

Explicit Newmark/Verlet algorithm for Time Integration of the Rotational Dynamics of Rigid Bodies

P. Krysl*, and L. Endres

April 20, 2004

Introduction

We consider the problem of integration of the initial value problem of the rotational rigid body dynamics. We are motivated by a very practical problem: time integration of the equations of motion of drill bits as they cut through rock. The drill bit geometry is fully three-dimensional, as is the surface of the rock to be cut, and the force laws of cutting, scraping, impacts, and persistent contacts are heuristic, but computationally intensive anyway. Consequently, evaluation of the forces acting at various points on the bit is very expensive. (Similar observations may be made in many molecular dynamics simulations, where the evaluation of the forcing may take as much as 90% or more of the CPU time, or in finite element dynamics calculations.) Therefore it makes eminent sense to try to limit the number of evaluations of the external torque, and we are thus led to consider methods that are explicit in the torque calculation, i.e. the torque is evaluated just once per time step, in preference to implicit methods (torque evaluated many times per time step).

We are generally looking for an integrator that is robust, accurate, and efficient. Conservation properties are often essential for the preservation of key qualitative features of the motion. Equally important are often the dissipative and forcing characteristics [19]. In addition, three-dimensional rotations pose a special challenge in that the configuration set is a curved manifold, not a vector space. The integrator should maintain the configuration on the manifold. Methods for this type of problem have been studied by many researchers, both in the computational mechanics community, and in the applied mathematics community [5, 27, 1, 24, 20, 32, 21, 29, 25, 42, 23, 8, 39, 2, 28, 10, 40, 22, 17, 33, 15, 41, 31, 26, 13, 6].

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

The issue of the integration of the equations of motion of a three-dimensional rotating body may be approached from the general viewpoint of geometric integration [4, 17]. A good example of a geometric integrator is the Verlet algorithm [38], whose generalization is the Newmark algorithm [30]. The Verlet/Newmark explicit integrator has been recently shown to possess very attractive geometrical properties [18, 19, 12]: Long-term energy and momentum near-conservation being perhaps the most important for our applications. Simo and Wong [35] have studied the algorithms for rigid body rotation, and have proposed that the particular version of their algorithm which used $\beta = 0$ was the explicit Newmark. However, their algorithm does not display the same kind of behaviors that we see in the classical vector space version of Newmark (crucially, it is not symplectic).

Our goal is to reformulate the traditional velocity based Newmark algorithm for vector spaces for the rotational dynamics of rigid bodies, that is for the setting of Lie groups. In fact, we show that the most naive re-write of the vector space algorithm does possess the desired properties, and is in fact one of the best-performing currently known explicit algorithms for rigid body dynamics. Curiously, this algorithm does not seem to have been considered in the open literature at all. We claim that this algorithm is the true rotation-group counterpart of the vector space velocity-based explicit Newmark.

1 Formulation

In this section we very briefly present the formulation of the kinematics and dynamics of the motion, chiefly to introduce the notation.

1.1 Kinematics

We shall consider rotations of a rigid body about a fixed point, located at the origin of the global Cartesian system. An arbitrary particle in the reference configuration is addressed by the vector $\bar{\mathbf{p}}_0$. The current position of the particle is expressed with the current rotation (attitude) matrix \mathbf{R} as

$$\bar{\mathbf{p}} = \mathbf{R} \cdot \bar{\mathbf{p}}_0 \tag{1}$$

where \mathbf{R} is an orthogonal transformation, $\mathbf{R}^{-1} = \mathbf{R}^T$.

1.1.1 Configuration

The orthogonal transformations \mathbf{R} that represent rotations constitute points of a particular manifold, the special orthogonal group $\text{SO}(3)$. The motion of the rigid body particles is therefore given as a mapping

$$t \mapsto \mathbf{R} \in \text{SO}(3) , \tag{2}$$

where the presence of the special orthogonal group is what makes the problem of devising suitable integrators interesting.

The rotation matrix may be parameterized in many different ways [3]. In this work we use the rotation vector to express rotation matrices. Thus,

$$\mathbf{R}(\Psi) = \sum_{k=0}^{\infty} \frac{\tilde{\Psi}^k}{k!} \quad (3)$$

or, in the form of the Rodrigues formula

$$\mathbf{R}(\Psi) = \mathbf{1} + \frac{\sin \|\Psi\|}{\|\Psi\|} \tilde{\Psi} + \frac{(1 - \cos \|\Psi\|)}{\|\Psi\|^2} \tilde{\Psi}^2 . \quad (4)$$

When needed, the rotation vector may be extracted from the rotation matrix with the Spurrier's algorithm (see for instance reference [34]).

1.1.2 Velocity

The velocity vector of a given particle is obtained from (1) as

$$\dot{\mathbf{p}} = \dot{\mathbf{R}}\bar{\mathbf{p}}_0 . \quad (5)$$

We may define

$$\dot{\mathbf{R}} = \mathbf{R}\tilde{\Omega} \quad (6)$$

where $\tilde{\Omega}$ is the body-attached (convected) angular velocity. The wave symbol is used to indicate a skew-symmetric matrix, which is related to the angular velocity Ω as a vector through $\tilde{\Omega} \cdot \Omega = \mathbf{0}$.

Alternatively, the time derivative (6) may be expressed as

$$\dot{\mathbf{R}} = \tilde{\omega}\mathbf{R} \quad (7)$$

where $\tilde{\omega}$ is the spatial angular velocity as a skew-symmetric matrix; the spatial vector of angular velocity ω is defined as $\tilde{\omega} \cdot \omega = \mathbf{0}$.

Skew-symmetric matrices are elements of the vector space

$$so(3) = \left\{ \tilde{\mathbf{h}} : \mathbb{R}^3 \mapsto \mathbb{R}^3 \mid \tilde{\mathbf{h}} + \tilde{\mathbf{h}}^T = \mathbf{0} \right\} \quad (8)$$

with a map into \mathbb{R}^3 defined as

$$\tilde{\mathbf{h}}\mathbf{v} = \mathbf{h} \times \mathbf{v} \quad (9)$$

The spatial and the convected angular velocity are related through the two equivalent relations

$$\tilde{\omega} = \mathbf{R}\tilde{\Omega}\mathbf{R}^T \quad \text{and} \quad \omega = \mathbf{R}\Omega . \quad (10)$$

The velocity of a given particle in the body may thus be written as

$$\dot{\mathbf{p}} = \mathbf{R}\tilde{\Omega}\bar{\mathbf{p}}_0 = \tilde{\omega}\mathbf{R}\bar{\mathbf{p}}_0 \quad (11)$$

1.1.3 Angular acceleration

Proceeding with time differentiation, we obtain for the acceleration

$$\ddot{\tilde{\mathbf{p}}} = \mathbf{R}(\tilde{\boldsymbol{\Omega}}^2 + \tilde{\mathbf{A}})\tilde{\mathbf{p}}_0 = (\tilde{\boldsymbol{\omega}}^2 + \tilde{\boldsymbol{\alpha}})\mathbf{R}\tilde{\mathbf{p}}_0 \quad (12)$$

where $\tilde{\mathbf{A}}$ is the body frame (convected) angular acceleration as a skew-symmetric tensor, $\tilde{\mathbf{A}} = \dot{\tilde{\boldsymbol{\Omega}}}$; $\tilde{\boldsymbol{\alpha}}$ is the spatial angular acceleration as a skew-symmetric tensor, $\tilde{\boldsymbol{\alpha}} = \dot{\tilde{\boldsymbol{\omega}}}$.

1.2 Dynamics

1.2.1 Principle of Virtual Power

The virtual velocity may be written as

$$\boldsymbol{\eta} = \mathbf{R}\tilde{\mathbf{H}}_\Omega\tilde{\mathbf{p}}_0 = -\mathbf{R}\tilde{\mathbf{p}}_0\mathbf{H}_\Omega \quad (13)$$

where \mathbf{H}_Ω is the virtual angular velocity in the body frame.

