Elastic Rod Dynamics: Validation of a Real-Time Implicit Approach

John Till *Student Member, IEEE* and D. Caleb Rucker *Member, IEEE*

***Abstract*— The large dynamic deflections of continuum robots, soft robots, and slender elastic objects can be accurately modeled with classical rod theories in nonlinear elasticity. In this paper, we propose a real-time computational approach for solving the partial differential equations of a dynamic Kirchhoff rod. Our approach is based on implicit time discretization of the Kirchhoff equations and subsequent solution of the resulting continuous spatial boundary value problem at each time step. This modular approach can exhibit low numerical damping, handle arbitrarily large time steps, and provide an accurate, high-order representation of the rod shape in steady-state. We experimentally validated the method by capturing footage of a dynamic rod with a high speed camera and comparing this experimental data with simulations using the proposed approach. Soft-real-time performance is achieved, and the relationship between time step and real-time performance is explored in a plot.**

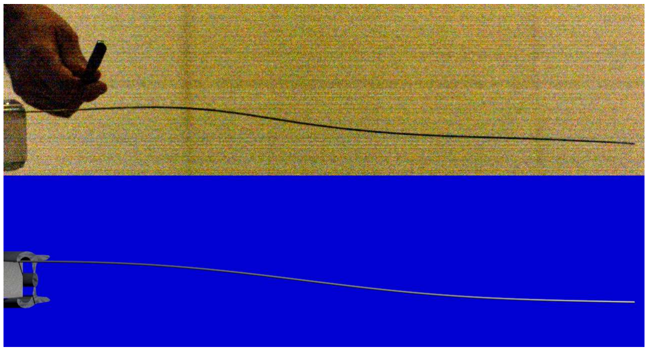

Fig. 1. An impulse point force is applied near the base of a cantilevered rod, resulting in a variety of vibration modes. The scenario is simulated using our proposed method and visualized with Blender.

I. Introduction

Slender elastic objects exhibiting large deflections are becoming increasingly prevalent in robotics, e.g. in the study of soft [1] and continuum [2] robots, robots with flexible links, and in the interaction with objects such as ropes, sutures, needles, and catheters [3]. Flexible robot dynamics have been researched for decades with applications to spacecraft arms, energy-saving lightweight robots, and collaborative robots [4]. While tractable solutions can often be found with Euler-Bernoulli beam theory, this relies on the assumption of small deflections. In applications with large deflections, such as those listed above, classical rod theories in nonlinear elasticity (Kirchhoff and Cosserat) are needed, and these are orders of magnitude more efficient than full 3D elasticity when using a finite-element method [5].

Some desirable characteristics of a modeling and computational approach for the dynamics of slender elastic objects undergoing large deflections are as follows:

- Numerical consistency with the continuous theory
- Real-time computation for simulation and control
- Good scalability with respect to spatial resolution
- Stability under large time steps
- High order of accuracy in steady-state / static cases
- Low numerical damping

Our purpose in this paper is to present and validate a computational method for elastic rod dynamics that satisfies the above criteria. Our proposed approach directly solves the nonlinear, hyperbolic, partial-differential equations (PDE) for 3D, large-deflection Kirchhoff rods using implicit discretization of time derivatives and numerical integration in the spatial dimension.

A. Related Work and Contributions

The continuous equations of motion for spatial rods were discovered in prior centuries [6], but the challenge to formulate and numerically solve discrete systems of equations consistent with the continuous physics has been the subject of recent works. Methods for simulating rod deformation range from non-physical techniques that achieve an aesthetic goal to accurate mechanics-based models [7]. In this work, we are concerned with the latter but also with computational speed and the other characteristics outlined previously.

Accurate models can be based on finite element methods [3], [8], finite differences [9]–[11], and differential algebraic equation solvers [12]. The most popular rod dynamics implementations first discretize the rod geometry in the spatial dimension and then derive equations of motion defining the accelerations of the generalized coordinates, which are typically integrated numerically with an explicit time-marching method. Obtaining the solution involves enforcing inextensibility and unshearability constraints (via minimal coordinates [13], [14], projection [15], or Lagrange multipliers with post-stabilization [16]), or explicitly modeling the stiff dynamics of shear and extension [11], [17] (equivalent to enforcement via a penalty method [3], [18]). Many explicit methods exhibit good computational efficiency and can run in real time. However, they require work-intensive derivations dependent on particular choices of model effects/assumptions

This material is based upon work supported by the National Science Foundation under CMMI-1427122 as part of the NSF/NASA/NIH/USDA/DOD National Robotics Initiative and under NSF CAREER Award IIS-1652588. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

J. Till and D. C. Rucker are with the University of Tennessee, Knoxville, TN 37996, USA (e-mail: JTill@vols.utk.edu, caleb.rucker@utk.edu).

and spatial discretization, which has a cost in terms of implementation effort and modularity. For instance, [15] mentions the use of a higher-order spatial discretization as future work, and this would involve re-deriving the equations of motion. Furthermore, for all methods using explicit time integration, the maximum time step is limited by the Courant-Friedrichs-Lewy stability condition, which is especially limiting if the shear and extension are modeled or enforced via a penalty method.

Our work in this paper does not follow the common pattern of spatial discretization followed by explicit time integration but is instead inspired by a relatively unexplored method suggested by [19] for modeling the planar motion of fly-fishing lines, and subsequently developed further by Lan and Lee in [20], [21] for planar compliant mechanisms. Starting with the continuous PDE, we discretize any time derivatives using a chosen implicit differentiation formula. This creates a continuous ordinary boundary value problem in the spatial dimension that can be solved to obtain the rod state at each time step. This implicit temporal-discretization approach provides consistency, stability under large time steps, and the potential for high-order accuracy, scalability, and efficiency through our proposed use of standard numerical integration routines and a stable variant of the shooting method. Any shear and extension behavior (including inextensibility) is automatically satisfied by the spatial integration of the strains. Current efforts using this type of approach have been limited to planar dynamics, and computation of solutions at interactive rates has not been demonstrated. We extend the basic approach to the 3D spatial case, explore the application of higher-order numerical schemes in both the space and time dimensions, and demonstrate real-time performance. In addition, we fill a gap in the experimental literature by including a highly dynamic experiment with multiple modes of vibration. A principle advantage of this method is that only simple changes are required to explore various model assumptions (neglecting shear, viscosity, etc.), external force terms such as tendon actuation [22], and discretization schemes in space or time, in contrast to methods which symbolically solve for acceleration terms. We believe that this modularity is worth the slight increase in run time, especially given the potential for greater accuracy by using high-order schemes.

