

Explicit Momentum-conserving Integrator for Dynamics of Rigid Bodies Approximating the Midpoint Lie Algorithm

P. Krysl*

May 4, 2004

1 Introduction

The focus of this work is the initial value problem of the rotational rigid body dynamics. Often the calculation of forcing (evaluation of the external torques) is computationally intensive. For instance, in many molecular dynamics simulations, the evaluation of the forcing may take as much as 90% or more of the CPU time. Consequently, methods that limit the number of evaluations of the external torque are of considerable interest, and we are thus led to contemplate algorithms that are *explicit* in the torque calculation, i.e. the torque is evaluated just *once per time step*.

The main result of this work follows from a reformulation of the midpoint Lie algorithm, which is implicit in the torque calculation as the impulse needs to be evaluated at the unknown midpoint of the incremental rotation. In order to make the algorithm explicit in the torque calculation, we approximate the impulse delivered over the time step with discrete impulses delivered at either the beginning of the time step or at the end of the time step. Therefore, we obtain two related variants, both of which are explicit in the torque calculation, but only first-order in the time step. Both of these algorithms are momentum-conserving and both are symplectic. Therefore, drawing on the properties of the composition of maps, we introduce another algorithm as a composition of these two variants. The resulting algorithm is then explicit, momentum-conserving, symplectic, and second order. Its accuracy is outstanding and consistently matches or exceeds currently known implicit and explicit integrators, as we show on a number of examples.

*University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093-0085, pkrysl@ucsd.edu

2 Midpoint Lie algorithm for free motion

To derive the implicit midpoint Lie integrator algorithm we could follow the abstract procedure in Iserles et al. [5] for general ordinary differential equations on Lie groups. (Application of the *implicit* midpoint Lie approximation to mechanics appeared in Reference [10].) However, in this paper the presentation will be strictly couched in mechanical terms as it affords the most insight.

We start with unforced, free rotations in this section, and then we shall proceed to introduce external torques in the next section. Our point of departure is the equation of motion which reads

$$\dot{\mathbf{\Pi}} = -\tilde{\mathbf{\Omega}}\mathbf{\Pi} . \quad (1)$$

where $\mathbf{\Pi}$ is the body-frame angular momentum; $\mathbf{\Omega}$ is the body-frame angular velocity; recall $\mathbf{\Pi} = \mathbf{I}\mathbf{\Omega}$, where \mathbf{I} is the time-independent body-frame tensor of inertia; and $\tilde{\mathbf{\Omega}}$ is the angular velocity as a skew-symmetric matrix; the vector of angular velocity $\mathbf{\Omega}$ is defined as $\tilde{\mathbf{\Omega}} \cdot \mathbf{\Omega} = \mathbf{0}$.

The angular momentum and the angular velocity may be expressed in the spatial frame as

$$\boldsymbol{\pi} = \mathbf{R}\mathbf{\Pi} , \quad \boldsymbol{\omega} = \mathbf{R}\mathbf{\Omega}$$

where \mathbf{R} is the rotation (attitude) matrix; an orthogonal transformation, $\mathbf{R}^{-1} = \mathbf{R}^T$.

In this work we use the rotation vector to express rotation matrices. Thus,

$$\mathbf{R}(\boldsymbol{\Psi}) = \sum_{k=0}^{\infty} \frac{\tilde{\boldsymbol{\Psi}}^k}{k!} \quad (2)$$

or, in the form of the Rodrigues formula

$$\mathbf{R}(\boldsymbol{\Psi}) = \exp[\tilde{\boldsymbol{\Psi}}] = \mathbf{1} + \frac{\sin \|\boldsymbol{\Psi}\|}{\|\boldsymbol{\Psi}\|} \tilde{\boldsymbol{\Psi}} + \frac{(1 - \cos \|\boldsymbol{\Psi}\|)}{\|\boldsymbol{\Psi}\|^2} \tilde{\boldsymbol{\Psi}}^2 . \quad (3)$$

In the spatial frame the equation of motion (1) reads

$$\dot{\boldsymbol{\pi}} = \mathbf{0}$$

where $\boldsymbol{\pi}$ the general solution is

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(t_0)$$

or, translating to the body frame angular momentum

$$\mathbf{R}(t)\mathbf{\Pi}(t) = \mathbf{R}(t_0)\mathbf{\Pi}(t_0)$$

This may be put as

$$\mathbf{\Pi}(t) = \mathbf{R}^T(t)\mathbf{R}(t_0)\mathbf{\Pi}(t_0)$$

or, expressing rotation from time t_0 to t with the help of an incremental rotation applied to the body frame, $\exp[\tilde{\boldsymbol{\Psi}}(t)]$

$$\mathbf{R}(t) = \mathbf{R}(t_0) \exp[\tilde{\boldsymbol{\Psi}}(t)]$$

we obtain

$$\mathbf{\Pi}(t) = \exp[-\tilde{\Psi}(t)]\mathbf{\Pi}(t_0) \quad (4)$$

Taking the time derivative, we arrive at (the dependence on the argument t is suppressed)

$$\dot{\mathbf{\Pi}} = \frac{d}{dt}(\exp[-\tilde{\Psi}]) \mathbf{\Pi}(t_0) = \text{dexp}_{-\tilde{\Psi}}(-\dot{\tilde{\Psi}}) \exp[-\tilde{\Psi}]\mathbf{\Pi}(t_0)$$

or, using equation (4)

$$\dot{\mathbf{\Pi}} = \text{dexp}_{-\tilde{\Psi}}(-\dot{\tilde{\Psi}})\mathbf{\Pi} \quad (5)$$

Note that we have introduced the differential map [5]

$$\text{dexp}_{\tilde{\Psi}} = \mathbf{1} + \frac{1 - \cos \|\Psi\|}{\|\Psi\|^2} \tilde{\Psi} + \left(1 - \frac{\sin \|\Psi\|}{\|\Psi\|}\right) \frac{\tilde{\Psi}^2}{\|\Psi\|^2}. \quad (6)$$

Now we set equal the right hand sides of equations (5) and (1) to arrive at a link between the angular velocity and the vector of incremental rotation

$$\text{dexp}_{-\tilde{\Psi}}(-\dot{\tilde{\Psi}})\mathbf{\Pi} = -\tilde{\Omega}\mathbf{\Pi}$$

or,

$$\text{dexp}_{-\tilde{\Psi}}(-\dot{\tilde{\Psi}}) = -\tilde{\Omega} \quad (7)$$

or, using vector quantities

$$\text{dexp}_{-\tilde{\Psi}}(-\dot{\tilde{\Psi}}) = -\Omega \quad (8)$$

Introducing the inverse of the differential map (6)

$$\text{dexpinv}_{\tilde{\Psi}} = \mathbf{1} - \frac{1}{2}\tilde{\Psi} - \left(\frac{\|\Psi\|}{2} \cot(\|\Psi\|/2) - 1\right) \frac{\tilde{\Psi}^2}{\|\Psi\|^2}, \quad (9)$$

we may write

$$\dot{\tilde{\Psi}} = \text{dexpinv}_{-\tilde{\Psi}} \Omega \quad (10)$$

2.1 Midpoint approximation

The midpoint approximation to equation (10) reads

$$\dot{\tilde{\Psi}}_{n-\frac{1}{2}} = \left(\text{dexpinv}_{-\tilde{\Psi}} \Omega\right)_{n-\frac{1}{2}} \approx \frac{1}{\Delta t}(\Psi_n - \Psi_{n-1})$$

From consistency considerations $\Psi_{n-1} = \mathbf{0}$, and thus we get

$$\left(\text{dexpinv}_{-\tilde{\Psi}} \Omega\right)_{n-\frac{1}{2}} \approx \frac{1}{\Delta t} \Psi_n$$

where $\left(\text{dexpinv}_{-\tilde{\Psi}} \Omega\right)_{n-\frac{1}{2}} = \text{dexpinv}_{-\frac{1}{2}\tilde{\Psi}_n} \Omega(\frac{1}{2}\Psi_n)$, and therefore to advance the solution, we need to solve

$$\text{dexpinv}_{-\frac{1}{2}\tilde{\Psi}_n} \Omega(\frac{1}{2}\Psi_n) = \frac{1}{\Delta t} \Psi_n \quad (11)$$

However, since we may write equation (11) as

$$\Omega(\frac{1}{2}\Psi_n) = \frac{1}{\Delta t} \text{dexp}_{-\frac{1}{2}\tilde{\Psi}_n} \Psi_n$$

we may use $\tilde{\Psi}_n \cdot \Psi_n = \mathbf{0}$, to show $\text{dexp}_{-\frac{1}{2}\tilde{\Psi}_n} \Psi_n = \Psi_n$, and thus we can conclude that equation (11) maybe simplified to

$$\Omega(\frac{1}{2}\Psi_n) = \frac{1}{\Delta t} \Psi_n \quad (12)$$

The angular velocity at the midpoint may be expressed using the equation of motion (4) as

$$\Omega(\frac{1}{2}\Psi_n) = \mathbf{I}^{-1} \exp[-\frac{1}{2}\tilde{\Psi}_n] \mathbf{\Pi}(t_{n-1}) \quad (13)$$

Substituting the above into equation (12), we arrive at the algorithm **LIEMID** for the time stepping of the torque-free motion.