1.2.2 Virtual power of the inertial forces

The virtual power of the inertial forces may be written as

$$\mathcal{P}_i = \int_V \boldsymbol{\eta} \cdot \rho \ddot{\tilde{\mathbf{p}}} \, dV \quad (14)$$

where $\boldsymbol{\eta}$ is the virtual velocity; ρ is the mass density; and V is the volume of the body.

$$\begin{aligned} \mathcal{P}_i &= \int_V (\mathbf{R}\tilde{\mathbf{p}}_0\mathbf{H}_\Omega) \cdot \rho(\mathbf{R}(\tilde{\boldsymbol{\Omega}}^2 + \tilde{\mathbf{A}})\tilde{\mathbf{p}}_0) \, dV \\ &= \int_V (\mathbf{R}\tilde{\mathbf{p}}_0\mathbf{H}_\Omega) \cdot \rho(\mathbf{R}(-\tilde{\boldsymbol{\Omega}}\tilde{\mathbf{p}}_0\boldsymbol{\Omega} - \tilde{\mathbf{p}}_0\tilde{\mathbf{A}})) \, dV \end{aligned} \quad (15)$$

which may be put into a familiar form using the tensor of inertia

$$\begin{aligned} \mathcal{P}_i &= \mathbf{H}_\Omega \cdot \tilde{\boldsymbol{\Omega}} \int_V \rho \tilde{\mathbf{p}}_0^T \tilde{\mathbf{p}}_0 \, dV \boldsymbol{\Omega} + \mathbf{H}_\Omega \cdot \int_V \rho \tilde{\mathbf{p}}_0^T \tilde{\mathbf{p}}_0 \, dV \mathbf{A} \\ &= \mathbf{H}_\Omega \cdot \tilde{\boldsymbol{\Omega}} \mathbf{I} \boldsymbol{\Omega} + \mathbf{H}_\Omega \cdot \mathbf{I} \mathbf{A} \end{aligned} \quad (16)$$

where $\mathbf{I} = \int_V \rho \tilde{\mathbf{p}}_0^T \tilde{\mathbf{p}}_0 \, dV$ is the tensor of inertia in the body frame.

1.3 Virtual power of the applied forces

For a given force \mathbf{f} applied at the point $\tilde{\mathbf{q}}$, the virtual power may be written as

$$\mathcal{P}_e = \boldsymbol{\eta} \cdot \mathbf{f} = (\mathbf{R}\tilde{\mathbf{q}}_0\mathbf{H}_\Omega) \cdot \mathbf{f} = \mathbf{H}_\Omega \tilde{\mathbf{q}}_0 \cdot \mathbf{R}^T \mathbf{f} = \mathbf{H}_\Omega \cdot \mathbf{T} \quad (17)$$

where $\mathbf{T} = \tilde{\mathbf{q}}_0 \cdot \mathbf{R}^T \mathbf{f}$ is the torque in the body frame.

1.4 Equation of motion

The equation of motion follows from the principle of virtual power as

$$\mathcal{P}_i = \mathcal{P}_e \quad (18)$$

or explicitly

$$\tilde{\Omega}I\Omega + IA = T \quad (19)$$

The above equation of motion in the body frame may be also cast in the spatial frame

$$\tilde{\omega}i\omega + ia = t \quad (20)$$

where $i = \mathbf{R}I\mathbf{R}^T$ is the tensor of inertia in the spatial frame; and $t = \mathbf{R}T$ is the torque in the spatial frame.

2 Time Discretization

The classical Newmark integration algorithm is formulated as follows, using two parameters, γ and β [30]

$$\Delta \mathbf{q}_n = \Delta t \mathbf{v}_{n-1} + \frac{\Delta t^2}{2} [(1 - 2\beta)\mathbf{a}_{n-1} + 2\beta\mathbf{a}_n] \quad (21)$$

$$\mathbf{v}_n = \mathbf{v}_{n-1} + \Delta t [(1 - \gamma)\mathbf{a}_{n-1} + \gamma\mathbf{a}_n] \quad (22)$$

where $\Delta \mathbf{q}_j = \mathbf{q}_j - \mathbf{q}_{j-1}$ is the increment of the algorithmic position vector at time t_j ; \mathbf{v}_j is the algorithmic velocity of the vector \mathbf{q}_j at time t_j ; \mathbf{a}_j is the algorithmic acceleration of the vector \mathbf{q}_j at time t_j . Note that all the vectors \mathbf{q}_j , \mathbf{v}_j , and \mathbf{a}_j must belong to a vector space S ,

$$\mathbf{q}_j \in S, \quad \mathbf{v}_j \in S, \quad \mathbf{a}_j \in S$$

in order for the linear combination in equations (21,22) to make sense.

For the particular choice $\beta = 0$ the Newmark algorithm becomes explicit:

$$\Delta \mathbf{q}_n = \Delta t \mathbf{v}_{n-1} + \frac{\Delta t^2}{2} \mathbf{a}_{n-1} \quad (23)$$

$$\mathbf{v}_n = \mathbf{v}_{n-1} + \Delta t [(1 - \gamma)\mathbf{a}_{n-1} + \gamma\mathbf{a}_n] \quad (24)$$

2.1 Compound rotations and tangents

While the increment $\Delta \mathbf{q}_n = \mathbf{q}_n - \mathbf{q}_{n-1}$ may be expressed as a difference of two configurations (locations) in (23), the transition in $\text{SO}(3)$ from the rotation \mathbf{R}_{n-1} to the rotation \mathbf{R}_n needs to be computed in terms of compound rotations.

A compound rotation curve may be constructed from a given rotation \mathbf{R} either by left or right multiplication with another finite rotation matrix, parameterized with

the scalar ϵ . Left multiplication by the matrix $\delta\mathbf{R}$ corresponds to an incremental rotation applied in the spatial setting

$$\mathbf{R}' = \delta\mathbf{R} \cdot \mathbf{R} , \quad (25)$$

right multiplication by the matrix $\Delta\mathbf{R}$ corresponds to an incremental rotation applied in the body-frame setting

$$\mathbf{R}' = \mathbf{R} \cdot \Delta\mathbf{R} \quad (26)$$

The incremental matrices are expressible as rotations about a given, fixed (rotation) vector. Thus, the left incremental rotation $\delta\mathbf{R}$ may be written as

$$\delta\mathbf{R} = \exp[\epsilon\widetilde{\delta\psi}] , \quad (27)$$

and the right incremental rotation $\Delta\mathbf{R}$ may be written as

$$\Delta\mathbf{R} = \exp[\epsilon\widetilde{\Delta\Psi}] . \quad (28)$$

Tangent vectors to the compound curves may be evaluated as

$$\left. \frac{d\mathbf{R}'}{d\epsilon} \right|_{\epsilon=0} = \widetilde{\delta\psi}\mathbf{R} = \mathbf{R}\widetilde{\Delta\Psi} \quad (29)$$

2.2 Rotation update

For now we shall proceed with formulation in the body frame. The incremental vector $\Delta\Psi$ belongs to a vector space, the tangent space $T_{\mathbf{R}}\text{SO}(3)$ to the manifold $\text{SO}(3)$ at the rotation \mathbf{R}

$$\Delta\Psi \in T_{\mathbf{R}}\text{SO}(3) .$$

Both the body-frame angular velocity $\boldsymbol{\Omega}$, and angular acceleration \mathbf{A} belong also to the same vector space

$$\boldsymbol{\Omega} \in T_{\mathbf{R}}\text{SO}(3), \quad \mathbf{A} \in T_{\mathbf{R}}\text{SO}(3) .$$

Notice that all quantities in equation (23) are expressed in the same configuration at time t_{n-1} . Therefore, it makes sense to construct the incremental rotation vector $\Delta\Psi_n$ that will take us from the rotation \mathbf{R}_{n-1} to the rotation \mathbf{R}_n by combining the angular velocity and the angular acceleration. In other words, the Newmark algorithm may be applied in the vector space $T_{\mathbf{R}_{n-1}}\text{SO}(3)$, so that equation (23) may be specialized as

$$\Delta\Psi_n = \Delta t\boldsymbol{\Omega}_{n-1} + \frac{\Delta t^2}{2}\mathbf{A}_{n-1}, \quad (30)$$

and the configuration will be therefore updated as

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

2.3 Angular velocity update

Now we proceed to the second Newmark equation, the update of the angular velocity (24). At variance with (23), this equation mixes together accelerations and velocities from two different time steps, and hence belonging to two different tangent spaces. Therefore, we need to introduce the concept of the dexp operator (differential of the exponential map) [17]. For the case of rotations in 3-D, we may refer for instance to [5], and relate increments to rotation vectors in two different tangent spaces, at \mathbf{R}_A and $\mathbf{R}_B = \mathbf{R}_A \exp(\Psi)$. We express a curve $\mathbf{R}_{B,\epsilon}$ using increments in the two tangent spaces, $\Theta_A \in T_{\mathbf{R}_A} \text{SO}(3)$ and $\Theta_B \in T_{\mathbf{R}_B} \text{SO}(3)$

$$\mathbf{R}_{B,\epsilon} = \mathbf{R}_A \exp(\Psi + \epsilon \Theta_A) = \mathbf{R}_B \exp(\epsilon \Theta_B) \quad (31)$$

As shown in Reference [5], the two rotation vectors may be then related as

$$\Theta_B = \mathbf{S}(\Psi) \Theta_A, \quad (32)$$

where the operator that relates the two tangent spaces is

$$\text{dexp}_\Psi = \mathbf{S}(\Psi) = \mathbf{1} + \frac{1 - \cos \|\Psi\|}{\|\Psi\|^2} \tilde{\Psi} + \left(1 - \frac{\sin \|\Psi\|}{\|\Psi\|}\right) \frac{\tilde{\Psi}^2}{\|\Psi\|^2}. \quad (33)$$

