



UNIVERSITÀ DI PARMA

ARCHIVIO DELLA RICERCA

University of Parma Research Repository

A geometrically exact isogeometric beam for large displacements and contacts

This is the peer reviewed version of the following article:

Original

A geometrically exact isogeometric beam for large displacements and contacts / Tasora, Alessandro; Benatti, Simone; Mangoni, Dario; Garziera, Rinaldo. - In: COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING. - ISSN 0045-7825. - 358:(2020), p. 112635. [10.1016/j.cma.2019.112635]

Availability:

This version is available at: 11381/2863527 since: 2019-09-30T13:49:05Z

Publisher:

Elsevier B.V.

Published

DOI:10.1016/j.cma.2019.112635

Terms of use:

Anyone can freely access the full text of works made available as "Open Access". Works made available

Publisher copyright

note finali coverpage

(Article begins on next page)

A geometrically exact isogeometric beam for large displacements and contacts

Alessandro Tasora, Simone Benatti, Dario Mangoni, Rinaldo Garziera

*Università degli Studi di Parma,
Dpt. of Engineering and Architecture
Parco Area delle Scienze, 181/A, 43124 Parma, Italy*

Abstract

This work discusses an efficient formulation of a geometrically exact three-dimensional beam which can be used in dynamical simulations involving large displacements, collisions and non-linear materials. To this end, we base our model on the shear-flexible Cosserat rod theory and we implement it in the context of Isogeometric Analysis (IGA). According to the IGA approach, the centerline of the beam is parameterized using splines; in our work the rotation of the section is parameterized by a spline interpolation of quaternions, and time integration of rotations is performed using the exponential map of quaternions. Aiming at an efficient and robust simulation of contacts, we propose the adoption of a non-smooth dynamics formulation based on differential-variational inequalities. The model has been implemented in an open-source physics simulation library that can simulate actuators, finite elements, rigid bodies, constraints, collisions and frictional contacts. This beam model has been tested on various benchmarks in order to assess its validity in non-linear static and dynamic analysis; in all cases the model behaved consistently with theoretical results and experimental data.

Keywords: IGA, beam, contact, multibody, non-smooth dynamics

Email addresses: alessandro.tasora@unipr.it (Alessandro Tasora),
simone.benatti@studenti.unipr.it (Simone Benatti), dario.mangoni@unipr.it (Dario Mangoni), rinaldo.garziera@unipr.it (Rinaldo Garziera)

1. Introduction

Deformable three-dimensional beams withstanding large displacements can be found in many scenarios of practical interest, this is the case of the blades in a helicopter rotor, for instance, or the case of the torsion bar in a car suspension, or the case of a flexible robot arm. In the last decades, similar problems motivated extensive research in the area of fast, robust and reliable computer methods for the simulation of beams, and most of those methods are based on Finite Element (FE) discretizations.

Recently, the Isogeometric Analysis (IGA) computational approach gained increased popularity as it blends the concept of spline interpolation in the context of FE. While traditional FE methods discretize the continuum using finite elements that share end nodes, the IGA approach uses a single spline with several nodes. Indeed, nowadays most CAD tools are based on splines of NURBS or B-spline type, hence one of the relevant advantages of IGA is that these geometries can be imported directly in the simulation, whereas FE methods require an intermediate pre-processing step in order to generate a mesh approximation of the original geometry [29]. Moreover, conventional finite elements using Lagrange polynomials feature C^0 continuity at nodal points regardless of their order, while spline functions of order p provide C^{p-1} continuity at all non-multiple knots. This means that for the same amount of degrees of freedom, IGA shows better robustness and accuracy compared to C^0 -continuous FE [16].

Although the most obvious application of IGA is in the modeling of structural elements which can map to a line parameterization, such as cables and beams, the same concept can be extended to generic PDEs involving membranes, shells and volumes [17, 11]. Among the vast literature on IGA, we cite in passing the developments on hierarchical refinement [60], the generalization to T-splines [9], the application to fluid dynamics [8] and to contact problems [20, 58].

In this paper we use IGA to implement a geometrically exact three-dimensional beam based on the Cosserat rod theory, hence capable of arbitrary

large rotations and displacements [33]. Non-linear geometric effects in beams have been studied extensively in computational mechanics and different approaches have been put forward: for example a straightforward method still used nowadays is based on reusing linear finite elements developed for infinitesimal-strain conventional beams of Euler-Bernoulli or Timoshenko type, by updating a local corotational reference frame that can have large displacements and rotations [18, 23]. In contrast to this, but at the cost of a more sophisticated formulation, the *geometrically exact beam model*, also referred to as Simo-Reissner beam model, draws on the theory of 1D Cosserat continua [15] and leads to a general formulation that makes no assumption on the amount of rotations and that allows finite strains, including shear and torsion [53, 4, 54]. The Cosserat rod theory does not require that cross-sections must be orthogonal to the tangent of the centerline, hence it can account for shear effects similarly to the Timoshenko beam theory, of whom it can be considered a generalization to the 3D case [59]. In fact, Reissner, Kirchhoff-Love, Timoshenko and Euler-Bernoulli beams can be interpreted as special cases of Cosserat rods.

Many references on IGA-based beams can be found in literature: for instance IGA for Kirchhoff-Love and Euler-Bernoulli beams are discussed in [26] and [63], straight and curved 2D Timoshenko beams are discussed in [12], to name a few. Similarly to FE beams, numerical locking artifacts can affect IGA beams when shear is taken into account. In this context, a collocation approach was used in [19, 5] to obtain a locking-free spatial IGA beam.

Contact between three-dimensional rods has been dealt in literature by various authors. Among the first results, in [65] a frictionless point-wise contact formulation was developed between pairs of cables with circular section, later extended to the frictional case [66]. The problem of contact between beams of rectangular section, that might generate multiple contact points between the edges, has been discussed in [39, 38]. The issue of self-contact between beams has been studied in [14, 24], a topic of great importance if one needs to simulate problems such as the tightening of knots, knitting machines or winches. Most contact models in literature are based on point-based formulations where the

position of contact points is obtained by solving local sub-problems of distance minimization between curved geometries [36]. However this approach is near singular when beams intersect with small contact angles: in [45, 46] an efficient method has been proposed that can solve this issue. In [62] the simulation of frictional contact and self collision is discussed for a class of efficient IGA-based Cosserat rods.

Similarly to [64], we address the case of a three-dimensional Cosserat rod of high generality, hence accounting for shear, torsion, geometric and material non-linearity. As an alternative to penalty-based beam contact formulations as presented in [61], we express it with a formalism that fits well in a time stepper for non-smooth multibody dynamics, using the theory of Differential Variational Inequality (DVI). To our knowledge, this is the first time an IGA formulation has been used in the framework of DVI non-smooth dynamics; the benefit being the fact that DVI formulations provide a robust and stable way to simulate contact problems even with large time-steps and many simultaneous contacts.

Early work on non-smooth dynamics can be traced back to the seminal research of Jean-Jacques Moreau on measure-differential inclusions, a special type of DVI [47, 32]. The non-smooth nature of dynamical problems originates from various phenomena, most often it is a consequence of the Coulomb friction model and impulsive collisions at contact points; instead than regularizing the contact forces -a conventional method that leads to smooth but stiff differential problems- Moreau proposed to embed such set-valued force laws directly in the formulation. This can be done at the cost of assuming that velocities can be discontinuous, hence accelerations are considered as distributions of vector signed measures. Since then, different authors contributed to the field of non-smooth dynamics, see for example [56, 3, 52]. The robustness and stability of DVI methods motivated their use in scenarios that involve millions of simultaneous contacts, such as granular dynamics [50], or in contexts that require real-time performance and large time steps, such as robotics and virtual reality [10]. Most time stepping schemes for DVI require the solution of a variational inequality (VI) per each time step: early approaches were based on fixed

point iterations that both represented a major computational bottleneck and performed badly in presence of finite elements, but recent researches address both issues and stimulate novel applications of DVI methods also to flexible structures [44, 27, 40].

In order to test our geometrically-exact rod formulation, we implemented it using the C++ language and we embedded it in our CHRONO open source simulation library [43]. We used this library to simulate IGA beams in a broader context involving rigid body dynamics, finite elements, motors, actuators and non-smooth frictional contacts: tests presented at the end of the paper show that the geometrically-exact IGA beam represents an efficient, compact and general way to simulate complex problems involving deformable beams.

2. B-Splines and NURBS

Since our beam model draws on Basis splines (B-Splines) or Non-Uniform Rational B-Splines (NURBS), basic concepts about these functions are introduced here.

2.1. B-Splines

A B-Spline of order p is a piecewise polynomial function of degree $p-1$ in a parametric variable $\tau \in \mathbb{R}$ (a curvilinear abscissa).

We introduce a set of $n+1$ control points $\mathbf{x}_i \in \mathbb{R}^3$ $i = 0 \dots n$ and a set $n+p+1$ of non-decreasing breaking points defining a *knot vector* $\mathbf{T} = (\tau_0, \tau_1 \dots \tau_{n+p})$.

Basis Functions. $N_{i,p}$ is an order p and $p-1$ degree *Basis Function* on the i -th knot of the B-Spline and it is recursively defined as follows:

$$N_{i,1}(\tau) = \begin{cases} 1 & \text{for } \tau_i \leq \tau \leq \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

And, for $p > 1$:

$$N_{i,p}(\tau) = \frac{\tau - \tau_i}{\tau_{i+p-1} - \tau_i} N_{i,p-1}(\tau) + \frac{\tau_{i+p} - \tau}{\tau_{i+p} - \tau_{i+1}} N_{i+1,p-1}(\tau) \quad (2)$$

Basis Function are a partition of unity: it means that $\sum_{i=0}^n N_{i,p}(\tau) = 1 \quad \forall \tau \in [\tau_o, \tau_n]$. The span of Basis Function increases with the order p .

Is useful to remind the derivative of the Basis Function with respect to the parametric variable since it is frequently used:

$$\dot{N}_{i,p}(\tau) = \frac{dN_{i,p}(\tau)}{d\tau} = \frac{p-1}{\tau_{i+p-1} - \tau_i} N_{i,p-1}(\tau) - \frac{p-1}{\tau_{i+p} - \tau_{i+1}} N_{i+1,p-1}(\tau) \quad (3)$$

B-Splines formulation. A B-Spline is a linear combination of control points \mathbf{x}_i and Basis Functions $N_{i,p}(\tau)$

$$\mathbf{r}(\tau) = \sum_{i=0}^n \mathbf{x}_i N_{i,p}(\tau) \quad n \geq p-1 \quad (4)$$

Given the number of control points $n+1$ and the order of the curve p the number of knots is $n+p+1$, which means there are more knots than control points, so some knots can be coincident on the same control point. When knots are distinct the first $p-1$ derivatives are continuous; when r nodes are coincident, only the first $p-r$ derivatives are continuous. In general, control points do not lie on the curve. When p knots are coincident, the spline passes through the control point with C^0 continuity.

2.2. NURBS

Non-Uniform Rational B-Splines introduce additional weights $w_i > 0$, ($i = 1 \dots n$), so that *rational* basis $R_{i,p}(\tau)$ are used in place of $N_{i,p}(\tau)$:

$$R_{i,p}(\tau) = \frac{N_{i,p}(\tau)w_i}{\sum_{j=1}^n N_{j,p}(\tau)w_j} \quad (5)$$

NURBS have the same properties listed for B-Splines: in particular for $w_i = 1 \forall i$, NURBS reduce to B-Splines. In addition, by using proper weights and few coarse control points, NURBS allow the exact (not approximated) representation of conic sections like circles and ellipses; this is a relevant feature because canonical primitives in CAD models are most often built from conical sections.

