Momentum and near-energy conserving/decaying time integrator for beams with higher-order interpolation on SE(3)

Tomec, Jan; Jelenić, Gordan

Computer Methods in Applied Mechanics and Engineering 419 (2024) 116665

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Momentum and near-energy conserving/decaying time integrator for beams with higher-order interpolation on $SE(3)$ ☆

Jan Tomec, Gordan Jelenić *

University of Rijeka, Faculty of Civil Engineering, Radmile Matejčić 3, 51000, Rijeka, Croatia

ARTICLE INFO

Keywords:
Geometrically exact beam theory
Numerical time integration
Energy–momentum conservation/decay
Implicit $SE(3)$ interpolation
Lie midpoint rule

ABSTRACT

This study aims to develop a simple yet robust time-integration scheme for configuration-interpolated beam finite elements. Geometrically exact theory is employed to model the beams. It utilises the rotations which belong to a non-commutative Lie group and thus require special attention. The configuration is interpolated using a two-node $SE(3)$ interpolation or its generalised implicit variant which enables higher orders. Such interpolation treats position and orientation as a unit and a member of the $SE(3)$ Lie group. This unified approach allows for elegant mathematical manipulation. By adapting the Lie midpoint rule appropriately, it becomes possible to express energy changes in terms of inertial and internal forces, thus enabling the derivation of a momentum conserving and almost energy conserving time integration algorithm. The precision of energy conservation depends solely on the length of the finite elements. With further modification, this algorithm can also become an energy-decaying algorithm. Despite the configuration-dependent nature of the interpolation, the need to deal with its derivatives is avoided, which simplifies the implementation. The method is tested using 3D numerical example with finite rotations which confirms, that the method indeed has the conservation properties and is stable and robust.

1. Introduction

This paper presents a new momentum conserving and almost energy conserving algorithm for stable long-term numerical time integration of nonlinear Cosserat beams. The stability of a time-stepping algorithm for mechanical systems has long been associated with its ability to conserve linear and angular momentum and energy. The same principle can be applied to Cosserat beam problems where the geometrically-exact strains present nonlinearities to the system due to the non-commutative properties of the special Euclidean group of rigid-body motions.

As pointed out in [1], the classical Newmark integration schemes generally fail to conserve the total angular momentum and energy, which can lead to unexpected blow-up of energy after a certain time [2]. A variety of energy and/or momentum conserving algorithms for nonlinear elastic beams have been developed in the past [3–7], and more recently in [8], primarily using the interpolation of incremental motion. However, it is often useful to have the centreline configuration explicitly known via, what we call, configuration interpolation, so that the configuration is interpolated directly. One example is in beam contact mechanics [9].

☆ This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860124, from the Croatian science foundation under the research project FIMCOS (HRZZ IP 2018-01-1732) and from the University of Rijeka grant No uniri tehnic-18-248 1415.

* Corresponding author.
E-mail addresses: jan.tomec@uniri.hr (J. Tomec), gordan.jelenic@uniri.hr (G. Jelenić).

https://doi.org/10.1016/j.cma.2023.116665
Received 24 July 2023; Received in revised form 28 October 2023; Accepted 22 November 2023
Available online 5 December 2023

This prompts the development of a new energy–momentum conservative algorithm that allows configuration-based interpolation, as opposed to the interpolation of incremental motion.

A geometrically-exact beam is characterised by the position of its centreline to which an oriented rigid cross-section is attached. As such, the beam configuration may be considered as a member of the special Euclidean group $SE(3)$. Traditionally, the position and the orientation have often been considered separately [2,3,10–12]. Recent developments [5,13–15] have shown advantages of treating the two fields together using a linked position-orientation interpolation with homogeneous configuration update and equilibrium equation.

Bottasso and Borri have in [5] transported the equilibrium equation to the so-called fixed-pole frame. It acts as a natural counterpart to the material frame since the two are mathematically left- and right-invariant vector fields of the Lie group $SE(3)$. It serves as the basis for the derivation of an energy–momentum conservative beam finite element. More precisely, the element algorithmically conserves the generalised momentum in the fixed-pole frame, which is equivalent to conserving the total generalised momentum in the inertial frame. The conservation is achieved using a midpoint integration rule. The element uses incremental interpolation, which is not strain invariant.

Strain invariant interpolation has been developed in [2] on the $\mathbb{R}^3 \times SO(3)$ manifold and later generalised for the $SE(3)$ group in [13]. To achieve this, the so-called motion approach has been employed where the linear and angular motion are treated as a unit. This leads also to unified configuration update and equilibrium equation. In [14], the principle has been extended to higher-order elements by using implicit interpolation scheme, introduced for rotations in [16]. However, it has been since noted in [15] that the implicit interpolation scheme leads to complicated expressions of inertial terms due to configuration-dependent interpolation. By configuration-dependent interpolation we mean that the resulting interpolation depends on the current state of the beam. The temporal derivatives thus include not only the terms due to nodal values, but also terms due to interpolation. As a result, an independent velocity field that is linked to the actual instantaneous velocity through a weak integral relationship is incorporated in [15].

These developments have motivated us to apply the midpoint-rule in the motion approach. By implementing suitable algorithmic midpoint inertial and internal forces, we have been able to formulate a momentum conservative time-stepping scheme with very good energy-preserving characteristics. Additionally, this method avoids the complicated second derivatives with respect to time, resulting in more straightforward expressions that, in turn, lead to faster computations.

This paper is structured as follows: in Section 2 we familiarise the reader with important Lie group properties for the groups $SO(3)$ and $SE(3)$. In Section 3, we summarise the geometrically-exact-beam theory and develop a weak formulation. Afterwards, we discuss time integration, the conservation laws, and the resulting algorithm in Section 4. The finite element is formulated in Section 5 and tested in Section 6.

2. Lie groups $SO(3)$ and $SE(3)$ and the corresponding Lie algebras $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$

Mathematically, the geometry of a beam can be represented using Lie groups. A Lie group is a differentiable manifold with an additional composition rule between the group members. In the context of a beam, the orientation of its cross-section can be described as a member of the $SO(3)$ group (a group of 3D rotations), whilst its position can be described as a member of the 3D linear space, $\mathbb{R}^3$. However, the position and the orientation can also be treated as a single member of the special Euclidean group $SE(3)$.

Each Lie group also has its corresponding Lie algebra, which is its tangent space at the identity. Tangent space is linear and thus isomorphic to the Euclidean space. We denote the mapping from a Euclidean space to a Lie algebra using a hat operator, which for the cases under consideration maps a 3D vector to Lie algebra $\mathfrak{so}(3)$ and a 6D vector to Lie algebra $\mathfrak{se}(3)$:

$$\hat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3) \wedge \mathbb{R}^6 \to \mathfrak{se}(3). \tag{1}$$

The matrix representation of the elements of the two Lie algebras can be written as

$$\hat{\boldsymbol{\omega}} = \widehat{\begin{Bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{Bmatrix}} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix}, \quad \hat{\boldsymbol{x}} = \widehat{\begin{Bmatrix}\boldsymbol{u}\\ \boldsymbol{\omega}\end{Bmatrix}} = \begin{bmatrix} \hat{\boldsymbol{\omega}} & \boldsymbol{u} \\ \boldsymbol{0}^T & 0 \end{bmatrix}, \tag{2}$$

where $\boldsymbol{u}, \boldsymbol{\omega} \in \mathbb{R}^3$ and $\boldsymbol{x} \in \mathbb{R}^6$.

An adjoint representation of a Lie group G is an alternative representation of a group element which transforms an element of the group's Lie algebra $\mathfrak{g}$ regarded as an element of a linear space. If $h \in G$ and $x \in \mathbb{R}^n$, where n is an appropriate dimension so that $\hat{x} \in \mathfrak{g}$, then

$$\mathrm{Ad}_h : \mathfrak{g} \to \mathfrak{g}; \quad \mathrm{Ad}_h(\hat{x}) = h\hat{x}h^{-1} \tag{3}$$

and using a linear tilde operator $\tilde{\cdot}$

$$\mathrm{Ad}_h(\hat{x}) = \widehat{\tilde{h}x}. \tag{4}$$

Specifically, the adjoint representation of the $SO(3)$ group maps to itself, i.e. $\tilde{h} = h$ (specifically, $\tilde{\mathbf{\Lambda}} = \mathbf{\Lambda}$, where $\mathbf{\Lambda} \in SO(3)$), whilst for the $SE(3)$ group with $\mathbf{H} = \begin{bmatrix} \mathbf{\Lambda} & \boldsymbol{r} \\ \mathbf{0}^T & 1 \end{bmatrix}$, where $\boldsymbol{r} \in \mathbb{R}^3$, $\tilde{\mathbf{H}}$ is defined using the matrix representation as

$$\widetilde{\begin{bmatrix} \mathbf{\Lambda} & \boldsymbol{r} \\ \mathbf{0}^T & 1 \end{bmatrix}} = \begin{bmatrix} \mathbf{\Lambda} & \hat{\boldsymbol{r}}\mathbf{\Lambda} \\ \mathbf{0} & \mathbf{\Lambda} \end{bmatrix}. \tag{5}$$

Adjoint representation of a Lie algebra is defined using a Lie bracket. For two elements x, y of the Lie algebra $\mathfrak{g}$ with a Lie bracket defined as $[x, y] = xy - yx$, the adjoint representation of $\mathfrak{g}$ is

$$\mathrm{ad}_x : \mathfrak{g} \to \mathfrak{g}; \quad \mathrm{ad}_x(y) = [x, y]. \tag{6}$$

Since Lie algebra is isomorphic to linear space, its adjoint representation can also be expressed as a linear operator $\widehat{\cdot}$. The following relationship exists

$$\mathrm{ad}_x(y) = \widehat{\tilde{x}}y. \tag{7}$$

Specifically, the adjoint representation of the $\mathfrak{so}(3)$ Lie algebra maps to itself, i.e. $\tilde{x} = x$, whilst for the $\mathfrak{se}(3)$ Lie algebra, it is defined using the matrix representation as

$$\widetilde{\begin{bmatrix} \hat{\boldsymbol{\omega}} & \boldsymbol{u} \\ \mathbf{0}^T & 0 \end{bmatrix}} = \begin{bmatrix} \hat{\boldsymbol{\omega}} & \hat{\boldsymbol{u}} \\ \mathbf{0} & \hat{\boldsymbol{\omega}} \end{bmatrix}. \tag{8}$$

A member of a Lie algebra can be mapped to its Lie group via exponentiation

$$\exp : \mathfrak{so}(3) \to SO(3) \wedge \mathfrak{se}(3) \to SE(3). \tag{9}$$

Exponential map can be expressed through a series as

$$\exp(\mathbf{X}) = \sum_{i=0}^{\infty} \frac{\mathbf{X}^i}{i!}, \tag{10}$$

for any square matrix $\mathbf{X}$. For $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$, its differential is expressible as

$$\mathrm{d}\exp(\hat{\boldsymbol{x}}) = \exp(\hat{\boldsymbol{x}})\widehat{\mathbf{T}(\tilde{\boldsymbol{x}})\mathrm{d}\boldsymbol{x}}, \tag{11}$$

where $\mathbf{T}$ is a tangential map with the following series expansion

$$\mathbf{T}(\tilde{\boldsymbol{x}}) = \sum_{i=0}^{\infty} (-1)^i \frac{\tilde{\boldsymbol{x}}^i}{(i+1)!}. \tag{12}$$