TABLE I

DEFINITIONS

Variable	Units	Definition
s	m	Reference arclength - a dimension of the PDE
t	s	Time - a dimension of the PDE
$\boldsymbol{p}$	m	Position in Cartesian coordinates
$\boldsymbol{R}$	none	Rotation matrix for the material orientation; the third column aligns with the rod tangent
$\boldsymbol{h}$	none	Quaternion for the material orientation
$\boldsymbol{n}$	N	Internal force expressed in the global frame
$\boldsymbol{m}$	Nm	Internal moment expressed in the global frame
$\boldsymbol{f}$	N/m	Distributed force in the global frame
$\boldsymbol{l}$	Nm/m	Distributed moment in the global frame
$\boldsymbol{v}$	none	Rate of change of position with respect to arclength in the local frame
$\boldsymbol{u}$	1/m	Curvature in the local frame
$\boldsymbol{q}$	m/s	Velocity in the local frame
$\boldsymbol{\omega}$	1/s	Angular velocity in the local frame
A	m^2	Cross-sectional area
ρ	kg/m^3	Material density
$\boldsymbol{J}$	m^4	Second mass moments of inertia
$\boldsymbol{u}^*$	1/m	Local curvature when $\boldsymbol{m} = \boldsymbol{u}_t = \boldsymbol{0}$. For a straight rod $\boldsymbol{u}^* = \boldsymbol{0}$.
$\boldsymbol{K}$	Nm^2	Stiffness matrix for bending and twisting; $\boldsymbol{K} = \begin{bmatrix} EI_{xx} & 0 & 0 \\ 0 & EI_{yy} & 0 \\ 0 & 0 & GI_{zz} \end{bmatrix}$
E	Pa	Young's modulus
G	Pa	Shear modulus
$\boldsymbol{I}$	m^4	Cross-sectional area moment of inertia
$\boldsymbol{B}$	Nm^2/s	Damping matrix for bending and twisting
$\boldsymbol{C}$	kg/m^2	Square law drag coefficient matrix
$\boldsymbol{g}$	m/s^2	Gravitational acceleration vector
$\boldsymbol{e}_3$	none	Unit vector; $\boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$
α	none	Coefficient of the BDF-α method

II. Partial Differential Equations

The rod is approximated as a one-dimensional continuum so that the state variables are parameterized by arclength along the rod centerline s and by the time t. We rely on Equations (23, 25-27) in [22] as the initial statement of the PDE. These equations are consistent with the derivation in chapter eight of Antman's classic text [6], but the notation differs in that the position is expressed as $\boldsymbol{p}$ instead of $\boldsymbol{r}$, the velocity is expressed as $\boldsymbol{q}$ instead of $\boldsymbol{p}$, and we introduce the rotation matrix $\boldsymbol{R}$ to express internal loads in the global frame whereas Antman's formulation is entirely in the local coordinate frame. The dynamics of an elastic rod are

$$\begin{aligned} \boldsymbol{p}_s &= \boldsymbol{R}\boldsymbol{v}, \qquad \boldsymbol{p}_t = \boldsymbol{R}\boldsymbol{q} \\ \boldsymbol{R}_s &= \boldsymbol{R}\widehat{\boldsymbol{u}}, \qquad \boldsymbol{R}_t = \boldsymbol{R}\widehat{\boldsymbol{\omega}} \\ \boldsymbol{n}_s &= \rho A \boldsymbol{R}\left(\widehat{\boldsymbol{\omega}}\boldsymbol{q} + \boldsymbol{q}_t\right) - \boldsymbol{f} \\ \boldsymbol{m}_s &= \partial_t(\boldsymbol{R}\rho\boldsymbol{J}\boldsymbol{\omega}) - \widehat{\boldsymbol{p}}_s\boldsymbol{n} - \boldsymbol{l} \\ \boldsymbol{q}_s &= \boldsymbol{v}_t - \widehat{\boldsymbol{u}}\boldsymbol{q} + \widehat{\boldsymbol{\omega}}\boldsymbol{v} \\ \boldsymbol{\omega}_s &= \boldsymbol{u}_t - \widehat{\boldsymbol{u}}\boldsymbol{\omega}. \end{aligned} \tag{1}$$

The symbols are described in Table I, and a subscript denotes a partial derivative with respect to the subscript. The "hat" operator $\widehat{\cdot}$ forms a skew-symmetric matrix from a vector, for example

$$\widehat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}.$$

It is reasonable to assume an inextensible and unshearable rod so that $\boldsymbol{v} = \boldsymbol{e}_3$ and $\boldsymbol{v}_t = \boldsymbol{0}$ (i.e. use Kirchhoff's rod theory rather than the Cosserat brothers' more general theory). The differential equations in (1) do not provide a unique solution; an appropriate material constitutive law must relate $\boldsymbol{m}$ to $\boldsymbol{u}$. We use a linear elastic law with Kelvin-Voigt type viscous damping as described in [23], that is

$$\boldsymbol{m} = \boldsymbol{R}[\boldsymbol{K}(\boldsymbol{u} - \boldsymbol{u}^*) + \boldsymbol{B}\boldsymbol{u}_t]. \tag{2}$$

We expand the distributed force term $\boldsymbol{f}$ to explicitly consider terms for weight and square-law-drag air resistance so that

$$\boldsymbol{f} = -\boldsymbol{R}\boldsymbol{C}\boldsymbol{q} \odot |\boldsymbol{q}| + \rho A \boldsymbol{g} + \bar{\boldsymbol{f}},$$

where the Hadamard product $\odot$ performs element-wise multiplication so that $\boldsymbol{q} \odot |\boldsymbol{q}| = \begin{bmatrix} q_1^2 \operatorname{sgn}(q_1) & q_2^2 \operatorname{sgn}(q_2) & q_3^2 \operatorname{sgn}(q_3) \end{bmatrix}^T$. $\bar{\boldsymbol{f}}$ contains forces not explicitly considered such as contact. $\bar{\boldsymbol{f}}$ and $\boldsymbol{l}$ are zero for the simulations and experiments but are kept in the development for generality.

