On the Use of Lie Group Time Integrators in Multibody Dynamics

Olivier Brüls[1] and Alberto Cardona[2]

[1]Department of Aerospace and Mechanical Engineering (LTAS), University of Liège
Chemin des Chevreuils 1, B52/3, 4000 Liège, Belgium

[2]CIMEC - INTEC, Universidad Nacional Litoral - Conicet
Güemes 3450, 3000 Santa Fe, Argentina

Abstract

This paper proposes a family of Lie group time integrators for the simulation of flexible multibody systems. The method provides an elegant solution to the rotation parameterization problem. As an extension of the classical generalized-α method for dynamic systems, it can deal with constrained equations of motion. Second-order accuracy is demonstrated in the unconstrained case. The performance is illustrated on several critical benchmarks of rigid body systems with high rotation speeds and second order accuracy is evidenced in all of them, even for constrained cases. The remarkable simplicity of the new algorithms opens some interesting perspectives for real-time applications, model-based control and optimization of multibody systems.

1 Introduction

The numerical simulation of articulated systems including rigid bodies, nonlinear force elements (e.g. springs and dampers) as well as flexible components requires advanced modelling strategies, see Wasfy and Noor [1]. The absolute coordinate method appears to be well-suited for the integrated modelling of complex flexible multibody systems, see Géradin and Cardona [2]. Still, an inherent difficulty of this approach comes from the need to treat large rotation variables in a consistent way. This paper considers some adaptation of Lie group time integrators, which have been initially developed in applied mathematics by Crouch and Grossmann [3] and Munthe-Kaas [4, 5], in order to provide a more natural answer to the rotation parameterization problem in multibody dynamics.

The Lie group nature of rotational fields already played a major role in the development of geometrically consistent models for mechanical systems with large rotations, see Simo and co-authors [6, 7] and Cardona and Géradin [8, 9]. In those contributions, dynamic problems were solved using integration algorithms based on the Newmark scheme [10] and on the HHT scheme by Hilber et al [11]. The generalized-α method described by Chung and Hulbert [12], which will be exploited in the present paper, is a further extension of the Newmark algorithm and it includes as special cases most popular algorithms in structural dynamics. An optimal design of the generalized-α method allows to combine second-order accuracy and the ability to filter spurious high frequency modes arising in finite element models. This method is also able to solve dynamic problems with kinematic constraints, as shown by Cardona and Géradin [9] and Arnold and Brüls [13]. Lunk and Simeon [14], Jay and Negrut [15] and Arnold [16] combined the

generalized-α method with an index reduction technique in order to solve systems of differential-algebraic equations.

Crouch and Grossmann [3] and Munthe-Kaas [4, 5] addressed some generalizations of classical Runge-Kutta and multistep time integration schemes to solve differential equations on Lie groups. From a mathematical viewpoint, a Lie group G is a differentiable manifold for which the product (or composition) and inversion operations are smooth maps. A differential equation on a Lie group G is then a differential equation whose solution remains on G for any initial condition on G. For instance, the rotational dynamics of a rigid body can be described as a differential equation on $SO(3)$, the group of proper orthogonal linear transformations. Some applications of Lie group time integrators in rigid body dynamics have been considered by Celledoni and Owren [17] and by Bottasso and Borri [18]. It is remarkable that Lie group time integrators do not require *a priori* the definition of local generalized coordinates. In other words, a Lie group integrator includes in its algorithmic structure an intrinsic and consistent strategy for the parameterization of the configuration manifold.

The present work concerns the numerical simulation of flexible multibody systems modelled according to the absolute coordinate method in a finite element context, see Géradin and Cardona [2]. The motion being described using nodal translation and rotation variables, the system evolves on the Lie group defined by a multiple cartesian product of $\mathbb{R}^3$ and $SO(3)$. In this formulation, the interconnections between the various bodies of the multibody system are modelled using nonlinear kinematic constraints. As a consequence, the motion of the system is restricted to a submanifold of the Lie group. In other words, the equations of motion have the structure of a *differential equation on a Lie group with algebraic constraints*.

We propose a new family of Lie group time integrators, which tries to inherit the favorable accuracy and stability properties of the generalized-α method and which can be used for the simulation of complex multibody systems. This rather broad family includes as special cases the classical generalized-α method for the analysis of dynamic systems on a linear space by Chung and Hulbert [12], the algorithm by Simo and Vu-Quoc [6], as well as the geometric algorithm studied by Brüls and Eberhard [19].

Recent research efforts lead to the development of structure-preserving time-integration schemes such as variational integrators or energy-preserving schemes, see e.g. [7, 18, 20, 21, 22, 23, 24, 25, 26, 27]. The design of those algorithms usually requires a deep investigation of the internal structure of the dynamic system. In contrast, the algorithms studied in this paper are not designed to preserve exactly energy or other first integrals; however, since they *only* exploit the Lie group structure of the problem, they can be implemented in a more generic way. In the work by Betsch and Steinmann [22], specific redundant coordinates are used in order to overcome the rotation parameterization problem. As opposed to this strategy which tends to increase the number of coordinates and kinematic constraints, the parameterization naturally induced by our Lie group formulation is based on a minimal number of rotation parameters.

The resulting simplicity of the proposed algorithms is attractive from several points of view: possible implementation in an existing industrial simulation code, easier code maintenance and improved efficiency with perspectives for real-time applications, optimization and model based-control. In addition, a simplified code may render possible the implementation of an exact linearization for a semi-analytical sensitivity analysis which would otherwise be cumbersome, see [19].

Using several critical benchmarks with large rotation speeds and kinematic constraints, this paper shows that the new and simpler Lie group methods can compete with the classical "linear" generalized-α scheme from the viewpoint of accuracy, stability and conservation of energy. Hence, these new methods are promising candidates for the development of robust, efficient and open simulation software for flexible multibody systems.

The paper is organized as follows. In Section 2, the equations of motion of a flexible multibody system are described according to the Lie group formalism. The new family of

time integrators is presented in Section 3 and some implementation issues are discussed in Section 4. Section 5 presents some examples of dynamic systems with large rotations, which can be simulated using the proposed approach. Second-order accuracy of the algorithms is demonstrated in the unconstrained case in Section 6. Some numerical results for dynamic systems with large rotation speeds are reported in Section 7. Three examples are presented with comparisons to analytical solutions given by Romano [28, 29]. Two more examples are solved and compared to reference solutions computed numerically with small time steps. Finally, in Section 8, conclusions of the study are drawn.

2 Equations of motion

We consider a flexible multibody system whose dynamics evolves on a n-dimensional manifold G with a Lie group structure (see Boothby [30] for mathematical details about Lie groups). In an absolute coordinate formulation, an element $q \in G$ is composed of several subsets of absolute nodal translations and rotations, *a priori* considered as independent variables. The composition operation $G \times G \to G$ is written as

$$q_{tot} = q_1 \circ q_2 \tag{1}$$

with $q_1, q_2, q_{tot} \in G$ and the identity element e is such that $q \circ e = e \circ q = q,\ \forall q \in G$. T_qG denotes the tangent space at a point $q \in G$ and the Lie algebra is defined as the tangent space at the identity $\mathfrak{g} = T_eG$. The Lie algebra is a vector space, which is isomorphic to $\mathbb{R}^n$ by an invertible linear mapping

$$\widetilde{(\bullet)} :\ \mathbb{R}^n \to \mathfrak{g},\ \mathbf{v} \mapsto \widetilde{\mathbf{v}}. \tag{2}$$

A tangent vector at any point q can be represented in the Lie algebra using the left translation map L_q. Indeed, L_q is a diffeomorphism of G

$$L_q :\ G \to G,\ y \mapsto q \circ y \tag{3}$$

and its derivative defines a diffeomorphism between T_yG and $T_{q\circ y}G$. In the particular case $y = e$, we thus have a bijection between $T_eG = \mathfrak{g}$ and T_qG :

$$DL_q(e) :\ \mathfrak{g} \to T_qG,\ \widetilde{\mathbf{w}} \mapsto DL_q(e) \cdot \widetilde{\mathbf{w}} \tag{4}$$

where $DL_q(e) \cdot \widetilde{\mathbf{w}}$ is the directional derivative of L_q evaluated at point e in the direction $\widetilde{\mathbf{w}} \in \mathfrak{g}$. Hence, a tangent vector $\widetilde{\mathbf{w}} \in \mathfrak{g}$ defines a left invariant vector field on G which is constructed by left translation of $\widetilde{\mathbf{w}}$ to the tangent space at any point of G.

In a multibody system, the nodal translation and rotation variables are generally not independent but they have to satisfy a set of m kinematic constraints $\mathbf{\Phi} : G \to \mathbb{R}^m$, which restrict the dynamics to the submanifold N of dimension $n - m$

$$N = \{q \in G : \mathbf{\Phi}(q) = \mathbf{0}\} \tag{5}$$

Using classical principles of mechanics [2], the equations of motion of a flexible multibody system have the following index-3 differential-algebraic structure

$$\dot{q} = DL_q(e) \cdot \widetilde{\mathbf{v}} \tag{6}$$

$$\mathbf{M}(q)\dot{\mathbf{v}} + \mathbf{g}(q, \mathbf{v}, t) + \mathbf{B}^T(q)\boldsymbol{\lambda} = \mathbf{0} \tag{7}$$

$$\mathbf{\Phi}(q) = \mathbf{0} \tag{8}$$

where $q \in G$ represents the configuration of the system, $\mathbf{v} \in \mathbb{R}^n$ is the velocity vector and $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the vector of Lagrange multipliers associated with the constraints $\mathbf{\Phi}$. $\mathbf{M}$ is the $n \times n$

symmetric mass matrix, $\mathbf{g}$ is the vector of external, internal and complementary inertia forces and $\mathbf{B}$ is the $m \times n$ matrix of constraint gradients such that

$$D\mathbf{\Phi}(q) \cdot \widetilde{\mathbf{w}} = \mathbf{B}(q)\mathbf{w}, \ \forall \mathbf{w} \in \mathbb{R}^n. \tag{9}$$

In the above equation, $D\mathbf{\Phi}(q) \cdot \widetilde{\mathbf{w}}$ is the directional derivative of $\mathbf{\Phi}$ evaluated at point q in the direction of the tangent vector defined by left translation of $\widetilde{\mathbf{w}}$.

The equations of motion (6-8) allow to represent the dynamics of a general class of flexible multibody systems, e.g. using the finite element approach described in [2].

