



Local coordinates on Lie groups for half-explicit time integration of Cosserat-rod models with constraints

Denise Tumiotto¹ · Martin Arnold¹

Received: 29 January 2024 / Accepted: 3 June 2024
© The Author(s) 2024

Abstract

Explicit Runge–Kutta methods are the gold standard of time-integration methods for nonstiff problems in system dynamics since they combine a small numerical effort per time step with high accuracy, error control, and straightforward implementation. For the analysis of beam dynamics, we couple them with a local coordinates approach in a Lie group setting to address large rotations. Stiff shear forces and inextensibility conditions are enforced by internal constraints in a coarse-grid discretization of a geometrically exact beam model. The resulting nonstiff constrained systems are handled by a half-explicit approach that relies on the constraints at velocity level and avoids all kinds of Newton–Raphson iteration. We construct half-explicit Runge–Kutta Lie group methods of order up to five that are equipped with an adaptive step-size strategy using embedded Runge–Kutta pairs for error estimation. The methods are tested successfully for a roll-up maneuver of a flexible beam and for the classical flying-spaghetti benchmark.

Keywords Lie group time integration · Half-explicit Runge-Kutta methods · Constrained systems

1 Introduction

Geometric numerical methods preserve the essential properties of the flow of differential equation models [13]. Such strategies are important for the robust, efficient, and numerically stable dynamical simulations of highly flexible structures. In space, they are used to obtain coarse-grid discretizations that reflect the characteristic behavior of geometrically

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860124. This publication reflects only the author’s view and the Research Executive Agency is not responsible for any use that may be made of the information it contains.

✉ D. Tumiotto
denise.tumiotto@mathematik.uni-halle.de

M. Arnold
martin.arnold@mathematik.uni-halle.de

¹ Institute of Mathematics, Martin-Luther-University Halle-Wittenberg, D-06099 Halle (Saale), Germany

exact beam models. In the time domain, geometric integration is tailored to nonlinear configuration spaces that are typical of mechanical systems with large rotations.

In the present paper, we address the latter aspect and present a novel Runge–Kutta Lie group integrator for constrained mechanical systems and its application to beam dynamics. The method is half-explicit and avoids all kinds of Newton–Raphson iterations that are a bottleneck for the efficiency of classical implicit integrators in beam analysis. Stiff shear forces are represented by internal constraints [19, 20] that may be used as well to enforce a beam’s inextensibility. These internal constraints are combined with a coarse-grid space discretization and result finally in a constrained system that is for typical application scenarios nonstiff in its differential part.

The equations of motion of constrained mechanical systems form differential-algebraic equations (DAEs) with Lagrange multipliers as algebraic variables that couple the constraints to the equilibrium equations for forces and moments [12, 13]. Following the approach of Brasey and Hairer [6], the equations of motion are transformed analytically before half-explicit time discretization, resulting in the index-2 formulation with constraint equations at the level of velocity coordinates. In that way, the velocity vector is constrained to the null space of the constraint Jacobian that is known from null-space approaches like the one of Betsch [4]. In the context of Lie group time integration, these velocity constraints have been used before in the RATTLie integrator [16] and in the application of generalized- α methods to the stabilized index-2 formulation of the equations of motion [2].

The successful application of generalized- α and other Newmark-type Lie group integrators in flexible multibody dynamics [7, 8, 12] relies on the specific Lie group structure of the configuration space for mechanical systems with large rotations [22, 23]. The Lie group setting is attractive in this context since it allows us to describe large rotations and the orientation of bodies and flexible structures without any singularities [27]. The spaces one comes across are the space of Special Orthogonal matrices $SO(3)$ or the space of unit quaternions \mathbb{S}^3 , their direct or semidirect product with \mathbb{R}^3 , and tensor products thereof.

The half-explicit Runge–Kutta Lie group integrator is tested for a geometrically exact beam model that goes back to Lang and Linn [19, 20]. The space discretization follows the variational integration approach of Hante [15, 17] with nodal variables $(q_i, x_i) \in \mathbb{S}^3 \times \mathbb{R}^3$. The semidirect product of \mathbb{S}^3 and \mathbb{R}^3 may be considered to be isomorphic to the Special Euclidean group $SE(3)$, taking into account that \mathbb{S}^3 is a double covering of the Special Orthogonal group $SO(3)$, i.e., $q, -q \in \mathbb{S}^3$ represent the same rotation matrix $R \in SO(3)$.

In the work of Munthe-Kaas [24], see also Iserles et al. [18], classical Runge–Kutta methods are generalized to Lie group integrators with the help of the exponential map that defines local coordinates on the Lie group by a tangent space parametrization. The possibility to work locally in the tangent space, which is a linear space with the well-known operations among its elements, simplifies substantially the numerical solution of a system set in a Lie group. The use of the exponential map assures that the numerical solution remains in the Lie group, given that we work with local coordinates.

In a general Lie group setting, the construction and implementation of higher-order Runge–Kutta Munthe-Kaas methods is challenging since they involve frequent evaluations of matrix exponentials [21] and the approximation of its derivatives [24]. There are no such problems in the application to $SO(3)$ since Rodrigues’ formula allows us to evaluate the exponential map in closed form and a similar expression may be found for its derivative [18]. Park and Chung generalized this approach to the Special Euclidean group $SE(3)$ and pointed out that any classical explicit Runge–Kutta method may be generalized in that way to a Lie group integrator on $SE(3)$ with the same order of convergence [26]. Several recent studies focus on the closed-form expression of the exponential map and its derivatives, as well as

their efficient evaluation [15, 30, 32]. Additionally, one may be concerned with singularities occurring at the origin, which are solved by extended investigation of the approximation of the closed form by a Taylor polynomial [15, Appendix B].

The present paper considers nonstiff integrators that are a quasistandard in nonlinear system dynamics, such as the default integrator ode45 in MATLAB’s ODE suite [28], addressing two main challenges: Nonlinear configuration spaces and constraints. We aim to solve the former by introducing Lie group integrators and the latter by adapting the existing numerical method of half-explicit Runge–Kutta type [6].

2 Half-explicit Runge–Kutta Lie group integrators

In the present section, we elaborate the definition of the method. In the early 1990s, Brasey and Hairer introduced the half-explicit Runge–Kutta methods for semiexplicit index-2 DAEs in Hessenberg form [6]. Some years later, Murua [25] and Arnold [1] proposed a modification, which simplifies the order conditions of higher order methods and makes the approach more efficient. In each time step, the methods start with an explicit stage and evaluate in all later stages velocity vectors in the null space of the corresponding constraint Jacobians.

2.1 Equations of motion and local coordinates

First, we consider a flexible multibody system in the Lie group $G := (\mathbb{S}^3 \times \mathbb{R}^3)^{N+1}$, with N denoting the number of beam edges after space discretization [15]. The elements $q \in G$ have $7(N + 1)$ components: 4 for the orientation and 3 for the position of each frame that is attached to one of the $N + 1$ nodes of this discretization. Let e be the identity element of the Lie group G . We introduce the elements of the tangent space $\tilde{\mathbf{v}} \in T_e G$, that summarize both angular and translational velocity. The tangent space $T_e G$ at the identity element has been identified with $\mathbb{R}^{6(N+1)}$ through an isomorphism called the tilde operator $\tilde{\bullet} : \mathbb{R}^{6(N+1)} \rightarrow T_e G$.

