Available online at www.sciencedirect.com

ScienceDirect

Computer-Aided Design 40 (2008) 35–48

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Geometrically exact dynamic splines

A. Theetten[a,*], L. Grisoni[a], C. Andriot[b], B. Barsky[c]

[a] *LIFL, IRCICA/INRIA Futurs, USTL, France*
[b] *CEA, Fontenay-aux-Roses, France*
[c] *University of California, Berkeley, USA*

Received 1 October 2006; accepted 3 May 2007

Abstract

In this paper, we propose a complete model handling the physical simulation of deformable 1D objects. We formulate continuous expressions for stretching, bending and twisting energies. These expressions are mechanically rigorous and geometrically exact. Both elastic and plastic deformations are handled to simulate a wide range of materials. We validate the proposed model in several classical test configurations. The use of geometrical exact energies with dynamic splines provides very accurate results as well as interactive simulation times, which shows the suitability of the proposed model for constrained CAD applications. We illustrate the application potential of the proposed model by describing a virtual system for cable positioning, which can be used to test compatibility between planned fixing clip positions, and mechanical cable properties.

Keywords: Spline; Beam theory; Elastic; Plastic; Lagrange equations

1. Introduction

One-dimensional flexible models are a key CAD element in number of practical situations. Cables of largely varying mechanical properties are nowadays used in industry. In fields such as car and plane design, virtual prototyping is used to improve quality and to reduce development costs. As a matter of fact, virtual prototyping includes more and more assembly simulations: it allows the early detection of potential problems, and also permits the study of ease of assembly. This implies the ability to accurately represent geometry, but also the mechanical behavior of involved parts. Among the many objects to be simulated, flexible one-dimensional objects are of significant importance. They are involved in vehicle engineering (e.g. electrical cable laying within the car structure [1]), but also in fields such as architecture (e.g. stiff electrical cable positioning within virtual buildings), or even in medical simulation (surgical thread simulation is currently an active research question in the medical simulation community [2]). For most unconstrained Computer Aided Design applications, splines are probably the most classical tool for 1D objects. As a matter of fact, NURBS have become an industry-standard representation for 1D objects. *Dynamic splines* have been introduced by Qin and Terzopoulos [3]. They combine physics-based constraining equations with spline geometry, in order to improve the design process. In this article, we propose an approach that extends the mechanical accuracy of previously proposed approaches. We propose, wherever possible, *geometrically exact* formal expressions that, along with spline analytical expressions, to provide a powerful, real-time model. We call this model *Geometrically Exact Dynamic Splines*, or GEDS for short.

In this paper, we propose a spline-based model for the real-time, mechanically accurate, simulation of one-dimensional objects. Our model can handle both reversible (*elastic*) and irreversible (*plastic*) deformations. The proposed formalism and energy expressions model stretching, bending and twisting loads; at the very limit of material constraint, we show how our model can be used to detect break points. We also show that the interactive rate is provided for a wide range of configurations. Finally, we describe a practical application of our model, oone that permits us to virtually validate electrical cable positioning and clipping along a path on a car door. Our specific scientific contributions are the following:

[*] Corresponding author. Tel.: +33 620755250.
E-mail addresses: adrien.theetten@lifl.fr (A. Theetten), laurent.grisoni@lifl.fr (L. Grisoni), claude.andriot@cea.fr (C. Andriot), barsky@cs.berkeley.edu (B. Barsky).

0010-4485/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2007.05.008

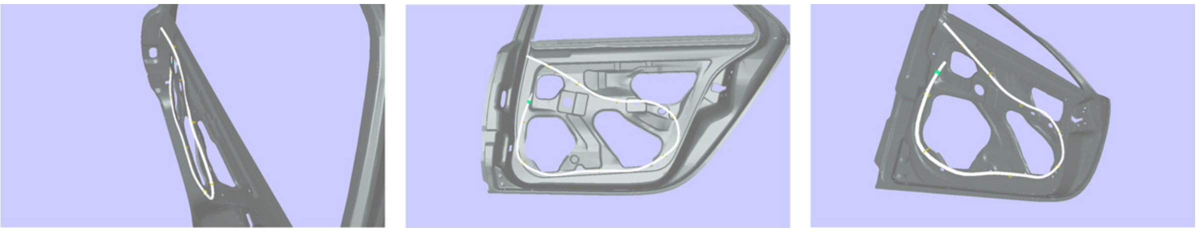


Fig. 1. Virtual cable positioning on a car door.

- An improvement in the dynamic spline state-of-the-art, namely a formulation of stretching, twisting and bending deformations in large rotations (or large displacements), through geometrically exact energy expressions. Such terms allow accurate results, while still running in interactive times. The proposed method is fully compatible with Lagrangian multipliers;
- The handle of twisting with an extended spline formulation, by decomposition of twisting in a *geometrical* part and a *roll* part. Such a separation ensures numerical stability for twisting energy evaluation. In addition to that, solving of the proposed mechanical model does not require a local frame, which makes it all the more accurate (frames are classically stabilized along a 1D curve using non-mechanical methods, see Section 8);
- An easy and efficient inclusion of plasticity within the Lagrange spline model.

The remainder of the paper is organized as follows: next section describes related work. In Section 3, we provide a short summary of elasticity and plasticity theory, which constitutes the core mechanical knowledge for understanding the remainder of the article. Then we define in Section 4 the formalism we use to describe geometrical model configurations. In Section 5, we propose a method (including elegant formalism) for handling elastic deformations of the model in a geometrically exact manner. In Section 6, we show how elastic deformation simulation can be combined with plastic behavior detection and simulation. For the sake of completeness, we provide in Section 7 the (classical) tools we use to handle the world's interaction with the 1D model. Section 8 provides some comments about twisting handling in the proposed deformation model, which is one of the very crucial points in the method. Finally, Section 9 describes tests and practical results: first, for ease of understanding, we provide a complete overview of the animation algorithm, which links the equations all together. Second, we compare numerical results of our model to several classical reference configurations. Third, we describe an advanced practical application of the model: a virtual system for electrical cable position testing on a car door (see visual example of Fig. 1).

2. Previous works

A constraints solver has now become a standard part of most CAD models, and is still a very active research field [4]; the range of applications of such techniques is potentially very large (e.g see [5]). Constraint solving most often relates to finding a compatible solution between user modeling requirements and pre-imposed geometric constraints. *Variational modeling* [6] minimizes the global energy of a constrained geometric deformable object, and can be seen as an introduction of physical behavior into constraint solving. Physics-based modeling is more and more involved in the field of constrained geometrical design: it permits one to extend constraints to the intrinsic mechanical properties of the modeled object.

The study of one-dimensional deformable objects has been a recurrent problem in computer graphics for about 20 years [7]. Many existing results, from the most computationally efficient to the most numerically accurate, try to capture the complexity of one-dimensional deformable models. Simple models, like particles, networks of mass-springs [8] or rigid articulated bodies [9], provide solutions on arbitrary linear discretization. Most of these models provide fast and interactive simulations, but are difficult to precisely tune for some given material, as their parameters do not directly relate to the coefficients provided by mechanics. These shortcomings have been addressed by developing more accurate models that underlie the physics. Inspired by classical numerical simulation methods, finite element and boundary element methods (see [10,11]) are known to yield accurate numerical results for fine discretization of the studied objects; such accuracy is difficult to combine with reasonable computation time in practice, however.

In the robotics community, several recent works use Cosserat theory. A Cosserat medium was first described in 1909 by Cosserat brothers [12]. This medium is described by a set of oriented micro-solids. Pai [13] first introduced Cosserat's rod theory in computer graphics to model cantilever objects. In this work, the animation step is done in two passes: the first one calculates the forces and torques iteratively along the rod discretization, and the second one evaluates the geometrical configuration in backward iterations. Wakamatsu [14] achieves a very accurate static solution of a cable simulation by considering it as a succession of oriented frames and by minimizing its potential energy; this approach is mechanically accurate, but demands a very high computation cost. Bertails et al. [15] define *Super-Helices* for simulating the dynamics of human hair strands. A Lagrangian formulation of inextensible Kirchhoff rods is used. This method is fast to compute and visually realistic for low resolution rods; the quadratic complexity of the algorithm is still a key problem for the real-time simulation of high complexity rods. The main drawback of these three methods is that it is difficult to combine such models with constraints. Moreover, these methods need at least a reference point for calculation, which might not always be available in practical cases (e.g. modeling rest state position of a deformable cable within a recipient shape). Grégoire [16] very recently proposed a mass-spring model based on Cosserat theory, which resembles realistic and interactive twisting and bending deformations. This model uses consistent mechanics, and is based on simple, continuous, energy terms that provide efficient computation time; it is anyhow unclear in which measure this model is numerically accurate, when compared to real-life objects with known material properties.

Spline-based techniques are still quite isolated within the physical animation literature. Terzopoulos et al. initiated deformable models in Computer Graphics, including physics-based curves [7], using a Lagrangian form of Newton's equation. The model of Qin and Terzopoulos [3], Dynamic Non-Uniform Rational B-Splines (DNURBS), first combined spline representation with physics laws. Nocent and Rémion [17] define the *Dynamic Material Splines* (DMS), a full Lagrange-based simulation framework for splines. They consider spline control points as the degrees of freedom of the underlying continuous object. Continuous stretching energy is defined. Lagrange multipliers are used to constrain point position and tangent orientation. Lenoir [18] propose a curvature energy formulation for DMS that is not geometrically exact, but provides real-time manipulations, as well as adaptive simulations [19]. We do not address in the present article the simulation of fractures: Lenoir et al. [19] provide an elegant manner to compute topological changes, by combining Lagrangian mechanics with B-Splines knot-refinement properties.

