2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

A Fast Geometric Framework for Dynamic Cosserat Rods with Discrete Actuated Joints

Hossain Samei[1] and Robin Chhabra[1*]

***Abstract*— Current dynamical models of Cosserat rods often use the finite element method limited by computational efficiency or the finite difference method in a Cartesian framework with a compromise to accuracy. We employ the finite difference method in a geometric framework to develop solutions that are both computationally efficient and accurate. A numerical study is conducted on various backward-differentiation discretization and Runge-Kutta-Munthe-Kaas integration schemes, focusing on their accuracy and computational efficiency. Case studies are conducted on a single-degree-of-freedom joint actuated Cosserat rod to mitigate additional sources of undesired error from the numerical analysis, e.g. multi-body interactions, moving base dynamics, etc. The proposed geometric integrators are demonstrated to improve solution accuracy compared to the published finite difference models. The presented solution is parameterization-free and also computationally efficient with the potential for use in real-time applications, e.g., model-based control of soft manipulators.**

I. Introduction

Soft and continuum manipulators have been extensively studied since their inception in the 1990's [1]. The early work by Chirikjian investigates the dynamics and control of hyper-redundant continuum manipulators [1]. The proposed framework uses the modal shape functions which can be superimposed to capture complex motions of a continuum manipulator [2]. Although this method has been proven effective, it is computationally expensive and functionally dependent on the choice of shape functions [2]. Constant Curvature (CC) models have been developed to address these shortcomings, and they remain one of the most prevalent methods used to model soft continuum robots [3]–[6]. Various experimental studies have been conducted to demonstrate the accuracy of these models to predict quasi-static behavior of soft continuum manipulators [6], [7]. To improve the accuracy of CC models, variable and piecewise-constant curvature models have been proposed which can handle complex manipulator geometries [8], [9].

The current state-of-the-art method to simulate soft continuum manipulators is Cosserat rod theory [10], [11]. This theory has been present in the material science community for over a century, but has only been recently applied to robotics [12]. Cosserat rod theory provides a geometrically exact dynamic model of a 1-Dimensional (1D) continuum in the form of a set of non-linear Partial Differential Equations (PDEs) that include both position and rotation of the cross-sections along the continuum [12]. This theory is more mathematically involved comparing to the CC models but it accurately predicts the dynamics, bifurcations and stability of a continuum [13], [14]. Although shape functions can achieve similar dynamic models, the Cosserat theory is not limited by the choice of a basis for solutions of the resulting PDEs [1], [15]. Research has also been done to adopt shape functions to parameterize the strain of a Cosserat rod instead of its displacement, but it is again limited by the choice of solutions [16], [17]. The dynamic partial differential equations of Cosserat rods do not have a known global analytical solution. This mandates the use of numerical tools to simulate the dynamics; most implementations use either the Finite Element Method (FEM) or Finite Difference Method (FDM) [13], [15]. The FEM produces accurate but computationally expensive solutions [18]. However, the solutions based on the FDM can be used in real-time applications, with compromise to the accuracy when operating under highly dynamic conditions [19], [20].

Screw theory is widely explored in the field of robotics to describe dynamics and develop control strategies [21]. It offers multiple advantages over classical theory but its key contribution is the implementation of global parameterization-free equations of motion [15], [21], [22]. The classical equations of motion need at least 3 parameters to define angular motion (e.g. Euler Angles and quaternions). This results in a set of highly nonlinear equations that may contain singularities due to the coordinate chart assignment and does not reflect the inherent symmetries of the dynamics [15], [21]. These issues are circumvented by the use of Lie groups to globally express the equations of motion without assuming any local parameterization [21]. Dynamic Cosserat rods have been recently formulated on the Special Euclidean group $\mathrm{SE}(3)$ and numerically simulated using the FEM by Grazioso [18]. This simulation method is limited by computational inefficiency that makes it unsuitable for real-time applications [18], [23].

The FDM has been widely used in computational fluid dynamics and heat transfer communities to simulate PDEs. The application of the FDM involves multiple sub-routines, e.g., Boundary Condition (BC) solver, semi-discretization and numerical integration. A discretization scheme can be categorized as implicit, e.g., Backward-Differentiation Formula (BDF), or explicit, e.g., Lax-Wendroff method [24], [25]. Explicit methods are computationally efficient but suffer from instability at large time-steps [24]. On the other hand, implicit methods need to solve a system of equations

*Corresponding author
[1]H. Samei and R. Chhabra are with the Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (Email: hossainsamei@cmail.carleton.ca & robin.chhabra@carleton.ca)

to satisfy the BCs and are computationally expensive [24]. Runge-Kutta (RK) integration schemes may include both implicit and explicit discretizations but are limited to applications on vector spaces [25]. An extension for explicit RK schemes, called Runge-Kutta-Munthe-Kaas (RKMK), has been developed for geometric integration on Lie groups [26], [27]. Geometric integrators implemented on the SE(3) offer improved accuracy and stability with a compromise to simplicity [28]–[30]. A comparison of various FDM discretizations for dynamic simulation of a Cosserat rod is reported in [19] but does not include geometric integrators and actuated joints.

