Soheil Fatehiboroujeni
Department of Mechanical Engineering,
University of California, Merced,
Merced, CA 95343
e-mail: sfatehiboroujeni@ucmerced.edu

Harish J. Palanthandalam-Madapusi
Mechanical Engineering,
Indian Institute of Technology,
Gandhinagar Palaj,
Gandhinagar 382355, India
e-mail: harish@iitgn.ac.in

Sachin Goyal
Department of Mechanical Engineering,
University of California, Merced,
Merced, CA 95343
e-mail: sgoyal2@ucmerced.edu

Computational Rod Model With User-Defined Nonlinear Constitutive Laws

Computational rod models have emerged as efficient tools to simulate the bending and twisting deformations of a variety of slender structures in engineering and biological applications. The dynamics of such deformations, however, strongly depends on the constitutive law in bending and torsion that, in general, may be nonlinear, and vary from material to material. Jacobian-based computational rod models require users to change the Jacobian if the functional form of the constitutive law is changed, and hence are not user-friendly. This paper presents a scheme that automatically modifies the Jacobian based on any user-defined constitutive law without requiring symbolic differentiation. The scheme is then used to simulate force-extension behavior of a coiled spring with a softening constitutive law. [DOI: 10.1115/1.4041028]

1 Introduction

The continuum mechanics-based elastic rod models [1,2] have evolved as viable tools to efficiently simulate the bending and twisting deformations of a variety of slender structures. Slender structures occur not only in structural engineering applications but also in several biological and nanoscale applications. Some of the recently emerging applications involving slender structures include nanoscale biological filaments [3–5] (e.g., DNA molecules [6–8], microtubules [9,10], cila and flagella [11] and several others [12–14]), carbon nanotubes [15], and silver nanowires [16]. Several of these slender structures undergo very large twisting and bending deformations. For example, biological filaments such as DNA perform their biological functions via well-regulated structural deformations that involve large twisting and bending deformations [17]. Computational rod models are capable of simulating the nonlinear dynamics of such deformations by employing appropriate constitutive laws in bending and torsion [18,19], including intertwining with self-contact [20].

A continuum-rod model, in general, consists of dynamic equilibrium equations and compatibility equations, which need to be solved respecting the prescribed constitutive law. Although dynamic equilibrium equations and compatibility conditions, which we henceforth refer to as the *rod model*, remain the same for all slender structures, the key distinguishing factor is the constitutive law. Traditional models assume linear constitutive laws in bending and torsion. However, nonlinearities in the constitutive laws strongly influence the dynamics of such large deformations. For example, both DNA [21,22] and microtubules [23,24] are known to kink suggesting that these filaments must have nonconvex stored energy functions [25]. The material nonlinearities are also known to influence the onset of buckling of nanorods and nanotubes [26,27] as well as the postbucking behavior [28]. Such details of buckling dynamics of filaments play an important role in biological systems [29]. Yet, highly limited knowledge of the constitutive laws of biological filaments and nanorods has been a major roadblock for applicability of continuum rod models. Accurate identification of constitutive law parameters that are not directly measurable involve challenging inverse methods, while a computational forward model can predict deformations given initial and boundary conditions as well as parameters defining the constitute law, inverse models seek to identify unknown parameters of the constitutive law given some measurements of deformations as well as the equilibrium and compatibility equations. However, in developing inverse methods, a review of which can be found here [30,31], the need first arises for a computational forward model that can efficiently simulate the deformations with any *user-defined* nonlinear constitutive law. Therefore, the inverse models for nonlinear constitutive laws have to evolve together with the development of corresponding forward models. Developing a computational rod model that is capable of incorporating user-defined constitutive laws, which is claimed as the contribution of this paper, is also a stepping stone for addressing the research on estimation and identification of constitutive law parameters.

Given the wide range of applications in which the computational rod model is being employed and the commonalities among these applications, a user-friendly computational rod model framework that is applicable across these applications will greatly impact research in these areas. As discussed earlier, while the equations of dynamic equilibrium and compatibility are the same across these varieties of applications, the key difference is the exact functional form of the nonlinear constitutive law. This exact functional form of nonlinear constitutive law varies from material to material depending on their atomistic structures [32,33]. Thus, to develop a user-friendly computational tool employing rod model, an important required feature is to allow the user to prescribe or input any functional form of the constitutive law. Currently available computational rod models, such as Ref. [34] that simulate the deformations, are unable to allow the user to define different constitutive laws in a user-friendly way. This is so because the computational model numerically solves the governing nonlinear differential equations with Jacobian-based methods, and the constitutive law contributes to the Jacobian. Any change in the form of the constitutive law thus necessitates extensive rewriting of the parts of the code that relate to the Jacobian. This poses a barrier in modularizing the constitutive law as a user input in the computer program. One way to circumvent this hurdle is to introduce symbolic differentiation to compute the Jacobian. However, symbolic differentiation is computationally very sluggish, and must be avoided, especially when they are required iteratively within "for" loops. Another brute-force approach is to preprogram a library of different functional forms of the nonlinear constitutive law (along with the respective Jacobians-related codes) and allow

Contributed by the Design Engineering Division of ASME for publication in the Journal of Computational and Nonlinear Dynamics. Manuscript received November 28, 2017; final manuscript received July 25, 2018; published online August 22, 2018. Assoc. Editor: Zdravko Terze.

users to choose from the list of available functions, but this approach is neither elegant nor can it handle unanticipated constitutive laws.

Yet another approach is automatic differentiation (AD) [35], which is as accurate as the method of symbolic differentiation and also faster than that. However, the approach that this paper presents is simpler in programming than AD. The main focus of this paper is to allow a user-friendly computational rod model environment wherein a user who is not necessarily familiar with programming of the computational rod model can easily input a constitutive law, the rod parameters, boundary conditions, and can compute the rod-deformations in a short span of time required to setup the simulation. We achieve this by programming into the model (and Jacobian) not a specific functional form of the constitutive law, but rather basis functions that can be used to represent arbitrary functions. For instance, if powers of the deformation variables are coded into the Jacobian, they can be used to represent any analytic function (by Taylor series expansion) of the deformation variables by simply supplying the appropriate coefficients. This is therefore an efficient approach that has one preprogrammed Jacobian that can accommodate any user-defined constitutive law without requiring symbolic differentiation. In this scheme, the user simply enters the constitutive law desired, and the computer program first expands the user-defined constitutive law into a series (such as Taylor series) or in terms of appropriately chosen basis functions. The coefficients of the series (or the basis functions) are automatically passed on to the Jacobian, the form of which is hard-coded based on the derivative of the series or the basis functions. Thus, this scheme numerically (not symbolically) passes on the information of the functional form of the user-defined constitutive law directly to the Jacobian via coefficients, and no reprogramming by the user is needed.

The above-proposed approach is demonstrated in this paper by implementing the above scheme into an existing computational rod model [34] using MATLAB, and testing it with hardening, softening, and other types of constitutive laws including the implicit forms. We analyze the results (including the accuracy and speed) for several loading scenarios leading to highly nonlinear rod deformations. The impact of the choice of the basis functions in the proposed method is also examined.

The paper is organized as follows: Section 2 summarizes the mathematical formulation of the dynamic rod model [2] with a general form of a nonlinear constitutive law in bending and torsion. The mathematical formulation consists of nonlinear partial differential equations that need to be integrated numerically for any given initial and boundary conditions. Section 3 begins with describing the computational approach for the numerical integration that was adopted in Ref. [2]. Section 3 then continues further with introducing the proposed strategy for incorporating the user-defined nonlinear constitutive law in the computational approach. Section 4 presents some case studies to compare the proposed strategy with existing (but non user-friendly) approaches.

