COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING
Commun. Numer. Meth. Engng (in press)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm.963

Dynamically equivalent implicit algorithms for the integration of rigid body rotations

P. Krysl*, †

University of California, San Diego, 9500 Gilman Dr #0085, La Jolla, CA 92093-0085, U.S.A.

SUMMARY

Two midpoint-trapezoid pairs of dynamically equivalent (conjugate) algorithms are derived as compositions of first-order forward Euler and backward Euler integrators as applied to an incremental form of the initial-value problem of three-dimensional rigid body rotation. The algorithms are related to the recently developed methodology of the so-called Munthe-Kaas Runge–Kutta methods. Selected examples are used to illustrate the excellent long-term integration properties. Copyright © 2006 John Wiley & Sons, Ltd.

Received 20 April 2006; Accepted 11 October 2006

KEY WORDS: rigid body dynamics; time integration; trapezoidal rule; midpoint rule; symplectic; angular momentum conservation; dynamic equivalence; composition of maps; Munthe-Kaas Runge–Kutta methods

1. INTRODUCTION

It is well-known that for the time integration of vector-space Hamiltonian mechanics the midpoint rule and the trapezoidal rule are dynamically equivalent [1]. The present paper addresses the question whether there is a corresponding pair of dynamically equivalent algorithms for the initial-value ordinary differential equation problem that describes three-dimensional rigid body rotations. The tool used in the construction of these algorithms is the composition of maps. It is common knowledge that both the midpoint rule and the trapezoidal rule in the vector-space setting result from the composition of the forward and backward (backward and forward, respectively) half-step Euler methods. In the first part, this procedure is reviewed to provide a template for the second part of the manuscript. Following the template, not one, but two midpoint-trapezoid pairs of dynamically equivalent (conjugate) algorithms are derived as compositions of first-order forward Euler and backward Euler integrators as applied to an incremental form of the initial-value problem.

*Correspondence to: P. Krysl, University of California, San Diego, 9500 Gilman Dr #0085, La Jolla, CA 92093-0085, U.S.A.
†E-mail: pkrysl@ucsd.edu

Copyright © 2006 John Wiley & Sons, Ltd.

The properties of these algorithms are investigated using a few previously published integrators as a reference. In particular, the algorithms are viewed in the context of the recently developed methodology of the so-called Munthe-Kaas Runge–Kutta methods [2]. A few selected examples are used to illustrate the excellent properties of the proposed algorithms, especially the stable and well-behaved response for very long integration intervals.

2. DYNAMICS ON VECTOR SPACES

The initial-value problem for a mechanical system (for instance, a system of interacting particles) described by a vector of configuration variables (displacements) $\mathbf{u} \in \mathbb{R}^n$ may be stated as

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{f}, \quad \mathbf{p}(0) = \mathbf{p}_0 \\ \dot{\mathbf{u}} &= \mathbf{M}^{-1}\mathbf{p}, \quad \mathbf{u}(0) = \mathbf{u}_0 \end{aligned} \tag{1}$$

where $\dot{\mathbf{p}}$ is the rate of linear momentum, $\dot{\mathbf{u}}$ is the velocity, and $\mathbf{f} = \mathbf{f}(\mathbf{u}, t)$ is the applied force. For simplicity we shall assume a time-independent mass matrix $\mathbf{M}$. The initial values are $\mathbf{p}_0$, and $\mathbf{u}_0$.

Using subscripts to indicate the time to which a given quantity belongs, and writing $\mathbf{f}_{t+h} = \mathbf{f}(\mathbf{u}_{t+h}, t+h)$, we may formulate a *forward Euler* time step applied to (1) as

$$\begin{aligned} \mathbf{p}_{t+h} &= \mathbf{p}_t + h\mathbf{f}_t \\ \mathbf{u}_{t+h} &= \mathbf{u}_t + h\mathbf{M}^{-1}\mathbf{p}_t \end{aligned} \tag{2}$$

and the *backward Euler* approximation as

$$\begin{aligned} \mathbf{p}_{t+h} &= \mathbf{p}_t + h\mathbf{f}_{t+h} \\ \mathbf{u}_{t+h} &= \mathbf{u}_t + h\mathbf{M}^{-1}\mathbf{p}_{t+h} \end{aligned} \tag{3}$$

Sequential composition of algorithms (2) and (3) in this order with time steps $h = \Delta t/2$ yields the *trapezoidal rule* [1]

$$\begin{aligned} \mathbf{p}_{t+\Delta t} &= \mathbf{p}_t + \frac{\Delta t}{2}(\mathbf{f}_t + \mathbf{f}_{t+\Delta t}) \\ \mathbf{u}_{t+\Delta t} &= \mathbf{u}_t + \frac{\Delta t}{2}\mathbf{M}^{-1}(\mathbf{p}_t + \mathbf{p}_{t+\Delta t}) \end{aligned} \tag{4}$$

Sequential composition of algorithms (3) and (2) with time steps $h = \Delta t/2$ yields the *midpoint rule* [1]

$$\begin{aligned} \mathbf{p}_{t+\Delta t} &= \mathbf{p}_t + \Delta t\mathbf{f}_{t+\Delta t/2} \\ \mathbf{u}_{t+\Delta t} &= \mathbf{u}_t + \frac{\Delta t}{2}\mathbf{M}^{-1}\mathbf{p}_{t+\Delta t/2} \end{aligned} \tag{5}$$

The above algorithms have been derived as forward or backward Euler approximations to the derivatives in Equation (1), but, importantly, it would equally make sense to understand them as

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

approximations of the integrals in these, equivalent, equations

$$\begin{aligned}\mathbf{p}_{t+h} &= \mathbf{p}_t + \int_t^{t+h} \mathbf{f}(\tau)\,\mathrm{d}\tau \\ \mathbf{u}_{t+h} &= \mathbf{u}_t + \mathbf{M}^{-1}\int_t^{t+h} \mathbf{p}(\tau)\,\mathrm{d}\tau\end{aligned} \tag{6}$$

For the vector-space problem there is no advantage to be gained from either form, but as we shall see next, these two approaches yield different algorithms for the rotational dynamics.

3. DYNAMICS OF ROTATIONS

An excellent discussion of the difficulties of interpolating on the curved manifold SO(3), which is an appropriate setting for this problem, has been given in Reference [3]. To begin our presentation, we shall show how to formulate rotational dynamics algorithms to bring out the parallels to the above vector-space trapezoidal/midpoint dynamically equivalent couple in Equations (4) and (5).

The initial-value problem may be written in the convected description (body frame) as

$$\begin{aligned}\dot{\mathbf{\Pi}} &= -\mathrm{skew}[\mathbf{I}^{-1}\mathbf{\Pi}]\mathbf{\Pi} + \mathbf{T}, \quad \mathbf{\Pi}(0) = \mathbf{\Pi}_0 \\ \dot{\mathbf{R}} &= \mathbf{R}\,\mathrm{skew}[\mathbf{I}^{-1}\mathbf{\Pi}], \quad \mathbf{R}(0) = \mathbf{R}_0\end{aligned} \tag{7}$$

where $\dot{\mathbf{\Pi}}$ is the rate of body-frame angular momentum, $\mathbf{R}$ is the rotation matrix (tensor), $\mathbf{R}^{-1} = \mathbf{R}^{\mathrm{T}}$ (orthogonal operator), $\mathbf{T} = \mathbf{T}(t, \mathbf{R}(t))$ is the applied torque in the body frame, skew[$\bullet$] is defined by $\mathrm{skew}[\mathbf{w}] \cdot \mathbf{w} = \mathbf{0}$, and $\mathbf{I}$ is the time-independent tensor of inertia in the body frame.