3 Formulation of Lie group time integrators

Inspired by the index-3 formulation of the generalized-α time integration scheme for classical systems of differential-algebraic equations [13] as well as by the work of Crouch and Grossmann [3] and Munthe-Kaas [4, 5], we propose a family of Lie group time integrators based on the following discretized set of equations:

$$\begin{aligned}
\mathbf{M}(q_{n+1})\dot{\mathbf{v}}_{n+1} + \mathbf{g}(q_{n+1}, \mathbf{v}_{n+1}, t_{n+1}) + \mathbf{B}^T(q_{n+1})\boldsymbol{\lambda}_{n+1} &= \mathbf{0} && (10)\\
\mathbf{\Phi}(q_{n+1}) &= \mathbf{0} && (11)\\
q_{n+1} &= \varphi_h(q_n, \mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) && (12)\\
\mathbf{v}_{n+1} &= \mathbf{v}_n + (1-\gamma)h\mathbf{a}_n + \gamma h\mathbf{a}_{n+1} && (13)\\
(1-\alpha_m)\mathbf{a}_{n+1} + \alpha_m\mathbf{a}_n &= (1-\alpha_f)\dot{\mathbf{v}}_{n+1} + \alpha_f\dot{\mathbf{v}}_n && (14)
\end{aligned}$$

$h = t_{n+1} - t_n$ is the time step size and the particular form of equation (12) makes this formulation applicable to dynamic systems on Lie groups. Indeed, $\varphi_h : G \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to G$ is a discrete time mapping which may be defined in several ways, e.g.

$$\begin{aligned}
\varphi_h^{(1)}(q_n, \mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) &= q_n \circ \exp\left(h\widetilde{\mathbf{v}_n} + h^2(0.5-\beta)\widetilde{\mathbf{a}_n} + \beta h^2\widetilde{\mathbf{a}_{n+1}}\right) && (15)\\
\varphi_h^{(2)}(q_n, \mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) &= q_n \circ \exp(h\widetilde{\mathbf{v}_n}) \circ \exp\left(h^2(0.5-\beta)\widetilde{\mathbf{a}_n} + \beta h^2\widetilde{\mathbf{a}_{n+1}}\right) && (16)\\
\varphi_h^{(3)}(q_n, \mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) &= q_n \circ \exp(h\widetilde{\mathbf{v}_n}) \circ \exp(h^2(0.5-\beta)\widetilde{\mathbf{a}_n}) \circ \exp\left(\beta h^2\widetilde{\mathbf{a}_{n+1}}\right) && (17)
\end{aligned}$$

where $\exp : \mathfrak{g} \to G$ is the exponential operator of the group [30]. The complete algorithm thus involves four parameters β, γ, α_m and α_f which should be selected as usual in order to obtain suitable convergence and stability properties.

Given some initial values $q(0)$ and $\mathbf{v}(0)$ at time $t_0 = 0$ and assuming that the kinematic constraints are satisfied at position and velocity level

$$\mathbf{\Phi}(q_0) = \mathbf{0}, \qquad \dot{\mathbf{\Phi}}(q_0, \mathbf{v}_0) = \mathbf{0} \tag{18}$$

the step-by-step recursive procedure is initialized by solving the equations:

$$\begin{aligned}
q_0 &= q(0) && (19)\\
\mathbf{v}_0 &= \mathbf{v}(0) && (20)\\
\mathbf{M}(q_0)\dot{\mathbf{v}}_0 + \mathbf{g}(q_0, \mathbf{v}_0, 0) + \mathbf{B}^T(q_0)\boldsymbol{\lambda}_0 &= \mathbf{0} && (21)\\
\ddot{\mathbf{\Phi}}(q_0, \mathbf{v}_0, \dot{\mathbf{v}}_0) &= \mathbf{0} && (22)\\
\mathbf{a}_0 &= \dot{\mathbf{v}}_0 && (23)
\end{aligned}$$

where Equation (22) represents the second time derivative of the kinematic constraints.

Remarks concerning the proposed integration scheme

1. The equations of motion are enforced exactly at time $n+1$ and there is no weighted combination of forces between time n and time $n+1$. This is especially important to guarantee the consistency of the algorithm when the mass matrix is not constant, see [13].

2. The variable $\mathbf{a}_{n+1}$ is an acceleration-like variable which is different from the true acceleration $\dot{\mathbf{v}}_{n+1}$ at time t_{n+1}.

3. $\boldsymbol{\lambda}_n$ is not involved in the computation of step $n+1$, i.e. we may say that the Lagrange multipliers have "no memory".

4. The derivative $\dot{q} \in T_q G$ is never explicitly evaluated in the numerical procedure. The algorithm only relies on tangent vectors in the Lie algebra $\mathfrak{g}$. Linear combinations of tangent vectors in the Lie algebra make perfect sense.

5. Nonlinearities are present not only in the equations of motion, but also in the integration formulae.

6. An analytical expression of the exponential map can be used as will be discussed later on in this paper.

7. Other variants of the algorithm may be obtained if the exponential map is replaced by another $\mathfrak{g} \to G$ mapping operator, e.g. the Cayley transform.

8. The proposed algorithm includes the classical generalized-α algorithm as a special case. Indeed, the vector space $\mathbb{R}^n$ is a particular case of a Lie group with the following composition and exponential operations:

$$\mathbf{q}_1 \circ \mathbf{q}_2 = \mathbf{q}_1 + \mathbf{q}_2 \tag{24}$$
$$\exp(\mathbf{q}) = \mathbf{q} \tag{25}$$

for $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q} \in \mathbb{R}^n$. The identity element is the null vector $\mathbf{0}$. In a vector space, the configuration space can be identified with the tangent space at any point and Equation (6) becomes simply

$$\dot{\mathbf{q}} = \mathbf{v}. \tag{26}$$

Consequently, the proposed algorithm degenerates into the classical generalized-α algorithm described in [13]. All integration formulae are then linear and the three variants defined by Equations (15-17) cannot be distinguished anymore.

4 Implementation

This section presents a Newton-Raphson algorithm which solves Equations (10-14) for all variables at time step $n+1$, starting from given values q_n, $\mathbf{v}_n$, $\dot{\mathbf{v}}_n$, $\mathbf{a}_n$. The linearization of the set of nonlinear equations (10-14) is described below.

First, the linearization of Equations (10,11,13,14) is achieved without difficulty. Let us define the residual vector $\mathbf{r}$

$$\mathbf{r}(q, \mathbf{v}, \boldsymbol{\lambda}, \dot{\mathbf{v}}, t) = \mathbf{M}(q)\dot{\mathbf{v}} + \mathbf{g}(q, \mathbf{v}, t) + \mathbf{B}^T(q)\boldsymbol{\lambda} \tag{27}$$

and the tangent stiffness and damping matrices $\mathbf{K}_t$ and $\mathbf{C}_t$ such that

$$D_1\mathbf{r}(q, \mathbf{v}, \boldsymbol{\lambda}, \dot{\mathbf{v}}, t) \cdot \widetilde{\Delta\mathbf{q}} = \mathbf{K}_t \Delta\mathbf{q} \tag{28}$$
$$D_2\mathbf{r}(q, \mathbf{v}, \boldsymbol{\lambda}, \dot{\mathbf{v}}, t) \cdot \Delta\mathbf{v} = D_2\mathbf{g}(q, \mathbf{v}, t) \cdot \Delta\mathbf{v} = \mathbf{C}_t \Delta\mathbf{v} \tag{29}$$

where D_1 (respectively, D_2) indicates the directional derivative with respect to the first argument $q \in G$ (respectively, to the second argument $\mathbf{v} \in \mathbb{R}^n$). The linearization of Equations (10,11,13,14) follows immediately:

$$\begin{aligned}
\Delta\mathbf{r} &= \mathbf{M}\Delta\dot{\mathbf{v}}_{n+1} + \mathbf{C}_t\Delta\mathbf{v}_{n+1} + \mathbf{K}_t\Delta\mathbf{q}_{n+1} + \mathbf{B}^T\Delta\boldsymbol{\lambda}_{n+1} && (30)\\
\Delta\boldsymbol{\Phi} &= \mathbf{B}\Delta\mathbf{q}_{n+1} && (31)\\
\Delta\mathbf{v}_{n+1} &= \gamma h\Delta\mathbf{a}_{n+1} && (32)\\
(1-\alpha_m)\Delta\mathbf{a}_{n+1} &= (1-\alpha_f)\Delta\dot{\mathbf{v}}_{n+1}. && (33)
\end{aligned}$$

Second, the linearization of the mapping φ_h in Equation (12) requires a more detailed study. We rewrite Equations (15-17) as

$$\varphi_h^{(a)}(q_n, \mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) = \varphi_{h*}^{(a)}(q_n, \mathbf{v}_n, \mathbf{a}_n) \circ \exp\left(\widetilde{\varphi_{hx}^{(a)}}(\mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1})\right), \qquad a = 1, 2 \text{ or } 3 \quad (34)$$

with the mappings $\varphi_{h*}^{(a)}$ and $\varphi_{hx}^{(a)}$ defined consistently as

$$\begin{aligned}
\varphi_{h*}^{(1)}(q_n, \mathbf{v}_n, \mathbf{a}_n) &= q_n && (35)\\
\varphi_{hx}^{(1)}(\mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) &= h\mathbf{v}_n + h^2(0.5-\beta)\mathbf{a}_n + \beta h^2\mathbf{a}_{n+1} && (36)\\
\varphi_{h*}^{(2)}(q_n, \mathbf{v}_n, \mathbf{a}_n) &= q_n \circ \exp(h\widetilde{\mathbf{v}_n}) && (37)\\
\varphi_{hx}^{(2)}(\mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) &= h^2(0.5-\beta)\mathbf{a}_n + \beta h^2\mathbf{a}_{n+1} && (38)\\
\varphi_{h*}^{(3)}(q_n, \mathbf{v}_n, \mathbf{a}_n) &= q_n \circ \exp(h\widetilde{\mathbf{v}_n}) \circ \exp(h^2(0.5-\beta)\widetilde{\mathbf{a}_n}) && (39)\\
\varphi_{hx}^{(3)}(\mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) &= \beta h^2\mathbf{a}_{n+1}. && (40)
\end{aligned}$$

