NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Exact computation of the solution of the free rigid body with applications

by

E. Celledoni, F. Fassò, N. Säfstöm, A. Zanna

PREPRINT
NUMERICS NO. 6/2007

NORWEGIAN UNIVERSITY OF
SCIENCE AND TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/numerics/2007/N6-2007.pdf
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Exact computation of the solution of the free rigid body with applications

E. Celledoni, F. Fassò, N. Säfstöm, A. Zanna

September 8, 2007

We discuss techniques for the direct, *exact* computation of the solution of the equations of motion of a Free Rigid Body. The method relies on the efficient and accurate computation of Jacobi ellipic functions and integrals. Here *exact* means that the approximation produces an error of the size of machine accuracy.

We show how these techniques combined with splitting methods can be profitably applied to the approximation of systems of torqued rigid bodies. The performance of the proposed strategy is illustrated in applications to molecular dynamics and satellite dynamics.

We discuss techniques for the direct, *exact* computation of the solution of the equations of motion of a Free Rigid Body. The method relies on the efficient and accurate computation of Jacobi ellipic functions and integrals. Here *exact* means that the approximation produces an error of the size of machine accuracy.

We show how these techniques combined with splitting methods can be profitably applied to the approximation of systems of torqued rigid bodies. The performance of the proposed strategy is illustrated in applications to molecular dynamics and satellite dynamics.

1 Introduction

We discuss techniques for the direct, *exact* computation of the solution of the equations of motion of a Free Rigid Body. The method relies on the efficient and accurate computation of Jacobi ellipic functions and integrals. Here *exact* means that the approximation produces an error of the size of machine accuracy.

The rigid body equations are *integrable*, meaning that their exact solution can be found in terms of known mathematical functions (elliptic functions). One of the earliest solutions dates back to Jacobi [13], although he attributed to Legendre the solution of the problem in terms of elliptic integrals of first and third type, [13]. As this is one of the most classical problems in mechanics, several other representations for the solution have been found (see for instance [25], [2] and [19]).

We are interested in accurate numerical simulations with rigid bodies, therefore we wish to single out algorithms that are amenable for numerical computations. By *accurate* we mean that the methods not only should produce a small global numerical error, but also they should exactly preserve as many as possible of the underlying geometrical properties of the system, as the kinetic energy, the Lie-Poisson structure, the time-symmetry of the flow, the Casimirs.

In [5] it was shown that methods solving exactly the angular momentum were numerically competitive and had good geometrical properties. In this paper we intend to explore whether the same is true for the attitude rotation. The equations for the attitude matrix are more complicated. The classical approach is to introduce Euler's angles, [2], [13]. As in many simulations it is preferable to work in Cartesian coordinates or in quaternions, we have investigated approaches that avoid spherical coordinates all together.

For the case of Cartesian coordinates, we propose a method that can be summarised as follows: it rotates one of the axes of the attitude matrix on the angular momentum, so that the other two axes lie on the *invariant plane*; then, the position of the two latter axes is parametrised by an angle with respect to a known (moving) reference frame; finally, the angle parameter is computed by evaluating an elliptic integral of the third kind, which is the most demanding part, from a computational point of view. After having derived the method, we have discovered similarities with an approach described in [6, 7]. What is particular for our approach, is the way the known moving reference frame is chosen.

The quaternion formulation is built on the same philosophy, we base our method on a method proposed by [14], which we reformulate and generalise. We obtain an algorithm which, for different choices of the parameters, gives a quaternion implementation of different expressions for the exact solution, some of which can be found in the classical mechanics literature. In all cases the accurate computation of an incomplete elliptic integral of the third kind is required.

To compute this integral we consider here two strategies, the well known method of Carlson [20], and Gaussian quadrature. The latter allows us to reduce the computational cost by 2/3.

Earlier work on accurate approximation of the solution of the free rigid body equations can be found in [16], and more recently in [12]. In this approach the classical discrete Moser-Veselov algorithm has been turned into a high order method using techniques of backward error analysis. As it turns out the exact solution can be tabulated at discrete time by applying the discrete Moser-Veselov algorithm to a modified free rigid body equation. The difference between the original free rigid body equation and the modified one is that in the latter the principal moments of inertia are given by an expansion in powers of the step-size h. Including more and more terms in this expansion improves the order of the methods at a very moderate increase in computational cost.

In this paper we perform comparisons of the implementation of the exact solution with these approaches. Following [11], we consider a range of different moments of inertia, (I_1, I_2, I_3), and investigate how the different methods perform for different choices. We obtain that all the methods perform well for oblate symmetric bodies and less well for very flat symmetric rigid bodies (with the first inertia moment close to zero). The *exact* methods presented in this paper perform well independently of the choice of the inertia moments. In particular our methods, perform better, compared to others, when using large step-sizes. Our conclusion is that the implementation of the exact solution of the free rigid body is in general a competitive approach compared to other numerical methods.

The numerical exact solution of the free rigid body equations is of interest as it can be used as a building block for splitting methods of high order. We show how the proposed techniques combined with splitting methods can be profitably applied to the approximation of systems of torqued rigid bodies. The performance of the proposed strategy is illustrated in applications to molecular dynamics and satellite dynamics.

The paper is organized as follows. In section two we review the exact integration of the angular momentum equation and present our approaches for the exact integration of the attitude matrix, using rotations and quaternions. We give a unifying overview of the relation between the two. Section three is devoted to extensive numerical experiments. Finally section four is devoted to some concluding remarks.

2 Accurate integration methods for the full dynamics of the free rigid body

To describe the motion of a free rigid body (FRB) we consider the Euler's equations of motion, [9], [10]. Consider a coordinate system fixed in space $O\xi_1\xi_2\xi_3$, with orthonormal coordinate frame $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ and a moving coordinate system $Ox_1x_2x_3$ with origin in the center of mass of the body (body coordinate frame). For the sake of simplicity we assume $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$ are the three canonical vectors in $\mathbb{R}^3$.

Assuming that the axes of the body frame, $Ox_1x_2x_3$, coincide with the principal axes of inertia, the equations of motion for the FRB are

$$\dot{\boldsymbol{m}} = \boldsymbol{m} \times I^{-1}\boldsymbol{m}, \tag{1}$$

$$\dot{Q} = Q\,\widehat{I^{-1}\boldsymbol{m}}, \tag{2}$$

with $\boldsymbol{m}(t_0) = \boldsymbol{m}_0$, and $Q(t_0) = \mathbb{1}$ the 3×3 identity matrix. Here we assume $I = \mathrm{diag}(I_1, I_2, I_3)$ is the inertia tensor, $\boldsymbol{m} = (m_1, m_2, m_3)^T \in \mathbb{R}^3$ is the angular momentum in the body frame, and $Q \in SO(3)$ is the attitude matrix describing the configuration of the body. The matrix Q transforms any vector in body coordinates into the corresponding vector in space coordinates. The columns of this matrix are an orthonormal basis for the body coordinate system. For ease of notation in (2) we have used the hat-map, $\widehat{\ }: \mathbb{R}^3 \to \mathfrak{so}(3)$. This map associates to any vector $\boldsymbol{v}$ a skew-symmetric matrix $\widehat{\boldsymbol{v}}$ as follows,

$$\boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}, \quad \widehat{\boldsymbol{v}} = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix},$$

and is such that $\widehat{\boldsymbol{v}}\boldsymbol{u} = \boldsymbol{v} \times \boldsymbol{u}$.

Equations (1) and (2) are completely integrable. Using (1) and (2) it is easy to show that the angular momentum is constant in the coordinate system fixed in space, i.e.

$$\boldsymbol{m}_0 = Q(t)\boldsymbol{m}(t).$$

The kinetic energy T, and the length of the angular momentum G are invariants of the the solution and have the following expressions in terms of the components of the solution

$$T = \frac{1}{2}\left(\frac{m_1^2}{I_1} + \frac{m_2^2}{I_2} + \frac{m_3^2}{I_3}\right), \quad G^2 = \|\boldsymbol{m}\|^2 = m_1^2 + m_2^2 + m_3^2. \tag{3}$$

Without loss of generality we will assume $G^2 = 1$ as other norms for $\boldsymbol{m}$ simply correspond to a time scaling, $t \to \|\boldsymbol{m}_0\|\, t$.

Jacobi derived the solution of the equations (1) and (2) in terms of Jacobi elliptic functions in 1849, [13]. In the sequel for the sake of completeness we recall the formulae for the angular momentum.

2.1 The solution of the equations for the momentum

Assume that the principal moments of inertia are in ascending order, $I_1 < I_2 < I_3$, and that $I = \mathrm{diag}(I_1, I_2, I_3)$ is the inertia tensor. Consider the constants

$$\begin{aligned} a_1^2 &= 2TI_3 - 1, & a_3^2 &= 1 - 2TI_1, \\ b_1^2 &= I_2(I_3 - I_2), & b_3^2 &= I_2(I_2 - I_1). \end{aligned} \tag{4}$$

In the case $b_3/a_3 = b_1/a_1$, when the trajectories of $\boldsymbol{m}$ coincide with the "separatrices", the equation (1) becomes easy to integrate and the solution can be written as

$$m_1(t) = \frac{a_1\sqrt{I_1(1-x^2)}}{\sqrt{I_3 - I_1}}, \quad m_2(t) = \frac{a_1 I_2}{b_1} x, \quad m_3(t) = \frac{a_3\sqrt{I_3(1-x^2)}}{\sqrt{I_3 - I_1}},$$

where

$$x = \frac{\mathrm{e}^{2u} - L}{\mathrm{e}^{2u} + L},$$

and

$$L = \frac{1 - x_0}{1 + x_0}, \quad u = \frac{b_1 a_3}{I_2\sqrt{I_1 I_3}}(t - t_0).$$

In the general case the solution of (1) can be written as follows. We distinguish two cases,

case(a) if $b_3/a_3 > b_1/a_1$

$$\begin{gathered} m_1 = \frac{\pm a_1\sqrt{I_1}\,\mathrm{dn}\ u}{\sqrt{I_3 - I_1}}, \quad m_2 = \frac{a_3 I_2\,\mathrm{sn}\ u}{b_3}, \quad m_3 = \frac{a_3\sqrt{I_3}\,\mathrm{cn}\ u}{\sqrt{I_3 - I_1}}, \\ \lambda = \pm a_1 b_3/(I_2\sqrt{I_1 I_3}), \quad k = b_1 a_3/(b_3 a_1), \quad u = \lambda(t - \nu), \end{gathered} \tag{5}$$

case(b) or if $b_3/a_3 < b_1/a_1$

$$\begin{gathered} m_1 = \frac{a_1\sqrt{I_1}\,\mathrm{cn}\ u}{\sqrt{I_3 - I_1}}, \quad m_2 = \frac{a_1 I_2\,\mathrm{sn}\ u}{b_1}, \quad m_3 = \frac{\pm a_3\sqrt{I_3}\,\mathrm{dn}\ u}{\sqrt{I_3 - I_1}}, \\ \lambda = \pm b_1 a_3/(I_2\sqrt{I_1 I_3}), \quad k = b_3 a_1/(b_1 a_3), \quad u = \lambda(t - \nu). \end{gathered} \tag{6}$$

In both cases $0 < k < 1$. Here, cn, sn and dn are the Jacobi elliptic functions defined by

$$\mathrm{cn}\ u = \cos\varphi, \quad \mathrm{sn}\ u = \sin\varphi, \quad \mathrm{dn}\ u = \sqrt{1 - k^2\sin^2\varphi}, \tag{7}$$

with $u(t) = \lambda(t-\nu)$, and ν a constant of integration depending on the initial condition. The amplitude φ is given implicitly as the solution of the equation

$$\int_0^{\varphi} \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} = u(t), \quad u(t) = \lambda(t-\nu) \tag{8}$$

where the left hand side is the Legendre elliptic integral of the first kind with modulus k. We will use the notation

$$\varphi = \mathrm{am}(u).$$

Given $u(t)$ and k, the Jacobi Elliptic functions cn, sn and dn can be efficiently calculated to machine precision by the method of ascending/descending Landen transformation, see for example [1].

2.2 Integration of the attitude rotation: the matrix case

In what follows we consider case (a), (5), when $b_3/a_3 > b_1/a_1$. The procedure is identical in case (b), (6), when $b_3/a_3 < b_1/a_1$, except for the variable $\xi_2 = \frac{a_1}{b_1} I_2$.

The following proposition gives an an expression for the attitude rotation matrix as a product of three rotations. The rotation $P(t_0)^T$ maps the axis $\boldsymbol{e}_3$ to the angular momentum and by virtue of (2) this axis coincides with $\boldsymbol{m}$ for all t, see (9). Thereafter, the other two axes of the frame lie on the invariant plane, and their position is determined by planar rotation $Y(t)$.

Proposition 2.1. *Assume $P(t) \in SO(3)$ is a rotation such that*

$$P(t)\boldsymbol{m} = \boldsymbol{e}_3. \tag{9}$$

If Q satisfies (2), then

$$Q(t) = P(t_0)^T Y P,$$

where

$$Y\boldsymbol{e}_3 = \boldsymbol{e}_3, \tag{10}$$

is a rotation around $\boldsymbol{e}_3$ with rotation angle ψ given by

$$\psi = 2T\,(t-t_0) + \int_{t_0}^{t} \dot{\boldsymbol{v}}(s)^T \boldsymbol{w}(s) ds \tag{11}$$

with $\boldsymbol{v} := P^T\boldsymbol{e}_1$, $\boldsymbol{w} := P^T\boldsymbol{e}_2$ the first two columns of P^T.

Proof. Equation (10) can be verified by direct calculation using the hypotheses (9). Differentiating the expression $Y = P(t_0)QP(t)^T$, under the assumption that Q satisfies (2), one obtains the following differential equation

$$\dot{Y} = Y(P\widehat{\boldsymbol{\omega}}P^T - \dot{P}P^T), \text{ and } Y(t_0) = P(t_0)^T Q(t_0) P(t_0) = \mathbb{1}. \tag{12}$$

The property (10) implies that Y is a planar rotation of the type

$$Y = \begin{pmatrix} c(t) & -s(t) & 0 \\ s(t) & c(t) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{with} \quad \begin{array}{rcl} c(t) & = & \cos(\psi(t)), \\ s(t) & = & \sin(\psi(t)), \end{array}$$

with $\psi(t)$ the rotation angle, and $\psi(t_0)=0$. This also implies that Y satisfies a differential equation of the type

$$\dot{Y}=YS, \quad \text{with} \quad S=\begin{pmatrix} 0 & -\gamma(t) & 0 \\ \gamma(t) & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{13}$$

and $\dot{\psi}=\gamma$. Comparing (13) with (12) we obtain that $S=(P\widehat{\boldsymbol{\omega}}P^T-\dot{P}P^T)$ and $\gamma=\boldsymbol{e}_2^T(P\widehat{\boldsymbol{\omega}}P^T-\dot{P}P^T)\boldsymbol{e}_1$, where $\boldsymbol{\omega}:=I^{-1}\boldsymbol{m}$ is the body angular velocity. Now since $\boldsymbol{\omega}$ and $P(t)$ are known functions of t, the problem reduces to a pure quadrature. Setting $\boldsymbol{w}:=P^T\boldsymbol{e}_2$, $\boldsymbol{v}:=P^T\boldsymbol{e}_1$ one obtains

$$\psi(t)=\int_{t_0}^{t}(\boldsymbol{w}(s)^T(\boldsymbol{\omega}(s)\times\boldsymbol{v}(s))-\dot{\boldsymbol{w}}(s)^T\boldsymbol{v}(s))ds,$$

and using $\boldsymbol{w}^T(\boldsymbol{\omega}\times\boldsymbol{v})=\boldsymbol{m}^T\boldsymbol{\omega}=2T$, and the derivative of $\boldsymbol{w}^T\boldsymbol{v}=0$, one arrives at (11). □

We will in what follows discuss in detail what happens for a particular choice of P.