None of the previously mentioned works deals with irreversible deformations. One-dimensional objects like electrical wires, telephone cables, and suturing threads remain partially deformed after relaxation. Few papers deal with plasticity. Terzopoulos and Fleischer [20] first introduced non-elastic behavior in the Computer Graphics community. They proposed physically-based models to simulate viscoelasticity, plasticity and fracture effects, for the purposes of animation, in the case of volumic objects. [21] describes a point-based method for animation that can handle fracture on shells. O'Brien et al. [22] propose a method for realistic ductile fracture animation in common solid materials, such as plastics and metals. In the present work, we use a similar decomposition of strains to handle plasticity.

3. Mechanics background

This section provides a very short overview of the mechanical background, which is necessary for understanding the extension of Dynamic spline we propose in the next section.

When a force is applied to a deformable object, object geometry is extended or compressed, and local topology may even change (i.e. material might break) if the force is large enough. Although both load and extension/compression are primary quantities, material scientists tend to use two derived quantities, *stress* σ and *strain* ε to analyze materials under load. Stress is the force per unit area and strain is the extension/compression per unit length. In the general case, both σ and ε are manipulated using tensors. However, in most practical cases, the relationship between the stress applied to a material and the resulting strain is described in mechanics by a simple relation separately *in each direction*: a curve can be produced that is characteristic of the material. Fig. 2 is a typical representation of a stress–strain curve.

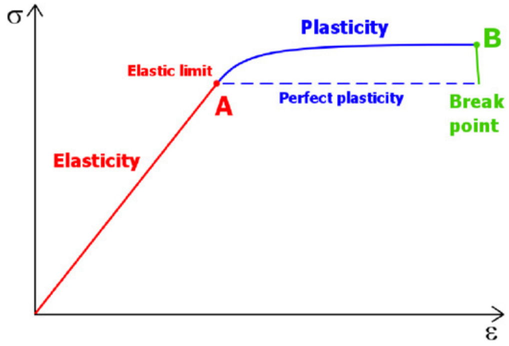


Fig. 2. Schematic representation of a typical stress–strain curve. Elasticity is usually linear, whereas plasticity is assumed to be perfect.

The behavior of the material fully depends on the strain magnitude: *elastic* (i.e. reversible) deformations occur for small strains, and *plasticity* (i.e. non-reversible) for more significant strain magnitudes. When strain intensity reaches the *fracture* point, the geometry of the local topology changes, and continuous mechanics can no longer be applied.

3.1. Elasticity

When strain magnitude is relatively small (assumption of *small strains* in mechanics), it can be considered as directly proportional to stress (*Hooke law*). The elasticity domain only contains *reversible* deformations: when stress is relaxed, the material returns to its rest state. A nonlinear elastic region needs to be studied at some point, depending on the material: the mechanical model is then said to be defined for *large strains*. In practice, a wide range of materials available in real-life one-dimensional objects do not need such non-linear methods for good simulation accuracy. A a result, we only consider in this paper *small strains* (we show in Section 9.2 that this approximation anyhow allows to reach good mechanical accuracy).

Several parameters characterize an elastic material. The longitudinal rate and the transversal rate between a stress and a related strain (i.e. the strain in the direction and the strain orthogonal to the stress, respectively) are given by *Young's modulus* E (longitudinal elasticity modulus) and *shear modulus* G (transversal elasticity modulus), respectively. These two moduli are interrelated by a formula incorporating Poisson's ratio ν, which links longitudinal and transversal rates:

$$G = \frac{E}{2(1+\nu)}. \tag{1}$$

The units of the two moduli are Pascals (1 Pa $=$ 1 N m^{-2}), whereas Poisson's ratio is dimensionless; specific values for several materials are provided in Table 1. These parameters completely identify a given material.

Table 1
Physical parameters for several materials [23]

Material	E (GPa)	ν	μ (Mg/m^3)
Nylon	2.7	0.33	1.1
Alluminium alloys	71.7	0.34	2.8
Copper	120.7	0.35	8.9
Steel, carbon	206.8	0.28	7.8
Steel, stainless	189.6	0.28	7.8

3.2. Plasticity and fracture

If the stress in a material exceeds the *elastic limit* A, also called the yield point, for the material, then the stress is no longer linearly proportional to strain, and the deformations become non-reversible; when the stress is relaxed, the material remains partially deformed. After a plastic deformation, the strain–stress curve and the elastic region is translated along the strain axis. In the case of *perfect* plasticity, measured stress is independent of the applied strain: external work above the threshold value is fully turned into material, non-reversible, deformations. Under growing strain, when material strain limit is reached, a break (potentially fracture if applied strain is more important) occurs.

These material properties are handled by simulation using equations that define the dynamic relation between the geometry of the object (i.e. its deformations) and the applied forces.

3.3. Energy balance

The *principle of least action*, first formulated by de Maupertuis in 1747, is that nature always finds the most efficient path from one configuration to another. The *Lagrange equations* are deduced from this principle. They involve the *kinetic energy* T and the *potential energy* U of the system. The kinetic energy is the energy of motion, whereas the potential energy is the stored energy of position possessed by an object. $\mathbf{F}$ is the sum of external forces. Assuming the mass distribution to be homogeneously distributed between the n degrees of freedom $\mathbf{q_i}$ of the object, the Lagrange equation, which can be used for movement resolving, is formulated as follows:

$$\forall i \in \{1, \ldots, n\}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_\mathbf{i}}\right) = \mathbf{F}_i - \frac{\partial U}{\partial \mathbf{q_i}} \tag{2}$$

where $\mathbf{F}_i$ is the sum of the generalized external forces at $\mathbf{q}_i$ (see 7 for more details).

In the case of Dynamic Splines, control points are used as the degrees of freedom if the object.

4. GEDS definition

4.1. Beam geometry definition

Beam theory is the study of one-dimensional objects in mechanics. Consider a *cross-section* of diameter D and area S, as shown in Fig. 3. The *neutral fiber* or neutral axis, denoted f, is the oriented curve of length L that passes through the center of every cross-section. The volume defined by these cross-sections is a *beam*.

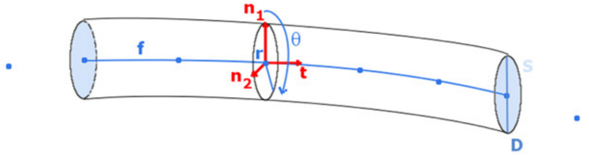


Fig. 3. Scheme of a beam with its geometrical parameters and local frame.

4.2. Spline formulation

The beam configuration is entirely described by two fields: a position field $\mathbf{r} = (x, y, z)$, which determines the neutral fiber f position, and a rotation field θ, which provides the roll of the cross-section. We propose to combine these two fields in a unique field described by a set of polynomial spline curves: $\mathbf{q} = (\mathbf{r}, \theta) = (x, y, z, \theta)$. Each resulting spline is given by

$$\mathbf{q}(u) = \sum_{i=1}^{n} b_i(u)\mathbf{q_i} \tag{3}$$

where b_i are the ith spline basis functions of the control points $\mathbf{q_i}$, and u is between 0 and ℓ, the length of the neutral fiber. The jth derivative of q with respect to u is given by:

$$\mathbf{q}^{(\mathbf{j})}(u) = \sum_{i=1}^{n} b_i{}^{(j)}(u)\mathbf{q_i}. \tag{4}$$

Arc length is denoted by s. The derivative of control point $\mathbf{q}$, position $\mathbf{r}$ and roll θ, with respect to u, are denoted by $\mathbf{q}'$, $\mathbf{r}'$ and θ' respectively. The displacement elements ds and du are interrelated by $\mathrm{d}s = \|r'\|\mathrm{d}u$.

Since control points completely define the position of the spline and the orientation of the cross-sections, they can be considered as the degrees of freedom of the system and used in the Lagrange equation (2).

Any kind of spline can potentially be used. In our test and without loss of generality, we used two different spline models: the cubic interpolatory Catmull–Rom spline, and the non-uniform rational B-spline (NURBS). The first model has the advantage of interpolating control points, which makes it quite easy to handle; in addition to that, scene definition is quite simple. By comparison, NURBS demand specific conversion numerical schemes to evaluate control points from sampling. Yet, this model has the nice potential feature of being available for geometric exact subdivision, which makes it interesting for further adaptive methods [19] (which is beyond the scope of this paper).

Once we have described our one-dimensional object geometrically with splines, we may now use mechanics to make it evolve in space. In the following subsection, we define the physical part of the spline, using linearly elastic but geometrically exact deformations.