Certain groups have also a closed form solution of the series, among which are also the groups in question. For $SO(3)$, the exponential map is analytically expressible as

$$\exp(\hat{\boldsymbol{\omega}}) = \mathbf{I} + \frac{\sin(\omega)}{\omega}\hat{\boldsymbol{\omega}} + \frac{1 - \cos(\omega)}{\omega^2}\hat{\boldsymbol{\omega}}^2, \qquad \omega = \sqrt{\boldsymbol{\omega}^T\boldsymbol{\omega}}, \tag{13}$$

whilst the tangential map in $SO(3)$ can be expressed as

$$\mathbf{T}(\tilde{\boldsymbol{\omega}}) = \mathbf{I} - \frac{1 - \cos(\omega)}{\omega^2}\hat{\boldsymbol{\omega}} + \frac{1 - \sin(\omega)/\omega}{\omega^2}\hat{\boldsymbol{\omega}}^2. \tag{14}$$

The maps in $SE(3)$ read

$$\exp(\hat{\boldsymbol{x}}) = \begin{bmatrix} \exp(\hat{\boldsymbol{\omega}}) & \mathbf{T}(\tilde{\boldsymbol{\omega}})^T\boldsymbol{u} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{15}$$

$$\mathbf{T}(\tilde{\boldsymbol{x}}) = \begin{bmatrix} \mathbf{T}(\tilde{\boldsymbol{\omega}}) & \mathbf{T}_{U\Omega}(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\omega}}) \\ \mathbf{0} & \mathbf{T}(\tilde{\boldsymbol{\omega}}) \end{bmatrix}, \tag{16}$$

where

$$\begin{aligned} \mathbf{T}_{U\Omega}(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{\omega}}) = &- \frac{1 - \cos(\omega)}{\omega^2}\hat{\boldsymbol{u}} + \frac{1 - \sin(\omega)/\omega}{\omega^2}(\hat{\boldsymbol{u}}\hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\omega}}\hat{\boldsymbol{u}}) \\ &+ \frac{\boldsymbol{u}^T\boldsymbol{\omega}}{\omega^2}\left[\left(2\frac{1 - \cos(\omega)}{\omega^2} - \frac{\sin(\omega)}{\omega}\right)\hat{\boldsymbol{\omega}} + \left(\frac{1 - \cos(\omega)}{\omega^2} - 3\frac{1 - \sin(\omega)/\omega}{\omega^2}\right)\hat{\boldsymbol{\omega}}^2\right]. \end{aligned} \tag{17}$$

A logarithmic map is the inverse function of an exponential map. Analytical expressions exist also for these and can be found in literature, e.g. [13,17], as can also the inverses of the tangent maps.

3. Geometrically exact beam theory

A beam is characterised by the position of its centreline to which an oriented rigid cross-section is attached. As such, a beam can be placed on the special Euclidean group $SE(3)$ by parametrising its configuration as

$$\mathbf{H} : \mathbb{R} \times [0, L] \subset \mathbb{R} \to SE(3); (t, s) \mapsto \mathbf{H}(t, s). \tag{18}$$

Traditionally, the position and the orientation have often been considered separately

$$\boldsymbol{r} \in \mathbb{R}^3, \quad \boldsymbol{\Lambda} \in SO(3). \tag{19}$$

In $SE(3)$, its configuration can be represented using a configuration matrix, expressed as

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{\Lambda} & \boldsymbol{r} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{20}$$

Differentials of $\mathbf{H}$ can be defined using left- and right-invariant vector fields, which are associated with the fixed-pole and material frame of reference, respectively. The following uses dot and prime notation for temporal and spatial derivatives, respectively. For example, velocity in the material frame can be defined as

$$\dot{\mathbf{H}} = \mathbf{H}\hat{\boldsymbol{v}}. \tag{21}$$

Velocity $\boldsymbol{v} \in \mathbb{R}^6$ represents a combined vector of linear and angular velocity. It should be noted, that the operations in the adjoint representation are analogous, i.e. $\dot{\tilde{\mathbf{H}}} = \tilde{\mathbf{H}}\tilde{\boldsymbol{v}}$. Similarly goes for spatial derivative

$$\mathbf{H}' = \mathbf{H}\hat{\boldsymbol{e}}, \tag{22}$$

where $\boldsymbol{e} \in \mathbb{R}^6$ represents the spatial derivative of the configuration, and also $\tilde{\mathbf{H}}' = \tilde{\mathbf{H}}\tilde{\boldsymbol{e}}$. The strain $\boldsymbol{\varepsilon} \in \mathbb{R}^6$ follows as

$$\boldsymbol{\varepsilon} = \boldsymbol{e} - \boldsymbol{e}_0 \tag{23}$$

with $\boldsymbol{e}_0$ being the spatial derivative of the initial state. The strain is a Biot-like six-dimensional vector expressed in the material frame where the linear component represent the tensile and the shear stresses in the direction of the cross-section axes, while the angular component represents the torsion and the bending curvatures about the cross-section axes.

It is worth noting that derivatives are non-commutative on a Lie algebra. Let us assume that we have a general derivative, for example wrt. time or arc-length coordinate denoted as $\mathrm{d}\mathbf{H} = \mathbf{H}\widehat{\mathrm{d}\boldsymbol{h}}$, with $\mathrm{d}\boldsymbol{h} \in \mathbb{R}^6$ and a different derivative, for example a variation $\delta\mathbf{H} = \mathbf{H}\widehat{\delta\boldsymbol{h}}$, with $\delta\boldsymbol{h} \in \mathbb{R}^6$, then $\delta(\mathrm{d}\boldsymbol{h}) \neq \mathrm{d}(\delta\boldsymbol{h})$ even though $\delta(\mathrm{d}\mathbf{H}) = \mathrm{d}(\delta\mathbf{H})$. To derive a relation between them we first note

$$\delta(\mathrm{d}\mathbf{H}) = \mathrm{d}(\delta\mathbf{H}) \quad \Rightarrow \quad \mathbf{H}\widehat{\delta\boldsymbol{h}}\widehat{\mathrm{d}\boldsymbol{h}} + \mathbf{H}\delta(\widehat{\mathrm{d}\boldsymbol{h}}) = \mathbf{H}\widehat{\mathrm{d}\boldsymbol{h}}\widehat{\delta\boldsymbol{h}} + \mathbf{H}\mathrm{d}(\widehat{\delta\boldsymbol{h}}) \tag{24}$$

After reducing the equation by multiplying from the left with $\mathbf{H}^{-1}$, the expression can be simplified to

$$\delta(\mathrm{d}\boldsymbol{h}) = \widetilde{\mathrm{d}\boldsymbol{h}}\delta\boldsymbol{h} + \mathrm{d}(\delta\boldsymbol{h}). \tag{25}$$

3.1. Potential and kinetic energy

While the utilised beam model is geometrically nonlinear and allows finite deformations, material nonlinearities are not supported by the formulation. The following developments thus assume that the material is linear elastic.

For a linear-elastic material, the kinetic and the potential energy of a beam are defined as [13]

$$EK = \frac{1}{2}\int_L \boldsymbol{v}^T\mathbf{M}\boldsymbol{v}\mathrm{d}s, \tag{26}$$

$$EP = \frac{1}{2}\int_L \boldsymbol{\varepsilon}^T\mathbf{K}\boldsymbol{\varepsilon}\mathrm{d}s. \tag{27}$$

Matrices $\mathbf{M}$ and $\mathbf{K}$ hold material data about stiffness and inertial properties of the cross-section. Usually they are defined around the principal axes of the cross-section as $\mathbf{M} = \mathrm{diag}(\rho A, \rho A, \rho A, \rho(I_1 + I_2), \rho I_1, \rho I_2)$ and $\mathbf{K} = \mathrm{diag}(EA, GA_1, GA_2, GI_t, EI_1, EI_2)$. Material parameters E and G are Young's and shear moduli, while the rest are the geometric properties of the cross-section:

- A... area,
- A_i... respective shear area (area multiplied by a shear-correction coefficient dependent on the cross-section shape),
- J_t... torsional coefficient dependent on the cross-section size and shape,
- I_i... second moment of area around the ith principal axis.

Variations of the kinetic and potential energy can now be computed. To ease the derivation we first resolve the variation of the strain using (23) and (25):

$$\delta\boldsymbol{\varepsilon} = \delta(\boldsymbol{e} - \boldsymbol{e}_0) = \delta\boldsymbol{e} = \tilde{\boldsymbol{e}}\delta\boldsymbol{h} + (\delta\boldsymbol{h})' \tag{28}$$

Eqs. (26) and (27) are varied with the help of derivatives (21) and (22) while acknowledging the relationship between mixed derivatives (25)

$$\begin{aligned} \delta EK &= \int_L \delta\boldsymbol{v}^T\mathbf{M}\boldsymbol{v}\mathrm{d}s \\ &= \int_L (\tilde{\boldsymbol{v}}\delta\boldsymbol{h} + \dot{\delta\boldsymbol{h}})^T\mathbf{M}\boldsymbol{v}\mathrm{d}s, \end{aligned} \tag{29}$$

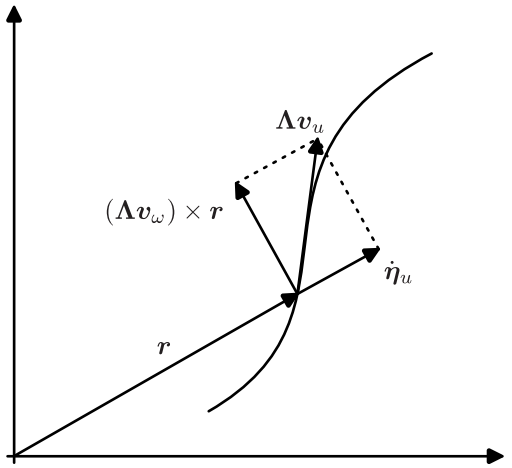


Fig. 1. Spatial linear velocity Λv_u, fixed-pole linear velocity $\dot{\eta}_u$ and the velocity due to change of orientation of the material particle $(\Lambda v_\omega) \times r$.

$$\begin{aligned} \delta EP &= \int_L \delta\varepsilon^T \mathbf{K}\varepsilon \,\mathrm{d}s \\ &= \int_L (\tilde{e}\delta h + \delta h')^T \mathbf{K}\varepsilon \,\mathrm{d}s \\ &= \left[\delta h^T \mathbf{K}\varepsilon\right]_0^L + \int_L \delta h^T \tilde{e}^T \mathbf{K}\varepsilon - \delta h^T \mathbf{K}\varepsilon' \,\mathrm{d}s. \end{aligned} \tag{30}$$

Virtual work of external forces in material frame can be expressed as

$$\delta W = \delta h_0^T \overline{P}_0 + \delta h_L^T \overline{P}_L + \int_L \delta h^T \overline{q} \,\mathrm{d}s, \tag{31}$$

where $\overline{q} \in \mathbb{R}^6$ is the distributed external load and $\overline{P}_0, \overline{P}_L \in \mathbb{R}^6$ are the concentrated external loads at the boundaries, while δh_0 and δh_L are the values of δh evaluated at the boundaries, all expressed in the material frame.