3. Kinematics of Cosserat rods

Our implementation of IGA beams draws on the Cosserat rod theory. This implies that the rotation of the beam section is independent from the centerline

position. This differs from the Euler-Bernoulli or Kirchhoff beam theory which implies that sections remain orthogonal to the centerline, an assumption that holds only for thin beams, because shear effects cannot be modelled.

3.1. Configuration

A Cosserat beam is represented by the line of its mass centroids (*centerline*), which is described by a curve parameterized [64] by a curvilinear abscissa $s \in [0, T]$:

$$\mathbf{r} = \mathbf{r}(s) \in \mathbb{R}^3 \quad (6)$$

Section rotations \mathbf{R} are parameterized by the same curvilinear abscissa $s \in [0, T]$ (even though not mandatory [5]), assuming that the X axis of the \mathbf{R} frame represents the normal of the beam section, and Y and Z axes represent the height and width directions, respectively:

$$\mathbf{R} = \mathbf{R}(s) \in \text{SO}(3) \quad (7)$$

That is, at some point s_a along the curve we have independent positions and rotations: $\mathbf{r}(s_a), \mathbf{R}(s_a)$.

We remark a first source of complication: \mathbf{R} is a 3D rotation matrix, that is a 3x3 orthogonal matrix (hence the *special orthogonal group* $\text{SO}(3)$ in the terminology of Lie groups). Its 3x3 elements are not independent as $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ must hold: in general it is better to avoid using all nine elements of such matrix and use quaternions or rotation angles to parameterize rotation frames [37].

3.2. Strains

Following [55], we can express $\underline{\epsilon}$ and $\underline{\kappa}$, hereafter called translational strains and rotational strains, respectively. The translational strain $\underline{\epsilon}$ is:

$$\underline{\epsilon} = \mathbf{R}^T \mathbf{r}' - \mathbf{e}_x \quad (8)$$

where we introduced $\mathbf{e}_x = \{1, 0, 0\}$ and we introduced the *curve gradient*

$$\mathbf{r}' = \frac{d\mathbf{r}}{ds}. \quad (9)$$

Note that for an undeformed beam we have $\mathbf{R}^T \mathbf{r}' = \{1, 0, 0\}$, that is we assume that s is a uniform arc-length parameterization. In other words, at the initial state, the curvilinear length of a curve from $s = a$ to $s = b$, i.e. $L = \int_a^b \|\mathbf{r}'\| ds$, is exactly $b - a$. Usually, however, splines are not parameterized with uniform arc-length abscissa, because a generic abscissa τ is used instead. If so, we have

$$\mathbf{r}' = \frac{d\mathbf{r}}{d\tau} \frac{d\tau}{ds} = \mathring{\mathbf{r}} J^{-1} \quad (10)$$

where a jacobian is defined as $J_{s\tau} = \frac{ds}{d\tau}$, and $\mathring{\mathbf{r}} = \frac{d\mathbf{r}}{d\tau}$ is the parametric derivative (something that is quite easy to compute when dealing with splines).

During an initialization phase, when the beam is still undeformed, values of jacobians $J_{s\tau}$ are computed and stored in memory for all the integration points - they will be used later, when computing the internal forces.

The rotational strain $\underline{\boldsymbol{\kappa}}$ is computed as [64]:

$$\underline{\boldsymbol{\kappa}} = \mathbf{R}^T \mathbf{R}' \quad (11)$$

Here $\mathbf{R}' = \frac{d\mathbf{R}}{ds}$, while the tilde operator means:

$$\tilde{\boldsymbol{\kappa}} = \begin{bmatrix} 0 & -\kappa_z & \kappa_y \\ \kappa_z & 0 & -\kappa_x \\ -\kappa_y & \kappa_x & 0 \end{bmatrix} \quad (12)$$

so it is obvious that one can build $\underline{\boldsymbol{\kappa}}$ from $\tilde{\boldsymbol{\kappa}}$ or vice versa.

Finally, in case the beam starts in an initially-curved configuration, initial values $\underline{\boldsymbol{\epsilon}}_0$ and $\underline{\boldsymbol{\kappa}}_0$ can be computed using the expression above, then one would compute effective strains with:

$$\boldsymbol{\epsilon} = \underline{\boldsymbol{\epsilon}} - \underline{\boldsymbol{\epsilon}}_0 \quad (13)$$

$$\boldsymbol{\kappa} = \underline{\boldsymbol{\kappa}} - \underline{\boldsymbol{\kappa}}_0 \quad (14)$$

3.3. Constitutive model

Once $\boldsymbol{\epsilon}$ and $\boldsymbol{\kappa}$ are computed, with some generic constitutive model one can compute their conjugate vectors, i.e. the generalized cut-section forces \mathbf{n} and cut-section torques \mathbf{m} , henceforth called translational stresses and rotational

stresses, respectively. The most generic constitutive model is expressed by a non-linear function:

$$\{\boldsymbol{\epsilon}, \boldsymbol{\kappa}\} \in \mathbb{R}^6 \rightarrow \{\mathbf{n}, \mathbf{m}\} \in \mathbb{R}^6 \quad (15)$$

Note that \mathbf{n} and \mathbf{m} are expressed in the local coordinate system of the section frame: if one needs the absolute values of such vectors, they can be computed easily, later, as

$$\mathbf{n}_a = \mathbf{R}\mathbf{n} \quad (16)$$

$$\mathbf{m}_a = \mathbf{R}\mathbf{m} \quad (17)$$

When structural damping is needed, instead of (15) we use:

$$\{\boldsymbol{\epsilon}, \boldsymbol{\kappa}, \dot{\boldsymbol{\epsilon}}, \dot{\boldsymbol{\kappa}}\} \in \mathbb{R}^{12} \rightarrow \{\mathbf{n}, \mathbf{m}\} \in \mathbb{R}^6 \quad (18)$$

by introducing the strain derivative as $\dot{\boldsymbol{\epsilon}}$ and the curvature derivative as $\dot{\boldsymbol{\kappa}}$.

See Appendix A for more details on the methods for computing generalized strains.

3.4. Equilibrium

The strong form for the equilibrium of the Cosserat beam is:

$$\mathbf{n}'_a + \underline{\mathbf{n}}_a = \mathbf{0} \quad (19)$$

$$\mathbf{m}'_a + \mathbf{r}'_a \times \mathbf{n}_a + \underline{\mathbf{m}}_a = \mathbf{0} \quad (20)$$

where, if provided, $\underline{\mathbf{n}}_a$ is the external force distributed on the beam, and $\underline{\mathbf{m}}_a$ is the external torque distributed on the beam, both expressed in absolute coordinates. The subscript a means we are using absolute coordinates for \mathbf{r}' too.

4. Discretization with IGA elements

The beam is approximated with IGA. To this end we assume that the beam is represented as a B-spline whose control points are nodes, each i -th node being a coordinate system $\{\mathbf{x}_i, \mathbf{R}_i\}$ with position $\mathbf{x}_i \in \mathbb{R}^3$ and rotation $\mathbf{R}_i \in \text{SO}(3)$, as shown in Fig. 1. For the rotation, however, we choose to parameterize rotation

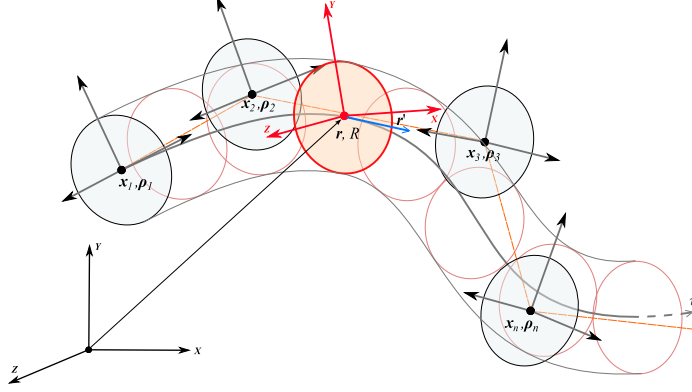


Figure 1: Spline discretization of the beam with independent position and rotation fields.

matrices \mathbf{R} using unit-length quaternions $\boldsymbol{\rho} \in \mathbb{H}_1$, so the configuration of each node is a compact set of 7 scalars: $\{\mathbf{x}_i, \boldsymbol{\rho}_i\}$. We recall the basic properties of quaternions: $\boldsymbol{\rho} = \rho_0 + \mathbf{i}\rho_1 + \mathbf{j}\rho_2 + \mathbf{k}\rho_3$ with $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$, often written succinctly $\boldsymbol{\rho} = [\rho_s, \boldsymbol{\rho}_v]$ to facilitate the expression of quaternion multiplication:

$$\boldsymbol{\tau}\boldsymbol{\rho} = [\boldsymbol{\tau}_s\rho_s - \boldsymbol{\tau}_v \cdot \boldsymbol{\rho}_v, \boldsymbol{\tau}_s\boldsymbol{\rho}_v + \rho_s\boldsymbol{\tau}_v + \boldsymbol{\tau}_v \times \boldsymbol{\rho}_v] \quad (21)$$

If needed, one can always convert quaternions into rotation matrices, as $\mathbf{R}_i = \mathbf{R}(\boldsymbol{\rho}_i)$, using the following property:

$$\mathbf{R}(\boldsymbol{\rho}) = \begin{bmatrix} \rho_0^2 + \rho_1^2 - \rho_2^2 - \rho_3^2 & 2(\rho_1\rho_2 - \rho_3\rho_0) & 2(\rho_1\rho_3 + \rho_2\rho_0) \\ 2(\rho_1\rho_2 + \rho_3\rho_0) & \rho_0^2 - \rho_1^2 + \rho_2^2 - \rho_3^2 & 2(-\rho_1\rho_0 + \rho_2\rho_3) \\ 2(\rho_1\rho_3 - \rho_2\rho_0) & 2(\rho_1\rho_0 + \rho_2\rho_3) & \rho_0^2 - \rho_1^2 - \rho_2^2 + \rho_3^2 \end{bmatrix} \quad (22)$$

Finally we denote the quaternion conjugate as $\boldsymbol{\rho}^*$, with $\boldsymbol{\rho}^* = \rho_0 - \mathbf{i}\rho_1 - \mathbf{j}\rho_2 - \mathbf{k}\rho_3$, such that $\mathbf{R}(\boldsymbol{\rho}_i^*)\mathbf{R}(\boldsymbol{\rho}_j) = \mathbf{R}(\boldsymbol{\rho}_i)^T\mathbf{R}(\boldsymbol{\rho}_j)$.

4.1. System state vectors

The state of the system contains velocities and angular velocities of each frame $\{\dot{\mathbf{x}}_i, \boldsymbol{\omega}_i\}$. We consider $\boldsymbol{\omega}_i$ expressed in frame-local coordinates.

This means that the state of the system, for n_n nodes, is given by the

system-level configuration \mathbf{q} and the system-state velocity \mathbf{v} as:

$$\mathbf{q} = \{\mathbf{x}_1, \boldsymbol{\rho}_1, \mathbf{x}_2, \boldsymbol{\rho}_2, \dots, \mathbf{x}_{n_n}, \boldsymbol{\rho}_{n_n}\} \quad (23)$$

$$\mathbf{v} = \{\dot{\mathbf{x}}_1, \boldsymbol{\omega}_1, \dot{\mathbf{x}}_2, \boldsymbol{\omega}_2, \dots, \dot{\mathbf{x}}_{n_n}, \boldsymbol{\omega}_{n_n}\} \quad (24)$$

In both statics and dynamics analysis it happens that one has to update by updating the \mathbf{q} configuration by applying some computed increment. For instance, in a linear static problem one has $\mathbf{K}\delta\mathbf{q} = \mathbf{b}$ and $\mathbf{q}_{new} = \mathbf{q} + \delta\mathbf{q}$. Also numerical methods for DAEs and ODEs in dynamics proceed by performing updates on the configurations at each time step. The problem is that, within our state \mathbf{q} , positions can be updated with straight sums as $\mathbf{x}_{i,new} = \mathbf{x}_i + \delta\mathbf{x}_i$, but quaternions cannot be updated as easily. In fact doing a straight sum with a $\delta\rho_i \in \mathbb{H}_1$ as in $\rho_{i,new} = \rho_i + \delta\rho_i$ could invalidate the unit-length of $\rho_{i,new}$. A more rigorous solution is to do incremental updates of quaternions using the exponential map. This requires some basic concepts of Lie algebras.

