REPORTS
IN
INFORMATICS

ISSN 0333-3590

The Discrete Moser–Veselov algorithm for the free Rigid Body, revisited

R. I. McLachlan and A. Zanna

REPORT NO 255 **October 2003**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL `http://www.ii.uib.no/publikasjoner/texrap/pdf/2003-255.pdf`

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at `http://www.ii.uib.no/publikasjoner/texrap/`.

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

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

The Discrete Moser–Veselov algorithm for the free Rigid Body, revisited

Robert I. McLachlan* and Antonella Zanna †

23rd October 2003

Abstract

In this paper we revisit the Moser–Veselov description for the free Rigid Body, which, in the 3×3 case, can be implemented as an *explicit, second order, integrable* approximation of the continuous solution. By backward error analysis, we study the modified vector field which is integrated exactly by the discrete algorithm. We deduce that the discrete Moser–Veselov (DMV) is well approximated to higher order by time-reparametrizations of the continuous equations (modified vector field). We use the modified vector field to preprocess the initial data to the DMV and show the equivalence of the DMV algorithm and the RATTLE algorithm. Numerical integration with these preprocessed initial data is several order of magnitude more accurate of the original DMV and RATTLE approach.

1 Introduction

In 1991 Moser and Veselov published a memorable paper (Moser & Veselov 1991) in which they described discrete versions of several classical integrable systems, among which, the Rigid Body (RB). The integrability of these discrete maps was shown with the help of a Lax-pair representation corresponding to a factorization of certain matrix polynomials.

There is a large interest of the computational community towards good methods for the integration of the RB equations, since they arise naturally in a number of applications, for instance celestial mechanics and molecular dynamics. RB equations are used here to follow particles (planets, atoms, molecules, etc.) in between other body-body interactions. It is of fundamental importance that some qualitative features of the system under consideration are preserved under integration, for instance energy. However, in the recent years it has become clear that numerical preservation of energy alone is not enough to obtain 'good' qualitative descriptions of the system. Other properties like symplecticity and time-reversibility have been shown to produce favourable propagation of errors, hence better numerical methods, especially when interested over long time behaviour.

Another interesting aspect of the work of Moser and Veselov is that their approach is based on the theory of *discrete Lagrangians*, which has become emergent in the computational mechanics community (see for instance (Marsden & West 2001) and references therein) because the methods derived by this approach naturally inherit the symplectic structure of the system under consideration, they preserve the discrete Lagrangian, preserve momentum and preserve energy. However, this approach has also some major drawbacks: *i)* it is difficult to obtain numerical methods of order higher than two, *ii)* it is usually difficult to estimate the numerical error, and *iii)* the derived methods (with some few exceptions) are highly implicit, hence computationally expensive. In facts, the DMV algorithm for the 3×3 RB is an exception, since, as we shall see in the sequel, it can be implemented as an *explicit* numerical method.

*Institute of Fundamental Sciences, Massey University, Private Bag 11-222, Palmerston North, New Zealand. Email: r.mclachlan@massey.ac.nz

†Institutt for informatikk, Universitet i Bergen, Høyteknologisenteret N-5020 Bergen, Norway. Email: anto@ii.uib.no

Motivated by this, in this paper we revisit the DMV algorithm for the RB equations and analyse it as a numerical algorithm for the their time integration. The original DMV is a second order method, but, unfortunately, it has a large error constant, so that it is not competitive with other numerical algorithms (like the splitting method of McLachlan (1993), which is second-order, explicit, reversible and norm-preserving, or the Implicit Midpoint Rule (IMR), which is second order, norm- and energy-preserving for this problem but implicit, though remarkably accurate) which are nowadays widely used for the integration of the RB.

Using tools dear to numerical analysis, like *backward error analysis* (BEA), we are able to construct a modified vector field which is integrated by the DMV up to order 6 and describe a general procedure for obtaining higher order approximations. The study of the modified vector field for the 3×3 RB reveals that the DMV algorithm solves a time-reparametrisation of the original RB equations. This observation is hence used to "preprocess" and "postprocess" the initial condition of the original DMV algorithm so that a 4th and a 6th order method are obtained, adding only a tiny percent to the total expense of the original DMV algorithm! These observations hold also for the RATTLE methods for the RB, which is shown to be equivalent to the DMV algorithm. Remarkably, for the 3×3 RB the DMV algorithm can be implemented as an *explicit* method, without losing any of its originaly integrability.

The present paper is organised as follows. In the rest of this section we describe briefly the connection between the DMV algorithm and the discrete Lagrangian approach. In Section 2 we study the modified vector field of the DMV algorithm deriving the components f_3 and f_5 explicitely and sketching the procedure for the higher order components. In Section 3 we focus on the case of the 3×3 RB equations and obtain explicit expressions for f_3 and f_5 as a constant times $[M, \Omega]$. These constants are then used in Section 4 to rescale the initial condition in order to obtain numerical approximation of order 4 and 6 respectively (DMV4, DMV6). In Section 5 we discuss the behaviour of the methods DMV, DMV4 and DMV6 as function of the step size h of integration. In Section 6 we establish the equivalence of the RATTLE method for the RB with the DMV algorithm and conclude that the scalings for order 4 and 6 for DMV can be used to improve RATTLE in an analogous fashion. In Section 7 we present some numerical experiments illustrating the results presented in this paper. Finally, in Section 8, we describe the explicit algorithm used to implement the DMV in this paper.

1.1 The DMV via the discrete Lagrangian approach

Moser & Veselov (1991) consider the fuctional $S(X)$ determined by

$$S = \sum_k \mathrm{tr}(X_k J X_{k+1}^\top) \tag{1.1}$$

where $X = \{X_k\}$ with $X_k \in \mathrm{O}(N)$ and J is a symmetric matrix. To obtain the stationary points of S, they take constrained variations

$$\sum_k \mathrm{tr}(X_k J X_{k+1}^\top) - \frac{1}{2} \sum_k \mathrm{tr}(\Lambda_k (X_k X_k^\top - I)),$$

(where $\Lambda_k = \Lambda_k^\top$ is a Lagrange multiplier), so that $\delta S = 0$ becomes

$$X_{k+1} J + X_{k-1} J = \Lambda_k X_k,$$

from which, multiplying by $X_k^\top$ on the left and taking into consideration the symmetry of Λ_k,

$$X_{k+1} J X_k^\top + X_{k-1} J X_k^\top = \Lambda_k = \Lambda_k^\top = X_k J X_{k+1}^\top + X_k J X_{k-1}^\top, \tag{1.2}$$

hence, the *discrete analogue of the angular momentum in space*,

$$m_k = X_k J X_{k-1}^\top - X_{k-1} J X_k^\top,$$

is conserved.

In the body variables, setting $\omega_k = X_k^\top X_{k-1} \in \mathrm{O}(N)$ and $M_k = X_{k-1}^{-1} m_k X_{k-1} = \omega_k^\top J - J\omega_k \in \mathfrak{so}(N)^*$ (angular momentum w.r.t. the body), (1.2) becomes

$$\begin{aligned} M_{k+1} &= \omega_k M_k \omega_k^\top \\ M_k &= \omega_k^\top J - J\omega_k. \end{aligned} \tag{1.3}$$

the **discrete Euler–Arnold** equation.

What is the relation between the discrete Lagrangian (1.1) and the Lagrangian of the continuous RB equations?

Let us consider the continuous RB equations in body coordinates,

$$M' = [M, \Omega], \qquad M = \Omega J + J\Omega, \tag{1.4}$$

(Arnold 1989), where M is the angular momentum and Ω the angular velocity. M and Ω are skew-symmetric matrices and J is a diagonal matrix, the inertia operator. The Lagrangian of the continuous RB equations, is the kinetic energy,

$$L = \frac{1}{2}\mathrm{tr}(\Omega^\top M) = \frac{1}{2}\mathrm{tr}(-\Omega^2 J - \Omega J \Omega) = \mathrm{tr}(\Omega^\top J \Omega), \tag{1.5}$$

where we take into account that $\Omega^\top = -\Omega$ and that the trace is invariant under cyclic permutations. Following (Marsden, Pekarsky & Shkoller 1999), discretise $\Omega = g^{-1}\dot{g}$, where $g \in \mathrm{SO}(N)$ is the configuration of the body, using a finite difference approximation of the derivative,

$$\Omega = g^{-1}\dot{g} \approx \frac{1}{h} g_{k+1}^\top (g_{k+1} - g_k), \qquad g_k, g_{k+1} \in \mathrm{SO}(N),$$

which gives

$$L \approx \frac{1}{h^2}\mathrm{tr}(J - g_k^\top g_{k+1} J - J g_{k+1}^\top g_k - g_k^\top g_{k+1} J g_{k+1}^\top g_k).$$

Due to the orthogonality of the g_k's and the cyclicity of the trace, the first and the last term cancel, and moreover, we can write

$$L \approx \frac{1}{h^2}\mathrm{tr}(g_k J g_{k+1}^\top).$$

Up a scaling factor, this is precisely the discrete Lagrangian (1.1) whereas X_k is replaced by g_k.

1.2 The DMV algorithm for the RB equations

Assume that the we wish to solve the RB equations (1.4) in the interval $[t_0, t_n]$, where $t_k = t_0 + kh$, h is the step size of integration and M_0 is initial condition. Without loss of generality, the matrix J is assumed to be diagonal.

The DMV algorithm:

1. Set $M_0 = M_0 h$.
2. For $k = 0, 1, \ldots, n-1$,
   find the unique w_k with the properties below such that $M_k = \omega_k^\top J - J\omega_k$
   set $M_{k+1} = \omega_k M_k \omega_k^\top$
   end
3. Reconstruct $M_n \approx M(t_n) = M_n/h$.

In step 2. one has to solve ripetutely the Moser–Veselov equation

$$M = \omega^\top J - J\omega \tag{1.6}$$

where M is a skew-symmetric matrix and J is diagonal with entries $J_1, J_2, \ldots, J_N$, for the unknown orthogonal matrix ω. The Moser–Veselov equation (1.6) does not have a unique solution; however, if the set S of eigenvalues λ_i of $W = \omega^\top J$ admits a splitting $S = S_+ \cup S_-$, with

$$\bar{S}_+ = S_+, \qquad \bar{S}_- = S_-, \qquad S_- = -S_+, \qquad S_+ \cap S_- = \emptyset, \tag{1.7}$$

then, there exists a unique $\omega = J^{-1}W^\top$ that satisfies (1.6), with $\mathrm{spec}W = S_+$ (Moser & Veselov 1991). We recall that the eigenvalues λ are the solutions of the characteristic equation

$$P(\lambda) = \det(\lambda^2 I - \lambda M - J^2) = 0. \tag{1.8}$$

For the 3×3 RB,

$$\begin{aligned} -P(\lambda) &= \lambda^6 - \lambda^4 \left(J_1^2 + J_2^2 + J_3^2 - m_{12}^2 - m_{13}^2 - m_{23}^2\right) \\ &\quad + \lambda^2 \left(J_1^2 J_2^2 + J_1^2 J_3^2 + J_2^2 J_3^2 - m_{12}^2 J_3^2 - m_{13}^2 J_2^2 - m_{23}^2 J_1^2\right) - J_1^2 J_2^2 J_3^2. \end{aligned}$$

Due to the skew-symmetry of M, $P(\lambda) = P(-\lambda)$, which implies that, introducing the auxiliary variable $\mu = \lambda^2$, $P(\mu)$ is a polynomial of degree three,

$$\begin{aligned} -P(\mu) &= \mu^3 - \mu^2 \left(J_1^2 + J_2^2 + J_3^2 - m_{12}^2 - m_{13}^2 - m_{23}^2\right) \\ &\quad + \mu^1 \left(J_1^2 J_2^2 + J_1^2 J_3^2 + J_2^2 J_3^2 - m_{12}^2 J_3^2 - m_{13}^2 J_2^2 - m_{23}^2 J_1^2\right) - J_1^2 J_2^2 J_3^2 \\ &= \mu^3 + a_2 \mu^2 + a_1 \mu + a_0, \end{aligned}$$

whose roots can be explicitely found (see (Abramowitz & Stegun 1965) p. 17). Once μ_1, μ_2 and μ_3 are determined, we recover the λ_is by

$$\lambda_{\pm i} = \pm\sqrt{\mu_i}, \qquad i = 1, 2, 3.$$

It must be noted that the coefficients of $P(\lambda)$ are invariants, therefore the computation of the roots λ_i can be done once and for all (before step 2.).