Proposition 2.2. *Assume P satisfies (9) and is such that*

$$\boldsymbol{v}=\frac{\dot{\boldsymbol{m}}}{\|\dot{\boldsymbol{m}}\|}, \quad \boldsymbol{w}=\frac{\boldsymbol{m}\times\dot{\boldsymbol{m}}}{\|\dot{\boldsymbol{m}}\|}, \quad P^T=[\boldsymbol{v},\boldsymbol{w},\boldsymbol{m}].$$

Then

$$\psi(t)=2T\,(t-t_0)+\frac{(1-2TI_2)}{\lambda I_2}(\Pi(\varphi,n,k)-\Pi(\varphi_0,n,k)), \tag{14}$$

where $\varphi=\operatorname{am}(\lambda(t-\nu))$, $\varphi_0=\operatorname{am}(\lambda(t_0-\nu))$ and

$$\Pi(\phi,n,k):=\int_0^u\frac{ds}{1-n\operatorname{sn}^2 s}, \quad -\infty<n<\infty, \quad \phi=\operatorname{am}(u),$$

is the elliptic integral of the third kind with

$$n=\frac{\xi_2^2(I_1-I_2)(I_3-I_2)}{I_2^2(1-2TI_1)(1-2TI_3)},$$

with $\xi_2=a_3I_2/b_3$, and λ, k defined in (5).

Proof. We begin observing that

$$\dot{\boldsymbol{v}}=\frac{\ddot{\boldsymbol{m}}\|\dot{\boldsymbol{m}}\|-2(\ddot{\boldsymbol{m}}^T\dot{\boldsymbol{m}})/\|\dot{\boldsymbol{m}}\|^{3/2}\dot{\boldsymbol{m}}}{\|\dot{\boldsymbol{m}}\|^2},$$

hence $\boldsymbol{w}^T\dot{\boldsymbol{v}}=\boldsymbol{w}^T\ddot{\boldsymbol{m}}/\|\dot{\boldsymbol{m}}\|$. Expanding

$$\ddot{\boldsymbol{m}}=-I^{-1}\dot{\boldsymbol{m}}\times\boldsymbol{m}-I^{-1}\boldsymbol{m}\times\dot{\boldsymbol{m}},$$

substituting in (11), we obtain that

$$\psi(t)=2T(t-t_0)-\int_{t_0}^{t}\left(2T+\boldsymbol{w}(s)^T(I^{-1}\boldsymbol{v}(s)\times\boldsymbol{m}(s))\right)ds.$$

This expression can be further simplified by decomposing $I^{-1}\boldsymbol{v}$ in the directions of the orthogonal system $\boldsymbol{m}$, $\boldsymbol{v}$, $\boldsymbol{w}$,

$$I^{-1}\boldsymbol{v} = \langle I^{-1}\boldsymbol{v}, \boldsymbol{m}\rangle \boldsymbol{m} + \langle I^{-1}\boldsymbol{v}, \boldsymbol{v}\rangle \boldsymbol{v} + \langle I^{-1}\boldsymbol{v}, \boldsymbol{w}\rangle \boldsymbol{w},$$

where we have used the notation $\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \boldsymbol{x}^T\boldsymbol{y}$ for the scalar product. Since

$$\boldsymbol{m}\times\boldsymbol{v} = \boldsymbol{w}, \quad \boldsymbol{w}\times\boldsymbol{m} = \boldsymbol{v}, \quad \boldsymbol{v}\times\boldsymbol{w} = \boldsymbol{m},$$

we have that

$$I^{-1}\boldsymbol{v}\times\boldsymbol{m} = -\langle I^{-1}\boldsymbol{v}, \boldsymbol{v}\rangle \boldsymbol{w} + \langle I^{-1}\boldsymbol{v}, \boldsymbol{w}\rangle \boldsymbol{v}.$$

Then we arrive to the simple expression

$$\psi(t) = \int_{t_0}^{t} \langle I^{-1}\boldsymbol{v}(s), \boldsymbol{v}(s)\rangle ds.$$

Using the equation for the momentum (1) we obtain that

$$\langle I^{-1}\boldsymbol{v}(s), \boldsymbol{v}(s)\rangle = \frac{\frac{A}{I_1}m_2^2m_3^2 + \frac{B}{I_2}m_1^2m_3^2 + \frac{C}{I_3}m_1^2m_2^2}{Am_2^2m_3^2 + Bm_1^2m_3^2 + Cm_1^2m_2^2},$$

where

$$A = a^2, \quad B = b^2, \quad C = c^2,$$

and

$$a = \left(\frac{1}{I_3} - \frac{1}{I_2}\right), \quad b = \left(\frac{1}{I_1} - \frac{1}{I_3}\right), \quad c = \left(\frac{1}{I_2} - \frac{1}{I_1}\right). \tag{15}$$

Next we eliminate the terms m_1^2 and m_3^2 using the integrals of motion (3) so that

$$\langle I^{-1}\boldsymbol{v}(s), \boldsymbol{v}(s)\rangle = 2T + \frac{(1-2TI_2)}{I_2}\frac{1}{1-n\,\mathrm{sn}^2(u(t))},$$

which, integrated between t_0 and t, gives the desired result. □

2.3 Integration of the attitude rotation: the quaternion case

For reasons of storage and computational complexity it might be convenient to represent rotations using quaternions of length 1 (Euler parameters) instead of rotation matrices. In this section we derive an algorithm formulated in quaternions for the implementation of the exact solution of the attitude rotation of the free rigid body problem. We obtain a general framework and we show how different approaches known in the literature fit within this framework.

The quaternions of length 1 are the elements of the unit sphere of $\mathbb{R}^4$,

$$\mathrm{S}^3 = \{q = (q_0, \boldsymbol{q}) \in \mathbb{R}\times\mathbb{R}^3\,;\, q_0^2 + \|\boldsymbol{q}\|^2 = 1\}.$$

This sphere S^3 has the structure of Lie group with the product

$$p\cdot q := (p_0q_0 - \boldsymbol{p}^T\boldsymbol{q}, p_0\boldsymbol{q} + q_0\boldsymbol{p} + \boldsymbol{p}\times\boldsymbol{q}),$$

and $p = (p_0, \boldsymbol{p})$, $q = (q_0, \boldsymbol{q}) \in \mathrm{S}^3$. The identity element is $e = (1, \boldsymbol{0})$. The inverse of a quaternion is defined by

$$q^{-1} = (q_0, -\boldsymbol{q}).$$

The product extends to $\mathbb{R}^4$ and can also be expressed by means of a matrix-vector product, namely, $p \cdot q = L_p q = R_q p$, where

$$L_p = \begin{bmatrix} p_0 & -\boldsymbol{p}^T \\ \boldsymbol{p} & (p_0 \mathbb{1} + \widehat{\boldsymbol{p}}) \end{bmatrix}, \quad R_q = \begin{bmatrix} q_0 & -\boldsymbol{q}^T \\ \boldsymbol{q} & (q_0 \mathbb{1} - \widehat{\boldsymbol{q}}) \end{bmatrix}, \tag{16}$$

and $\mathbb{1}$ is the 3×3 identity matrix. The Lie algebra of S^3, $\mathrm{s}^3 = \mathrm{T}_e\mathrm{S}^3$, can be identified with $\mathbb{R}^3$, equipped with the cross product as commutator. Let us denote with $vec : \mathbb{R}^4 \to \mathbb{R}^3$ the projection corresponding to discarding the first component.

Consider the inner automorphism of S^3, $I_q(p) = q \cdot p \cdot q^{-1}$, the adjoint representation of S^3 is the derivative mapping of I_q at the identity e, i.e. $\mathrm{Ad}_q := \mathrm{T}_e I_q : \mathrm{s}^3 \to \mathrm{s}^3$, and we have

$$\mathrm{Ad}_q(\boldsymbol{u}) = vec(q \cdot u \cdot q^{-1}), \quad q \in \mathrm{S}^3, \quad u = (0, \boldsymbol{u}), \quad \boldsymbol{u} \in \mathrm{s}^3.$$

The map $\mathcal{E} : \mathrm{S}^3 \to SO(3)$, defined by

$$\mathcal{E}(q) = \mathbb{1} + 2q_0\widehat{\boldsymbol{q}} + 2\widehat{\boldsymbol{q}}^2, \tag{17}$$

is a group homomorphism since

$$\mathcal{E}(q \cdot p) = \mathcal{E}(q)\mathcal{E}(p), \quad \forall q, p \in \mathrm{S}^3.$$

Its derivative at e,

$$\mathcal{E}_* = \mathrm{T}_e\mathcal{E} : \mathrm{s}^3 \to \mathfrak{so}(3),$$

is thus a Lie algebra isomorphism. By a simple calculation it is possible to verify that

$$\mathcal{E}_*(u) = 2\widehat{\boldsymbol{u}}, \quad u \in \mathrm{s}^3. \tag{18}$$

The map $\mathcal{E}_*$ intertwines the two adjoint representations and we have

$$\mathcal{E}_*(q \cdot u \cdot q^{-1}) = \mathcal{E}(q)\mathcal{E}_*(u)\mathcal{E}(q^{-1}), \quad \forall\, u = (0, \boldsymbol{u}), \quad \boldsymbol{u} \in \mathbb{R}^3.$$

The map $\mathcal{E} : \mathrm{S}^3 \to SO(3)$ is not injective as $\mathcal{E}(q) = \mathcal{E}(-q)$.

The following proposition is a direct consequence of (18) and gives the rigid body equations of motion on S^3.

Proposition 2.3. *If $\boldsymbol{m}(t) \in \mathbb{R}^3$ satisfies (1) and $q(t) \in \mathrm{S}^3$ is such that*

$$\dot{q} = \frac{1}{2} q \cdot \omega, \tag{19}$$

with $\omega = (0, \boldsymbol{\omega}) \in \mathbb{R}^4$, $\boldsymbol{\omega} = I^{-1}\boldsymbol{m}$, then

$$\frac{d}{dt}\mathcal{E}(q) = \mathcal{E}(q)\widehat{\boldsymbol{\omega}}. \tag{20}$$

From the previous proposition we have that if $q \in \mathrm{S}^3$ satisfies (19), then $\mathcal{E}(q)$ satisfies (2).

We will derive formulae for q as a product of three quaternions. We begin by stating the following lemma.

Lemma 2.4. *If $p = (p_0, \boldsymbol{p}) \in \mathrm{S}^3$ with components p_0, p_1, p_2, p_3, is such that*

$$p \cdot m \cdot p^{-1} = e_3, \tag{21}$$

with $e_3 = (0, \boldsymbol{e}_3)$ and $m = (0, \boldsymbol{m})$ in S^3, then

$$p_0^2 + p_3^2 = \frac{1 + m_3}{2}, \tag{22}$$

$$p_1 = \frac{p_3 m_1 + p_0 m_2}{1 + m_3}, \tag{23}$$

$$p_2 = \frac{p_3 m_2 - p_0 m_1}{1 + m_3}. \tag{24}$$

Proof. From (23), we have $p \cdot m - e_3 \cdot p = 0$, hence, expanding the quaternion products, we obtain the conditions

$$\begin{array}{rcl} \boldsymbol{p}^T(\boldsymbol{m} - \boldsymbol{e}_3) & = & 0, \\ p_0 & = & \frac{(\boldsymbol{m} \times \boldsymbol{e}_3)^T \boldsymbol{p}}{1 - m_3}, \end{array} \tag{25}$$

or, equivalently

$$\begin{array}{rcl} p_1 m_1 + p_2 m_2 + p_3(m_3 - 1) & = & 0, \\ p_0(1 - m_3) - m_2 p_1 + m_1 p_2 & = & 0. \end{array} \tag{26}$$

By multiplying the first equation by m_2, the second by m_1, adding them together and solving for p_2, we obtain (24). The procedure for (23) is similar. Finally, using the obtained expressions for p_1 and p_2 and requiring $|p| = 1$, we obtain (22). □

Solving for p_0 in (22) and substituting in (23) and (24), we can express p_0, p_1 and p_2 in by means of p_3. These expressions are used in our implementation.

In what follows, we state the analogous of proposition 2.1.

Proposition 2.5. *Assume $p \in \mathrm{S}^3$ satisfies (21). If $q(t)$ is the solution of (19) with $q(t_0) = 1$, then we have*

$$q(t) = p(t_0)^{-1} \cdot y \cdot p(t), \tag{27}$$

where

$$y \cdot e_3 \cdot y^{-1} = e_3, \tag{28}$$

is a rotation around $\boldsymbol{e}_3$ with rotation angle

$$\psi = 2T(t - t_0) - 2 \int_{t_0}^{t} \boldsymbol{e}_3^T vec(\dot{p} \cdot p^{-1})\, ds. \tag{29}$$

Proof. Assume $q = p(t_0)^{-1} \cdot y \cdot p$ with p satisfying (21). One can easily verify that $y \cdot e_3 \cdot y^{-1} = e_3$. Therefore y has the form $y = (\cos(\psi/2), \sin(\psi/2)\boldsymbol{e}_3)$. From $1 = q(t_0) = p(t_0)^{-1} \cdot y(t_0) \cdot p(t_0)$, we have that $y(t_0)$ is the identity quaternion and therefore $\psi(t_0) = 0$.

By differentiating $q = p(t_0)^{-1} \cdot y \cdot p$ with respect to t, using (21) and substituting into (19), one obtains a differential equation for y,

$$\dot{y} = y \cdot \left(\frac{1}{2} p \cdot \omega \cdot p^{-1} - \dot{p} \cdot p^{-1} \right). \tag{30}$$

Since y and $\dot{y}$ have the second and third component equal to zero then also the quaternion product $y^{-1}\cdot\dot{y}$ does. Since $p\cdot\omega\cdot p^{-1}$ and $\dot{p}\cdot p^{-1}$ have both the first component equal to zero it follows that the three first components of $y^{-1}\cdot\dot{y}$ are zero, and therefore from (30) one obtains that

$$\dot{\psi} = 2\boldsymbol{e}_3^T vec(\frac{1}{2}p(t)\cdot\omega(t)\cdot p(t)^{-1} - \dot{p}(t)\cdot p(t)^{-1}), \tag{31}$$

with $\psi(t_0)=0$.

Observing that

$$\frac{1}{2}\boldsymbol{e}_3^T vec(p\cdot\omega\cdot p^{-1}) = \frac{1}{2}\boldsymbol{e}_3^T\mathcal{E}(p)\boldsymbol{\omega} = \frac{1}{2}\boldsymbol{m}^T\boldsymbol{\omega} = T,$$

from (31) we obtain

$$\psi = 2T(t-t_0) - 2\int_{t_0}^{t}\boldsymbol{e}_3^T vec(\dot{p}\cdot p^{-1})ds.$$

□

We stress here, for the sake of clarity, that the quaternion $y=(\cos(\psi/2),\sin(\psi/2)\boldsymbol{e}_3)$ is defined by using $\psi/2$, while ψ given by (29) is the angle of the corresponding planar rotation.