Relation (32) may be applied to the construction of the update of angular velocities. For instance, we could push forward quantities from configuration at time t_{n-1} to configuration at t_n by the rotation increment $\Delta \Psi_n$, which yields

$$\Omega_n = \mathbf{S}(\Delta \Psi_n) \Omega_{n-1} + \Delta t [(1 - \gamma) \mathbf{S}(\Delta \Psi_n) \mathbf{A}_{n-1} + \gamma \mathbf{A}_n] \quad (34)$$

We could stop here, but there's more that can be done. If we rewrite equation (34) as

$$\Omega_n = \mathbf{S}(\Delta \Psi_n) [\Omega_{n-1} + \Delta t (1 - \gamma) \mathbf{A}_{n-1}] + \Delta t \gamma \mathbf{A}_n, \quad (35)$$

we may realize that the contents of the bracket are co-axial with the incremental rotation vector $\Delta \Psi_n$ from equation (30), provided $\gamma = 1/2$. Thus, for this choice of the coefficient γ , the shift with $\mathbf{S}(\Delta \Psi_n)$ has no effect [hint: use equation (9)], and may be omitted. Since the choice of $\gamma = 1/2$ is known to be optimal from the point of view of accuracy, it also appears in the Verlet algorithm, and we are going to use this numerical value too. Therefore, we shall be using

$$\Omega_n = \Omega_{n-1} + \frac{\Delta t}{2} (\mathbf{A}_{n-1} + \mathbf{A}_n), \quad (36)$$

as the angular velocity update equation.

2.3.1 Evaluated acceleration

The last essential component of the algorithm (21) is the algorithmic (evaluated) acceleration, which is readily obtained from the equation of motion (19). As will become clear in a moment, we need to solve a system of nonlinear equations to obtain the acceleration, since it is coupled with the angular velocity due to the velocity update.

The three ingredients, the configuration update, the velocity update, and the use of the evaluated acceleration combine, in some way that might not be at this point completely understood, to balance good behaviors (energy and momentum representation) in a nearly optimal way.

2.4 Explicit Newmark algorithm in the body frame

To summarize equations (30) and (36), the explicit Newmark algorithm ($\beta = 0$, and $\gamma = 1/2$) in the body frame (NMB) may be written as

```

Algorithm NMB:
  Given  $\mathbf{\Omega}_0, \mathbf{R}_0,$ 
   $\mathbf{A}_0 = \mathbf{I}^{-1}(\mathbf{R}_0^T \mathbf{T}_0 - \mathbf{\Omega}_0 \times \mathbf{I} \cdot \mathbf{\Omega}_0)$ 
  for  $n = 1, 2, \dots$ 
     $\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\Delta t \tilde{\mathbf{\Omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\mathbf{A}}_{n-1}]$ 
     $\mathbf{I} \mathbf{A}_n - \mathbf{R}_n^T \mathbf{T}_n + \mathbf{\Omega}_n \times \mathbf{I} \cdot \mathbf{\Omega}_n = \mathbf{0}$ 
     $\mathbf{\Omega}_n = \mathbf{\Omega}_{n-1} + \frac{\Delta t}{2}(\mathbf{A}_{n-1} + \mathbf{A}_n)$ 
  end

```

Note that the algorithm is self-starting, which is particularly nice from the point of view of implementation. The integrator algorithm may be called repeatedly with the output of the previous invocation as the initial conditions, and there will be no loss of accuracy involved in doing so.

Also of note is the fact that we never actually need the components of the rotation vector (or those of the associated quaternion): the update is performed on the rotation matrix, and in our numerical experiments we have not come across numerical difficulties. However, should such a need arise it is always possible to re-orthogonalize \mathbf{R} , or extract the rotation vector components. Of course, that is what we have to do when the rotation vector components, or the components of the incremental rotation vector are needed for output.

2.5 Explicit Newmark algorithm in the spatial frame

Contrary to popular belief, the spatial form of the above body-frame Newmark algorithm as given below makes geometric sense and is completely equivalent to the body-frame version.

Algorithm NMS:

Given $\boldsymbol{\omega}_0, \mathbf{R}_0$,
 $\boldsymbol{\alpha}_0 = \mathbf{i}_0^{-1}(\mathbf{t}_0 - \boldsymbol{\omega}_0 \times \mathbf{i}_0 \cdot \boldsymbol{\omega}_0)$
for $n = 1, 2, \dots$
 $\mathbf{R}_n = \exp[\Delta t \tilde{\boldsymbol{\omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\boldsymbol{\alpha}}_{n-1}] \mathbf{R}_{n-1}$
 $\mathbf{i}_n \boldsymbol{\alpha}_n - \mathbf{t}_n + \boldsymbol{\omega}_n \times \mathbf{i}_n \cdot \boldsymbol{\omega}_n = \mathbf{0}$
 $\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2} (\boldsymbol{\alpha}_{n-1} + \boldsymbol{\alpha}_n)$
end

To show the equivalence, we may start with the rotation update equation and to substitute the convected angular velocity and convected angular acceleration

$$\begin{aligned}
\mathbf{R}_n &= \mathbf{R}_{n-1} \exp[\Delta t \tilde{\boldsymbol{\Omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\mathbf{A}}_{n-1}] \\
&= \mathbf{R}_{n-1} \sum_{k=0}^{\infty} \frac{(\Delta t \tilde{\boldsymbol{\Omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\mathbf{A}}_{n-1})^k}{k!} \\
&= \mathbf{R}_{n-1} \sum_{k=0}^{\infty} \frac{(\Delta t \mathbf{R}_{n-1}^T \tilde{\boldsymbol{\omega}}_{n-1} \mathbf{R}_{n-1} + \frac{\Delta t^2}{2} \mathbf{R}_{n-1}^T \tilde{\boldsymbol{\alpha}}_{n-1} \mathbf{R}_{n-1})^k}{k!} \\
&= \sum_{k=0}^{\infty} \frac{(\Delta t \tilde{\boldsymbol{\omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\boldsymbol{\alpha}}_{n-1})^k}{k!} \mathbf{R}_{n-1} \\
&= \exp[\Delta t \tilde{\boldsymbol{\omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\boldsymbol{\alpha}}_{n-1}] \mathbf{R}_{n-1}
\end{aligned} \tag{37}$$

Analogously, we obtain the spatial version of the equation of motion

$$\begin{aligned}
\mathbf{I} \mathbf{A}_n - \mathbf{R}_n^T \mathbf{T}_n + \boldsymbol{\Omega}_n \times \mathbf{I} \cdot \boldsymbol{\Omega}_n &= \\
\mathbf{R}_n^T \mathbf{i}_n \boldsymbol{\alpha}_n - \mathbf{R}_n^T \mathbf{T}_n + \mathbf{R}_n^T \boldsymbol{\omega}_n \times \mathbf{i}_n \cdot \boldsymbol{\omega}_n &= \mathbf{0}
\end{aligned} \tag{38}$$

where we have to substitute for the angular velocity at time t_n from

$$\begin{aligned}
\boldsymbol{\Omega}_n &= \boldsymbol{\Omega}_{n-1} + \frac{\Delta t}{2} (\mathbf{A}_{n-1} + \mathbf{A}_n) \\
&= \mathbf{R}_n^T \boldsymbol{\omega}_n = \mathbf{R}_{n-1}^T \boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2} (\mathbf{R}_{n-1}^T \boldsymbol{\alpha}_{n-1} + \mathbf{R}_n^T \boldsymbol{\alpha}_n)
\end{aligned} \tag{39}$$

Thus, we obtain

$$\boldsymbol{\omega}_n = \mathbf{R}_n \mathbf{R}_{n-1}^T \boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2} (\mathbf{R}_n \mathbf{R}_{n-1}^T \boldsymbol{\alpha}_{n-1} + \boldsymbol{\alpha}_n) \tag{40}$$

However, from equation (37) we have $\mathbf{R}_n \mathbf{R}_{n-1}^T = \exp[\Delta t \tilde{\boldsymbol{\omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\boldsymbol{\alpha}}_{n-1}]$. To complete the proof, we need to realize that the vector $\boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2} \boldsymbol{\alpha}_{n-1}$ is a scalar multiple of the axial vector of the rotation matrix $\exp[\Delta t \tilde{\boldsymbol{\omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\boldsymbol{\alpha}}_{n-1}]$, and is not affected by the rotation, and therefore

$$\begin{aligned}
\boldsymbol{\omega}_n &= \exp[\Delta t \tilde{\boldsymbol{\omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\boldsymbol{\alpha}}_{n-1}] (\boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2} \boldsymbol{\alpha}_{n-1}) + \frac{\Delta t}{2} \boldsymbol{\alpha}_n \\
&= \boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2} (\boldsymbol{\alpha}_{n-1} + \boldsymbol{\alpha}_n)
\end{aligned} \tag{41}$$