The equation of motion can be derived using Hamilton's principle, which asserts that the action integral within an arbitrary time interval $[t_a, t_b] \subset \mathbb{R}$, remains unchanged (i.e. stationary) for the actual path taken by the system between t_a and t_b:

$$\int_{t_a}^{t_b} (\delta EK - \delta EP + \delta W) \,\mathrm{d}t = 0. \tag{32}$$

Utilising the result from (29) and *per-partes* integration allows us to rewrite the inertial term in (32) as

$$\int_{t_a}^{t_b} \delta EK \,\mathrm{d}t = \left[\int_L \delta h^T \mathbf{M} v \,\mathrm{d}s\right]_{t_a}^{t_b} - \int_{t_a}^{t_b} \int_L \delta h^T \mathbf{M}\dot{v} - \delta h^T \tilde{v}^T \mathbf{M} v \,\mathrm{d}s \,\mathrm{d}t.$$

Since the variations are fixed at t_a and t_b, the first term on the right hand side vanishes [13]. Substituting this result, as well as (30) and (31) in (32), and recognising integration over an arbitrary time segment between t_a and t_b, yields

$$-\delta h_0^T \overline{P}_0 - \delta h_L^T \overline{P}_L + \left[\delta h^T \mathbf{K}\varepsilon\right]_0^L + \int_L \delta h^T \left(\mathbf{M}\dot{v} - \tilde{v}^T \mathbf{M} v - \mathbf{K}\varepsilon' + \tilde{e}^T \mathbf{K}\varepsilon - \overline{q}\right) \mathrm{d}s = 0. \tag{33}$$

Variations can be equivalently expressed also in the fixed-pole frame as

$$\delta \mathbf{H} = \widehat{\delta\eta}\mathbf{H}, \tag{34}$$

where $\delta\eta \in \mathbb{R}^6$. They are related to material frame variations via

$$\delta\eta = \tilde{\mathbf{H}}\delta h. \tag{35}$$

Remark 1. The relationship between the material and the fixed-pole variations in (35) may be understood by illustrating its expression over an infinitesimal time segment in which case those variations turn into respective velocities

$$\dot{\eta} = \tilde{\mathbf{H}}\dot{h},$$

where $\dot{h}$ is the vector of material velocities $v^T = \{v_u^T \quad v_\omega^T\}$, with v_u and v_ω being the respective linear and angular material velocities. Likewise, the fixed-pole velocity vector $\dot{\eta}^T = \{\dot{\eta}_u^T \quad \dot{\eta}_\omega^T\}$ consist of the linear fixed-pole velocity $\dot{\eta}_u$ and the angular

fixed-pole velocity $\dot{\boldsymbol{\eta}}_\omega$. The above relationship thus provides

$$\dot{\boldsymbol{\eta}}_u = \boldsymbol{\Lambda}\boldsymbol{v}_u + \hat{\boldsymbol{r}}\boldsymbol{\Lambda}\boldsymbol{v}_\omega = \boldsymbol{\Lambda}\boldsymbol{v}_u - \widehat{\boldsymbol{\Lambda}\boldsymbol{v}_\omega}\boldsymbol{r}$$
$$\dot{\boldsymbol{\eta}}_\omega = \boldsymbol{\Lambda}\boldsymbol{v}_\omega$$

indicating that the fixed-pole angular velocity is in fact equal to the spatial angular velocity $\boldsymbol{\Lambda}\boldsymbol{v}_u$ obtained by push-forward. The fixed-pole linear velocity differs from the spatial linear velocity owing to the presence of the term $\dot{\boldsymbol{\eta}}_\omega \times \boldsymbol{r}$, which represents the part of the linear velocity caused by the change of orientation of the material particle. The fixed-pole linear velocity thus represents the part of the spatial velocity vector from which the velocity due to the change in the particle orientation has been extracted. A simple 2D illustration is provided in Fig. 1. Likewise, the fixed-pole variation $\delta\boldsymbol{\eta}$ consists of a linear and an angular part, of which the latter is the same as the spatial angular variation, while the former is obtained from the spatial linear variation by extracting the variation due to the change of orientation of the material particle.

If we define the fixed-pole momentum $\boldsymbol{p} \in \mathbb{R}^6$ as

$$\boldsymbol{p} = \tilde{\mathbf{H}}^{-T}\mathbf{M}\boldsymbol{v}, \tag{36}$$

the fixed-pole internal forces $\boldsymbol{f} \in \mathbb{R}^6$ as

$$\boldsymbol{f} = \tilde{\mathbf{H}}^{-T}\mathbf{K}\boldsymbol{\varepsilon} \tag{37}$$

and the fixed-pole external distributed loads $\boldsymbol{q} \in \mathbb{R}^6$ and concentrated forces $\boldsymbol{P} \in \mathbb{R}^6$ as

$$\boldsymbol{q} = \tilde{\mathbf{H}}^{-T}\overline{\boldsymbol{q}}, \quad \boldsymbol{P} = \tilde{\mathbf{H}}^{-T}\overline{\boldsymbol{P}} \tag{38}$$

to obtain the following auxiliary results

- from (35) and (38) $\delta\boldsymbol{h}^T\overline{\boldsymbol{P}} = \delta\boldsymbol{\eta}^T\boldsymbol{P}$ and $\delta\boldsymbol{h}^T\overline{\boldsymbol{q}} = \delta\boldsymbol{\eta}^T\boldsymbol{q}$,
- from (35) and (37) $\delta\boldsymbol{h}^T\mathbf{K}\boldsymbol{\varepsilon} = \delta\boldsymbol{\eta}^T\boldsymbol{f}$,
- from (21), (35) and (36) $\delta\boldsymbol{\eta}^T\dot{\boldsymbol{p}} = \delta\boldsymbol{h}^T\tilde{\mathbf{H}}^T\left(\tilde{\mathbf{H}}^{-T}\mathbf{M}\dot{\boldsymbol{v}} - \tilde{\mathbf{H}}^{-T}\tilde{\boldsymbol{v}}^T\tilde{\mathbf{H}}^T\tilde{\mathbf{H}}^{-T}\mathbf{M}\boldsymbol{v}\right) = \delta\boldsymbol{h}^T\left(\mathbf{M}\dot{\boldsymbol{v}} - \tilde{\boldsymbol{v}}^T\mathbf{M}\boldsymbol{v}\right)$,
- from (22), (35) and (37) $\delta\boldsymbol{\eta}^T\boldsymbol{f}' = \delta\boldsymbol{h}^T\tilde{\mathbf{H}}^T\left(\tilde{\mathbf{H}}^{-T}\mathbf{K}\boldsymbol{\varepsilon}' - \tilde{\mathbf{H}}^{-T}\tilde{\boldsymbol{e}}^T\tilde{\mathbf{H}}^T\tilde{\mathbf{H}}^{-T}\mathbf{K}\boldsymbol{\varepsilon}\right) = \delta\boldsymbol{h}^T\left(\mathbf{K}\boldsymbol{\varepsilon}' - \tilde{\boldsymbol{e}}^T\mathbf{K}\boldsymbol{\varepsilon}\right)$,

it can be easily shown that the equilibrium equation in the material frame (33) reduces to

$$-\delta\boldsymbol{\eta}_0^T\boldsymbol{P}_0 - \delta\boldsymbol{\eta}_L^T\boldsymbol{P}_L + \left[\delta\boldsymbol{\eta}^T\boldsymbol{f}\right]_0^L + \int_L \delta\boldsymbol{\eta}^T\left(\dot{\boldsymbol{p}} - \boldsymbol{f}' - \boldsymbol{q}\right)\mathrm{d}s = 0. \tag{39}$$

Integration *per partes* leads to an alternative expression for the weak formulation in the fixed-pole frame

$$-\delta\boldsymbol{\eta}_0^T\boldsymbol{P}_0 - \delta\boldsymbol{\eta}_L^T\boldsymbol{P}_L + \int_L \delta\boldsymbol{\eta}^T\left(\dot{\boldsymbol{p}} - \boldsymbol{q}\right) + \delta\boldsymbol{\eta}'^T\boldsymbol{f}\,\mathrm{d}s = 0. \tag{40}$$

3.2. Spatial discretisation

The domain of a beam is now subdivided into N_{ele} finite elements with N_{nod} total global nodes (i.e. not involving the inner nodes in higher-order elements) in the mesh and the weak formulation (40) becomes

$$-\sum_{i=1}^{N_{\mathrm{nod}}}\delta\underline{\boldsymbol{\eta}}_i^T\underline{\boldsymbol{P}}_i + \sum_{k=1}^{N_{\mathrm{ele}}}\int_{L_k}\delta\boldsymbol{\eta}_k^T\left(\dot{\boldsymbol{p}}_k - \boldsymbol{q}_k\right) + \delta\boldsymbol{\eta}_k'^T\boldsymbol{f}_k\mathrm{d}s_k = 0, \tag{41}$$

where $L_k \in \mathbb{R}$ represents the undeformed length of kth finite element. The two summations control the assembly of finite elements, see e.g. [18]. The underlined symbols represent nodal values. On the element-level we introduce interpolation of $\delta\boldsymbol{\eta}$ (index k will be omitted on the element-level)

$$\delta\boldsymbol{\eta} = \boldsymbol{\Psi}\underline{\delta\boldsymbol{\eta}}, \tag{42}$$

where $\underline{\delta\boldsymbol{\eta}}$ is a column vector consisting of the time-dependent nodal values and the space-dependent shape function matrix $\boldsymbol{\Psi}$ is defined as

$$\boldsymbol{\Psi} = \begin{bmatrix}\psi_1\mathbf{I} & \psi_2\mathbf{I} & \dots & \psi_i\mathbf{I} & \dots & \psi_N\mathbf{I}\end{bmatrix}, \tag{43}$$

where N is the number of nodes on the element, ψ_i is the ith interpolation function and $\mathbf{I}$ is an identity matrix of size 6. We set the following constraint on the choice of the interpolation functions (the reason will become clear in the next section)

$$\sum_{j=1}^{N}\Psi_j(s) = 1 \quad \Rightarrow \quad \sum_{j=1}^{N}\Psi_j'(s) = 0. \tag{44}$$

Eq. (41) can now be written as

$$-\underline{\delta\boldsymbol{\eta}}^T\underline{\boldsymbol{P}} + \sum_{k=1}^{N_{\mathrm{ele}}}\underline{\delta\boldsymbol{\eta}}_k\int_{L_k}\boldsymbol{\Psi}_k^T\left(\dot{\boldsymbol{p}}_k - \boldsymbol{q}_k\right) + \boldsymbol{\Psi}_k'^T\boldsymbol{f}_k\mathrm{d}s_k = 0, \tag{45}$$

where $\underline{\delta\boldsymbol{\eta}}_k$ refers to nodal variations which contribute to element k.

Remark 2. The discretisation introduced so far is incomplete as we have so far only interpolated the test functions. The interpolation of the fields is introduced in Section 5, as it is independent of the interpolation of the test functions.

