{q}}_i \frac{m_i l_i^2}{12} \dot{\mathbf{q}}_i \\ &= \frac{1}{2} \dot{\mathbf{q}} \mathbf{M} \dot{\mathbf{q}}, \end{aligned}$$

where $\dot{\mathbf{q}} = \left[ \dot{\mathbf{q}}_1^T \cdots \dot{\mathbf{q}}_n^T \right]^T$ and

$$\mathbf{M} \overset{\text{def}}{=} \operatorname{diag}(M_1 \mathbf{I}, \dots, M_n \mathbf{I}) \overset{\text{def}}{=} \operatorname{diag}\left( \frac{m_1 l_1^2}{12} \mathbf{I}, \dots, \frac{m_n l_n^2}{12} \mathbf{I} \right).$$

The Lagrangian of the system is hence

$$L(\mathbf{q}_1,\dots,\mathbf{q}_n,\dot{\mathbf{q}}_1,\dots,\dot{\mathbf{q}}_n)=\frac{1}{2}\dot{\mathbf{q}}\mathbf{M}\dot{\mathbf{q}}-\sum_{\substack{i=1\\j>i}}^{n}\frac{\mu_0\nu_i\nu_j}{4\pi\|\mathbf{r}_{ij}\|^3}\left(\mathbf{q}_i\cdot\mathbf{q}_j-\frac{3}{\|r_{ij}\|^2}(\mathbf{q}_i\cdot\mathbf{r}_{ij})(\mathbf{q}_j\cdot\mathbf{r}_{ij})\right);$$

the constrained equations of Euler-Lagrange are

$$\begin{cases}\mathbf{v}_i=\dot{\mathbf{q}}_i,\\ \dot{\mathbf{v}}_i=-\dfrac{1}{M_i}\displaystyle\sum_{\substack{j=1\\j\neq i}}^{n}\frac{\mu_0}{\nu_i\nu_j}4\pi\|\mathbf{r}_{ij}\|^3\left[\mathbf{q}_j-\frac{3}{\|\mathbf{r}_{ij}\|^2}(\mathbf{q}_j\cdot\mathbf{r}_{ij})\mathbf{r}_{ij}\right]-\dfrac{1}{M_i}\mathbf{J}_{f_i}(\mathbf{q}_i)\lambda_i,\\ f(\mathbf{q}_i)=0,\end{cases}$$

for $i=1,\dots,n$.

Equivalently, one can instead consider the Euler-Lagrange equations on the tangent unit sphere, as shown in [LLM09], obtaining for $i=1,\dots,n$

$$\begin{cases}\dot{\mathbf{q}}_i=\boldsymbol{\omega}_i\times\mathbf{q}_i,\\ \boldsymbol{\omega}_i=-\dfrac{1}{M_i}\mathbf{q}_i\times\displaystyle\sum_{\substack{j=1\\j\neq i}}^{n}\frac{\mu_0}{\nu_i\nu_j}4\pi\|\mathbf{r}_{ij}\|^3\left[\mathbf{q}_j-\frac{3}{\|\mathbf{r}_{ij}\|^2}(\mathbf{q}_j\cdot\mathbf{r}_{ij})\mathbf{r}_{ij}\right].\end{cases}\tag{7.8}$$

In the numerical simulation we run, we choose the same initial conditions and parameter values as in [LLM09]: the same mass, length and magnitude of magnetic moment for all dipoles, $m_i=m=0.05$, $l_i=l=0.02$, $\nu_i=\nu=0.1$, and the magnetic needles are placed at vertices of a $4\times 4$ square grid in which the edge of a unit square is $1.2l$. The initial conditions are

$$\mathbf{q}_{i,0}=\begin{bmatrix}1\\0\\0\end{bmatrix},\quad\boldsymbol{\omega}_{i,0}=\begin{bmatrix}0\\0\\0\end{bmatrix},\quad\mathbf{v}_{i,0}=\boldsymbol{\omega}_{i,0}\times\mathbf{q}_{i,0}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

for all dipoles except

$$\mathbf{q}_{16,0}=\begin{bmatrix}0.353508623948073\\0.353608626387662\\-0.866021126843087\end{bmatrix},\quad\boldsymbol{\omega}_{1,0}=\begin{bmatrix}0\\0.5\\0\end{bmatrix},\quad\mathbf{v}_{1,0}=\begin{bmatrix}0\\0\\-0.5\end{bmatrix}.$$

We note that $\nu=0.1$ is a standard magnetic moment value for a rod with good magnetic properties.

The preservation of the total energy of the system is shown in Figure 7.7. Even if none of the methods exactly preserve energy, the ELN method exhibits a very similar energy behavior than RATTLE (whose energy error is bounded over exponentially long times); and RKMKGL4 error is really small.