Algorithm **LIEMID**:

Given Ω_0, \mathbf{R}_0 ,

for $n = 1, 2, \dots$

Solve $\Psi_n = \Delta t \mathbf{I}^{-1} \exp[-\frac{1}{2}\tilde{\Psi}_n] \mathbf{I} \Omega_{n-1}$

$\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\tilde{\Psi}_n]$

$\Omega_n = \mathbf{I}^{-1} (\exp[-\tilde{\Psi}_n] \mathbf{I} \Omega_{n-1})$

end

2.2 Freely spinning body (McLachlan, Zanna 2003)

This example is discussed in the report [8] in the context of discrete Moser-Veselov integrators for the free rigid body. The initial condition is $\Omega = (0.45549, 0.82623, 0.03476)$, and the diagonal entries of the inertia tensor are $\text{diag} \mathbf{I} = [0.9144, 1.098, 1.66]$.

Figure 1 demonstrates how well-behaved the **LIEMID** algorithm is in terms of kinetic energy. Even though the kinetic energy is not conserved, it oscillates with unchanging amplitude and without drift (promising mark of symplecticness). Note that the time steps are quite large, which can be clearly seen in Figure 2 showing the magnitude of the incremental rotation vector in degrees.

Figure 3 shows the components of the body-frame angular velocity. One can see that the qualitative character is well preserved even for large time steps. This is an attractive feature of the present midpoint Lie algorithm.

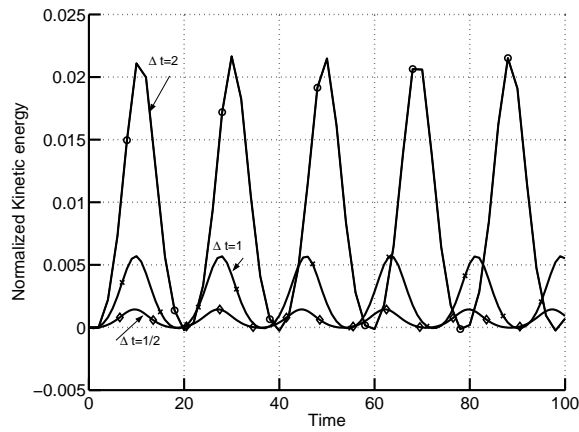


Figure 1: Freely spinning rigid body; kinetic energy for a step sizes 4, 2, and 1.

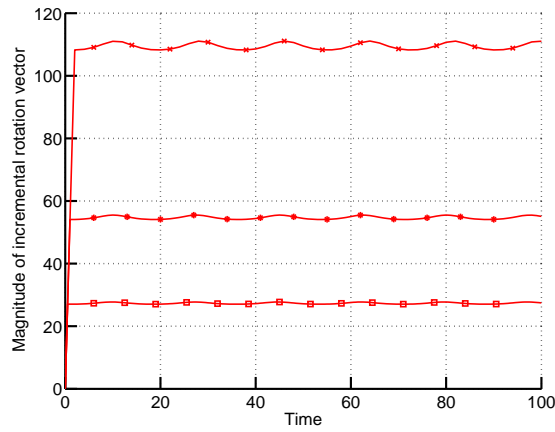


Figure 2: Freely spinning rigid body; magnitude of the incremental rotation vector in degrees for step sizes 4, 2, and 1.

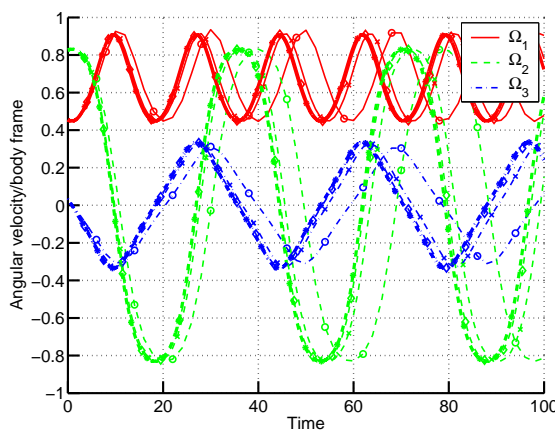


Figure 3: Freely spinning rigid body; components of the body-frame angular velocity for a step sizes 4, 2, 1, and 1/16 (thick line).

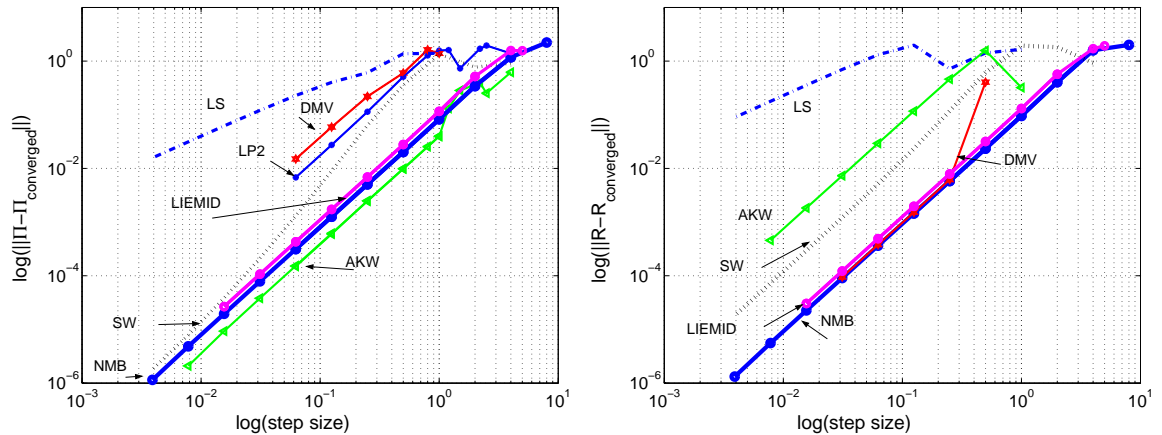


Figure 4: Freely spinning rigid body; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix. **LIEMID**: the present midpoint Lie algorithm; (\triangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (\circ , dashed line) solutions with the Simo and Wong [9] algorithm; **NMB** (\circ , solid line) Krysl, Endres Newmark algorithm [6]; (\star): **DMV** (discrete Moser-Veselov) [8]; **LS** Lewis and Simo symplectic conserving algorithm [7]. Step size 2^{-8} , 2^{-7} , ..., 2^3 .

Figure 4 illustrates the convergence behavior in the norm $\|\mathbf{R} - \mathbf{R}_{converged}\|_2$ and the norm $\|\mathbf{\Pi} - \mathbf{\Pi}_{converged}\|_2$, where the orientation matrix $\mathbf{R}_{converged} = \mathbf{R}(t = 100)$ and the angular momentum in the body frame $\mathbf{\Pi}_{converged} = \mathbf{\Pi}(t = 100)$ have been obtained with an extremely small step size of 0.001. On this problem the **LIEMID** algorithm is on on par with the currently best explicit second-order algorithm, the Krysl, Endres Newmark algorithm [6]. In fact, the performance of the best second-order implicit algorithm, the canonical midpoint rule of Austin, Krishnaprasad, Wang [1] is marginally more accurate for the body frame momentum, but lags behind significantly in the accuracy of the attitude matrix.