By the use of the previous elements, we set the equations of motion for a constrained flexible multibody system as expressed in [7]:

$$\dot{q} = DL_q(e) \cdot \tilde{\mathbf{v}} \tag{1a}$$

$$\mathbf{M}(q)\dot{\mathbf{v}} = -\mathbf{g}(t, q, \mathbf{v}) - \mathbf{B}^\top(q)\lambda \tag{1b}$$

$$\mathbf{0} = \Phi(q), \tag{1c}$$

where $\mathbf{M}(q)$ is the mass and inertia matrix, $\mathbf{g}(t, q, \mathbf{v})$ is the vector of external and internal forces, and $\mathbf{B}(q)$ is the gradient of the constraint function at the position level $\Phi(q)$ in the sense of [7, Eq. (9)]:

$$D\Phi(q) \cdot (DL_q(e) \cdot \tilde{\mathbf{w}}) = \mathbf{B}(q)\mathbf{w}, \quad \mathbf{w} \in \mathbb{R}^{6(N+1)}. \tag{2}$$

Equations (1a)–(1c) are the index-3 formulation of the equations of motion. The right-hand side $DL_q(e) \cdot \tilde{\mathbf{v}}$ of the kinematic equation (1a) evaluates at $y = e$ the directional derivative of the left translation $L_q(y) = q \circ y$ in the direction of $\tilde{\mathbf{v}} \in T_e G$:

$$\dot{q}(t) = \lim_{\epsilon \rightarrow 0} \frac{q(t) \circ \exp(\epsilon \tilde{\mathbf{v}}(t)) - q(t)}{\epsilon}. \tag{3}$$

We can perform an index reduction by differentiating in time the constraints in Equation (1c):

$$\mathbf{0} = \frac{d\Phi}{dt}(q(t)) = D\Phi(q(t)) \cdot \dot{q}(t) = D\Phi(q(t)) \cdot (DL_{q(t)}(e) \cdot \tilde{\mathbf{v}}(t)) = \mathbf{B}(q(t))\mathbf{v}(t). \tag{4}$$

The index-2 formulation of the equations of motion is

$$\dot{q} = DL_q(e) \cdot \tilde{\mathbf{v}} \tag{5a}$$

$$\mathbf{M}(q)\dot{\mathbf{v}} = -\mathbf{g}(t, q, \mathbf{v}) - \mathbf{B}^\top(q)\lambda \tag{5b}$$

$$\mathbf{0} = \mathbf{B}(q)\mathbf{v}. \tag{5c}$$

It is analytically equivalent to (1a)–(1c) and avoids the time-consuming evaluation of curvature terms in the index-1 formulation of the equations of motion [31, 33].

Given the equations of motion, the solution in local coordinates, in the neighborhood of $t = t_n$, is

$$q(t) = q(t_n) \circ \exp(\tilde{\boldsymbol{\theta}}_n(t)), \tag{6}$$

where $\boldsymbol{\theta}_n(t)$ denotes a local parametrization of the Lie group with $\boldsymbol{\theta}_n(t_n) = \mathbf{0}$. Differentiating (6) in time, the time derivative of $q(t)$ is given by

$$\dot{q}(t) = \lim_{\epsilon \rightarrow 0} \frac{q(t + \epsilon) - q(t)}{\epsilon} = \lim_{\epsilon \rightarrow 0} \frac{q(t_n) \circ \exp(\tilde{\boldsymbol{\theta}}_n(t + \epsilon)) - q(t)}{\epsilon}. \tag{7}$$

Comparing this equation with (3), they have to coincide for finite ϵ up to higher-order terms:

$$q(t) \circ \exp(\epsilon \tilde{\mathbf{v}}) = q(t_n) \circ \exp(\tilde{\boldsymbol{\theta}}_n(t) + \epsilon \dot{\tilde{\boldsymbol{\theta}}}_n(t)) + o(\epsilon). \tag{8}$$

Given that

$$\left. \frac{d}{d\epsilon} \exp(\tilde{\boldsymbol{\theta}}_n(t) + \epsilon \dot{\tilde{\boldsymbol{\theta}}}_n(t)) \right|_{\epsilon=0} = \exp(\tilde{\boldsymbol{\theta}}_n(t)) \text{dexp}_{-\tilde{\boldsymbol{\theta}}_n(t)} \dot{\tilde{\boldsymbol{\theta}}}_n(t) \tag{9}$$

with $\text{dexp}_{-\tilde{\boldsymbol{\theta}}_n(t)}$ denoting the right trivialized differential of \exp , see [18, Definition 2.18], we obtain

$$q(t_n) \circ \exp(\tilde{\boldsymbol{\theta}}_n(t) + \epsilon \dot{\tilde{\boldsymbol{\theta}}}_n(t)) = q(t_n) \circ \exp(\tilde{\boldsymbol{\theta}}_n(t)) \circ \exp(\epsilon \text{dexp}_{-\tilde{\boldsymbol{\theta}}_n(t)} \dot{\tilde{\boldsymbol{\theta}}}_n(t)) + o(\epsilon)$$

and (8) results in

$$\tilde{\mathbf{v}}(t) = \text{dexp}_{-\tilde{\boldsymbol{\theta}}_n(t)} \dot{\tilde{\boldsymbol{\theta}}}_n(t). \tag{10}$$

For practical computations, the equivalent representation in terms of the tangent operator [7, 8] proves to be favorable:

$$\mathbf{v}(t) = \mathbf{T}(\boldsymbol{\theta}_n(t))\dot{\boldsymbol{\theta}}_n(t). \tag{11}$$

The dexp operator may be evaluated in terms of iterations of the adjoint operator [18, Equation (2.44)] resulting in a series expansion of the tangent operator \mathbf{T} , see [8], that may be summarized to closed-form expressions for all Lie groups of practical relevance in multi-body dynamics, see [18, 26] and the more detailed discussion in Sect. 2.3.

2.2 Half-explicit Runge–Kutta Lie group methods

In classical Runge–Kutta methods, we use parameters a_{ij} , b_j , c_i , ($i, j = 1, \dots, s$) to build stage vectors and numerical solution. In particular, we evaluate s stage vectors using a_{ij} , c_i and we compute the numerical solution as a linear combination of the s stage vectors weighted by the b_j parameters. When using the half-explicit Runge–Kutta methods, we may introduce more stage vectors ($\bar{s} \geq s$), and use the notation $b_j = a_{s+1,j}$, ($j = 1, \dots, s$). In the Lie group integrator, the stage vectors are

$$Q_{ni} = q_n \circ \exp(\tilde{\Theta}_{ni}), \quad \Theta_{ni} = h \sum_{j=1}^{i-1} a_{ij} \dot{\Theta}_{nj}, \quad \mathbf{V}_{ni} = \mathbf{v}_n + h \sum_{j=1}^{i-1} a_{ij} \dot{\mathbf{V}}_{nj}, \quad (12)$$

$(i = 1, \dots, \bar{s} + 1),$

with

$$\dot{\Theta}_{ni} = \mathbf{T}^{-1}(\Theta_{ni})\mathbf{V}_{ni}, \quad (i = 1, \dots, \bar{s}), \quad (13a)$$

$$\mathbf{M}(Q_{ni})\dot{\mathbf{V}}_{ni} = -\mathbf{g}(t_n + c_i h, Q_{ni}, \mathbf{V}_{ni}) - \mathbf{B}^\top(Q_{ni})\Lambda_{ni}, \quad (i = 1, \dots, \bar{s}). \quad (13b)$$

The first stage ($i = 1$) is explicit with

$$Q_{n1} = q_n, \quad \Theta_{n1} = \mathbf{0}, \quad \mathbf{V}_{n1} = \mathbf{v}_n, \quad \Lambda_{n1} = \lambda_n.$$

For $i = 2, \dots, \bar{s}$, we enforce the hidden constraints at velocity level (5c) for stage vectors $Q_{n,i+1}$, $\mathbf{V}_{n,i+1}$ resulting in

$$\mathbf{0} = \mathbf{B}(Q_{n,i+1})\mathbf{V}_{n,i+1} = \mathbf{B}(Q_{n,i+1})(\mathbf{v}_n + h \sum_{j=1}^{i-1} a_{i+1,j} \dot{\mathbf{V}}_{nj} + h a_{i+1,i} \dot{\mathbf{V}}_{ni}), \quad (13c)$$

$(i = 2, \dots, \bar{s}).$

Equations (13b) and (13c) form a system of linear equations in terms of Λ_{ni} , $\dot{\mathbf{V}}_{ni}$, ($i = 2, \dots, \bar{s}$), that motivate the notation *half-explicit* for this class of time-integration methods [6]:

$$\begin{pmatrix} \mathbf{M}(Q_{ni}) & \mathbf{B}^\top(Q_{ni}) \\ h a_{i+1,i} \mathbf{B}(Q_{n,i+1}) & \mathbf{0} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{V}}_{ni} \\ \Lambda_{ni} \end{pmatrix} = \begin{pmatrix} -\mathbf{g}(t_n + c_i h, Q_{ni}, \mathbf{V}_{ni}) \\ -\mathbf{B}(Q_{n,i+1})(\mathbf{v}_n + h \sum_{j=1}^{i-1} a_{i+1,j} \dot{\mathbf{V}}_{nj}) \end{pmatrix}$$

$(i = 2, \dots, \bar{s}). \quad (14)$

The total number of linear systems to be solved in each time step is $\bar{s} - 1$. Note that the half-explicit approach avoids all iterations and does not rely on some kind of Newton–Raphson method.