5. GEDS in the elastic domain

To obtain the motion of control points with the Lagrange equations, deformation energies must be first formulated from physical parameters, and then differentiated with respect to the degrees of freedom. In this section, we propose a unified

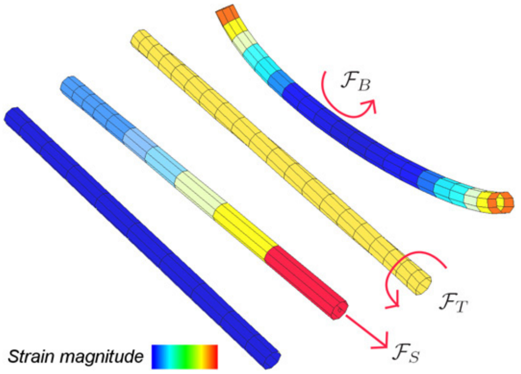


Fig. 4. From left to right: rest state, stretching, twisting and bending on a beam.

formulation to describe the deformations of a one dimensional object and the exact calculation of the corresponding forces.

5.1. Constitutive laws and strain energies

Every action on a beam can be modeled by forces and torques on the neutral fiber f; we express the force and the torque in the local frame instead of using stresses. They are here proportional to stresses, and moreover are easier to manipulate.

We elucidate here some terminology and notation, as shown in Fig. 4.

The *normal force* to the cross-section, which results in stretching, is denoted by $\mathcal{F}_S$. The cross-section may rotate around the neutral fiber, due to the *torsional torque*, denoted by $\mathcal{F}_T$. The *bending torque*, denoted by $\mathcal{F}_B$, corresponds to the oriented curvature of the neutral fiber f. The *Kirchhoff assumption* presumes that cross-sections are stiff; only the neutral axis is distorted : we neglect the shearing force in the plane of the cross-section. This yields the *Rayleigh model*.

Let us define the vector $\mathcal{F}$ as the data of $\mathcal{F}_S$, $\mathcal{F}_T$ and $\mathcal{F}_B$:

$$\mathcal{F} = \begin{pmatrix} \mathcal{F}_S \\ \mathcal{F}_T \\ \mathcal{F}_B \end{pmatrix} \tag{5}$$

$\mathcal{F}$ is related to the strain ϵ. The rest of the strain is denoted ϵ^0.

The elastic relationship between ϵ and $\mathcal{F}$ is well described in beam theory by Courbon [24]. Considering that linear elasticity facilitates the calculation of strain energies of a beam, as strains are directly proportional to stresses, we use the *small strain* assumption, which is accurate if the radius of curvature is large relative to the radius of the cross-section (usually about 5 times).

From these considerations, we provide the following result, which derives from Hooke's law:

$$\mathcal{F} = \mathbb{H}\left(\epsilon - \epsilon^0\right) = \begin{pmatrix} ES & 0 & 0 \\ 0 & GI_o & 0 \\ 0 & 0 & EI_s \end{pmatrix} \left(\epsilon - \epsilon^0\right) \tag{6}$$

where I_o is the polar momentum of inertia whereas I_s is the cross-section momentum of inertia; ES, GI_o and EI_s are the stretching, the twisting and the bending rigidities respectively (see [24]). We call $\mathbb{H}$ the Hooke matrix.

Assuming the cross-section is circular and its diameter constant, we can obtain the following expression for $\mathbb{H}$:

$$\mathbb{H} = \frac{D^2\pi}{4} \begin{pmatrix} E & 0 & 0 \\ 0 & \dfrac{GD^2}{8} & 0 \\ 0 & 0 & \dfrac{ED^2}{16} \end{pmatrix} \tag{7}$$

Strain energy U is formulated by the following integration along the beam:

$$U = \frac{1}{2} \int_0^L \left(\epsilon - \epsilon^0\right)^t \mathcal{F} \mathrm{d}s. \tag{8}$$

Using expression (6) of $\mathcal{F}$, we get:

$$U = \frac{1}{2} \int_0^L \left(\epsilon - \epsilon^0\right)^t \mathbb{H} \left(\epsilon - \epsilon^0\right) \mathrm{d}s. \tag{9}$$

We now have all the background in mechanics necessary to determine the motion of our one-dimensional object by solving the Lagrange equation (2). In the two next subsections, we will study the two terms of the equations so as to obtain a numerical solution.

5.2. Handling twisting in dynamic splines

Since the one-dimensional object is specified by position and rotation, its kinetic energy comprises translational energy and rotational energy. Translation energy corresponds to the displacement of control points and rotation energy is due to the motion of cross-sections around the neutral axis. We define the inertia matrix, denoted $\mathbb{J}$, which is the same everywhere along the spline, since the diameter is constant:

$$\mathbb{J} = \begin{pmatrix} \mu & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & I_o \end{pmatrix} \tag{10}$$

μ corresponds to linear density and I_o to the polar momentum of inertia. We thus propose a simple definition of the spline kinetic energy T:

$$T = \frac{1}{2} \int_0^L \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t}^t \mathbb{J} \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} \mathrm{d}s \tag{11}$$

where ^{t} denotes a transpose. Differentiating kinetic energy T with respect to $\mathbf{q_i}$ yields the left term of the Lagrange equations:

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial T}{\partial \dot{\mathbf{q}}_i} = \frac{1}{2} \int_0^L \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t}^t \mathbb{J} \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t}}{\partial \dot{\mathbf{q}}_i} \mathrm{d}s. \tag{12}$$

Replacing $\mathbf{q}$ by the expression given in Eq. (3) in a similar way to that described by Nocent and Rémion [17], yields:

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial T}{\partial \dot{\mathbf{q}}_i} = \sum_{j=1}^{n} \mathbb{J} \int_0^L (b_i(s) b_j(s)) \mathrm{d}s \frac{\mathrm{d}^2 \mathbf{q}_j}{\mathrm{d}t^2}. \tag{13}$$

Considering a matrix $\mathbb{M}$ and vector A of components $\mathbb{M}_{i,j} = \mathbb{J} \int_0^L (b_i(s) b_j(s)) \mathrm{d}s$ and $\mathbf{A}_j = \frac{\mathrm{d}^2 \mathbf{q}_j}{\mathrm{d}t^2}$ respectively, this equation yields:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial T}{\partial \dot{\mathbf{q}}_{\mathbf{i}}} = \sum_{j=1}^{n} \mathbb{M}_{i,j}\mathbf{A}_j. \tag{14}$$

Considering all degrees of freedom, this sum or Kinetic term can consequently be written as a matrix–vector product:

$$\mathbb{M}\mathbf{A}. \tag{15}$$

To obtain Nocent and Rémion kinetic energy, replace $\mathbb{J}$ by the following matrix: $\begin{pmatrix} \mu & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \mu \end{pmatrix}$. The use of $\mathbb{J}$ considers twisting inertia.

5.3. Geometrically exact energy evaluation

In this subsection, we propose an expression of the derivatives of strain energies with respect to generalized coordinates. They compose the right hand term of the Lagrange equation (2);

$$P^i = -\frac{\partial U}{\partial \mathbf{q}_i} = -\frac{1}{2}\int_0^L \frac{\partial \left(\epsilon - \epsilon^0\right)^t \mathbb{H} \left(\epsilon - \epsilon^0\right)}{\partial \mathbf{q}_i} \mathrm{d}s \tag{16}$$

they are homogeneous to three generalized forces : the stretching force P_s, twisting force P_t and bending force P_b.

To solve the Lagrange equations, we need to express these generalized forces about position $\mathbf{r}$ and its derivatives $\mathbf{r}'$, $\mathbf{r}''$, $\mathbf{r}'''$, and the spline basis functions b_i and their derivatives b_i', b_i'', b_i'''. This allows us to evaluate the integral terms using classical Riemann sums [25].

The expressions involving generalized forces are quite complicated, but fast enough to compute. For the sake of clarity and brevity, we will provide them here, but not every detail of the calculation; please see the Appendix A for more details. We also consider separatly differentiations of strains with respect to position $\mathbf{r}$ and roll θ. Note that $\frac{\partial \mathbf{r}^{(j)}}{\partial \mathbf{r}_i} = b_i^j$ and $\frac{\partial \theta^{(j)}}{\partial \theta_i} = b_i^{(j)}$, where $b_i^{(j)}$ is the jth derivative of b_i with respect to u.

Furthermore, we introduce the following variables for compactness of the equations:

- $\mathcal{C} = \mathbf{r}' \times \mathbf{r}''$
- $\mathcal{P} = \frac{\partial \mathbf{r}' \times \mathbf{r}''}{\partial \mathbf{r}_i}$
- $\mathcal{T} = \mathcal{C} b_i''' - \mathcal{P} \times \mathbf{r}''' - 2\tau \left(\mathcal{C} \times \mathcal{P}\right)$.

where × denotes the cross product. The geometrical twisting τ will be detailed ahead.

We lay stress on the fact again that we consider our object to be materially linear elastic (small strains) but geometrically exact (large transformations).

- Stretching force

  In small strains, the stretching strain is defined by $\epsilon_s = 1 - \|\mathbf{r}'\|$. Nocent [17] used the large strain assumption, but the difference of accuracy between small strains and large strains is not significative for high rigidities.