2.6 Solving the equation of motion

Note that both algorithms **NMB** and **NMS** need to obtain the angular accelerations by solving a nonlinear algebraic equation. For instance, for the algorithm **NMS** the following equation in residual form needs to be solved

$$\mathcal{R}(\boldsymbol{\alpha}_n) = \mathbf{I}_n \boldsymbol{\alpha}_n - \mathbf{T}_n + \boldsymbol{\omega}_n \times \mathbf{I}_n \cdot \boldsymbol{\omega}_n = \mathbf{0} \quad (42)$$

where $\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2}(\boldsymbol{\alpha}_{n-1} + \boldsymbol{\alpha}_n)$, or; $\boldsymbol{\omega}_n = \boldsymbol{\omega}_{n,p} + \frac{\Delta t}{2}\boldsymbol{\alpha}_n$, with the definition $\boldsymbol{\omega}_{n,p} = \boldsymbol{\omega}_{n-1} + \frac{\Delta t}{2}\boldsymbol{\alpha}_{n-1}$. We may write a straightforward Newton's solver as

$$\begin{aligned} &\boldsymbol{\alpha}_n^{(0)} = \mathbf{0}, i = 1 \\ &\text{while } \|\mathcal{R}(\boldsymbol{\alpha}_n^{(i-1)})\| > TOL \\ &\quad \boldsymbol{\alpha}_n^{(i)} = \boldsymbol{\alpha}_n^{(i-1)} - \left[\frac{\partial \mathcal{R}(\boldsymbol{\alpha}_n^{(i-1)})}{\partial \boldsymbol{\alpha}_n} \right]^{-1} \mathcal{R}(\boldsymbol{\alpha}_n^{(i-1)}) \\ &\text{end} \end{aligned}$$

with the tangent $\partial \mathcal{R} / \partial \boldsymbol{\alpha}_n$

$$\frac{\partial \mathcal{R}(\boldsymbol{\alpha}_n^{(i-1)})}{\partial \boldsymbol{\alpha}_n} = \mathbf{I}_n + \frac{\Delta t}{2} \left(\widetilde{\boldsymbol{\omega}_{n,p}} \mathbf{I}_n - \mathbf{I}_n \widetilde{\boldsymbol{\omega}_{n,p}} \right) + \frac{\Delta t^2}{2} \left(\widetilde{\boldsymbol{\alpha}_n} \mathbf{I}_n - \mathbf{I}_n \widetilde{\boldsymbol{\alpha}_n} \right) \quad (43)$$

3 Numerical Experiments

We are going to compare the performance of the Newmark algorithm (**NMB**, **NMS**) with the performance of the two top-performing explicit algorithms, the staggered momentum-conserving explicit algorithm [16] (**HSTAGM**) (see also discussion in Svanberg [36]), and the explicit time integration algorithm of Simo and Wong [35] (**SW**). For reader's convenience, both algorithms are summarized in the Appendix A. For comparison we also include references to the canonical implicit midpoint almost-Lie-Poisson algorithm of Austin, Krishnaprasad, Wang [1] (**AKW**).

3.1 Freely spinning body (McLachlan, Zanna 2003)

This example is discussed in the report [26] in the context of discrete Moser-Veselov integrators for the rigid body. The initial condition is $\boldsymbol{\Omega} = (0.45549, 0.82623, 0.03476)$, 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 Newmark algorithm is in terms of kinetic energy. Even though the kinetic energy is not conserved, it oscillates with unchanging amplitude and without drift. 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 very well preserved even for large time steps. This

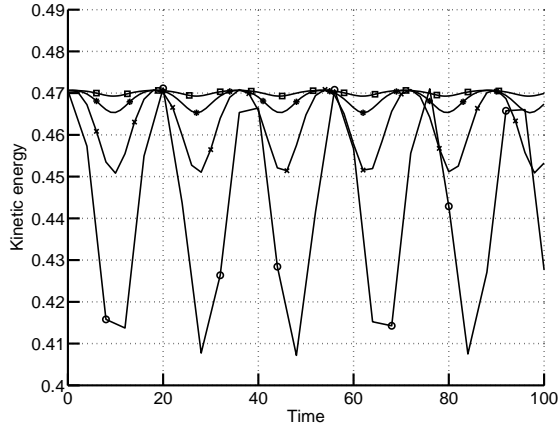


Figure 1: Freely spinning rigid body; kinetic energy for a step sizes 4 (\circ), 2 (\times), 1 (\star), $1/2$ (\square).

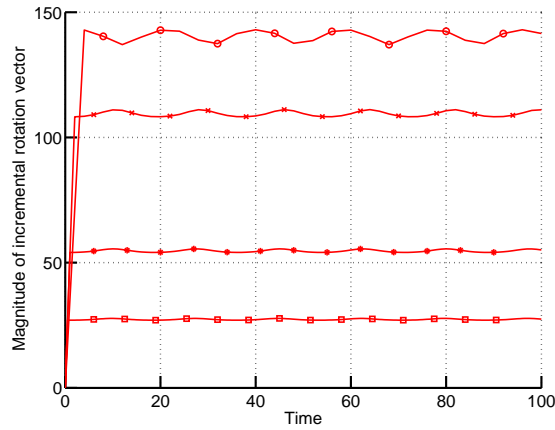


Figure 2: Freely spinning rigid body; magnitude of the incremental rotation vector in degrees for step sizes 4 (\circ), 2 (\times), 1 (\star), $1/2$ (\square).

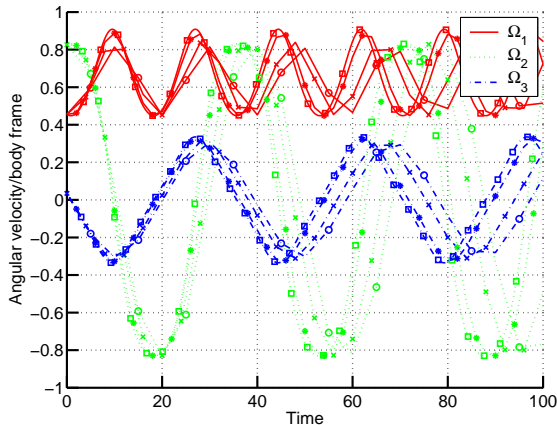


Figure 3: Freely spinning rigid body; components of the body-frame angular velocity for a step sizes 5 (\circ), 4 (\times), 2 (\star), $1/8$ (\square).

is a very attractive feature of the present Newmark algorithm. When compared with other algorithms discussed in reference [26] in figure 7.7, for the largest timestep, $\Delta t = 5$, our algorithm performs on par with the implicit midpoint rule (IMR, our **AKW**) which is the only one that is able to capture the character of the solution, while all the discrete Moser-Veselov integrators fail.

Figure 4 shows the convergence of the Newmark algorithm in the norm $\|\mathbf{R} - \mathbf{R}_{converged}\|_2$, where the orientation matrix $\mathbf{R}_{converged} = \mathbf{R}(t = 100)$ has been obtained with an extremely small step size of 0.001. Note that our algorithm outperforms the **AKW** by two orders of magnitude.

3.2 Unstable rotation: book toss

Consider a body described in the body frame whose directions coincide with the principal directions of the tensor of inertia. It is well known that if the three principal values of the tensor of inertia are different, rotation about the axis corresponding to the intermediate principal value is unstable [37]. Real world examples include the toss of a book. When the book is tossed with a spin imparted about the intermediate axis, the book will as likely as not flip in mid-flight due to the slightest disturbance.

The data for this example appear to be due to Simo and Wong [35]. It had also been studied by Hulbert [16]. A constant torque is applied about the axis of intermediate moment of inertia to a body that is initially at rest. Then, at time $t = t_d$, the constant torque is removed and a constant disturbance torque is applied about an axis perpendicular to that of the intermediate moment of inertia for the duration of a single timestep Δt . The body frame tensor of inertia is diagonal, $\text{diag}\mathbf{I} = (5, 10, 1)$. The spatial components of the torque are

$$\begin{cases} (20, 0, 0), & \text{for } 0 \leq t \leq t_d; \\ (0, 1/(5\Delta t), 0), & \text{for } t_d \leq t \leq t_d + \Delta t; \\ (0, 0, 0), & \text{for } t \geq t_d + \Delta t. \end{cases}$$

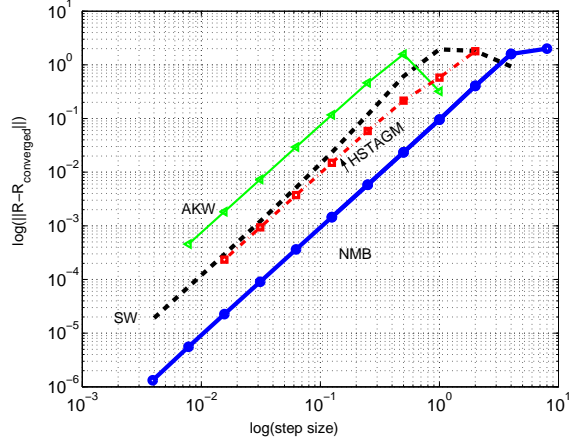


Figure 4: Freely spinning rigid body; convergence in the norm $\|\mathbf{R} - \mathbf{R}_{converged}\|$. Thick solid line (\circ) present Newmark (**NMB**); thick dotted line Simo-Wong explicit algorithm (**SW**); thick dash-dot line Lewis-Simo implicit algorithm (**LS**): for step sizes 8, 4, 2, 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, and 1/256;.