3.1. Vector parametrization

The second equation (7) is not in a form suitable for forward or backward Euler discretization: the rotation tensor constitutes points of the Lie group SO(3), which is not a vector space and linear combinations are not legal operations on the rotation tensors. Therefore, an inevitable loss of orthogonality of the rotation tensor would result when the time stepping was applied directly. To transform the initial-value problem to a form suitable for our purposes, we shall introduce the rotation vector representation of the rotation tensor.

As is standard, the equation of motion is written in the spatial frame as $\dot{\boldsymbol{\pi}} = \mathbf{RT}$, where $\boldsymbol{\pi} = \mathbf{R\Pi}$ is the spatial angular momentum. Integrating the spatial equation of motion, and converting back to the body frame, we may write the equation of motion in integral form in the body frame as

$$\mathbf{\Pi}(t + \Delta t) = \exp[-\mathbf{\Psi}]\left(\mathbf{\Pi}(t) + \mathbf{R}^{\mathrm{T}}(t)\int_t^{t+\Delta t} \mathbf{R}(\tau)\mathbf{T}(\tau)\,\mathrm{d}\tau\right) \tag{8}$$

where $\exp[-\mathbf{\Psi}] = \exp[-\mathrm{skew}\,\mathbf{\Psi}] = \mathbf{R}^{\mathrm{T}}(t+\Delta t)\mathbf{R}(t)$ is the incremental rotation through vector $(-\mathbf{\Psi})$. By time differentiation of Equation (8), and by comparison with the original differential equation of motion, we obtain

$$\dot{\mathbf{\Psi}} = (d\exp_{-\mathbf{\Psi}})^{-1}\mathbf{I}^{-1}\mathbf{\Pi}$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

where $d\exp_{-\boldsymbol{\Psi}}$ is the differential of the exponential map [3]

$$d\exp_{\boldsymbol{\Theta}} = \mathbf{1} + \frac{1-\cos\|\boldsymbol{\Theta}\|}{\|\boldsymbol{\Theta}\|^2}\text{skew}\,\boldsymbol{\Theta} + \left(1 - \frac{\sin\|\boldsymbol{\Theta}\|}{\|\boldsymbol{\Theta}\|}\right)\frac{\text{skew}\,\boldsymbol{\Theta}^2}{\|\boldsymbol{\Theta}\|^2} \tag{9}$$

The initial-value problem (in the second equation it is of incremental nature) may be therefore rewritten as

$$\begin{aligned} \dot{\boldsymbol{\Pi}} &= -\text{skew}[\mathbf{I}^{-\mathbf{1}}\boldsymbol{\Pi}]\boldsymbol{\Pi} + \mathbf{T}, \quad \boldsymbol{\Pi}(0) = \boldsymbol{\Pi}_0 \\ \dot{\boldsymbol{\Psi}} &= (d\exp_{-\boldsymbol{\Psi}})^{-1}\mathbf{I}^{-1}\boldsymbol{\Pi}, \quad \boldsymbol{\Psi}(0) = \mathbf{0} \end{aligned} \tag{10}$$

3.2. Forward and backward Euler algorithms

We may write the forward Euler step for these two differential equations as

$$\boldsymbol{\Pi}_{t+h} = \boldsymbol{\Pi}_t - h\,\text{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_t]\boldsymbol{\Pi}_t + h\mathbf{T}_t$$

$$\boldsymbol{\Psi}_{t+h} = \boldsymbol{\Psi}_t + h(d\exp_{-\boldsymbol{\Psi}_t})^{-1}\mathbf{I}^{-1}\boldsymbol{\Pi}_t$$

Since the incremental rotation vector is such that $\boldsymbol{\Psi}_t = \mathbf{0}$, and $(d\exp_{\mathbf{0}})^{-1} = \mathbf{1}$, we obtain the *forward Euler* approximation as

$$\begin{aligned} \boldsymbol{\Pi}_{t+h} &= \boldsymbol{\Pi}_t - h\,\text{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_t]\boldsymbol{\Pi}_t + h\mathbf{T}_t \\ \boldsymbol{\Psi}_{t+h} &= h\mathbf{I}^{-1}\boldsymbol{\Pi}_t \end{aligned} \tag{11}$$

The backward Euler approximation is similarly written as

$$\boldsymbol{\Pi}_{t+h} = \boldsymbol{\Pi}_t - h\,\text{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+h}]\boldsymbol{\Pi}_{t+h} + h\mathbf{T}_{t+h}$$

$$\boldsymbol{\Psi}_{t+h} = h(d\exp_{-\boldsymbol{\Psi}_{t+h}})^{-1}\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+h}$$

which may be simplified by noting

$$d\exp_{\boldsymbol{\Psi}_{t+h}}\boldsymbol{\Psi}_{t+h} = \boldsymbol{\Psi}_{t+h}$$

to give the *backward Euler* step

$$\begin{aligned} \boldsymbol{\Pi}_{t+h} &= \boldsymbol{\Pi}_t - h\,\text{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+h}]\boldsymbol{\Pi}_{t+h} + h\mathbf{T}_{t+h} \\ \boldsymbol{\Psi}_{t+h} &= h\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+h} \end{aligned} \tag{12}$$

It is well-known that these two methods in the vector-space setting are *mutual adjoints* [1]. This means that for the forward Euler method Φ_h^{fE} we can undo its step by applying the backward Euler method with the reversed time step Φ_{-h}^{bE}:

$$\Phi_{-h}^{\text{bE}} \circ \Phi_h^{\text{fE}} = \text{Id}$$

with Id the identity map, and also the backward Euler step Φ_h^{bE} may be undone with Φ_{-h}^{fE}

$$\Phi_{-h}^{\text{fE}} \circ \Phi_h^{\text{bE}} = \text{Id}$$

Since the availability of this property is not obvious for (11) and (12), we present a brief proof.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

First, we show that $\Phi^{\mathrm{bE}}_{-h} \circ \Phi^{\mathrm{fE}}_{h} = \mathrm{Id}$ holds. The step Φ^{fE}_{h} may be written

$$\boldsymbol{\Pi}_1 = \boldsymbol{\Pi}_0 - h\,\mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_0]\boldsymbol{\Pi}_0 + h\mathbf{T}_0, \quad \mathbf{R}_1 = \mathbf{R}_0 \exp[h\mathbf{I}^{-1}\boldsymbol{\Pi}_0]$$

with the obvious notation $\bullet_0 = \bullet_t$ and $\bullet_1 = \bullet_{t+h}$. Next, the step Φ^{bE}_{-h} is applied to the quantities $\boldsymbol{\Pi}_1$ and $\mathbf{R}_1$, and reads

$$\boldsymbol{\Pi}_2 = \boldsymbol{\Pi}_1 + h\,\mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_2]\boldsymbol{\Pi}_2 - h\mathbf{T}_2, \quad \mathbf{R}_2 = \mathbf{R}_1 \exp[-h\mathbf{I}^{-1}\boldsymbol{\Pi}_2]$$

with the notation $\bullet_2 = \bullet_{t+h-h}$. Substituting for $\boldsymbol{\Pi}_1$ one obtains

$$\begin{aligned} \boldsymbol{\Pi}_2 &= \boldsymbol{\Pi}_0 - h\,\mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_0]\boldsymbol{\Pi}_0 + h\mathbf{T}_0 + h\,\mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_2]\boldsymbol{\Pi}_2 - h\mathbf{T}_2 \\ \mathbf{R}_2 &= \mathbf{R}_1 \exp[h\mathbf{I}^{-1}\boldsymbol{\Pi}_2] = \mathbf{R}_0 \exp[h\mathbf{I}^{-1}\boldsymbol{\Pi}_0] \exp[-h\mathbf{I}^{-1}\boldsymbol{\Pi}_2] \end{aligned} \tag{13}$$

with the obvious solution $\boldsymbol{\Pi}_2 = \boldsymbol{\Pi}_0$, $\mathbf{R}_2 = \mathbf{R}_0$. (By definition of the torque, $\mathbf{T}_2 = \mathbf{T}(t+h-h, \mathbf{R}_2) = \mathbf{T}(t, \mathbf{R}_0) = \mathbf{T}_0$.)