The product $q=p(t_0)^{-1}\cdot y\cdot p$ is the analogous of $Q=P_0^TYP$ of proposition 2.1. In fact

$$\mathcal{E}(p(t_0)^{-1}\cdot y\cdot p) = \mathcal{E}(p(t_0)^{-1})\mathcal{E}(y)\mathcal{E}(p),$$

with $\mathcal{E}(p)$ satisfying (9) and $\mathcal{E}(y)$ satisfying (10).

In the following we express the angle (31) by means for the components of p.

Corollary 2.6. *Under the hypotheses of the previous proposition one obtains that*

$$\psi = 2\int_{t_0}^{t}\left(\frac{2T+\omega_3}{2(1+m_3)} + 2\frac{\dot{p}_0p_3 - p_0\dot{p}_3}{1+m_3}\right)ds. \tag{32}$$

Proof. By expanding the quaternion product in (29), we obtain

$$\psi = 2\int_{t_0}^{t}\left(T + (\dot{p}_0p_3 - p_0\dot{p}_3) + (\dot{p}_1p_2 - p_1\dot{p}_2)\right)ds.$$

Next, we expand the expression

$$\dot{p}_1p_2 - p_1\dot{p}_2 = p_2^2\frac{d}{dt}\frac{p_1}{p_2} = \frac{(p_3m_2-p_0m_1)^2}{(1+m_3)^2}\frac{d}{dt}\left(\frac{p_3m_1+p_0m_2}{p_3m_2-p_0m_1}\right),$$

and, by using Lemma 2.4, we obtain

$$\dot{p}_1p_2 - p_1\dot{p}_2 = \frac{\dot{m}_1m_2 - m_1\dot{m}_2}{2(1+m_3)} + (\dot{p}_0p_3 - p_0\dot{p}_3)\frac{1-m_3}{(1+m_3)}.$$

The final expression (32) is obtained by using the equations of motion to express $\dot{m}_1$ and $\dot{m}_2$. □

2.4 Relation between the quaternion and the matrix approach

In the sequel we discuss different possible choices of the quaternion p giving rise to different exact algorithms for the attitude rotation of the free rigid body. Most of these algorithms are known in the literature and we propose here a unified quaternion framework.

In order to be able to identify the quaternion p corresponding to the a given rotation matrix $P \in SO(3)$ such that $P^T\boldsymbol{e}_3 = \boldsymbol{m}$, we consider the inverse of the restriction of $\mathcal{E}$ to an appropriate subset of $SO(3)$. The mapping $\mathcal{E} : S^3 \to SO(3)$ is not injective as the counter image $\mathcal{E}^{-1}(P) = \{p, -p\}$. Consider the following equivalence relation,

$$p \sim q \quad \text{if} \quad p = \pm q.$$

The restriction of $\mathcal{E}$ to $S^3/\!\sim$ is invertible. In the following proposition we give explicit formulae for $\mathcal{E}^{-1}(P)$, with $\mathcal{E}^{-1} : SO(3) \to S^3/\sim$.

Proposition 2.7. *Given $P \in SO(3)$ such that $P^T = [\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{m}]$, assume*

$$\begin{aligned} \alpha &:= \sqrt{\tfrac{v_2 m_2(1+m_3)+v_3(1+m_3-m_2^2)}{4m_1} + \tfrac{1+m_3}{4}}, \\ \beta &:= \sqrt{\tfrac{1+m_3}{2} - \alpha^2}, \\ \gamma &:= \tfrac{(4\alpha^2-1-m_3)m_1 m_2 - v_2(1+m_3)^2}{4(1+m_3-m_2^2)}, \end{aligned} \tag{33}$$

then the quaternion with components

$$\begin{aligned} p_0 &= \beta, \\ p_3 &= \operatorname{sign}(\gamma)\alpha, \end{aligned} \tag{34}$$

and p_1, p_2 given as in (23), (24) of Lemma 2.4, is such that $\mathcal{E}(p) = P$.

Proof. Using (17) one easily obtains the following explicit expression,

$$\mathcal{E}(p)^T = \begin{pmatrix} 1-2(p_3^2+p_2^2) & 2p_0p_3+2p_1p_2 & -2p_0p_2+2p_3p_1 \\ -2p_0p_3+2p_1p_2 & 1-2(p_3^2+p_1^2) & 2p_0p_1+2p_3p_2 \\ 2p_0p_2+2p_3p_1 & -2p_0p_1+2p_3p_2 & 1-2(p_2^2+p_1^2) \end{pmatrix},$$

that we want to be equal to $P^T = [\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{m}]$. This gives us a system of 9 equations in 4 unknowns, $p_0, \dots, p_3$. The choice $P^T\boldsymbol{e}_3 = \boldsymbol{m}$ implies that $\mathcal{E}(p)^T\boldsymbol{e}_3 = \boldsymbol{m}$. Therefore the hypotheses of Lemma 2.4 are fulfilled for the quaternion p, and (22), (23), (24) hold.

This allows to derive an expression for p_3^2. One can proceed as follows. Equating the components $(1,3)$ and $(3,1)$ in $\mathcal{E}(p) = P$ gives

$$m_1 = -2p_0p_2 + 2p_3p_1, \quad v_3 = 2p_0p_2 + 2p_3p_1.$$

Adding these two equations and expanding the product p_3p_1 using Lemma 2.4, we obtain

$$\frac{v_3 + m_1}{4} = \frac{p_3^2 m_1 + p_3 p_0 m_2}{1 + m_3}. \tag{35}$$

Next, we express p_0p_3 by means of p_3^2 only. To this end, we use $v_2 = -2p_0p_3 + 2p_1p_2$, and expand the product p_1p_2 using Lemma 2.4. After some algebra we obtain

$$p_0p_3 = \frac{(4p_3^2 - 1 - m_3)m_1m_2 - v_2(1+m_3)^2}{4(1+m_3-m_2^2)}. \tag{36}$$

Substituting in (35) and solving for p_3^2 we obtain

$$p_3^2 = \frac{v_2m_2(1+m_3) + v_3(1+m_3-m_2^2)}{4m_1} + \frac{1+m_3}{4}, \tag{37}$$

and therefore $p_3^2 = \alpha^2$, and $p_0p_3 = \gamma$. The absolute values of p_0 and p_3 are obtained extracting the square roots of $p_0^2 = \frac{1+m_3}{2} - p_3^2$, and (37) respectively. The sign of p_0 and p_3 must be consistent with the sign of p_0p_3 given by (36), by choosing $p_0 = \beta$ positive, we have $p_3 = \operatorname{sign}(p_0p_3)\alpha = \operatorname{sign}(\gamma)\alpha$. □

In what follows we consider a list of different choices of p corresponding to different formulations of the exact solution known in the literature.

case(a) Choosing $p_0 = c_1\sqrt{1+m_3}$ and $p_3 = c_2\sqrt{1+m_3}$ with c_1 and c_2 constants such that $c_1^2 + c_2^2 = \frac{1}{2}$ (for example $c_1 = \frac{1}{\sqrt{2}}$ and $c_2 = 0$), the integrand of (32) simplifies to

$$\beta(t) = \frac{2T + \omega_3}{2(1+m_3)}.$$

This expression can be further rewritten in the form

$$\beta(t) = \frac{1}{2I_3} + \frac{2TI_3 - 1}{2I_3\left[1 + (I_3\xi_3)\operatorname{cn} u\right]},$$

where $\xi_3 = a_3/\sqrt{I_3(I_3 - I_1)}$. Integration of $\beta(t)$, (see [4, page 215]), yields

$$\psi = \tfrac{1}{I_3}(t - t_0) + \tfrac{2TI_3-1}{I_3\lambda(1-\mu^2)}\left[\Pi(\varphi, n, k) - \Pi(\varphi_0, n, k) - \mu\left(F(t) - F(t_0)\right)\right], \tag{38}$$

where

$$\mu := I_3\xi_3, \quad (0 < \mu^2 < 1), \quad n := \frac{\mu^2}{\mu^2 - 1},$$

$$F(t) = \sqrt{\frac{1}{k^2 - n}}\tan^{-1}\left(\operatorname{sd} u(t)\sqrt{k^2 - n}\right), \quad n < k^2,$$

and Π is the incomplete elliptic integral of the third kind defined by

$$\Pi(\phi, n, k) := \int_0^u \frac{ds}{1 - n\operatorname{sn}^2 s} \quad -\infty < n < \infty, \quad \phi = \operatorname{am}(u(t)), \quad u(t) = \lambda(t-\nu).$$

This choice produces a rescaled and simplified version of the algorithm presented by Kosenko in [14].

case(b) Assume we wish to determine p by choosing the rotation matrix $\mathcal{E}(p)^T = [\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{m}]$ as considered in section 2.2.

By taking $\boldsymbol{v} = \frac{\dot{\boldsymbol{m}}}{\|\dot{\boldsymbol{m}}\|}$ and using proposition 2.7, one obtains

$$p_3^2 = \frac{1+m_3}{4} + \frac{-a\,m_2m_3 + b\,m_2m_3^2 + c\,m_2(1-m_2^2)}{4\|\dot{\boldsymbol{m}}\|},$$

with a, b, c defined in (15).

Substituting in the formulas of Lemma 2.4 and taking ψ as in (14) one obtains a quaternion formulation of the algorithm in section 2.2.

case (c) Analogously, taking $\boldsymbol{v} = \frac{\boldsymbol{m}\times\boldsymbol{e}_3}{\|\boldsymbol{m}\times\boldsymbol{e}_3\|}$ and $\boldsymbol{w} = \frac{\boldsymbol{m}\times(\boldsymbol{m}\times\boldsymbol{e}_3)}{\|\boldsymbol{m}\times(\boldsymbol{m}\times\boldsymbol{e}_3)\|}$ one obtains

$$p_3^2 = \frac{1+m_3}{4} - \frac{m_2(1+m_3)}{4\sqrt{1-m_3^2}},$$

giving rise to a different algorithm analogous to the one recently considered by van Zon and Schofield in [23]. The rotation angle is

$$\psi = \int_{t_0}^{t} \frac{2T - \frac{1}{I_3}m_3^2}{1-m_3^2} ds = \frac{1}{I_3}(t-t_0) + \frac{(2TI_3-1)(I_3-I_1)}{\lambda\left((I_3-I_1)I_3 - a_3^2 I_3^2\right)} \left(\Pi(\varphi, n, k) - \Pi(\varphi_0, n, k)\right),$$

with

$$n = \frac{a_3^2 I_3}{a_3^2 I_3 - (I_3 - I_1)},$$

where $\varphi = \mathrm{am}(\lambda(t-\nu))$ and $\varphi_0 = \mathrm{am}(\lambda(t_0-\nu))$.

3 Numerical experiments

3.1 Numerical implementation

The exact algorithm described in this paper require the computation of elliptic integrals of the first and third kind. Elliptic integrals of the first kind are computed very fast by using standard algorithms like AGM (arithmetic geometric mean) and ascending/descending Landen trasformations [1]. These can be used also for the elliptic integral of the third kind, but their performance is not so uniform and other algorithms are preferred instead. In [23] the authors use a method based on theta functions. Our implementation makes use of Carlson's algorithms `rf`, `rj`, `rc`, that have been acclaimed to produce accurate values for large sets of parameters. These methods are described in details in [20] and are the most common routines in several scientific libraries.

An alternative to the exact computation of this elliptic integral is the approximation by a quadrature method. In [24], the integral

$$\int_{u_0}^{u} \frac{ds}{1 - n\,\mathrm{sn}^2 s}$$

is approximated by a quadrature based on Hermite interpolation, as the function sn and its derivative can be easily computed at the endpoints of the interval. Instead, we prefer to write the same integral in the Legendre form,

$$\int_{\varphi_0}^{\varphi} \frac{d\theta}{(1-n\sin^2\theta)\sqrt{1-k^2\sin^2\theta}}, \quad -\infty < n < \infty, \qquad \varphi = \mathrm{am}(u). \tag{39}$$

To our opinion, this format is more suitable to approximation by quadrature formulae because it requires tabulating the sine function in the quadrature nodes instead of $\mathrm{sn}(\lambda(t-\nu))$. Thus, (39) can be approximated as

$$\int_{\varphi_0}^{\varphi} f(\theta)\mathrm{d}\theta \approx \sum_{i=1}^{p} b_i f(\varphi_0 + a_i \Delta\varphi),$$

where $\Delta\varphi = \varphi - \varphi_0$ and b_i, a_i are weights and nodes of a quadrature formula respectively. We use Gaussian quadrature, as it is known to give the highest quadrature order ($2p$) for a given number of nodes (p). The coefficients and weights for Gaussian quadrature of order 10 used in this paper are reported in Appendix. Our numerical experiments indicate that this approximation is very effective. For instance, a 5 point Gaussian quadrature (order 10) gives very accurate results even for moderately large step-sizes, and reduces the overall cost of the methods by $2/3$.

3.2 Free rigid body

In this section we compare several FORTRAN algorithms for the numerical approximation of the free rigid body. The methods are: `dmv6, dmv8, dmv10`, the methods based on the modified rattle algorithms of order 6, 8 and 10, respectively, for the free rigid body [12]; `jacobirot` and `jacobi_quat` computing the exact solution with the rotation and the quaternion method proposed in this paper along with their variants in which the elliptic integral is approximated by Gauss quadrature formulae of order 6, 8 and 10. All the methods, except the `jacobirot` and `jacobirot_gauss`, employ a quaternion description for the attitude matrix Q.

In the first experiment, we compute the number of significant digits (precision) for the attitude matrix[1] and the cpu-time averages of the different methods in an interval of integration $[0, 10]$, with twenty different stepsizes ranging from about 0.34 down to 0.014. We choose a random inertia tensor, normalized so that $I_1 < I_2 < I_3 = 1$, thereafter a random initial angular moment in the first quadrant. This is not a restriction, as both scaling the inertia tensor and the angular momentum are equivalent to a time reparametrization. The initial condition for the attitude matrix is the identity matrix (for rotations) and the quaternion $(1, 0, 0, 0)$ for the other methods. The exact solution is computed with Matlab's `ode45` routine, setting both relative and absolute error to machine precision. In Figure 1 we show the average number of significant digits and the time averages for the methods under investigation, the samples are computed over 100 runs. The exact methods are clearly more expensive, but they always converge (against 75 successes for the methods `dmv6, dmv8, dmv10`, that are depending on a step size "small enough" for the fixed point iterations to converge). The diverging runs of the `dmv` methods are not taken into account when computing averages. Good behaviour is displayed also by the methods using Gaussian quadrature instead of the exact elliptic integrals of third kind. Their cost is about $1/3$ of the methods using the exact elliptic integral (and this is reasonable, because the exact routines compute 3 elliptic integrals of the third kind: the complete one between 0 and $\pi/2$, and two incomplete ones between 0 and ϕ, where $0 \leq \phi \leq \pi/2$).