3 Midpoint Lie algorithm for forced motion

In this section we introduce forcing. We start with the implicit version of the midpoint Lie algorithm. The equation of motion in the spatial frame reads

$$\dot{\boldsymbol{\pi}} = \mathbf{t} \quad (14)$$

where

$$\mathbf{t} = \text{space-frame applied external torque.}$$

The general solution is

$$\boldsymbol{\pi}(t) = \boldsymbol{\pi}(t_0) + \int_{t_0}^t \mathbf{t}(\tau) d\tau$$

or, translating to the body frame

$$\mathbf{R}(t)\mathbf{\Pi}(t) = \mathbf{R}(t_0)\mathbf{\Pi}(t_0) + \int_{t_0}^t \mathbf{R}(\tau)\mathbf{T}(\tau)d\tau$$

where $\mathbf{T} = \mathbf{R}^T \mathbf{t}$ is the body frame applied torque. Analogously to equation (4), we may write

$$\mathbf{\Pi}(t) = \exp[-\tilde{\Psi}(t)] \left(\mathbf{\Pi}(t_0) + \mathbf{R}^T(t_0) \int_{t_0}^t \mathbf{R}(\tau)\mathbf{T}(\tau)d\tau \right) \quad (15)$$

Differentiating with respect to t we obtain

$$\dot{\mathbf{\Pi}}(t) = \text{dexp}_{-\tilde{\Psi}}(-\dot{\tilde{\Psi}}) \exp[-\tilde{\Psi}(t)] \left(\mathbf{\Pi}(t_0) + \mathbf{R}^T(t_0) \int_{t_0}^t \mathbf{R}(\tau)\mathbf{T}(\tau)d\tau \right) + \mathbf{T}(t) . \quad (16)$$

or, substituting from (15)

$$\dot{\mathbf{\Pi}}(t) = \text{dexp}_{-\tilde{\Psi}}(-\dot{\tilde{\Psi}})\mathbf{\Pi}(t) + \mathbf{T}(t) . \quad (17)$$

With the equation of motion in the body frame, $\dot{\mathbf{\Pi}} = -\tilde{\Omega}\mathbf{\Pi} + \mathbf{T}$, at hand, we see that we again arrive at equation (7), or equivalently equation (8).

3.1 Implicit Midpoint approximation

Applying results of Section 2.1 we find yet again the condition from which to advance the solution as

$$\Omega(\frac{1}{2}\Psi_n) = \frac{1}{\Delta t} \Psi_n .$$

However, now the midpoint angular velocity $\Omega(\frac{1}{2}\Psi_n)$ needs to be expressed from the forced equation of motion (15), where as before $\mathbf{\Pi} = \mathbf{I}\Omega$. Therefore, we need to approximate the torque impulse, and the midpoint form may be put as

$$\mathbf{R}^T(t_0) \int_{t_0}^{t_0+\Delta t/2} \mathbf{R}(\tau)\mathbf{T}(\tau)d\tau \approx \frac{\Delta t}{2} \mathbf{R}^T(t_0)\mathbf{R}(t_0 + \Delta t/2)\mathbf{T}(t_0 + \Delta t/2) \quad (18)$$

where $\mathbf{T}(t_0 + \Delta t/2) = \mathbf{T}(t_0 + \Delta t/2, \frac{1}{2}\Psi_n)$, and

$$\begin{aligned} \mathbf{R}(t_0 + \Delta t/2) &= \mathbf{R}(t_0) \exp[\frac{1}{2}\tilde{\Psi}_n] \\ \mathbf{R}^T(t_0) \int_{t_0}^{t_0+\Delta t/2} \mathbf{R}(\tau)\mathbf{T}(\tau)d\tau &\approx \frac{\Delta t}{2} \exp[\frac{1}{2}\tilde{\Psi}_n]\mathbf{T}(t_0 + \Delta t/2) \end{aligned} \quad (19)$$

Substituting this approximation into equation (15) expressed at time $t + \Delta t/2$, we obtain

$$\mathbf{I}^{-1} \left(\exp[-\frac{1}{2}\tilde{\Psi}_n]\mathbf{\Pi}(t_{n-1}) + \frac{\Delta t}{2}\mathbf{T}(t_0 + \Delta t/2, \frac{1}{2}\Psi_n) \right) = \frac{1}{\Delta t} \Psi_n .$$

Thus, we arrive at the implicit midpoint Lie algorithm

Algorithm **LIEMID**[I]:

Given $\mathbf{\Omega}_0, \mathbf{R}_0,$

FOR $n = 1, 2, \dots$

Solve $\mathbf{\Psi}_n = \Delta t \mathbf{I}^{-1} \left(\exp[-\frac{1}{2} \tilde{\mathbf{\Psi}}_n] \mathbf{I} \mathbf{\Omega}_{n-1} + \frac{\Delta t}{2} \mathbf{T}_{n-1/2} \right)$

$\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\tilde{\mathbf{\Psi}}_n]$

$\mathbf{\Omega}_n = \mathbf{I}^{-1} \left(\exp[-\tilde{\mathbf{\Psi}}_n] \mathbf{I} \mathbf{\Omega}_{n-1} + \Delta t \exp[-\frac{1}{2} \tilde{\mathbf{\Psi}}_n] \mathbf{T}_{n-1/2} \right)$

end

3.2 Explicit Midpoint approximation

As for the implicit version, we need to approximate the torque impulse. It seems natural to consider a forward Euler approximation

$$\mathbf{R}^T(t_0) \int_{t_0}^{t_0+\Delta t/2} \mathbf{R}(\tau) \mathbf{T}(\tau) d\tau \approx \frac{\Delta t}{2} \mathbf{R}^T(t_0) \mathbf{R}(t_0) \mathbf{T}(t_0) = \frac{\Delta t}{2} \mathbf{T}(t_0) \quad (20)$$

Thus, we get the algorithm **LIEMID**[E].

Algorithm **LIEMID**[E]:

Given $\mathbf{\Omega}_0, \mathbf{R}_0,$

for $n = 1, 2, \dots$

Solve $\mathbf{\Psi}_n = \Delta t \mathbf{I}^{-1} \exp[-\frac{1}{2} \tilde{\mathbf{\Psi}}_n] (\mathbf{I} \mathbf{\Omega}_{n-1} + \frac{\Delta t}{2} \mathbf{T}_{n-1})$

$\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\tilde{\mathbf{\Psi}}_n]$

$\mathbf{\Omega}_n = \mathbf{I}^{-1} \exp[-\tilde{\mathbf{\Psi}}_n] (\mathbf{I} \mathbf{\Omega}_{n-1} + \Delta t \mathbf{T}_{n-1})$

end

The visual differences between the implicit version and **LIEMID**[E] are innocuous, but of tremendous importance. The algorithm **LIEMID**[E] is in fact no good: very poor accuracy, linear convergence rate compared to the quadratic rate of the implicit version, and apparently only conditional stability (compare with figures in the following sections, for instance Figure 6). How can we improve on this?

We suggest that the solution be considered in terms of discrete impulses as a way of approximating the integral in equation (20). Thus, let us assume that the effect of external torques is delivered by discrete impulses, whose sum over a given interval approximates the integral of the external torques over the same interval, but which are concentrated at times t_i . In other words, we assume that the motion is free in between the time instants t_{i-1} and t_i , and that there are torque impulses at the boundaries of the time steps that deliver instantaneous kicks to the body. This is easily accommodated in the algorithm for the free rigid body **LIEMID**. We assume the body moves without external torques in the interval $t_{n-1} \leq t < t_n$, and therefore we may solve for the incremental rotation vector from (13). Then, at time t_n the torque impulse $\Delta t \mathbf{T}_n$ is delivered, which is reflected in the update equation for the angular velocity. This narrative is summarized in the algorithm **LIEMID**[E1], which

should be compared with the algorithm **LIEMID** for the free body. Clearly, the only difference is the added torque impulse.

