An implicit 3D corotational formulation for frictional contact dynamics of beams against rigid surfaces using discrete signed distance fields

Miquel Aguirre, Stéphane Avril

An implicit 3D corotational formulation for frictional contact dynamics of beams against rigid surfaces using discrete signed distance fields

Miquel Aguirre, Stéphane Avril

Mines Saint-Etienne, Univ Lyon, Univ Jean Monnet, INSERM, U 1059 Sainbiose, Centre CIS, F-42023 Saint-Etienne

Abstract

The interaction of beam-like structures against surfaces is a challenging problem with applications in engineering (wheel-rail contact, pipeline-soil interaction, ropes sliding on the seabed) and in medical applications such as surgical planning and training (catheter navigation, aneurysm coiling, stent deployment). This contact problem is traditionally solved using Total Lagrangian beam formulations, which interact against Lagrangian triangulated surfaces. Overall, the computational speed is affected, due to the nonlinearities of the beam formulation and, as well as due to the expensive search algorithms, required to find the close point projection. In the search towards an efficient beam to surface contact algorithm, this paper explores the combined use of 1) a corotational beam formulation, where the motion of the beam element is decomposed in rigid body and pure deformational parts and 2) an implicit description of the surface by means of discrete signed distance fields (SDF), which can be seen as a Lagrangian beam immersed within an Eulerian (rigid) solid. To do so, a previously reported implicit corotational formulation for beam dynamics is modified to account for frictional contact, by means of a penalty term. The new contributions are fully linearized to update the tangent operator and the system is integrated in time by means of the HHT-α method. Overall, a consistent implicit contact dynamics formulation is provided. A SDF, defined in a voxel-type grid, is used to represent implicitly the surface geometry. The SDF values and derivatives are computed at the Lagrangian point of integration of the beam by means of an efficient tensor product of compact, high order, 1D Kernels, as widely used in immersed Fluid-Structure Interaction techniques. A wide variety of validation tests are presented which prove the accuracy and robustness of the proposed algorithm.

Keywords: Contact, 3D Corotational beams, Nonlinear dynamics, Finite Elements, Eulerian Solids, Signed Distance Fields

1. Introduction

Modeling beam to surface contact has applications in many fields of engineering, such as: civil engineering (pipeline-soil interaction [1], drill-string dynamics in oil extraction [2]), offshore and naval engineering (mooring lines and capstains [3, 4], offshore risers sliding on the seabed [5, 6]) and mechanical engineering (deployment of flexible antennas [7], belt-drives used in power transmission [8], vibration of loosened joints [9]). This problem has also attracted the interest in biomedical applications, specifically in the simulation of Minimally Invasive Surgery in which guidewires and catheters are inserted through the opaque patient tissue with limited visual aids [10–13]. In many of the above instances, there is also a need of fast computation, even real-time, such that decisions can be made as the physical process evolves.

The beam to surface contact problem can be accurately solved by combining Total Lagrangian formulations for the beam model, and triangulated or NURBS Lagrangian surfaces for the surface model [14, 15]. The overall process is computationally expensive, due to the nonlinearities involved in the beam formulation but also due to the expensive search algorithms, required to solve a

minimal distance problem to find the close point projection [16]. In the pursue of real-time computations alternative approaches have been used to simplify the beam model, on one hand, and simplify the contact interaction on the other hand. Regarding the beam model, simplified approaches have been used such as one dimensional string equations [17, 18], mass-spring models [19], rigid multibody links [20] or hybrid models including the last two [21]. Regarding contact (or collision) detection, this can be simplified by using bounding boxes or spheres surrounding the target object [22]. Also, specific boundary conditions can be designed, as in the case of sliding beams in small orifices [23–25], however this is limited to a limited number of problems. In cardiovascular applications, some authors assume circular cross sections for the vessels, which allows detecting penetration [26]. Each of these approaches provide faster computational speed at the expense of a decrease in accuracy. On top of this, most of the publications use explicit time integration, which limits the maximum time step size allowed for stability reasons. Alternatively, in the search towards an efficient, yet accurate, beam to surface contact algorithm, this paper explores using the following original contributions: 1) development of a consistent implicit 3D corotational formulation for beam against master surface contact and 2) an implicit description of the surface by means of discrete signed distance fields (SDF) defined in a voxel type background mesh. The overall approach can be seen as a Lagrangian beam immersed within an Eulerian (rigid) solid. The use of level sets and/or Eulerian solids in computational contact, has been explored before for 3D solids in [27–32] but, to the best of the authors knowledge, this is the first application where it is combined with a Lagrangian beam.

Corotational beams assume a split of the beam deformation into rigid and purely deformational parts and have been largely used in the last decades [33–40]. This formulation allows modeling all the nonlinearities in the rigid rotation, and using linear constitutive models for the deformational part. In many thin beams applications, this allows for an accurate, yet computationally efficient solution, and makes it one of the preferred choices in real-time simulation [41]. Among the challenges faced in corotational beam formulations, comes the parametrization of 3D rotations. Numerous papers use the 'rotational vector' which allows for an additive update of the rotations but it limits its use to rotations under 2π. Instead, Battini and co-workers [38] have largely explored the use of a spatial form of the incremental rotation vector, which still allows for additive updates, but only at the level of the iterative corrections. In [40], Le and coauthors extended this formulation to dynamics, comparing very well against the total Lagrangian formulation of Simo and Vu-Quoc [42, 43]. Regarding corotational beams and contact, the only contribution in implicit dynamics formulation is found in [44] but it is limited to 2D planar elements. This paper proposes a 3D implicit formulation of frictional dynamic contact by building up from the formulation of Le and coauthors [40]. To do so, a friction penalty term is added into the variational formulation, which is integrated along the beam centerline in spatial coordinates and gives rise to a distributed line contact force [45–48]. The formulation assumes that the normal gap with respect to the surface is given, as well as its spatial gradient and Hessian (which will be computed via interpolation of a SDF). To facilitate convergence of the Newton Raphson algorithm, the normal contact pressure is regularized via a quadratic contact potential [48, 49], whereas the stick-slip behavior is regularized using a square root function [16]. The semi-discrete system of equations is solved in time using HHT$-\alpha$ method [50]. In addition to the contact formulation, a new description of the corotational beam kinematics is proposed, traditionally described by means of change of variables between a moving and a global reference frame [35, 38]. To do so, finite-elasticity three dimensional theory is used, as proposed in [51] for geometrically exact beams. By particularizing it to the corotational formulation, interesting links are found with geometrically exact beam kinematics, while making the previous developments in corotational beams still fully valid.

Regarding contact detection, the distance from the beam to the surface is found by interpolating a discrete SDF (level set with gradient of norm one) given in a background voxel-type mesh, to the Lagrangian point of integration. Specifically, the interpolation is carried out using kernel functions as used in Material Point Method [52, 53] and immersed Fluid-Structure Interaction techniques

[54–60]. Taking advantage of the Cartesian structure of the voxel mesh, a very efficient tensor product of 1D kernels is used, which allows precomputing the interpolation stencils. Specifically, the spline based 1D kernel proposed in [59] is chosen, which ensures continuity of the first and second spatial derivatives, as required by the beam contact formulation. The paper analyzes numerically the accuracy of the interpolation technique in approximating the SDF, its gradient and Hessian by comparing it against an analytical solution. Next, five examples of beam to rigid surface contact are shown to prove the accuracy and robustness of the proposed formulation.

The remainder of the paper is organized as follows. Section 2 summarizes the applied beam formulation. Section 3 presents the frictional contact penalty formulation and its linearization. Section 4 presents the interpolation used for the discrete SDF and the numerical analysis of its order of accuracy. Section 5 presents the time integration method and computational aspects. Section 6 a series of examples. Finally, section 7 presents the concluding remarks.

2. Beam formulation

We largely follow the notation in [50] in which a fully consistent tangent operator is proposed for the dynamics of co-rotational beams. In what follows, a global reference system is defined in the material configuration, which is given by the triad of Cartesian vectors $\boldsymbol{E}_i$ $(i = 1, 2, 3)$. Similarly, a reference system is defined in the spatial (Eulerian) configuration, given by the triad $\boldsymbol{e}_i$ $(i = 1, 2, 3)$. As in [61], both triads are made coincident, i.e. $\boldsymbol{E}_i = \delta_{ij}\boldsymbol{e}_j$ with δ_{ij} being the Kronecker delta. They are denoted with different symbols to differentiate between material and spatial objects.

2.1. Parametrization of finite rotations

Rotations are parametrized by means of the rotational vector [38, 62, 63], defined as

$$\boldsymbol{\theta} = \theta \boldsymbol{n}, \tag{1}$$

with $\boldsymbol{n}$ the axis of rotation and θ the rotation angle given by

$$\theta = \sqrt{\boldsymbol{\theta} \cdot \boldsymbol{\theta}}. \tag{2}$$

The relationship between the rotation angle and the rotation matrix is provided by the Rodrigues formula, given by

$$\boldsymbol{R}(\boldsymbol{\theta}) = \boldsymbol{I} + \frac{\sin\theta}{\theta}\boldsymbol{S}(\boldsymbol{\theta}) + \frac{1-\cos\theta}{\theta^2}\boldsymbol{S}(\boldsymbol{\theta})\boldsymbol{S}(\boldsymbol{\theta}) = \exp(\boldsymbol{S}(\boldsymbol{\theta})), \tag{3}$$

where $\boldsymbol{I}$ is the 3×3 identity matrix and $\boldsymbol{S}(\boldsymbol{\theta})$ is the skew-symmetric matrix of $\boldsymbol{\theta}$, given by

$$\boldsymbol{S}(\boldsymbol{\theta}) = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix}. \tag{4}$$

The variation of $\boldsymbol{R}(\boldsymbol{\theta})$ is obtained by first constructing a superimposed infinitesimal rotation [43]. This is done by using the exponential map (3) and by noticing that, since $\boldsymbol{R}$ is a two point tensor, the superimposed rotation needs to be a spatial quantity, i.e.

$$\boldsymbol{R}_\epsilon = \exp(\epsilon \boldsymbol{S}(\boldsymbol{w}))\boldsymbol{R} \tag{5}$$

and, therefore,

$$\delta\boldsymbol{R} = \frac{d}{d\epsilon}\boldsymbol{R}_\epsilon\big|_{\epsilon=0} = \boldsymbol{S}(\delta\boldsymbol{w})\boldsymbol{R}, \tag{6}$$

with $\delta\boldsymbol{w}$ and $\boldsymbol{S}(\delta\boldsymbol{w})$ spatial objects defined as

$$\delta\boldsymbol{w} = w_i\boldsymbol{e}_i, \tag{7}$$
$$\boldsymbol{S}(\delta\boldsymbol{w}) = [\boldsymbol{S}(\delta\boldsymbol{w})]_{ij}\boldsymbol{e}_i \otimes \boldsymbol{e}_j, \tag{8}$$

where $\delta\boldsymbol{w}$ is named the spatial spin vector and are infinitesimal spatial rotations superimposed to the rotation $\boldsymbol{R}$. Instead of a spatial spin vector, a material one can be used, defined as

$$\delta\boldsymbol{W} = \delta W_i\boldsymbol{E}_i, \tag{9}$$
$$\boldsymbol{S}(\delta\boldsymbol{W}) = [\boldsymbol{S}(\delta\boldsymbol{W})]_{ij}\boldsymbol{E}_i \otimes \boldsymbol{E}_j, \tag{10}$$

where $\delta\boldsymbol{W}$ and $\boldsymbol{S}(\delta\boldsymbol{W})$ are the pull back of $\delta\boldsymbol{w}$ and $\boldsymbol{S}(\delta\boldsymbol{w})$, respectively, i.e.

$$\delta\boldsymbol{W} = \boldsymbol{R}^T\delta\boldsymbol{w}, \tag{11}$$
$$\boldsymbol{S}(\delta\boldsymbol{W}) = \boldsymbol{R}^T\boldsymbol{S}(\delta\boldsymbol{w})\boldsymbol{R}. \tag{12}$$

Using (12) into (6) yields [43]

$$\delta\boldsymbol{R} = \boldsymbol{R}^T\boldsymbol{S}(\delta\boldsymbol{W}). \tag{13}$$

The relationship between the spatial spin vector and the variation of the rotational vector are given by

$$\delta\boldsymbol{w} = \boldsymbol{T}_s(\boldsymbol{\theta})\delta\boldsymbol{\theta}, \tag{14a}$$
$$\delta\boldsymbol{\theta} = \boldsymbol{T}_s^{-1}(\boldsymbol{\theta})\delta\boldsymbol{w}, \tag{14b}$$

with

$$\boldsymbol{T}_s(\boldsymbol{\theta}) = \boldsymbol{I} + \frac{1-\cos\theta}{\theta^2}\boldsymbol{S}(\boldsymbol{\theta}) + \frac{\theta-\sin\theta}{\theta^3}\boldsymbol{S}(\boldsymbol{\theta})\boldsymbol{S}(\boldsymbol{\theta}), \tag{15a}$$
$$\boldsymbol{T}_s^{-1}(\boldsymbol{\theta}) = \frac{(\theta/2)}{\tan(\theta/2)}\boldsymbol{I}_{3\times3} + \frac{1}{\theta^2}\left(1 - \frac{(\theta/2)}{\tan(\theta/2)}\right)\boldsymbol{\theta}\boldsymbol{\theta}^T - \frac{1}{2}S(\boldsymbol{\theta}). \tag{15b}$$

The spatial form of the angular velocity and acceleration can be obtained from equation (6) as

$$\boldsymbol{S}(\dot{\boldsymbol{w}}) = \dot{\boldsymbol{R}}\boldsymbol{R}^T, \tag{16a}$$
$$\boldsymbol{S}(\ddot{\boldsymbol{w}}) = \ddot{\boldsymbol{R}}\boldsymbol{R}^T + \dot{\boldsymbol{R}}\dot{\boldsymbol{R}}^T, \tag{16b}$$

with $\dot{(\cdot)} = \frac{d}{dt}(\cdot)$ and $\ddot{(\cdot)} = \frac{d^2}{dtdt}(\cdot)$. Finally, the spatial angular velocity can be calculated from the time derivative of the rotational vector via equation (14b) to give

$$\dot{\boldsymbol{w}} = \boldsymbol{T}_s(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \tag{17}$$

2.2. Corotational beam kinematics

As an addition to the formulation presented in [38], the corotational beam kinematics is described using finite-elasticity three dimensional theory as proposed in [51] for geometrically exact beams.

A straight beam with circular cross section is defined with deformed configuration Ω and material configuration Ω_0. Two additional configurations, $\bar{\Omega}_0$ and $\bar{\Omega}$ are also defined. These configurations contain, respectively, the positions of the beam in the initial and current positions absent of rigid body motions. Both configurations are assumed to be very close, i.e. $\bar{\Omega} \simeq \bar{\Omega}_0$ and therefore they are uniquely denoted by $\bar{\Omega}$. The basis vectors of $\bar{\Omega}$ are $\bar{\boldsymbol{E}}_i$ $(i = 1, 2, 3)$ and are made coincident with the material and spatial basis, i.e. $\bar{\boldsymbol{E}}_i = \delta_{ij}\boldsymbol{E}_j$, $\bar{\boldsymbol{E}}_i = \delta_{ij}\boldsymbol{e}_j$.

To define the beam position, an orthogonal system $\boldsymbol{T}_i$ $(i = 1, 2, 3)$ is placed at the centroid of the beam cross section in the material configuration. This system defines the local coordinates $\{\xi_1, \xi_2, \xi_3\}$ with the arc-length parameter $s = \xi_1 \in [0, l_0]$ that coincides with the axis of the beam, defined by $\boldsymbol{T}_1$. The cross sections of the beam lie in planes defined by the basis vectors $\{\boldsymbol{T}_2, \boldsymbol{T}_3\}$, which depend on the arc-length parameter s. Accordingly, the beam is deformed into a current configuration Ω, with an orthonormal frame $\boldsymbol{t}_i$ defined also at the centroid of the beam cross section. These vectors are defined through the following transformations

$$\boldsymbol{T}_i(s) = \boldsymbol{\Lambda}_0(s)\bar{\boldsymbol{E}}_i; \quad \boldsymbol{\Lambda}_0(s) = \boldsymbol{T}_i(s) \otimes \bar{\boldsymbol{E}}_i = [\boldsymbol{\Lambda}_0]_{ij}\, \boldsymbol{E}_i \otimes \bar{\boldsymbol{E}}_j, \tag{18}$$

$$\boldsymbol{t}_i(s,t) = \boldsymbol{\Lambda}(s,t)\bar{\boldsymbol{E}}_i; \quad \boldsymbol{\Lambda}(s,t) = \boldsymbol{t}_i(s,t) \otimes \bar{\boldsymbol{E}}_i = [\boldsymbol{\Lambda}]_{ij}\, \boldsymbol{e}_i \otimes \bar{\boldsymbol{E}}_j. \tag{19}$$

with $\boldsymbol{\Lambda}_0$, $\boldsymbol{\Lambda}$ being two point tensors that give, respectively, orthonormal triads $\boldsymbol{T}_i$, $\boldsymbol{t}_i$ (material and spatial objects, respectively) as a function of the basis vectors $\bar{\boldsymbol{E}}_i$. By comparing relationships (18) and (19), a map between $\boldsymbol{T}_i$ and $\boldsymbol{t}_i$ can be defined as

$$\boldsymbol{t}_i(s,t) = \boldsymbol{R}(s,t)\boldsymbol{T}_i(s); \quad \boldsymbol{R}(s,t) = \boldsymbol{\Lambda}(s,t)\boldsymbol{\Lambda}_0^{-T}(s) = \boldsymbol{t}_i \otimes \boldsymbol{T}_i = [\boldsymbol{R}]_{ij}\, \boldsymbol{e}_i \otimes \boldsymbol{E}_i, \tag{20}$$

which is the mapping from the material to the current configuration which in fact uniquely describes the rotation of the cross section. Using the above definitions, the material and spatial positions of the beam are given by

$$\boldsymbol{X}(s, \xi_2, \xi_3) = \boldsymbol{X}_0(s) + \xi_2 \boldsymbol{T}_2(s) + \xi_3 \boldsymbol{T}_3(s), \tag{21}$$

$$\boldsymbol{x}(s, \xi_2, \xi_3) = \boldsymbol{x}_0(s) + \xi_2 \boldsymbol{t}_2(s) + \xi_3 \boldsymbol{t}_3(s), \tag{22}$$

where $\boldsymbol{X}_0(s)$, $\boldsymbol{x}_0(s)$ are, respectively, the position of the beam centerline in the material and current configurations. In order to remove rigid body rotations, two mappings are defined between $\bar{\Omega}$ and Ω_0 and between $\bar{\Omega}$ and Ω. To do so, two frames that move rigidly with the beam axis are considered [37]. This is a material frame $\boldsymbol{V}_i$ $(i = 1, 2, 3)$ and a spatial frame $\boldsymbol{v}_i$ $(i = 1, 2, 3\ )$. In order to compute $\boldsymbol{v}_i$ $(i = 1, 2, 3)$, the procedure detailed in [38] is used,

$$\boldsymbol{v}_1(t) = \frac{\boldsymbol{x}_0(l_0, t) - \boldsymbol{x}_0(0, t)}{\|\boldsymbol{x}_0(l_0, t) - \boldsymbol{x}_0(0, t)\|}; \quad \boldsymbol{v}_3(t) = \frac{\boldsymbol{v}_1(t) \times \boldsymbol{p}(t)}{\|\boldsymbol{v}_1(t) \times \boldsymbol{p}(t)\|}; \quad \boldsymbol{v}_2(t) = \boldsymbol{v}_3(t) \times \boldsymbol{v}_1(t), \tag{23}$$

with the auxiliary vector $\boldsymbol{p}$ given by

$$\boldsymbol{p}(t) = \frac{1}{2}(\boldsymbol{t}_2(0, t) + \boldsymbol{t}_2(l_0, t)), \tag{24}$$

with an identical procedure to obtain, i.e $\boldsymbol{V}_i = \boldsymbol{v}_i(0)$ $(i = 1, 2, 3)$. The above basis vectors allow defining the following mappings

$$\boldsymbol{V}_i = \boldsymbol{R}_e^0 \bar{\boldsymbol{E}}_i; \quad \boldsymbol{R}_e^0 = \boldsymbol{V}_i \otimes \bar{\boldsymbol{E}}_i = \left[\boldsymbol{R}_e^0\right]_{ij} \boldsymbol{E}_i \otimes \bar{\boldsymbol{E}}_j, \tag{25}$$

$$\boldsymbol{v}_i(t) = \boldsymbol{R}_e(t)\bar{\boldsymbol{E}}_i; \quad \boldsymbol{R}_e(t) = \boldsymbol{v}_i(t) \otimes \bar{\boldsymbol{E}}_i = [\boldsymbol{R}_e]_{ij}\, \boldsymbol{e}_i \otimes \bar{\boldsymbol{E}}_j, \tag{26}$$

with $\boldsymbol{R}_e^0$ being a two point tensor that goes from $\bar{\Omega}$ to Ω_0 and $\boldsymbol{R}_e$ a two point tensor that goes from $\bar{\Omega}$ to Ω. It is worth noticing that $\boldsymbol{R}_e(0) = \boldsymbol{R}_e^0$ and that neither $\boldsymbol{R}_e^0$ nor $\boldsymbol{R}_e$ depend on the arc-length parameter s, and hence define a rigid body rotation of the beam segment. These mappings allow defining the cross sectional triads in the rigid body free configuration $\bar{\Omega}$, i.e.

$$\bar{\boldsymbol{T}}_i(s) = (\boldsymbol{R}_e^0)^T \boldsymbol{T}_i(s), \tag{27}$$

$$\bar{\boldsymbol{t}}_i(s,t) = \boldsymbol{R}_e^T(t)\boldsymbol{t}_i(s,t). \tag{28}$$

Equivalently to equation (20), a mapping can be defined between $\bar{\boldsymbol{T}}_i$ and $\bar{\boldsymbol{t}}_i$ as

$$\bar{\boldsymbol{t}}_i(s,t) = \bar{\boldsymbol{R}}(s,t)\bar{\boldsymbol{T}}_i(s), \quad \bar{\boldsymbol{R}}(s,t) = \boldsymbol{R}_e^T(t)\boldsymbol{R}(s,t)\boldsymbol{R}_e^0 = \bar{\boldsymbol{t}}_i(s,t) \otimes \bar{\boldsymbol{T}}_i(s) = [\bar{\boldsymbol{R}}]_{ij}\bar{\boldsymbol{E}}_i \otimes \bar{\boldsymbol{E}}_j. \tag{29}$$

where $\bar{\boldsymbol{R}}$ is a small rotation of the cross sectional vectors $\bar{\boldsymbol{T}}_i$ in $\bar{\Omega}$, now absent of rigid body motions. Assuming the material beam as straight, the rotation matrix $\boldsymbol{\Lambda}_0$ can be made coincident with $\boldsymbol{R}_e^0$, this is $\boldsymbol{\Lambda}_0(s) = \boldsymbol{R}_e^0$, while $\boldsymbol{T}_i = \boldsymbol{V}_i$ and $\bar{\boldsymbol{T}}_i = \bar{\boldsymbol{E}}_i$. Taking all this into account, and omitting the dependence with respect to t for clarity purposes, the material and spatial positions of the beam centerline are given by

$$\boldsymbol{X}_0(s) = \boldsymbol{X}_0(0) + s\boldsymbol{V}_1, \tag{30}$$

$$\boldsymbol{x}_0(s) = \boldsymbol{x}_0(0) + s\boldsymbol{v}_1 + u_1(s)\boldsymbol{v}_1 + u_2(s)\boldsymbol{v}_2 + u_3(s)\boldsymbol{v}_3, \tag{31}$$

where $u_i(s)$ $(i = 1, 2, 3)$ are the positions of the beam centerline respect the rigid frame $\boldsymbol{v}_i$. Given the definition of $\boldsymbol{v}_1$ in equation (23), the displacements $u_i(s)$ follow

$$u_1(0) = 0; \quad u_1(l_0) = \bar{u}; \quad \bar{u} = l_n - l_0, \tag{32a}$$

$$u_2(0) = 0; \quad u_2(l_0) = 0, \tag{32b}$$

$$u_3(0) = 0; \quad u_3(l_0) = 0. \tag{32c}$$

By removing rigid body rotations using $\boldsymbol{R}_e^0$, $\boldsymbol{R}_e$ and rigid body translations using the position of the centerline node at $s = 0$, the position of the beam points in the $\bar{\Omega}$ configuration can be defined as

$$\bar{\boldsymbol{X}}(s, \xi_2, \xi_3) = \bar{\boldsymbol{X}}_0(s) + \xi_2\bar{\boldsymbol{E}}_2 + \xi_3\bar{\boldsymbol{E}}_3; \quad \bar{\boldsymbol{X}}_0(s) = s\bar{\boldsymbol{E}}_1 \tag{33}$$

$$\bar{\boldsymbol{x}}(s, \xi_2, \xi_3) = \bar{\boldsymbol{x}}_0(s) + \xi_2\bar{\boldsymbol{R}}(s)\bar{\boldsymbol{E}}_2 + \xi_3\bar{\boldsymbol{R}}(s)\bar{\boldsymbol{E}}_3; \quad \bar{\boldsymbol{x}}_0(s) = s\bar{\boldsymbol{E}}_1 + u_1(s)\bar{\boldsymbol{E}}_1 + u_2(s)\bar{\boldsymbol{E}}_2 + u_3(s)\bar{\boldsymbol{E}}_3. \tag{34}$$

which defines the beam deformational displacement $\bar{\boldsymbol{u}} = \bar{\boldsymbol{x}} - \bar{\boldsymbol{X}}$ in $\bar{\Omega}$ as

$$\bar{\boldsymbol{u}}(s, \xi_2, \xi_3) = \bar{\boldsymbol{u}}_0(s) + \xi_2(\bar{\boldsymbol{R}}(s) - \boldsymbol{I})\bar{\boldsymbol{E}}_2 + \xi_3(\bar{\boldsymbol{R}}(s) - \boldsymbol{I})\bar{\boldsymbol{E}}_3; \quad \bar{\boldsymbol{u}}_0(s) = u_1(s)\bar{\boldsymbol{E}}_1 + u_2(s)\bar{\boldsymbol{E}}_2 + u_3(s)\bar{\boldsymbol{E}}_3, \tag{35}$$

which is assumed to be very small, i.e. $\bar{\boldsymbol{x}} \simeq \bar{\boldsymbol{X}}$. Via equation (33), it is worth noticing that $s = \bar{X}_1$, $\xi_\alpha = \bar{X}_\alpha$.

It is useful now to write the deformation gradient using finite-elasticity three dimensional theory. Following Auricchio and coauthors [51] this is defined as

$$\boldsymbol{F}(s) = \frac{\partial \boldsymbol{x}}{\partial \xi_i} \otimes \frac{\partial \xi_i}{\partial \boldsymbol{X}} = \boldsymbol{R}(s) + \boldsymbol{a}(s) \otimes \boldsymbol{V}_1, \tag{36}$$

where equation (20) has been used alongside with $\boldsymbol{T}_i = \boldsymbol{V}_i$. The vector $\boldsymbol{a}$ is defined as

$$\boldsymbol{a}(s) = \boldsymbol{\gamma}(s) + \xi_\alpha \boldsymbol{\kappa}_\alpha(s), \tag{37}$$

with $\boldsymbol{\gamma}$ and $\boldsymbol{\kappa}_\alpha$ defined as (see [51])

$$\boldsymbol{\gamma}(s) = \boldsymbol{x}_0'(s) - \boldsymbol{t}_1(s), \tag{38}$$

$$\boldsymbol{\kappa}_\alpha(s) = \boldsymbol{a}_\alpha'(s). \tag{39}$$

where $(\cdot)' = \frac{d(\cdot)}{ds}$.

Remark 2.2.1. In comparison with [51], it is worth noticing that $\boldsymbol{R}$ is the rotation from the orthonormal basis $\boldsymbol{V}_i$ in the initial configuration (straight, but not parallel to the $\bar{\boldsymbol{E}}_i$ axis) to the orthonormal triad $\boldsymbol{t}_i$ in the current configuration. In [51], the orthonormal triad in the material configuration coincides with the reference axes, i.e. $\boldsymbol{V}_i = \delta_{ij}\bar{\boldsymbol{E}}_j$ and, therefore, $\boldsymbol{\Lambda}_0 = \boldsymbol{I}$ and, via equation (20), $\boldsymbol{R} = \boldsymbol{\Lambda}$.

The deformation gradient (36) can be decomposed using a left decomposition to yield

$$\boldsymbol{F}(s) = \boldsymbol{R}(s)\boldsymbol{A}^r(s); \quad \boldsymbol{A}^r(s) = \boldsymbol{I} + \boldsymbol{a}^r(s) \otimes \boldsymbol{V}_1 \tag{40}$$

with $\boldsymbol{a}^r$ being, as defined in [51], the *rotate-back* of $\boldsymbol{a}$ in the material configuration, i.e.

$$\boldsymbol{a}^r(s) = \boldsymbol{R}(s)^T \boldsymbol{a}(s). \tag{41}$$

Equation (40) shows that the deformation gradient can be decomposed by a pure rotation, on the left, followed by a pure material stretch, on the right.

Alternatively, another interesting result can be obtained for the corotational formulation, which links the formalisms described in [51], with the decomposition between rigid body motions and small deformational displacements as described in, for example, [35, 38]. In fact, using the rigid mappings defined in equations (25), (26), the following multiplicative decomposition can be carried out

$$\boldsymbol{F}(s) = \boldsymbol{R}_e \bar{\boldsymbol{F}}(s) (\boldsymbol{R}_e^0)^T \tag{42}$$

with $\bar{\boldsymbol{F}}$ being the deformation gradient in the configuration $\bar{\Omega}$, which is defined by

$$\bar{\boldsymbol{F}}(s) = \bar{\boldsymbol{R}}(s) + \bar{\boldsymbol{a}}(s) \otimes \bar{\boldsymbol{E}}_1 \tag{43}$$

with $\bar{\boldsymbol{a}}$ being now the *rigid-rotate-back* of $\boldsymbol{a}$ to the configuration $\bar{\Omega}$, i.e.

$$\bar{\boldsymbol{a}}(s) = \boldsymbol{R}_e^T \boldsymbol{a}(s) = \bar{\boldsymbol{\gamma}}(s) + \xi_\alpha \bar{\boldsymbol{\kappa}}_\alpha(s), \tag{44}$$

with

$$\bar{\boldsymbol{\gamma}} = \bar{\boldsymbol{x}}_0'(s) - \bar{\boldsymbol{t}}_1, \tag{45}$$

$$\bar{\boldsymbol{\kappa}}_\alpha = \boldsymbol{R}_e^T \boldsymbol{t}_\alpha'(s) = (\boldsymbol{R}_e^T \boldsymbol{t}_\alpha(s))' = \bar{\boldsymbol{a}}_\alpha' = \bar{\boldsymbol{R}}'(s) \bar{\boldsymbol{E}}_\alpha. \tag{46}$$

where the fact that $\boldsymbol{R}_e$ is independent of the arc-length coordinate s has been used.