Recalling concepts of differential geometry, for an element \mathbf{R} in Lie group $\text{SO}(3)$ and an element $\delta\Theta$ in the corresponding Lie algebra $\mathfrak{so}(3)$, one has

$$\mathbf{R} = \exp(\delta\Theta) \quad (25)$$

$$\delta\Theta = \log(\mathbf{R}) \quad (26)$$

One can extract the three dimensional rotation pseudovector $\delta\theta$ from $\delta\Theta$ via $\delta\theta = \text{axis}(\delta\Theta)$, recalling that $\delta\Theta = \text{skew}(\delta\theta) = \delta\tilde{\theta}$. For our purposes, $\delta\theta$ can be considered a (not necessarily infinitesimal) incremental rotation; for example in a time stepper it could be $\delta\theta = \boldsymbol{\omega}dt$.

For sake of compactness and performance, we avoid updating \mathbf{R} and we rather exploit the fact that \mathbb{H}_1 is a double cover of $\text{SO}(3)$. Just like in (25) and (26), an exponential map links \mathbb{H}_1 , (unit quaternions), and its Lie algebra $\text{Im}(\mathbb{H})$ of *pure quaternions* $\delta\rho = [0, \delta\theta]$:

$$\rho = \exp(\delta\rho) \quad (27)$$

$$\delta\rho = \log(\rho) \quad (28)$$

We can define two operators to convert pure quaternions from and to rotation pseudovectors: $\delta\boldsymbol{\theta} = \text{imag}([0, \delta\boldsymbol{\theta}])$, and $[0, \delta\boldsymbol{\theta}] = \text{pure}(\delta\boldsymbol{\theta})$. The exponential map (27) can be explicitly computed from $\delta\boldsymbol{\theta}$ as $\boldsymbol{\rho} = \exp(\text{pure}(\delta\boldsymbol{\theta}))$ using the following closed-form expression:

$$\boldsymbol{\rho} = \exp([0, \delta\boldsymbol{\theta}]) = \left\{ \cos\left(\frac{\|\delta\boldsymbol{\theta}\|}{2}\right), \frac{\delta\boldsymbol{\theta}}{\|\delta\boldsymbol{\theta}\|} \sin\left(\frac{\|\delta\boldsymbol{\theta}\|}{2}\right) \right\} \quad (29)$$

This said, in our code we can work with increments in the following form:

$$\delta\mathbf{q} = \{\delta\mathbf{x}_1, \delta\boldsymbol{\theta}_1, \delta\mathbf{x}_2, \delta\boldsymbol{\theta}_2, \dots, \delta\mathbf{x}_{n_n}, \delta\boldsymbol{\theta}_{n_n}\} \quad (30)$$

and, where one has to perform the incremental update to find \mathbf{q}_{new} , the rotational parts are incremented pre-multiplying the quaternion by an exponential map, as:

$$\mathbf{x}_{i,new} = \mathbf{x}_i + \delta\mathbf{x}_i \quad (31)$$

$$\boldsymbol{\rho}_{i,new} = \boldsymbol{\rho}_i^\delta \boldsymbol{\rho}_i \quad (32)$$

where one computes $\boldsymbol{\rho}_i^\delta = \exp([0, \delta\boldsymbol{\theta}_i])$ using (29), then computes the product $\boldsymbol{\rho}_i^\delta \boldsymbol{\rho}_i$ using (21). More succinctly, the incremental update is a map $\mathbf{q}_{new} = \Lambda(\mathbf{q}, \delta\mathbf{q})$.

In the following, for expressing formulas in a easier way, we group all the translational degrees of freedom and all the rotational degrees of freedom in two separate vectors \mathbf{q}_x and \mathbf{q}_ρ , and we do the same for the increments $\delta\mathbf{q}_x$ and $\delta\mathbf{q}_\rho$:

$$\mathbf{q}_x = \{\delta\mathbf{x}_1, \delta\mathbf{x}_2, \dots, \delta\mathbf{x}_{n_n}\} \quad (33)$$

$$\mathbf{q}_\rho = \{\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \dots, \boldsymbol{\rho}_{n_n}\} \quad (34)$$

$$\delta\mathbf{q}_x = \{\delta\mathbf{x}_1, \delta\mathbf{x}_2, \dots, \delta\mathbf{x}_{n_n}\} \quad (35)$$

$$\delta\mathbf{q}_\rho = \{\delta\boldsymbol{\theta}_1, \delta\boldsymbol{\theta}_2, \dots, \delta\boldsymbol{\theta}_{n_n}\} \quad (36)$$

4.2. Interpolation

In IGA, the role of FEA shape functions is done by Basis Functions of the spline. For a given knot abscissa τ along the spline, one can compute interpolated position and gradient using Basis Functions $N_i(\tau)$ and their derivatives

$\dot{N}_i(\tau) = \frac{dN_i(\tau)}{d\tau}$, that is:

$$\mathbf{r}(\tau) = \sum_i N_i(\tau) \mathbf{x}_i \quad (37)$$

$$\mathbf{r}'(\tau) = \frac{d\mathbf{r}(\tau)}{d\tau} J_{s\tau}^{-1} \quad (38)$$

$$= \sum_i \dot{N}_i(\tau) \mathbf{x}_i J_{s\tau}^{-1} \quad (39)$$

Once this is computed, $\underline{\mathbf{e}}(\tau)$ can be computed with (8).

Similarly, one would be tempted to interpolate the rotation at a given abscissa τ using a similar method, however applying directly $\boldsymbol{\rho}(\tau) = \sum_i N_i(\tau) \boldsymbol{\rho}_i$ is not possible because the linear sum of unit quaternions does not represent, in general, a rotation. In fact, the spline interpolation of quaternions is still a debated problem and exact solutions presented in literature, such as [35, 34], often lead to complex formulas that add complication and computational overhead. As a simplification we introduce an auxiliary co-rotated system $\underline{\boldsymbol{\rho}}$, for instance an average of rotations of the closest nodes, and we interpolate rotations by doing a weighted sum of rotation pseudovectors later recast as a quaternion using the exponential map:

$$\boldsymbol{\rho}(\tau) = \underline{\boldsymbol{\rho}} \exp \left(\text{pure} \sum_i N_i(\tau) \text{imag}(\log(\underline{\boldsymbol{\rho}}^* \boldsymbol{\rho}_i)) \right) \quad (40)$$

Then, the \mathbf{R} matrix can be computed as $\mathbf{R}(\boldsymbol{\rho})$. Also, the curvature $\boldsymbol{\kappa}$ is computed as:

$$\boldsymbol{\kappa}(\tau) = \sum_i J_{s\tau}^{-1} \dot{N}_i(\tau) \text{imag}(\log(\boldsymbol{\rho}(\tau)^* \boldsymbol{\rho}_i)) \quad (41)$$

These allow the computation of local stresses $\mathbf{m}(\tau)$ and $\mathbf{n}(\tau)$ according to (15). If a constitutive material includes damping effects, one can compute also $\dot{\boldsymbol{\epsilon}}(\tau) = \mathbf{R}^T \sum_i \dot{N}_i(\tau) \dot{\mathbf{x}}_i J_{s\tau}^{-1}$ and $\dot{\boldsymbol{\kappa}}(\tau) = \mathbf{R}^T \sum_i \dot{N}_i(\tau) \mathbf{R}_i \boldsymbol{\omega}_i J_{s\tau}^{-1}$ to be used in (18).

4.3. Gauss quadrature

As in most FEA frameworks, also in IGA the backbone of the process is the computation of three main ingredients: the vector of internal forces \mathbf{f}_{int} , the

tangent stiffness matrix \mathbf{K}_t , and the mass matrix \mathbf{M} . Once these are computed, one can solve, for instance, linear elastic problems as $\mathbf{K}_t \delta \mathbf{q} = \mathbf{f}_{ext}$, non linear elastic problems that iterate on $\mathbf{K}_t \delta \mathbf{q} = \mathbf{f}_{ext} + \mathbf{f}_{int}$ up to convergence, explicit dynamic integration as in $\mathbf{M} \ddot{\mathbf{q}} = \mathbf{f}_{ext} + \mathbf{f}_{int}$ and so on. Therefore, in the following, we present the procedure for computing these terms, starting from the most relevant part, that is the vector of internal forces \mathbf{f}_{int} .

Here we focus on computing \mathbf{f}_{int} via typical Gauss integration. We remark that an appropriate choice of integration points must be used in order to avoid shear locking phenomena, to this end we used selective reduced integration requiring fewer quadrature points than standard integration techniques. Efficient choice of Gauss points is discussed by various authors, for instance [30, 28, 1]. In fact, a more recent and efficient approach to IGA consists in using collocation instead of Gauss integration [5, 41, 62].

Also, we use the same idea of FEA of computing internal forces and matrices on a per-element basis, where later all element terms will be assembled/overlapped in global system-level matrices and vectors. In this sense, in our embodiment the j -th element is the j -th span of the B-spline, so for each element we compute $\mathbf{f}_{int,j}$, and later all those per-element internal forces are assembled in a single vector ¹.

Over the j -th span from $s = s_A$ to $s = s_B$ the weak form of the equilibrium reads:

$$\delta \Pi = \int_{s_A}^{s_B} (\delta \epsilon_a \mathbf{n}_a + \delta \boldsymbol{\kappa}_a \mathbf{m}_a) ds - \int_{s_A}^{s_B} (\delta \mathbf{x}_a \mathbf{n}_a + \delta \boldsymbol{\theta}_a \mathbf{m}_a) ds \quad (42)$$

Recalling (8), evaluating its increment and referring it to the global coordinates:

$$\delta \boldsymbol{\epsilon}_a = \delta \mathbf{r}'_a - \delta \boldsymbol{\theta}_a \times \mathbf{r}'_a \quad (43)$$

So, at a point at abscissa τ along the spline, $\delta \boldsymbol{\epsilon}$ and $\delta \boldsymbol{\kappa}$ can be expressed

¹Differently from FEA, in IGA one could compute the system-level vector without passing through intermediate per-element vectors, but here for clarity we develop our formulation on a per-element basis

using interpolation of the B-spline given the following terms:

$$\delta \mathbf{r}'_a = \sum_i J_{s\tau}^{-1} \dot{N}_i \delta \mathbf{x}_i \quad (44)$$

$$\delta \boldsymbol{\kappa}_a = \sum_i J_{s\tau}^{-1} \dot{N}_i \mathbf{R}_i \delta \boldsymbol{\theta}_i \quad (45)$$

$$\delta \boldsymbol{\theta}_a = \sum_i N_i \mathbf{R}_i \delta \boldsymbol{\theta}_i \quad (46)$$

We rewrite the weak equilibrium introducing the vector of internal forces

$$\delta \mathbf{q}_j^T \mathbf{f}_{int,j} - \delta \mathbf{q}_j^T \mathbf{f}_{ext,j} = \mathbf{0} \quad (47)$$

where $\mathbf{f}_{int,j}$ and $\mathbf{f}_{ext,j}$ are the internal and external forces acting on the j -th node.