Algorithm **LIEMID[E1]**:

```

Given  $\mathbf{\Omega}_0, \mathbf{R}_0,$ 
for  $n = 1, 2, \dots$ 
    Solve  $\mathbf{\Psi}_n = \Delta t \mathbf{I}^{-1} \exp[-\frac{1}{2} \tilde{\mathbf{\Psi}}_n] (\mathbf{I} \mathbf{\Omega}_{n-1})$ 
     $\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\tilde{\mathbf{\Psi}}_n]$ 
     $\mathbf{\Omega}_n = \mathbf{I}^{-1} \left( \exp[-\tilde{\mathbf{\Psi}}_n] \mathbf{I} \mathbf{\Omega}_{n-1} + \Delta t \mathbf{T}_n \right)$ 
end

```

While we have assumed that the kick from the external torque had been delivered at the end of the time step, evidently it would equally make sense to give the shove at the beginning of the time step. Therefore, we may assume the body moves without external torques in the interval $t_{n-1} < t \leq t_n$, but at time t_{n-1} the torque impulse $\Delta t \mathbf{T}_{n-1}$ changes the initial angular velocity for the step. This is put into formulas in the algorithm **LIEMID[E2]**.

Algorithm **LIEMID[E2]**:

```

Given  $\mathbf{\Omega}_0, \mathbf{R}_0,$ 
for  $n = 1, 2, \dots$ 
    Solve  $\mathbf{\Psi}_n = \Delta t \mathbf{I}^{-1} \exp[-\frac{1}{2} \tilde{\mathbf{\Psi}}_n] (\mathbf{I} \mathbf{\Omega}_{n-1} + \Delta t \mathbf{T}_{n-1})$ 
     $\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\tilde{\mathbf{\Psi}}_n]$ 
     $\mathbf{\Omega}_n = \mathbf{I}^{-1} \exp[-\tilde{\mathbf{\Psi}}_n] (\mathbf{I} \mathbf{\Omega}_{n-1} + \Delta t \mathbf{T}_{n-1})$ 
end

```

3.3 Slow Lagrangian top

It is time to look at an example to help us assess the qualities of the above algorithms. In the first forced-motion example we consider the *slow* symmetrical top with total mass M in a uniform gravitational field. The body frame tensor of inertia is diagonal, $\text{diag} \mathbf{I} = [5, 5, 1]$. 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

$$\mathbf{R}_0 = \exp[\tilde{\mathbf{\Psi}}_0]$$

where $\mathbf{\Psi}_0 = [0.05; 0; 0]$, and

$$\mathbf{\Omega}_0 = [0; 0; 5]$$

Figure 5 shows the spatial angular momenta for the slow top obtained with $\Delta t = 1/16$. Notice that the component of angular momentum in the direction of gravity is

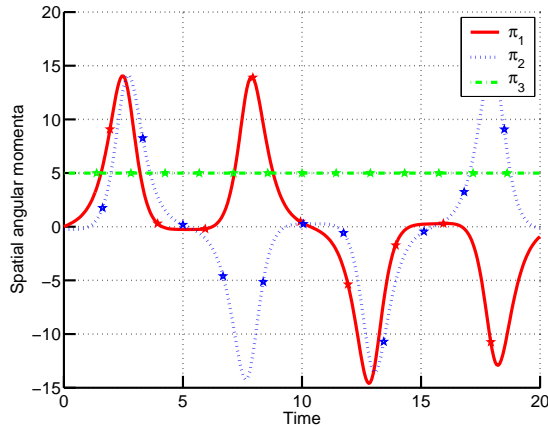


Figure 5: Slow Lagrangian top: Components of the spatial angular momentum. Implicit midpoint Lie **LIEMID**[**I**].)

conserved (simulation with the implicit midpoint Lie method). Figure 6 illustrates the global convergence by using a numerical 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 $\|\mathbf{R} - \mathbf{R}_{converged}\|_2$, and the norm $\|\mathbf{\Pi} - \mathbf{\Pi}_{converged}\|_2$, where the reference values are the orientation matrix $\mathbf{R}_{converged} = \mathbf{R}(t = 20)$ and the body frame angular momentum $\mathbf{\Pi}_{converged} = \mathbf{\Pi}(t = 20)$. The explicit midpoint Lie algorithm does not converge at all. The implicit algorithm converges at the expected quadratic rate, and the relative accuracy is also excellent. The explicit midpoint Lie variants have excellent accuracy for larger time steps, but then seem to be experiencing trouble for smaller time steps, especially in the representation of the attitude matrix. This is worrying, since that would indicate that these two variants have asymptotically linear convergence rate. Since the story told in Figure 6 is being corroborated by other numerical and theoretical evidence, some of it reported below, we have to ask ourselves if we could do better.

Figure 7 shows the Hamiltonian for the slow top calculated for $\Delta t = 1/8$ with the implicit algorithm **LIEMID**[**I**] and also with all the above explicit algorithms. Couple of intriguing observations may be made in that figure. Firstly, all the algorithms display periodic behavior, and the implicit algorithm and the two variants **LIEMID**[**E1**] and **LIEMID**[**E2**] are apparently symplectic. Secondly, the two explicit variants appear to possess some sort of symmetry, since the errors of the calculated Hamiltonians cancel, not quite perfectly but well enough to suggest taking advantage of this. Some symmetry may be also discerned in the algorithms themselves.

3.4 Improved explicit midpoint Lie algorithm

Indeed the idea presented at this point is very simple: we have two algorithms whose errors seem to be canceling, could we then combine them in some suitable way to take advantage of this cancellation? There may be more than one answer to this question,

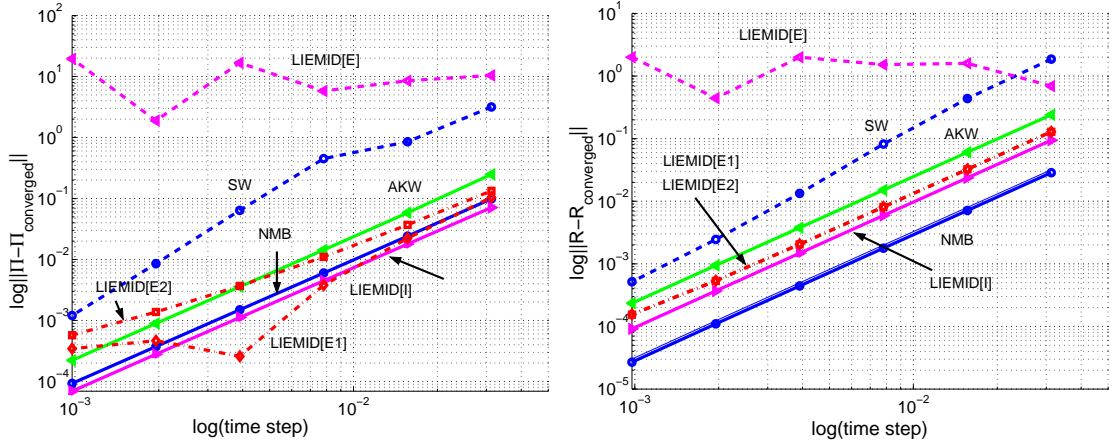


Figure 6: Slow Lagrangian top; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix: (\triangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (\circ , dashed line) solutions with the Simo and Wong [9] algorithm; **NMB** (\circ , solid line) Krysl, Endres Newmark algorithm [6]; (\triangleright): implicit midpoint Lie; (\triangleleft , dashed line): explicit midpoint Lie; (\diamond): explicit midpoint Lie variant 1; (\square): explicit midpoint Lie variant 2.

