Reducing round-off errors in rigid body dynamics

Gilles Vilmart [1]

INRIA Rennes, ENS Cachan Bretagne, avenue Robert Schuman, 35170 Bruz, France
Université de Genève, Section de mathématiques, 2-4 rue du Lièvre, CP 64, 1211 Genève 4, Switzerland

Abstract

In several recent publications, numerical integrators based on Jacobi elliptic functions are proposed for solving the equations of motion of the rigid body. Although this approach yields theoretically the exact solution, a standard implementation shows an unexpected linear propagation of round-off errors. We explain how deterministic error contribution can be avoided, so that round-off behaves like a random walk.

Key words: rigid body integrator, Jacobi elliptic functions, probabilistic error propagation, long-time integration, compensated summation, quaternion, Discrete Moser–Veselov algorithm.
PACS: 02.60.Jh, 45.10.-b, 02.70.Ns, 45.40.-f

1. Introduction

There exists a large choice of numerical integrators for solving the equations of motion of the rigid body, mainly based on splitting methods (see e.g. [5, Sect. VII.5]), and recently a high-order modification of the Discrete-Moser-Veselov algorithm [7].

In several recent publications [3,2,13,14], it is proposed to integrate the equations of motion of the free rigid body analytically, using the Jacobi elliptic functions [10]. Although this approach yields the exact solution, a standard implementation yields an unexpected linear propagation (accumulation) of round-off errors (see Figure 1 and [2, Fig. 1]). The aim of this article is to analyze this propagation of rounding errors and explain how it can be reduced, to retrieve the optimal probabilistic error growth model, as developed in [8]. We focus on the conservation of the first integrals of the system.

Email address: `Gilles.Vilmart@math.unige.ch` (Gilles Vilmart).
URL: `http://www.unige.ch/~vilmart` (Gilles Vilmart).

[1] This work was partially supported by the Fonds National Suisse, project No. 200020-109158.

This article is organized as follows. Section 2 recalls the rigid body equations of motion. In section 3, we study the propagation of round-off errors for a standard implementation of the algorithm based on Jacobi elliptic functions. Then, we present an implementation of this algorithm that allows to reduce the effect of round-off errors, both qualitatively and quantitatively (Sect. 4). Finally (Sect. 5), we compare the results with the preprocessed DMV algorithm [7], which has a suitable form to apply compensated summation and reduce round-off errors, for an accurate long-term integration.

2. Equations of motion and first integrals

The motion of a rigid rigid body, relative to a fixed coordinate system, is determined by a Hamiltonian system constrained to the Lie group $SO(3)$ (see [5, Sect. VII.5]). In the absence of an external potential, the Euler equations of motion of the free rigid body are

$$\dot{y}_1 = (I_3^{-1} - I_2^{-1})y_2y_3, \; \dot{y}_2 = (I_1^{-1} - I_3^{-1})y_3y_1, \; \dot{y}_3 = (I_2^{-1} - I_1^{-1})y_1y_2 \tag{1}$$

where the vector $y(t) = (y_1(t), y_2(t), y_3(t))^T$ is the angular momentum, and the constants $I_1, I_2, I_3 > 0$ are the three moments of inertia. The orientation of the rigid body, relative to a fixed coordinate system, is then represented by an orthogonal matrix $Q(t)$ satisfying

$$\dot{Q} = Q \begin{pmatrix} 0 & y_3/I_3 & -y_2/I_2 \\ -y_3/I_3 & 0 & y_1/I_1 \\ y_2/I_2 & -y_1/I_1 & 0 \end{pmatrix}. \tag{2}$$

The flow of (1)-(2) exactly conserves the energy and the angular momentum relative to the fixed frame. This means that Qy and

$$C(y) = \frac{1}{2}\Big(y_1^2 + y_2^2 + y_3^2\Big) \qquad \text{and} \qquad H(y) = \frac{1}{2}\Big(\frac{y_1^2}{I_1} + \frac{y_2^2}{I_2} + \frac{y_3^2}{I_3}\Big)$$

(Casimir and Hamiltonian) are first integrals of the system.