The updated solution at time step $t = t_{n+1}$ is

$$q_{n+1} = Q_{n,s+1} = q_n \circ \exp(\tilde{\Theta}_{n,s+1}), \quad \mathbf{v}_{n+1} = \mathbf{V}_{n,s+1}, \quad \lambda_{n+1} = \sum_{i=1}^{\bar{s}} d_i \Lambda_{ni}, \quad (15)$$

Table 1 Parameters of half-explicit methods up to order $p = 5$

Order 2, $s = \bar{s} = 2$		Order 3, $s = \bar{s} = 3$		Order 4, $s = 4, \bar{s} = 5$														
0	1	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	1	$\frac{1}{2}$	$\frac{3}{8}$	$\frac{1}{8}$								
1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{2}{9}$	$\frac{2}{3}$	$\frac{4}{9}$	1	$-\frac{1}{2}$	$-\frac{1}{2}$	2	$\frac{1}{6}$	0	$\frac{2}{3}$	$\frac{1}{6}$					
$d = (1/2, 1/2)$		$d = (0, -2, 3)$		$d_5 = 1, d_i = 0, (i = 1, \dots, 4)$						$a_{61} = \frac{1}{6} - \frac{\sqrt{3}}{108}, a_{63} = \frac{1}{3} - \frac{4\sqrt{3}}{27}$	a_{63}	$-\frac{7\sqrt{3}}{108}$	$\frac{\sqrt{3}}{18}$					
$\frac{1}{2} - \frac{\sqrt{3}}{6}$		$\frac{1}{6}$		0		$\frac{2}{3}$		$\frac{1}{6}$		0		0		0				
Order 5, $s = 6, \bar{s} = 7$																		
0	$\frac{1}{5}$																	
$\frac{1}{5}$	$\frac{3}{40}$	$\frac{9}{40}$																
$\frac{3}{10}$	$\frac{4}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$															
$\frac{4}{5}$	$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$													
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$													
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$												
$\frac{19}{20}$	a_{81}	$\frac{59332529}{14479296}$	a_{83}	$\frac{2763523204159}{3289696051200}$	a_{85}	$\frac{46310205821}{287848404480}$	$-\frac{3280}{75413}$											
$d_7 = 1, d_i = 0, (i = 1, \dots, 6), \text{ i.e. } \lambda_{n+1} = \Lambda_n$		$\frac{35}{384}$		0		$\frac{500}{1113}$		$\frac{125}{192}$		$-\frac{2187}{6784}$		$\frac{11}{84}$		0				
$a_{81} = -\frac{18611506045861}{19738176307200}, a_{83} = -\frac{2509441598627}{893904224850}, a_{85} = -\frac{41262869588913}{116235927142400}$																		

where d_j are additional parameters for the computation of the Lagrange multipliers. Their value, as for the other parameters, is set by order conditions and by a contractivity condition to ensure zero stability. The method has been tested up to order five. We list in Table 1 the Butcher tableaux with parameters $a_{ij}, (i = 1, \dots, \bar{s} + 1, j < i)$ for the implemented methods [1].

In a local coordinates approach, the order analysis in [1], valid for linear spaces, appears to be applicable also for nonlinear configuration spaces. In our case, the tests are performed for the Lie groups $SE(3)$ and $SO(3) \times \mathbb{R}^3$ in Sect. 2.4 and for Lie group $(\mathbb{S}^3 \times \mathbb{R}^3)^{N+1}$ for the applications to beam dynamics.

2.3 Implementation issues

As stated in the previous subsection, the Lie groups of interest are $SO(3), \mathbb{S}^3$, their direct or semidirect product with \mathbb{R}^3 , and tensor products thereof. Inspired by Rodrigues' formula [18]

$$\exp_{SO(3)}(\tilde{\Omega}) = \mathbf{I}_3 + \frac{\sin \omega}{\omega} \tilde{\Omega} + \frac{1 - \cos \omega}{\omega^2} \tilde{\Omega}^2, \tag{16}$$

we consider the closed-form expression, see, e.g., [5]

$$\exp_{\mathbb{S}^3}(\tilde{\Omega}) = \begin{bmatrix} \cos(\frac{\omega}{2}) \\ \frac{\Omega}{\omega} \sin(\frac{\omega}{2}) \end{bmatrix}. \tag{17}$$

Here and later in the current section, we set $\omega = \|\Omega\|_2$.

Following the approach of Hante [15], we consider the semidirect product $\mathbb{S}^3 \ltimes \mathbb{R}^3$ that introduces a coupling between the orientation and the position variables, which results in the appearance of the tangent operator in the exponential map as it is known from $SE(3)$. Given $\mathbf{v} = [\Omega^\top, \mathbf{U}^\top]^\top$

$$\exp_{\mathbb{S}^3 \ltimes \mathbb{R}^3}(\tilde{\mathbf{v}}) = \left(\exp_{\mathbb{S}^3}(\tilde{\Omega}), \mathbf{T}_{\mathbb{S}^3}^\top(\Omega) \cdot \mathbf{U} \right), \tag{18}$$

where the tangent operator $\mathbf{T}_{\mathbb{S}^3}(\Omega)$ is

$$\mathbf{T}_{\mathbb{S}^3}(\Omega) = \mathbf{I}_3 + \frac{\cos \omega - 1}{\omega^2} \tilde{\Omega} + \frac{\omega - \sin \omega}{\omega^3} \tilde{\Omega}^2. \tag{19}$$

At the same time, we need to specify the tangent operator on $\mathbb{S}^3 \ltimes \mathbb{R}^3$

$$\mathbf{T}_{\mathbb{S}^3 \ltimes \mathbb{R}^3}(\mathbf{v}) = \begin{pmatrix} \mathbf{T}_{\mathbb{S}^3}(\Omega) & \mathbf{0} \\ \mathbf{C}_1(\Omega, \mathbf{U}) & \mathbf{T}_{\mathbb{S}^3}(\Omega) \end{pmatrix}, \tag{20}$$

where the function $\mathbf{C}_1(\Omega, \mathbf{U})$ is

$$\begin{aligned} \mathbf{C}_1(\Omega, \mathbf{U}) = & \frac{\cos \omega - 1}{\omega^2} \tilde{\mathbf{U}} + \frac{\omega - \sin \omega}{\omega^3} (\tilde{\mathbf{U}}\tilde{\Omega} + \tilde{\Omega}\tilde{\mathbf{U}}) + \\ & + \frac{2 - 2 \cos \omega - \omega \sin \omega}{\omega^4} (\Omega^\top \mathbf{U}) \tilde{\Omega} + \\ & - \frac{\omega(2 + \cos \omega) - 3 \sin \omega}{\omega^5} (\Omega^\top \mathbf{U}) \tilde{\Omega}^2. \end{aligned}$$

In the space of interest for beam analysis $G = (\mathbb{S}^3 \ltimes \mathbb{R}^3)^{N+1}$, the exponential map and the tangent operator are diagonal block matrices, where each block is of the form (18) and (20), respectively.

We would like to remark that due to the form of the expression of the operators, one has to consider alternatives for the singularities at $\Omega \approx \mathbf{0}$. In [15, Appendix B], a deep investigation is performed to obtain the expression needed to have the numerical method without loss of accuracy. In Table 2, we present a limited output of this study, focusing on the functions that are needed in the group $\mathbb{S}^3 \ltimes \mathbb{R}^3$.

2.4 Case study

As a proof of concept, we refer to the classical benchmark of the heavy top. We use the model as described in [3], where the orientation of the top is parameterized by rotation matrices in $SO(3)$.