Remark 2.2.2. Differently from $\boldsymbol{a}^r$ in (41), $\bar{\boldsymbol{a}}$ has been obtained in (44) by removing the rigid body motions via $\boldsymbol{R}_e$ and hence the denomination *rigid-rotate-back*. The relation between the two is given by $\boldsymbol{a}^r = \boldsymbol{R}_e^0 \bar{\boldsymbol{R}}^T \bar{\boldsymbol{a}}$.

Remark 2.2.3. Equation (40) decomposes the deformation gradient into one material tensor, $\boldsymbol{A}^r$, followed by one rotation, $\boldsymbol{R}$, both dependent on the arc-length parameter s. On the other hand, equation (42) decomposes the deformation gradient into one rotation $(\boldsymbol{R}_e^0)^T$, followed by one deformation gradient $\bar{\boldsymbol{F}}$, followed by another rotation $\boldsymbol{R}_e$. In this case only $\bar{\boldsymbol{F}}$ is dependent on the arc-length coordinate, which shows why the corotational formulations allow writing the variational equations purely in $\bar{\Omega}$.

Assuming small rotations in $\bar{\Omega}$, the rotation tensor $\bar{\boldsymbol{R}}$ can be parametrized by a first order approximation of (3),i.e.

$$\bar{\boldsymbol{R}}(s) = \boldsymbol{I} + \boldsymbol{S}(\bar{\boldsymbol{\Theta}}(s)) \tag{47}$$

where $\bar{\boldsymbol{\Theta}}$ is the rotation vector in the configuration $\bar{\Omega}$. At the same time, assuming also small displacements, it can be proved that the gradient of the displacements $\bar{\boldsymbol{u}}$ defined in (35) is given by (see Appendix A)

$$\nabla \bar{\boldsymbol{u}} = \boldsymbol{S}(\bar{\boldsymbol{\Theta}}) + \bar{\boldsymbol{a}} \otimes \bar{\boldsymbol{E}}_1. \tag{48}$$

Using equations (47) and (48) into (43) yields the following approximation of $\bar{\boldsymbol{F}}$

$$\bar{\boldsymbol{F}} \simeq \boldsymbol{I} + \boldsymbol{\varepsilon}(\nabla \bar{\boldsymbol{u}}) + \boldsymbol{\Omega}(\nabla \bar{\boldsymbol{u}}) \tag{49}$$

which is the additive decomposition of the deformation gradient.

Remark 2.2.4. Equations (42) and (49) show that, in the corotational approach, the global deformation gradient can be seen as the multiplicative decomposition of a rigid rotation (from Ω_0 to $\bar{\Omega}$), followed by a small deformation (in $\bar{\Omega}$), followed by another rigid rotation (from $\bar{\Omega}$ to Ω).

From equation (42), the Green-Lagrange strain tensor can be computed as

$$\boldsymbol{E} = \boldsymbol{R}_e^0 \bar{\boldsymbol{E}} (\boldsymbol{R}_e^0)^T, \tag{50}$$

with

$$\bar{\boldsymbol{E}} = \frac{1}{2}(\bar{\boldsymbol{F}}^T \bar{\boldsymbol{F}} - \boldsymbol{I}) = \bar{E}_{11} \bar{\boldsymbol{E}}_1 \otimes \bar{\boldsymbol{E}}_1 + \bar{E}_{1\alpha} \bar{\boldsymbol{E}}_1 \otimes \bar{\boldsymbol{E}}_\alpha + \bar{E}_{\alpha 1} \bar{\boldsymbol{E}}_\alpha \otimes \bar{\boldsymbol{E}}_1, \tag{51}$$

and

$$\bar{E}_{11} = \frac{1}{2}(\bar{\boldsymbol{a}} \cdot \bar{\boldsymbol{a}}) + (\bar{\boldsymbol{a}} \cdot \bar{\boldsymbol{t}}_1), \tag{52a}$$

$$\bar{E}_{1\alpha} = \bar{E}_{\alpha 1} = \frac{1}{2}(\bar{\boldsymbol{a}} \cdot \bar{\boldsymbol{t}}_\alpha), \tag{52b}$$

where equation (43) has been used alongside equation (29). The total Green-Lagrange strain tensor depends only on $\bar{\boldsymbol{a}}$, this is the derivative of the local centerline displacement, $\bar{\boldsymbol{u}}_0'$, the local rotation matrix $\bar{\boldsymbol{R}}$ and its derivative along the centerline $\bar{\boldsymbol{R}}'$ (see equations (44,45,46)). Assuming again small strains, via equation (49), the Green-Lagrange strain tensor can be written as

$$\bar{\boldsymbol{E}} = \frac{1}{2}(\bar{\boldsymbol{F}}^T \bar{\boldsymbol{F}} - \boldsymbol{I}) \simeq \boldsymbol{\varepsilon}(\nabla \bar{\boldsymbol{u}}) \tag{53}$$

which proves that by assuming small strains and small rotations in $\bar{\Omega}$, $\bar{\boldsymbol{E}}$ approximates to the small strain tensor $\boldsymbol{\varepsilon}(\nabla \bar{\boldsymbol{u}})$. Using equations (50) (53), the variation of the internal energy can be computed as

$$\delta W_{int} = \int_{\Omega_0} \boldsymbol{S} : \delta \boldsymbol{E} \, d\Omega_0 = \int_{\bar{\Omega}} \bar{\boldsymbol{\sigma}} : \delta \bar{\boldsymbol{\varepsilon}} \, d\bar{\Omega} = \int_{\bar{\Omega}} \left[\bar{\sigma}_{11} \delta \bar{\varepsilon}_{11} + \bar{\sigma}_{1\alpha} \delta \bar{\varepsilon}_{1\alpha} + \bar{\sigma}_{\alpha 1} \delta \bar{\varepsilon}_{\alpha 1} \right] d\bar{\Omega} \tag{54}$$

where the notation $\bar{\boldsymbol{\varepsilon}} \equiv \boldsymbol{\varepsilon}(\nabla \bar{\boldsymbol{u}})$ has been used alongside the definition of

$$\bar{\boldsymbol{\sigma}} = \left(\boldsymbol{R}_e^0\right)^T \boldsymbol{S} \boldsymbol{R}_e^0. \tag{55}$$

Therefore, the internal virtual work can be written in terms of linear elasticity components in $\bar{\Omega}$, which contributes in the efficiency of the corotational formulation, as all the nonlinearities are taken care of via the elemental rigid body rotation $\boldsymbol{R}_e$. Specifically, the following relationship between stresses and strains in $\bar{\Omega}$ is used

$$\bar{\boldsymbol{\sigma}} = \lambda \, \mathrm{tr}(\bar{\boldsymbol{\varepsilon}}) \boldsymbol{I} + 2G \bar{\boldsymbol{\varepsilon}} \tag{56}$$

where λ, G are, respectively, the first and second Lamé parameters. Using the approximation $E \simeq \lambda + 2G$ [51], the stress components contributing into equation (54) can be written as

$$\bar{\sigma}_{11} = (\lambda + 2G)\bar{\varepsilon}_{11} \simeq E \bar{\varepsilon}_{11}, \tag{57a}$$

$$\bar{\sigma}_{1\alpha} = \bar{\sigma}_{\alpha 1} = 2G \bar{\varepsilon}_{1\alpha} = 2G \bar{\varepsilon}_{\alpha 1}, \tag{57b}$$

where E is the Young's modulus. Combining equations (54), (57), the internal energy of the beam element can be written as

$$W_{int} = \int_{l_0} \left(\frac{1}{2} E \int_A \bar{\varepsilon}_{11}^2 \, dA + \frac{1}{2} G \int_A \left((2\bar{\varepsilon}_{12})^2 + (2\bar{\varepsilon}_{13})^2 \right) dA \right) ds. \tag{58}$$

where $\bar{\Omega} = l_0 A$ alongside the fact that the cross sectional area remains unchanged have been used. Finally, and before moving to the Finite Element discretization, it is useful to derive the relationship between the variation of the different rotation tensors involved in the corotational formulation. From equation (29), the following relationship is obtained

$$\bar{\boldsymbol{R}}(s) = \boldsymbol{R}_e^T \boldsymbol{R}(s) \boldsymbol{R}_e^0. \tag{59}$$

The variations of the above rotation tensors are obtained via equation (6). Omitting from now on the dependence on s, t for clarity, this is

$$\delta \boldsymbol{R} = \boldsymbol{S}(\delta \boldsymbol{w}) \boldsymbol{R}, \tag{60}$$
$$\delta \boldsymbol{R}_e = \boldsymbol{S}(\delta \boldsymbol{w}_e) \boldsymbol{R}_e, \tag{61}$$
$$\delta \bar{\boldsymbol{R}} = \boldsymbol{S}(\delta \bar{\boldsymbol{W}}) \bar{\boldsymbol{R}}. \tag{62}$$

The relationship between these variations is key in the development of the co-rotational formulation. To do so, the variations $\delta \boldsymbol{w}$ and $\delta \boldsymbol{w}_e$ are firstly *rigidly rotated-back* to the local configuration, i.e.

$$\delta \boldsymbol{W} = \boldsymbol{R}_e^T \delta \boldsymbol{w}, \tag{63}$$
$$\delta \boldsymbol{W}_e = \boldsymbol{R}_e^T \delta \boldsymbol{w}_e, \tag{64}$$

which after some algebra (see [38] for details) and in combination with equation (59) yields the following relationship

$$\delta \bar{\boldsymbol{W}} = \delta \boldsymbol{W} - \delta \boldsymbol{W}_e. \tag{65}$$

2.3. Finite Element Discretization

The beam centerline is discretized in N initially straight elements of length $l_0^{(e)}$ such that $l_0 \simeq \cup_N l_0^{(e)}$. [1] Each of these segments is defined by two nodes $a = 1, 2$ with material positions $\boldsymbol{X}_a$ and spatial positions $\boldsymbol{x}_a = \boldsymbol{X}_a + \boldsymbol{u}_a$, with $\boldsymbol{u}_a$ being the nodal displacement. The cross section rotation at the node is defined by the nodal rotation $\boldsymbol{R}_a$, such that

$$\boldsymbol{u}_1 = \boldsymbol{u}_0(0); \quad \boldsymbol{R}_1 = \boldsymbol{R}(0), \tag{66}$$
$$\boldsymbol{u}_2 = \boldsymbol{u}_0(l_0); \quad \boldsymbol{R}_2 = \boldsymbol{R}(l_0), \tag{67}$$

(see Figure 1). The rotation matrices $\boldsymbol{R}_a$ are parametrized via the rotational vector $\boldsymbol{\theta}_a$, while their variation requires the spatial spin vector $\boldsymbol{w}_a$, i.e.

$$\boldsymbol{R}_a = \exp(\boldsymbol{S}(\boldsymbol{\theta_a})); \quad \delta \boldsymbol{R}_a = \boldsymbol{S}(\delta \boldsymbol{w}_a) \boldsymbol{R}_a, \quad (a = 1, 2) \tag{68}$$

which defines the elemental vector of global displacements and rotations, $\boldsymbol{d}$, and their variation $\delta \boldsymbol{d}$

$$\boldsymbol{d} = \left[\begin{array}{cccc} \boldsymbol{u}_1^T & \boldsymbol{\theta}_1^T & \boldsymbol{u}_2^T & \boldsymbol{\theta}_2^T \end{array} \right]^T; \quad \delta \boldsymbol{d} = \left[\begin{array}{cccc} \delta \boldsymbol{u}_1^T & \delta \boldsymbol{w}_1^T & \delta \boldsymbol{u}_2^T & \delta \boldsymbol{w}_2^T \end{array} \right]^T. \tag{69}$$

Following the procedure explained in the previous section, two orthonormal triads $\boldsymbol{V}_i$, $\boldsymbol{v}_i$ are build tangent to the axis connecting nodes 1 and 2 in the material and spatial configurations, respectively, which are rigidly rotated using $\boldsymbol{R}_e^0$, $\boldsymbol{R}_e$ (see equations (25), (26)). These mappings allow to do *rigid-rotate-back* of the variations $\delta \boldsymbol{d}$ in the $\bar{\Omega}$ configurations, via

$$\delta \boldsymbol{\mathcal{D}} = \boldsymbol{E}^T \delta \boldsymbol{d}; \quad \delta \boldsymbol{\mathcal{D}} = \left[\begin{array}{cccc} \delta \boldsymbol{U}_1^T & \delta \boldsymbol{W}_1^T & \delta \boldsymbol{U}_2^T & \delta \boldsymbol{W}_2^T \end{array} \right]; \quad \boldsymbol{E} = \frac{\partial \boldsymbol{\mathcal{D}}}{\partial \boldsymbol{d}} = \mathrm{diag}(\boldsymbol{R}_e). \tag{70}$$

[1] In what follows, the superscript (e) denoting an elemental quantity, is dropped for notational convenience, letter $a \in \{1, 2\}$ is reserved to nodal quantities and letter $i \in \{1, 2, 3\}$ is used for defining dimensions.

Additionally, the nodal centerline displacements and rotations in $\bar{\Omega}$, $\bar{\boldsymbol{u}}_a$ and $\bar{\boldsymbol{R}}_a$ are defined as

$$\bar{\boldsymbol{u}}_1 = \bar{\boldsymbol{u}}_0(0); \quad \bar{\boldsymbol{R}}_1 = \bar{\boldsymbol{R}}(0) = \boldsymbol{R}_e^T \boldsymbol{R}(0) \boldsymbol{R}_e^0 = \boldsymbol{R}_e^T \boldsymbol{R}_1 \boldsymbol{R}_e^0, \tag{71}$$

$$\bar{\boldsymbol{u}}_2 = \bar{\boldsymbol{u}}_0(l_0); \quad \bar{\boldsymbol{R}}_2 = \bar{\boldsymbol{R}}(l_0) = \boldsymbol{R}_e^T \boldsymbol{R}(l_0) \boldsymbol{R}_e^0 = \boldsymbol{R}_e^T \boldsymbol{R}_2 \boldsymbol{R}_e^0, \tag{72}$$

and via equations (32), (35), the displacement nodal values are given by

$$\bar{\boldsymbol{u}}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad \bar{\boldsymbol{u}}_2 = \begin{bmatrix} \bar{u} \\ 0 \\ 0 \end{bmatrix} \tag{73}$$

and therefore the only non-zero component is $\bar{u}$, this is, the displacement of node 2 in the $\bar{X}_1$ direction. At the same time, the rotation matrices $\bar{\boldsymbol{R}}_a$ are parametrized via the rotational vector $\bar{\boldsymbol{\Theta}}_a$, while their variation requires the spin vector $\bar{\boldsymbol{W}}_a$, i.e

$$\bar{\boldsymbol{R}}_a = \exp(\boldsymbol{S}(\bar{\boldsymbol{\Theta}}_a)); \quad \delta\bar{\boldsymbol{R}}_a = \boldsymbol{S}(\delta\bar{\boldsymbol{W}}_a)\bar{\boldsymbol{R}}_a \quad (a = 1, 2). \tag{74}$$

From the nodal quantities $\bar{u}$, $\bar{\boldsymbol{\Theta}}_1$, $\bar{\boldsymbol{\Theta}}_2$ the centerline displacements $\bar{\boldsymbol{u}}_0(\xi)$, and the cross section rotational vector $\bar{\boldsymbol{\Theta}}(\xi)$ can be interpolated at any point of the centerline. To do so, the Bernoulli hypothesis with linear interpolation for the axial displacement and axial rotation and cubic interpolation for the transverse displacements and rotations is chosen [40].

$$\bar{\boldsymbol{u}}_0 = u_1(\xi)\boldsymbol{E}_1 + \boldsymbol{u}^t(\xi), \tag{75}$$

with

$$u_1(\xi) = N_2(\xi)\bar{u}; \quad \boldsymbol{u}^t(\xi) = \begin{bmatrix} 0 \\ u_2(\xi) \\ u_3(\xi) \end{bmatrix} = \boldsymbol{P}_1(\xi) \begin{bmatrix} \bar{\boldsymbol{\Theta}}_1 \\ \bar{\boldsymbol{\Theta}}_2 \end{bmatrix}. \tag{76}$$

The rotations are interpolated as

$$\bar{\boldsymbol{\Theta}}(\xi) = \boldsymbol{P}_2(\xi) \begin{bmatrix} \bar{\boldsymbol{\Theta}}_1 \\ \bar{\boldsymbol{\Theta}}_2 \end{bmatrix}, \tag{77}$$

with the expression of $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ given in Appendix B. The above interpolations allow defining a set of local nodal quantities and their variations, which will be used in the discretized variational equation (54), i.e.

$$\bar{\boldsymbol{\mathcal{D}}} = \begin{bmatrix} \bar{u} & \bar{\boldsymbol{\Theta}}_1^T & \bar{\boldsymbol{\Theta}}_2^T \end{bmatrix}^T; \quad \delta\bar{\boldsymbol{\mathcal{D}}} = \begin{bmatrix} \delta\bar{u} & \delta\bar{\boldsymbol{\Theta}}_1^T & \delta\bar{\boldsymbol{\Theta}}_2^T \end{bmatrix}^T. \tag{78}$$

with the following relationship between $\delta\bar{\boldsymbol{\mathcal{D}}}$ and $\delta\boldsymbol{d}$

$$\delta\bar{\boldsymbol{\mathcal{D}}} = \boldsymbol{B}\delta\boldsymbol{d}, \tag{79}$$

with $\boldsymbol{B}$ given in Appendix C. On top of this, the dynamic and frictional terms requires the variation and time derivatives of the centerline displacement $\boldsymbol{u}_0$ as well as those of the centerline spin-vector $\boldsymbol{w}_0$. From equation (75), the centerline displacement in the current configuration is given by

$$\boldsymbol{u}_0^h = \boldsymbol{u}_1 N_1(\xi) + \boldsymbol{u}_2 N_2(\xi) + \boldsymbol{R}_e \boldsymbol{u}^t(\xi). \tag{80}$$

The variation of $\boldsymbol{u}_0^h$ is obtained from the equation above

$$\delta\boldsymbol{u}_0^h = \boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T \delta\boldsymbol{d}, \tag{81}$$

and the first and second time derivatives as

$$\dot{\boldsymbol{u}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{82}$$

$$\ddot{\boldsymbol{u}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T \ddot{\boldsymbol{d}} + \boldsymbol{R}_e \boldsymbol{C}_1 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{83}$$

with $\boldsymbol{H}_1$, $\boldsymbol{C}_1$ given in Appendix D. Regarding the rotational variables, equation (65) is used to write

$$\delta \bar{\boldsymbol{W}}_0^h = \delta \boldsymbol{W}_0^h - \delta \boldsymbol{W}_e, \tag{84}$$

which yields the following variations and first and second derivatives of the rotational variables

$$\delta \boldsymbol{w}_0^h = \boldsymbol{R}_e \boldsymbol{H}_2 \boldsymbol{E}^T \delta \boldsymbol{d}, \tag{85}$$

$$\dot{\boldsymbol{w}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_2 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{86}$$

$$\ddot{\boldsymbol{w}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_2 \boldsymbol{E}^T \ddot{\boldsymbol{d}} + \boldsymbol{R}_e \boldsymbol{C}_2 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{87}$$

where again the expressions of $\boldsymbol{H}_2$, $\boldsymbol{C}_2$ are given in Appendix D.

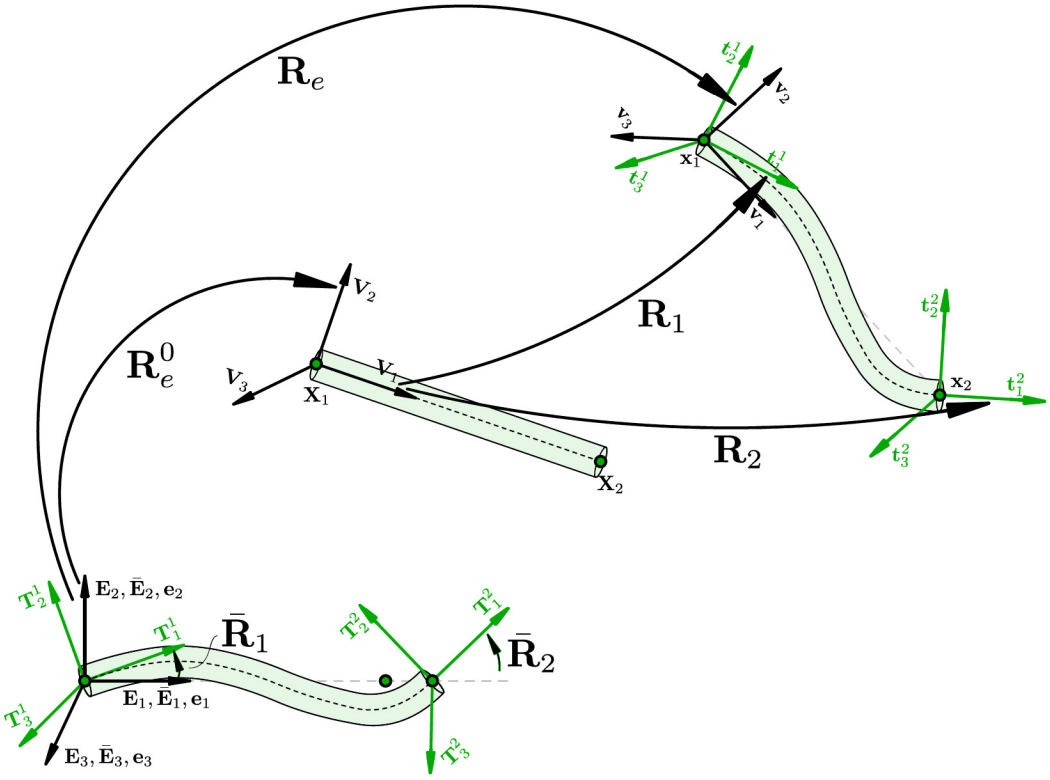


Figure 1: Co-rotational element

2.4. Variational problem for beam dynamics

By making use of the Hamilton principle, the weak form of the equations of motion is stated as

$$\delta W = \delta W_{int} + \delta W_{kin} - \delta W_{ext} = 0, \tag{88}$$

with W_{kin} being the dynamic potential and W_{ext} the potential of the external forces. The variation of the internal energy is obtained via equation (58) and following the procedure described in [38]. This variation leads to the elemental internal forces, given by

$$\delta W_{int} = (\delta \boldsymbol{d})^T \boldsymbol{T}_{int}, \tag{89}$$

with $\boldsymbol{T}_{int}$ the elemental internal forces in Ω, defined as

$$\boldsymbol{T}_{int} = \frac{\partial W_{int}}{\partial \boldsymbol{d}} = \boldsymbol{B}^T \bar{\boldsymbol{T}}_{int} \equiv \begin{bmatrix} \boldsymbol{F}_1^T & \boldsymbol{M}_1^T & \boldsymbol{F}_2^T & \boldsymbol{M}_2^T \end{bmatrix}^T, \tag{90}$$

with $\boldsymbol{B}$ the matrix expressing the relationship between $\delta\boldsymbol{d}$ and $\delta\bar{\boldsymbol{\mathcal{D}}}$ via equation (79), $\boldsymbol{F}_a$ the nodal force and $\boldsymbol{M}_a$ the nodal moment. Finally $\bar{\boldsymbol{T}}_{int}$ are the elemental internal forces in $\bar{\Omega}$ defined as

$$\bar{\boldsymbol{T}}_{int} = \frac{\partial W_{int}}{\partial \bar{\boldsymbol{\mathcal{D}}}} \equiv \begin{bmatrix} \bar{N} & \bar{\boldsymbol{M}}_1^T & \bar{\boldsymbol{M}}_2^T \end{bmatrix}^T, \tag{91}$$

with $\bar{N}$ the elemental axial force and $\bar{\boldsymbol{M}}_a$ the nodal moments in $\bar{\Omega}$. As described in [38], given the definition of the internal energy for a linear elastic material (58) and the use small strains in $\bar{\Omega}$ (51) a closed form expression can be obtained for $\bar{\boldsymbol{T}}_{int}$ for a given set of displacements $\bar{\boldsymbol{\mathcal{D}}}$, without resorting to numerical integration. Specifically, given the choice of Finite Element interpolation for the centerline displacements (75) and rotations (77), this is given by (see for example [35, 38])

$$\bar{\boldsymbol{T}}_{int} = \bar{\boldsymbol{K}}_{int} \bar{\boldsymbol{\mathcal{D}}}, \tag{92}$$

with

$$\bar{\boldsymbol{K}}_{int} = \frac{1}{l_0} \begin{bmatrix} AE & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & GJ & 0 & 0 & -GJ & 0 & 0 \\ 0 & 0 & 4EI_{33} & 0 & 0 & 2EI_{33} & 0 \\ 0 & 0 & 0 & 4EI_{22} & 0 & 0 & 2EI_{22} \\ 0 & -GJ & 0 & 0 & GJ & 0 & 0 \\ 0 & 0 & 2EI_{33} & 0 & 0 & 4EI_{33} & 0 \\ 0 & 0 & 0 & 2EI_{22} & 0 & 0 & 4EI_{22} \end{bmatrix}, \tag{93}$$

where I_{22}, I_{33} are the moments of inertia respect to the $\bar{X}_2$, $\bar{X}_3$ axes and J is the torsional constant. In order to obtain the tangent stiffness matrix, the variation of $\boldsymbol{T}_{int}$ is required. This is given by

$$\delta\boldsymbol{T}_{int} = \boldsymbol{K}_{int} \delta\boldsymbol{d}, \tag{94}$$

with

$$\boldsymbol{K}_{int} = \boldsymbol{B}^T \bar{\boldsymbol{K}}_{int} \boldsymbol{B} + \tilde{\boldsymbol{K}}_{int} \tag{95}$$

and with $\tilde{\boldsymbol{K}}$ provided in Appendix C.

The kinetic energy functional is given by (see for example [40, 64])

$$W_{kin} = \frac{1}{2} \int_{l_0} \left(A_\rho (\dot{\boldsymbol{u}}_0 \cdot \dot{\boldsymbol{u}}_0) + \dot{\boldsymbol{w}}_0 \cdot \boldsymbol{I}_\rho \dot{\boldsymbol{w}}_0 \right) ds. \tag{96}$$

The variation of the potential W_{kin} in (96) combined with equations (81), (83), (85), (86), (87) gives the inertia force vector

$$\delta W_{kin} = \boldsymbol{T}_k^T \delta\boldsymbol{d}, \tag{97}$$

with $\boldsymbol{T}_k$ given by

$$\boldsymbol{T}_k = \boldsymbol{E} \int_{l_0} \left(A_\rho \boldsymbol{H}_1^T \boldsymbol{R}_e^T \ddot{\boldsymbol{u}}_0 + \boldsymbol{H}_2^T \bar{\boldsymbol{I}}_\rho \boldsymbol{R}_e^T \ddot{\boldsymbol{w}}_0 + \boldsymbol{H}_2^T \boldsymbol{S}(\dot{\boldsymbol{W}}_0) \bar{\boldsymbol{I}}_\rho \dot{\boldsymbol{W}}_0 \right) ds, \tag{98}$$

with

$$\dot{\boldsymbol{W}}_0 = \boldsymbol{R}_e^T \dot{\boldsymbol{w}}_0, \tag{99}$$

$$\bar{\boldsymbol{I}}_\rho = \bar{\boldsymbol{R}} \boldsymbol{J}_\rho \bar{\boldsymbol{R}}^T. \tag{100}$$

and with $\ddot{\boldsymbol{u}}_0$, $\ddot{\boldsymbol{w}}_0$, $\ddot{\boldsymbol{w}}_0$ given, respectively, by equations (83), (86), (87). Finally, to obtain the tangent dynamic matrix, the variation of $\boldsymbol{T}_k$ is needed. As in [40], this is computed considering only the contributions from the acceleration and velocity, this is, the centrifugal and dynamic matrices

$$\delta \boldsymbol{T}_k \simeq \boldsymbol{M}\delta\ddot{\boldsymbol{d}} + \boldsymbol{C}_k\delta\dot{\boldsymbol{d}}, \tag{101}$$

with

$$\boldsymbol{M} = \boldsymbol{E}\left(\int_{l_0} A_\rho \boldsymbol{H}_1^T\boldsymbol{H}_1 + \boldsymbol{H}_2^T\bar{\boldsymbol{I}}_\rho\boldsymbol{H}_2\, ds\right)\boldsymbol{E}^T, \tag{102}$$

$$\boldsymbol{C}_k = \boldsymbol{E}\left(\int_{l_0}\left(A_\rho\boldsymbol{H}_1^T\left(\boldsymbol{C}_1+\boldsymbol{C}_3\right) + \boldsymbol{H}_2^T\bar{\boldsymbol{I}}_\rho\left(\boldsymbol{C}_2+\boldsymbol{C}_4\right) + \boldsymbol{H}_2^T\left(\boldsymbol{S}(\dot{\boldsymbol{W}}_0)\bar{\boldsymbol{I}}_\rho - \boldsymbol{S}(\bar{\boldsymbol{I}}_\rho\dot{\boldsymbol{W}}_0)\right)\boldsymbol{H}_2\right) ds\right)\boldsymbol{E}^T, \tag{103}$$

with the expressions for $\boldsymbol{C}_3$, $\boldsymbol{C}_4$ provided in Appendix D.