For very slender rods, the dynamics are dominated by the mass distribution along the length, and the cross-section rotational inertia matrix $\boldsymbol{J}$ has a negligible effect on the dynamic behavior [24], thus we can let $\partial_t(\boldsymbol{R}\rho\boldsymbol{J}\boldsymbol{\omega}) = \boldsymbol{0}$. Methods based on discrete differential geometry [15] also typically neglect cross sectional rotational inertia.

Finally, using (2) as the constitutive law, the partial derivatives with respect to arc length are

$$\begin{aligned}
\boldsymbol{p}_s &= \boldsymbol{R}\boldsymbol{e}_3 \\
\boldsymbol{R}_s &= \boldsymbol{R}\widehat{\boldsymbol{u}} \\
\boldsymbol{n}_s &= \rho A \boldsymbol{R}\left(\widehat{\boldsymbol{\omega}}\boldsymbol{q} + \boldsymbol{q}_t\right) + \boldsymbol{R}\boldsymbol{C}\boldsymbol{q} \odot |\boldsymbol{q}| - \rho A \boldsymbol{g} - \bar{f} \\
\boldsymbol{u}_s &= \boldsymbol{u}_s^* - \boldsymbol{K}^{-1}[\widehat{\boldsymbol{u}}\boldsymbol{K}(\boldsymbol{u} - \boldsymbol{u}^*) + \widehat{\boldsymbol{e}}_3\boldsymbol{R}^T\boldsymbol{n} \\
&\quad + \boldsymbol{R}^T\boldsymbol{l} + \widehat{\boldsymbol{u}}\boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{B}\boldsymbol{u}_{st}] \\
\boldsymbol{q}_s &= -\widehat{\boldsymbol{u}}\boldsymbol{q} + \widehat{\boldsymbol{\omega}}\boldsymbol{e}_3 \\
\boldsymbol{\omega}_s &= \boldsymbol{u}_t - \widehat{\boldsymbol{u}}\boldsymbol{\omega}.
\end{aligned} \tag{3}$$

A. Quaternions

Numerically integrating the equation for $\boldsymbol{R}_s$ along the rod length can produce a "rotation" matrix that is not perfectly orthonormal due to truncation error. In many cases this effect can reasonably be ignored, but for completeness we introduce a quaternion representation of the rod orientation using the element ordering $\boldsymbol{h} = h_1 + h_2 i + h_3 j + h_4 k$. The differential equation for $\boldsymbol{R}_s$ is replaced by

$$\boldsymbol{h}_s = F_1(\boldsymbol{h}, \boldsymbol{u}) = \frac{1}{2}\begin{bmatrix} 0 & -u_1 & -u_2 & -u_3 \\ u_1 & 0 & u_3 & -u_2 \\ u_2 & -u_3 & 0 & u_1 \\ u_3 & u_2 & -u_1 & 0 \end{bmatrix}\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}.$$

The other equations relying on the rotation matrix are left unchanged; the matrix is found by

$$\begin{aligned}
&\boldsymbol{R} = F_2(\boldsymbol{h}) = \\
&\boldsymbol{I} + \frac{2}{\boldsymbol{h}^T\boldsymbol{h}}\begin{bmatrix} -h_3^2 - h_4^2 & h_2h_3 - h_4h_1 & h_2h_4 + h_3h_1 \\ h_2h_3 + h_4h_1 & -h_2^2 - h_4^2 & h_3h_4 - h_2h_1 \\ h_2h_4 - h_3h_1 & h_3h_4 + h_2h_1 & -h_2^2 - h_3^2 \end{bmatrix}.
\end{aligned}$$

Note that in constructing the rotation matrix, the quaternion magnitude is not required to be unity, and the differential equation for $\boldsymbol{h}_s$ is correct regardless of the magnitude of $\boldsymbol{h}$. Additionally, this formulation requires no square root or norm operations.

B. Semi-discretization in Time

The system in (3) would be an ODE in s except for the presence of time derivatives $\boldsymbol{u}_t$, $\boldsymbol{q}_t$, and $\boldsymbol{u}_{st}$ on the right-hand side. An ODE can be obtained by discretizing these derivatives in the time variable. We denote the time index with a left superscript, e.g. for a general state variable $\boldsymbol{y}$, $\boldsymbol{y}(t_i, s) = {}^{(i)}\boldsymbol{y}(s)$. Any general backward finite difference for a first derivative can be written in the form

$$\begin{aligned}
{}^{(i)}\boldsymbol{y}_t &= c_0\, {}^{(i)}\boldsymbol{y} + \sum_{k=1}^{\infty}\left[c_k\, {}^{(i-k)}\boldsymbol{y} + d_k\, {}^{(i-k)}\boldsymbol{y}_t\right] \\
&:= c_0\, {}^{(i)}\boldsymbol{y} + {}^{(i)}\boldsymbol{Y}.
\end{aligned}$$

The only term in ${}^{(i)}\boldsymbol{y}_t$ corresponding to time t_i is $c_0\, {}^{(i)}\boldsymbol{y}$. ${}^{(i)}\boldsymbol{Y}$ is defined to encapsulate all terms relying on the prior history of the rod.