Using those mappings, q_{n+1} can be evaluated by four successive operations

$$\begin{aligned}
q^* &:= \varphi_{h*}^{(a)}(q_n, \mathbf{v}_n, \mathbf{a}_n) && (41)\\
\mathbf{x} &:= \varphi_{hx}^{(a)}(\mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1}) && (42)\\
q_{inc} &:= \exp(\widetilde{\mathbf{x}}) && (43)\\
q_{n+1} &:= q^* \circ q_{inc} && (44)
\end{aligned}$$

with the intermediate variables $q^*, q_{inc} \in G$ and $\mathbf{x} \in \mathbb{R}^n$. One observes that q^* is computed explicitly from the known variables at time step n whereas the increment $\mathbf{x}$ implicitly depends on $\mathbf{a}_{n+1}$. One also notes that $\varphi_{hx}^{(a)}$ only involves linear operations. The linearization of the first two relations (41) and (42) is straightforward considering the special form of the operator $\varphi_{h*}^{(a)}$ and $\varphi_{hx}^{(a)}$ in Equations (35-40)

$$\begin{aligned}
DL_{q^*}(e) \cdot \widetilde{\Delta\mathbf{q}^*} &= 0 && (45)\\
\Delta\mathbf{x} &= \beta h^2\Delta\mathbf{a}_{n+1} && (46)
\end{aligned}$$

and Equation (45) yields $\Delta\mathbf{q}^* = \mathbf{0}$. The linearization of Equation (43)

$$DL_{q_{inc}}(e) \cdot \widetilde{\Delta\mathbf{q}_{inc}} = D\exp(\widetilde{\mathbf{x}}) \cdot \Delta\widetilde{\mathbf{x}} \quad (47)$$

defines a linear relation between the vectors $\Delta\mathbf{x}$ and $\Delta\mathbf{q}_{inc} \in \mathbb{R}^n$ which is written in compact form as

$$\Delta\mathbf{q}_{inc} = \mathbf{T}(\mathbf{x})\Delta\mathbf{x} \quad (48)$$

where $\mathbf{T}(\mathbf{x})$ is the $n \times n$ tangent operator of the exponential map [5]. Finally, developing the linearized form of Equation (44)

$$DL_{q_{n+1}}(e) \cdot \widetilde{\Delta \mathbf{q}_{n+1}} = DL_{q^*}(q_{inc}) \cdot \left(DL_{q_{inc}}(e) \cdot \widetilde{\Delta \mathbf{q}_{inc}} \right), \tag{49}$$

and observing that its right-hand-side is equal to $DL_{q^* \circ q_{inc}}(e) \cdot \widetilde{\Delta \mathbf{q}_{inc}}$, we obtain

$$\Delta \mathbf{q}_{n+1} = \Delta \mathbf{q}_{inc}. \tag{50}$$

If the variables $\Delta \mathbf{q}_{n+1}$, $\Delta \mathbf{v}_{n+1}$, $\Delta \dot{\mathbf{v}}_{n+1}$, $\Delta \mathbf{a}_{n+1}$ and $\Delta \mathbf{q}_{inc}$ are eliminated from the linearized Equations (30-33,46,48,50), we obtain the linearized form

$$\begin{bmatrix} \Delta \mathbf{r} \\ \Delta \mathbf{\Phi} \end{bmatrix} = \mathbf{S}_t \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} \tag{51}$$

with the $(n+m) \times (n+m)$ tangent matrix

$$\mathbf{S}_t(q, \mathbf{x}, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\lambda}, t) = \begin{bmatrix} \mathbf{M}(q)\beta' + \mathbf{C}_t(q, \mathbf{v}, t)\gamma' + \mathbf{K}_t(q, \mathbf{v}, \dot{\mathbf{v}}, \boldsymbol{\lambda}, t)\mathbf{T}(\mathbf{x}) & \mathbf{B}^T(q) \\ \mathbf{B}(q)\mathbf{T}(\mathbf{x}) & \mathbf{0} \end{bmatrix} \tag{52}$$

and the algorithmic parameters

$$\beta' = \frac{1 - \alpha_m}{\beta h^2 (1 - \alpha_f)}, \qquad \gamma' = \frac{\gamma}{\beta h}. \tag{53}$$

The following algorithm ($a = 1$, 2 or 3) computes all system variables at time step $n+1$ based on their value at time step n:

function solveTimeStep($q_n, \mathbf{v}_n, \dot{\mathbf{v}}_n, \mathbf{a}_n$)
  $\dot{\mathbf{v}}_{n+1} := \mathbf{0}$
  $\boldsymbol{\lambda}_{n+1} := \mathbf{0}$
  $\mathbf{a}_{n+1} := (\alpha_f \dot{\mathbf{v}}_n - \alpha_m \mathbf{a}_n)/(1 - \alpha_m)$
  $\mathbf{v}_{n+1} := \mathbf{v}_n + h(1-\gamma)\mathbf{a}_n + \gamma h \mathbf{a}_{n+1}$
  $q^* := \varphi_{h*}^{(a)}(q_n, \mathbf{v}_n, \mathbf{a}_n)$
  $\mathbf{x} := \varphi_{hx}^{(a)}(\mathbf{v}_n, \mathbf{a}_n, \mathbf{a}_{n+1})$
  for $i = 1$ to i_{max}
    $q_{n+1} := q^* \circ \exp(\widetilde{\mathbf{x}})$
    $\mathbf{res} := \begin{bmatrix} \mathbf{r}(q_{n+1}, \mathbf{v}_{n+1}, \boldsymbol{\lambda}_{n+1}, \dot{\mathbf{v}}_{n+1}, t_{n+1}) \\ \mathbf{\Phi}(q_{n+1}) \end{bmatrix}$
    if $\|\mathbf{res}\| < \text{tol}$
      break
    end
    $\mathbf{S}_t := \mathbf{S}_t(q_{n+1}, \mathbf{x}, \mathbf{v}_{n+1}, \dot{\mathbf{v}}_{n+1}, \boldsymbol{\lambda}_{n+1}, t_{n+1})$
    $\begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} := -\mathbf{S}_t^{-1} \mathbf{res}$
    $\mathbf{x} := \mathbf{x} + \Delta \mathbf{x}$
    $\mathbf{v}_{n+1} := \mathbf{v}_{n+1} + \gamma' \Delta \mathbf{x}$
    $\dot{\mathbf{v}}_{n+1} := \dot{\mathbf{v}}_{n+1} + \beta' \Delta \mathbf{x}$
    $\boldsymbol{\lambda}_{n+1} := \boldsymbol{\lambda}_{n+1} + \Delta \boldsymbol{\lambda}$
  end
  $\mathbf{a}_{n+1} := \mathbf{a}_{n+1} + (1 - \alpha_f)/(1 - \alpha_m) \dot{\mathbf{v}}_{n+1}$
return $q_{n+1}, \mathbf{v}_{n+1}, \dot{\mathbf{v}}_{n+1}, \boldsymbol{\lambda}_{n+1}, \mathbf{a}_{n+1}$

5 Examples

5.1 Example 1 : Motion of a single rotating body

The motion of a rotating body fixed to the ground by a spherical joint is represented by the 3×3 rotation matrix $\mathbf{R}$, which belongs to the group of proper orthogonal linear transformations $SO(3)$. The composition operation is the matrix product

$$\mathbf{R}_1 \circ \mathbf{R}_2 = \mathbf{R}_1\mathbf{R}_2 \tag{54}$$

and the identity element is the 3×3 identity matrix $\mathbf{I}_3$. At any point $\mathbf{R}$, the tangent space is noted $T_{\mathbf{R}}SO(3)$ and the Lie algebra $\mathfrak{so}(3) = T_{\mathbf{I}_3}SO(3)$ is the set of skew-symmetric matrices:

$$\mathfrak{so}(3) = \{\widetilde{\mathbf{\Omega}} :\ \widetilde{\mathbf{\Omega}} + \widetilde{\mathbf{\Omega}}^T = \mathbf{0}\} \tag{55}$$

The Lie algebra can be identified to $\mathbb{R}^3$ since any matrix $\widetilde{\mathbf{\Omega}} \in \mathfrak{so}(3)$

$$\widetilde{\mathbf{\Omega}} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix} \tag{56}$$

can be represented by the 3×1 axial vector $\mathbf{\Omega} = [\Omega_1\ \Omega_2\ \Omega_3]^T$. The tangent space $T_{\mathbf{R}}SO(3)$ is thus isomorphic to $\mathbb{R}^3$ and we have

$$\dot{\mathbf{R}} = DL_{\mathbf{R}}(\mathbf{I}_3) \cdot \widetilde{\mathbf{\Omega}} = \mathbf{R}\widetilde{\mathbf{\Omega}}. \tag{57}$$

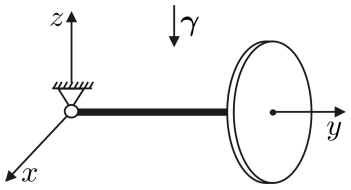

Figure 1: Heavy top.

The equations of motion of the rotating rigid body are written as

$$\mathbf{J}\dot{\mathbf{\Omega}} + \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} - \mathbf{C}(\mathbf{R}, t) = \mathbf{0} \tag{58}$$

where $\mathbf{J}$ is the 3×3 symmetric inertia tensor around the fixed point, $\mathbf{\Omega} \in \mathbb{R}^3$ is the material angular velocity and $\mathbf{C}$ is the applied torque in body coordinates. In the particular case of a heavy top (see Fig. 1), the gravity torque is computed as

$$\mathbf{C}(\mathbf{R}) = \widetilde{\mathbf{X}}\mathbf{R}^T m\boldsymbol{\gamma} \tag{59}$$

where $\boldsymbol{\gamma}$ is the 3×1 vector of gravity acceleration, m is the mass of the top and $\mathbf{X}$ is the 3×1 material position vector of the center of gravity with respect to the fixed point.

Clearly, Equations (57) and (58) are special cases of Equations (6) and (7) with

$$\mathbf{M} = \mathbf{J} \tag{60}$$

$$\mathbf{g}(\mathbf{R}, \mathbf{\Omega}, t) = \mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} - \mathbf{C}(\mathbf{R}, t) \tag{61}$$

$$\mathbf{C}_t(\mathbf{\Omega}) = \widetilde{\mathbf{\Omega}}\mathbf{J} - \widetilde{\mathbf{J}\mathbf{\Omega}} \tag{62}$$

$$\mathbf{K}_t \cdot \delta\mathbf{\Theta} = -D\mathbf{C}(\mathbf{R}, t) \cdot \delta\widetilde{\mathbf{\Theta}} \tag{63}$$

In the heavy top case, the last equation becomes

$$\mathbf{K}_t = -\widetilde{\mathbf{X}}\widetilde{\mathbf{R}^T m\boldsymbol{\gamma}} \tag{64}$$

For $SO(3)$, the exponential operator can be computed using the analytic Rodrigues formula:

$$\exp(\widetilde{\boldsymbol{\psi}}) = \mathbf{I}_3 + \frac{\sin\phi}{\phi}\widetilde{\boldsymbol{\psi}} + \frac{1-\cos\phi}{\phi^2}\widetilde{\boldsymbol{\psi}}\widetilde{\boldsymbol{\psi}} \tag{65}$$

where $\phi = \|\boldsymbol{\psi}\|$ and the vector $\boldsymbol{\psi} \in \mathbb{R}^3$ is the so-called Cartesian rotation vector. The exponential map admits the series expansion

$$\exp(\widetilde{\boldsymbol{\psi}}) = \mathbf{I}_3 + \frac{\widetilde{\boldsymbol{\psi}}}{1!} + \frac{\widetilde{\boldsymbol{\psi}}^2}{2!} + \dots \tag{66}$$

and the tangent operator is given by the formula

$$\mathbf{T}(\boldsymbol{\psi}) = \mathbf{I}_3 + \frac{\cos\phi - 1}{\phi^2}\widetilde{\boldsymbol{\psi}} + \left(1 - \frac{\sin\phi}{\phi}\right)\frac{\widetilde{\boldsymbol{\psi}}\widetilde{\boldsymbol{\psi}}}{\phi^2}. \tag{67}$$

Hence, the proposed Lie group time integration algorithm can be used to solve the equations of motion (58). In the case $\alpha_m = \alpha_f = 0$, the integration algorithm with the discrete operator $\varphi_h^{(1)}$ is equivalent to the algorithm proposed by Simo and Vu-Quoc [6].