  The stretching force term P_s^i yields:

$$P_s^i(\mathbf{r}) = -\frac{\pi E D^2}{4}\int_0^L \left(1 - \frac{\|\mathbf{r}_0'\|}{\|\mathbf{r}'\|}\right) \mathbf{r}' {b'}_i \mathrm{d}s. \tag{17}$$

  Since stretching strain energy U_s does not depend on θ,

$$P_s^i(\theta) = 0. \tag{18}$$

- Twisting force

  The twisting comprises two scalar parts: *geometrical or Frenet twisting* τ and *roll* θ. Geometrical twisting is due to the bending of the neutral fiber and is reponsible for bending-twisting coupling, whereas roll corresponds to the rotation of the material around the neutral fiber, as described in [26]. Chouaïeb [27] established that twisting is the sum of Frenet twisting and a rotation about the tangent.

  The twisting results in the following expression:

$$\begin{aligned} \epsilon_t &= \theta' + \tau \\ \tau &= \frac{\mathbf{r}' \times \mathbf{r}'' \cdot \mathbf{r}'''}{\|\mathbf{r}' \times \mathbf{r}''\|^2} = \frac{\mathcal{C} \cdot \mathbf{r}'''}{\|\mathcal{C}\|^2}. \end{aligned} \tag{19}$$

  This expression is also considered in [16].

  The geometrical twisting contribution $P_t^i(\mathbf{r})$ yields:

$$P_t^i(\mathbf{r}) = -\frac{\pi G D^4}{32}\int_0^L \left(\epsilon_t - \epsilon_t^0\right) \frac{\mathcal{T}}{\|\mathcal{C}\|^2} \mathrm{d}s. \tag{20}$$

  The roll contribution $P_t^i(\theta)$ yields:

$$P_t^i(\theta) = -\frac{\pi G D^4}{64}\int_0^L \left(\epsilon_t - \epsilon_t^0\right) \left(\frac{{b_i}'}{\|r'\|}\right) \mathrm{d}s. \tag{21}$$

- Bending force

  The bending force is a function of the scalar Frenet curvature k, which is equal to bending strain ϵ_b:

$$\epsilon_b = k = \frac{\|\mathbf{r}' \times \mathbf{r}''\|}{\|\mathbf{r}'\|^3} = \frac{\|\mathcal{C}\|}{\|\mathbf{r}'\|^3}. \tag{22}$$

  Lenoir approximated k by r'', considering small rotations. In order to have accurate results, we keep the original definition of the curvature in our calculation.

  The bending force term P_b^i yields:

$$P_b^i(\mathbf{r}) = -\frac{\pi E D^4}{64}\int_0^L \frac{\epsilon_b - \epsilon_b^0}{\|r'\|^2} \left(\frac{\mathcal{C} \times \mathcal{P}}{\|\mathcal{C}\| \|\mathbf{r}'\|} - 3k{b'}_i r'\right) \mathrm{d}s. \tag{23}$$

  Since the bending strain energy U_b does not depend on θ, $P_b^i(\theta) = 0$.

The strain force is the sum of vectors provided by the strain equations in stretching (17), twisting (20) and (21) and bending (23), and its calculation is expanded in Appendix A.

Specifying some arbitrary rest state configuration of the GEDS is quite simple. Yet, deformation energies calculated from an initial geometric configuration have to remain "reasonable" for preserving numerical stability.

5.4. Numerical solving of the Lagrange equations

The Lagrange equation (2) can now be transformed into its matrix form,

$$\mathbb{M}A = F + P. \tag{24}$$

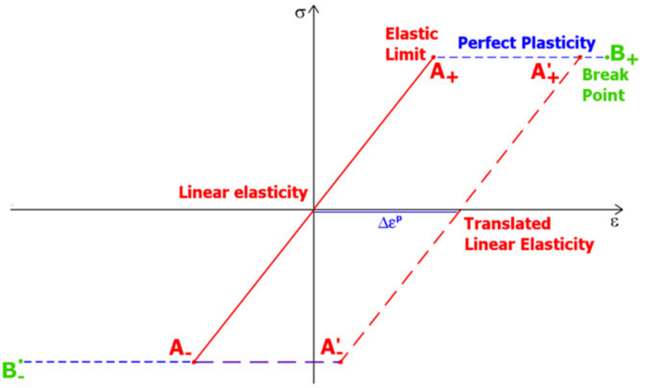


Fig. 5. Strain regions that rule one-dimensional objects: linear elasticity, perfect plasticity and break point. $\Delta\varepsilon^p$ corresponds to the elastic strain offset induced by a plastic strain.

The four sub-systems corresponding to x, y, z and θ can be solved independently. Since the spline possesses the local control property, the matrix $\mathbb{M}$ is banded with width $2l - 1$, where l is the spline locality.

We solve the system by using a simple LU decomposition at each simulation step. Accelerations are then integrated, at every time step, to determine control point velocities and positions. We use two integration methods: the explicit Runge Kutta method is fast but unstable for high rigidities; the implicit Euler Broyden method described in [28] guarantees numerical stability, but adds damping to the simulation.

We can now simulate an elastic physically-based spline. Adding plasticity to the model only requires a few other considerations, which we describe in the following section.

6. GEDS in the plastic domain

In this paper, we may also treat one-dimensional objects as perfectly plastic and breakable, with a stress–strain curve of the form shown in Fig. 2 with perfect plasticity. It is possible to simulate real plasticity using a function of ϵ and ϵ_A, which gives the part of the force $\mathcal{F}$ to convert into strain ϵ. In practice, we used ideal plasticity materials: as one can see Section 9.4, our practical results show convincing simulations using ideal plasticity. In addition to that, it is known that such an approximation provides almost exact results for some specific material, such as nylon wires or soft steel. In fact, this curve is quite similar to the curve for nylon wire given by Shuttleworth [29]; this justifies in this paper the use of perfect plasticity.

Stress and strain are not proportional beyond the elastic limits A_+ or A_-, as shown in Fig. 5.

In perfect plasticity, the force cannot exceed the positive yield force $\mathcal{F}_{A_+}$ or be less than the negative yield stress $\mathcal{F}_{A_-}$; more external works are entirely turned into a plastic strain offset $\Delta\epsilon^p$. The elastic region is translated by $\Delta\epsilon^p$. When the strain is greater than the breaking point strain ϵ_{B_+} or smaller than ϵ_{B_-}, the material irreversibly breaks.

Very little calculation is required to model perfect plasticity or to detect break points. Algorithm 1 can be used separately for each strain scalar component, since the strains are all independent. We do not directly use elastic limit stress, but we precompute corresponding strains ϵ_{A_+} and ϵ_{A_-}. We also assume that the plastic strain offset $\Delta\epsilon^p$ is zero everywhere at the start of the simulation, since the material is not damaged yet. At each simulation step, we check if the stress $\epsilon^e = \epsilon - \epsilon^0$ has exceeded the elastic region. If ϵ^e exceeds $\epsilon_{A_+} + \Delta\epsilon^p$, irreversible strain occurs and the elastic region is translated by their difference ϵ_+. So we update $\Delta\epsilon^p$ by adding ϵ_+.

Algorithm 1 Plasticity and break point algorithm

```
Initialization
for all spline samples do
    Compute $\epsilon^0$
    $\Delta\epsilon^p \leftarrow 0$
end for
while simulation do
    for all spline samples do
        $\epsilon^e \leftarrow \epsilon - \epsilon^0$
        $\epsilon^{e-p} \leftarrow \epsilon^e - \Delta\epsilon^p$
        $\epsilon_+ \leftarrow \epsilon^{e-p} - \epsilon_{A_+}$
        $\epsilon_- \leftarrow \epsilon^{e-p} - \epsilon_{A_-}$
        if $\epsilon_+ > 0$ then
            $\Delta\epsilon^p \leftarrow \Delta\epsilon^p + \epsilon_+$
            if $\epsilon^e > \epsilon_{B_+}$ then
                Simulate fracture
            end if
        else if $\epsilon_- < 0$ then
            $\Delta\epsilon^p \leftarrow \Delta\epsilon^p + \epsilon_-$
            if $\epsilon^e < \epsilon_{B_-}$ then
                Simulate fracture
            end if
        end if
    end for
end while
```

To deal with plasticity, the energy formulation (9) now yields:

$$U = \frac{1}{2}\int_0^L \left(\epsilon - \epsilon^0 - \Delta\epsilon^p\right)^t \mathbb{H} \left(\epsilon - \epsilon^0 - \Delta\epsilon^p\right) \mathrm{d}s \qquad (25)$$

ϵ_{A_+} implicitly becomes $\epsilon_{A'_+}$. When ϵ^e exceeds ϵ_{B_+}, the material breaks. The algorithm is the same for negative algebraic values.