3. Beam to rigid surface contact formulation

In order to consider the contact against a rigid surface, the weak form (88) is extended using the variation of a penalty potential δW_c as

$$\delta W = \delta W_{int} + \delta W_{kin} - \delta W_{ext} - \delta W_c = 0. \tag{104}$$

Following [16], a penalty potential is defined. This potential takes into account the normal and frictional contributions of the beam contact against a rigid master surface Γ_s (see Figure 2a)

$$\delta W_c = \int_0^{l_0} \left(p_N\delta g_N + \boldsymbol{t}_T\cdot\delta\boldsymbol{g}_T\right) ds, \tag{105}$$

with g_N being the normal gap respect the master surface, $\boldsymbol{g}_T$ the relative displacement in the tangential direction, p_N the contact pressure and $\boldsymbol{t}_T$ the frictional contact traction. All values in (105) are evaluated at the beam centerline by assuming that the beam radius is small enough to disregard the couples generated by the frictional force [65]. Given the fact that circular beams are used, g_N is defined as

$$g_N(s) = d_s(s) - r, \tag{106}$$

with $d_s(s)$ being the distance of a point on the beam centerline to the surface and r the radius of the beam, which is assumed constant. The variation of g_N is given by

$$\delta g_N = \frac{\partial g_N}{\partial \boldsymbol{x}}\cdot\delta\boldsymbol{x}_0 = \frac{\partial d_{\Gamma_s}}{\partial \boldsymbol{x}}\cdot\delta\boldsymbol{u}_0 = \boldsymbol{n}\cdot\delta\boldsymbol{u}_0, \tag{107}$$

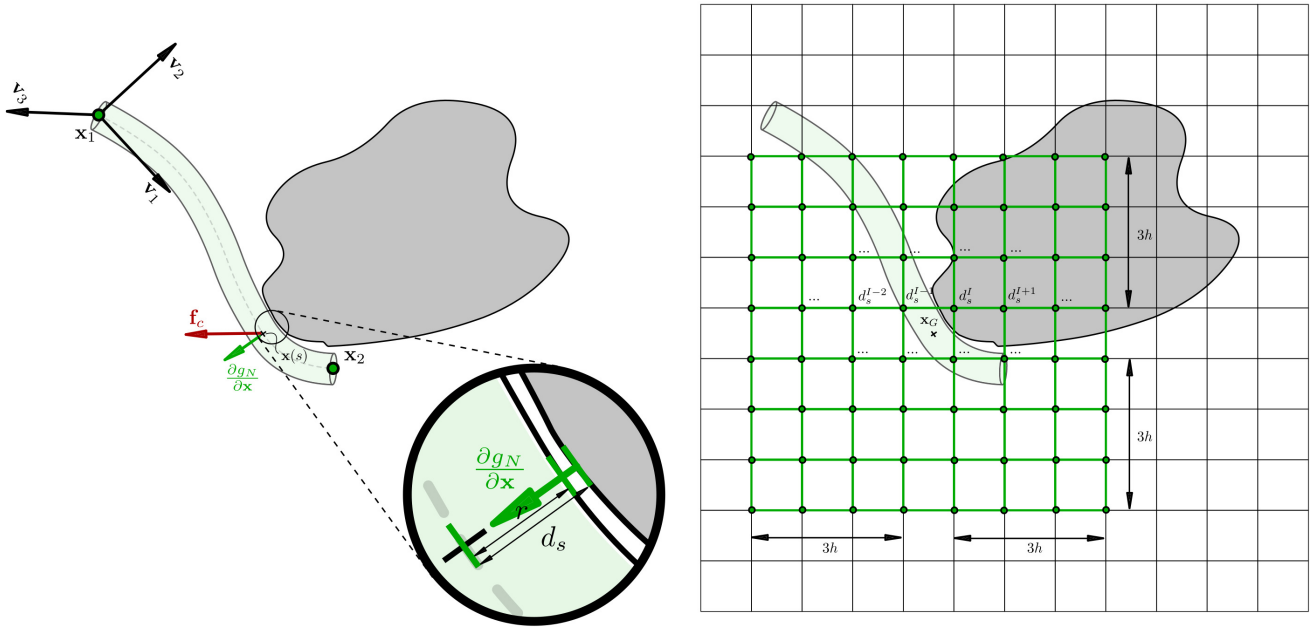
with $\boldsymbol{n}$ being the surface normal. In order to obtain the variation of $\boldsymbol{g}_T$, the centerline displacement is decomposed into its normal and tangential components, respect to the master surface [66], i.e.

$$\delta\boldsymbol{u}_0 = \left(\delta\boldsymbol{u}_0\cdot\boldsymbol{n}\right)\boldsymbol{n} + \delta\boldsymbol{u}_T = \delta g_N\frac{\partial g_N}{\partial\boldsymbol{x}} + \delta\boldsymbol{g}_T. \tag{108}$$

Isolating $\delta\boldsymbol{g}_T$ in the above expression yields

$$\delta\boldsymbol{g}_T = \left(\boldsymbol{I} - \frac{\partial g_N}{\partial\boldsymbol{x}}\otimes\frac{\partial g_N}{\partial\boldsymbol{x}}\right)\delta\boldsymbol{u}_0, \tag{109}$$

$$\dot{\boldsymbol{g}}_T = \left(\boldsymbol{I} - \frac{\partial g_N}{\partial\boldsymbol{x}}\otimes\frac{\partial g_N}{\partial\boldsymbol{x}}\right)\dot{\boldsymbol{u}}_0. \tag{110}$$



(a) Continuum.

(b) Discrete voxel mesh with a SDF.

Figure 2: Contact of a spatial beam element against a rigid surface.

In order to facilitate the convergence of the Newton Raphson algorithm, both the contact pressure p_N and frictional contact force $\boldsymbol{t}_T$ are regularized. The contact pressure is evaluated using the penalty regularization used in [45, 47–49]

$$p_N = p_N(g_N) = \begin{cases} \bar{p}_N - \varepsilon_c g_N, & g \leq 0 \\ \dfrac{\varepsilon_c \bar{g}_N - \bar{p}_N}{\bar{g}_N^2} g_N^2 - \varepsilon_c g_N + \bar{p}_N, & 0 < g_N \leq \bar{g}_N \\ 0, & g > \bar{g}_N \end{cases} \tag{111}$$

with ε_c being the penalty parameter, $\bar{p}_N = \frac{1}{2}\varepsilon_c \bar{g}_N$ and $\bar{g}_N$ being the normal gap value at which the contact pressure starts increasing (see Figure 3a). As in [47], $\bar{g}_N$ is taken as 10% of the beam radius. The frictional contact traction is evaluated using the regularization proposed in [16]

$$\boldsymbol{t}_T = \boldsymbol{t}_T(g_N, \dot{\boldsymbol{g}}_T) = -\mu p_N(g_N) \frac{\dot{\boldsymbol{g}}_T}{\sqrt{\|\dot{\boldsymbol{g}}_T\|^2 + \varepsilon_T}}. \tag{112}$$

with ε_T being a regularization parameter (see Figure 3b).

Replacing equations (81), (107), (109), (111) and (112) into the potential (105) yields the following discrete contact force vector

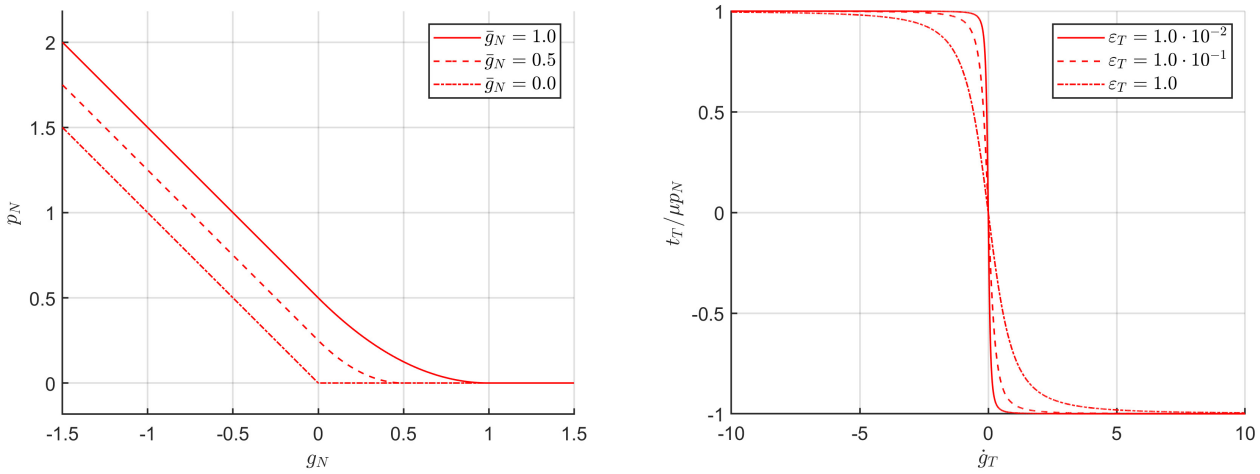
$$\boldsymbol{T}_c = \int_0^{l_0} \boldsymbol{E}\boldsymbol{H}_1^T \boldsymbol{R}_e^T \boldsymbol{f}_c \, ds, \tag{113}$$

with $\boldsymbol{H}_1$ given in Appendix D, $\boldsymbol{E}$ given in equation (70) and with $\boldsymbol{f}_c$ being the contact force, given by

$$\boldsymbol{f}_c = p_N(\boldsymbol{d}) \boldsymbol{\mathcal{G}}(\boldsymbol{d}; \dot{\boldsymbol{d}}), \tag{114}$$

with

$$\boldsymbol{\mathcal{G}}(\boldsymbol{x}; \boldsymbol{d}; \dot{\boldsymbol{d}}) = \frac{\partial g_N}{\partial \boldsymbol{x}} - \mu \frac{\dot{\boldsymbol{g}}_T^h}{\sqrt{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon}}. \tag{115}$$



(a) Regularization of p_N

(b) Regularization of $\boldsymbol{t}_T$

Figure 3: Regularization of the normal and frictional forces.

The contact tangent matrix is obtained by linearizing $\boldsymbol{T}_c$ with respect the displacement and velocity contributions (see Appendix E for details),

$$\delta \boldsymbol{T}_c = \boldsymbol{K}_c \delta \boldsymbol{d} + \boldsymbol{C}_c \delta \dot{\boldsymbol{d}}, \tag{116}$$

with $\boldsymbol{K}_c$ defined as

$$\boldsymbol{K}_c = \boldsymbol{K}_c^1 + \boldsymbol{K}_c^2 + \boldsymbol{K}_c^3 + \boldsymbol{K}_c^4, \tag{117}$$

and with each of the terms given by

$$\boldsymbol{K}_c^1 = -\int_0^{l_0} \boldsymbol{E}\hat{\boldsymbol{S}}(\boldsymbol{H}_1^T \boldsymbol{\mathcal{F}}_c)\boldsymbol{G}^T \boldsymbol{E}^T \, ds, \tag{118a}$$

$$\boldsymbol{K}_c^2 = \int_0^{l_0} \left(\frac{N_7}{l_n^2} \boldsymbol{E}\boldsymbol{A}_1^T \boldsymbol{\mathcal{F}}_c \boldsymbol{r} - \boldsymbol{E}\boldsymbol{G}\boldsymbol{S}(\boldsymbol{\mathcal{F}}_c)\boldsymbol{P}_1 \boldsymbol{P}\boldsymbol{E}^T \right) ds, \tag{118b}$$

$$\boldsymbol{K}_c^3 = \int_0^{l_0} \boldsymbol{E}\boldsymbol{H}_1^T \boldsymbol{S}(\boldsymbol{\mathcal{F}}_c)\boldsymbol{G}^T \boldsymbol{E}^T \, ds, \tag{118c}$$

$$\boldsymbol{K}_c^4 = \int_0^{l_0} \boldsymbol{E}\boldsymbol{H}_1^T \boldsymbol{R}_e^T \boldsymbol{K}_{\boldsymbol{f}_c} \, ds, \tag{118d}$$

where the full expression of $\boldsymbol{K}_{\boldsymbol{f}_c}$ is given in Appendix E, $\boldsymbol{\mathcal{F}}_c$ is the contact force *rigidly-rotated-back* to the $\bar{\Omega}$ configuration, i.e.

$$\boldsymbol{\mathcal{F}}_c = \boldsymbol{R}_e^T \boldsymbol{f}_c \tag{119}$$

and $\hat{\boldsymbol{S}}$ is the operator that transforms the 12×1 array into a 12×3 matrix as

$$\hat{\boldsymbol{S}}(\boldsymbol{a}) = \begin{bmatrix} \boldsymbol{S}(\boldsymbol{a}_1) \\ \boldsymbol{S}(\boldsymbol{a}_2) \\ \boldsymbol{S}(\boldsymbol{a}_3) \\ \boldsymbol{S}(\boldsymbol{a}_4) \end{bmatrix} \quad \boldsymbol{a}_I = \begin{bmatrix} a_{3(I-1)+1} \\ a_{3(I-1)+2} \\ a_{3(I-1)+3} \end{bmatrix}; \quad I \in \{1,2,3,4\}. \tag{120}$$

Finally, the dynamic contribution is given by

$$\boldsymbol{C}_c = \int_0^{l_0} \boldsymbol{E}\boldsymbol{H}_1^T \boldsymbol{R}_e^T \boldsymbol{C}_{\boldsymbol{f}_c} \, ds, \tag{121}$$

with $\boldsymbol{C}_{\boldsymbol{f}_c}$ given in Appendix E.

4. Interpolation of a discrete Signed Distance Field

The contact formulation presented in the previous section depends on the normal gap $g_N(\boldsymbol{x})$, and its spatial gradient and Hessian, i.e. $\frac{\partial g_N(\boldsymbol{x})}{\partial \boldsymbol{x}}$, $\frac{\partial^2 g_N(\boldsymbol{x})}{\partial \boldsymbol{x}\partial \boldsymbol{x}}$. This section presents the necessary tools to use a discrete SDF to compute the above quantities. To do so, an efficient interpolation based on the tensor product of 1D kernels is presented in section 4.1. Section 4.2 assesses the accuracy of the proposed framework to interpolate the signed distance.

4.1. Kernel Functions

At any Gauss point of the beam, represented by the isoparametric coordinate ξ_G, which is located in the spatial domain at $\boldsymbol{x}_G \equiv \boldsymbol{x}_0(\xi_G)$, the gap function, first and second derivatives are needed, i.e. from equation (106),

$$g_N(\boldsymbol{x}_G) = d_s(\boldsymbol{x}_G) - r, \tag{122}$$

$$\frac{\partial g_N(\boldsymbol{x}_G)}{\partial \boldsymbol{x}} = \frac{\partial d_s(\boldsymbol{x}_G)}{\partial \boldsymbol{x}}, \tag{123}$$

$$\frac{\partial^2 g_N(\boldsymbol{x}_G)}{\partial \boldsymbol{x}\partial \boldsymbol{x}} = \frac{\partial^2 d_s(\boldsymbol{x}_G)}{\partial \boldsymbol{x}\partial \boldsymbol{x}}. \tag{124}$$

It is assumed that the SDF is provided in a Cartesian grid in a three dimensional prismatic domain $\Omega_s = [x_1^o, x_1^f] \times [x_2^o, x_2^f] \times [x_3^o, x_3^f]$, discretized with N nodes, $N = N_1 N_2 N_3$, defined as

$$\boldsymbol{x}^A = \left(x_1^{A_1}, x_2^{A_2}, x_3^{A_3}\right)^T; \quad A \equiv (A_1, A_2, A_3), \tag{125}$$

with

$$x_i^{A_i} = x_i^o + (A_i - 1)h_i; \quad A_i \in \{1, N_i\}; \quad h_i = (x_i^f - x_i^o)/(N_i - 1); \quad i \in 1, 2, 3 \tag{126a}$$

and h_i being the grid size in the direction i. For simplicity, from now a uniform size will be considered in all directions i.e. $h = h_1 = h_2 = h_3$. At each of the voxel grid nodes, the signed distance is given, i.e.

$$d_s(\boldsymbol{x}^A) = d_s^A \tag{127}$$

and at any other point within the domain Ω_s, the value of d_s can be interpolated. Specifically, at the integration point of the beam, i.e.

$$d_s(\boldsymbol{x}_G) = \mathcal{I}_A(d_s^A), \tag{128}$$

where $\mathcal{I}$ is some interpolant function. Given the Cartesian grid chosen for the discrete SDF, one dimensional kernel functions are chosen, as used in Immersed Boundary Fluid Structure Interaction methods [58, 59, 67]. This is, for any point in $\boldsymbol{x}$ in space, the signed distance $d_s(\boldsymbol{x})$ is interpolated as

$$d_s(\boldsymbol{x}) = \sum_A V_A d_s^A \varphi(\boldsymbol{x} - \boldsymbol{x}^A), \tag{129}$$

where $V_A = h^3$ and ϕ defined as a tensor product of 1D kernels, i.e.

$$\varphi(\boldsymbol{z}) = \phi\left(\frac{z_1}{h}\right)\phi\left(\frac{z_2}{h}\right)\phi\left(\frac{z_3}{h}\right). \tag{130}$$

This allows computing the first and second derivatives of d_s as

$$\frac{\partial d_s(\boldsymbol{x})}{\partial \boldsymbol{x}} = \sum_A V_A d_s^A \frac{\partial \varphi}{\partial \boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{x}^A), \tag{131}$$

$$\frac{\partial^2 d_s(\boldsymbol{x})}{\partial \boldsymbol{x}\partial \boldsymbol{x}} = \sum_A V_A d_s^A \frac{\partial^2 \varphi}{\partial \boldsymbol{x}\partial \boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{x}^A), \tag{132}$$

with

$$\frac{\partial \varphi}{\partial z_i} = \frac{1}{h}\phi'\left(\frac{z_i}{h}\right)\phi\left(\frac{z_j}{h}\right)\phi\left(\frac{z_k}{h}\right), \tag{133}$$

$$\frac{\partial^2 \varphi}{\partial z_i \partial z_i} = \frac{1}{h^2}\phi''\left(\frac{z_i}{h}\right)\phi\left(\frac{z_j}{h}\right)\phi\left(\frac{z_k}{h}\right), \tag{134}$$

$$\frac{\partial^2 \varphi}{\partial z_i \partial z_j} = \frac{1}{h^2}\phi'\left(\frac{z_i}{h}\right)\phi'\left(\frac{z_j}{h}\right)\phi\left(\frac{z_k}{h}\right), \tag{135}$$

where $i \neq j \neq k$. The use of the above interpolation strategy is very advantageous when using Cartesian meshes (as the given voxel mesh), as the stencils involved in the interpolation (129) can be precomputed in advance, and therefore no search is involved in computing the distance d_s. There is a wealth of 1D kernels functions ϕ proposed in the immersed FSI literature that could be used in the above expressions. Yet, some of the proposed kernels do not satisfy continuity of the second derivative [67], which would be detrimental for the proposed as second order derivatives of ϕ are required (see equation 132). Instead, the spline-based kernel proposed in [59] is used, which has a support of $[-3h, 3h]$ and ensures continuity of the second derivative. This is given by

$$\phi(r) = \begin{cases} 0 & r \leq -3 \\ -\frac{29}{7560}r^7 - \frac{5}{72}r^6 - \frac{21}{40}r^5 - \frac{17}{8}r^4 - \frac{39}{8}r^3 - \frac{243}{40}r^2 - \frac{27}{8}r - \frac{81}{280} & -3 < r \leq -2 \\ \frac{17}{1512}r^7 + \frac{1}{8}r^6 + \frac{13}{24}r^5 + \frac{79}{72}r^4 + \frac{65}{72}r^3 + \frac{7}{120}r^2 + \frac{13}{72}r + \frac{1447}{2520} & -2 < r \leq -1 \\ -\frac{11}{756}r^7 - \frac{1}{18}r^6 + \frac{7}{36}r^4 - \frac{29}{60}r^2 + \frac{691}{1260} & -1 < r \leq 0 \\ \frac{11}{756}r^7 - \frac{1}{18}r^6 + \frac{7}{36}r^4 - \frac{29}{60}r^2 + \frac{691}{1260} & 0 < r \leq 1 \\ -\frac{17}{1512}r^7 + \frac{1}{8}r^6 - \frac{13}{24}r^5 + \frac{79}{72}r^4 - \frac{65}{72}r^3 + \frac{7}{120}r^2 - \frac{13}{72}r + \frac{1447}{2520} & 1 < r \leq 2 \\ \frac{29}{7560}r^7 - \frac{5}{72}r^6 + \frac{21}{40}r^5 - \frac{17}{8}r^4 + \frac{39}{8}r^3 - \frac{243}{40}r^2 + \frac{27}{8}r - \frac{81}{280} & 2 < r \leq 3 \\ 0 & 3 < r \end{cases} \tag{136}$$

which is designed, by construction, to interpolate exactly linear functions and with second-order accuracy smooth functions [59, 67].

4.2. Numerical analysis of the signed distance interpolation error

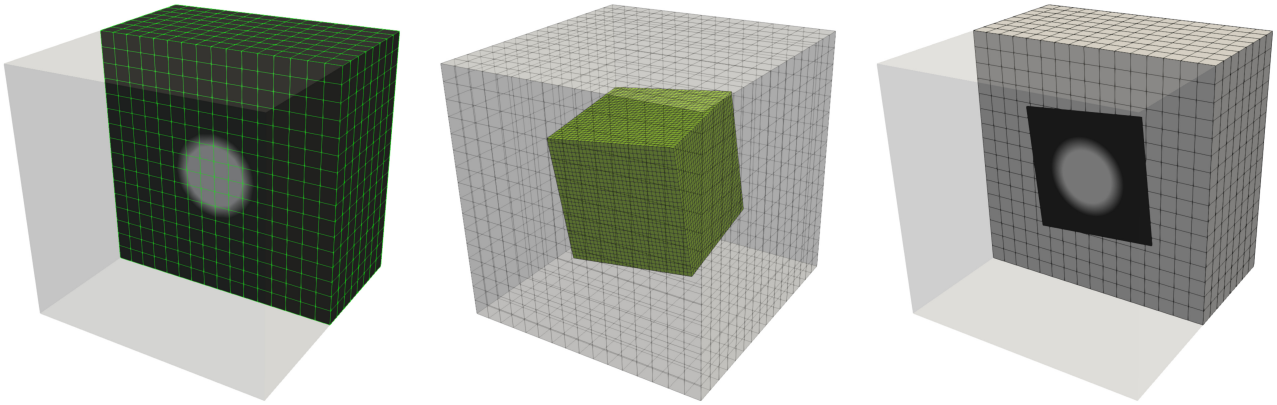
In this section the accuracy of the interpolation strategy presented in the previous section is analyzed numerically. To do so, the analytical signed distance function of a sphere is used, which is defined as

$$d_s(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{P}_0)\cdot(\boldsymbol{x} - \boldsymbol{P}_0)} - R_0, \tag{137}$$

with $\boldsymbol{P}_0$ being the location of the center of the sphere and R_0 its radius. For the purpose of the current analysis, these are chosen as $\boldsymbol{P}_0 = \begin{bmatrix} 0.5 & 0.5 & 0.5 \end{bmatrix}^T$ and $R_0 = 0.3$.

Five sets of voxel meshes are created in the domain $\Omega_s = [-0.5, 1.5] \times [-0.5, 1.5] \times [-0.5, 1.5]$ with mesh sizes $h = \{1/4, 1/8, 1/16, 1/32, 1/64\}$. At each of the nodes A of a given voxel mesh, the signed distance d_s^A is stored as $d_s^A = d_s(\boldsymbol{x}^A)$ via equation (137), which creates a SDF (see Figure 4a). At the same time, an auxiliary mesh is created with the final purpose of interpolating the SDF values created in the previous step, mimicking what will be done at the beam Gauss point $\boldsymbol{x}_G$. Additionally, the spatial gradient and Hessian are also computed at each of the nodes of this auxiliary mesh. The auxiliary mesh is centered at $\boldsymbol{P}_0$, has a side length of $l = 1$, element size $h_{aux} = 1/32$ and, in order to avoid overlapping of nodes with the background voxel mesh (which could potentially benefit the global interpolation error), it is rotated 5 degrees respect to the X_1, X_2 and X_3 axes (see Figure 4b). Additionally, at each of the points of the auxiliary mesh, the exact value of the signed distance, its gradient and Hessian can be obtained from equation (137) which, in turn, allows computing the approximation error. As it can easily be proved, the gradient and Hessian of the signed distance function (137) have a singularity at $\boldsymbol{P}_0$. Therefore, in order to avoid numerical issues, the error analysis is limited to the region in which $|d_s(\boldsymbol{x})| \leq 0.8R_0$.

Figure 5 shows the L2-norms for the interpolated signed distance, d_s^h and the computed spatial gradient and Hessian ∇d_s^h, $\nabla^2 d_s^h$. It can be seen that all values converge to their analytical counterpart with second-order accuracy, i.e. $\mathcal{O}(h^2)$, which proves the adequacy of the algorithm, initially designed for velocity fields, to interpolate distances and their derivatives. Particularly, it is worth stressing the capacity of this interpolation strategy to compute gradients and Hessians with the same order of accuracy without having the discrete values of these operators initially. Yet, contact against such a smooth surface cannot always be warranted as sharp gradients are present in real-life geometries. In such cases, as proved for the immersed boundary method, first-order accuracy, i.e. $\mathcal{O}(h)$ is expected [67, 68]. Importantly, the discrete contact force $\boldsymbol{f}_c$ in (114) and its spatial derivatives will also approximated at the same order of the distance function $d_s(\boldsymbol{x})$ and its spatial derivatives (i.e. second-order for smooth surfaces, first-order for non-smooth surfaces) which, alongside the beam discretization will determine the contact algorithm order of accuracy.



(a) SDF in voxel mesh (b) Auxiliary mesh (c) SDF interpolated in auxiliary mesh

Figure 4: Interpolation of a SDF in a voxel mesh of $h = 1/8$

5. Time integration and computational aspects

Following the discretization in equations (90), (98) and (113), and upon assembly, the discretization of the weak form (104) is given by

$$\mathbf{R}(\mathbf{D}, \dot{\mathbf{D}}, \ddot{\mathbf{D}}, t) = \mathbf{T}_{int}(\mathbf{D}, t) + \mathbf{T}_k(\mathbf{D}, \dot{\mathbf{D}}, \ddot{\mathbf{D}}, t) - \mathbf{T}_c(\mathbf{D}, \dot{\mathbf{D}}, t) - \mathbf{F}_{ext}(\mathbf{D}, t) = \mathbf{0} \tag{138}$$

where $\mathbf{R}$ is the residual, $\mathbf{T}_{int}$, $\mathbf{T}_k$, $\mathbf{T}_c$ are, respectively, the assembled vectors of nodal internal, kinetic and contact forces and $\mathbf{F}_{ext}$ is the vector of nodal external forces. At the same time, $\mathbf{D}$ is the vector of nodal solutions, containing displacements and rotations, and $\dot{\mathbf{D}}, \ddot{\mathbf{D}}$ are respectively, its first and second time derivatives. Following equation (69), these are given by

$$\mathbf{D} = \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{u}_N \\ \boldsymbol{\theta}_N \end{bmatrix}; \quad \Delta\mathbf{D} = \begin{bmatrix} \Delta\boldsymbol{u}_1 \\ \Delta\boldsymbol{w}_1 \\ \vdots \\ \Delta\boldsymbol{u}_N \\ \Delta\boldsymbol{w}_N \end{bmatrix}; \quad \dot{\mathbf{D}} = \begin{bmatrix} \dot{\boldsymbol{u}}_1 \\ \dot{\boldsymbol{w}}_1 \\ \vdots \\ \dot{\boldsymbol{u}}_N \\ \dot{\boldsymbol{w}}_N \end{bmatrix}; \quad \ddot{\mathbf{D}} = \begin{bmatrix} \ddot{\boldsymbol{u}}_1 \\ \ddot{\boldsymbol{w}}_1 \\ \vdots \\ \ddot{\boldsymbol{u}}_N \\ \ddot{\boldsymbol{w}}_N \end{bmatrix}, \tag{139}$$

with N the number of nodes of the beam mesh. It is worth noticing that, in equation (138), $\mathbf{T}_c$ depends both on $\mathbf{D}$ and $\dot{\mathbf{D}}$, due to the frictional contact term (see equations (114), (115)). Equation (138) is discretized in time using the HHT-α method [69], as used in corotational beam dynamics in [40, 50]. This yields

$$\mathbf{R}^{n+\alpha} = (1+\alpha)(\mathbf{T}_{int}^{n+1} - \mathbf{T}_c^{n+1} - \mathbf{F}_{ext}^{n+1}) - \alpha(\mathbf{T}_{int}^{n} - \mathbf{T}_c^{n} - \mathbf{F}_{ext}^{n}) + \mathbf{T}_k^{n+1} = \mathbf{0} \tag{140}$$

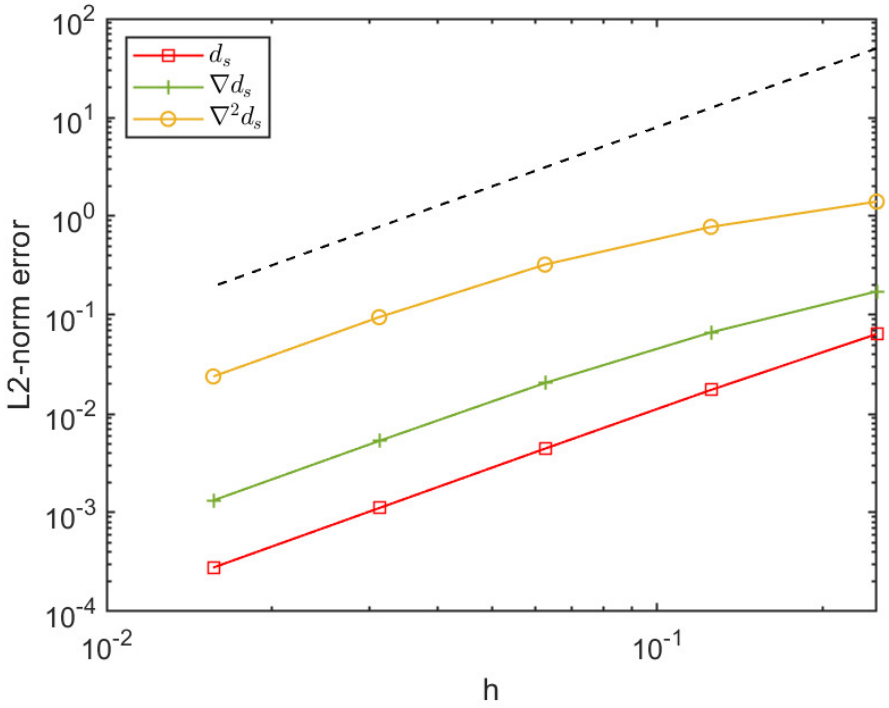


Figure 5: Error in the discrete approximation of the signed distance of a sphere surface and its first and second derivatives. Discontinuous line indicates second-order convergence slope.

with α a user-dependent parameter and where the superscript n (respectively $n+1$) indicates that the operator is evaluated at the time step $t = n\Delta t$ (respectively $t = (n+1)\Delta t$). Finally, the system is solved via a Newton Raphson algorithm using the following linearization

$$\tilde{\mathbf{K}}\Delta\mathbf{D} = -\mathbf{R}^{n+\alpha} \tag{141}$$

with $\tilde{\mathbf{K}}$ given by

$$\tilde{\mathbf{K}} = (1+\alpha)\left(\mathbf{K}_{int} - \mathbf{K}_c - \mathbf{K}_{ext}\right) + \frac{1}{\beta\Delta t^2}\mathbf{M} + \frac{\gamma}{\beta\Delta t}\left(\mathbf{C}_k - (1+\alpha)\mathbf{C}_c\right), \tag{142}$$

with $\beta = \frac{1}{4}(1-\alpha)^2$, $\gamma = \frac{1}{2}(1-2\alpha)$ and $\mathbf{K}_{int}$, $\mathbf{K}_c$, $\mathbf{M}$, $\mathbf{C}_k$, $\mathbf{C}_c$ assembled, respectively, from equations (95), (117), (102), (103) and (121). Upon obtaining $\Delta\mathbf{D}$, the update of the displacement and rotational variables is carried out following the procedure described in [40].

At the beginning of each time step, prior to the Newton Raphson algorithm, the predictor proposed in [70] is used. Following [47], the Newton-Raphson algorithm is solved by checking the norm of the increment vector in (139) and the norm of the residual in (140). The iteration finishes once both norms are smaller than their respective prescribed tolerances, ϵ_R, ϵ_D, i.e. $\|\mathbf{R}\| \leq \epsilon_R$, $\|\Delta\mathbf{D}\| \leq \epsilon_D$. Also, following [47], if the Newton-Raphson algorithm fails to converge after 10 iterations, the time step is divided by two. Afterwards, and once the algorithm successfully converges in 8 solves, the time step is multiplied by two (this increase stops once the initial time step is recovered).