5.2 Parameterization-based time integration algorithm

In the following numerical investigations, the performance of the new Lie group time integrators will be compared with a more classical integration algorithm based on an incremental parameterization of the equations of motion. This algorithm, presented in [9], is summarized below.

Even though it is not possible to find a global parameterization of $SO(3)$ with a set of 3 coordinates, it is possible to use a local coordinate chart for limited rotational amplitudes around a fixed reference rotation $\mathbf{R}_{ref}$. If we consider the natural parameterization of $SO(3)$ using the Cartesian rotation vector $\boldsymbol{\psi} \in \mathbb{R}^3$, the current rotation at time t is computed from $\boldsymbol{\psi}(t)$ using

$$\mathbf{R}(t) = \mathbf{R}_{ref}\exp(\widetilde{\boldsymbol{\psi}}). \tag{68}$$

At velocity level, we have the relation

$$\boldsymbol{\Omega}(t) = \mathbf{T}(\boldsymbol{\psi})\dot{\boldsymbol{\psi}} \tag{69}$$

where the tangent operator $\mathbf{T}(\boldsymbol{\psi})$ was already defined in Equation (67). Due to the existence of singularities, the validity range of this parameterization is limited to $\phi \in]-2\pi, 2\pi[$.

Introducing Equation (69) in the equation of motion (58) leads to the parameterized form of the equations of motion:

$$\mathbf{M}(\boldsymbol{\psi})\ddot{\boldsymbol{\psi}} + \mathbf{g}(\boldsymbol{\psi}, \dot{\boldsymbol{\psi}}, t) = \mathbf{0} \tag{70}$$

with

$$\mathbf{M}(\boldsymbol{\psi}) = \mathbf{T}^T(\boldsymbol{\psi})\mathbf{J}\mathbf{T}(\boldsymbol{\psi}) \tag{71}$$

$$\mathbf{g}(\boldsymbol{\psi}, \dot{\boldsymbol{\psi}}, t) = \mathbf{T}^T(\boldsymbol{\psi})\left(\mathbf{J}\dot{\mathbf{T}}(\boldsymbol{\psi}, \dot{\boldsymbol{\psi}})\dot{\boldsymbol{\psi}} + (\mathbf{T}(\boldsymbol{\psi})\dot{\boldsymbol{\psi}}) \times (\mathbf{J}\mathbf{T}(\boldsymbol{\psi})\dot{\boldsymbol{\psi}}) - \mathbf{C}(\mathbf{R}_{ref}\exp(\widetilde{\boldsymbol{\psi}}), t)\right). \tag{72}$$

The dynamics of the system is thus described in the vector space $\mathbb{R}^3$ and it can be solved using the classical generalized-α method (we recall that it is a special case of the Lie group method

described in this paper). The detailed expressions of the tangent matrices $\mathbf{C}_t$ and $\mathbf{K}_t$ can be found in [2], but they are not reproduced here for reasons of space.

The coordinate system can only be used locally for limited rotations around the reference point. In order to deal with large rotational amplitudes without singularity problems, a usual technique is to change the reference $\mathbf{R}_{ref}$ during the integration procedure. At time t_{n+1}, one can move from an old reference to the new one $\mathbf{R}_{ref} := \mathbf{R}_{n+1}$, provided that the values of $\boldsymbol{\psi}_{n+1}$, $\dot{\boldsymbol{\psi}}_{n+1}$, $\ddot{\boldsymbol{\psi}}_{n+1}$ and $\mathbf{a}_{n+1}$ are mapped to the new coordinate chart according to the following algorithm [8]

$$\begin{aligned}
\ddot{\boldsymbol{\psi}}_{n+1} &:= \mathbf{T}(\boldsymbol{\psi}_{n+1})\ddot{\boldsymbol{\psi}}_{n+1} + \dot{\mathbf{T}}(\boldsymbol{\psi}_{n+1}, \dot{\boldsymbol{\psi}}_{n+1})\dot{\boldsymbol{\psi}}_{n+1} && (73)\\
\mathbf{a}_{n+1} &:= \mathbf{T}(\boldsymbol{\psi}_{n+1})\mathbf{a}_{n+1} + \dot{\mathbf{T}}(\boldsymbol{\psi}_{n+1}, \dot{\boldsymbol{\psi}}_{n+1})\dot{\boldsymbol{\psi}}_{n+1} && (74)\\
\dot{\boldsymbol{\psi}}_{n+1} &:= \mathbf{T}(\boldsymbol{\psi}_{n+1})\dot{\boldsymbol{\psi}}_{n+1} && (75)\\
\boldsymbol{\psi}_{n+1} &:= \mathbf{0} && (76)
\end{aligned}$$

The update for acceleration-like variables in Equation (74) is required for the consistency of the algorithm. A pragmatic approach is to implement the above update procedure at the end of each time step, so that the current local coordinate system is systematically centered on the total rotation at the preceding time step.

5.3 Example 2 : Heavy top with kinematic constraints

Even though the problem is the same as in Example 1, the following example relies on a differential-algebraic formulation of the dynamic system. According to the absolute coordinate formulation, the equations of motion of a rotating top about a fixed point are written as [2]

$$\begin{aligned}
m\ddot{\mathbf{x}} - \boldsymbol{\lambda} &= m\boldsymbol{\gamma} && (77)\\
\mathbf{J}\dot{\boldsymbol{\Omega}} + \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} + \widetilde{\mathbf{X}}\mathbf{R}^T\boldsymbol{\lambda} &= \mathbf{0} && (78)\\
-\mathbf{x} + \mathbf{R}\mathbf{X} &= \mathbf{0} && (79)
\end{aligned}$$

The vector $\mathbf{x}$ represents the position of the center of mass in the inertial frame and $\mathbf{X}$ represents the position of the center of mass in the body-fixed frame. m is the mass of the top and the inertia tensor $\mathbf{J}$ is now defined with respect to the center of mass (and not with respect to the fixed point as in Example 1). The third equation is a set of 3 algebraic constraints and $\boldsymbol{\lambda}$ is the associated 3×1 vector of Lagrange multipliers.

The set $\mathbb{R}^3 \times SO(3)$ with the composition operation

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{R}_1 \end{bmatrix} \circ \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{R}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 + \mathbf{x}_2 \\ \mathbf{R}_1\mathbf{R}_2 \end{bmatrix} \tag{80}$$

defines a 6-dimensional Lie group denoted as G_6. The exponential map and the tangent operator are constructed in a straightforward way by using independently their definition in $\mathbb{R}^3$ for the translation variables and their definition in $SO(3)$ for the rotation variables. Due to the constraints, the motion is restricted to a 3-dimensional submanifold of G_6 and we have

$$\begin{aligned}
&\mathbf{M} = \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} -m\boldsymbol{\gamma} \\ \boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} \end{bmatrix}, \quad \mathbf{C}_t = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{\boldsymbol{\Omega}}\mathbf{J} - \widetilde{\mathbf{J}\boldsymbol{\Omega}} \end{bmatrix},\\
&\mathbf{K}_t = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \widetilde{\widetilde{\mathbf{X}}\mathbf{R}^T\boldsymbol{\lambda}} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -\mathbf{I}_3 & -\mathbf{R}\widetilde{\mathbf{X}} \end{bmatrix}
\end{aligned} \tag{81}$$

Let us remark that the Lie group G_6 should not be confused with the special Euclidean group $SE(3)$, which is also isomorphic to $\mathbb{R}^3 \times SO(3)$, but whose composition operation is defined by the product of homogeneous transformation matrices of the form

$$\left[\begin{array}{cc} \mathbf{R} & \mathbf{x} \\ \mathbf{0}_{1\times 3} & 1 \end{array} \right]. \tag{82}$$

6 Consistency of the method in the unconstrained case

Let us consider a dynamic system on a n-dimensional Lie group $G = \mathbb{R}^3 \times \ldots \times \mathbb{R}^3 \times SO(3) \times \ldots \times SO(3)$ without any kinematic constraint. The local error of the time integration algorithm is analysed in a local coordinate system $\mathbf{x} = \mu^{-1}(q)$, where μ is an invertible coordinate mapping $\mathbb{R}^n \rightarrow G$. If the exact solution is denoted by $(\mathbf{x}(t), \mathbf{v}(t))$ and the numerical solution after one time step $(\mathbf{x}_1(h), \mathbf{v}_1(h))$, the algorithm is consistent of order two provided that the local errors at position and velocity levels satisfy $\mathbf{x}(h) - \mathbf{x}_1(h) = \mathcal{O}(h^3)$ and $\mathbf{v}(h) - \mathbf{v}_1(h) = \mathcal{O}(h^3)$, respectively. For convenience, the analysis will be performed in the canonical coordinate system centered on q_0, i.e. μ is defined by

$$\mu(\mathbf{x}) = q_0 \circ \exp(\widetilde{\mathbf{x}}), \tag{83}$$

with the properties $\mathbf{x}(0) = \mathbf{0}$ and $\mu(\mathbf{0}) = q_0$.

Theorem. *Using the canonical coordinate system $\mathbf{x} = \mu^{-1}(q)$ centered on q_0, the local errors of the time integration algorithm verify*

$$\mathbf{x}(h) - \mathbf{x}_1(h) = \mathcal{O}(h^3), \quad \mathbf{v}(h) - \mathbf{v}_1(h) = \mathcal{O}(h^3) \tag{84}$$

provided that the algorithmic parameters satisfy the standard second-order condition

$$\gamma = 0.5 + \alpha_f - \alpha_m. \tag{85}$$

Proof. First, we consider the Taylor series expansion for the exact solution

$$\mathbf{x}(h) = h\frac{d\mathbf{x}}{dh}(0) + \frac{h^2}{2}\frac{d^2\mathbf{x}}{dh^2}(0) + \mathcal{O}(h^3), \tag{86}$$

$$\mathbf{v}(h) = \mathbf{v}(0) + h\frac{d\mathbf{v}}{dh}(0) + \frac{h^2}{2}\frac{d^2\mathbf{v}}{dh^2}(0) + \mathcal{O}(h^3). \tag{87}$$

The relation between $\dot{\mathbf{x}}$ and $\mathbf{v}$ is given by

$$\mathbf{T}(\mathbf{x})\dot{\mathbf{x}} = \mathbf{v}, \quad \mathbf{T}(\mathbf{x})\ddot{\mathbf{x}} + \dot{\mathbf{T}}(\mathbf{x})\dot{\mathbf{x}} = \dot{\mathbf{v}}. \tag{88}$$