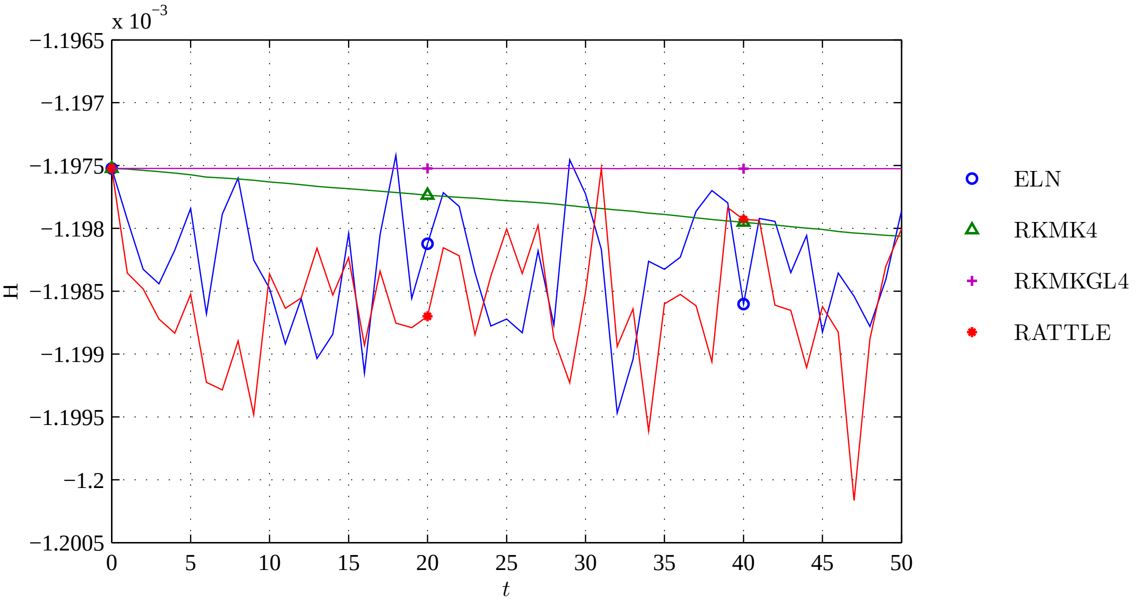


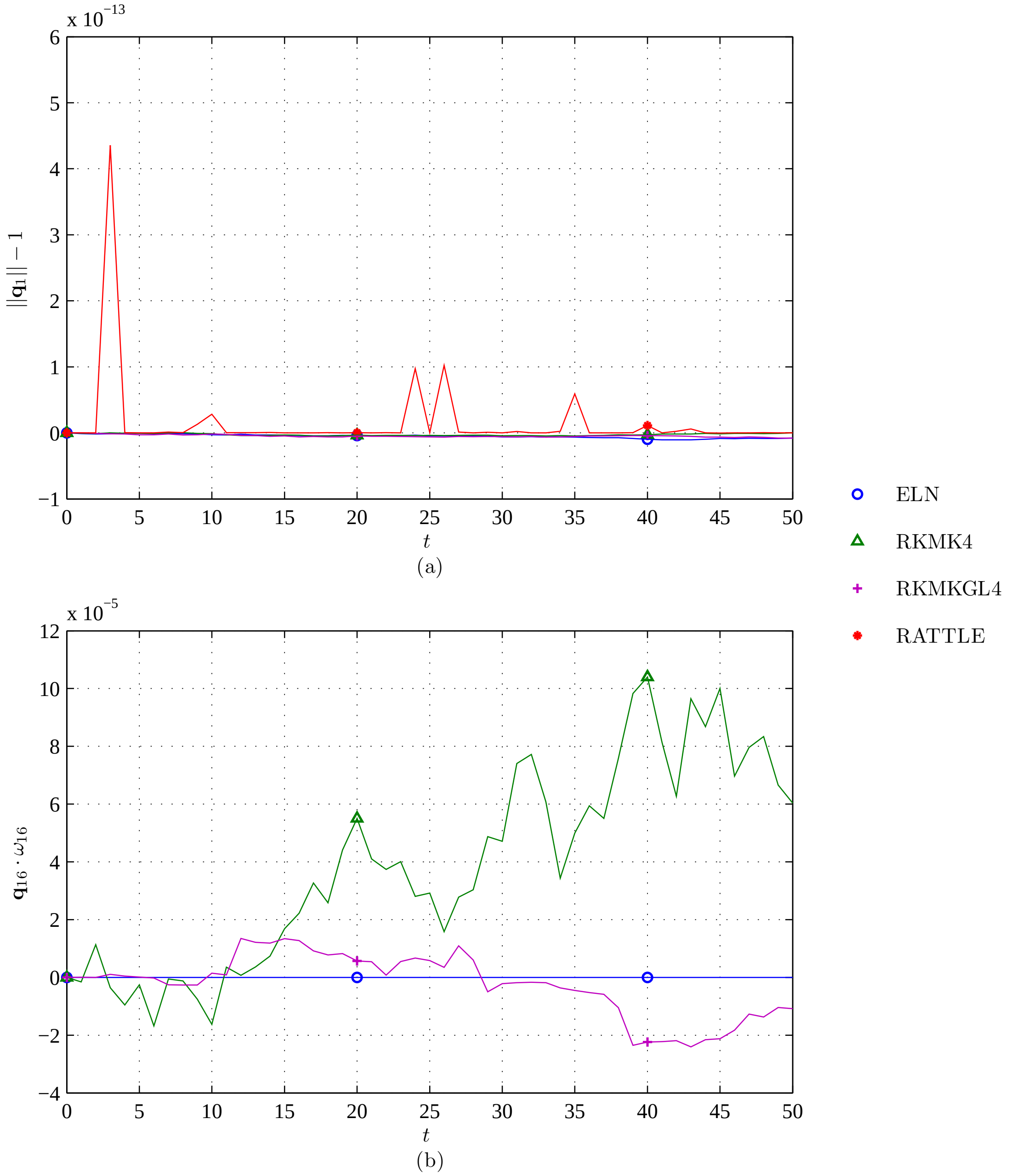
Figure 7.7: Total energy of the array of magnetic dipoles computed with ELN, RKMK4, RKMKGL4, RATTLE integrators. ELN and RATTLE show a very similar oscillatory behavior, while RKMK methods exhibit a linear drift. Initial conditions and parameters used are provided in the text.