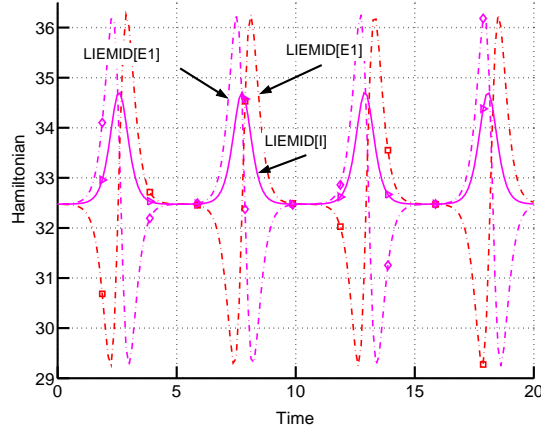


Figure 7: Slow Lagrangian top: Hamiltonian. (\triangleright): implicit midpoint Lie **LIEMID[I]**; (\diamond): explicit midpoint Lie variant 1 **LIEMID[E1]**; (\square): explicit midpoint Lie variant 2 **LIEMID[E2]**. (Hamiltonian for explicit midpoint Lie **LIEMID[E]** was growing linearly.)

and the proposed algorithm **LIEMID[EA]** presents a particularly simple one: We alternate the two variants **LIEMID[E1]** and **LIEMID[E2]**: push forward the solution with variant 1, then continue with variant 2 using the output of variant 1 as the initial conditions, then variant 1 again and so on. One might think that the alternation may be performed at the level of time steps, that is odd time steps performed with variant 1, and even time steps performed with a variant 2. However, upon closer inspection

we see that the torque would be calculated only once per every *two* time steps. It is possible then to deduce that we can produce an algorithm with exactly one torque evaluation per time step if the two algorithms advance the solution by half a time step each. The resulting algorithm **LIEMID[EA]** is summarized below. Note that we update the rotation matrix twice only for presentation purposes; for efficiency and robustness it is actually updated only once as $\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\tilde{\Psi}_{n-1/2}] \exp[\tilde{\Psi}_n]$.

Algorithm **LIEMID[EA]**:

Given Ω_0, \mathbf{R}_0 ,

for $n = 1, 2, \dots$

% half-step with algorithm 2

Solve $\Psi_{n-1/2} = \frac{\Delta t}{2} \mathbf{I}^{-1} \exp[-\frac{1}{2} \tilde{\Psi}_{n-1/2}] (\mathbf{I} \Omega_{n-1} + \frac{\Delta t}{2} \mathbf{T}_{n-1})$

$\mathbf{R}_{n-1/2} = \mathbf{R}_{n-1} \exp[\tilde{\Psi}_{n-1/2}]$

$\Omega_{n-1/2} = \mathbf{I}^{-1} \exp[-\tilde{\Psi}_{n-1/2}] (\mathbf{I} \Omega_{n-1} + \frac{\Delta t}{2} \mathbf{T}_{n-1})$

% half-step with algorithm 1

Solve $\Psi_n = \frac{\Delta t}{2} \mathbf{I}^{-1} \exp[-\frac{1}{2} \tilde{\Psi}_n] (\mathbf{I} \Omega_{n-1/2})$

$\mathbf{R}_n = \mathbf{R}_{n-1/2} \exp[\tilde{\Psi}_n]$

$\Omega_n = \mathbf{I}^{-1} \left(\exp[-\tilde{\Psi}_n] \mathbf{I} \Omega_{n-1/2} + \frac{\Delta t}{2} \mathbf{T}_n \right)$

end

The above mechanically inspired formulation may be couched in mathematical terms as a composition of maps. It is straightforward to show that **LIEMID[E1]** and **LIEMID[E2]** are mutually adjoint. (Recall the definition of adjoint methods [2]: the adjoint method of Φ_h is Φ_h^* such that $\Phi_h^* = \Phi_{-h}^{-1}$.)

It is well known that composition of adjoint maps is going to be time-symmetric, that is for $\Psi_h = \Phi_h^* \circ \Phi_h$ we will have $\Psi_h^* = \Psi_h$. Therefore, since **LIEMID[EA]** is a composition of mutually adjoint first-order maps, it is time-symmetric and consequently we expect its accuracy to be of second order [2].

If we can prove that **LIEMID[E1]** is symplectic (work in progress, but numerical evidence is very strong), then also its adjoint is going to be symplectic, as well as their composition. In addition, both **LIEMID[E1]** and **LIEMID[E2]** conserve momentum in the absence of external forcing, and therefore **LIEMID[EA]** is momentum conserving too. Consequently, we expect the algorithm **LIEMID[EA]** to have a very desirable set of basic properties: symplecticness and momentum conservation.

We wish to stress that we choose to label both the basic algorithms **LIEMID[E1]** and **LIEMID[E2]** and their composition **LIEMID[EA]** as *explicit midpoint Lie*, since

1. all of these maps are *identical* to the midpoint Lie in the absence of forcing; the only difference between them is the approximation of the torque impulse;
2. these algorithms are explicit in the torque evaluation, as it needs to be computed just once per time step.

The last remark concerns the analogy between the present algorithm and a composition of two first-order methods from the vector-space setting, the symplectic Euler and the adjoint of the symplectic Euler. The composition of these two Euler methods is the well-known Newmark/Verlet algorithm [2]. That algorithm may be also derived from the concept of concentrated impulses, with the basic discretization in the form of a midpoint method (work in progress).

Next we shall study a few more examples to demonstrate the performance of the improved explicit midpoint Lie algorithm, **LIEMID[EA]**.

3.5 Fast Lagrangian top

The data for this example appear to be due to Simo and Wong [9]. It had also been studied by Hulbert [4]. In this example we consider the motion of a symmetrical top with total mass M and axis of symmetry that coincides with the direction of uniform gravitational field. The body frame tensor of inertia is diagonal, $\text{diag}\mathbf{I} = [5, 5, 1]$. 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

$$\mathbf{R}_0 = \exp[\widetilde{\Psi}_0]$$

where $\Psi_0 = [0.3; 0; 0]$, and

$$\Omega_0 = [0; 0; 50]$$

It is well known that the symmetric Lagrangian heavy top model possesses four conserved quantities:

1. the Hamiltonian,

$$H(\boldsymbol{\pi}, \boldsymbol{\gamma}) = \frac{1}{2}\boldsymbol{\pi} \cdot \mathbf{i}^{-1}\boldsymbol{\pi} + Mlg\boldsymbol{\gamma} \cdot \boldsymbol{\chi}$$

where

- $\boldsymbol{\pi}$ = spatial angular momentum, $\boldsymbol{\pi} = \boldsymbol{\pi} \cdot \boldsymbol{\omega}$;
- $\boldsymbol{\chi}$ = unit vector along the axis of the top pointing from the fixed point;
- $\boldsymbol{\gamma}$ = unit vector in the direction of gravity;
- Mlg = product of the mass, distance of the center of mass and the fixed point, and gravitational acceleration, $Mlg = \frac{\sqrt{3}}{2}9.81$

2. the projection of the spatial angular momentum on the axis of the top, $\boldsymbol{\pi} \cdot \boldsymbol{\chi}$,
3. the projection of the spatial angular momentum on the gravity vector, $\boldsymbol{\pi} \cdot \boldsymbol{\gamma}$,
4. the norm of the gravity vector in the body frame, $\|\mathbf{R}\boldsymbol{\gamma}\|$.