Regarding the interpolation of the SDF, as described in section 4, this can be efficiently implemented due to the Cartesian structure of the voxel mesh. Specifically, given a Lagrangian point of integration, it is very simple to find in which voxel element it has fallen (if any). Once this is found, the interpolation (129) is carried out using a precomputed stencil. By doing this, the computation of the distance at the Lagrangian point of integration becomes a quick sum over the stencil nodal values, therefore avoiding any search. In case the Lagrangian point of integration

falls outside the background voxel mesh, a numerically large value is assigned to d_s, which makes $p_N = 0$ via equation (111).

6. Verification and examples

Each of the SDF involved in these examples has been generated using SDFgen [71]. All examples have been simulated using 3 Gauss points for integrating the dynamic contributions and 5 Gauss points for integrating the contact contributions. The tolerances in the Newton-Raphson iterations are set to to $\epsilon_R = \epsilon_D = 1 \cdot 10^{-5}$. All the examples show quadratic convergence in the Newton-Raphson iterations.

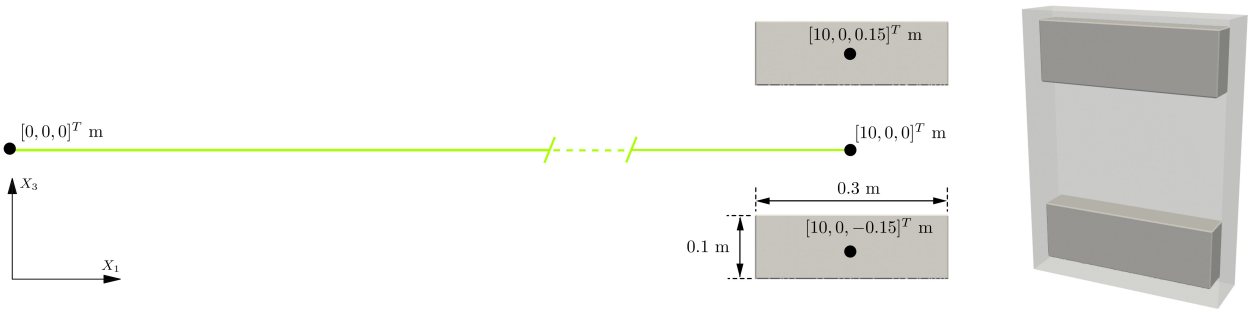
6.1. Vibrations of a beam constrained by two end stops

This example is used as a benchmark test and was first published in [9, 17, 18]. A beam constrained to move in the $X_1 - X_3$ plane has a length $L = 10$ m, moment of inertia $I_{22} = 1.688 \cdot 10^{-3}$ m^4, cross sectional area $A = 1.4923 \cdot 10^{-5}$ m^2, Young's modulus $E = 2 \cdot 10^{11}$ Pa, density $\rho_0 = 8000$ Kg/m^3 and Poisson's ratio $\nu = 0$. The beam has its left end clamped and two rigid stops located at its right end that constrain the displacement above and below $X_3 = \pm 0.1$ m. The original results published in [9, 17, 18] were obtained by modeling the one dimensional of a string and, therefore, the end-stops where modeled as a singular constrain located exactly at $X_3 = 10$ m. In this case, a two dimensional beam is modeled and the tip can move freely in the X_1 and X_3 directions. Because of this, the end-stops are created as two rigid prisms of $0.3 \times 0.04 \times 0.1$ m centered at $\boldsymbol{X} = \begin{bmatrix} 10 & 0 & \pm 0.15 \end{bmatrix}^T$ m (see Figure 6a), while non-frictional contact is imposed only at the right end node. In order to mimic the literature results, no radius is considered when computing the normal gap g_N, i.e. $r = 0$ in equation (106). A uniform time-varying body force is applied along the beam, with the value $\boldsymbol{b} = \begin{bmatrix} 0 & 0 & \sin(10t) \end{bmatrix}^T$, which causes the beam right tip to impact against the end-stops and excite high frequency modes. The beam is modeled using three choices for the number of elements, $n_{el} = \{20, 80, 320\}$ with corresponding time step of $\Delta t = \{5 \cdot 10^{-4}, 1.25 \cdot 10^{-4}, 3.125 \cdot 10^{-5}\}$ s. The penalty parameter is set to $\varepsilon_c = 1 \cdot 10^6$. The end-stops are modeled using a SDF in a voxel-type grid $175 \times 45 \times 219$ nodes of element size $h = 0.002$ m. Figure 6b shows the isovolume of this SDF.

Figure 7 shows the variation of the u_3 component of the displacement of the beam tip with time for different refinements of the beam mesh. Both the displacement and time are made adimensional using the same procedure as in [18]. It can be seen that high frequency modes are excited with successive impacts and chattering as the beam tip impacts the top and bottom stops. This is highlighted in Figure 8, which zooms in the right end displacement at the first impact. The solution compares very well against that of [18], with some differences when the beam tip is freely moving between the two stops. This difference is probably due to the two-dimensional character of the solution, as the beam tip can freely move in the X_1 and X_3 directions, which is a substantial difference with respect to the string vibration solution obtained in [18]. Figure 9 shows the pathlines of the beam tip in the $X_1 - X_3$ plane, which shows this two dimensional motion. This plot also shows how the solution converges with successive refinement of the beam mesh.

6.2. Beam against sphere

This example compares the accuracy of the discrete SDF versus the use of an analytical signed distance function. A beam of length $L = 10$ cm and radius $R = 0.3$ mm has its base point constrained to oscillate back and forth in the X_1 direction (the rest of the beam points are free to move and rotate in any direction). The beam has a Young's modulus $E = 5$ GPa, Poisson ratio $\nu = 0.33$ and density $\rho_0 = 7850$. The beam impacts against a sphere centred at $\boldsymbol{P}_0 = \begin{bmatrix} 5 & 0.5 & 8 \end{bmatrix}^T$ cm and radius $R_0 = 4$ cm. Figure 10a describes the problem setup while Figure 10b shows the prescribed displacement of the beam base point. The sphere surface is defined



(a) Initial Setup

(b) Isovolume of the SDF

Figure 6: Vibrations of a beam constrained by two end stops. Initial setup and SDF

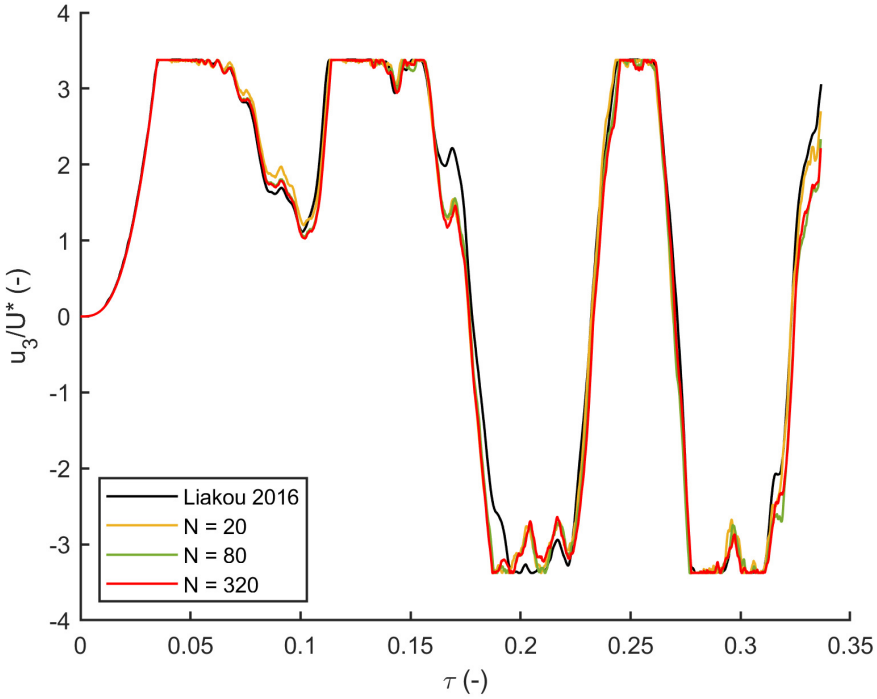
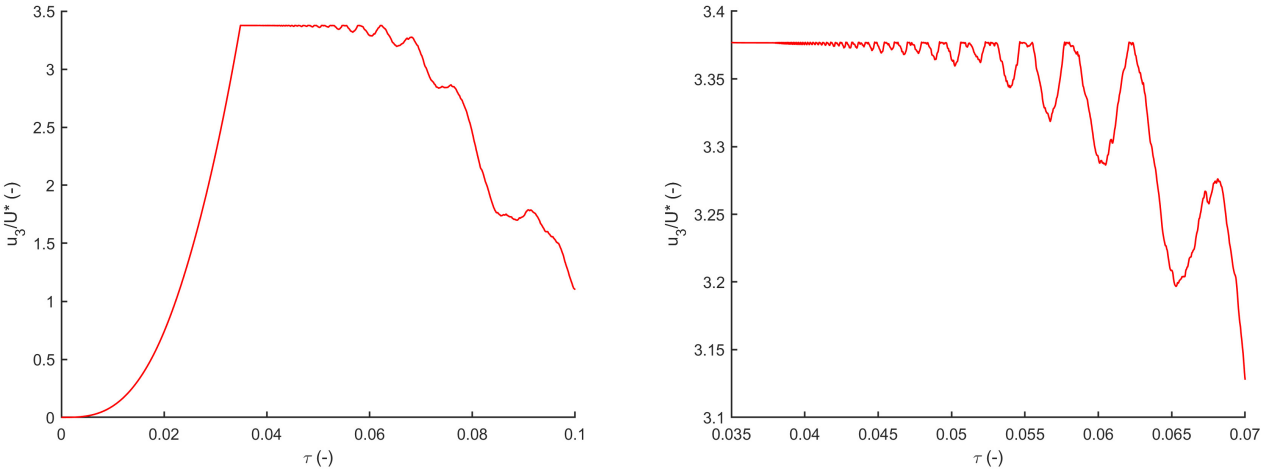


Figure 7: Vibrations of a beam constrained by two end stops for different discretizations. Normalized u_3 displacement of the beam tip.

first using an analytical signed distance function, see equation (137), and then by using 3 discrete SDF defined in a domain $\Omega_s = [-0.05, 0.15] \times [-0.1, 0.1] \times [-0.05, 0.15]$ of $(20 \times 20 \times 20)$ voxels, $(40 \times 40 \times 40)$ voxels and $(80 \times 80 \times 80)$ voxels (see Figure 11). The penalty parameter is set to $\varepsilon_c = 5 \cdot 10^4$ and no friction is considered, i.e. $\mu = 0.0$.

Figure 12 shows snapshots of the solution at $t = 1.25$ s, comparing different voxel discretizations with the analytical SDF. Additionally, a video animation is provided in Appendix F in the electronic version of the document. Visually, it is observed that for coarser voxel meshes, the beam undergoes some penetration within the sphere surface. In order to quantify that penetration, the analytical SDF is used to compute the maximum penetration at the centerline. To do so, the centerline position is postprocessed via the interpolation defined in equation (80) and using 80 points per element between $\xi = 0$ and $\xi = l_0$. At each of this postprocessed points, the penetration is evaluated via equation (106) and the analytical signed distance function (137). Using this procedure, Figure 13 shows the maximum normalized penetration. It can be seen how the maximum penetration is reduced with the refinement of the voxel mesh. Also, the finest voxel



(a) Zoom 1

(b) Zoom 2

Figure 8: Vibrations of a beam constrained by two end stops $n_{el} = 320$. Normalized u_3 displacement of the beam tip.

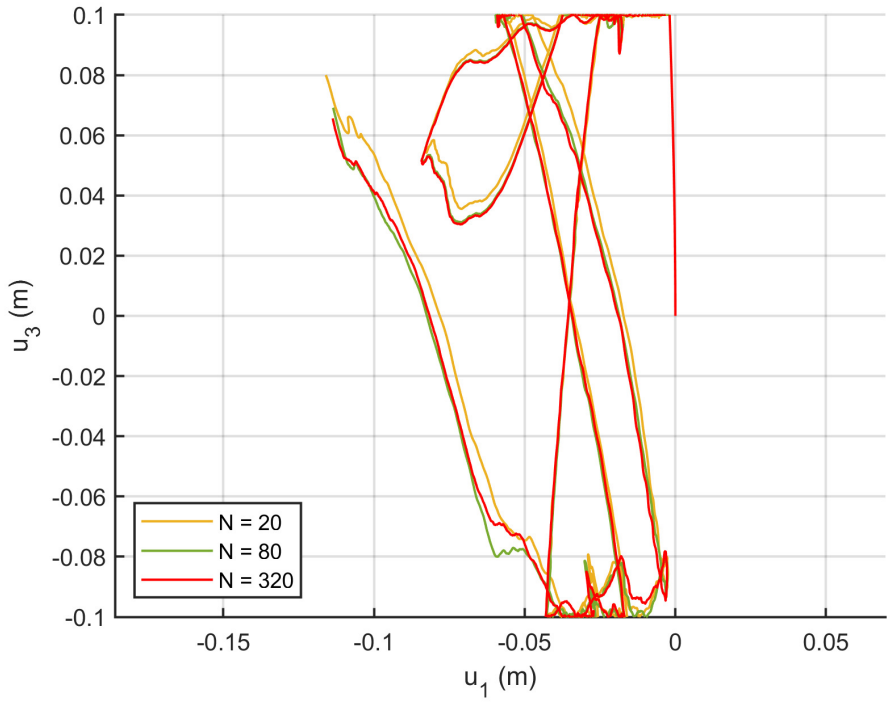
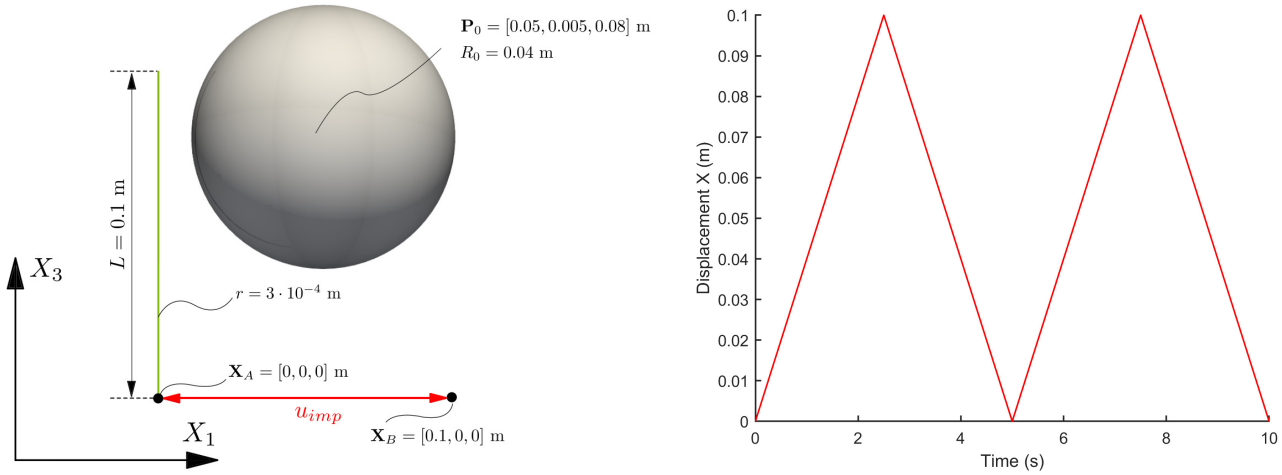


Figure 9: Vibrations of a beam constrained by two end stops, different discretizations. u_1 versus u_3 displacement of the beam tip.

mesh gives maximum penetration levels of the order of the analytical solution, which is around 50% of the beam radius.

Finally, the computational efficiency of the proposed distance interpolation algorithm is assessed qualitatively. Figure 14 shows the ratio of computational time (wall clock time) to physical time when using different discretizations of the SDF or the analytical signed distance. It can be seen that when using the discrete SDF the computational time is increased around 50% as compared to the computation when using the analytical signed distance equation. The differences in computational time between the different discretizations are significantly smaller than that, which proves that the interpolation algorithm is barely affected by the size of the background mesh h. This is a significant difference with respect to traditional contact search methods, where the size of the master surface mesh plays a big role in the computational time. Again, this assessment is purely qualitative and a more accurate assessments should be carried out by comparing against standard contact methods. Figure 14 also shows that the ratios of computational to physical time are of the order of 300, approximately. This is due to the fact that the algorithm has been implemented in Matlab using dense matrices and no vectorization and therefore, is open to substantial optimizations. This would allow reducing very significantly the ratio shown in the figure and approach real-time computation (ratio = 1) for this example. Finally, it is worth mentioning that the ratio fluctuations as shown in Figure 14 are due to the adaptive time step strategy described in section 5, which forces a reduction of the time step when contact occurs.



(a) Initial Setup

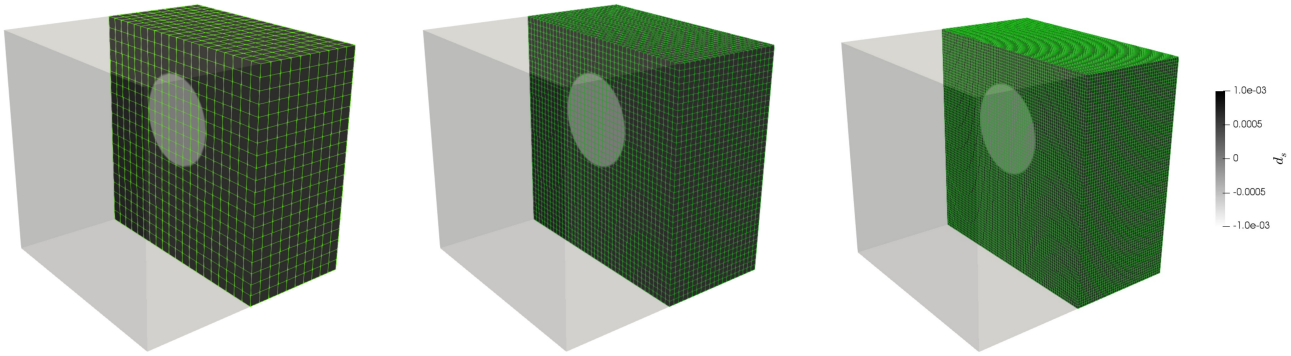
(b) Imposed displacement

Figure 10: Beam against sphere. Initial setup and imposed displacement.

6.3. Contact between a cable and a rigid cylinder

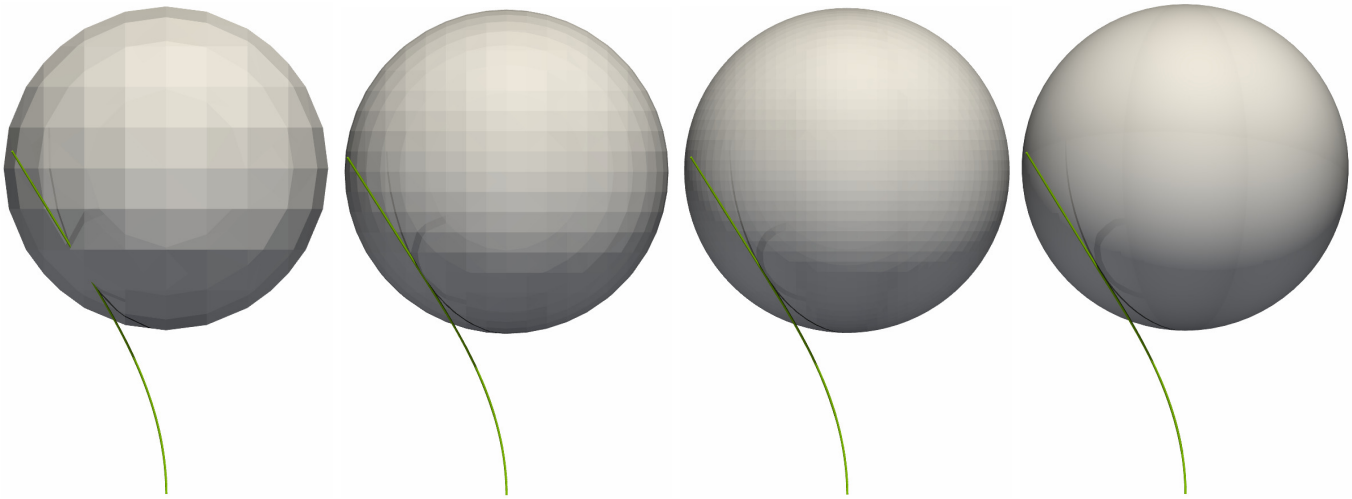
This example is taken from [20, 72] and shows the robustness of the contact algorithm in a friction dominated scenario. A circular cable of length 3.0 m is inclined 0.925 rad respect to the X_3 axis with a lumped mass of 5.0 kg placed at its end. The cable has a radius $r = 0.01$ m, density $\rho_0 = 7919$ kg/m^3 and Young's modulus $E = 1 \cdot 10^8$ Pa. The rigid cylinder has a length of 3.0 m and radius of 0.1 m. The cable has an initial angular speed $\boldsymbol{w} = [0, 0, 2]$ rad/s and is subjected to gravitational force, which is imposed through a constant body force $\boldsymbol{b} = \rho_0 \, [0, 0, -9.8]^T$ m/s^2. The position of the top end of the cable is set to $\boldsymbol{X}_o = [-0.8, -0.115, 3.0]$ m[2], which accounts for an initial gap of $5 \cdot 10^{-3}$ m (see Figure 15). The surface SDF is given in a voxel-type grid of $25 \times 25 \times 250$ voxels, element size $h = 0.0125$ m and it is centered at the cylinder longitudinal

[2]The first component of $\boldsymbol{X}_o$ has been decided by the authors, as it was not detailed in [20, 72]



(a) $20 \times 20 \times 20$ voxels (b) $40 \times 40 \times 40$ voxels (c) $80 \times 80 \times 80$ voxels

Figure 11: Beam against sphere. Different SDF voxel meshes used for the simulation.



(a) $20 \times 20 \times 20$ (b) $40 \times 40 \times 40$ (c) $80 \times 80 \times 80$ (d) Analytical

Figure 12: Beam against sphere. View on the $X_1 - X_3$ plane of beam contacting the sphere for different levels of discretization of the SDF as compared to the analytical solution. Time step $t = 1.25$ s.

axis as shown in Figure 16. The cable is first modeled using 20 beam elements and time step size $\Delta t = 5 \cdot 10^{-4}$. The penalty parameter is set to $\varepsilon_c = 1 \cdot 10^6$ and three different friction coefficients are used, $\mu = \{0.00, 0.10, 0.30\}$. Figure 17 shows three snapshots of the solution for $\mu = 0.1$ at different time instants. Figure 18 quantifies the vertical displacement of the beam tip for the different friction coefficients. Additionally, a video animation is provided in Appendix F in the electronic version of the document, where the evolution of the solution is compared between $\mu = 0.00$ and $\mu = 0.30$. As expected, the higher the friction coefficient, the smaller the vertical oscillations, as the cable sticks more onto the cylinder surface. Next, the cable is modeled using three choices for the number of beam elements, $n_{el} = \{10, 20, 40\}$ with corresponding time steps of $\Delta t = \{1 \cdot 10^{-3}, 5 \cdot 10^{-4}, 2.5 \cdot 10^{-4}\}$ s and $\mu = 0.30$. Figure 19 shows the vertical displacement of the cable tip point for these different discretizations. As it can be seen, the difference in the solution between 20 and 40 elements is very small, in contrast to the Absolute Nodal Coordinate formulation reported in [73], which required at least 40 elements to obtain a converged solution. Finally, Figure 20 shows the normal and frictional force vectors at time $t = 3.0$ s for a friction coefficient $\mu = 0.1$ s. As it can be seen, the normal force vectors are perpendicular to the isosurface of the discrete SDF, while the frictional force vectors are tangent to that isosurface, proving the accuracy of the SDF interpolation (see section 4) in modeling the gradients of the surface normal, $\frac{\partial g_N}{\partial \boldsymbol{x}}$.

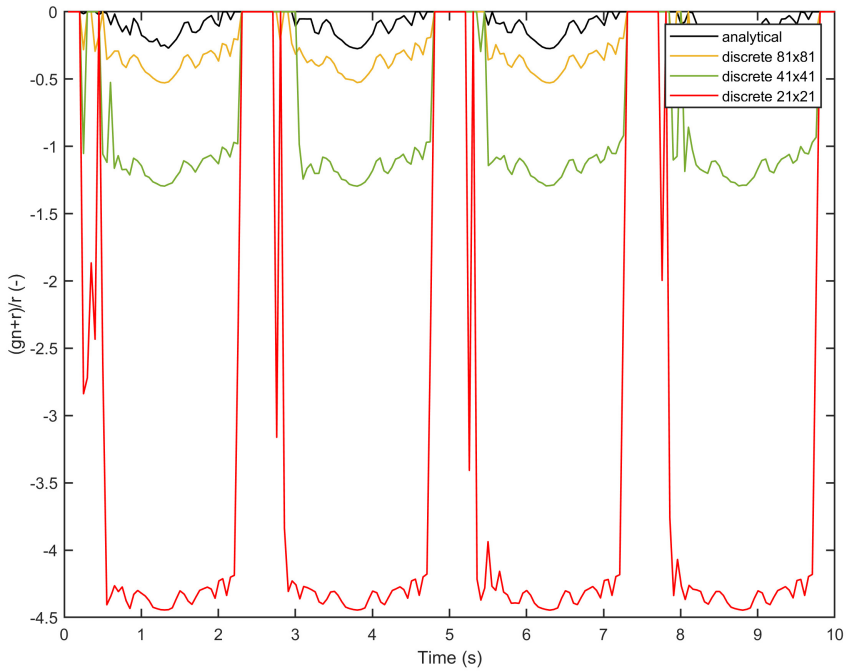


Figure 13: Beam against sphere. Normalized centerline maximum penetration using the different discretizations of the SDF as compared to using the analytical signed distance function.

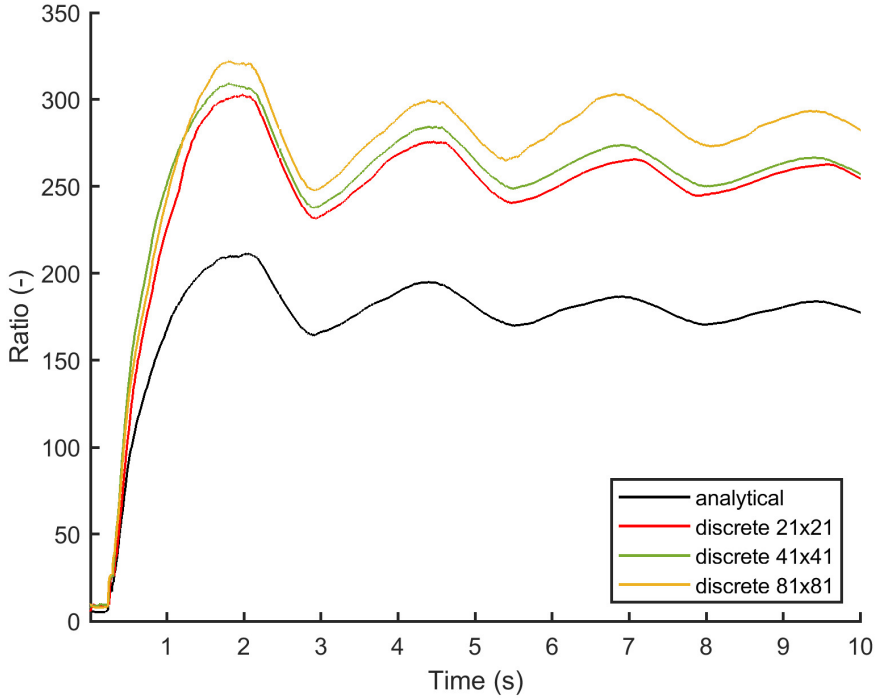


Figure 14: Beam against sphere. Ratio of the computational (wall clock) time against physical time for different discretizations of the voxel mesh.

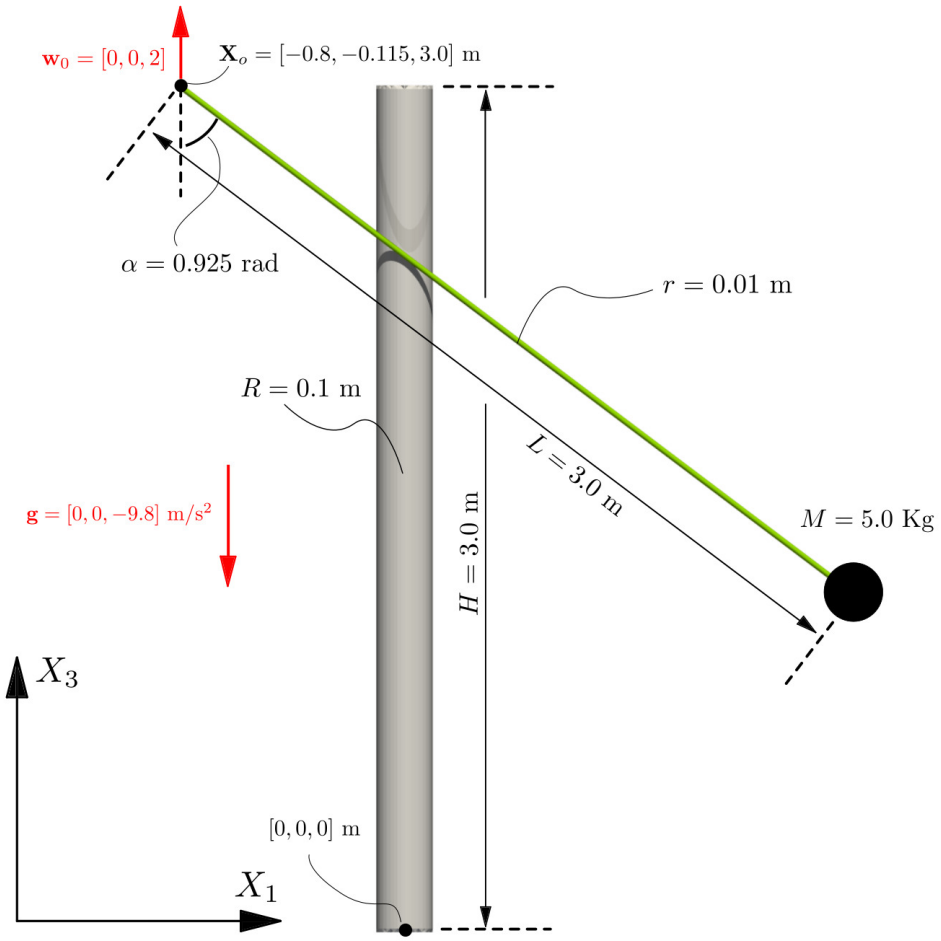
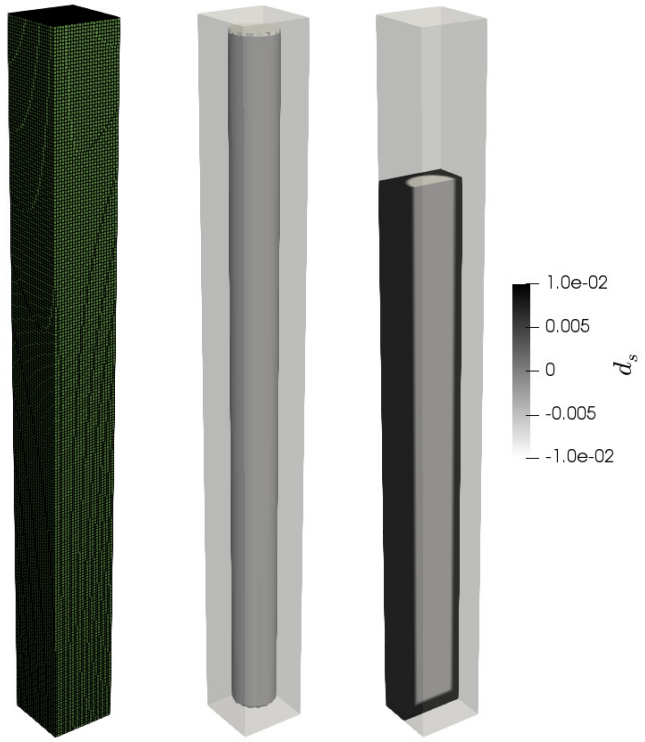
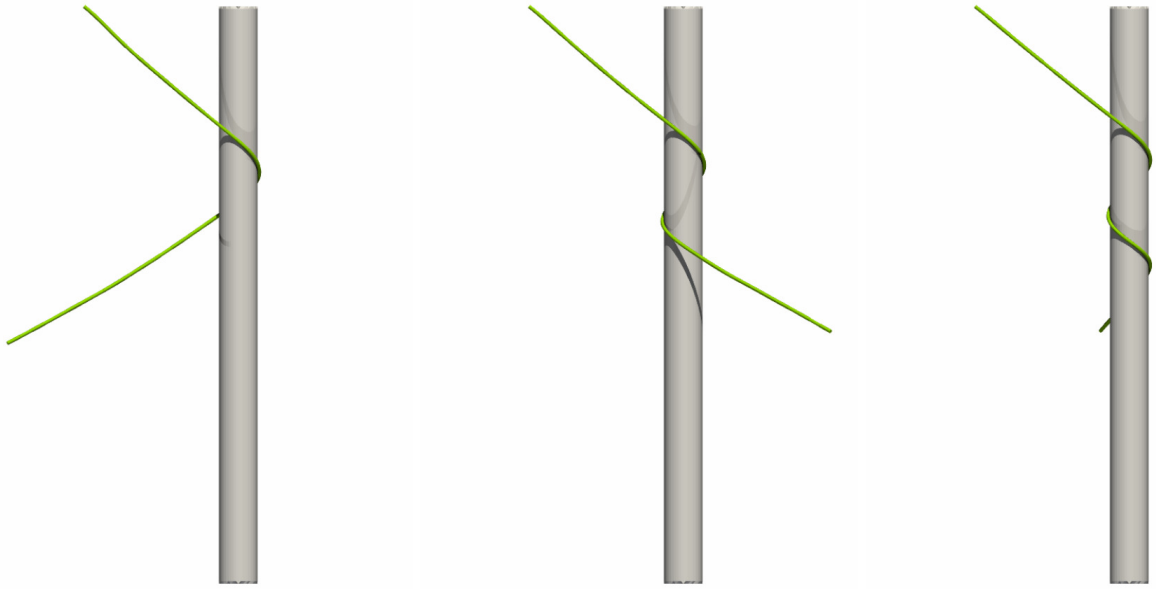


Figure 15: Contact between a cable and a rigid cylinder. Initial Setup.



