International Journal of Solids and Structures 113–114 (2017) 192–208

Contents lists available at ScienceDirect

International Journal of Solids and Structures

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

Real time simulation of inextensible surgical thread using a Kirchhoff rod model with force output for haptic feedback applications

Zhujiang Wang[a], Marco Fratarcangeli[b], Annie Ruimi[c], A.R. Srinivasa[a,*]

[a] Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843-3123, United States
[b] Chalmers University of Technology, Gothenburg, Sweden
[c] Department of Mechanical Engineering, Texas A&M University, Qatar

ARTICLE INFO

Article history:
Received 22 February 2016
Revised 24 October 2016
Available online 15 February 2017

Keywords:
Kirchhoff rod
Elastic thread
Variational integrator
Inextensibility
Constraint force
Haptic feedback

ABSTRACT

We discuss the dynamics of an inextensible thin Kirchhoff rod used in the modeling of surgical threads, and demonstrate a very efficient scheme to not only simulate the motion of the thread in real-time (up to 1 ms per frame) but also obtain the constraint axial forces which can be fed back to a haptic system. The numerical scheme is based on a family of schemes called geometric or discrete variational integrators guaranteeing that the momentum and energy are exactly conserved over long periods of time for conservative systems. Besides, we report on an efficient numerical procedure to handle the inextensibility of the thread through physically based Lagrange multipliers, as well as the internal dissipation of the thread. We have performed simulations to verify the capabilities of our model to conserve momentum and energy, accurately calculate the axial constraint forces along the thread for haptic feedback, and capture bending-torsion coupling leading to the formation of plectonemes. While many of the ideas are well known in the computer graphics community (especially in hair modeling), we have implemented several improvements for the specific purpose of speeding up the computations for developing physically based haptic interfaces for knot tying and suturing.

1. Introduction

Surgical training that relies on practicing on animals, cadavers or foam kits is still widely used by the medical profession (Gallagher and Traynor, 2008). These traditional methods are known to have limitations because: (i) dead tissues do not exhibit the same response as healthy ones, (ii) humans and animals have different physiologies, (iii) plastic or foam models lack realism, (iv) novice trainees have a higher risk of being exposed to contaminated blood while manipulating sharp instruments, (v) experiments on animal raise ethical considerations and (vi) students of Muslim or Jewish faith may have religious conflicts in practicing on porcine tissues. In addition, there has been rising pressure from animal right activists and religious groups having conflicts touching porcine tissues (SL, 2002).

To address these problems, large research institutions and training hospitals (i.e. Massachusetts General Hospital, Stanford, Harvard) have undertaken projects to support simulation training (https://www.harvardmedsim.org/) for the last fifteen years. Within these efforts, the modeling of tissues and organs (such as brain or heart) has received more attention than the modeling of surgical devices. While there are already commercially available virtual surgery simulators, A large majority of them only provide visual feedback, i.e. there is no force feedback in the simulation devices. The ability to provide force feedback is essential to realistic training simulators which require not only the geometry to be right, but also the physics of the process. In particular, especially for surgical threads, the ability to simulate entangling and self intersection, as well as the inextensibility and strongly dissipative nature of the thread are critical to providing a realistic effect. Also, the axial force engendered by the inextensibility of the constraint has to be computed since this is the force felt by the user.

Furthermore, not all surgical threads are inextensible. For example the surgical thread *M468 MONODOX* (made by CP Medical Portland, OR, USA) has some limited extensibility as compared to *833S Silk Black Braided Sutures* (made by CP Medical Portland, OR, USA) and so has a different “hand feel”. Thus it is necessary to develop a method that will deal with both extensible and inextensible threads (so that the user can select the appropriate thread type).

* Corresponding author. Tel: +19798623999.
E-mail addresses: arun-r-srinivasa@tamu.edu, arun.r.srinivasa@gmail.com (A.R. Srinivasa).

http://dx.doi.org/10.1016/j.ijsolstr.2017.02.017
0020-7683/

2. Related works

2.1. Purely geometrical simulations

Challenges associated with the simulation in space and time of one dimensional objects in a three-dimensional space have been recognized by the computer animation industry (Nealen et al., 2006). However, the limitation of the computational power of early years meant that physical accuracy had to be sacrificed to generate visually plausible simulations quickly. Moreover, animating large amounts of hair and animal fur also led to the development of purely geometry-based simulations wherein the models focus only on visual accuracy.

For instance, Brown et al. (2004) develop an algorithm known as Follow the Leader (FTL) which guarantees thread's inextensibility and enables interactive simulations of knot tying. In their work, the thread is discretized into N nodes. If a node is moved, motion propagates throughout the rest of the nodes according to the FTL algorithm. Nonetheless, Bending and torsional effects are not taken into account in the FTL formulation. Furthermore, haptic feedback is not possible since forces are not computed. Müller et al. (2012) propose a Dynamic Follow-the-Leader (DTFL) algorithm (a revised version of the FTL algorithm), which is better suited to simulate the dynamic behavior of hair strands.

Umetani et al. (2014) propose another geometrically based model to simulate elastic rod following the Position-Based Dynamics (PBD) (Müller et al., 2007). In the PBD framework, the deformation of the rod is characterized by the positions of discrete points. The orientation of the rod is measured by ghost points distributed around the edge.

With geometrically based approaches, the simulation systems require prior knowledge of the objects being manipulated and the deformations must be explicitly specified, since no constitutive relation or balance law is used to predict the motion. This has prompted the computer CG community to begin exploring physically-based methods for animation. These methods relying on physical principles and computational power can model complex processes which would be difficult or impossible for geometrical approaches (Moore and Molloy, 2007).

2.2. Physical based simulations

We try here to provide a brief view of the physics based approaches used to simulate the dynamics of rod-like structures by classifying previous work into four main categories: mass-spring, Euler beam, Cosserat theory, and Kirchhoff theory.

Mass-spring: As noted in a recent survey (Moore and Molloy, 2007), the CG community has devoted a great deal of effort on mass-spring systems to simulate deformable bodies. One of the earliest mass-spring models for rod-like structures appears in the nineties (see Rosenblum et al., 1991) to simulate large quantities of hairs. In their work, the hair is discretized into mass points which are connected by springs. Additional hinge springs provide bending resistance. Wang et al. (2005) improved the mass-spring model by introducing torsional springs. Kubiak et al. (2007) evaluate the amount of torque using torsion angle – the angle difference between the material frame and the Bishop frame associated with the thread (Bishop, 1975); and the bending stiffness of each point is measured through the bending springs connecting its two adjacent neighbor points. Other approaches based on a mass-spring model (Nakajima et al., 1999; Bertails et al., 2003; Ward and Lin, 2003; Choe et al., 2005; Iben et al., 2013; Michels et al., 2015) have been developed to simulate hair in the past. Mass-spring systems have the advantage of being easy to implement, computationally efficient and well suited to parallel computing, which makes real-time animations possible. The main drawbacks are significant approximations of the physical body, the tendency to oscillate, poor system stability for large spring constants, and the delay in the propagation of force effects (Moore and Molloy, 2007).

Euler beam: Anjyo et al. (1992) propose a model based on Euler beam theory in hair simulation, in which each hair strand is considered as a cantilever Euler beam. Following the work by Anjyo et al., Kurihara et al. (1993); Daldegan et al. (1993), and Lee and Ko (2001) improve the Euler beam based hair model; wherein the internal damping of hair is considered. This formulation is efficient and easy to implement. However, torsional stiffness is hard to be accounted for (Ward et al., 2007).

Cosserat theory: The origins of the Cosserat rod theory date back to the work of Cosserat and Cosserat (1909). In fact, many solutions for loop formation have been described by Love (2013) as early as 1924 based on a graphical argument. A more contemporary account was provided by Antman (1995) and Rubin (2013) in a numerical context. Pai (2002) appears to have been the first to advocate and implement the Cosserat rod theory for surgical simulation. However, the implicit representation of the centerline complicates the handling of collisions. An improvement over Pai's model reports in the work of Spillmann and Teschner (2007a, 2008) who explicitly express the rod centerline and use constraints to couple the implicit (orientation of rod cross section) and explicit schemes (rod centerline). Punak and Kurenov (2011) optimize the performance through neglecting kinetic effects as well as internal friction and other dissipative effects. These simplifications result in an increase of computational speed while keeping the model relatively accurate. The models based on Cosserat rod theory achieve a good trade-off between accuracy and computational cost. However, most of these models starting with Cosserat rod theory are essentially methods based on Kirchhoff rod theory (Dill, 1992b).

Kirchhoff theory: Many endeavors have been made by the CG community to explore Kirchhoff rod theory (Dill, 1992b) due to its intrinsic properties of handling the bending and twisting of rods in recent years. Models for thin elastic rods based on Kirchhoff theory can be categorized into two subfamilies: implicit and explicit approaches, according to the different methods expressing rod.

Bertails et al. (2006) formulate an implicit approach known as super-helices model based on the Kirchhoff theory of elastic rods (Dill, 1992a; Langer and Singer, 1996). In the super-helices model, the rod is represented by its discrete curvatures and thus this approach naturally handles inextensibility. The model has been remarkably successful in simulating large numbers of strands of hair with frictional contact between them (Bertails-Descoubes et al., 2011a). The computational complexity of the super helices model is further reduced from quadratic to linear in the number of discrete elements (Bertails, 2009). The scheme automatically enforces inextensibility. Later Bertails-Descoubes (2012) develop another implicit approach named as super-clothoids, in which the rod is modeled as piecewise clothoids; additionally, Casati and Bertails-Descoubes (2013) improve the accuracy by building the dynamics of a G^2-continuous piecewise 3D clothoid model. Other implicit approaches based on the super-helices model have also been made depending on their applications, such as the works (Bonanni et al., 2009; Derouet-Jourdan et al., 2013).

Bergou et al. (2008) develop a model for a discrete elastic rod with an explicit representation of rod centerline based on the Kirchhoff rod theory (Dill, 1992b; 1992a; Langer and Singer, 1996). Torsion is measured by the difference of angles between the material frame and the Bishop frame of the rod (Bishop, 1975) so that no additional degree of freedom needs to be introduced to represent the torsional behavior. Furthermore, torsion is assumed to propagate instantaneously because the rod has negligible polar moment of inertia and this reduces the computation time needed to capture the temporal evolution of torsional waves. Bergou et al. (2010) later improve the efficiency in hair simulation by introduc-

ing time-parallel reference frame in the representation of adapted framed curves. Other recent noteworthy articles (see for example Spillmann and Harders, 2010; Tang et al., 2010; Sueda et al., 2011, Huang et al., Tang et al., 2012b; Kir, 2014; Gornowicz and Borac, 2015) report on modeling discrete elastic rods with the Kirchhoff theory.

All the implicit and explicit models based on Kirchhoff rod theory do a very effective job of simulating *physically realistic motion*. The implicit approaches have incorporated the inextensibility constraint into the parameterization, and therefore are suitable to simulate the inextensible rod-like structures. However, to the author's knowledge they have not *reported* on the computation of the axial forces in any of the papers cited, perhaps because that was not central to the problem they were considering. On the other hand for the haptic interface problem, this is a key need. Furthermore, the super helix model (as currently developed) incorporates the inextensibility directly. So it would have to be rewritten with additional degrees of freedom also to account for extensible threads. The explicit approaches are capable of modeling both inextensible (additional constraints are needed) and extensible rod-like structures, as well as calculating the axial forces, which can be used to provide a realistic "feel" for force feedback systems. Consequently, we propose a model based on the explicit approaches, considering the requirement of force feedback and the merits of Kirchhoff rod theory.

There also exist many other schemes to model the dynamics of thin rods. For example, the models based on an "articulated body" have been reported (Choe and Ko, 2005; Wan et al., 2012), where the properties of joints influence the models' performance. Additionally, Grégoire and Schömer (2006) propose a novel hybrid model that combines mass-spring system and Cosserat rod. In it, the rod is modeled as a mass-spring system, but bending and torsion follow the Cosserat theory.

3. Our approach

In this paper, our aim is to develop a fast robust scheme for threads that are inextensible using a model that can be extended to extensible threads as well and also get axial and other force information, so that we can provide true force feedback, which requires haptic update rates to be approximately 1000 Hz (El Saddik, 2007). We are focusing on the fact that the physics is as close to the accepted physics as possible while still maintaining an efficient simulation. Based on the model of discretized elastic rod (Bergou et al., 2008), we propose a scheme that meets the requirements of a real-time haptic simulator by making many subtle but needed improvements to existing schemes aiming towards long term stability, inextensibility (axial constraint force calculation), and internal dissipation. The uniqueness of our approach stems from the fact that we

(1) Overcome the long time energy or momentum boost or loss issues by adopting the discrete variational integrator technique (Marsden and West, 2001) to develop an explicit time integration scheme that is fast and stable over long simulation times. The conditional stability criterion for this explicit scheme is that the time steps are to be smaller than $\frac{1}{2}$ of the highest natural frequency of the system (Skeel et al., 1997). In our case, since the torsional modes have been "condensed out" of the system the time step determining frequency is the bending wave frequency which is approximately given by $\sqrt{mL^4/B}$, where m is the mass per unit length L is the link length, B is the bending modulus (Ginsberg, 2001). For a thread which has very low bending modulus, the time steps can be reasonably large. Once this condition is satisfied, the discrete variational integrator scheme will conserve energy and momentum when there is no further boundary motion

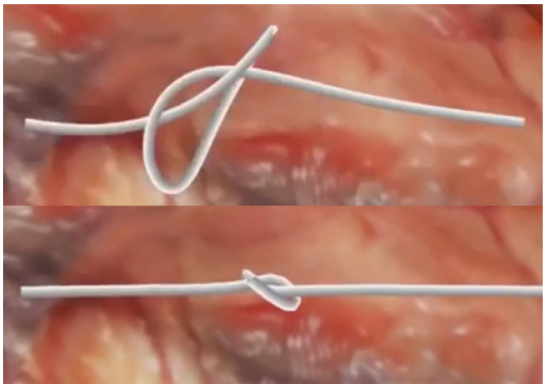


Fig. 1. Top: the initial loose square knot; bottom: the tight square knot[1] (Note the straightening thread due to apply axial forces).

(Marsden and West, 2001). Of course, with damping, the energy will decrease with time, but this will not be due to numerical damping artifacts. Thus, the variational integrator used here does not provide better numerical stability than other explicit schemes, but it is one of the ideal techniques that naturally prevent energy boost caused by the discretization error for a system. Thus it is suitable for the cases where long time simulation stability is required, particularly for the systems with low damping or energy dissipation.

(2) Develop a new technique based on a contraction mapping to enforce inextensibility exactly at every time step and compute the constraint forces, without constraint drift (compensated by a feedback term) reported in the work by Spillmann and Harders (2010) and sacrificing speed. We compare this approach to a Newton Raphson scheme and show that, because of the fewer calculations involved per time step, the approach presented here results in a significant speedup.