Now, we show that $\Phi^{\mathrm{fE}}_{-h} \circ \Phi^{\mathrm{bE}}_{h} = \mathrm{Id}$ holds. The step Φ^{bE}_{h} may be written

$$\boldsymbol{\Pi}_1 = \boldsymbol{\Pi}_0 - h\,\mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_1]\boldsymbol{\Pi}_1 + h\mathbf{T}_1, \quad \mathbf{R}_1 = \mathbf{R}_0 \exp[h\mathbf{I}^{-1}\boldsymbol{\Pi}_1]$$

and the step Φ^{fE}_{-h} is applied to the quantities $\boldsymbol{\Pi}_1$ and $\mathbf{R}_1$

$$\boldsymbol{\Pi}_2 = \boldsymbol{\Pi}_1 + h\,\mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_1]\boldsymbol{\Pi}_1 - h\mathbf{T}_1, \quad \mathbf{R}_2 = \mathbf{R}_1 \exp[-h\mathbf{I}^{-1}\boldsymbol{\Pi}_1]$$

Substituting for $\boldsymbol{\Pi}_1$ and $\mathbf{R}_1$ it trivially follows $\boldsymbol{\Pi}_2 = \boldsymbol{\Pi}_0$, $\mathbf{R}_2 = \mathbf{R}_0$.

3.3. TRAP: trapezoidal algorithm

Composition of the forward Euler step (11) followed by the backward Euler step (12), both with $h = \Delta t/2$, yields the following *trapezoidal rule*:

$$\begin{aligned} \boldsymbol{\Pi}_{t+\Delta t} &= \boldsymbol{\Pi}_t + \frac{\Delta t}{2}(-\mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_t]\boldsymbol{\Pi}_t + \mathbf{T}_t - \mathrm{skew}[\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+\Delta t}]\boldsymbol{\Pi}_{t+\Delta t} + \mathbf{T}_{t+\Delta t}) \\ \boldsymbol{\Psi}_{t+\Delta t/2} &= \frac{\Delta t}{2}\mathbf{I}^{-1}\boldsymbol{\Pi}_t, \quad \boldsymbol{\Psi}_{t+\Delta t} = \frac{\Delta t}{2}\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+\Delta t} \end{aligned} \tag{14}$$

The second line should be understood as producing two incremental rotation vectors, so that the update of the orthogonal rotation tensor is

$$\mathbf{R}_{t+\Delta t} = \mathbf{R}_t \exp[\boldsymbol{\Psi}_{t+\Delta t/2}] \exp[\boldsymbol{\Psi}_{t+\Delta t}]$$

The trapezoidal rule is formally given in the algorithm **TRAP**: (note that the torque at time t_n depends on the current configuration, $\mathbf{T}_n = \mathbf{T}(t_n, \mathbf{R}_n)$).

Algorithm **TRAP**
Given $\boldsymbol{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations

$$\mathbf{R}_n = \mathbf{R}_{n-1} \exp\left[\tfrac{\Delta t}{2}\mathrm{skew}\left[\mathbf{I}^{-1}\boldsymbol{\Pi}_{n-1}\right]\right] \exp\left[\tfrac{\Delta t}{2}\,\mathrm{skew}\left[\mathbf{I}^{-1}\boldsymbol{\Pi}_n\right]\right]$$

$$\boldsymbol{\Pi}_n = \boldsymbol{\Pi}_{n-1} + \tfrac{\Delta t}{2}\left(-\mathrm{skew}\left[\mathbf{I}^{-1}\boldsymbol{\Pi}_{n-1}\right]\boldsymbol{\Pi}_{n-1} + \mathbf{T}_{n-1} - \mathrm{skew}\left[\mathbf{I}^{-1}\boldsymbol{\Pi}_n\right]\boldsymbol{\Pi}_n + \mathbf{T}_n\right)$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

3.4. IMID: midpoint algorithm

Similarly, composition of the backward Euler step (12) followed by the forward Euler step (11), both with $h = \Delta t/2$, yields the following *midpoint rule*:

Algorithm **IMID**:
Given $\mathbf{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations
$\overline{\mathbf{\Pi}} = \mathbf{\Pi}_{n-1} - \frac{\Delta t}{2}\,\mathrm{skew}[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}]\overline{\mathbf{\Pi}} + \frac{\Delta t}{2}\mathbf{T}_{n-1/2}$
$\mathbf{R}_n = \mathbf{R}_{n-1}\exp\left[\Delta t\,\mathrm{skew}\left[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}\right]\right]$
$\mathbf{\Pi}_n = \mathbf{\Pi}_{n-1} - \Delta t\,\mathrm{skew}[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}]\overline{\mathbf{\Pi}} + \Delta t\mathbf{T}_{n-1/2}$

The torque is evaluated as $\mathbf{T}_{n-1/2} = \mathbf{T}(t_n + \Delta t/2, \mathbf{R}_{n-1}\exp[(\Delta t/2)\,\mathrm{skew}[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}]])$.

3.5. Properties of TRAP and IMID

It is easy to ascertain that neither algorithm **TRAP** nor **IMID** will conserve spatial momenta in the absence of external forcing. The reason is evidently our use of the rate balance equation.

It is of considerable interest that the algorithm **IMID** conserves kinetic energy (in the absence of forcing). The kinetic energy in time step n is written as

$$K_n = \tfrac{1}{2}\mathbf{\Pi}_n^{\mathrm{T}}\mathbf{I}^{-1}\mathbf{\Pi}_n$$

Using the first equation in **IMID** we may express

$$\overline{\mathbf{\Pi}} - \mathbf{\Pi}_{n-1} = -\frac{\Delta t}{2}\,\mathrm{skew}[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}]\overline{\mathbf{\Pi}} \tag{15}$$

which upon substitution into the last equation in **IMID** yields

$$\mathbf{\Pi}_n = 2\overline{\mathbf{\Pi}} - \mathbf{\Pi}_{n-1}$$

Therefore, the kinetic energy in time step n may be rewritten

$$\begin{aligned} K_n &= \tfrac{1}{2}\mathbf{\Pi}_n^{\mathrm{T}}\mathbf{I}^{-1}\mathbf{\Pi}_n = \tfrac{1}{2}\mathbf{\Pi}_{n-1}^{\mathrm{T}}\mathbf{I}^{-1}\mathbf{\Pi}_{n-1} + 2\overline{\mathbf{\Pi}}^{\mathrm{T}}\mathbf{I}^{-1}\overline{\mathbf{\Pi}} - 2\overline{\mathbf{\Pi}}^{\mathrm{T}}\mathbf{I}^{-1}\mathbf{\Pi}_{n-1} \\ &= K_{n-1} + 2\overline{\mathbf{\Pi}}^{\mathrm{T}}\mathbf{I}^{-1}(\overline{\mathbf{\Pi}} - \mathbf{\Pi}_{n-1}) \end{aligned} \tag{16}$$