(a) Voxel grid. (b) Iso-volume. (c) Clip.

Figure 16: Contact between a cable and a rigid cylinder. SDF voxel mesh.



(a) $t = 1.0$ s (b) $t = 1.5$ s (c) $t = 2.0$ s

Figure 17: Contact between a cable and a rigid cylinder. Snapshots of the solution at different time steps for $\mu = 0.1$.

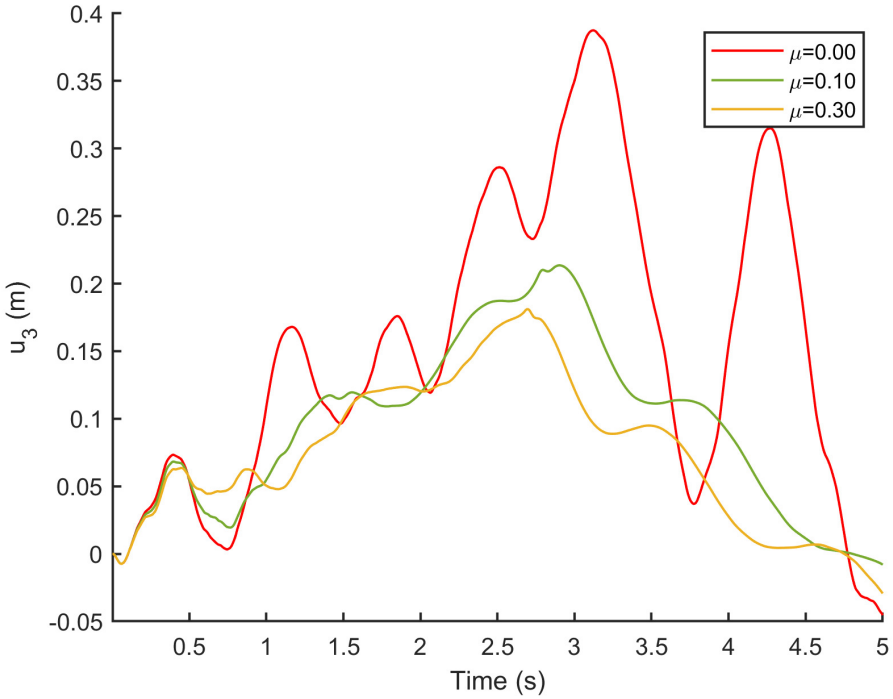


Figure 18: Contact between a cable and a rigid cylinder. u_3 displacement of the tip point for different friction coefficients.

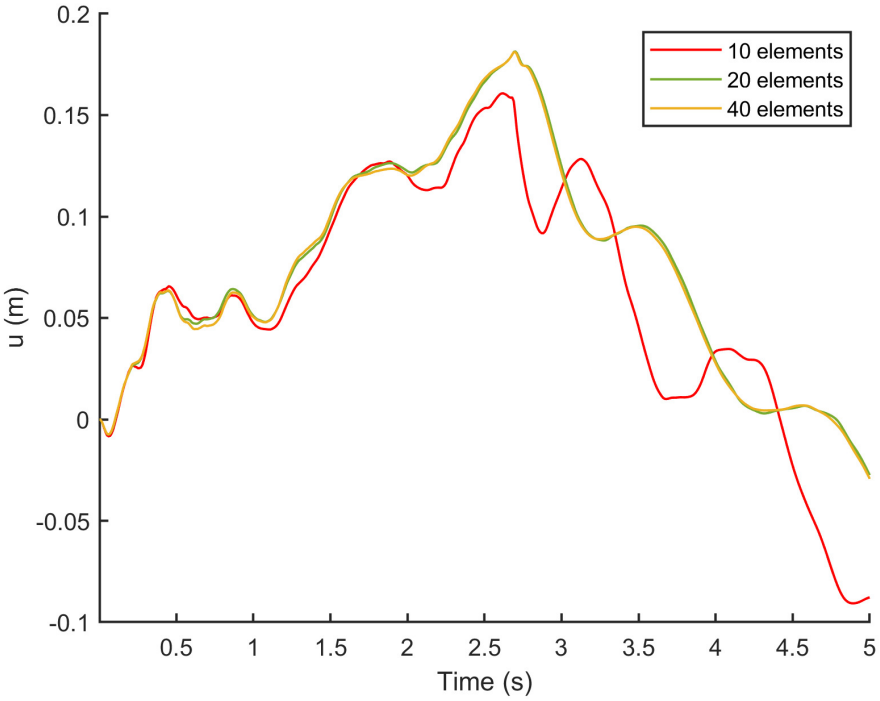
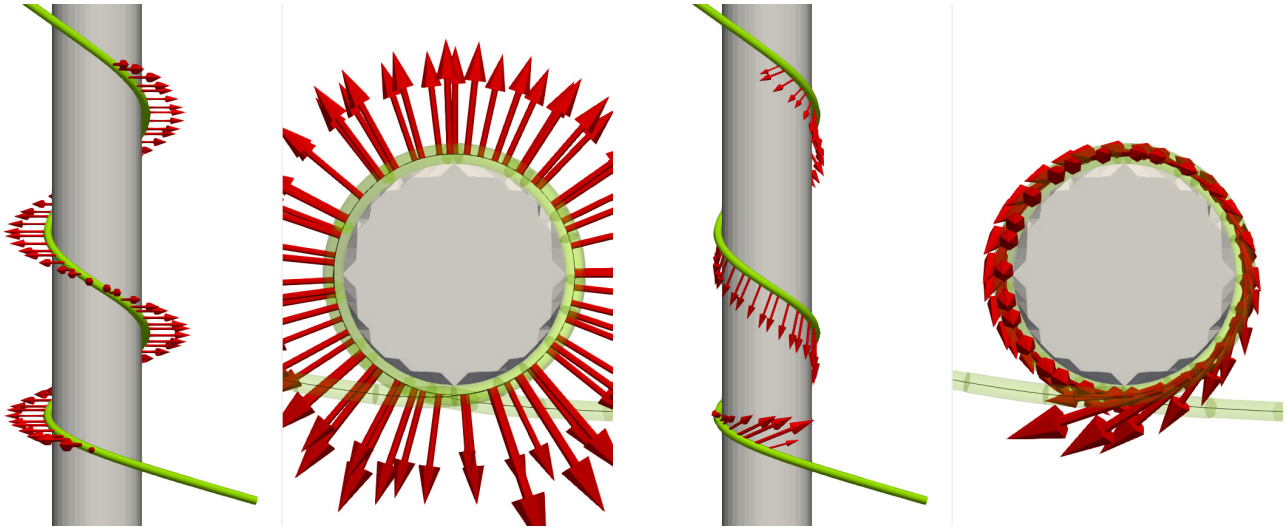


Figure 19: Contact between a cable and a rigid cylinder. u_3 displacement of the beam tip with different discretizations for $\mu = 0.3$.



(a) Normal contact vectors.

(b) Frictional contact vectors

Figure 20: Contact between a cable and a rigid cylinder. Close up of the contact friction vectors at the Gauss points. Time $t = 3$ s, friction coefficient $\mu = 0.1$.

6.4. Impact of a ring against a rigid surface

This example is designed to test the conservation properties of the algorithm [74, 75]. A flexible ring with a centerline radius $R = 9.5$ m has a circular cross section of radius $r = 0.5$ m. The ring is made of flexible material with properties $E = 100$ Pa, $\nu = 1 \cdot 10^{-4}$ and $\rho_0 = 1 \cdot 10^{-2}$ Kg/m^3. The ring has an initial uniform speed of $\boldsymbol{v}_0 = \left[\sqrt{2}, -\sqrt{2}, 0\right]^T$ m/s with its center initially located at $\boldsymbol{X}_0 = [0, 0, 0]^T$ m. A rigid surface is placed at $X_3 = -12$ m[3]. The penalty parameter is taken as $\varepsilon_c = 5000$, the time step size as $\Delta t = 0.2$ s and the ring is discretized using 64 beam elements. The simulation is run using three different friction coefficients, $\mu = \{0.00, 0.15, 0.30\}$. Figure 22 shows a snapshot of the solution at $t = 40.0$ s alongside the pathlines for $\mu = 0.00$ and $\mu = 0.30$. It can be seen that the reflection angle increases with increasing friction, as reported in [74]. Regarding the conservation of energy, Figure 23 shows the evolution of the kinetic, internal and total energy for $\mu = 0.15$. It can be seen that, by using the HHT-α method, the solution shows an overall dissipative trend, and is stable up to very long time steps, not experiencing any energy blow up, which is a crucial test in impact problems [75].

Figure 24 compares the conservation of the total energy for different friction coefficients. The energy dissipation increases with the friction coefficient, as expected. It can also be seen that the total energy dissipates even in the frictionless case, when no physical dissipation is added into the system. A large amount of this dissipation, of the order of 9 %, occurs during the contact (and sliding) against the rigid surface. This is due to the combination of the proposed contact penalty algorithm with the non-conservative HHT-α time integrator. This result is consistent with other reported results in the literature when a contact spatial discretization is combined with a non-conservative time integrator (see [18, 76, 77] for different impact examples with numerical energy loss due to a non-conservative time integrator). Briefly explained, this is due to the non-satisfaction of the persistency condition, i.e. $p_N \dot{g}_N = 0$, which is one of the fourth Kuhn-Tucker contact conditions, and the only one that involves a rate of the normal gap [76]. Given that a material time derivative is involved in this condition, the joint spatial discretization and time integration should be designed to ensure its satisfaction. Failing to do this might result in energy blow up, which is not the case for the current algorithm, or dissipation of the total energy during

[3] As the main purpose of this example is evaluating the conservation properties of the joint contact algorithm and time integrator, in this case the rigid surface is simply modeled using an analytical signed distance (trivial in the case of a flat surface).

impact (see [17] for successive energy losses during consecutive impact), which is due to negative power introduced into the system during contact [76]. Energy preserving contact time integrators have been proposed in [76–80]. It is also worth mentioning that energy preservation in the context of beam dynamics poses also a significant challenge, as time integration of the rotational variables adds additional complexity, and has been explored in, for example, [62, 81].

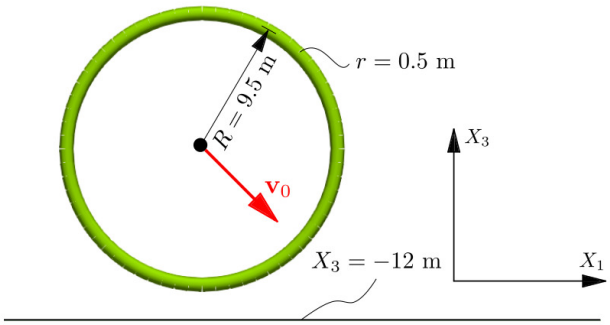
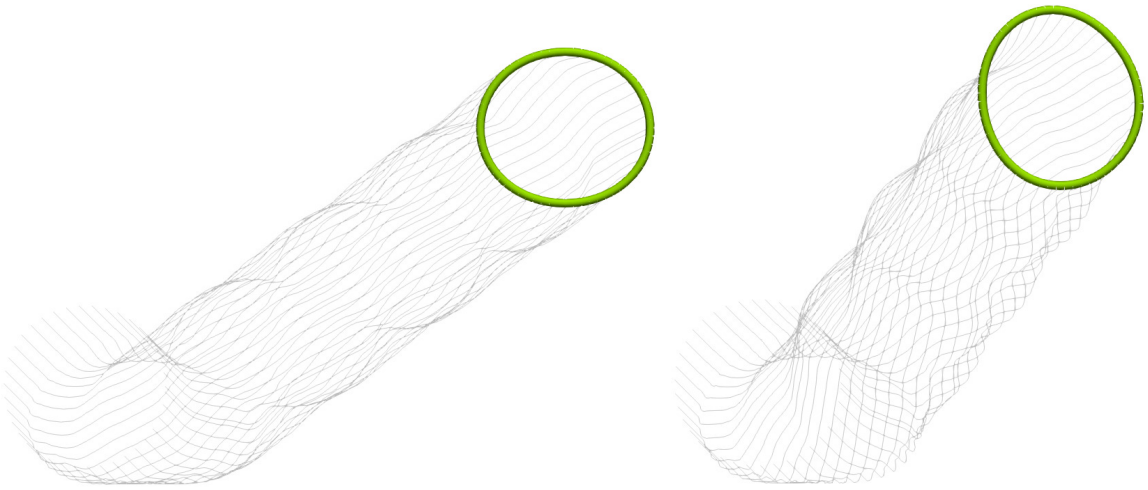


Figure 21: Impact of a ring against a rigid surface. Initial Setup.



(a) $\mu = 0.00$ (b) $\mu = 0.30$

Figure 22: Impact of a ring against a rigid surface. Comparison of the ring pathlines until $t = 40.0$ s for different friction coefficients.

6.5. Dropping net on the Stanford Dragon

This example is designed to show the robustness of the algorithm in complex contact scenarios. A net of 11×13 threads of lengths L_1 and L_2 is placed immediately above the Stanford Dragon [82], covering a larger area of that corresponding to its floor projection. The net geometry and position are detailed in Figure 25 and Table 1. The threads have a circular cross section of radius $r = 2 \cdot 10^{-4}$ m, and their material properties are $E = 10$ GPa, $\nu = 0.0$, $\rho_0 = 1500$ Kg/m^3. The net is initially at rest and under the action of gravity, defined as a constant body force $\boldsymbol{b} = [0, -9.8, 0]^T$ m/s^2. Regarding the discretization, 786 beam elements are used to model the net and the initial time step is set to $\Delta t = 5 \cdot 10^{-4}$ s. The penalty parameter is chosen as $\varepsilon_c = 1 \cdot 10^8$ and the friction coefficient as $\mu = 0.1$. The simulation is run from time $t = 0.0$ s to time $t = 19 \cdot 10^{-2}$ s, when

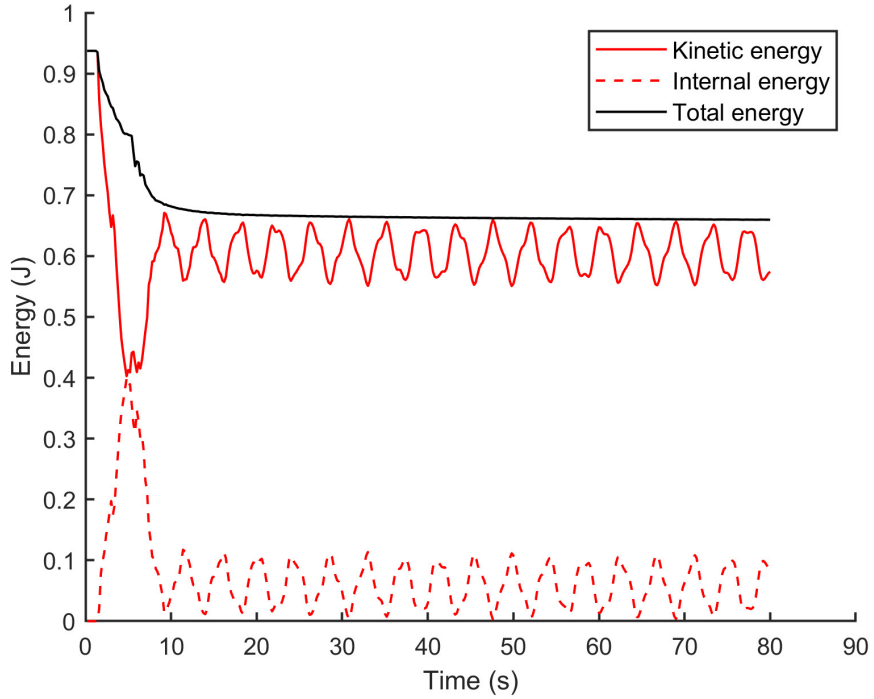


Figure 23: Impact of a ring against a rigid surface. Evolution of the kinetic, internal and contact energies for $\mu = 0.15$.

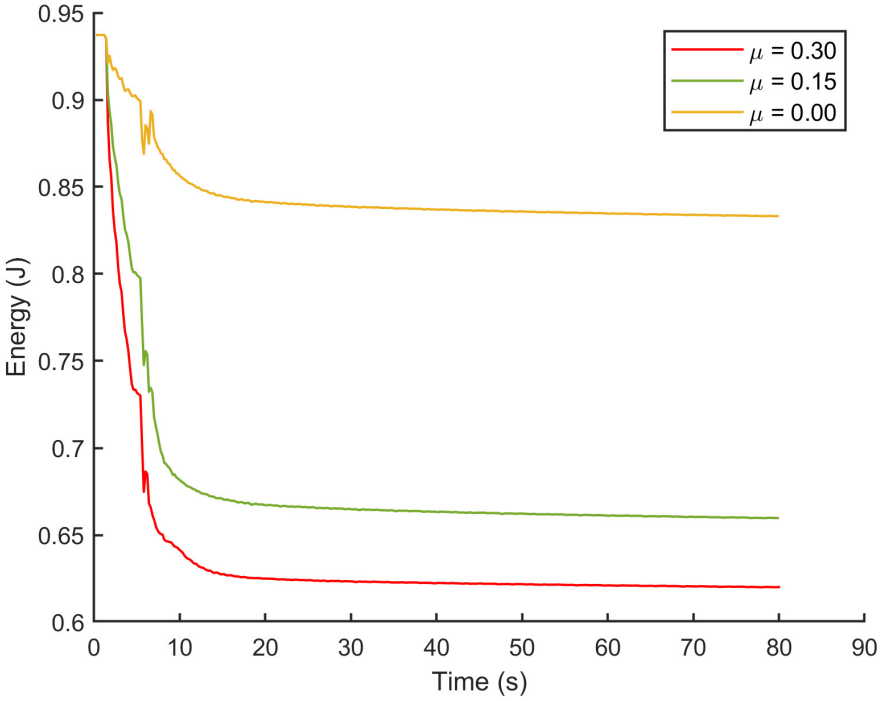
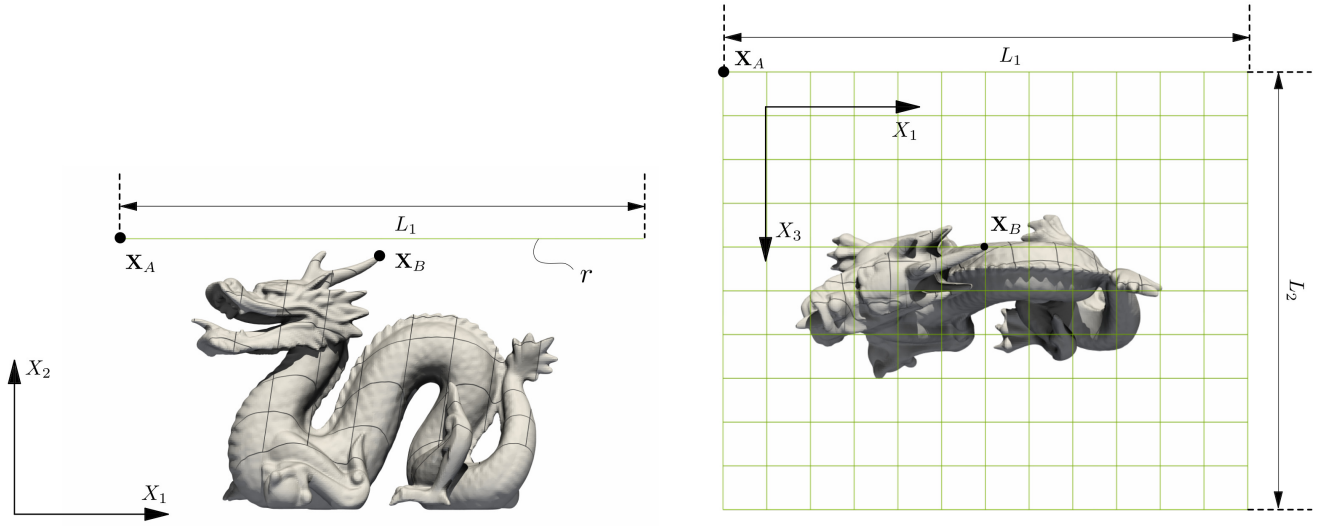


Figure 24: Impact of a ring against a rigid surface. Comparison of the total energy conservation for different friction coefficients.

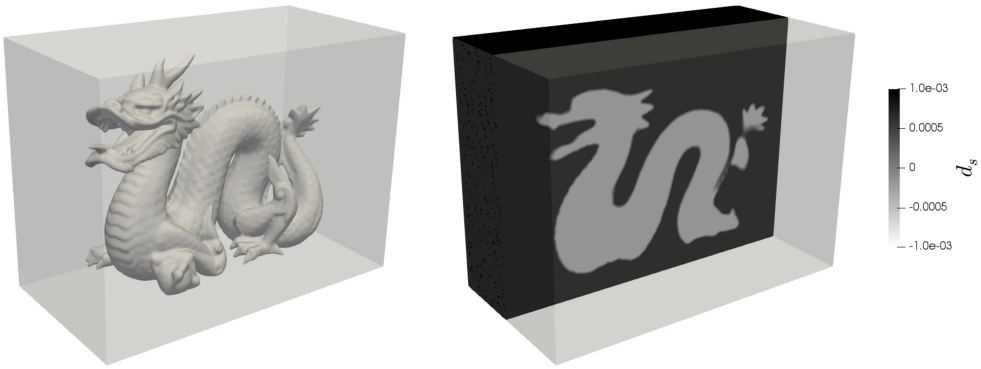
the net has covered the whole solid and prior to any self contact of the net. The Stanford Dragon SDF is given in a voxel-type grid of $236 \times 176 \times 123$ nodes with element size $h = \cdot 10^{-3}$ m (see Figure 26). Figure 27 shows the solution at different time steps. Additionally, a video animation is provided in Appendix F in the electronic version of the document. During the falling process, complex interactions between the net and the different parts of the dragon geometry occur, which are accurately captured by the proposed algorithm. Figure 28 highlights the solution at the final time step, by showing different views. In order to assess the penetration, Figure 29a evaluates the percentage of Gauss points that are in contact or penetrating the body. Gauss points in contact are identified as those that have a normal gap $g_N \leq \bar{g}_N = 0.1r$ (see section 3). Further to that, the number of Gauss points at different levels of penetration (represented as a fraction of the beam radius) are also computed. As it can be seen, all Gauss points are either contacting ($0.0 \leq g_N \leq \bar{g}_N$) or non-contacting the Dragon surface, with no Gauss point penetrating the surface. Next, the same evaluation is carried out by processing the penetration of the beam points in Figure 29b. To do so, the centerline position is postprocessed in the same way as in example 6.2, but now the distance $d_s(\boldsymbol{x})$ is evaluated using the interpolation technique described in section 4, as no analytical distance function is available. As it can be seen, the majority of the beam length in the contact region (i.e. $g_N \leq \bar{g}_N$) has a penetration between the range $-r \leq g_N \leq 0.1r$ (i.e. the majority of the points in the contact region do not penetrate further than the beam radius). In short, the algorithm ensures that no Gauss point penetrates the surface and this results into a small percentage of the beam length penetrating the dragon surface. This is specially important as the voxel mesh element size is significantly larger than the beam radius, i.e. $h/r = 5.0$, which proves the accuracy of the contact algorithm proposed in section 3. To further strengthen this last point, Figure 30 shows the maximum and mean penetration of the centerline points with $g_N \leq 0$. It can be seen that the maximum penetration is smaller than $2r$ ($0.4h$), while the mean penetration is of the order of $0.4r$ ($0.08h$).



(a) Lateral view.

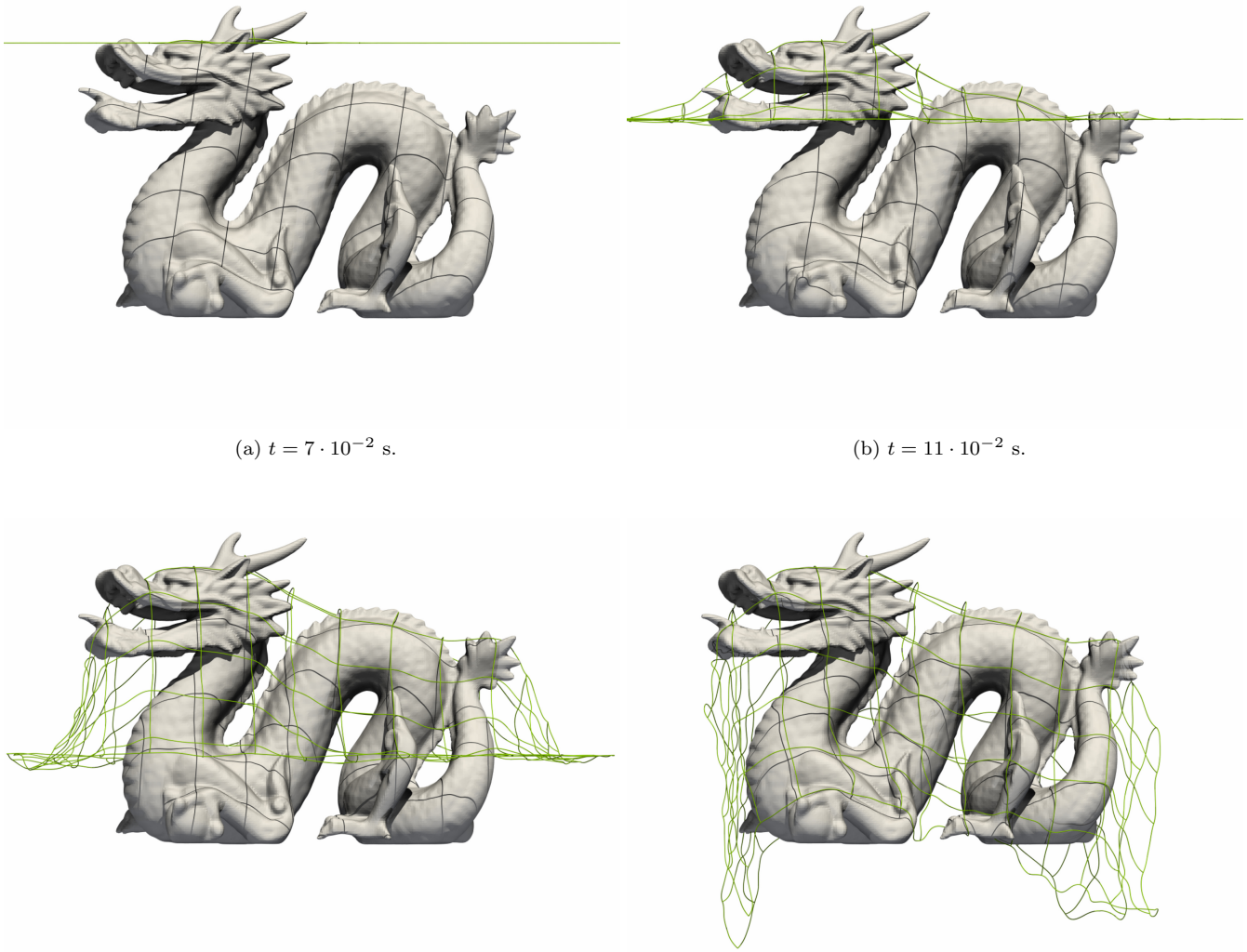
(b) Top view.

Figure 25: Dropping net on the Stanford Dragon. Initial setup.



(a) Iso-volume. (b) Clip.

Figure 26: Dropping net on the Stanford Dragon. SDF voxel mesh.