Having computed the λ_is, one can then proceed to the computation of the matrix W by solving for eigenvectors corresponding to the λ_is having positive real part, hence $\omega = J^{-1}W^\top$ is constructed for each iteration k. We refer to this method as the *direct method* and it is essentially described in (Moser & Veselov 1991).

The matrix ω can be computed also indirectly, via Schur real forms, as proposed by Cardoso & Leite (2001). This approach, and our explicit approach, is described in greater details in Section 8.

2 Modified vector field of the DMV

In this section, we consider the Discrete Moser–Veselov algorithm for the dynamics of the rigid body,

$$\begin{aligned} M_k &= \omega_k^\top J - J\omega_k \\ M_{k+1} &= \omega_k M_k \omega_k^\top, \end{aligned} \tag{2.1}$$

and assume that $\omega_k \approx I - h\Omega(t_k)$, $t_k = t_0 + hk$, where $\Omega(t_k)$ is the angular momentum of the continuous RB equation at time $t = t_k$.

Setting

$$M_{k+1} = \Phi_h(M_k) = M_k + h[M_k, \Omega_k] + h^2 d_2 + h^3 d_3 + h^4 d_4 + \cdots, \tag{2.2}$$

we look for a modified vector field (see for instance (Hairer, Lubich & Wanner 2002)),

$$\tilde{M}' = [\tilde{M}, \tilde{\Omega}] + hf_2(\tilde{M}, \tilde{\Omega}) + h^2 f_3(\tilde{M}, \tilde{\Omega}) + h^3 f_4(\tilde{M}, \tilde{\Omega}) + \cdots \tag{2.3}$$

such that $\Phi_h(M_k)$ equals the solution $\tilde{M}(t_{k+1})$ at time $t_{k+1} = t_0 + (k+1)h$ of the modified vector field (2.3).

2.1 Deriving the map $\Phi_h(M_k)$

To arrive at the formulation (2.2), we write

$$\omega_k = \exp(-h\Omega_0 - h^2\Omega_1 - h^3\Omega_2 - h^4\Omega_3 - h^5\Omega_4 + \cdots), \tag{2.4}$$

where $\Omega_0, \Omega_1, \Omega_2, \ldots,$ are skew-symmetric matrices computed so that

$$\omega_k^\top J - J\omega_k = h(\Omega(t_k)J + J\Omega(t_k)). \tag{2.5}$$

The next step consists of expanding $M_{k+1} = \Phi_h(M_k)$ in (2.1) in terms of powers of h using the series expansion for ω_k. Finally, comparison of the Taylor expansion of the solution of (2.3) and (2.2) allows us to derive the expressions for the f_i's, $i = 3, 4, 5, \ldots$ in (2.3).

We commence by expanding ω_k in powers of h,

$$\begin{aligned}\omega_k = I - h\Omega_0 &+ h^2(-\Omega_1 + \tfrac{1}{2}\Omega^2) + h^3(-\Omega_2 + \tfrac{1}{2}(\Omega_0\Omega_1 + \Omega_1\Omega_0) - \tfrac{1}{6}\Omega^3) \\ &+ h^4\Big(- \Omega_3 + \tfrac{1}{2!}(\Omega_1^2 + \Omega_0\Omega_2 + \Omega_2\Omega_0) - \tfrac{1}{3!}(\Omega_0^2\Omega_1 + \Omega_0\Omega_1\Omega_0 + \Omega_1\Omega_0^2) + \tfrac{1}{4!}\Omega_0^4\Big) \\ &+ h^5\Big(- \Omega_4 + \tfrac{1}{2!}(\Omega_3\Omega_0 + \Omega_1\Omega_2 + \Omega_2\Omega_1 + \Omega_0\Omega_3) \\ &\quad - \tfrac{1}{3!}(\Omega_0^2\Omega_2 + \Omega_0\Omega_1^2 + \Omega_0\Omega_2\Omega_0 + \Omega_1\Omega_0\Omega_1 + \Omega_1^2\Omega_0 + \Omega_2\Omega_0^2) \\ &\quad + \tfrac{1}{4!}(\Omega_0^3\Omega_1 + \Omega_0^2\Omega_1\Omega_0 + \Omega_0\Omega_1\Omega_0^2 + \Omega_1\Omega_0^3) - \tfrac{1}{5!}\Omega_0^5\Big) \\ &+ \mathcal{O}(h^6).\end{aligned} \tag{2.6}$$

Substituting in (2.5), we obtain

$$\begin{aligned}h(\Omega(t_k)J + J\Omega(t_k)) = h(\Omega_0 J + J\Omega_0) &+ h^2(\Omega_1 J + J\Omega_1 + \tfrac{1}{2}(\Omega_0^2 J - J\Omega_0^2)) \\ &+ h^3(\Omega_2 J + J\Omega_2 + \tfrac{1}{2}[(\Omega_0\Omega_1 + \Omega_1\Omega_0), J] + \tfrac{1}{6}(\Omega_0^3 J + J\Omega_0^3)) + \cdots.\end{aligned}$$

Comparing left and right-hand-sides, it is trivially observed that the order-h term disappears if $\Omega_0 = \Omega$ (to simplify notation, we omit the dependence of Ω on t_k). In order to annihilate the h^2-term, we require that

$$\Omega_1 J + J\Omega_1 + \frac{1}{2}(\Omega_0^2 J - J\Omega_0^2) = 0.$$

Recall that $M = \Omega J + J\Omega$ and hence $M' = \Omega' J + J\Omega'$. On the other hand, $M' = [M, \Omega] = -(\Omega^2 J - J\Omega^2)$. Hence we can write

$$O = \Omega_1 J + J\Omega_1 - \frac{1}{2}M' = \Omega_1 J + J\Omega_1 - \frac{1}{2}(\Omega' J + J\Omega')$$

and the identity is satisfied by if and only if

$$\Omega_1 = \frac{1}{2}\Omega'. \tag{2.7}$$

$M = \Omega J + J\Omega$	
$M' = -[\Omega^2, J]$	$= \Omega' J + J\Omega'$
$M'' = -[\Omega'\Omega + \Omega\Omega', J]$	$= \Omega'' J + J\Omega''$
$M''' = -[\Omega''\Omega + 2\Omega'^2 + \Omega\Omega'', J]$	$= \Omega''' J + J\Omega'''$
$M^{(iv)} = -[\Omega'''\Omega + 3(\Omega''\Omega' + \Omega'\Omega'') + \Omega\Omega''', J]$	$= \Omega^{(iv)} J + J\Omega^{(iv)}$

Table 1: M and its first derivatives with respect to time as functions of Ω (the commutator is the usual matrix commutator)

$M = \Omega J + J\Omega$
$M' = [M, \Omega]$
$M'' = [[M, \Omega], \Omega] + [M, \Omega']$
$M''' = [[[M, \Omega], \Omega], \Omega] + [[M, \Omega'], \Omega] + 2[[M, \Omega], \Omega'] + [M, \Omega'']$
$M^{(iv)} = [[[[M, \Omega], \Omega], \Omega], \Omega] + [[[M, \Omega'], \Omega], \Omega] + 2[[[M, \Omega], \Omega'], \Omega]$
$\quad + 3[[[M, \Omega], \Omega], \Omega'] + [[M, \Omega''], \Omega] + 3[[M, \Omega'], \Omega'] + 3[[M, \Omega], \Omega'']] + [M, \Omega''']$

Table 2: M and its first derivatives with respect to time in terms of a single occurrence of M.

By a similar token, we require that

$$\begin{aligned}\Omega_2 J + J\Omega_2 &= -\tfrac{1}{3!}(\Omega_0^3 J + J\Omega_0^3) - \tfrac{1}{2!}[(\Omega_0\Omega_1 + \Omega_1\Omega_0), J] \\ &= -\tfrac{1}{6}(\Omega^3 J + J\Omega^3) + \tfrac{1}{4}M'' \\ &= -\tfrac{1}{6}(\Omega^3 J + J\Omega^3) + \tfrac{1}{4}(\Omega'' J + J\Omega''),\end{aligned}$$

From which we deduce that

$$\Omega_2 = \frac{1}{4}\Omega'' - \frac{1}{6}\Omega^3. \tag{2.8}$$

The equation for Ω_3 is

$$\begin{aligned}0 \quad = \quad & \Omega_3 J + J\Omega_3 + \frac{1}{2!}[\Omega_1^2 + \Omega_0\Omega_2 + \Omega_2\Omega_0, J] \\ & + \frac{1}{3!}\Big((\Omega_0^2\Omega_1 + \Omega_0\Omega_1\Omega_0 + \Omega_1\Omega_0^2)J + J(\Omega_0^2\Omega_1 + \Omega_0\Omega_1\Omega_0 + \Omega_1\Omega_0^2)\Big) \\ & + \frac{1}{4!}[\Omega_0^4, J],\end{aligned}$$

from which we deduce that

$$\Omega_3 = \frac{1}{8}\Omega''' - \frac{1}{12}(\Omega^2\Omega' + \Omega\Omega'\Omega + \Omega'\Omega^2) + \frac{1}{8}\mathcal{J}^{-1}[\Omega^4, J] + \frac{1}{8}\mathcal{J}^{-1}[{\Omega'}^2, J],$$

where the operator $\mathcal{J}$ is the linear operator such that $\mathcal{J}S = SJ + JS$.

Using the relations between M, Ω and J and their derivatives, after some tedious computations we find

$$[{\Omega'}^2, J] = -\Big((\Omega'\Omega^2 + \Omega^2\Omega')J + J(\Omega'\Omega^2 + \Omega^2\Omega')\Big) - [\Omega^4, J],$$

Hence, Ω_3 reduces simply to

$$\Omega_3 = \frac{1}{8}\Omega''' - \frac{1}{24}(5\Omega^2\Omega' + 2\Omega\Omega'\Omega + 5\Omega'\Omega^2). \tag{2.9}$$

Next, we consider the equation for Ω_4. Setting the coefficient of h^5 to zero, we obtain

$$0 \quad = \quad \Omega_4 J + J\Omega_4 + \frac{1}{2!}[\Omega_3\Omega_0 + \Omega_2\Omega_1 + \Omega_1\Omega_2 + \Omega_0\Omega_3, J]$$

$$
\begin{aligned}
&+\frac{1}{3!}\Big((\Omega_0^2\Omega_2+\Omega_0\Omega_2\Omega_0+\Omega_2\Omega_0^2+\Omega_0\Omega_1^2+\Omega_1\Omega_0\Omega_1+\Omega_1^2\Omega_0)J\\
&\quad +J(\Omega_0^2\Omega_2+\Omega_0\Omega_2\Omega_0+\Omega_2\Omega_0^2+\Omega_0\Omega_1^2+\Omega_1\Omega_0\Omega_1+\Omega_1^2\Omega_0)\Big)\\
&+\frac{1}{4!}[\Omega_0^3\Omega_1+\Omega_0^2\Omega_1\Omega_0+\Omega_0\Omega_1\Omega_0^2+\Omega_1\Omega_0^3,J]\\
&+\frac{1}{5!}(\Omega_0^5J+J\Omega_0^5).
\end{aligned}
$$

At this point, we repeat the tedious work of substituting in the above expression the known values of $\Omega_0,\Omega_1,\ldots$, and of expressing the whole as $\Omega_4J+J\Omega_4=CJ+JC$, from which we will deduce $\Omega_4=C$.