In Figure 7.8 the geometric properties of the discrete flows are analyzed. Figure 7.8(a) shows the preservation of the unitary norm of $\|\mathbf{q}_1\|$, while in Figure 7.8(b) is plotted $\mathbf{q}_{16} \cdot \boldsymbol{\omega}_{16}$. All the Lie methods preserve up to machine precision the unitary norm; on the other hand, the orthogonality is not conserved bu RKMK methods: on the other hand, ELN seems to preserve the structure of $T\mathbb{S}^2$.

The run-time required by the methods for the simulation previously discussed is reported in Table 7.3. As in the single spherical pendulum case, the ELN method is faster than RATTLE, while showing an improved accuracy; we remark that in this case ELN is fully explicit. The high computational cost of the RKMKGL4 is due to its implicitness and to its high order.

| Algorithm | Run-time [s] |
|---|---|
| ELN | 118.74 |
| RKMK4 | 289.28 |
| RKMKGL4 | 1785.53 |
| RATTLE | 125.22 |

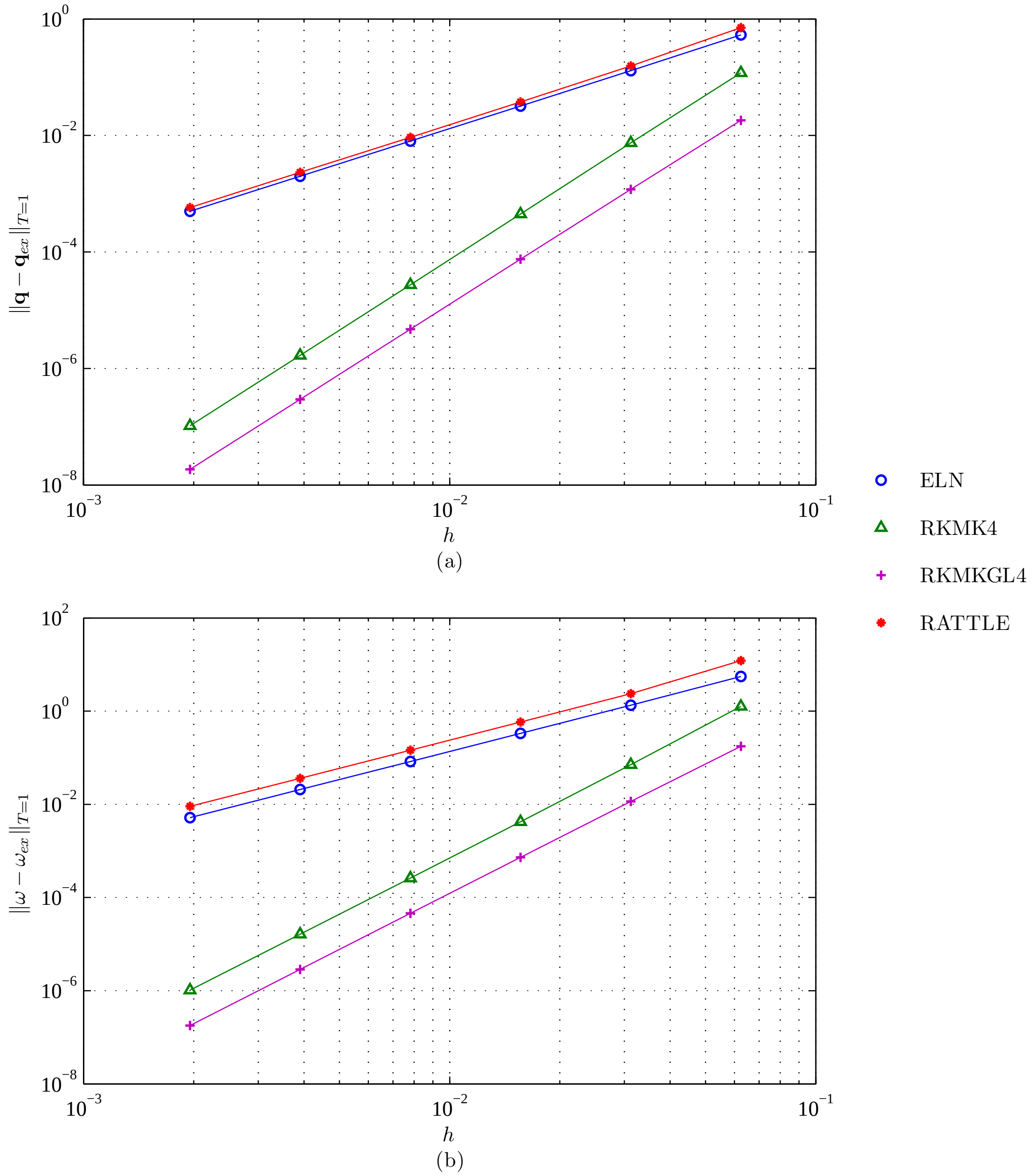
Table 7.3: Mean running time over three simulations for the double spherical pendulum described in the text.