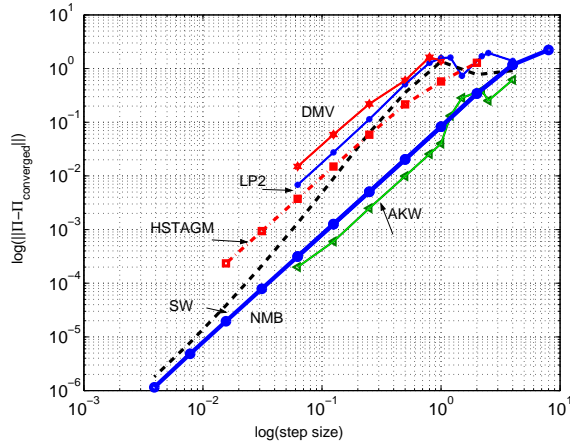


Figure 5: Freely spinning rigid body; convergence in the body-frame momenta, norm $\|\mathbf{M} - \mathbf{M}_{converged}\|$. Thick line (\circ , (**NMB**)): Newmark for step sizes 8, 4, 2, 1, 1/2, 1/4, 1/8 and 1/16 (1/32, 1/64, 1/128, 1/256); thick dotted line Simo-Wong explicit algorithm (**SW**); thick dash-dot line Lewis-Simo implicit algorithm (**LS**); (\ast): LP2 (explicit Lie-Poisson integrator of the second order); (\triangleleft): IMR (implicit midpoint rule); (\star): DMV (discrete Moser-Veselov); (\diamond): DMV4 (fourth order discrete Moser-Veselov); (\times): DMV6 (sixth order discrete Moser-Veselov).

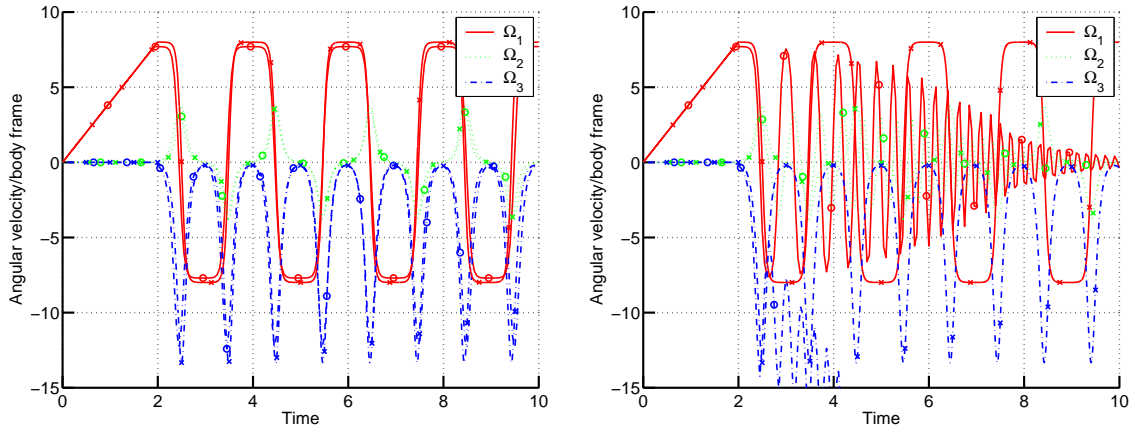


Figure 6: Book toss; convected angular velocities; left: **NMB** time step 0.05 (\circ), and 0.001 (\times), right: **SW** time step 0.05 (\circ), and 0.001 (\times).

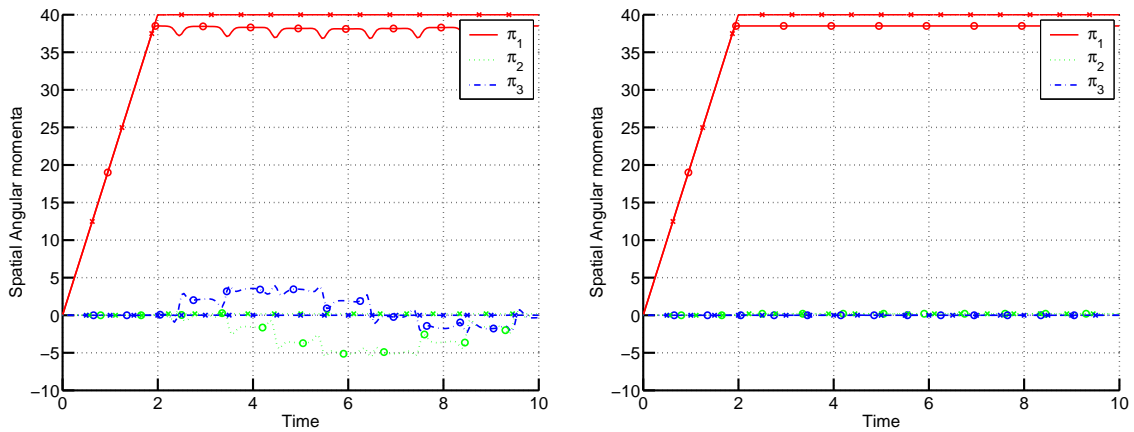


Figure 7: Book toss; spatial angular momenta; left: **NMB** time step 0.05 (\circ), and 0.001 (\times), right: **SW** time step 0.05 (\circ), and 0.001 (\times).

In Figure 6 the characteristic pattern of the body-frame angular velocities is displayed for two different time steps. The tendency of the **SW** algorithm to become unstable for larger time steps is quite clearly visible. On the other hand, the Newmark algorithm does not seem to have any trouble maintaining stability, and even the accuracy of the angular velocities seems quite acceptable.

Conservation of the angular momenta is illustrated in Figure 7: the Newmark algorithm clearly does not conserve momenta exactly, however, the conservation is achieved in a different sense. The oscillations of the momenta are limited, and seem to be more or less periodic in time. On the other hand, the **SW** algorithm conserves momenta exactly, even in those cases where it is actually unstable. This only illustrates that the conservation of momenta on its own does not necessarily mean that good results follow.

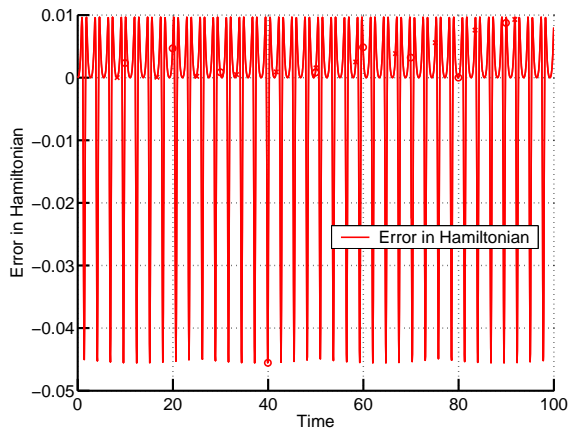


Figure 8: Lagrangian top; error in the Hamiltonian.

3.3 Lagrangian top

This particular problem had been studied with the number of methods [11, 9]. It is well known that the symmetric Lagrangian heavy top model possesses four conserved quantities:

1. the Hamiltonian,

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

where $\boldsymbol{\pi}$ is the spatial angular momentum, $\boldsymbol{\pi} = \boldsymbol{\pi} \cdot \boldsymbol{\omega}$; $\boldsymbol{\chi}$ is the unit vector along the axis of the top pointing from the fixed point; $\boldsymbol{\gamma}$ is the unit vector in the direction of gravity; Mlg is the 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, $\|\boldsymbol{R}\boldsymbol{\gamma}\|$.

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

The step size of the integration was $\Delta = 0.05$, and the dynamics was integrated on the interval $0 \leq t < 100$. The error in Hamiltonian is shown in Figure 8. Notice that the error in the Hamiltonian does not drift and stays fairly small (less than half a percent of the exact Hamiltonian).

The algorithm **NMB** (**NMS**) is about as accurate both in the preservation of the invariants and in the global error (convergence curve constant as well as slope) as the Crouch-Grossman method [7] with the Runge-Kutta RK4 tableau [11].