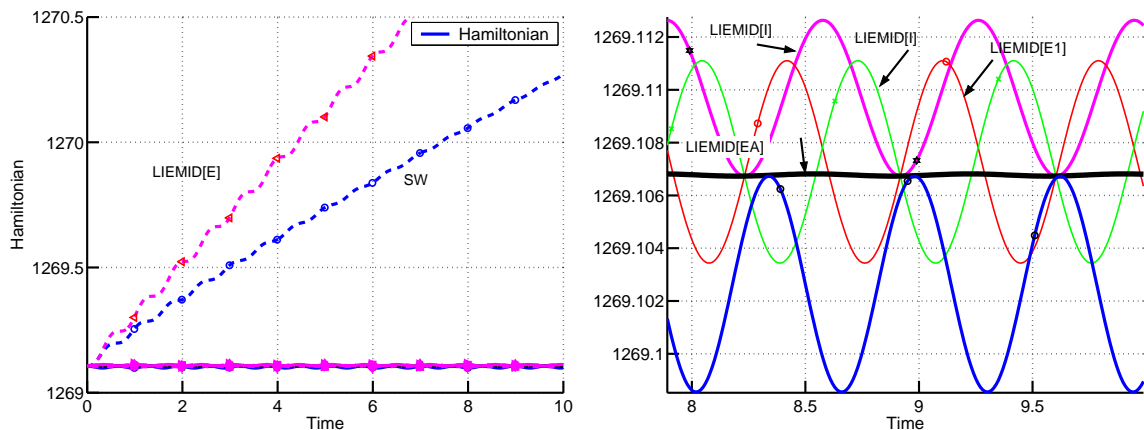


Figure 8: Fast Lagrangian top; Hamiltonian. Left-hand side: overall view, right hand side: zoom in to the neighborhood of the time axis. (\triangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]): **SW** (\circ , dashed line) solutions with the Simo and Wong [9] algorithm; **NMB** (\circ , solid line) Krysl, Endres Newmark algorithm [6] algorithm; (\triangleright): implicit midpoint Lie **LIEMID[EI]**; (\triangleleft , dashed line): explicit midpoint Lie **LIEMID[E]**; (\diamond): explicit midpoint Lie variant 1 **LIEMID[E1]**; (\square): explicit midpoint Lie variant 2 **LIEMID[E2]**; alternating explicit midpoint Lie method **LIEMID[EA]**.

The last two are the Casimirs of the Poisson bracket that defines the Hamiltonian structure.

The Hamiltonian is shown in figure 8. The two methods **LIEMID[E]** and **SW** display a distinct non-conservation of the total energy. On the other hand, as shown in the zoomed in viewport on the right, the Newmark method **NMB**, the two variants **LIEMID[E1]** and **LIEMID[E2]**, the alternating explicit midpoint Lee method **LIEMID[EA]**, and the implicit midpoint Lie method **LIEMID[I]** all show essentially periodic character of the Hamiltonian with no drift. Notice that the error of the implicit midpoint is larger than the errors of the explicit variants, both linear and quadratic.

We investigate the global convergence by using a numerical solution (obtained with an extremely small timestep $\Delta t = 0.000005$) for the attitude matrix and the body frame angular momenta at time $t = 10$. We measure the norm $\|\mathbf{R} - \mathbf{R}_{converged}\|_2$, and the norm $\|\mathbf{\Pi} - \mathbf{\Pi}_{converged}\|_2$, where the reference values are the orientation matrix $\mathbf{R}_{converged} = \mathbf{R}(t = 10)$ and the body frame angular momentum $\mathbf{\Pi}_{converged} = \mathbf{\Pi}(t = 10)$. The convergence curves are shown in Figure 9. Noticed excellent behavior of the present alternating explicit midpoint method **LIEMID[EA]**. It is the best performer, even compared with the implicit midpoint Lee method. Interestingly, the first order variants of the explicit midpoint the method produce errors comparable to those attained by the best quadratic method for smaller time steps. As evidence of symplecticness, Figure 10 shows the scaled Hamiltonian error (by $1/\Delta t^2$) for the alternating explicit midpoint lead method **LIEMID[EA]** for 90 times larger integration

interval. Note that there is no perceivable drift, or change in amplitude.

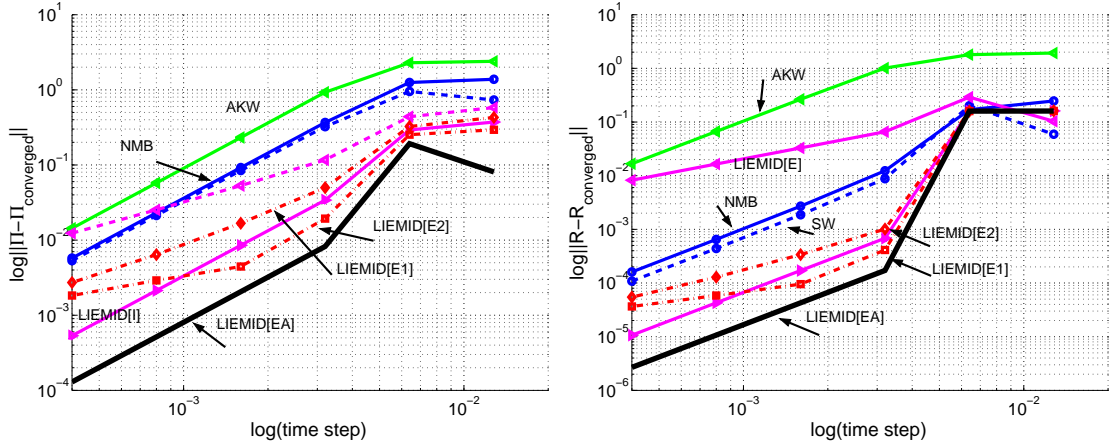


Figure 9: Fast Lagrangian top; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix: (\triangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (\circ , dashed line) solutions with the Simo and Wong [9] algorithm; **NMB** (\circ , solid line) Krysl, Endres Newmark algorithm [6] algorithm; (\triangleright): implicit midpoint Lie **LIEMID[EI]**; (\triangleleft , dashed line): explicit midpoint Lie **LIEMID[E]**; (\diamond): explicit midpoint Lie variant 1 **LIEMID[E1]**; (\square): explicit midpoint Lie variant 2 **LIEMID[E2]**; thick solid line: alternating explicit midpoint Lie method **LIEMID[EA]**.

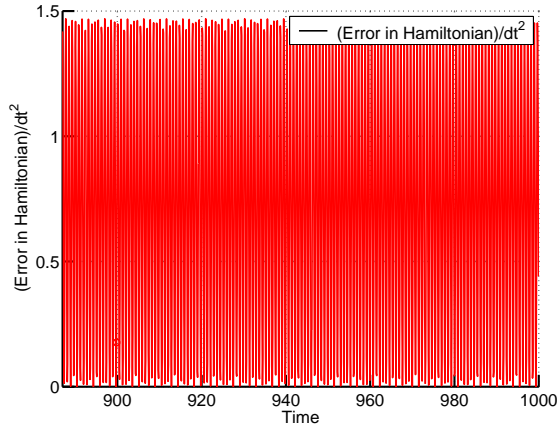


Figure 10: Fast Lagrangian top: Hamiltonian error scaled by $1/\Delta t^2$. Alternating explicit midpoint Lie **LIEMID[EA]**.

3.6 Slow Lagrangian top

In this example we continue with the example of the *slow* symmetrical top from Section 3.3. Figure 11 adds to the data of Figure 6 the curves for at the alternating explicit algorithm, **LIEMID[EA]**, and it is clearly the most accurate algorithm overall.

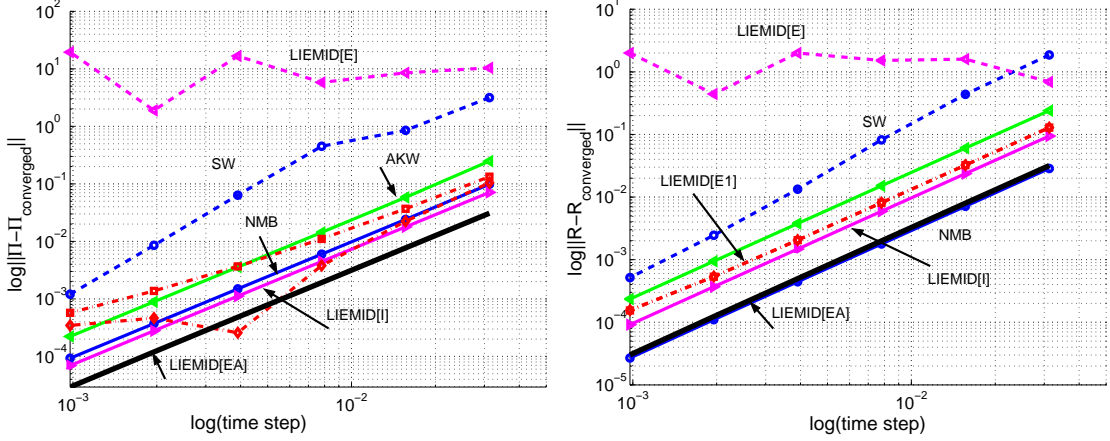


Figure 11: Slow Lagrangian top; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix: (\blacktriangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (\circ , dashed line) solutions with the Simo and Wong [9] algorithm; **NMB** (\circ , solid line) Krysl, Endres Newmark algorithm [6] algorithm; (\blacktriangleright): implicit midpoint Lie; (\blacktriangleleft , dashed line): explicit midpoint Lie **LIEMID[E]**; (\blacklozenge): explicit midpoint Lie variant 1 **LIEMID[E1]**; (\square): explicit midpoint Lie variant 2 **LIEMID[E2]**.