2 Mathematical Formulation of Rod Model

The mathematical formulation of the dynamic rod model that we use [2] employs the classical approach of the Kirchhoff [36], which assumes each cross section of the rod to be rigid. The position and orientation of each cross section are determined in space s and time t by tracking the transformation of a body-fixed frame $\hat{a}_i(s,t)$ with respect to an inertial frame of reference $\hat{e}_i$ that are shown in Fig. 1, where subscript $i = 1, 2, 3$.

Vector $\mathbf{R}(s,t)$ defines the position of the body-fixed reference frame $\hat{a}_i(s,t)$ relative to the inertial frame of reference $\hat{e}_i$. The spatial derivative of $\mathbf{R}(s,t)$, which we denote by $\mathbf{r}(s,t)$, is in the tangential direction along the centerline. Its deviation from the unit normal of the cross section determines the shear and stretch at the cross section. The change in magnitude of $\mathbf{r}(s,t)$ quantifies the extension or compression along s, and the change in its orientation with respect to the body-fixed frame $\hat{a}_i(s,t)$ quantifies

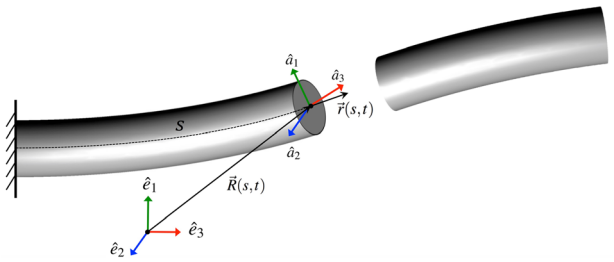

Fig. 1 The motion of each cross section of the rod at length *s* and time *t* is determined by tracking the transformations of the body-fixed frame $\hat{a}_i(s,t)$ with respect to the inertial frame of reference $\hat{e}_i$

shear. Furthermore, the spatial rate of change of cross section orientation is denoted by $\boldsymbol{\kappa}(s,t)$, and describes the curvature and twist of the rod. In general, the rod may be intrinsically curved and twisted in its stress-free state. We denote this stress-free curvature and twist by $\boldsymbol{\kappa}_0(s)$. The stress distribution over the cross section of the rod results in a net internal force $\mathbf{f}(s,t)$ and a net internal moment $\mathbf{q}(s,t)$. Moreover, the translational velocity $\mathbf{v}(s,t)$ and angular velocity $\boldsymbol{\omega}(s,t)$ of the frame $\hat{a}_i(s,t)$ are used to describe the rigid-body motion of the cross section.

Thus, the dynamics of rod deformation is described by six vector fields, $\boldsymbol{\kappa}(s,t)$ and $\mathbf{r}(s,t)$ that represent deformation of the rod, $\mathbf{q}(s,t)$ and $\mathbf{f}(s,t)$ that represent the restoring moment and force, and $\mathbf{v}(s,t)$ and $\boldsymbol{\omega}(s,t)$ that represent motion of each cross section. These six vector fields must satisfy equations of dynamic equilibrium and compatibility as well as a constitutive law.

2.1 Equilibrium and Compatibility Equations.

By applying Newton's second law to an element of the rod with infinitesimal length, the equations of equilibrium (1) and (2) are derived. The compatibility Eqs. (3) and (4) follow from the space–time continuity of the cross section position $\mathbf{R}(s,t)$, and the space-time continuity of the transformation that maps $\hat{a}_i(s,t)$ to $\hat{e}_i$, respectively. A detailed discussion and derivation of these equations are given in Ref. [2]

$$m\left(\frac{\partial \mathbf{v}}{\partial t} + \boldsymbol{\omega} \times \mathbf{v}\right) - \left(\frac{\partial \mathbf{f}}{\partial s} + \boldsymbol{\kappa} \times \mathbf{f}\right) - \mathbf{F} = \mathbf{0} \tag{1}$$

$$\underline{\mathbf{I_m}}\,\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\omega} \times \underline{\mathbf{I_m}}\,\boldsymbol{\omega} - \left(\frac{\partial \mathbf{q}}{\partial s} + \boldsymbol{\kappa} \times \mathbf{q}\right) + \mathbf{f} \times \mathbf{r} - \mathbf{Q} = \mathbf{0} \tag{2}$$

$$\frac{\partial \mathbf{r}}{\partial t} + \boldsymbol{\omega} \times \mathbf{r} - \left(\frac{\partial \mathbf{v}}{\partial s} + \boldsymbol{\kappa} \times \mathbf{v}\right) = \mathbf{0} \tag{3}$$

$$\frac{\partial \boldsymbol{\kappa}}{\partial t} - \left(\frac{\partial \boldsymbol{\omega}}{\partial s} + \boldsymbol{\kappa} \times \boldsymbol{\omega}\right) = \mathbf{0} \tag{4}$$

In Eqs. (1)–(4), all the derivatives are relative to the body-fixed reference frame, m is the mass of the rod per unit length, and $\underline{\mathbf{I_m}}(s)$ is a 3×3 tensor of the moments of inertia per unit length. The interaction of the rod with the environment is captured with external force per unit length $\mathbf{F}(s,t)$ as well as the external moment per unit length $\mathbf{Q}(s,t)$.

2.2 Constitutive Law.

The differential equations of equilibrium and compatibility have to be solved together with a constitutive law to find the six unknown vector fields. The constitutive law describes the relationship between the deformation of the rod and the restoring internal force and moment. In general, an elastic constitutive law can be written as the following set of implicit algebraic equations in a $\mathbb{R}^6$ space:

$$\Psi(\mathbf{q},\mathbf{f},\boldsymbol{\kappa},\mathbf{r},s)=0,\quad \Psi\in\mathbb{R}^3\times\mathbb{R}^3\times\mathbb{R}^3\times\mathbb{R}^3\times\mathbb{R}\rightarrow\mathbb{R}^6 \tag{5}$$

However, for an inextensible rod, $\mathbf{r}(s,t)$ has unit magnitude, and for an unshearable rod, its orientation relative to the body-fixed frame is constant. Therefore, for an inextensible and unshearable rod, $(\partial\mathbf{r}(s,t)/\partial t)=\mathbf{0}$. Extension and shear are indeed negligible in majority of applications of thin rods in low tension or compression, for which deformation of the rod is dominated by large bending and twisting. So, we describe our numerical approach in this paper for inextensible and unshearable rods, but recognize that the same approach is applicable to a geometrically exact rod as well. We further assume the constitutive law to be such that the restoring moment is an explicit function of curvature and twist and not dependent on the internal forces. Thus, the constitutive law takes the following form:

$$\mathbf{q}=\boldsymbol{\psi}(\boldsymbol{\kappa}-\boldsymbol{\kappa}_0,s),\quad \boldsymbol{\psi}\in\mathbb{R}^3\times\mathbb{R}\rightarrow\mathbb{R}^3 \tag{6}$$

where we recall that $\boldsymbol{\kappa}_0$ describes the initial stress-free curvature and twist of the rod. Note that in this case, we have one less vector field to solve for ($\mathbf{r}(s,t)$ is constant, not variable). Furthermore, the constitutive law (Eq. (6)) can be directly substituted in the angular momentum equation (Eq. (2)) to eliminate $\mathbf{q}(s,t)$. In particular, the constitutive law is used to express the internal moment $\mathbf{q}$ and its derivatives in terms of $\boldsymbol{\kappa}$ using the total derivative of function $\boldsymbol{\psi}$ given in the below equation:

$$\frac{\partial\mathbf{q}}{\partial s}=\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\kappa}}\frac{\partial\boldsymbol{\kappa}}{\partial s}-\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\kappa}_0}\frac{d\boldsymbol{\kappa}_0}{ds}+\frac{\partial\boldsymbol{\psi}}{\partial s} \tag{7}$$

With this substitution, we are left with four vector differential equations to solve for four unknown vector fields, namely, $\mathbf{v}(s,t)$, $\boldsymbol{\omega}(s,t)$, $\boldsymbol{\kappa}(s,t)$, and $\mathbf{f}(s,t)$. The four partial differential equations of equilibrium and compatibility can be assembled as described next to apply a numerical integration scheme.