In the historical article [10], Jacobi derived the analytic solution for the motion of a free rigid body and defined to this aim the so-called 'Jacobi analytic functions' as

$$\mathrm{sn}(u,k) = \sin(\varphi), \quad \mathrm{cn}(u,k) = \cos(\varphi), \quad \mathrm{dn}(u,k) = \sqrt{1 - k^2\sin^2(\varphi)}, \tag{3}$$

where the Jacobi amplitude $\varphi = \mathrm{am}(u,k)$ is defined implicitly by an elliptic integral of the first kind.

3. Standard implementation

We consider here the numerical algorithm based on Jacobi elliptic functions as proposed in [3,2,13,14], and we focus on the numerical resolution of the Euler equations (1), see e.g. Proposition 2.1 in [2].

Algorithm 3.1 Assume $I_1 \leq I_2 \leq I_3$ and $y(t_0)$ is not a saddle point. Consider the quantities

$$a_1 = \sqrt{2H(y)I_3 - 2C(y)} \qquad a_3 = \sqrt{2C(y) - 2H(y)I_1},$$

which are conserved along time. To simulate the presence of an external potential, they are recalculated before each step. For $(I_2 - I_1)a_1^2 \le (I_3 - I_2)a_3^2$, the solution of the Euler equations at time $t = t_0 + h$ is

$$y_1(t) = b_1 a_1 \mathrm{cn}(u,k), \quad y_2(t) = b_2 a_1 \mathrm{sn}(u,k), \quad y_3(t) = b_3 a_3 \delta \mathrm{dn}(u,k),$$

where $\delta = \mathrm{sign}(y_3) = \pm 1$ and

$$b_1 = \sqrt{I_1/(I_3 - I_1)}, \quad b_2 = \sqrt{I_2/(I_3 - I_2)}, \quad b_3 = \sqrt{I_3/(I_3 - I_1)}.$$

Here, $\mathrm{cn}(u,k)$, $\mathrm{sn}(u,k)$ and $\mathrm{dn}(u,k)$ are the Jacobi elliptic functions (3) with modulus k and parameter u,

$$k^2 = b_0 a_1^2/a_3^2, \quad b_0 = (I_2 - I_1)/(I_3 - I_2), \quad u = h\delta\sqrt{(I_3 - I_2)/(I_1 I_2 I_3)}a_3 + \nu,$$

where ν is a constant of integration (see [3, Sect. 3] for details). Similar formulas hold for $(I_2 - I_1)a_1^2 \ge (I_3 - I_2)a_3^2$.

Notice that round-off errors in the computation of u and φ for the Jacobi elliptic functions (3) have no influence on the preservation of first integrals, because it can be interpreted as a time transformation.

3.1. *Numerical experiments*

In all numerical experiments, we consider the following initial condition with norm 1,

$$y_1(0) = 0.5, \quad y_2(0) = 0.2, \quad y_3(0) = \sqrt{1 - y_1(0)^2 - y_2(0)^2} \tag{4}$$

and integrate on the interval of time $[0, 10^4]$ with stepsize $h = 0.01$ (one million steps). We consider a rigid body with moment of inertia $I_1 = 0.345, I_2 = 0.653, I_3 = 1.0$, which corresponds to the water molecule, as considered in [4]. The angular momentum $y(t)$ is a periodic function of time (in the absence of an external potential), and we integrate over about 822 periods. We also tried many different initial values and moments of inertia, and numerical results where similar to those presented in this paper.

The algorithm based on Jacobi elliptic functions is fully explicit and no iterative solution of non-linear equations is involved (excepted the code for computing Jacobi elliptic functions). However, the standard implementation (Algorithm 3.1) shows a linear growth of round-off errors (see left picture in Figure 1). The error for the Hamiltonian is about 1.25×10^{-17} per step, or $0.056 \times eps$ per step, where $eps = 2^{-52}$ is the machine precision. The error is a superposition of a small statistical error and a deterministic error which grows linearly with time, due to a tiny non-zero bias in the pattern of positive and negative rounding errors.

In [6], it is shown that for implicit Runge-Kutta methods, the use of rounded coefficients a_{ij} and b_j induces a systematic error in long-time integrations. Here, the situation is similar, because there are many constants involved: $b_0, b_1, b_2, b_3, I_1, I_3, \ldots$ The same rounded coefficients are used along the numerical integration, and this induces a deterministic error which propagates linearly with time.