and we can show that the last term is identically zero by substituting from (15) and using $\mathbf{a}^{\mathrm{T}}\mathrm{skew}[\mathbf{a}] = \mathbf{0}$ (product of a skew-symmetric matrix with its axial vector)

$$2\overline{\mathbf{\Pi}}^{\mathrm{T}}\mathbf{I}^{-1}(\overline{\mathbf{\Pi}} - \mathbf{\Pi}_{n-1}) = 2\overline{\mathbf{\Pi}}^{\mathrm{T}}\mathbf{I}^{-1}\left(-\frac{\Delta t}{2}\,\mathrm{skew}[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}]\overline{\mathbf{\Pi}}\right) = 0$$

The algorithm **TRAP** does not conserve energy at the time stations t_{n-1}, t_n. However, **TRAP** and **IMID** are *conjugate* algorithms [1] as we have

$$\Phi_{\Delta t}^{\mathrm{T}} = \Phi_{\Delta t/2}^{\mathrm{bE}} \circ \Phi_{\Delta t/2}^{\mathrm{fE}}, \quad \Phi_{\Delta t}^{\mathrm{M}} = \Phi_{\Delta t/2}^{\mathrm{fE}} \circ \Phi_{\Delta t/2}^{\mathrm{bE}}$$

and consequently

$$\Phi_{\Delta t}^{\mathrm{M}} = (\Phi_{\Delta t/2}^{\mathrm{bE}})^{-1} \circ \Phi_{\Delta t}^{\mathrm{T}} \circ \Phi_{\Delta t/2}^{\mathrm{bE}}$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

where Φ^{T} is **TRAP**, Φ^{M} is **IMID**, Φ^{fE} is the forward Euler (11), and Φ^{bE} is the backward Euler (12). Thus, the algorithm **TRAP** produces at the time instant t_n quantities that can be pushed to time $t_{n+1/2}$ by one half-step of the forward Euler (11). These same quantities are available by stepping with the **IMID** algorithm starting from the initial values at $t_{1/2}$, and then advancing with the midpoint algorithm by a full step. Therefore, the trapezoidal rule algorithm **TRAP** actually conserves the 'midpoint' energy exactly in torque-free motion. In this sense, both **IMID** and **TRAP** are energy-conserving methods.

3.6. Momentum-conserving TRAPM and IMIDM algorithms

Concerning the conservation of angular momentum: there is an alternative, since Equation (8) expresses evolution of the body-frame momenta in integral form (this device has been also used by Simo and Wong to derive a momentum-conserving integrator [4]). Writing the initial-value problem in the integral form

$$\begin{aligned}
&\boldsymbol{\Pi}(t+h) = \exp[-\boldsymbol{\Psi}(t+h)]\left(\boldsymbol{\Pi}(t) + \mathbf{R}^{\mathrm{T}}(t)\int_t^{t+h}\mathbf{R}(\tau)\mathbf{T}(\tau)\,\mathrm{d}\tau\right), \quad \boldsymbol{\Pi}(t) = \boldsymbol{\Pi}_t \\
&\boldsymbol{\Psi}(t+h) = \int_t^{t+h}(d\exp_{-\boldsymbol{\Psi}})^{-1}\mathbf{I}^{-1}\boldsymbol{\Pi}(\tau)\,\mathrm{d}\tau, \quad \boldsymbol{\Psi}(t) = \mathbf{0}
\end{aligned} \tag{17}$$

we therefore arrive, with the simplifications discussed below Equation (11), at the *momentum-conserving* form of the *forward Euler* discretization as

$$\begin{aligned}
\boldsymbol{\Pi}_{t+h} &= \exp[-\boldsymbol{\Psi}_{t+h}](\boldsymbol{\Pi}_t + h\mathbf{T}_t) \\
\boldsymbol{\Psi}_{t+h} &= h\mathbf{I}^{-1}\boldsymbol{\Pi}_t
\end{aligned} \tag{18}$$

The *momentum-conserving* form of the *backward Euler* approximation is similarly written as

$$\begin{aligned}
\boldsymbol{\Pi}_{t+h} &= \exp[-\boldsymbol{\Psi}_{t+h}](\boldsymbol{\Pi}_t + h\exp[\boldsymbol{\Psi}_{t+h}]\mathbf{T}_{t+h}) \\
\boldsymbol{\Psi}_{t+h} &= h\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+h}
\end{aligned} \tag{19}$$

Therefore, composing Equations (18) and (19) with time step $h = \Delta t/2$, we can write the *momentum-conserving trapezoidal rule* as

$$\begin{aligned}
&\boldsymbol{\Pi}_{t+\Delta t} = \exp[-\boldsymbol{\Psi}_{t+\Delta t}]\exp[-\boldsymbol{\Psi}_{t+\Delta t/2}]\left(\boldsymbol{\Pi}_t + \frac{\Delta t}{2}\mathbf{T}_t\right) + \frac{\Delta t}{2}\mathbf{T}_{t+\Delta t} \\
&\boldsymbol{\Psi}_{t+\Delta t/2} = \frac{\Delta t}{2}\mathbf{I}^{-1}\boldsymbol{\Pi}_t, \quad \boldsymbol{\Psi}_{t+\Delta t} = \frac{\Delta t}{2}\mathbf{I}^{-1}\boldsymbol{\Pi}_{t+\Delta t}
\end{aligned} \tag{20}$$

Evidently, the trapezoidal rule exactly conserves spatial momenta in the absence of external torques. It is summarized in the algorithm **TRAPM**:

Algorithm **TRAPM**:
Given $\boldsymbol{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations

$$\mathbf{R}_n = \mathbf{R}_{n-1}\exp\left[\tfrac{\Delta t}{2}\,\mathrm{skew}\left[\mathbf{I}^{-1}\boldsymbol{\Pi}_{n-1}\right]\right]\exp\left[\tfrac{\Delta t}{2}\mathrm{skew}\left[\mathbf{I}^{-1}\boldsymbol{\Pi}_n\right]\right]$$

$$\boldsymbol{\Pi}_n = \mathbf{R}_n^{\mathrm{T}}\mathbf{R}_{n-1}\left(\boldsymbol{\Pi}_{n-1} + \tfrac{\Delta t}{2}\mathbf{T}_{n-1}\right) + \tfrac{\Delta t}{2}\mathbf{T}_n$$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

Analogously, we may formulate the dynamically equivalent *momentum-conserving midpoint rule* by composing Equations (19) and (18) with time step $h = \Delta t/2$, arriving at the algorithm **IMIDM**.