4. Time integration

The following discusses the time stepping algorithm used to advance time $t \in \mathbb{R}$ from time t_n to $t_{n+1} = t_n + \Delta t$. All quantities indexed with n or $n+1$ refer to these two time instants. The configuration is updated using a Lie group version of the update equation so that the solution remains on the manifold. The update can be expressed using the time-step increments in the fixed-pole frame as

$$\mathbf{H}_{n+1} = \exp(\hat{\boldsymbol{\vartheta}})\mathbf{H}_n, \tag{46}$$

where increments $\boldsymbol{\vartheta} \in \mathbb{R}^6$. They can be mapped between the fixed-pole $\boldsymbol{\vartheta}$ and the material frame version $\boldsymbol{t} \in \mathbb{R}^6$ using the following relation

$$\boldsymbol{\vartheta} = \tilde{\mathbf{H}}_{n+1}\boldsymbol{t}. \tag{47}$$

Using the definition (46), the above expression can be written as $\boldsymbol{\vartheta} = \exp(\hat{\boldsymbol{\vartheta}})\mathbf{H}_n\boldsymbol{t}$. After multiplying both sides with $\exp(-\hat{\boldsymbol{\vartheta}})$ from the left and using identity (C.3), an alternative form of the map (47) is obtained

$$\boldsymbol{\vartheta} = \mathbf{H}_n\boldsymbol{t}. \tag{48}$$

The midpoint rule, which is an implicit second-order time integration scheme, suggests to evaluate the equilibrium equation at the middle of the step, instead of at the end at t_{n+1}. This instance can be interpreted as $t_m = (t_n + t_{n+1})/2$, however, it should be emphasised that the midpoint quantities need not be determined through time evolution; rather, they are chosen in a manner that enables algorithmic conservation of mechanical properties. For this reason, we first present the undetermined quantities in the balance equation set at the midpoint and only later, as we inspect the properties of the conservation of momentum and energy, reverse engineer the proper definitions. The Lie midpoint rule requires the update equation to be conducted using the Lie group operations as presented in (46) to respect the specific solution manifold. Following the Lie midpoint rule, the equilibrium equation at the middle of the step becomes

$$-\underline{\delta\boldsymbol{\eta}}^T\underline{\boldsymbol{P}}_m + \sum_{k=1}^{N_{\text{ele}}} \underline{\delta\boldsymbol{\eta}}_k^T \int_{L_k} \boldsymbol{\Psi}_k^T\left(\dot{\boldsymbol{p}}_{k,m} - \boldsymbol{q}_{k,m}\right) + \boldsymbol{\Psi}_k'^T\boldsymbol{f}_{k,m}\mathrm{d}s_k = 0 \tag{49}$$

where index m denotes the middle of the step.

4.1. Conservation of momentum

Conservation of the total generalised momentum is established when $\boldsymbol{J}_{n+1} - \boldsymbol{J}_n = \mathbf{0}$. To check if this is true we start with

$$\boldsymbol{J}_{n+1} - \boldsymbol{J}_n = \sum_{k=1}^{N_{\text{ele}}} \int_{L_k} (\boldsymbol{p}_{k,n+1} - \boldsymbol{p}_{k,n})\mathrm{d}s_k. \tag{50}$$

Because of the first property in (44) of the interpolation functions, we can write

$$\boldsymbol{J}_{n+1} - \boldsymbol{J}_n = \sum_{k=1}^{N_{\text{ele}}} \int_{L_k} \sum_{j=1}^{N} \Psi_{kj}(\boldsymbol{p}_{k,n+1} - \boldsymbol{p}_{k,n})\mathrm{d}s_k. \tag{51}$$

The upper form suggests to use as the midpoint approximation of the generalised momentum the first-order Taylor series expansion, which is

$$\dot{\boldsymbol{p}}_m \approx \frac{\boldsymbol{p}_{n+1} - \boldsymbol{p}_n}{\Delta t}, \tag{52}$$

which renders (51) to be

$$\boldsymbol{J}_{n+1} - \boldsymbol{J}_n = \Delta t \sum_{k=1}^{N_{\text{ele}}} \int_{L_k} \sum_{j=1}^{N} \Psi_{kj}\dot{\boldsymbol{p}}_{k,m}\mathrm{d}s_k. \tag{53}$$

Proving the conservation of the momentum now requires us to show that this quantity is indeed changed only by external forces when solving the balance Eq. (49). The first term in (49) vanishes for the assumed Neumann boundary conditions. This leads to the following element residual

$$\boldsymbol{R} = \int_L \boldsymbol{\Psi}^T\left(\dot{\boldsymbol{p}}_m - \boldsymbol{q}_m\right) + \boldsymbol{\Psi}'^T\boldsymbol{f}_m\mathrm{d}s \tag{54}$$

assembled by the nodal contributions $\boldsymbol{R}_j$, which can be inserted into (53) to give

$$\boldsymbol{J}_{n+1} - \boldsymbol{J}_n = \Delta t \sum_{k=1}^{N_{\text{ele}}} \sum_{j=1}^{N} \left(\boldsymbol{R}_j + \int_{L_k} \Psi_{kj}\boldsymbol{q}_{k,m} - \Psi'_{kj}\boldsymbol{f}_{k,m}\mathrm{d}s_k \right). \tag{55}$$

Because of the second property of the interpolation function (44) and in the absence of external loading we are thus left with

$$\boldsymbol{J}_{n+1} - \boldsymbol{J}_n = \Delta t \sum_{k=1}^{N_{\text{ele}}} \sum_{j=1}^{N} \boldsymbol{R}_j \tag{56}$$

which obviously vanishes in the equilibrium thus proving the total momentum conservation in the absence of external loading.

We have thus showed that in order to algorithmically conserve generalised momentum, assuming the Neumann boundary conditions and the absence of distributed loading, in combination with the selected discretised midpoint equilibrium Eq. (49), the midpoint approximation of the generalised momentum should be set as the first-order Taylor series expansion of the momentum time derivative (52).

4.2. Conservation of energy

The difference in the mechanical energy after a time step can be computed as

$$E_{n+1} - E_n = EK_{n+1} - EK_n + EP_{n+1} - EP_n \tag{57}$$

$$= \frac{1}{2}\int_L (\boldsymbol{v}_{n+1} + \boldsymbol{v}_n)^T \mathbf{M} \left(\boldsymbol{v}_{n+1} - \boldsymbol{v}_n\right) \mathrm{d}s + \frac{1}{2}\int_L \left(\boldsymbol{\varepsilon}_{n+1} + \boldsymbol{\varepsilon}_n\right)^T \mathbf{K} \left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n\right) \mathrm{d}s. \tag{58}$$

Let us relate increments and velocities via a midpoint-rule approximation in the material frame

$$\Delta t \frac{\boldsymbol{v}_{n+1} + \boldsymbol{v}_n}{2} = \boldsymbol{t}. \tag{59}$$

Using the fixed-pole-material frame relation (47) and (48), the update can be expressed also using the increment in the fixed-pole frame

$$\Delta t \frac{\boldsymbol{v}_{n+1} + \boldsymbol{v}_n}{2} = \tilde{\mathbf{H}}_{n+1}^{-1} \boldsymbol{\vartheta} = \tilde{\mathbf{H}}_n^{-1} \boldsymbol{\vartheta}. \tag{60}$$

The velocities in the above expression conveniently remain in the material frame.

Remark 3. The use of numerical integration relaxes the relation between the time step and the velocity from (60) to hold only at integration points.

Let us consider now the increment of strain obtained by differentiation of the update Eq. (46)

$$\mathbf{H}_{n+1}\hat{\boldsymbol{e}}_{n+1} = \exp(\hat{\boldsymbol{\vartheta}})\mathbf{H}_n\hat{\boldsymbol{e}}_n + \exp(\hat{\boldsymbol{\vartheta}})\widehat{\mathbf{T}(\tilde{\boldsymbol{\vartheta}})\boldsymbol{\vartheta}'}\mathbf{H}_n \tag{61}$$

$$\hat{\boldsymbol{e}}_{n+1} = \hat{\boldsymbol{e}}_n + \mathbf{H}_n^{-1}\widehat{\mathbf{T}(\tilde{\boldsymbol{\vartheta}})\boldsymbol{\vartheta}'}\mathbf{H}_n. \tag{62}$$

After transportation from the Lie algebra $\mathfrak{se}(3)$ to linear space $\mathbb{R}^6$ we obtain

$$\tilde{\mathbf{H}}_n^{-1}\mathbf{T}(\tilde{\boldsymbol{\vartheta}})\boldsymbol{\vartheta}' = \boldsymbol{e}_{n+1} - \boldsymbol{e}_n = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n. \tag{63}$$

Expressions (60) and (63) inserted back into (58) return

$$E_{n+1} - E_n = \int_L \boldsymbol{\vartheta}^T \frac{\tilde{\mathbf{H}}_{n+1}^{-T}\mathbf{M}\boldsymbol{v}_{n+1} - \tilde{\mathbf{H}}_n^{-T}\mathbf{M}\boldsymbol{v}_n}{\Delta t} + \boldsymbol{\vartheta}'^T\mathbf{T}^T(\tilde{\boldsymbol{\vartheta}})\tilde{\mathbf{H}}_n^{-T}\mathbf{K}\frac{\boldsymbol{\varepsilon}_{n+1} + \boldsymbol{\varepsilon}_n}{2}\mathrm{d}s, \tag{64}$$

equivalently written using identity (C.11) and expressions (36), (37) and (46) also as

$$E_{n+1} - E_n = \int_L \boldsymbol{\vartheta}^T \frac{\boldsymbol{p}_{n+1} - \boldsymbol{p}_n}{\Delta t} + \boldsymbol{\vartheta}'^T \frac{\mathbf{T}(-\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_{n+1} + \mathbf{T}(\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_n}{2}\mathrm{d}s \tag{65}$$

and with

$$\boldsymbol{f}_m = \frac{\mathbf{T}(-\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_{n+1} + \mathbf{T}(\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_n}{2} \tag{66}$$

also as

$$E_{n+1} - E_n = \int_L \boldsymbol{\vartheta}^T\dot{\boldsymbol{p}}_m + \boldsymbol{\vartheta}'^T\boldsymbol{f}_m\mathrm{d}s, \tag{67}$$

which alludes to the form of the weak formulation (40) and (49) and implies energy conservation for $\dot{\boldsymbol{p}}_m$ and $\boldsymbol{f}_m$ as defined in (52) and (66). For a time-stepping algorithm to have the energy conservative property, the time-step increment $\boldsymbol{\vartheta}$ must be discretised using the same shape functions as the variation $\delta\boldsymbol{\eta}$.

4.3. Energy decaying algorithm

Energy decaying algorithms are desired to dampen higher frequencies in order to improve the stability of the algorithm [6,19]. As demonstrated by [5,6], energy decay in the midpoint rule is accomplished by including a damping term in the midpoint force (66) to obtain

$$\boldsymbol{f}_m^d = \overbrace{\frac{1}{2}\left(\mathbf{T}(-\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_{n+1} + \mathbf{T}(\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_n\right)}^{\boldsymbol{f}_m} + \frac{1-\tau}{2}\left(\mathbf{T}(-\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_{n+1} - \mathbf{T}(\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_n\right) \tag{68}$$

$$= \left(1-\frac{\tau}{2}\right)\mathbf{T}(-\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_{n+1} + \frac{\tau}{2}\mathbf{T}(\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_n \tag{69}$$

for $\tau \in (0, 1]$. Such definition dampens out the deformations while having no effect on the rigid-body motion. Setting $\tau = 1$ recovers the energy-preserving midpoint-force (66).