Anyone who has tightened a knot quickly recognizes that the constraint force due to the inextensibility of the thread is acutely felt. Simulations in which the inextensibility constraint is eliminated (as the implicit approaches by Bertails et al., 2006) or in which constraints are enforced by projection onto a constrained manifold at each time step (such as in the work by Bergou et al., 2008) have not reported a means for computing these forces (and in some cases require recomputing Newton's laws to find them), in part because they do not have a need for it unlike the current case. The Lagrange multiplier approach that we utilize here, is fast and has the advantage of providing the constraint forces (the Lagrange multiplier) as part of the solution, but requires additional computational steps during the motion computation itself. So our efforts to implement the Lagrange multiplier approach are central to proper force feedback. As a different example, CFD simulations of Newtonian fluids need to compute the pressure, which is the Lagrange multiplier that enforces incompressibility correctly. In practice, the time step we take is actually limited by the needs of identifying self-contact where the controlling dimension is the thread thickness and the requirements of true force feedback feel (high haptic update rates requires that time step should be around 1ms). Table 1 shows that our scheme can capture the axial force at 1000 Hz. The accuracy of the axial force is verified in Section 6.1.

(3) Develop a new way of incorporating internal string inelasticity as well as external air damping, which is critical for simulating realistic motions with rapid cessation of motion. Again experience with threads shows that they are susceptible not only to air

[1] The simulation is generated by SimInsights Inc. based on the thread model in this paper. Those who are interested in the animation can also check the videos *knot tying*: https://www.youtube.com/watch?v=ekBV0qhNyoo and *plectoneme formation*: https://www.youtube.com/watch?v=LeMP1Ikoipc.

Table 1
Comparison of recent Kirchhoff rod based models with force feedback and our model. We should note that the first row indicates, *Explicit*: if the model is explicit approach; *Inext*: if the rod is inextensible; F_{axial}: if the force feedback is the axial force; *Rates*: the haptic update rates; N_{Nodes}: number of nodes in the simulation.

	Explicit	Inext	F_{axial}	Rates	N_{Nodes}	Platform
Bonanni et al. (2009)	No	Yes	No	1000 Hz	-	-
Huang et al. (2011)	Yes	No	No	90 Hz	-	P7350 2.0 GHz
Kir (2014)	Yes	Yes	No	177 Hz	100	E3-1230V2 3.3 GHz
Our Model	Yes	Yes	Yes	1000 Hz	100	i7-3520M 2.9 GHz

drag but to internal dissipation (resistance to bending) due to the intra fiber friction. We have explicitly accounted for this, and we can show the realistic aspects of the effects only in a video (see the hyperlink in Fig. 1). Here we need to point out that all the current inextensible thread models based on Kirchhoff rod theory reported here, assume that the air drag on the string is linear in the velocity. As can be seen in any elementary textbook in fluid dynamics, the magnitude of the drag on such bodies is quadratic in the speed. The model that we develop accounts for this physically correct drag force on the body.[2]

4. Simulation scheme

A surgical thread can be modeled as a circular rod with a diameter much smaller than its length and is termed a one-dimensional object. When subjected to input forces and (or) moments, the rod takes on three-dimensional configurations. To describe the configurations of the rod in space and time requires the knowledge of the position of the rod centerline and the orientation of the cross section. The thread may bend, twist or assume a distinct coiling effect but it cannot stretch (inextensibility constraint).

4.1. Mathematical model

The mathematical model we use to describe the thread is a Kirchhoff elastic rod. The core geometry of the rod follows the work by Bergou et al. (2008), in which the configuration of a rod with the length L can be described by an adapted framed curve $\Gamma = \{\boldsymbol{\gamma}(s); \mathbf{T}(s), \mathbf{M_1}(s), \mathbf{M_2}(s)\}$ where $s \in [0, L]$. Here $\boldsymbol{\gamma}(s)$ represents the position of the rod's centerline, and the material frame $\{\mathbf{T}(s), \mathbf{M_1}(s), \mathbf{M_2}(s)\}$ describe the orientation of the cross section of the thread. The torsion of the rod is measured by the angle difference $\theta(s)$ between its material frame $\{\mathbf{T}(s), \mathbf{M_1}(s), \mathbf{M_2}(s)\}$ and the corresponding Bishop frame $\{\mathbf{T}(s), \mathbf{U}(s), \mathbf{V}(s)\}$ (see Fig. 2), the orientation of an untwist rod taking the same centerline $\boldsymbol{\gamma}(s)$.[3]

To proceed to a numerical solution, we discretize the rod centerline into $N_L \in \mathbb{N}$ segments each having a mass (m_i) concentrated on the segment vertices. Thus, the discretized curve Γ consists of a centerline comprised of $(N_L + 1)$ vertices $\mathbf{x} = (\mathbf{x}_1 \dots \mathbf{x}_{N_L+1})$ and N_L straight edges $\mathbf{e}_1 \dots \mathbf{e}_{N_L}$ such that $\mathbf{e}_i = (\mathbf{x}_{i+1} - \mathbf{x}_i)/|\mathbf{x}_{i+1} - \mathbf{x}_i|$. Also each straight edge $\mathbf{e}_i$ has the discrete material frame $\mathbf{M}^i = \left\{\mathbf{T}^i, \mathbf{M}_1^i, \mathbf{M}_2^i\right\}$ and the discrete Bishop frame $\mathbf{B}^i = \left\{\mathbf{T}^i, \mathbf{U}^i, \mathbf{V}^i\right\}$ to represent the twisted and untwisted orientation of the discrete rod respectively (Bergou et al., 2008) (see Fig. 3). The inextensibility constraints of the discrete surgical thread can then be expressed as,

$$\boldsymbol{\Psi}(\mathbf{x}) = [\psi_1, \psi_2, \cdots, \psi_i]^T = \mathbf{0} \tag{1}$$

where

$$\psi_i(\mathbf{x}) = |\mathbf{x}_{i+1} - \mathbf{x}_i|^2 - l_i^2, \qquad i = 1, 2, \cdots, N_L \tag{2}$$

and l_i is the rest length of the i_{th} discrete straight segment.

Since the cross-section of the thread has a tiny rotational inertia, we may neglect the torsional inertia in such a way that the material frame instantly evolves along the centerline of the thread at any time. This indicates that for the naturally straight and isotropic rod used in this paper, the elastic energy of the discrete rod consisting of bending and twisting energy can be expressed as,[4]

$$E(\Gamma) = \sum_{i=2}^{N_L} \frac{k_b}{\bar{l}_i}\left(\frac{2}{1 + \mathbf{e}_{i-1} \cdot \mathbf{e}_i} - 1\right) + k_t \frac{(\theta_{N_L} - \theta_1)^2}{2\bar{L}} \tag{3}$$

where $\theta_{N_L} - \theta_1$ is the total torsion angle of the rod; k_b and k_t measure the bending and twisting stiffness respectively; $\bar{l}_i = |\mathbf{x}_{i+2} - \mathbf{x}_{i+1}| + |\mathbf{x}_{i+1} - \mathbf{x}_i|$ and $2\bar{L} = \sum_{i=2}^{N_L} \bar{l}_i$.

4.2. Air drag

Because the density of the surgical thread is very small, one must take into account the drag caused by the motion of thread in air. Here (as in Munson et al., 2012), we assume that the air drag is an external force whose magnitude is proportional to the square of the thread velocity and whose direction is opposed to the velocity so that,

$$\mathbf{F}_a(\mathbf{v}_i) = -k_{air}(\mathbf{v_i}^T \mathbf{v_i})\hat{\mathbf{v}}_i \tag{4}$$

where k_{air} is the drag coefficient, and $\hat{\mathbf{v}}_i = \dfrac{\mathbf{v_i}}{|\mathbf{v_i}|}$. We note that this is a different assumption than that in the papers in the field. In particular, the quadratic dependence implies that for very slow motions, the drag is effectively zero (to the first approximation).

4.3. Internal dissipation

Unlike purely elastic Cosserat rods, surgical threads tend to have high internal dissipation (presumably due to the internal sliding of the fibers) and this has to be accounted for. Since the thread is inextensible, this means that the thread has resistance to internal bending. Spillmann and Teschner (2007b) calculate the relative angular velocity of each vertex to account for the dissipation. We introduce a simpler way to account for the internal dissipation that does not require additional computations of angular velocities. In this paper, since we assume the surgical thread is inextensible and the evolution of the twist of the thread is instantaneous,

[2] While it may be possible to incorporate the correct drag force in the theories of Bertails (2009), and Bergou(2008)) they have not reported this possibility. Furthermore, for models that are based on using an implicit time integration scheme (such as the one developed by Bergou (2008)), changing the drag force will change the scheme.

[3] The details of the kinematics of the rod using the Bishop frame can be found in the work by Bergou et al. (2008) and so we do not repeat it here.

[4] For a uniform circular rod which is naturally straight and untwisted, it can be shown that the angle of twist is uniformly distributed along the length (if we ignore rotational inertia) (Bergou et al., 2008). This considerably simplifies the calculations of the energy since the twist contribution now depends only upon the end twist. Furthermore, it is relatively simple to deal with extensible rods by simply including the term $\Sigma EA^* \psi_i$ to the total energy of the beam, where E is the modulus and A is the cross sectional area of the beam. It is the inextensibility constraint that needs special care and so for the remainder of the paper, we will focus attention only on this case.

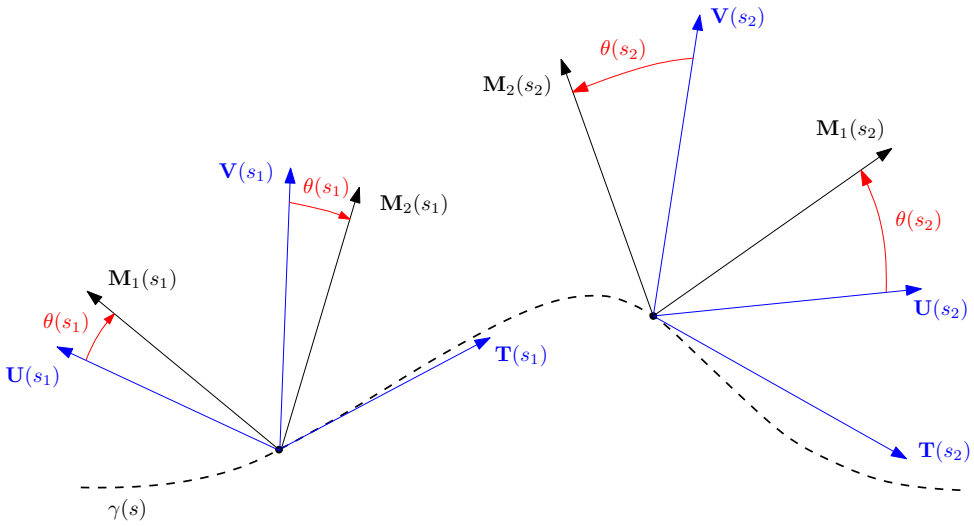


Fig. 2. The configuration of an elastic thread is represented by a curve $\boldsymbol{\gamma}(s)$ and a material frame $\{\mathbf{T}(s), \mathbf{M_1}(s), \mathbf{M_2}(s)\}$. The torsion can be measured by the angle difference $\theta(s)$ between the material frame and the Bishop frame $\{\mathbf{T}(s), \mathbf{U}(s), \mathbf{V}(s)\}$.

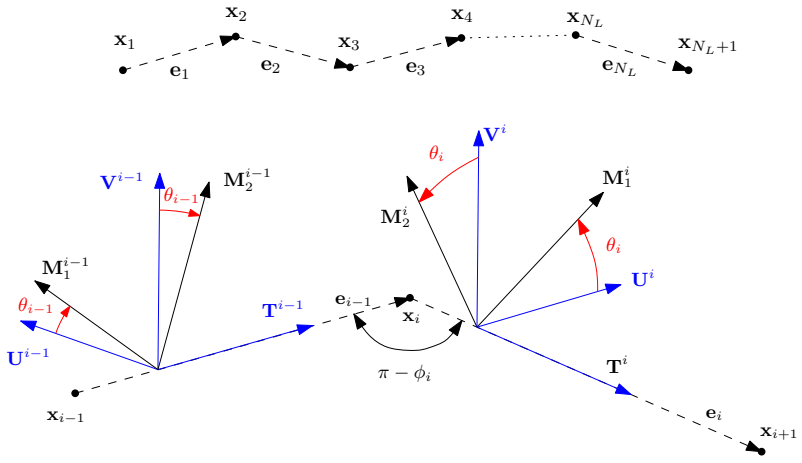


Fig. 3. Discrete framed curves.

we neglect the dissipation caused by stretch and torsion, and only consider the dissipation due to bending. We thus assume that the power dissipated due to bending is a function of the bending rate and is given by

$$P_d(\Gamma) = \sum_{i=2}^{N_L} \frac{1}{2} k_d \dot{\beta}_i^2 \tag{5}$$

where $\beta_i = \cos\phi_i$ measures the bending of the discrete thread (see Fig. 3). Then, we compute the internal force due to bending dissipation to be

$$\mathbf{F_b} = \frac{\partial P_d(\Gamma)}{\partial \dot{\mathbf{x}}} \tag{6}$$

Details pertaining to the derivation of the force $\mathbf{F}_b$ are shown in Appendix A.

5. Discrete variational integrator for Kirchhoff rods

A scheme to obtain a numerical solution for the deformations in space and time of a very thin and long circular rod (i.e. a thread) is presented here. The simulations are expected to run for a relatively long period so it is essential that the algorithm is able to prevent energy boost due to the numerical error introduced through the discretization process. The discrete variational integrators derived from Lagrangian mechanics have the property of being symplectic and thus guarantee the conservation of energy and momentum (Marsden and West, 2001; West, 2004). The method has been found to be very successful for path planning applications and has also been used to simulate the dynamics of rubbery materials. It is always employed to investigate the energy change caused by dissipation effects of a system (West, 2004). The results have demonstrated that the discrete variational integrator scheme is well suited for developing long time stable algorithms (Kharevych et al., 2006).

5.1. Discrete time Lagrangian mechanics

In order to apply this method to the rod theory being developed here, we note that the potential energy is given by Eq. (3) while the dissipative forces are given by Eqs. (4) and (6). Since we are ignoring the torsional inertia of the rod, the kinetic energy takes the simple form $T = \frac{1}{2}\sum(m_i \dot{x}_i \dot{x}_i)$. To this we add the constraint of inextensibility through the use of Lagrange multipliers. Now, following the well established discrete variational integrator approach (West, 2004), we discretize the action integral and extremize it (see Appendix B for details) to obtain a discrete set of

equations of the form

$$-\frac{\mathbf{M}}{h}(\mathbf{x}^{k+1}-2\mathbf{x}^{k}+\mathbf{x}^{k-1})+h\left[-\frac{dE(\mathbf{x}^{k})}{d\mathbf{x}^{k}}+\mathbf{F}_{d}\left(\mathbf{x}^{k-1},\mathbf{x}^{k}\right)+\frac{\partial\boldsymbol{\Psi}(\mathbf{x}^{k})}{\partial\mathbf{x}^{k}}\frac{\boldsymbol{\lambda}^{k}}{h}\right]=0 \tag{7a}$$

$$\boldsymbol{\Psi}(\mathbf{x}^{k+1})=0 \tag{7b}$$

where $\mathbf{M}$ is a $3(N_L+1)\times 3(N_L+1)$ mass matrix associated to vertices of discrete surgical thread; h is the time step; $\mathbf{F}_d(\mathbf{x},\dot{\mathbf{x}})=\mathbf{F}_a(\dot{\mathbf{x}})+\mathbf{F}_b(\mathbf{x})$ are the nonconservative forces shown in Eqs. (4) and (6); and the inextensibility constraint $\boldsymbol{\Psi}(\mathbf{x}^{k+1})$ is shown in Eq. (1).