Looking at the expressions above, one can write the expression of the contribution from the i -th control node to $\mathbf{f}_{int,j}$ as an integral in s arc-length coordinate:

$$\mathbf{f}_{int,j,i} = \int_{s_A}^{s_B} \begin{bmatrix} J_{s\tau}^{-1} \dot{N}_i \mathbf{I} & N_i \mathbf{R}_i \tilde{\mathbf{r}}' \\ 0 & J_{s\tau}^{-1} \dot{N}_i \mathbf{R}_i \end{bmatrix}^T \begin{bmatrix} \mathbf{n}_a \\ \mathbf{m}_a \end{bmatrix} ds \quad (48)$$

The integral is computed with a sum over n_{gp} Gauss points, so we introduce Gauss point weights w_k , jacobians $J_{\tau\zeta} = \frac{d\tau}{d\zeta}$ where $\zeta \in [-1, +1]$ is the coordinate used for Gauss quadrature. Because of change of coordinates, we also have to multiply the integrand by $J_{s\tau}$, so some jacobians will simplify as $J_{s\tau}^{-1} J_{s\tau} = 1$. Finally one has:

$$\mathbf{f}_{int,j,i} = \sum_k^{n_{gp}} w_k J_{\tau\zeta} \begin{bmatrix} \dot{N}_i \mathbf{I} & J_{s\tau} N_i \mathbf{R}_i \tilde{\mathbf{r}}' \\ 0 & \dot{N}_i \mathbf{R}_i \end{bmatrix}^T \begin{bmatrix} \mathbf{n}_a \\ \mathbf{m}_a \end{bmatrix} \quad (49)$$

Here all the terms of the integrand (namely, N_i , \dot{N}_i , $J_{s\tau}$, $\tilde{\mathbf{r}}'$, \mathbf{n}_a and \mathbf{m}_a) must be computed for the k -th knot abscissa $\tau_k = \left(\frac{s_B - s_A}{2} \zeta_k + \frac{s_B + s_A}{2} \right)$ where ζ_k is one of the n_{gp} tabulated abscissas for Gauss integration, i.e. coupled to the corresponding weight w_k . By the way one can see that $J_{\tau\zeta} = \frac{s_B - s_A}{2}$.

We remark that control points will affect neighbouring spans (elements) hence terms $\mathbf{f}_{int,j,i}$ from different spans will overlap and must be summed to obtain $\mathbf{f}_{int,j}$.

The number of nodes influencing a single span increases with the spline order. IGA spline Basis Functions can be considered like shape functions affecting more knot spans, combining each other in the shared segments of support. For this reason an higher spline order leads to more overlapped Basis Functions, while increasing the element order in FE analysis adds nodes within the element but internal nodes do not affect adjacent elements. This leads to a band-diagonal stiffness matrix whose band width increases with the spline order, as opposed to traditional FE elements characterized by a block-diagonal stiffness matrix whose blocks overlap only at the boundary nodes.

5. Stiffness, mass and damping matrices

For the time integration of the equations of motion, the mass matrix is needed. If implicit integration schemes are used, also the tangent stiffness and damping matrices are required.

In our implementation we use a lumped mass matrix where each i -th IGA control point has an atomic mass m_i and a tensor of inertia $I_i \in \mathbb{R}^{3 \times 3}$. This has a computational advantage with respect to the explicit quadrature of the beam shape, because the tensor of inertia $I_i = \text{diag}(I_{xx_i}, I_{yy_i}, I_{zz_i})$ is diagonal and constant as a consequence of the fact that we assumed angular velocities ω_i to be expressed in the local coordinate system. Values of $m_i, I_{xx_i}, I_{yy_i}, I_{zz_i}$ can be computed from the geometry of the section and from the density of the material.

The tangent stiffness matrix and the damping matrix are computed by numerical differentiation of \mathbf{f}_{int} .

$$\mathbf{K}_t = -\nabla_{\mathbf{q}} \mathbf{f}_{int} = -\left[\frac{\partial \mathbf{f}_{int}}{\partial \mathbf{q}} \right], \quad \mathbf{R}_t = -\nabla_{\mathbf{v}} \mathbf{f}_{int} = -\left[\frac{\partial \mathbf{f}_{int}}{\partial \mathbf{v}} \right] \quad (50)$$

Since the procedure for computing (49) is quite simple, the computational overhead is not excessive with respect to an analytic evaluation of the matrix, yet it has the useful property of being of general validity even when using black-box nonlinear constitutive laws for materials. For a practical implementation, for the sake of high performance, one should exploit the sparsity pattern of \mathbf{K}_t .

In detail, the numerical differentiation operates on a per-span basis, to obtain per-span stiffness matrices \mathbf{K}_{t_j} that are summed afterward to obtain the large and sparse \mathbf{K}_t system-level matrix.

6. Contacts

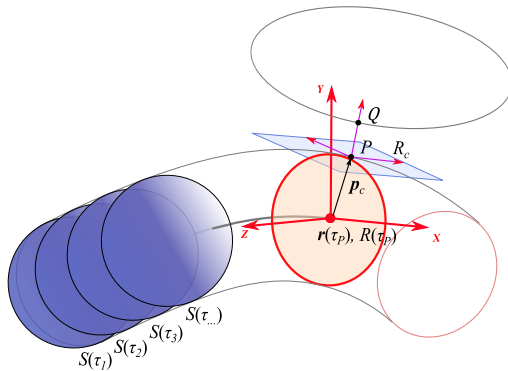


Figure 2: Sampling the beam with contact shapes (proxies) for fast collision detection.

We developed a framework for simulating contacts between moving parts, either beams and other objects (rigid bodies, other finite elements, etc.).

When targeting complex scenarios with thousands of elements, we experienced that an efficient approach is to sample the rod with simple collision geometries that can be easily handled by fast collision algorithms. For example a rod with a circular section can be approximated with invisible collision spheres S_i , evenly spaced along the beam centerline, at given parametric abscissas τ_i , as shown in Fig. 2. Those proxies are passed to the collision engine that computes candidate collision points at each time step: to this end we use a Sweep-And-Prune (SAP) broad-phase optimization that allows a near linear-time complexity even in case of many shapes, and a Gilbert-Johnson-Keerthi (GJK) algorithm for the narrow-phase.

The narrow-phase returns a variable number of candidate contact pairs, each represented by two points \mathbf{P} and \mathbf{Q} belonging to two colliding shapes, and a coordinate system \mathbf{R}_c whose x axis represent the contact normal. We assume

that the contact distance $\Phi = |\mathbf{P} - \mathbf{Q}|$ is differentiable. The distance vector, in the contact coordinates, is $\mathbf{D} = \mathbf{R}_c^T(\mathbf{P} - \mathbf{Q})$.

In order to write the equation of motion in the following section, we need to express the relative speed of the two points, $\mathbf{u} = \mathbf{R}_c^T(\dot{\mathbf{P}} - \dot{\mathbf{Q}})$ as a function of generalized coordinates \mathbf{v} .

Assuming point \mathbf{P} belongs to a beam, we can recover τ_P , the abscissa at which the colliding shape was sampled. This allows the immediate ² computation of $\mathbf{R}(\tau_P)$, the rotation of the section at the contact, and \mathbf{p}_c , the position of the contact point in the section coordinate system, as in $\mathbf{P}(\tau_P) = \mathbf{r}(\tau_P) + \mathbf{R}(\tau_P)\mathbf{p}_c$.

Taking the time derivative, simplifying for $\dot{\mathbf{p}}_c = 0$ and recalling the spline Basis Functions:

$$\dot{\mathbf{P}}(\tau_P) = \dot{\mathbf{r}}(\tau_P) + \dot{\mathbf{R}}(\tau_P)\mathbf{p}_c + \mathbf{R}(\tau_P)\dot{\mathbf{p}}_c \quad (51a)$$

$$= \sum_i N_i(\tau_P)\dot{\mathbf{x}}_i + \mathbf{R}(\tau_P)\dot{\boldsymbol{\omega}}(\tau_P)\mathbf{p}_c + 0 \quad (51b)$$

$$= \sum_i N_i(\tau_P)\dot{\mathbf{x}}_i - \mathbf{R}(\tau_P)\tilde{\mathbf{p}}_c\boldsymbol{\omega}(\tau_P) \quad (51c)$$

$$= \sum_i N_i(\tau_P)\dot{\mathbf{x}}_i - \sum_i N_i(\tau_P)\mathbf{R}(\tau_P)\tilde{\mathbf{p}}_c\boldsymbol{\omega}_i \quad (51d)$$

At this point it is possible to write the contact speed of the k -th contact using generalized coordinates, $\mathbf{u}_k = \nabla_{\mathbf{v}}^T \mathbf{D}_k \mathbf{v}$, where the jacobian $\nabla_{\mathbf{v}}^T \mathbf{D}_k \in \mathbb{R}^{3 \times n_v}$ has a sparse structure with two blocks:

$$\nabla_{\mathbf{v}}^T \mathbf{D}_k = [\dots, \nabla_{\mathbf{v}}^T \mathbf{D}_{kP}, \dots, \nabla_{\mathbf{v}}^T \mathbf{D}_{kQ}, \dots]. \quad (52)$$

²An optional refinement can happen at this point: since the collision shape -a sphere, a capsule...- was only an approximation of the smoothly bent surface of the beam, one can iteratively correct the (discrete-sampled) contact abscissa in order to satisfy $\mathbf{P}(\tau_P) = \mathbf{r}(\tau_P) + \mathbf{R}(\tau_P)\mathbf{p}_c$ until the position of the contact point in the beam section frame, \mathbf{p}_c , has perfectly zero off-section value: $p_{c,x} = 0$

In detail, from (51d) one has

$$\begin{aligned} \nabla_{\mathbf{v}}^T \mathbf{D}_{\mathbf{k}P} = & \left[N_1(\tau_P) \mathbf{R}_{\mathbf{c}}^T, -N_1(\tau_P) \mathbf{R}_{\mathbf{c}}^T \mathbf{R}(\tau_P) \tilde{\mathbf{p}}_{\mathbf{c}}, \right. \\ & N_2(\tau_P) \mathbf{R}_{\mathbf{c}}^T, -N_2(\tau_P) \mathbf{R}_{\mathbf{c}}^T \mathbf{R}(\tau_P) \tilde{\mathbf{p}}_{\mathbf{c}}, \\ & \left. \dots, N_n(\tau_P) \mathbf{R}_{\mathbf{c}}^T, -N_n(\tau_P) \mathbf{R}_{\mathbf{c}}^T \mathbf{R}(\tau_P) \tilde{\mathbf{p}}_{\mathbf{c}} \right] \quad (53) \end{aligned}$$

whereas an equivalent expression holds for $\nabla_{\mathbf{v}}^T \mathbf{D}_{\mathbf{k}Q}$ if also \mathbf{Q} belongs to a beam³.

As a special case, very thin beams have $\|\mathbf{p}_{\mathbf{c}}\| \approx 0$ hence $-N_n(\tau_P) \mathbf{R}_{\mathbf{c}}^T \mathbf{R}(\tau_P) \tilde{\mathbf{p}}_{\mathbf{c}}$ terms could be removed from the jacobian in sake of performance.

Finally, it is possible to group all n_c contact jacobians in a system-level sparse matrix $\nabla_{\mathbf{v}}^T \mathbf{D} \in \mathbb{R}^{3n_c \times n_v}$.