Algorithm **IMIDM**:
Given $\mathbf{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations
$\mathbf{\Psi}_n = \Delta t \mathbf{I}^{-1}(\exp[-\frac{1}{2}\widetilde{\mathbf{\Psi}}_n]\mathbf{\Pi}_{n-1} + \frac{\Delta t}{2}\mathbf{T}_{n-1/2})$
$\mathbf{R}_n = \mathbf{R}_{n-1}\exp[\widetilde{\mathbf{\Psi}}_n]$
$\mathbf{\Pi}_n = \exp[-\widetilde{\mathbf{\Psi}}_n]\mathbf{\Pi}_{n-1} + \Delta t \exp[-\frac{1}{2}\widetilde{\mathbf{\Psi}}_n]\mathbf{T}_{n-1/2}$

Note that the midpoint torque is calculated as $\mathbf{T}_{n-1/2} = \mathbf{T}(t_n + \Delta t/2, \mathbf{R}_{n-1}\exp[\frac{1}{2}\widetilde{\mathbf{\Psi}}_n])$. The algorithm **IMIDM** has been analysed in the open literature; see for instance [5–7]. It was also introduced in References [8, 9] as **LIEMID[I]** to serve a discussion of explicit composition algorithms for the rigid body rotation.

3.7. Properties of TRAPM and IMIDM

Their behaviour in long-term integrations depends on which quantities (if any) are conserved. The design of the algorithms equipped both with the conservation of the spatial angular momentum. Let us now look at the other properties.

A quick calculation shows that neither of the algorithms conserves exactly kinetic energy in torque-free motion. It appears from numerical evidence (the fact that the energy error oscillates but is bounded for long times, and also that the scaling of the energy error with the time step squared yields approximately the same magnitude of the energy error), and from an incomplete proof, that the algorithm **IMIDM** is symplectic (for a readable discussion of the symplecticity of some dynamics algorithms see Reference [10]). Similar to the **IMID**/**TRAP** pair, the pair **TRAPM**/**IMIDM** is a composition of forward and backward Euler steps. Therefore, **TRAPM** and **IMIDM** are also conjugate algorithms. **TRAPM** may or may not be symplectic, but it shares the excellent long-term behaviour with **IMIDM**.

Integration on manifolds is currently attracting much attention in the mathematical literature. It will be useful to analyse the algorithms **TRAPM**/**IMIDM** in the setting of the so-called Munthe-Kaas Runge–Kutta methods that have been formally described by Munthe-Kaas [2]. Paraphrased in our notation these methods read (an actual algorithm requires the specification of a tableau)

Algorithm **MKRK**:
Given $\mathbf{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
for $k = 1, \ldots, s$ stages
$\mathbf{\Psi}_{(k)} = \Delta t \sum_{j=1}^{s} a_{kj}(d\exp_{-\mathbf{\Psi}_{(j)}})^{-1}\mathbf{I}^{-1}\mathbf{\Pi}_{(j)}$
$\mathbf{R}_{(k)} = \mathbf{R}_{n-1}\exp[\mathbf{\Psi}_{(k)}]$
$\mathbf{\Pi}_{(k)} = \exp[-\mathbf{\Psi}_{(k)}]\left(\mathbf{\Pi}_{n-1} + \Delta t \sum_{j=1}^{s} a_{kj}\exp[\mathbf{\Psi}_{(j)}]\mathbf{T}_{(k)}\right)$
end
$\mathbf{\Psi}_n = \Delta t \sum_{j=1}^{s} b_j(d\exp_{-\mathbf{\Psi}_{(j)}})^{-1}\mathbf{I}^{-1}\mathbf{\Pi}_{(j)}$
$\mathbf{R}_n = \mathbf{R}_{n-1}\exp[\mathbf{\Psi}_n]$
$\mathbf{\Pi}_n = \exp[-\mathbf{\Psi}_n]\left(\mathbf{\Pi}_{n-1} + \Delta t \sum_{j=1}^{s} b_j\exp[\mathbf{\Psi}_{(j)}]\mathbf{T}_{(j)}\right)$

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

where $\mathbf{T}_{(j)} = \mathbf{T}(t_{n-1} + c_j \Delta t, \mathbf{R}_{(j)})$, and a_{kj}, b_j, and c_j are the coefficients of the Runge–Kutta tableau.

The forward and backward Euler sub-algorithms (18) and (19) may be recognized as Munthe-Kaas Runge–Kutta methods with the tableaus

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

for the forward Euler, and

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

for the backward Euler. These methods are mutual adjoints, as proven in Section 3.2, and therefore the midpoint algorithm **IMIDM** is self-adjoint (symmetric)

$$(\Phi^*_{-\Delta t/2} \circ \Phi_{\Delta t/2})^* = \Phi^*_{-\Delta t/2} \circ \Phi_{\Delta t/2}$$

This method is being discussed by Celledoni and Owren [7] (referred to as implicit midpoint) in relation to time-symmetry and long-term properties of integrators on manifolds. The conjugate algorithm **TRAPM** is also symmetric.

The **IMIDM** algorithm is a composition of two Munthe-Kaas Runge–Kutta methods, and it is itself also classified as a Munthe-Kaas Runge–Kutta method with the tableau

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array}$$

The conjugate algorithm **TRAPM** does not seem to be associated to any Munthe-Kaas Runge–Kutta method. In particular, the trapezoidal Runge–Kutta rule in the context of the Munthe-Kaas Runge–Kutta method yields the Bottasso and Borri algorithm **BBTRAPWD** [11] to be discussed next as one of the reference methods.

4. COMPARABLE INTEGRATORS

We shall be comparing the present two pairs of methods with three well-known second-order algorithms. Simo and Wong [4] derived an *ad hoc* momentum- and energy-conserving algorithm.

Algorithm **SWC1**:
Given $\mathbf{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations
$\mathbf{\Psi}_n = \frac{\Delta t}{2}(\mathbf{I}^{-1}\mathbf{\Pi}_n + \mathbf{I}^{-1}\mathbf{\Pi}_{n-1})$
$\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\mathbf{\Psi}_n]$
$\mathbf{\Pi}_n = \exp[-\mathbf{\Psi}_n]\mathbf{\Pi}_{n-1} + \Delta t \exp[-\frac{1}{2}\mathbf{\Psi}_n]\mathbf{T}_{n-1/2}$

Austin *et al.* [12] proposed the following spatial momentum- and energy-conserving algorithm (it was shown to conserve the Hamiltonian even for the heavy top, not only for torque-free motion).

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

Algorithm **AKW**:
Given $\mathbf{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations
$\mathbf{\Pi}_n = \mathbf{\Pi}_{n-1} - \Delta t\ \mathrm{skew}[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}]\overline{\mathbf{\Pi}} + \frac{\Delta t}{2}(\mathbf{T}_{n-1} + \mathbf{T}_n)$
$\mathbf{R}_n = \mathbf{R}_{n-1}\,\mathrm{cay}[\Delta t\ \mathrm{skew}[\mathbf{I}^{-1}\overline{\mathbf{\Pi}}]]$
where $\overline{\mathbf{\Pi}} = \frac{1}{2}(\mathbf{\Pi}_n + \mathbf{\Pi}_{n-1})$

Bottasso and Borri [11] proposed an energy-conserving algorithm based on a modification of the classical Runge–Kutta. It is really a Munthe-Kaas Runge–Kutta method [2] with the tableau (the trapezoidal rule)

0	0	0
1	1/2	1/2
	1/2	1/2

and truncation of the $(d\ \exp_\bullet)^{-1}$ operator. Written out in full, but with the stages of the Runge–Kutta algorithm compressed whenever possible, the algorithm with the $(d\,\exp_\bullet)^{-1}$ operator reads

Algorithm **BBTRAPWD** (with the map $(d\ \exp_\bullet)^{-1}$):
Given $\mathbf{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations
$\mathbf{\Psi}_n = \frac{\Delta t}{2}((d\ \exp_{-\mathbf{\Psi}_n})^{-1}\mathbf{I}^{-1}\mathbf{\Pi}_n + \mathbf{I}^{-1}\mathbf{\Pi}_{n-1})$
$\mathbf{R}_n = \mathbf{R}_{n-1}\exp[\mathbf{\Psi}_n]$
$\mathbf{\Pi}_n = \exp[-\mathbf{\Psi}_n](\mathbf{\Pi}_{n-1} + \frac{\Delta t}{2}\mathbf{T}_{n-1}) + \frac{\Delta t}{2}\mathbf{T}_n$

Bottasso and Borri have shown that truncating the $(d\ \exp_\bullet)^{-1}$ furnishes energy conservation, while preserving the asymptotic rate of convergence.