(a)

(b)

Figure 7.8: Preservation of the configuration space properties by ELN, RKMK4, RKMKGL4, RATTLE integrators: (a) position norm for the first dipole $\|\mathbf{q}_1\|$; (b) orthogonality between the position and the angular velocity for the 16-th dipole $\|\mathbf{q}_{16} \cdot \boldsymbol{\omega}_{16}\|$. In (b) RATTLE is not plotted, since it works with the linear velocity. The unitary norm of $\mathbf{q}$ is preserved up to machine precision by Lie group methods, with a better performance than RATTLE. Notably, ELN seems to conserve also the orthogonality between $\mathbf{q}$ and $\boldsymbol{\omega}$, while RKMK methods show a non-negligible error. Initial conditions and parameters used are provided in the text.



(a)

(b)

Figure 7.9: Global error after $T = 1$ on (a) the position $\mathbf{q}$ and (b) the corresponding angular velocity $\boldsymbol{\omega}$ for the case of the array of magnetic dipoles. The error is evaluated for timesteps $h = 2^{i}$, $i = -9, \ldots, -4$. The reference solution is obtained integrating (7.8) using MATLAB function ode45 with low tolerance. The discrete flow on $T\mathbb{S}^2$ generated by ELN is second-order accurate, while the discrete flows generated by RKMK4 and RKMKGL4 are fourth-order accurate. Notably, all the Lie group methods are more accurate than RATTLE. Initial conditions and parameters used are provided in the text.

## 7.3 Conclusions

In Part II of this Thesis a new, straightforward approach to use Lie methods for the rigid body integration to dynamical systems with a separable Hamiltonian whose configuration space is $\left(\mathbb{S}^2\right)^n$ has been described and tested on several relevant cases. The approach is based on the Euler-Lagrange equations written on $\mathbb{S}^2$ using a variational approach [LLM09]. The results have been compared to RATTLE method, a standard choice for constrained integration, which preserves the configuration structure explicitly enforcing the constraints; moreover, RATTLE is symplectic and therefore exhibits excellent geometric properties and long-time performances. It turns out that the performances yielded by a second-order Lie method are better that RATTLE, at an even lower computational cost. Moreover, all the tests shown in Chapter 7 constitute a numerical proof of the fact that this approach preserves the accuracy order of the Lie method also for the integration on $\mathbb{S}^2$. The flow on $\mathbb{S}^2$ can therefore be approximated with an arbitrarily high accuracy just employing, e.g., a high-order Runge-Kutta Munthe-Kaas method, which is very known and well-studied in literature.

We are currently investigating if this approach can be extended to generical Hamiltonian systems on $\mathbb{S}^2$. The basic idea is re-write the dynamics in terms of $\mathbf{q} \in \mathbb{S}^2$ and $\boldsymbol{\omega} \in T\mathbb{S}^2$, $\mathbf{q} \cdot \boldsymbol{\omega} = 0$, using the following fact. The dynamics of a system on the unit sphere can be expressed as

$$\dot{\mathbf{q}} = f(\mathbf{q}), \text{ with } f(\mathbf{q}) \cdot \mathbf{q} = 0. \tag{7.9}$$

It can be easily proved that (7.9) can be equivalent to

$$\dot{\mathbf{q}} = g(\mathbf{q}) \times \mathbf{q}, \text{ with } g(\mathbf{q}) = \mathbf{q} \times f(\mathbf{q}), \; g(\mathbf{q}) \cdot \mathbf{q} = 0.$$

This comes from the Lagrange's formula for the cross product

$$(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{b} \cdot \mathbf{c})\mathbf{a};$$

therefore we have

$$\begin{aligned} g(\mathbf{q}) \times \mathbf{q} &= \left(\mathbf{q} \times f(\mathbf{q})\right) \times \mathbf{q} \\ &= \underbrace{(\mathbf{q} \cdot \mathbf{q})}_{=1} f(\mathbf{q}) - \underbrace{(f(\mathbf{q}) \times \mathbf{q})}_{=0} \mathbf{q} \\ &= f(\mathbf{q}). \end{aligned}$$

We can then consider $g(\mathbf{q})$ as $\boldsymbol{\omega}$, and we can apply our approach for integrating the flow on $\mathbb{S}^2$.