7. Time stepping and implementation

In the most general setting, we assume that the system is made by a several IGA beams, as well as other finite elements and rigid bodies, and we assume that all their states are grouped in the system-level vectors \mathbf{q} and \mathbf{v} . Also, we assume that the configuration is subject to holonomic-rheonomic constraint equations $\mathbf{C}(\mathbf{q}, t) = \mathbf{0}$. Differentiation of constraints leads to $\frac{\partial \mathbf{C}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{C}}{\partial t} = \mathbf{0}$, that is $\nabla_{\mathbf{v}}^T \mathbf{C} \mathbf{v} + \mathbf{C}_t = \mathbf{0}$ where $\nabla_{\mathbf{v}}^T \mathbf{C} = \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \Gamma(\mathbf{q})$ is the constraint jacobian, $\Gamma(\mathbf{q})$ is a linear operator for $\dot{\mathbf{q}} = \Gamma(\mathbf{q}) \mathbf{v}$, and \mathbf{C}_t is a null term in case of scleronomic constraints, but can appear in case of time-dependant constraints (motors, imposed trajectories, etc.).

We perform time integration either with implicit integrators⁴ for smooth Differential-Algebraic-Equations (DAE), or with a custom time stepper for non-smooth DVI. The latter allows also an efficient treatment of contacts. In fact,

³For instance, if \mathbf{Q} belongs to a rigid body, the corresponding jacobian block has a simpler expression: $\nabla_{\mathbf{v}}^T \mathbf{D}_{\mathbf{k}Q} = [-\mathbf{R}_{\mathbf{c}}^T, \mathbf{R}_{\mathbf{c}}^T \mathbf{R}_{\mathbf{b}} \hat{\mathbf{q}}_{\mathbf{c}}]$ where $\mathbf{R}_{\mathbf{b}}$ is the rotation of the body and $\hat{\mathbf{q}}_{\mathbf{c}}$ is the position of \mathbf{Q} in body local coordinates.

⁴HHT [31] or Newmark [48] in our implementation, here not described for compactness.

if one wants to simulate contacts in the framework of conventional smooth dynamics, in most cases contact constraints would be dealt with regularization, i.e. penalty functions that would require very short time steps. On the other hand, the DVI approach embeds the set-valued non-smooth nature of contact forces directly in the formulation [49]. At l -th time step we solve:

$$\begin{bmatrix} \mathbf{H} & \nabla_{\mathbf{v}}\mathbf{C} & \nabla_{\mathbf{v}}\mathbf{D} \\ \nabla_{\mathbf{v}}^T\mathbf{C} & 0 & 0 \\ \nabla_{\mathbf{v}}^T\mathbf{D} & 0 & 0 \end{bmatrix} \begin{pmatrix} \mathbf{v}^{(l+1)} \\ \boldsymbol{\lambda}_C \\ \boldsymbol{\lambda}_D \end{pmatrix} - \begin{pmatrix} \mathbf{H}\mathbf{v}^{(l)} + h(\mathbf{f}_{int} + \mathbf{f}_{est}) \\ -\frac{1}{h}\mathbf{C}(\mathbf{q}, t) - \mathbf{C}_t \\ -\frac{1}{h}\boldsymbol{\Phi}(\mathbf{q}) \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{y} \end{pmatrix} \quad (54)$$

$$\mathbf{y} \in \Upsilon^* \perp \boldsymbol{\lambda}_D \in \Upsilon \quad (55)$$

$$\mathbf{q}^{(l+1)} = \Lambda(\mathbf{q}^{(l)}, h\mathbf{v}^{(l+1)}) \quad (56)$$

where $\mathbf{H} = \mathbf{M} + h^2\mathbf{K}_t + h\mathbf{R}_t$ is a sparse matrix, h is the time step, contact jacobians are computed as in (53), $\Upsilon = \oplus_i \Upsilon_i$ is the Cartesian product of all Coulomb friction cones generated by contacts, and $\Upsilon^* = \{\boldsymbol{\lambda}_D \in \Upsilon^* : \langle \boldsymbol{\lambda}_D, \mathbf{z} \rangle \geq 0 \quad \forall \mathbf{z} \in \Upsilon\} = \oplus_i \Upsilon_i^*$ is its dual. Finally, (56) is updated with exponential maps as in (31)-(32).

In absence of contacts, it can be demonstrated that the algorithm (54)-(56) simplifies to a backward Euler linearly implicit integrator, of whom it inherits the same properties of first order accuracy in time and L-stability. If higher order accuracy or less numerical damping is needed, recent research demonstrated the possibility of second-order DVI time steppers at the cost of additional complexity in the implementation [13].

The problem (54)-(55) is a Second-Order Cone Complementarity Problem (SO-CCP) with unknowns $\mathbf{v}^{(l+1)}$, the speed at the end of time step, and $\boldsymbol{\lambda}$, the reaction impulses in contacts or bilateral constraints. The convex cone complementarity problem is also a special case of a Variational Inequality (VI), hence the DVI acronym [51].

The complementarity constraint (55) introduces a major numerical difficulty, however the monolithic SO-CCP (54)-(55) can be solved efficiently using the primal-dual interior point method that we presented in [40] and that we implemented in C++.

Although the DVI integrator (54)-(56) is stable also when using large time steps, we remark that there are scenarios where the time step must be extremely short anyway, for example because one wants to simulate collisions between very thin beams, a case that requires small steps in order to avoid tunneling and aliasing effects in the computation of contact points; in such cases, the advantage of using the DVI is less evident.

Comparing this non-smooth DVI integrator to conventional smooth explicit integrators, one can see that DVI requires more computational effort per each time step, because of the SO-CCP overhead. On the other hand, explicit integrators require a much larger number of time steps to achieve stability, depending on how stiff is the contact formulation.

Comparing the DVI integrator to conventional smooth implicit integrators embedded with penalty formulations leads to similar conclusions: the SO-CCP solver overhead is still higher than the few linear system solves required in the Newton iterations of implicit steppers, but on the other side the total number of time steps can be smaller. Implicit solvers, in fact, could use large time steps as the DVI, but the highly non-linear nature of the contact forces often impair their performance to the point that they need short time steps anyway.

In general, since the SO-CCP solution dominates the overall computational effort per each time step of the DVI, the efficiency of the DVI method respect to smooth integrators depends mostly on how optimized is the SO-CCP solver. In order to maximize the computational performance, the inner loop of our interior-point method leverages the MKL parallel direct solver from Intel. As an alternative, we tested also CVXOPT, a Python library for large-scale cone programming, again based on an interior-point method [2].

8. Numerical tests

8.1. The ring bending test

This is a classical benchmark where a cantilever beam is subject to a concentrated torque T acting on the tip to obtain a ring bending deformation. Since the length of the neutral axis is constant under pure bending, the initial beam

length L must be equal to the final circumference, hence the curvature radius must be $\rho = \frac{L}{2\pi}$. For a linear elastic rod it holds $\rho = \frac{EI}{T}$, so the torque required to obtain a perfect circle must be

$$T = \frac{2\pi EI}{L}$$

Because of the large deformations, we performed a non-linear static analysis where the torque value has been increased gradually to avoid divergence. Our simulations converge consistently with the analytical solution, as shown in Fig. 3.

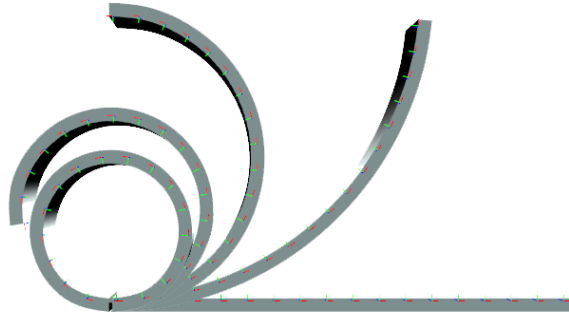


Figure 3: Progression of ring bending

8.2. Princeton beam experiment

A thin beam, depicted in Fig. 4, is subject to large deformations and large rotations because of a tip load at E , for different angles θ . A non-linear analysis shows that, for large displacements, a strong twisting action couples to the bending action, hence obtaining out-of-plane displacements even if the load is vertical.

Three loading conditions are tested: $P_1 = 4.448\text{N}$, $P_2 = 8.896\text{N}$, and $P_3 = 13.345\text{N}$, for θ ranging in the $[0^\circ, 90^\circ]$ interval. The beam has length $L = 0.508\text{m}$, section height $H = 12.77\text{mm}$, section thickness $T = 3.2024\text{mm}$, Young modulus $E = 71.7\text{GPa}$, $\nu = 0.31$, $G = E \frac{(1+\nu)}{2} = 27.37\text{GPa}$.

Because of the geometric nonlinearities, the solver must perform Newton-Raphson steps before obtaining a zero residual in the equations of static equilibrium.

Results in Figs. 5, 6 and 7 show a good agreement between the present IGA formulation and reference data [6]. In detail, there is a good agreement with other geometrically-exact beam formulations (non IGA-based) discussed in [7] for Dymore and in [25] for MBDyn, as well as an agreement with the experimental results in [21, 22], obtained with a beam made with 7075 aluminium alloy.

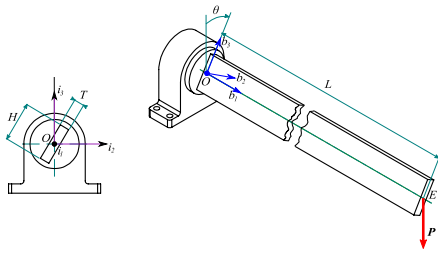


Figure 4: Setup of the benchmark for the Princeton beam experiment.

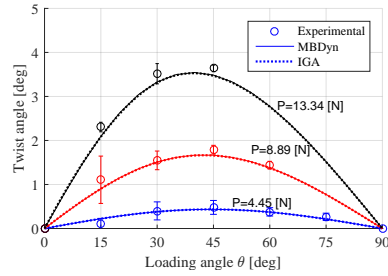


Figure 5: Twist rotation of the beam for the Princeton experiment.

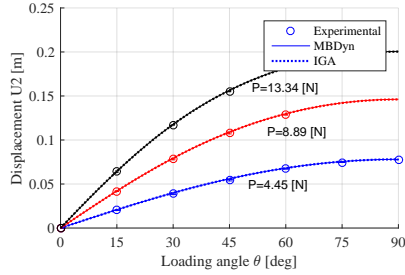


Figure 6: Flapwise displacement at the beam tip versus loading angle for three loading conditions.

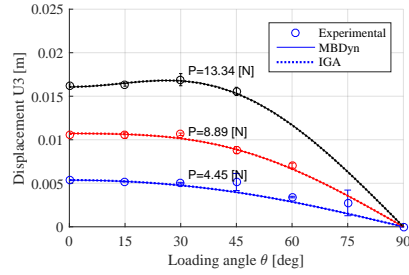


Figure 7: Chordwise displacement at the beam tip versus loading angle for three loading conditions.

8.3. Jeffcott rotor

This benchmark explores the reliability of the numerical method in the non-linear dynamical analysis of a flexible system rotating at finite angular velocity. A rotating unbalanced shaft of length $L = 6$ m is integrated in time. As shown in Fig. 8, a rigid disk is connected to the shaft at mid-span, above the reference shaft axis by an offset $d = 0.05$ m. The shaft is made of steel (density $\rho =$

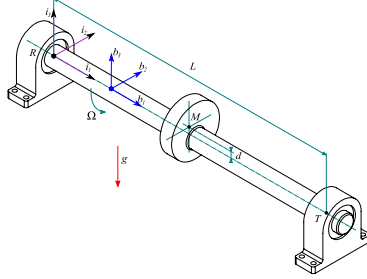


Figure 8: Setup of the unbalanced rotating shaft benchmark.

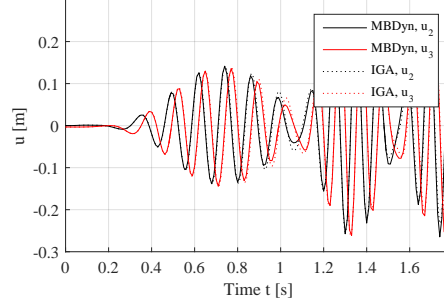


Figure 9: Mid-point transverse displacement of unbalanced rotating shaft.