2.3 Assembled System of Equations. The equations of the inextensible and unshearable rod model (Eqs. (1)–(4)) are assembled after substitution of the constitutive law (Eq. (6) and its spatial derivative Eq. (7)) to write them in the following compact form:

$$\mathbf{M}\frac{\partial\mathbf{Y}}{\partial t}+\mathbf{K}\frac{\partial\mathbf{Y}}{\partial s}+\mathbf{F}=\mathbf{0} \tag{8}$$

Here

$$\mathbf{Y}=\begin{bmatrix}\mathbf{v}\\ \boldsymbol{\omega}\\ \boldsymbol{\kappa}\\ \mathbf{f}\end{bmatrix} \tag{9}$$

is a 12×1 column matrix of unknowns to be solved for that describe the dynamic state of the system. $\mathbf{M}$ and $\mathbf{K}$ are 12×12 matrices that describe the overall inertia and stiffness of the system, and $\mathbf{F}$ is a 12×1 column matrix of nonhomogeneous terms. For our numerical code, we assembled the four equations in the following order: Eqs. (3),(4),(2), and (1) that resulted in the following form of $\mathbf{M}$, $\mathbf{K}$, and $\mathbf{F}$:

$$\mathbf{M}=\begin{bmatrix}\mathbf{O}&\mathbf{O}&\mathbf{O}&\mathbf{O}\\ \mathbf{O}&\mathbf{O}&\mathbf{I}&\mathbf{O}\\ \mathbf{O}&\underline{\mathbf{I}_\mathbf{m}}&\mathbf{O}&\mathbf{O}\\ m\mathbf{I}&\mathbf{O}&\mathbf{O}&\mathbf{O}\end{bmatrix} \tag{10}$$

$$\mathbf{K}=-\begin{bmatrix}\mathbf{I}&\mathbf{O}&\mathbf{O}&\mathbf{O}\\ \mathbf{O}&\mathbf{I}&\mathbf{O}&\mathbf{O}\\ \mathbf{O}&\mathbf{O}&\dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\kappa}}&\mathbf{O}\\ \mathbf{O}&\mathbf{O}&\mathbf{O}&\mathbf{I}\end{bmatrix} \tag{11}$$

$$\mathbf{F}=\begin{bmatrix}\boldsymbol{\omega}\times\mathbf{r}-\boldsymbol{\kappa}\times\mathbf{v}\\ -\boldsymbol{\kappa}\times\boldsymbol{\omega}\\ -\left(\dfrac{\partial\boldsymbol{\psi}}{\partial s}-\dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\kappa}_0}\dfrac{d\boldsymbol{\kappa}_0}{ds}\right)+\boldsymbol{\omega}\times\underline{\mathbf{I}_\mathbf{m}}\boldsymbol{\omega}+\mathbf{f}\times\mathbf{r}-\boldsymbol{\kappa}\times\boldsymbol{\psi}-\mathbf{Q}\\ m(\boldsymbol{\omega}\times\mathbf{v})-\boldsymbol{\kappa}\times\mathbf{f}-\mathbf{F}\end{bmatrix} \tag{12}$$

In Eqs. (10) and (11), the symbols **I** and **O,** respectively, refer to the identity tensor and null tensor (with the dimensions of 3×3). Note that rows in **M**, **K**, and **F** correspond to the four equations, and their order is immaterial.

The *Generalized-α method* [37] is adopted to compute the numerical solution of this system, subjected to necessary and sufficient initial and boundary conditions. To compute the geometric shape of the rod, we use the method of incremental rotation [2,38] to construct the transformation matrix from body-fixed frame to an inertial frame. The rod model formulation presented here is distinctly different from geometrically local approaches that were first proposed by Simo and Vu-Quoc [39] and describe configuration of a slender structure locally, by using displacements and the rotation of a cross section. There are many choices to parametrize the rotation of a cross section in geometrically local approaches and various approaches such as Euler angles [40], the rotation vector [41], and Euler parameters [42] have been used. In Sec. 3, we illustrate how the Generalized-α method is applicable to a rod with nonlinear and nonhomogeneous constitutive law. Here, note that **K** and **F** have the contribution of the nonlinear constitutive law from its derivative given by Eq. (7), and therefore must have contribution to the Jacobian of the system. In the next section, we introduce our strategy for implementation of the user-defined nonlinear constitutive law in the computational approach.

3 Numerical Algorithm With User-Defined Constitutive Law

We first devise a numerical algorithm in Subsec. 3.1 that is general enough to allow for any constitutive law by expressing it in terms of the arbitrary function $\boldsymbol{\psi}$. In doing so, we identify the parts of the algorithm that get affected by $\boldsymbol{\psi}$, the constitutive law. Then, in Subsec. 3.2, we introduce how a symbolic implementation would (traditionally) take care of any constitutive law given any arbitrary function $\boldsymbol{\psi}$. This is the most accurate approach, but computationally extremely sluggish due to recurring symbolic differentiations. Finally, in Subsec. 3.3, we introduce our user-friendly and computationally efficient strategy for inputting the user-defined nonlinear constitutive law that circumvents the need of symbolic differentiation.

3.1 Generalized-α Discretization. For a rod with linear and homogenous constitutive law, the matrices **M** and **K** are constant and are not discretized in space and time. Therefore, Eq. (8) can be discretized as

$$\mathbf{M}\left(\frac{\partial\mathbf{Y}}{\partial t}\right)_{1-\alpha_s}^{1-\alpha_t}+\mathbf{K}\left(\frac{\partial\mathbf{Y}}{\partial s}\right)_{1-\beta_s}^{1-\beta_t}+\mathbf{F}_{1-\beta_s}^{1-\beta_t}=\mathbf{0} \tag{13}$$

in which the notation $\mathbf{A}^{1-x}$ and $\mathbf{A}_{1-x}$ for any quantity **A** and any parameter x are defined as follows:

$$\mathbf{A}^{1-x} = (1-x)\mathbf{A}^{\mathbb{i}} + x\mathbf{A}^{\mathbb{i}-1} \tag{14}$$

$$\mathbf{A}_{1-x} = (1-x)\mathbf{A}_{\mathbb{j}} + x\mathbf{A}_{\mathbb{j}-1} \tag{15}$$

The indices $\mathbb{i}$ and $\mathbb{j}$ enumerate the discretized nodes in time and space, respectively. The parameters α_t and α_s are called mass averaging parameters and β_t and β_s are stiffness averaging.