Firstly, we have

$$
\begin{aligned}
0 \;=\;& \Omega_4J+J\Omega_4-\frac{1}{16}(\Omega^{(iv)}J+J\Omega^{(iv)})\\
&-\frac{1}{8}[\Omega''\Omega'+\Omega'\Omega''+\Omega^2\Omega'\Omega+\Omega\Omega'\Omega^2+\Omega^3\Omega'+\Omega'\Omega^3,J]\\
&+\frac{1}{24}\Big((\Omega^2\Omega''+\Omega\Omega''\Omega+\Omega''\Omega^2+\Omega\Omega'^2+\Omega'\Omega\Omega'+\Omega'^2\Omega)J\\
&\quad +J(\Omega^2\Omega''+\Omega\Omega''\Omega+\Omega''\Omega^2+\Omega\Omega'^2+\Omega'\Omega\Omega'+\Omega'^2\Omega)\Big)\\
&-\frac{3}{40}(\Omega^5J+J\Omega^5).
\end{aligned}
$$

Secondly, using the equalities

$$
\begin{aligned}
[\Omega''\Omega'+\Omega'\Omega'',J] &= -[M',\Omega'']-[M'',\Omega']\\
&= [[\Omega^2,J],\Omega'']+[[\Omega'\Omega+\Omega\Omega',J],\Omega']\\
&= -\Big((\Omega''\Omega^2+\Omega^2\Omega'')J+J(\Omega''\Omega^2+\Omega^2\Omega'')\Big)+\Omega^2M''+M''\Omega^2\\
&\quad +[[\Omega'\Omega+\Omega\Omega',J],\Omega'],\\
&= -\Big((\Omega''\Omega^2+\Omega^2\Omega'')J+J(\Omega''\Omega^2+\Omega^2\Omega'')\Big)-\Big(\Omega^2[\Omega'\Omega+\Omega\Omega',J]\\
&\quad +[\Omega'\Omega+\Omega\Omega',J]\Omega^2\Big)+[[\Omega'\Omega+\Omega\Omega',J],\Omega'],\\
[\Omega^2\Omega'\Omega+\Omega\Omega'\Omega^2+\Omega^3\Omega'+\Omega'\Omega^3,J] &= [\Omega^2(\Omega'\Omega+\Omega\Omega')+(\Omega'\Omega+\Omega\Omega')\Omega^2,J]\\
&= \Omega^2[\Omega'\Omega+\Omega\Omega',J]+[\Omega'\Omega+\Omega\Omega',J]\Omega^2\\
&\quad -(M'(\Omega'\Omega+\Omega\Omega')+(\Omega'\Omega+\Omega\Omega')M').
\end{aligned}
$$

we deduce that

$$
\begin{aligned}
&[\Omega''\Omega'+\Omega'\Omega''+\Omega^2\Omega'\Omega+\Omega\Omega'\Omega^2+\Omega^3\Omega'+\Omega'\Omega^3,J]\\
&=-\Big((\Omega''\Omega^2+\Omega^2\Omega''+\Omega'^2\Omega+2\Omega'\Omega\Omega'+\Omega\Omega'^2)J\\
&\quad +J(\Omega''\Omega^2+\Omega^2\Omega''+\Omega'^2\Omega+2\Omega'\Omega\Omega'+\Omega\Omega'^2)\Big),
\end{aligned}
$$

from which we obtain

$$
\Omega_4=\frac{1}{16}\Omega^{(iv)}-\frac{1}{24}(4\Omega''\Omega^2+\Omega\Omega''\Omega+4\Omega^2\Omega''+4\Omega'^2\Omega+7\Omega'\Omega\Omega'+4\Omega\Omega'^2)+\frac{3}{40}\Omega^5. \qquad (2.10)
$$

In general, the algorithm to derive Ω_i, for $i=1,2,\ldots$, is

1. Find the coefficient of h^{i+1} in (2.5) and set it equal to zero. This will give an equation of the type $\Omega_iJ+J\Omega_i=C_iJ+JC_i+[D_i,J]$. Note that the terms C_iJ+JC_i have an odd occurrence of the

$\Omega_0 = \Omega$
$\Omega_1 = \frac{1}{2}\Omega'$
$\Omega_2 = \frac{1}{4}\Omega'' - \frac{1}{6}\Omega^3$
$\Omega_3 = \frac{1}{8}\Omega''' - \frac{1}{24}(5\Omega^2\Omega' + 2\Omega\Omega'\Omega + 5\Omega'\Omega^2)$

Table 3: The functions Ω_i

Ω_js, while the terms of the type $[D_i, J]$ have an even occurrence of the Ω_js (this statement can be rigorously proved by writing $\omega = \exp(-f(h))$, where $f(h) = \sum_{j=0}^{\infty} h^{j+1}\Omega_j$, and then using the relations between $\exp$ and $\sinh, \cosh$).

2. Use the derivatives of M and Ω to express the term $[D_i, J]$ as $\tilde{C}_i J + J\tilde{C}_i$.
3. Deduce $\Omega_i = C_i + \tilde{C}_i$.

Once the Ω_is are known, substituting back in (2.1) and using the well known identity

$$\exp(X)Y\exp(-X) = \exp_{\mathrm{ad}_X} Y = \sum_{k=0}^{\infty} \frac{1}{k!}\mathrm{ad}_X^k(Y),$$

where $\mathrm{ad}_X(Y) = [X, Y]$ and, recursively, $\mathrm{ad}_X^k(Y) = [X, \mathrm{ad}_X^{k-1}(Y)]$ (see for instance (Iserles, Munthe-Kaas, Nørsett & Zanna 2000)), we obtain the Taylor expansion for the DMV algorithm,

$$\begin{aligned} M_{k+1} = M_k - h[\Omega_0, M_k] &+ h^2(-[\Omega_1, M_k] + \tfrac{1}{2!}[\Omega_0, [\Omega_0, M_k]]) \\ &+ h^3\Big(- [\Omega_2, M_k] + \tfrac{1}{2!}([\Omega_0, [\Omega_1, M_k]] + [\Omega_1, [\Omega_0, M_k]]) - \tfrac{1}{3!}[\Omega_0, [\Omega_0, [\Omega_0, M_k]]]\Big) \\ &+ \mathcal{O}\big(h^4\big) \end{aligned}$$

hence,

$$\begin{aligned} M_{k+1} = M_k - h[\Omega, M_k] &+ \tfrac{1}{2}h^2(-[\Omega', M_k] + [\Omega, [\Omega, M_k]]) \\ &+ h^3(-\tfrac{1}{4}[\Omega'', M_k] + \tfrac{1}{4}[\Omega, [\Omega', M_k]] + \tfrac{1}{4}[\Omega', [\Omega, M_k]] - \tfrac{1}{6}[\Omega, [\Omega, [\Omega, M_k]]] + \tfrac{1}{6}[\Omega^3, M]) \\ &+ \mathcal{O}\big(h^4\big) . \end{aligned}$$

Using the expressions for the Ω_is, by the skew-symmetry of the commutator and by comparison with (2.2), we obtain the explicit expressions for the functions $d_2, d_3, \ldots$,

$$\begin{aligned} d_2 &= \tfrac{1}{2}([M, \Omega'] + [[M, \Omega], \Omega]), \\ d_3 &= \tfrac{1}{4}[M, \Omega''] + \tfrac{1}{4}[[M, \Omega'], \Omega] + \tfrac{1}{4}[[M, \Omega], \Omega'] + \tfrac{1}{6}[[[M, \Omega], \Omega], \Omega] - \tfrac{1}{6}[M, \Omega^3], \\ d_4 &= \ldots, \end{aligned} \tag{2.11}$$

where $M \equiv M_k \approx M(t_k)$. The general expression for d_i as a function of $\Omega_0, \ldots, \Omega_{i-1}$ is

$$d_i = \sum_{j=1}^{i} \frac{(-1)^j}{j!} \sum_{k_1+k_2+\cdots+k_j=i-j} \mathrm{ad}_{\Omega_{k_1}}\mathrm{ad}_{\Omega_{k_2}}\cdots\mathrm{ad}_{\Omega_{k_j}} M, \qquad k_1, \ldots k_j \in \{0, 1, \ldots, i-1\}. \tag{2.12}$$

2.2 Taylor expansion of the solution of the modified equation

The derivation of the Taylor expansion of the solution of the modified vector field can be done according to (Hairer et al. 2002). Consider

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{y} = f(\tilde{y}) + hf_2(\tilde{y}) + h^2 f_3(\tilde{y}) + \cdots,$$

where $f(M) = [M, \Omega] = [M, \mathcal{J}^{-1}M]$ is the original vector field of the RB equations, where $\mathcal{J}$ is a linear operator, defined such that $\mathcal{J}\Omega = \Omega J + J\Omega = M$. Putting $\tilde{y}(t) = M(t)$, we expand the solution of the above equation in a Taylor series and collect corresponding powers of h,

$$\begin{aligned}\tilde{y}(t+h) \quad = \quad & M(t) + hf(M) + h^2\left(f_2(M) + \frac{1}{2!}f'f(M)\right) \\ & + h^3\left(f_3(M) + \frac{1}{2!}(f'f_2(M) + f_2'f(M)) + \frac{1}{3!}(f''(f,f)(M) + f'f'f(M))\right) + \cdots,\end{aligned}$$

where f' is considered as a linear operator, f'' as a bilinear operator and so on and so forth. In our case,

$$\begin{aligned} f'(z)(M) &= [z, \mathcal{J}^{-1}M] + [M, \mathcal{J}^{-1}z] \\ &= [z, \Omega] + [M, \mathcal{J}^{-1}z] \\ f''(z_1, z_2)(M) &= 2[z_1, \mathcal{J}^{-1}z_2], \end{aligned}$$

and, since f is quadratic, f''' and all the other higher derivatives equal zero.

Taking into accout the equalities displayed in Tables (1)-(2), we have

$$\begin{aligned} f'f(M) &= [[M,\Omega],\Omega] + [M, [\mathcal{J}^{-1}[M,\Omega]]] = [[M,\Omega],\Omega] + [M,\Omega'] \\ f''(f,f)(M) &= 2[[M,\Omega], \mathcal{J}^{-1}[M,\Omega]] = 2[[M,\Omega],\Omega'] \\ f'f'f(M) &= [[[M,\Omega],\Omega],\Omega] + [M, \mathcal{J}^{-1}[[M,\Omega],\Omega]] + [[M,\Omega'],\Omega] + [M, \mathcal{J}^{-1}[M,\Omega']] \\ &= [[[M,\Omega],\Omega],\Omega] + [M,\Omega''] + [[M,\Omega'],\Omega], \end{aligned}$$

from which we immediately deduce that

$$f_2 = d_2 - \frac{1}{2!}f'f(M) = O,$$

as we would expect from a second-order methods, and, moreover,

$$\begin{aligned} f_3 &= d_3 - \tfrac{1}{3!}(f''(f,f)(M) + f'f'f(M)) \\ &= \tfrac{1}{12}[M, \Omega'' - [\Omega,\Omega'] - 2\Omega^3], \end{aligned}$$

hence f_3 is a quartic polynomial in M. For our computations, it will be conventient to write f_3 as

$$f_3 = \frac{1}{12}\Big([M,\Omega''] - [[M,\Omega],\Omega'] + [[M,\Omega'],\Omega] - 2[M,\Omega^3]\Big). \tag{2.13}$$

Note that f_3 is a quartic polynomial in M. This confirms what had been already proved in (Deift, Li & Tomei 1992) by a completely different argument: Deift et al. used the theory of loop groops and rank-2 extension and their expression for $K(M) = f_3(M)$ is given as a limit of a double integral. In the sequel, we shall see that, due to its simplicity, our formula (2.13) is much more amenable and easy to use in numerical computations to dramatically improve the performance of the DMV algorithm.

At this point it is important to stress an important difference between the expressions for the modified vector field of (Hairer et al. 2002) and ours. While the vector field discussed in (Hairer et al. 2002) is in $\mathbb{R}^n$, hence the f'' is a symmetric quadratic operator, this is not the case for our vector field which is on matrices, thus in general $f''(f'f,f) \neq f''(f,f'f)$. This non-commutative case is discussed with more generality in (Munthe-Kaas & Krogstad 2002). However, we observe that *all* the terms containing combinations of f'', f' and f correspond simply to higher derivatives of f. The mixed terms are treated instead specifically. Therefore, we have

$$\begin{aligned} f_4 &= d_4 - \tfrac{1}{4!}M^{(iv)} - \tfrac{1}{2!}(f'f_3 + f_3'f) \\ f_5 &= d_5 - \tfrac{1}{5!}M^{(v)} - \tfrac{1}{2!}(f'f_4 + f_4f' + \tfrac{1}{2!}\tfrac{\mathrm{d}}{\mathrm{d}t}(f_3'f + f'f_3)). \end{aligned} \tag{2.14}$$

whereby all the vector fields are computed at M.