Under the assumptions stated in Section 4.2, the total energy of the beam is conserved. By rewriting the expression from (67) using the decaying midpoint force (68) we get

$$\int_L \boldsymbol{\vartheta}^T\dot{\boldsymbol{p}}_m + \boldsymbol{\vartheta}'^T\boldsymbol{f}_m^d \mathrm{d}s = 0. \tag{70}$$

Expanding this expression reveals

$$\begin{aligned}&\int_L \boldsymbol{\vartheta}^T\dot{\boldsymbol{p}}_m + \boldsymbol{\vartheta}'^T\boldsymbol{f}_m + \frac{1-\tau}{2}\boldsymbol{\vartheta}'^T\left(\mathbf{T}(-\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_{n+1} - \mathbf{T}(\tilde{\boldsymbol{\vartheta}})^T\boldsymbol{f}_n\right)\mathrm{d}s\\ &= \frac{1}{2}\int_L (\boldsymbol{v}_{n+1}+\boldsymbol{v}_n)^T\mathbf{M}\left(\boldsymbol{v}_{n+1}-\boldsymbol{v}_n\right)\mathrm{d}s + \frac{1}{2}\int_L \left(\boldsymbol{\varepsilon}_{n+1}+\boldsymbol{\varepsilon}_n\right)^T\mathbf{K}\left(\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}_n\right)\mathrm{d}s\\ &\quad + \frac{1-\tau}{2}\int_L \left(\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}_n\right)^T\mathbf{K}\left(\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}_n\right)\mathrm{d}s = 0.\end{aligned} \tag{71}$$

Defining the decaying term $\mathcal{D}$ as

$$\mathcal{D} = \frac{1-\tau}{2}\int_L \left(\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}_n\right)^T\mathbf{K}\left(\boldsymbol{\varepsilon}_{n+1}-\boldsymbol{\varepsilon}_n\right)\mathrm{d}s \tag{72}$$

allows us to reorganise the terms into

$$E_{n+1} - E_n + \mathcal{D} = 0. \tag{73}$$

As the decaying term is positive $\mathcal{D} \geq 0$, it can be concluded that the mechanical energy decays with each time step.

4.4. External forces

Since the increment of work done by external forces can be calculated as

$$\Delta W = \int_L \boldsymbol{\vartheta}^T\boldsymbol{q}_m \mathrm{d}s, \tag{74}$$

the midpoint load can be defined as the average load

$$\boldsymbol{q}_m = \frac{1}{2}\left(\boldsymbol{q}_{n+1} + \boldsymbol{q}_n\right). \tag{75}$$

5. Finite element formulation

A finite element has its configuration state interpolated using the implicit $SE(3)$ interpolation in a manner similar to that defined in [14] for static problems. Such interpolation provides objective, strain-invariant and locking-free elements as theoretically explained and numerically proven in [14,20]. An analytical expression can be used to define the first-order interpolation, which involves two nodes, see [13]. However, for higher-order interpolations, an implicit definition is required, which involves solving an additional nonlinear 6D equation. It should be noted, however, that the present formulation is intrinsically dynamic, with the weak form of the equations of motion established at an algorithmic mid-point configuration, while the actual unknowns are computed at the end of a time step, thus making the tangent operator obviously non-symmetric.

Let us define the implicit $SE(3)$ configuration interpolation for N nodes per element as

$$\sum_{i=1}^{N} \psi_i\boldsymbol{\phi}_i = \mathbf{0}, \quad \exp(\hat{\boldsymbol{\phi}}_i) = \underset{\sim}{\mathbf{H}}_i^{-1}\mathbf{H}, \tag{76}$$

where $\boldsymbol{\phi}_i \in \mathbb{R}^6$ and shape functions ψ_i are Lagrange polynomials. The auxiliary quantities $\boldsymbol{\phi}_i(s)$ can be physically interpreted as generalised displacement-rotation vectors between the configuration of the ith node and the configuration at the arc-length coordinate s. This equation needs to be solved using a Newton–Raphson algorithm as presented in A.

The finite element is designed to have $\boldsymbol{\vartheta}$ as the nodal unknowns as they are conveniently in the tangent space.

Remark 4. The interpolation of the configuration has been introduced independently of the test functions' interpolation, which have been defined in (44) as complete to enable momentum conservation. The selected configuration interpolation is in violation of the conservation of energy requirement that the time-step increment $\boldsymbol{\vartheta}$ must be expressible with the same shape functions as the variation $\delta\boldsymbol{\eta}$ after discretisation. However, using the described algorithm, mesh refinement leads to a smaller discrepancy between the two, which in turn leads to better energy conservation. See B, where this is theoretically shown. This is also illustrated with numerical examples.

5.1. Explicit first-order interpolation

For two nodes, the $SE(3)$ interpolation (76) reduces to the helicoidal interpolation and can be thus expressed explicitly. Between nodes 1 and 2, the configuration is interpolated as

$$\mathbf{H}(s) = \underline{\mathbf{H}}_1 \exp\left(\frac{s}{L}\hat{\boldsymbol{d}}\right), \quad \exp(\hat{\boldsymbol{d}}) = \underline{\mathbf{H}}_1^{-1}\underline{\mathbf{H}}_2, \tag{77}$$

where $\boldsymbol{d} \in \mathbb{R}^6$. The spatial derivative of the curve $\mathbf{H}(s)$ is constant

$$\mathbf{H}'(s) = \frac{1}{L}\mathbf{H}\hat{\boldsymbol{d}}, \tag{78}$$

yielding

$$\boldsymbol{e} = \frac{\boldsymbol{d}}{L}. \tag{79}$$

Instantaneous velocity $\boldsymbol{v}$ is not interpolated, see the explanation under the next section. The algorithm is described in Box 1.

5.2. Implicit interpolation

Spatial differentiation of (76) yields

$$\sum_{i=1}^{N} \psi_i'\boldsymbol{\phi}_i + \psi_i\boldsymbol{\phi}_i' = \mathbf{0}, \quad \exp(\hat{\boldsymbol{\phi}}_i)\widehat{\mathbf{T}(\tilde{\boldsymbol{\phi}}_i)\boldsymbol{\phi}_i'} = \exp(\hat{\boldsymbol{\phi}}_i)\hat{\boldsymbol{e}}. \tag{80}$$

From the second equation in (80) we can express $\boldsymbol{\phi}_i'$

$$\boldsymbol{\phi}_i' = \mathbf{T}^{-1}(\tilde{\boldsymbol{\phi}}_i)\boldsymbol{e}. \tag{81}$$

Inserting it into the first equation in (80) yields

$$\boldsymbol{e} = -\boldsymbol{\Psi}^* \sum_{i=1}^{N} \psi_i'\boldsymbol{\phi}_i, \tag{82}$$

where

$$\boldsymbol{\Psi}^* = \left[\sum_{i=1}^{N} \psi_i \mathbf{T}^{-1}(\tilde{\boldsymbol{\phi}}_i)\right]^{-1} \tag{83}$$

Note that this expression can be explicitly evaluated if the implicitly defined $\mathbf{H}(s)$ has already been computed.

Time differentiation of (76) yields

$$\sum_{i=1}^{N} \psi_i\dot{\boldsymbol{\phi}}_i = \mathbf{0}, \quad \exp(\hat{\boldsymbol{\phi}}_i)\widehat{\mathbf{T}(\tilde{\boldsymbol{\phi}}_i)\dot{\boldsymbol{\phi}}_i} = \exp(\hat{\boldsymbol{\phi}}_i)\hat{\boldsymbol{v}} - \underline{\hat{\boldsymbol{v}}}_i \exp(\hat{\boldsymbol{\phi}}_i) \tag{84}$$

from where we get

$$\dot{\boldsymbol{\phi}}_i = \mathbf{T}^{-1}(\tilde{\boldsymbol{\phi}}_i)\boldsymbol{v} - \mathbf{T}^{-1}(-\tilde{\boldsymbol{\phi}}_i)\underline{\boldsymbol{v}}_i \tag{85}$$

and finally

$$\boldsymbol{v} = \boldsymbol{\Psi}^* \sum_{i=1}^{N} \psi_i \mathbf{T}^{-1}(-\tilde{\boldsymbol{\phi}}_i)\underline{\boldsymbol{v}}_i. \tag{86}$$

Interpolation of velocity can be now expressed using an interpolation matrix as

$$\boldsymbol{v} = \boldsymbol{\Psi}^*\mathbf{Q}\underline{\boldsymbol{v}}, \tag{87}$$

where $\mathbf{Q}$ is a $6 \times 6N$ matrix, defined as

$$\mathbf{Q} = \begin{bmatrix} \psi_1\mathbf{T}^{-1}(-\tilde{\boldsymbol{\phi}}_1) & \dots & \psi_N\mathbf{T}^{-1}(-\tilde{\boldsymbol{\phi}}_N) \end{bmatrix}. \tag{88}$$

Again, as with the spatial derivative (82), the velocity interpolation matrix is explicitly defined. Furthermore, matrix $\boldsymbol{\Psi}^*$ is the same in both cases and can be reused.

The above interpolation of velocity is, however, in conflict with Eq. (60) which relates the time step $\boldsymbol{\eta}$ computed from (46) and velocity. Explicit expression of velocity in terms of the nodal increments is thus not possible. However, since we only require the relation from (60) to hold at integration points, we can avoid an explicit definition of the velocity field and resort to storing the values at the integration points and update them according to (60). See the algorithm in Box 1.

5.3. Time integrator

The presented formulation can be solved using the Newton–Raphson iterative procedure as explained in the algorithm in Box 1. The system of equations to be solved at each iteration i based on residual (54) reads

$$\mathbf{K}_{\text{glob}}(\boldsymbol{\vartheta}^i_{\text{glob}})\Delta\boldsymbol{\vartheta}^i_{\text{glob}} = -\boldsymbol{R}_{\text{glob}}(\boldsymbol{\vartheta}^i_{\text{glob}}), \tag{89}$$

where Δ is the linearisation operator, $\mathbf{K}_{\text{glob}}$ is the gradient of the assembled residual $\boldsymbol{R}_{\text{glob}}$ wrt. the global unknowns, i.e. time-step increments $\boldsymbol{\vartheta}_{\text{glob}}$. The time-step increments are updated between iterations straightforwardly as

$$\boldsymbol{\vartheta}^{i+1}_{\text{glob}} = \boldsymbol{\vartheta}^i_{\text{glob}} + \Delta\boldsymbol{\vartheta}^i_{\text{glob}}. \tag{90}$$

To evaluate the time-step increment at the integration points, one can simply invert (46) and obtain

$$\hat{\boldsymbol{\vartheta}}(s) = \log(\mathbf{H}_{n+1}(s)\mathbf{H}_n^{-1}(s)), \tag{91}$$

where values $\mathbf{H}_n$ and $\mathbf{H}_{n+1}$ are obtained via interpolation.

Box 1: Algorithm for updating the finite element and solving the balance equation.

1. Initiate new time step by storing the converged values in nodes $\mathbf{H}_{n+1} \leftarrow \mathbf{H}_n$ and in integration points $\boldsymbol{v}_{n+1} \leftarrow \boldsymbol{v}_n$, $\boldsymbol{q}_{n+1} \leftarrow \boldsymbol{q}_n$ and $\boldsymbol{\vartheta} \leftarrow \Delta t\boldsymbol{v}_n$.
2. Update $\mathbf{H}_{n+1}$ in nodes using Eq. (46) and $\boldsymbol{v}_{n+1}$ in integration points using Eq. (60).
3. Compute interpolated values of $\mathbf{H}_{n+1}$ and $\mathbf{H}_n$ in the integration points using Eq. (76) as explained in Appendix A or use Eq. (77) for first-order elements. Evaluate strains $\boldsymbol{\varepsilon}_{n+1}$ and $\boldsymbol{\varepsilon}_n$ in the integration points using Eq. (76) and Eq. (82) or Eq. (23), Eq. (77) and Eq. (79) for first-order elements.
4. Compute the momentum $\boldsymbol{p}_{n+1}$ and $\boldsymbol{p}_n$ using Eq. (36) and internal forces $\boldsymbol{f}_{n+1}$ and $\boldsymbol{f}_n$ using Eq. (37), all in the integration points.
5. Compute the residual vector from Eq. (52), Eq. (54), Eq. (69) and Eq. (75).
6. Check the norm of the residual vector against the tolerance.
7. Solve Eq. (89) and update $\boldsymbol{\vartheta}$ using Eq. (90).
8. Continue with 2.

