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, France

Received 9 May 2020; received in revised form 4 July 2020; accepted 6 July 2020
Available online 5 August 2020

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.

© 2020 Elsevier B.V. All rights reserved.

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

* Corresponding author.
E-mail address: miquel.aguirre@emse.fr (M. Aguirre).

https://doi.org/10.1016/j.cma.2020.113275
0045-7825/© 2020 Elsevier B.V. All rights reserved.
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 pursuit 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 $2\pi$. 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-α 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 \( \mathbf{E}_i \) \((i = 1, 2, 3)\). Similarly, a reference system is defined in the spatial (Eulerian) configuration, given by the triad \( \mathbf{e}_i \) \((i = 1, 2, 3)\). As in [61], both triads are made coincident, i.e. \( \mathbf{E}_i = \delta_{ij} \mathbf{e}_j \) with \( \delta_{ij} \) 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
\[
\mathbf{\theta} = \vartheta \mathbf{n},
\]
with \( \mathbf{n} \) the axis of rotation and \( \vartheta \) the rotation angle given by
\[
\vartheta = \sqrt{\mathbf{\theta} \cdot \mathbf{\theta}}.
\]

The relationship between the rotation angle and the rotation matrix is provided by the Rodrigues formula, given by
\[
\mathbf{R}(\mathbf{\theta}) = \mathbf{I} + \frac{\sin \vartheta}{\vartheta} \mathbf{S(\theta)} + \frac{1 - \cos \vartheta}{\vartheta^2} \mathbf{S(\theta)S(\theta)} = \exp(\mathbf{S(\theta)}),
\]
where \( \mathbf{I} \) is the \(3 \times 3\) identity matrix and \( \mathbf{S(\theta)} \) is the skew-symmetric matrix of \( \mathbf{\theta} \), given by
\[
\mathbf{S(\theta)} = \begin{bmatrix}
0 & -\theta_3 & \theta_2 \\
\theta_3 & 0 & -\theta_1 \\
-\theta_2 & \theta_1 & 0
\end{bmatrix}.
\]
The variation of \( \mathbf{R}(\mathbf{\theta}) \) is obtained by first constructing a superimposed infinitesimal rotation [43]. This is done by using the exponential map (3) and by noticing that, since \( \mathbf{R} \) is a two point tensor, the superimposed rotation needs to be a spatial quantity, i.e.
\[
\mathbf{R}_\epsilon = \exp(\epsilon \mathbf{S(w)}) \mathbf{R}
\]
and, therefore,
\[
\delta \mathbf{R} = \frac{d}{d\epsilon} \mathbf{R}_\epsilon|_{\epsilon=0} = \mathbf{S(\delta w)R},
\]
with \( \delta \mathbf{w} \) and \( \mathbf{S(\delta w)} \) spatial objects defined as
\[
\delta \mathbf{w} = w_i \mathbf{e}_i,
\]
\[
\mathbf{S(\delta w)} = [\mathbf{S(\delta w)}]_{ij} \mathbf{e}_i \otimes \mathbf{e}_j,
\]
where \( \delta \mathbf{w} \) is named the spatial spin vector and are infinitesimal spatial rotations superimposed to the rotation \( \mathbf{R} \). Instead of a spatial spin vector, a material one can be used, defined as
\[
\delta \mathbf{W} = \delta \mathbf{W}_i \mathbf{E}_i,
\]
\[
\mathbf{S(\delta W)} = [\mathbf{S(\delta W)}]_{ij} \mathbf{E}_i \otimes \mathbf{E}_j,
\]
where $\delta W$ and $S(\delta W)$ are the pull back of $\delta w$ and $S(\delta w)$, respectively, i.e.

$$\delta W = R^T \delta w,$$

$$S(\delta W) = R^T S(\delta w) R.$$  \hspace{1cm} (11)

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

$$\delta R = R^T S(\delta W).$$  \hspace{1cm} (13)

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

$$\delta w = T_s(\theta) \delta \theta,$$

$$\delta \theta = T_s^{-1}(\theta) \delta w,$$

with

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

$$T_s^{-1}(\theta) = \frac{(\theta/2)}{\tan(\theta/2)} I_{3x3} + \frac{1}{\theta^2} \left( 1 - \frac{(\theta/2)}{\tan(\theta/2)} \right) \theta \theta^T - \frac{1}{2} S(\theta).$$ \hspace{1cm} (15a)

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

$$S(\dot{w}) = \dot{R} R^T,$$

$$S(\ddot{w}) = \dot{R} R^T + \ddot{R} R^T,$$ \hspace{1cm} (16a)

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

$$\dot{w} = T_s(\theta) \dot{\theta}.$$  \hspace{1cm} (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 $\Omega$ and material configuration $\Omega_0$. 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. $\bar{\Omega} \approx \Omega_0$ and therefore they are uniquely denoted by $\bar{\Omega}$. The basis vectors of $\bar{\Omega}$ are $\bar{\xi}_i$ ($i = 1, 2, 3$) and are made coincident with the material and spatial basis, i.e. $\bar{\xi}_i = \delta_{ij} e_j$, $\bar{\xi}_i = \delta_{ij} e_j$.

To define the beam position, an orthogonal system $T_i$ ($i = 1, 2, 3$) 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 $s = \xi_t$ in $[0, l_0]$ that coincides with the axis of the beam, defined by $T_1$. The cross sections of the beam lie in planes defined by the basis vectors $[T_2, T_3]$, which depend on the arc-length parameter $s$. Accordingly, the beam is deformed into a current configuration $\Omega$, with an orthonormal frame $t_i$ defined also at the centroid of the beam cross section. These vectors are defined through the following transformations

$$T_i(s) = \Lambda_0(s) \bar{\xi}_i; \quad \Lambda_0(s) = T_i(s) \otimes \bar{\xi}_i = [\Lambda_0]_{ij} e_i \otimes \bar{\xi}_j,$$

$$t_i(s, t) = \Lambda(s, t) \bar{\xi}_i; \quad \Lambda(s, t) = t_i(s, t) \otimes \bar{\xi}_i = [\Lambda]_{ij} e_i \otimes \bar{\xi}_j,$$ \hspace{1cm} (18)

with $\Lambda_0, \Lambda$ being two point tensors that give, respectively, orthonormal triads $T_i, t_i$ (material and spatial objects, respectively) as a function of the basis vectors $\bar{\xi}_i$. By comparing relationships (18) and (19), a map between $T_i$ and $t_i$ can be defined as

$$t_i(s, t) = \dot{R}(s, t) T_i(s); \quad R(s, t) = \Lambda(s, t) \Lambda_0^{-1}(s) = t_i \otimes T_i = [R]_{ij} e_i \otimes E_j,$$ \hspace{1cm} (19)
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

\[ X(s, \xi_2, \xi_3) = X_0(s) + \xi_2 T_2(s) + \xi_3 T_3(s), \]

\[ x(s, \xi_2, \xi_3) = x_0(s) + \xi_2 t_2(s) + \xi_3 t_3(s), \]

where \( X_0(s), x_0(s) \) 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 \( \tilde{\Omega} \) and \( \Omega_0 \) and between \( \tilde{\Omega} \) and \( \Omega \). To do so, two frames that move rigidly with the beam axis are considered [37]. This is a material frame \( V_i \) \((i = 1, 2, 3)\) and a spatial frame \( v_i \) \((i = 1, 2, 3)\). In order to compute \( v_i \) \((i = 1, 2, 3)\), the procedure detailed in [38] is used,

\[ v_1(t) = \frac{x_0(l_0, t) - x_0(0, t)}{\|x_0(l_0, t) - x_0(0, t)\|}; \quad v_3(t) = \frac{v_1(t) \times p(t)}{\|v_1(t) \times p(t)\|}; \quad v_2(t) = v_3(t) \times v_1(t), \]

with the auxiliary vector \( p \) given by

\[ p(t) = \frac{1}{2} (t_2(0, t) + t_2(l_0, t)), \]

with an identical procedure to obtain, i.e \( V_i = v_i(0) \) \((i = 1, 2, 3)\). The above basis vectors allow defining the following mappings

\[ V_i = R^0 E_i; \quad R^0 = V_i \otimes \bar{E}_i = \left[R^0\right]_{ij} E_i \otimes \bar{E}_j, \]

\[ v_i(t) = R_e(t) \bar{E}_i; \quad R_e(t) = v_i(t) \otimes \bar{E}_i = \left[R_e\right]_{ij} e_i \otimes \bar{E}_j, \]

with \( R^0 \) being a two point tensor that goes from \( \tilde{\Omega} \) to \( \Omega_0 \) and \( R_e \) a two point tensor that goes from \( \tilde{\Omega} \) to \( \Omega \). It is worth noticing that \( R_e(0) = R^0 \) and that neither \( R^0 \) nor \( R_e \) 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 \( \tilde{\Omega} \), i.e.

\[ \tilde{T}_i(s) = (R^0)^T T_i(s), \]

\[ \tilde{t}_i(s, t) = R^0 t_i(s, t). \]

Equivalently to Eq. (20), a mapping can be defined between \( \tilde{T}_i \) and \( \bar{t}_i \) as

\[ \bar{t}_i(s, t) = R(s, t) \tilde{T}_i(s), \quad \tilde{R}(s, t) = R^T_e(t) R(s, t) R^0 = \tilde{t}_i(s, t) \otimes \tilde{T}_i(s) = [\tilde{R}]_{ij} \bar{E}_i \otimes \bar{E}_j. \]

where \( \tilde{R} \) is a small rotation of the cross sectional vectors \( \tilde{T}_i \) in \( \tilde{\Omega} \), now absent of rigid body motions. Assuming the material beam as straight, the rotation matrix \( \Lambda_0 \) can be made coincident with \( R^0_e \), this is \( \Lambda_0(s) = R^0_e \), while \( T_i = V_i \) and \( \tilde{T}_i = \bar{E}_i \). 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

\[ X_0(s) = X_0(0) + s V_1, \]

\[ x_0(s) = x_0(0) + s v_1 + u_1(s) v_1 + u_2(s) v_2 + u_3(s) v_3, \]

where \( u_i(s) \) \((i = 1, 2, 3)\) are the positions of the beam centerline respect the rigid frame \( v_i \). Given the definition of \( v_1 \) in Eq. (23), the displacements \( u_i(s) \) follow

\[ u_1(0) = 0; \quad u_1(l_0) = \bar{u}; \quad \bar{u} = l_n - l_0, \]

\[ u_2(0) = 0; \quad u_2(l_0) = 0, \]

\[ u_3(0) = 0; \quad u_3(l_0) = 0. \]