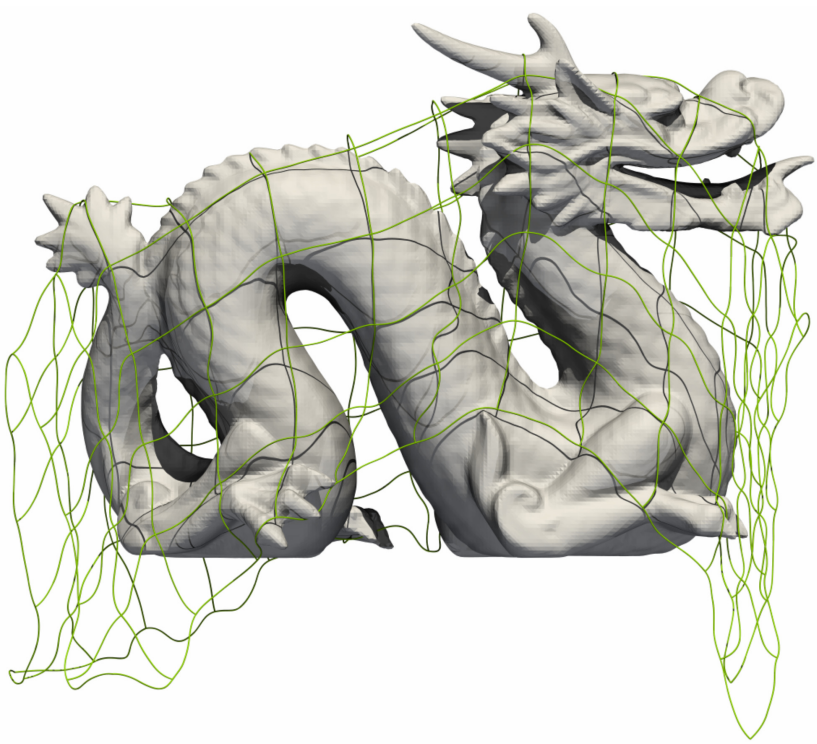
(a) $t = 7 \cdot 10^{-2}$ s. (b) $t = 11 \cdot 10^{-2}$ s.

(c) $t = 15 \cdot 10^{-2}$ s. (d) $t = 19 \cdot 10^{-2}$ s.

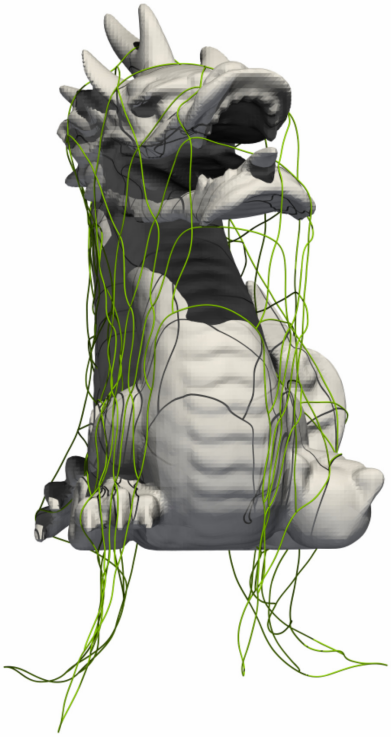
Figure 27: Dropping net on the Stanford Dragon. Position of the grid at different time steps.

$\boldsymbol{X}_A$	$\left[9.375 \cdot 10^{-2}, 2.55 \cdot 10^{-1}, -1.875 \cdot 10^{-3}\right]^T$ m
$\boldsymbol{X}_B$	$\left[3.8625 \cdot 10^{-1}, 2.55 \cdot 10^{-1}, 241875 \cdot 10^{-1}\right]^T$ m
L_1	$2.925 \cdot 10^{-1}$ m
L_2	$2.4375 \cdot 10^{-1}$ m

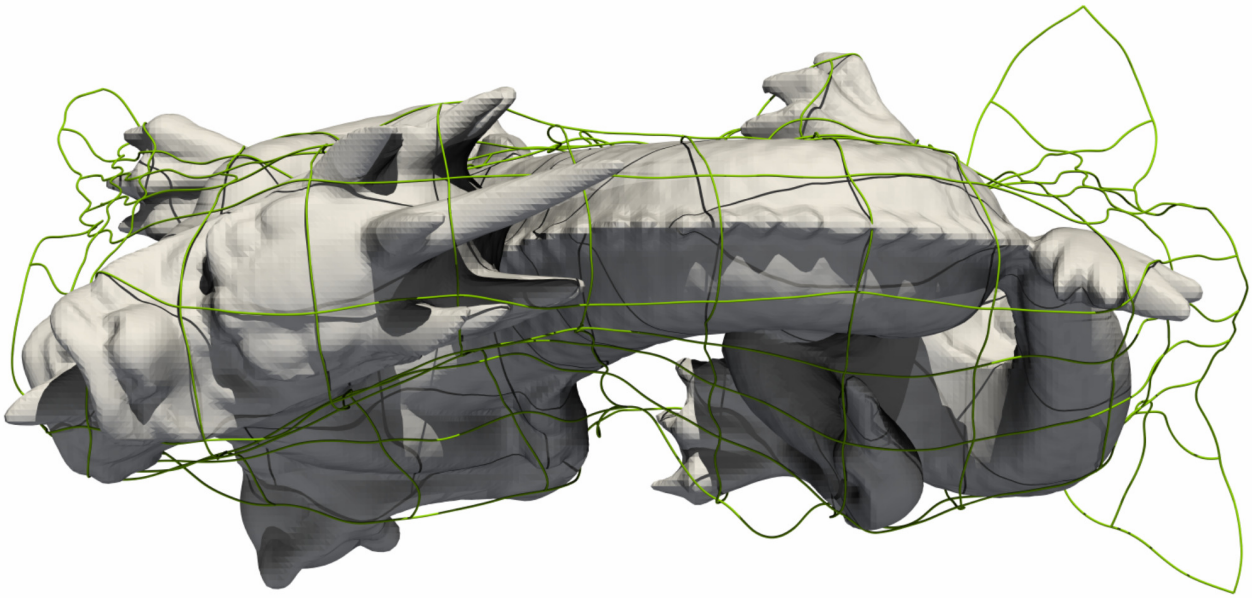
Table 1: Dropping net on the Stanford Dragon. Geometric information of the initial setup.



(a) Lateral view.

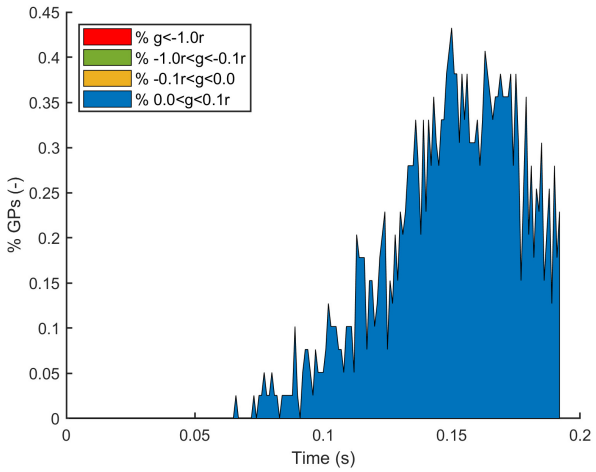


(b) Front view.

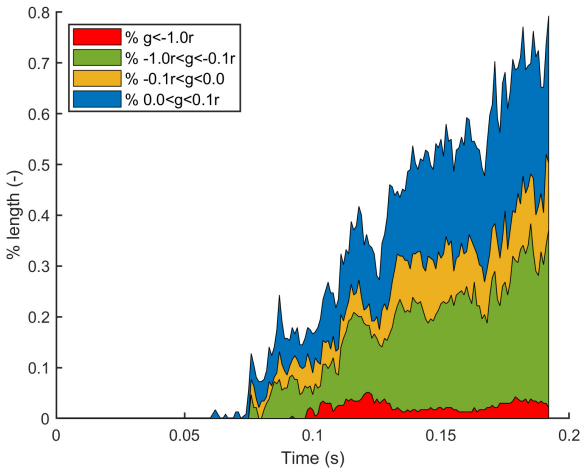


(c) Top view.

Figure 28: Dropping net on the Stanford Dragon. Different views of the solution at time $t = 19 \cdot 10^{-2}$ s.



(a) Percentage of Gauss points.



(b) Percentage of beam centerline length.

Figure 29: Dropping net on the Stanford Dragon. Stacked area plots showing the percentage of Gauss points and length contacting / penetrating the surface.

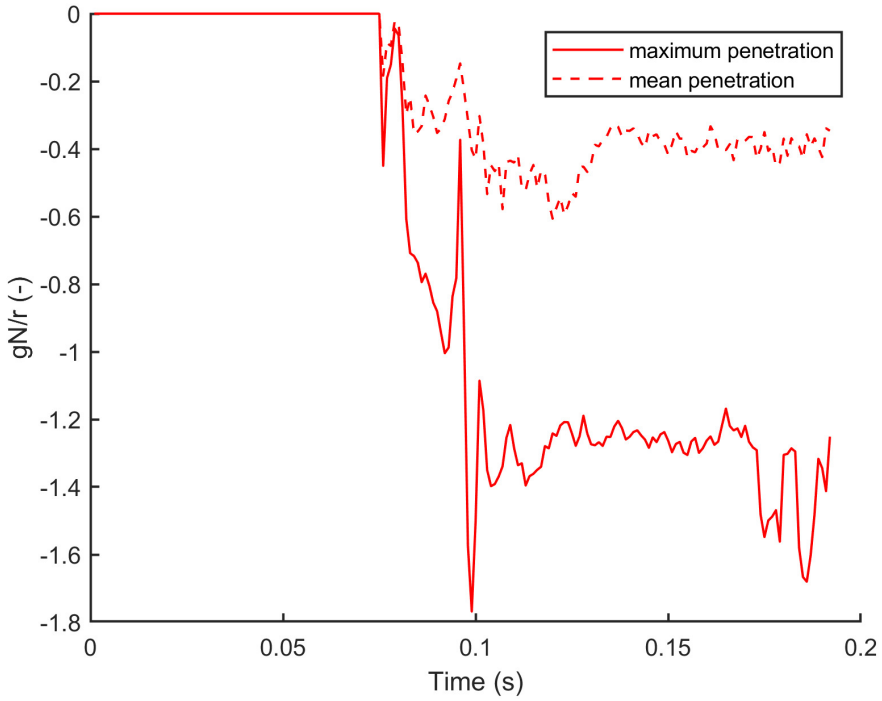


Figure 30: Dropping net on the Stanford Dragon. Maximum and mean normalized penetration.

7. Conclusions

This paper pursued the development of a fast, yet accurate and robust, algorithm for beam to surface contact. To do so, a new implicit dynamics formulation for the simulation of 3D beam contact against rigid surfaces has been presented. The formulation builds from an efficient corotational beam dynamics formulation previously reported. The corotational beam kinematics has been rewritten using 3D kinematics description, linking it with geometrically exact beam theory. Frictional contact terms have been added and linearized, providing a consistent implicit dynamics formulation for frictional contact. Contact search is avoided by using a discrete signed distance field to represent the rigid surface, in which the Lagrangian beam is immersed. The contact distance, gradient and Hessian, as required by the beam formulation, are computed at the beam Gauss point by using an interpolation based in a tensor product of 1D kernels, as used in immersed fluid-structure interaction techniques. The use of this interpolation, facilitated by the Cartesian structure of the voxel mesh, is highly efficient as the stencil can be precomputed and no search is required when computing the distance to the surface.

Regarding the numerical results, it has been numerically proved that the interpolated signed distance, spatial gradient and Hessian converge with second-order accuracy to the analytical solution, i.e. $\mathcal{O}(h^2)$, for smooth surfaces while first-order convergence, i.e. $\mathcal{O}(h)$, is expected for sharp surfaces. This fact, alongside the beam discretization, determines the contact algorithm order of accuracy. Next, the robustness and applicability of the overall contact framework has been proven by providing a series of challenging examples. Firstly, the solution matches well against literature results in a beam fast impact problem. Next, an example has been proposed to test the accuracy of the surface discretization by comparing the penetration for different voxel sizes. Using the same parameters for the beam material and discretization, it can be seen that the penetration decreases for finer voxel mesh sizes, as expected. In the same example, the efficiency of the interpolation strategy has been assessed qualitatively, proving that the computational time is barely affected by the refinement of the voxel mesh. A literature example in a friction dominated scenario shows that the proposed algorithm converges faster than previously reported results using Absolute Nodal Coordinate formulation. Using a standard benchmark, it has also been proved that the combination of the proposed frictional contact terms with the HHT-α time integrator provide a globally dissipative solution, avoiding energy blow-ups after long physical time. Finally, the algorithm has been tested in a complex contact scenario by using a SDF of the Stanford Dragon. The penetration levels of the beam centerline are of the order of the beam radius, which is smaller than the voxel mesh size and which proves the robustness of the contact algorithm.

The proposed framework could be improved in different ways. Firstly, the algorithm has been implemented in Matlab in a non-optimized code (dense matrix, no vectorization). Therefore, the current implementation allows for substantial optimization via use of sparse matrices and lower level languages. Once this is done, and keeping in mind that both the corotational beam models and the 1D kernel interpolation have been proven to be effective algorithms in their respective areas of application, the overall framework could be an excellent candidate for accurate real-time contact simulation. Yet, a detailed comparison, in terms of efficiency, of the proposed framework against other standard contact algorithms is required in order to prove that hypothesis. Regarding the time integration, despite the HHT-α method provides a globally dissipative solution, as pointed in [50], energy-preserving time integrators could be implemented, which would improve the solution in terms of time accuracy. A significant limitation of the current formulation is that it only considers contact against rigid surfaces. In order to extend it to contact against deformable surfaces, the use of an Eulerian formulation for hyperelastic solids could be explored [29]. Finally, the current formulation does not consider contact (or self contact) of the beam elements. To model such phenomena, a unified beam to beam contact approach as the one presented in [47] could be adapted from a Total Lagrangian to a (Lagrangian) corotational beam formulation as presented in this paper. Overall, this would result into a Lagrangian beam to beam contact formulation

combined with the presented SDF beam to surface contact formulation.

Finally, it is worth mentioning that the current algorithm is well-suited to its application in cardiovascular medicine and, specifically, simulation of guidewire navigation and stent-graft deployment [12, 83]. In these applications efficient or even real time simulations are needed for training purposes and for decision making during the pre-operative and intra-operative phases. Additionally, medical imaging data is traditionally provided in voxel type meshes where SDF can be obtained automatically [84]. By using the proposed framework, the creation of patient-specific triangular meshes would be avoided, which usually requires experienced user input and takes an important part of the simulation work-flow [85]. Yet, cardiovascular applications, in general, require taking into account the deformation of the vessel wall [12]. Therefore, the presented work can be seen as a first step towards this final goal.

8. Acknowledgments

The authors are grateful to the European Research Council for ERC grant Biolochanics, grant number 647067.

Appendix A. Kinematics assuming small displacements and rotations

Assuming small rotations in $\bar{\Omega}$, the rotation tensor $\bar{\boldsymbol{R}}$ can be approximated as

$$\bar{\boldsymbol{R}}(\bar{\boldsymbol{\Theta}}) \simeq \boldsymbol{I} + \boldsymbol{S}(\bar{\boldsymbol{\Theta}}) \tag{A.1}$$

and its first derivative as

$$\bar{\boldsymbol{R}}' \simeq \boldsymbol{S}(\bar{\boldsymbol{\Theta}}'). \tag{A.2}$$

At the same time, by assuming small displacement of the centerline in $\bar{\Omega}$, the derivatives of the transversal displacements are given by

$$u_2'(s) \simeq \bar{\Theta}_3(s), \tag{A.3}$$
$$u_3'(s) \simeq -\bar{\Theta}_2(s). \tag{A.4}$$

The above approximations are used to better analyze the structure of the deformation gradient $\bar{\boldsymbol{F}}$, as defined in (43), which is repeated for convenience,

$$\bar{\boldsymbol{F}}(s) = \bar{\boldsymbol{R}}(s) + \bar{\boldsymbol{a}}(s) \otimes \bar{\boldsymbol{E}}_1, \tag{A.5}$$

with $\bar{\boldsymbol{a}}$ given by (see equation (44))

$$\bar{\boldsymbol{a}}(s) = \bar{\boldsymbol{\gamma}}(s) + \xi_\alpha \bar{\boldsymbol{\kappa}}_\alpha(s), \tag{A.6}$$

and, via equations (45), (46),

$$\bar{\boldsymbol{\gamma}} = \bar{\boldsymbol{x}}_0'(s) - \bar{\boldsymbol{t}}_1, \tag{A.7}$$
$$\bar{\boldsymbol{\kappa}}_\alpha = \bar{\boldsymbol{R}}'(s)\bar{\boldsymbol{E}}_\alpha. \tag{A.8}$$

Using equations (34), (35) alongside approximations (A.3), (A.4), $\bar{\boldsymbol{x}}_0'$ can be written as

$$\bar{\boldsymbol{x}}_0' = \bar{\boldsymbol{E}}_1 + u_1'(s)\bar{\boldsymbol{E}}_1 + u_2'(s)\bar{\boldsymbol{E}}_2 + u_3'(s)\bar{\boldsymbol{E}}_3 \simeq u_1'(s)\bar{\boldsymbol{E}}_1 + \bar{\boldsymbol{t}}_1, \tag{A.9}$$

which yields

$$\bar{\boldsymbol{\gamma}} = u_1'(s)\bar{\boldsymbol{E}}_1. \tag{A.10}$$

At the same time, via approximation (A.2) and by omitting the high order terms, $\bar{\boldsymbol{\kappa}}_\alpha$ can be rewritten as

$$\bar{\boldsymbol{\kappa}}_2 = \left(\bar{\boldsymbol{\Theta}}'(s) \times \bar{\boldsymbol{T}}_2\right) \simeq \begin{bmatrix} \bar{\Theta}'_1 \\ \bar{\Theta}'_2 \\ \bar{\Theta}'_3 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} \times \begin{bmatrix} -\bar{\Theta}_3 \\ 1 \\ \bar{\Theta}_1 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} = \begin{bmatrix} \bar{\Theta}'_2\bar{\Theta}_1 - \bar{\Theta}'_3 \\ -\bar{\Theta}_3\bar{\Theta}'_3 - \bar{\Theta}'_1\bar{\Theta}_1 \\ \bar{\Theta}'_1 - \bar{\Theta}_3\bar{\Theta}'_2 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} \simeq \begin{bmatrix} -\bar{\Theta}'_3 \\ 0 \\ \bar{\Theta}'_1 \end{bmatrix}_{\bar{\boldsymbol{E}}_i}, \tag{A.11}$$

$$\bar{\boldsymbol{\kappa}}_3 = \left(\bar{\boldsymbol{\Theta}}'(s) \times \bar{\boldsymbol{T}}_3\right) \simeq \begin{bmatrix} \bar{\Theta}'_1 \\ \bar{\Theta}'_2 \\ \bar{\Theta}'_3 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} \times \begin{bmatrix} \bar{\Theta}_2 \\ -\bar{\Theta}_1 \\ 1 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} = \begin{bmatrix} \bar{\Theta}'_2 + \bar{\Theta}'_3\bar{\Theta}_1 \\ \bar{\Theta}_2\bar{\Theta}'_3 - \bar{\Theta}'_1 \\ -\bar{\Theta}'_1\bar{\Theta}_1 - \bar{\Theta}'_2\bar{\Theta}_2 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} \simeq \begin{bmatrix} \bar{\Theta}'_2 \\ -\bar{\Theta}'_1 \\ 0 \end{bmatrix}_{\bar{\boldsymbol{E}}_i}. \tag{A.12}$$

Using the results in equations (A.10), (A.11), (A.12) into (A.6) finally yields

$$\bar{\boldsymbol{a}} = \begin{bmatrix} u'_1 - \xi_2\bar{\Theta}'_3 + \xi_3\bar{\Theta}'_2 \\ -\xi_3\bar{\Theta}'_1 \\ \xi_2\bar{\Theta}'_1 \end{bmatrix}_{\bar{\boldsymbol{E}}_i}. \tag{A.13}$$

On the other hand, the displacement $\bar{\boldsymbol{u}}$ can be approximated assuming small rotations by replacing approximation (A.1) into equation (35), to yield

$$\bar{\boldsymbol{u}}(s) = \begin{bmatrix} u_1(s) \\ u_2(s) \\ u_3(s) \end{bmatrix}_{\bar{\boldsymbol{E}}_i} + \xi_2 \begin{bmatrix} -\bar{\Theta}_3 \\ 1 \\ \bar{\Theta}_1 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} + \xi_3 \begin{bmatrix} \bar{\Theta}_2 \\ -\bar{\Theta}_1 \\ 1 \end{bmatrix}_{\bar{\boldsymbol{E}}_i} = \begin{bmatrix} u_1(s) - \xi_2\bar{\Theta}_3(s) + \xi_3\bar{\Theta}_2(s) \\ u_2(s) + \xi_2 - \xi_3\bar{\Theta}_1(s) \\ u_3(s) + \xi_2\bar{\Theta}_1(s) + \xi_3 \end{bmatrix}_{\bar{\boldsymbol{E}}_i}, \tag{A.14}$$

and the gradient can be obtained by reminding that, in the configuration $\bar{\Omega}$, $\bar{X}_1 = s$, $\bar{X}_\alpha = \xi_\alpha$, which yields

$$\nabla\bar{\boldsymbol{u}} = \frac{\partial\bar{\boldsymbol{u}}}{\partial\bar{\boldsymbol{X}}} \simeq \begin{bmatrix} u'_1(s) - \xi_2\bar{\Theta}'_3(s) + \xi_3\bar{\Theta}'_2(s) & -\bar{\Theta}_3(s) & \bar{\Theta}_2(s) \\ \bar{\Theta}_3(s) - \xi_3\bar{\Theta}'_1(s) & 1 & -\bar{\Theta}_1(s) \\ -\bar{\Theta}_2(s) + \xi_2\bar{\Theta}'_1(s) & \bar{\Theta}_1(s) & 1 \end{bmatrix} \tag{A.15}$$

where, again, the small displacement assumption (A.3), (A.4) has been used. Finally, comparing the result above with equation (A.13), yields

$$\nabla\bar{\boldsymbol{u}} \simeq \boldsymbol{S}(\bar{\boldsymbol{\Theta}}) + \bar{\boldsymbol{a}} \otimes \boldsymbol{E}_1. \tag{A.16}$$

The above gradient can be approximated using a sum of symmetric and antisymmetric tensors

$$\nabla\bar{\boldsymbol{u}} = \boldsymbol{\varepsilon}(\nabla\bar{\boldsymbol{u}}) + \boldsymbol{\Omega}(\nabla\bar{\boldsymbol{u}}), \tag{A.17}$$

with

$$\boldsymbol{\varepsilon}(\nabla\bar{\boldsymbol{u}}) = \frac{1}{2}\begin{bmatrix} 2\left(u'_1(s) - \xi_2\bar{\Theta}'_3(s) + \xi_3\bar{\Theta}'_2(s)\right) & -\xi_3\bar{\Theta}'_1(s) & \xi_2\bar{\Theta}'_1(s) \\ -\xi_3\bar{\Theta}'_1(s) & 0 & 0 \\ \xi_2\bar{\Theta}'_1(s) & 0 & 0 \end{bmatrix} = \frac{1}{2}\left(\bar{\boldsymbol{a}} \otimes \boldsymbol{E}_1 + \boldsymbol{E}_1 \otimes \bar{\boldsymbol{a}}\right), \tag{A.18}$$

$$\boldsymbol{\Omega}(\nabla\bar{\boldsymbol{u}}) = \frac{1}{2}\begin{bmatrix} 0 & -2\bar{\Theta}_3 + \xi_3\bar{\Theta}'_1 & 2\bar{\Theta}_2 - \xi_2\bar{\Theta}'_1 \\ 2\bar{\Theta}_3 - \xi_3\bar{\Theta}'_1 & 0 & -2\bar{\Theta}_1 \\ -2\bar{\Theta}_2 + \xi_2\bar{\Theta}'_1 & 2\bar{\Theta}_1 & 0 \end{bmatrix} = \boldsymbol{S}(\bar{\boldsymbol{\Theta}}) + \frac{1}{2}\left(\bar{\boldsymbol{a}} \otimes \boldsymbol{E}_1 - \boldsymbol{E}_1 \otimes \bar{\boldsymbol{a}}\right). \tag{A.19}$$

Appendix B. Isoparametric interpolation

The Finite Element interpolation matrices are

$$\boldsymbol{P}_1(\xi) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N_3(\xi) & 0 & 0 & N_4(\xi) \\ 0 & -N_3(\xi) & 0 & 0 & -N_4(\xi) & 0 \end{bmatrix} \tag{B.1}$$

$$\boldsymbol{P}_2(\xi) = \begin{bmatrix} N_1(\xi) & 0 & 0 & N_2(\xi) & 0 & 0 \\ 0 & N_5(\xi) & 0 & 0 & N_6(\xi) & 0 \\ 0 & 0 & N_5(\xi) & 0 & 0 & N_6(\xi) \end{bmatrix} \tag{B.2}$$

with $\xi \in [0, l_0]$ and the shape functions given by

$$N_1(\xi) = 1 - \frac{\xi}{l_0}, \quad N_2(\xi) = 1 - N_1, \tag{B.3}$$

$$N_3(\xi) = \xi \left(1 - \frac{\xi}{l_0}\right)^2, \quad N_4(\xi) = -\left(1 - \frac{\xi}{l_0}\right)\frac{\xi^2}{l_0}, \tag{B.4}$$

$$N_5(\xi) = \left(1 - \frac{3\xi}{l_0}\right)\left(1 - \frac{\xi}{l_0}\right), \quad N_6(\xi) = \left(\frac{3\xi}{l_0} - 2\right)\frac{\xi}{l_0}. \tag{B.5}$$

Appendix C. Internal force vector, stiffness matrix

The relationship between $\delta\boldsymbol{d}$ and $\delta\bar{\boldsymbol{\mathcal{D}}}$ requires several steps, as explained in [38]. For the sake of completeness, the main results are summarized in what follows. Firstly, the variation of $\bar{\boldsymbol{R}}_a$ requires the spin vector $\delta\bar{\boldsymbol{W}}_a$, which does not appear in the variational equation (54). Therefore, an additional set of nodal quantities is considered that inclues the variation of the spin vectors $\bar{\boldsymbol{W}}_a$,

$$\bar{\boldsymbol{\mathcal{D}}}_* = \begin{bmatrix} \bar{u} & \bar{\boldsymbol{W}}_1^T & \bar{\boldsymbol{W}}_2^T \end{bmatrix}^T; \quad \delta\bar{\boldsymbol{\mathcal{D}}}_* = \begin{bmatrix} \delta\bar{u} & \delta\bar{\boldsymbol{W}}_1^T & \delta\bar{\boldsymbol{W}}_2^T \end{bmatrix}^T, \tag{C.1}$$

The relationship between $\delta\bar{\boldsymbol{\mathcal{D}}}$ and $\delta\bar{\boldsymbol{\mathcal{D}}}_*$ given via equation (14a) as

$$\delta\bar{\boldsymbol{\mathcal{D}}} = \bar{\boldsymbol{B}}\delta\bar{\boldsymbol{\mathcal{D}}}_*; \quad \bar{\boldsymbol{B}} = \begin{bmatrix} 1 & \boldsymbol{0}^T & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{T}_s^{-1}(\bar{\boldsymbol{\Theta}}_1) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0} & \boldsymbol{0}_{3\times3} & \boldsymbol{T}_s^{-1}(\bar{\boldsymbol{\Theta}}_2) \end{bmatrix}. \tag{C.2}$$

Next, the relationship between $\delta\bar{\boldsymbol{\mathcal{D}}}_*$ and $\delta\boldsymbol{d}$ is given by

$$\delta\bar{\boldsymbol{\mathcal{D}}}_* = \bar{\boldsymbol{B}}_*\delta\boldsymbol{d}, \quad \bar{\boldsymbol{B}}_* = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{P}\boldsymbol{E}^T \end{bmatrix}, \tag{C.3}$$

with $\boldsymbol{E}$ given in equation (70). The vector $\boldsymbol{r}$ gives the relationship between the variation $\delta\bar{u}$ and the variation of the global displacements, i.e. and the expressions for $\boldsymbol{r}$ and $\boldsymbol{P}$ given by

$$\delta\bar{u} = \boldsymbol{r}\delta\boldsymbol{d}; \quad \boldsymbol{r} = \begin{bmatrix} -\boldsymbol{v}_1{}^T & \boldsymbol{0}^T & \boldsymbol{v}_1{}^T & \boldsymbol{0}^T \end{bmatrix}. \tag{C.4}$$

At the same time, using equation (65), $\boldsymbol{P}$ gives the relationship between the variation of the spin variables in $\bar{\Omega}$, $\delta\bar{\boldsymbol{W}}_a$ and $\delta\boldsymbol{\mathcal{D}}$, i.e.

$$\begin{bmatrix} \delta\bar{\boldsymbol{W}}_1 \\ \delta\bar{\boldsymbol{W}}_2 \end{bmatrix} = \boldsymbol{P}\delta\boldsymbol{\mathcal{D}} = \boldsymbol{P}\boldsymbol{E}^T\delta\boldsymbol{d}; \quad \boldsymbol{P} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} - \begin{bmatrix} \boldsymbol{G}^T \\ \boldsymbol{G}^T \end{bmatrix}, \tag{C.5}$$

and

$$\boldsymbol{G}^T = \frac{\partial \boldsymbol{W}_e}{\partial \boldsymbol{\mathcal{D}}} = \begin{bmatrix} 0 & 0 & \eta/l_n & \eta_{12}/2 & -\eta_{11}/2 & 0 & 0 & 0 & -\eta/l_n & \eta_{22}/2 & -\eta_{21}/2 & 0 \\ 0 & 0 & 1/l_n & 0 & 0 & 0 & 0 & 0 & -1/l_n & 0 & 0 & 0 \\ 0 & -1/l_n & 0 & 0 & 0 & 0 & 0 & 1/l_n & 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.6}$$

with

$$\eta = \frac{q_1}{q_2}, \quad \eta_{11} = \frac{q_{11}}{q_2}, \quad \eta_{12} = \frac{q_{12}}{q_2}, \quad \eta_{21} = \frac{q_{21}}{q_2}, \quad \eta_{22} = \frac{q_{22}}{q_2}. \tag{C.7}$$

Combining equations (C.2) and (C.3) yields the relationship between $\delta \boldsymbol{d}$ and $\delta \bar{\boldsymbol{\mathcal{D}}}$

$$\delta \boldsymbol{d} = \boldsymbol{B} \delta \bar{\boldsymbol{\mathcal{D}}}; \quad \boldsymbol{B} = \bar{\boldsymbol{B}} \bar{\boldsymbol{B}}_*. \tag{C.8}$$