7. World interaction

The spline-based model is continuous, that is, mechanically defined everywhere along the one-dimensional object. An applied force $\mathbf{F}$ on the point $\mathbf{q}$ of the spline provides generalized forces $\mathbf{F}_i$. Differentiating the power $W = \mathbf{Fq}$ with respect to the generalized coordinate $\mathbf{q}_i$ yields the corresponding generalized force $\mathbf{F}_i$:

$$\frac{\partial W}{\partial \mathbf{q}_i} = \mathbf{F}\frac{\partial \mathbf{q}}{\partial \mathbf{q}_i} = \mathbf{F}b_i. \qquad (26)$$

A force may consequently be applied everywhere, but interacting with the manipulated object remains quite difficult. This is the reason why we use Lagrangian multipliers: they allow us to set the position or the direction of any point of the one-dimensional object. Introducing them into the Lagrange equation (2), we get:

$$\forall i \in \{1, \ldots, n\}, \quad \begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\left(\dfrac{\partial K}{\partial \dot{\mathbf{q}}_i}\right) = \mathbf{F}_i - \dfrac{\partial T}{\partial \mathbf{q}_i} + \mathbf{L}^t \cdot \boldsymbol{\lambda} \\ \phi\left(\mathbf{q}_i, \dot{\mathbf{q}}_i\right) = 0 \end{cases} \tag{27}$$

where L is a matrix defined using the different constraints ϕ relative to all degrees of freedom [30], and λ are the Lagrange multipliers which correspond to the force required to maintain the constraints; ^{t} still denotes a transpose. The derived linear system thus yields:

$$\begin{pmatrix} \mathbb{M} & \mathbb{L}^t \\ \mathbb{L} & 0 \end{pmatrix} \begin{pmatrix} A \\ -\lambda \end{pmatrix} = \begin{pmatrix} F + P \\ E \end{pmatrix} \tag{28}$$

where $\mathbf{E}$ is a vector coding the desired behavior of the constraint, position or orientation.

Collision is dealt with using a classical penalty method.

8. Comments about twist control

8.1. From mechanical point of view

A cross-sectional orientation field is not required to solve mechanics, but only to visualize twisting and apply textures. As a matter of fact, bending and geometrical twisting only depend on control point positions, whereas the roll is not directly considered in the mechanical equations, but its derivative with respect to the spline parameter u. A major convenience of our model is that its accuracy does not rely on frames. This allows a real continuity of the one-dimensional object.

8.2. From visual point of view

However, we need to visualize the one-dimensional object to interact with it. A frame that minimizes geometrical twisting has the following major advantage: the aim is to add only a rotation about the tangent of an angle equal to the roll, in the plane of the cross-section. One of the more intuitive frames is due to Frenet. It consists of a unit length tangent $\mathbf{t}$, a principal normal $\mathbf{n}$ and a binormal $\mathbf{b}$. $\mathbf{t}$ is simply the unit length velocity vector $\mathbf{t} = \frac{\mathbf{r}'}{\|\mathbf{r}'\|}$. The Frenet frame is convenient because it can be analytically computed at any arbitrary point of the curve, but it is undefined wherever the curvature vanishes. Bloomenthal [31] proposed to define an initial frame at the beginning of the curve and to propagate it along the curve using small, local rotations. We will use the index k to enumerate frames from the beginning. The rotation matrix R between two frames may be given by the Olinde–Rodrigues formula. Boyer [32] gives a convenient expression which does not involve a rotation angle but only the two successive units tangents. However, it remains expensive. This is the reason why we use Kenneth Sloan's method as described by Bloomenthal to propagate the frames:

$$\begin{cases} \mathbf{b}_k = \mathbf{t}_k \times \mathbf{n}_{k-1} \\ \mathbf{n}_k = \mathbf{b}_k \times \mathbf{t}_k. \end{cases} \tag{29}$$

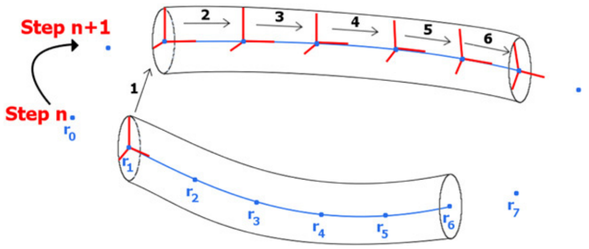


Fig. 6. Update of the local frames.

To update the frames between two steps n and $n + 1$ of the simulation, we use the Olinde–Rodrigues formula for its robustness, considering the unit tangents $\mathbf{t_{0^n}}$ and $\mathbf{t_0^{n+1}}$, as described in Fig. 6.

9. Implementation and validations

9.1. Implementation

The overall algorithm 2 recalls the required steps to simulate a physically-based spline, including elasticity and plasticity.

Algorithm 2 Physically-based spline algorithm

```
Initialization
for all spline samples do
  Compute Initial strains: $\epsilon^0$
  $\Delta\epsilon^p \leftarrow 0$
  Compute Inertia Matrix: $\mathbb{J}$ (Eq. 10)
  Compute Hooke Matrix: $\mathbb{H}$ (Eq. 6)
  Compute first initial frame, then propagate it along the spline (Eq. 29)
end for
while simulation do
  for all spline samples do
    Compute Right Term:
    Compute potential strain forces: $\mathbf{P}_s, \mathbf{P}_t, \mathbf{P}_b$ (Sec. 5.3)
    Add external forces and Lagrange multipliers: $\mathbf{F}, \mathbb{L}$ (Sec. 7)
    Plasticity and Break Point algorithm (Alg. 1)
    Compute Left Term:
    Compute $\mathbb{M}$ (Sec. 5.2)
    Solve matrix form of the Lagrange equations
    LU decomposition and solving: get accelerations $\mathbf{A}$ (see [25])
    Explicit or Implicit Integration: get new control points $\mathbf{q}_i$ (see [28])
    Update control points $\mathbf{q}_i$
    Update frames (Eq. 29)
    Display Splines (Eq. 3)
  end for
end while
```

A one-dimensional object is completely defined by a spline specified by an arbitrary number of control points as well as by some physical parameters. These parameters comprise the cross-section diameter D and material density μ, as well as two parameters of elasticity (that is, two of the three interrelated constants of Young's modulus, shear modulus and Poisson's ratio). The yield point A and break point B are optional. Strain forces are numerically evaluated, using a classical Riemann sum method with k samples per spline. Sampling is discussed in next Section 9.2.

Every point of the spline may be constrained in position and orientation using Lagrange multipliers. Reasonable constraint violation values may be used to move the GEDS. A value below 5 cm per step ensures stability. We also use a red icosahedron as an interactor with the scene. We may attach it to any object via a spring. This provides soft coupling with deformable objects and allows a more comprehensive interaction, but introduces oscillations between the interactor and the GEDS.

The model described in this paper has been implemented in C++. We performed several tests using a 3 GHz Pentium IV. The following subsections provide some accuracy measurements in static state, as well as performance and realism evaluations in motion state.

9.2. Classical static states

To validate our strain formulation, we have performed two classical experiments. The first one is the catenary whose shape is that formed by a perfectly flexible chain suspended by its ends and acted on by gravity. Its equation was obtained by Leibniz, Huygens and Johann Bernoulli in 1691:

$$y = \frac{h(\mathcal{F}_s)}{S\mu}\left(ch\left(\frac{S\mu x}{h(\mathcal{F}_s)}\right) - 1\right) \tag{30}$$

where $h(\mathcal{F}_s)$ is the horizontal component of the normal effort $\mathcal{F}_s$, μ the density, and S the area of the cross-section. The shape only relies on *stretching* and not on bending neither twisting. $h(\mathcal{F}_s)$ can only be numerically computed with the formulation of the length of the catenary l:

$$l = \frac{2h(\mathcal{F}_s)}{S\mu} sh\left(\frac{S\mu L}{2h(\mathcal{F}_s)}\right) \tag{31}$$

L is the distance between the wire ends.

Several configurations have been tested, corresponding to theoretical curves.

The second experiment is the classical problem of deflection of a cantilever beam of linear elastic material, under the action of an external vertical concentrated load at the free end. It was analysed by Beléndez [33]. The beam curvature and deflection only involve *bending* energy.

The total length l of the beam corresponds to the unknown slope ϕ_0 at the free end of the beam:

$$l = \sqrt{\frac{EI_\Delta}{2F}} \int_0^{\phi_0} \frac{\mathrm{d}\phi}{\sqrt{\sin\phi_0 - \sin\phi}}. \tag{32}$$

This equation allows us to obtain the slope ϕ_0 at the free end of the beam as a function of the external load F. The horizontal and vertical deflections at any point of the neutral axis of the cantilever beam are found as follows:

$$x = \sqrt{\frac{EI_\Delta}{2F}} (\sqrt{\sin\phi_0} - \sqrt{\sin\phi_0 - \sin\phi}) \tag{33}$$

$$y = \sqrt{\frac{EI_\Delta}{2F}} \int_0^{\phi} \frac{\sin\phi\, \mathrm{d}\phi}{\sqrt{\sin\phi_0 - \sin\phi}}. \tag{34}$$

The solutions to this problem are elliptic equations, which have no closed form solutions: they have to be numerically solved. Experiments illustrate in Fig. 12 the fact that theoretical and simulation curves we have computed are close to each other. Increasing the sampling density improves accuracy, but only 5 samples between two successive control points are necessary to provide accurate results. Concerning control point density, a minimum of 12 control points is required for the GEDS converge to the exact deflection. However, this corresponds to only 3 points per meter. Other methods like mass-springs, FEM or even Strands require many more degrees of freedom for the same result.

Experimental results demonstrate that the simulation curves we have computed are close to the theoretical curves, depending on the number of control points per length unit. Convergence of the deflection towards theory relies here on geometrical considerations, not on material ones. The assumption of small strains/large transformations is consequently validated a posteriori.

9.3. Dynamic simulation

There are a number of situations that cannot be handled by static simulation.

The motion pendulum is a simple animation that very much depends on physical parameters. Moreover, its behavior corresponds to its material properties, see Figs. 7 and Table 2.