The exact methods discussed in this paper reveal a worse accumulation of roundoff error (see Figure 1, top plot). However, given to the exact nature of the method, it is not necessary to perform many tiny steps for integrating to the final time: a single time stepping is enough, and this avoids the problems related to the accumulation of roundoff error. Numerical experiments reveal that the accuracy of the two exact methods is very comparable and also their cost (see Figure 2 and Table 1).

[1] Our methods use an exact approximations for the angular momentum, while the `dmv` methods do not. However both the two classes of methods preserve exactly the kinetic energy, the norm of the angular

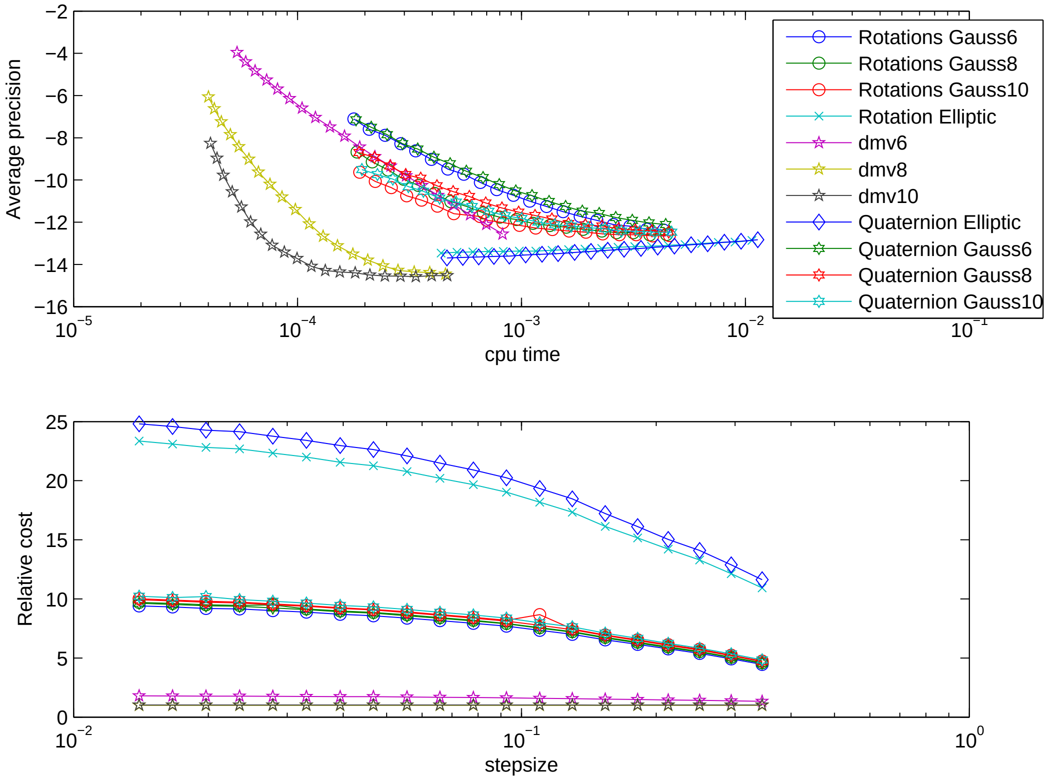

Figure 1: Top: Average precision versus average cpu times in the attitude matrix (100 runs) for random initial conditions and random inertia moments. Bottom: Relative cost (with respect to the cheapest method) versus stepsize. The methods computing the exact solutions are more expensive then the approximated ones, but their relative cost rate improves for large time-steps. The methods `dvm6`, `dmv8` and `dmv10` converge 75 out of the 100 runs. The failures are not taken into account when computing the averages.

In general, when these exact methods are applied within a splitting method, the value of the parameters (angular momentum, attitude, energy) will change before and after one free rigid body step, hence the problem of roundoff accumulation will not be present.

Our extensive numerical experiments revealed that the performance of the methods depended heavily on the inertia matrix I and the initial condition $\boldsymbol{m}_0$ for the angular momentum. To understand this dependence, we have followed a procedure similar to the one used in [11]. Since normalizing the inertia matrix is equivalent to a time reparametrization, it is sufficient to consider values of the form $I_1/I_3 < I_2/I_3 < 1$. This reduces to considering two parameters, say $x = I_1/I_3$ and $y = I_2/I_3$. By a symmetry argument, the problem can be further reduced to considering values of x and y in the triangle

$$\mathcal{T} = \{(x, y) \in \mathbb{R}^2 : 0 < 1 - y \leq x < y < 1\},$$

(see Figure 3).

We construct a discretization of this triangle (100 points in the x direction and 50 in the y direction), and for each point $(x, y, 1)$ we solve 20 initial value

momentum, its projection, are time-reversible and Lie–Poisson integrators.

	Median error ($\times 10^{-13}$)	σ ($\times 10^{-12}$)	Median cpu ($\times 10^{-5}$)	σ ($\times 10^{-6}$)
Rotation	7.0865	4.4725	2.2466	2.4379
Quaternion	3.3383	2.3748	2.1452	3.0631

Table 1: Average error and cpu for the two exact methods (rotation and quaternion) for a single timestepping ($h = 5$), over 100 experiments with 100 random initial angular momentum and identity matrix as initial attitude. The exact reference solutions is computed with `ode45`. See also Figure 2. The computational effort is dominated by the cost of computing the elliptic integrals of the third kind.

problems with initial condition $\boldsymbol{m}_0$ in the first quadrant (again, by a symmetry argument, this choice is representative of all possible behaviors). Thereafter, we average the number of significant digits of the methods (non converging runs for the `dmv` methods are discarded). The results of the experiments are shown in Figures (4–7), computed with integration stepsize $h = 0.4$ and Figures (8–11), computed with integration stepsize $h = 0.04$. For the largest stepsize, $h = 0.4$, the exact methods described in this paper perform very similarly and show an uniform accuracy. We compare then approximate methods of the same order (6, 8, and 10), for the rotation method, the quaternion method and the `dmv` method. The rotation and quaternion methods employ a Gauss–Legendre approximation of the elliptic integral. All the approximate methods reveal a worse approximation in the proximity of the top left corner

$$0 \approx x = \frac{I_1}{I_3} \ll y = \frac{I_2}{I_3} \approx 1 = \frac{I_3}{I_3},$$

namely when one of the moments of inertia is much smaller than the two others, that are quite close (very thin disc case). The `dmv` methods have in average less accuracy and they failed to converge for several initial conditions. Nevertheless, for this choice of the stepsize (see Figure 1), they are 5 times faster than their approximate counterparts based on the algorithms described in this paper. For the next value of the stepsize ($h = 0.04$) the exact methods reveal a worse accumulation of roundoff error, already observed in Figure 1. The `dmv`, in particular `dmv10`, performs very well in the whole triangle, except for the top left corner.

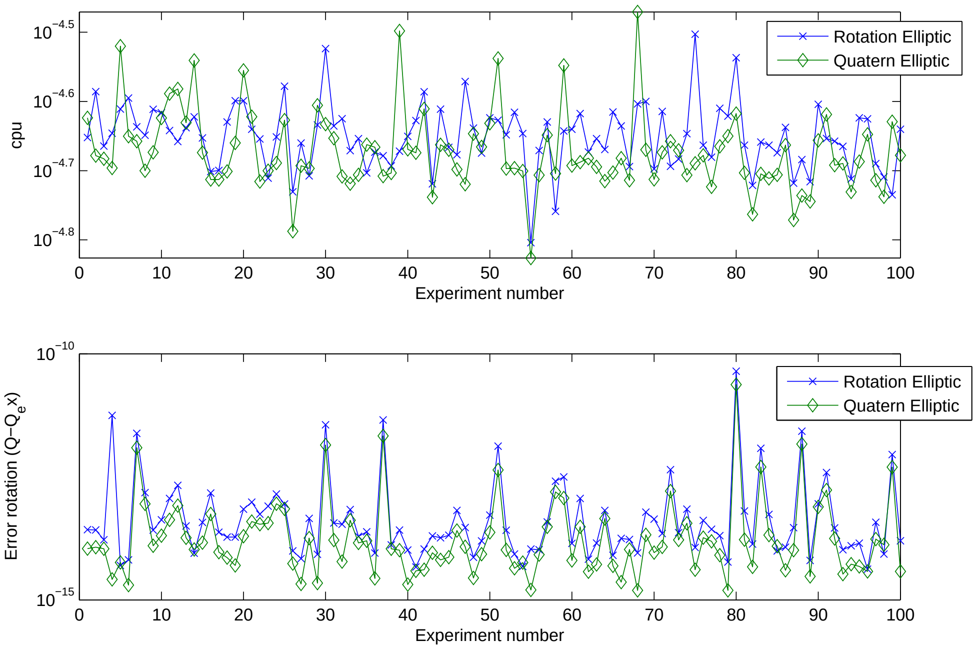

Figure 2: Top: Cpu in 100 runs for the exact method based on rotations (crosses) and the quaternion method (diamonds). Bottom: Error in the rotation matrix (infinity norm) for the rotation method and the quaternion method. Symbols as above. The exact reference solutions is computed with `ode45`. See Table 1 for median and standard deviation values.

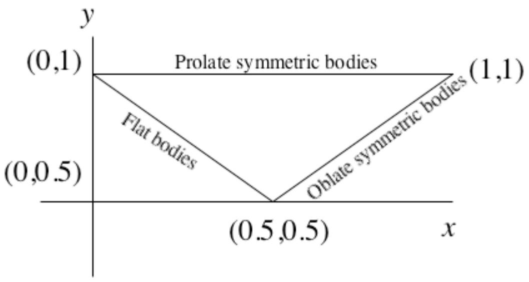

Figure 3: Parametrization domain for the Inertia matrix. x-axis: I_1/I_3, y-axis: I_2/I_3.

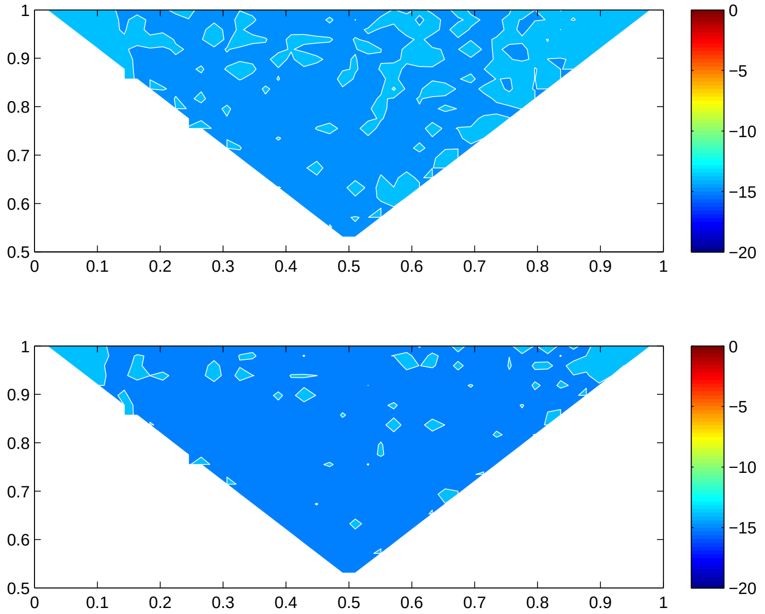

Figure 4: Significant digits for the various values of the inertia matrix with stepsize $h = 0.4$. Comparison of exact methods. Top: Rotation elliptic. Bottom: Quaternion elliptic.

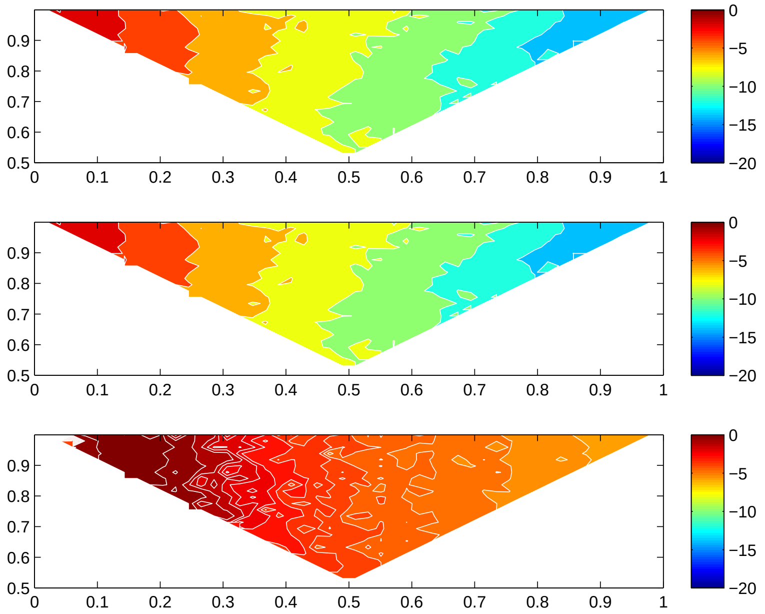

Figure 5: Significant digits for the various values of the inertia matrix with stepsize $h = 0.4$. Comparison of order 6 methods. Top: Rotation G6. Middle: Quaternion G6. Bottom: `dmv6`.

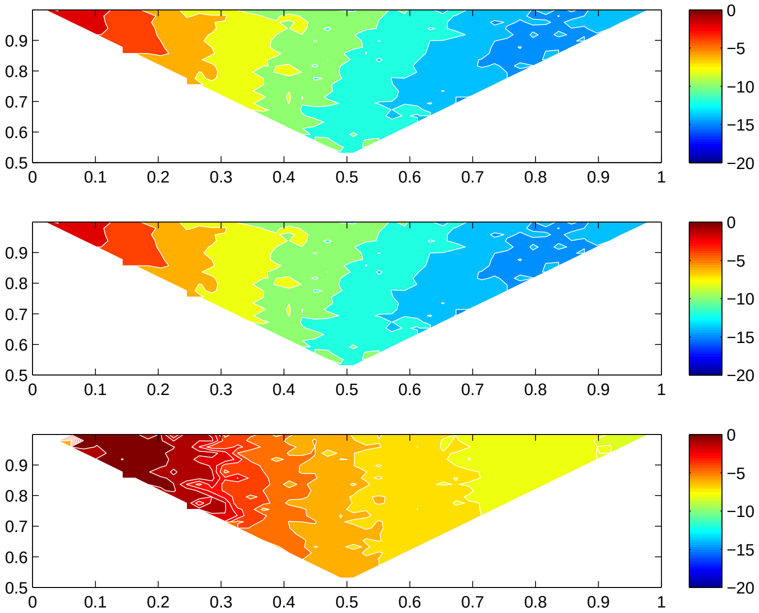

Figure 6: Significant digits for the various values of the inertia matrix with stepsize $h = 0.4$. Comparison of order 8 methods. Top: Rotation G8. Middle: Quaternion G8. Bottom: `dmv8`.