In the case of nonlinear constitutive laws, the term $(\partial\boldsymbol{\psi}/\partial\boldsymbol{\kappa})$ in the stiffness matrix $\mathbf{K}$, depends on the curvature and twist vector; therefore, it varies in space and time and is discretized as given in the following equation:

$$\begin{aligned}\mathbf{K}_{1-\beta_s}^{1-\beta_t} = [&(1-\beta_t)((1-\beta_s)\mathbf{K}_{\mathbb{j}}^{\mathbb{i}} + \beta_s\mathbf{K}_{\mathbb{j}-1}^{\mathbb{i}}) \\ &+ \beta_t((1-\beta_s)\mathbf{K}_{\mathbb{j}}^{\mathbb{i}-1} + \beta_s\mathbf{K}_{\mathbb{j}-1}^{\mathbb{i}-1})]\end{aligned} \tag{16}$$

For nonhomogeneous rods, the mass matrix $\mathbf{M}$ varies along the length of rod but not in time. Therefore, the discretized form of the system can be written as

$$\mathbf{M}_{1-\alpha_s}\left(\frac{\partial\mathbf{Y}}{\partial t}\right)_{1-\alpha_s}^{1-\alpha_t} + \mathbf{K}_{1-\beta_s}^{1-\beta_t}\left(\frac{\partial\mathbf{Y}}{\partial s}\right)_{1-\beta_s}^{1-\beta_t} + \mathbf{F}_{1-\beta_s}^{1-\beta_t} = \mathbf{0} \tag{17}$$

The derivatives of $\mathbf{Y}$ are then discretized with Newmark-like formulation. In Eqs. (18) and (19), the Newmark constants γ_t and γ_s control the averaging of time and space derivatives

$$\left(\frac{\partial\mathbf{Y}}{\partial t}\right)^{\mathbb{i}} = \frac{\mathbf{Y}^{\mathbb{i}} - \mathbf{Y}^{\mathbb{i}-1}}{\gamma_t\Delta t} - \frac{1-\gamma_t}{\gamma_t}\left(\frac{\partial\mathbf{Y}}{\partial t}\right)^{\mathbb{i}-1} \tag{18}$$

$$\left(\frac{\partial\mathbf{Y}}{\partial s}\right)_{\mathbb{j}} = \frac{\mathbf{Y}_{\mathbb{j}} - \mathbf{Y}_{\mathbb{j}-1}}{\gamma_s\Delta s} - \frac{1-\gamma_s}{\gamma_s}\left(\frac{\partial\mathbf{Y}}{\partial s}\right)_{\mathbb{j}-1} \tag{19}$$

Applying this scheme to Eq. (17) results in an algebraic equation with nonlinear terms of $\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}}$ and $\mathbf{Y}_{\mathbb{j}-1}^{\mathbb{i}}$ and linear terms of $(\partial\mathbf{Y}/\partial s)_{\mathbb{j}-1}^{\mathbb{i}}$. In Eq. (20), $\mathbf{A}(\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}})$ and $\mathbf{B}(\mathbf{Y}_{\mathbb{j}-1}^{\mathbb{i}})$ represent the nonlinear terms, $\mathbf{H}$ contains all of the known terms from previous time-step $(\mathbb{i}-1)$, and the matrix $\hat{\mathbf{K}}$ is given in the following equation:

$$\hat{\mathbf{K}}\left(\frac{\partial\mathbf{Y}}{\partial s}\right)_{\mathbb{j}-1}^{\mathbb{i}} + \mathbf{A}\left(\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}}\right) + \mathbf{B}\left(\mathbf{Y}_{\mathbb{j}-1}^{\mathbb{i}}\right) = \mathbf{H} \tag{20}$$

$$\hat{\mathbf{K}} = (1-\beta_t)\left(\beta_s - (1-\beta_s)\left(\frac{1-\gamma_s}{\gamma_s}\right)\right)\mathbf{K}_{1-\beta_s}^{1-\beta_t} \tag{21}$$

To derive an integrable linear algebraic equation in the space, Eq. (20) is linearized about a guessed solution. The linearization requires the calculation of the Jacobian of the terms $\mathbf{A}(\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}})$ and $\mathbf{B}(\mathbf{Y}_{\mathbb{j}-1}^{\mathbb{i}})$ that we will show with $\mathbf{A}_{\mathbf{Y}}$ and $\mathbf{B}_{\mathbf{Y}}$

$$\mathbf{A}_{\mathbf{Y}} = (1-\alpha_t)(1-\alpha_s)\left(\frac{\mathbf{M}_{1-\alpha_s}}{\gamma_t\Delta t}\right) + (1-\beta_t)(1-\beta_s)\left(\frac{\mathbf{K}_{1-\beta_s}^{1-\beta_t}}{\gamma_s\Delta s} + \mathbf{F}_{\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}}}\right) \tag{22}$$

$$\mathbf{B}_{\mathbf{Y}} = (1-\alpha_t)(\alpha_s)\left(\frac{\mathbf{M}_{1-\alpha_s}}{\gamma_t\Delta t}\right) + (1-\beta_t)\left((1-\beta_s)\frac{\mathbf{K}_{1-\beta_s}^{1-\beta_t}}{\gamma_s\Delta s} + \beta_s\mathbf{F}_{\mathbf{Y}_{\mathbb{j}-1}^{\mathbb{i}}}\right) \tag{23}$$

To linearize the terms $A(\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}})$ and $B(\mathbf{Y}_{\mathbb{j}-1}^{\mathbb{i}})$, the Jacobian of the vector $\mathbf{F}$ needs to be calculated. The Jacobian of $\mathbf{F}$, which is called $\mathbf{F}_{\mathbf{Y}}$, is given in Eq. (24) for the case that external force and external moment, $\mathbf{F}$ and $\mathbf{Q}$ do not depend on $\mathbf{Y}$

$$\mathbf{F}_{\mathbf{Y}} = \begin{bmatrix} -\widetilde{\boldsymbol{\kappa}} & -\widetilde{\mathbf{r}} & \widetilde{\mathbf{v}} & \mathbf{O} \\ \mathbf{O} & -\widetilde{\boldsymbol{\kappa}} & \widetilde{\boldsymbol{\omega}} & \mathbf{O} \\ \mathbf{O} & \widetilde{\boldsymbol{\omega}}\underline{\mathbf{I_m}} - \widetilde{(\underline{\mathbf{I_m}}\boldsymbol{\omega})} & \widetilde{\boldsymbol{\psi}} - \widetilde{\boldsymbol{\kappa}}\dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\kappa}} + \mathbf{J}_{\psi s} & -\widetilde{\mathbf{r}} \\ m\widetilde{\boldsymbol{\omega}} & -m\widetilde{\mathbf{v}} & \widetilde{\mathbf{f}} & -\widetilde{\boldsymbol{\kappa}} \end{bmatrix} \tag{24}$$

The symbol $\widetilde{\mathbf{v}}$ in Eq. (24) represents the skew-symmetric tensor associated with the vector $\mathbf{v}$ generated as follows from its components:

$$\widetilde{\mathbf{v}} = \begin{bmatrix} 0 & -\nu_3 & \nu_2 \\ \nu_3 & 0 & -\nu_1 \\ -\nu_2 & \nu_1 & 0 \end{bmatrix} \tag{25}$$

and the $\mathbf{J}_{\psi s}$ is given below:

$$\mathbf{J}_{\psi s} = -\frac{\partial\left(\dfrac{\partial\boldsymbol{\psi}}{\partial s} - \dfrac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{\kappa}_0}\dfrac{d\boldsymbol{\kappa}_0}{ds}\right)}{\partial\boldsymbol{\kappa}} \tag{26}$$

Equation (20) is linearized around a guessed solution. This means that the terms $\mathbf{A}_{\mathbf{Y}}$, $\mathbf{B}_{\mathbf{Y}}$, and $\hat{\mathbf{K}}(\partial\mathbf{Y}/\partial s)_{\mathbb{j}-1}^{\mathbb{i}}$ are calculated using the guessed solution so that Eq. (20) rendered integrable with respect to $\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}}$ in space

$$\mathbf{A}_{\mathbf{Y}}\mathbf{Y}_{\mathbb{j}}^{\mathbb{i}} + \mathbf{B}_{\mathbf{Y}}\mathbf{Y}_{\mathbb{j}-1}^{\mathbb{i}} = \mathbf{H}^{*} \tag{27}$$

The matrix $\mathbf{H}^{*}$ contains all of the known terms from previous time-step $(\mathbb{i}-1)$ as well as the linearization terms that depend on the guessed solution.