The modelization of a spring using an helical version of our model validates the bending and twisting initial states as well as energies (see Fig. 8). It is an interesting example of what can be done with initial states. Springs are treated as purely elastic: they always return to their initial state after being relaxed.

The angular behavior is validated with the following experiment: twisting one end of an L-shaped cable make the opposite end have a circular trajectory (see Fig. 9).

9.4. Plastic strains and fracture detection

Plastic deformations can spoil electrical or pneumatic performance in cable laying, whereas in surgery, they enforce suturing quality and are thus welcomed by surgeons. We are able to handle these deformations and determine their magnitudes in real-time. When the fracture occurs, its location is indicated by the display of a sphere; here we do not simulate rupture as the object manipulation having failed, but it could be easily done with B-Splines [19] We illustrate perfect plasticity and fractures in Fig. 10.

9.5. A practical example of GEDS use in a constrained CAD application

We illustrate the potential benefit of our model in an application of virtual cable positioning on the inner structure of a car door. A result is shown in Fig. 1. The purpose is to test the compatibility between the planned fixing clip positions

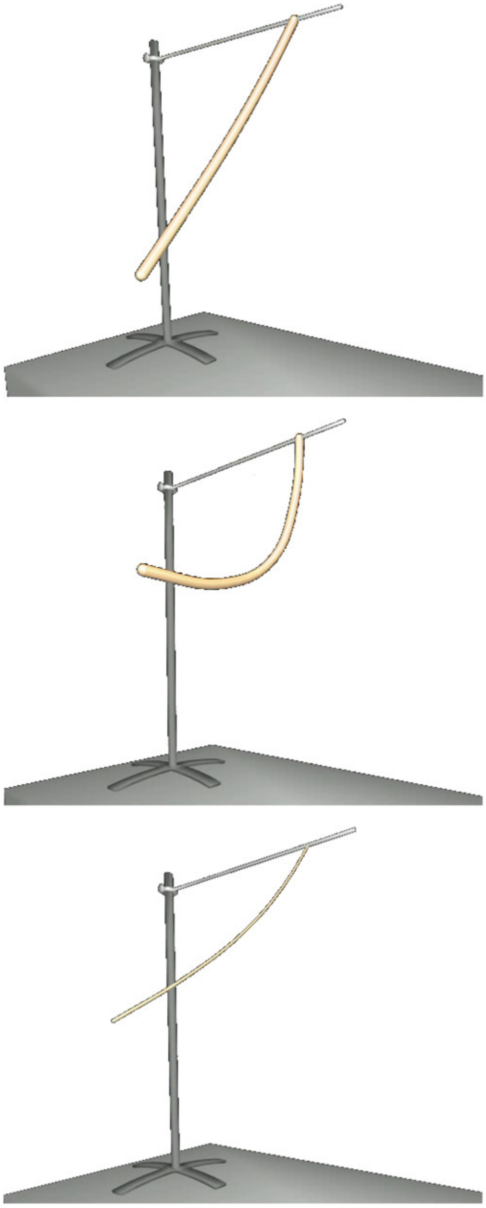


Fig. 7. Pendulums with several parameters; from top to bottom: ($E = 5e6$ Pa, $D = 0.04$ m), ($E = 1e6$ Pa, $D = 0.04$ m), ($E = 1e8$ Pa, $D = 0.01$ m). Varying the diameter and Young's Modulus induces different behaviors.

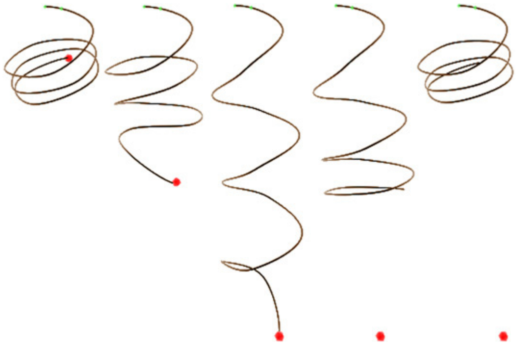


Fig. 8. Elongation of spring with 16 control points, $E = 1.5e7$ Pa, $D = 1$ cm.

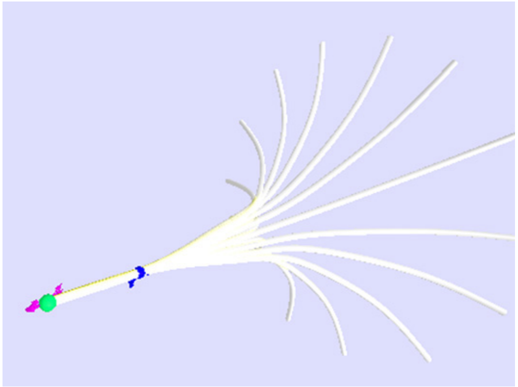


Fig. 9. Successive steps of an experiment, twisting one end of an L-shaped cable. The opposite end circular trajectory validates our twisting formulation.

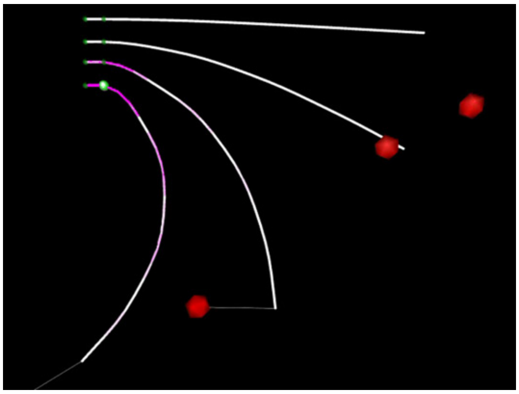


Fig. 10. Successive steps from elasticity to plasticity from the companion video. Pink indicates plastic strain intensity and the green sphere tags a break. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and the mechanical cable properties. Car engineers still need to build prototypes, since existing solutions are not accurate enough. Our model can prevent them from undergoing this fastidious and inevitable step. The simulation has to address the following properties: elastoplastic stretching, bending, twisting. In this application, fixing clips are mechanically modeled as a set of Lagrangian constraints. In our application, we consider simple fixed point constraints; if larger clips were needed, a combination of fixed points and fixed first derivative constraints would provide satisfactory results. During the interactive manipulation, the cable meets a fixing clip, and we create a point constraint. Solving these constraints gives the required forces or multipliers λ to maintain the global equilibrium of the cable. If a resulting force overwhelms the fixing clip strength in a determined direction, the clip fails to keep the cable: the corresponding constraint is deleted. This behavior only requires a simple test to be carried out, which consists in the following statement: if the scalar product of the force λ with the normal component N of the fixing clip frame is greater than the orientated fixing clip strength, the position constraint is deleted. This test is illustrated in Fig. 11.

Physical parameters and time necessary to compute one step of the simulation of 1 ms in these experiments is provided in Table 2. All of these correspond to interactive time and prove the model's efficiency.

Table 2
Average calculation time for various material parameters (simulation timestep 1 ms)

Experiment	n	ℓ (m)	D (cm)	μ (g/cm^3)	E (MPa)	ν	PDE solver	Comput. cost (ms)
Catenary	16	4.06	2	6	5	0.33	IE	0.60
Cantilever	7	1.60	4	2	35	0.33	IE	0.57
Cantilever	7	1.60	4	2	70	0.33	IE	0.64
Cantilever	7	1.60	4	2	700	0.33	IE	0.84
Cantilever	7	1.60	4	2	7000	0.33	IE	4.7
Pendulum	7	1.14	4	4	7	0.33	RK4	0.39
Pendulum	7	1.14	4	4	35	0.33	RK4	0.41
Pendulum	7	1.14	4	4	700	0.33	RK4	0.42
Pendulum	7	1.14	1	4	700	0.33	IE	0.94
Buckling	8	3.25	4	2.4	56	0.45	RK4	1.54
Buckling	8	2.00	4	2.4	56	0.45	RK4	1.06
Buckling	8	1.25	4	2.4	56	0.45	RK4	0.84
Buckling	8	1.125	4	2.4	56	0.45	RK4	0.77
Buckling	8	0.63	4	2.4	56	0.45	RK4	0.59
Car door	18	2.00	1.08	5	5000	0.33	IE	1.30

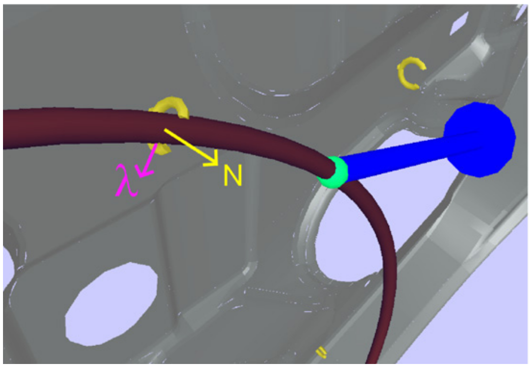


Fig. 11. A cable constrained by a fixing clip. Its normal component N and the force λ are illustrated by an arrow. The green sphere and its blue arrow correspond to the external force applied to the cable. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

10. Conclusion and future work

Using a background in mechanics consisting of elasticity and plasticity theories, we have proposed a deformable model for one-dimensional objects. Our approach addresses reversible and irreversible deformations, like stretching, twisting and bending, and can even detect fractures. This model provides both accurate mechanical simulation as well as quick calculations. Moreover, we can impose positions and orientations everywhere along the object. We can also simulate a wide range of materials in straight or distorted rest states; this is generally at interactive rates, except for very hard stiffness, for which the integration method requires excessive computation time for stability. The next step would be to provide a more efficient integration method for our model. Dynamic adaptive repartition of control points would also reduce computational effort, and therefore lower necessary computation time. Finally, adding the capability to handle contact and friction with the environment would improve realism and simulation possibilities.