We have $\mathbf{T} = \mathrm{diag}(\mathbf{T}_{trans}, \ldots, \mathbf{T}_{trans}, \mathbf{T}_{rot}, \ldots, \mathbf{T}_{rot})$, with $\mathbf{T}_{trans} = \mathbf{I}_3$ the tangent operator for translation variables and $\mathbf{T}_{rot}$ the tangent operator for rotation variables as defined in Equation (67). We have $\mathbf{T}(\mathbf{0}) = \mathbf{I}_n$. Moreover, observing that $\dot{\mathbf{T}}_{rot}(\mathbf{0}) = -0.5\widetilde{\dot{\mathbf{x}}}$, we also get $\dot{\mathbf{T}}(\mathbf{0})\dot{\mathbf{x}} = \mathbf{0}$, so that Equations (86) and (87) become

$$\mathbf{x}(h) = h\mathbf{v}(0) + 0.5h^2\dot{\mathbf{v}}(0) + \mathcal{O}(h^3), \tag{89}$$

$$\mathbf{v}(h) = \mathbf{v}(0) + h\dot{\mathbf{v}}(0) + 0.5h^2\ddot{\mathbf{v}}(0) + \mathcal{O}(h^3). \tag{90}$$

Second, we develop Taylor series expansions for the numerical solution. Let us observe that

$$\mathbf{a}_1 = \dot{\mathbf{v}}_0 + h(1 + \alpha_m - \alpha_f)\ddot{\mathbf{v}}_0 + \mathcal{O}(h^2) \tag{91}$$

$$\mathbf{a}_0 = \dot{\mathbf{v}}_0 + h(\alpha_m - \alpha_f)\ddot{\mathbf{v}}_0 + \mathcal{O}(h^2) \tag{92}$$

At position level, depending of the selected algorithmic variant, we have

$$\exp(\widetilde{\mathbf{x}_1}^{(1)}) = \exp(h\widetilde{\mathbf{v}_0} + 0.5h^2\widetilde{\dot{\mathbf{v}}_0} + \mathcal{O}(h^3)) \tag{93}$$

$$\exp(\widetilde{\mathbf{x}_1}^{(2)}) = \exp(h\widetilde{\mathbf{v}_0}) \circ \exp(0.5h^2\widetilde{\dot{\mathbf{v}}_0} + \mathcal{O}(h^3)) \tag{94}$$

$$\exp(\widetilde{\mathbf{x}_1}^{(3)}) = \exp(h\widetilde{\mathbf{v}_0}) \circ \exp(h^2(0.5-\beta)\widetilde{\dot{\mathbf{v}}_0} + \mathcal{O}(h^3)) \circ \exp(\beta h^2\widetilde{\dot{\mathbf{v}}_1} + \mathcal{O}(h^3)) \tag{95}$$

and those three formulae only differ in the treatment of rotation variables. For algorithm (1), we immediately get

$$\mathbf{x}_1(h) = h\widetilde{\mathbf{v}_0} + 0.5h^2\widetilde{\dot{\mathbf{v}}_0} + \mathcal{O}(h^3) \tag{96}$$

This result is also verified for algorithms (2) and (3) using the series expansion of the exponential map for $SO(3)$, see Equation (66). At velocity level, the numerical solution satisfies

$$\mathbf{v}_1(h) = \mathbf{v}_0 + h\dot{\mathbf{v}}_0 + ((1-\gamma)(\alpha_m - \alpha_f) + \gamma(1 + \alpha_m - \alpha_f))h^2\ddot{\mathbf{v}}_0 + \mathcal{O}(h^3) \tag{97}$$

The conclusion follows from a comparison of Equations (89), (90) and Equations (96), (97).

7 Numerical results

This section compares the performance of three different algorithms:

- `geom1` is the geometric Lie group algorithm with the discrete operator $\varphi_h^{(1)}$,
- `geom2` is the geometric Lie group algorithm with the discrete operator $\varphi_h^{(2)}$,
- `linear` is the classical generalized-α algorithm used with an incremental parameterization of the rotation field as described in Section 5.2.

For all methods, the algorithmic parameters are selected according to the formulae described by Chung and Hulbert [12]

$$\alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \quad \alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1}, \quad \beta = \frac{1}{4}(1 + \alpha_f - \alpha_m)^2, \quad \gamma = \frac{1}{2} + \alpha_f - \alpha_m \tag{98}$$

where ρ_∞ represents the desired value of the spectral radius at infinite frequencies. We will consider either a value ρ_∞ close or equal to one (small or no numerical damping), or the value $\rho_\infty = 0.6$ (significant numerical damping).

Five different problems are studied. In all cases, the system is a single object with fast rotating speeds and the problems differ in the value of the inertia tensor, the applied load, the initial conditions and the structure of the equations of motion (without constraint as in Example 1 or with constraints as in Example 2).

For the purpose of convergence analysis, numerical errors are evaluated on the displacement $\mathbf{x}(t_i) = \mathbf{R}(t_i)\mathbf{X}$ at a set of specified times t_i :

$$\text{error} = \frac{1}{n_e}\sum_{i=1}^{n_e} \|\mathbf{x}(t_i) - \mathbf{x}^{ref}(t_i)\| \tag{99}$$

In the first three problems, the reference value is computed using analytical solutions presented by Romano [28, 29]. For the last two problems, for each convergence curve we use the reference $\mathbf{x}^{ref}(t_i)$ computed using the same algorithm but a smaller time step.

7.1 Rotating body with spherical ellipsoid of inertia and follower torque

In this problem, the inertia tensor is $\mathbf{J} = \text{diag}(3., 3., 3.)$, the reference point is $\mathbf{X} = [0.\ 0.\ -0.6]^T$, the applied following torque $\mathbf{C} = [0.\ 0.\ 30.]^T$ is constant and the equations of motion are formulated without any kinematic constraint, as in Example 1. The initial conditions are $\mathbf{R}(0) = \mathbf{I}_3$, $\mathbf{\Omega}(0) = [10.\ 15.\ 20.]^T$ rad/s.

With this particular value of the inertia tensor, the nonlinear gyroscopic and centrifugal force vector is zero, i.e. $\mathbf{\Omega} \times \mathbf{J}\mathbf{\Omega} = \mathbf{0}$. The analytical solution described by Romano [28] is used for the convergence analysis in Fig. 2.

In the results, as predicted by the theory, the numerical errors decrease as $\mathcal{O}(h^2)$. Although numerical errors are quite small for all algorithms, we observe that the geometric algorithms are slightly more accurate than the linear algorithm.

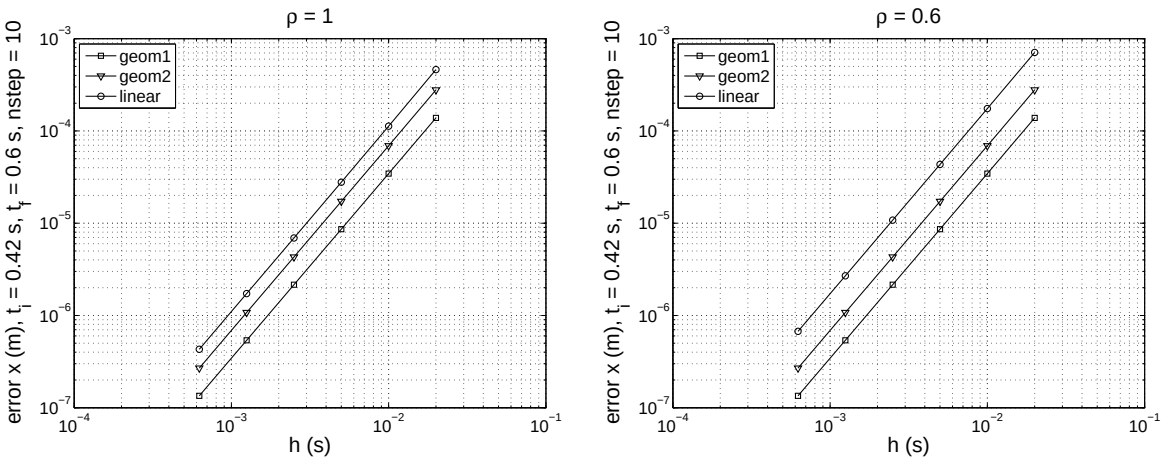

Figure 2: Rotating body with spherical ellipsoid of inertia and follower torque. Error norm evaluated at $t = 0.42 : 0.02 : 0.6$ s.

7.2 Free rotating body

In this problem, the inertia tensor $\mathbf{J} = \text{diag}(21.959, 21.959, 6.671)$ has an axial symmetry, the reference point is $\mathbf{X} = [0.\,0.\ -0.646]^T$, there is no applied torque ($\mathbf{C} = \mathbf{0}$) and the equations of motion are formulated without any kinematic constraint, as in Example 1. The initial conditions are $\mathbf{R}(0) = \mathbf{I}_3$, $\mathbf{\Omega}(0) = [0.\ 0.02\ 36.]^T$ rad/s. The analytical solution described by Romano [29] is used for the convergence analysis.

Time domain and convergence results are given in Fig. 3. After several seconds of simulation, the numerical solutions depart from the analytical solution. The geometric algorithms tend to preserve the amplitude of oscillations, leading to good energy conservation, but the period is underestimated. The converse is observed for the linear algorithm, with an increase in period and in amplitude. In both cases, numerical damping increases the dissipation of energy during the simulation.

Again, all methods are second-order accurate but the error constants are now larger for geometric algorithms, which are slightly less accurate than the linear algorithm.

Trajectories in the $x - y$ plane are also represented on Fig. 4.