The time-discretized equation for $\boldsymbol{u}_s$ is obtained by setting $\boldsymbol{u}_t = c_0\, {}^{(i)}\boldsymbol{u} + {}^{(i)}\boldsymbol{U}$ and $\boldsymbol{u}_{st} = c_0\, {}^{(i)}\boldsymbol{u}_s + {}^{(i)}\boldsymbol{U}_s$, which yields

$$\begin{aligned}
{}^{(i)}\boldsymbol{u}_s =& \boldsymbol{u}_s^* - \boldsymbol{K}^{-1}[{}^{(i)}\widehat{\boldsymbol{u}}\boldsymbol{K}({}^{(i)}\boldsymbol{u} - \boldsymbol{u}^*) + {}^{(i)}\boldsymbol{R}^T\boldsymbol{l} \\
&+ \widehat{\boldsymbol{e}}_3{}^{(i)}\boldsymbol{R}^{T(i)}\boldsymbol{n} + {}^{(i)}\widehat{\boldsymbol{u}}\boldsymbol{B}(c_0\, {}^{(i)}\boldsymbol{u} + {}^{(i)}\boldsymbol{U}) \\
&+ \boldsymbol{B}(c_0\, {}^{(i)}\boldsymbol{u}_s + {}^{(i)}\boldsymbol{U}_s)].
\end{aligned}$$

This can be solved so that $\boldsymbol{u}_s$ is not on the right-hand side:

$$\begin{aligned}
{}^{(i)}\boldsymbol{u}_s =& (\boldsymbol{K} + c_0\boldsymbol{B})^{-1}\{\boldsymbol{K}\boldsymbol{u}_s^* - [{}^{(i)}\widehat{\boldsymbol{u}}\boldsymbol{K}({}^{(i)}\boldsymbol{u} - \boldsymbol{u}^*) \\
&+ {}^{(i)}\boldsymbol{R}^T\boldsymbol{l} + \widehat{\boldsymbol{e}}_3{}^{(i)}\boldsymbol{R}^{T(i)}\boldsymbol{n} + {}^{(i)}\widehat{\boldsymbol{u}}\boldsymbol{B}(c_0\, {}^{(i)}\boldsymbol{u} + {}^{(i)}\boldsymbol{U}) \\
&+ \boldsymbol{B}^{(i)}\boldsymbol{U}_s]\}.
\end{aligned}$$

The other equations require no special effort to discretize, so that the time-discretized ODEs in s are

$$\begin{aligned}
\boldsymbol{p}_s &= \boldsymbol{R}\boldsymbol{e}_3 \\
\boldsymbol{h}_s &= F_1(\boldsymbol{h}, \boldsymbol{u}) \\
\boldsymbol{n}_s &= \rho A \boldsymbol{R}\left(\widehat{\boldsymbol{\omega}}\boldsymbol{q} + c_0\boldsymbol{q} + \boldsymbol{Q}\right) + \boldsymbol{R}\boldsymbol{C}\boldsymbol{q} \odot |\boldsymbol{q}| - \rho A \boldsymbol{g} - \bar{f} \\
\boldsymbol{u}_s &= (\boldsymbol{K} + c_0\boldsymbol{B})^{-1}\{\boldsymbol{K}\boldsymbol{u}_s^* - [\widehat{\boldsymbol{u}}\boldsymbol{K}(\boldsymbol{u} - \boldsymbol{u}^*) + \boldsymbol{R}^T\boldsymbol{l} \\
&\quad + \widehat{\boldsymbol{e}}_3\boldsymbol{R}^T\boldsymbol{n} + \widehat{\boldsymbol{u}}\boldsymbol{B}(c_0\boldsymbol{u} + \boldsymbol{U}) + \boldsymbol{B}\boldsymbol{U}_s]\} \\
\boldsymbol{q}_s &= -\widehat{\boldsymbol{u}}\boldsymbol{q} + \widehat{\boldsymbol{\omega}}\boldsymbol{e}_3 \\
\boldsymbol{\omega}_s &= c_0\boldsymbol{u} + \boldsymbol{U} - \widehat{\boldsymbol{u}}\boldsymbol{\omega} \\
\boldsymbol{R} &= F_2(\boldsymbol{h}),
\end{aligned} \tag{4}$$

where the superscript (i) has been omitted. This discretization encompasses many choices of schemes such as backward Euler, the backward differentiation formulas of order one through six, or the trapezoidal method.

To implement the method in simulation, we use the BDF-α method [25], which is $O(dt^2)$ accurate. This is described by

$$\begin{aligned}
{}^{(i)}\boldsymbol{y}_t &= c_0\, {}^{(i)}\boldsymbol{y} + c_1\, {}^{(i-1)}\boldsymbol{y} + c_2\, {}^{(i-2)}\boldsymbol{y} + d_1\, {}^{(i-1)}\boldsymbol{y}_t \\
&:= c_0\, {}^{(i)}\boldsymbol{y} + {}^{(i)}\boldsymbol{Y},
\end{aligned}$$

where

$$\begin{aligned}
c_0 &= (1.5 + \alpha)/[dt(1 + \alpha)] \\
c_1 &= (-2 - 2\alpha)/[dt(1 + \alpha)] \\
c_2 &= (0.5 + \alpha)/[dt(1 + \alpha)] \\
d_1 &= \alpha/(1 + \alpha).
\end{aligned}$$