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

\[ \tilde{X}(s, \xi_2, \xi_3) = \tilde{X}_0(s) + \xi_2 \tilde{E}_2 + \xi_3 \tilde{E}_3; \quad \tilde{X}_0(s) = s \bar{E}_1 \]

\[ \tilde{x}(s, \xi_2, \xi_3) = \tilde{x}_0(s) + \xi_2 \tilde{R}(s) \bar{E}_2 + \xi_3 \tilde{R}(s) \bar{E}_3; \quad \tilde{x}_0(s) = s \bar{E}_1 + u_1(s) \bar{E}_1 + u_2(s) \bar{E}_2 + u_3(s) \bar{E}_3. \]
which defines the beam deformaional displacement \( \ddot{u} = \dddot{x} - \dot{X} \) in \( \tilde{\Omega} \) as
\[
\ddot{u}(s, \xi_2, \xi_3) = \ddot{u}_0(s) + \xi_2(\dddot{R}(s) - \dot{I})\ddot{E}_2 + \xi_3(\dddot{R}(s) - \dot{I})\ddot{E}_3; \quad \ddot{u}_0(s) = u_1(s)\dddot{E}_1 + u_2(s)\dddot{E}_2 + u_3(s)\dddot{E}_3,
\]
which is assumed to be very small, i.e. \( \dddot{x} \simeq \dddot{X} \). Via equation (33), it is worth noticing that \( s = \dot{X}_1, \xi_a = \dddot{X}_a \).

It is useful now to write the deformation gradient using finite-elasticity three dimensional theory. Following Auricchio and coauthors [51] this is defined as
\[
F(s) = \frac{\partial x}{\partial \xi_i} \otimes \frac{\partial \xi_i}{\partial X} = R(s) + a(s) \otimes V_1,
\]
where Eq. (20) has been used alongside with \( T_i = V_i \). The vector \( a \) is defined as
\[
a(s) = \gamma(s) + \xi_a \kappa_a(s),
\]
with \( \gamma \) and \( \kappa_a \) defined as (see [51])
\[
\gamma(s) = x'_0(s) - t_1(s), \quad \kappa_a(s) = \dot{a}_a(s).
\]
where \( (\cdot)' = \frac{d(\cdot)}{ds} \).

**Remark 2.2.1.** In comparison with [51], it is worth noticing that \( R \) is the rotation from the orthonormal basis \( V_i \) in the initial configuration (straight, but not parallel to the \( \dddot{E}_i \) axis) to the orthonormal triad \( t_i \) in the current configuration. In [51], the orthonormal triad in the material configuration coincides with the reference axes, i.e. \( V_i = \delta_{ij} \dddot{E}_j \) and, therefore, \( \Lambda_0 = I \) and, via equation (20), \( R = \Lambda \).

The deformation gradient (36) can be decomposed using a left decomposition to yield
\[
F(s) = R(s)A'(s); \quad A'(s) = I + a'(s) \otimes V_1
\]
with \( a' \) being, as defined in [51], the \textit{rotate-back} of \( a \) in the material configuration, i.e.
\[
a'(s) = R(s)^T a(s).
\]
Eq. (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 Eqs. (25), (26), the following multiplicative decomposition can be carried out
\[
F(s) = R_e \ddot{F}(s)(R_e^0)^T
\]
with \( \ddot{F} \) being the deformation gradient in the configuration \( \ddot{\Omega} \), which is defined by
\[
\ddot{F}(s) = \ddot{R}(s) + \ddot{a}(s) \otimes \dddot{E}_1
\]
with \( \dddot{a} \) being now the \textit{rigid-rotate-back} of \( a \) to the configuration \( \ddot{\Omega} \), i.e.
\[
\ddot{a}(s) = R_e^T a(s) = \ddot{\gamma}(s) + \xi_a \dddot{\kappa}_a(s),
\]
with
\[
\ddot{\gamma} = x'_0(s) - \dddot{t}_1, \quad \dddot{\kappa}_a = R_e^T t'_a(s) = (R_e^T t_a(s))' = \ddot{a}_a = \dddot{R}(s)\dddot{E}_a.
\]
where the fact that \( R_e \) is independent of the arc-length coordinate \( s \) has been used.

**Remark 2.2.2.** Differently from \( a' \) in (41), \( \ddot{a} \) has been obtained in (44) by removing the rigid body motions via \( R_e \) and hence the denomination \textit{rigid-rotate-back}. The relation between the two is given by \( a' = R_e^0 \ddot{R}^T \dddot{a} \).
Remark 2.2.3. Eq. (40) decomposes the deformation gradient into one material tensor, \( A' \), followed by one rotation, \( R \), both dependent on the arc-length parameter \( s \). On the other hand, Eq. (42) decomposes the deformation gradient into one rotation (\( R^0 \)), followed by one deformation gradient \( \bar{F} \), followed by another rotation \( R_c \). In this case only \( \bar{F} \) 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 \( \bar{R} \) can be parametrized by a first order approximation of (3), i.e.

\[
\bar{R}(s) = I + S(\bar{\Theta}(s))
\]

(47)

where \( \bar{\Theta} \) 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)

\[
\nabla \bar{\boldsymbol{u}} = S(\bar{\Theta}) + \bar{a} \otimes \bar{E}_1.
\]

(48)

Using Eqs. (47) and (48) into (43) yields the following approximation of \( \bar{F} \)

\[
\bar{F} \simeq I + \varepsilon(\nabla \bar{\boldsymbol{u}}) + \Omega(\nabla \bar{\boldsymbol{u}})
\]

(49)

which is the additive decomposition of the deformation gradient.

Remark 2.2.4. Eqs. (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 \( \Omega_0 \) to \( \bar{\Omega} \)), followed by a small deformation (in \( \bar{\Omega} \)), followed by another rigid rotation (from \( \bar{\Omega} \) to \( \Omega \)).

From Eq. (42), the Green–Lagrange strain tensor can be computed as

\[
E = R^0_1 \tilde{E}(R^0_e)^T.
\]

(50)

with

\[
\tilde{E} = \frac{1}{2}(\bar{F}^T \bar{F} - I) = \tilde{E}_{11} \tilde{E}_1 \otimes \tilde{E}_1 + \tilde{E}_{1a} \tilde{E}_1 \otimes (\bar{a}_a \tilde{E}_a),
\]

and

\[
\tilde{E}_{11} = \frac{1}{2}(\bar{a} \cdot \bar{a}) + (\bar{a} \cdot \bar{r}_1),
\]

\[
\tilde{E}_{1a} = \bar{a}_{a1} = \frac{1}{2} (\bar{a} \cdot \bar{r}_a),
\]

(51)

(52a)

(52b)

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

\[
\tilde{E} = \frac{1}{2}(\bar{F}^T \bar{F} - I) \simeq \varepsilon(\nabla \bar{\boldsymbol{u}})
\]

(53)

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

\[
\delta W_{int} = \int_{\Omega_0} S : \delta E d\Omega_0 = \int_{\bar{\Omega}} \bar{\sigma} : \delta \tilde{e} d\bar{\Omega} = \int_{\bar{\Omega}} [\tilde{\sigma}_{11} \delta \tilde{e}_{11} + \tilde{\sigma}_a \delta \tilde{e}_{1a} + \tilde{\sigma}_{a1} \delta \tilde{e}_{a1}] d\bar{\Omega}
\]

(54)

where the notation \( \tilde{e} = \varepsilon(\nabla \bar{\boldsymbol{u}}) \) has been used alongside the definition of

\[
\bar{\sigma} = (R^0_e)^T S R^0_e.
\]

(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 \( R_c \). Specifically, the following relationship between stresses and strains in \( \bar{\Omega} \) is used

\[
\bar{\sigma} = \lambda \text{tr}(\tilde{e}) I + 2G \tilde{e}
\]

(56)
where \( \lambda, G \) are, respectively, the first and second Lamé parameters. Using the approximation \( E \simeq \lambda + 2G \) \([51]\), the stress components contributing into Eq. (54) can be written as

\[
\begin{align*}
\bar{\sigma}_{11} &= (\lambda + 2G)e_{11} \simeq E\tilde{e}_{11}, \\
\bar{\sigma}_{1a} &= \bar{\sigma}_{a1} = 2G\tilde{e}_{a1},
\end{align*}
\]  

(57a,b)

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

\[
W_{int} = \int_0^l \left( \frac{1}{2} E \int_A \tilde{e}_{11}^2 dA + \frac{1}{2} G \int_A ((2\tilde{e}_{12})^2 + (2\tilde{e}_{13})^2) dA \right) ds.
\]

(58)

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

\[
\begin{align*}
\delta R &= S(\delta w)R, \\
\delta R_c &= S(\delta w_c)R_c, \\
\delta \bar{R} &= S(\delta \bar{W})\bar{R}.
\end{align*}
\]

(60, 61, 62)

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

\[
\begin{align*}
\delta \bar{W} &= R^T \delta \bar{w}, \\
\delta W_c &= R_c^T \delta w_c,
\end{align*}
\]

(63, 64)

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

\[
\delta \bar{W} = \delta W - \delta W_c.
\]

(65)

2.3. Finite Element discretization

The beam centerline is discretized in \( N \) initially straight elements of length \( l^e_0 \) such that \( l_0 \simeq \bigcup_{i=1}^N l^e_i \). Each of these segments is defined by two nodes \( a = 1, 2 \) with material positions \( X_a \) and spatial positions \( x_a = X_a + u_a \), with \( u_a \) being the nodal displacement. The cross section rotation at the node is defined by the nodal rotation \( R_a \), such that

\[
\begin{align*}
u_1 &= u_0(0); \quad R_1 = R(0), \\
u_2 &= u_0(l_0); \quad R_2 = R(l_0),
\end{align*}
\]