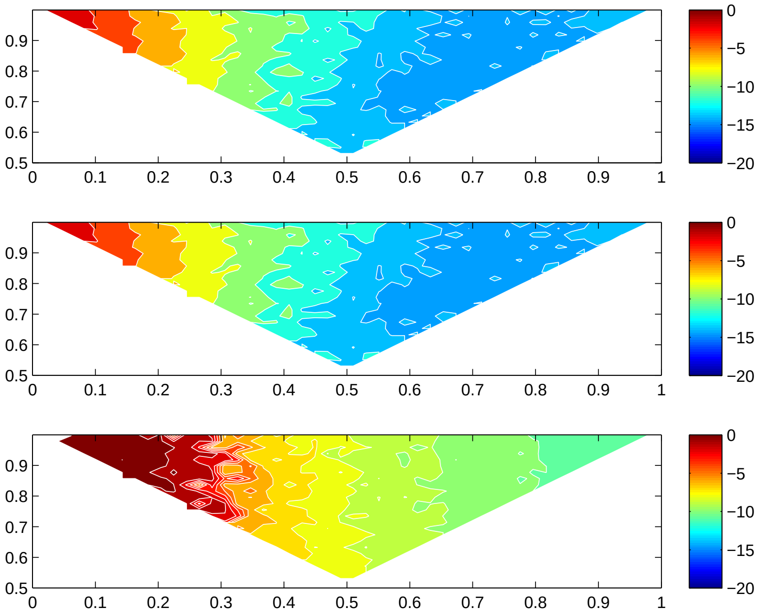

Figure 7: Significant digits for the various values of the inertia matrix with stepsize $h = 0.4$. Comparison of order 10 methods. Top: Rotation G10. Middle: Quaternion G10. Bottom: `dmv10`.

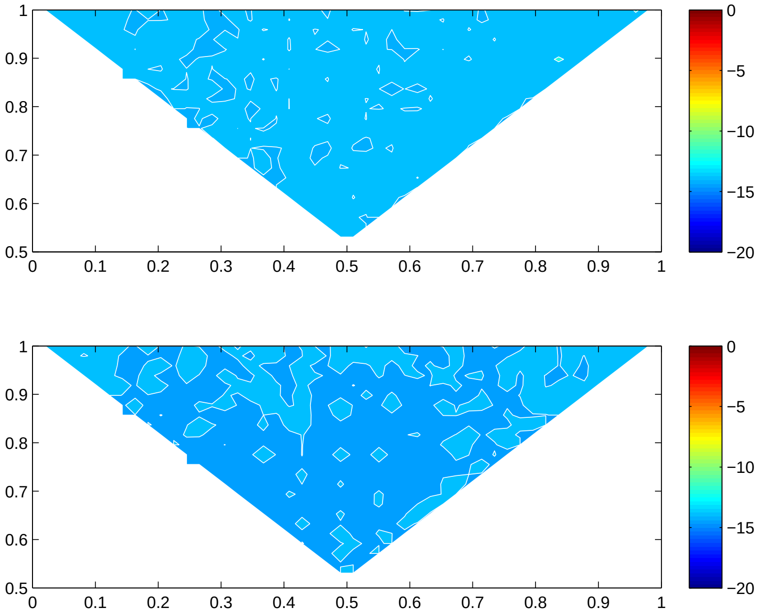

Figure 8: Significant digits for the various values of the inertia matrix with stepsize $h = 0.04$. Comparison of exact methods. Top: Rotation elliptic. Bottom: Quaternion elliptic.

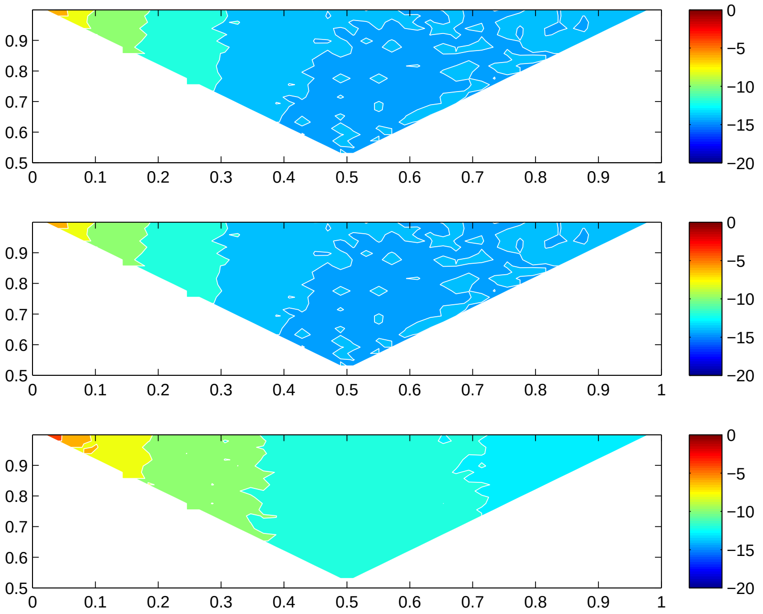

Figure 9: Significant digits for the various values of the inertia matrix with stepsize $h = 0.04$. Comparison of order 6 methods. Top: Rotation G6. Middle: Quaternion G6. Bottom: `dmv6`.

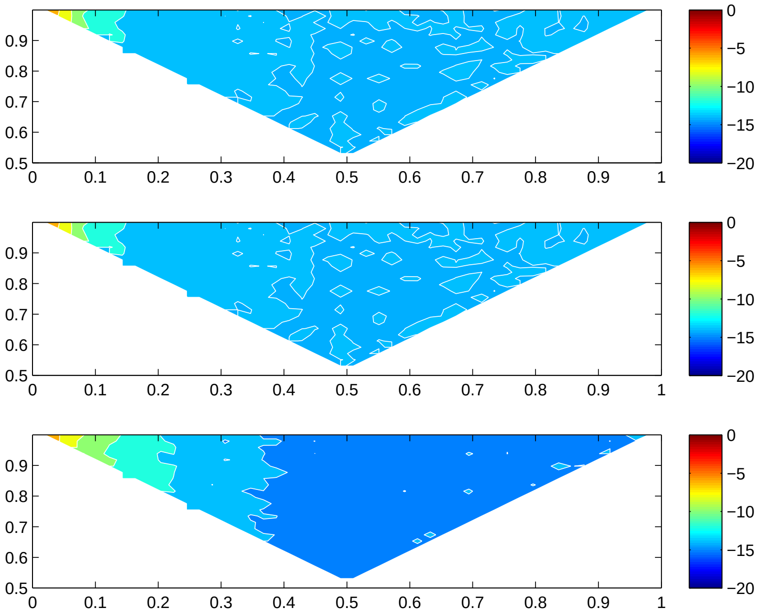

Figure 10: Significant digits for the various values of the inertia matrix with stepsize $h = 0.04$. Comparison of order 8 methods. Top: Rotation G8. Middle: Quaternion G8. Bottom: `dmv8`.

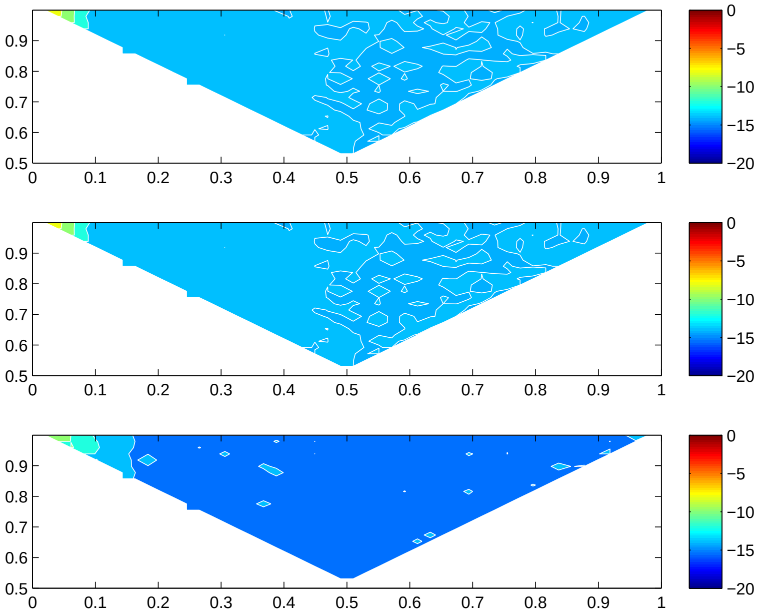

Figure 11: Significant digits for the various values of the inertia matrix with stepsize $h = 0.04$. Comparison of order 10 methods. Top: Rotation G10. Middle: Quaternion G10. Bottom: `dmv10`.

3.3 Torqued systems and perturbations of free rigid body motions

In this section we consider systems of the form

$$H(\boldsymbol{m}, Q) = T + V, \qquad T(\boldsymbol{m}) = \frac{1}{2}\boldsymbol{m}^T(I^{-1}\boldsymbol{m}), \quad V = V(Q). \tag{40}$$

A standard approach to solve this problem is to split it into a free rigid body motion plus a torqued motion, namely

$$S_1 = \begin{cases} \dot{\boldsymbol{m}} = \boldsymbol{m} \times I^{-1}\boldsymbol{m}, \\ \dot{Q} = Q\,\widehat{I^{-1}\boldsymbol{m}}, \end{cases} \tag{41}$$

free motion corresponding to the kinetic part, and

$$S_2 = \begin{cases} \dot{\boldsymbol{m}} = \mathbf{f}(Q), \\ \dot{Q} = 0, \end{cases} \tag{42}$$

Thereafter, the flows of the S_1 and S_2 systems are composed for instance by means of a splitting method [17].

The most commonly used is the symplectic second order Störmer/Verlet scheme

$$(\boldsymbol{m}, Q)^{(j+1)} = \varphi_{h/2}^{[S_2]} \circ \varphi_h^{[S_1]} \circ \varphi_{h/2}^{[S_2]}((\boldsymbol{m}, Q)^{(j)}), \quad j = 0, 1, \ldots,$$

where $\varphi_h^{[S_1]}$ and $\varphi_h^{[S_2]}$ represent the exact flows of S_1 and S_2, respectively. Some higher order schemes are presented in the appendix.

3.3.1 The heavy top

For this system we have

$$V(Q) = \varepsilon \boldsymbol{e}_3^T Q^T \boldsymbol{u}_0,$$

as potential energy. The vector $\boldsymbol{u} = Q^T\boldsymbol{u}_0$ describes position of the center of mass times the gravitational force. This potential V corresponds to $\mathbf{f}(Q) = (u_2, -u_1, 0)^T$, where u_1 and u_2 are components of $\boldsymbol{u}$.

One of the most popular (and elegant) methods for approximating the free rigid body system (41) is a second-order method designed by McLachlan and Reich MR (see [8]). This method has very nice geometric properties, as it is time-reversible and preserves the Poisson structure of the system. In brief, the MR method is based on a splitting of the Hamiltonian (40) into four parts,

$$\tilde{H}_1 = \frac{m_1^2}{2I_1}, \quad \tilde{H}_2 = \frac{m_2^2}{2I_2}, \quad \tilde{H}_3 = \frac{m_3^2}{2I_3}, \quad \tilde{H}_4 = V(Q).$$

Each of the corresponding Hamiltonian vector fields can be integrated exactly ($\tilde{H}_1$, $\tilde{H}_2$, $\tilde{H}_3$ correspond to the vector fields (1)), the symmetric composition of the flows gives rise to the approximation scheme,

$$(\boldsymbol{m}, Q)^{(j+1)} = \Phi_{MR}((\boldsymbol{m}, Q)^{(j)}),$$

where

$$\Phi_{MR} = \varphi_{4,h/2} \circ \Phi_{T,h} \circ \varphi_{4,h/2}.$$

Here

$$\Phi_{T,h} = \varphi_{1,h/2} \circ \varphi_{2,h/2} \circ \varphi_{3,h} \circ \varphi_{2,h/2} \circ \varphi_{1,h/2}$$

is the contribution from the kinetic parts, $\tilde{H}_1$, $\tilde{H}_2$ and $\tilde{H}_3$, where the flows of the kinetic parts corresponds to elementary rotations in $\mathbb{R}^3$.

We consider next perturbations of free rigid body motions,

$$V(Q) = \varepsilon\, \boldsymbol{e}_3^T Q^T \boldsymbol{u}_0, \tag{43}$$

with ε small. We compare different splitting methods for the following values of the kinetic energy $T_0 = T(\boldsymbol{m}_0) = \varepsilon^2, \varepsilon, 1, 1/\varepsilon, 1/\varepsilon^2$. The total Hamiltonian energy is of the form

$$H = T_0 + \mathcal{O}(\varepsilon).$$

The initial conditions are chosen as follows. Having fixed a value of ε, we choose a random inertia tensor, normalized so that $I_1 = 1$ (other choices correspond to time reparametrizations of the flow). Having chosen the first two components of $\boldsymbol{m}_0$ randomly, the remaining one is determined to match T_0. The vector $\boldsymbol{u}_0$ is a random vector normalized to have unit norm. Q_0 is the identity matrix. For a splitting method of order p, it is reasonable to expect a leading error term of the form εh^p, i.e. linear in ε. Several splitting methods are compared, each timing and relative Hamiltonian error is averaged (mean value) over 20 different initial conditions (each with new $I, \boldsymbol{m}_0, \boldsymbol{u}_0$). The methods are implemented so that all the splitting schemes perform the same number force $\mathbf{f}$ evaluations. This is done as follows: start with the following basic time steps: $h \in \{8, 5, 4, 2, 1.75, 1.5, 1.25, 1, 0.5\}$. For a splitting method with s stages (s is the number of evaluations of the force), we use $h_s = c_s h = \frac{s}{10} h$. For instance, for the 6th order 10-stages method $S6_{10}$, $c_s = 1$, for the Verlet splitting $c_s = \frac{1}{10}$. The integration is performed in the interval $[0, 2000]$ (see Figures 12-13). Note that only the methods using the exact integrator produce an error that is smaller than ε for all the choices of the step size.