6. Numerical examples

6.1. Flying spaghetti

This examples is taken from [21]. An impulse initiates the motion of a free-flying beam, as presented in Fig. 2. The impulse is generated by a generalised force $\{F_x, 0, 0, 0, M_y, M_z\}$, where $F_x = M_z/10$, $M_y = M_z/2$ and M_z is given as a function of time in Fig. 3. The following are the material parameters: $\rho A = 1$, $\rho I = 10$, $EA = GA = 10^4$, $GJ_t = EI = 500$. All integrals are computed using full integration. The motion of the beam at various times, revealing its large deformations and finite rotations, is displayed in Fig. 4.

The mechanical energy and generalised momentum are expected to grow during the loading part due to the work of external forces. In the second part (load-free flight), the energy and momentum are expected to remain constant as no external forces work on the system. The generalised momentum is plotted in Fig. 5 where we can observe conservation of momentum during the flight. The combined kinetic and potential energy is plotted in Fig. 8 for different spatial discretisations with time step 0.1. As expected from the theory, we can observe deviations from the constant value for coarser meshes, however, with mesh refinement the energy quickly converges towards a constant value.

This is shown also in Fig. 9, which plots standard deviation of energy during the flight for different mesh sizes. From the standard deviation, the magnitude of the absolute error can be estimated as one σ, which is where, for large-enough sets, lays 68% of all samples. With the average value of the mechanical energy in the unloaded portion of the test, of 723, the relative error can be estimated as $\varepsilon_{\text{rel.}} = 10^{\log_2(N^{-0.85}/2)}$, where N is the number of nodes in the mesh. For an example, for the case using 4 second-order elements, the relative error is 4×10^{-5}.

To test convergence with respect to the time-step size, a mesh consisting of ten first-order elements has been subjected to $t \in [0, 6]$ analysis using different time steps. The convergence is plotted in Fig. 6. Since no analytical solution to the problem exists, the solution

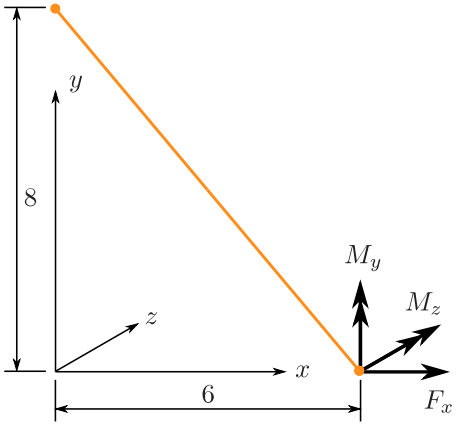


Fig. 2. Initial setup of the flying spaghetti.

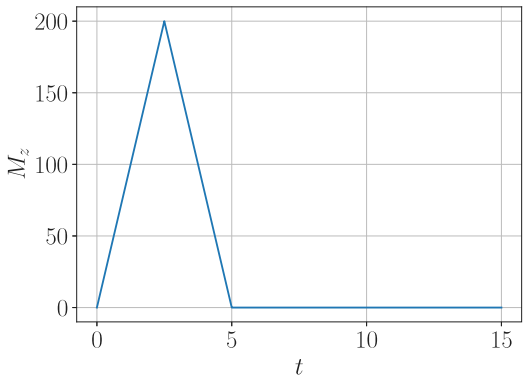


Fig. 3. Time dependency of load for the flying spaghetti.

computed with the smallest time step 2^{-8} is taken as the reference solution. The error is defined as the norm of the difference in displacement for the loaded end node compared to the reference solution at $t_t = 6$.

$$\epsilon = \text{norm}(\boldsymbol{r}_L(t_t) - \boldsymbol{r}_{L,\text{ref}}(t_t)). \tag{92}$$

The simulation has been run with the integration parameters $\tau = 1$ and $\tau = 0.7$. The order of convergence has been estimated from the sequence for $\tau = 1$ cases as 2.02 which is expected for the midpoint rule. When using the energy decay mechanism the convergence order is reduced as expected for this sort of energy decaying mechanism — we measured it at 1.57.

Fig. 7 shows a comparison of mechanical energy over time for different values of τ, which introduces damping. The results are obtained using 4 second-order elements and $\Delta t = 0.1$.

7. Conclusion

We propose a novel time integration algorithm for beam dynamics, which conserves momentum and exhibits near energy-conservation or controlled energy decay. Our approach combines beam elements interpolated on the $SE(3)$ Lie group with an adapted Lie midpoint rule for the equilibrium equation in the fixed-pole frame. In addition, we apply the algorithm to higher-order elements, employing implicit interpolation on $SE(3)$. By employing configuration-independent test functions in the equation of motion, we ensure the preservation of momentum in the absence of external forces.

In an attempt to achieve algorithmic conservation of energy, we approximate the velocity field at integration points using the midpoint approximation and construct an appropriate midpoint approximation of internal forces. However, due to the mismatch between test and shape functions, conservation can only be approached through mesh refinement. To introduce an energy decaying behaviour, we incorporate an additional term into the midpoint force expression, controlled by a parameter $\tau \in (0, 1]$. Setting $\tau = 1$ yields an energy-conservative algorithm, while $0 < \tau < 1$ guarantees strict energy decay for deformed beams. The conservation of momentum holds for any value of τ.

One advantage of our method is its simplicity in implementation, as it avoids time derivatives of a configuration-dependent interpolation. The numerical example demonstrates the robustness of our approach, which is capable of handling large time steps.

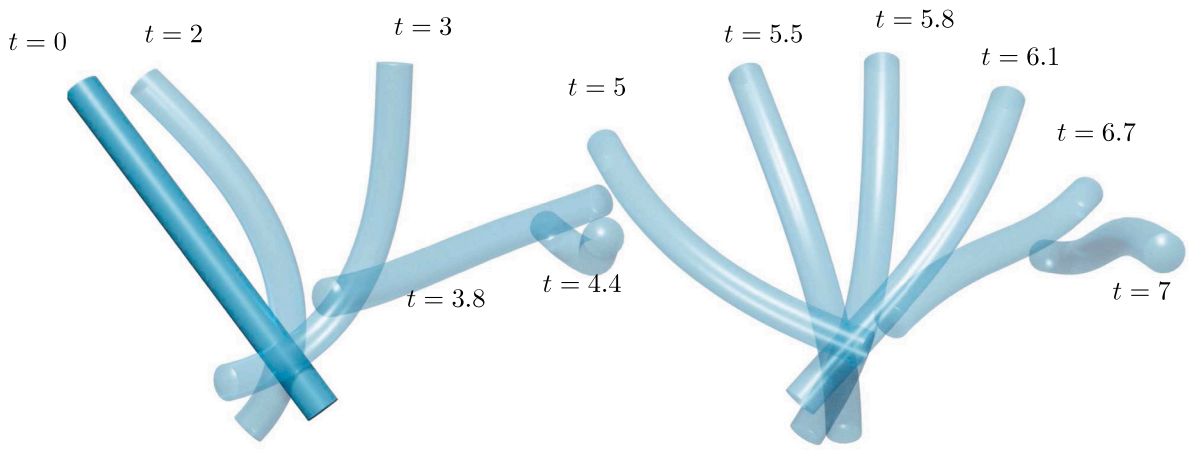
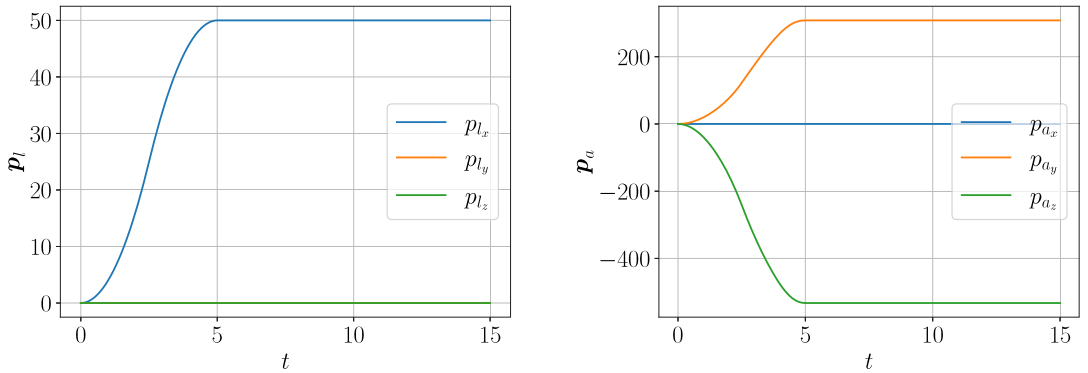


Fig. 4. Trace of deformed configurations at given times for 10 first-order elements and $\Delta t = 0.1$.



(a) Linear momentum (b) Angular momentum

Fig. 5. Plot of linear and angular momentum for two first-order elements with time step 0.1. Components of linear momentum p_{l_y} and p_{l_z} and angular momentum p_{a_x} are zero.

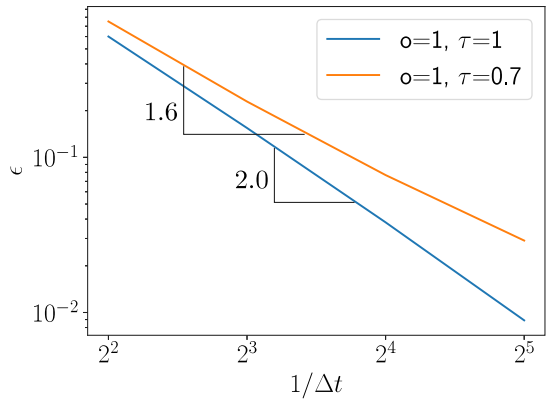


Fig. 6. Convergence with respect to the time-step size using 10 first-order elements.

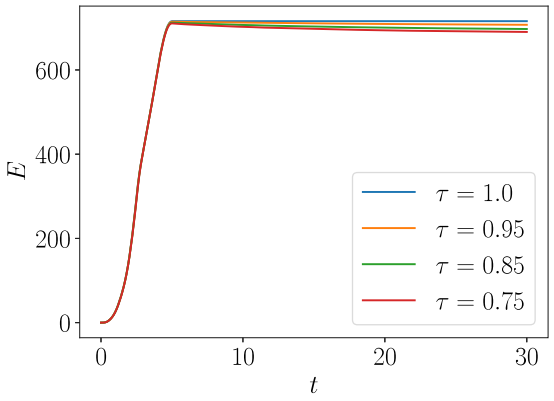


Fig. 7. Comparison of energy decay for different values of parameter τ.

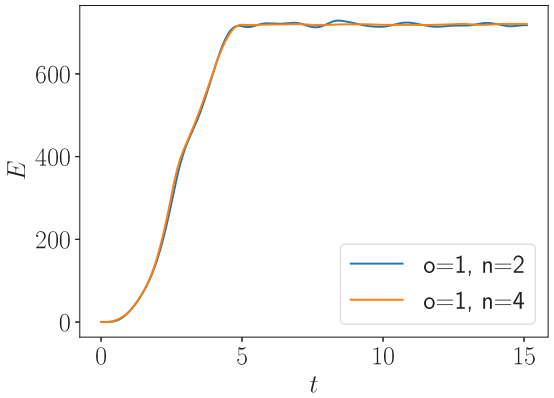


Fig. 8. Plot of kinetic and potential energy for two different meshes with two and four first-order elements.