(66, 67)

(see Fig. 1). The rotation matrices \( R_a \) are parametrized via the rotational vector \( \theta_a \), while their variation requires the spatial spin vector \( w_a \), i.e.

\[
R_a = \exp(S(\theta_a)); \quad \delta R_a = S(\delta w_a)R_a, \quad (a = 1, 2)
\]

(68)

which defines the elemental vector of global displacements and rotations, \( d \), and their variation \( \delta d \)

\[
d = [u^T_1 \quad \theta^T_1 \quad u^T_2 \quad \theta^T_2]^T; \quad \delta d = [\delta u^T_1 \quad \delta \theta^T_1 \quad \delta u^T_2 \quad \delta \theta^T_2]^T.
\]

(69)

Following the procedure explained in the previous section, two orthonormal triads \( V_i, v_i \) are built tangent to the axis connecting nodes 1 and 2 in the material and spatial configurations, respectively, which are rigidly rotated

\[1\] 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.
using $\mathbf{R}_0^c$, $\mathbf{R}_e$ (see Eqs. (25), (26)). These mappings allow to do rigid-rotate-back of the variations $\delta \mathbf{d}$ in the $\tilde{\Omega}$ configurations, via

$$\delta \mathbf{D} = E^T \delta \mathbf{d}; \quad \delta \mathbf{D} = \begin{bmatrix} \delta \mathbf{U}_1^T & \delta \mathbf{W}_1^T & \delta \mathbf{U}_2^T & \delta \mathbf{W}_2^T \end{bmatrix}; \quad E = \frac{\partial \mathbf{D}}{\partial \mathbf{d}} = \text{diag}(R_e).$$

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

$$\bar{u}_1 = \bar{u}_0(0); \quad \bar{R}_1 = \bar{R}(0) = \mathbf{R}_c^T \mathbf{R}(0) \mathbf{R}_e^0 = \mathbf{R}_c^T \mathbf{R}_1 \mathbf{R}_e^0,$$

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

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

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

and therefore the only non-zero component is $\bar{u}$, this is, the displacement of node 2 in the $X_1$ direction. At the same time, the rotation matrices $\bar{R}_a$ are parametrized via the rotational vector $\bar{\Theta}_a$, while their variation requires the spin vector $\bar{W}_a$, i.e

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

From the nodal quantities $\bar{u}$, $\bar{\Theta}_1$, $\bar{\Theta}_2$ the centerline displacements $\bar{u}_0(\xi)$, and the cross section rotational vector $\bar{\Phi}(\xi)$ 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].

$$\bar{u}_0 = u_1(\xi)E_1 + u'(\xi),$$

with

$$u_1(\xi) = N_2(\xi)\bar{u}; \quad u'(\xi) = \begin{bmatrix} 0 \\ u_2(\xi) \\ u_3(\xi) \end{bmatrix} = P_1(\xi) \begin{bmatrix} \bar{\Theta}_1 \\ \bar{\Theta}_2 \end{bmatrix}.$$
The rotations are interpolated as
\[ \bar{\Theta}(\xi) = P_2(\xi) \left[ \begin{array}{c} \bar{\Theta}_1 \\ \bar{\Theta}_2 \end{array} \right], \] (77)
with the expression of \( P_1, P_2 \) 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.
\[ \bar{\mathbf{D}} = [\bar{u} \quad \bar{\Theta}_1 \quad \bar{\Theta}_2]^T; \quad \delta \bar{\mathbf{D}} = [\delta \bar{u} \quad \delta \bar{\Theta}_1 \quad \delta \bar{\Theta}_2]^T. \] (78)
with the following relationship between \( \delta \bar{\mathbf{D}} \) and \( \delta \bar{d} \)
\[ \delta \bar{\mathbf{D}} = B \delta \bar{d}, \] (79)
with \( B \) given in Appendix C. On top of this, the dynamic and frictional terms require the variation and time derivatives of the centerline displacement \( u_0 \) as well as those of the centerline spin-vector \( w_0 \). From Eq. (75), the centerline displacement in the current configuration is given by
\[ u^h_0 = u_1 N_1(\xi) + u_2 N_2(\xi) + R_e u^l(\xi). \] (80)
The variation of \( u^h_0 \) is obtained from the equation above
\[ \delta u^h_0 = R_e H_1 E^T \delta d, \] (81)
and the first and second time derivatives as
\[ \dot{u}^h_0 = R_e H_1 E^T \dot{d}, \] (82)
\[ \ddot{u}^h_0 = R_e H_1 E^T \ddot{d} + R_e C_1 E^T \dot{d}, \] (83)
with \( H_1, C_1 \) given in Appendix D. Regarding the rotational variables, Eq. (65) is used to write
\[ \delta \bar{W}^h_0 = \delta W^h_0 - \delta W_e, \] (84)
which yields the following variations and first and second derivatives of the rotational variables
\[ \delta w^h_0 = R_e H_2 E^T \delta d, \] (85)
\[ \dot{w}^h_0 = R_e H_2 E^T \dot{d}, \] (86)
\[ \ddot{w}^h_0 = R_e H_2 E^T \ddot{d} + R_e C_2 E^T \dot{d}, \] (87)
where again the expressions of \( H_2, C_2 \) are given in Appendix D.

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
\[ \delta W = \delta W_{int} + \delta W_{kin} - \delta W_{ext} = 0, \] (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
\[ \delta W_{int} = (\delta d)^T T_{int}, \] (89)
with \( T_{int} \) the elemental internal forces in \( \Omega \), defined as
\[ T_{int} = \frac{\partial W_{int}}{\partial d} = B^T \tilde{T}_{int} \equiv \left[ \begin{array}{ccc} F_1^T & M_1^T & F_2^T \\ \tilde{M}_1 & \tilde{M}_2 \end{array} \right]^T, \] (90)
with \( B \) the matrix expressing the relationship between \( \delta d \) and \( \delta \bar{\mathbf{D}} \) via equation (79), \( F_a \) the nodal force and \( M_a \) the nodal moment. Finally \( \tilde{T}_{int} \) are the elemental internal forces in \( \tilde{\Omega} \) defined as
\[ \tilde{T}_{int} = \frac{\partial W_{int}}{\partial \tilde{\mathbf{D}}} \equiv \left[ \begin{array}{ccc} \tilde{N} & \tilde{M}_1^T & \tilde{M}_2^T \end{array} \right]^T, \] (91)
with \( \bar{N} \) the elemental axial force and \( \bar{M}_a \) 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 of small strains in \( \bar{\Omega}(51) \) a closed form expression can be obtained for \( \bar{T}_{int} \) for a given set of displacements \( \bar{D} \), 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])

\[
\bar{T}_{int} = \bar{K}_{int} \bar{D},
\]

with

\[
\bar{K}_{int} = \frac{1}{l_0} \begin{bmatrix}
AE & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & GJ & 0 & 0 & \text{ }_GJ & 0 & 0 \\
0 & 0 & 4EI_{33} & 0 & 0 & 2EI_{33} & 0 \\
0 & \text{ }_GJ & 0 & 0 & 4EI_{22} & 0 & 0 \\
0 & 0 & 2EI_{33} & 0 & 0 & 4EI_{33} & 0 \\
0 & 0 & 0 & 2EI_{22} & 0 & 0 & 4EI_{22}
\end{bmatrix},
\]

where \( l_{22}, l_{33} \) 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 \( T_{int} \) is required. This is given by

\[
\delta T_{int} = K_{int} \delta d,
\]

with

\[
K_{int} = B^T \bar{K}_{int} B + \bar{K}_{int}
\]

and with \( \bar{K} \) provided in Appendix C.

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

\[
W_{\text{kin}} = \frac{1}{2} \int_{l_0} (A_\rho (\dot{u}_0 \cdot \dot{u}_0) + \dot{w}_0 \cdot I_\rho \dot{w}_0) \, ds.
\]

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

\[
\delta W_{\text{kin}} = T_k^T \delta d,
\]

with \( T_k \) given by

\[
T_k = E \int_{l_0} (A_\rho H_1^T R_e^T \dot{u}_0 + H_2^T \bar{I}_\rho R_e^T \dot{w}_0 + H_2^T S(\dot{W}_0) \bar{I}_\rho \dot{W}_0) \, ds,
\]

with

\[
\dot{W}_0 = R_e^T \dot{w}_0,
\]

\[
\bar{I}_\rho = RJ_\rho R^T.
\]

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

\[
\delta T_k \simeq M \delta \ddot{d} + C_k \delta \dot{d},
\]

with

\[
M = E \left( \int_{l_0} A_\rho H_1^T H_1 + H_2^T \bar{I}_\rho H_2 \, ds \right) E^T,
\]

\[
C_k = E \left( \int_{l_0} (A_\rho H_1^T (C_1 + C_3) + H_2^T \bar{I}_\rho (C_2 + C_4) + H_2^T (S(\dot{W}_0) \bar{I}_\rho - S(\bar{I}_\rho \dot{W}_0)) H_2) \, ds \right) E^T,
\]

with the expressions for \( C_3, C_4 \) 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 $\delta W_c$ as

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

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 $\Gamma_s$ (see Fig. 2a)

$$\delta W_c = \int_{0}^{l_0} (p_N \delta g_N + t_T \cdot \delta g_T) \, ds, \quad (105)$$

with $g_N$ being the normal gap respect the master surface, $g_T$ the relative displacement in the tangential direction, $p_N$ the contact pressure and $t_T$ 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

$$g_N(s) = d_i(s) - r, \quad (106)$$

with $d_i(s)$ 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

$$\delta g_N = \frac{\partial g_N}{\partial x} \cdot \delta x_0 = \frac{\partial d_i}{\partial x} \cdot \delta u_0 = n \cdot \delta u_0, \quad (107)$$