Such a method can be applied, e.g., to relevant problems for numerical integration like the case of $N$ point vortices on the sphere [New01], or micromagnetics applications [LN03]. The former problem is an important issue for Geophysics, and a lot of literature has been devoted to the study of a 2D case. Nevertheless, if we wish to simulate the motion of vortices on the Earth sphere, the curvature effects are no longer negligible and one cannot use a tangent plane approximation anymore. Recently, Ma and Rowley [MR09] includes the example of $N$ point vortices on a sphere to test the performance of their numerical integrators for Lie-Poisson Hamiltonian systems. The latter problem is a hot issue, since magnetic materials are used in a wide range of industrial application and integrating magnetization dynamics can become crucial. The Landau-Lifshitz-Landau model for the state of magnetization in a ferromagnetic sensor has been studied in depth by Lewis and Nigam in [LN03], and a geometric approach - which uses the isotropy algebra in order to gain some accuracy in the method - has been proposed there. Nevertheless, their methods require elaborate computations, and some simulations are provided only in a simplistic case.

# $\mathcal{A}$ Appendix

## A.1 A cross product property

**Proposition A.1.1.** *Be* $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$ *and* $\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^T$ *two vectors in* $\mathbb{R}^3$, *and* $\mathbf{c} = \mathbf{a} \times \mathbf{b}$. *If* $\mathbf{a} \cdot \mathbf{b} = 0$ *and* $||\mathbf{b}|| = 1$, *then* $\mathbf{a} = \mathbf{b} \times \mathbf{c}$.

*Proof.* For hypothesis, we have

$$a_1 b_1 + a_2 b_2 + a_3 b_3 = 0, \tag{A.1}$$

$$b_1^2 + b_2^2 + b_3^2 = 1. \tag{A.2}$$

Let us rewrite **c** in a component-wise manner:

$$\mathbf{c} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}.$$

Straightforward computation yields:

$$\begin{aligned} \mathbf{b} \times \mathbf{c} &= \begin{bmatrix} 0 & -b_3 & b_2 \\ b_3 & 0 & -b_1 \\ -b_2 & b_1 & 0 \end{bmatrix} \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix} \\ &= \begin{bmatrix} a_1(b_2^2 + b_3^2) - b_1(a_2 b_2 + a_3 b_3) \\ a_2(b_1^2 + b_3^2) - b_2(a_1 b_1 + a_3 b_3) \\ a_3(b_1^2 + b_2^2) - b_3(a_1 b_1 + a_2 b_2) \end{bmatrix} \end{aligned}$$

where, using (A.1) and (A.2), one gets

$$\begin{aligned} \mathbf{b} \times \mathbf{c} &= \begin{bmatrix} a_1(1 - b_1^2) - b_1(-a_1 b_1) \\ a_2(1 - b_2^2) - b_2(-a_2 b_2) \\ a_3(1 - b_3^2) - b_3(-a_3 b_3) \end{bmatrix} \\ &= \begin{bmatrix} a_1 - a_1 b_1^2 + a_1 b_1^2 \\ a_2 - a_2 b_2^2 + a_2 b_2^2 \\ a_3 - a_3 b_3^2 + a_3 b_3^2 \end{bmatrix} = \mathbf{a}, \end{aligned}$$

which is what we want to prove. □

## A.2 Inversion of a block matrix

Consider the block matrix $\mathbf{M} \in \mathbb{R}^{(n+m)\times(n+m)}$,

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix},$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{m\times n}$, $\mathbf{D} \in \mathbb{R}^{m\times m}$.

**Proposition A.2.1.** *If* $\mathbf{A}$, $\mathbf{D}$, $(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})$, $(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})$ *are invertible, then* $\mathbf{M}$ *is invertible and*

$$\mathbf{M}^{-1} = \begin{bmatrix} (\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}. \tag{A.3}$$

*Proof.* It is sufficient to show that (A.3) denotes the inverse of $\mathbf{M}$. Direct computation yields

$$\begin{aligned}
\mathbf{M}\mathbf{M}^{-1} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} (\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}+\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ \mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}-\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{C}\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}+\mathbf{D}(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix} \\
&= \begin{bmatrix} (\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & \mathbf{0}_3 \\ \mathbf{0}_3 & (\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} = \mathbf{I}_6.
\end{aligned}$$

The same calculation shows that $\mathbf{M}^{-1}\mathbf{M} = \mathbf{I}_6$. □

**Proposition A.2.2.** *If* $\mathbf{M}$ *and* $\mathbf{A}$ *(or* $\mathbf{D}$*) are invertible, then* $(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ *(or* $(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})$*) is invertible.*