The heavy top has a fixed point in the origin of the inertial frame and it is free to rotate about that point. The motion under the influence of gravity is described via the position of the center of mass in the inertial frame $\mathbf{x} \in \mathbb{R}^3$ and the orientation of the body using a rotation matrix $\mathbf{R} \in SO(3)$, i.e., the variable q in Eq. (5a)–(5c) is $q(t) =$

Table 2 Singular functions and their Taylor polynomials [15, Appendix B]

Function	ω threshold	Taylor polynomial
$\frac{\sin \omega}{\omega}$	$\omega < 10^{-4}$	$1 - \frac{\omega^2}{6}$
$\frac{\cos \omega - 1}{\omega^2}$	$\omega < 10^{-2}$	$-\frac{1}{2} + \frac{\omega^2}{24} - \frac{\omega^4}{720}$
$\frac{\omega - \sin \omega}{\omega^3}$	$\omega < 10^{-4}$	$\frac{1}{6} - \frac{\omega^2}{120} + \frac{\omega^4}{5040}$
$\frac{2 - 2 \cos \omega - \omega \sin \omega}{\omega^4}$	$\omega < 10^{-1}$	$\frac{1}{12} - \frac{\omega^2}{180} + \frac{\omega^4}{6720} - \frac{\omega^6}{453600}$
$\frac{\omega(2 + \cos \omega) - 3 \sin \omega}{\omega^5}$	$\omega < 10^{-1}$	$\frac{1}{60} - \frac{\omega^2}{1260} + \frac{\omega^4}{60480} - \frac{\omega^6}{4989600}$

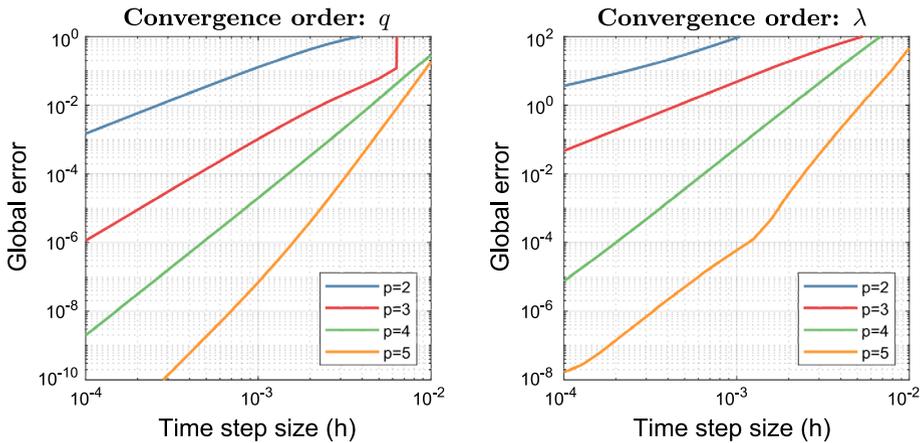


Fig. 1 Heavy top modeled in $SO(3) \times \mathbb{R}^3$. Convergence study for the variable at position level q (left plot) and for the Lagrange multipliers λ (right plot). (Color figure online)

$(\mathbf{R}(t), \mathbf{x}(t))$. The data, omitting physical units, include the center of mass in the body-attached frame $\mathbf{X} = (0, 1, 0)^\top$, the mass $m = 15.0$, the inertia tensor w.r.t. the center of mass $\mathbf{J} = \text{diag}(0.234375, 0.46875, 0.234375)$, and the constant gravitational acceleration vector $\gamma = (0, 0, -9.81)^\top$. The constraints at position level refer to the rotation about a fixed point, which can be modeled with the constraint $\Phi(q) = \mathbf{R}^\top \mathbf{x} - \mathbf{X}$. As an initial configuration, we set the rotation $\mathbf{R}(0) = \mathbf{I}_3$ and the angular velocity $\Omega(0) = (0, 150, -4.61538)^\top$ with initial position and translational velocity consistent with the holonomic constraints at position level and corresponding hidden constraints at velocity level. In Fig. 1 and Fig. 2, the proposed numerical methods are applied to the heavy-top benchmark. In the plots, one may observe that the expected order of convergence is preserved for the test cases in the Lie group setting. Moreover, the same order of convergence applies also to the Lagrange multipliers λ , right plots in Figs. 1 and 2. The experiment solves the system in its index-2 formulation, where the constraints are at the velocity level. A known limitation is the risk of linear growth of the residual in the holonomic constraints (*drift-off*) and it appears to be group dependent. One may observe in Fig. 3 that the choice of the Lie group determines the presence of the drift-off. When using a direct product $SO(3) \times \mathbb{R}^3$, the constraints at position level are not completely satisfied and the numerical solution diverges systematically from the manifold of constraints. On the contrary, for the semidirect product $SE(3) := SO(3) \ltimes \mathbb{R}^3$ the solution maintains the constraints both at position and velocity levels. This is in line with a detailed theoretical analysis for other Lie group integrators [3, Sect. 3.6]. The drift-off effect will be

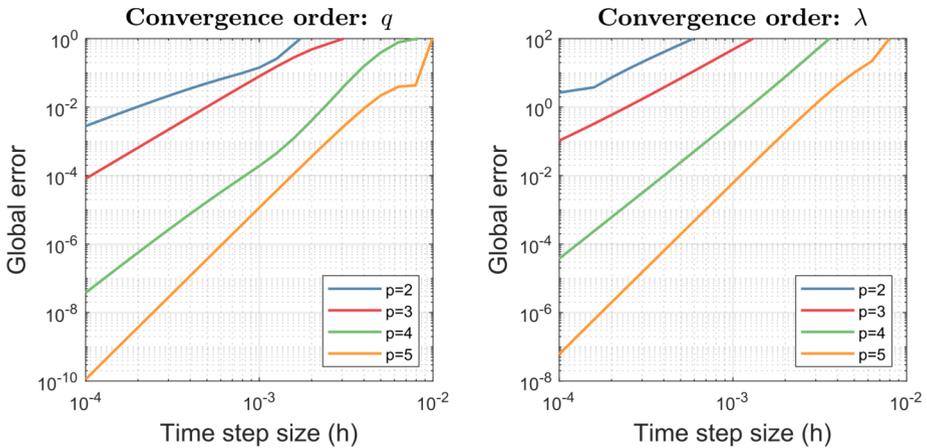


Fig. 2 Heavy top modeled in $SE(3)$. Convergence study for the variable at position level q (left plot) and for the Lagrange multipliers λ (right plot). (Color figure online)

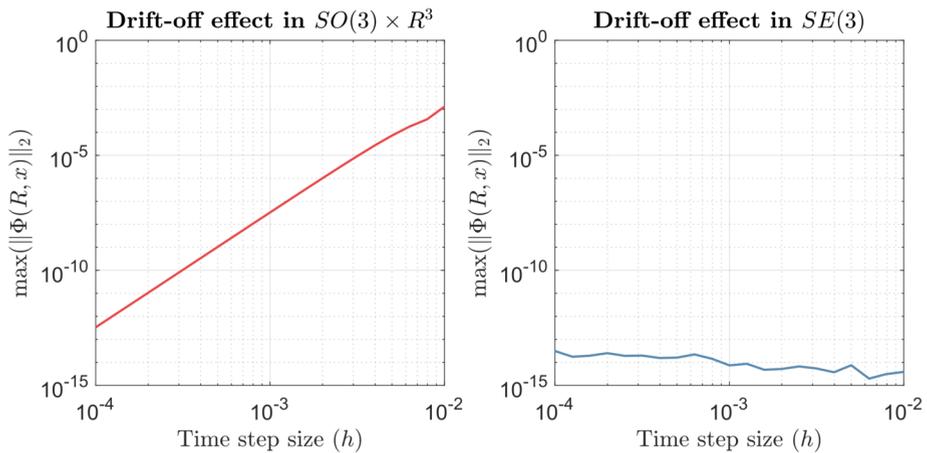


Fig. 3 Drift-off effect of the residual of the constraint at position level: $SO(3) \times \mathbb{R}^3$ (left plot) and $SE(3)$ (right plot)

further investigated in future research. It does not appear in the specific applications to beam dynamics we are going to show later in the paper. In a direct comparison with the Lie group DAE integrators that rely on the index-1 formulation of the equations of motion [31, 33], the reduced risk of drift in the index-2 formulation (5a)–(5c) is a clear benefit of the proposed method (12)–(14), see also [13, Theorem VII.2.1]. In the literature, the heavy-top benchmark is set in the direct or semidirect product of $SO(3)$ and \mathbb{R}^3 . For the setting presented in the current paper and for homogeneity with the following tests, we repeated the tests with unit quaternions, i.e., in the spaces $S^3 \times \mathbb{R}^3$ and $S^3 \ltimes \mathbb{R}^3$. In Figs. 4 to 6, we obtained qualitatively the same results, with a slightly larger error constant.