In most scenarios, the boundary condition contains partial information on $\mathbf{Y}$ at one end $(s=0)$ and the rest is known at the other end $(s=L)$. For example, in Fig. 1 the left-hand side of the rod is fixed by a clamp which imposes $\mathbf{v}(0,t) = 0$ and $\boldsymbol{\omega}(0,t) = 0$, while on the right-hand side, the external forces and moments, $\mathbf{f}(L,t)$ and $\mathbf{q}(L,t)$, are prescribed. We use the shooting method at each time-step as explained by Sun et al. [43] to start integration from one end and match the boundary conditions at the other end. Alternatively, an assembled matrix approach can also be used to match the boundary conditions at both the ends simultaneously. Figure 2 shows the algorithm of how shooting method is applied to this problem.

Now, the aim is to implement the numerical solution of the rod model as explained here in a way that user has to provide the initial and boundary conditions for the simulation, the parameters for the numerical scheme (α_t, α_s, β_t, β_s, γ_t, γ_s, Δ_t, and Δ_s), the physical properties of the rod (L, m, $\underline{\mathbf{I_m}}$), and the function $\rightarrow\psi$ that

```
for 𝕚 = 1,2,...
    choose a guessed solution Y^{*𝕚}
    while shooting iteration not converged
        for 𝕛 = 1,2,...
            Calculate A_Y, B_Y, and H*
            Integrate Eq. (27) in space to solve Y_𝕛^𝕚
        end for
        update the guessed solution
        check the shooting method convergence (||Y^{*𝕚} − Y^𝕚||_2 < ε)
    end while
end for
```

Fig. 2 The algorithm of the numerical scheme. The guessed solution for all spatial nodes is shown with $\mathbf{Y}^{*\mathbb{i}}$. At each time-step, the linearized equation is integrated in space. The spatial integration is iterated and is used to update the guessed solution until it converges to the true solution $\mathbf{Y}^{\mathbb{i}}$ bounded by a small tolerance ε.

describes the constitutive law of the rod. The constitutive law is allowed to have any arbitrary functional form and its derivatives need to be calculated for this formulation. In particular, the matrices **K** and **F** in Eqs. (11) and (12) depend on the constitutive law and its derivatives. In addition, the Jacobian of the matrix **F** as represented in Eq. (24) also depends on the derivatives of the constitutive law.

In Secs. 3.2–3.3, we discuss two approaches to incorporate a user-defined constitutive law with an arbitrary functional form in the numerical solution of the rod model. The first approach uses accurate description of the constitutive law and its derivatives while the second approach, which we propose, approximates the constitutive law using polynomial functions. It is important to note that by approximating the matrices **K** and **F**, the true solution of the system **Y** will be directly affected while by approximating the Jacobian $\mathbf{F}_\mathbf{Y}$ merely the number of iterations for guess solution to converge may get affected and the true (converged) solution will remain unchanged. Such approximation of the Jacobian does not necessarily increase the Newton–Raphson iterations. Instead, a slightly inexact Jacobian helps overcome some known singularities that are encountered with exact Jacobian.

It is also important to note that an alternative way to assemble the system of equations in comparison to Eqs. (8) and (9) would be to use a five-variable formulation. By defining the state variable to be $\mathbf{Y} = [\mathbf{v}, \boldsymbol{\omega}, \mathbf{f}, \mathbf{q}, \boldsymbol{\kappa}]$, there would be no substitution of the constitutive law in Eq. (2) and therefore the matrices **K** and **F** will not contain any term that depends on the derivatives of the constative law. Instead of the substitution, the constitutive law will be captured through the fifth row of the system of equations in order to complete the five-variable formulation. However, elaborating on the implementation of our strategy of handling user-defined constitutive law in the five-variable variable formulation is not in the scope of this paper.

3.2 Symbolic Implementation. We will compare the performance of our method with that of the symbolic implementation, which is the most accurate approach. So, here we describe how we used the method of symbolic implementation in getting our benchmark results.

The constitutive law is defined as an input by the user in terms of the scalar symbolic variables $\kappa_1, \kappa_2, \kappa_3$, and s in the form similar to Eq. (6).

The matrices **K**, **F**, and $\mathbf{F}_\mathbf{Y}$ are implemented in terms of the symbols κ_1, κ_2, κ_3, and s and MATLAB substitute command is used to compute the value of these matrices. The algorithm as shown in Fig. 2 requires the calculation of matrices **K**, **F**, and $\mathbf{F}_\mathbf{Y}$ at each space-step. The shooting method iteration will repeat the spatial integration until it converges, which requires reevaluation of all matrices. Therefore, the accuracy of this description comes with a high computational cost due to iterative calculation of the matrices of symbolic type. A comparison of the computational costs among the two methods is presented in Sec. 4.

3.3 Least-Square Polynomial Approximation. The second method that this paper contributes has a similar user interface in which the constitutive law is defined symbolically as expressed in Eq. (6). However, in this approach, the least-square polynomial fitting is used to approximate the functions ψ_i where subscript $i = 1, 2, 3$ as given in the following equation:

$$q_i = \sum_{j=0}^{n_1} \sum_{k=0}^{n_2} \sum_{l=0}^{n_3} \sum_{m=0}^{n_4} p_{jklm} \kappa_1^j \kappa_2^k \kappa_3^l s^m \tag{28}$$

The user can control and choose the order of the polynomial functions so that approximations match well with the true constitutive law or it can be automated with a convergence criteria.

For the cases in which the two axes of bending of the rod and its one axis of twist are decoupled, the polynomial approximation reduces to

$$q_i = \sum_{j=0}^{n_1} \sum_{k=0}^{n_2} p_{jk} \kappa_i^j s^k \tag{29}$$

And finally, if in addition to the decoupling, the rod also has homogeneous elastic properties, the function approximation can be written as

$$q_i = \sum_{j=0}^{n} p_j \kappa_i^j \tag{30}$$

Therefore, in this approach, the matrices **K**, **F**, and $\mathbf{F}_\mathbf{Y}$ are defined in terms of arrays of polynomial coefficients, for example p_j's that are calculated before entering the iteration loops. For instance, the derivative of the function ψ_i with respect to the curvature or twist κ_i, which appears in both **K** and $\mathbf{F}_\mathbf{Y}$, has the following form, for any arbitrary constitutive law:

$$\frac{\partial q_i}{\partial \kappa_i} = \sum_{j=1}^{n} j p_j \kappa_i^{j-1} \tag{31}$$

Thus, there is no need for symbolic description of the constitutive law and it is expected to gain a significant computation advantage by using this method. This will be addressed in Secs. 4 and 5 with more detail.

To compute the derivative of the constitutive law and using these values in the Jacobian and other matrices, one might also use an existing technique of *automatic differentiation* as described in Ref. [35]. Automatic differentiation (AD) uses exact formulas along with floating-point values, instead of expression strings as in symbolic differentiation, and it involves no approximation error as in numerical differentiation using difference quotients. This method is as accurate as symbolic differentiation and very quick in computation also.

But using this technique might not be of great help when compared to the method mentioned above. The constitutive laws in real life will not be too complex which cannot be approximated accurately by Taylor Series. The current method as shown in the previous examples is giving very accurate results when compared with symbolic differentiation. Also, understanding and then implementing Automatic Differentiation in the rod theory will take some extra effort. Introducing this new technique might just make the current code complex and might not be of significant help when studying the nonconvex constitutive laws.

4 Results

In this section, a case study is defined to illustrate and compare the results of the two methods that are previously explained. In order to show the effect of nonlinear constitutive laws on the overall dynamics and mechanics of the rod, we also present the results for a rod with linearized constitutive laws in bending and torsion.