7800kg/m³, Young's modulus $E = 210$ GPa, Poisson's ratio $\nu = 0.3$). The cross section is annular ($r_i = 0.045$ m, $r_o = 0.05$ m). The mass of the disk is $m_d = 70.573$ kg, the radius is $r_d = 0.24$ m, and the thickness is $t_d = 0.05$ m. The system is subjected to gravity ($g = 9.81$ m s⁻²) directed transversely. The end R of the shaft is connected to the ground by a cylindrical joint (displacement along and rotation about the shaft's axis are permitted). The end T is supported by a revolute joint; the relative angular velocity about the shaft axis is prescribed as a function of time,

$$\Omega(t) = \begin{cases} A_1\omega(1 - \cos(\pi t/T_1))/2 & 0 \leq t \leq T_1 \\ A_1\omega & T_1 < t \leq T_2 \\ A_1\omega + (A_2 - A_1)\omega(1 - \cos(\pi(t - T_2)/(T_3 - T_2)))/2 & T_2 < t \leq T_3 \\ A_2\omega & T_3 < t \end{cases}$$

with $A_1 = 0.8$, $A_2 = 1.2$, $T_1 = 0.5$ s, $T_2 = 1$ s, $T_3 = 1.25$ s, and $\omega = 60$ rad s⁻¹, close to the first natural frequency of the system. The shaft accelerates from zero and passes from sub-critical to super-critical regime; when passing through the first natural bending frequency of the system, lateral oscillations occur and significant forces take place, as predicted by the linear theory of unbalanced rotors.

Results plotted in Fig.9 for the case of a 3rd order IGA with nine nodes, using a second order implicit time stepping integrator, show a good agreement with reference results obtained with MBDyn.

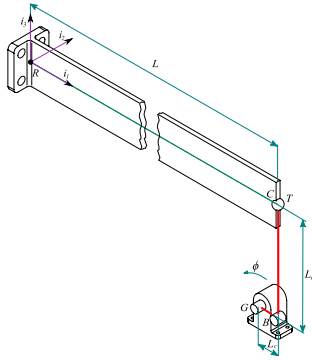


Figure 10: Setup of the benchmark for lateral buckling dynamics.

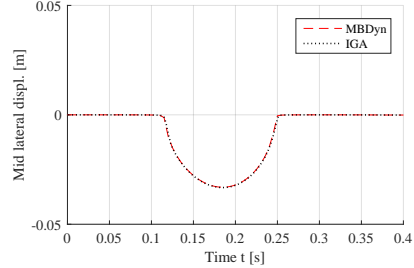


Figure 11: Static displacement of the beam along i_2 , at the mid point.

8.4. Lateral buckling

This is a benchmark that tests the beam formulation in the context of a dynamical problem of difficult integration.

In detail, a flat beam is bent in its plane of highest flexural rigidity, up to the point where lateral buckling is instantly triggered. In a quasi-static non-linear analysis, results are visible in Fig. 11. In the context of dynamics, when buckling occurs, the beam snaps laterally and twists, inducing highly oscillatory motions. The IGA beam discussed in this paper can capture the nonlinear nature of this phenomena.

As depicted in Fig. 10, a RC beam with length $L = 1\text{m}$ and rectangular section $H = 100\text{mm}$, $B = 10\text{mm}$, is clamped at the R end point. The snapping is caused by a tip load at C , generated by mean of a rotating crank GB and a vertical rod TB , with a spherical joint in C and a revolute joint in B . An initial imperfection is simulated by introducing a small offset $d = 0.1\text{mm}$ in the off-plane direction i_2 between the crank and the vertical bar.

The crank has length $L_c = 0.05\text{m}$ and its circular section has diameter $D_r = 24\text{mm}$, the vertical rod has length $L_r = 0.25\text{m}$ and its circular section has diameter $D_r = 48\text{mm}$. The rotation of the crank is enforced by a prescribed motion function $\phi_c(t) = \pi(1 - \cos(\pi t/T_c))/2$, with $T_c = 0.4\text{s}$, then after $t > T_c$ it holds $\phi_c(t) = \pi$.

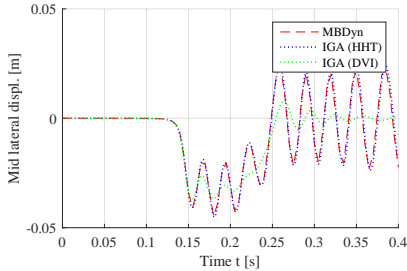


Figure 12: Displacement of the beam along i_2 , at the mid point.

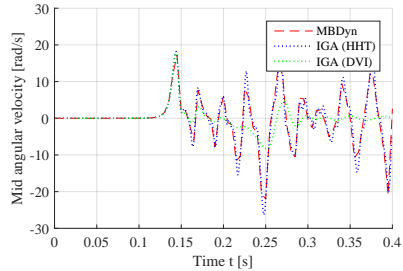


Figure 13: Angular velocity of the beam, at the mid point.

All parts have Young modulus $E = 73\text{GPa}$ and Poisson ratio $\nu = 0.3$. For the three beams, inertia values I_{zz} and I_{yy} and torsion constants J are computed using formulas available in classical textbooks.

In our tests the crank and the rod are modeled with 4 nodes each, using first order IGA beams, whereas the *RC* beam is modeled with 12 nodes, using third order IGA beams.

We performed the dynamical analysis both with a conventional HHT time integrator and with the DVI time stepper of (54)-(55). Comparing the results with the reference data obtained with MBDyn in Figs. 12 and 13, a remarkable fact is that the lateral buckling is triggered exactly at the same moment for all the formulations, and the resulting oscillations have the same period. However, as expected, the DVI time stepper introduces some numerical damping. In fact, in scenarios that do not involve frictional contacts as in this benchmark, the conic complementarity constraint (55) disappears and the DVI time stepper boils down to a linearly-implicit first-order scheme, hence it shows the same damping effect of an implicit Euler method [57]. When using the HHT second-order implicit time integrator, results are affected by numerical damping to a lower degree. The drawback of the numerical damping in the DVI method can still be accepted when its superior stability and its efficiency in contact problems are attractive, as shown in the following example.

8.5. *Contacts with rigid body*

This benchmark shows that the proposed IGA beam performs well in a dynamical simulation with complex spatial contacts, especially when using the non-smooth formulation embedded in the DVI time stepper. A bundle of IGA beams has been fixed between two shafts, one of which is rotating at constant speed. A fixed central rod has been added, so that it will be wrapped by the IGA beams during the simulation, see Fig. 14.

The bundle consists in a set of eight beams, each being an IGA beam of third order with 57 nodes, for a total length of 0.5 m. The section is circular with a diameter of 0.01 m, the Young modulus is $E = 0.5$ GPa, the shear modulus $G = 0.35$ GPa, the density is $\rho = 1000$ kg m⁻³. The internal cylinder diameter is 0.5 m, while the beams are distributed on a circular pattern of diameter 0.5 m. The rotation speed is 1 rad s⁻¹.

The interior-point solver used in this simulation is request to achieve a tolerance of 1×10^{-10} over the residuals and complementarity gap.

The dynamical analysis has been performed using the DVI time stepper (54)-(55). For this non-smooth dynamics problem, frictional contacts are enforced as complementarity constraints that do not require any tuning of penalty. Note that when using conventional implicit or explicit integrators for smooth dynamics, high penalty stiffness would be needed to approximate contact between rigid materials without contact compenetration, but in turn this requires very short time steps to avoid numerical instability.

Multiple tests have been run with increasing time steps, from 1 ms up to 75 ms. This range includes values that are unusually large for this type of analysis; just for reference we report that we ran the same benchmark using the HHT time stepper and penalty contacts, but to avoid divergence the time step could not be larger than $h = 1 \times 10^{-5}$ s.

The forces induced by beams on the shafts have been reported in Fig.15 and the relative error respect to a reference solution (computed with much shorter time step) is shown: being always under 3% even for the largest time step is a good proof for the robustness of the DVI solver. It is remarkable that the

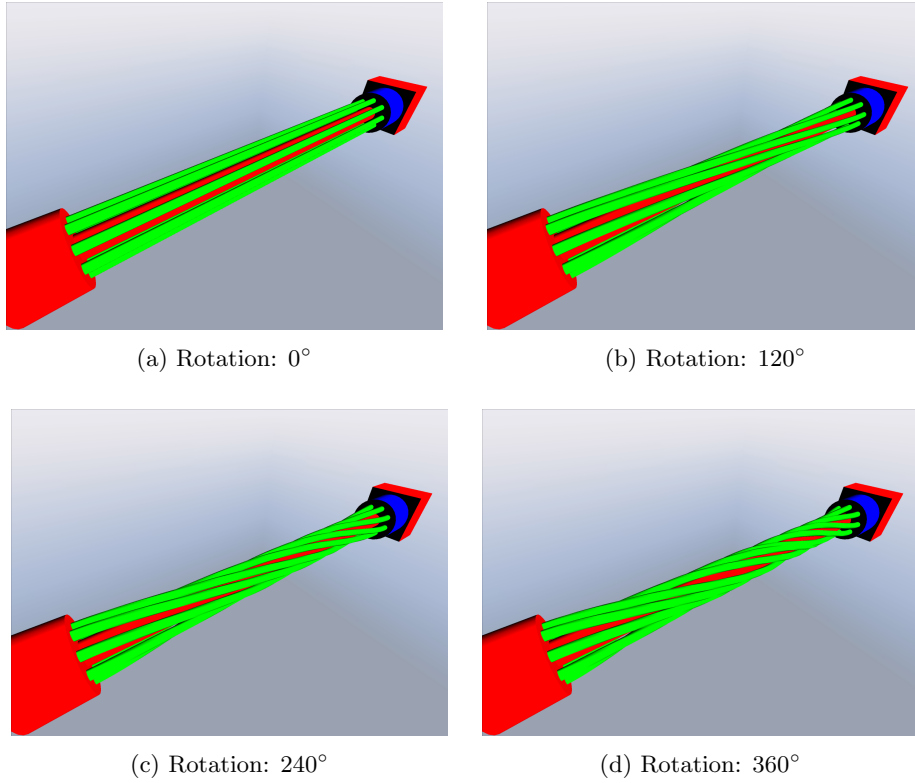


Figure 14: Contact with rigid body (in red, elements that are fixed; in blue, the rotating shaft; in green, the IGA beams).

solver was stable even for time steps larger than 100 ms, but at that point the precision started to be compromised by other factors such as the precision of the collision detection algorithms etc.

8.6. Woven mesh

In order to assess the performance of the DVI method for the case of mutual contact between IGA beams, a woven mesh of IGA beams has been reproduced and compared to literature [62]. The experiment consists in a 7×7 mesh where wires are clamped at one end, free at the other. Each wire is modeled with 64 nodes and a third-order IGA rod with cross-sectional radius of 1 mm, Young modulus $E = 10 \times 10^8$ and Poisson ratio of $\nu = 0.5$, for a total length $L = 4L_w = 0.12$ m where $L_w = 0.03$ m is the wavelength of the curve used to weave

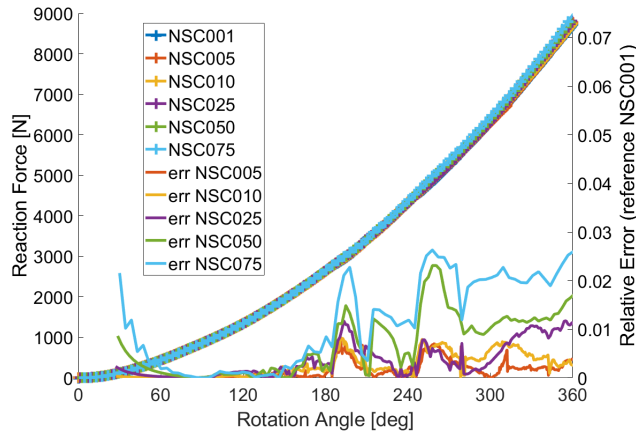


Figure 15: Reaction force on the shaft for different time steps. In the legend, labels like NSC010 means Non-Smooth Contact test with time step $h = 10$ ms

the mesh. Because of the non-smooth contact model no additional compliance parameter is required.