Different from the manifold projection methods employed in the work by Bergou et al. (2008) to ensure the inextensibility, Eq. (7b) is ensured through a Lagrange multiplier that has to be calculated as part of the solution and not by a projection method. While this appears to be more cumbersome, the approach turns out be surprisingly fast and allows for the computation of the axial force (related to the Lagrange multiplier) without further computational steps. Also, it should be noted that the form (7) hides considerable complexity in the derivation of explicit expressions for $\frac{dE(\mathbf{x}^k)}{d\mathbf{x}^k}$ and $\mathbf{F}_d$ which must be carefully done (accounting for the neglect of the torsional inertia). However, the structure of the equations is such that these terms can be calculated systematically (see the Appendix B). For the positional variables $\mathbf{x}$ and Lagrange multipliers $\boldsymbol{\lambda}$, the superscripts denote the variables associated with time, while the subscripts denote the variables associated with the geometric position.

We should notice that the term $-\frac{dE(\mathbf{x}^k)}{d\mathbf{x}^k}+\mathbf{F}_d\left(\mathbf{x}^{k-1},\mathbf{x}^k\right)+\frac{\partial\boldsymbol{\Psi}(\mathbf{x}^k)}{\partial\mathbf{x}^k}\frac{\boldsymbol{\lambda}^k}{h}$ in Eq. (7) can be taken as external forces applied to the vertices of the discrete thread. $-\frac{dE(\mathbf{x}^k)}{d\mathbf{x}^k}$ is the conservative force (see the Appendix C); $\mathbf{F}_d\left(\mathbf{x}^{k-1},\mathbf{x}^k\right)$ is force caused by dissipation. Particularly we should emphasize the constraint force item

$$\mathbf{F}_{constraint}:=\frac{\partial\boldsymbol{\Psi}(\mathbf{x}^{k})}{\partial\mathbf{x}^{k}}\frac{\boldsymbol{\lambda}^{k}}{h} \tag{8}$$

which can be considered as the forces ensuring the inextensibility of the thread.

So far, we have set up the time evolution of the discrete surgical thread through the discrete Euler–Lagrange Eq. (7). Given $\mathbf{x}^{k-1}$ and $\mathbf{x}^k$ (time step k), we can find the unknowns $\mathbf{x}^{k+1}$ and $\boldsymbol{\lambda}^k$ (time step $k+1$) by solving Eq. (7). Next, we outline a simple iterative algorithm that gives the subsequent configurations of the thread in time.

5.2. Enforcing inextensibility and obtaining the corresponding Lagrange multipliers through a contraction mapping

Starting from the discrete Euler–Lagrange Eq. (7), we can rearrange it to obtain

$$\mathbf{x}^{k+1}=\mathbf{x}^{k}+h\mathbf{v}^{k}+h^{2}\mathbf{M}^{-1}\left[-\frac{dE}{d\mathbf{x}^{k}}+\mathbf{F}_{a}\left(\frac{\mathbf{x}^{k}-\mathbf{x}^{k-1}}{h}\right)\right]+h\mathbf{M}^{-1}\frac{\partial\boldsymbol{\Psi}(\mathbf{x}^{k})}{\partial\mathbf{x}^{k}}\boldsymbol{\lambda}^{k} \tag{9}$$

Substituting the above expressions $\mathbf{x}^{k+1}$ into the inextensibility constraints (1), the following matrix equation is obtained,

$$[\mathbf{A}(\mathbf{x}^{\mathbf{k}},\mathbf{x}^{\mathbf{k-1}})+\mathbf{B}(\mathbf{x}^{\boldsymbol{k}},\mathbf{x}^{\boldsymbol{k-1}},\boldsymbol{\lambda}^{\boldsymbol{k}})]\boldsymbol{\lambda}^{k}=\mathbf{R}(\mathbf{x}^{k},\mathbf{x}^{k-1}) \tag{10}$$

Appendix D contains the detailed expressions for matrices $\mathbf{A}(\mathbf{x}^{\mathbf{k}},\mathbf{x}^{\mathbf{k-1}})$, $\mathbf{B}(\mathbf{x}^{\boldsymbol{k}},\mathbf{x}^{\boldsymbol{k-1}},\boldsymbol{\lambda}^{\boldsymbol{k}})$, and the residue $\mathbf{R}(\mathbf{x}^{k},\mathbf{x}^{k-1})$. Eq. (10) can now be solved using a contraction mapping algorithm of the form $[\mathbf{A}+\mathbf{B}(\boldsymbol{\lambda}_n^k)]\boldsymbol{\lambda}_{n-1}^k=\mathbf{R}^k$. A key question that has to be answered is whether the contraction mapping algorithm converges and if it does, why should it be any better than a classical Newton–Raphson method which has quadratic convergence. We will answer this question next.

5.3. Enforcing inextensibility and obtaining the corresponding Lagrange multipliers through Newton–Raphson method

We now compare the algorithm that was introduced in the lines following Eq. (10) with a standard Newton–Raphson technique. The Eq. (10), derived from the discrete Euler–Lagrange Eq. (7), can be reformed as

$$G(\boldsymbol{\lambda}^{k})=[\mathbf{A}(\mathbf{x}^{\mathbf{k}},\mathbf{x}^{\mathbf{k-1}})+\mathbf{B}(\mathbf{x}^{\boldsymbol{k}},\mathbf{x}^{\boldsymbol{k-1}},\boldsymbol{\lambda}^{\boldsymbol{k}})]\boldsymbol{\lambda}^{k}-\mathbf{R}(\mathbf{x}^{k},\mathbf{x}^{k-1})=0$$

We neglect the upper script k and the position variable $\mathbf{x}$ for the sake of clarity, and then get

$$G(\boldsymbol{\lambda})=[\mathbf{A}+\mathbf{B}(\boldsymbol{\lambda})]\boldsymbol{\lambda}-\mathbf{R}=0 \tag{11}$$

Following the Newton–Raphson method, we can obtain $\boldsymbol{\lambda}$ through

$$\left.\frac{\partial G}{\partial\boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}_n}(\boldsymbol{\lambda}_{n+1}-\boldsymbol{\lambda}_n)=-G(\boldsymbol{\lambda}_n) \tag{12}$$

Substituting Eq. (11) into the above equation, we arrive at

$$\left[\mathbf{A}+2\mathbf{B}(\boldsymbol{\lambda}_n^k)\right](\boldsymbol{\lambda}_{n+1}^k-\boldsymbol{\lambda}_n^k)=-\left\{\left[\mathbf{A}+\mathbf{B}(\boldsymbol{\lambda}_n^k)\right]\boldsymbol{\lambda}_n^k-\mathbf{R}\right\} \tag{13}$$

where n is iteration number.

We might expect the Newton–Raphson scheme to outperform the naive iterative scheme (10). However, since the time steps are very small and the previous values of $\boldsymbol{\lambda}$ form a very good initial guess, both methods require only 3–5 iterations to converge. However the Newton–Raphson is slightly more expensive due to the repeated need to evaluate the right-hand side of Eq. (13) and over the course of the simulation that involves many tens of thousands of time steps, the cost adds up to a substantial value.

In summary, the numerical scheme we propose has three steps

i) Solve the Eq. (10) or (13) to obtain the value of $\boldsymbol{\lambda}^k$ using contraction mapping.
ii) Substitute $\boldsymbol{\lambda}^k$ into Eq. (9) to update the position information of vertices $\mathbf{x}^{k+1}$ at time step $k+1$.
iii) Repeat (i) and (ii) until the final time step is reached.

6. Results

In the previous section, we have claimed that our thread model is not only guaranteeing the conservation of energy and momentum, but also able to be used to calculate the axial force, which is given by the Eq. (8). So we will first try to verify these claims.

6.1. Thread model validation

Energy Conservation An initial straight thread is pinned at the top, and its centerline forms an angle $\varphi=0.45\pi$ with respect to the horizontal $x-axis$ (see Fig. 4). In order to eliminate energy change caused by air drag and internal dissipation of the thread, we set $k_{air}=0$ and $k_d=0$. In the simulation, the thread swings back and forth, and the configurations at $t=0.3$ s and $t=0.8$ s are plotted in Fig. 4. The discretization error and the neglect of the thread's kinetic energy in our model cause the energy difference between the simulation results and the analytical solution shown in Fig. 5. However, there is no drift in the thread's energy, which is thus well preserved during the long time simulation (1000 s).

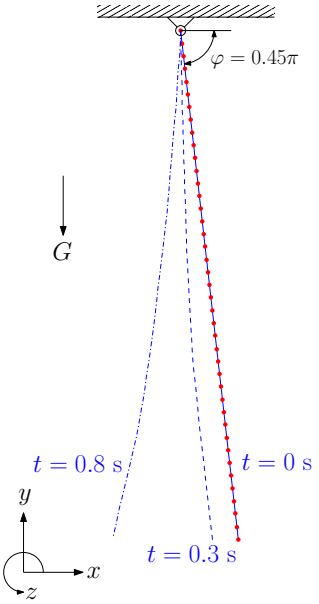


Fig. 4. Energy conservation test. An initial straight thread of length $L = 1$ m and mass $M = 0.01$ kg is pinned at the top. The centerline forms an angle $\varphi = 0.45\pi$ with respect to the horizontal $x - axis$. In the simulation, the dissipation are neglected ($k_{air} = 0$ and $k_d = 0$), and thus the thread keeps swinging back and forth without energy boost or lost as shown in Fig. 5.

Momentum Conservation An intuitive method to verify the momentum conservation of our model's intrinsic properties is observing the momentum of the discrete thread without external forces in a simulation. Thus we run a simulation (see Fig. 6), in which the gravity, air drag, and internal dissipation are all set to zero ($g = 0$, $k_{air} = 0$, and $k_d = 0$). The initial configuration of the thread is represented by the horizontal straight solid line, with the initial speeds of $\mathbf{v_L} = [0, 0.05, 0]$ m/s and $\mathbf{v_R} = [0, -0.05, 0]$ m/s at left and right ends of thread. Although the mass point of the thread obviously moves (see the configurations at $t = 0.8$, 20, 80, 300, and 1000 s in Fig. 6) and the magnitude of the thread's momentum does not show visual drift during the 1000 s simulation (see Fig. 7). So we claim that our model is able to intrinsically conserve momentum.

Axial force calculation To demonstrate that our model can accurately capture the axial force along the thread, we run a simulation shown in Fig. 8. The left end of the thread is pinned to a wall, and the right end is subject to an external force along the positive $x - axis$ with magnitude $F = 0.1$ N. The comparison of an analytical solution of the axial force for an equivalent static problem and the magnitude of the axial force along the thread (obtained at t = 100 s, when the kinetic energy greatly reduced due to the dissipation and thus the thread nearly reach a static state) in the simulation is shown in Fig. 9. The difference between the static analytical solution and dynamic simulation results of the axial force is mainly contributed by two parts, the discretization error and the kinetic energy of the thread. To the best of our knowledge, our model can be viewed as a first step towards accurately capturing the axial force of an inextensible thread at real-time rate shown in Fig. 9.

6.2. Simulation of plectoneme formation and tying knot

The slenderness of the surgical thread allows for very large bending and torsion leading to the possibility of self intersection. This is particularly true when tying knots, where the thread often collides with itself and inter-penetration and pass-through must be avoided. This is a difficult problem for one dimensional models (with zero thickness) since, within a single time step a portion of the rod can pass through a different portion and collisions or inter-penetration may not be detected at all. This collision of portions of the thread with itself raises difficulties in collision detection and response (Akutsu and Wadati, 1988; Kaufman et al., 2014).

Collision detection There are two aspects to the management of collisions: (1) fast collision detection and (2) application of forces to prevent inter-penetration and pass-through. Robust, fast collision detection algorithms have been developed in the CG community and here we provide only a brief introduction of the fast collision detection algorithm. The details are in the works by Ericson (2004) and Bresenham (1965). We first partition the three dimensional space into a uniform grid, and then calculate the cells of the grid occupied by the discrete thread through Bresenham's line algorithm (Bresenham, 1965). Since the collision is determined by the distance between portions of the thread, the grid size can be adjusted so that possible self-collision of the thread can only happen among the neighbor cells. Now we can check the distance between each two segments in the neighbor cells. If the distant $d(t)$ between two segments of the discrete surgical thread is smaller than a constant value Δd ($d(t) < \Delta d$), we define that the collision is detected (see Fig. 10).

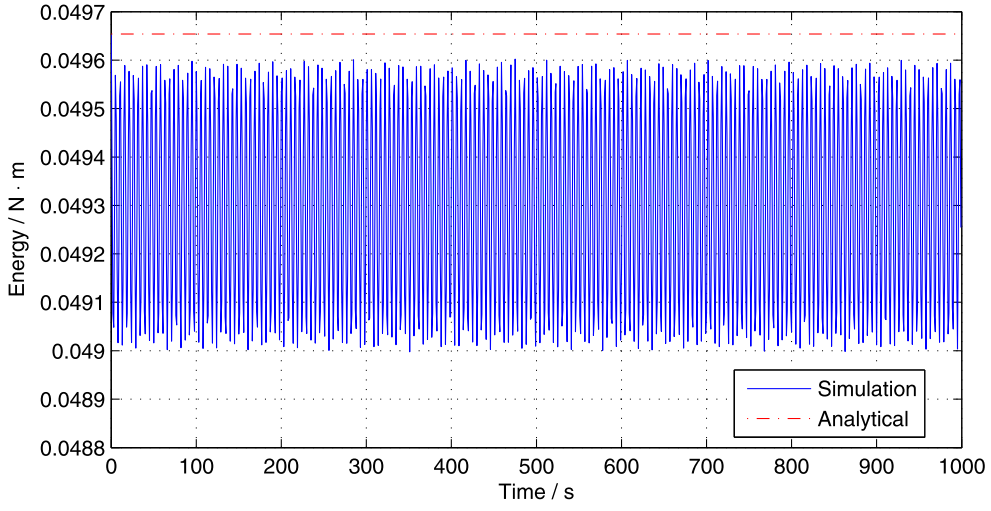


Fig. 5. The total energy of the thread during the 1000 s simulation (see Fig. 4). The energy is sampled at each second.

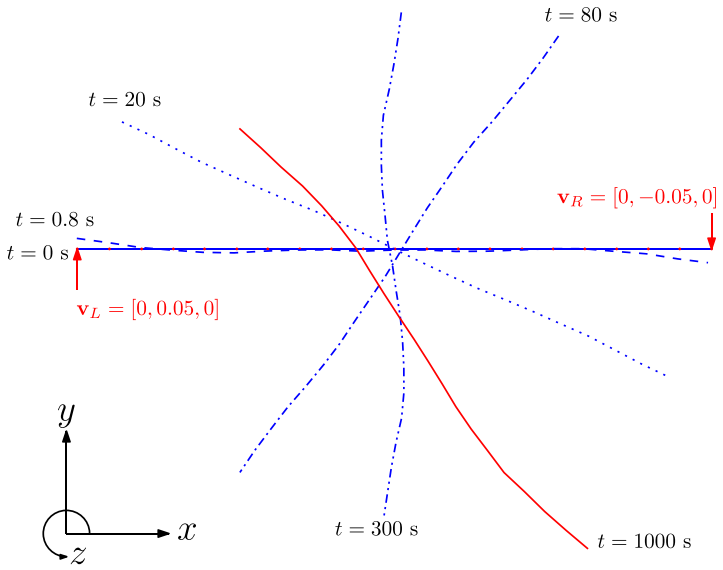


Fig. 6. Momentum conservation test. A thread of length $L = 1$ m and mass $M = 0.01$ kg is represented by the horizontal straight solid line, with the initial speeds of $\mathbf{v_L} = [0, 0.05, 0]$ m/s and $\mathbf{v_R} = [0, -0.05, 0]$ m/s at left and right ends respectively. In the simulation, the gravity, air drag, and internal dissipation are all neglected ($g = 0$, $k_{air} = 0$, and $k_d = 0$), the thread keep rotating and the configurations at $t = 0.8$, 20, 80, 300, and 1000 s are plotted. The magnitude of the thread's momentum is shown in Fig. 7.