3.3.2 Satellite simulation

We consider a simplified model describing the motion of a satellite in a circular orbit of radius r around the earth [15]. Denote $\mu = gM$, where g is the gravitational constant and M is the mass of the earth. The potential energy of this system is given by

$$V(Q) = 3\frac{\mu}{2r^3}(Q^T \mathbf{e}_3)^T I (Q^T \mathbf{e}_3), \tag{44}$$

where I is the inertia tensor and $\mathbf{e}_3$ is the canonical vector $(0,0,1)^T$ in $\mathbb{R}^3$. The torque associated to this potential becomes

$$\mathbf{f}(Q) = 3\frac{\mu}{r^3}(Q^T \mathbf{e}_3) \times I(Q^T \mathbf{e}_3). \tag{45}$$

We simulate the motion of the satellite using the same parameters as in [18], namely

$$I_1 = 1.7 \times 10^4, \qquad I_2 = 3.7 \times 10^4, \qquad I_3 = 5.4 \times 10^4,$$

with

$$\mu = 3.986 \times 10^{14}, \qquad r = 1.5 \times 10^5,$$

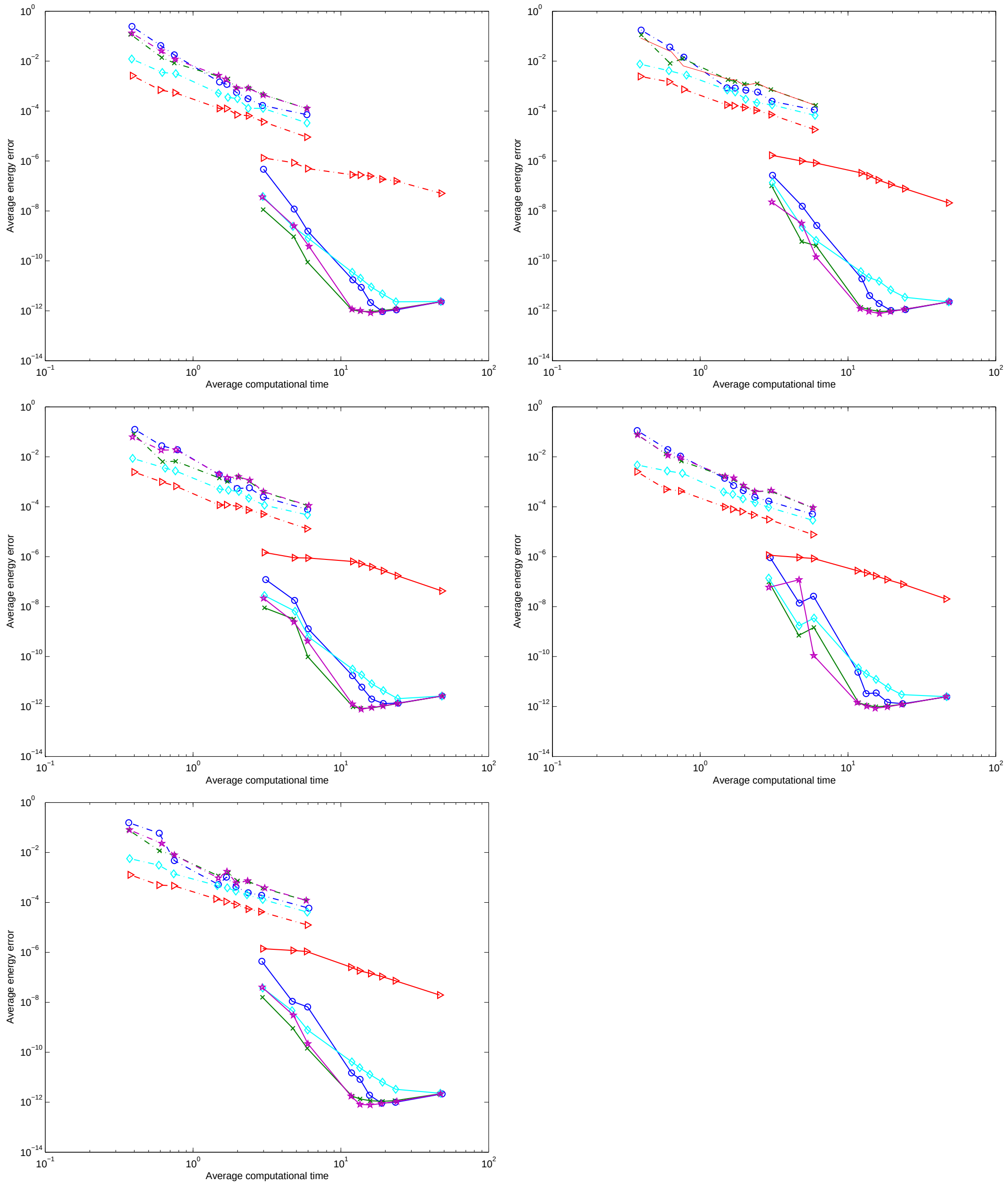

Figure 12: Average relative errors versus computational time, perturbed rigid body, $\varepsilon = 10^{-6}$. Initial kinetic energies (from top left) $T_0 = \varepsilon^2, \varepsilon, 1, 1/\varepsilon, 1/\varepsilon^2$. Solid lines: splitting methods using exact RB solutions. Dash-dotted lines: splitting methods using McL2 approximation for RB. See text for details. Circles: S6_{10}, triangles: V2+McL2, times: SRKN4_6^b, diamonds: S4_6, pentagrams: SRKN6_{14}^a.

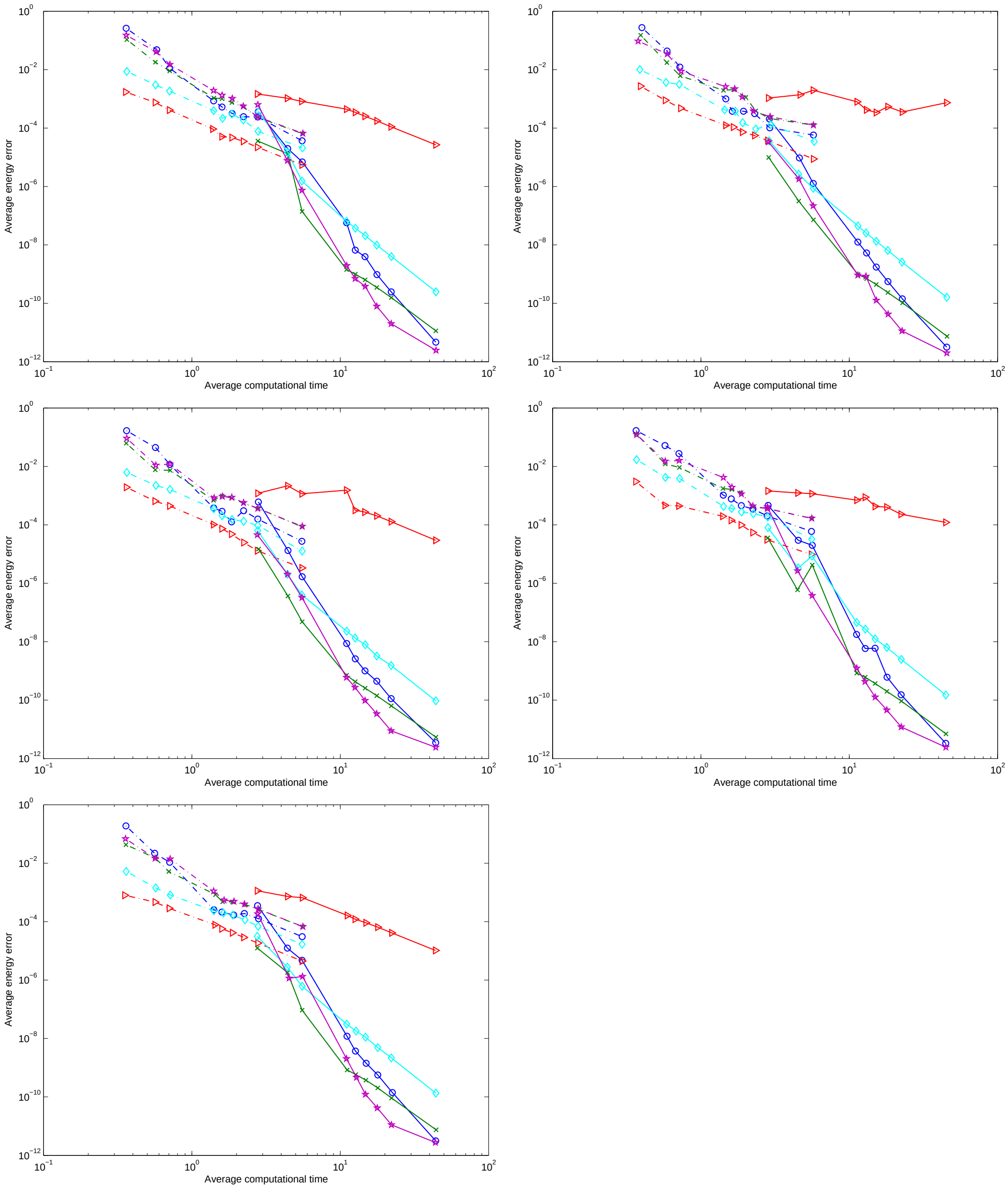

Figure 13: Average relative errors versus computational time, perturbed rigid body, $\varepsilon = 10^{-3}$. Initial kinetic energies (from top left) $T_0 = \varepsilon^2, \varepsilon, 1, 1/\varepsilon, 1/\varepsilon^2$. Solid lines: splitting methods using exact RB solutions. Dash-dotted lines: splitting methods using McL2 approximation for RB. See text for details. Circles: S6_{10}, triangles: V2+McL2, times: SRKN4_6^b, diamonds: S4_6, pentagrams: SRKN6_{14}^a.

in the interval $[0, 400]$. The initial angular velocity is $\boldsymbol{\omega}_0 = (15, -15, 15)^T$, corresponding to an angular momentum $\boldsymbol{m}_0 = I\boldsymbol{\omega}_0$. The initial attitude Q_0 is the identity matrix. The system has an energy $H_0 = 1.21595664 \times 10^7$, which is conserved in time. The splitting method based on the exact approximation of the rigid body is very accurate. The motion of the center of mass (left column) and the relative error on the energy H_0 (right column) for the splitting method SRKN6^a_{14} employing our exact solution, are shown in Figure 15. The integration is performed in the interval $[0, 400]$ with with stepsize $h = 0.1$ (top) and $h = 0.05$ (bottom). The relative error on the energy (see Figure 15), which is of the order of 10^{-7} for $h = 0.1$ and 10^{-10} for $h = 0.05$, indicates that H_0 is preserved to 7 and 10 digits respectively. The corresponding plots for the MR method are shown in Figure (14).

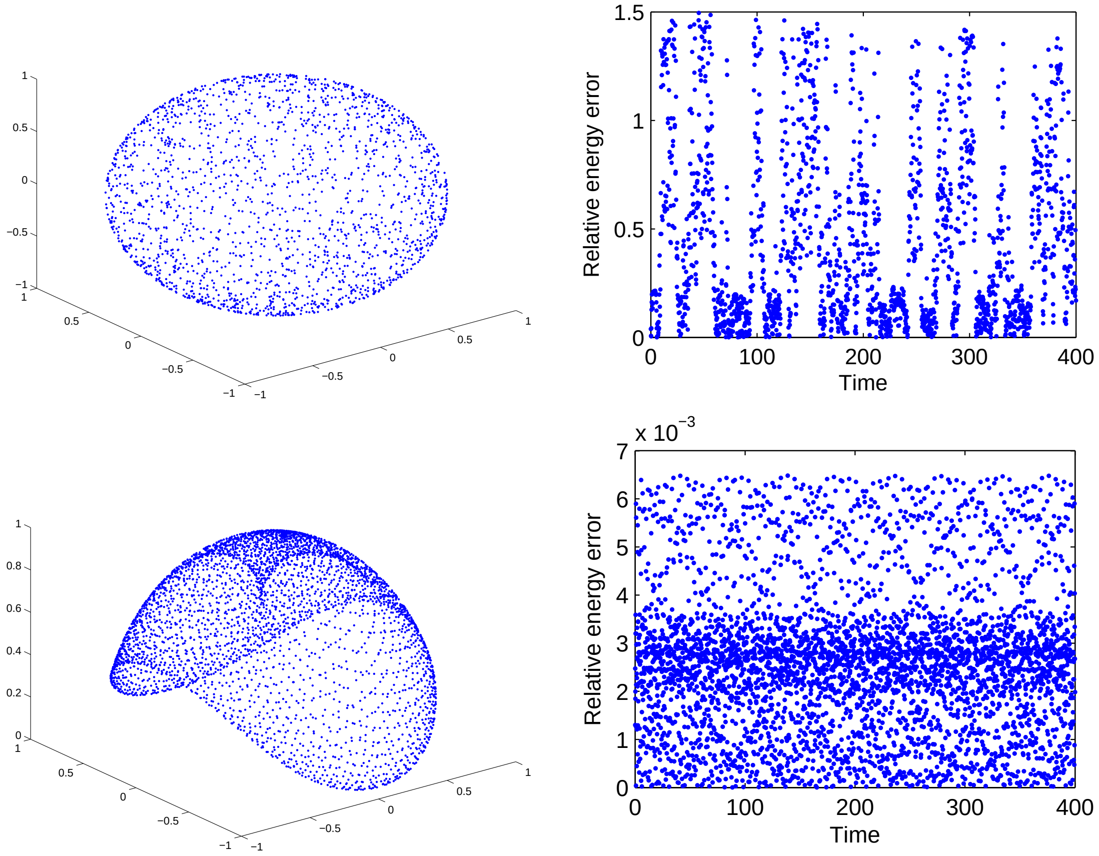

Figure 14: Satellite simulation. Left column: Center of mass $(Q^T \mathbf{e}_3)$ by the splitting method MR with stepsize $h = 0.1$ (top) and $h = 0.05$ (bottom). Right: Relative error on the energy corresponding to the same stepsizes. See text for details.

3.4 Molecular dynamics simulation: Soft dipolar spheres

We consider a molecular dynamics simulation, where molecules are modeled as dipolar soft spheres. This model is of interest because it can be used to study water and aqueous solutions, as water molecules can be described as small dipoles. We consider the system described in example b in Appendix A of [8].

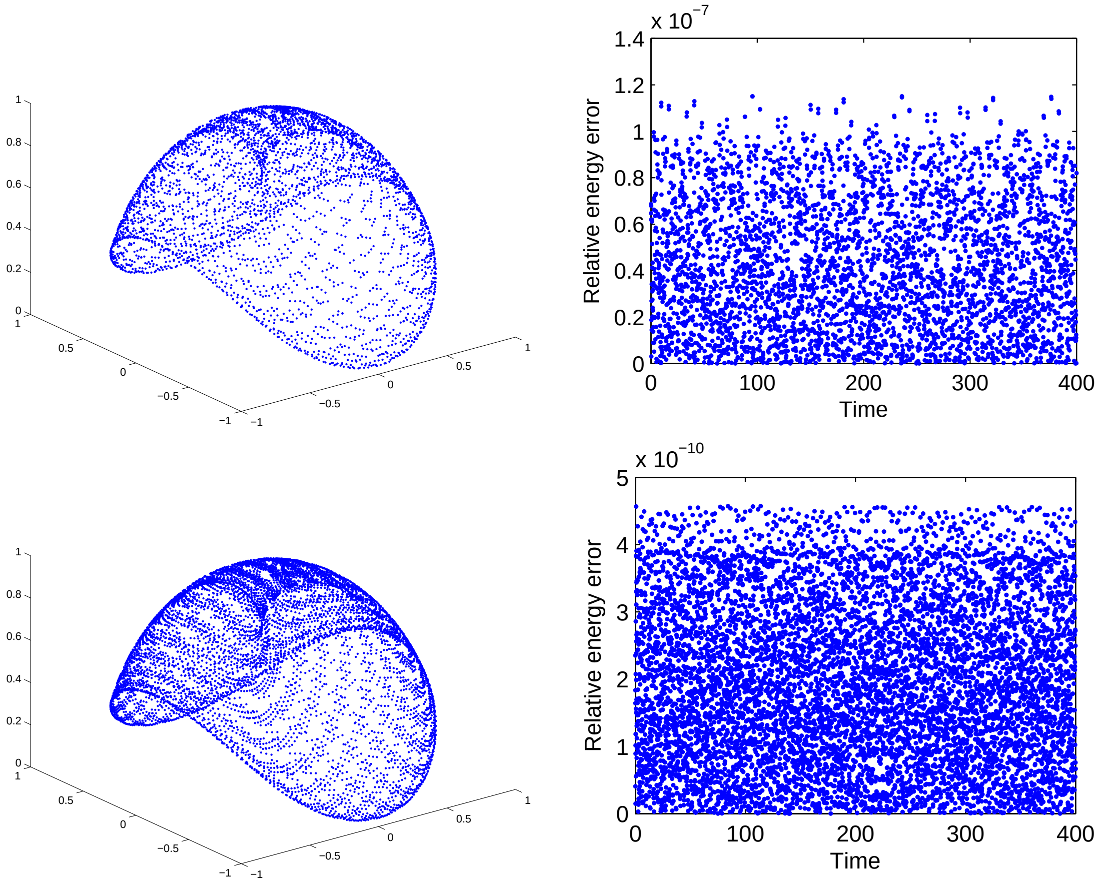

Figure 15: Satellite simulation. Left column: Center of mass $(Q^T \mathbf{e}_3)$ by the splitting method SRKN6^a_{14} with stepsize $h = 0.1$ (top) and $h = 0.05$ (bottom). Right: Relative error on the energy corresponding to the same stepsizes. See text for details.