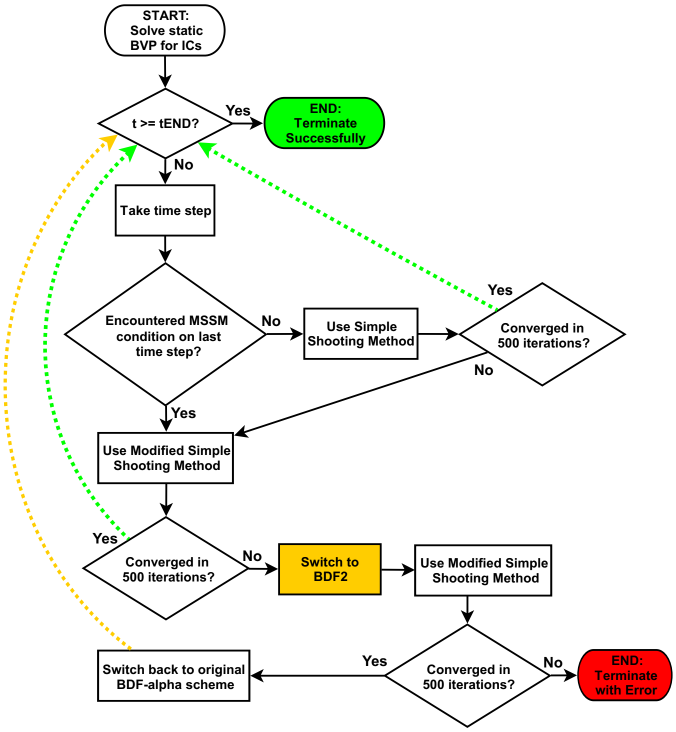

Fig. 2. The high level simulation strategy is illustrated in a flowchart. If the problem is well posed, it will be solved with a simple shooting method. In the event of numerical issues, the simulation attempts a solution using the modified simple shooting method and second-order backwards differentation formula. Although the modified shooting method raises the computational cost, it does not affect the final solution. Switching to BDF2 will alter the solution by contributing more numerical damping, but the time discretization is still $O(dt^2)$ accurate.

The variable α is a parameter that may be assigned. The trapezoidal method is obtained for $\alpha = -0.5$, and the second-order backward differentiation formula BDF2 is obtained for $\alpha = 0$. Using this approximation, the solution to the ODE in s at t_i is dependent on the solutions to previous ODEs at times t_{i-1} and t_{i-2}.

C. *ODE Solution with the Shooting Method*

When considering a specific problem, the ODE system described by (4) is accompanied by a set of boundary conditions. For instance, a cantilevered rod has known values of $\boldsymbol{p}(t_i, 0) = \boldsymbol{p}_0$, $\boldsymbol{R}(t_i, 0) = \boldsymbol{R}_0$, and $\boldsymbol{q}(t_i, 0) = \boldsymbol{\omega}(t_i, 0) = \mathbf{0}$. If the rod has a length L and no forces at the free end, the distal constraints are $\boldsymbol{n}(t_i, L) = \boldsymbol{m}(t_i, L) = \mathbf{0}$.

Such BVPs can be solved by a shooting method that iteratively guesses the unknown initial values (e.g. $\boldsymbol{n}(t_i, 0)$ and $\boldsymbol{u}(t_i, 0)$ for the cantilever problem) and evaluates the boundary conditions at the other end after numerical integration. There is freedom in the choice of numerical integration method; for the simulations herein, we will consider and compare both Euler's first-order method and the classic fourth-order Runge-Kutta integration (RK4), but the modularity of the approach makes it trivial to explore other spatial integration schemes. The RK4 method requires interpolation to estimate ${}^{(i)}\boldsymbol{Y}(s_{j+0.5})$ since it lies between the grid points of previous ODE solutions. For the purposes of this paper, we use linear interpolation, although the use of a Hermite interpolant would be an interesting topic for future work. Using a fourth-order method in space should provide superior error scaling and computational efficiency, and it will produce highly accurate steady-state solutions.

The guessed values are corrected by nonlinear optimization. This aspect of the shooting method is implemented by a Levenberg-Marquardt algorithm with an adaptive damping coefficient as described in [26]. The existence of multiple solutions to the BVP is related to the time step. An infinite time step recovers the static ODE, which may have multiple solutions [27] due to the geometric nonlinearities associated with large deflection, and large time steps may share this difficulty. However, as the time step approaches zero, the discretization approaches the continuous PDE for which there is only one solution except at specific times where dynamic bifurcations exist (e.g. the exact moment of buckling) and there are multiple valid solution paths for the future time evolution. While the issue of multiple solution branches and dynamic bifurcations is inherent to the physics, our approach stays on its current solution branch by keeping the time step relatively low and using the solution at the previous time step as the initial guess at the current time.

Often it is necessary to solve a static BVP to find the initial conditions for the dynamic BVP. The static ODEs are

$$\begin{aligned}
\boldsymbol{p}_s &= \boldsymbol{R}\boldsymbol{e}_3 \\
\boldsymbol{h}_s &= F_1(\boldsymbol{h}, \boldsymbol{u}) \\
\boldsymbol{n}_s &= -\rho A \boldsymbol{g} - \bar{\boldsymbol{f}} \\
\boldsymbol{u}_s &= \boldsymbol{u}_s^* - \boldsymbol{K}^{-1}[\widehat{\boldsymbol{u}}\boldsymbol{K}(\boldsymbol{u} - \boldsymbol{u}^*) + \widehat{\boldsymbol{e}}_3 \boldsymbol{R}^T \boldsymbol{n} + \boldsymbol{R}^T \boldsymbol{l}] \\
\boldsymbol{R} &= F_2(\boldsymbol{h}),
\end{aligned}$$

and of course $\boldsymbol{q}$ and $\boldsymbol{\omega}$ are initialized to zero.

D. *Improving Convergence and Stability at Small Time Steps*

Although arbitrarily large time steps are possible by using an asymptotically-stable finite difference scheme, the IVP is actually poorly conditioned at small time steps. Under normal operating conditions, the rod state is continuous in both time and space so that

$$\lim_{dt \to 0} {}^{(i)}\boldsymbol{u} - {}^{(i-1)}\boldsymbol{u} = \mathbf{0}.$$

Thus, performing the calculation ${}^{(i)}\boldsymbol{u} - {}^{(i-1)}\boldsymbol{u}$ for small time steps is a "catastrophic cancellation," where subtracting two nearly equal numbers causes a loss of significant digits [28]. For instance, subtracting two floating-point numbers with 16 digits of precision that are the same in the first 10 digits will yield a resultant having only 6 digits of precision. This is an inherent property of the implicit time-discretization, and this effect seems to be exacerbated by the numerical integration. While the breakdown of the time-discretized scheme is inevitable with decreasing time steps, the simulations of a small steel rod described later are able to run with a time step as small as 2-5ms using just a simple shooting method. However, we emphasize that robotics applications are usually more concerned with real-time computations and