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

$$\delta u_0 = (\delta u_0 \cdot n) n + \delta u_T = \delta g_N \frac{\partial g_N}{\partial x} + \delta g_T. \quad (108)$$

Isolating $\delta g_T$ in the above expression yields

$$\delta g_T = \left( I - \frac{\partial g_N}{\partial x} \otimes \frac{\partial g_N}{\partial x} \right) \delta u_0, \quad (109)$$

$$\dot{g}_T = \left( I - \frac{\partial g_N}{\partial x} \otimes \frac{\partial g_N}{\partial x} \right) \dot{u}_0. \quad (110)$$
In order to facilitate the convergence of the Newton Raphson algorithm, both the contact pressure $p_N$ and frictional contact force $t_T$ are regularized. The contact pressure is evaluated using the penalty regularization used in [45,47–49]

$$p_N = p_N(g_N) = \begin{cases} \tilde{p}_N - \varepsilon_c \tilde{g}_N, & g \leq 0 \\ \frac{\varepsilon_c \tilde{g}_N - \tilde{p}_N}{\tilde{g}_N^2} \tilde{g}_N^2 - \varepsilon_c \tilde{g}_N + \tilde{p}_N, & 0 < g_N \leq \tilde{g}_N \\ 0, & g_N > \tilde{g}_N \end{cases}$$  \hspace{1cm} (111)

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

$$t_T = t_T(g_N, \dot{g}_T) = -\mu p_N(g_N) \frac{\dot{g}_T}{\sqrt{\|\dot{g}_T\|^2 + \varepsilon_T}}.$$  \hspace{1cm} (112)

with $\varepsilon_T$ being a regularization parameter (see Fig. 3b).

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

$$T_c = \int_0^{l_0} E H_1^T R_c^T f_c ds,$$  \hspace{1cm} (113)

with $H_1$ given in Appendix D, $E$ given in Eq. (70) and with $f_c$ being the contact force, given by

$$f_c = p_N(d) G(d; \dot{d}),$$  \hspace{1cm} (114)

with

$$G(x; d; \dot{d}) = \frac{\partial g_N}{\partial x} - \frac{\dot{g}_N}{\mu \sqrt{\|\dot{g}_T\|^2 + \varepsilon}}.$$  \hspace{1cm} (115)

The contact tangent matrix is obtained by linearizing $T_c$ with respect the displacement and velocity contributions (see Appendix E for details),

$$\delta T_c = K_c \delta d + C_c \delta \dot{d},$$  \hspace{1cm} (116)

with $K_c$ defined as

$$K_c = K_c^1 + K_c^2 + K_c^3 + K_c^4,$$  \hspace{1cm} (117)
and with each of the terms given by

\begin{align}
K_c^1 &= -\int_0^l E\hat{S}(H^T_{c}\mathcal{F}_c)G^T E^T \, ds, \\
K_c^2 &= \int_0^l \left( \frac{N^T}{l^2} EA_1 T\mathcal{F}_c r - EGS(\mathcal{F}_c) P_1 PE^T \right) \, ds, \\
K_c^3 &= \int_0^l EH_1^T S(\mathcal{F}_c)G^T E^T \, ds, \\
K_c^4 &= \int_0^l EH_1^T R_c^T K_{f_c} \, ds,
\end{align}

where the full expression of \(K_{f_c}\) is given in Appendix E. \(\mathcal{F}_c\) is the contact force rigidly-rotated-back to the \(\tilde{Q}\) configuration, i.e.

\[ \mathcal{F}_c = R_c^T f_c \]

and \(\hat{S}\) is the operator that transforms the \(12 \times 1\) array into a \(12 \times 3\) matrix as

\[ \hat{S}(a) = \begin{bmatrix} S(a_1) \\ S(a_2) \\ S(a_3) \\ S(a_4) \end{bmatrix}, \quad a_I = \begin{bmatrix} a_{3(I-1)+1} \\ a_{3(I-1)+2} \\ a_{3(I-1)+3} \end{bmatrix}; \quad I \in \{1, 2, 3, 4\}. \]

Finally, the dynamic contribution is given by

\[ C_c = \int_0^l EH_1^T R_c^T C_{f_c} \, ds, \]

with \(C_{f_c}\) 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 \(g_N(x)\), and its spatial gradient and Hessian, i.e. \(\frac{\partial g_N(x)}{\partial x}, \frac{\partial^2 g_N(x)}{\partial x \partial x^T}\). 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 \(\xi_G\), which is located in the spatial domain at \(x_G \equiv x_0(\xi_G)\), the gap function, first and second derivatives are needed, i.e. from Eq. (106),

\begin{align}
g_N(x_G) &= d_s(x_G) - r, \\
\frac{\partial g_N(x_G)}{\partial x} &= \frac{\partial d_s(x_G)}{\partial x}, \\
\frac{\partial^2 g_N(x_G)}{\partial x \partial x} &= \frac{\partial^2 d_s(x_G)}{\partial x \partial x}. \tag{124}
\end{align}

It is assumed that the SDF is provided in a Cartesian grid in a three dimensional prismatic domain \(\Omega_s = [x_1^s, x_1^o] \times [x_2^s, x_2^o] \times [x_3^s, x_3^o]\), discretized with \(N\) nodes, \(N = N_1N_2N_3\), defined as

\[ x^A = (x_1^{A_1}, x_2^{A_2}, x_3^{A_3})^T; \quad A \equiv (A_1, A_2, A_3), \]

with

\[ x_i^{A_i} = x_i^o + (A_i - 1)h_i; \quad A_i \in \{1, N_i\}; \quad h_i = (x_i^f - x_i^o)/(N_i - 1); \quad i \in 1, 2, 3 \]
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.

\[
d_s(x^A) = d_s^A
\]

(127)

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

\[
d_s(x_G) = \mathcal{I}_A(d_s^A),
\]

(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 \( x \) in space, the signed distance \( d_s(x) \) is interpolated as

\[
d_s = \sum_A V_A d_s^A \varphi(x - x^A),
\]

(129)

where \( V_A = h^3 \) and \( \varphi \) defined as a tensor product of 1D kernels, i.e.

\[
\varphi(z) = \phi \left( \frac{z_1}{h} \right) \phi \left( \frac{z_2}{h} \right) \phi \left( \frac{z_3}{h} \right).
\]

(130)

This allows computing the first and second derivatives of \( d_s \) as

\[
\frac{\partial d_s(x)}{\partial x} = \sum_A V_A d_s^A \frac{\partial \varphi}{\partial x}(x - x^A),
\]

(131)

\[
\frac{\partial^2 d_s(x)}{\partial x \partial x} = \sum_A V_A d_s^A \frac{\partial^2 \varphi}{\partial x \partial x}(x - x^A),
\]

(132)

with

\[
\frac{\partial \varphi}{\partial z_i} = \frac{1}{h} \phi' \left( \frac{z_i}{h} \right) \phi \left( \frac{z_j}{h} \right) \phi \left( \frac{z_k}{h} \right),
\]

(133)

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

(134)

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

(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 \( \varphi \) 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 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 \( \varphi \) are required (see Eq. (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

\[
\phi(r) = \begin{cases} 
0 & r < 0 \\
\frac{29}{7560} r^7 - \frac{5}{72} r^6 - \frac{21}{40} r^5 - \frac{17}{8} r^4 - \frac{39}{8} r^3 - \frac{243}{40} r^2 - \frac{27}{8} r - \frac{81}{280} & r \leq -3 \\
\frac{17}{1512} r^7 + \frac{1}{18} r^6 + \frac{13}{24} r^5 + \frac{7}{12} r^4 + \frac{65}{24} r^3 + \frac{7}{120} r^2 + \frac{13}{72} r + \frac{1447}{2520} & -3 < r \leq -2 \\
\frac{11}{7560} r^7 - \frac{1}{18} r^6 + \frac{7}{36} r^5 - \frac{29}{60} r^4 + \frac{691}{1260} & -2 < r \leq -1 \\
\frac{11}{7560} r^7 - \frac{1}{18} r^6 + \frac{7}{36} r^5 - \frac{29}{60} r^4 + \frac{691}{1260} & -1 < r \leq 0 \\
0 & 0 \leq r \leq 1 \\
\frac{17}{1512} r^7 + \frac{1}{18} r^6 + \frac{13}{24} r^5 + \frac{7}{12} r^4 + \frac{65}{24} r^3 + \frac{7}{120} r^2 - \frac{13}{72} r + \frac{1447}{2520} & 1 < r \leq 2 \\
\frac{29}{7560} r^7 - \frac{5}{72} r^6 + \frac{21}{40} r^5 - \frac{17}{8} r^4 + \frac{39}{8} r^3 - \frac{243}{40} r^2 + \frac{27}{8} r - \frac{81}{280} & 2 < r \leq 3 \\
0 & 3 < r
\end{cases}
\]

(136)

which is designed, by construction, to interpolate exactly linear functions and with second-order accuracy smooth functions [59,67].
To do so, the analytical signed distance function of a sphere is used, which is defined as

$$d_s(x) = \sqrt{(x - P_0) \cdot (x - P_0)} - R_0,$$ \hspace{1cm} (137)

with \( P_0 \) 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 \( P_0 = [0.5 \ 0.5 \ 0.5]^T \) and \( R_0 = 0.3 \).

Five sets of voxel meshes are created in the domain \( \Omega = [-0.5, 1.5] \times [-0.5, 1.5] \times [-0.5, 1.5] \) 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 \( d_s^A = d_s(x^A) \) via equation (137), which creates a SDF (see Fig. 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 \( x_G \). Additionally, the spatial gradient and Hessian are also computed at each of the nodes of this auxiliary mesh. The auxiliary mesh is centered at \( P_0 \), 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 Fig. 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 Eq. (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 \( P_0 \). Therefore, in order to avoid numerical issues, the error analysis is limited to the region in which \( |d_s(x)| \leq 0.8R_0 \).