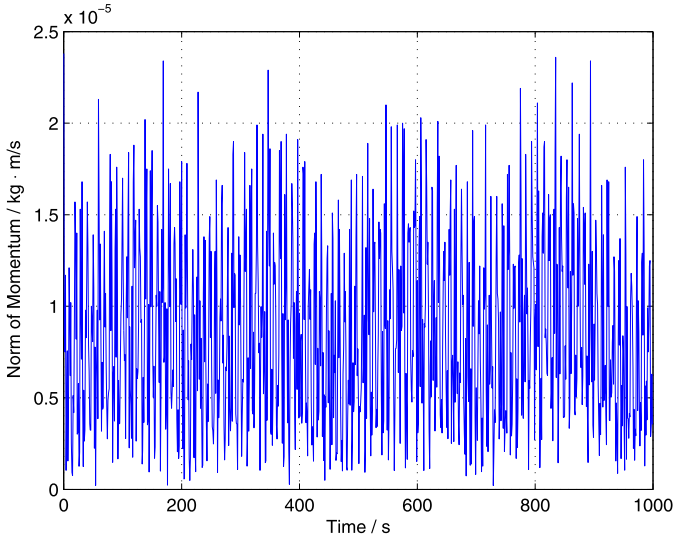


Fig. 7. The magnitude of the thread's momentum with respect to time during the 1000 s simulation (see Fig. 6). The momentum is sampled at each second.

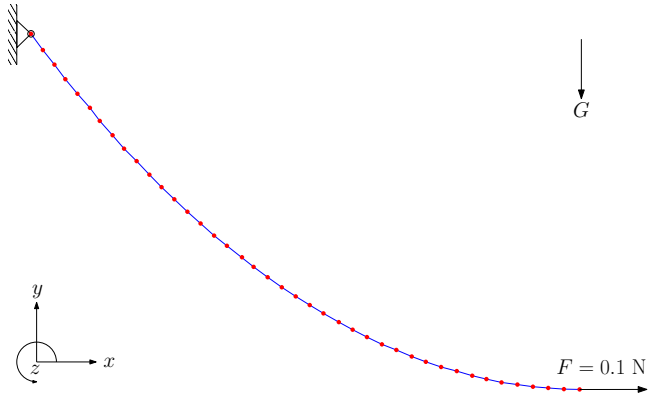


Fig. 8. The left end of a thread is pinned to a wall, and the right end is subject to an external force along the positive $x-axis$ with magnitude $F = 0.1$ N. The blue solid dot line represents the thread configuration at $t = 100$ s, when the kinetic energy greatly reduced due to the dissipation and thus the thread nearly reach a static state. The magnitude of axial force along the thread centerline is shown in Fig. 9. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Collision response The other challenge lies in a simulation of tying knots is collision response algorithms, which can be classified into three major categories (Tang et al., 2012a; Witkin, 1997): constraint-based formulations (Bridson et al., 2002; Duriez et al., 2006; Otaduy et al., 2009), penalty-based methods (Pen, 1990), and impulse-based methods (Mirtich, 1996). In general, constraint-based methods result in a more plausible simulation at the cost of extra computation (Hüsken, 2014; Bertails-Descoubes et al., 2011b). Impulse-based methods are more suitable for rigid body collision response due to the requirement of precise dynamic collision detections at each time step. Traditional penalty based methods, which are the easiest to use, unfortunately suffer from various kinds of problems, such as jitter effect. Tang et al. (2012a) improve the penalty method and develop a novel model-continuous penalty force method for application to rigid body collisions. We adapt this approach for flexible rods here to handle the collision response.

The continuous penalty force created by the collision during the time interval Δt is (see Fig. 10),

$$\mathbf{F}_c = \frac{\mathbf{I}}{\Delta t} = \frac{1}{\Delta t}\int_t^{t+\Delta t} \mathbf{F}_p(t)\, dt \tag{14}$$

where k_s is a stiffness constant, $\mathbf{F}_p(t) = k_s(\Delta d - |\mathbf{d}(t)|)\frac{\mathbf{d}(t)}{|\mathbf{d}(t)|}$ represents the penalty force created at time t, and $\mathbf{I} = \int_t^{t+\Delta t} \mathbf{F}_p(t)\, dt$ is the impulse produced by the penalty force $\mathbf{F}_p(t)$ during time interval $[t, t + \Delta t]$ (see the details in Appendix E). The key idea used here is that rather than directly use the penalty force (which causes jittery behavior and bounce back), we "smoothen it out" using a windowing scheme and utilize the time averaged collision force. Hence the continuous penalty force can essentially be taken as the average of the penalty force during a time interval.

To combine the collision response and thread model, we can insert the continuous penalty force (14) into the external force term in the constrained & forced discrete Euler–Lagrange Eq. (7) keeping

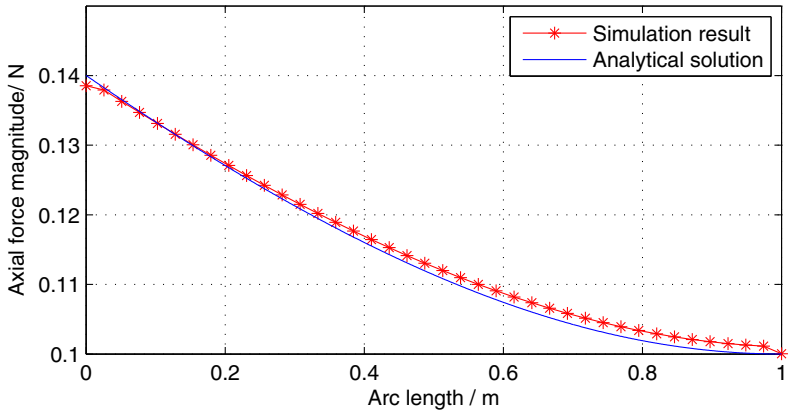


Fig. 9. The comparison between the magnitude of axial force along the thread centerline of a static analytical solution and that of our dynamic simulation (see Fig. 8). The solid line is the analytical solution and the solid line with asterisk markers represents the simulation results.

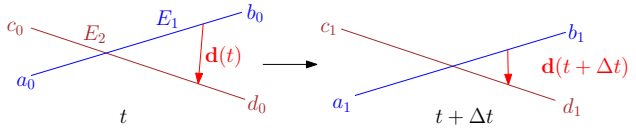


Fig. 10. Continuous penalty force for Edge-Edge (EE) contact. $\mathbf{d}(t)$ is the displacement between the edge a_0b_0 (E_1) and edge c_0d_0 (E_2) at time t; $\mathbf{d}(t+\Delta t)$ is the displacement between E_1 and E_2 at time $t+\Delta t$.

the same simulation scheme. This allows for the collision response to be easily handled.

6.2.1. Plectoneme formation

The simulation scheme we proposed was used to illustrate the formation of the plectoneme. Noteworthy parameters utilized in the simulations are a rod of length $L = 1$ m and mass $M = 0.02$ kg. The rod is discretized into $N_L = 20$ segments of equal length. The bending and torsional stiffness of the rod are $k_b = 0.02$ N· m^2 and $k_t = 0.02$ N· m^2 respectively. The time step is h = 0.001 s. For the collision detection and response, the distance criterion is $\Delta d = 0.3\frac{L}{N_L}$, and the stiffness constant is $k_s = 100$ N/m. The initial configuration of the rod is represented by a straight blue line in Fig. 11.

In the simulation, the right end of the thread moves toward the left at the speed of $\mathbf{V}_e = [-0.5, 0, 0]^T$. When the distance between the two ends reaches the value of $L_e = 0.075$ m, the right end of the rod is twisted with the angular velocity of $\omega_e = 0.8$ rad/s until the angle of twist reaches $\theta_{in}^{\max} = 20$ rad (maximum value). The simulation shows that the formation of two loops is accompanied by a decrease and then an increase of torque (points A and B in Fig. 12). Because when a loop starts formatting, the twisting energy begins to transfer to bending energy, resulting in the torque decrease; however, after the loop formats, the self-collision happens and prohibit this energy transfer, which causes the torque first suddenly stop decreasing and then increases again as the more input torsion angle is applied. Also, we notice that because part

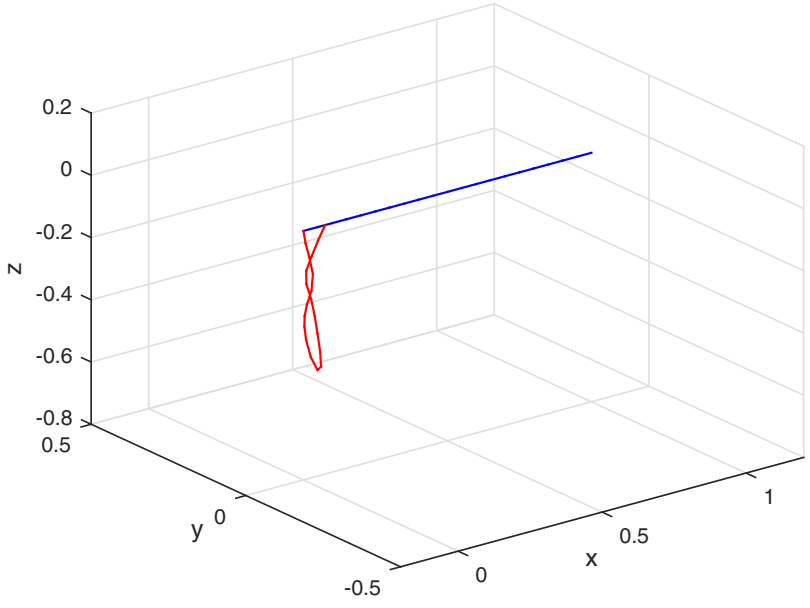


Fig. 11. The left end of the thread is pinned, and the right end is moving toward the left end at the speed of $\mathbf{V}_e = [-0.5, 0, 0]^T$ until the distance between the two ends is $L_e = 0.075$. Then the torsion angle is input with the angular velocity of $\omega_e = 0.8$ rad/s until reaching $\theta_{in}^{\max} = 25$ rad. The straight line represent the initial configuration of the thread at $t = 0.0$ s; the red curve (helix) is the configuration at $t = 40.0$ s.

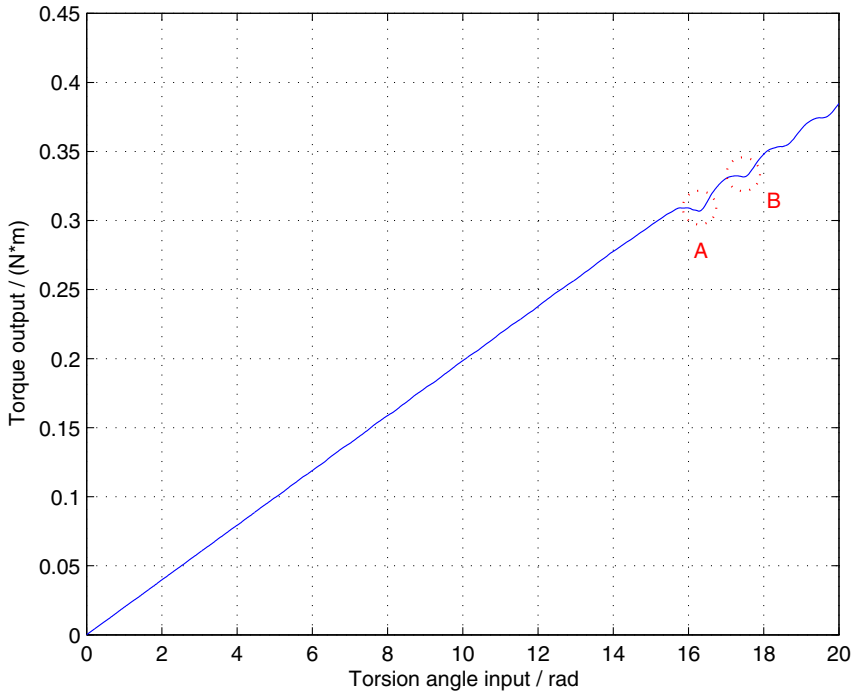


Fig. 12. The relation between the torque on the thread and the input torsion angle during the simulation. Note that every contact causes an oscillation (the contacts resulting in the plectoneme are shown as points A and B).

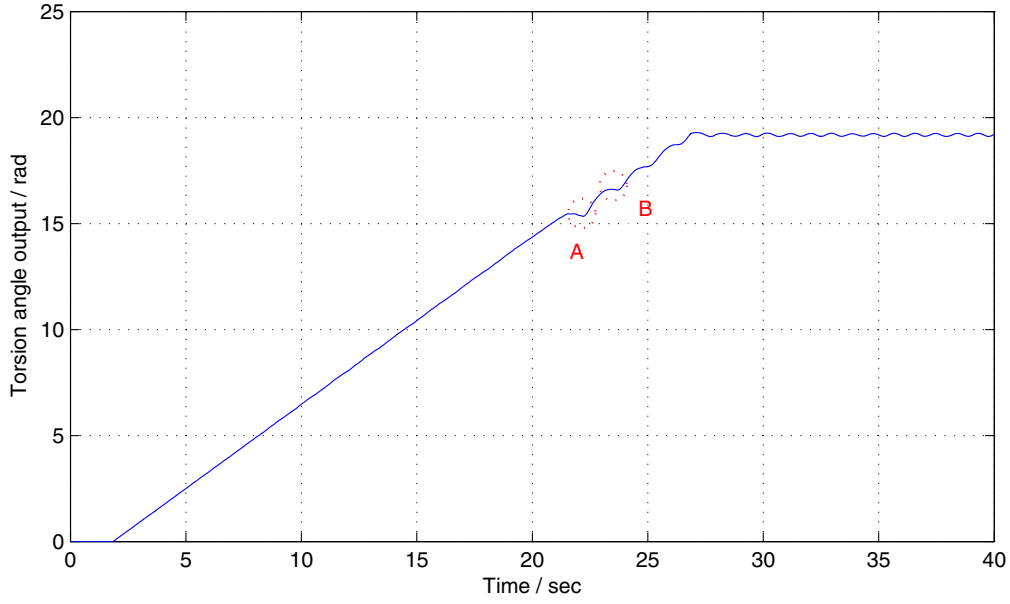


Fig. 13. The figure illustrates the change of torsion angle of the thread with respect to time during the simulation. Note that every contact causes an oscillation (the contact resulting in the plectoneme are shown as points A and B). The final torsion angle of the thread is smaller than the input torsion angle 20 rad.

of the twisting energy has transferred to bending energy, the final torsion angle of the thread decrease and is smaller than the input torsion angle 20 rad (see Fig. 13). Moreover, the simulation shows some oscillations of the torque, because twisting energy and bending energy are keeping transfer to each other after the formation of the loops due to the assumption of the elasticity of the thread model, and the dissipation are not large enough to stabilize the system immediately. The total energy of the system with respect to time is plotted in Fig. 14. After the application of torsion load, the total energy keeps to a certain level with little oscillations.