Algorithm **BBTRAP** (truncation: $(d\ \exp_\bullet)^{-1} \approx \mathbf{1}$):
Given $\mathbf{\Pi}_{n-1}, \mathbf{R}_{n-1}$,
Solve as a coupled system of equations
$\mathbf{\Psi}_n = \frac{\Delta t}{2}(\mathbf{I}^{-1}\mathbf{\Pi}_n + \mathbf{I}^{-1}\mathbf{\Pi}_{n-1})$
$\mathbf{R}_n = \mathbf{R}_{n-1}\ \exp[\mathbf{\Psi}_n]$
$\mathbf{\Pi}_n = \exp[-\mathbf{\Psi}_n](\mathbf{\Pi}_{n-1} + \frac{\Delta t}{2}\mathbf{T}_{n-1}) + \frac{\Delta t}{2}\mathbf{T}_n$

Interestingly, the algorithm **BBTRAP** is almost identical to the algorithm of Simo and Wong [4] (in fact, for torque-free motion it *is* identical). **BBTRAP** differs from the proposed **TRAPM** in subtle, but evidently important, ways.

5. EXAMPLES

5.1. Torque-free motion

This example is discussed in the report [13] in the context of discrete Moser–Veselov integrators for the rigid body. The initial condition is $\mathbf{\Omega} = (0.45549, 0.82623, 0.03476)$, and the diagonal entries of the inertia tensor are $\mathrm{diag}\,\mathbf{I} = (0.9144, 1.098, 1.66)$.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

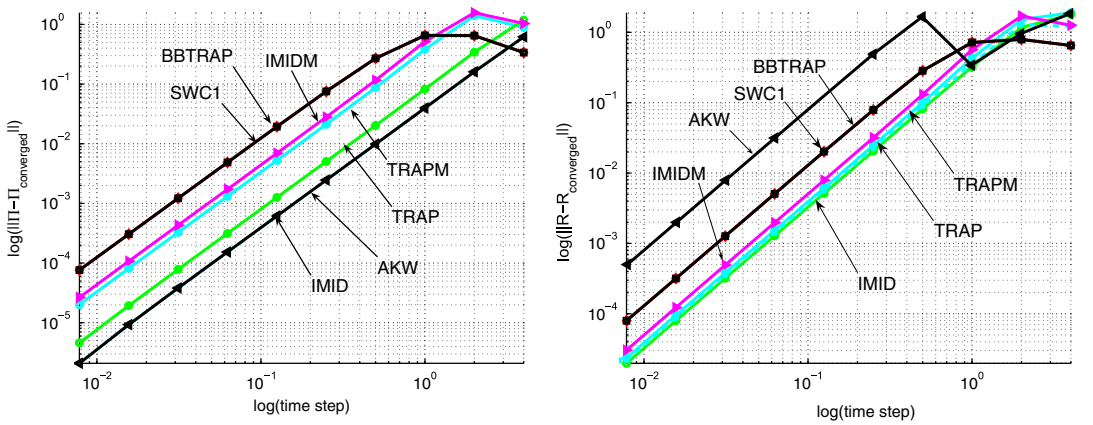

Figure 1. Free body rotation; on the left-hand side convergence in the norm of the error in body-frame angular momentum; on the right-hand side convergence in the norm of the error in the attitude matrix: **AKW**: implicit midpoint rule of Austin *et al.* [12]; **BBTRAP**: Bottasso and Borri [11]; **SWC1**: implicit Simo and Wong algorithm ALGO_C1 [4]; **TRAP**: trapezoidal rule; **IMID**: midpoint rule; **TRAPM**: momentum-conserving trapezoidal rule; **IMIDM**: momentum-conserving midpoint rule.

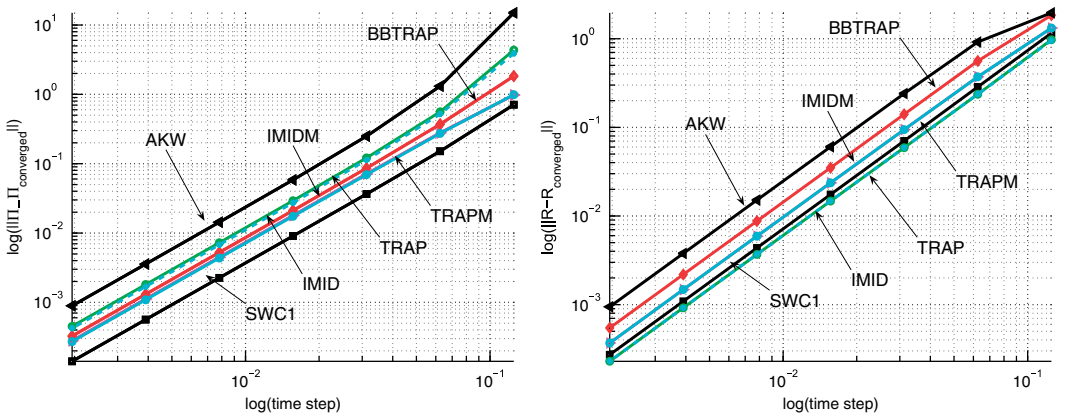

Figure 2. Slow Lagrangian top; on the left-hand side convergence in the norm of the error in body-frame angular momentum; on the right-hand side convergence in the norm of the error in the attitude matrix: **AKW**: implicit midpoint rule of Austin *et al.* [12]; **BBTRAP**: Bottasso and Borri [11]; **SWC1**: implicit Simo and Wong algorithm ALGO_C1 [4]; **TRAP**: trapezoidal rule; **IMID**: midpoint rule; **TRAPM**: momentum-conserving trapezoidal rule; **IMIDM**: momentum-conserving midpoint rule.