Denote by m_i the total mass of the ith body, by $\mathbf{q}_i$ the position of its center of mass, by $\mathbf{p}_i$ its linear momentum, by $\mathbf{Q}_i$ its orientation and, finally, by $\boldsymbol{m}_i$ its angular momentum in body frame. We consider systems with Hamiltonian (total energy) of the type

$$H(\mathbf{q}, \mathbf{p}, \boldsymbol{m}, \mathbf{Q}) = T(\mathbf{p}, \boldsymbol{m}) + V(\mathbf{q}, \mathbf{Q}), \tag{46}$$

where T refers to the total kinetic energy,

$$T(\mathbf{p}, \boldsymbol{m}) = \sum_i (T_i^{\text{trans}}(\mathbf{p}_i) + T_i^{\text{rot}}(\boldsymbol{m}_i)),$$

consisting of the sum of the translational and rotational kinetic energies of each body,

$$T_i^{\text{trans}}(\mathbf{p}_i) = \frac{\|\mathbf{p}_i\|^2}{2}, \qquad T_i^{\text{rot}}(\boldsymbol{m}_i) = \frac{1}{2}\boldsymbol{m}_i \cdot (I_i^{-1}\boldsymbol{m}_i),$$

(here $I_i = \operatorname{diag}(I_1, I_2, I_3)$ is the inertia tensor of the ith body and $\cdot$ the standard scalar product), and V is a potential energy, describing the interaction between dipoles, that is assumed to depend on the position and orientation only. Furthermore, $V = \sum_{j>i} V_{i,j}$, where $V_{i,j}$ describes the interaction between dipole i and dipole j. We suppose

$$V_{i,j}(\mathbf{q}_i, \mathbf{Q}_i, \mathbf{q}_j, \mathbf{Q}_j) = V_{i,j}^{\text{short}} + V_{i,j}^{\text{dip}},$$

where

$$V_{i,j}^{\text{short}} = 4\epsilon \left(\frac{\sigma}{r_{i,j}}\right)^{12}, \qquad \mathbf{r}_{i,j} = \mathbf{q}_i - \mathbf{q}_j, \quad r_{i,j} = \|\mathbf{r}_{i,j}\|,$$

describes the short range interaction between particles i and j, while

$$V_{i,j}^{\text{dip}} = \frac{1}{r_{i,j}^3}\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j - \frac{3}{r_{i,j}^5}(\boldsymbol{\mu}_i \cdot \mathbf{r}_{i,j})(\boldsymbol{\mu}_j \cdot \mathbf{r}_{i,j}),$$

is the term modeling the dipole interaction, where $\boldsymbol{\mu}_i$ being the orientation of the ith dipole vector. If $\bar{\boldsymbol{\mu}}_i$ is an initial fixed reference orientation for the dipole, then $\boldsymbol{\mu}_i = \mathbf{Q}_i\bar{\boldsymbol{\mu}}_i$.

The Hamiltonian (46) is separable, as the potential does not depend on the position or on the angular momenta. Therefore, we split the system as $H = H_T + H_V$, yielding

$$\begin{array}{rcl} \frac{d}{dt}\mathbf{q}_i &=& \frac{\mathbf{p}_i}{m_i}, \\ \frac{d}{dt}\mathbf{p}_i &=& 0, \\ \frac{d}{dt}\boldsymbol{m}_i &=& \boldsymbol{m}_i \times (I_i^{-1}\boldsymbol{m}_i), \\ \frac{d}{dt}\mathbf{Q}_i &=& \mathbf{Q}_i\widehat{(I_i^{-1}\boldsymbol{m}_i)}, \end{array} \tag{47}$$

and

$$\begin{array}{rcl} \frac{d}{dt}\mathbf{q}_i &=& 0, \\ \frac{d}{dt}\mathbf{p}_i &=& -\frac{\partial V}{\partial \mathbf{q}_i}, \\ \frac{d}{dt}\boldsymbol{m}_i &=& -\operatorname{rot}(\mathbf{Q}_i^\top \frac{\partial V}{\partial \mathbf{Q}_i}) \\ \frac{d}{dt}\mathbf{Q}_i &=& 0. \end{array} \tag{48}$$

Here, rot-function maps matrices to vectors, first by associating to a matrix a skew-symmetric one, and then identifying the latter with a vector,

$$\operatorname{rot}(A) = \operatorname{skew}^{-1}(A - A^T),$$

where $\text{skew}(\mathbf{v}) = \hat{\mathbf{v}}$, see also [21].

In our approach, the original system with full Hamiltonian (46) is then replaced by a composition of the flows of (47) and (48), according to the coefficients of a splitting method. Note that (48) is exactly integrable by a step of the Forward Euler method. This is not the case of (47) because, even though the first equation is,

$$\mathbf{q}_i(t + \Delta t) = \mathbf{q}_i(t) + \Delta t \mathbf{p}_i(t)/m_i,$$

the last two equations are precisely the equations of a free rigid body. In [8], the authors propose to use a Verlet-splitting for the original system (i.e. solve (47) for half time step, then (48) for full time step and then again (47) with half time step) and to split further the rigid-body part. This corresponds to the MR method described earlier in §3.3.1. Here, we will denote the same method by V2+McL2. This is the standard method which is used in several packages for molecular dynamics simulations, for instance the ORIENT package [22].

To obtain methods of order higher than two with the second order splitting scheme V2+McL2, one has use composition rules for self-adjoint scheme described in [26]: these splitting schemes might have large error constants compared to other splitting schemes, whose coefficients are obtained by including extra stages and using the free parameters for minimizing error coefficients [3]. On the other hand, by treating the rigid body part of (47) with our exact methods, we obtain an exact integrator for (47). This is more expensive per step, however, it allows us to use directly higher order splitting methods – in particular, the best one known to-date, see [3, 17]. See the appendix for the coefficients of the splitting methods.

It is important to stress that, for sufficiently large number of particles, using the cheap method McL2 or the more expensive exact method RB for solving the rigid body equations is irrelevant, as the cost of this part grows only linearly with the number of particles. The computationally most demanding part in this simulation, that dominates the cost of the simulation, is the solution of (48), namely the computation of the potential, which grows quadratically with the number of particles (unless treated with more sophisticate methods like fast multipole expansions, but that is beyond the scope of the present work).

This appears clearly in our first example: we compare different splitting methods for a system of 100 particles, for a relatively short time integration ($T = 1$). All the methods use fixed step size, appropriately scaled for each splitting scheme, to require the same number of function evaluations. For the reference method, the V2+McL2, we use step size $h = 10^{-1} \times 1/2^i$, for $i = 0, \ldots, 7$, i.e. for the largest stepsize $h = 0.1$ one has 10 potential evaluations, thus the x-axis in Figure 16 can be interpreted as number of function evaluations as well. Similarly, the sixth-order splitting method S6_{10}+RB, with 10 internal stages requiring potential evaluations, is implemented with stepsize $h = 1$. The results of the simulation are displayed in Figure 16. The figure should be read as follows: there are 3 basic splitting methods V2, SRKN4_6b, S6_{10} (Verlet; a Runge–Kutta–Nyström type splitting of order 4 and 7(6) internal stages with optimized coefficients, effective error $E_f = 0.28$; a partitioned RK splitting of order 6 and 10 internal stages with optimized coefficients, effective error $E_f = 1.12$). The methods are implemented both solving exactly the RB equations (solid line) and using the McL2 method (dash-dotted line). Coalescence of stages is exploited for all methods.

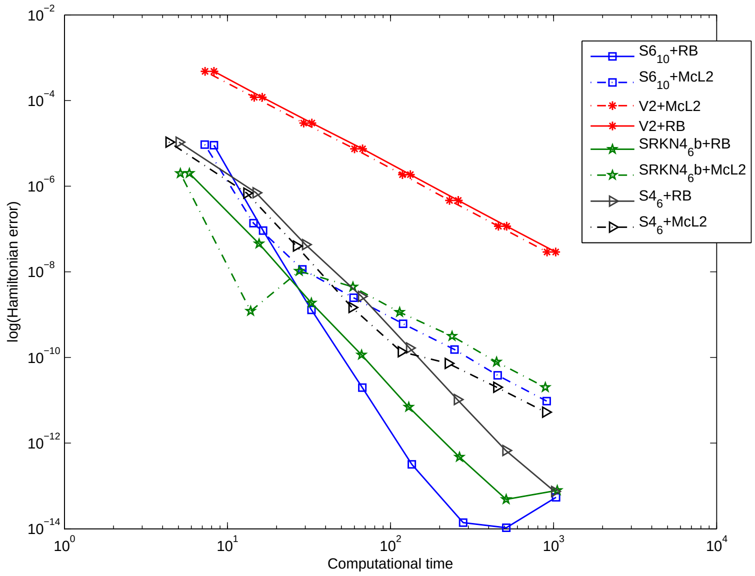

Figure 16: Error in the Hamiltoninan versus computational time for 100 particles. Several splitting methods are compared. See text for details.

The figure indicates that the use of higher order splitting method is indeed favourable. All methods using McL2 will ultimately behave as a second-order method, and the corresponding lines will ultimately be parallel to the lines of the Verlet splitting. Notice however the gain in accuracy: this is due to the fact that the splitting method have very small leading-error coefficients. In particular, the Nyström 4th order scheme is extremely accurate. The methods using the exact RB solutions display the correct order of convergence producing more accurate solutions – at the same computational cost.

The initial conditions for the experiment were taken as follows: $m_i = 1$, $\mathbf{q_i} = N \times \mathtt{randn}(3,1)$, $N = 100$ being the number of particles, and $\mathtt{randn}(3,1)$ a vector with random components (gaussian distribution) between -1 and 1; $\mathbf{p}_i = 0, \boldsymbol{m}_i = 0$, $\mathbf{Q_i}$ random orthogonal matrix, $\boldsymbol{\mu}_i = [0,1,1]^T$, $\sigma = \epsilon = 1$, with a resulting energy $H_0 = 0.141341856111814$. The inertia moments are those of water ($I_1 = 1, I_2 = 1.88, I_3 = 2.88$).

In the next numerical example (Figure 17), we test the same methods for different energies. The initial conditions are chosen as follows: we take 125 particles that we position on a lattice of dimension $5 \times 5 \times 5$. The initial positions are then perturbed by 1% (Gaussian normal distribution). The initial orientations are random orthogonal matrices. With these parameters, we compute the initial energy and then we change the linear momentum of the particles in positions $\mathbf{q}_1 = [1,1,1]^T$ and $\mathbf{q}_{125} = [5,5,5]^T$ to achieve the target energy H_0. For each stepsize $h = 1, 1/2, 1/4, 1/8$ of the basic method $\mathrm{SR6}_{10}$, we perform 100 simulations (choosing every time a different initial condition), and we average the error and the computational time (arithmetic mean). We observe that for small energies, the splitting methods using the exact RB integrator perform very well. For higher energies, the leading error term in the splitting comes from V^{short}, and the effect of having an exact integrator for the RB part is less relevant.

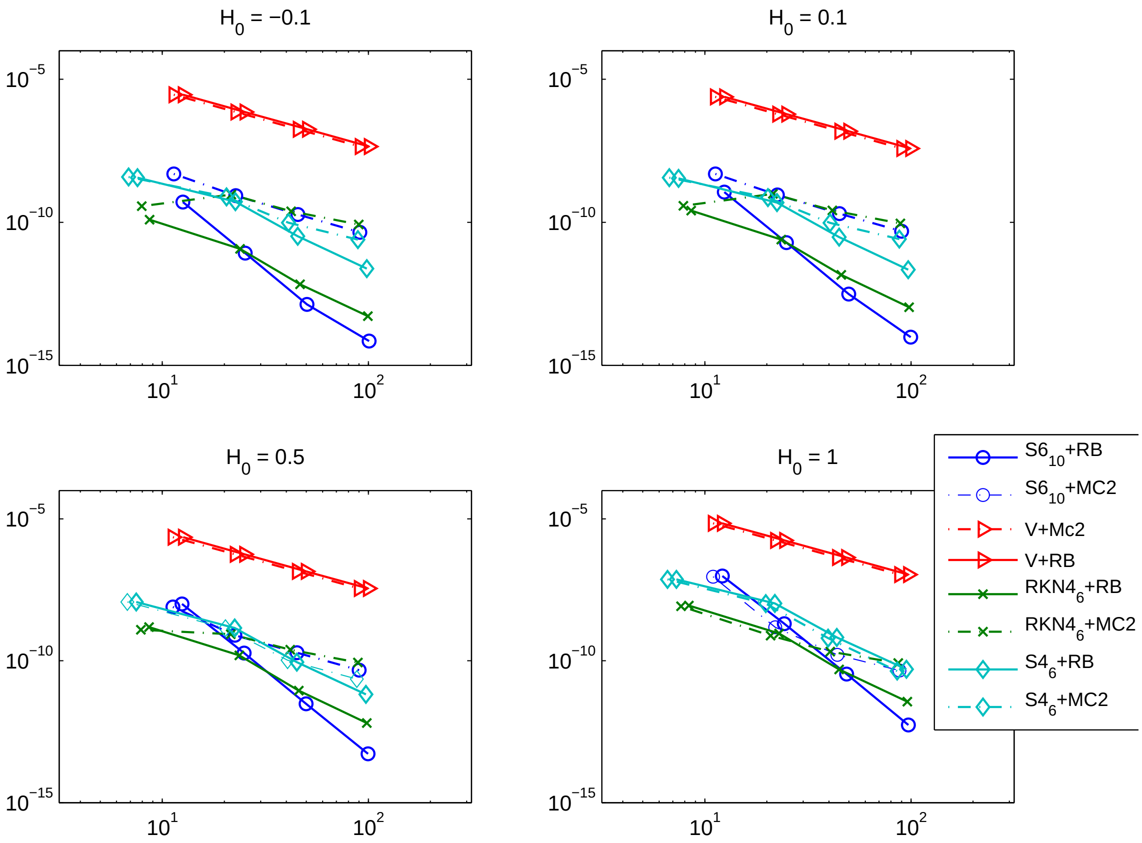

Figure 17: Average errors for different values of the energy H_0, 100 runs per each of the stepsizes $1, 1/2, 1/4, 1/8$. Number of particles $N = 125$. For small energy values, the splitting methods based on the exact RB integrator perform better than those with the MC2 splitting. For higher values of the energy, the error of due to the splitting is much higher than the error for the RB-part and it dominates the total error

Finally, in Figure 18, the method SRKN4_6b is compared to another Nyström splitting schemes, this time of order 6 and 14 stages, SRKN6^a_{14}. The number of function evaluations is the same. The initial conditions as before, except for the number of averages (which is 1), and the time of integration, with $T = 10$.