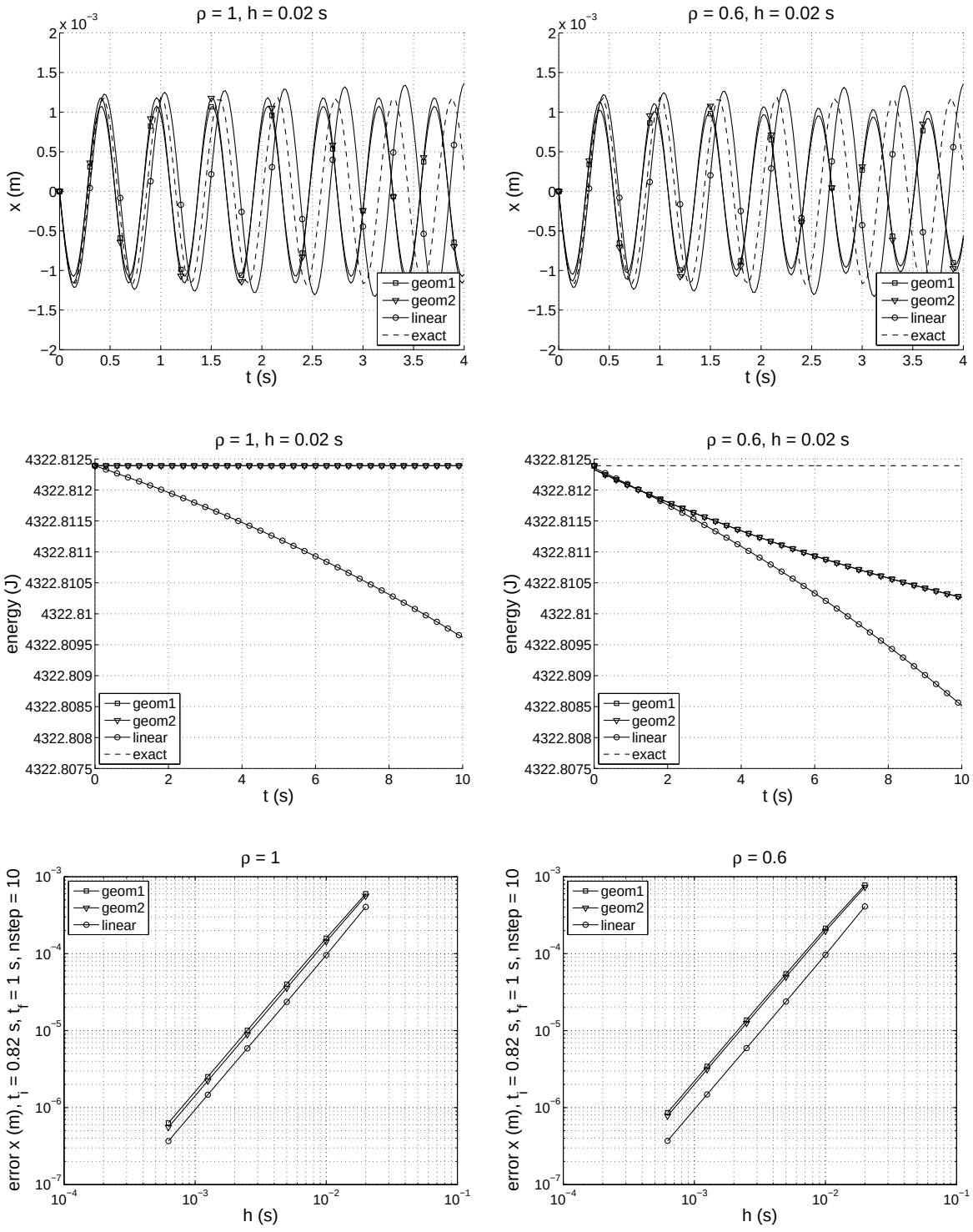

Figure 3: Free rotating body. Error norm evaluated at $t = 0.82 : 0.02 : 1.$ s.

7.3 Rotating object with follower torque

As in the case before, the inertia tensor is $\mathbf{J} = \text{diag}(21.959, 21.959, 6.671)$, the reference point is $\mathbf{X} = [0.\,0.\,-0.646]^T$, but there is an applied following torque $\mathbf{C} = [0.\ 0.\ 40.]^T$ aligned with the third axis of the body. The equations of motion are formulated without any kinematic constraint, as in Example 1. The initial conditions are $\mathbf{R}(0) = \mathbf{I}_3$, $\boldsymbol{\Omega}(0) = [0.\ 0.1\ 10.]^T$ rad/s.

Time domain and convergence results are given in Fig. 5. The numerical solutions depart from the analytical solution with the advancement of computations. Numerical damping increases the dissipation of energy during the simulation.

All methods are second-order accurate but the error constants are again larger for geometric algorithms, which are slightly less accurate than the linear algorithm.

7.4 Heavy top (without kinematic constraints)

In this example, the inertia tensor with respect to the fixed point is $\mathbf{J} = \text{diag}(15.234375, 0.46875, 15.234375)$, the reference point is $\mathbf{X} = [0.\,1.\,0.]^T$, and the applied gravity torque torque $\mathbf{C} = \widetilde{\mathbf{X}}\mathbf{R}^T m\boldsymbol{\gamma}$ with the mass $m = 15$ kg and gravity acceleration $\boldsymbol{\gamma} = [0.\ 0.\ -9.81]^T$ m/s^2. The equations of motion are formulated without any kinematic constraint, as in Example 1. The initial conditions are $\mathbf{R}(0) = \mathbf{I}_3$, $\boldsymbol{\Omega}(0) = [0.\ 150.\ -4.61538]^T$ rad/s.

For each convergence curve, the reference solution is computed using a small time step $h = 1.5625 \times 10^{-5}$ s.

Even if numerical damping is not introduced, the algorithms do not preserve the energy anymore. The linear algorithm shows a very small decrease in total energy, while the geometric algorithms present some oscillations with a larger drift in the `geom2` algorithm.

All methods are second-order accurate but the error constants are again larger for geometric algorithms, which are slightly less accurate than the linear algorithm.

7.5 Heavy top (with kinematic constraints)

The previous example is now formulated with kinematic constraints as in Example 2. The inertia tensor is now evaluated with respect to the center of mass (and not with respect to the fixed point as before), with $\mathbf{J} = \text{diag}(0.234375, 0.46875, 0.234375)$. The other parameters of the model are as in Section 7.4. Note that the initial acceleration is computed as $\dot{\boldsymbol{\Omega}}(0) = \overline{\mathbf{J}}^{-1}(\mathbf{X} \times m\boldsymbol{\gamma} - \boldsymbol{\Omega}(0) \times \overline{\mathbf{J}}\boldsymbol{\Omega}(0))$, where $\overline{\mathbf{J}} = \mathbf{J} - m\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}$ is the inertia tensor with respect to the fixed point (as in Section 7.4). We also have, according to the constraints, $\mathbf{x}(0) = \mathbf{X}$, $\dot{\mathbf{x}}(0) = \boldsymbol{\Omega}(0) \times \mathbf{X}$ and $\ddot{\mathbf{x}}(0) = \dot{\boldsymbol{\Omega}}(0) \times \mathbf{X} + \boldsymbol{\Omega}(0) \times (\boldsymbol{\Omega}(0) \times \mathbf{X})$.

The case without numerical damping was not considered because of stability problems in the presence of kinematic constraints. Instead, a lightly damped case was considered ($\rho = 0.9$).

Again, for each convergence curve, the reference solution is computed using a small time step $h = 1.5625 \times 10^{-5}$ s.

The results are displayed in Fig. 7. We observe that the results are more accurate in this case that in the unconstrained case. After several seconds of simulation, small drifts in the energy are observed in all cases (the apparent energy conservation of geometric algorithm 1 is a coincidence in this case). All algorithms are of order 2, with different error constants. The improved accuracy might be explained by the smaller value of nonlinear gyroscopic forces when the inertia tensor is evaluated with respect to the center of mass.

Figure 8 represents the large rotation angles (here, the component 2 of the Cartesian rotation vector $\boldsymbol{\psi}$, which is defined such that $\|\boldsymbol{\psi}\| \leq \pi$) and the oscillations in the multipliers at the beginning of the simulation. A stability analysis might allow a better understanding of these phenomena; this issue should be investigated in a future work.

8 Conclusions

This paper proposes a family of Lie group time integrators for the simulation of flexible multibody systems. This method provides an elegant solution to the rotation parameterization problem and it does not suffer from parameterization singularities. As an extension of the classical generalized-α method for dynamic systems, it can deal with constrained equations of motion. Second-order accuracy has been demonstrated for three algorithms of this family in the unconstrained case, and has been verified for several examples in the constrained and unconstrained cases.

In several critical benchmarks of rigid body systems with large rotation speeds, the performance of the Lie group algorithms has been compared to classical parameterization-based methods. Even though the performance of the different algorithms is problem dependent, we conclude that the new methods can compete reasonably well with classical approaches from the viewpoint of accuracy, stability and energy conservation.

The main advantages of the proposed approach are related with its particular formulation, which is both generic and remarkably simple compared to parameterization-based algorithms. Tangent operators can be easily derived analytically without any approximation, at least for rigid multibody problems. This allows an efficient implementation of the Newton algorithm at each time step of the implicit numerical integration. The improved efficiency of the resulting code is an advantage for real-time applications, large scale problems or parametric studies. The availability of simple formulations of tangent operators is also useful for the implementation of a sensitivity analysis based on a semi-analytical approach, which opens some interesting perspectives for the optimization of multibody systems. Finally, model simplicity and efficiency are valuable for the development of model-based control schemes. Hence, the proposed Lie group time integrators are promising candidates for the development of robust, efficient and open simulation software for flexible multibody systems.

For constrained dynamic systems on Lie group, further studies may include a stability analysis as well as a global convergence analysis.

Acknowledgment

The first author gratefully acknowledges Peter Eberhard (University of Stuttgart) who participated in initial experiments on Lie group methods for the simulation of multibody systems.

References

[1] Wasfy, T., and Noor, A., 2003. "Computational strategies for flexible multibody systems". *Applied Mechanics Review,* **56**(6), pp. 553–613.

[2] Géradin, M., and Cardona, A., 2001. *Flexible Multibody Dynamics: A Finite Element Approach.* John Wiley & Sons, New York.

[3] Crouch, P., and Grossman, R., 1993. "Numerical integration of ordinary differential equations on manifolds". *Journal of Nonlinear Science,* **3**, pp. 1–33.

[4] Munthe-Kaas, H., 1995. "Lie-Butcher theory for Runge-Kutta methods". *BIT,* **35**, pp. 572–587.

[5] Munthe-Kaas, H., 1998. "Runge-Kutta methods on Lie groups". *BIT,* **38**, pp. 92–111.

[6] Simo, J., and Vu-Quoc, L., 1988. "On the dynamics in space of rods undergoing large motions - a geometrically exact approach". *Computer Methods in Applied Mechanics and Engineering,* **66**, pp. 125–161.

[7] Simo, J., and Wong, K., 1991. "Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum". *International Journal for Numerical Methods in Engineering,* **31**, pp. 19–52.

[8] Cardona, A., and Géradin, M., 1988. "A beam finite element non-linear theory with finite rotations". *International Journal for Numerical Methods in Engineering,* **26**, pp. 2403–2438.

[9] Cardona, A., and Géradin, M., 1989. "Time integration of the equations of motion in mechanism analysis". *Computers and Structures,* **33**, pp. 801–820.

[10] Newmark, N., 1959. "A method of computation for structural dynamics". *ASCE Journal of the Engineering Mechanics Division,* **85**, pp. 67–94.

[11] Hilber, H., Hughes, T., and Taylor, R., 1977. "Improved numerical dissipation for time integration algorithms in structural dynamics". *Earthquake Engineering and Structural Dynamics,* **5**, pp. 283–292.

[12] Chung, J., and Hulbert, G., 1993. "A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method". *ASME Journal of Applied Mechanics,* **60**, pp. 371–375.

[13] Arnold, M., and Brüls, O., 2007. "Convergence of the generalized-α scheme for constrained mechanical systems". *Multibody System Dynamics,* **18**(2), pp. 185–202.

[14] Lunk, C., and Simeon, B., 2006. "Solving constrained mechanical systems by the family of Newmark and α-methods". *Journal of Applied Mathematics and Mechanics (ZAMM),* **86**(10), pp. 772–784.