The projection of the spatial angular momentum onto the axis of the top is shown in Figure 9. The two curves correspond to a simulation which performs the rotation

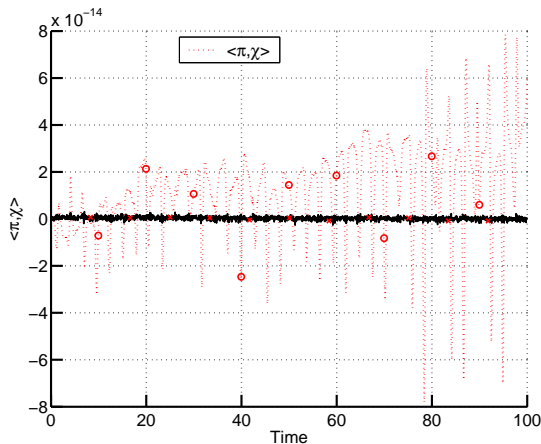


Figure 9: Lagrangian top; error in the projection of the spatial angular momentum onto the axis of the top.

update directly on the attitude matrix by matrix-matrix multiplication (larger error), and to a simulation in which the attitude matrix is re-orthogonalized in each timestep. Clearly, the error is then down to machine precision. Hardly any effect may be seen for either simulation in the physical characteristics of the motion, however.

3.4 Fast Lagrangian top

The data for this example appear to be due to Simo and Wong [35]. It had also been studied by Hulbert [16]. 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 $\mathbf{R}_0 = \exp[\widehat{\boldsymbol{\Psi}}_0]$ where $\boldsymbol{\Psi}_0 = [0.3; 0; 0]$, and $\boldsymbol{\Omega}_0 = [0; 0; 50]$. 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 $\|\mathbf{R} - \mathbf{R}_{\text{converged}}\|_2$, and the norm $\|\boldsymbol{\Pi} - \boldsymbol{\Pi}_{\text{converged}}\|_2$, where the reference values are the orientation matrix $\mathbf{R}_{\text{converged}} = \mathbf{R}(t = 10)$ and the body frame angular momentum $\boldsymbol{\Pi}_{\text{converged}} = \boldsymbol{\Pi}(t = 10)$. The convergence curves are shown in Figure 10. The **SW** and **HSTAGM** algorithms yield essentially identical answers except for the largest time steps.

3.5 Slow Lagrangian top

In this example we consider the motion of a *slow* symmetrical top with total mass M and axis of symmetry that coincides with the direction of uniform gravitational field.

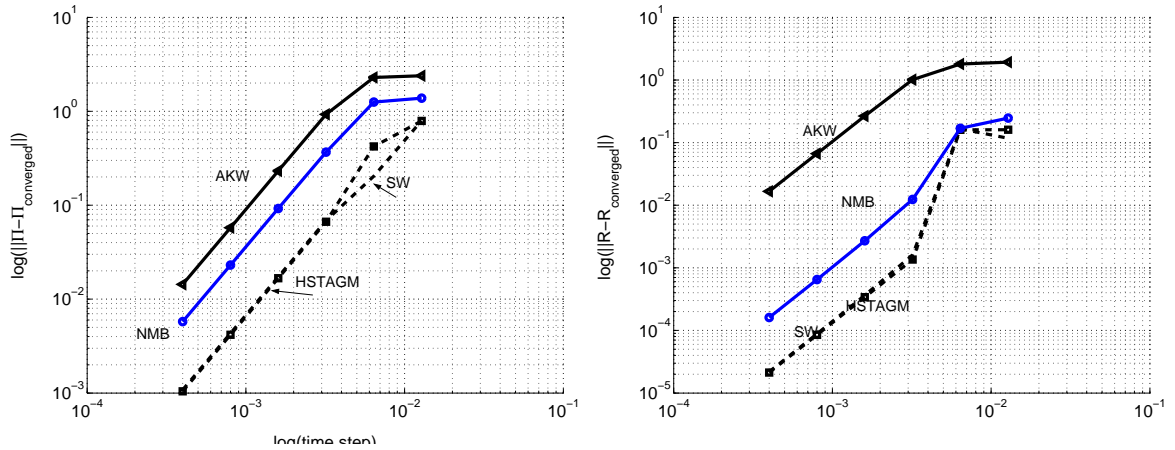


Figure 10: Fast Lagrangian top; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix: (\blacktriangle): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); (\square): **SW** (dashed line, no symbol) solutions with the Simo and Wong [35] algorithm; (\circ , solid line) **NMB** Newmark algorithm; (\square): explicit staggered algorithm with midpoint torque evaluation **HSTAGM** [16].

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 $\Psi_0 = [0.05; 0; 0]$, and $\Omega_0 = [0; 0; 5]$.

Figure 11 illustrates the convergence for the slow Lagrangian top. The Newmark algorithm is clearly a robust performer, delivering accuracy that improves on that of the implicit midpoint rule. The staggered explicit integrator **HSTAGM** has stability problems for the slow top, and also for the next example. Figure 12 illustrates this for the free rigid body and the slow Lagrangian top. For the free rigid body, the integrator **HSTAGM** produces a Hamiltonian that closely corresponds to the picture for the Newmark integrator, which leads us to believe that **HSTAGM** is perhaps symplectic, or possibly related to a symplectic algorithm. However, for the slow top we can clearly see instability developing early on, which leads to a blowup at the end of the simulation. If this algorithm is indeed symplectic it may be compromised by some numerically unstable process. The leap-frog (staggered) algorithm has been discussed more recently by Svanberg [36], and the inherent implicitness has been discussed in relation to the lack of stability of the explicit versions. (Note that the implicit leap-frog is indeed implicit in the torque evaluation.)

3.6 Body in Coulombic potential with soft wall contact

This problem had been used by the Holder, Leimkuhler, and Reich [15] to investigate the need for adaptive variable step size methods. The problem they consider is a rigid

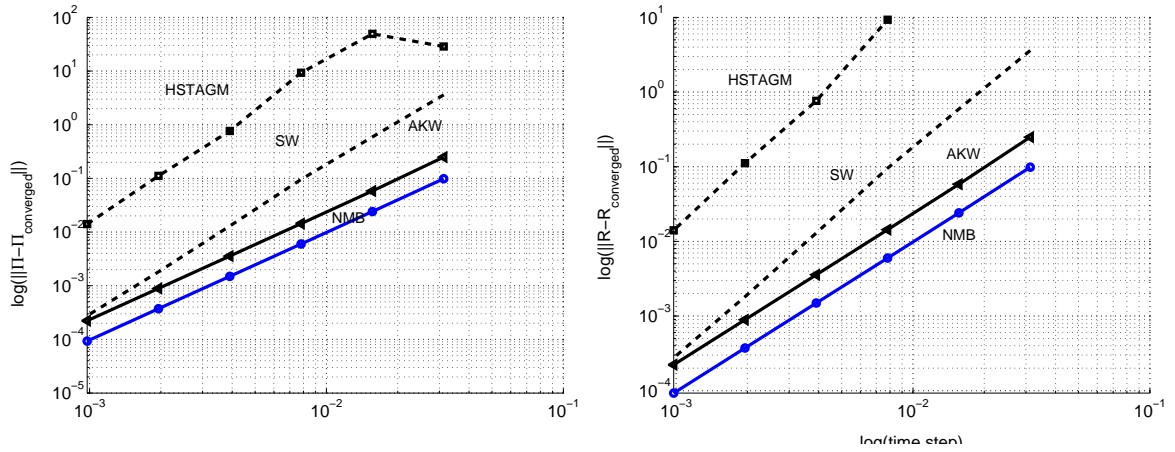


Figure 11: Slow Lagrangian top; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix: (\triangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (dashed line, no symbol) solutions with the Simo and Wong [35] algorithm; **NMB** (\circ , solid line) Newmark algorithm; (\square): explicit staggered algorithm with midpoint torque evaluation **HSTAGM** [16].