6.2.2. Tying knot

We also simulated the tying of a square knot. The parameters used in the simulations are a rod of length $L = 1$ m and mass $M = 0.02$ kg. The rod is discretized into $N_L = 40$ segments of equal length. The bending and torsional stiffness of the rod are $k_b = 0.002$ N· m^2 and $k_t = 0.002$ N· m^2 respectively. The time step is $h = 0.0001$ s. For the collision detection and response, the distance criterion is $\Delta d = \frac{L}{N_L}$, and the stiffness constant is $k_s = 50$ N/m. The result is shown in Fig. 15. In this simulation, the loose knot has no contact points at the beginning. The simulation is robust enough to ensure that no inter-penetration or pass through occur in this simulation. The constraint force that is to provide to haptic system can

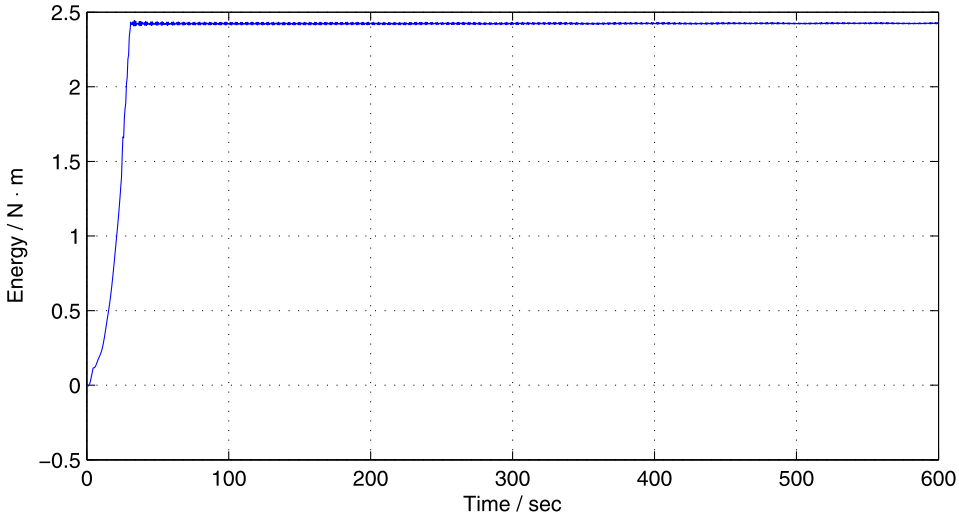


Fig. 14. The figure shows total energy (the sum of kinetic energy, potential energy, and elastic energy) with respect to time during the 600 s simulation. Note that total energy increase when the torsion load applied. After the application of torsion load, the total energy keeps to a certain level with little oscillations.

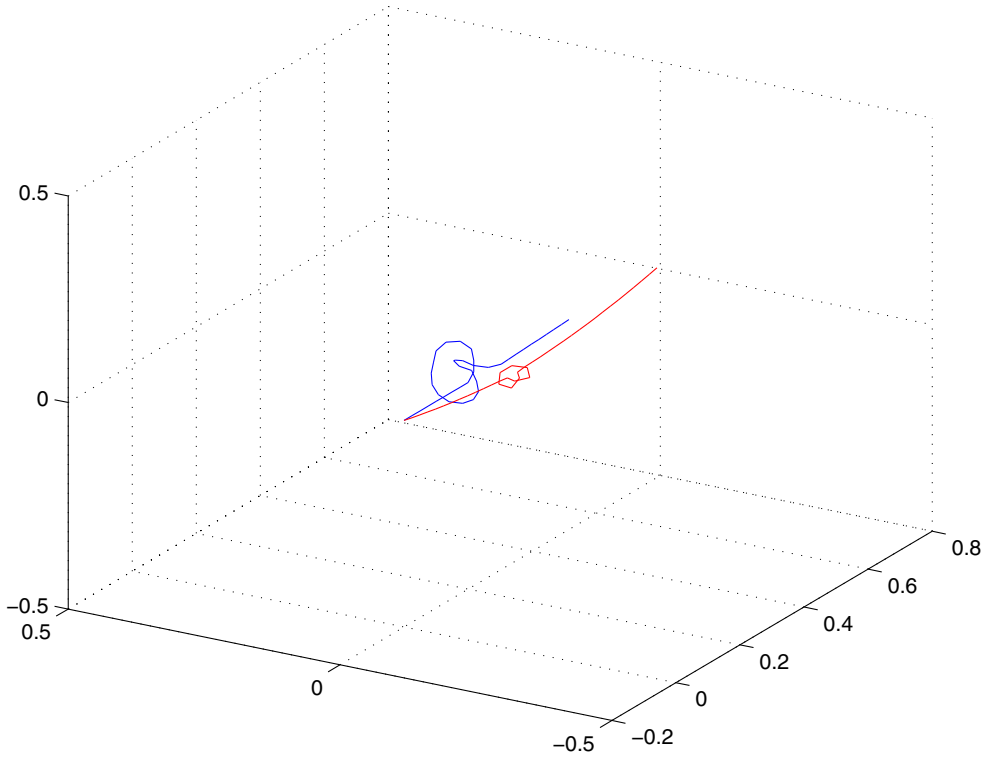


Fig. 15. The result of tying a square knot. The left end of the thread is pinned, and the right end is moving toward the right side. The curve with the loose knot is the initial condition of the thread at $t = 0.0$ s; the curve with the tight knot is the configuration at $t = 10.0$ s.

be obtained through the product of Lagrange multiplier and the direction vector of the last segment (see Eq. (8)). The magnitude of constraint force and the ratio of the distance between two ends of thread to its un-knot length with respect to time are shown in Fig. 16. As you tie the knot, the distance between two ends of the thread gradually increases until it reaches a static state. The simulation clear shows the magnitude of the constraint force first increases and then reaches a steady state when the knot is tightened tight. The fluctuations of the constraint forces are very tiny and can be smoothed by low pass filter when the haptic device is applied in future.

6.3. Comparison of numerical schemes based on contraction mapping and Newton–Raphson method

In almost all of the simulations, we obtain the value of Lagrange multipliers $\boldsymbol{\lambda}$ from Eqs. (10) and (13) within three iterations. As a consequence, numerical schemes based on contraction mapping performs at least 20% faster than that based on Newton–Raphson method, because solving Lagrange multipliers is the only difference existing in both numerical schemes. Therefore, we select the numerical scheme relying on contraction mapping (see Section 5.2) to run the real-time simulation in the following part.

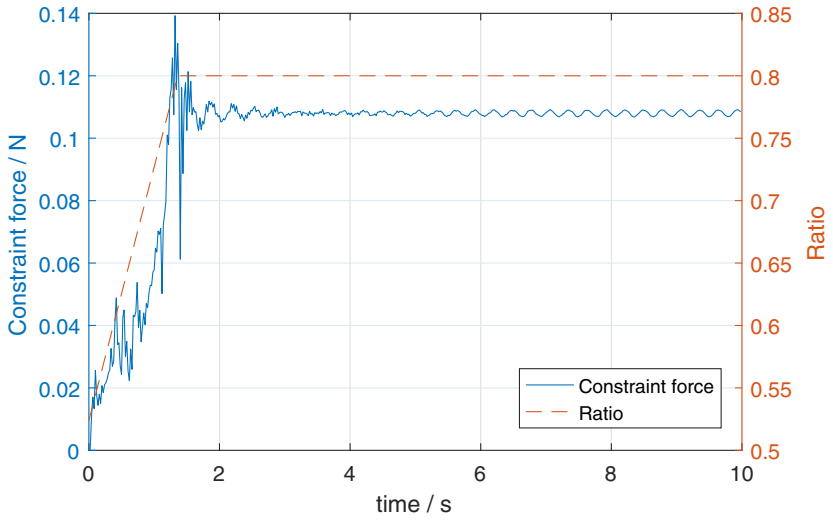
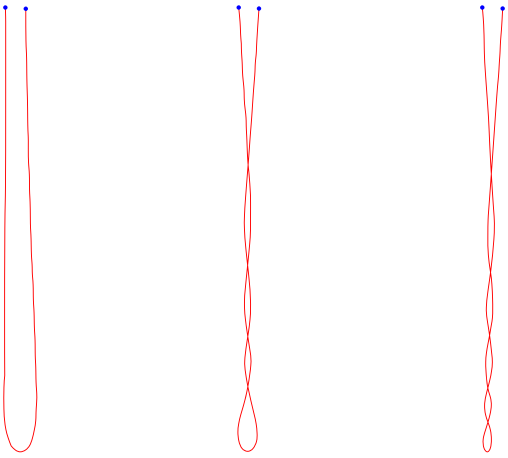


Fig. 16. The solid line shows the magnitude of constraint force to maintain the inextensibility of the thread with respect to time during the process of tying the tight knot. The dashed line illustrates the ratio of the distance between two ends of thread to its un-knot length.



(a) $k_b = 0.001\ N \cdot m^2$ $k_t = 0.001\ N \cdot m^2$ (b) $k_b = 0.0005\ N \cdot m^2$ $k_t = 0.001\ N \cdot m^2$ (c) $k_b = 0.00025\ N \cdot m^2$ $k_t = 0.001\ N \cdot m^2$

Fig. 17. The thread is initially straight with length $L = 1$ m and mass $M = 0.01$ kg, and discretized into $N = 100$ equal segments. The simulations are run with different stiffness as shown in the figures (a), (b), and (c). During all the simulations, the time step is $h = 0.001$/s, and the same load is applied (the left end of the thread moves toward the fixed right end until the distance between the two ends is $L_e = 0.02$ m. Then the torsion angle increases with the angular velocity of $\omega_e = 20$ rad/s until reaching $\theta_{in}^{\max} = 50\pi$). We note that for the same torsional modulus, the thread with the lower bending modulus has more loops.

6.4. Real-time simulation

The goal of our paper is to develop a physics-based real-time simulator, so we speed up the simulation through parallel programming techniques. Fig. 17 shows the three phenomena of plectoneme with different thread properties under the same load. The thread is initially straight with length $L = 1$ m and mass $M = 0.01$ kg, and is discretized into $N = 100$ equal segments. The simulations are run with different stiffness shown in the Fig. 17(a) $k_b = 0.001$ N· m^2, $k_t = 0.001$ N· m^2, (b) $k_b = 0.0005$ N· m^2, $k_t = 0.001$ N· m^2, and (c) $k_b = 0.00025$ N· m^2, $k_t = 0.001$ N· m^2. During all the simulations, the time step is $h = 0.001$ s, and the same load is applied (the left end of the thread moves toward the fixed right end until the distance between the two ends is $L_e = 0.02$ m. Then the torsion angle increases with the angular velocity of $\omega_e = 20$ rad/s until reaching $\theta_{in}^{\max} = 50\pi$). We notice that with the same torsion stiffness, the thread with softer bending stiffness can generate more loops under the same torsional load. The simulation runs on a computer with Intel(R) Core(TM) i7-3520M CPU @ 2.90 GHz, 8.00GB RAM. The average frame rate of the graphical display remains above 1.00 KHz.

7. Conclusion

In this paper, the surgical thread, a long and thin flexible structure, is modeled as an elastic Kirchhoff rod subject to end forces load. The combination of Lagrangian multiplier to maintain exact inextensibility, together with the use of the discrete variational integrator technique and continuous collision force technique shows that our model is able to handle very high bending angles and intimate contact without instability. The symplectic properties of the variational integrator guarantee that the momentum and energy will not boost during long time simulation, resulting in the long time stable simulation of the thread. The adoption of Lagrangian multiplier enable us not only accurately but also efficiently (on average over 1000 Hz for 100 elements) calculate the axial force along the thread.

The limitation of our work is that in order to capture the curvature of the knot, very fine discrete thread is required, due to the adoption of equal length discretization strategy. We can improve this by taking the adaptive discretization strategy (the fine segments are set where the curvature of the centerline is large, and coarse segments are where the curvature of the centerline of the thread is small) in the future.

Acknowledgments

This publication was made possible by the NPRP Award # 5-353-2-138 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors. We also acknowledge the support of Texas A&M University at Qatar for the visits of Dr. Arun Srinivasa and Mr. Zhujiang Wang to Doha, Qatar.

Appendix A. Derivation of the force caused by the bending dissipation

To find the forces caused by internal dissipation, we first derive the individual summands,

$$\begin{aligned}
\mathbf{S}_i^W &:= k_d\dot{\beta}_{i-1}\frac{\partial\dot{\beta}_{i-1}}{\partial\dot{\mathbf{x}}_i} = k_d\dot{\beta}_{i-1}\frac{\partial\beta_{i-1}}{\partial\mathbf{x}_i} = k_d\dot{\beta}_{i-1}\frac{\mathbf{x}_{i-1}-\mathbf{x}_{i-2}}{l_{i-2}\cdot l_{i-1}}\\
\mathbf{S}_i^O &:= k_d\dot{\beta}_i\frac{\partial\dot{\beta}_i}{\partial\dot{\mathbf{x}}_i} = k_d\dot{\beta}_i\frac{\partial\beta_i}{\partial\mathbf{x}_i} = k_d\dot{\beta}_i\frac{\mathbf{x}_{i+2}-2\mathbf{x}_i+\mathbf{x}_{i-1}}{l_{i-1}\cdot l_i}\\
\mathbf{S}_i^E &:= k_d\dot{\beta}_{i+1}\frac{\partial\dot{\beta}_{i+1}}{\partial\dot{\mathbf{x}}_i} = k_d\dot{\beta}_{i+1}\frac{\partial\beta_{i+1}}{\partial\mathbf{x}_i} = k_d\dot{\beta}_{i+1}\frac{\mathbf{x}_{i+2}-\mathbf{x}_{i+1}}{l_i\cdot l_{i+1}}
\end{aligned} \tag{A.1}$$

With the definition of the above items, the forces obtained through Eq. (6) can be expressed as,

$$\mathbf{F}_b = \left[\mathbf{F}_1^b, \mathbf{F}_2^b, \ldots, \mathbf{F}_N^b\right]^T \tag{A.2}$$

where

$$\begin{aligned}
&\mathbf{F}_1^b = \mathbf{S}_1^E; \qquad \mathbf{F}_2^b = \mathbf{S}_2^O + \mathbf{S}_2^E\\
&\mathbf{F}_i^b = \mathbf{S}_i^W + \mathbf{S}_i^O + \mathbf{S}_i^E, \qquad i = 3\ldots N-2\\
&\mathbf{F}_{N-1}^b = \mathbf{S}_{N-1}^O + \mathbf{S}_{N-1}^W; \qquad \mathbf{F}_N^b = \mathbf{S}_N^W
\end{aligned}$$