Fig. 5 shows the L2-norms for the interpolated signed distance, \( d_s^A \) and the computed spatial gradient and Hessian \( \nabla d_s^A, \nabla^2 d_s^A \). It can be seen that all values converge to their analytical counterpart with second-order accuracy, i.e. \( O(h^2) \), 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. \( O(h) \) is expected [67,68]. Importantly, the discrete contact force \( f_c \) in (114) and its spatial derivatives will also approximated at the same order of the distance function \( d_s(x) \) 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.

5. Time integration and computational aspects

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

$$R(D, \dot{D}, \ddot{D}, t) = T_{int}(D, t) + T_x(D, \dot{D}, \ddot{D}, t) - T_c(D, \dot{D}, t) - F_{ext}(D, t) = 0 \hspace{1cm} (138)$$
where $R$ is the residual, $T_{int}$, $T_k$, $T_c$ are, respectively, the assembled vectors of nodal internal, kinetic and contact forces and $F_{ext}$ is the vector of nodal external forces. At the same time, $D$ is the vector of nodal solutions, containing displacements and rotations, and $\dot{D}$, $\ddot{D}$ are respectively, its first and second time derivatives. Following Eq. (69), these are given by

$$D = \begin{bmatrix} u_1 \\ \theta_1 \\ \vdots \\ u_N \\ \theta_N \end{bmatrix}, \quad \Delta D = \begin{bmatrix} \Delta u_1 \\ \Delta \theta_1 \\ \vdots \\ \Delta u_N \\ \Delta \theta_N \end{bmatrix}, \quad \dot{D} = \begin{bmatrix} \dot{u}_1 \\ \dot{\theta}_1 \\ \vdots \\ \dot{u}_N \\ \dot{\theta}_N \end{bmatrix}, \quad \ddot{D} = \begin{bmatrix} \ddot{u}_1 \\ \ddot{\theta}_1 \\ \vdots \\ \ddot{u}_N \\ \ddot{\theta}_N \end{bmatrix},$$

(139)

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

$$R^{n+\alpha} = (1 + \alpha)(T_{int}^{n+1} - T_c^{n+1} - F_{ext}^{n+1}) - \alpha(T_{int}^n - T_c^n - F_{ext}^n) + T_k^{n+1} = 0$$

(140)

with $\alpha$ 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

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

(141)

with $\tilde{K}$ given by

$$\tilde{K} = (1 + \alpha)(K_{int} - K_c - K_{ext}) + \frac{1}{\beta\Delta t^2}M + \frac{\gamma}{\beta\Delta t} (C_k - (1 + \alpha)C_c),$$

(142)

with $\beta = \frac{1}{2}(1 - \alpha)^2$, $\gamma = \frac{1}{2}(1 - 2\alpha)$ and $K_{int}$, $K_c$, $M$, $C_k$, $C_c$ assembled, respectively, from Eqs. (95), (117), (102), (103) and (121). Upon obtaining $\Delta D$, 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, $\epsilon_R$, $\epsilon_D$, i.e. $\|R\| \leq \epsilon_R$, $\|\Delta D\| \leq \epsilon_D$. 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 $\epsilon_R = \epsilon_D = 1 \cdot 10^{-5}$. 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 $X_1\text{--}X_3$ 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 $X = [10 \ 0 \ \pm 0.15]^T$ m (see Fig. 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 Eq. (106). A uniform time-varying body force is applied along the beam, with the value $b = [0 \ 0 \ \sin(10t)]^T$, 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. Fig. 6b shows the isovolume of this SDF.

Fig. 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 Fig. 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]. Fig. 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 centered at $P_0 = [5 \ 0.5 \ 8]^T$ cm and radius $R_0 = 4$ cm. Fig. 10a describes the problem setup while Fig. 10b shows the prescribed displacement of the beam base point. The sphere surface is defined first using an analytical signed distance function, see Eq. (137), and then by using 3 discrete SDF defined in a domain...
Fig. 9. Vibrations of a beam constrained by two end stops, different discretizations. $u_1$ versus $u_3$ displacement of the beam tip.

Fig. 10. Beam against sphere. Initial setup and imposed displacement.

$\Omega_s = [-0.05, 0.15] \times [-0.1, 0.1] \times [-0.05, 0.15]$ of $(20 \times 20 \times 20)$ voxels, $(40 \times 40 \times 40)$ voxels and $(80 \times 80 \times 80)$ voxels (see Fig. 11). The penalty parameter is set to $\varepsilon_c = 5 \cdot 10^4$ and no friction is considered, i.e. $\mu = 0.0$.

Fig. 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 Eq. (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, Fig. 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 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. Fig. 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. Fig. 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 Fig. 14 are due to the adaptive time step strategy described in Section 5, which forces a reduction of the time step when contact occurs.

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 $\omega = [0, 0, 2]$
rad/s and is subjected to gravitational force, which is imposed through a constant body force $b = \rho_0 [0, 0, -9.8]^T$ m/s$^2$. The position of the top end of the cable is set to $X_o = [-0.8, -0.115, 3.0]$ m, which accounts for an initial gap of $5 \cdot 10^{-3}$ m (see Fig. 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 axis as shown in Fig. 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, $\mu = \{0.00, 0.10, 0.30\}$. Fig. 17 shows three snapshots of the solution for $\mu = 0.1$ at different time instants. Fig. 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,

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

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

2 The first component of $X_o$ has been decided by the authors, as it was not detailed in [20,72].
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 $\mu = 0.30$. Fig. 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, Fig. 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, $\frac{\partial g}{\partial x}$.

### 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, $v = 1 \cdot 10^{-4}$ and $\rho_0 = 1 \cdot 10^{-2}$ kg/m$^3$. The ring has an initial uniform speed of $\mathbf{v}_0 = \left[\sqrt{2}, -\sqrt{2}, 0\right]^T$ m/s with its center initially located at $X_0 = [0, 0, 0]^T$ m. A rigid surface is placed at $X_3 = -12$ m (see Fig. 21). 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, $\mu = \{0.00, 0.15, 0.30\}$. Fig. 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, Fig. 23 shows the evolution of the kinetic, internal and total energy for $\mu = 0.15$. It can be seen that, by using the HHT-$\alpha$ method, the solution shows an overall dissipative trend.

---

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).
and is stable up to very long time steps, not experiencing any energy blow up, which is a crucial test in impact problems [75].

**Fig. 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-\(\alpha\) 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 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].
Fig. 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 $\mu = 0.1$.

Fig. 21. Impact of a ring against a rigid surface. Initial setup.

Fig. 22. Impact of a ring against a rigid surface. Comparison of the ring pathlines until $t = 40.0$ s for different friction coefficients.
Fig. 23. Impact of a ring against a rigid surface. Evolution of the kinetic, internal and contact energies for $\mu = 0.15$.

Fig. 24. Impact of a ring against a rigid surface. Comparison of the total energy conservation 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 $11 \times 13$ 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 Fig. 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 $b = [0, -9.8, 0]^T$ 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 $\epsilon_c = 1 \cdot 10^8$ 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 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 $h = 10^{-3}$ m (see Fig. 26). Fig. 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. Fig. 28 highlights the solution at the final time step, by showing different views. In order to assess the penetration, Fig. 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 Fig. 29b. To do so, the centerline position is postprocessed in the same way as in example 6.2, but now the distance $d_s(x)$ 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.
Table 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_A$</td>
<td>$[9.375 \cdot 10^{-2}, 2.55 \cdot 10^{-1}, -1.875 \cdot 10^{-3}]^T$ m</td>
</tr>
<tr>
<td>$X_B$</td>
<td>$[3.8625 \cdot 10^{-1}, 2.55 \cdot 10^{-1}, 241875 \cdot 10^{-1}]^T$ m</td>
</tr>
<tr>
<td>$L_1$</td>
<td>$2.925 \cdot 10^{-1}$ m</td>
</tr>
<tr>
<td>$L_2$</td>
<td>$2.4375 \cdot 10^{-1}$ m</td>
</tr>
</tbody>
</table>

proposed in Section 3. To further strengthen this last point, Fig. 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 $0.4r$ ($0.08h$).

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. $O(h^2)$, for smooth surfaces while first-order convergence, i.e. $O(h)$, 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
Fig. 29. Dropping net on the Stanford Dragon. Stacked area plots showing the percentage of Gauss points and length contacting/penetrating the surface.

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

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-\(\alpha\) 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.

Declaration of competing interest

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

Acknowledgment

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 \( \tilde{\Omega} \), the rotation tensor \( \tilde{R} \) can be approximated as

\[
\tilde{R}(\tilde{\Theta}) \simeq I + S(\tilde{\Theta}) \tag{A.1}
\]

and its first derivative as

\[
\tilde{R}' \simeq S(\tilde{\Theta}') \tag{A.2}
\]

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

\[
u_2'(s) \simeq \hat{\Theta}_3(s), \tag{A.3}
\]

\[
u_3'(s) \simeq -\hat{\Theta}_2(s). \tag{A.4}
\]

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

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

with \( \tilde{a} \) given by (see Eq. (44))

\[
\tilde{a}(s) = \tilde{\gamma}(s) + \xi_\alpha \tilde{\kappa}_\alpha(s), \tag{A.6}
\]

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

\[
\tilde{\gamma} = \tilde{x}_0'(s) - \tilde{l}_1, \tag{A.7}
\]
\( \mathbf{\kappa}_a = \mathbf{\tilde{R}}'(s) \mathbf{\tilde{E}}_a. \) \hspace{1cm} (A.8)

Using Eqs. (34), (35) alongside approximations (A.3), (A.4), \( \mathbf{\tilde{x}}' \) can be written as
\[
\mathbf{\tilde{x}}'_0 = \mathbf{\tilde{E}}_1 + u'_1(s) \mathbf{\tilde{E}}_1 + u'_2(s) \mathbf{\tilde{E}}_2 + u'_3(s) \mathbf{\tilde{E}}_3 \simeq u'_1(s) \mathbf{\tilde{E}}_1 + \mathbf{i}_1, \quad \text{(A.9)}
\]
which yields
\[
\mathbf{\tilde{y}} = u'_1(s) \mathbf{\tilde{E}}_1. \quad \text{(A.10)}
\]