It is worthwhile to mention that the RB integrator that we have used in these experiments is algorithm based on rotations described in §2.2. The main cost of the method is the computation of the elliptic integral of the third kind. The latter could be approximated at a much lower cost using Gaussian quadrature. Similarly, one could use other RB methods, like those based on quaternions (like the exact one presented in this paper, or the methods of Hairer and Vilmart, that instead produce approximate solutions). The resulting lines would then lie somewhere between the solid and dash-dotted lines of the same colors in fig. 17-18, but this is less relevant, as the overall cost of the implementation lies in the computation of the potential part, which is the same for all the implementations. One marginal advantage of the quaternion representation is the storage, as quaternions require only 4 elements to store an orthogonal matrix, instead of 9; thus, the number of variables to store for each particle is reduced from 24 to 19.

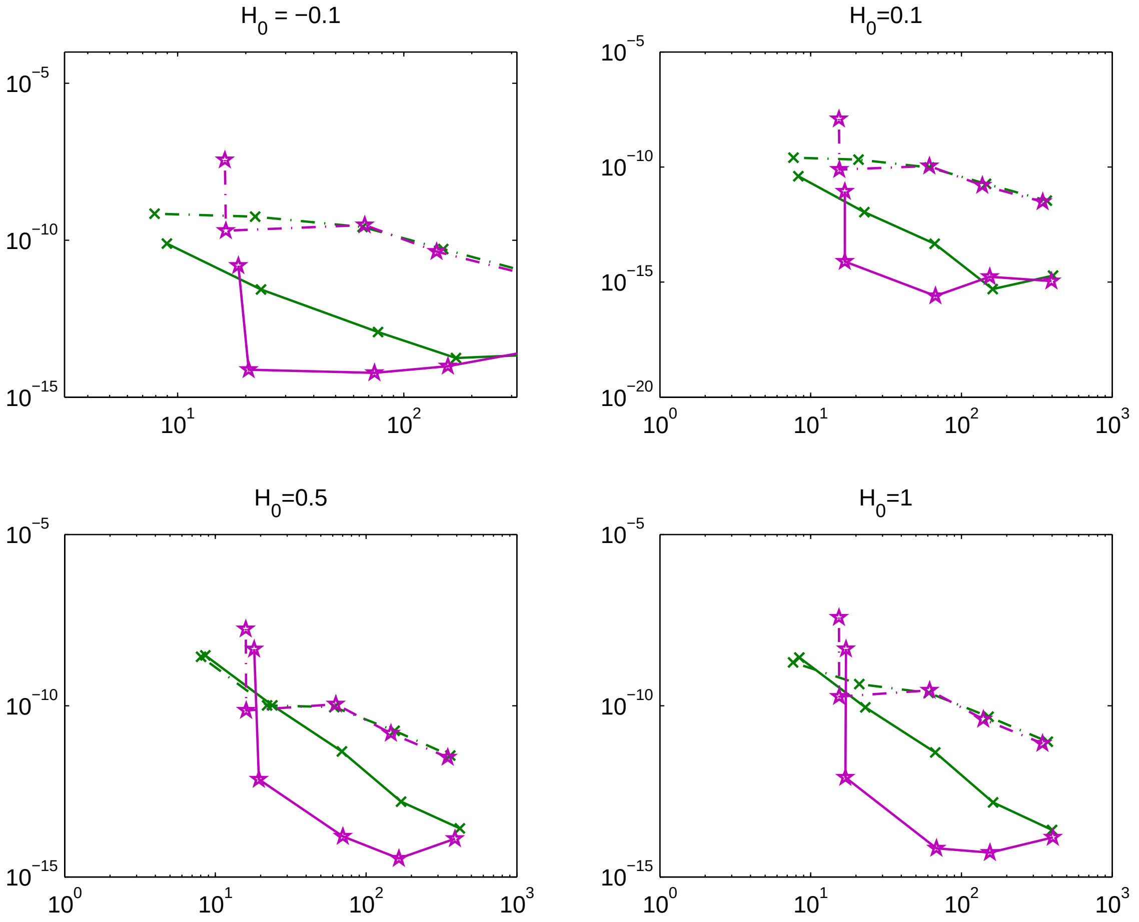

Figure 18: Comparison of two RKN splittings of order 4 and 6, on the interval [0,10], 125 particles, for some initial conditions. The sharp increase of the error for the 6th order method is due to the fact that the stepsize is greater than one. The green method is the same as in figure 2.

4 Conclusions

The main purpose of this paper has been to understand whether and when methods employing the exact solution of the free rigid body equations could compete with state of the art geometric integrators. As the exact solution of the momentum equations has been discussed in the literature before, we have focussed on the computation of the attitude rotation. We have presented two concrete approaches, based on rotation matrices and quaternions, and we have shown how other formulations of the solution fit into our framework. Thereafter, we have considered the implementation of the exact and *semi-exact* methods discussed in this paper and we have tested them thoroughly for several problems.

We have found out that the exact methods, though more expensive, are very robust and behave uniformly well for all choices of the principal moments of inertia and initial conditions, independently of the step-size of integration.

If cost is an issue, *semi-exact* methods are a good compromise. They are much cheaper than the exact ones, while sharing all the geometric properties and being robust for large step-sizes and arbitrary values of the principal moments of inertia. This is an advantage with respect to implicit methods using fixed-point iteration, that might require small step-sizes to converge.

Our conclusion is that the implementation of the exact solution of the free rigid body is competitive as a numerical approach.

The numerical exact solution of the free rigid body equations is of interest as it can be used as a building block for splitting methods of high order. The main argument is that one would like to use step-sizes as large as possible to reduce the number of force evaluations. This property is appealing in several important applications, like molecular dynamics simulations, where other aspects (like force evaluation) are the computationally heavy part of the problem.

Acknowledgments

The authors would like to thank E. Hairer, B. Carlson, E. Karatsuba for useful discussion and comments. We acknowledge the kind hospitality and support of the Newton Institute of Mathematical Sciences in Cambridge UK. Special thanks to A. Iserles.

Appendices

Coefficients of the Gauss quadrature

For completeness, we report the coefficients of the Gaussian quadrature of order 10 shifted to the interval $[0, 1]$.

$$\begin{aligned} a_1 &= 0.04691007703067 & b_1 &= 0.11846344252809 \\ a_2 &= 0.23076534494716 & b_2 &= 0.23931433524968 \\ a_3 &= 0.5 & b_3 &= 0.28444444444444 \\ a_4 &= 0.76923465505284 & b_4 &= b_2 \\ a_5 &= 0.95308992296933 & b_5 &= b_1. \end{aligned} \tag{49}$$

For the qudrature of order 6 and 8 the coefficients have closed form and can be found for instace in [1].

Coefficients of the splitting schemes

Given the differential equation

$$y' = F(y) = A(y) + B(y),$$

denote by $\varphi_\tau^{[F]}$ the flow of the vector-field F from time t to time $t+\tau$. Given a numerical approximations $y^{(j)} \approx y(t_j)$, we consider symmetric splitting schemes of the type

$$y^{(j+1)} = \varphi_{a_1 h}^{[A]} \circ \varphi_{b_1 h}^{[B]} \circ \varphi_{a_2 h}^{[A]} \circ \cdots \circ \varphi_{a_{m+1} h}^{[A]} \circ \cdots \varphi_{b_1 h}^{[B]} \circ \varphi_{a_1 h}^{[A]}\, y^{(j)},$$

where $h = t_{j+1} - t_j$. A typical splitting is obtained separating the contributions arising from the from kinetic (A) and potential (B) energy of the system. For this reason, the (twice) the number s of the coefficients b_i is called the *stage number* of the splitting method. The effective error is defined as $E_f = s\sqrt[p]{||\mathbf{c}||_2}$, where $\mathbf{c}$ is the vector of coefficients of the elementary differentials of the leading error term and p is the order of the method. We refer to [3, 17] for background and notation.

In this notation, the Störmer–Verlet scheme V2 has coefficients

$$a_1 = 1/2, \qquad b_1 = 1, \tag{50}$$

(order 2, one stage).

For completeness, we report the coefficients of the methods used in this paper: S6$_{10}$ method (order 6, 10 stages, effective error $E_f = 1.12$)

$$\begin{aligned}
a_1 &= 0.0502627644003922, & b_1 &= 0.148816447901042,\\
a_2 &= 0.413514300428344, & b_2 &= -0.132385865767784,\\
a_3 &= 0.0450798897943977, & b_3 &= 0.067307604692185,\\
a_4 &= -0.188054853819569, & b_4 &= 0.432666402578175,\\
a_5 &= 0.541960678450780, & b_5 &= 1/2 - (b_1 + \cdots + b_4),\\
a_6 &= 1 - 2(a_1 + \cdots + a_5).
\end{aligned} \tag{51}$$

In the implementation, the A part (coefficients a_i) corresponds to (47) while the B part (b_i) corresponds to (48). This is a general splitting and the A and B part are interchangeable. Similarly, for $S4_6$ (order 4, 6 stages, effective error $E_f = 0.56$), we have

$$\begin{aligned}
a_1 &= 0.07920369643119565, & b_1 &= 0.209515106613362,\\
a_2 &= 0.353172906049774, & b_2 &= 0.143851773179818,\\
a_3 &= -0.04206508035771952, & b_3 &= 1/2 - (b_1 + b_2),\\
a_4 &= 1 - 2(a_1 + a_2 + a_3).
\end{aligned} \tag{52}$$

For SRKN4$_6^b$ (order 4, (7)6 stages, effective error $E_f = 0.28$)

$$\begin{aligned}
b_1 &= 0.0829844064174052, & a_1 &= 0.245298957184271,\\
b_2 &= 0.396309801498368, & a_2 &= 0.604872665711080,\\
b_3 &= -0.0390563049223486, & a_3 &= 1/2 - (a_1 + a_2),\\
b_4 &= 1 - 2(b_1 + b_2 + b_3)
\end{aligned} \tag{53}$$

but for this last method, the coefficients for the A and B part are not interchangeable. Also for the method SRKN6^a_{14} (order 6, 14 stages, effective error $E_f = 0.63$)

$$\begin{aligned}
a_1 &= 0.0378593198406116, & b_1 &= 0.09171915262446165,\\
a_2 &= 0.102635633102435, & b_2 &= 0.183983170005006,\\
a_3 &= -0.0258678882665587, & b_3 &= -0.05653436583288827,\\
a_4 &= 0.314241403071477, & b_4 &= 0.004914688774712854,\\
a_5 &= -0.130144459517415, & b_5 &= 0.143761127168358,\\
a_6 &= 0.106417700369543, & b_6 &= 0.328567693746804,\\
a_7 &= -0.00879424312851058, & b_7 &= 1/2 - (b_1 + \cdots + b_6)\\
a_8 &= 1 - 2(a_1 + \cdots + a_7)
\end{aligned} \tag{54}$$

the stages A and B are not interchangeable.

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series, 55.* Reprint of the 1972 edition. Dover Publications, Inc., New York, 1992.

[2] P. E. Appell. *Traité de mécanique rationnelle*, volume 2. Gauthier Villars, Paris, 1924/26.

[3] S. Blanes and P. C. Moan. Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods. *J. Comp. Appl. Math.*, 142(2):313–330, 2002.

[4] P.F. Byrd and M.D. Friedman. *Handbook of elliptic integrals for engineers and scientists.* Die Grundlehren der mathematischen Wissenschaften, Band 67. Springer-Verlag, New York-Heidelberg, second edition edition, 1971.

[5] E. Celledoni and N. Säfstöm. Efficient time-symmetric simulation of torqued rigid bodies using Jacobi elliptic functions. *Journal of Physics A*, 39:5463–5478, 2006.

[6] R. Cushman. No polar coordinates. Lecture notes, MASIE summer school, Peyresq, France, September 2-16, 2000.

[7] R. H. Cushman and L. Bates. *Global aspects of classical integrable systems.* Birkhauser, Basel, 1997.

[8] A. Dullweber, B. Leimkuhler, and R. McLachlan. Symplectic splitting methods for rigid body molecular dynamics. *J. Chem. Phys.*, 107:5840–5851, 1997.

[9] M. Euler. Decouverte d'un nouveau principe de mecanique. *Memoires de l'academie des sciences de Berlin*, 6:185–217, 1752.

[10] M. Euler. De motu corporum circa punctum fixum mobilium. In *Opera postuma*, number 2, pages 46–62. 1862.

[11] Francesco Fassò. Comparison of splitting algorithm for the rigid body. *J. Comput. Phys.*, 189(2):527–538, 2003.

[12] E. Hairer and G. Vilmart. Preprocessed discrete Moser-Veselov algorithm for the full dynamics of a rigid body. *J. Phys. A*, 39:13225–13235, 2006.

[13] C.G.J. Jacobi. Sur la rotation d'un corps. *Crelle Journal für die reine und angewandte Matematik*, Bd. 39:293–350, 1849.

[14] I. I. Kosenko. Integration of the equations of a rotational motion of rigid body in quaternion algebra. The Euler case. *J. Appl. Maths Mechs*, 62(2):193–200, 1998.

[15] M. Leok, T. Lee, and N.H. McClamroch. Attitude maneuvers of a rigid spacecraft in a circular orbit. In *Proc. American Control Conf.*, 2005. Submitted.

[16] R. I. McLachlan and A. Zanna. The discrete Moser–Veselov algorithm for the free rigid body, revisited. *Found. of Comp. Math.*, 5(1):87–123, 2005.

[17] Robert I. McLachlan and G. Reinout W. Quispel. Splitting methods. *Acta Numer.*, 11:341–434, 2002.

[18] J.Wm. Mitchell. *A simplified variation of parameters solution for the motion of an arbirarily torqued mass asymmetric rigid body.* PhD thesis, University of Cincinnati, 2000.

[19] Harold S. Morton, Jr., John L. Junkins, and Jeffrey N. Blanton. Analytical solutions for Euler parameters. *Celestial Mech.*, 10:287–301, 1974.

[20] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in Fortran 77 and Fortran 90.* Cambridge University Press, Cambridge, 1996. The art of scientific and parallel computing, Second edition diskette v 2.06h.

[21] S. Reich. Symplectic integrators for systems of rigid bodies. Integration algorithms and classical mechanics (Toronto, ON, 1993). *Fields Inst. Commun.*, 10:181–191, 1996.

[22] A. J. Stone, A. Dullweber, M. P. Hodges, P. L. A. Popelier, and D. J. Wales. ORIENT Version 3.2: A program for studying interaction between molecules.

[23] R. van Zon and J. Schofield. Numerical implementation of the exact dynamics of free rigid bodies. *J. of Comput. Phys.*, 2007. To appear.

[24] R. van Zon and J. Schofield. Symplectic algorithms for simulations of rigid body systems using the exact solution of free motion. *Phys. Rev. E*, 75, 2007.

[25] E. T. Whittaker. *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies.* Cambridge University Press, 4th edition, 1937.

[26] H. Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150:262–268, 1990.