All the simulations are conducted for a rod with an intrinsic curvature $\boldsymbol{\kappa}_0$ that corresponds to a helix and resembles a coil spring. There are many scenarios in which these types of structures are important. For example, recently it is shown [44] how the injection mechanism of the viral genome, which involves compressing and stressing helical proteins can be understood and modeled by the continuum rod model.

4.1 Geometry and Properties of the Rod. As explained in Sec. 2, the vector $\boldsymbol{\kappa}_0$ captures the stress-free configuration of the rod. In this section, a rod with helical shape is simulated by prescribing the $\boldsymbol{\kappa}_0$. The rod has an arc-length of 1 m, the radius and the pitch of the helix are, respectively, called R and P and are

chosen to be both equal to $(\sqrt{2}/10\pi)$. Therefore, the stress-free curvature and twist of the helix are calculated as follows:

$$\boldsymbol{\kappa}_0 = \frac{1}{R^2 + P^2}(R, R, P) \tag{32}$$

The rod has a circular cross section with the radius of 1 cm and its density is $2766.67(\mathrm{kg/m^3})$. The rod is assumed to be composed of homogenous isotropic material with the following constitutive laws:

$$\psi_1 = EI_1 4 \arctan\left(\frac{\kappa_1 - \kappa_{01}}{4}\right) \tag{33}$$

$$\psi_2 = EI_2 4 \arctan\left(\frac{\kappa_2 - \kappa_{02}}{4}\right) \tag{34}$$

$$\psi_3 = GI_3 4 \arctan\left(\frac{\kappa_3 - \kappa_{03}}{4}\right) \tag{35}$$

In Eqs. (33)–(35), I_1 and I_2 represent the second moment of area of the rod's cross section about the axes $\hat{a}_1$ and $\hat{a}_2$ and I_3 is the polar moment of area of the cross section about $\hat{a}_3$. The values of the E and G are chosen to correspond to the Young's modulus and shear modulus of the aluminum in its linear elastic regime where $E = 68.95 \times 10^9$ Pa and $G = 27.58 \times 10^9$ Pa.

4.2 Loading Scenario. The helical rod that is explained in Sec. 4.1 is equivalent to a coil spring. The following boundary conditions are devised to stretch and compress the rod along the axis of the helix as depicted in Fig. 3. The end of the rod at $s = 0$ is clamped by imposing $\mathbf{v}(0, t) = 0$ and $\boldsymbol{\omega}(0, t) = 0$. The other end at $s = L$ is also clamped but slowly moves toward or away from the clamp at $s = 0$ by prescribing $\mathbf{v}(L, t) = h(t)\mathbf{N}$ and $\boldsymbol{\omega}(L, t) = 0$. The vector $\mathbf{N} = (1, 1, 1)$ is along the axis of the helix and the scalar function $h(t)$ is given by the following expression:

$$h(t) = \pm \begin{cases} 5t \text{ m/s}, & \text{if } t \leq 0.1 \\ 0.5 \text{ m/s}, & \text{otherwise} \end{cases} \tag{36}$$

The positive and negative signs on the right-hand side of Eq. (36), respectively, correspond to the extension and compression of the spring.

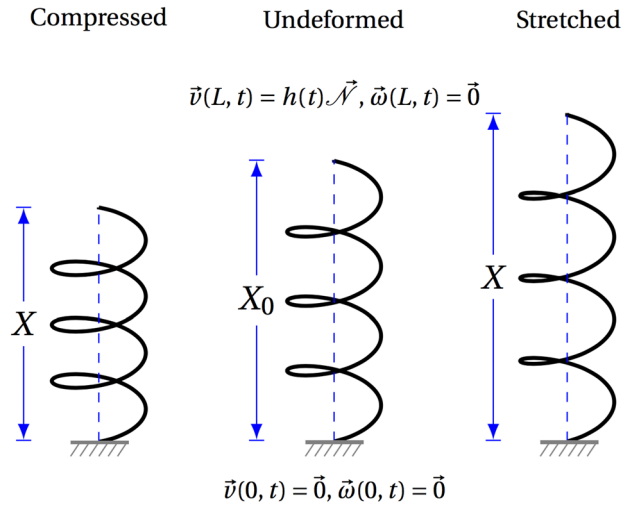

Fig. 3 Loading scenarios of compressing and stretching the helical rod

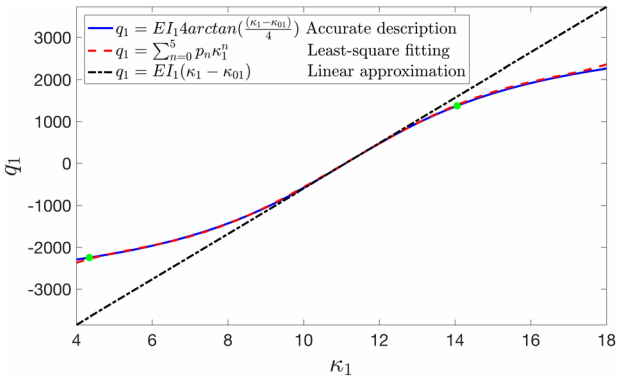

Fig. 4 Bending constitutive law that captures the relationship between the restoring moment q_1 and curvature κ_1. The rod chosen to be isotropic with a circular cross section. Therefore, the bending constitutive law for both planes of bending $q_1 - \kappa_1$, and $q_2 - \kappa_2$ are the same.

In Sec. 4.3, the force–extension relationship of the rod is extracted from the results of the simulations that are described here.

4.3 Force–Extension Relation. In this section, the simulation results of the loading scenario that explained previously are compared when the symbolic implementation of the accurate constitutive laws are used versus the case in which the constitutive laws are approximated by a polynomial function as explained in Sec. 3. The least-square fitting is used to find fifth order polynomial functions that approximate ψ_i's in the interval $0.4\kappa_{0i} < \kappa_i < 1.6\kappa_{0i}$. The order of polynomial and the range of κ_i's can be controlled by the user for the desirable accuracy. Figures 4 and 5 show the accurate constitutive laws and the corresponding polynomial approximations as well as the linearized constitutive laws about the initial curvature and twist. The symmetry of the circular cross section of the rod in addition to the isotropic mechanical properties result into having equal bending constitutive laws or equivalently $I_1 = I_2$ and $\psi_1 = \psi_2$.

Figure 6 is showing the force–extension relationships obtained by the two different methods proposed in Sec. 3 as well as the results using the linearized constitutive law. We can see in this figure that the method explained in Sec. 3.3 is able to closely reproduce the same results as the method of Sec. 3.2, which incorporates accurate description of the constitutive laws. We also observe that the linearized constitutive law is overestimating the hardening behavior of this coil spring in stretching while underestimating its softening behavior in compression.