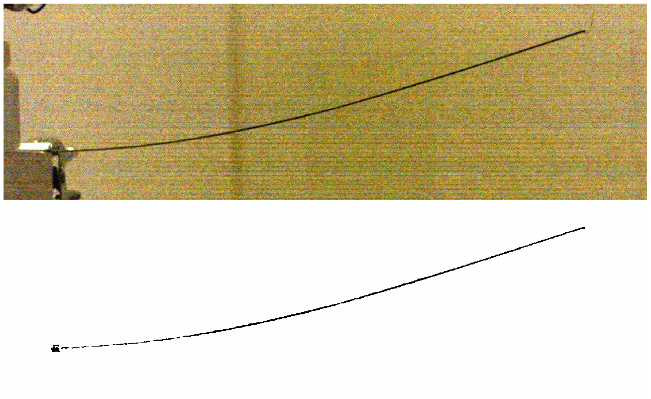

Fig. 3. The experimental rod shape was quantified by obtaining binary data based on darkness. The rightmost pixel is taken as the tip position, specifically the top rightmost pixel when multiple rightmost pixels exist.

good long-term accuracy. Thus, an implicit method is an appropriate choice, and the tradeoff of a minimum time step is acceptable.

If small time step resolution of high-frequency modes is desired, there are modifications of the shooting method that can improve performance. The modified simple shooting method is an excellent tool for solving badly-posed shooting problems [29]. For the simulations presented here, the modified shooting method is used when the simple shooting method fails to converge within five hundred iterations. If the modified shooting method is used and makes a correction midway through the integration, it will also be used for the next time step. If modified shooting is used and no correction is necessary, simple shooting will be used on the next time step. This allows us to solve well-posed problems without a performance penalty while still tackling slightly ill-posed shooting problems.

From our experience, the choice of time derivative discretization affects the difficulty of the problem; for example, it is easier to achieve convergence with BDF2 than with the trapezoidal method, although this results in more numerical damping. Thus, if the modified shooting method fails to converge, it is attempted again with $\alpha = 0$ for that time step. The error handling strategy is described in Figure 2.

Although we have experimented with high precision floating-point numbers and found the convergence to improve slightly, we ultimately decided to use the current standard of 64-bit precision due to the adequacy of possible time steps and the computational cost of higher precision.

III. Validation

The proposed method was implemented in C++. The only external dependency was the matrix library "eigen" [30]. We experimentally validated the proposed model by comparing simulation results to high-speed footage of a cantilevered rod clamped to a table. Our simulation implements the full 3-dimensional model described in the previous sections, but the experimental data was taken from planar cases for simplicity. The rod was spring steel with a 1.42mm diameter. There were two scenarios. First, a 20g weight was hung by a string at the tip of a rod with a cantilevered length of 0.408m, and after equilibrium was reached, the string was cut. Second, the cantilevered length was increased to 0.517m to obtain larger vibrations, and the rod was hit with a rigid object near its base to excite high-frequency vibration modes. The BDF-α coefficient was $\alpha = -0.48$ for all simulations, which is close to the trapezoidal method and thus exhibits very little numerical damping.

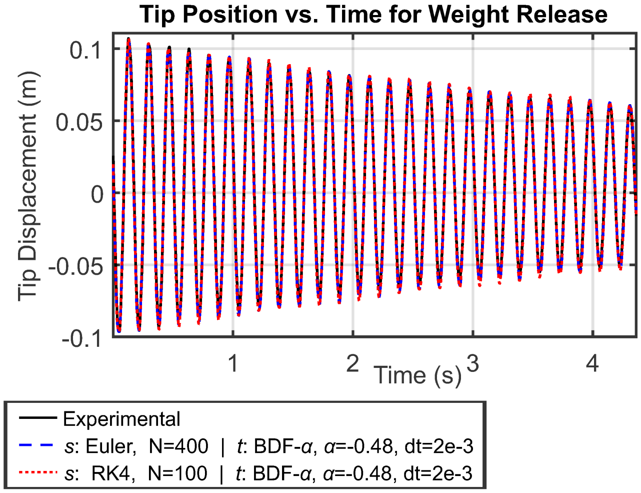

Fig. 4. A weight was attached to the free end of a cantilevered rod by a string. After the weighted rod reached equilibrium, the string was cut. This scenario was simulated, and the simulation parameters were calibrated so that the simulation response matches the experimental response. These calibrated values were used while validating the other impulse experiment.

The camera was placed about three meters from the rod with the viewing plane parallel to the rod's plane of motion. The camera recorded a frame every millisecond. The rod was darker than the background so that the experimental rod position could be easily extracted by comparing pixel brightness values, as shown in Figure 3.

A. Weight Release

The weight release trial was used to calibrate the rod parameters EI, ρ, and C. This calibration is implemented in MATLAB using the Levenberg-Marquardt algorithm to minimize error between experimental data and model prediction. The weight release response is very nearly a decaying sine wave, as shown in Figure 4. The calibration objective function was evaluated by running the simulation for a set of parameters and evaluating the characteristics of the simulated response versus the experimental data. The magnitude of the first peak, magnitude of the first valley, magnitude of the final peak, and frequency are compared and combined to form the objective function residual. MATLAB's "findpeaks" command can easily detect peaks in the smooth simulation data. The experimental data has some noise, but since the experimental response only needs to be analyzed once, this was done manually. The calibrated values are shown in Table II. For steel, ρ is typically around 7800 kg/m^3. With an