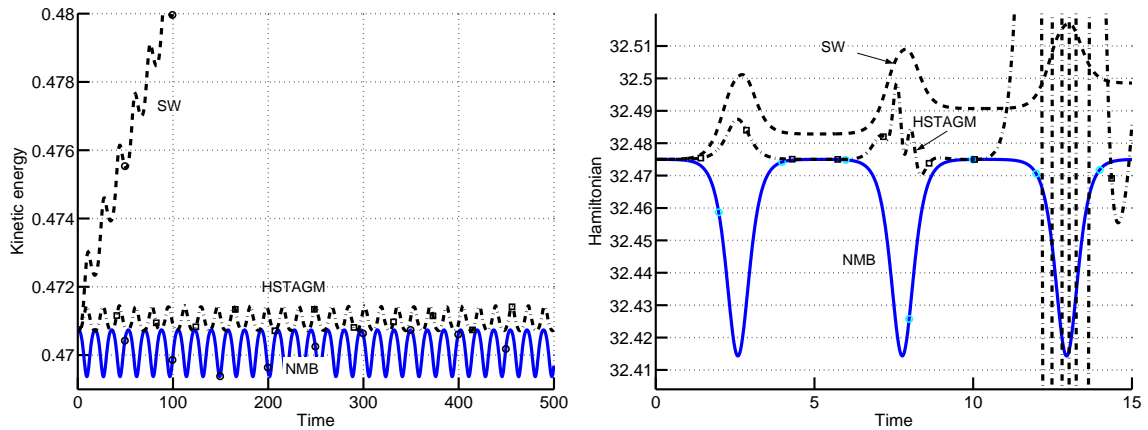


Figure 12: Free rigid body and Slow Lagrangian top Hamiltonians: **SW** (dashed line, no symbol) solutions with the Simo and Wong [35] algorithm; **NMB** (\circ , solid line) Newmark algorithm; (\square): explicit staggered algorithm with midpoint torque evaluation **HSTAGM** [16].

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 13 compares the convergence of the various algorithms discussed here. The Newmark algorithm clearly outperforms all the others, including the canonical implicit midpoint method.

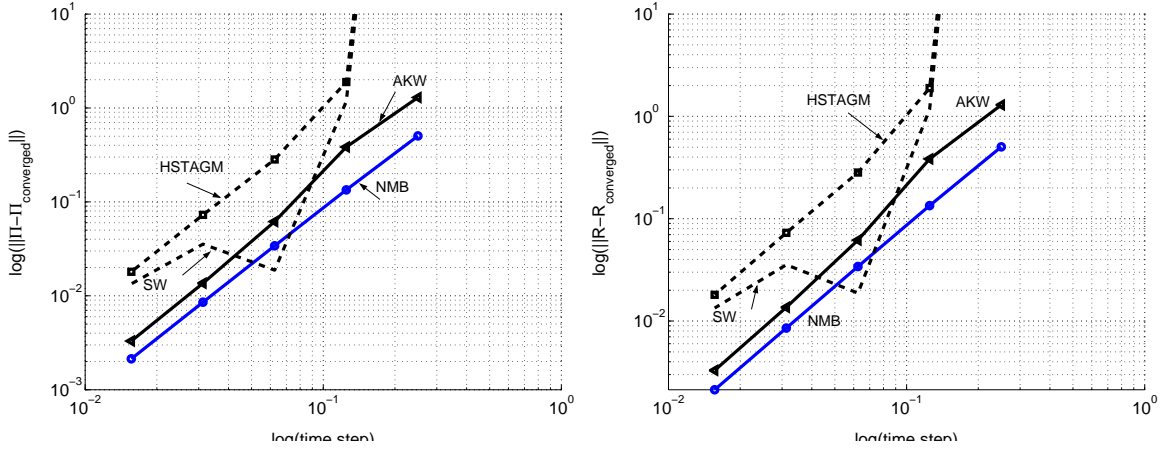


Figure 13: Body in Coulombic potential with soft wall contact; on the left hand side convergence in the norm of the error in body-frame angular momentum; on the right hand side convergence in the norm of the error in the attitude matrix: (\triangleleft): **AKW** (implicit midpoint rule of Austin, Krishnaprasad, Wang [1]); **SW** (dashed line, no symbol) solutions with the Simo and Wong [35] algorithm; **NMB** (\circ , solid line) Newmark algorithm; (\square): explicit staggered algorithm with midpoint torque evaluation **HSTAGM** [16].

Discussion

The explicit Newmark algorithms discussed here, namely **NMB**, and **NMS**, and also the **SW** algorithm, are time symmetric (for background see for example [14]). To show that for instance for the proposed Newmark algorithm, we need to swap subscripts n and $n - 1$ and change the sign of the time step. Thus, the rotation update

$$\mathbf{R}_n = \mathbf{R}_{n-1} \exp\left[\Delta t \tilde{\boldsymbol{\Omega}}_{n-1} + \frac{\Delta t^2}{2} \tilde{\mathbf{A}}_{n-1}\right] \quad (44)$$

will become after the swap

$$\mathbf{R}_{n-1} = \mathbf{R}_n \exp\left[-\Delta t \tilde{\boldsymbol{\Omega}}_n + \frac{\Delta t^2}{2} \tilde{\mathbf{A}}_n\right] \quad (45)$$

while the angular velocity update

$$\boldsymbol{\Omega}_n = \boldsymbol{\Omega}_{n-1} + \frac{\Delta t}{2}(\mathbf{A}_{n-1} + \mathbf{A}_n) \quad (46)$$

will read

$$\boldsymbol{\Omega}_{n-1} = \boldsymbol{\Omega}_n - \frac{\Delta t}{2}(\mathbf{A}_n + \mathbf{A}_{n-1}) \quad (47)$$

However, we may express

$$\boldsymbol{\Omega}_{n-1} + \frac{\Delta t}{2}\mathbf{A}_{n-1} = \boldsymbol{\Omega}_n - \frac{\Delta t}{2}\mathbf{A}_n$$

which upon substitution into (45) yields algorithm identical to equations (44), (46). As pointed out for some integrators [41], time symmetry often leads to good energy behavior. However, clearly that is not a sufficient condition since the Simo and Wong integrator displays a drift in energy for all time steps.

It would appear that the present Newmark is a symplectic integrator. A proof has not been completed yet, but the numerical evidence is very strong.

Conclusions

We have explored the issue of time stepping for the dynamics of rigid bodies in three-dimensional space with explicit time integrators. In particular, we show that the most naive and straightforward rewrite of the Verlet/Newmark algorithm for the rotation group inherits a collection of good behaviors essentially analogous to those of the vector-space version. On the other hand, the explicit momentum-conservation algorithm as proposed by Simo and Wong [35] that is sometimes referred to as an implementation of the explicit Newmark does not display the characteristics we would expect.

We propose that our algorithm is *the* explicit velocity-based Newmark for the rotation dynamics setting. It is a very good performer, robust and accurate. In ways that are likely not fully understood at this moment it manages to balance good behaviors in momentum and energy representation.

ACKNOWLEDGMENTS

This research was partially supported by a Hughes-Christensen research award. This support is gratefully acknowledged. The author wishes to thank Jerry Marsden for insightful comments and suggestions.

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] A.I. Bobenko, B. Lorbeer, and Yu. B. Suris. Integrable discretization's of the Euler top. *Journal of mathematical physics*, 39(12):6668–6683, 1998.
- [3] M. Borri, L. Trainelli, and C. L. Bottasso. On representations and parametrizations of motion. *Multibody system dynamics*, 4:129–193, 2000.
- [4] CJ Budd and A Iserles. Geometric integration: numerical solution of differential equations on manifolds. *PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY OF LONDON SERIES A-MATHEMATICAL PHYSICAL AND ENGINEERING SCIENCES*, 357(1754):945 – 956, 1999.
- [5] A. Cardona and M. Geradin. A beam the finite element nonlinear theory with finite rotations. *International Journal for Numerical Methods in Engineering*, 26:2403–2438, 1988.
- [6] E. Celledoni and B. Owren. Lie methods for rigid body dynamics and time integration on manifolds. *Computer Methods In Applied Mechanics And Engineering*, 192:421–438, 2003.
- [7] P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear Sci.*, 3:1–33, 1993.
- [8] A Dullweber, B Leimkuhler, and R McLachlan. Symplectic splitting methods for rigid body molecular dynamics. *JOURNAL OF CHEMICAL PHYSICS*, 107(15):5840–5851, 1997.
- [9] K. Eng and A. Marthinsen. A note on the numerical solution of the heavy top equations. Technical report, Department of Informatics, University of Bergen, 1999.
- [10] K. Engo and S. Faltinsen. Numerical integration of Lie-Poisson systems while preserving coadjoint orbits and energy. Reports in Informatics 179, University of Bergen, 1999.
- [11] K. Engo and A. Marthinsen. Application of geometric integration to some mechanical problems. *Multibody System Dynamics*, 2:71–88, 1998.
- [12] Christian Lubich Ernst Hairer and Gerhard Wanner. Geometric numerical integration illustrated by the Störmer-Verlet method. *Acta Numerica*, pages 399–450, 2003.
- [13] Francesco Fassò. Comparison of splitting algorithms for the rigid body. *Journal of computational physics*, 189:527–538, 2003.
- [14] 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.