*Proof.* We will prove just one case. Be $\mathbf{v} \in \mathbb{R}^m$ a nonzero vector, and define $\mathbf{u} = -\mathbf{A}^{-1}\mathbf{B}\mathbf{v} \in \mathbb{R}^n$. Since $\mathbf{M}$ is invertible, we have that

$$\mathbf{0} \neq \mathbf{M}\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{u}+\mathbf{B}\mathbf{v} \\ \mathbf{C}\mathbf{u}+\mathbf{D}\mathbf{v} \end{bmatrix} = \begin{bmatrix} -\mathbf{B}\mathbf{v}+\mathbf{B}\mathbf{v} \\ -\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\mathbf{v}+\mathbf{D}\mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ (\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})\mathbf{v} \end{bmatrix}.$$

Since the above formula must hold for a generic nonzero vector $\mathbf{v}$, then $(\mathbf{D}-\mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ is invertible. □

**Corollary A.2.3.** *If* $\mathbf{M}$, $\mathbf{A}$, $\mathbf{D}$ *are invertible, then the inverse of* $\mathbf{M}$ *is given by* (A.3).

# Bibliography

[Act90] Forman S. Acton, *Numerical methods that usually work*, Mathematical Association of America, 1990, Updated and revised from the 1970 edition.

[AKW93] Mark Austin, Perinkulam S. Krishnaprasad, and Li-Sheng Wang, *Almost Poisson Integration of Rigid Body Systems*, Journal of Computational Physics **107** (1993), 105–117.

[And83] Hans C. Andersen, *Rattle: a "velocity" version of the Shake algorithm for molecular dynamics calculations*, Journal of Computational Physics **52** (1983), 24–34.

[Bak02] Andrew Baker, *Matrix Groups: An Introduction to Lie Group Theory*, Springer, 2002.

[BG94] Giancarlo Benettin and Antonio Giorgilli, *On the Hamiltonian Interpolation of Near-to-the-Identity Symplectic Mappings with Application to Symplectic Integration Algorithms*, Journal of Statistical Physics **74** (1994), 1117–1143.

[BR07] Nawaf Bou-Rabee, *Hamilton-Pontryagin on Lie Groups*, Ph.D. thesis, California Institute of Technology, 2007.

[BRM09] Nawaf Bou-Rabee and Jerrold E. Marsden, *Hamilton-Pontryagin Integrators on Lie Groups Part I: Introduction and Structure-Preserving Properties*, Foundations of Computational Mathematics **9** (2009), 197–219.

[CG93] Peter E. Crouch and Robert L. Grossman, *Numerical Integration of Ordinary Differential Equations on Manifolds*, Journal of Nonlinear Science **3** (1993), 1–33.

[CO02] Elena Celledoni and Brynjulf Owren, *A class of intrinsic schemes for orthogonal integration*, SIAM Journal of Numerical Analysis **40** (2002), no. 6, 2069–2084.

[CO03] ______, *On the implementation of Lie group methods on the Stiefel manifold*, Numerical Algorithms **32** (2003), 163–183.

[CS72] Bruce Chartres and Robert Stepleman, *A General Theory of Convergence for Numerical Methods*, SIAM Journal on Numerical Analysis **9** (1972), no. 3, 476–492.

[DdA08] Christine C. Dantas and Luiz de Andrade, *Micromagnetic simulations of small arrays of submicron ferromagnetic particles*, Physical Review **78** (2008), 24441.

[DP80] John R. Dormand and P. J. Prince, *A family of embedded Runge-Kutta formulae*, Journal of Computational and Applied Mathematics **6** (1980), 19–26.

[EM98] Kenth Engø and Arne Marthinsen, *Modeling and Solution of Some Mechanical Problems on Lie Groups*, Multibody System Dynamics **2** (1998), 71–88.

[Eng03] Kenth Engø, *Partitioned Runge-Kutta methods in Lie group setting*, BIT Numerical Mathematics **43** (2003), 21–39.

[FHP04] Erwan Faou, Ernst Hairer, and Truong-Linh Pham, *Energy conservation with non-symplectic methods: examples and counter-examples*, BIT Numerical Mathematics **44** (2004), 699–709.

[Get08] Mathias Getzlaff, *Fundamentals of Magnetism*, Springer, Berlin, 2008.

[Hai08] Ernst Hairer, *Conjugate-symplecticity of linear multistep methods*, Journal of Computational Mechanics **26** (2008), no. 5, 657–659.

[HC87] Thomas J. R. Hughes and Englewood Cliffs, *The finite element method: linear static and dynamic finite element analysis*, Prentice-Hall, 1987.

[HH55] Preston C. Hammer and Jack W. Hollingsworth, *Trapezoidal Methods of Approximating Solutions of Differential Equations*, Mathematical Tables and other Aids for computation **9** (1955), no. 51, 92–96.

[HLW06] Ernst Hairer, Christian Lubich, and Gerhard Wanner, *Geometric Numerical Integration*, 2nd ed., Springer Series in Computational Mathematics, vol. 31, Springer, 2006.

[HMS09] Ernst Hairer, Robert I. McLachlan, and Robert D. Skeel, *On energy conservation of the simplified takahashi-imada method*, ESAIM> Mathematical Modeling and Numerical Analysis **43** (2009), 631–644.