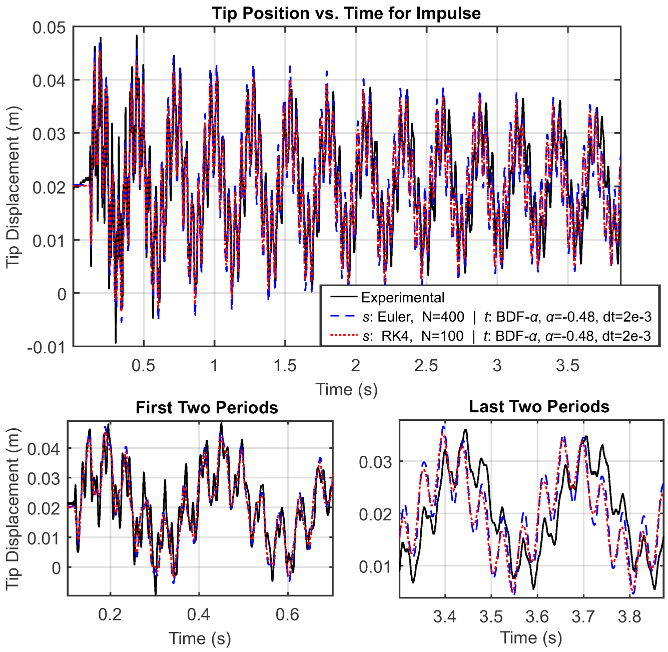

Fig. 5. An impulse was applied near the base of a cantilevered rod. The experimental impulse response is compared to two simulations that used the proposed method. The multimedia attachment for this paper includes a video of this trial.

assumed Young's modulus of 200GPa, the 1.42mm diameter rod would have a bending stiffness EI of 0.03992 Nm^2. Thus, the calibrated values are within reason.

TABLE II
CALIBRATED PARAMETERS

Parameter	Euler, N=400, dt=2e-3	RK4, N=100, dt=2e-3
EI (Nm^2)	0.03823	0.03803
ρ (kg/m^3)	7640	7621
C (kg/m^2)	0.003507	0.003556

B. Impulse Near Base

After calibration of the model parameters using the weight release dataset, we evaluated the model prediction versus data taken from the impulse response experiment. The impulse point force was modeled as a hat function in time with

$$F(t) = \begin{cases} M\frac{t}{0.5d}, & t < 0.5d \\ M(2 - \frac{t}{0.5d}), & 0.5d \leq t \leq d \\ 0, & t > d. \end{cases}$$

Appropriate values for the impulse's peak magnitude and duration were found: $M = 5$N and $d = 0.016$s. The impulse point force is included in the simulation by performing piecewise integration of the ODEs in space and applying the point force at the transition. The impulse response is shown in Figure 5. A simulation video is included with this paper submission, and a still frame is shown in Figure 1.

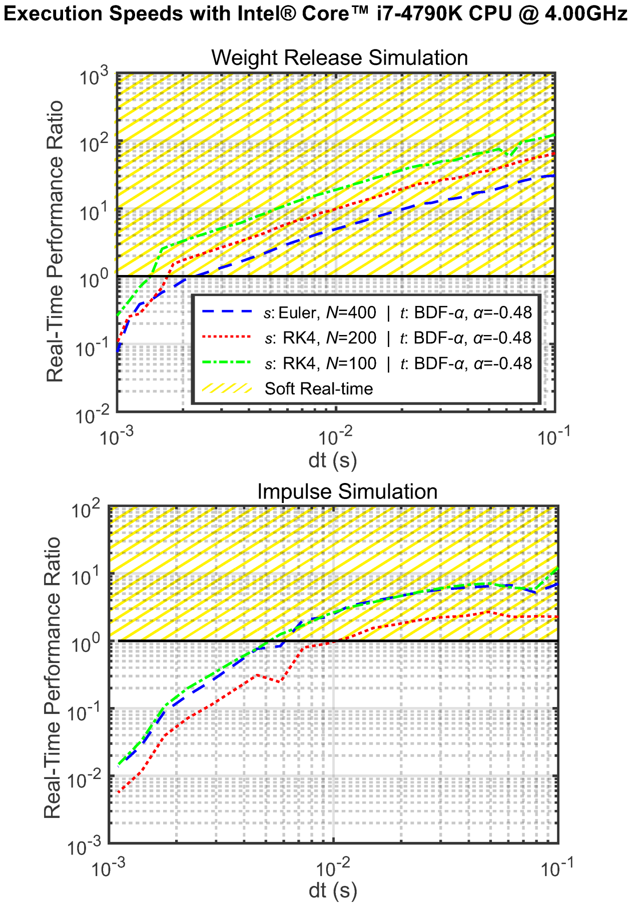

Fig. 6. We plot the real-time performance ratio versus the time step on a log-log scale for our two simulation datasets. The real-time performance ratio is the amount of time simulated divided by the wall-clock time spent running the simulation. A ratio greater than or equal to one indicates soft real-time performance, as shown by the gold-shaded region. The weight release scenario requires significantly less effort to solve than the impulse scenario. For the difficult impulse scenario, the simulation can run in soft real-time with $dt = 6$ms. The smallest time step for the impulse simulations is slighter larger than 1ms because of convergence issues with small time steps.

C. Real-Time Performance

To evaluate computational speed, we ran many simulations of both the weight release and impulse response scenarios using increasing values of dt (logarithmically spaced). Results are shown in the log-log plots of Figure 6. The real-time performance ratio is the amount of time simulated divided by the wall-clock time spent running the simulation, and the gold-shaded region indicates real-time performance. The plot confirms that most of the simulations ran in real-time. At higher time steps, the dependence of the run time on the time step is nearly linear. The run time also appears linear in the number of spatial steps, N, because the runtime approximately doubles when N doubles.

Not surprisingly, the impulse response case requires higher computational times due to the increased presence of faster dynamic modes that require more solver iterations per time step. A time step of 2 milliseconds captured the high-frequency dynamics very accurately, as shown in Figure 5, but this simulation required more computation time than it simulated, and the speed is further reduced when the solver begins to encounter numerical ill-conditioning at smaller time steps.

IV. Discussion and Conclusions

We have validated the proposed rod discretization in a highly dynamic impulse scenario. The simulation behavior is remarkably close to experimental data. It can be seen in Figure 5, and is more evident in the multimedia attachment, that the impulse simulation does not capture the very highest frequency modes of the experimental rod. Even with this unfortunate discrepancy, the comparison makes a compelling case for the physical accuracy of this method.

It is important to consider the issue of catastrophic cancellation when calculating a finite difference with a small increment. This was mainly a problem for the impulse experiment, where high frequency modes played an important role in the rod behavior, making small time steps necessary to resolve the high-frequency behavior accurately. However, the modified simple shooting method allowed us to accurately and stably predict the impulse response at small time steps.