A distributed load of 0.1 N/m is gradually applied to the beams in the vertical direction, and the DVI method is used to perform a non-linear analysis with a time step of 10 ms using the Interior-Point solver described in the previous example. Results are shown in Fig.16. We remark that the time step could be increased up to 50 ms without incurring in divergence. Although the tolerance of the solver has been kept at the very strict threshold of 1×10^{-10} on residuals and complementarity, the computational time never exceeded 0.997 s/step on a 2.4 GHz Intel i7-4700 processor.

9. Conclusions

A geometrically-exact spatial Cosserat rod has been implemented via IGA discretization, using quaternions as rotational coordinates. We implemented the model in a open-source physics library using C++ language and we tested it in various non-linear static and dynamic problems, showing generality, robustness and efficiency of the formulation.

Using a DVI time stepping scheme and a SO-CCP solver, we proved that simulations with non-smooth frictional contacts between IGA beams and moving

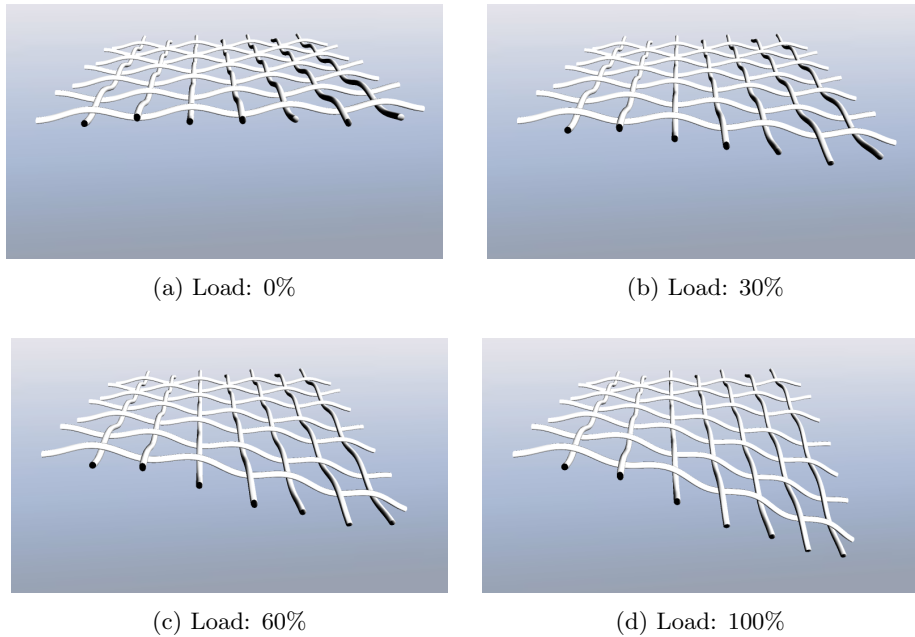


Figure 16: Snapshots from the simulation of the woven mesh, for increasing load.

shapes can be performed with large time steps. The enhanced robustness and stability indicate that the DVI method is a viable alternative to conventional methods based on contact penalty.

References

- [1] C. Adam, T.J.R. Hughes, S. Bouabdallah, M. Zarroug, and H. Maitournam. Selective and reduced numerical integrations for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:732 – 761, 2015. Isogeometric Analysis Special Issue.
- [2] Martin Andersen, Joachim Dahl, Zhang Liu, Lieven Vandenbergh, S Sra, S Nowozin, and SJ Wright. Interior-point methods for large-scale cone programming. *Optimization for machine learning*, 5583, 2011.
- [3] Mihai Anitescu and Gary D. Hart. A fixed-point iteration approach for multibody dynamics with contact and friction. *Mathematical Programming, Series B*, 101(1)(ANL/MCS-P985-0802):3–32, 2004.

- [4] Stuart S. Antman. Kirchhoff's problem for nonlinearly elastic rods. *Quarterly of Applied Mathematics*, 32(3):221–240, 1974.
- [5] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina, and A. Reali. Locking-free isogeometric collocation methods for spatial Timoshenko rods. *Computer Methods in Applied Mechanics and Engineering*, 263:113 – 126, 2013.
- [6] O. A. Bauchau, G. Wu, P. Betsch, A. Cardona, J. Gerstmayr, B. Jonker, P. Masarati, and V. Sonnevile. Validation of flexible multibody dynamics beam formulations using benchmark problems. In *IMSD 2014*, Korea, June 30-July 3 2014.
- [7] Olivier A. Bauchau and N. K. Kang. A multibody formulation for helicopter structural dynamic analysis. *Journal of the American Helicopter Society*, 38(2):3–14, 1993.
- [8] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43(1):3–37, Dec 2008.
- [9] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5):229 – 263, 2010. Computational Geometry and Analysis.
- [10] Jan Bender, Kenny Erleben, and Jeff Trinkle. Interactive simulation of rigid body dynamics in computer graphics. In *Computer Graphics Forum*, volume 33, pages 246–270. Wiley Online Library, 2014.
- [11] D. J. Benson, Y. Bazilevs, M. C. Hsu, and T. J.R. Hughes. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):276–289, 2010.

- [12] Robin Bouclier, Thomas Elguedj, and Alain Combescure. Locking free isogeometric formulations of curved thick beams. *Computer Methods in Applied Mechanics and Engineering*, 245-246:144 – 162, 2012.
- [13] Olivier Bruls, Vincent Acary, and Alberto Cardona. On the constraints formulation in the nonsmooth generalized- α method. In *Advanced Topics in Nonsmooth Dynamics*, pages 335–374. Springer, 2018.
- [14] Mourad Chamekh, Saloua Mani-Aouadi, and Maher Moakher. Modeling and numerical treatment of elastic rods with frictionless self-contact. *Computer Methods in Applied Mechanics and Engineering*, 198(47):3751 – 3764, 2009.
- [15] F. Cosserat and E. Cosserat. *Thorie des corps dformables*. A. Hermann et fils, 1909.
- [16] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196(41):4160 – 4183, 2007.
- [17] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5257 – 5296, 2006. John H. Argyris Memorial Issue. Part II.
- [18] M.A. Crisfield. A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements. *Computer Methods in Applied Mechanics and Engineering*, 81(2):131 – 150, 1990.
- [19] L. Beiro da Veiga, C. Lovadina, and A. Reali. Avoiding shear locking for the Timoshenko beam problem via isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 241-244:38 – 51, 2012.
- [20] L. De Lorenzis, P. Wriggers, and G. Zavarise. A mortar formulation for 3d large deformation contact using NURBS-based isogeometric analysis and

- the augmented lagrangian method. *Computational Mechanics*, 49(1):1–20, Jan 2012.
- [21] E. H. Dowell and J. J. Traybar. An experimental study of the nonlinear stiffness of a rotor blade undergoing flap, lag, and twist deformations. Aerospace and Mechanical Science Report 1194, Princeton University, January 1975.
- [22] E. H. Dowell and J. J. Traybar. An experimental study of the nonlinear stiffness of a rotor blade undergoing flap, lag, and twist deformations. Aerospace and Mechanical Science Report 1257, Princeton University, December 1975.
- [23] C.A. Felippa and B. Haugen. A unified formulation of small-strain corotational finite elements: I. theory. *Computer Methods in Applied Mechanics and Engineering*, 194(21?24):2285 – 2335, 2005. Computational Methods for Shells.
- [24] Alfredo Gay Neto, Paulo M. Pimenta, and Peter Wriggers. Self-contact modeling on beams experiencing loop formation. *Computational Mechanics*, 55(1):193–208, 2015.
- [25] Gian Luca Ghiringhelli, Pierangelo Masarati, and Paolo Mantegazza. A multi-body implementation of finite volume beams. *AIAA Journal*, 38(1):131–138, January 2000.
- [26] L. Greco and M. Cuomo. B-spline interpolation of Kirchhoff-Love space rods. *Computer Methods in Applied Mechanics and Engineering*, 256:251 – 269, 2013.
- [27] Toby Heyn, Mihai Anitescu, Alessandro Tasora, and Dan Negrut. Using Krylov subspace and spectral methods for solving complementarity problems in many-body contact dynamics simulation. *International Journal for Numerical Methods in Engineering*, 95(7):541–561, 2013.

- [28] M. Hillman, J.S. Chen, and Y. Bazilevs. Variationally consistent domain integration for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:521 – 540, 2015. Isogeometric Analysis Special Issue.
- [29] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135 – 4195, 2005.
- [30] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):301–313, 2010.
- [31] L. O. Jay and D. Negrut. Extensions of the HHT- α method to Differential-Algebraic Equations in mechanics. *Electronic Transactions on Numerical Analysis*, 26:190–208, 2007.
- [32] M. Jean and J. J. Moreau. Dynamics in the presence of unilateral contacts and dry friction: A numerical approach. In G. Del Piero and F. Maceri, editors, *Unilateral Problems in Structural Analysis — 2*, pages 151–196, Vienna, 1987. Springer Vienna.
- [33] Pascal Jung, Sigrid Leyendecker, Joachim Linn, and Michael Ortiz. A discrete mechanics approach to the Cosserat rod theory - part 1: static equilibria. *International Journal for Numerical Methods in Engineering*, 85(1):31–60, 2011.
- [34] Myoung-Jun Kim, Myung-Soo Kim, and Sung Yong Shin. A C^2 -continuous B-spline quaterion curve interpolating a given sequence of solid orientations. In *Proc. Computer Animation'95*, pages 72–81, 1995.
- [35] Myoung-Jun Kim, Myung-Soo Kim, and Sung Yong Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. In *Computer Graphics (Proc. SIGGRAPH'95)*, pages 369–376, 1995.

- [36] Alexander Konyukhov and Karl Schweizerhof. Geometrically exact covariant approach for contact between curves. *Computer Methods in Applied Mechanics and Engineering*, 199(37):2510 – 2531, 2010.
- [37] Holger Lang, Joachim Linn, and Martin Arnold. Multi-body dynamics simulation of geometrically exact Cosserat rods. *Multibody System Dynamics*, 25(3):285–312, 2011.
- [38] P Litewka and P Wriggers. Frictional contact between 3D beams. *Computational mechanics*, 28(1):26–39, 2002.
- [39] Przemyslaw Litewka and Peter Wriggers. Contact between 3D beams with rectangular cross-sections. *International Journal for Numerical Methods in Engineering*, 53(9):2019–2041, 2002.
- [40] Dario Mangoni, Alessandro Tasora, and Rinaldo Garziera. A primal-dual predictor-corrector interior point method for non-smooth contact dynamics. *Computer Methods in Applied Mechanics and Engineering*, 330:351 – 367, 2018.
- [41] Enzo Marino. Locking-free isogeometric collocation formulation for three-dimensional geometrically exact shear-deformable beams with arbitrary initial curvature. *Computer Methods in Applied Mechanics and Engineering*, 324:546 – 572, 2017.
- [42] Pierangelo Masarati. On the choice of the reference frame for beam section stiffness properties. *Intl. J. Solids Structures*, 51(13):2439–2447, June 2014. doi:10.1016/j.ijsolstr.2014.03.011.
- [43] H. Mazhar, T. Heyn, A. Pazouki, D. Melanz, A. Seidl, A. Bartholomew, A. Tasora, and D. Negrut. CHRONO: a parallel multi-physics library for rigid-body, flexible-body, and fluid dynamics. *Mechanical Sciences*, 4(1):49–64, 2013.