Figure 1 illustrates the second-order accuracy of all the algorithms introduced above in the norms $\|\mathbf{R}-\mathbf{R}_{\text{converged}}\|_2$ and $\|\mathbf{\Pi}-\mathbf{\Pi}_{\text{converged}}\|_2$, where the orientation matrix $\mathbf{R}_{\text{converged}}=\mathbf{R}(t=100)$ and the angular momentum $\mathbf{\Pi}_{\text{converged}}=\mathbf{\Pi}(t=100)$ have been obtained with an extremely small step size of 0.001. Interestingly, the most accurate algorithm both in the body-frame momentum

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

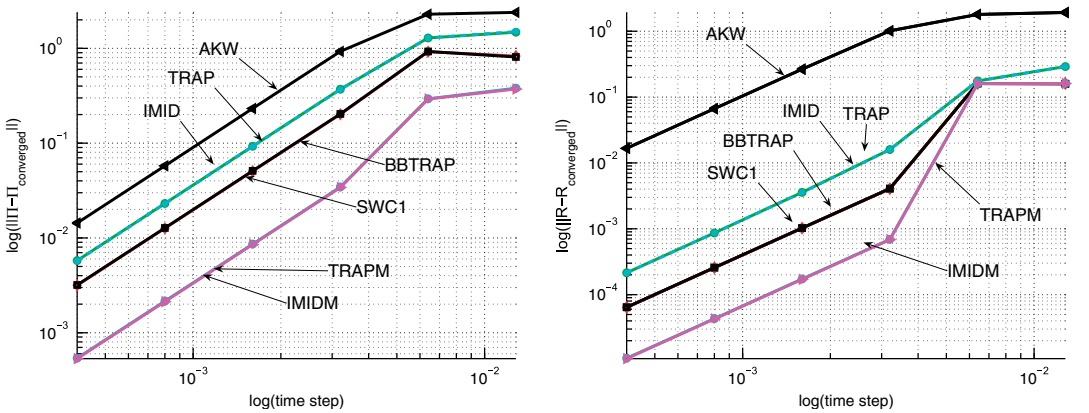

Figure 3. Fast Lagrangian top; on the left-hand side convergence in the norm of the error in body-frame angular momentum; on the right-hand side convergence in the norm of the error in the attitude matrix: **AKW**: implicit midpoint rule of Austin *et al.* [12]; **BBTRAP**: Bottasso and Borri [11]; **SWC1**: implicit Simo and Wong algorithm ALGO_C1 [4]; **TRAP**: trapezoidal rule; **IMID**: midpoint rule; **TRAPM**: momentum-conserving trapezoidal rule; **IMIDM**: momentum-conserving midpoint rule.

and in the error in the attitude matrix is the energy-conserving **IMID**, even though it does not conserve spatial angular momentum. The algorithm **AKW** matches **IMID** in the body-frame momenta, but is in order of magnitude less accurate in the attitude matrix. The conjugate algorithm **TRAP** is slightly less accurate than **IMID** in the body-frame momenta because of the half-step shift. The energy- and momentum-conserving pair **BBTRAP**/**SWC1** is rather inaccurate in the body-frame momenta.

5.2. *Slow Lagrangian top*

In the next example we consider the *slow* symmetrical top in a uniform gravitational field. The body-frame tensor of inertia is diagonal, $\operatorname{diag}\mathbf{I} = [5, 5, 1]$. The spatial torque is

$$\mathbf{t} = -20\mathbf{R}(:, 3) \times [0; 0; 1]$$

where $\mathbf{R}(:, 3)$ is the third column of the attitude matrix, and $[0; 0; 1]$ is the 'up' vector. The initial conditions are

$$\mathbf{R}_0 = \exp[\operatorname{skew}\boldsymbol{\Psi}_0]$$

where $\boldsymbol{\Psi}_0 = [0.05; 0; 0]$, and $\boldsymbol{\Omega}_0 = [0; 0; 5]$.

The good behaviour of the present algorithm is also corroborated by convergence data shown in Figure 2. The global convergence is assessed using a numerical reference solution (obtained with an extremely small time step $\Delta t = 0.0001$) for the attitude matrix and the body-frame angular momenta at time $t = 20$. We measure the norm $\|\mathbf{R} - \mathbf{R}_{\text{converged}}\|_2$ and the norm $\|\boldsymbol{\Pi} - \boldsymbol{\Pi}_{\text{converged}}\|_2$, where the reference values are the orientation matrix $\mathbf{R}_{\text{converged}} = \mathbf{R}(t = 20)$ and the body-frame angular momentum $\boldsymbol{\Pi}_{\text{converged}} = \boldsymbol{\Pi}(t = 20)$. Clearly, the present algorithm converges at a quadratic rate, as all the others algorithms do, and its absolute accuracy is also outstanding.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

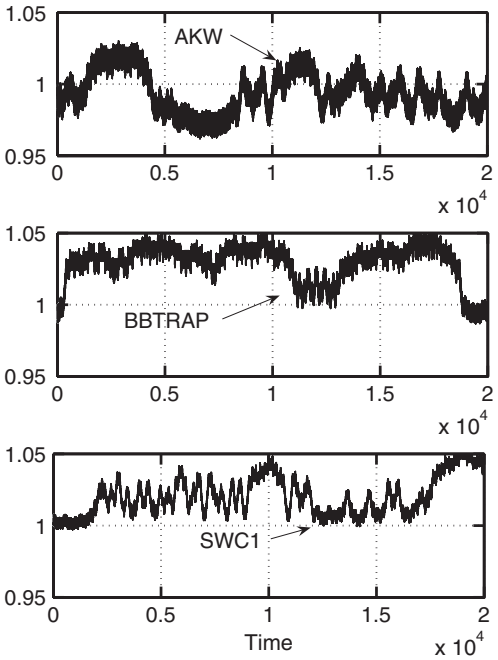

Figure 4. Body in Coulombic potential with soft wall contact, normalized Hamiltonian, step $\Delta t = 0.5$, 40 000 steps; **AKW**: implicit midpoint rule of Austin *et al.* [12]; **BBTRAP**: Bottasso and Borri [11]; **SWC1**: implicit Simo and Wong algorithm ALGO_C1 [4].

5.3. Fast Lagrangian top

In the next example we study the *fast* Lagrangian top. The data for this example appear to be due to Simo and Wong [4]. It had also been investigated by Hulbert [14]. The data are as for the slow top above, except for the initial conditions which are $\mathbf{\Psi}_0 = [0.3; 0; 0]$ and $\mathbf{\Omega}_0 = [0; 0; 50]$.

The kinetic energy for the fast top is dominant, and a numerical method has to effectively deal with precession and nutation which are motions of distinct frequencies. We show convergence graphs for the fast spinning heavy top in Figure 3. The present **IMIDM**/**TRAPM** pair are significantly more accurate than the other algorithms. Perhaps the conservation of the spatial momentum combined with the time-symmetry properties are a decisive advantage here.

5.4. Body in Coulombic potential with soft wall contact

This problem has been investigated by Holder *et al.* [15]. It is a pinned rigid body that rotates under an external torque coming from an attractive Coulombic potential coupled with a repulsive potential with steep gradient that represents a soft wall from which the rotating body is repeatedly repelled. As the authors point out, the repelling torque is troublesome from the point of view of time resolution: the sharp knock imparted to the body when it hits the wall needs to be accurately represented, which calls (typically) for a relatively short time step. Note that none of the algorithms discussed in this paper is able to conserve the Hamiltonian of this system exactly.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