[15] Jay, L., and Negrut, D., 2007. "Extensions of the HHT-method to differential-algebraic equations in mechanics". *Electronic Transactions on Numerical Analysis,* **26**, pp. 190–208.

[16] Arnold, M., 2009. "The generalized-α method in industrial multibody system simulation". In Proceedings of Multibody Dynamics 2009, Eccomas Thematic Conference, K. Arczewski, J. Fraczek, and M. Wojtyra, eds.

[17] Celledoni, E., and Owren, B., 2003. "Lie group methods for rigid body dynamics and time integration on manifolds". *Computer Methods in Applied Mechanics and Engineering,* **192**(3-4), pp. 421–438.

[18] Bottasso, C., and Borri, M., 1998. "Integrating finite rotations". *Computer Methods in Applied Mechanics and Engineering,* **164**, pp. 307–331.

[19] Brüls, O., and Eberhard, P., 2008. "Sensitivity analysis for dynamic mechanical systems with finite rotations". *International Journal for Numerical Methods in Engineering,* **74**(13), pp. 1897–1927.

[20] Gonzalez, O., 1996. "Time integration and discrete hamiltonian systems". *Journal of Nonlinear Science,* **6**, pp. 449–467.

[21] Bauchau, O., and Bottasso, C., 1999. "On the design of energy preserving and decaying schemes for flexible nonlinear multi-body systems". *Computer Methods in Applied Mechanics and Engineering,* **169**, pp. 61–79.

[22] Betsch, P., and Steinmann, P., 2001. "Constrained integration of rigid body dynamics". *Computer Methods in Applied Mechanics and Engineering,* **191**, pp. 467–488.

[23] Ibrahimbegovic, A., and Mamouri, S., 2002. "Energy conserving/decaying implicit time-stepping scheme for nonlinear dynamics of three-dimensional beams undergoing finite rotations". *Computer Methods in Applied Mechanics and Engineering,* **191**, pp. 4241–4258.

[24] Kuhl, D., and Crisfield, M., 1999. "Energy-conserving and decaying algorithms in non-linear structural dynamics". *International Journal for Numerical Methods in Engineering,* **45**, pp. 569–599.

[25] Lens, E., Cardona, A., and Géradin, M., 2004. "Energy preserving time integration for constrained multibody systems". *Multibody System Dynamics,* **11**(1), pp. 41–61.

[26] Kane, C., Marsden, J., Ortiz, M., and West, M., 2000. "Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems". *International Journal for Numerical Methods in Engineering,* **49**, pp. 1295–1325.

[27] Hairer, E., Lubich, C., and Wanner, G., 2006. *Geometric Numerical Integration - Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed. Springer-Verlag.

[28] Romano, M., 2008. "Exact analytic solution for the rotation of a rigid body having spherical ellipsoid of inertia and subjected to a constant torque". *Celestial Mechanics and Dynamical Astronomy,* **100**, pp. 181–189.

[29] Romano, M., 2008. "Exact analytic solutions for the rotation of an axially symmetric rigid body subjected to a constant torque". *Celestial Mechanics and Dynamical Astronomy,* **101**, pp. 375–390.

[30] Boothby, W., 2003. *An Introduction to Differentiable Manifolds and Riemannian Geometry*, 2nd ed. Academic Press.

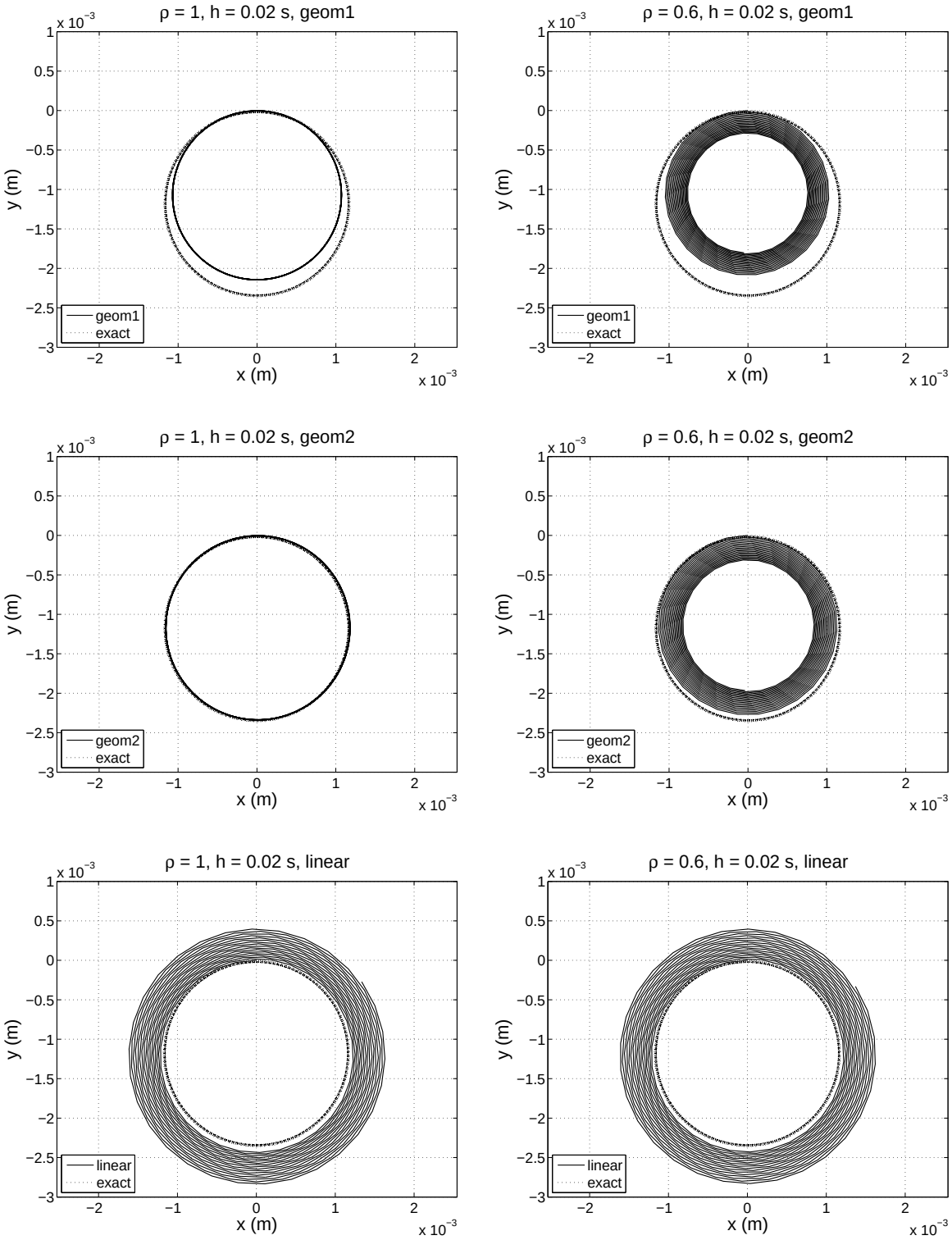

Figure 4: Free rotating body. Trajectories in $x - y$ plane for all algorithms, compared to analytical solution.

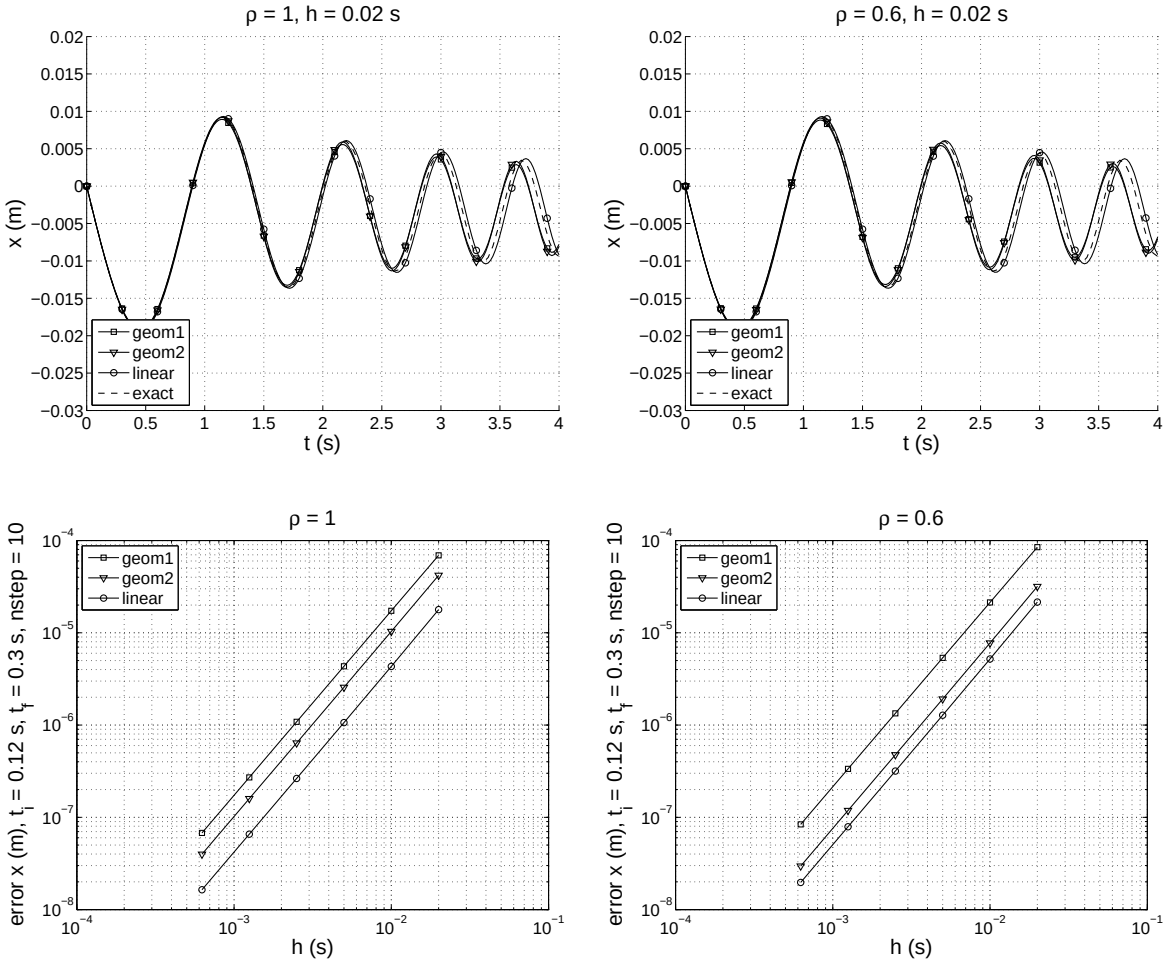

Figure 5: Rotating body with follower torque. Error norm evaluated at $t = 0.12 : 0.02 : 0.3$ s.