By direct computation,

$$\begin{aligned} f'f_3 &= \frac{1}{12}\Big([[M,\Omega''-[\Omega,\Omega']-2\Omega^3],\Omega]+[M,\mathcal{J}^{-1}([M,\Omega'']-[[M,\Omega],\Omega']+[[M,\Omega'],\Omega]-2[M,\Omega^3])]\Big) \\ &= \frac{1}{12}\Big([[M,\Omega''-[\Omega,\Omega']-2\Omega^3],\Omega]+[M,\Omega'''-3(\Omega^2\Omega'+\Omega'\Omega^2)]\Big) \end{aligned}$$

$$\begin{aligned} f_3'f &= \frac{1}{12}(-2([[M,\Omega],\Omega^3]+[M,\Omega'\Omega^2+\Omega\Omega'\Omega+\Omega^2\Omega'])-([[M,\Omega],[\Omega,\Omega']]+[M,[\Omega,\Omega'']]) \\ &\quad+[[M,\Omega],\Omega'']+[M,\mathcal{J}^{-1}([[[M,\Omega],\Omega],\Omega])]+[M,\mathcal{J}^{-1}([[M,\Omega'],\Omega])] \\ &\quad+[M,\mathcal{J}^{-1}([[M,\Omega],\Omega'])]+[M,\mathcal{J}^{-1}([[M,\Omega],\Omega'])] \\ &\quad+[M,\mathcal{J}^{-1}([M,\mathcal{J}^{-1}[[M,\Omega],\Omega]])]+[M,\mathcal{J}^{-1}([M,\mathcal{J}^{-1}[M,\Omega']])]) \\ &= \frac{1}{12}(-2([[M,\Omega],\Omega^3]+[M,\Omega'\Omega^2+\Omega\Omega'\Omega+\Omega^2\Omega'])-([[M,\Omega],[\Omega,\Omega']]+[M,[\Omega,\Omega'']]) \\ &\quad+[[M,\Omega],\Omega'']+[M,\Omega''']). \end{aligned}$$

To reduce the second term in $f'f_3$ we have used the equation for M''' in Table 2, thence the identities $[M,\Omega]=M'=\Omega'J+J\Omega'=-[\Omega^2,J]$.

We have

$$\begin{aligned} f'f_3+f_3f' &= \frac{1}{12}\Big(2[M,\Omega''']+2[[M,\Omega''],\Omega]-4[[M,\Omega],\Omega^3] \\ &\quad-[M,5\Omega^2\Omega'+2\Omega\Omega'\Omega+5\Omega'\Omega^2]+[[[M,\Omega'],\Omega],\Omega]-[[[M,\Omega],\Omega],\Omega']. \end{aligned}$$

By a long and tedious computation, it is verified that indeed

$$f_4=d_4-\frac{1}{4!}M^{(iv)}-\frac{1}{2!}(f_3'f+f'f_3)=0.$$

Similarly,

$$\begin{aligned} f_5 = {}& \tfrac{1}{80}[M,\Omega^{(iv)}]-\tfrac{1}{80}[M,[\Omega,\Omega''']]+\tfrac{3}{40}[M,\Omega^5-\Omega'\Omega\Omega'] \\ &+\tfrac{1}{80}[M,[\Omega',\Omega'']]-\tfrac{1}{40}[M,\Omega\Omega''\Omega]-\tfrac{1}{20}[M,\Omega^2\Omega''+\Omega''\Omega^2] \\ &+\tfrac{1}{20}[M,[\Omega^3,\Omega']]-\tfrac{1}{40}[M,\Omega'^2\Omega+\Omega\Omega'^2+\Omega[\Omega,\Omega']\Omega]. \end{aligned} \tag{2.15}$$

The fact that $f_2=f_4=0$ is not unexpected. Indeed, it can be proved that all the functions $f_{2i}=0$ for $i=1,2,\ldots$, as shown in the result below.

Theorem 2.1 *The discrete Moser–Veselov algorithm* (2.1) *is time-reversible, hence* $f_{2i}=0$ *for* $i=1,2,\ldots$.

Proof. It is sufficient to show that if $M_{k+1}=\Phi_h(M_k)$ and $M_{k+2}=\Phi_{-h}(M_{k+1})$, then $M_{k+2}=M_k$.

By construction,

$$\begin{aligned} M_k &= \omega_k^\top J-J\omega_k, \\ M_{k+1} &= \Phi_k(M_k)=-\omega_kJ+J\omega_k^\top. \end{aligned}$$

Recall that, for a positive time step, the orthogonal solution ω of the matrix equation (1.6)

$$M=\omega^\top J-J\omega$$

is computed as the unique matrix such that $W=\omega^\top J$ has eigenvalues with positive real part (see $\S$ 1.2).

Next, we compute $M_{k+2} = \Phi_{-h}(M_{k+1})$. For negative step size, either i) we compute the unique solution corresponding to eigenvalues λ solving (1.8) having negative real parts, or ii) we take the solution with positive real parts and exchange ω by $\omega^\top$; both procedures are equivalent. Considering the first case, we have

$$M_{k+1} = \omega_{k+1}^\top J - J\omega_{k+1},$$

which, by comparison with $M_{k+1} = -\omega_k J + J\omega_k^\top$ gives immediately

$$\omega_{k+1}^\top = -\omega_k$$

for sufficiently small step size, because of the uniqueness of the orthogonal solution with the required eigenvalues properties and the $\mathcal{O}(h^2)$ consistency. Hence,

$$\begin{aligned} M_{k+2} &= \omega_{k+1} M_{k+1} \omega_{k+1}^\top \\ &= \omega_k^\top(-\omega_k J + J\omega_k^\top)\omega_k \\ &= \omega_k^\top J - J\omega_k \\ &= M_k, \end{aligned}$$

which implies that $\Phi_{-h} = \Phi_h^{-1}$. This condition is equivalent to $f_{2i} = 0$, for $i = 1, 2, \ldots$, see (Hairer et al. 2002). □

Moser and Veselov (Moser & Veselov 1991) prove that, for general N, the algorithm (2.1) is a time-reparamtetrisation of the flow of the original vector field of the rigid body: since the mapping preserves the underlying Poisson structure and all the integrals $F_i = c_i$ of the system, it commutes with all commuting Hamiltonian flows generated by the F_is, $M' = \{M, \nabla F_i\}$. The nonsingular compact level sets $T_c = \cap_i(F_i = c_i)$ consists of a finite union of tori and on each torus the DMV mapping is a shift along the trajectory depending on an integral quantity H_2 which will be introduced later.

This, in combination with Theorem 2.1, means that (2.1) solves the modified equation

$$M' = (1 + h^2\tau_3 + h^4\tau_5 + \cdots + h^{2i}\tau_{2i+1} + \cdots)[M, \Omega],$$

where h is the stepsize of integration and the τ_{2i+1}, for $i = 1, 2, \ldots$, are constants that depend only on the function H_2, the matrix J and the Casimirs of the system.

In what follows, we use our results on the modified vector field of the DMV algorithm to find an explicit expression for some of the τ_is in the 3×3 case.

3 Modified vector fields and time reparametrisation of the Discrete Moser–Veselov algorithm for the 3×3 rigid body

The previous results were general and independent of the dimensions of the rigid body under consideration. In what follows, otherwise specified, we focus on the 3×3 case. In this case, the equations of motion for the components $m_3 = -m_{1,2}, m_1 = -m_{1,3}, m_2 = m_{2,3}$ are

$$\begin{aligned} m_1' &= -\frac{(J_2 - J_3)m_2 m_3}{(J_1 + J_2)(J_1 + J_3)} \\ m_2' &= \frac{(J_1 - J_3)m_1 m_3}{(J_1 + J_2)(J_2 + J_3)} \\ m_3' &= -\frac{(J_1 - J_2)m_1 m_2}{(J_2 + J_3)(J_1 + J_3)}, \end{aligned}$$

The correspondence between the matrix and vector notation is well known and is given by the *hatmap* function,

$$\hat{\mathbf{a}} = A = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}, \qquad \mathbf{a} = (a_1, a_2, a_3)^\top,$$

while commutation $[A, B]$ corresponds to computing the cross-product $\mathbf{a} \times \mathbf{b}$ of the corresponding vectors ($A = \hat{\mathbf{a}}$, $B = \hat{\mathbf{b}}$). The Hamiltonian is

$$H = \frac{m_{1,2}^2}{J_1 + J_2} + \frac{m_{1,3}^2}{J_1 + J_3} + \frac{m_{2,3}^2}{J_2 + J_3}. \tag{3.1}$$

In Figure 3.1 we display the components m_1, m_2, m_3 of the solution of the DMV algorithm (dotted line) compared with the exact solutions of the RB equations (1.4) (solid line) and the exact solutions of the modified vector fields $f + h^2 f_3$ (dash-dotted line) and $f + h^2 f_3 + h^4 f5$ (dashed line) in the interval $[0, 50]$. The initial condition $\mathbf{m}_0$ and the matrix J are reported in §7. The stepsize of integration h for the DMV is chosen as $h = \frac{8}{10}$. The "exact solutions" are computed using the MATLAB routine `ode45` with absolute and relative errors set to machine precision. The numerics confirm that the order of approximations are two, for the RB; four for the modified vector field $f + h^2 f_3$ and six for $f + h^2 f_3 + h^4 f_5$. Figures 3.1 and 3.2

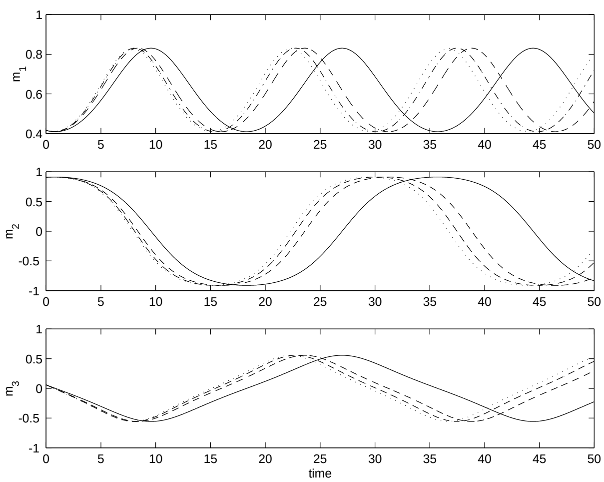

Figure 3.1: The DMV solution of the RB equations (dotted line), the exact solution (solid line) and the trajectories corresponding to the modified vector fields $f + h^2 f_3$ (dash-dotted line) and $f + h^2 f_3 + h^4 f_5$ (dashed line) in the interval $[0, 50]$ with $h = \frac{8}{10}$.

indicate that DMV approximation, though preserving very well the 'qualitative' features of the solution of (1.4), oscillates too rapidly compared to exact solution.

In what follows, we focus on the 3×3 case and we set

$$H_2 = m_{1,2}^2 J_3^2 + m_{1,3}^2 J_2^2 + m_{2,3}^2 J_1^2, \tag{3.2}$$

and

$$\Delta = (J_1 + J_2)(J_1 + J_3)(J_2 + J_3). \tag{3.3}$$

Proposition 3.1 *The function f_3 is of the form*

$$f_3 = \tau_3[M, \Omega],$$

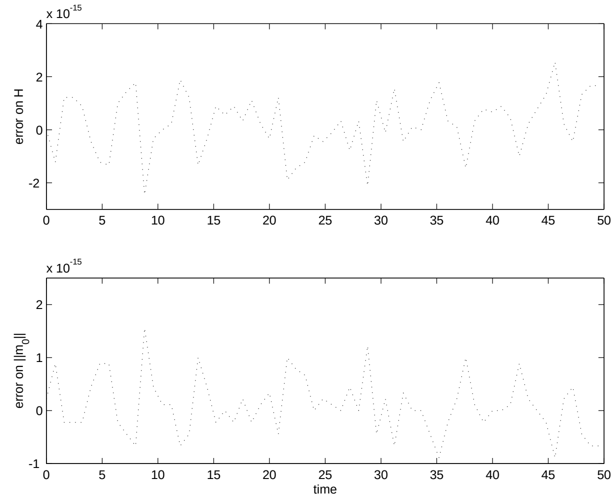

Figure 3.2: Error on the Hamiltonian (top) and Casimir, 2-norm of the solution, (bottom) for the DMV with $h = \frac{8}{10}$.

where

$$\tau_3 = \frac{1}{6\Delta^2}((3\det(J)\mathrm{tr}(J) + J_1^2J_2^2 + J_1^2J_3^2 + J_2^2J_3^2)\|\mathbf{m}\|_2^2 + (3(J_1J_2 + J_1J_3 + J_2J_3) + \mathrm{tr}(J^2))H_2).$$

is a constant which depends the integral function H_2, on the Casimir $\|\mathbf{m}\|_2 = \|\mathbf{m}(\mathbf{t_0})\|_2$ and on the entries of the matrix J.

Proof. We start recalling that H_2 in (3.2) is an integral of the system[1], though it is not independent of the Hamiltonian H. Reducing the right-hand-side of (3.1) to a common denominator and eliminating it, we obtain

$$(J_1 + J_2)(J_1 + J_3)(J_2 + J_3)H = (J_1J_2 + J_1J_3 + J_2J_3)(m_{1,2}^2 + m_{1,3}^2 + m_{2,3}^2) + H_2.$$

Note that $m_{1,2}^2 + m_{1,3}^2 + m_{2,3}^2 = \|\mathbf{m}\|_2^2$ is constant, therefore H_2 cannot depend on time and must be a constant too.