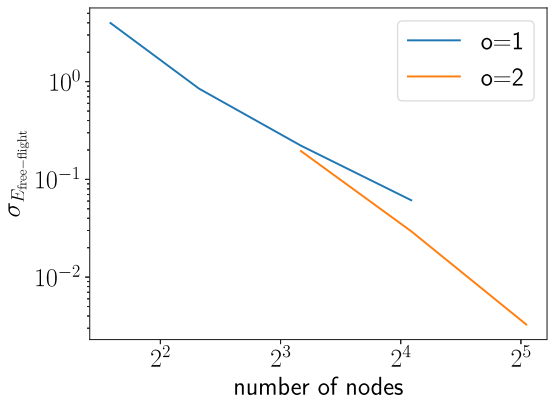


Fig. 9. Convergence of the load-free flight phase mechanical energy towards a constant value for different discretisations with ten first and ten second-order elements.

Moreover, the conservation properties of our element are algorithmic in nature, so that the momentum is independent of the chosen time-step size, while the energy approaches algorithmic conservation with mesh refinement. The element is interpolated using objective interpolation functions, effectively preventing locking and enabling strain-invariant interpolation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Solving implicit interpolation

Evaluation of implicit interpolation (76) at a certain point s requires the use of Newton–Raphson algorithm. The size of the resulting system of equations corresponds to the number of degrees of freedom — that is 6. The residual can be defined as

$$\boldsymbol{R}_H = \sum_{i=1}^{N} \psi_i \boldsymbol{\phi}_i = \boldsymbol{0}, \quad \exp(\hat{\boldsymbol{\phi}}_i)\widehat{\mathbf{T}(\tilde{\boldsymbol{\phi}}_i)\boldsymbol{\phi}'_i} = \exp(\hat{\boldsymbol{\phi}}_i)\hat{\boldsymbol{e}}. \tag{A.1}$$

Linearisation of the above expression goes as follows

$$\Delta \sum_{i=1}^{N} \psi_i(s)\boldsymbol{\phi}_i = \sum_{i=1}^{N} \psi_i(s)\Delta\boldsymbol{\phi}_i. \tag{A.2}$$

Let $\mathbf{H}(s) = \exp(\hat{\boldsymbol{u}})$. Then

$$\begin{aligned}
&\Delta \exp\left(\hat{\boldsymbol{\phi}}_i\right) = \underline{\mathbf{H}}_i^{-1} \Delta \mathbf{H}(s),\\
&\exp\left(\hat{\boldsymbol{\phi}}_i\right) \widehat{T\left(\tilde{\boldsymbol{\phi}}_i\right)\Delta\boldsymbol{\phi}_i} = \underline{\mathbf{H}}_i^{-1} \exp(\hat{\boldsymbol{u}})\, \widehat{T(\tilde{\boldsymbol{u}})\,\Delta\boldsymbol{u}},\\
&T\left(\tilde{\boldsymbol{\phi}}_i\right)\Delta\boldsymbol{\phi}_i = T(\tilde{\boldsymbol{u}})\,\Delta\boldsymbol{u}.
\end{aligned} \tag{A.3}$$

This leads to the following iteration

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \left[\sum_{i=1}^{N} \psi_i(s) T\left(\tilde{\boldsymbol{\phi}}_i^k\right)^{-1} T\left(\tilde{\boldsymbol{u}}^k\right)\right]^{-1} \left(\sum_{i=1}^{N} \psi_i(s)\boldsymbol{\phi}_i^k\right). \tag{A.4}$$

Appendix B. Energy conservation

1. Energy conservation is provided if the test functions

   $$\delta\boldsymbol{\eta}(s) = \boldsymbol{\Psi}(s)\underline{\delta\boldsymbol{\eta}}$$

   are interpolated in the same way as the increments $\boldsymbol{\vartheta}$ over the time step. Our algorithm is not such.
2. Numerical evidence suggests that in our algorithm the energy-conservation error reduces with the element length. But can this to be proven?
3. To do so, let us limit our attention to the first-order interpolation only, where

   $$\begin{aligned}
   &\delta\boldsymbol{\eta}(s) = \frac{L-s}{L}\underline{\delta\boldsymbol{\eta}}_1 + \frac{s}{L}\underline{\delta\boldsymbol{\eta}}_2\\
   &\mathbf{H}(s) = \underline{\mathbf{H}}_1 \exp\left(\frac{s}{L}\hat{\boldsymbol{d}}\right), \qquad \exp(\hat{\boldsymbol{d}}) = \underline{\mathbf{H}}_1^{-1}\underline{\mathbf{H}}_2.
   \end{aligned}$$

   Take the assumption that L is small, such that s/L stays within the range $[0,1]$, but $\boldsymbol{d}$ becomes increasingly smaller, such that $\hat{\boldsymbol{d}}^2 \to \boldsymbol{0}$ (and also $\widehat{\boldsymbol{\vartheta}_2 - \boldsymbol{\vartheta}_1}^2 \to \boldsymbol{0}$). Can it be shown that under this assumption the above interpolation for $\mathbf{H}(s)$ is equivalent to

   $$\boldsymbol{\vartheta}(s) = \frac{L-s}{L}\underline{\boldsymbol{\vartheta}}_1 + \frac{s}{L}\underline{\boldsymbol{\vartheta}}_2,$$

   where $\mathbf{H}_{n+1}(s) = \exp\left(\hat{\boldsymbol{\vartheta}}(s)\right)\mathbf{H}_n(s)$? If yes, this provides the proof under #2.
4. The actual interpolation for $\boldsymbol{\vartheta}(s)$ is

   $$\begin{aligned}
   \exp(\hat{\boldsymbol{\vartheta}}) &= \mathbf{H}_{n+1}(s)\mathbf{H}_n^{-1}(s)\\
   &= \mathbf{H}_{1,n+1} \exp\left(\frac{s}{L}\hat{\boldsymbol{d}}_{n+1}\right) \exp\left(-\frac{s}{L}\hat{\boldsymbol{d}}_n\right) \mathbf{H}_{1,n}^{-1}\\
   &= \exp(\hat{\boldsymbol{\vartheta}}_1)\mathbf{H}_{1,n} \sum_{i=0}^{\infty} \frac{1}{i!}\left(\frac{s}{L}\hat{\boldsymbol{d}}_{n+1}\right)^i \sum_{j=0}^{\infty}\left(-\frac{s}{L}\hat{\boldsymbol{d}}_n\right)^j \mathbf{H}_{1,n}^{-1},
   \end{aligned}$$

   where

   $$\begin{aligned}
   \exp\left(\frac{s}{L}\hat{\boldsymbol{d}}_{n+1}\right)\exp\left(-\frac{s}{L}\hat{\boldsymbol{d}}_n\right) = \mathbf{I} &+ \frac{s}{L}\hat{\boldsymbol{d}}_{n+1} - \frac{s}{L}\hat{\boldsymbol{d}}_n\\
   &+ \frac{1}{2!}\left(\frac{s}{L}\hat{\boldsymbol{d}}_{n+1}\right)^2 - \left(\frac{s}{L}\right)^2 \hat{\boldsymbol{d}}_{n+1}\hat{\boldsymbol{d}}_n + \frac{1}{2!}\left(\frac{s}{L}\hat{\boldsymbol{d}}_n\right)^2\\
   &+ \frac{1}{3!}\left(\frac{s}{L}\hat{\boldsymbol{d}}_{n+1}\right)^3 - \frac{1}{2!}\left(\frac{s}{L}\right)^3 \hat{\boldsymbol{d}}_{n+1}^2\hat{\boldsymbol{d}}_n + \frac{1}{2!}\left(\frac{s}{L}\right)^3 \hat{\boldsymbol{d}}_{n+1}\hat{\boldsymbol{d}}_n^2 - \frac{1}{3!}\left(\frac{s}{L}\hat{\boldsymbol{d}}_n\right)^3 + \cdots
   \end{aligned}$$

$$
\begin{aligned}
&= \mathbf{I} + \frac{s}{L}\left(\hat{\boldsymbol{d}}_{n+1} - \hat{\boldsymbol{d}}_n\right) \\
&\quad + \left(\frac{s}{L}\right)^2 \left(\frac{1}{2}\hat{\boldsymbol{d}}_{n+1}^2 - \hat{\boldsymbol{d}}_{n+1}\hat{\boldsymbol{d}}_n + \frac{1}{2}\hat{\boldsymbol{d}}_n^2\right) \\
&\quad + \frac{1}{2!}\left(\frac{s}{L}\right)^3 \left(\frac{1}{3}\hat{\boldsymbol{d}}_{n+1}^3 - \hat{\boldsymbol{d}}_{n+1}\hat{\boldsymbol{d}}_n - \hat{\boldsymbol{d}}_{n+1}\hat{\boldsymbol{d}}_n - \frac{1}{3}\hat{\boldsymbol{d}}_n^3\right) + \cdots \\
&= \mathbf{I} + \frac{s}{L}(\hat{\boldsymbol{d}}_{n+1} - \hat{\boldsymbol{d}}_n) + \mathcal{O}\left(\hat{\boldsymbol{d}}^2\right).
\end{aligned}
$$

Likewise

$$
\begin{aligned}
\exp(-\hat{\boldsymbol{\vartheta}}_1)\exp\left(\hat{\boldsymbol{\vartheta}}(s)\right) &= \mathbf{I} - \hat{\boldsymbol{\vartheta}}_1 + \hat{\boldsymbol{\vartheta}}(s) \\
&\quad + \frac{1}{2!}\hat{\boldsymbol{\vartheta}}_1^2 - \hat{\boldsymbol{\vartheta}}_1\hat{\boldsymbol{\vartheta}}(s) + \frac{1}{2!}\left(\hat{\boldsymbol{\vartheta}}(s)\right)^2 \\
&\quad - \frac{1}{3!}\hat{\boldsymbol{\vartheta}}_1^3 + \frac{1}{2!}\hat{\boldsymbol{\vartheta}}_1^2\hat{\boldsymbol{\vartheta}}(s) - \frac{1}{2!}\hat{\boldsymbol{\vartheta}}_1\left(\hat{\boldsymbol{\vartheta}}(s)\right)^2 + \frac{1}{3!}\left(\hat{\boldsymbol{\vartheta}}(s)\right)^3 + \cdots \\
&= \mathbf{I} + \hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1 + \mathcal{O}\left((\hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1)^2\right).
\end{aligned}
$$

5. Hence

$$
\mathbf{I} + \hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1 + \mathcal{O}\left((\hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1)^2\right) = \mathbf{I} + \frac{s}{L}\quad\widehat{\tilde{\mathbf{H}}_{1,n}(\boldsymbol{d}_{n+1} - \boldsymbol{d}_n)} + \mathcal{O}\left(\hat{\boldsymbol{d}}^2\right),
$$

i.e.

$$
\hat{\boldsymbol{\vartheta}}(s) = \hat{\boldsymbol{\vartheta}}_1 + \frac{s}{L}\quad\widehat{\tilde{\mathbf{H}}_{1,n}(\boldsymbol{d}_{n+1} - \boldsymbol{d}_n)} + \mathcal{O}\left(\hat{\boldsymbol{d}}^2\right) + \mathcal{O}\left((\hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1)^2\right)
$$

where, according to (48),

$$
\tilde{\mathbf{H}}_{1,n}(\boldsymbol{d}_{n+1} - \boldsymbol{d}_n) = \boldsymbol{\vartheta}_2 - \boldsymbol{\vartheta}_1.
$$

From here,

$$
\begin{aligned}
\hat{\boldsymbol{\vartheta}}(s) &= \hat{\boldsymbol{\vartheta}}_1 + \frac{s}{L}\widehat{\boldsymbol{\vartheta}_2 - \boldsymbol{\vartheta}_1} + \mathcal{O}\left(\hat{\boldsymbol{d}}^2\right) + \mathcal{O}\left((\hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1)^2\right) \\
&= \frac{L-s}{L}\hat{\boldsymbol{\vartheta}}_1 + \frac{s}{L}\hat{\boldsymbol{\vartheta}}_2 + \mathcal{O}\left(\hat{\boldsymbol{d}}^2\right) + \mathcal{O}\left((\hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1)^2\right).
\end{aligned} \tag{*}
$$

6. Clearly, therefore, the actual interpolation

$$
\exp\left(\hat{\boldsymbol{\vartheta}}(s)\right) = \mathbf{H}_{n+1}(s)\mathbf{H}_n^{-1}(s)
$$

from which Eq. (*) is derived, is equivalent to

$$
\boldsymbol{\vartheta}(s) = \frac{L-s}{L}\hat{\boldsymbol{\vartheta}}_1 + \frac{s}{L}\hat{\boldsymbol{\vartheta}}_2
$$

if

$$
\mathcal{O}\left(\hat{\boldsymbol{d}}^2\right) + \mathcal{O}\left((\hat{\boldsymbol{\vartheta}}(s) - \hat{\boldsymbol{\vartheta}}_1)^2\right) = \mathbf{0}
$$

which is true following the initial assumption. Hence, for small elements, same interpolation is used for the test and the trial functions leading to increased energy conservation as the mesh is refined.