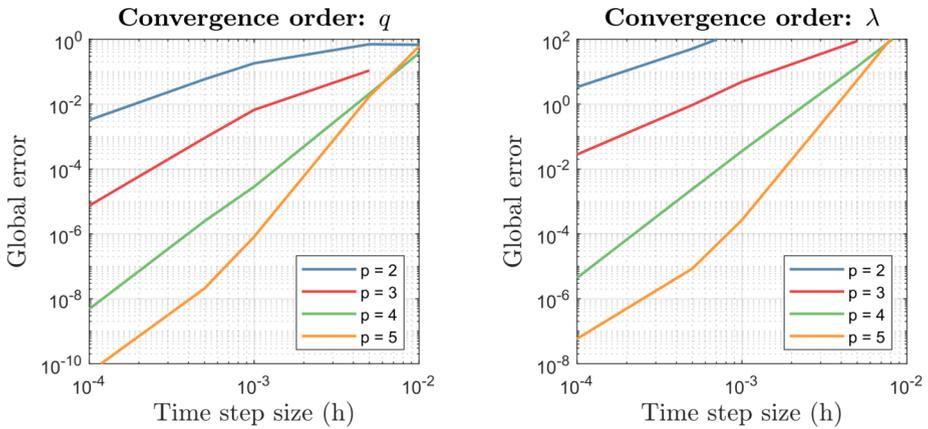


Fig. 4 Heavy top modeled in $\mathbb{S}^3 \times \mathbb{R}^3$. Convergence study for the variable at position level q (left plot) and for the Lagrange multipliers λ (right plot). (Color figure online)

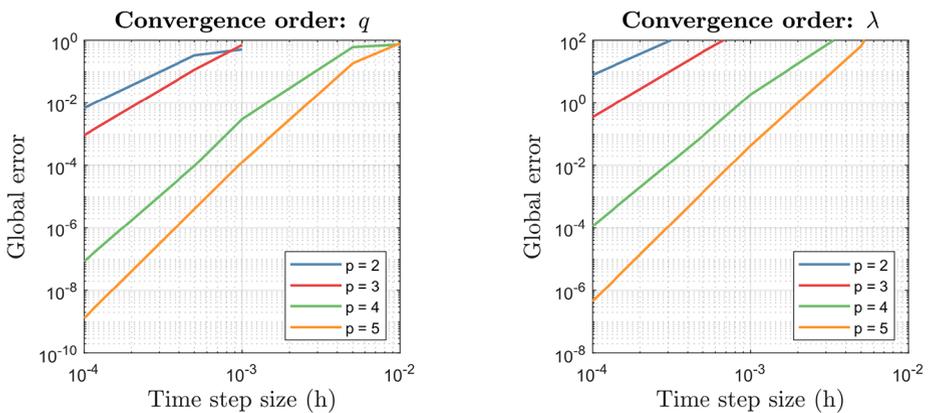


Fig. 5 Heavy top modeled in $\mathbb{S}^3 \times \mathbb{R}^3$. Convergence study for the variable at position level q (left plot) and for the Lagrange multipliers λ (right plot). (Color figure online)

3 Error-controlled variable time step-size

In Sect. 2, we introduced a modified half-explicit Runge–Kutta method for Lie group settings. In the current section, we are going to expand the study to variable time step-sizes. The advantages of using an error-controlled variable time step-size are the reduction of the computational time, obtaining at the same time a control over the error. In particular, when solving the equations with fixed time steps, we do not know how well the simulation is performing while it is running. For variable step sizes, we have to estimate the local error to establish a new time step-size. The theory behind the approach used here can be found in [14].

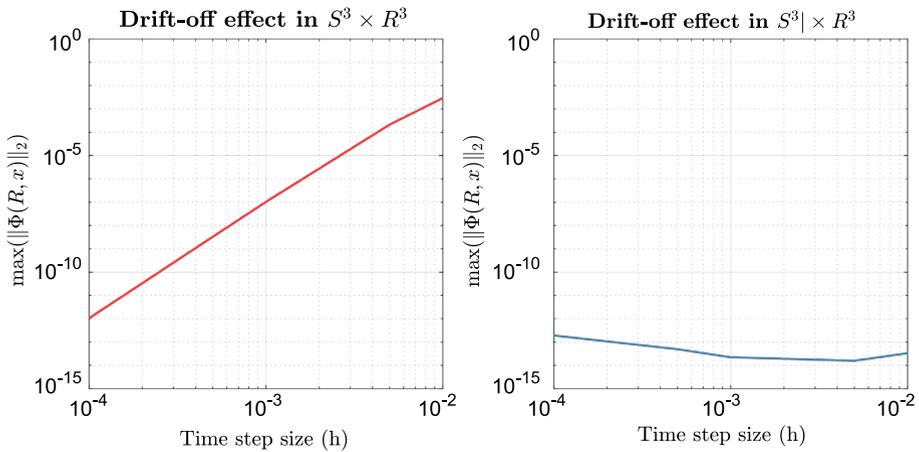


Fig. 6 Drift-off effect of the residual of the constraint at position level: $S^3 \times \mathbb{R}^3$ (left plot) and $S^3 \times \mathbb{R}^3$ (right plot)

3.1 Methodology

The main challenge is the estimation of the local error for the position variable. Since we are working in a Lie group setting, the position variable, indicated before by $q(t)$, cannot be used in sums and differences as an element of a linear space. The decision we make is to estimate the local error of the local parametrization $\theta_n(t)$. In this way, we are able to use the established theory of embedded Runge–Kutta formulas [14]. To implement it, we need to introduce a new variable, which does not lie in the Lie group, and for which we can perform operations of sums and difference: $\mathbf{y} = (\boldsymbol{\theta}^\top, \mathbf{v}^\top)$. Here, $\boldsymbol{\theta}$ and \mathbf{v} , as before, are the local parametrization of the Lie group and the velocities, respectively.

Local error estimate First, we introduce the embedded Runge–Kutta formulas. At the same time step, we are going to evaluate two numerical solutions obtained with two Runge–Kutta methods with the same stage vectors Q_{ni} , Θ_{ni} , \mathbf{V}_{ni} but different weights b_i , such that we obtain a first numerical approximation \mathbf{y}_1 of order p and a second numerical approximation $\hat{\mathbf{y}}_1$ of order \hat{p} . Usually, we have $\hat{p} = p - 1$ or $\hat{p} = p + 1$. The approximation \mathbf{y}_1 is used to continue the integration, while the approximation $\hat{\mathbf{y}}_1$ is used to evaluate the estimate of the local error. Given user-defined tolerances, the error indicator at time $t = t_n$ is

$$err = \sqrt{\frac{1}{m} \sum_{i=1}^m \left(\frac{y_{n,i} - \hat{y}_{n,i}}{Atol_i + \max(|y_{n-1,i}|, |y_{n,i}|) \cdot Rtol_i} \right)^2}, \tag{21}$$

where $m = 6(N + 1)$ is the number of components of the solution vectors, $Atol_i$, $Rtol_i$, ($i = 1, \dots, m$) are vector-valued user-defined tolerances.