Appendix B. Derivation of constrained & forced discrete Euler–Lagrange equations

Following the theories on variational integrator(West, 2004), a discrete Lagrangian $\mathcal{L}_d(\mathbf{x}^k, \mathbf{x}^{k+1})$, approximating the action integral $\int_{t_k}^{t_{k+1}} \mathcal{L}(\mathbf{x}, \dot{\mathbf{x}})\, dt$ during time period of $[t_k, t_{k+1}]$, can be expressed as

$$\mathcal{L}_d(\mathbf{x}^k, \mathbf{x}^{k+1}) = h\left[\frac{1}{2}\left(\frac{\mathbf{x}^{k+1}-\mathbf{x}^k}{h}\right)^T\mathbf{M}\left(\frac{\mathbf{x}^{k+1}-\mathbf{x}^k}{h}\right) - E(\Gamma(\mathbf{x}^k))\right] \tag{B.1}$$

Taking consideration of the inextensibility constraints (1) and the dissipation forces (4) and (6), we can obtain the following equation according to the integral Lagrange–d'Alembert principle,

$$\delta\int_0^T \mathcal{L}(\mathbf{x}, \dot{\mathbf{x}}) + \boldsymbol{\lambda}\cdot\boldsymbol{\Psi}(\mathbf{x})\, dt + \int_0^T \mathbf{F}_d(\mathbf{x}, \dot{\mathbf{x}})\cdot\delta\mathbf{x}\, dt = 0 \tag{B.2}$$

The corresponding discrete form can be expressed as,

$$\begin{aligned}
&\delta\sum_{k=0}^{N_t-1}\left[\mathcal{L}_d(\mathbf{x}^k, \mathbf{x}^{k+1}) + \boldsymbol{\lambda}^{k+1}\cdot\boldsymbol{\Psi}(\mathbf{x}^{k+1})\right]\\
&\sum_{k=0}^{N_t-1}\left[\mathbf{F}_d^-(\mathbf{x}^k, \mathbf{x}^{k+1})\cdot\delta\mathbf{x}^k + \mathbf{F}_d^+(\mathbf{x}^k, \mathbf{x}^{k+1})\cdot\delta\mathbf{x}^{k+1}\right] = 0
\end{aligned} \tag{B.3}$$

where N_t is the number of time steps. Taking the variation of the Eq. (B.3), we can obtain

$$\begin{aligned}
&\sum_{k=1}^{N_t-1}\Big(D_1\mathcal{L}_d(\mathbf{x}^k, \mathbf{x}^{k+1}) + D_2\mathcal{L}_d(\mathbf{x}^{k-1}, \mathbf{x}^k) + \mathbf{F}_d^-(\mathbf{x}^k, \mathbf{x}^{k+1})\\
&\quad + \mathbf{F}_d^+(\mathbf{x}^{k-1}, \mathbf{x}^k) + \frac{\partial\boldsymbol{\Psi}}{\partial\mathbf{x}^k}\boldsymbol{\lambda}^k\Big)\cdot\delta\mathbf{x}^k\\
&+\sum_{k=0}^{N_t-1}\left[\boldsymbol{\Psi}(\mathbf{x}^{k+1})\cdot\delta\boldsymbol{\lambda}^{k+1}\right] + \left[D_1\mathcal{L}_d(\mathbf{x}^0, \mathbf{x}^1) + \mathbf{F}_d^-(\mathbf{x}^0, \mathbf{x}^1)\right]\cdot\delta\mathbf{x}^0\\
&+\left[D_2\mathcal{L}_d(\mathbf{x}^{N_t-1}, \mathbf{x}^{N_t}) + \mathbf{F}_d^+(\mathbf{x}^{N_t-1}, \mathbf{x}^{N_t})\right]\cdot\delta\mathbf{x}^{N_t} = 0
\end{aligned} \tag{B.4}$$

where

$$\begin{aligned}
D_1\mathcal{L}_d(\mathbf{x}^k, \mathbf{x}^{k+1}) &= -\frac{\mathbf{M}}{h}(\mathbf{x}^{k+1}-\mathbf{x}^k) - h\frac{dE(\mathbf{x}^k)}{d\mathbf{x}^k}\\
D_2\mathcal{L}_d(\mathbf{x}^{k-1}, \mathbf{x}^k) &= \frac{\mathbf{M}}{h}(\mathbf{x}^k-\mathbf{x}^{k-1})
\end{aligned}$$

Let $\mathbf{v}^k = \frac{\mathbf{x}^k-\mathbf{x}^{k-1}}{h}$, $\dot{\beta}_i^k = \frac{\beta^k-\beta^{k-1}}{h}$. According to the Eqs. (4) and (6), we can get the discrete form of the air drag and force caused by internal dissipation as $\mathbf{F}_a^k = \mathbf{F}_a(\mathbf{x}^{k-1}, \mathbf{x}^k)$ and $\mathbf{F}_b^k = \mathbf{F}_b(\mathbf{x}^{k-1}, \mathbf{x}^k)$ respectively. And therefore, we can set the discrete dissipation forces as

$$\begin{aligned}
&\mathbf{F}_d^-(\mathbf{x}^k, \mathbf{x}^{k+1}) = 0\\
&\mathbf{F}_d^+(\mathbf{x}^{k-1}, \mathbf{x}^k) \;=\; \mathbf{F}_d(\mathbf{x}^{k-1}, \mathbf{x}^k) = \mathbf{F}_a(\mathbf{x}^{k-1}, \mathbf{x}^k) + \mathbf{F}_b(\mathbf{x}^{k-1}, \mathbf{x}^k)
\end{aligned}$$

Substituting the above terms into Eq. (B.4), we can get

$$\begin{aligned}
&\sum_{k=1}^{N_t-1}\left(D_1\mathcal{L}_d(\mathbf{x}^k, \mathbf{x}^{k+1}) + D_2\mathcal{L}_d(\mathbf{x}^{k-1}, \mathbf{x}^k) + \mathbf{F}_d(\mathbf{x}^{k-1}, \mathbf{x}^k) + \frac{\partial\boldsymbol{\Psi}}{\partial\mathbf{x}^k}\boldsymbol{\lambda}^k\right)\cdot\delta\mathbf{x}^k\\
&+\sum_{k=0}^{N_t-1}\left[\boldsymbol{\Psi}(\mathbf{x}^{k+1})\cdot\delta\boldsymbol{\lambda}^{k+1}\right] + \left[D_1\mathcal{L}_d(\mathbf{x}^0, \mathbf{x}^1) + \mathbf{F}_d^-(\mathbf{x}^0, \mathbf{x}^1)\right]\cdot\delta\mathbf{x}^0\\
&+\left[D_2\mathcal{L}_d(\mathbf{x}^{N_t-1}, \mathbf{x}^{N_t}) + \mathbf{F}_d^+(\mathbf{x}^{N_t-1}, \mathbf{x}^{N_t})\right]\cdot\delta\mathbf{x}^{N_t} = 0
\end{aligned} \tag{B.5}$$

Since the above equation is true for any choice of $\delta\mathbf{x}^k$ and $\delta\lambda^k$ with $\delta\mathbf{x}^0 = \delta\mathbf{x}^{N_t} = 0$, then we obtain the *constrained & forced discrete Euler–Lagrange equations*,

$$\begin{aligned}
&-\frac{\mathbf{M}}{h}(\mathbf{x}^{k+1} - 2\mathbf{x}^k + \mathbf{x}^{k-1}) - h\frac{dE(\mathbf{x}^k)}{d\mathbf{x}^k}\\
&\quad + h\mathbf{F}_d(\mathbf{x}^{k-1}, \mathbf{x}^k) + \frac{\partial\boldsymbol{\Psi}(\mathbf{x}^k)}{\partial\mathbf{x}^k}\boldsymbol{\lambda}^k = 0\\
&\boldsymbol{\Psi}(\mathbf{x}^{k+1}) = 0
\end{aligned} \tag{B.6}$$

Appendix C. Derivation of conservative forces

The elastic energy of the natural straight and isotropic discrete thread (Eq. (3)) only depends on the dispositional variables due to the quasistatic updating material frame assumption. So the conservative force $\mathbf{F}_i^e$ obtained through the elastic energy is (Bergou et al., 2008),

$$\mathbf{F}_i^e = -\frac{d\mathbf{E}(\Gamma)}{d\mathbf{x}_i} = -\frac{\partial\mathbf{E}(\Gamma)}{\partial\mathbf{x}_i} - \frac{\partial\mathbf{E}(\Gamma)}{\partial\theta_{N_L}}\frac{\partial\theta_{N_L}}{\partial\mathbf{x}_i} \tag{C.1}$$

where

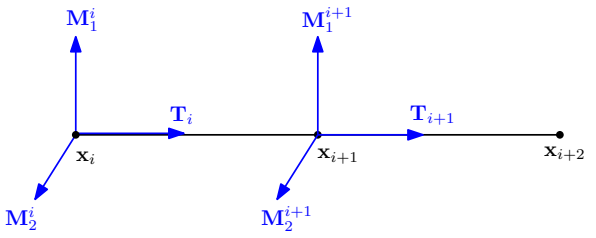
$$E(\Gamma) = \sum_{i=2}^{N_L}\frac{k_b}{\bar{l}_i}\left(\frac{2}{1+\mathbf{e}_{i-1}\cdot\mathbf{e}_i} - 1\right) + k_t\frac{(\theta_{N_L}-\theta_1)^2}{2\bar{L}}$$

and

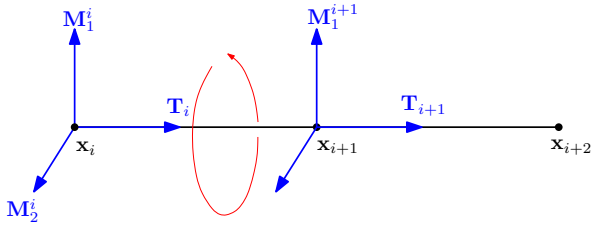
$$\mathbf{e}_i = \frac{\mathbf{x}_{i+1}-\mathbf{x}_i}{|\mathbf{x}_{i+1}-\mathbf{x}_i|}$$

The term $-\frac{\partial\mathbf{E}(\Gamma)}{\partial\mathbf{x}_i}$ is contributed by the bending elastic energy and can be expressed as,

$$\begin{aligned}
\frac{\partial\mathbf{E}(\Gamma)}{\partial\mathbf{x}_i} &= \frac{k_b}{\bar{l}_{i-1}}\frac{1}{(1+\mathbf{e}_{i-2}\cdot\mathbf{e}_{i-1})^2}\frac{\mathbf{x}_{i-1}-\mathbf{x}_{i-2}}{l_{i-2}\cdot l_{i-1}}\\
&\quad + \frac{k_b}{\bar{l}_i}\frac{1}{(1+\mathbf{e}_{i-1}\cdot\mathbf{e}_i)^2}\frac{\mathbf{x}_{i+1}-2\mathbf{x}_i+\mathbf{x}_{i-1}}{l_{i-1}\cdot l_i}\\
&\quad - \frac{k_b}{\bar{l}_{i+1}}\frac{1}{(1+\mathbf{e}_i\cdot\mathbf{e}_{i+1})^2}\frac{\mathbf{x}_{i+2}-\mathbf{x}_{i+1}}{l_i\cdot l_{i+1}}
\end{aligned}$$



(a) Torsion angle between segment i and segment $i+1$ is zero.



(b) Torsion angle between segment i and segment $i+1$ is 2π.

Fig. C.18. Top: the torsion angle between segment i and segment $i+1$ is zero. Bottom: the torsion angle between segment i and segment $i+1$ is 2π. One cannot calculate the torsion angle between segment i and segment $i+1$ purely based on the material frame of the discrete segments.

The term $-\frac{\partial \mathbf{E}(\Gamma)}{\partial \theta_{N_L}} \frac{\partial \theta_{N_L}}{\partial \mathbf{x}_i}$ is contributed by the torsional elastic energy. Following the work by Bergou et al. (2008), we know

$$-\frac{\partial \mathbf{E}(\Gamma)}{\partial \theta_{N_L}} \frac{\partial \theta_{N_L}}{\partial \mathbf{x}_i} = \frac{k_t(\theta_{N_L}-\theta_1)}{\bar{L}}\left(-\frac{(\mathbf{kb})_{i-1}}{2|\mathbf{e}_{i-1}|} - \frac{(\mathbf{kb})_i}{2|\mathbf{e}_{i-1}|} + \frac{(\mathbf{kb})_i}{2|\mathbf{e}_i|} + \frac{(\mathbf{kb})_{i+1}}{2|\mathbf{e}_i|}\right) \tag{C.2}$$

where

$$(\mathbf{kb})_i = \frac{2\mathbf{e}_{i-1}\times\mathbf{e}_i}{1+\mathbf{e}_{i-1}\cdot\mathbf{e}_i}$$

To calculate the total torsion angle $\theta_{N_L}-\theta_1$, we adopt a new accumulative technique. The angular velocity of i_{th} segment can be expressed as (Shuster, 1993)

$$\hat{\boldsymbol{\omega}}_i = \dot{\phi}_i\hat{\mathbf{n}}_i + \sin\phi\dot{\hat{\mathbf{n}}}_i + (1-\cos\phi_i)\hat{\mathbf{n}}_i\times\dot{\hat{\mathbf{n}}}_i \tag{C.3}$$

where $\hat{\boldsymbol{n}}_i = \frac{(\mathbf{kb})_i}{|(\mathbf{kb})_i|}$ is the unit vector along $(\mathbf{kb})_i$. Thus, the torsion angle change along each segment during each time step can be given by

$$\Delta\phi_i = h(\hat{\boldsymbol{\omega}}_i\cdot\mathbf{e}_i) = -\frac{h}{1+\mathbf{e}_{i-1}\cdot\mathbf{e}_i}(\mathbf{e}_{i-1}\times\mathbf{e}_i)\cdot(\dot{\mathbf{x}}_{i+1}-\dot{\mathbf{x}}_{i-1}) \tag{C.4}$$

Thus, the total torsion angle is

$$\theta_{N_L}-\theta_1 = \theta_{in} + h\sum_{i=2}^{N_L}\frac{(\mathbf{e}_{i-1}\times\mathbf{e}_i)\cdot(\dot{\mathbf{x}}_{i+1}-\dot{\mathbf{x}}_{i-1})}{1+\mathbf{e}_{i-1}\cdot\mathbf{e}_i} \tag{C.5}$$

where θ_{in} is the input torsion angle for the thread.

To calculate the total torsion angle in the numerical simulation, the accumulative angle technique (Eq. (C.5)) requires only some inexpensive float manipulations, and thus is very computational efficient. As shown in Fig. C.18, the top plot shows the torsion angle between segment i and segment $i+1$ is zero, and the bottom plot shows the torsion angle between segment i and segment $i+1$ is 2π. Since these two discrete thread segment have the same material frame, one cannot calculate the torsion angle between segment i and segment $i+1$ purely based on the material frame of the discrete segments. Thus the torsion angle between two continuous discrete segments is usually limited into 2π. However, our accumulative angle technique (Eq. (C.5)) does not have such limitations. As we can see from Eq. (C.5), the torsion angle is calculated based on the angle change between two continuous discrete segments in each time step. This indicates that the torsion angle between two continuous discrete segments can be any large value, as long as the angle change between two continuous discrete segments is smaller than 2π during a time step h, which is impossible to occur in our case due to the small time step we adopted.