- [44] Hammad Mazhar, Toby Heyn, Dan Negrut, and Alessandro Tasora. Using Nesterov’s method to accelerate multibody dynamics with friction and contact. *ACM Transactions on Graphics*, 34(3):32:1–32:14, May 2015.
- [45] Christoph Meier, Alexander Popp, and Wolfgang A. Wall. A finite element approach for the line-to-line contact interaction of thin beams with arbitrary orientation. *Computer Methods in Applied Mechanics and Engineering*, 308:377 – 413, 2016.
- [46] Christoph Meier, Wolfgang A. Wall, and Alexander Popp. A unified approach for beam-to-beam contact. *Computer Methods in Applied Mechanics and Engineering*, 315:972 – 1010, 2017.
- [47] J. J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In J. J. Moreau and P. D. Panagiotopoulos, editors, *Nonsmooth Mechanics and Applications*, pages 1–82, Berlin, 1988. Springer-Verlag.
- [48] D. Negrut, B. Gavrea, and F. Potra. Multibody dynamics simulation with Newmark integrators. *In preparation*, 2005.
- [49] Dan Negrut, Radu Serban, and Alessandro Tasora. Posing multibody dynamics with friction and contact as a differential complementarity problem. *Journal of Computational and Nonlinear Dynamics*, 13(1):014503, 2018.
- [50] Dan Negrut, Alessandro Tasora, Hammad Mazhar, Toby Heyn, and Philipp Hahn. Leveraging parallel computing in multibody dynamics. *Multibody System Dynamics*, 27:95–117, 2012.
- [51] J. S. Pang and D. E. Stewart. Differential variational inequalities. *Mathematical Programming*, 113:1–80, 2008.
- [52] Friedrich Pfeiffer and Christian Glocker. *Multibody Dynamics with Unilateral Contacts*. John Wiley, New York City, 1996.
- [53] E. Reissner. On one-dimensional large-displacement finite-strain beam theory. *Studies in Applied Mathematics*, 52(2):87–95, 1973.

- [54] J. C. Simo. A finite strain beam formulation. the three-dimensional dynamic problem. part I. *Comput. Meth. Appl. Mech. Engng.*, 49:55–70, 1985.
- [55] J.C. Simo and L. Vu-Quoc. A three-dimensional finite-strain rod model. part II: Computational aspects. *Computer Methods in Applied Mechanics and Engineering*, 58(1):79 – 116, 1986.
- [56] D.E. Stewart. Convergence of a time-stepping scheme for rigid body dynamics and resolution of Painlevé’s problems. *Archive Rational Mechanics and Analysis*, 145(3):215–260, 1998.
- [57] Alessandro Tasora and Mihai Anitescu. A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):439 – 453, 2011.
- [58] I. Temizer, M.M. Abdalla, and Z. Gürdal. An interior point method for isogeometric contact. *Computer Methods in Applied Mechanics and Engineering*, 276:589 – 611, 2014.
- [59] S.P. Timoshenko. On the correction for shear of the differential equation for transverse vibrations of prismatic bars. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(245):744–746, 1921.
- [60] A.-V. Vuong, C. Giannelli, B. Jttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49):3554 – 3567, 2011.
- [61] Oliver Weeger, Bharath Narayanan, Laura DeLorenzis, Josef Kiendl, and MartinL. Dunn. An isogeometric collocation method for frictionless contact of Cosserat rods. *Computer Methods in Applied Mechanics and Engineering*, 321:361 – 382, 2017.

- [62] Oliver Weeger, Bharath Narayanan, and Martin L Dunn. Isogeometric collocation for nonlinear dynamic analysis of Cosserat rods with frictional contact. *Nonlinear Dynamics*, 91(2):1213–1227, 2018.
- [63] Oliver Weeger, Utz Wever, and Bernd Simeon. Isogeometric analysis of nonlinear Euler–Bernoulli beam vibrations. *Nonlinear Dynamics*, 72(4):813–835, Jun 2013.
- [64] Oliver Weeger, Sai-Kit Yeung, and Martin L. Dunn. Isogeometric collocation methods for Cosserat rods and rod structures. *Computer Methods in Applied Mechanics and Engineering*, 316:100 – 122, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
- [65] P. Wriggers and G. Zavarise. On contact between three-dimensional beams undergoing large deflections. *Communications in Numerical Methods in Engineering*, 13(6):429–438, 1997.
- [66] G. Zavarise and P. Wriggers. Contact with friction between beams in 3-D space. *International Journal for Numerical Methods in Engineering*, 49(8):977–1006, 2000.

Appendix A. Constitutive models for beams

In the most generic case, (15) may be a non-linear function, but in the following we present special cases of practical interest.

Appendix A.1. Generic linear elasticity

The generic case of constitutive model for the Cosserat rod, with linear elasticity, requires a matrix $\mathbf{E} \in \mathbb{R}^{6 \times 6}$, not necessarily sparse. Such matrix can be provided by some detailed 3D FEA analysis of a chunk of beam, in a preprocessing stage.

$$\begin{pmatrix} \mathbf{n} \\ \mathbf{m} \end{pmatrix} = \mathbf{E} \begin{pmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\kappa} \end{pmatrix} \tag{A.1}$$

We remark that, depending on the 36 values used in the 6x6 \mathbf{E} matrix, \mathbf{n} and \mathbf{m} might have coupled effects. Figure A.17 shows the reference coordinate system of the section.

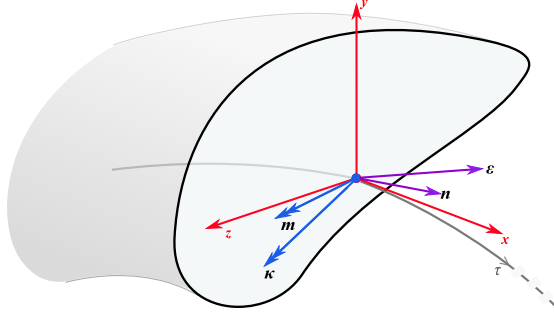


Figure A.17: Section of the beam. Generic case.

Appendix A.2. Basic diagonal linear elasticity

For centered symmetric sections, the previous relation can be simplified. In fact one can express the constitutive relation via a linear mapping where \mathbf{n} and \mathbf{m} effects are uncoupled:

$$\mathbf{n} = \mathbf{n}(\boldsymbol{\epsilon}) \quad (\text{A.2})$$

$$\mathbf{m} = \mathbf{m}(\boldsymbol{\kappa}) \quad (\text{A.3})$$

In detail, with linear elasticity, a very common case is the linear mapping:

$$\mathbf{n} = \mathbf{C}\boldsymbol{\epsilon} \quad (\text{A.4})$$

$$\mathbf{m} = \mathbf{D}\boldsymbol{\kappa} \quad (\text{A.5})$$

where one simply uses the material matrices:

$$\mathbf{C} = \begin{bmatrix} EA & 0 & 0 \\ 0 & GAk_y & 0 \\ 0 & 0 & GAk_z \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} GJ & 0 & 0 \\ 0 & EI_y & 0 \\ 0 & 0 & EI_z \end{bmatrix} \quad (\text{A.6})$$

for given material Young's modulus E , shear modulus G , area A , Timoshenko shear correction factors k_y, k_z , torsion constant J , and second moments of area I_y, I_z computed in the section reference. Note that the center of axial forces C_a and the shear center C_s are in the origin, by assumption. See Fig. A.18.

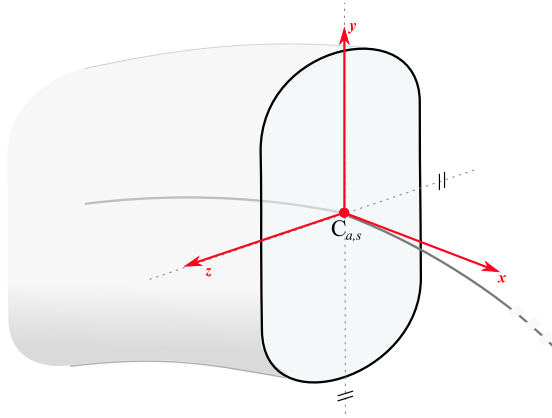


Figure A.18: Section of the beam. Simplified case (diagonal \mathbf{E}).

Appendix A.3. Advanced section for linear elasticity

A more general constitutive model is the one depicted in Fig. A.19, where the I_y, I_z are computed respect to an auxiliary reference C_a , center of axial forces, that has displacement y_a, z_a and rotation α respect to the reference center line of the beam [42].

Also, in case of non-symmetric sections, it may happen that the shear center C_s does not coincide with C_a ; if so, one can provide displacements y_s, z_s and rotation β of the reference of the shear center C_s respect to the reference center line of the beam.

Usually, for symmetric sections, it tends to $y_s = y_a, z_s = z_a, \alpha = \beta$. For simple problems like rectangular or circular sections centered on the reference line of the beam, one has $y_s = y_a = 0, z_s = z_a = 0, \alpha = \beta = 0$.

The linear model becomes:

$$\begin{Bmatrix} \mathbf{n} \\ \mathbf{m} \end{Bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 & 0 & a_{12} & a_{13} \\ 0 & s_{11} & s_{12} & s_{13} & 0 & 0 \\ 0 & s_{21} & s_{22} & s_{23} & 0 & 0 \\ 0 & s_{31} & s_{32} & s_{33} & 0 & 0 \\ a_{21} & 0 & 0 & 0 & a_{22} & a_{23} \\ a_{31} & 0 & 0 & 0 & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\kappa} \end{Bmatrix} \quad (\text{A.7})$$

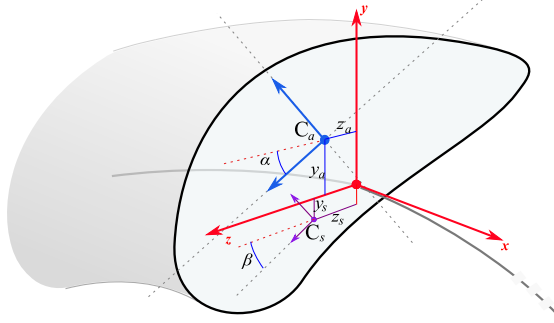


Figure A.19: Section of the beam. Advanced case.

Here the components $a_{ij} = \mathbf{A}$ and $s_{ij} = \mathbf{S}$ are obtained by rotations \mathbf{R} and translations \mathbf{T} of the diagonal constitutive matrices \mathbf{A}_{C_a} and \mathbf{S}_{C_s} :

$$\mathbf{A} = \mathbf{T}_{C_a} \mathbf{R}_{C_a} \mathbf{A}_{C_a} \mathbf{R}_{C_a}^T \mathbf{T}_{C_a}^T \quad (\text{A.8})$$

$$\mathbf{S} = \mathbf{R}_{C_s}^T \mathbf{T}_{C_s}^{-1} \mathbf{S}_{C_s} \mathbf{T}_{C_s} \mathbf{R}_{C_s} \quad (\text{A.9})$$

where

$$\mathbf{A}_{C_a} = \begin{bmatrix} EA & 0 & 0 \\ 0 & EI_y & 0 \\ 0 & 0 & EI_z \end{bmatrix} \quad \mathbf{S}_{C_s} = \begin{bmatrix} GAk_y & 0 & 0 \\ 0 & GAk_z & 0 \\ 0 & 0 & GJ \end{bmatrix} \quad (\text{A.10})$$

The above model requires the following parameters: Young modulus E , shear modulus G , area A , Timoshenko shear correction factors k_y, k_z , torsion constant J , and second moments of area I_y, I_z , as the diagonal simplified model, plus the position and rotation of C_a as y_a, z_a and α , plus the position and rotation of C_s as y_s, z_s and β .