The method of Sec. 3.2 as mentioned before uses symbolic variables to implement the accurate constitutive laws. This increases

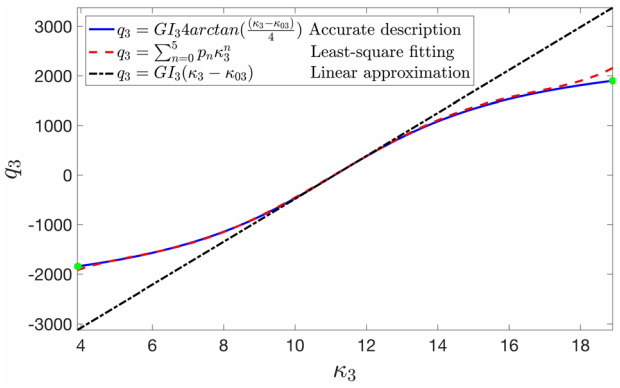

Fig. 5 Twisting constitutive law that captures the relationship between the restoring torque q_3 and torsion κ_3

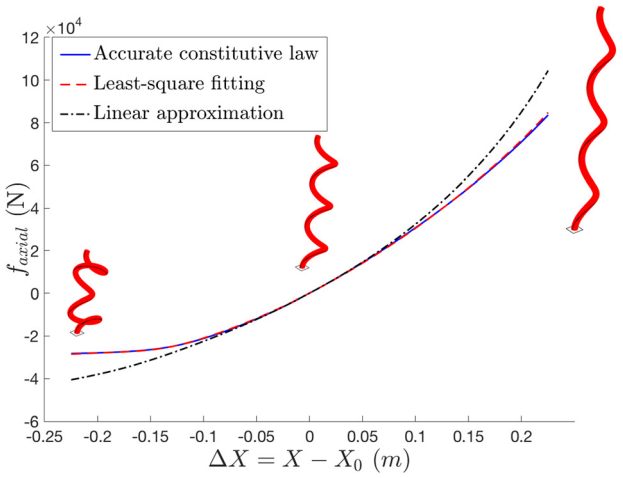

Fig. 6 This diagram shows the relationship between the magnitude of the force along the axis of the helix f_{axial} and the end-to-end distance ΔX

the computational cost drastically in comparison to the method of Sec. 3.3 in which the constitutive laws are approximated by polynomial and there is no need to use the symbolic variables. In particular, the simulations that are presented here take about 4 s per time-step for the method of Sec. 3.3 on a 3.1 GHz Intel Core i7 machine with 16 GB of memory, while it takes 400 s per time-step when the method of Sec. 3.2 is used on the same machine.

4.4 Dynamic Response. In this section, to compare the performance of the three methods on the dynamic of the rod, a loading scenario is defined as follows: The rod is initially at rest. The end of the rod at $s=0$ is clamped similar to Sec. 4.2. The other end of the rod at $s=L$ is stretched along the axis of helix and then released by prescribing the $\mathbf{v}(L,t) = g(t)\mathbf{N}$ where $\mathbf{N} = (1,1,1)$ and

$$g(t) = \begin{cases} 1000t \text{ m/s}, & \text{if } t \le 0.005 \\ 5 \text{ m/s}, & \text{if } 0.005 \le t \le 0.015 \end{cases} \quad (37)$$

For the time $t \ge 0.015$, the end of the rod at $s=0$ is released to freely vibrate by imposing $\mathbf{f}(L,t) = 0$ and $\boldsymbol{\kappa}(L,t) = \boldsymbol{\kappa}_0$.

Figure 7 is showing the results of the three simulations using the accurate description of the constitutive law, the least-square fitting, and the linear approximation of the constitutive law. This figure shows that the polynomial approximation of the constitutive law closely captures the dynamics of the rod when it is compared to the simulation with the accurate description of the constitutive law. Although, the deviations of the results, using the linear approximation of the constitutive law, grow as the time elapses. In these simulations, the rod is surrounded by water and the hydraulic drag which is captured through the external force $\mathbf{F}$ in Eq. (1) is calculated based on the Morison law as explained by Goyal et al. [20]

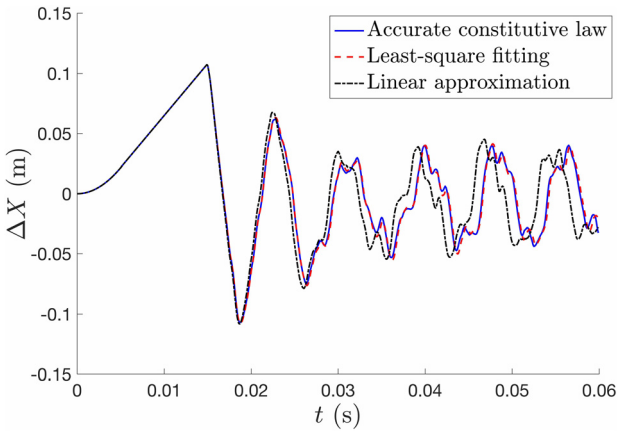

Fig. 7 The vibrations of the rod are calculated via three different descriptions of the constitutive law. The end-to-end distance of the rod ΔX is plotted versus time using the accurate description of the constitutive law, the least-square fitting, and the linear approximation of the constitutive law.

$$\mathbf{F}_{\text{drag}} = -\frac{1}{2}\rho_f d\left(C_n|\mathbf{v}\times\hat{t}|\hat{t}\times(\mathbf{v}\times\hat{t}) + \pi C_t(\mathbf{v}\cdot\hat{t})|\mathbf{v}\times\hat{t}|\hat{t}\right) \quad (38)$$

Here, the diameter of the rod is $d = 2\times10^{-2}$ m, the normal drag coefficient is $C_n = 0.1$, the tangential drag coefficient is $C_t = 0.01$, and ρ_f is the surrounding fluid density, which is chosen to be water. The vector $\hat{t}$ represents the vector $\mathbf{r}$ for an inextensible and unshearable rod.

5 Conclusions

This paper contributes a simple and fast method of implementing any arbitrary user-defined constitutive law in a computational rod model. The method avoids symbolic differentiation by expanding the user-input constitutive law function in a series and using the derivative of the series in the Jacobian. Thus, the method automatically modifies the Jacobian based on the coefficients in the series expansion. The performance of the method is presented for hardening and softening constitutive laws in the force-extension behavior of a helical spring. The effect of nonlinearity in the constitutive law is also emphasized by comparing the results with those for linearized constitutive laws.

References

[1] Simo, J. C., Marsden, J. E., and Krishnaprasad, P. S., 1988, "The Hamiltonian Structure of Nonlinear Elasticity: The Material and Convective Representations of Solids, Rods, and Plates," Arch. Ration. Mech. Anal., **104**(2), pp. 125–183.

[2] Goyal, S., Perkins, N. C., and Lee, C. L., 2005, "Nonlinear Dynamics and Loop Formation in Kirchhoff Rods With Implications to the Mechanics of DNA and Cables," J. Comput. Phys., **209**(1), pp. 371–389.

[3] Hwang, W., 2015, "Biofilament Dynamics: Line-to-Rod-Level Descriptions," *Multiscale Modeling in Biomechanics and Mechanobiology*, S. De, W. Hwang, and E. Kuhl, eds., Springer, London, pp. 63–83.

[4] Neukirch, S., and van der Heijden, G., 2002, "Geometry and Mechanics of Uniform n-Plies: From Engineering Ropes to Biological Filaments," J. Elasticity, **69**(1/3), pp. 41–72.

[5] Klapper, I., 1996, "Biological Applications of the Dynamics of Twisted Elastic Rods," J. Comput. Phys., **125**(2), pp. 325–337.

[6] Lillian, T. D., Goyal, S., Kahn, J. D., Meyhfer, E., and Perkins, N., 2008, "Computational Analysis of Looping of a Large Family of Highly Bent {DNA} by Laci," Biophys. J., **95**(12), pp. 5832–5842.

[7] Hoffman, K. A., 2004, "Methods for Determining Stability in Continuum Elastic-Rod Models of DNA," Philos. Trans. R. Soc. London A: Math., Phys. Eng. Sci., **362**(1820), pp. 1301–1315.

[8] Goyal, S., Perkins, N. C., and Meiners, J. C., 2008, "Resolving the Sequence-Dependent Stiffness of DNA Using Cyclization Experiments and a Computational Rod Model," ASME J. Comput. Nonlinear Dyn., **3**(1), p. 011003.

[9] Sept, D., and MacKintosh, F. C., 2010, "Microtubule Elasticity: Connecting All-Atom Simulations With Continuum Mechanics," Phys. Rev. Lett., **104**(1), p. 018101.