Secondly, let us denote by $\mathbf{f}_3 = (f_{3;1}, f_{3;2}, f_{3;3})^\top$ the three-vector such that $\hat{\mathbf{f}}_3 = f_3$. We show that the first component of this vector is of the form

$$f_{3;1} = \tau_3 m_1'.$$

To this scope, we compute explicitly $f_{3;1}$ using (2.13). Collecting terms carefully, we have

$$\begin{aligned} f_{3;1} \quad = \quad & \frac{-(J_2 - J_3)m_2m_3}{6(J_1 + J_2)^3(J_1 + J_3)^3(J_2 + J_3)^2}\Big((3J_1^2(J_1J_2 + J_1J_3 + J_2J_3) + J_1^2(J_1^2 + J_2^2 + J_3^2))m_1^2 \\ & + (3J_2^2(J_1J_2 + J_1J_3 + J_2J_3) + J_2^2(J_1^2 + J_2^2 + J_3^2))m_2^2 \\ & + (3J_3^2(J_1J_2 + J_1J_3 + J_2J_3) + J_3^2(J_1^2 + J_2^2 + J_3^2))m_3^2 \\ & (3J_1J_2J_3(J_1 + J_2 + J_3) + J_1^2J_2^2 + J_1^2J_3^2 + J_2^2J_3^2)(m_1^2 + m_2^2 + m_3^2)\Big), \end{aligned}$$

from which, using the above expression for H_2, the identity $m_1^2 + m_2^2 + m_3^2 = \|\mathbf{m}\|_2^2 = \|\mathbf{m}(t_0)\|_2^2$, and the equation for m_1', the result follows.

[1]Moser and Veselov call this quantity H, while we reserve H for the Hamiltonian function (3.1).

The procedure is analogous for the other components of the vector $\mathbf{f}_3$. □

Before proceding, we introduce the following simplifying notation: for $i, j = 1, 2, \ldots$, let

$$\begin{aligned} C_{J,i,j} &= J_1^i J_2^j + J_1^i J_3^j + J_2^i J_3^j \\ C_{J,i} &= C_{J,i,i} \\ C_J &= C_{J,1}. \end{aligned}$$

Proposition 3.2 *For the* 3×3 *RB, the function* f_5 *is of the form*

$$f_5 = \tau_5 [M, \Omega],$$

where

$$\begin{aligned} \tau_5 = \; & \frac{1}{40\Delta^4}\Big((3\mathrm{tr}(J^4) + 27C_{J,2} + 15\mathrm{tr}(J^2)C_J + 45\det(J)\mathrm{tr}(J))H_2^2 \\ & + (10C_{J,3} + 50\det(J)\mathrm{tr}(J)C_J + 10\det(J)\mathrm{tr}(J)\mathrm{tr}(J^2) + 2C_{J,2}\mathrm{tr}(J^2) - 28\det(J^2))\|\mathbf{m}\|_2^2 H_2 \\ & + (60\det(J^2)C_J + 3C_{J,4} + 27\det(J^2)\mathrm{tr}(J^2) + 15\det(J)(C_{J,2,3} + C_{J,3,2}))\|\mathbf{m}\|_2^4 \Big) \end{aligned}$$

is a positive constant which depends on the constant function H_2*, the Casimir* $\|\mathbf{m}\|_2$*, and on the entries of the matrix* J*.*

Proof. By direct computation, as in the case of τ_3. The positivity of τ_5 is not immediate, because of the presence of the term $-28\|\mathbf{m}\|_2^2 \det(J^2)H_2$, but can also be verified by expanding the above expression (τ_5 is ultimately a combination of positive terms with positive coefficients). □

4 Constructing higher order integrable approximations

In this section, we wish to use the results derived in the previous section to obtain higher order integrable approximations to the RB equations. To preserve the properties of the integrable discretisations, we propose to simply scale the initial condition (since this amounts to a time reparametrisation for the continuous equations). The scaling must, of course, depend on the step size h of integration.

As our ansatz we choose to scale the initial condition as

$$\frac{h(\Omega(t_k)J + J\Omega(t_k))}{1 + \tilde{\tau}_3 h^2 + \tilde{\tau}_5 h^4 + \cdots}. \tag{4.1}$$

We perform again the backward error analysis. We set now $\tilde{\omega} = \exp(-h\tilde{\Omega}_0 - h^2\tilde{\Omega}_1 + \cdots)$ and solve for the $\tilde{\Omega}_i$s as the skew-symmetric matrices that solve

$$h(1 - \tilde{\tau}_3 h^2 + (\tilde{\tau}_3^2 - \tilde{\tau}_5)h^4 + \cdots)(\Omega J + J\Omega) = \tilde{\omega}^\top J - J\tilde{\omega}. \tag{4.2}$$

It is immediately verified that

$$\tilde{\Omega}_0 = \Omega_0 = \Omega, \qquad \tilde{\Omega}_1 = \Omega_1 = \frac{1}{2}\Omega',$$

hence it is clear that $\tilde{f}_1 = f_1 = 0$ and $\tilde{f}_2 = f_2 = 0$ for the new modified vector field. The first $\tilde{\Omega}_i$ to differ is $\tilde{\Omega}_2$. Equating the h^3 term and taking into account (2.8), we have

$$\tilde{\Omega}_2 J + J\tilde{\Omega}_2 = -\tilde{\tau}_3(\Omega J + J\Omega) - \frac{1}{3!}(\Omega_0^3 J + J\Omega_0^3) - \frac{1}{2!}[(\Omega_0\Omega_1 + \Omega_1\Omega_0), J],$$

hence

$$\tilde{\Omega}_2 = -\tilde{\tau}_3\Omega + \Omega_2.$$

Thus,

$$\tilde{d}_3 \quad = \quad -[\tilde{\Omega}_2, M] + \frac{1}{2!}[\tilde{\Omega}_1, [\tilde{\Omega}_1, M]] = \tilde{\tau}_3[\Omega, M] + d_3 = -\tilde{\tau}_3[M, \Omega] + d_3.$$

Moreover,

$$\begin{aligned} \tilde{f}_3 &= \tilde{d}_3 - \frac{1}{3!}M''' = -\tilde{\tau}_3[M, \Omega] + d_3 - \frac{1}{3!}M''' \\ &= -\tilde{\tau}_3[M, \Omega] + f_3 = (-\tilde{\tau}_3 + \tau_3)[M, \Omega], \end{aligned}$$

hence, in order to have an order-four scheme, we must set $\tilde{f}_3 = 0$ which corresponds to the choice

$$\tilde{\tau}_3 = \tau_3.$$

One could be tempted to extrapolate that $\tilde{\tau}_5 = \tau_5$ does the job of rendering the methods of order six, but unfortunately this is not the case, since there are more complicate nonlinear effects arising from the scaling of the initial condition. Therefore we need to proceed with our backward error analysis.

The equation for $\tilde{\Omega}_3$ is identical to (2.9), whereby Ω_2 is replaced by $\tilde{\Omega}_2$, hence,

$$\tilde{\Omega}_3 = \Omega_3 - \tilde{\tau}_3\Omega'.$$

For what $\tilde{\Omega}_4$ is concerned, we need to take into account also the h^5 term on the left-hand-side of (4.2), hence

$$\begin{aligned} (\tilde{\tau}_3^2 - \tilde{\tau}_5)(\Omega J + J\Omega) &= \tilde{\Omega}_4 J + J\tilde{\Omega}_4 + \frac{1}{2!}[\tilde{\Omega}_3\Omega_0 + \tilde{\Omega}_2\Omega_1 + \Omega_1\tilde{\Omega}_2 + \Omega_0\tilde{\Omega}_3, J] \\ &+ \frac{1}{3!}\Big((\Omega_0^2\tilde{\Omega}_2 + \Omega_0\tilde{\Omega}_2\Omega_0 + \tilde{\Omega}_2\Omega_0^2 + \Omega_0\Omega_1^2 + \Omega_1\Omega_0\Omega_1 + \Omega_1^2\Omega_0)J \\ &\quad + J(\Omega_0^2\tilde{\Omega}_2 + \Omega_0\tilde{\Omega}_2\Omega_0 + \tilde{\Omega}_2\Omega_0^2 + \Omega_0\Omega_1^2 + \Omega_1\Omega_0\Omega_1 + \Omega_1^2\Omega_0)\Big) \\ &+ \frac{1}{4!}[\Omega_0^3\Omega_1 + \Omega_0^2\Omega_1\Omega_0 + \Omega_0\Omega_1\Omega_0^2 + \Omega_1\Omega_0^3, J] \\ &+ \frac{1}{5!}(\Omega_0^5 J + J\Omega_0^5). \end{aligned}$$

from which we deduce

$$\tilde{\Omega}_4 = (\tilde{\tau}_3^2 - \tilde{\tau}_5)\Omega + \frac{1}{2}\tilde{\tau}_3\Omega^3 - \frac{3}{4}\tilde{\tau}_3\Omega'' + \Omega_4.$$

We now compute $\tilde{d}_5$ according to (2.12),

$$\tilde{d}_5 = -(\tilde{\tau}_3^2 - \tilde{\tau}_5)[\Omega, M] + \frac{3}{4}\tilde{\tau}_3[\Omega'', M] - \frac{1}{2}\tilde{\tau}_3[\Omega^3, M] - \frac{3}{4}\tilde{\tau}_3([\Omega, [\Omega', M]] + [\Omega', [\Omega, M]]) + \frac{1}{2}\tilde{\tau}_3[\Omega, [\Omega, [\Omega, M]]] + d_5.$$

Because of the nature of the scaling, it is evident that $\tilde{f}_4 = f_4 = 0$, therefore

$$\tilde{f}_5 = \tilde{d}_5 - \frac{1}{5!}M^{(v)},$$

having chosen $\tilde{\tau}_3$ so that $\tilde{f}_3 = 0$. We have

$$\tilde{f}_5 \quad = \quad -(\tilde{\tau}_3^2 - \tilde{\tau}_5)[\Omega, M] + \frac{3}{4}\tilde{\tau}_3[\Omega'', M] - \frac{1}{2}\tilde{\tau}_3[\Omega^3, M] - \frac{3}{4}\tilde{\tau}_3([\Omega, [\Omega', M]] + [\Omega', [\Omega, M]]) + \frac{1}{2}\tilde{\tau}_3[\Omega, [\Omega, [\Omega, M]]]$$

$$
\begin{aligned}
& \quad + d_5 - \frac{1}{5!}M^{(v)} \\
&= -(\tilde\tau_3^2 - \tilde\tau_5)[\Omega, M] + \frac{3}{4}\tilde\tau_3[\Omega'', M] - \frac{1}{2}\tilde\tau_3[\Omega^3, M] - \frac{3}{4}\tilde\tau_3([\Omega,[\Omega',M]] + [\Omega',[\Omega,M]]) + \frac{1}{2}\tilde\tau_3[\Omega,[\Omega,[\Omega,M]]] \\
& \quad + f_5 + \frac{1}{5!}M^{(v)} + \frac{1}{4}\frac{\mathrm{d}}{\mathrm{d}t}(f'f_3 + f_3'f) - \frac{1}{5!}M^{(v)} \\
&= (\tilde\tau_3^2 - \tilde\tau_5 + \tau_5)[M,\Omega] + \frac{3}{4}\tilde\tau_3[\Omega'', M] - \frac{1}{2}\tilde\tau_3[\Omega^3, M] - \frac{3}{4}\tilde\tau_3([\Omega,[\Omega',M]] + [\Omega',[\Omega,M]]) + \frac{1}{2}\tilde\tau_3[\Omega,[\Omega,[\Omega,M]]] \\
& \quad + \frac{1}{4}\frac{\mathrm{d}}{\mathrm{d}t}(f'f_3 + f_3'f).
\end{aligned}
$$

Since $f_3 = \tau_3[M,\Omega]$, we have $f'f_3 + f_3'f = 2\tau_3 f'f$, hence $\frac{\mathrm{d}}{\mathrm{d}t}(f'f_3 + f_3'f) = 2\tau_3(f''(f,f) + f'f'f) = 2\tau_3 M'''$. Moreover, recall that $f_5 = \tau_5[M,\Omega]$. It follows that

$$
\begin{aligned}
\tilde f_5 &= (\tilde\tau_3^2 - \tilde\tau_5 + \tau_5)[M,\Omega] - \frac{3}{4}\tilde\tau_3[M,\Omega''] + \frac{1}{2}\tilde\tau_3[M,\Omega^3] - \frac{3}{4}\tilde\tau_3([[M,\Omega'],\Omega] + [[M,\Omega],\Omega']) - \frac{1}{2}\tilde\tau_3[[[M,\Omega],\Omega],\Omega] \\
& \quad + \frac{1}{2}\tau_3([[[M,\Omega],\Omega],\Omega] + [[M,\Omega'],\Omega] + 2[[M,\Omega],\Omega'] + [M,\Omega'']) \\
&= (\tilde\tau_3^2 - \tilde\tau_5 + \tau_5)[M,\Omega] - \frac{1}{4}\tilde\tau_3([M,\Omega''] + [[M,\Omega'],\Omega] - [[M,\Omega],\Omega'] - 2[M,\Omega^3]).
\end{aligned}
$$

We recognise that the last term is proportional to $f_3 = \tau_3 f$. Hence,

$$
\tilde f_5 = (\tilde\tau_3^2 - \tilde\tau_5 + \tau_5 - 3\tilde\tau_3^2)[M,\Omega] = (-2\tilde\tau_3^2 - \tilde\tau_5 + \tau_5)[M,\Omega],
$$

and setting $\tilde f_5 = 0$ we find

$$
\tilde\tau_5 = \tau_5 - 2\tau_3^2.
$$

This value of $\tilde\tau_5$ gives indeed a method of order six.