Appendix D. The components of the matrix in contraction mapping

Let $\bar{\mathbf{x}} = \mathbf{x}^k + h\mathbf{v}^k + h^2\mathbf{M}^{-1}\left[-\frac{dE}{d\mathbf{x}^k} + \mathbf{F}_a\left(\frac{\mathbf{x}^k-\mathbf{x}^{k-1}}{h}\right)\right]$, and then the Eq. (9) can be reformed as

$$\mathbf{x}^{k+1} = \bar{\mathbf{x}} + h^2\mathbf{M}^{-1}\left[\frac{1}{h}\frac{\partial\boldsymbol{\Psi}}{\partial\mathbf{x}^k}\boldsymbol{\lambda}^k\right] \tag{D.1}$$

Substituting the Eq. (1) into the above equation, we can get,

$$\mathbf{x}_i^{k+1} = \bar{\mathbf{x}}_i + \frac{h^2}{m_i}\left(\frac{1}{h}(-2\lambda_i^k\mathbf{e}_i^k + 2\lambda_{i-1}^k\mathbf{e}_{i-1}^k)\right) \tag{D.2}$$

Let $\alpha_i = -\frac{2h\lambda_i}{m_i}$, such that

$$\mathbf{x}_i^{k+1} = \bar{\mathbf{x}}_i + \alpha_i^k\mathbf{e}_i^k - \alpha_{i-1}^k\mathbf{e}_{i-1}^k \tag{D.3}$$

Let $\bar{\mathbf{e}}_i = \bar{\mathbf{x}}_{i+1} - \bar{\mathbf{x}}_i$. Since $\mathbf{e}_i^{k+1} = \mathbf{x}_{i+1}^{k+1} - \mathbf{x}_i^{k+1}$, we can get,

$$\mathbf{e}_i^{k+1} = \bar{\mathbf{e}}_i + \alpha_{i+1}^k\mathbf{e}_{i+1}^k - 2\alpha_i^k\mathbf{e}_i^k + \alpha_{i-1}^k\mathbf{e}_{i-1}^k \tag{D.4}$$

Let $\underline{\boldsymbol{\alpha}}_i = [\alpha_{i-1}, \alpha_i, \alpha_{i+1}]^T$ and $\underline{\mathbf{e}}_i = [\mathbf{e}_{i-1}, -2\mathbf{e}_i, \mathbf{e}_{i+1}]$. The above equation can be reformed as

$$\mathbf{e}_i^{k+1} = \bar{\mathbf{e}}_i + \underline{\mathbf{e}}_i^k\underline{\boldsymbol{\alpha}}_i^k \tag{D.5}$$

Due to the inextensible constraints (1), $\mathbf{e}_i^{k+1}$ satisfies

$$[\mathbf{e}_i^{k+1}]^T\mathbf{e}_i^{k+1} = l_i^2 \tag{D.6}$$

where l_i is the original length of the i_{th} segment of discrete surgical thread. Since

$$[\mathbf{e}_i^{k+1}]^T\mathbf{e}_i^{k+1} = [\bar{\mathbf{e}}_i]^T\bar{\mathbf{e}}_i + 2[\bar{\mathbf{e}}_i]^T\underline{\mathbf{e}}_i^k\underline{\boldsymbol{\alpha}}_i^k + [\underline{\boldsymbol{\alpha}}_i^k]^T[\underline{\mathbf{e}}_i^k]^T\underline{\mathbf{e}}_i^k\underline{\boldsymbol{\alpha}}_i^k$$

Substituting Eq. (D.6) into the above equation, we know,

$$\left[2[\bar{\mathbf{e}}_i]^T\underline{\mathbf{e}}_i^k + [\underline{\boldsymbol{\alpha}}_i^k]^T[\underline{\mathbf{e}}_i^k]^T\underline{\mathbf{e}}_i^k\right]\underline{\boldsymbol{\alpha}}_i^k = l_i^2 - [\bar{\mathbf{e}}_i]^T\bar{\mathbf{e}}_i \tag{D.7}$$

Assembling the above equation, we can get,

$$[\mathbf{A}+\mathbf{B}(\boldsymbol{\alpha})]\boldsymbol{\alpha} = \mathbf{R} \tag{D.8}$$

where

$$A_{ij} = \begin{cases} [\bar{\mathbf{e}}_i]^T\mathbf{e}_{i-1}^k & \text{if } j = i-1 \\ -2[\bar{\mathbf{e}}_i]^T\mathbf{e}_i^k & \text{if } j = i \\ [\bar{\mathbf{e}}_i]^T\mathbf{e}_{i+1}^k & \text{if } j = i+1 \\ 0 & \text{if } j = others. \end{cases}$$

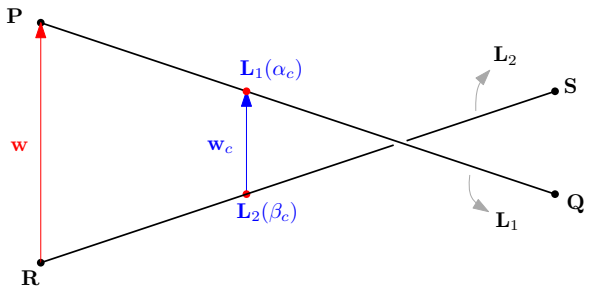


Fig. E.19. Two line segments $\mathbf{L}_1(\alpha) = \mathbf{P} + \alpha\mathbf{d}_1$ ($\mathbf{d}_1 = \mathbf{Q} - \mathbf{P}$, $\alpha \in [0, 1]$) and $\mathbf{L}_2(\beta) = \mathbf{R} + \beta\mathbf{d}_2$ ($\mathbf{d}_2 = \mathbf{S} - \mathbf{R}$, $\beta \in [0, 1]$) collide with each other ($|\mathbf{w}_c| < \Delta d$), where the shortest displacement between them is $\mathbf{w}_c$ with start point $\mathbf{L}_2(\beta_c)$ on $\mathbf{L}_2$ and end point $\mathbf{L}_1(\alpha_c)$ on $\mathbf{L}_1$.

$$B_{ij} = \begin{cases} \left[\alpha_{i-1}(\mathbf{e}^k_{i-1})^T - 2\alpha_i(\mathbf{e}^k_i)^T + \alpha_{i+1}(\mathbf{e}^k_{i+1})^T\right]\mathbf{e}^k_{i-1} & \text{if } j = i-1 \\ -2\left[\alpha_{i-1}(\mathbf{e}^k_{i-1})^T - 2\alpha_i(\mathbf{e}^k_i)^T + \alpha_{i+1}(\mathbf{e}^k_{i+1})^T\right]\mathbf{e}^k_i & \text{if } j = i \\ \left[\alpha_{i-1}(\mathbf{e}^k_{i-1})^T - 2\alpha_i(\mathbf{e}^k_i)^T + \alpha_{i+1}(\mathbf{e}^k_{i+1})^T\right]\mathbf{e}^k_{i+1} & \text{if } j = i+1 \\ 0 & \text{if } j = others. \end{cases}$$

$$\mathbf{R} = \left[l_1^2 - [\tilde{\mathbf{e}}_1]^T\tilde{\mathbf{e}}_1, \quad l_2^2 - [\tilde{\mathbf{e}}_2]^T\tilde{\mathbf{e}}_2, \quad \ldots, \quad l_i^2 - [\tilde{\mathbf{e}}_i]^T\tilde{\mathbf{e}}_i, \quad \ldots, \quad l_{N_L}^2 - [\tilde{\mathbf{e}}_{N_L}]^T\tilde{\mathbf{e}}_{N_L}\right]^T$$

Since $\alpha_i = -\frac{2h\lambda_i}{m_i}$, the equation is essentially

$$[\mathbf{A} + \mathbf{B}(\boldsymbol{\lambda}^k)]\boldsymbol{\lambda}^k = \mathbf{R}$$

Appendix E. Derivation of continuous penalty forces

Our method to handle the collision response is based on the model of continuous penalty force is developed by Tang et al. (2012a); hence our notation here follows their work. Since we are going to develop a model that is required to provide force feedback, the time step is very small (the simulation rate is at least 1000 Hz). As shown in Fig. E.19, we assume that two line segments $\mathbf{L}_1(\alpha) = \mathbf{P} + \alpha\mathbf{d}_1$ ($\mathbf{d}_1 = \mathbf{Q} - \mathbf{P}$, $\alpha \in [0, 1]$) and $\mathbf{L}_2(\beta) = \mathbf{R} + \beta\mathbf{d}_2$ ($\mathbf{d}_2 = \mathbf{S} - \mathbf{R}$, $\beta \in [0, 1]$) collide with each other ($|\mathbf{w}_c| < \Delta d$), where the shortest displacement between them is $\mathbf{w}_c$ with start point $\mathbf{L}_2(\beta_c)$ on $\mathbf{L}_2$ and end point $\mathbf{L}_1(\alpha_c)$ on $\mathbf{L}_1$. The motion of the four ends points of the two line segments are given by

$$\mathbf{P}(t) = \mathbf{P}_0 + \mathbf{V}_P t \qquad \mathbf{Q}(t) = \mathbf{Q}_0 + \mathbf{V}_Q t$$
$$\mathbf{R}(t) = \mathbf{R}_0 + \mathbf{V}_R t \qquad \mathbf{S}(t) = \mathbf{S}_0 + \mathbf{V}_S t$$

Let $a = \mathbf{d}_1 \cdot \mathbf{d}_1$, $b = \mathbf{d}_1 \cdot \mathbf{d}_2$, $c = \mathbf{d}_2 \cdot \mathbf{d}_2$, $d = \mathbf{d}_1 \cdot \mathbf{w}$, $e = \mathbf{d}_2 \cdot \mathbf{w}$, where $\mathbf{w} = \mathbf{P} - \mathbf{R}$, then we can obtain (Tang et al., 2012a)

$$\alpha_c(t) = \frac{be - cd}{ac - b^2}, \qquad \beta_c(t) = \frac{ae - bd}{ac - b^2} \tag{E.1}$$

Therefore we know

$$\begin{aligned} \mathbf{w}_c(t) &= \mathbf{L}_1(\alpha_c) - \mathbf{L}_2(\beta_c) \\ &= \mathbf{w} + \frac{(be - ac)\mathbf{d}_1 - (ae - bd)\mathbf{d}_2}{ac - b^2} \\ &= \mathbf{P} - \mathbf{R} + \frac{(be - ac)(\mathbf{Q} - \mathbf{P}) - (ae - bd)(\mathbf{S} - \mathbf{R})}{ac - b^2} \\ &= \left(1 - \frac{be - cd}{ac - b^2}\right)\mathbf{P} + \left(\frac{be - cd}{ac - b^2}\right)\mathbf{Q} - \left(1 - \frac{ae - bd}{ac - b^2}\right)\mathbf{R} \\ &\quad - \frac{ae - bd}{ac - b^2}\mathbf{S} \\ &= \omega_P(t)\mathbf{P}(t) + \omega_Q(t)\mathbf{Q}(t) - \omega_R(t)\mathbf{R}(t) - \omega_S(t)\mathbf{S}(t) \end{aligned} \tag{E.2}$$

where $\omega_P(t) = 1 - \frac{be-cd}{ac-b^2}$, $\omega_Q(t) = \frac{be-cd}{ac-b^2}$, $\omega_R(t) = 1 - \frac{ae-bd}{ac-b^2}$, and $\omega_S(t) = \frac{ae-bd}{ac-b^2}$. When $ac - b^2 = 0$, $\mathbf{L}_1$ and $\mathbf{L}_2$ are parallel to each other. The direction of $\mathbf{w}_c(t)$ can be given by,

$$\begin{aligned} \mathbf{m}'(t) &= (\mathbf{Q}(t) - \mathbf{P}(t)) \times (\mathbf{R}(t) - \mathbf{S}(t)) \\ &= [(\mathbf{Q}_0 - \mathbf{P}_0) + (\mathbf{V}_Q - \mathbf{V}_P)t] \times [(\mathbf{R}_0 - \mathbf{S}_0) + (\mathbf{V}_R - \mathbf{V}_S)t] \\ &= (\mathbf{Q}_0 - \mathbf{P}_0) \times (\mathbf{R}_0 - \mathbf{S}_0) + [(\mathbf{V}_Q - \mathbf{V}_P) \times (\mathbf{R}_0 - \mathbf{S}_0) \\ &\quad + (\mathbf{Q}_0 - \mathbf{P}_0) \times (\mathbf{V}_R - \mathbf{V}_S)]t + (\mathbf{V}_Q - \mathbf{V}_P) \times (\mathbf{V}_R - \mathbf{V}_S)t^2 \\ &= \mathbf{n}'_0 + \mathbf{n}'_1 t + \mathbf{n}'_2 t^2 \end{aligned} \tag{E.3}$$

where

$$\begin{aligned} \mathbf{n}'_0 &= (\mathbf{Q}_0 - \mathbf{P}_0) \times (\mathbf{R}_0 - \mathbf{S}_0) \\ \mathbf{n}'_1 &= (\mathbf{V}_Q - \mathbf{V}_P) \times (\mathbf{R}_0 - \mathbf{S}_0) + (\mathbf{Q}_0 - \mathbf{P}_0) \times (\mathbf{V}_R - \mathbf{V}_S) \\ \mathbf{n}'_2 &= (\mathbf{V}_Q - \mathbf{V}_P) \times (\mathbf{V}_R - \mathbf{V}_S) \end{aligned}$$

The unit direction of $\mathbf{w}_c(t)$ can then be expressed as

$$\mathbf{n}_E(t) = \frac{\mathbf{n}'_0 + \mathbf{n}'_1 t + \mathbf{n}'_2 t^2}{|\mathbf{n}'_0 + \mathbf{n}'_1 t + \mathbf{n}'_2 t^2|} \tag{E.4}$$

We can now find the impulse during the time interval h is (Tang et al., 2012a),

$$\mathbf{I} = k_s \int_t^{t+h} \mathbf{n}_E(t)^T[\Delta d\, \mathbf{n}_E(t) - \mathbf{w}_c(t)]\mathbf{n}_E(t)dt \tag{E.5}$$