- [15] Thomas Holder, Ben Leimkuhler, and Sebastian Reich. Explicit variable step size and time reversible integration. *Applied Numerical Mathematics*, 39:367–377, 2001.
- [16] G. Hulbert. Explicit momentum conserving algorithms for rigid body dynamics. *Computers and Structures*, 44(6):1291–1303, 1992.
- [17] A. Iserles, H.Z. Munthe-Kaas, S. P. Norsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.
- [18] C Kane, JE Marsden, and M Ortiz. Symplectic-energy-momentum preserving variational integrators. *JOURNAL OF MATHEMATICAL PHYSICS*, 40(7):3353–3371, 1999.
- [19] C Kane, JE Marsden, M Ortiz, and M West. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, 49(10):1295–1325, 2000.
- [20] B. Leimkuhler and S. Reich. Symplectic integraton of constrained Hamiltonian systems. *Math. Comp.*, 65:589–605, 1994.
- [21] D. Lewis and J. C. Simo. Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups. *J. Nonlinear Sci.*, 4:253–299, 1995.
- [22] S. Pekarsky Marsden, J. E. and . S. Shkoller. Discrete Euler-Poincaré and Lie-Poisson equations. *Nonlinearity*, 12:1647–1662, 1999.
- [23] A Marthinsen, H MuntheKaas, and B Owren. Simulation of ordinary differential equations on manifolds: Some numerical experiments and verifications. *MODELING IDENTIFICATION AND CONTROL*, 18(1):75–88, 1997.
- [24] R. I. McLachlan. Explicit Lie-Poisson integration and the Euler equations. *Physical Review Letters*, 71:3043–3046, 1993.
- [25] R. I. McLachlan and C. Scovel. Equivariant constrained symplectic integration. *J. Nonlinear Sci.*, 16:233–256, 1995.
- [26] 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.
- [27] J. Moser and A.P. Veselov. Discrete versions of some classical integrable systems and factorization of matrix polynomials. *Communications in mathematical physics*, 139:217 – 243, 1991.
- [28] H Munthe-Kaas. Runge-Kutta methods on Lie groups. *BIT*, 38(1):92–111, 1998.

- [29] Hans Munthe-Kaas. Lie–Butcher theory for Runge–Kutta methods. *BIT*, 35:572–587, 1995.
- [30] N. M. Newmark. A method of computation for structural dynamics. *Journal of engineering mechanics, ASCE*, pages 67–94, 1959.
- [31] Michael Anthony Puso. An energy and momentum conserving method for rigid-flexible body dynamics. *International Journal for numerical methods in engineering*, 53:1393–1414, 2002.
- [32] S. Reich. Symplectic integrators for systems of rigid bodies. *Physica D*, 76:375–383, 1994.
- [33] F. A. Rochinha and R. Sampaio. Nonlinear rigid body dynamics: Energy and momentum conserving algorithm. *Computational Methods In Engineering Sciences*, 1(2):7–18, 2000.
- [34] J. C. Simo and L. Vu-Quoc. On the dynamics in space of rods undergoing large motions – a geometrically exact approach. *Computer methods in applied mechanics and engineering*, 66:125–161, 1988.
- [35] 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.
- [36] Marcus Svanberg. An improved leap-frog rotational algorithm. *Molecular physics*, 92(6):1085–1088, 1997.
- [37] W. T. Thompson. *Introduction to space dynamics*. Dover publications, New York, 1986.
- [38] L. Verlet. Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review D*, pages 98–103, 1967.
- [39] J. M. Wendlandt and J. E. Marsden. Mechanical integrators derived from a discrete variational principle. *Physica D*, 106:223–246, 1997.
- [40] A. Zanna. Collocation and relaxed collocation for the Fer and Magnus expansions. *SIAM Journal of numerical analysis*, 36(4):1145–1182, 1999.
- [41] A. Zanna, K. Eng, and H. Munthe-Kaas. Adjoint and selfadjoint Lie-group methods. *BIT*, 41(2):395–421, 2001.
- [42] Antonella Zanna. The method of iterated commutators for ordinary differential equations on Lie groups. Technical Report 1996/NA12, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, England, 1996.

4 Appendix A

The explicit time integration algorithm of Simo and Wong [35], which had been designed to conserve angular momentum, is one of the best performing explicit algorithms published in the open literature. It is not a self starting algorithm, and following the recommendation of the above reference, the algorithm is started by calculating the initial acceleration from the balance equation.

Algorithm **SW**:

```

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

```

Hulbert has proposed an explicit algorithm that achieves momentum conservation [16]. It is somewhat reminiscent of the leap-frog version of the Newmark algorithm, but as noted by the author it cannot be exactly equivalent since the equations of motion produce an additional coupling between accelerations and velocities (this has been also discussed above).

Algorithm **HSTAGM**:

```

Given  $\mathbf{\Omega}_0, \mathbf{R}_0,$ 
 $\mathbf{A}_0 = \mathbf{I}^{-1}(\mathbf{t}_0 - \mathbf{\Omega}_0 \times \mathbf{I} \cdot \mathbf{\Omega}_0)$ 
 $\mathbf{\Theta}_{1/2} = \frac{\Delta t}{2} \mathbf{\Omega}_0 + \frac{\Delta t^2}{8} \mathbf{A}_0$ 
 $\mathbf{R}_{1/2} = \mathbf{R}_0 \exp[\tilde{\mathbf{\Theta}}_{1/2}]$ 
 $\mathbf{R}_{1/2} \mathbf{I} \mathbf{\Omega}_{1/2} = \mathbf{R}_0 \mathbf{I} \mathbf{\Omega}_0 + \frac{\Delta t}{2} \mathbf{t}_{1/4}$ 
for  $n = 1, 2, \dots$ 
   $\mathbf{R}_n = \mathbf{R}_{n-1} \exp[\Delta t \mathbf{\Omega}_{n-1/2}]$ 
   $\mathbf{R}_n \mathbf{I} \mathbf{\Omega}_n = \mathbf{R}_{n-1} \mathbf{I} \mathbf{\Omega}_{n-1} + \Delta t \mathbf{t}_{n-1/2}$ 
   $\mathbf{R}_{n+1/2} = \mathbf{R}_{n-1/2} \exp[\Delta t \mathbf{\Omega}_n]$ 
   $\mathbf{R}_{n+1/2} \mathbf{I} \mathbf{\Omega}_{n+1/2} = \mathbf{R}_{n-1/2} \mathbf{I} \mathbf{\Omega}_{n-1/2} + \Delta t \mathbf{t}_n$ 
end

```

Here the torques are evaluated as $\mathbf{t}_{1/4} = \mathbf{t}(t_{1/4}, \mathbf{R}_0 \exp[\frac{1}{2} \tilde{\mathbf{\Theta}}_{1/2}])$, $\mathbf{t}_{n-1/2} = \mathbf{t}(t_{n-1/2}, \mathbf{R}_{n-1/2})$, and $\mathbf{t}_n = \mathbf{t}(t_n, \mathbf{R}_n)$.

5 Appendix B

For reference we present the Matlab method `advance()` for an integrator class of the explicit Newmark in the body frame.


```

function retobj = advance(self,torque,dt,nsteps)
Omeگان1 = self.Omega;
Rn1 = self.R;
t=get(self.rotint,'t');
I=self.I;
% initial acceleration at t_n-1
Alphan1=I\r(Rn1'*feval(torque, t, Rn1) - skewmat(Omeگان1)*I*Omeگان1);
for i=1:nsteps
    Rn=Rn1*rotmat(skewmat(dt*Omeگان1+dt^2/2*Alphan1));
    Tn = Rn'*feval(torque, t+dt, Rn); % solve equation of motion at t_n
    Alphan = solve_Alpha(I, Tn, dt, Alphan1, Omeگان1, 100*eps);
    Omeگان = Omeگان1+dt/2*(Alphan1+Alphan); % update velocity
    % swap variables for next step
    Rn1=Rn;    Omeگان1=Omeگان;    Alphan1=Alphan;
    t=t+dt;
end
self.R = Rn;
self.Omega = Omeگان;
self.rotint = set(self.rotint,'t',t);
retobj=self;

```

For completeness, we also include the method that solves the nonlinear equations of motion, solve_Alpha.

```

function Alphan = solve_Alpha(I, Tn, dt, Alphan1, Omeگان1, ceps)
maxi=12;
Omeگانp=Omeگان1+dt/2*Alphan1;
B = I + dt/2*skewmat(Omeگانp)*I - skewmat(dt/2*I*(Omeگانp));
Alphan=Alphan1;
i=0;
res = I*Alphan - Tn + skewmat(Omeگانp+dt/2*Alphan)*I*(Omeگانp+dt/2*Alphan);
while (max(abs(res)) > ceps)
    Alphan = Alphan - (B+(dt/2)^2*(skewmat(Alphan)*I-skewmat(I*(Alphan))))\res;
    res = I*Alphan - Tn + skewmat(Omeگانp+dt/2*Alphan)*I*(Omeگانp+dt/2*Alphan);
    i=i+1;
    if (i > maxi)
        if (max(abs(res)) > ceps)
            warning(['Failed to converge: ||residual||=' num2str(norm(res))]);
        end
        break;
    end
end
end

```