Appendix C. Useful identities

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{3 \text{ or } 6}$ and $\mathbf{Z} \in SO(3)$ or $SE(3)$:

$$
\exp(\hat{\boldsymbol{x}}) = \sum_{i=0}^{\infty} \frac{\hat{\boldsymbol{x}}^i}{i!} \tag{C.1}
$$

$$
\exp(\tilde{\boldsymbol{x}}) = \widehat{(\exp(\hat{\boldsymbol{x}}))} \tag{C.2}
$$

$$
\exp(\alpha\hat{\boldsymbol{x}})\boldsymbol{x} = \boldsymbol{x} \tag{C.3}
$$

$$
\mathbf{T}(\alpha\tilde{\boldsymbol{x}})\boldsymbol{x} = \boldsymbol{x} \tag{C.4}
$$

$$
\mathbf{T}^{-1}(\tilde{\boldsymbol{x}}) - \mathbf{T}^{-1}(-\tilde{\boldsymbol{x}}) = \hat{\boldsymbol{x}} \tag{C.5}
$$

$$
\exp(\tilde{\boldsymbol{x}}) = \mathbf{I} + \mathbf{T}(-\tilde{\boldsymbol{x}})\tilde{\boldsymbol{x}} \tag{C.6}
$$

$$
\mathbf{T}(-\tilde{\boldsymbol{x}})\mathbf{T}(\tilde{\boldsymbol{x}}) = \mathbf{T}(\tilde{\boldsymbol{x}})\mathbf{T}(-\tilde{\boldsymbol{x}}) \tag{C.7}
$$

$$
\mathbf{T}(-\tilde{\boldsymbol{x}})^{-1}\mathbf{T}(\tilde{\boldsymbol{x}}) = \mathbf{T}(\tilde{\boldsymbol{x}})\mathbf{T}(-\tilde{\boldsymbol{x}})^{-1} \tag{C.8}
$$

$$
\mathbf{T}(-\tilde{\boldsymbol{x}})\mathbf{T}(\tilde{\boldsymbol{x}})^{-1} = \mathbf{T}(\tilde{\boldsymbol{x}})^{-1}\mathbf{T}(-\tilde{\boldsymbol{x}}) \tag{C.9}
$$

$$\mathbf{T}(-\tilde{\boldsymbol{x}})^{-1}\mathbf{T}(\tilde{\boldsymbol{x}})^{-1} = \mathbf{T}(\tilde{\boldsymbol{x}})^{-1}\mathbf{T}(-\tilde{\boldsymbol{x}})^{-1} \tag{C.10}$$

$$\exp(\tilde{\boldsymbol{x}}) = \mathbf{T}(-\tilde{\boldsymbol{x}})\mathbf{T}^{-1}(\tilde{\boldsymbol{x}}) \tag{C.11}$$

$$\mathrm{d}\exp(\hat{\boldsymbol{x}}) = \exp(\hat{\boldsymbol{x}})\,\widehat{\mathbf{T}(\tilde{\boldsymbol{x}})\mathrm{d}\boldsymbol{x}} = \widehat{\mathbf{T}(-\tilde{\boldsymbol{x}})\mathrm{d}\boldsymbol{x}}\exp(\hat{\boldsymbol{x}}) \tag{C.12}$$

$$\frac{\mathrm{d}T(s\tilde{\boldsymbol{x}})}{\mathrm{d}s} = \exp(-\tilde{\boldsymbol{x}}(s)) - \mathbf{T}(\tilde{\boldsymbol{x}}) = \frac{\mathbf{I}}{s} - \mathbf{T}(s\tilde{\boldsymbol{x}})\left(\frac{\mathbf{I}}{s} + \tilde{\boldsymbol{x}}\right) \tag{C.13}$$

$$\exp(\widehat{\mathbf{Z}\boldsymbol{x}}) = \mathbf{Z}\exp(\boldsymbol{x})\mathbf{Z}^{-1} \tag{C.14}$$

$$\mathbf{T}(\widetilde{\mathbf{Z}\boldsymbol{x}}) = \mathbf{Z}\mathbf{T}(\tilde{\boldsymbol{x}})\mathbf{Z}^{-1} \tag{C.15}$$

$$\widehat{\mathbf{Z}\boldsymbol{x}} = \mathbf{Z}\hat{\boldsymbol{x}}\mathbf{Z}^{-1} \tag{C.16}$$

$$\widehat{\hat{\boldsymbol{x}}\boldsymbol{y}} = \hat{\boldsymbol{x}}\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}\hat{\boldsymbol{x}} \tag{C.17}$$

References

[1] J. Simo, N. Tarnow, K. Wong, Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics, Comput. Methods Appl. Mech. Engrg. 100 (1) (1992) 63–116, http://dx.doi.org/10.1016/0045-7825(92)90115-z.

[2] G. Jelenić, M. Crisfield, Geometrically exact 3D beam theory: implementation of a strain-invariant finite element for statics and dynamics, Comput. Methods Appl. Mech. Engrg. 171 (1–2) (1999) 141–171, http://dx.doi.org/10.1016/s0045-7825(98)00249-7.

[3] J.C. Simo, N. Tarnow, M. Doblare, Non-linear dynamics of three-dimensional rods: Exact energy and momentum conserving algorithms, Internat. J. Numer. Methods Engrg. 38 (9) (1995) 1431–1473, http://dx.doi.org/10.1002/nme.1620380903.

[4] O.A. Bauchau, G. Damilano, N.J. Theron, Numerical integration of non-linear elastic multi-body systems, Internat. J. Numer. Methods Engrg. 38 (16) (1995) 2727–2751, http://dx.doi.org/10.1002/nme.1620381605.

[5] C.L. Bottasso, M. Borri, Integrating finite rotations, Comput. Methods Appl. Mech. Engrg. 164 (3–4) (1998) 307–331, http://dx.doi.org/10.1016/s0045-7825(98)00031-0.

[6] A. Ibrahimbegović, S. Mamouri, Energy conserving/decaying implicit time-stepping scheme for nonlinear dynamics of three-dimensional beams undergoing finite rotations, Comput. Methods Appl. Mech. Engrg. 191 (37–38) (2002) 4241–4258, http://dx.doi.org/10.1016/s0045-7825(02)00377-8.

[7] I. Romero, F. Armero, An objective finite element approximation of the kinematics of geometrically exact rods and its use in the formulation of an energy-momentum conserving scheme in dynamics, Internat. J. Numer. Methods Engrg. 54 (12) (2002) 1683–1716, http://dx.doi.org/10.1002/nme.486.

[8] E. Zupan, D. Zupan, On conservation of energy and kinematic compatibility in dynamics of nonlinear velocity-based three-dimensional beams, Nonlinear Dynam. 95 (2) (2018) 1379–1394, http://dx.doi.org/10.1007/s11071-018-4634-y.

[9] J. Tomec, G. Jelenić, Analysis of static frictionless beam-to-beam contact using mortar method, Multibody Syst. Dyn. (2022) http://dx.doi.org/10.1007/s11044-022-09823-2.

[10] J. Simo, A finite strain beam formulation. The three-dimensional dynamic problem. Part I, Comput. Methods Appl. Mech. Engrg. 49 (1) (1985) 55–70, http://dx.doi.org/10.1016/0045-7825(85)90050-7.

[11] A. Ibrahimbegović, M.A. Mikdad, Finite rotations in dynamics of beams and implicit time-stepping schemes, Internat. J. Numer. Methods Engrg. 41 (5) (1998) 781–814, http://dx.doi.org/10.1002/(sici)1097-0207(19980315)41:5<781::aid-nme308>3.0.co;2-9.

[12] S. Hante, D. Tumiotto, M. Arnold, A Lie group variational integration approach to the full discretization of a constrained geometrically exact Cosserat beam model, Multibody Syst. Dyn. 54 (1) (2021) 97–123, http://dx.doi.org/10.1007/s11044-021-09807-8.

[13] V. Sonneville, A. Cardona, O. Brüls, Geometrically exact beam finite element formulated on the special Euclidean group SE(3), Comput. Methods Appl. Mech. Engrg. 268 (2014) 451–474, http://dx.doi.org/10.1016/j.cma.2013.10.008.

[14] V. Sonneville, O. Brüls, O.A. Bauchau, Interpolation schemes for geometrically exact beams: A motion approach, Internat. J. Numer. Methods Engrg. 112 (9) (2017) 1129–1153, http://dx.doi.org/10.1002/nme.5548.

[15] V. Sonneville, M. Géradin, Two-field formulation of the inertial forces of a geometrically-exact beam element, Multibody Syst. Dyn. (2022) http://dx.doi.org/10.1007/s11044-022-09867-4.

[16] T. Merlini, M. Morandini, The helicoidal modeling in computational finite elasticity. Part II: Multiplicative interpolation, Int. J. Solids Struct. 42 (3–4) (2005) 1269, http://dx.doi.org/10.1016/j.ijsolstr.2004.09.004.

[17] R.A. Spurrier, Comment on " singularity-free extraction of a quaternion from a direction-cosine matrix", J. Spacecr. Rockets 15 (4) (1978) 255, http://dx.doi.org/10.2514/3.57311.

[18] O.C. Zienkiewicz, R.L. Taylor, J.Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, sixth ed., Butterworth-Heinemann, 2005.

[19] M. Arnold, O. Brüls, Convergence of the generalized-α scheme for constrained mechanical systems, Multibody Syst. Dyn. 85 (2007) 187–202, http://dx.doi.org/10.1007/s11044-007-9084-0.

[20] V. Sonneville, A Geometric Local Frame Approach for Flexible Multibody Systems (Ph.D. thesis), FRIA - Fonds pour la Formation à la Recherche dans l'Industrie et dans l'Agriculture, ULiège - Université de Liège, 2015.

[21] J. Simo, L. Vu-Quoc, On the dynamics in space of rods undergoing large motions — A geometrically exact approach, Comput. Methods Appl. Mech. Engrg. 66 (2) (1988) 125–161, http://dx.doi.org/10.1016/0045-7825(88)90073-4.