To reduce round-off errors in the Jacobi elliptic functions based algorithm, our first idea was to compute all above constants in quadruple-precision arithmetic, and then make all corresponding multiplications in quadruple-precision. Alternatively, we explain in the next section how round-off errors can be reduced using only standard double-precision arithmetic.

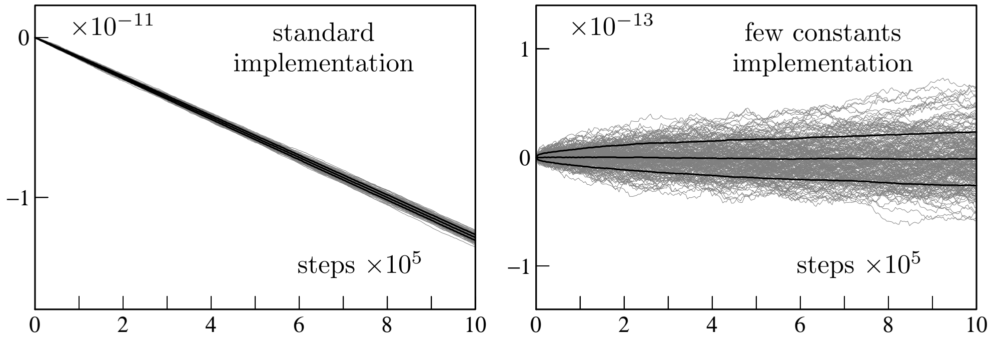

Fig. 1. Hamiltonian errors for the integrators based on Jacobi elliptic functions. One million steps with stepsize $h = 0.01$. Left picture: standard implementation (Algorithm 3.1). Right picture: new implementation (Algorithm 4.1). The plots show the error as a function of time for 200 initial values (with norm 1) randomly chosen close to the one in (4). The mean as a function of time and the standard deviation over all 1000 trajectories are included as bold curves.

3.2. *Probabilistic explanation of the Error Growth.*

The long-time behavior of round-off errors can be explained using probability, as developed in the classical book of Henrici [8]. It is often called Brouwer's law [1] in celestial mechanics, see also [5, Section VIII.5]. The error contribution over one step in the Hamiltonian $H(y)$ (and similarly for the other invariants) can be interpreted as a sequence of independent random variables

$$H(y_{n+1}) - H(y_n) = \varepsilon_n$$

with variance $\mathrm{Var}(\varepsilon_n)$ proportional to the square of the round-off unit *eps* of the computer.

If the mean average of the ε_n's is different from zero, due to a deterministic error source, then the round-off errors accumulate linearly (see left picture in Figure 1).

Under the assumption that the mean of all ε_n is zero, then the sum for N steps of the ε_n's is a random variable with mean zero and variance proportional to $N\,eps^2$. This shows that the error err_N in the Hamiltonian after N steps grows like (random walk)

$$\mathrm{Var}(err_N)^{1/2} = \sigma\, eps\, \sqrt{N}$$

for some constant σ (e.g. $\sigma \approx 0.11$ in right picture of Figure 1).

4. Reducing round-off errors

In this section, we present a modification of Algorithm 3.1 which makes round-off behave like a random walk (see right picture of Figure 1). The idea is to reduce the number of constants involved, so that, in the spirit of backward error analysis, all constants can be interpreted as exact values corresponding to modified moments of inertia. We show that this can be achieved with Algorithm 4.1 which uses only two independent constants c_1 and λ defined below.

Algorithm 4.1 Consider the constants (we still assume $I_1 \le I_2 \le I_3$),

$$c_1 = \frac{I_1(I_3 - I_2)}{I_2(I_3 - I_1)}, \qquad c_2 = 1 - c_1, \tag{5}$$

and the quantities

$$d_1 = \sqrt{y_1^2 + c_1 y_2^2}, \qquad d_3 = \sqrt{c_2 y_2^2 + y_3^2},$$

(recalculated before each step). For $c_2 d_1^2 \leq c_1 d_3^2$, the solution of the Euler equations at time $t = t_0 + h$ is