[10] Hawkins, T., Mirigian, M., Yasar, M. S., and Ross, J. L., 2010, "Mechanics of Microtubules," J. Biomech., **43**(1), pp. 23–30.

[11] Hilfinger, A., Chattopadhyay, A. K., and Jülicher, F., 2009, "Nonlinear Dynamics of Cilia and Flagella," Phys. Rev. E, **79**(5), p. 051918.

[12] Qin, Z., Buehler, M. J., and Kreplak, L., 2010, "A Multi-Scale Approach to Understand the Mechanobiology of Intermediate Filaments," J. Biomech., **43**(1), pp. 15–22.

[13] Goldstein, R. E., and Goriely, A., 2006, "Dynamic Buckling of Morphoelastic Filaments," Phys. Rev. E, **74**(1), p. 010901.

[14] Odijk, T., 1998, "Microfibrillar Buckling Within Fibers Under Compression," J. Chem. Phys., **108**(16), pp. 6923–6928.

[15] Kumar, A., Mukherjee, S., Paci, J. T., Chandraseker, K., and Schatz, G. C., 2011, "A Rod Model for Three Dimensional Deformations of Single-Walled Carbon Nanotubes," Int. J. Solids Struct., **48**(20), pp. 2849–2858.

[16] Chen, Y., Dorgan, B. L., McIlroy, D. N., and Eric Aston, D., 2006, "On the Importance of Boundary Conditions on Nanomechanical Bending Behavior and Elastic Modulus Determination of Silver Nanowires," J. Appl. Phys., **100**(10), p. 104301.
[17] Calladine, C. R., Drew, H. R., Luisi, B. F., and Travers, A. A., 2004, *Understanding DNA, the Molecule and How It Works*, Elsevier Academic Press, Amsterdam, The Netherlands.
[18] Goyal, S., Lillian, T., Blumberg, S., Meiners, J. C., Meyhofer, E., and Perkins, N. C., 2007, "Intrinsic Curvature of DNA Influences LacR-Mediated Looping," Biophys. J., **93**(12), pp. 4342–4359.
[19] Goyal, S., and Perkins, N., 2008, "Looping Mechanics of Rods and DNA with Non-Homogeneous and Discontinuous Stiffness," Int. J. Non-Linear Mech., **43**(10), pp. 1121–1129.
[20] Goyal, S., Perkins, N., and Lee, C. L., 2008, "Non-Linear Dynamic Intertwining of Rods With Self-Contact," Int. J. Non-Linear Mech., **43**(1), pp. 65–73.
[21] Cloutier, T. E., and Widom, J., 2004, "Spontaneous Sharp Bending of Double-Stranded DNA," Mol. Cell, **14**(3), pp. 355–362.
[22] Wiggins, P. A., Phillips, R., and Nelson, P. C., 2005, "Exact Theory of Kinkable Elastic Polymers," Phys. Rev. E, **71**(2), p. 021909.
[23] Swati Verma, G. S., and Palanthandalam-Madapusi, H. J., 2012, "Simulation Based Analysis of Constitutive Behavior of Microtubules," Asian Conference on Mechanics of Functional Materials and Structures, New Delhi, India, Dec. 5–8, pp. 679–681.
[24] Fatehiboroujeni, S., and Goyal, S., 2016, "Deriving Mechanical Properties of Microtubules From Molecular Simulations," Biophys. J., **110**(Suppl. 1), p. 129A.
[25] Fosdick, R. L., and James, R. D., 1981, "The Elastica and the Problem of the Pure Bending for a Non-Convex Stored Energy Function," J. Elasticity, **11**(2), pp. 165–186.
[26] Gupta, P., and Kumar, A., 2017, "Effect of Material Nonlinearity on Spatial Buckling of Nanorods and Nanotubes," J. Elasticity, **126**(2), pp. 155–171.
[27] Smith, M. L., and Healey, T. J., 2008, "Predicting the Onset of DNA Supercoiling Using a Nonlinear Hemitropic Elastic Rod," Int. J. Non-Linear Mech., **43**(10), pp. 1020–1028.
[28] Haslach, H. W., Jr., 1985, "Post-Buckling Behavior of Columns With Non-Linear Constitutive Equations," Int. J. Non-Linear Mech., **20**(1), pp. 53–67.
[29] Baczynski, K. K., 2009, "Buckling Instabilities of Semiflexible Filaments in Biological Systems," Ph.D. dissertation, University of Potsdam, Potsdam, Germany.
[30] Avril, S., Bonnet, M., Bretelle, A.-S., Grédiac, M., Hild, F., Ienny, P., Latourte, F., Lemosse, D., Pagano, S., Pagnacco, E., and Pierron, F., 2008, "Overview of Identification Methods of Mechanical Parameters Based on Full-Field Measurements," Exp. Mech., **48**(4), pp. 381–402.
[31] Bonnet, M., and Constantinescu, A., 2005, "Inverse Problems in Elasticity," Inverse Probl., **21**(2), p. R1.
[32] Hinkle, A. R., Goyal, S., and Palanthandalam-Madapusi, H. J., 2009, "An Estimation Method of a Constitutive-Law for the Rod Model of DNA Using Discrete-Structure Simulations," ASME Paper No. DETC2009–87763.
[33] Palanthandalam-Madapusi, H. J., and Goyal, S., 2011, "Robust Estimation of Nonlinear Constitutive Law From Static Equilibrium Data for Modeling the Mechanics of DNA," Automatica, **47**(6), pp. 1175–1182.
[34] Goyal, S., 2006, "A Dynamic Rod Model to Simulate Mechanics of Cables and DNA," Ph.D. dissertation, University of Michigan, Ann Arbor, MI.
[35] Neidinger, R. D., 2010, "Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming," SIAM Rev., **52**(3), pp. 545–563.
[36] Kirchhoff, G., 1859, "Uber Das Gleichgewicht Und Die Bewegung Eines Unendlich Dunnen Elastischen Stabes," J. Reine Angew. Math. (Crelle), **1859**(56), pp. 285–343.
[37] Chung, J., and Hulbert, G. M., 1993, "A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation—The Generalized-Alpha Method," ASME J. Appl. Mech., **60**(2), pp. 371–375.
[38] Bottasso, C. L., and Borri, M., 1998, "Integrating Finite Rotations," Comput. Methods Appl. Mech. Eng., **164**(3–4), pp. 307–331.
[39] Simo, J., and Vu-Quoc, L., 1986, "A Three-Dimensional Finite-Strain Rod Model—Part II: Computational Aspects," Comput. Methods Appl. Mech. Eng., **58**(1), pp. 79–116.
[40] Zhang, Z., Qi Zhaohui, W. Z., and Huiqing, F., 2015, "A Spatial Euler-Bernoulli Beam Element for Rigid-Flexible Coupling Dynamic Analysis of Flexible Structures," J. Shock Vib., **2015**, p. 208127.
[41] Cardona, A., and Geradin, M., 1988, "A Beam Finite Element Non-Linear Theory With Finite Rotations," Int. J. Numer. Methods Eng., **26**(11), pp. 2403–2438.
[42] Fan, W., and Zhu, W. D., 2018, "An Accurate Singularity-Free Geometrically Exact Beam Formulation Using Euler Parameters," Nonlinear Dyn., **91**(2), pp. 1095–1112.
[43] Sun, Y., Leonard, J. W., and Chiou, R. B., 1994, "Simulation of Unsteady Oceanic Cable Deployment by Direct Integration With Suppression," Ocean Eng., **21**(3), pp. 243–256.
[44] Maghsoodi, A., Chatterjee, A., Andricioaei, I., and Perkins, N. C., 2016, "A First Model of the Dynamics of the Bacteriophage t4 Injection Machinery," ASME J. Comput. Nonlinear Dyn., **11**(4), p. 041026.