The new proposed algorithms of order four and six are described below.

The DMV4 algorithm:

1. Compute τ_3 and set $M_0 = M_0 h/(1 + h^2\tau_3)$.
2. Compute the roots of (1.8) having positive real parts.
3. For $k = 0, 1, \ldots, n-1$,
   find the unique w_k as in § 1.2 such that $M_k = \omega_k^\top J - J\omega_k$
   set $M_{k+1} = \omega_k M_k \omega_k^\top$
   end
4. Reconstruct $M_n \approx M(t_n) = M_n(1 + h^2\tau_3)/h$.

The DMV6 algorithm:

1. Compute τ_3, τ_5 and set $\tilde\tau_5 = \tau_5 - 2\tau_3^2$ and $M_0 = M_0 h/(1 + h^2\tau_3 + h^4\tilde\tau_5)$.
2. Compute the roots of (1.8) having positive real parts.
3. For $k = 0, 1, \ldots, n-1$,
   find the unique w_k as in § 1.2 such that $M_k = \omega_k^\top J - J\omega_k$
   set $M_{k+1} = \omega_k M_k \omega_k^\top$
   end
4. Reconstruct $M_n \approx M(t_n) = M_n(1 + h^2\tau_3 + h^4\tilde\tau_5)/h$.

5 How large can the step size h be?

The choice of the step size is intimately related to the accuracy of the method. Other limitations to the choice of the step size for a method can be loss of stability or convergence (for instance for an implicit scheme). For the DMV-type algorithms described in this paper, there is a further limitation, which comes from the spectrum S of the polynomial $P(\lambda)$ in (1.8): when two opposite-sign roots become purely imaginary, then the choice of ω in the Moser–Veselov equation (1.6) is no longer unique; in other words the relation (1.6) cannot give rise to a one-to-one map.

In this section we intend to investigate this limit on the step size h for the original DMV algorithm and the new proposed algorithms DMV4 and DMV6.

In Figure 5.3 we plot real and imaginary parts of the eigenvalues λ_i of the characteristic equation (1.8) versus step size h for our running test problem (see §7). The roots are computed using MATLAB's routine `roots` for finding zeros of polynomials. When h is very small, all the roots are real (they correspond to plus/minus the squares of the moments of inertia in J). For larger steps, the typical behaviour is that some roots become complex.

For the DMV algorithm (circles joined by a dotted line) the maximum allowed step size is $h \approx 1.2$; for this value, two couples of opposite-sign complex and conjugate roots hit the imaginary axis. This occurres also for the DMV6 algorithm, though at a larger step size, $h \approx 2.2$; afterwards the spectrum recovers its original splitting (1.7). Remarkably, the roots of the DMV4 algorithm do not appear to hit the imaginary axis at all for this test problem and preserve the desired spectrum splitting (1.7) for all values of h. Does

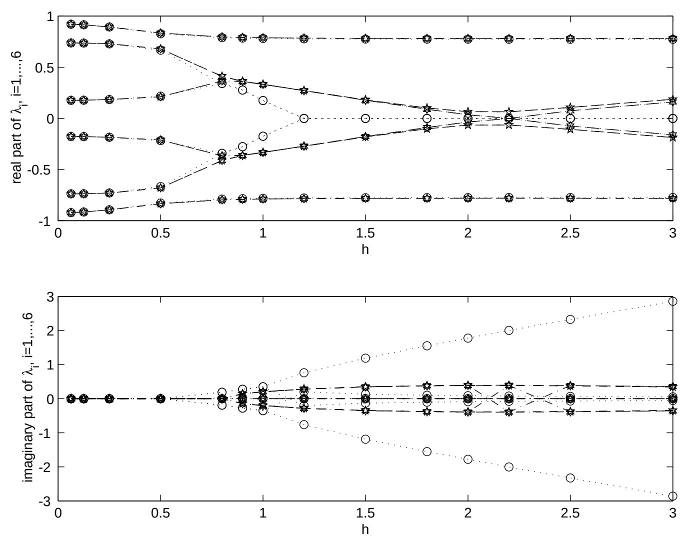

Figure 5.3: Real (top) and imaginary (bottom) part of the eigenvalues λ_i as a function of the step size h. Legend: DMV corresponds to circles joined by dotted line; DMV4 to pentagrams joined by dash-dotted line; DMV6 to hexagons joined by dashed line

this mean that the limitation on the step size are dictated only by the accuracy required in the computations? The answer is no: let us look more carefully at the polynomial $-P(\lambda)$ in (1.8). Assume that $\mathbf{m}_0$ is the given initial condition, that we scale through the algorithms DMV4 and DMV6 to $\mathbf{m}_0/(1+h^2\tau_3)$ for DMV4 and $\mathbf{m}_0/(1+h^2\tau_3+h^4\tilde{\tau}_5)$ for DMV6. Let us denote by $-P_4(\lambda)$ and $-P_6(\lambda)$ the corresponding characteristic polynomials in which we factor out the dependence on the scaling from the initial condition,

$$-P_{4,h}(\lambda) = \lambda^6 - \Big(J_1^2 + J_2^2 + J_3^2 - \frac{h^2}{(1+h^2\tau_3)^2}\|\mathbf{m}_0\|^2\Big)\lambda^4 \qquad (5.1)$$

$$+\Big(J_1^2J_2^2+J_1^2J_3^2+J_2^2J_3^2-\frac{h^2}{(1+h^2\tau_3)^2}H_2\Big)\lambda^2-J_1^2J_2^2J_3^2$$

$$-P_{6,h}(\lambda) \quad = \quad \lambda^6-\Big(J_1^2+J_2^2+J_3^2-\frac{h^2}{\big(1+h^2\tau_3+h^4\tilde{\tau}_5\big)^2}\|\mathbf{m}_0\|^2)\lambda^4 \qquad (5.2)$$

$$+\Big(J_1^2J_2^2+J_1^2J_3^2+J_2^2J_3^2-\frac{h^2}{(1+h^2\tau_3+h^4\tilde{\tau}_5)^2}H_2\Big)\lambda^2-J_1^2J_2^2J_3^2,$$

where $H_2=\frac{H\Delta}{\|\mathbf{m}_0\|^2(J_1J_2+J_1J_3+J_2J_3)}$ is independent of h. By a similar token,

$$-P_h(\lambda)=\lambda^6-(J_1^2+J_2^2+J_3^2-h^2\|\mathbf{m}_0\|^2)\lambda^4+(J_1^2J_2^2+J_1^2J_3^2+J_2^2J_3^2-h^2H_2)\lambda^2-J_1^2J_2^2J_3^2.$$

We consider the polynomial as parametrised by the h and we analyse their roots letting $h\to\infty$. Starting with P_h, we have

$$-P_\infty(\lambda)=\|\mathbf{m}_0\|^2\lambda^4-H_2\lambda^2.$$

Two pure imaginary roots are have disappeared at $\pm\infty$, two pure imaginary roots go to zero, and two stay real. In particular, this means that (1.6) cannot be made into a map which is symplectic.

The situation is different for the scaled methods DMV4 and DMV6. Since

$$\lim_{h\to\infty}\frac{h^2}{(1+h^2\tau_3)^2}=\lim_{h\to\infty}\frac{h^2}{(1+h^2\tau_3+h^4\tilde{\tau}_5)^2}=0,$$

we have

$$-P_{4,\infty}=-P_{6,\infty}=-P_{4,0}=-P_{6,0}=\lambda^6-(J_1^2+J_2^2+J_3^2)\lambda^4+(J_1^2J_2^2+J_1^2J_3^2+J_2^2J_3^2)\lambda^2-J_1^2J_2^2J_3^2,$$

which is the same as having $h=0$ — in other words, no motion! Hence, the map is still integrable, symplectic and momentum preserving, but the motion becomes slower and slower until it stops at infinity. This seems to indicate that there exist optimal values of the stepsize for moderate h and we plan to explore the phenomenon in more details in a latter paper.

The behaviour of the roots of $-P_h,-P_{4,h},-P_{6,h}$ for large h is displayed below in Figure 5.4.

6 RATTLE $\equiv$ DMV

In this section we consider the RATTLE algorithm for the integration of the RB and prove that RATTLE is the same as DMV.

In brief we mention that the RATTLE algorithm can be derived as a Lie–Poisson integrator and it is described at a larger detail in (Hairer et al. 2002). A reformulation of RATTLE for the RB equations is in (Hairer et al. 2002), pg. 247, which we report here for convenience. Note that we replace h with $-h$ to obtain the same equations. The notation is as in this paper.

The RATTLE algorithm for the RB equations:
With the same notation of this paper, $M(t_0)=J\Omega(t_0)+\Omega(t_0)J$ is the initial condition.

1. Set $Y_0=J\Omega(t_0)$.
2. For $k=0,1,2,\ldots,n-1$,

- determine a symmetric matrix S such that $Q=I-hJ^{-1}Y_{1/2}$ is orthogonal, $Y_{1/2}=Y_0-hS$;

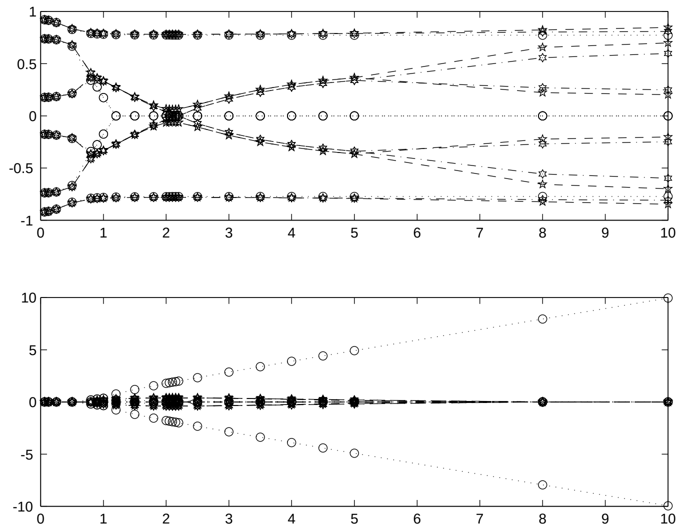

Figure 5.4: Real (top) and imaginary (bottom) part of the eigenvalues λ_i as a function of the step size h letting $h \to \infty$. Legend: DMV corresponds to circles joined by dotted line; DMV4 to pentagrams joined by dash-dotted line; DMV6 to hexagons joined by dashed line

- determine a symmetric matrix T such that $J^{-1}Y_1$ is skew-symmetric, where $Y_1 = Y_0 - hY_{1/2}Y_{1/2}^\top J^{-1} - hT$;
- set $Y_0 = Y_1$;

end

3. Reconstruct $\Omega_n = J^{-1}Y_1 \approx \Omega(t_n)$, $M_n = J\Omega_n + \Omega_n J \approx M(t_n)$.

The first step in 2. is equivalent to solving the Ricatti equation

$$J^{-1}S + SJ^{-1} = -(Y_0 - hS)^\top J^{-2}(Y_0 - hS).$$

This method is of order 2, it preserves the Casimir $\|\mathbf{m}\|_2^2$ and the Hamiltonian H in (3.1).