Figure 12 supplements the data in Figure 7 with the algorithm **LIEMID[EA]**, and illustrates the cancellation of the errors of the Hamiltonian for the slow top calculated for $\Delta t = 1/8$. Even though the error of **LIEMID[EA]** is not zero (that would mean the algorithm has become momentum and *energy* conserving), it substantially decreased with respect to its constituent parts. Interestingly, is evident that the error of the present explicit algorithm **LIEMID[EA]** is much lower than the error of the fully implicit midpoint Lie method **LIEMID[I]**.

Figure 13 shows the error of the Hamiltonian exhibited by the alternating explicit midpoint Lie method **LIEMID[EA]** scaled by the step size squared for three different Δt 's. The three curves are similar in appearance, of roughly the same amplitude, and display no perceivable drift in time, which seems to confirm our guess that the method is second order and symplectic.

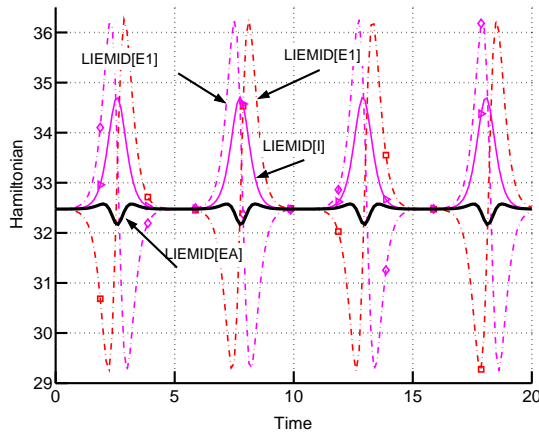


Figure 12: Slow Lagrangian top: Hamiltonian. (\triangleright): implicit midpoint Lie **LIEMID[I]**; (\diamond): explicit midpoint Lie variant 1 **LIEMID[E1]**; (\square): explicit midpoint Lie variant 2 **LIEMID[E2]**; thick black line: alternating explicit midpoint Lie **LIEMID[EA]**. (Hamiltonian for explicit midpoint Lie **LIEMID[E]** was growing linearly.)

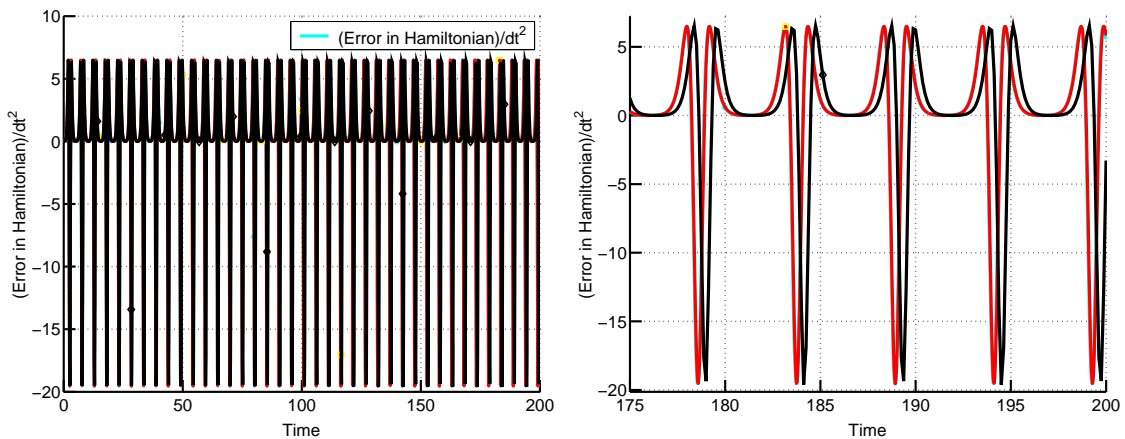


Figure 13: Slow Lagrangian top: Hamiltonian error scaled by $1/\Delta t^2$. Alternating explicit midpoint Lie **LIEMID[EA]**.

3.7 Body in Coulombic potential with soft wall contact

This problem had been used by the Holder, Leimkuhler, and Reich [3] to investigate the need for adaptive variable step size methods. The problem they consider is a 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 resolution.

The body frame tensor of inertia is diagonal, $\text{diag} \mathbf{I} = [2, 3, 4.5]$. The components

of the spatial torque are

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

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

Figure 14 compares the convergence of the various algorithms discussed here with some good representatives of implicit and explicit algorithms from the literature. By some lucky cancellation of errors, the two first order methods, **LIEMID[E1]** and **LIEMID[E2]** outperform all the second order methods in the representation of body-frame angular momentum for larger time steps. Interestingly, they also maintain second order rate in the representation of the attitude matrix. The alternating explicit midpoint Lie method **LIEMID[EA]** performs very well, and maintains second-order accuracy.

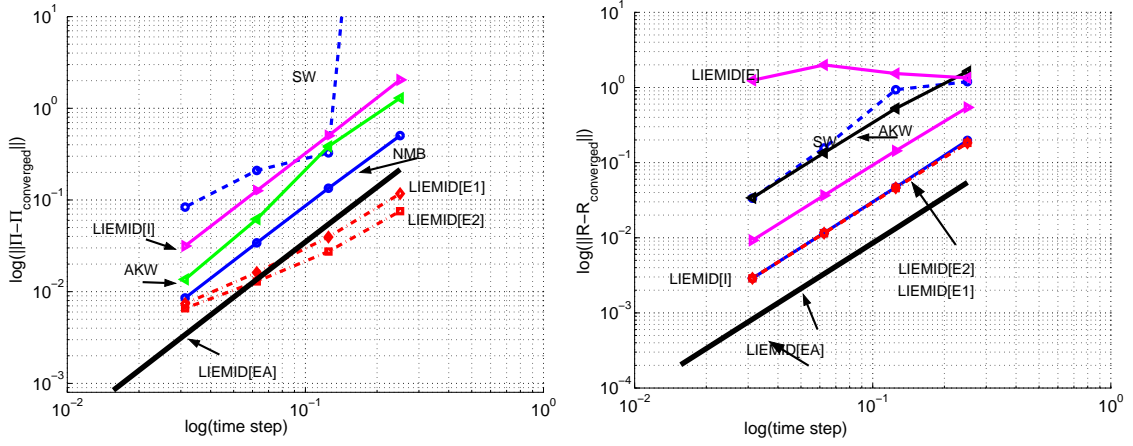


Figure 14: Body in Coulombic potential with soft wall contact; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix: (\blacktriangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (\circ , dashed line) solutions with the Simo and Wong [9] algorithm; **NMB** (\circ , solid line) Krysl, Endres Newmark algorithm [6]; (\blacktriangleright): implicit midpoint Lie; (\blacktriangleleft , dashed line): explicit midpoint Lie; (\blacklozenge): explicit midpoint Lie variant 1 **LIEMID[E1]**; (\blacksquare): explicit midpoint Lie variant 2 **LIEMID[E2]**; thick solid line: alternating explicit midpoint Lie **LIEMID[EA]**.

Figure 15 shows one of the non-conserved components of the spatial angular momentum. The reference solution was obtained with the timestep $\Delta t = 0.01$. The other solutions had been obtained with a large timestep $\Delta t = 0.75$. As can be seen, the implicit midpoint Lie, and both variants of the the explicit midpoint Lie outperform the classical midpoint rule **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]).