We note that the tip responses of simulations using Euler's method with N=400 and RK4 with N=100 (requiring the same number of function evaluations) are visibly identical, while RK4 ran consistently faster, perhaps because its superior error scaling resulted in faster shooting method convergence. A thorough investigation of the relation between the time step, spatial step, solver tolerance, order of accuracy, and computational speed is left to future work. Overall, the execution speeds of this implementation of a dynamic rod simulation are encouraging for the prospect of model-based control of continuum robots and manipulation of elastic rods.

V. Acknowledgments

The high-speed camera was provided and operated by Christopher Combs, Phillip Kreth, and John Schmisseur of the University of Tennessee Space Institute.

References

[1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, may 2015.

[2] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications : A Survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[3] J. Spillmann and M. Teschner, "C O R D E : Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects," vol. 1, 2007.

[4] M. Benosman and G. Le Vey, "Control of flexible manipulators: A survey," *Robotica*, vol. 22, no. 5, pp. 533–545, 2004.

[5] I. Tunay, "Spatial Continuum Models of Rods Undergoing Large Deformation and Inflation," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 297–307, apr 2013.

[6] S. S. Antman, *Nonlinear Problems of Elasticity Second Edition*, 2005, vol. 107.

[7] S. F. F. Gibson and B. Mirtich, "A Survey of Deformable Modeling in Computer Graphics," *Merl - a Mitsubishi Electric Research Laboratory*, pp. 1–31, 1997.

[8] Y. Duan, D. Li, and P. F. Pai, "Geometrically exact physics-based modeling and computer animation of highly flexible 1D mechanical systems," *Graphical Models*, vol. 75, no. 2, pp. 56–68, 2013.

[9] H. Lang, J. Linn, and M. Arnold, "Multi-body dynamics simulation of geometrically exact Cosserat rods," *Multibody System Dynamics*, vol. 25, no. 3, pp. 285–312, 2011.

[10] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic Model of a Multibending Soft Robot Arm Driven by Cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, oct 2014.

[11] D. Rucker, R. Webster III, R. Webster III, and R. Webster III, "Statics and Dynamics of Continuum Robots with General Tendon Routing and External Loading," *IEEE Transactions on Robotics*, vol. 27, no. 6, 2011.

[12] S. Hadap, "Oriented strands: dynamics of stiff multi-body system," *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 91–100, 2006.

[13] F. Bertails, B. Audoly, B. Querleux, F. Leroy, J.-l. Lévêque, and M.-p. Cani, "Predicting Natural Hair Shapes by Solving the Statics of Flexible Rods," 2005.

[14] W. S. Rone and P. Ben-Tzvi, "Continuum robot dynamics utilizing the principle of virtual power," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 275–287, 2014.

[15] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," *ACM Transactions on Graphics*, vol. 27, no. 3, p. 1, 2008.

[16] S. Sueda, G. L. Jones, D. I. W. Levin, and D. K. Pai, "Large-scale dynamic simulation of highly constrained strands," *ACM Transactions on Graphics*, vol. 30, no. 4, p. 1, 2011.

[17] F. Renda, M. Cianchetti, M. Giorelli, A. Arienti, and C. Laschi, "A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm." *Bioinspiration & biomimetics*, vol. 7, no. 2, p. 025006, jun 2012.

[18] W. Tang, P. Lagadec, D. Gould, T. R. Wan, J. Zhai, and T. How, "A realistic elastic rod model for real-time simulation of minimally invasive vascular interventions," *Visual Computer*, vol. 26, no. 9, pp. 1157–1165, 2010.

[19] C. Gatti-Bono and N. Perkins, "Physical and Numerical Modelling of the Dynamic Behavior of a Fly Line," *Journal of Sound and Vibration*, vol. 255, no. 3, pp. 555–577, 2002.

[20] C.-C. Lan and K.-M. Lee, "Generalized Shooting Method for Analysing Compliant Mechanisms with Curved Members," *Journal of Mechanical Design*, vol. 128, no. July 2006, pp. 765–775, 2006.

[21] C. C. Lan, K. M. Lee, and J. H. Liou, "Dynamics of highly elastic mechanisms using the generalized multiple shooting method: Simulations and experiments," *Mechanism and Machine Theory*, vol. 44, no. 12, pp. 2164–2178, 2009.

[22] D. C. Rucker and R. J. Webster III, "Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading," *Robotics, IEEE Transactions on*, vol. 27, no. 6, pp. 1033–1044, 2011.

[23] J. Linn, H. Lang, and a. Tuganov, "Geometrically exact Cosserat rods with Kelvin-Voigt type viscous damping," no. 2, pp. 79–96, 2012.

[24] F. Baus, A. Klar, N. Marheineke, and R. Wegener, "Low-Mach-Number-Slenderness Limit for Elastic Cosserat Rods," *Preprint arXiv:math.AP/1507.03432*, jul 2015.

[25] E. A. Celaya and J. Jos, "BDF- α : A Multistep Method with Numerical Damping Control," vol. 1, no. 3, pp. 96–108, 2013.

[26] J. Till, C. E. Bryson, S. Chung, A. Orekhov, and D. C. Rucker, "Efficient Computation of Multiple Coupled Cosserat Rod Models for Real-Time Simulation and Control of Parallel Continuum Manipulators," *Proc. IEEE Conference on Robotics and Automation*, pp. 5067–5074, 2015.

[27] T. Bretl and Z. McCarthy, "Mechanics and Quasi-Static Manipulation of Planar Elastic Kinematic Chains," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 1–14, feb 2013.

[28] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Computing Surveys*, vol. 23, no. 1, pp. 5–48, 1991.

[29] R. Holsapple, R. Venkataraman, and D. Doman, "A modified simple shooting method for solving two-point boundary-value problems," *IEEE Aerospace Conference Proceedings*, vol. 6, no. 0, pp. 2783–2790, 2003.

[30] G. Guennebaud, B. Jacob, and Others, "Eigen v3," http://eigen.tuxfamily.org, 2010.