In the present paper, we implement the variable time step-size for embedded half-explicit Runge–Kutta Lie group methods of the fifth order, using the scheme previously reported with $s = 6$, $\bar{s} = 7$. The coefficients for the embedded method are those that are known from the seminal work of Dormand and Prince [10]:

$$\hat{a}_{81} = 5179/57600, \hat{a}_{82} = 0, \hat{a}_{83} = 7571/16695, \hat{a}_{84} = 393/640,$$

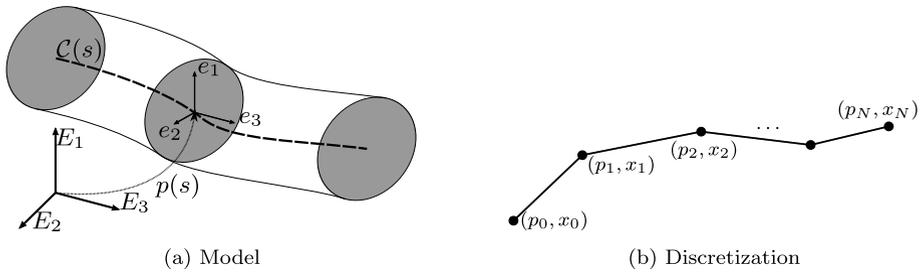


Fig. 7 Cosserat rod

$$\hat{a}_{85} = -92097/339200, \hat{a}_{86} = 187/2100, \hat{a}_{87} = 1/40.$$

New time step-size After the local error estimate is evaluated, the value of the error indicator err is compared with 1. If $err \leq 1$, the computed step is accepted and the computation will continue for the time step $t_{n+2} = t_{n+1} + h_{\text{new}}$. On the contrary, if $err > 1$, the computed step is rejected and a new computation for the time step $t_n \rightarrow t_{n+1}$ will be performed using a new time step-size, i.e., $t_{n+1} = t_n + h_{\text{new}}$. As one may observe, in both cases we need to compute a new value for the time step-size. We are going to evaluate the new time step-size in terms of the local error estimate, but we are going to use some multiplicative factors to avoid too large a change in the step size from one step to the next. In particular, given the previous time step-size h , the new one will be [14]

$$h_{\text{new}} = h \cdot \min(\text{facmax}, \max(\text{facmin}, \text{fac} \cdot (1/err)^{1/(\mu+1)})), \quad (22)$$

where $\mu = \min(p, \hat{p})$. We observe that for $err > 1$, $(1/err) < 1$ and the time step-size will surely diminish, while for $err \leq 1$ the new time step-size could still be smaller than the previous one due to the role of the multiplicative factor fac , which is set to be smaller than one. Typical values of these factors are $\text{fac} = 0.8$, $\text{facmin} = 0.2$, and $\text{facmax} = 5.0$, see [14].

4 Applications to geometrically exact beam model

4.1 Roll-up and flying spaghetti

The numerical experiments in the present section are performed for Cosserat beams with internal constraints. In detail, the geometrically exact model of a rod describes the rod as a curve, the centerline $\mathcal{C}(s)$, parameterized by the curvilinear abscissa s , along which we position the cross sections with a specific orientation dictated by the rotation matrix $R(s)$ or the unit quaternion $p(s)$ [17, 19], which indicates the rotation of a body-fixed reference system $\{e_1, e_2, e_3\}$ with respect to the inertial system $\{E_1, E_2, E_3\}$. In Fig. 7a, we may observe the model previously stated and in Fig. 7b the discretization in space is depicted. The discretization of the beam is motivated by the staggered grid in [19], but only from a computational point of view. In our practical applications, both the quaternions, indicating the orientation of the cross section, and the position coordinates are set on the nodes of the discretization. The methodology goes back to the work of Hante [15] and has been introduced in [17]. We

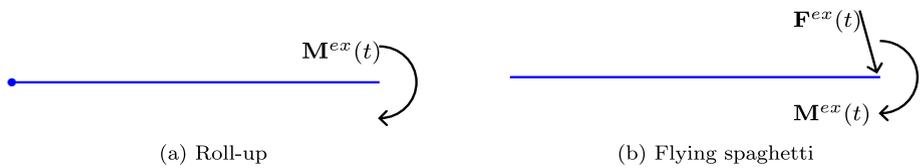


Fig. 8 Cosserat-rod test cases

evaluate in the midpoints of the edges the spatial derivatives of the configuration variables

$$\mathbf{w}_{i-1/2}(t) = \frac{1}{\Delta s_{i-1/2}} \widetilde{\log}(q_{i-1}^{-1}(t) \circ q_i(t)), \quad i = 1, \dots, N, \tag{23}$$

where N is the total number of discretization edges in which the beam is divided, $s \in [0, L]$ is the curvilinear abscissa, $q_i = (p_i, x_i) \in \mathbb{S}^3 \times \mathbb{R}^3$ and $\widetilde{\log}$ denotes the inverse of the exponential map with image in $\mathbb{R}^{6(N+1)}$. Finally, the trapezoidal and midpoint rule are used, respectively, to discretize the kinetic and the potential energy.

The validation of the numerical method through numerical experiments is performed with two test cases: the *roll-up maneuver* and the *flying spaghetti*. Both tests involve a constrained Cosserat rod as presented in [19], see also [17]. They go back to previous work of Gérardin and Cardona [11] and Simo and Vu-Quoc [29], respectively, and are fully described in [15]. The roll-up maneuver was first introduced as a static benchmark problem [11] and later its dynamics were studied [9].

The physical parameters involve the use of material parameters such as Young’s modulus and geometric parameters such as the area of the cross section or inertia values in different directions. In Table 3, we summarize these parameters, the initial conditions and the boundary conditions for both experiments. We define \mathbf{d} and a time-dependent input $g(t)$ as

$$\mathbf{d} = \begin{bmatrix} -6 \\ 8 \\ 0 \end{bmatrix}, \quad g(t) = \begin{cases} 200t/2.5, & t \in [0, 2.5] \\ 200(5 - t)/2.5, & t \in [2.5, 5] \\ 0, & t > 5. \end{cases} \tag{24}$$

The *roll-up maneuver* experiment considers a rod, fixed at one of the ends and subject to a moment on the other one, see Fig. 8a. The *flying spaghetti*, already introduced by [29], on the contrary, does not have any Dirichlet boundary conditions, having both ends free, but it has both a force and a momentum applied to one of its ends, see Fig. 8b. For both test cases, the constraints arise from the use of the Kirchhoff model [19]. In particular, the model avoids shear deformations of the cross sections, which remain perpendicular to the centerline. The constraint at position level is

$$\begin{bmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \end{bmatrix} \cdot \boldsymbol{\Gamma}(s, t) = \mathbf{0}, \tag{25}$$

where $\boldsymbol{\Gamma}$ is the material strain vector. Since we require (25), the first two components of $\boldsymbol{\Gamma}$ need to vanish.

For the experiments, we apply the half-explicit Runge–Kutta Lie group methods described in the sections above with coefficients that allow the methods to perform up to order five, see Table 1. In Fig. 9, we observe the expected order of convergence for the method applied to the roll-up maneuver test case with $N = 8$ edges and for the flying-spaghetti test

Table 3 Physical parameters in dimensionless form, see [15]

	Roll-up	Flying spaghetti
L	10	10
N	8	16
$\rho A $	1	1
$\rho\mathbf{J}$	diag(10, 10, 10)	diag(10, 10, 10)
$\mathbf{C}^{\mathbf{K}}$	diag(5×10^2 , 5×10^2 , 5×10^2)	diag(5×10^2 , 5×10^2 , 5×10^2)
$\mathbf{C}^{\dot{\mathbf{K}}}$	diag(10^2 , 10^2 , 10^2)	$\mathbf{0}$
\mathbf{C}^{Γ}	diag(10^4 , 10^4 , 10^4)	diag(10^4 , 10^4 , 10^4)
$\mathbf{C}^{\dot{\Gamma}}$	diag(10^2 , 10^2 , 10^2)	$\mathbf{0}$
$p_0(s)$	$\begin{bmatrix} \cos(\pi/4) \\ \sin(\pi/4)\mathbf{e}_3 \times \mathbf{e}_1 \end{bmatrix}$	$\begin{bmatrix} \cos(\pi/4) \\ \sin(\pi/4)\mathbf{e}_3 \times (\mathbf{d}/\ \mathbf{d}\ _2) \end{bmatrix}$
$\mathbf{x}_0(s)$	$s\mathbf{e}_1$	$\begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} + \frac{s}{L}\mathbf{d}$
fixed \mathbf{x}_0	yes	no
fixed p_0	yes	no
fixed \mathbf{x}_N	no	no
fixed p_N	no	no
$\mathbf{M}^{ex}(t)$	$2\pi \frac{\mathbf{C}^{\mathbf{K}}_{1,1}}{L}$	$\begin{bmatrix} 0 \\ -1/2 \\ -1 \end{bmatrix} g(t)$
$\mathbf{F}^{ex}(t)$	-	$\begin{bmatrix} 1/10 \\ 0 \\ 0 \end{bmatrix} g(t)$