[Hun74] Thomas W. Hungerford, *Algebra*, Springer, 1974.

[IMKNZ00] Arieh Iserles, Hans Munthe-Kaas, Syvert P. N/orsett, and Antonella Zanna, *Lie-group methods*, Acta Numerica **9** (2000), 215–365.

[Jay96] Laurent Jay, *Symplectic partitioned Runge-Kutta methods for constrained Hamiltonian systems*, SIAM Journal on Numerical Analysis **33** (1996), no. 1, 368–387.

[KB10] Tomasz Koziara and Nenad Bićanić, *Simple and efficient integration of rigid rotations suitable for constraint solvers*, International Journal for Numerical Methods in Engineering **81** (2010), 1073–1092.

[KE05] Petr Krysl and Lance Endres, *Explicit Newmark/Verlet algorithm for time integration of the rotational dynamics of rigid bodies*, International Journal for Numerical Methods in Engineering **62** (2005), 2154–2177.

[KMOW00] Couro Kane, Jerrold E. Marsden, Michael Ortiz, and Matthew West, *Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems*, International Journal for Numerical Methods in Engineering **49** (2000), 1295–1325.

[KN63] Shoshichi Kobayashi and Katsumi Nomizu, *Foundations of Differential Geometry*, Interscience Publishers, 1963.

[Kry05] Petr Krysl, *Explicit momentum-conserving integrator for dynamics of rigid bodies approximating the midpoint Lie algorithm*, International Journal for Numerical Methods in Engineering **63** (2005), 2171–2193.

[Kry08] ______, *Dynamically equivalent implicit algorithms for the integration of rigid body rotations*, International Journal for Numerical Methods in Engineering **24** (2008), 141–156.

[Leo04] Melvin Leok, *Foundations of Computational Geometric Mechanics*, Ph.D. thesis, California Institute of Technology, 2004.

[LLM09] Taeyoung Lee, Melvin Leok, and N. Harris McClamroch, *Lagrangian mechanics and variational integrators on two-spheres*, International Journal for Numerical Methods in Engineering **79** (2009), 1147–1174.

[LN03] Debra Lewis and Nilima Nigam, *Geometric integration on spheres and some interesting applications*, Journal of Computational and Applied Mathematics **151** (2003), 141–170.

[LO02] Debra Lewis and Peter J. Olver, *Geometric Integration Algorithms on Homogeneous Manifolds*, Foundations of Computational Mathematics **2** (2002), no. 4, 363–392.

[LR04] Benedict Leimkuhler and Sebastian Reich, *Simulating Hamiltonian Dynamics*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2004.

[LS94a] Benedict Leimkuhler and Robert D. Skeel, *Symplectic numerical integratiors in constrained Hamiltonian systems*, Journal of Computational Physics **112** (1994), 117–125.

[LS94b] Debra Lewis and Juan Carlos Simo, *Conserving algorithms for the dynamics of Hamiltonian systems on Lie groups*, Journal of Nonlinear Science **4** (1994), 253–299.

[MK95] Hans Munthe-Kaas, *Lie-Butcher theory forRunge-Kutta methods*, BIT Numerical Mathematics **35** (1995), 572–587.

[MK98] ______, *Runge-Kutta methods on Lie groups*, BIT Numerical Mathematics **38** (1998), 92–111.

[MK99] ______, *High order Runge-Kutta methods on manifolds*, Applied Numerical Mathematics **29** (1999), 115–127.

[MKZ97] Hans Munthe-Kaas and Antonella Zanna, *Numerical Integration of Differential Equations on Homogeneous Manifolds,* Selected papers of a conference on Foundations of Computational Mathematics (New York, NY, USA), Springer-Verlag New York, Inc., 1997, pp. 305–315.

[MLS94] Richard M. Murray, Zexiang Li, and S. Shankar Sastry, *A mathematical introduction to robotic manipulation*, CRC Press, 1994.

[MNS74] Matti Mäkela, Olavi Nevanlinna, and Aarne H. Sipilä, *On the Concepts of Convergence, Consistency and Stability in Connection with some Numerical Methods,* Numerische Mathematik **22** (1974), 261–274.

[MPS99] Jerrold E. Marsden, Sergej Pekarsky, and Steve Shkoller, *Discrete Euler-Poincarè and Lie-Poisson equations,* Nonlinearity **12** (1999), 1647–1662.

[MR99] Jerrold E. Marsden and Tudor Ratiu, *Introduction to Mechanics and Symmetry,* Springer, 1999.

[MR09] Zhanhua Ma and Clarence W. Rowley, *Lie-Poisson integrators: A Hamiltonian, variational approach*, International Journal for Numerical Methods in Engineering **82** (2009), no. 13, 1609–1644.

[MS93] J. E. Marsden and J. Scheurle, *Lagrangian reduction and the double spherical pendulum,* Zeitschrift für Angewandte Mathematik und Physik **44** (1993), 17–43.

[MS95] Robert McLachlan and Clint Scovel, *Equivariant Constrained Symplectic Integration,* Journal of Nonlinear Science **5** (1995), 223–256.