Theorem 6.1 *The RATTLE method for the RB equation is equivalent to the DMV algorithm.*

Proof. It is sufficient to demonstrate that the two algorithms are equivalent for the first step.

By construction, we have $M(t_0) = J\Omega(t_0) + \Omega(t_0)J = Y_0 - Y_0^\top$, and $Y_0 = \frac{J(I-Q)}{h} + hS$, where S is a symmetric matrix determined so that Q is orthogonal. Hence, by direct computation,

$$hM(t_0) = h(Y_0 - Y_0^\top) = Q^\top J - JQ,$$

namely, Q is a solution of the Moser–Veselov equation (1.6). Similarly, since T is a symmetric matrix and $Y_0 - hS = J(I-Q)/h$,

$$\begin{aligned} h(Y_1 - Y_1^\top) &= h(Y_0 - Y_0^\top) - h^2 \frac{J(I-Q)}{h}\frac{(I-Q^\top)J}{h}J^{-1} + h^2 J^{-1}\frac{J(I-Q)}{h}\frac{(I-Q^\top)J}{h} \\ &= Q^\top J - JQ - J(2I - Q - Q^\top) + (2I - Q - Q^\top)J \\ &= JQ^\top - QJ, \end{aligned}$$

which, compared with (2.1), reveals that $Y_1 - Y_1^\top$ is precisely the DMV approximation for $M(t_1) = M(t_0 + h)$, since, for sufficiently small h, the DMV gives the only order-2 solution of (2.1) consistent with the dynamics of the flow. Hence RATTLE coincides with DMV. $\square$

As a consequence, our results on the DMV can be immediately be extended to RATTLE for the Rigid Body, and it is of no surprise that the method preserves both the Hamiltonian and the Casimirs of the system, since the method is integrable.

Moreover, our analysis allows us to improve the order of the RATTLE approximation to order 4 and 6 simply by scaling the initial condition and the scaling back the final approximation!

The RATTLE4, RATTLE6 algorithms:

1. Compute τ_3 for order 4 and $\tau_3, \tau_5, \tilde{\tau}_5 = \tau_5 - 2\tau_3^2$ for order 6. Set $Y_0 = J\Omega(t_0)/(1 + h^2\tau_3)$ for order 4 and $Y_0 = J\Omega(t_0)/(1 + h^2\tau_3 + h^4\tilde{\tau}_5)$ for order 6.
2. For $k = 0, 1, 2, \dots, n-1$,
   - determine a symmetric matrix S such that $Q = I - hJ^{-1}Y_{1/2}$ is orthogonal, $Y_{1/2} = Y_0 - hS$;
   - determine a symmetric matrix T such that $J^{-1}Y_1$ is skew-symmetric, where $Y_1 = Y_0 - hY_{1/2}Y_{1/2}^\top J^{-1} - hT$;
   - set $Y_0 = Y_1$;

   end
3. Reconstruct $\Omega_n = (1 + h^2\tau_3)J^{-1}Y_1 \approx \Omega(t_n)$, $M_n = J\Omega_n + \Omega_n J \approx M(t_n)$ for order 4 and $\Omega_n = (1 + h^2\tau_3 + h^4\tilde{\tau}_5)J^{-1}Y_1 \approx \Omega(t_n)$, $M_n = J\Omega_n + \Omega_n J \approx M(t_n)$ for order 6.

This equivalence is particularly interesting since the RATTLE algorithm can be efficiently implemented by using quaternions (Hairer 2003).

7 Comparison with other methods

In this section we compare the numerical results of the DMV algorithm with the new proposed schemes DMV4 and DMV6. Moreover, we also compare with

- the explicit second-order Lie–Poisson integrator for the RB of McLachlan (LP2) (McLachlan 1993), which is obtained by splitting the Hamiltonian H in three pieces, $H = \mathcal{H}_1 + \mathcal{H}_2 + \mathcal{H}_3$, whose corresponding vector fields can be integrated exactly, generating the flows $\phi_{1,h}, \phi_{2,h}, \phi_{3,h}$. Then, at each step, the numerical approximation is computed as

  $$\mathbf{m}_{k+1} = \phi_{3,h/2} \circ \phi_{2,h/2} \circ \phi_{1,h} \circ \phi_{2,h/2} \circ \phi_{3,h/2}(\mathbf{m}_k), \qquad k = 0, 1, 2, \dots;$$

  and
- the well known implicit midpoint rule (IMR), reading

  $$y_{k+1} = y_k + hf\left(\frac{y_k + y_{k+1}}{2}\right), \qquad k = 0, 1, 2 \dots,$$

  for the problem $y' = f(y)$, which is second-order, energy and momentum preserving – but implicit.

In Figure 7.5 we display the error of the methods versus step size $h = \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$, and then $0.8, 1, 1.2, 1.5, 2.2, 2.5, 4$. The error is computed at $T = 100$ for our running example,

$$\mathbf{m}_0 = \begin{pmatrix} 0.4165 \\ 0.9072 \\ 0.0577 \end{pmatrix}, \qquad J = \begin{pmatrix} 0.9218 & 0 & 0 \\ 0 & 0.7382 & 0 \\ 0 & 0 & 0.1762 \end{pmatrix}.$$

(Without loss of generality, $\mathbf{m}_0$ is normalised so that $\|\mathbf{m}_0\|_2 = 1$.) The slope of the lines agrees with the order of the methods under consideration. The original DMV method is the one that has lowest accuracy and the most limitations on the step size: it fails at around $h = 1.2$ (see also Table 4) never to recover. Also DMV6 fails, but at around $h = 2.2$ to recover afterwards.

Method	$h = \frac{1}{16}$	$h = \frac{1}{2}$	$h = 1.2$	$h = 2.2$	$h = 2.5$	$h = 4$
LP2	6.7903e-03	5.1043e-01	1.6055e+00	1.7002e+00	1.9489e+00	1.3902e+00
IMR	1.5494e-04	9.9329e-03	1.3119e-01	3.9514e-01	2.5276e-01	6.1905e-01
DMV	1.5014e-02	5.9899e-01	NaN	NaN	NaN	NaN
DMV4	1.757e-07	7.6167e-04	1.0785e-01	1.6094e+00	5.0245e-01	5.1624e-01
DMV6	1.962e-10	1.6440e-06	7.0269e-02	NaN	7.0407e-01	1.0519e-01

Table 4: Error for the various methods and selected step sizes

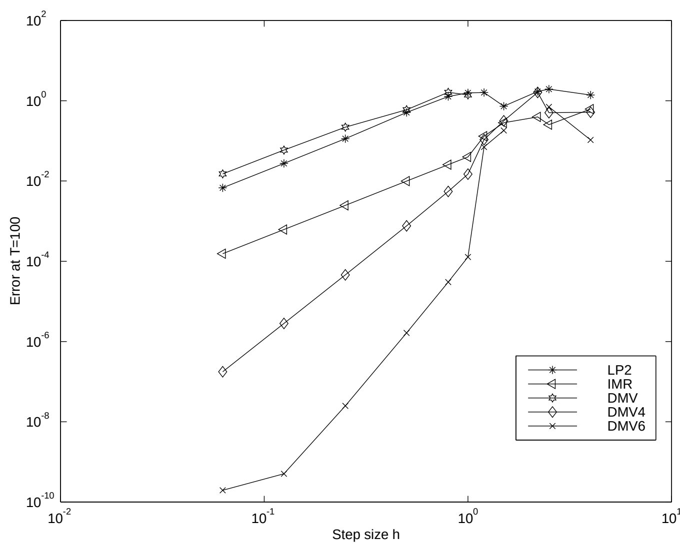

Figure 7.5: Error versus step size computed at $T = 100$ for the methods LP2, IMR, DMV, DMV4, DMV6.

In Figure 7.6 we plot instead the cost of the methods (in floating point operations) versus their accuracy. We have not taken into consideration the computation of τ_3 and $\tilde{\tau}_5$, but these are very small and done only once for all the computations, hence irrelevant to the overall estimate of the methods. As one can see, the methods DMV, DMV4 and DMV6 have the same floating point operations – they only differ on the scaling of the initial condition. The gain in accuracy for the latter two is however so significant that DMV6 is the method which is the most efficient, especially when good accuracy is required.

The DMV, DMV4 and DMV6 are implemented using the algorithm described in the appendix and we have done very little effort to optimise the computations.

Finally, in Figure 7.7 we plot the solutions obtained with the methods LP2, IMR, DMV4 and DMV6 for a large value of h, $h = 5$. The method LP2 performs very poorly in this case, the solution loses completely

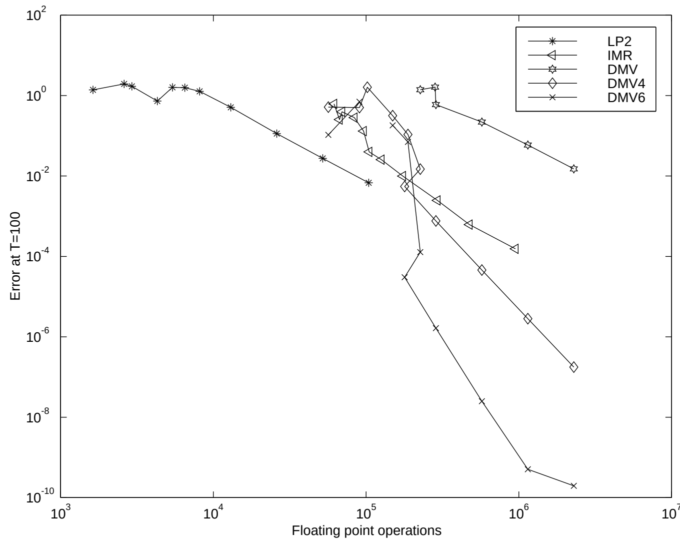

Figure 7.6: Floating point operations versus accuracy ($T = 100$) for the methods LP2, IMR, DMV, DMV4, DMV6.

its shape, DMV is not plotted because it does not generate a solution for this choice of the step size. As our analysis predicted, DMV4 and DMV6 still produce a solution, which has precisely the same integrals of the original problem, but is much too slow compared with the true solution. Remarkably, the IMR still preserves some form and the integrals of the original problem. For this test example the IMR fails to converge at $h \approx 6.5$.

For comparison, in Figure 7.8 we plot the solutions for $h = 1$. In this case, both IMR, DMV4 and DMV6 give a very good agreement with the exact solution.

8 On the explicit solution of the Moser–Veselov equation for the 3×3 rigid body

8.1 Reduction to a matrix Ricatti equation

Consider the matrix equation

$$M = X^\top J - JX, \tag{8.1}$$

where M is a skew-symmetric matrix, and J is diagonal. Cardoso & Leite (2001) have shown that every solution of (8.1) (not necessarily orthogonal) can be written as

$$X = J^{-1}(-M/2 + S),$$

for some symmetric matrix S. The proof is immediate. Clearly, if S is symmetric, $X = J^{-1}(-M/2+S)$ is a solution of (8.1). Next, assume that X solves (8.1) and set $\tilde{X} = JX$. Then (8.1) becomes $\tilde{X}^\top - \tilde{X} = M$. Recall that any matrix $\tilde{X}$ can be uniquely decomposed in its symmetric and skew-symmetric part,

$$\tilde{X} = \frac{1}{2}(\tilde{X} + \tilde{X}^\top) + \frac{1}{2}(\tilde{X} - \tilde{X}^\top),$$

from which follows that the skew-symmetric part of $\tilde{X}$ is precisely $-M/2$, while the symmetric part is an arbitray symmetric matrix S. From this it follows immediately that $X = J^{-1}(-M/2 + S)$.

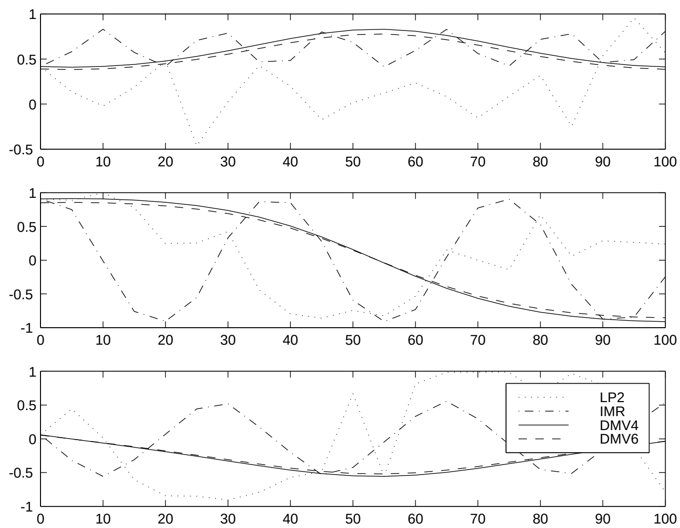

Figure 7.7: Numerical approximations of the RB equations for h large ($h = 5$).