case with $N = 16$ edges. Here, the reference solutions for the two sets of tests are evaluated with a finer time grid ($h_{\text{ref}} = 10^{-6}$), but maintaining the space grid fixed. We note that the convergence properties of the method in the classical setting are maintained in the Lie group setting. We now consider the solution of the two problems when using an algorithm for variable time step-size. Here, we first observe the behavior of the local error for fixed time step-size. In Fig. 10, the local error estimate $\|\hat{\mathbf{y}} - \mathbf{y}\|$ is plotted vs. time for various values of a fixed time step-size h . The left plot shows the local error estimate for the roll-up maneuver test and we note that it decreases while reaching a more stable configuration. On the other side, the flying-spaghetti plot shows more variations due to the nature of the movement of the rod in this experiment. What we observe is how the local error estimates change as a function of the fixed time step-size. In particular, given a fixed time step-size h , the simulation is then performed two more times, once with time step-size $2h$ and once with time step-size $h/2$. One may note that the local error estimate for the solution with time step-size h is about 2^{4+1} times larger than the local error for the solution with time step-size $h/2$, similarly when computing the quotient between the local error for $2h$ and h . In fact, the method we use to estimate the local error is of order four, and we know that

$$\text{err} = \mathcal{O}(h^{p+1}), \quad (26)$$

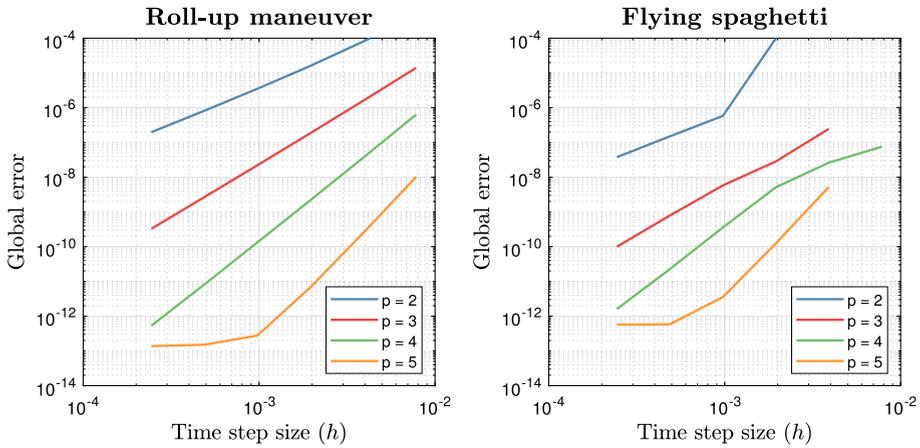


Fig. 9 Order of convergence (relative global error in the position variable vs. time step-size) for the roll-up maneuver test case (left) and the flying spaghetti (right) solved with half-explicit Runge–Kutta Lie group methods of order $p \leq 5$. (Color figure online)

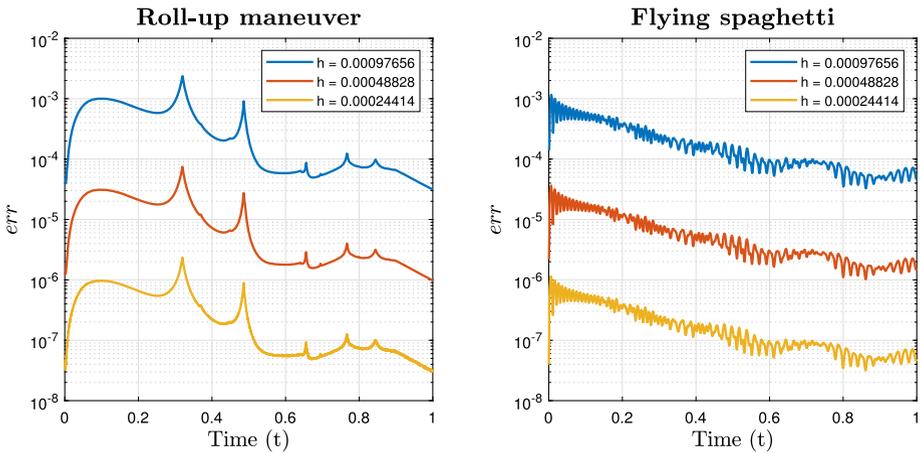


Fig. 10 Estimate of the local error indicator for fixed time step-size. On the x -axis there is the time t , on the y -axis the variable err in logarithmic scale. (Color figure online)

where p is the order of the method.

We implemented the fifth-order method with variable step size and in Fig. 11 we observe the results for both the roll-up maneuver and the flying-spaghetti example. In Fig. 11, the simulations run in the time interval $t \in [0, 5]$ and we note that after a first time interval in which the time step-size is comparable in size between the two test experiments, for $t > 1$ s the step size increases substantially for the roll-up maneuver and remains bounded for the flying spaghetti. The difference in the size of the time step could be explained by the nature of the test case itself: In the flying spaghetti there is no equilibrium state that could be reached, while for the roll-up maneuver the closed-ring position is an equilibrium point of the motion. The increase in the time step-size could then be associated with the achieved

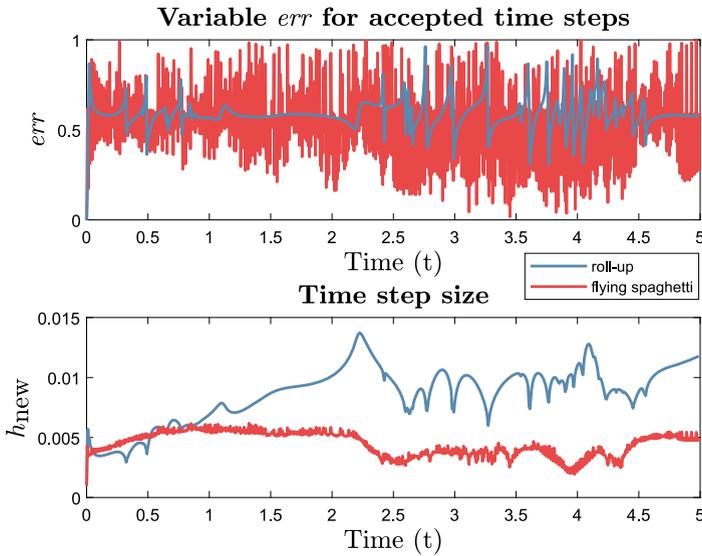
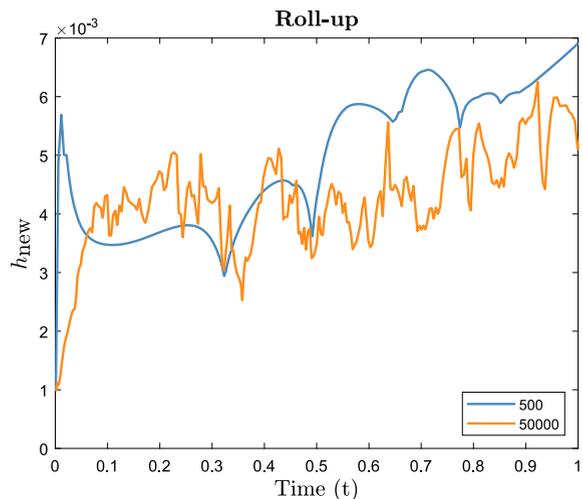


Fig. 11 The blue lines represent the test data for the roll-up test with $N = 8$ edges, the red lines for the flying spaghetti with $N = 16$ edges. The upper plot represents the trend in the variable err that is the estimate of the local error against the user-defined tolerance $Atol = 10^{-8}$ and $Rtol = 10^{-6}$, the lower plot is the new time step-size after each accepted step. (Color figure online)

Fig. 12 The blue line represents the solution for the roll-up test with bending stiffness $\mathbf{C}^{\mathbf{K}} = 5 \cdot 10^2$, the orange line represents the solution with bending stiffness $\mathbf{C}^{\mathbf{K}} = 5 \cdot 10^4$. (Color figure online)



equilibrium status. Other observations may regard the behavior in the change of time step-size as a function of the variable err . When err decreases we note a spike in the step-size sequence, since the error reaches lower values.