[MV91] Jürgen Moser and Alexander P. Veselov, *Discrete Versions of Some Classical Integrable Systems and Factorization of Matrix Polynomials*, Communications in Mathematical Physics **139** (1991), 217–243.

[MW01] Jerrold E. Marsden and Matthew West, *Discrete mechanics and variational integrators*, Acta Numerica **10** (2001), 357–524.

[MZ05] Robert I. McLachlan and Antonella Zanna, *The Discrete Moser-Veselov Algorithm for the Free Rigid Body, Revisited*, Foundations of Computational Mathematics **5** (2005), no. 1, 87–123.

[New59] Nathan A. Newmark, *A method of computation for structural dynamics*, Journal of the Engineering Mechanics Division **EM3** (1959), 67–94.

[New01] Paul K. Newton, *The N-vortex problem: analytical techniques*, Applied Mathematical Sciences, Springer, 2001.

[NJ07] Phani Kumar N. N. Nukala and William Shelton Jr., *Semi-implicit reversible algorithms for rigid body rotational dynamics*, International Journal for Numerical Methods in Engineering **69** (2007), 2636–2662.

[OM99] Brynjulf Owren and Arne Marthinsen, *Runge-Kutta methods adapted to manifolds and based on rigid frames*, BIT Numerical Mathematics **39** (1999), no. 1, 116–142.

[Rei94] Sebastian Reich, *Momentum sconserving symplectic integrators*, Physica D: Nonlinear Phenomena **76** (1994), 375–383.

[Rei99] ______, *Backward Error Analysis for Numerical Integrators*, SIAM Journal on Numerical Analysis **36** (1999), no. 5, 1549–1570.

[Sac09] Alessandro Saccon, *Midpoint rule for variational integrators on Lie groups*, International Journal for Numerical Methods in Engineering **78** (2009), 1345–1364.

[Shu93] Malcolm D. Shuster, *Survey of Attitude Representations*, Journal of Astronautical Sciences **41** (1993), 439–517.

[SVQ88] Juan Carlos Simo and Loc Vu-Quoc, *On the dynamics in space of rods undergoing large motions – A geometrically exact approach*, Computer Methods in Applied Mechanics and Engineering **66** (1988), 125–161.

[SW91] Juan Carlos Simo and Kachung Kevin Wong, *Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum*, International Journal for Numerical Methods in Engineering **31** (1991), 19–52.

[Ves98] Alexander P. Veselov, *Integrable discrete-time systems and difference operators*, Translated from Funktsional’nyi Analiz i Ego Prilozheniya **22** (1998), 1–13.

# Acknowledgements

First of all, I would like to thank sincerely my advisor, Alessandro Beghi, for all his advices, his presence and his guidance during all my time as a graduate student. Alessandro is an excellent teacher, a brilliant scientist and a very educated and witty man. I am heartily grateful to him for the support he provided me during all these years, and for the help he offered and is offering me to find work opportunities.

I wish to acknowledge Alessandro Saccon, with whom I am working closely since my Master Degree. Despite our - sometimes - hard discussion and fights, I really hope he appreciates the results we got to with our research project.

During my PhD, I spent four months at the Department of Mathematics of the University of California San Diego, and I had the great opportunity and pleasure to know and work side-by-side with Melvin Leok. Melvin has always been helpful and amenable to me, and he did all his best to assure my stay was comfortable. Moreover, Melvin is a great scientist and an outstanding communicator: a short discussion with him, one of his talks have always been more useful and enlightening than several hours spent on books and papers.

Even if the research I did with him does not compare in this Thesis, I would also take the opportunity to thank Angelo Cenedese. Angelo is modest, brilliant and hard-worker, and the time I spent working with him has been really valuable and extremely enjoyable.

Of course, my PhD years wouldn't have been so pleasant without all the officemates I met. Thanks to my friends Saverio Bolognani, Giulio Bottegal, Mattia Bruschetta, Simone Del Favero, Fabio Maran, Andrea Masiero, Maura Pasquotti, Mirco Rampazzo, Lucia Seno, Gian Antonio Susto, Damiano Varagnolo, Filippo Zanella: thanks not only for all the jokes, the fun, and the great parties but also for the affection and understanding you all displayed to me anytime I had some hard times.

Finally, I wish to thank some people outside the academic world: in the first place, my mother Marilena and my grandmother Edvige. They have been and always will be an example for me, and I owe to them my strength, my constancy, my tenacity; and, mostly important, I know that they will always be there for me. I will do my best to be there for them, too.

I want to say “thank you” to Chiara (aka Talpa), for all our evenings, our nights, our trips, our intercontinental phone calls. Thank you for all the delicious meals you cooked for me, thank you for being near to me every time I need support.

Last but not least, I want to thank a very special person who has been really near to me in these months: Elisa. Firstly, you made my passion for playing violin revive again, and I hope our common music project will have a long and healthy life. Then, in spite of all the troubles you faced during this time, you have always been able to stand and support me while I was writing this Thesis, with patience and understanding. Without you everything would have been much more harder.