In this paper, a Cosserat rod connected to a single degree of freedom actuated joint is simulated to compare different discretization schemes. This model captures the full dynamics of a flexible body without introducing undesired complexities, e.g., multi-body interaction, multi-degree of freedom joints, etc., to allow for the accurate analysis of different discretizations on the system response. There are also limited references for complex systems making it difficult to compare the solutions. The major contributions of this paper are:

1) Development of a geometric framework using the finite difference method to model a dynamic Cosserat rod.
2) Studying the accuracy and computational efficiency of geometric integrators under different discretizations.
3) Comparison with the existing finite difference solutions to demonstrate the advantages of the proposed geometric framework.

A software library of the implemented geometric solution for a joint actuated Cosserat rod is also distributed.

This paper is organized as follows: Section II briefly reviews the Lie group SE(3) and its Lie algebra. In Section III, we present the set of PDEs describing the dynamics of a Cosserat rod on the SE(3). Section IV employs the FDM to approximate the equations of motion by a set of ODEs. The boundary condition solver used to iteratively solve the joint wrench is elaborated in Section V. We present case studies in Section VI studying different discretizations, comparing to alternative formulations and demonstrating the complexity of the model. Section VII includes some concluding remarks.

II. Preliminaries

This section briefly reviews the basics of the Lie group SE(3) and its Lie algebra $\mathfrak{se}(3)$. Orientation-preserving orthonormal frame transformations that do not change distance between points in the 3-Dimensional Euclidean space are members of the SE(3). This is a matrix Lie group and the semi-direct product of the Special Orthogonal group SO(3) and $\mathbb{R}^3$. An element $g \in \text{SE}(3)$ is a transformation and can be represented by a position vector as $P \in \mathbb{R}^3$ and a rotation matrix as R in the Special Orthogonal group SO(3):

$$g = \begin{bmatrix} R & P \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \text{SE}(3).$$

Members of $\mathfrak{se}(3)$ correspond to infinitesimal transformations and are called twists. The Lie algebra $\mathfrak{se}(3)$ is the Cartesian product of $\mathfrak{so}(3)$, the Lie algebra of SO(3), and $\mathbb{R}^3$. There are vector space isomorphisms from vectors in $\mathbb{R}^3$ and $\mathbb{R}^6$ to matrices in $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$, respectively, which are denoted by the $\times$ and $\wedge$ operators:

$$\xi_\omega^\times = \begin{bmatrix} 0 & -\xi_{\omega,3} & \xi_{\omega,2} \\ \xi_{\omega,3} & 0 & -\xi_{\omega,1} \\ -\xi_{\omega,2} & \xi_{\omega,1} & 0 \end{bmatrix} \in \mathfrak{so}(3),$$

$$\xi^\wedge = \begin{bmatrix} \xi_\omega^\times & \xi_v \\ \mathbf{0}_{1\times 3} & 0 \end{bmatrix} \in \mathfrak{se}(3).$$

Here, the vector $\xi_\omega \in \mathbb{R}^3$ is defined as $[\xi_{\omega,1}, \xi_{\omega,2}, \xi_{\omega,3}]^T$, the vector $\xi_v \in \mathbb{R}^3$, and the vector $\xi = [\xi_v^T, \xi_\omega^T]^T \in \mathbb{R}^6$. An element $\xi^\wedge \in \mathfrak{se}(3)$ can be represented by a linear component $\xi_v \in \mathbb{R}^3$ and a rotational component $\xi_\omega^\times \in \mathfrak{so}(3)$, which is a 3×3 skew-symmetric matrix. We also define the $\vee$ operator to be the inverse of the $\wedge$ operator, converting the matrices in $\mathfrak{se}(3)$ to their corresponding twists in $\mathbb{R}^6$.