Figure 12 shows the difference in step-size history when increasing the bending stiffness for the roll-up maneuver test setup. As expected, the step size increases towards the end of the simulation, since the solution approaches a stationary state. The blue line correspond to a value of the stiffness $\mathbf{C}^{\mathbf{K}} = \text{diag}(5 \cdot 10^2, 5 \cdot 10^2, 5 \cdot 10^2)$, while the orange line corresponds

to $\mathbf{C}^K = \text{diag}(5 \cdot 10^4, 5 \cdot 10^4, 5 \cdot 10^4)$. We observe that a smaller value of the stiffness will make the numerical integration smoother, in the sense that the local error will have fewer oscillations and the time step-size will have less variation. We note that when increasing the stiffness, spikes in the local error will increase, which will require the decreasing of the time step by quite a large factor. Those differences are not unexpected, on the contrary, we know that stiffer systems may require smaller time step-sizes for their resolution if explicit integrators are used.

5 Conclusion

The time integrator we introduced in the present paper shows favorable properties: Having the structure of a half-explicit integrator, it does not need the use of a Newton–Raphson method to solve possible nonlinear systems of equations, and being designed to maintain the solution on the Lie group, it is advantageous for the evaluation of the solution of mechanical systems subject to large rotations. We observed that the application of the Lie group does not influence the stability behavior and the classical orders of convergence are preserved in the corresponding Lie group integrator.

On the other hand, we extended the study to the control of the local error. The implementation of a variable time step-size is not only helpful to reduce the computing time, but also to control the local error. The classical fifth-order explicit Runge–Kutta method of Dormand and Prince [10] has been made available to a constrained system on Lie groups.

Acknowledgements The numerical test results rely in part on open source software at <https://github.com/StHante> that Stefan Hante developed in his PhD project for testing implicit integrators for constrained systems on Lie groups.

Author contributions D.T. and M.A. wrote the main manuscript text based on previous research results of M.A. In addition, D.T. performed the numerical tests and prepared the figures. Both authors reviewed the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Arnold, M.: Half-explicit Runge-Kutta methods with explicit stages for differential-algebraic systems of index 2. *BIT Numer. Math.* **38**(3), 415–438 (1998)

2. Arnold, M., Brüls, O., Cardona, A.: Error analysis of generalized- α Lie group time integration methods for constrained mechanical systems. *Numer. Math.* **129**, 149–179 (2015)
3. Arnold, M., Cardona, A., Brüls, O.: A Lie algebra approach to Lie group time integration of constrained systems. In: *Structure-Preserving Integrators in Nonlinear Structural Dynamics and Flexible Multibody Dynamics*, pp. 91–158. Springer, Berlin (2016)
4. Betsch, P.: The discrete null space method for the energy consistent integration of constrained mechanical systems: part I: holonomic constraints. *Comput. Methods Appl. Mech. Eng.* **194**, 5159–5190 (2005)
5. Betsch, P., Siebert, R.: Rigid body dynamics in terms of quaternions: Hamiltonian formulation and conserving numerical integration. *Int. J. Numer. Methods Eng.* **79**, 444–473 (2009)
6. Brasey, V., Hairer, E.: Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2. *SIAM J. Numer. Anal.* **30**(2), 538–552 (1993)
7. Brüls, O., Cardona, A.: On the use of Lie group time integrators in multibody dynamics. *J. Comput. Nonlinear Dyn.* **5**(3), 031002 (2010)
8. Brüls, O., Cardona, A., Arnold, M.: Lie group generalized- α time integration of constrained flexible multibody systems. *Mech. Mach. Theory* **48**, 121–137 (2012)
9. Češarek, P., Saje, M., Zupan, D.: Kinematically exact curved and twisted strain-based beam. *Int. J. Solids Struct.* **49**(13), 1802–1817 (2012)
10. Dormand, J.R., Prince, P.J.: A family of embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **6**, 19–26 (1980)
11. Géradin, M., Cardona, A.: Kinematics and dynamics of rigid and flexible mechanisms using finite elements and quaternion algebra. *Comput. Mech.* **4**, 115–135 (1988)
12. Géradin, M., Cardona, A.: *Flexible Multibody Dynamics: A Finite Element Approach*. Wiley, New York (2001)
13. Hairer, E., Wanner, G.: *Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems*, 2nd edn. Springer, Berlin (1996)
14. Hairer, E., Nørsett, S.P., Wanner, G.: *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer, Berlin (1993)
15. Hante, S.: Geometric integration of a constrained Cosserat beam model. PhD thesis, Martin-Luther-Universität Halle-Wittenberg (March 2022)
16. Hante, S., Arnold, M.: RATTLie: a variational Lie group integration scheme for constrained mechanical systems. *J. Comput. Appl. Math.* **387**, 112492 (2021)
17. Hante, S., Tumiotto, D., Arnold, M.: A Lie group variational integration approach to the full discretization of a constrained geometrically exact Cosserat beam model. *Multibody Syst. Dyn.* **54**, 97–123 (2022)
18. Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie group methods. *Acta Numer.* **9**, 215–365 (2000)
19. Lang, H., Linn, J., Arnold, M.: Multibody dynamics simulation of geometrically exact Cosserat rods. *Multibody Syst. Dyn.* **25**, 285–312 (2011)
20. Linn, J.: Discrete Cosserat rod kinematics constructed on the basis of the difference geometry of framed curves—part I: discrete Cosserat curves on a staggered grid. *J. Elast.* **139**(2), 177–236 (2020)
21. Moler, C., Van Loan, C.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* **45**, 3–49 (2003)
22. Müller, A.: Screw and Lie group theory in multibody dynamics – recursive algorithms and equations of motion of tree-topology systems. *Multibody Syst. Dyn.* **42**, 219–248 (2018)
23. Müller, A., Terze, Z.: On the choice of configuration space for numerical Lie group integration of constrained rigid body systems. *J. Comput. Appl. Math.* **262**, 3–13 (2014)
24. Munthe-Kaas, H.: Runge-Kutta methods on Lie groups. *BIT Numer. Math.* **38**, 92–111 (1998)
25. Murua, A.: Partitioned half-explicit Runge–Kutta methods for differential-algebraic systems of index 2. *Computing* **59**, 43–61 (1997)
26. Park, J., Chung, W.K.: Geometric integration on Euclidean group with application to articulated multibody systems. *IEEE Trans. Robot.* **21**, 850–863 (2005)
27. Romero, I., Arnold, M.: Computing with rotations: algorithms and applications. In: *Encyclopedia of Computational Mechanics*, 2nd edn., pp. 1–27 (2017)
28. Shampine, L.F., Reichelt, M.W.: The Matlab ODE suite. *SIAM J. Sci. Comput.* **18**(1), 1–22 (1997)
29. Simo, J.C., Vu-Quoc, L.: On the dynamics in space of rods undergoing large motions – a geometrically exact approach. *Comput. Methods Appl. Mech. Eng.* **66**(2), 125–161 (1988)
30. Sonnevill, V.: A geometric local frame approach for flexible multibody systems. PhD thesis, Université de Liège (2015)
31. Terze, Z., Müller, A., Zlatar, D.: Lie group integration method for constrained multibody systems in state space. *Multibody Syst. Dyn.* **34**, 275–305 (2015)
32. Todesco, J., Brüls, O.: Highly accurate differentiation of the exponential map and its tangent operator. *Mech. Mach. Theory* **190**, 105451 (2023)

33. Zhou, P., Ren, H.: Stabilized explicit integrators for local parametrization in multi-rigid-body system dynamics. *ASME J. Comput. Nonlinear Dyn.* **17**, 101005 (11 pages) (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.