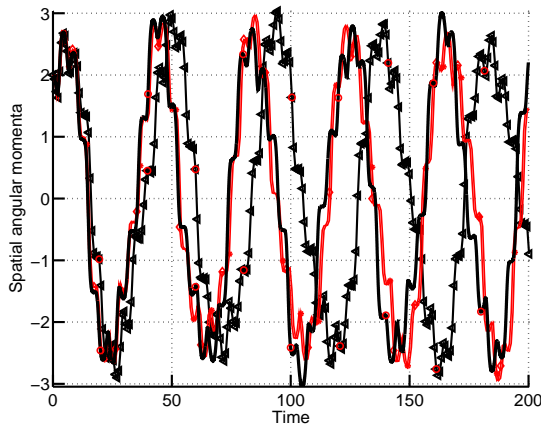


Figure 15: Body in Coulombic potential with soft wall contact; one of the non-conserved components of the spatial angular momentum: solid line without symbol is the reference solution; (\blacktriangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); (\blacktriangleright): implicit midpoint Lie; (\blacklozenge): explicit midpoint Lie variant 1; (\blacksquare): explicit midpoint Lie variant 2.

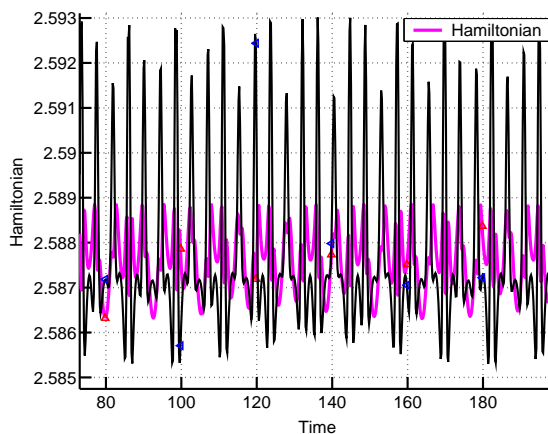


Figure 16: Body in Coulombic potential with soft wall contact: Hamiltonian for $\Delta t = 0.2$. (\blacktriangleright): implicit midpoint Lie **LIEMID[I]**; alternating explicit midpoint Lie **LIEMID[EA]**.

Figure 17 compares the performance of some good performing with implicit and explicit algorithms with respect to the conservation of the projection of the spatial angular momentum onto the symmetry axis (direction of the potential gradient). All the midpoint Lie algorithms presented in this paper conserve this projection to within machine accuracy. On the other hand, some of the best-performing algorithms from the literature do not conserve the projection exactly, but only within a certain bound. In particular, it can be seen that the algorithm of Simo and Wong [9] conserves this projection, while the Newmark algorithm of Krysl, Endres Newmark algorithm [6] and the algorithm **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]) do

not.

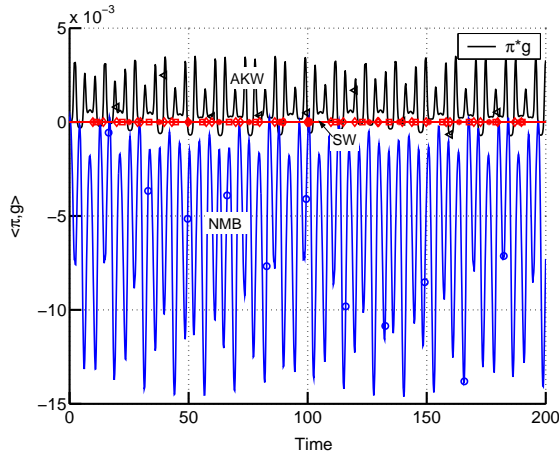


Figure 17: Body in Coulombic potential with soft wall contact; error in the projection of the spatial angular momentum on the symmetry axis: (\blacktriangle): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (\circ) solutions with the Simo and Wong [9] algorithm; **NMB** (\circ) Krysl, Endres Newmark algorithm [6] algorithm.

Conclusions

We have presented an algorithm that is explicit in the evaluation of the forcing (torque), is momentum conserving, and apparently also symplectic. The numerical evidence we have presented indicates that this algorithm is the best explicit second-order integrator for rigid body dynamics the date, as it consistently matches or exceeds the accuracy of the currently known best-performing algorithms, both implicit and explicit.

The result we present follows from a reformulation of the implicit midpoint Lie algorithm. This integrator is implicit in the torque calculation because the impulse needs to be evaluated at the unknown midpoint of the incremental rotation. In order to make the algorithm explicit in the torque calculation, we approximate the impulse delivered over the time step with discrete impulses delivered at either the beginning of the time step or at the end of the time step. Therefore, we obtain two related variants, both of which are explicit in the torque calculation. It turns out that this approximation reduces the order of these algorithms from two to one, but they are momentum-conserving and apparently both are symplectic. To improve the convergence rate, we take advantage of the apparent possibility to cancel discretization errors of these two variants, since they tend to go in opposite directions. Thus, we introduce another algorithm as a composition of these two variants. As is well-known, such a composition will also be symplectic and the momentum conservation is preserved too. The resulting algorithm is then explicit, momentum-conserving,

(apparently) symplectic, and second order. Its error constant is excellent, and its accuracy consistently matches or exceeds (sometimes very substantially) accuracy of the best known implicit and explicit integrators on a number of problems, including the free rigid body, both fast and slow Lagrangian tops, and dynamics in the presence of more complex potentials (soft wall obstacle).

Acknowledgments

This research was supported by a Hughes-Christensen research award. The author wishes to thank Antonella Zanna and Robert McLachlan for their comments.

References

- [1] M. Austin, P. S. Krishnaprasad, and L. S. Wang. Almost Lie-Poisson integrators for the rigid body. *Journal of Computational Physics*, 107:105–117, 1993.
- [2] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations.*, volume 31. Springer Series in Comput. Mathematics, Springer-Verlag, 2002.
- [3] Thomas Holder, Ben Leimkuhler, and Sebastian Reich. Explicit variable step size and time reversible integration. *Applied Numerical Mathematics*, 39:367–377, 2001.
- [4] G. Hulbert. Explicit momentum conserving algorithms for rigid body dynamics. *Computers and Structures*, 44(6):1291–1303, 1992.
- [5] A. Iserles, H.Z. Munthe-Kaas, S. P. Norsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.
- [6] P. Krysl and L. Endres. Explicit Newmark/Verlet algorithm for time integration of the rotational dynamics of rigid bodies. *International Journal for Numerical Methods in Engineering*, (?):?, 2004. Submitted.
- [7] D. Lewis and J. C. Simo. Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups. *J. Nonlinear Sci.*, 4:253–299, 1995.
- [8] Robert I. McLachlan and A. Zanna. The discrete Moser-Veselov algorithm for the free rigid body, revisited. Reports in Informatics 255, University of Bergen, 2003.
- [9] J. C. Simo and K. K. Wong. Unconditionally stable algorithms for the orthogonal group that exactly preserve energy and momentum. *International Journal for Numerical Methods in Engineering*, 31:19–52, 1991.

- [10] A. Zanna, K. Eng, and H. Munthe-Kaas. Adjoint and selfadjoint Lie-group methods. *BIT*, 41(2):395–421, 2001.

Appendix

It needs to be realized that the classical midpoint rule **AKW** (almost Lie-Poisson integrator) of Austin, Krishnaprasad, Wang [1] differs substantially from the present midpoint Lie algorithms. For easy reference, the algorithm is summarized here.

Algorithm **AKW**:

Given $\mathbf{\Omega}_0, \mathbf{R}_0$,

for $n = 1, 2, \dots$

Solve $I\mathbf{\Omega}_n = I\mathbf{\Omega}_{n-1} - \tilde{\mathbf{\Omega}}I\bar{\mathbf{\Omega}} + \frac{\Delta t}{2}(\mathbf{T}_n + \mathbf{T}_{n-1})$

where $\bar{\mathbf{\Omega}} = \frac{1}{2}(\mathbf{\Omega}_n + \mathbf{\Omega}_{n-1})$

$\mathbf{R}_n = \mathbf{R}_{n-1}\text{cay}[\Delta t\tilde{\mathbf{\Omega}}]$

end