Appendix A. Strain force calculation

A.1. *Stretching force*

The stretching force depends on the square of the stretching strain defined by $\epsilon_s = 1 - \|\mathbf{r}'\|$, that is, $\left(\epsilon_s - \epsilon_s^0\right)^2 = (\|\mathbf{r}'_0\| - \|\mathbf{r}'\|)^2$.

The stretching force term P_s^i yields:

$$P_s^i(\mathbf{r}) = -\frac{\partial U_s}{\partial \mathbf{r}_i} = -\frac{\pi E D^2}{8} \int_0^L \frac{\partial \left(\epsilon_s - \epsilon_s^0\right)^2}{\partial \mathbf{r}_i} \mathrm{d}s. \tag{A.1}$$

Differentiating $\left(\epsilon_s - \epsilon_s^0\right)^2$ with respect to $\mathbf{r}_i$ yields:

$$\frac{\partial \left(\epsilon_s - \epsilon_s^0\right)^2}{\partial \mathbf{r}_i} = -2\left(\|\mathbf{r}'_0\| - \|\mathbf{r}'\|\right) \frac{\partial \|\mathbf{r}'\|}{\partial \mathbf{r}_i}. \tag{A.2}$$

As the derivation of $\|\mathbf{r}'\|$ with respect to $\mathbf{r}_i$ is defined by

$$\frac{\partial \|r'\|}{\partial \mathbf{r}_i} = \frac{\mathbf{r}' \frac{\partial \mathbf{r}'}{\partial \mathbf{r}_i}}{\|\mathbf{r}'\|} \tag{A.3}$$

the differentiation results in:

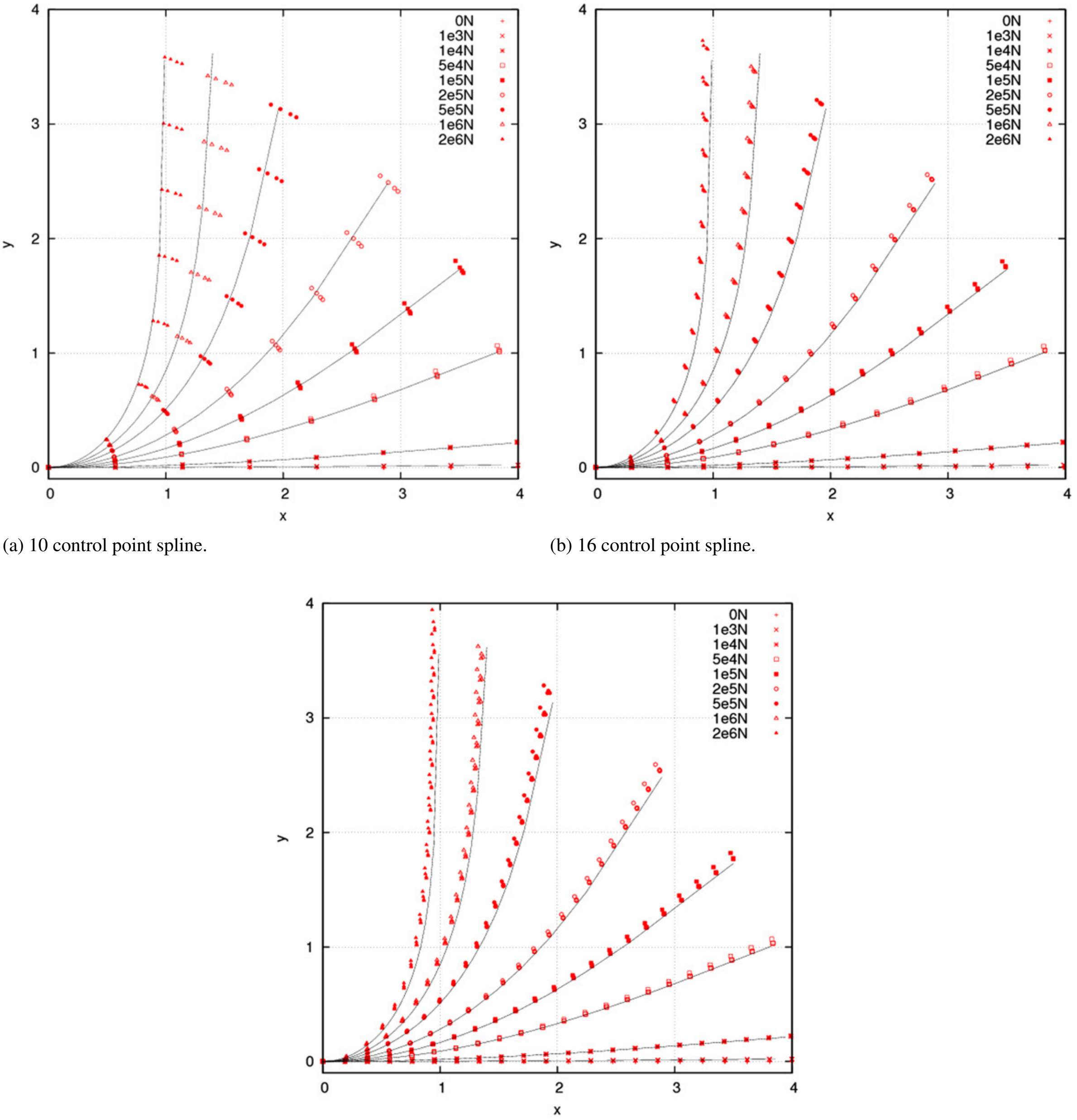
$$\frac{\partial \left(\epsilon_s - \epsilon_s^0\right)^2}{\partial \mathbf{r}_i} = 2\left(1 - \frac{\|\mathbf{r}'_0\|}{\|\mathbf{r}'\|}\right) \mathbf{r}' b'_i. \tag{A.4}$$

Thus, substituting Eq. (A.4) in Eq. (A.1) results in the following expression for the stretching force:

$$P_s^i(\mathbf{r}) = -\frac{\pi E D^2}{4} \int_0^L \left(1 - \frac{\|\mathbf{r}'_0\|}{\|\mathbf{r}'\|}\right) \mathbf{r}' b'_i \mathrm{d}s. \tag{A.5}$$

A.2. *Twisting force*

The twisting comprises two scalar parts: *geometrical or Frenet twisting* τ and *roll* θ. Geometrical twisting is due to the bending of the neutral fiber, and is responsible for bending-twisting coupling, whereas roll corresponds to the rotation of



(a) 10 control point spline.

(b) 16 control point spline.

(c) 24 control point spline.

Fig. 12. Bending validation of our model with the deflection of a four meter long cantilever beam, loaded with a vertical force at its end. Dots correspond to simulation results with different samplings (3, 5, 10 and 20 samples between 2 successive control points), whereas lines are theoretical deflections computed with elliptic equations.

material around the neutral fiber. The twisting results in the following expression:

$$\epsilon_t = \theta' + \tau$$
$$\tau = \frac{\mathbf{r}' \times \mathbf{r}'' \cdot \mathbf{r}'''}{\|\mathbf{r}' \times \mathbf{r}''\|^2} = \frac{\mathcal{C} \cdot \mathbf{r}'''}{\|\mathcal{C}\|^2}. \tag{A.6}$$

Frenet twisting is not defined for a straight line, but we assume it is zero in this case.

The geometrical twisting force term $P_t^i(\mathbf{r})$ yields:

$$P_t^i(\mathbf{r}) = -\frac{\partial U_t}{\partial \mathbf{r}_i} = -\frac{\pi G D^4}{64} \int_0^L \frac{\partial \left(\epsilon_t - \epsilon_t^0\right)^2}{\partial \mathbf{r}_i} \mathrm{d}s. \tag{A.7}$$

Differentiating $\left(\epsilon_t - \epsilon_t^0\right)^2$ with respect to $\mathbf{r}_i$ yields:

$$\frac{\partial \left(\epsilon_t - \epsilon_t^0\right)^2}{\partial \mathbf{r}_i} = 2(\epsilon_t - \epsilon_t^0) \frac{\partial \epsilon_t - \epsilon_t^0}{\partial \mathbf{r}_i} \tag{A.8}$$

and thus after some calculation:

$$\frac{\partial \left(\epsilon_t - \epsilon_t^0\right)^2}{\partial \mathbf{r}_i} = 2(\epsilon_t - \epsilon_t^0) \left(\frac{\mathcal{T}}{\|\mathcal{C}\|^2}\right). \tag{A.9}$$

Substituting Eq. (A.9) in Eq. (A.7), results in the following expression for the geometrical twisting force term:

$$P_t^i(\mathbf{r}) = -\frac{\pi G D^4}{32} \int_0^L \left(\epsilon_t - \epsilon_t^0\right) \frac{\mathcal{T}}{\|\mathcal{C}\|^2} \mathrm{d}s. \tag{A.10}$$