At the same time, via approximation (A.2) and by omitting the high order terms, \( \mathbf{\kappa}_a \) can be rewritten as
\[
\mathbf{\tilde{k}}_2 = (\mathbf{\tilde{\Theta}}'(s) \times \mathbf{\tilde{T}}_2) \simeq \begin{bmatrix} \tilde{\Theta}'_1 \\ \tilde{\Theta}'_2 \\ \tilde{\Theta}'_3 \end{bmatrix}_{\mathbf{E}_i} \times \begin{bmatrix} 1 \\ -\tilde{\Theta}_3 \end{bmatrix}_{\mathbf{E}_i} = \begin{bmatrix} \tilde{\Theta}'_2 \tilde{\Theta}_1 - \tilde{\Theta}'_3 \\ -\tilde{\Theta}_3 \tilde{\Theta}'_1 - \tilde{\Theta}'_3 \tilde{\Theta}_1 \\ \tilde{\Theta}'_1 - \tilde{\Theta}_2 \tilde{\Theta}'_2 \end{bmatrix}_{\mathbf{E}_i} \simeq \begin{bmatrix} -\tilde{\Theta}'_3 \\ \tilde{\Theta}'_1 \\ 0 \end{bmatrix}_{\mathbf{E}_i}. \quad \text{(A.11)}
\]

Using the results in Eqs. (A.10), (A.11), (A.12) into (A.6) finally yields
\[
\mathbf{\bar{a}} = \begin{bmatrix} u'_1 - \xi_2 \tilde{\Theta}'_3 + \xi_3 \tilde{\Theta}'_2 \\ -\xi_3 \tilde{\Theta}'_1 \\ \xi_2 \tilde{\Theta}'_1 \end{bmatrix}_{\mathbf{E}_i}. \quad \text{(A.13)}
\]

On the other hand, the displacement \( \mathbf{\tilde{u}} \) can be approximated assuming small rotations by replacing approximation (A.1) into Eq. (35), to yield
\[
\mathbf{\tilde{u}}(s) = \begin{bmatrix} u_1(s) \\ u_2(s) \\ u_3(s) \end{bmatrix}_{\mathbf{E}_i} = \begin{bmatrix} -\tilde{\Theta}_1 \\ 1 \\ -\tilde{\Theta}_1 \end{bmatrix}_{\mathbf{E}_i} + \xi_2 \begin{bmatrix} \tilde{\Theta}'_2 \\ -\tilde{\Theta}_3 \\ -\tilde{\Theta}_1 \end{bmatrix}_{\mathbf{E}_i} = \begin{bmatrix} u_1(s) - \xi_2 \tilde{\Theta}_3(s) + \xi_3 \tilde{\Theta}_2(s) \\ u_2(s) + \xi_2 - \xi_3 \tilde{\Theta}_1(s) \\ u_3(s) + \xi_2 \tilde{\Theta}_1(s) + \xi_3 \end{bmatrix}_{\mathbf{E}_i}, \quad \text{(A.14)}
\]
and the gradient can be obtained by rewriting that, in the configuration \( \mathbf{\tilde{\Omega}}, \mathbf{\tilde{X}}_1 = s, \mathbf{\tilde{X}}_a = \xi_a \), which yields
\[
\nabla \mathbf{\tilde{u}} = \frac{\partial \mathbf{\tilde{u}}}{\partial \mathbf{X}} \simeq \begin{bmatrix} u'_1(s) - \xi_2 \tilde{\Theta}'_3(s) + \xi_3 \tilde{\Theta}'_2(s) - \tilde{\Theta}_3(s) \tilde{\Theta}'_2(s) \\ \tilde{\Theta}_3(s) - \xi_3 \tilde{\Theta}'_1(s) \\ -\tilde{\Theta}_2(s) + \xi_2 \tilde{\Theta}'_1(s) \tilde{\Theta}_1(s) \\ \end{bmatrix}_{\mathbf{E}_i} \quad \text{(A.15)}
\]
where, again, the small displacement assumption (A.3), (A.4) has been used. Finally, comparing the result above with Eq. (A.13), yields
\[
\nabla \mathbf{\tilde{u}} \simeq S(\tilde{\Theta}) + \mathbf{\bar{a}} \otimes \mathbf{E}_1. \quad \text{(A.16)}
\]

The above gradient can be approximated using a sum of symmetric and antisymmetric tensors
\[
\nabla \mathbf{\tilde{u}} = \mathbf{\varepsilon}(\nabla \mathbf{\tilde{u}}) + \mathbf{\Omega}(\nabla \mathbf{\tilde{u}}), \quad \text{(A.17)}
\]
with
\[
\mathbf{\varepsilon}(\nabla \mathbf{\tilde{u}}) = \frac{1}{2} \begin{bmatrix} 2 \left( u'_1(s) - \xi_2 \tilde{\Theta}'_3(s) + \xi_3 \tilde{\Theta}'_2(s) \right) & -\xi_3 \tilde{\Theta}'_1(s) & \xi_2 \tilde{\Theta}'_1(s) \\ -\xi_3 \tilde{\Theta}'_1(s) & 0 & 0 \\ \xi_2 \tilde{\Theta}'_1(s) & 0 & 0 \end{bmatrix} = \frac{1}{2} \left( \mathbf{\bar{a}} \otimes \mathbf{E}_1 + \mathbf{E}_1 \otimes \mathbf{\bar{a}} \right). \quad \text{(A.18)}
\]

\[
\mathbf{\Omega}(\nabla \mathbf{\tilde{u}}) = \frac{1}{2} \begin{bmatrix} 0 & -2 \tilde{\Theta}_3 + \xi_3 \tilde{\Theta}'_1 & 2 \tilde{\Theta}_2 - \xi_2 \tilde{\Theta}'_1 \\ -2 \tilde{\Theta}_3 + \xi_3 \tilde{\Theta}'_1 & 0 & -2 \tilde{\Theta}_1 \\ 2 \tilde{\Theta}_2 - \xi_2 \tilde{\Theta}'_1 & 2 \tilde{\Theta}_1 & 0 \end{bmatrix} = S(\tilde{\Theta}) + \frac{1}{2} \left( \mathbf{\bar{a}} \otimes \mathbf{E}_1 - \mathbf{E}_1 \otimes \mathbf{\bar{a}} \right). \quad \text{(A.19)}
\]
Appendix B. Isoparametric Interpolation

The Finite Element interpolation matrices are

\[
P_1(\xi) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & N_3(\xi) & 0 & 0 & N_4(\xi) \\
0 & -N_3(\xi) & 0 & 0 & -N_4(\xi) & 0
\end{bmatrix}
\]

\[
P_2(\xi) = \begin{bmatrix}
N_1(\xi) & 0 & 0 & N_2(\xi) & 0 & 0 \\
0 & N_3(\xi) & 0 & 0 & N_4(\xi) & 0 \\
0 & 0 & N_5(\xi) & 0 & 0 & N_6(\xi)
\end{bmatrix}
\]

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

\[
N_1(\xi) = 1 - \frac{\xi}{l_0}, \quad N_2(\xi) = 1 - N_1, \\
N_3(\xi) = \xi \left(1 - \frac{\xi}{l_0}\right)^2, \quad N_4(\xi) = -\left(1 - \frac{\xi}{l_0}\right) \frac{\xi^2}{l_0}, \\
N_5(\xi) = \left(1 - \frac{3\xi}{l_0}\right) \left(1 - \frac{\xi}{l_0}\right), \quad N_6(\xi) = \left(\frac{3\xi}{l_0} - 2\right) \frac{\xi}{l_0}.
\]

Appendix C. Internal Force Vector, Stiffness Matrix

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

\[
\bar{D}_s = \begin{bmatrix}
\bar{u} & \bar{W}_1^T & \bar{W}_2^T
\end{bmatrix}^T; \quad \delta \bar{D}_s = \begin{bmatrix}
\delta \bar{u} & \delta \bar{W}_1^T & \delta \bar{W}_2^T
\end{bmatrix}^T.
\]

The relationship between \( \delta \bar{D} \) and \( \delta \bar{D}_s \) given via equation (14a) as

\[
\delta \bar{D} = \bar{B} \delta \bar{D}_s; \quad \bar{B} = \begin{bmatrix}
1 & 0^T & 0^T \\
0 & T_s^{-1}(\bar{\Omega}_1) & 0_{3\times3} \\
0 & 0_{3\times3} & T_s^{-1}(\bar{\Omega}_2)
\end{bmatrix}.
\]

Next, the relationship between \( \delta \bar{D}_s \) and \( \delta d \) is given by

\[
\delta \bar{D}_s = \bar{B}_s \delta d, \quad \bar{B}_s = \begin{bmatrix}
\frac{\mathbf{r}}{P E^T}
\end{bmatrix},
\]

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

\[
\delta \bar{u} = \mathbf{r} \delta d; \quad \mathbf{r} = \begin{bmatrix}
-v_1^T & 0^T & v_1^T & 0^T
\end{bmatrix}.
\]

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

\[
\begin{bmatrix}
\delta \bar{W}_1 \\
\delta \bar{W}_2
\end{bmatrix} = \begin{bmatrix}
P\delta \bar{D} = P E^T \delta d; \quad P = \begin{bmatrix}
0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\
0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3}
\end{bmatrix} - \begin{bmatrix}
G^T \\
G^T
\end{bmatrix},
\]

and

\[
\frac{\partial \bar{W}_s}{\partial \bar{D}} = \begin{bmatrix}
0 & 0 & \eta/\kappa & \eta_{12}/2 & -\eta_{11}/2 & 0 & 0 & 0 & -\eta/\kappa & \eta_{22}/2 & -\eta_{21}/2 & 0 \\
0 & 0 & 1/\kappa & 0 & 0 & 0 & 0 & -1/\kappa & 0 & 0 & 0 & 0 \\
0 & -1/\kappa & 0 & 0 & 0 & 0 & 0 & 1/\kappa & 0 & 0 & 0 & 0
\end{bmatrix}
\]

with

\[
\eta = \frac{q_1}{q_2}, \quad \eta_{11} = \frac{q_{11}}{q_2}, \quad \eta_{12} = \frac{q_{12}}{q_2}, \quad \eta_{21} = \frac{q_{21}}{q_2}, \quad \eta_{22} = \frac{q_{22}}{q_2}.
\]
Combining Eqs. (C.2) and (C.3) yields the relationship between $\delta d$ and $\delta \mathbf{D}$

$$
\delta d = B \delta \mathbf{D}; \quad B = \mathbf{B} B_s.
$$

(C.8)