$$y_1(t) = d_1 \mathrm{cn}(u,k), \quad y_2(t) = d_2 \mathrm{sn}(u,k), \quad y_3(t) = \delta\sqrt{d_3^2 - c_2 y_2(t)^2},$$

where $d_2 = \sqrt{y_1^2/c_1 + y_2^2}$ and $d_3^2 = c_2 y_2^2 + y_3^2$. Here, $\mathrm{cn}(u,k)$, $\mathrm{sn}(u,k)$ are the Jacobi elliptic functions (3) with

$$k^2 = (c_2 d_1^2)/(c_1 d_3^2), \quad u = \delta h \lambda d_3 + \nu, \quad \lambda = \sqrt{(I_3 - I_2)(I_3 - I_1)/(I_1 I_2 I_3^2)},$$

$\delta = \mathrm{sign}(y_3) = \pm 1$, and ν is a constant of integration. We have similar formulas for $c_2 d_1^2 \geq c_1 d_3^2$.

It is essential in (5) that the identity $c_1 + c_2 = 1$ holds exactly. This can be done as follows:

$$\text{compute } c_1;$$

$$c_2 = 1 - c_1;$$

$$c_1 = 1 - c_2;$$

It makes c_1 and c_2 have a floating point arithmetic representation with the same exponent.

5. Compensated summation

Unlike the algorithm based on Jacobi elliptic functions, the Preprocessed Discrete Moser-Veselov algorithm [7] requires for solving the Euler equations (1) the computation of a recursion of the form

$$y_{n+1} = y_n + \delta_n$$

where the increment δ_n has size $\mathcal{O}(h)$. It is thus possible to apply the so-called 'compensated summation' algorithm due to [11,12] for reducing round-off errors in floating point arithmetic. A famous analysis and presentation is given in [9] (see [5, VIII]).

Applying compensated summation allows to compute the above recursion in high-precision and thus to reduce by a factor h the effect of round-off error, as illustrated in bottom pictures in Figure 2. Notice that we do not lose information if we do not normalize to 1 the quaternions $q_n, n = 0, 1, 2, \ldots$ representing rotation matrices. This allows to apply compensated summation also for the attitude $Q(t)$ (2) of the rigid body (see bottom right picture in Figure 2). The round-off errors in the preservation of invariants $H(y), C(y)$ and Qy now grow like $eps\, h\sqrt{N}$, or equivalently $eps\, h^{1/2} t^{1/2}$ where $t = nh$.

The idea of the Preprocessed DMV algorithm is to apply the standard DMV algorithm (order 2) with modified values of moments of inertia $\widetilde{I}_1, \widetilde{I}_2, \widetilde{I}_3$, which depend on initial conditions only via the conserved quantities $C(y)$ and $H(y)$. They are given by formal series expansion in powers of the stepsize, and truncating theses series yields numerical integrators of arbitrarily high-order that preserves all first integrals. In the numerical implementation, the only constants involved are the modified moments of inertia $\widetilde{I}_1, \widetilde{I}_2, \widetilde{I}_3$, and we avoid using other constants in the numerical implementation. The DMV algorithm shows the correct behavior (see Figure 2). As recommended in [6], the fixed point iteration