Given the above relationship, the tangent stiffness can be obtained via equations (90), (94)

$$\delta \boldsymbol{T}_{int} = \delta(\boldsymbol{B}^T \bar{\boldsymbol{T}}_{int}) = \delta(\bar{\boldsymbol{B}}_*^T \bar{\boldsymbol{B}}^T \bar{\boldsymbol{T}}_{int}) = \boldsymbol{K}_{int} \delta \boldsymbol{d}, \tag{C.9}$$

with

$$\boldsymbol{K}_{int} = \boldsymbol{B}^T \bar{\boldsymbol{K}}_{int} \boldsymbol{B} + \tilde{\boldsymbol{K}}; \quad \tilde{\boldsymbol{K}} = \bar{\boldsymbol{B}}_*^T \bar{\boldsymbol{K}}_h \bar{\boldsymbol{B}}_* + \boldsymbol{K}_m \tag{C.10}$$

with $\bar{\boldsymbol{K}}$ given in equation (93) and $\boldsymbol{K}_h$ given by

$$\bar{\boldsymbol{K}}_h = \left. \frac{\partial \bar{\boldsymbol{B}}^T \bar{\boldsymbol{T}}_{int}}{\partial \bar{\boldsymbol{\mathcal{D}}}_*} \right|_{\bar{\boldsymbol{T}}_{int}} = \begin{bmatrix} 0 & \boldsymbol{0}^T & \boldsymbol{0}^T \\ \boldsymbol{0} & \bar{\boldsymbol{K}}_{h1} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0} & \boldsymbol{0}_{3\times 3} & \bar{\boldsymbol{K}}_{h2} \end{bmatrix}; \quad \bar{\boldsymbol{K}}_{ha} = \left. \frac{\partial \boldsymbol{T}_s^{-T}(\bar{\boldsymbol{\Theta}}_a) \bar{\boldsymbol{M}}_a}{\partial \bar{\boldsymbol{W}}_a} \right|_{\bar{\boldsymbol{M}}_a}, \tag{C.11}$$

$$\begin{aligned} \bar{\boldsymbol{K}}_{ha} = {} & \eta_a \left(\bar{\boldsymbol{\Theta}}_a \bar{\boldsymbol{M}}_a^T - 2 \bar{\boldsymbol{M}}_a \bar{\boldsymbol{\Theta}}_a^T + (\bar{\boldsymbol{\Theta}}_a \cdot \bar{\boldsymbol{M}}_a) \boldsymbol{I}_{3\times 3} \right) \boldsymbol{T}_s^{-1}(\bar{\boldsymbol{\Theta}}_a) \\ & + \mu_i \left(\boldsymbol{S}(\bar{\boldsymbol{\Theta}}_a) \boldsymbol{S}(\bar{\boldsymbol{\Theta}}_a) \bar{\boldsymbol{M}}_a \bar{\boldsymbol{\Theta}}_a^T \right) \boldsymbol{T}_s^{-1}(\bar{\boldsymbol{\Theta}}_a) \\ & - \frac{1}{2} \boldsymbol{S}(\bar{\boldsymbol{M}}_a) \boldsymbol{T}_s^{-1}(\bar{\boldsymbol{\Theta}}_a) \end{aligned} \tag{C.12}$$

and

$$\eta_a = \frac{2 \sin \bar{\Theta}_a - \bar{\Theta}_a (1 + \cos \bar{\Theta}_a)}{2 (\bar{\Theta}_a)^2 \sin \bar{\Theta}_a}, \quad \mu_a = \frac{\bar{\Theta}_a (\bar{\Theta}_a + \sin \bar{\Theta}_a) - \sin^2(\bar{\Theta}_a/2)}{4 (\bar{\Theta}_a)^4 \sin^2(\bar{\Theta}_a/2)}. \tag{C.13}$$

Finally $\boldsymbol{K}_m$ is given by

$$\boldsymbol{K}_m = \boldsymbol{D} \bar{N} - \boldsymbol{E} \boldsymbol{Q} \boldsymbol{G}^T \boldsymbol{E}^T + \boldsymbol{E} \boldsymbol{G} \boldsymbol{a} \boldsymbol{r}, \tag{C.14}$$

with $\bar{N}$ being the first component of $\bar{\boldsymbol{T}}_{int}$ (see equation (91))), $\boldsymbol{r}$ given in equation (C.4) and $\boldsymbol{D}$, $\boldsymbol{Q}$, $\boldsymbol{a}$ given by

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_3 & \boldsymbol{0}_{3\times 3} & -\boldsymbol{D}_3 & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} \\ -\boldsymbol{D}_3 & \boldsymbol{0}_{3\times 3} & \boldsymbol{D}_3 & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} \end{bmatrix}; \quad \boldsymbol{D}_3 = \frac{1}{l_n} (\boldsymbol{I}_{3\times 3} - \boldsymbol{v}_1 \boldsymbol{v}_1^T), \tag{C.15}$$

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{S}(\boldsymbol{Q}_1) \\ \boldsymbol{S}(\boldsymbol{Q}_2) \\ \boldsymbol{S}(\boldsymbol{Q}_3) \\ \boldsymbol{S}(\boldsymbol{Q}_4) \end{bmatrix}, \quad \boldsymbol{a} = \begin{bmatrix} 0 \\ \eta([\bar{\boldsymbol{M}}_{*_1}]_1 + [\bar{\boldsymbol{M}}_{*_2}]_1)/l_n - ([\bar{\boldsymbol{M}}_{*_1}]_2 + [\bar{\boldsymbol{M}}_{*_2}]_2)/l_n \\ ([\bar{\boldsymbol{M}}_{*_1}]_3 + [\bar{\boldsymbol{M}}_{*_2}]_3)/l_n \end{bmatrix}. \tag{C.16}$$

In the above definition of $\boldsymbol{Q}$, $\boldsymbol{Q}_i$ are the block components of

$$\boldsymbol{P}^T\boldsymbol{m} = \left[\begin{array}{cccc} \boldsymbol{Q}_1^T & \boldsymbol{Q}_2^T & \boldsymbol{Q}_3^T & \boldsymbol{Q}_4^T \end{array}\right]; \quad \boldsymbol{m} = \left[\begin{array}{cc} \bar{\boldsymbol{M}}_{*1}^T & \bar{\boldsymbol{M}}_{*2}^T \end{array}\right], \tag{C.17}$$

with $\bar{\boldsymbol{M}}_{*a}^T$ being the nodal moments related to the displacements $\bar{\boldsymbol{\mathcal{D}}}_*$,

$$\bar{\boldsymbol{T}}_* = \frac{\partial W_{int}}{\partial \bar{\boldsymbol{\mathcal{D}}}_*} = \bar{\boldsymbol{B}}_* \bar{\boldsymbol{T}}_{int} \equiv \left[\begin{array}{ccc} \bar{N} & \bar{\boldsymbol{M}}_{*1}^T & \bar{\boldsymbol{M}}_{*2}^T \end{array}\right]^T. \tag{C.18}$$

Appendix D. Time derivatives and variations of the centerline displacements and rotations

In order to derive the dynamic force vector as well as the tangent dynamic matrices, the variations of the centerline displacements $\boldsymbol{u}_0^h$ and their time derivatives, as well as the variations of the centerline spin vectors $\boldsymbol{w}_0^h$ and their time derivatives are needed. For completness, the main results of [40] are summarized, but for more details the reader is refered to that paper. The variation of the centerline displacement, $\delta\boldsymbol{u}_0^h$ is obtained via equation (80),

$$\delta\boldsymbol{u}_0^h = \delta\boldsymbol{u}_1 N_1(\xi) + \delta\boldsymbol{u}_2 N_2(\xi) + \delta\boldsymbol{R}_e\boldsymbol{u}^t(\xi) + \boldsymbol{R}_e\delta\boldsymbol{u}^t(\xi). \tag{D.1}$$

The variation $\delta\boldsymbol{R}_e$ is obtained via equations (61), (64) as

$$\delta\boldsymbol{R}_e = \boldsymbol{R}_e\boldsymbol{S}(\delta\boldsymbol{W}_e). \tag{D.2}$$

with (see equations (70), (C.6)),

$$\delta\boldsymbol{W}_e = \frac{\partial \boldsymbol{W}_e}{\partial \boldsymbol{\mathcal{D}}}\delta\boldsymbol{\mathcal{D}} = \boldsymbol{G}^T\boldsymbol{E}^T\delta\boldsymbol{d}. \tag{D.3}$$

The variation of $\boldsymbol{u}^t$ is given by

$$\delta\boldsymbol{u}^t = \boldsymbol{P}_1 \left[\begin{array}{c} \delta\bar{\boldsymbol{\Theta}}_1 \\ \delta\bar{\boldsymbol{\Theta}}_2 \end{array}\right] \simeq \boldsymbol{P}_1 \left[\begin{array}{c} \delta\bar{\boldsymbol{W}}_1 \\ \delta\bar{\boldsymbol{W}}_2 \end{array}\right] = \boldsymbol{P}_1\boldsymbol{P}\boldsymbol{E}^T\delta\boldsymbol{d}, \tag{D.4}$$

where equation (C.5) has been used and where, as in [40], it has been assumed that $\boldsymbol{T}_s(\bar{\boldsymbol{\Theta}}_a) \simeq \boldsymbol{I}$ due to small rotations (see equation 14b). Replacing the above results into (D.1),

$$\delta\boldsymbol{u}_0^h = \boldsymbol{R}_e\boldsymbol{H}_1\boldsymbol{E}^T\delta\boldsymbol{d}, \tag{D.5}$$

with

$$\boldsymbol{H}_1(\xi) = \boldsymbol{N}(\xi) + \boldsymbol{P}_1(\xi)\boldsymbol{P} - \boldsymbol{S}(\boldsymbol{u}^t(\xi))\boldsymbol{G}^T. \tag{D.6}$$

with

$$\boldsymbol{N}(\xi) = \left[\begin{array}{cccc} N_1(\xi)\boldsymbol{I} & \boldsymbol{0} & N_2(\xi)\boldsymbol{I} & \boldsymbol{0} \end{array}\right]. \tag{D.7}$$

The variation of $\boldsymbol{H}_1$ is useful to compute second derivatives and also to obtain the tangent dynamic matrices. This is given by

$$\delta\boldsymbol{H}_1 = \frac{N_7}{l_n^2}\boldsymbol{A}_1\boldsymbol{r}\delta\boldsymbol{d} - \boldsymbol{S}(\delta\boldsymbol{u}^t)\boldsymbol{G}^T, \tag{D.8}$$

with

$$N_7 = N_3 + N_4, \tag{D.9}$$

$$\boldsymbol{A}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{D.10}$$

Via equation (D.5), the first and second derivatives of $\boldsymbol{u}_0^h$ can be computed as

$$\dot{\boldsymbol{u}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{D.11}$$

$$\ddot{\boldsymbol{u}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T \ddot{\boldsymbol{d}} + \boldsymbol{R}_e \boldsymbol{C}_1 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{D.12}$$

with

$$\boldsymbol{C}_1 = \boldsymbol{S}(\dot{\boldsymbol{W}}_e)\boldsymbol{H}_1 + \dot{\boldsymbol{H}}_1 - \boldsymbol{H}_1 \boldsymbol{E} \tag{D.13}$$

and $\dot{\boldsymbol{H}}_1$ obtained from equations (D.8), (D.4) as

$$\dot{\boldsymbol{H}}_1 = \frac{N_7}{l_n^2} \boldsymbol{A}_1 \boldsymbol{r} \dot{\boldsymbol{d}} - \boldsymbol{S}(\dot{\boldsymbol{u}}^t)\boldsymbol{G}^T; \quad \dot{\boldsymbol{u}}^t = \boldsymbol{P}_1 \boldsymbol{P} \boldsymbol{E}^T \dot{\boldsymbol{d}}. \tag{D.14}$$

Regarding the rotation variables, the varition of $\boldsymbol{w}_0^h$ is obtained fom equation (84) and doing a *rigid-push-forward* with $\boldsymbol{R}_e$

$$\delta \boldsymbol{w}_0^h = \boldsymbol{R}_e \boldsymbol{H}_2 \boldsymbol{E}^T \delta \boldsymbol{d}, \tag{D.15}$$

with

$$\boldsymbol{H}_2(s) = \boldsymbol{P}_2(s)\boldsymbol{P} + \boldsymbol{G}^T. \tag{D.16}$$

As for $\boldsymbol{H}_1$, it is useful to obtain the variation of $\boldsymbol{H}_2$. This is given by

$$\delta \boldsymbol{H}_2 = \frac{N_8}{l_n^2} \boldsymbol{A}_2 \boldsymbol{r} \delta \boldsymbol{d}, \tag{D.17}$$

with

$$N_8 = N_5 + N_6 - 1, \tag{D.18}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{D.19}$$

Via equation (D.15), the angular velocity and angular acceleration are obtained as

$$\dot{\boldsymbol{w}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_2 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{D.20}$$

$$\ddot{\boldsymbol{w}}_0^h = \boldsymbol{R}_e \boldsymbol{H}_2 \boldsymbol{E}^T \ddot{\boldsymbol{d}} + \boldsymbol{R}_e \boldsymbol{C}_2 \boldsymbol{E}^T \dot{\boldsymbol{d}}, \tag{D.21}$$

with

$$\boldsymbol{C}_2 = \boldsymbol{S}(\dot{\boldsymbol{W}}_e)\boldsymbol{H}_2 + \dot{\boldsymbol{H}}_2 - \boldsymbol{H}_2 \boldsymbol{E}. \tag{D.22}$$

In order to obtain the tangent matrix, the variation of $\boldsymbol{T}_k$ in equation (98) is needed. As in [40], this is computed considering only the contributions from the acceleration and velocity, this is, the centrifugal and dynamic matrices

$$\delta \boldsymbol{T}_k \simeq \boldsymbol{M} \delta \ddot{\boldsymbol{d}} + \boldsymbol{C}_k \delta \dot{\boldsymbol{d}} \tag{D.23}$$

The following variables require linearization

$$\tilde{\boldsymbol{U}}_0^h \equiv \boldsymbol{R}_e^T \ddot{\boldsymbol{u}}_0^h = \boldsymbol{H}_1(\boldsymbol{d})\boldsymbol{E}(\boldsymbol{d})^T \ddot{\boldsymbol{d}} + \boldsymbol{C}_1(\boldsymbol{d},\dot{\boldsymbol{d}})\boldsymbol{E}(\boldsymbol{d})^T \dot{\boldsymbol{d}}, \tag{D.24}$$

$$\tilde{\boldsymbol{W}}_0^h \equiv \boldsymbol{R}_e^T \ddot{\boldsymbol{w}}_0^h = \boldsymbol{H}_2(\boldsymbol{d})\boldsymbol{E}(\boldsymbol{d})^T \ddot{\boldsymbol{d}} + \boldsymbol{C}_2(\boldsymbol{d},\dot{\boldsymbol{d}})\boldsymbol{E}(\boldsymbol{d})^T \dot{\boldsymbol{d}}, \tag{D.25}$$

$$\dot{\boldsymbol{W}}_0^h \equiv \boldsymbol{R}_e^T \dot{\boldsymbol{w}}_0^h = \boldsymbol{H}_2(\boldsymbol{d})\boldsymbol{E}(\boldsymbol{d})^T \dot{\boldsymbol{d}}, \tag{D.26}$$

where equations (83), (86) and (87) have been used for the definition of with $\ddot{\boldsymbol{u}}_0$, $\dot{\boldsymbol{w}}_0$, $\ddot{\boldsymbol{w}}_0$. The variations of the above quantities are obtained by linearizing only respect to $\dot{\boldsymbol{d}}$ and $\ddot{\boldsymbol{d}}$ to give

$$\delta\tilde{\boldsymbol{U}}_0^h \simeq \boldsymbol{H}_1\boldsymbol{E}^T\delta\ddot{\boldsymbol{d}} + (\boldsymbol{C}_1 + \boldsymbol{C}_3)\boldsymbol{E}^T\delta\dot{\boldsymbol{d}}, \tag{D.27}$$

$$\delta\tilde{\boldsymbol{W}}_0^h \simeq \boldsymbol{H}_2\boldsymbol{E}^T\delta\ddot{\boldsymbol{d}} + (\boldsymbol{C}_2 + \boldsymbol{C}_4)\boldsymbol{E}^T\delta\dot{\boldsymbol{d}}, \tag{D.28}$$

$$\delta\dot{\boldsymbol{W}}_0^h \simeq \boldsymbol{H}_2(\boldsymbol{d})\boldsymbol{E}(\boldsymbol{d})^T\delta\dot{\boldsymbol{d}}. \tag{D.29}$$

In the above equations, $\boldsymbol{C}_3$ and $\boldsymbol{C}_4$ are defined as

$$\boldsymbol{C}_3 = -\boldsymbol{S}(\boldsymbol{h}_1)\boldsymbol{G}^T + \frac{N_7}{l_n^2}\boldsymbol{A}_1\dot{\boldsymbol{\mathcal{D}}}\boldsymbol{r}\boldsymbol{E} + \boldsymbol{S}(\dot{\boldsymbol{W}}_e)\boldsymbol{P}_1\boldsymbol{P} + \boldsymbol{H}_1\boldsymbol{F}_1\boldsymbol{G}^T, \tag{D.30}$$

$$\boldsymbol{C}_4 = -\boldsymbol{S}(\boldsymbol{h}_2)\boldsymbol{G}^T + \frac{N_8}{l_n^2}\boldsymbol{A}_2\dot{\boldsymbol{\mathcal{D}}}\boldsymbol{r}\boldsymbol{E} + \boldsymbol{H}_2\boldsymbol{F}_1\boldsymbol{G}^T, \tag{D.31}$$

with $\dot{\boldsymbol{\mathcal{D}}}$ taken from equation (70) as

$$\dot{\boldsymbol{\mathcal{D}}} = \boldsymbol{E}^T\dot{\boldsymbol{d}} = \begin{bmatrix} \dot{\boldsymbol{U}}_1^T & \dot{\boldsymbol{W}}_1^T & \dot{\boldsymbol{U}}_2^T & \dot{\boldsymbol{W}}_2^T \end{bmatrix}^T, \tag{D.32}$$

$\boldsymbol{h}_1$, $\boldsymbol{h}_2$ defined as

$$\boldsymbol{h}_1 = \boldsymbol{H}_1\dot{\boldsymbol{\mathcal{D}}}, \tag{D.33}$$

$$\boldsymbol{h}_2 = \boldsymbol{H}_2\dot{\boldsymbol{\mathcal{D}}}, \tag{D.34}$$

and, finally, $\boldsymbol{F}_1$ given by

$$\boldsymbol{F}_1 = \begin{bmatrix} \boldsymbol{S}(\dot{\boldsymbol{U}}_1) \\ \boldsymbol{S}(\dot{\boldsymbol{W}}_1) \\ \boldsymbol{S}(\dot{\boldsymbol{U}}_2) \\ \boldsymbol{S}(\dot{\boldsymbol{W}}_2) \end{bmatrix}. \tag{D.35}$$

Appendix E. Linearization of the contact force vector

This appendix explains in detail the linearization of the discrete contact force vector, which leads to the different terms desribed in equations (116), (117), (118) and (121). To start with, the contact force vector defined in equation (113) is repeated here for convenience,

$$\boldsymbol{T}_c = \int_0^{l_0} \boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{R}_e^T\boldsymbol{f}_c\, ds, \tag{E.1}$$

with $\boldsymbol{f}_c$ given by

$$\boldsymbol{f}_c = p_N(\boldsymbol{d})\boldsymbol{\mathcal{G}}(\boldsymbol{d};\dot{\boldsymbol{d}}); \quad \boldsymbol{\mathcal{G}}(\boldsymbol{x};\boldsymbol{d};\dot{\boldsymbol{d}}) = \frac{\partial g_N}{\partial \boldsymbol{x}} - \mu\frac{\dot{\boldsymbol{g}}_T^h}{\sqrt{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon}}. \tag{E.2}$$

$$\delta\boldsymbol{T}_c = \boldsymbol{K}_c\delta\boldsymbol{d} + \boldsymbol{C}_c\delta\dot{\boldsymbol{d}}; \quad \boldsymbol{K}_c = \boldsymbol{K}_c^1 + \boldsymbol{K}_c^2 + \boldsymbol{K}_c^3 + \boldsymbol{K}_c^4. \tag{E.3}$$

and, in order to facilitate their derivation, the following abbreviations are used (see for example [47])

$$\boldsymbol{K}_c^1 \delta \boldsymbol{d} = \int_0^{l_0} \boldsymbol{t}_c^1, ds, \tag{E.4a}$$

$$\boldsymbol{K}_c^2 \delta \boldsymbol{d} = \int_0^{l_0} \boldsymbol{t}_c^2 \, ds, \tag{E.4b}$$

$$\boldsymbol{K}_c^3 \delta \boldsymbol{d} = \int_0^{l_0} \boldsymbol{t}_c^3 \, ds, \tag{E.4c}$$

$$\boldsymbol{K}_c^4 \delta \boldsymbol{d} + \boldsymbol{C}_c \delta \dot{\boldsymbol{d}} = \int_0^{l_0} \boldsymbol{t}_c^4 \, ds, \tag{E.4d}$$

with

$$\boldsymbol{t}_c^1 = \delta \boldsymbol{E} \boldsymbol{H}_1^T \boldsymbol{R}_e^T \boldsymbol{f}_c, \tag{E.5a}$$

$$\boldsymbol{t}_c^2 = \boldsymbol{E} (\delta \boldsymbol{H}_1)^T \boldsymbol{R}_e^T \boldsymbol{f}_c, \tag{E.5b}$$

$$\boldsymbol{t}_c^3 = \boldsymbol{E} \boldsymbol{H}_1^T (\delta \boldsymbol{R}_e)^T \boldsymbol{f}_c, \tag{E.5c}$$

$$\boldsymbol{t}_c^4 = \boldsymbol{E} \boldsymbol{H}_1^T \boldsymbol{R}_e^T \delta \boldsymbol{f}_c. \tag{E.5d}$$

Before moving forward with the above linearizations, it is useful to provide some preliminary results. Firstly, $\boldsymbol{f}_c$ is rigidly rotated back to give

$$\boldsymbol{\mathcal{F}}_c = \boldsymbol{R}_e^T \boldsymbol{f}_c. \tag{E.6}$$

At the same time the operator $\hat{\boldsymbol{S}}(\cdot)$ is defined, which transforms a 12×1 array $\boldsymbol{a}$ into a 12×3 matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = \hat{\boldsymbol{S}}(\boldsymbol{a}) = \begin{bmatrix} \boldsymbol{S}(\boldsymbol{a}_1) \\ \boldsymbol{S}(\boldsymbol{a}_2) \\ \boldsymbol{S}(\boldsymbol{a}_3) \\ \boldsymbol{S}(\boldsymbol{a}_4) \end{bmatrix}; \quad \boldsymbol{a}_I = \begin{bmatrix} a_{3(I-1)+1} \\ a_{3(I-1)+2} \\ a_{3(I-1)+3} \end{bmatrix}; \quad I \in \{1,2,3,4\}, \tag{E.7}$$

with the operator $\boldsymbol{S}$ as defined in equation (4).

In order to obtain $\boldsymbol{t}_c^1$, the variation $\delta \boldsymbol{E}$ is needed. This can be obtained from its definition (70) and equation (D.2), to give

$$\delta \boldsymbol{E} = \boldsymbol{E} \begin{bmatrix} \boldsymbol{S}(\delta \boldsymbol{W}_e) & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}(\delta \boldsymbol{W}_e) & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{S}(\delta \boldsymbol{W}_e) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{S}(\delta \boldsymbol{W}_e) \end{bmatrix}. \tag{E.8}$$

And, in order to obtain $\boldsymbol{t}_c^r$, the variation of $\delta \boldsymbol{f}_c$ is needed. From equation (E.2) this can be obtained as

$$\delta \boldsymbol{f}_c = \delta p_N \boldsymbol{\mathcal{G}} + p_N \delta \boldsymbol{\mathcal{G}}, \tag{E.9}$$

which in turn requires the variations δp_N and $\delta \boldsymbol{\mathcal{G}}$. The variation of p_N depends only on $\delta \boldsymbol{d}$ and is given as

$$\delta p_N = p_N'(g_N) \delta g_N = p_N'(g_N) \frac{\partial g_N}{\partial \boldsymbol{x}} \cdot \left(\boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T \delta \boldsymbol{d} \right), \tag{E.10}$$

with p'_N obtained from (111)

$$p'_N = p'_N(g_N) = \begin{cases} -\varepsilon_c, & g \le 0 \\ 2\dfrac{\varepsilon_c \bar{g}_N - \bar{p}_N}{\bar{g}_N^2} g_N - \varepsilon_c, & 0 < g_N \le \bar{g}_N \\ 0. & g > \bar{g}_N \end{cases} \tag{E.11}$$

The variation of $\delta\boldsymbol{\mathcal{G}}$ has contributions in $\delta\boldsymbol{d}$ and $\delta\dot{\boldsymbol{d}}$. This can be obtained from (E.2) as

$$\begin{aligned} \delta\boldsymbol{\mathcal{G}} &= \delta\left(\frac{\partial g_N}{\partial \boldsymbol{x}} - \mu \frac{\dot{\boldsymbol{g}}_T^h}{\sqrt{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon}}\right) \\ &= \frac{\partial^2 g_N}{\partial \boldsymbol{x} \partial \boldsymbol{x}} \boldsymbol{R}_e \boldsymbol{H}_1(s) \boldsymbol{E}^T \delta\boldsymbol{d} \\ &- \frac{\mu}{\sqrt{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon}} \left(\boldsymbol{I} - \frac{1}{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon} (\dot{\boldsymbol{g}}_T^h \otimes \dot{\boldsymbol{g}}_T^h)\right) \delta\dot{\boldsymbol{g}}_T^h. \end{aligned} \tag{E.12}$$

where equation (D.5) has been used. The above result requires the variation $\delta\dot{\boldsymbol{g}}_T^h$, which is given by

$$\delta\dot{\boldsymbol{g}}_T^h = \boldsymbol{\mathcal{A}}\delta\boldsymbol{d} + \boldsymbol{\mathcal{B}}\delta\dot{\boldsymbol{d}}; \quad \boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}}^1 + \boldsymbol{\mathcal{A}}^2 + \boldsymbol{\mathcal{A}}^3 + \boldsymbol{\mathcal{A}}^4, \tag{E.13}$$

with

$$\begin{aligned} \boldsymbol{\mathcal{A}}^1 &= -\left(\frac{\partial g_N}{\partial \boldsymbol{x}} \cdot (\boldsymbol{R}_e \boldsymbol{H}_1 \dot{\boldsymbol{\mathcal{D}}})\right) \frac{\partial^2 g_N}{\partial \boldsymbol{x} \partial \boldsymbol{x}} \boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T \\ &\quad - \frac{\partial g_N}{\partial \boldsymbol{x}} \otimes \left(\boldsymbol{E}\boldsymbol{H}_1^T \boldsymbol{R}_e^T \frac{\partial^2 g_N}{\partial \boldsymbol{x} \partial \boldsymbol{x}} \boldsymbol{R}_e \boldsymbol{H}_1 \dot{\boldsymbol{\mathcal{D}}}\right), \end{aligned} \tag{E.14a}$$

$$\boldsymbol{\mathcal{A}}^2 = -\left(\boldsymbol{I} - \frac{\partial g_N}{\partial \boldsymbol{x}} \otimes \frac{\partial g_N}{\partial \boldsymbol{x}}\right) \boldsymbol{R}_e \boldsymbol{S}(\boldsymbol{H}_1 \dot{\boldsymbol{\mathcal{D}}}) \boldsymbol{G}^T \boldsymbol{E}^T, \tag{E.14b}$$

$$\boldsymbol{\mathcal{A}}^3 = \left(\boldsymbol{I} - \frac{\partial g_N}{\partial \boldsymbol{x}} \otimes \frac{\partial g_N}{\partial \boldsymbol{x}}\right) \left(\frac{N_7}{l_n^2} \boldsymbol{R}_e \boldsymbol{A}_1 \dot{\boldsymbol{\mathcal{D}}} \boldsymbol{r} + \boldsymbol{R}_e \boldsymbol{S}(\boldsymbol{G}^T \dot{\boldsymbol{\mathcal{D}}}) \boldsymbol{P}_1 \boldsymbol{P} \boldsymbol{E}^T\right), \tag{E.14c}$$

$$\boldsymbol{\mathcal{A}}^4 = \left(\boldsymbol{I} - \frac{\partial g_N}{\partial \boldsymbol{x}} \otimes \frac{\partial g_N}{\partial \boldsymbol{x}}\right) \boldsymbol{R}_e \boldsymbol{H}_1 \hat{\boldsymbol{S}}(\dot{\boldsymbol{\mathcal{D}}}) \boldsymbol{G}^T \boldsymbol{E}^T, \tag{E.14d}$$

$$\boldsymbol{\mathcal{B}} = \left(\boldsymbol{I} - \frac{\partial g_N}{\partial \boldsymbol{x}} \otimes \frac{\partial g_N}{\partial \boldsymbol{x}}\right) \boldsymbol{R}_e \boldsymbol{H}_1 \boldsymbol{E}^T. \tag{E.14e}$$

Using (E.10), (E.12), (E.13) into (E.9) yields

$$\delta\boldsymbol{f}_c = \boldsymbol{K}_{\boldsymbol{f}_c} \delta\boldsymbol{d} + \boldsymbol{C}_{\boldsymbol{f}_c} \delta\dot{\boldsymbol{d}}, \tag{E.15}$$

with

$$\begin{aligned} \boldsymbol{K}_{\boldsymbol{f}_c} &= p'_N(g_N) \left(\boldsymbol{\mathcal{G}} \otimes (\boldsymbol{E}\boldsymbol{H}_1^T \boldsymbol{R}_e^T \frac{\partial g_N}{\partial \boldsymbol{x}})\right) \\ &\quad + p_N \left(\frac{\partial^2 g_N}{\partial \boldsymbol{x} \partial \boldsymbol{x}} \boldsymbol{R}_e \boldsymbol{H}_1(s) \boldsymbol{E}^T - \frac{\mu}{\sqrt{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon}} \left(\boldsymbol{I} - \frac{1}{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon} (\dot{\boldsymbol{g}}_T^h \otimes \dot{\boldsymbol{g}}_T^h)\right) \boldsymbol{\mathcal{A}}\right), \end{aligned} \tag{E.16}$$

$$\boldsymbol{C}_{\boldsymbol{f}_c} = -p_N \frac{\mu}{\sqrt{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon}} \left(\boldsymbol{I} - \frac{1}{\|\dot{\boldsymbol{g}}_T^h\|^2 + \varepsilon} (\dot{\boldsymbol{g}}_T^h \otimes \dot{\boldsymbol{g}}_T^h)\right) \boldsymbol{\mathcal{B}}. \tag{E.17}$$