The elements of $\mathfrak{se}(3)$ can define infinitesimal motions in time or deformations in space. Elements of $\mathfrak{se}(3)$ can be mapped to transformations in SE(3) using the exponential map $\exp : \mathfrak{se}(3) \to \text{SE}(3)$. The transformation in SE(3) corresponds to propagating for one unit time or length with the constant twist. The exponential map of SO(3) and SE(3) can be calculated based on the Rodriguez formula as presented in [21], [31]. The Jacobian of the exp map at an element $\xi \in \mathfrak{se}(3)$ is the map $\text{dexp}_\xi : \mathfrak{se}(3) \to \mathfrak{se}(3)$ that is used in the RKMK integrator and can be approximated using the Bernoulli number as presented in [27], [30]. The Adjoint operator, $\mathbf{Ad}_g : \mathfrak{se}(3)^\vee \to \mathfrak{se}(3)^\vee$, is used to transform twists from one coordinate frame to another, based on the transformation between the two frames, g. In this paper, this operator is used to move between the local and global coordinate frames. The lie bracket operator is also called the adjoint operator and denoted by $\mathbf{ad}_\xi : \mathfrak{se}(3)^\vee \to \mathfrak{se}(3)^\vee$, where $\xi \in \mathfrak{se}(3)^\vee$ is a twist.

$$\mathbf{Ad}_g := \begin{bmatrix} R & P^\times R \\ \mathbf{0}_{3\times 3} & R \end{bmatrix}, \quad \mathbf{ad}_\xi := \begin{bmatrix} \xi_\omega^\times & \xi_v^\times \\ \mathbf{0}_{3\times 3} & \xi_\omega^\times \end{bmatrix}.$$

III. Cosserat-Rod Dynamical Equations

The Cosserat-Rod theory is a 1D model of a continuum that determines the position and rotation of its Cross-Sections (CSs) along its arc-length. Fig. 1 depicts an arbitrary transformation between a local coordinate frame $\mathcal{B}$ and the global coordinate frame $\mathcal{I}$ for a generic soft manipulator. The continuum is assumed extensible with a constant CS and is subject to shear, bending and torsion. The theory is based on local constitutive laws for the material stress-strain relations to retain accuracy over global large deformations. The required transformations between global and local frames naturally calls for a Lie group framework to formulate the dynamics of a 1D Cosserat rod on SE(3). The position and rotation of any CS of the continuum with respect to $\mathcal{I}$ at time $t \in \mathbb{R}$ and reference arc length $s \in [0, 1]$ is a transformation:

$$g_{\mathcal{B}}^{\mathcal{I}} : \mathbb{R}^2 \to \text{SE}(3)$$

$$(t, s) \mapsto g_{\mathcal{B}}^{\mathcal{I}}(t, s) = \begin{bmatrix} R_{\mathcal{B}}^{\mathcal{I}}(t, s) & P_{\mathcal{B}}^{\mathcal{I}}(t, s) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \in \text{SE}(3).$$

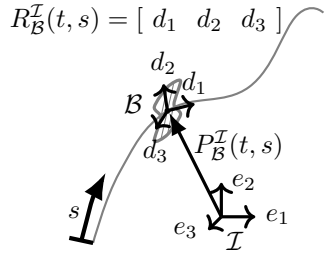

Fig. 1. Definition of parameters on a Cosserat rod

In this paper, the partial derivatives with respect to time and reference arclength for any variable x are respectively expressed by $\dot{x} = \frac{\partial x}{\partial t}$ and $x' = \frac{\partial x}{\partial s}$. The local twists associated with the infinitesimal changes of transformation in time and space, i.e., the local velocity $\eta(t,s)$ and total strain $f(t,s)$, are defined as follows:

$$\eta(t,s) := (g^{-1}\dot{g})^{\vee} \in \mathfrak{se}(3)^{\vee}, \tag{1}$$

$$f(t,s) := (g^{-1}g')^{\vee} \in \mathfrak{se}(3)^{\vee}. \tag{2}$$

Here, f is the sum of the free local strain $f_* \in \mathfrak{se}(3)^{\vee}$ and the local strain due to applied loads $\epsilon \in \mathfrak{se}(3)^{\vee}$, i.e.,

$$f(t,s) = \epsilon(t,s) + f_*(s).$$

The material properties are the inertia matrix $\mathbf{M}$, the stiffness matrix $\mathbf{K}$ and the damping matrix $\mathbf{B}$, all defined per unit length and constant over the continuum. We adopt a linear visco-elastic Kelvin-Voigt material model, as the constitutive law, to relate the local internal stress $\sigma \in \mathfrak{se}^*(3)^{\vee}$ to the local applied strain ϵ:

$$\sigma = \mathbf{K}\epsilon + \mathbf{B}\dot{\epsilon} = \mathbf{K}(f - f_*) + \mathbf{B}\dot{f}\ .$$

Based on the Hamilton's Principle, the weak form of the equations of motion for a dynamic Cosserat rod under applied distributed body wrench $F_{dst}(t,s) \in \mathfrak{se}^*(3)^{\vee}$ can be derived as the following set of PDEs [15]:

$$\eta' - \dot{f} = \mathbf{ad}_{\eta} f\ , \tag{3}$$

$$\mathbf{M}\dot{\eta} - \mathbf{ad}^T_{\eta}\,\mathbf{M}\eta - \mathbf{K}f' - \mathbf{B}\dot{f}' + \mathbf{ad}^T_f\ \big(\mathbf{K}(f - f_*) + \mathbf{B}\dot{f}\,\big) = F_{dst}\ . \tag{4}$$

To obtain the shape of a dynamic Cosserat rod, in time and space, these equations must be solved along with (1) and (2).