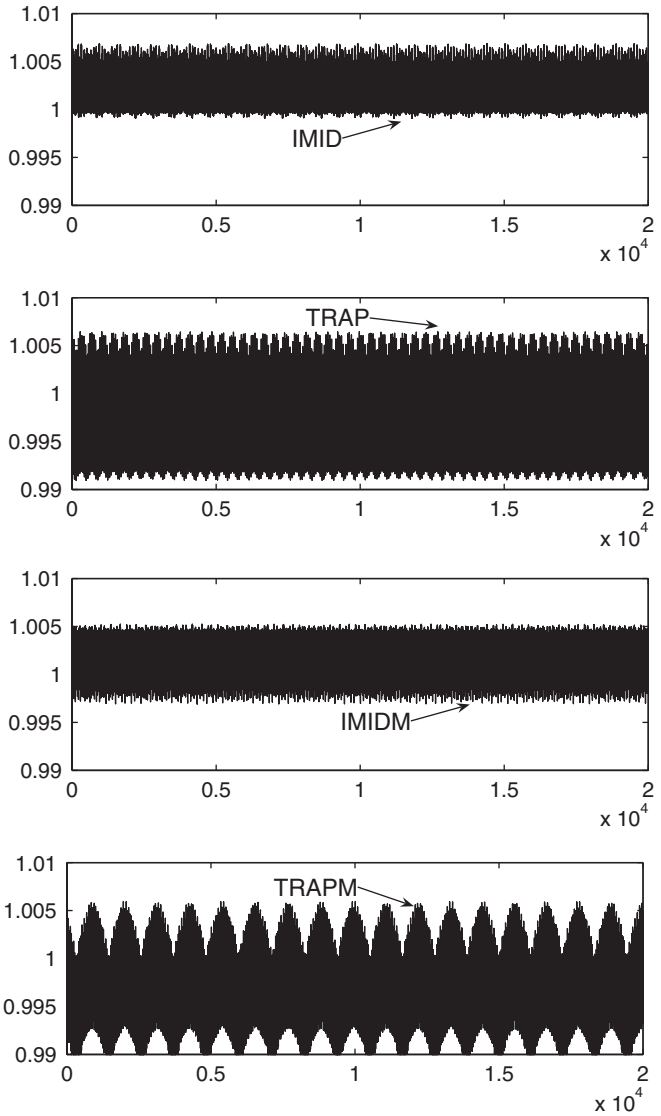

Figure 5. Body in Coulombic potential with soft wall contact, normalized Hamiltonian, step $\Delta t = 0.5$, 40 000 steps; present algorithms **TRAP**: trapezoidal rule; **IMID**: midpoint rule; **TRAPM**: momentum-conserving trapezoidal rule; **IMIDM**: momentum-conserving midpoint rule.

The body-frame tensor of inertia is diagonal, $\mathrm{diag}\,\mathbf{I} = [2, 3, 4.5]$. The components of the spatial torque are

$$[\mathbf{t}] = (-(1.1 + R_{3,3})^{-2} + 0.01(1.1 + R_{3,3})^{-11})[-R_{2,3};\, R_{1,3};\, 0]$$

where R_{ij} are the components of the attitude matrix. The initial conditions are $\mathbf{R}_0 = \mathbf{1}$ and $\boldsymbol{\pi}_0 = [2; 2; 2]$.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

Figures 4 and 5 investigate the long-term representation of the computed Hamiltonian using a relatively long time step, $\Delta t = 0.5$. This leads to rotations within a time step of well over 30°. Consequently, the sharp repelling action of the potential near the wall is likely to be represented rather poorly. Figure 4 illustrates the three reference algorithms, **AKW**, **BBTRAP**, and **SWC1**. Clearly, even though none of the solutions blow up, the long-term behaviour displays an irregular pattern. Figure 5 illustrates the present algorithms; note that the vertical scale in this figure is $\frac{1}{5}$ that of Figure 4, that is the error in the Hamiltonian is generally much smaller. The present algorithms yield very clean and well behaved solutions. The trapezoidal rules display more pronounced periodic pattern of lower frequency and somewhat higher error than the midpoint rules. In the overall quality of the representation of the Hamiltonian, all of the present algorithms are superior to the reference algorithms.

6. CONCLUSIONS

Two midpoint-trapezoid pairs of dynamically equivalent (conjugate) algorithms were derived for the dynamics of rigid body rotation. Both pairs are constructed as compositions of first-order forward Euler and backward Euler integrators as applied to an incremental form of the initial-value problem.

The first pair, **IMID/TRAP**, is energy-conserving for torque-free motions (the latter at the midpoint). The spatial momenta are not conserved. Both are time symmetric.

The second pair, **IMIDM/TRAPM** proceeds from an integral form of the equation of motion (as opposed to the differential form for the first pair), which endows both algorithms with the conservation of the spatial momenta (for torque-free motion). Again, both algorithms are self-adjoint.

The performance of both pairs is excellent, in terms of absolute accuracy and behaviour for larger time steps. An especially attractive feature is the stable and bounded response for large steps and very long time integrations, as demonstrated on the example of a rigid body rotating in an attractive Coulombic potential coupled with a repulsive potential with steep gradient that represents a soft wall.

It is of interest to note that composition of first-order algorithms (explicit midpoint Lie and its adjoint, which in a way are the analogues of the symplectic Euler and its adjoint) has also been used previously in References [8, 9] to derive a high-performance explicit (Newmark) algorithm for the rigid body rotation. The present work continues this line of inquiry, effectively yielding as one of the variants the implicit version of the Newmark algorithm.

ACKNOWLEDGEMENTS

Support for this research by a Hughes-Christensen award is gratefully acknowledged.

REFERENCES

1. Hairer E, Nørsett SP, Wanner G. *Solving Ordinary Differential Equations I. Nonstiff Problems* (revised 2nd edn). Springer Series in Computational Mathematics, vol 8. Springer: Berlin, 1993.
2. Munthe-Kaas H. Runge–Kutta methods on Lie groups. *British Information Technology* 1998; **38**(1):92–111.
3. Iserles A, Munthe-Kaas HZ, Nørsett SP, Zanna A. Lie-group methods. *Acta Numerica* 2000; **9**:215–365.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm

4. Simo JC, Wong KK. Unconditionally stable algorithms for the orthogonal group that exactly preserve energy and momentum. *International Journal for Numerical Methods in Engineering* 1991; **31**:19–52.
5. Zanna A, Eng AK, Munthe-Kaas H. Adjoint and selfadjoint Lie-group methods. *British Information Technology* 2001; **41**(2):395–421.
6. Hairer E, Lubich C, Wanner G. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Computational Mathematics, vol. 31. Springer: Berlin, 2002.
7. Celledoni E, Owren B. Lie methods for rigid body dynamics and time integration on manifolds. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:421–438.
8. Krysl P. Explicit momentum-conserving integrator for dynamics of rigid bodies approximating the midpoint Lie algorithm. *International Journal for Numerical Methods in Engineering* 2005; **63**(15):2171–2193.
9. Krysl P. Direct time integration of rigid body motion with discrete-impulse midpoint approximation: explicit Newmark algorithms. *Communications in Numerical Methods in Engineering* 2006; **22**(5):441–451.
10. Simo JC, Tarnow NN, Wong KK. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Computer Methods in Applied Mechanics and Engineering* 1992; **100**:63–116.
11. Bottasso CL, Borri M. Integrating finite rotations. *Computer Methods in Applied Mechanics and Engineering* 1998; **164**:307–331.
12. Austin M, Krishnaprasad PS, Wang LS. Almost Lie–Poisson integrators for the rigid body. *Journal of Computational Physics* 1993; **107**:105–117.
13. McLachlan RI, Zanna A. The discrete Moser–Veselov algorithm for the free rigid body, revisited. *Reports in Informatics 255*, University of Bergen, 2003.
14. Hulbert G. Explicit momentum conserving algorithms for rigid body dynamics. *Computers and Structures* 1992; **44**(6):1291–1303.
15. Holder T, Leimkuhler B, Reich S. Explicit variable step size and time reversible integration. *Applied Numerical Mathematics* 2001; **39**:367–377.

Copyright © 2006 John Wiley & Sons, Ltd.

DOI: 10.1002/cnm