Using (E.6), (E.7), (E.8), (E.15) alongside (D.3), (D.4), (D.8) into (E.5) gives

$$\begin{aligned}
\boldsymbol{t}_c^1 &= \delta\boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c = \boldsymbol{E}\begin{bmatrix} \boldsymbol{S}(\delta\boldsymbol{W}_e)\left[\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c\right]_{1:3} \\ \boldsymbol{S}(\delta\boldsymbol{W}_e)\left[\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c\right]_{4:6} \\ \boldsymbol{S}(\delta\boldsymbol{W}_e)\left[\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c\right]_{7:9} \\ \boldsymbol{S}(\delta\boldsymbol{W}_e)\left[\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c\right]_{10:12} \end{bmatrix} \\
&= -\boldsymbol{E}\hat{\boldsymbol{S}}(\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c)\delta\boldsymbol{W}_e = -\boldsymbol{E}\hat{\boldsymbol{S}}(\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c)\boldsymbol{G}^T\boldsymbol{E}^T\delta\boldsymbol{d}, \qquad \text{(E.18a)} \\
\boldsymbol{t}_c^2 &= \boldsymbol{E}\delta\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c = \boldsymbol{E}\left(\frac{N_7}{l_n^2}\boldsymbol{A}_1\boldsymbol{r}\delta\boldsymbol{d} - \boldsymbol{S}(\delta\boldsymbol{u}_l^t)\boldsymbol{G}^T\right)^T\boldsymbol{\mathcal{F}}_c \\
&= \frac{N_7}{l_n^2}\boldsymbol{E}\boldsymbol{A}_1^T\boldsymbol{\mathcal{F}}_c\left(\boldsymbol{r}\delta\boldsymbol{d}\right) + \boldsymbol{E}\boldsymbol{G}\boldsymbol{S}(\delta\boldsymbol{u}^t)\boldsymbol{\mathcal{F}}_c \\
&= \frac{N_7}{l_n^2}\boldsymbol{E}\boldsymbol{A}_1^T\boldsymbol{\mathcal{F}}_c\boldsymbol{r}\delta\boldsymbol{d} - \boldsymbol{E}\boldsymbol{G}\boldsymbol{S}(\boldsymbol{\mathcal{F}}_c)\boldsymbol{P}_1\boldsymbol{P}\boldsymbol{E}^T\delta\boldsymbol{d}, \qquad \text{(E.18b)} \\
\boldsymbol{t}_c^3 &= \boldsymbol{E}\boldsymbol{H}_1^T(\delta\boldsymbol{R}_e)^T\boldsymbol{f}_c = \boldsymbol{E}\boldsymbol{H}_1^T\left(\boldsymbol{R}_e\boldsymbol{S}(\delta\boldsymbol{W}_e)\right)^T\boldsymbol{f}_c = -\boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{S}(\delta\boldsymbol{W}_e)\boldsymbol{\mathcal{F}}_c \\
&= \boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{S}(\boldsymbol{\mathcal{F}}_c)\delta\boldsymbol{W}_e = \boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{S}(\boldsymbol{\mathcal{F}}_c)\boldsymbol{G}^T\boldsymbol{E}^T\delta\boldsymbol{d}, \qquad \text{(E.18c)} \\
\boldsymbol{t}_c^4 &= \boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{R}_e^T\delta\boldsymbol{f}_c = \boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{R}_e^T\left(\boldsymbol{K}_{\boldsymbol{f}_c}\delta\boldsymbol{d} + \boldsymbol{C}_{\boldsymbol{f}_c}\delta\dot{\boldsymbol{d}}\right). \qquad \text{(E.18d)}
\end{aligned}$$

Finally, using the results in (E.18) into (E.4), the different tangent terms in (E.3) are given by

$$\boldsymbol{K}_c^1 = -\int_0^{l_0}\boldsymbol{E}\hat{\boldsymbol{S}}(\boldsymbol{H}_1^T\boldsymbol{\mathcal{F}}_c)\boldsymbol{G}^T\boldsymbol{E}^T\,ds, \qquad \text{(E.19)}$$

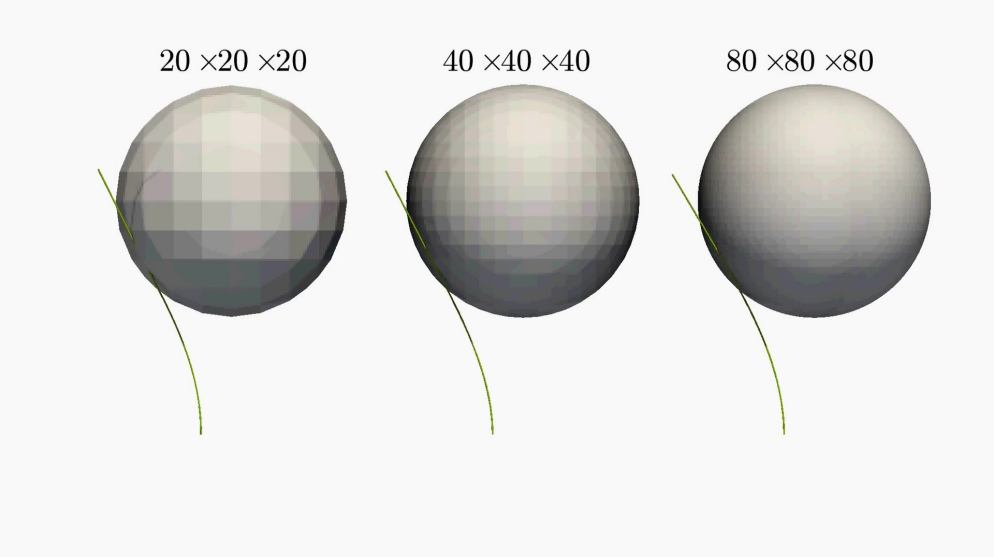
$$\boldsymbol{K}_c^2 = \int_0^{l_0}\left(\frac{N_7}{l_n^2}\boldsymbol{E}\boldsymbol{A}_1^T\boldsymbol{\mathcal{F}}_c\boldsymbol{r} - \boldsymbol{E}\boldsymbol{G}\boldsymbol{S}(\boldsymbol{\mathcal{F}}_c)\boldsymbol{P}_1\boldsymbol{P}\boldsymbol{E}^T\right)ds, \qquad \text{(E.20)}$$

$$\boldsymbol{K}_c^3 = \int_0^{l_0}\boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{S}(\boldsymbol{\mathcal{F}}_c)\boldsymbol{G}^T\boldsymbol{E}^T\,ds, \qquad \text{(E.21)}$$

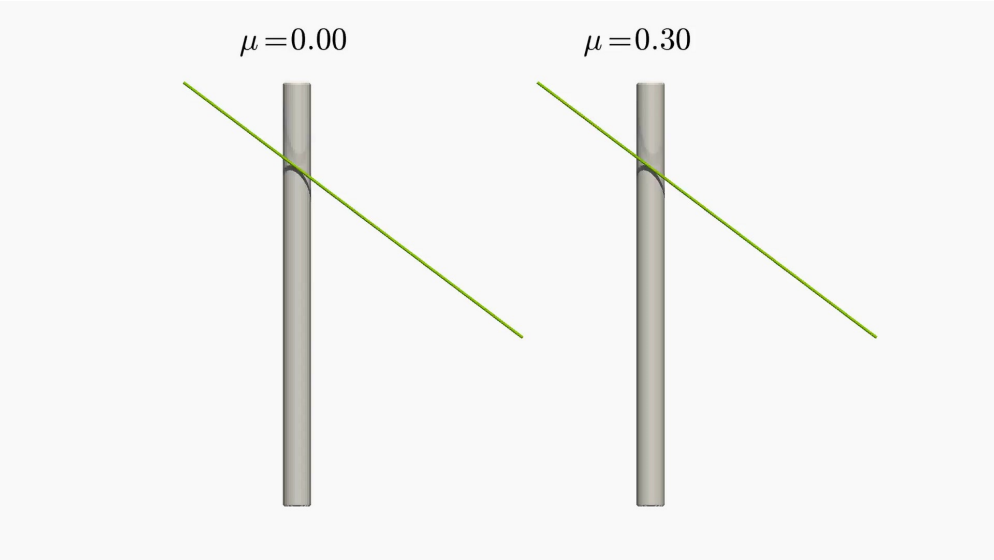
$$\boldsymbol{K}_c^4 = \int_0^{l_0}\boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{R}_e^T\boldsymbol{K}_{\boldsymbol{f}_c}\,ds, \qquad \text{(E.22)}$$

$$\boldsymbol{C}_c = \int_0^{l_0}\boldsymbol{E}\boldsymbol{H}_1^T\boldsymbol{R}_e^T\boldsymbol{C}_{\boldsymbol{f}_c}\,ds. \qquad \text{(E.23)}$$

Appendix F. Supplementary data



Video 1: Beam against sphere. Comparison of the solutions obtained using 3 different voxel mesh discretizations.



Video 2: Contact between a cable and a rigid cylinder. Comparison of the solutions obtained using $\mu = 0.00$ and $\mu = 0.30$.

References

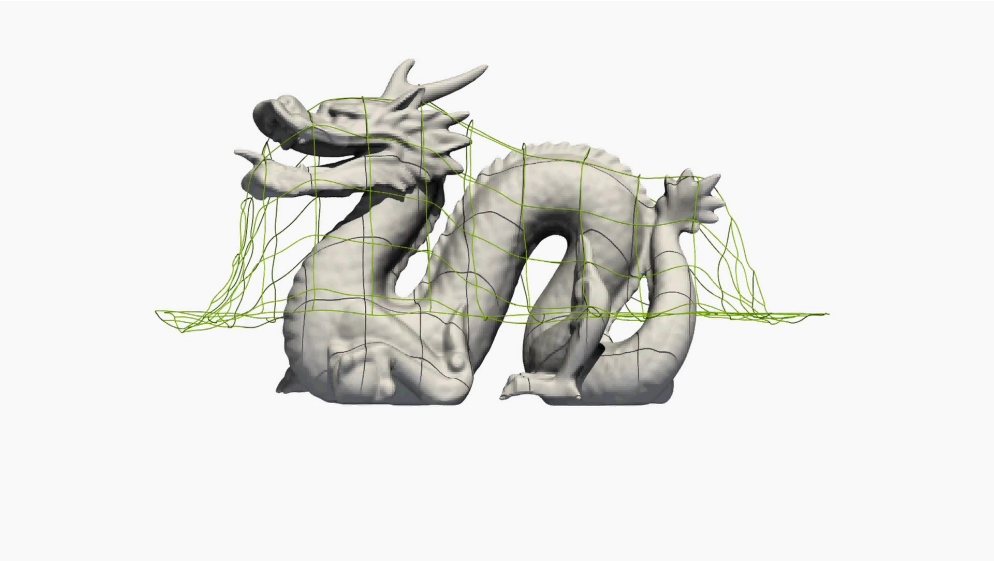
[1] D. J. White, M. F. Randolph, Seabed characterisation and models for pipeline-soil interaction, in: Proceedings of the International Offshore and Polar Engineering Conference, International Society of Offshore and Polar Engineers, 2007, pp. 758–769.

[2] H. E. Goicoechea, F. S. Buezas, M. B. Rosales, A non-linear Cosserat rod model for drill-string dynamics in arbitrary borehole geometries with contact and friction, International Journal of Mechanical Sciences 157-158 (2019) 98–110. doi:10.1016/j.ijmecsci.2019.04.023.

[3] J. E. De Cursi, Stress unilateral analysis of mooring cables, International Journal for Numerical Methods in Engineering 34 (1) (1992) 279–302. doi:10.1002/nme.1620340118.

[4] H. W. Lee, M. I. Roh, S. H. Ham, N. K. Ku, Coupled analysis method of a mooring system and a floating crane based on flexible multibody dynamics considering contact with the seabed, Ocean Engineering 163 (2018) 555–569. doi:10.1016/j.oceaneng.2018.06.025.

[5] A. G. Neto, P. M. Pimenta, P. Wriggers, Contact between rolling beams and flat surfaces, International Journal for Numerical Methods in Engineering 97 (9) (2014) 683–706. doi:



Video 3: Dropping net on the Stanford Dragon.

10.1002/nme.4611.

[6] A. G. Neto, C. A. Martins, P. M. Pimenta, Static analysis of offshore risers with a geometrically-exact 3D beam model subjected to unilateral contact, Computational Mechanics 53 (1) (2014) 125–145. doi:10.1007/s00466-013-0897-9.

[7] B. Tabarrok, C. M. Leech, Y. I. Kim, On the dynamics of an axially moving beam, Journal of the Franklin Institute 297 (3) (1974) 201–220. doi:10.1016/0016-0032(74)90104-5.

[8] T. M. Wasfy, M. Leamy, Effect of bending stiffness on the dynamic and steady-state responses of belt-drives, in: Proceedings of the ASME Design Engineering Technical Conference, Vol. 5 A, 2002, pp. 217–224. doi:10.1115/detc2002/mech-34223.

[9] Y. Dumont, L. Paoli, Dynamic contact of a beam against rigid obstacles: Convergence of a velocity-based approximation and numerical results, Nonlinear Analysis: Real World Applications 22 (2015) 520–536. doi:10.1016/j.nonrwa.2014.09.009.

[10] S. B. Solomon, T. Dickfeld, H. Calkins, Real-time cardiac catheter navigation on three-dimensional CT images, Journal of Interventional Cardiac Electrophysiology 8 (1) (2003) 27–36. doi:10.1023/A:1022379612437.

[11] Y. Wei, S. Cotin, J. Dequidt, C. Duriez, J. Allard, E. Kerrie, A (Near) Real-Time Simulation Method of Aneurysm Coil Embolization, in: Aneurysm, InTech, 2012, pp. 223–248. doi:10.5772/48635.

[12] D. Perrin, P. Badel, L. Orgéas, C. Geindreau, A. Dumenil, J. N. Albertini, S. Avril, Patient-specific numerical simulation of stent-graft deployment: Validation on three clinical cases, Journal of Biomechanics 48 (10) (2015) 1868–1875. doi:10.1016/j.jbiomech.2015.04.031.

[13] E. Kerrien, A. Yureidini, J. Dequidt, C. Duriez, R. Anxionnat, S. Cotin, Blood vessel modeling for interactive simulation of interventional neuroradiology procedures, Medical Image Analysis 35 (2017) 685–698. doi:10.1016/j.media.2016.10.003.

[14] I. Temizer, P. Wriggers, T. J. Hughes, Contact treatment in isogeometric analysis with NURBS, Computer Methods in Applied Mechanics and Engineering 200 (9-12) (2011) 1100–1112. doi:10.1016/j.cma.2010.11.020.

[15] A. Konyukhov, K. Schweizerhof, On some aspects for contact with rigid surfaces: Surface-to-rigid surface and curves-to-rigid surface algorithms, Computer Methods in Applied Mechanics and Engineering 283 (2014) 74–105. doi:10.1016/j.cma.2014.08.013.

[16] P. Wriggers, Computational contact mechanics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. doi:10.1007/978-3-540-32609-0.

[17] Y. Dumont, L. Paoli, Numerical simulation of a model of vibrations with joint clearance,

International Journal of Computer Applications in Technology 33 (1) (2008) 41. doi:10.1504/IJCAT.2008.021884.
[18] A. Liakou, V. Denoël, E. Detournay, Fast In-Plane Dynamics of a Beam with Unilateral Constraints, Journal of Engineering Mechanics 143 (2) (2017) 04016116. doi:10.1061/(ASCE)EM.1943-7889.0001175.
[19] C. Basdogan, C. H. Ho, M. A. Srinivasan, Virtual environments for medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration, IEEE/ASME Transactions on Mechatronics 6 (3) (2001) 269–285. doi:10.1109/3516.951365.
[20] Y. Wang, S. Guo, T. Tamiya, H. Hirata, H. Ishihara, X. Yin, A virtual-reality simulator and force sensation combined catheter operation training system and its preliminary evaluation, The International Journal of Medical Robotics and Computer Assisted Surgery 13 (3) (2017) e1769. doi:10.1002/rcs.1769.
[21] W. Tang, T. R. Wan, D. A. Gould, T. How, N. W. John, A stable and real-time nonlinear elastic approach to simulating guidewire and catheter insertions based on cosserat rod, IEEE Transactions on Biomedical Engineering 59 (8) (2012) 2211–2218. doi:10.1109/TBME.2012.2199319.
[22] H. Wang, J. Wu, M. Wei, X. Ma, A robust and fast approach to simulating the behavior of guidewire in vascular interventional radiology, Computerized Medical Imaging and Graphics 40 (2015) 160–169. doi:10.1016/j.compmedimag.2014.10.006.
[23] L. Vu-Quoc, S. Li, Dynamics of sliding geometrically-exact beams: large angle maneuver and parametric resonance, Computer Methods in Applied Mechanics and Engineering 120 (1-2) (1995) 65–118. doi:10.1016/0045-7825(94)00051-N.
[24] A. Humer, Dynamic modeling of beams with non-material, deformation-dependent boundary conditions, Journal of Sound and Vibration 332 (3) (2013) 622–641. doi:10.1016/j.jsv.2012.08.026.
[25] I. Steinbrecher, A. Humer, L. Vu-Quoc, On the numerical modeling of sliding beams: A comparison of different approaches, Journal of Sound and Vibration 408 (2017) 270–290. doi:10.1016/j.jsv.2017.07.010.
[26] S. Li, J. Guo, Q. Wang, Q. Meng, Y. P. Chui, J. Qin, P. A. Heng, A catheterization-training simulator based on a fast multigrid solver, IEEE Computer Graphics and Applications 32 (6) (2012) 56–70. doi:10.1109/MCG.2012.32.
[27] T. Belytschko, W. J. Daniel, G. Ventura, A monolithic smoothing-gap algorithm for contact-impact based on the signed distance function, International Journal for Numerical Methods in Engineering 55 (1) (2002) 101–125. doi:10.1002/nme.568.
[28] J. Bruchon, H. Digonnet, T. Coupez, Using a signed distance function for the simulation of metal forming processes: Formulation of the contact condition and mesh adaptation. From a Lagrangian approach to an Eulerian approach, International Journal for Numerical Methods in Engineering 78 (8) (2009) 980–1008. arXiv:1010.1724, doi:10.1002/nme.2519.
[29] G. L. Jones, S. Sueda, J. Litven, D. K. Pai, D. I. W. Levin, Eulerian solid simulation with contact, ACM Transactions on Graphics 30 (4) (2011) 1. doi:10.1145/2010324.1964931.
[30] S. Wolff, C. Bucher, Distance fields on unstructured grids: Stable interpolation, assumed gradients, collision detection and gap function, Computer Methods in Applied Mechanics and Engineering 259 (100) (2013) 17–25. doi:10.1016/j.cma.2013.02.015.
[31] S.-W. Chi, C.-H. Lee, J.-S. Chen, P.-C. Guan, A level set enhanced natural kernel contact algorithm for impact and penetration modeling, International Journal for Numerical Methods in Engineering 102 (3-4) (2015) 839–866. doi:10.1002/nme.4728.
[32] A. Leichner, H. Andrä, B. Simeon, A contact algorithm for voxel-based meshes using an implicit boundary representation, Computer Methods in Applied Mechanics and Engineering 352 (2019) 276–299. doi:10.1016/j.cma.2019.04.008.
[33] C. C. Rankin, F. A. Brogan, An element independent corotational procedure for the treatment of large rotations, Journal of Pressure Vessel Technology, Transactions of the ASME 108 (2)

(1986) 165–174. doi:10.1115/1.3264765.
[34] C. C. Rankin, B. Nour-Omid, The use of projectors to improve finite element performance, Computers and Structures 30 (1-2) (1988) 257–267. doi:10.1016/0045-7949(88)90231-3.
[35] M. Crisfield, A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements, Computer Methods in Applied Mechanics and Engineering 81 (2) (1990) 131–150. doi:10.1016/0045-7825(90)90106-V.
[36] C. Pacoste, A. Eriksson, Element behavior in post-critical plane frame analysis, Computer Methods in Applied Mechanics and Engineering 125 (1-4) (1995) 319–343. doi:10.1016/0045-7825(95)00813-G.
[37] C. Pacoste, A. Eriksson, Beam elements in instability problems, Computer Methods in Applied Mechanics and Engineering 144 (1-2) (1997) 163–197. doi:10.1016/S0045-7825(96)01165-6.
[38] J.-M. Battini, C. Pacoste, Co-rotational beam elements with warping effects in instability problems, Computer Methods in Applied Mechanics and Engineering 191 (17-18) (2002) 1755–1789. doi:10.1016/S0045-7825(01)00352-8.
[39] C. Felippa, B. Haugen, A unified formulation of small-strain corotational finite elements: I. Theory, Computer Methods in Applied Mechanics and Engineering 194 (21-24) (2005) 2285–2335. doi:10.1016/J.CMA.2004.07.035.
[40] T.-N. Le, J.-M. Battini, M. Hjiaj, Corotational formulation for nonlinear dynamics of beams with arbitrary thin-walled open cross-sections, Computers & Structures 134 (2014) 112–127. doi:10.1016/J.COMPSTRUC.2013.11.005.
[41] D. Marinkovic, M. Zehn, Survey of finite element method-based real-time simulations (jul 2019). doi:10.3390/app9142775.
[42] J. C. Simo, A finite strain beam formulation. The three-dimensional dynamic problem. Part I, Computer Methods in Applied Mechanics and Engineering 49 (1) (1985) 55–70. doi:10.1016/0045-7825(85)90050-7.
[43] J. C. Simo, L. Vu-Quoc, A three-dimensional finite-strain rod model. part II: Computational aspects, Computer Methods in Applied Mechanics and Engineering 58 (1) (1986) 79–116. doi:10.1016/0045-7825(86)90079-4.
[44] H. Cho, H. S. Joo, S. J. Shin, H. Kim, Elastoplastic and contact analysis based on consistent dynamic formulation of co-rotational planar elements, International Journal of Solids and Structures 121 (2017) 103–116. doi:10.1016/j.ijsolstr.2017.05.019.
[45] C. Meier, A. Popp, W. A. Wall, A finite element approach for the line-to-line contact interaction of thin beams with arbitrary orientation, Computer Methods in Applied Mechanics and Engineering 308 (2016) 377–413. doi:10.1016/J.CMA.2016.05.012.
[46] C. Meier, A. Popp, W. A. Wall, Geometrically Exact Finite Element Formulations for Slender Beams: Kirchhoff–Love Theory Versus Simo–Reissner Theory, Archives of Computational Methods in Engineering 26 (1) (2019) 163–243. doi:10.1007/s11831-017-9232-5.
[47] C. Meier, W. A. Wall, A. Popp, A unified approach for beam-to-beam contact, Computer Methods in Applied Mechanics and Engineering 315 (2017) 972–1010. doi:10.1016/j.cma.2016.11.028.
[48] C. Meier, M. J. Grill, W. A. Wall, A. Popp, Geometrically exact beam elements and smooth contact schemes for the modeling of fiber-based materials and structures, International Journal of Solids and Structuresdoi:10.1016/J.IJSOLSTR.2017.07.020.
[49] D. Durville, Contact-friction modeling within elastic beam assemblies: an application to knot tightening, Computational Mechanics 49 (2012) 687–707. doi:10.1007/s00466-012-0683-0.
[50] T. N. Le, J. M. Battini, M. Hjiaj, A consistent 3D corotational beam element for nonlinear dynamic analysis of flexible structures, Computer Methods in Applied Mechanics and Engineering 269 (2014) 538–565. doi:10.1016/j.cma.2013.11.007.
[51] F. Auricchio, P. Carotenuto, A. Reali, On the geometrically exact beam model: A consistent, effective and simple derivation from three-dimensional finite-elasticity, International Journal

of Solids and Structures 45 (17) (2008) 4766–4781. doi:10.1016/J.IJSOLSTR.2008.04.015.
[52] D. Sulsky, Z. Chen, H. L. Schreyer, A particle method for history-dependent materials, Computer Methods in Applied Mechanics and Engineering 118 (1-2) (1994) 179–196. doi:10.1016/0045-7825(94)90112-0.
[53] A. R. York, D. Sulsky, H. L. Schreyer, Fluid-membrane interaction based on the material point method, International Journal for Numerical Methods in Engineering 48 (6) (2000) 901–924. doi:10.1002/(SICI)1097-0207(20000630)48:6<901::AID-NME910>3.0.CO;2-T.
[54] X. Wang, W. K. Liu, Extended immersed boundary method using FEM and RKPM, Computer Methods in Applied Mechanics and Engineering 193 (12-14) (2004) 1305–1321. doi:10.1016/j.cma.2003.12.024.
[55] W. K. Liu, Y. Liu, D. Farrell, L. Zhang, X. S. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, J. Lee, J. Hong, X. Chen, H. Hsu, Immersed finite element method and its applications to biological systems, Computer Methods in Applied Mechanics and Engineering 195 (13-16) (2006) 1722–1749. doi:10.1016/j.cma.2005.05.049.
[56] H. Wang, J. Chessa, W. K. Liu, T. Belytschko, The immersed/fictitious element method for fluid-structure interaction: Volumetric consistency, compressibility and thin members, International Journal for Numerical Methods in Engineering 74 (1) (2008) 32–55. doi:10.1002/nme.2153.
[57] X. Wang, L. T. Zhang, Interpolation functions in the immersed boundary and finite element methods, Computational Mechanics 45 (4) (2010) 321–334. doi:10.1007/s00466-009-0449-5.
[58] A. J. Gil, A. Arranz Carreño, J. Bonet, O. Hassan, The immersed structural potential method for haemodynamic applications, Journal of Computational Physics 229 (22) (2010) 8613–8641. doi:10.1016/j.jcp.2010.08.005.
[59] A. J. Gil, A. Arranz Carreño, J. Bonet, O. Hassan, An enhanced Immersed Structural Potential Method for fluid-structure interaction, Journal of Computational Physics 250 (2013) 178–205. doi:10.1016/j.jcp.2013.05.011.
[60] C. Hesch, a. J. Gil, A. Arranz Carreño, J. Bonet, On continuum immersed strategies for Fluid-Structure Interaction, Computer Methods in Applied Mechanics and Engineering 247-248 (2012) 51–64. doi:10.1016/j.cma.2012.07.021.
[61] A. Ibrahimbegović, F. Frey, Stress resultant geometrically nonlinear shell theory with drilling rotations-Part II. Computational aspects, Computer Methods in Applied Mechanics and Engineering 118 (3-4) (1994) 285–308. doi:10.1016/0045-7825(94)90004-3.
[62] M. A. Crisfield, U. Galvanetto, G. Jelenic, Dynamics of 3-D co-rotational beams, Computational Mechanics 20 (6) (1997) 507–519. doi:10.1007/s004660050271.
[63] A. Ibrahimbegović, M. Al Mikdad, Finite rotations in dynamics of beams and implicit time-stepping schemes, International Journal for Numerical Methods in Engineering 41 (5) (1998) 781–814. doi:10.1002/(SICI)1097-0207(19980315)41:5<781::AID-NME308>3.0.CO;2-9.
[64] A. Cardona, M. Geradin, A beam finite element non-linear theory with finite rotations, International Journal for Numerical Methods in Engineering 26 (11) (1988) 2403–2438. doi:10.1002/nme.1620261105.
[65] G. Zavarise, P. Wriggers, Contact with friction between beams in 3-D space, International Journal for Numerical Methods in Engineering 49 (8) (2000) 977–1006. doi:10.1002/1097-0207(20001120)49:8<977::AID-NME986>3.0.CO;2-C.
[66] R. Ortega, J. C. García Orden, M. Cruchaga, C. García, Energy-consistent simulation of frictional contact in rigid multibody systems using implicit surfaces and penalty method, Multibody System Dynamics 41 (3) (2017) 275–295. doi:10.1007/s11044-017-9565-8.
[67] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (January 2002) (2002) 479–517. doi:10.1017/S0962492902000077.
[68] B. E. Griffith, C. S. Peskin, On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems, Journal of Computational

Physics 208 (1) (2005) 75–105. doi:10.1016/j.jcp.2005.02.011.
[69] H. M. Hilber, T. J. Hughes, R. L. Taylor, Improved numerical dissipation for time integration algorithms in structural dynamics, Earthquake Engineering & Structural Dynamics 5 (3) (1977) 283–292. doi:10.1002/eqe.4290050306.
[70] T. N. Le, J. M. Battini, M. Hjiaj, Dynamics of 3D beam elements in a corotational context: A comparative study of established and new formulations, Finite Elements in Analysis and Design 61 (2012) 97–111. doi:10.1016/j.finel.2012.06.007.
[71] C. Batty, SDFGen (2012).
URL https://github.com/christopherbatty/SDFGen
[72] Q. Wang, Q. Tian, H. Hu, Dynamic simulation of frictional multi-zone contacts of thin beams, Nonlinear Dynamics 83 (4) (2016) 1919–1937. doi:10.1007/s11071-015-2456-8.
[73] Q. Wang, Q. Tian, H. Hu, Dynamic simulation of frictional contacts of thin beams during large overall motions via absolute nodal coordinate formulation, Nonlinear Dynamics 77 (4) (2014) 1411–1425. doi:10.1007/s11071-014-1387-0.
[74] P. Wriggers, T. Vu Van, E. Stein, Finite element formulation of large deformation impact-contact problems with friction, Computers and Structures 37 (3) (1990) 319–331. doi:10.1016/0045-7949(90)90324-U.
[75] T. A. Laursen, V. Chawla, Design of energy conserving algorithms for frictionless dynamic contact problems, International Journal for Numerical Methods in Engineering 40 (5) (1997) 863–886. doi:10.1002/(SICI)1097-0207(19970315)40:5<863::AID-NME92>3.0.CO;2-V.
[76] T. A. Laursen, V. Chawla, Design of energy conserving algorithms for frictionless dynamic contact problems, International Journal for Numerical Methods in Engineering 40 (5) (1997) 863–886. doi:10.1002/(SICI)1097-0207(19970315)40:5<863::AID-NME92>3.0.CO;2-V.
[77] P. Hauret, P. Le Tallec, Energy-controlling time integration methods for nonlinear elastodynamics and low-velocity impact, Computer Methods in Applied Mechanics and Engineering 195 (37-40) (2006) 4890–4916. doi:10.1016/j.cma.2005.11.005.
[78] F. Armero, E. Petocz, Formulation and analysis of conserving algorithms for frictionless dynamic contact/impact problems, Computer Methods in Applied Mechanics and Engineering 158 (3-4) (1998) 269–300. doi:10.1016/S0045-7825(97)00256-9.
[79] T. A. Laursen, G. R. Love, Improved implicit integrators for transient impact problems - Geometric admissibility within the conserving framework, International Journal for Numerical Methods in Engineering 53 (2) (2002) 245–274. doi:10.1002/nme.264.
[80] V. Acary, Energy conservation and dissipation properties of time-integration methods for nonsmooth elastodynamics with contact, ZAMM Zeitschrift fur Angewandte Mathematik und Mechanik 96 (5) (2016) 585–603. arXiv:1410.2499, doi:10.1002/zamm.201400231.
[81] A. Ibrahimbegovic, S. Mamouri, Energy conserving/decaying implicit time-stepping scheme for nonlinear dynamics of three-dimensional beams undergoing finite rotations, Computer Methods in Applied Mechanics and Engineering 191 (37-38) (2002) 4241–4258. doi:10.1016/S0045-7825(02)00377-8.
[82] Stanford_University, The Stanford 3D Scanning Repository (2013).
URL http://graphics.stanford.edu/data/3Dscanrep/
[83] H. Sharei, T. Alderliesten, J. J. van den Dobbelsteen, J. Dankelman, Navigation of guidewires and catheters in the body during intervention procedures: a review of computer-based models, Journal of Medical Imaging 5 (01) (2018) 1. doi:10.1117/1.jmi.5.1.010902.
[84] M. W. Jones, J. A. Bærentzen, M. Sramek, 3D distance fields: A survey of techniques and applications, in: IEEE Transactions on Visualization and Computer Graphics, Vol. 12, 2006, pp. 581–599. doi:10.1109/TVCG.2006.56.
[85] A. Neic, M. A. Gsell, E. Karabelas, A. J. Prassl, G. Plank, Automating image-based mesh generation and manipulation tasks in cardiac modeling workflows using Meshtool, SoftwareX 11 (2020) 100454. doi:10.1016/j.softx.2020.100454.