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

$$
\delta \mathbf{T}_{int} = \delta (B^T \mathbf{T}_{int}) = \delta (B_s^T \mathbf{B}^T \mathbf{T}_{int}) = K_{int} \delta d,
$$

(C.9)

with

$$
K_{int} = B^T \mathbf{K}_{int} B + \mathbf{K}; \quad \mathbf{K} = \mathbf{B}_s^T \mathbf{K}_h \mathbf{B}_s + K_m
$$

(C.10)

with $\mathbf{K}$ given in Eq. (93) and $K_h$ given by

$$
\mathbf{K}_h = \frac{\partial B^T \mathbf{T}_{int}}{\partial \mathbf{D}_s} \bigg|_{\mathbf{T}_{int}} = \begin{bmatrix} 0 & 0^T & 0^T \\ 0 & \mathbf{K}_{h1} & 0_{3 \times 3} \\ 0 & 0_{3 \times 3} & \mathbf{K}_{h2} \end{bmatrix}; \quad \mathbf{K}_{ha} = \frac{\partial \mathbf{T}_s^T (\hat{\Theta}_a) \mathbf{M}_a}{\partial W_a} \bigg|_{\mathbf{M}_a},
$$

(C.11)

\begin{align*}
\mathbf{K}_{ha} &= \eta_a \left( \hat{\Theta}_a \mathbf{M}_a^T - 2 \mathbf{M}_a \hat{\Theta}_a^T + (\hat{\Theta}_a \cdot \mathbf{M}_a) I_{3 \times 3} \right) T_s^{-1}(\hat{\Theta}_a) \\
&\quad + \mu_a \left( S(\hat{\Theta}_a) S(\hat{\Theta}_a) \mathbf{M}_a \hat{\Theta}_a^T \right) T_s^{-1}(\hat{\Theta}_a) \\
&\quad - \frac{1}{2} S(\mathbf{M}_a) T_s^{-1}(\hat{\Theta}_a)
\end{align*}

(C.12)

and

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

(C.13)

Finally $K_m$ is given by

$$
K_m = D \tilde{N} - E \mathbf{Q}_G \mathbf{Q}^T E^T + E \mathbf{G} \mathbf{a} r,
$$

(C.14)

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

$$
D = \begin{bmatrix} D_3 & 0_{3 \times 3} & -D_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & D_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ -D_3 & 0_{3 \times 3} & D_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}; \quad D_3 = \frac{1}{l_n}(I_{3 \times 3} - v_1 v_1^T),
$$

(C.15)

$$
\mathbf{Q} = \begin{bmatrix} S(Q_1) \\ S(Q_2) \\ S(Q_3) \\ S(Q_4) \end{bmatrix}, \quad a = \begin{bmatrix} 0 \\ \eta((\mathbf{M}_{s1})_1 + [\mathbf{M}_{s2}])/(l_n - ((\mathbf{M}_{s1})_2 + [\mathbf{M}_{s2}])/(l_n) \\ ((\mathbf{M}_{s1})_3 + [\mathbf{M}_{s2}])/(l_n) \end{bmatrix}.
$$

(C.16)

In the above definition of $\mathbf{Q}$, $Q_i$ are the block components of

$$
\mathbf{P}^T \mathbf{m} = \begin{bmatrix} Q_1^T & Q_2^T & Q_3^T & Q_4^T \end{bmatrix}; \quad \mathbf{m} = \begin{bmatrix} \mathbf{M}_{s1}^T & \mathbf{M}_{s2}^T \end{bmatrix}.
$$

(C.17)

with $\mathbf{M}_{s1}$ being the nodal moments related to the displacements $\mathbf{D}_s$,

$$
\tilde{T}_s = \frac{\partial W_{int}}{\partial \mathbf{D}_s} = \mathbf{B}_s \mathbf{T}_{int} = \begin{bmatrix} \tilde{N} & \mathbf{M}_{s1}^T & \mathbf{M}_{s2}^T \end{bmatrix}^T.
$$

(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 $u_0^h$ and their time derivatives, as well as the variations of the centerline spin vectors $w_0^h$ and their time derivatives are needed. For completeness, the main results of [40] are summarized, but for more details the reader is referred to that paper. The variation of the centerline displacement, $\delta u_0^h$ is obtained via equation (80),

$$
\delta u_0^h = \delta u_1 N_1(\xi) + \delta u_2 N_2(\xi) + \delta R u^i(\xi) + R_c \delta u^i(\xi).
$$

(D.1)
The variation $\delta R_e$ is obtained via equations (61), (64) as
\[
\delta R_e = R_e S(\delta W_e),
\] (D.2)
with (see Eqs. (70), (C.6)),
\[
\delta W_e = \frac{\partial W_e}{\partial \mathbf{D}} \delta \mathbf{D} = G^T E^T \delta \mathbf{d}.
\] (D.3)
The variation of $u'$ is given by
\[
\delta u' = P_1 \begin{bmatrix} \delta \bar{\Theta}_1 \\ \delta \bar{\Theta}_2 \end{bmatrix} \simeq P_1 \begin{bmatrix} \delta \bar{W}_1 \\ \delta \bar{W}_2 \end{bmatrix} = P_1 P E^T \delta \mathbf{d},
\] (D.4)
where Eq. (C.5) has been used and where, as in [40], it has been assumed that $T_s(\bar{\Theta}_a) \simeq I$ due to small rotations (see Eq. (14b)). Replacing the above results into (D.1),
\[
\delta u_h^0 = R_e H_1 E^T \delta \mathbf{d},
\] (D.5)
with
\[
H_1(\xi) = N(\xi) + P_1(\xi) P - S(u'(\xi))G^T.
\] (D.6)
with
\[
N(\xi) = \begin{bmatrix} N_1(\xi) I & 0 & N_2(\xi) I & 0 \end{bmatrix}.
\] (D.7)
The variation of $H_1$ is useful to compute second derivatives and also to obtain the tangent dynamic matrices. This is given by
\[
\delta H_1 = \frac{N_7}{l^2_n} A_1 r \delta \mathbf{d} - S(\delta u')G^T,
\] (D.8)
with
\[
N_7 = N_3 + N_4,
\] (D.9)
\[
A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.
\] (D.10)
Via equation (D.5), the first and second derivatives of $u_h^0$ can be computed as
\[
\dot{u}_0^h = R_e H_1 E^T \dot{\mathbf{d}},
\] (D.11)
\[
\ddot{u}_0^h = R_e H_1 E^T \ddot{\mathbf{d}} + R_e C_1 E^T \dot{\mathbf{d}},
\] (D.12)
with
\[
C_1 = S(\dot{W}_e) H_1 + \dot{H}_1 - H_1 E
\] (D.13)
and $\dot{H}_1$ obtained from Eqs. (D.8), (D.4) as
\[
\dot{\dot{H}}_1 = \frac{N_7}{l^2_n} A_1 r \dot{\mathbf{d}} - S(\dot{u'})G^T; \quad \dot{u'} = P_1 P E^T \dot{\mathbf{d}}.
\] (D.14)
Regarding the rotation variables, the variation of $w_h^0$ is obtained from Eq. (84) and doing a rigid-push-forward with $R_e$
\[
\delta w_h^0 = R_e H_2 E^T \delta \mathbf{d},
\] (D.15)
with
\[
H_2(s) = P_2(s) P + G^T.
\] (D.16)
As for $H_1$, it is useful to obtain the variation of $H_2$. This is given by
\[
\delta H_2 = \frac{N_8}{l^2_n} A_2 r \delta \mathbf{d},
\] (D.17)
considering only the contributions from the acceleration and velocity, this is, the centrifugal and dynamic matrices. The following variables require linearization:

\[ \dot{\omega}_0^h = R_c H_2 E^T \ddot{d}, \]
\[ \ddot{\omega}_0^h = R_c H_2 E^T \dddot{d} + R_c C_2 E^T \dddot{d}, \]

with

\[ C_2 = S(\dot{W}_c) H_2 + \dot{H}_2 - H_2 E. \]

In order to obtain the tangent matrix, the variation of \( T_k \) in Eq. (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:

\[ \delta T_k \approx M \delta \ddot{d} + C_\delta \delta \dot{d} \]

The following variables require linearization:

\[ \tilde{U}_0^h = R_c^T \ddot{u}_0^h = H_1(d)E(d)^T \ddot{d} + C_1(d, \dot{d})E(d)^T \dddot{d}, \]
\[ \tilde{W}_0^h = R_c^T \dddot{w}_0^h = H_2(d)E(d)^T \ddot{d} + C_2(d, \dot{d})E(d)^T \dddot{d}, \]
\[ \tilde{W}_0^h = R_c^T \dddot{w}_0^h = H_2(d)E(d)^T \dddot{d}, \]

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

\[ \delta \tilde{U}_0^h \approx H_1 E^T \delta \ddot{d} + (C_1 + C_3)E^T \delta \dot{d}, \]
\[ \delta \tilde{W}_0^h \approx H_2 E^T \delta \ddot{d} + (C_2 + C_4)E^T \delta \dot{d}, \]
\[ \delta \tilde{W}_0^h \approx H_2(d)E(d)^T \delta \dot{d}. \]

In the above equations, \( C_3 \) and \( C_4 \) are defined as

\[ C_3 = -S(h_1)G^T + \frac{N_t}{l_n^2} A_1 \dot{\mathcal{D}} r E + S(\dot{W}_c) P_1 P + H_1 F_1 G^T, \]
\[ C_4 = -S(h_2)G^T + \frac{N_s}{l_n^2} A_2 \dot{\mathcal{D}} r E + H_2 F_1 G^T, \]

with \( \dot{\mathcal{D}} \) taken from Eq. (70) as

\[ \dot{\mathcal{D}} = E^T \ddot{d} = \begin{bmatrix} \dot{U}_1^T & \dot{W}_1^T & \dot{U}_2^T & \dot{W}_2^T \end{bmatrix}^T. \]