Since the time step h is very small in our model, we can get the approximations for $\mathbf{n}_E(t)$ and $\mathbf{w}_c(t)$ as (Tang et al., 2012a),

$$\begin{aligned} \mathbf{w}_c(t) &= \omega_P(t)\mathbf{P}(t) + \omega_Q(t)\mathbf{Q}(t) - \omega_R(t)\mathbf{R}(t) - \omega_S(t)\mathbf{S}(t) \\ &\approx \mathbf{w}_c^{app}(t) = \omega_{P_0}\mathbf{P}(t) + \omega_{Q_0}\mathbf{Q}(t) - \omega_{R_0}\mathbf{R}(t) - \omega_{S_0}\mathbf{S}(t) \end{aligned}$$

where $\omega_{P_0} = 1 - \frac{b_0e_0 - c_0d_0}{a_0c_0 - b_0^2}$, $\omega_{Q_0} = \frac{b_0e_0 - c_0d_0}{a_0c_0 - b_0^2}$, $\omega_{R_0} = 1 - \frac{a_0e_0 - b_0d_0}{a_0c_0 - b_0^2}$, and $\omega_{S_0} = \frac{a_0e_0 - b_0d_0}{a_0c_0 - b_0^2}$, and

$$\begin{aligned} \mathbf{n}_E(t) &= \frac{\mathbf{n}'_0 + \mathbf{n}'_1 t + \mathbf{n}'_2 t^2}{|\mathbf{n}'_0 + \mathbf{n}'_1 t + \mathbf{n}'_2 t^2|} \approx \mathbf{n}_E^{app}(t) = \frac{\mathbf{n}'_0 + \mathbf{n}'_1 t + \mathbf{n}'_2 t^2}{|\mathbf{n}'_0|} \\ &= \frac{\mathbf{n}'_0}{|\mathbf{n}'_0|} + \frac{\mathbf{n}'_1}{|\mathbf{n}'_0|}t + \frac{\mathbf{n}'_2}{|\mathbf{n}'_0|}t^2 = \mathbf{n}_0 + \mathbf{n}_1 t + \mathbf{n}_2 t^2 \end{aligned}$$

where $\mathbf{n}_0 = \frac{\mathbf{n}'_0}{|\mathbf{n}'_0|}$, $\mathbf{n}_1 = \frac{\mathbf{n}'_1}{|\mathbf{n}'_0|}$, and $\mathbf{n}_2 = \frac{\mathbf{n}'_2}{|\mathbf{n}'_0|}$. And thus we can get the discrete impulse during the time interval $[t_1, t_2]$ as

$$\begin{aligned} \mathbf{I}_d^{dis} = \frac{h}{2}\Big\{ &\left(\mathbf{n}_E^{app}(t_1)\right)^T\left[\Delta d\, \mathbf{n}_E^{app}(t_1) - \mathbf{w}_c^{app}(t_1)\right]\mathbf{n}_E^{app}(t_1) \\ &+ \left(\mathbf{n}_E^{app}(t_2)\right)^T\left[\Delta d\, \mathbf{n}_E^{app}(t_2) - \mathbf{w}_c^{app}(t_2)\right]\mathbf{n}_E^{app}(t_2)\Big\} \end{aligned} \tag{E.6}$$

So the continuous penalty forces created during the time interval $[t_1, t_2]$ in numerical scheme can be expressed as

$$\begin{aligned} \mathbf{F}_c^{dis} = \frac{\mathbf{I}_d^{dis}}{h} = \frac{1}{2}\Big\{ &\left(\mathbf{n}_E^{app}(t_1)\right)^T\left[\Delta d\, \mathbf{n}_E^{app}(t_1) - \mathbf{w}_c^{app}(t_1)\right]\mathbf{n}_E^{app}(t_1) \\ &+ \left(\mathbf{n}_E^{app}(t_2)\right)^T\left[\Delta d\, \mathbf{n}_E^{app}(t_2) - \mathbf{w}_c^{app}(t_2)\right]\mathbf{n}_E^{app}(t_2)\Big\} \end{aligned} \tag{E.7}$$

References

Akutsu, Y., Wadati, M., 1988. Knots, links, braids and exactly solvable models in statistical mechanics. Commun. Math. Phys. 117 (2), 243–259.
Anjyo, K.-i., Usami, Y., Kurihara, T., 1992. A simple method for extracting the natural beauty of hair. SIGGRAPH Comput. Graph. 26 (2), 111–120.
Antman, S.S., 1995. Nonlinear Problems of Elasticity. Springer.
Bergou, M., Audoly, B., Vouga, E., Wardetzky, M., Grinspun, E., 2010. Discrete viscous threads. ACM Trans. Graph. (TOG) 29 (4), 116.

Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., Grinspun, E., 2008. Discrete elastic rods. ACM Trans. Graph. 27 (3), 63:1–63:12.
Bertails, F., 2009. Linear time super-helices. In: Computer Graphics Forum, 28. Wiley Online Library, pp. 417–426.
Bertails, F., Audoly, B., Cani, M.-P., Querleux, B., Leroy, F., Lévêque, J.-L., 2006. Super-helices for predicting the dynamics of natural hair. ACM Trans. Graph. (TOG) 25 (3), 1180–1187.
Bertails, F., Kim, T.-Y., Cani, M.-P., Neumann, U., 2003. Adaptive wisp tree: a multiresolution control structure for simulating dynamic clustering in hair motion. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, pp. 207–213.
Bertails-Descoubes, F., 2012. Super-clothoids. In: Computer Graphics Forum, 31. Wiley Online Library, pp. 509–518.
Bertails-Descoubes, F., Cadoux, F., Daviet, G., Acary, V., 2011. A nonsmooth Newton solver for capturing exact coulomb friction in fiber assemblies. ACM Trans. Graph. (TOG) 30 (1), 6.
Bertails-Descoubes, F., Cadoux, F., Daviet, G., Acary, V., 2011. A nonsmooth Newton solver for capturing exact coulomb friction in fiber assemblies. ACM Trans. Graph. (TOG) 30 (1), 6.
Bishop, R.L., 1975. There is more than one way to frame a curve. Am. Math. Mon. 82 (3), pp.246–251.
Bonanni, U., Kmoch, P., Magnenat-Thalmann, N., 2009. Haptic interaction with one-dimensional structures. In: Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology. ACM, pp. 75–78.
Bresenham, J.E., 1965. Algorithm for computer control of a digital plotter. IBM Syst. J. 4 (1), 25–30.
Bridson, R., Fedkiw, R., Anderson, J., 2002. Robust treatment of collisions, contact and friction for cloth animation. ACM Trans. Graph. 21 (3), 594–603.
Brown, J., Latombe, J.-C., Montgomery, K., 2004. Real-time knot-tying. Visual Comput. (2) 165–179.
Casati, R., Bertails-Descoubes, F., 2013. Super space clothoids. ACM Trans. Graph. (TOG) 32 (4), 48.
Choe, B., Choi, M.G., Ko, H.-S., 2005. Simulating complex hair with robust collision handling. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM, pp. 153–160.
Choe, B., Ko, H.-S., 2005. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. IEEE Trans. Visualization Comput.Graph. 11 (2), 160–170.
Cosserat, E., Cosserat, F., 1909. Theory of Deformable Bodies. Scientific Library A. Hermann and Sons, Paris.
Daldegan, A., Thalmann, N.M., Kurihara, T., Thalmann, D., 1993. An integrated system for modeling, animating and rendering hair. In: Computer Graphics Forum, 12. Wiley Online Library, pp. 211–221.
Derouet-Jourdan, A., Bertails-Descoubes, F., Daviet, G., Thollot, J., 2013. Inverse dynamic hair modeling with frictional contact. ACM Trans. Graph. (TOG) 32 (6), 159.
Dill, E., 1992a. Kirchhoff's theory of rods. Arch. Hist. Exact Sci. 44 (1), 1–23.
Dill, E.H., 1992b. Kirchhoff's theory of rods. Arch. Hist. Exact Sci. 44 (1), 1–23.
Duriez, C., Dubois, F., Kheddar, A., Andriot, C., 2006. Realistic haptic rendering of interacting deformable objects in virtual environments. IEEE Trans. Visualization Comput.Graph. 12 (1), 36–47.
El Saddik, A., 2007. The potential of haptics technologies. IEEE Instrum. Meas. Mag. 10 (1), 10–17.
Ericson, C., 2004. Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
Gallagher, A.G., Traynor, O., 2008. Simulation in surgery: opportunity or threat? Ir. J. Med. Sci. 177 (4), 283–287.
Ginsberg, J.H., 2001. Mechanical and Structural Vibrations: Theory and Applications, 1 Wiley.
Gornowicz, G., Borac, S., 2015. Efficient and stable approach to elasticity and collisions for hair animation. In: Proceedings of the 2015 Symposium on Digital Production. ACM, pp. 41–49.
Grégoire, M., Schömer, E., 2006. Interactive simulation of one-dimensional flexible parts. In: Proceedings of the 2006 ACM Symposium on Solid and Physical Modeling, pp. 95–103.
https://www.harvardmedsim.org/.
Huang, D., Tang, W., Wan, T. R., John, N. W., Gould, D., Ding, Y., Chen, Y.,. A new approach to haptic rendering of guidewires for use in minimally invasive surgical simulation. Comput. Anim. Virtual Worlds 22 (2-3).
Huang, D., Tang, W., Wan, T.R., John, N.W., Gould, D., Ding, Y., Chen, Y., 2011. A new approach to haptic rendering of guidewires for use in minimally invasive surgical simulation. Comput. Anim. Virtual Worlds 22 (2-3), 261–268.
Hüsken, N., 2014. Realtime Simulation of Stiff Threads for Microsurgery Training Simulation, Ph.D. thesis. Ruperto-Carola University of Heidelberg, Germany.
Iben, H., Meyer, M., Petrovic, L., Soares, O., Anderson, J., Witkin, A., 2013. Artistic simulation of curly hair. In: Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. ACM, pp. 63–71.
Kaufman, D., Tamstorf, R., Smith, B., Aubry, J., Grinspun, E., 2014. Adaptive nonlinearity for collisions in complex rod assemblies. ACM Trans. Graph. 33 (4). 12 pages. doi: 10.1145/2601097.2601100.
Kharevych, L., Yang, W., Tong, Y., Kanso, E., Marsden, J.E., Schröder, P., Desbrun, M., 2006. Geometric, variational integrators for computer animation. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 43–51.
A robust and real-time vascular intervention simulation based on Kirchhoff elastic rod. Comput. Med. Imaging Graph. 38 (8) (2014) 735–743
Kubiak, B., Pietroni, N., Ganovelli, F., Fratarcangeli, M., 2007. A robust method for real-time thread simulation. In: Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology. ACM, New York, NY, USA, pp. 85–88.
Kurihara, T., Anjyo, K.-i., Thalmann, D., 1993. Hair animation with collision detection. In: Models and Techniques in Computer Animation. Springer, pp. 128–138.
Langer, J., Singer, D.A., 1996. Lagrangian aspects of the Kirchhoff elastic rod. SIAM Rev. 38 (4), 605–618.
Lee, D.-W., Ko, H.-S., 2001. Natural hairstyle modeling and animation. Graphical Models 63 (2), 67–85.
Love, A.E.H., 2013. A Treatise on the Mathematical Theory of Elasticity, 1. Cambridge University Press.
Marsden, J.E., West, M., 2001. Discrete mechanics and variational integrators. Acta Numerica (10) 357–514.
Michels, D.L., Mueller, J.P.T., Sobottka, G.A., 2015. A physically based approach to the accurate simulation of stiff fibers and stiff fiber meshes. Comput. Graph. 53, 136–146.
Mirtich, B.V., 1996. Impulse-Based Dynamic Simulation of Rigid Body Systems Ph.D. thesis. AAI9723116
Moore, P., Molloy, D., 2007. A survey of computer-based deformable models. In: Machine Vision and Image Processing Conference, 2007. IMVIP 2007. International, pp. 55–66.
Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J., 2007. Position based dynamics. J. Vis. Commun. Image Represent. 18 (2), 109–118.
Müller, M., Kim, T.-Y., Chentanez, N., 2012. Fast simulation of inextensible hair and fur. In: Bender, J., Kuijper, A., Fellner, D.W., Guérin, E. (Eds.), VRIPHYS. Eurographics Association, pp. 39–44.
Munson, B.R., Rothmayer, A.P., Okiishi, T.H., Huebsch, W.W., 2012. Fundamentals of Fluid Mechanics. Wiley.
Nakajima, M., Ming, K.W., Takashi, H., 1999. Generation of 3d hair model from multiple pictures. IEEE Comput. Graph. Appl. 12.
Nealen, A., Mller, M., Keiser, R., Boxerman, E., Carlson, M., 2006. Physically based deformable models in computer graphics. Comput. Graph. Forum 25 (4), 809–836.
Otaduy, M.A., Tamstorf, R., Steinemann, D., Gross, M., 2009. Implicit contact handling for deformable objects. Comput. Graph. Forum 28 (2), 559–568.
Pai, D.K., 2002. STRANDS: interactive simulation of thin solids using Cosserat models. Comput. Graph. Forum 21, 347–352.
Finite element formulation of large deformation impact-contact problems with friction. Comput. Struct. 37 (3), 319–331
Punak, S., Kurenov, S., Simplified Cosserat rod for interactive suture modeling.Stud. Health Technol. Inform. 163, 466–472.
Rosenblum, R.E., Carlson, W.E., Tripp, E., 1991. Simulating the structure and dynamics of human hair: modelling, rendering and animation. J. Visualization Comput.Anim. 2 (4), 141–148.
Rubin, M.B., 2013. Cosserat Theories: Shells, Rods and Points, 79. Springer Science & Business Media.
Shuster, M.D., 1993. A survey of attitude representations. J. Astronaut. Sci. 41 (4), 439–517.
Skeel, R.D., Zhang, G., Schlick, T., 1997. A family of symplectic integrators: stability, accuracy, and molecular dynamics applications. SIAM J. Sci. Comput. 18 (1), 203–222.
SL, D., 2002. A critical approach to medical simulation. Bull. Am. Coll. Surg. 87 (11), 12–18.
Spillmann, J., Harders, M., 2010. Inextensible elastic rods with torsional friction based on Lagrange multipliers. Comput. Anim. Virtual Worlds 21 (6), 561–572.
Spillmann, J., Teschner, M., 2007. Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 63–72.
Spillmann, J., Teschner, M., 2007. CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 63–72.
Spillmann, J., Teschner, M., 2008. An adaptive contact model for the robust simulation of knots. Comput. Graph. Forum 27 (2), 497–506.
Sueda, S., Jones, G.L., Levin, D.I., Pai, D.K., 2011. Large-scale dynamic simulation of highly constrained strands. In: ACM Transactions on Graphics (TOG), 30. ACM, p. 39.
Tang, M., Manocha, D., Otaduy, M. A., Tong, R., 2012a. Continuous penalty forces.
Tang, W., Lagadec, P., Gould, D., Wan, T., Zhai, J., How, T., 2010. A realistic elastic rod model for real-time simulation of minimally invasive vascular interventions. Vis. Comput. 26 (9), 1157–1165.
Tang, W., Wan, T.R., Gould, D., How, T., John, N., 2012. A stable and real-time nonlinear elastic approach to simulating guidewire and catheter insertions based on Cosserat rod. Biomed. Eng. IEEE Trans. 59 (8), 2211–2218.
Umetani, N., Schmidt, R., Stam, J., 2014. Position-based elastic rods. In: ACM SIGGRAPH 2014 Talks, pp. 47:1–47:1.
Wan, T.R., Tang, W., Huang, D., 2012. Real-time simulation of long thin flexible objects in interactivevirtual environments. In: Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology. ACM, pp. 85–92.
Wang, F., Burdet, E., Vuillemin, R., Bleuler, H., 2005. Knot-tying with visual and force feedback for vr laparoscopic training. In: Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the, pp. 5778–5781.

Ward, K., Bertails, F., Kim, T.-Y., Marschner, S.R., Cani, M.-P., Lin, M.C., 2007. A survey on hair modeling: styling, simulation, and rendering. IEEE Trans. Visualization Comput.Graph. 13 (2), 213–234.
Ward, K., Lin, M.C., 2003. Adaptive grouping and subdivision for simulating hair dynamics. In: Computer Graphics and Applications, 2003. Proceedings. 11th Pacific Conference on. IEEE, pp. 234–243.
West, M., 2004. Variational Integrators, Ph.D. thesis. California Institute of Technology, Pasadena, California, USA.
Witkin, A., 1997. Physically based modeling: principles and practice. In: COMPUTER GRAPHICS, pp. 11–21.