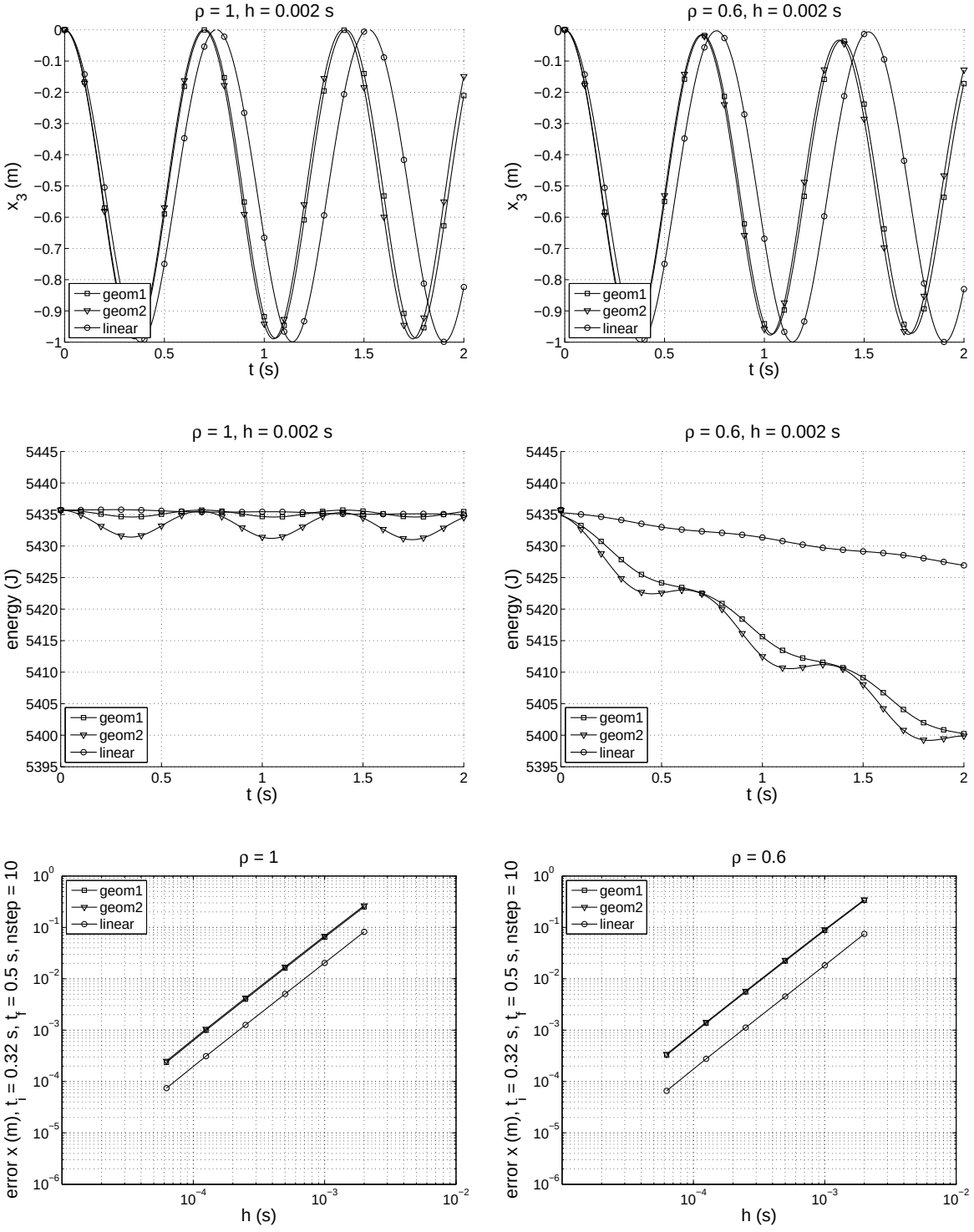

Figure 6: Heavy top (without kinematic constraints). Error norm evaluated at $t = 0.32 : 0.02 : 0.5$ s.

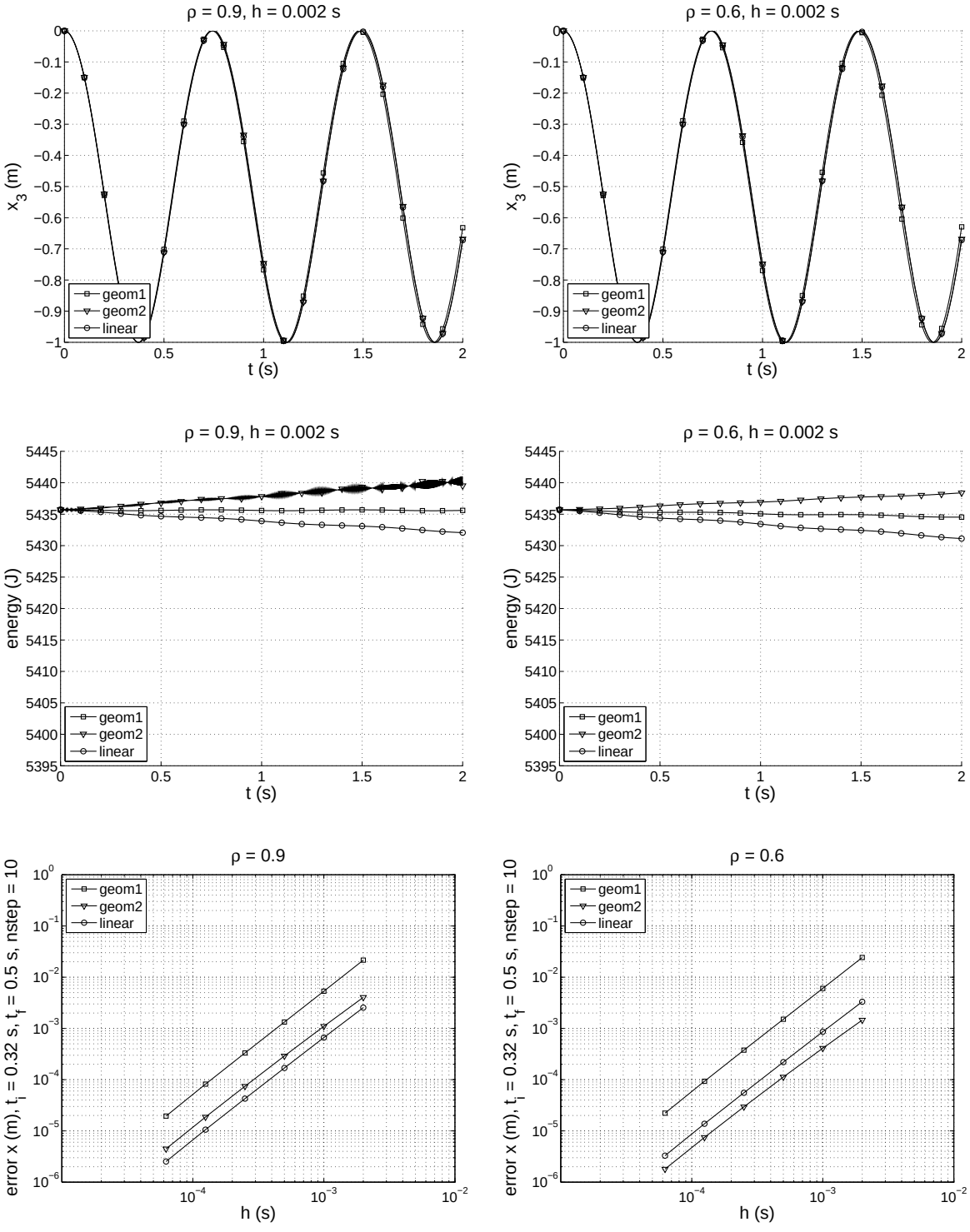

Figure 7: Heavy top (with kinematic constraints).

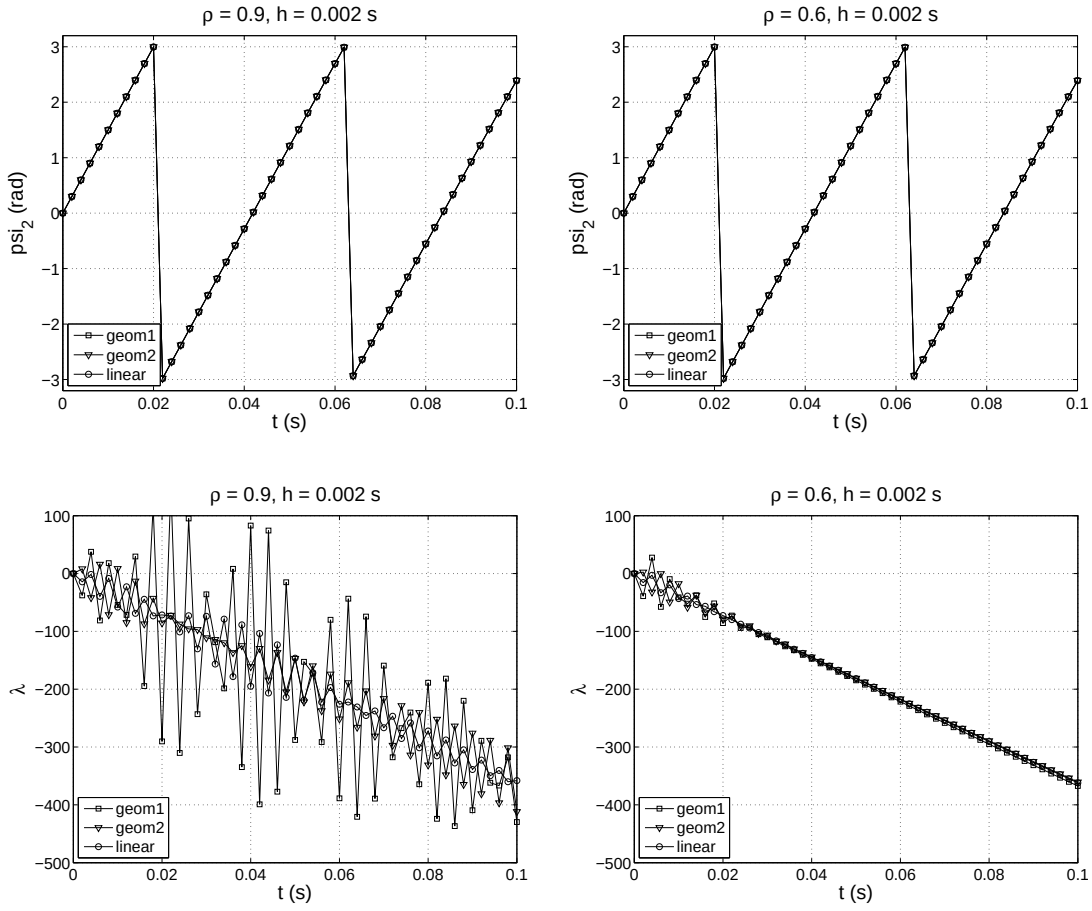

Figure 8: Heavy top (with kinematic constraints).