The roll contribution $P_t^i(\theta)$ yields:

$$P_t^i(\theta) = -\frac{\partial U_t}{\partial \theta_i} = -\frac{\pi G D^4}{64} \int_0^L \frac{\partial \left(\epsilon_t - \epsilon_t^0\right)^2}{\partial \theta_i} \mathrm{d}s. \tag{A.11}$$

Differentiating $\left(\epsilon_t - \epsilon_t^0\right)^2$ with respect to θ_i yields:

$$\frac{\partial \left(\epsilon_t - \epsilon_t^0\right)^2}{\partial \theta} = 2(\epsilon_t - \epsilon_t^0) \frac{{b_i}'}{\|r'\|}. \tag{A.12}$$

Substituting Eq. (A.12) in Eq. (A.11), results in the following expression for the roll force term:

$$P_t^i(\theta) = -\frac{\pi G D^4}{32} \int_0^L \left(\epsilon_t - \epsilon_t^0\right) \frac{{b_i}'}{\|r'\|} \mathrm{d}s. \tag{A.13}$$

A.3. Bending force

The bending force is a function of the scalar Frenet curvature k which is equal to bending strain ϵ_b:

$$\epsilon_b = k = \frac{\|\mathbf{r}' \times \mathbf{r}''\|}{\|\mathbf{r}'\|^3}. \tag{A.14}$$

The bending force term P_b^i yields:

$$P_b^i(\mathbf{r}) = -\frac{\partial U_b}{\partial \mathbf{r}_i} = -\frac{\pi E D^4}{128} \int_0^L \frac{\partial \left(\epsilon_b - \epsilon_b^0\right)^2}{\partial \mathbf{r}_i} \mathrm{d}s. \tag{A.15}$$

Differentiating $\left(\epsilon_b - \epsilon_b^0\right)^2$ with respect to $\mathbf{r}_i$ yields:

$$\frac{\partial \left(\epsilon_b - \epsilon_b^0\right)^2}{\partial \mathbf{r}_i} = 2\frac{\epsilon_b - \epsilon_b^0}{\|r'\|^2} \left(\frac{\mathcal{C} \times \mathcal{P}}{\|\mathcal{C}\|\|r'\|} - 3k{b'}_i r'\right). \tag{A.16}$$

Substituting Eq. (A.16) in Eq. (A.15), results in the following expression for the bending force term:

$$P_b^i(\mathbf{r}) = -\frac{\pi E D^4}{64} \int_0^L \frac{\epsilon_b - \epsilon_b^0}{\|r'\|^2} \left(\frac{\mathcal{C} \times \mathcal{P}}{\|\mathcal{C}\|\|r'\|} - 3k{b'}_i r'\right) \mathrm{d}s. \tag{A.17}$$

This expression is nevertheless undefined for $\|\mathcal{C}\| = 0$ (i.e. $\mathbf{r}'' = 0$ and $k = 0$), corresponding to a straight line, but considering a neighborhood of 0 yields a value of the bending force term for the rectilinear state. The magnitude of the tangent vector of the neutral axis must be nonzero; however, $\mathcal{C}$ is zero when the curvature is zero. We may assume that $\frac{\mathcal{C}}{\|\mathcal{C}\|}\underset{0}{\sim}\mathbf{1}$. Consequently, the bending force term for the rectilinear state yields:

$$P_b^i(\mathbf{r})\underset{0}{\sim} - \frac{\pi E D^4}{64} \epsilon_b^0 \int_0^L \frac{\mathcal{P} \times \mathbf{1}}{\|\mathbf{r}'\|^3} \mathrm{d}s. \tag{A.18}$$

Using this approximation for $\|\mathcal{C}\| < 10^{-8}$ does not influence either the stability or the precision of the GEDS near-rectilinear states. In our test we use this simple threshold value to determine whether the initial expression, or approximation, should be used for small curvature configurations.

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.cad.2007.05.008.

References

[1] Flexilution. URL: http://flexilution.com.

[2] Lenoir J, Cotin S, Duriez C, Neumann P. Interactive physically-based simulation of catheter and guidewire. Computer and Graphics 2006;30(3): 417–23.

[3] Qin H, Terzopoulos D. D-NURBS: A physics-based framework for geometric design. Transactions on visualization and computer graphics, vol. 2-1. IEEE; 1996. p. 85–96.

[4] Michelucci D, Foufou S, Lamarque L, Schreck P. Geometric constraints solving: Some tracks. In: SPM '06: Proceedings of the 2006 ACM symposium on solid and physical modeling. New York (NY, USA): ACM Press; 2006. p. 185–96.

[5] Benko P, Kos G, Varady T, Andor L, Martin R. Constrained fitting in reverse engineering. Computer Aided Geometrical Design 2002;19(3): 173–205.

[6] Welch W, Witkin A. Variational surface modeling. Computer Graphics 1992;26(2):157–66.

[7] Terzopoulos D, Platt J, Barr A, Fleischer K. Elastically deformable models. Computer Graphics 1987;21(4):205–14.

[8] Jeong I-K, Lee I. An oriented particle and generalized spring model for fast prototyping deformable objects. Eurographics 2004.

[9] Redon S, Galoppo N, Lin MC. Adaptive dynamics of articulated bodies. ACM Transactions on Graphics 24(3).

[10] James DL, Pai DK. Artdefo: Accurate real time deformable objects.

[11] Mueller M, Dorsey J, McMillan L, Jagnow R, Cutler B. Stable real-time deformations. In: Symposium on computer animation, ACM SIGGRAPH/eurographics symposium on computer animation. San Antonio (Texas): ACM Press; 2002. p. 49–54.

[12] Cosserat E, Cosserat F. Théorie des objets déformables. Hermann; 1909.

[13] Pai D. STRANDS: Interactive simulation of thin solids using Cosserat models, Computer Graphics Forum 21–3.

[14] Wakamatsu H, Hirai S. Static modeling of linear object deformation based on differential geometry. The International Journal of Robotics Research 2004;23:293–311.

[15] Bertails F, Audoly B, Cani M-P, Querleux B, Leroy F, Lévêque J-L. Super-helices for prediciting the dynamics of natural hair. ACM Transactions on Graphics 2006.

[16] Grégoire M, Schömer E. Interactive simulation of one-dimensional flexible parts. In: Symposium on solid and physical modeling. ACM; 2006. p. 95–103.

[17] Nocent O, Rémion Y. Continuous deformation energy for dynamic material splines subject to finite displacements. In: Proceedings of the eurographic workshop on computer animation and simulation (ACM). New York (NY, USA): Springer-Verlag New York, Inc.; 2001. p. 88–97.

[18] Lenoir J, Meseure P, Grisoni L, Chaillou C. A suture model for surgical simulation. In: 2nd international symposium on medical simulation. 2004. p. 105–13.

[19] Lenoir J, Grisoni L, Meseure P, Chaillou C. Adaptive resolution of 1d mechanical B-spline. In: Graphite. Dunedin (New Zealand); 2005. p. 395–403.

[20] Terzopoulos D, Fleischer K. Modeling inelastic deformation: Viscolelasticity, plasticity, fracture. In: SIGGRAPH'88: Proceedings of the 15th annual conference on computer graphics and interactive techniques. New York (NY, USA): ACM Press; 1988. p. 269–78.

[21] Wicke M, Steinemann D, Gross M. Efficient animation of point-based thin shells. In: Proceedings of eurographics'05. 2005. p. 667–76.

[22] O'Brien JF, Bargteil AW, Hodgins JK. Graphical modeling and animation of ductile fracture. In: SIGGRAPH'02: Proceedings of the 29th annual conference on computer graphics and interactive techniques. New York (NY, USA): ACM Press; 2002. p. 291–4.

[23] Howell LL. Compliant mechanisms. Wiley-Interscience; 2001.

[24] Courbon J. Théorie des poutres. 1980.

[25] Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical recipes in C. Cambridge University Press; 1988.

[26] Boyer F, Primault D. Finite element of slender beams in finite transformations: A geometrically exact approach. International Journal of Numerical Methods in Engineering 2004;59:669–702.

[27] Chouaïeb N. Kirchhoff's problem of helical solutions of uniform rods and their stability properties. Ph.D. thesis. 2004.

[28] Hilde L, Meseure P, Chaillou C. A fast implicit integration method for solving dynamic equations of movement. In: ACM symposium on virtual reality software and technology. New York (NY, USA): ACM Press; 2001. p. 71–6.

[29] Shuttleworth GN, Vaughn LF, Hoh HB. Material properties of ophthalmic sutures after sterilization and disinfection. Journal of Cataract and Refractive Surgery 1999;25:1270–4.

[30] Lenoir J, Grisoni L, Meseure P, Rémion Y, Chaillou C. Smooth constraints for spline variational modeling. In: Graphite. Singapore: Nanyang Technological University; 2004. p. 58–64.

[31] Bloomenthal J. Calculation of reference frames along a space curve. 1990. p. 567–71.

[32] Boyer F, Primault D. Finite element of nonlinear cables : Applications to robotics. Far East Journal of Applied Mathematics.

[33] Beléndez T, Neipp C, Beléndez A. Large and small deflections of a cantilever beam. European Journal of Physics 2002;23:371–9.