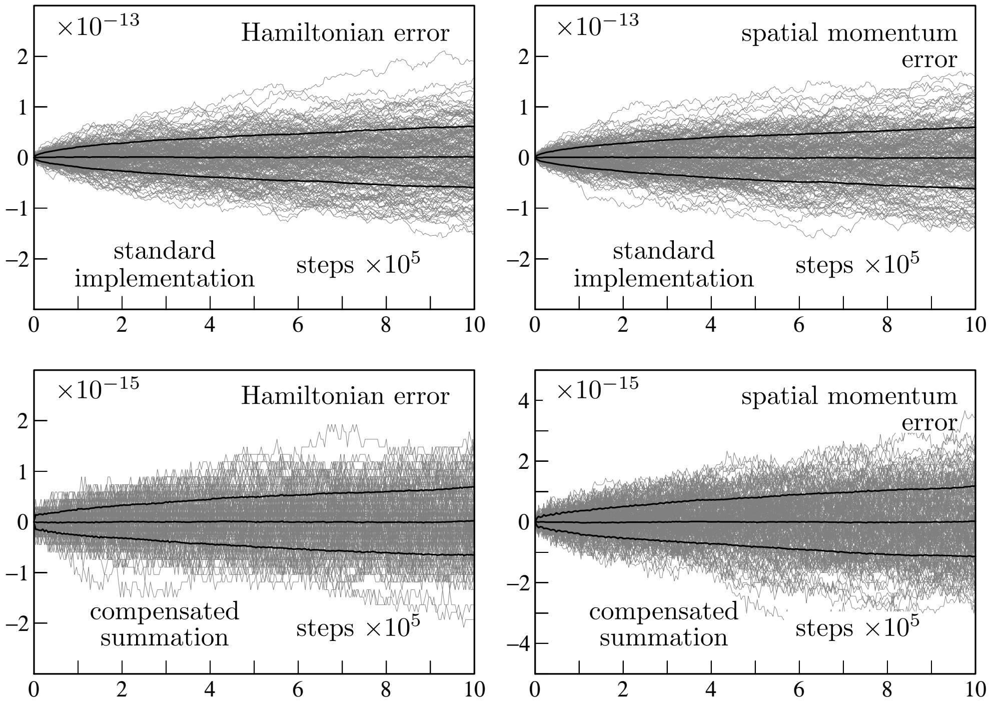

Fig. 2. Discrete Moser-Veselov algorithm of order 10. Roundoff errors in Hamiltonian and spatial momentum (first component of Qy) for 200 initial values randomly chosen close to the one in (4). One million steps with stepsize $h = 0.01$. Top pictures: standard implementation. Bottom pictures: compensated summation. The average as a function of time and the standard deviation over all 1000 trajectories are included as bold curves.

is performed until convergence: the stopping criterion is $\Delta^{(k)} = 0$ or $\Delta^{(k)} > \Delta^{(k-1)}$ which indicates that the increments $\Delta^{(k)}$ of the iteration starts to oscillate due to round-off.

Acknowledgement. The author is grateful to Philippe Chartier and Ernst Hairer for helpful discussions, and thanks the participants of the 'ARMOR–IPSO seminar' in Saint-Malo (January 2008) for stimulating comments.

References

[1] D. Brouwer. On the accumulation of errors in numerical integration. *Astronomical Journal*, 46:149–153, 1937.

[2] E. Celledoni, F. Fassò, N. Säfström, and A. Zanna. The exact computation of the free rigid body motion and its use in splitting methods. *to appear in SIAM J. Sci. Comp.*, 2007.

[3] E. Celledoni and N. Säfström. Efficient time-symmetric simulation of torqued rigid bodies using Jacobi elliptic functions. *J. Phys. A*, 39:5463–5478, 2006.

[4] F. Fassò. Comparison of splitting algorithm for the rigid body. *J. Comput. Phys.*, 189:527–538, 2003.

[5] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations.* Springer Series in Computational Mathematics 31. Springer-Verlag, Berlin, second edition, 2006.

[6] E. Hairer, R.I. McLachlan, and A. Razakarivony. Achieving Brouwer's law with implicit Runge-Kutta methods. *to appear in BIT*, 2008.

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

[8] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations.* John Wiley & Sons Inc., New York, 1962.

[9] N. J. Higham. The accuracy of floating point summation. *SIAM J. Sci. Comput.*, 14:783–799, 1993.

[10] C. G. J. Jacobi. Sur la rotation d'un corps. *Journal für die reine und angewandte Matematik (Journal de Crelle)*, 39:293–350, publ. 1850 (lu dans la séance du 30 juillet 1849 à l'académie des sciences de Paris).

[11] W. Kahan. Further remarks on reducing truncation errors. *Comm. ACM*, 8:40, 1965.

[12] O. Møller. Quasi double-precision in floating point addition. *BIT*, 5:251–255, 1965.

[13] R. van Zon and J. Schofield. Numerical implementation of the exact dynamics of free rigid bodies. *J. Comput. Phys.*, 225(1):145–164, 2007.

[14] R. van Zon and J. Schofield. Symplectic algorithms for simulations of rigid body systems using the exact solution of free motion. *Phys. Rev. E*, 75:056701, 2007.