\( h_1, h_2 \) defined as

\[ h_1 = H_1 \dot{\mathcal{D}}, \]
\[ h_2 = H_2 \dot{\mathcal{D}}, \]

and, finally, \( F_1 \) given by

\[ F_1 = \begin{bmatrix} S(\dot{U}_1) \\ S(\dot{W}_1) \\ S(\dot{U}_2) \\ S(\dot{W}_2) \end{bmatrix}. \]
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 described in Eqs. (116), (117), (118) and (121). To start with, the contact force vector defined in Eq. (113) is repeated here for convenience,

\[ T_c = \int_0^{l_0} E H_1^T R_1^T f_c \, ds, \]  

(E.1)

with \( f_c \) given by

\[ f_c = p_N(d)\mathcal{G}(d; \dot{d}); \quad \mathcal{G}(x; d; \dot{d}) = \frac{\partial g_N}{\partial x} - \mu \frac{\dot{g}_b^b}{\sqrt{\|\dot{g}_b^b\|^2 + \varepsilon}}. \]  

(E.2)

\[ \delta T_c = K_c \delta d + C_c \delta \dot{d}; \quad K_c = K_1^c + K_2^c + K_3^c + K_4^c. \]  

(E.3)

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

\[ K_1^c \delta d = \int_0^{l_0} t_1^c \, ds, \]  

(E.4a)

\[ K_2^c \delta d = \int_0^{l_0} t_2^c \, ds, \]  

(E.4b)

\[ K_3^c \delta d = \int_0^{l_0} t_3^c \, ds, \]  

(E.4c)

\[ K_4^c \delta d + C_c \delta \dot{d} = \int_0^{l_0} t_4^c \, ds, \]  

(E.4d)

with

\[ t_1^c = \delta E H_1^T R_1^T f_c, \]  

(E.5a)

\[ t_2^c = E(\delta H_1)^T R_1^T f_c, \]  

(E.5b)

\[ t_3^c = E H_1^T (\delta R_1)^T f_c, \]  

(E.5c)

\[ t_4^c = E H_1^T R_1^T \delta f_c. \]  

(E.5d)

Before moving forward with the above linearizations, it is useful to provide some preliminary results. Firstly, \( f_c \) is rigidly rotated back to give

\[ \mathcal{F}_c = R_1^T f_c. \]  

(E.6)

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

\[ A = \hat{S}(a) = \begin{bmatrix} S(a_1) \\ S(a_2) \\ S(a_3) \\ S(a_4) \end{bmatrix}; \quad a_I = \begin{bmatrix} a_{3(I-1)+1} \\ a_{3(I-1)+2} \\ a_{3(I-1)+3} \end{bmatrix}; \quad I \in \{1, 2, 3, 4\}, \]  

(E.7)

with the operator \( S \) as defined in Eq. (4).

In order to obtain \( t_1^c \), the variation \( \delta E \) is needed. This can be obtained from its definition (70) and Eq. (D.2), to give

\[ \delta E = E \begin{bmatrix} S(\delta W_e) & 0 & 0 & 0 \\ 0 & S(\delta W_e) & 0 & 0 \\ 0 & 0 & S(\delta W_e) & 0 \\ 0 & 0 & 0 & S(\delta W_e) \end{bmatrix}. \]  

(E.8)

And, in order to obtain \( t_4^c \), the variation of \( \delta f_c \) is needed. From Eq. (E.2) this can be obtained as

\[ \delta f_c = \delta p_N \mathcal{G} + p_N \delta \mathcal{G}, \]  

(E.9)
The variation of $\delta$ with $\delta p_N$ and $\delta \mathcal{G}$. The variation of $p_N$ depends only on $\delta d$ and is given as

$$
\delta p_N = p'_N(g_N)\delta g_N = p'_N(g_N) \frac{\partial g_N}{\partial x} \cdot (R_e H_1 E^T \delta d),
$$

(E.10)

with $p'_N$ obtained from (111)

$$
p'_N = p'_N(g_N) = \begin{cases} \frac{-\varepsilon_c}{2} g_N - \varepsilon_c, & g \leq 0 \\
\frac{\varepsilon_c g_N - \bar{p}_N}{g_N^2} g_N - \varepsilon_c, & 0 < g_N \leq \bar{g}_N \\
0, & g > \bar{g}_N
\end{cases}
$$

(E.11)

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

$$
\delta \mathcal{G} = \mathcal{A} \delta d + \mathcal{B} \delta \dot{d}; \quad \mathcal{A} = \mathcal{A}^1 + \mathcal{A}^2 + \mathcal{A}^3 + \mathcal{A}^4,
$$

(E.13)

with

$$
\mathcal{A}^1 = \left( \frac{\partial g_N}{\partial x} \cdot (R_e H_1 \mathcal{D}) \right) \frac{\partial^2 g_N}{\partial x \partial x} R_e H_1 E^T
$$

(E.14a)

$$
\mathcal{A}^2 = \left( I - \frac{\partial g_N}{\partial x} \otimes \frac{\partial g_N}{\partial x} \right) R_e S(H_1 \mathcal{D}) G^T E^T,
$$

(E.14b)

$$
\mathcal{A}^3 = \left( I - \frac{\partial g_N}{\partial x} \otimes \frac{\partial g_N}{\partial x} \right) \left( \frac{N_1}{T_1^2} R_e A_1 \mathcal{D} \mathcal{D} r + R_e S(G^T \mathcal{D}) P_e P_e^T \right),
$$

(E.14c)

$$
\mathcal{A}^4 = \left( I - \frac{\partial g_N}{\partial x} \otimes \frac{\partial g_N}{\partial x} \right) R_e H_1 \dot{S}(\mathcal{D}) G^T E^T,
$$

(E.14d)

$$
\mathcal{B} = \left( I - \frac{\partial g_N}{\partial x} \otimes \frac{\partial g_N}{\partial x} \right) R_e H_1 E^T.
$$

(E.14e)

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

$$
\delta f_c = K_f e \delta d + C_f \delta \dot{d}.
$$

(E.15)

with

$$
K_f = p'_N(g_N) \left( \mathcal{G} \otimes (E H_1^T R_e^T \frac{\partial g_N}{\partial x}) \right)
$$

$$
+ p_N \left( \frac{\partial^2 g_N}{\partial x \partial x} R_e H_1(s) E^T - \frac{\mu}{\sqrt{\|g_N^h\|^2 + \varepsilon}} \left( I - \frac{1}{\|g_N^h\|^2 + \varepsilon} (g_N^h \otimes g_N^h) \right) \mathcal{A} \right),
$$

(E.16)

$$
C_f = -p_N \frac{\mu}{\sqrt{\|g_N^h\|^2 + \varepsilon}} \left( I - \frac{1}{\|g_N^h\|^2 + \varepsilon} (g_N^h \otimes g_N^h) \right) \mathcal{B}.
$$

(E.17)
Using (E.6), (E.7), (E.8), (E.15) alongside (D.3), (D.4), (D.8) into (E.5) gives

\[
t^1_c = \delta EH^T_c \mathbf{F}_c = E \begin{bmatrix} S(\delta W_c) \left[ H^T_1 \mathbf{F}_c \right]_{1:3} \\ S(\delta W_c) \left[ H^T_1 \mathbf{F}_c \right]_{4:6} \\ S(\delta W_c) \left[ H^T_1 \mathbf{F}_c \right]_{7:9} \\ S(\delta W_c) \left[ H^T_1 \mathbf{F}_c \right]_{10:12} \end{bmatrix} \]

\[
= -E \hat{S}(H^T_1 \mathbf{F}_c) \delta W_c = -E \hat{S}(H^T_1 \mathbf{F}_c) G^T E^T \delta d, \tag{E.18a}
\]

\[
t^2_c = E \delta H^T_1 \mathbf{F}_c = E \left( \frac{N_T}{T_n} A_1 r \delta d - S(\delta u_i')G^T \right)^T \mathbf{F}_c
\]

\[
= \frac{N_T}{T_n} E A^T_1 \mathbf{F}_c (r \delta d) + E G S(\delta u_i') \mathbf{F}_c
\]

\[
= \frac{N_T}{T_n} E A^T_1 \mathbf{F}_c (r \delta d) - E G S(\mathbf{F}_c) P_1 P E^T \delta d, \tag{E.18b}
\]

\[
t^3_c = EH^T_1 (\delta R_c^T) f_c = EH^T_1 \left( R_c S(\delta W_c) \right)^T f_c = -EH^T_1 S(\delta W_c) \mathbf{F}_c
\]

\[
= EH^T_1 S(\mathbf{F}_c) \delta W_c = EH^T_1 S(\mathbf{F}_c) G^T E^T \delta d, \tag{E.18c}
\]

\[
t^4_c = EH^T_1 R_c^T \delta f_c = EH^T_1 R_c^T (K_f c \delta d + C_f c \delta d). \tag{E.18d}
\]

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

\[
K^1_c = - \int_0^{l_0} E \hat{S}(H^T_1 \mathbf{F}_c) G^T E^T ds, \tag{E.19}
\]

\[
K^2_c = \int_0^{l_0} \left( \frac{N_T}{T_n} E A^T_1 \mathbf{F}_c r - E G S(\mathbf{F}_c) P_1 P E^T \right) ds, \tag{E.20}
\]

\[
K^3_c = \int_0^{l_0} EH^T_1 S(\mathbf{F}_c) G^T E^T ds, \tag{E.21}
\]

\[
K^4_c = \int_0^{l_0} EH^T_1 R_c^T K_f c ds, \tag{E.22}
\]

\[
C_c = \int_0^{l_0} EH^T_1 R_c^T C_f c ds. \tag{E.23}
\]

Appendix F. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cma.2020.113275.