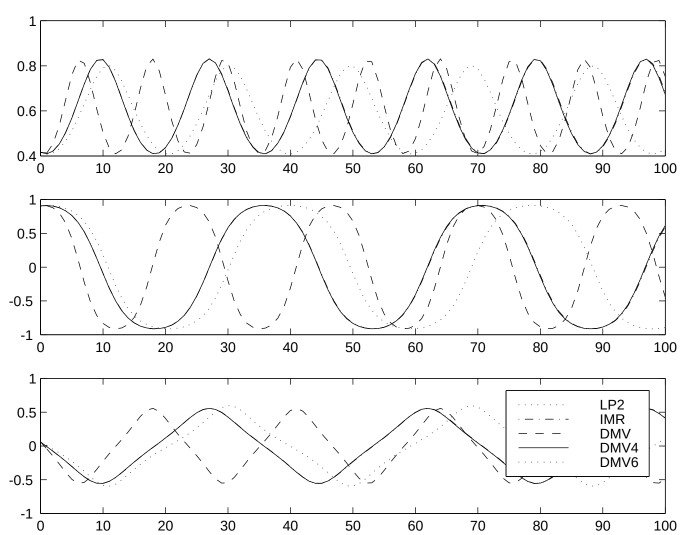

Figure 7.8: Numerical approximations of the RB equations for $h = 1$.

Furthermore, they have shown that X is an orthogonal solution of (8.1) if and only if S is a symmetric solution of the Ricatti equation

$$SS^\top + S(M/2) + (M/2)^\top S^\top - (M^2/4 + J^2) = 0. \tag{8.2}$$

Thus, the problem is reduced to the well known problem of solving Ricatti equations.

Ricatti equations can be associated to symplectic matrices. For this problem, the corresponding symplectic matrix is

$$H_{\mathrm{sympl}} = \begin{bmatrix} \frac{M}{2} & I \\ \frac{M^2}{4} + J^2 & \frac{M}{2} \end{bmatrix}. \tag{8.3}$$

If $\frac{M^2}{4} + J^2$ is positive definite, it has been shown in (Cardoso & Leite 2001) that (8.2) has a unique solution S which is symmetric, positive definite, and such that the eigenvalues of

$$W = (-M/2 + S)^\top$$

have positive real parts. This matrix W is precisely the same matrix in Moser & Veselov (1991) such that $WW^\top = J^2$ and

$$\omega = J^{-1}W^\top.$$

In this case, Cardoso & Leite (2001) propose the following algorithm for the computation of X, as the unique solution of (8.1) in the special orthogonal group $\mathrm{SO}(N)$.

1. Find a real Schur form of H_{sympl},

$$\tilde{Q}^\top H_{\mathrm{sympl}} \tilde{Q} = \begin{bmatrix} T_{11} & T_{12} \\ O & T_{22} \end{bmatrix}, \tag{8.4}$$

   where T_{11} and T_{22} are block upper-triangular matrices such that the real parts of the spectrum of T_{11} are positive and the real parts of the spectrum of T_{22} are negative definite.

2. Partition $\tilde{Q}$ accordingly,

$$\tilde{Q} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}.$$

   Then, compute

$$S = Q_{21} Q_{11}^{-1}.$$

3. Compute

$$X = J^{-1}\left(-\frac{M}{2} + S\right).$$

A real Schur form of H_{sympl} can be computed using the QR iteration for eigenvalues (Golub & van Loan 1989), which is a $\mathcal{O}(N^3)$ operation for a $N \times N$ matrix M, a cost that is comparable with that of using implicit ODE methods for the solution of the rigid body equations.

In what follows, we focus on the particular case when $N = 3$. In this case, the eigenvalues of H_{sympl} are explicitely known (they are the roots of the polynomial $P(\lambda)$), hence it is possible to find an explicit spectral decomposition of H_{sympl} (without using the QR eigenvalue method). Only the eigenvectors corresponding to the roots λ_1, λ_2 and λ_3 of (1.8) with positive real part need be computed. We construct the partial real Schur decomposition (8.4) of this eigenspace and hence X. This yields an *explicit* numerical method for thereduced RB equations.

To this purpose, assume that $\boldsymbol{x}_\lambda = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix}$ is an eigenvector of H_{sympl} with eigenvalue λ. From

$$\begin{bmatrix} \frac{M}{2} & I \\ \frac{M^2}{4} + J^2 & \frac{M}{2} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix}$$

we deduce that $\boldsymbol{x}_2 = \lambda \boldsymbol{x}_1 - \frac{M}{2}\boldsymbol{x}_1$, hence $\boldsymbol{x}_1$ is an eigenvector for the quadratic eigenvalue problem

$$J^2 \boldsymbol{x}_1 + \lambda M \boldsymbol{x}_1 = \lambda^2 \boldsymbol{x}_1. \tag{8.5}$$

8.2 Computation of the eigenvectors

For the irreducible case (all the roots are real $\lambda_1, \lambda_2, \lambda_3 > 0$), we solve the quadratic eigenvalue problem (8.5)

$$A_{\lambda_i} \boldsymbol{x}_{1,\lambda_i} = (J^2 - \lambda_i^2 I + \lambda_i M)\boldsymbol{x}_{1,\lambda_i} = 0, \qquad i = 1, 2, 3,$$

by computing the QR factorization of $A_{\lambda_i} = Q_{\lambda_i} R_{\lambda_i}$, where R_{λ_i} is now an upper-triangular matrix with the last diagonal element equal to zero (A_{λ_i} is singular, rank-2). We choose the third component of $\boldsymbol{x}_{1,\lambda_i} \neq 0$ arbitrary and determine the remaining components by solving the upper triangular system

$$R_{\lambda_i} \boldsymbol{x}_{1,\lambda_i} = 0.$$

Then, we construct $\boldsymbol{x}_{2,\lambda_i} = \lambda_i \boldsymbol{x}_{1,\lambda_i} - \frac{M}{2}\boldsymbol{x}_{1,\lambda_i}$, and $\boldsymbol{x}_{\lambda_i} = \begin{bmatrix} \boldsymbol{x}_{1,\lambda_i} \\ \boldsymbol{x}_{2,\lambda_i} \end{bmatrix}$

The procedure is analogous in the case of one real and two complex and conjugate roots, except for the fact that the complex eigenvalues require complex arithmetic. In that case, complex arithmetic is avoided in the usual way, computing real and imaginary part of the corresponding eigenvectors. Specifically, assume that

$$\lambda = \mu + \mathrm{i}\nu$$

is a complex eigenvalue with eigenvector $\boldsymbol{y}_1 = \boldsymbol{u}_1 + \mathrm{i}\boldsymbol{v}_1$. From the eigenvector equation $J^2 \boldsymbol{y}_1 + \lambda M \boldsymbol{y}_1 - \lambda^2 \boldsymbol{y}_1 = 0$, separating real and imaginary part we obtain

$$\begin{aligned} J^2\boldsymbol{u}_1 + \mu M\boldsymbol{u}_1 - \nu M\boldsymbol{v}_1 + (\nu^2 - \mu^2)\boldsymbol{u}_1 + 2\mu\nu\boldsymbol{v}_1 &= \mathbf{0}, \\ J^2\boldsymbol{v}_1 + \nu M\boldsymbol{u}_1 + \mu M\boldsymbol{v}_1 - 2\mu\nu\boldsymbol{u}_1 + (\nu^2 - \mu^2)\boldsymbol{v}_1 &= \mathbf{0}, \end{aligned}$$

which can be written as

$$\begin{bmatrix} J^2 + \mu M + (\nu^2 - \mu^2)I & -\nu + 2\mu\nu I \\ \nu M - 2\mu\nu I & J^2 + \mu M + (\nu^2 - \mu^2)I \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{v}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

If $\mu \neq 0$, the 6×6 coefficient-matrix on the right-hand-side of the above linear system has rank four. The vectors $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ can be computed by performing a QR factorization factorization of the coefficient matrix (eventually, with column pivoting) and then by solving an upper triangular system by backward substitution.

Once $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ are computed, we compute also

$$\begin{aligned} \boldsymbol{u}_2 &= \mu\boldsymbol{u}_1 - \nu\boldsymbol{v}_1 - \frac{M}{2}\boldsymbol{u}_1 \\ \boldsymbol{v}_2 &= \nu\boldsymbol{u}_1 + \mu\boldsymbol{v}_1 - \frac{M}{2}\boldsymbol{v}_1, \end{aligned}$$

hence, we set $\mathbf{u} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix}$.

Note that it is not necessary to compute the real Schur form of H_{sympl} explicitely. What is needed is only the an orthogonal basis that spans the eigenvectors space. This basis can be obtained either by QR or simply by a Grahm–Schmit orthogonalization.

- Construct the 6×3 matrix $V = [\boldsymbol{x}_{\lambda_1}, \boldsymbol{x}_{\lambda_2}, \boldsymbol{x}_{\lambda_3}]$ if λ_i, $i = 1, 2, 3$ are all real and positive; or $V = [\boldsymbol{x}_{\lambda_1}, \boldsymbol{u}, \boldsymbol{v}]$ if λ_1 is real and positive and $\lambda_2 = \bar{\lambda}_3$ are the two complex and conjugate roots with real part $\mu > 0$.

- Compute the QR decomposition (or the Grahm–Schmit orthogonalization) $\tilde{V}R = V$, and decompose $\tilde{V} = \begin{bmatrix} \tilde{V}_1 \\ \tilde{V}_2 \end{bmatrix}$, where $\tilde{V}_1, \tilde{V}_2$ are 3×3 blocks.

- Compute $S = \tilde{V}_2 \tilde{V}_1^{-1}$
- Compute $W = -M/2 + S$
- Compute $\omega = J^{-1} W^\top$.

This is the core of the explicit algorithm used to implement step 2 of DMV and step 3 of DMV4,DM6 for the 3×3 RB.

In view of the connection with RATTLE, it would be interesting to see whether the RATTLE algorithm can be implemented more efficiently, although we have done very little effort to optimise our implementation.

Acknowledgments

This work was initiated while the authors were guest of the program "Special year in Geometric Integration", hosted at the Center for Advanced Study, Oslo. We wish to thank the staff for making our stay so pleasant and productive. Thanks to Ben Leimkhuler and Sebastian Reich for indicating possible connections with the RATTLE algorithm.

References

Abramowitz, M. & Stegun, I. A. (1965), *Handbook of Mathematical Functions*, Dover, New York.

Arnold, V. I. (1989), *Mathematical Methods of Classical Mechanics*, Vol. 60 of *GTM*, second edn, Springer-Verlag.

Cardoso, J. R. & Leite, F. S. (2001), The Moser–Veselov equation, To appear in *Lin. Alg. Appl.*

Deift, P., Li, L.-C. & Tomei, C. (1992), 'Loop groups, discrete versions of some classical integrable systems and rank-2 extensions', *Memoirs of the American Mathematical Society* **100**(479), 1–101.

Golub, G. H. & van Loan, C. F. (1989), *Matrix Computations*, 2nd edn, John Hopkins, Baltimore.

Hairer, E. (2003), 'Global modified Hamiltonian for constrained symplectic integrators', *Numer. Math.* **95**, 325–336.

Hairer, E., Lubich, C. & Wanner, G. (2002), *Geometric Numerical Integration*, number 31 *in* 'Springer Series in Computational Mathematics', Springer, Berlin.

Iserles, A., Munthe-Kaas, H., Nørsett, S. P. & Zanna, A. (2000), 'Lie-group methods', *Acta Numerica* **9**, 215–365.

Marsden, J. E. & West, M. (2001), 'Discrete mechanics and variational integrators', *Acta Numerica* **10**, 357–514.

Marsden, J. E., Pekarsky, S. & Shkoller, S. (1999), 'Discrete Euler–Poincaré and Lie–Poisson equations', *Nonlinearity* **12**, 1647–1662.

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

Moser, J. & Veselov, A. P. (1991), 'Discrete Version of Some Classical Integrable Systems and Factorization of Matrix Polynomials', *Commun. Math. Phys.* **139**, 217–243.

Munthe-Kaas, H. Z. & Krogstad, S. (2002), On enumeration problems in Lie–Butcher theory, Report in informatics, Institutt for informatikk, University of Bergen, Norway. To appear in *Future Generation Computer Systems*, Special issue.