IV. Finite Difference Solution to PDEs

The set of PDEs defined by (1) - (4) describe the full dynamics of a Cosserat rod but has no analytical solution. The FDM has been adopted to numerically solve these equations assuming the discretized spatial and temporal step sizes of Δs and Δt respectively. First, the equations must be semi-discretized to simplify all the temporal derivatives from the set of PDEs.

Backward difference formula schemes derived using the Lagrange interpolation polynomial are linear implicit approximations [25]. Multi-step extensions exist offering higher-order accuracy, e.g. BDF-2, BDF-3, etc. but all schemes beyond BDF-2 are not A-stable [25]. Since $\mathfrak{se}(3)$ is a vector space, we can directly apply BDF schemes to approximate the time derivatives of any twist $X \in \mathfrak{se}(3)^{\vee}$. An N step BDF scheme takes the form

$$\dot{X}(t,s) = c_0\, X(t,s) + X^{\mathfrak{h}}\ , \tag{5}$$

$$X^{\mathfrak{h}} := \sum_{i=1}^{N} c_i\, X(t - i\,\Delta t, s)\ , \tag{6}$$

and $X^{\mathfrak{h}}$ is referred to as the history terms from the solution at previous time steps. The discretizations studied in this paper include BDF-1 and BDF-2 which use the vectors of coefficients $c = \frac{1}{\Delta t}[\,1,\ -1\,]$ and $c = \frac{1}{\Delta t}[\,1.5,\ -2,\ .5\,]$, respectively. These implicit schemes are applied to discretize $\dot{f}$, $\dot{f}'$ and $\dot{\eta}$ to transform the set of PDEs into the following set of semi-discretized ODEs

$$g' = g\, f^{\wedge}\ , \tag{7}$$

$$\eta' = (c_0\, \mathbb{I}_{6\times 6} + \mathbf{ad}_{\eta})\, f + f^{\mathfrak{h}}\ , \tag{8}$$

$$f' = \big(\mathbf{K} + c_0\,\mathbf{B}\big)^{-1}\Big(\mathbf{M}\dot{\eta} - \mathbf{ad}^T_{\eta}\,\mathbf{M}\eta - \mathbf{B}f'^{\mathfrak{h}} + \mathbf{ad}^T_f\ \big(\mathbf{K}(f - f_*) + \mathbf{B}(c_0\, f + f^{\mathfrak{h}})\,\big) - F_{dst}\Big)\ . \tag{9}$$

The states of the system are g, η and f and their ODEs are coupled functions of the states, e.g. $g' := \Phi(s,\, g,\, \eta,\, f)$.

Remark 1: Equation (1) is omitted as the transformation g can be computed by spatially integrating (2).

Runge-Kutta schemes are a general family of discretizations for integrating ODEs. For an explicit RK scheme, the next state is computed using only the values from previous iterations. Hence, considering a twist $X(s) \in \mathfrak{se}(3)^{\vee}$ at a fixed time step and $X' := \Psi(s, X)$, an N stage explicit RK scheme takes the form

$$X(s + \Delta s) = X(s) + \sum_{i=1}^{N} b_i\, k_i\ ,$$

$$k_i = \Delta s\ \Psi\Big(s + c_i\,\Delta s,\ X(s) + \sum_{j=1}^{i-1} a_{ij}\, k_j\Big)\ ,$$

where $b,\, c \in \mathbb{R}^N$ and $a \in \mathbb{R}^{N\times N}$, such that $a_{ij} = 0$ for $j \geq i$, are the weights associated with a specific scheme. The discretizations studied in this paper include Euler, RK2 and RK4 and their corresponding weights can be found in [25]. The discretized ODEs need to be evaluated at intermediate locations within the step size Δs where the history terms are not saved from previous iterations. Cubic interpolation can be applied using the discrete nodes as

$$X^{\mathfrak{h}}(s + \mu\,\Delta s) = a\,\mu^3 + b\,\mu^2 + \mu\,\big(X^{\mathfrak{h}}(s + \Delta s) - X^{\mathfrak{h}}(s - \Delta s)\big) + X^{\mathfrak{h}}(s)\ ,$$

$$a = X^{\mathfrak{h}}(s + 2\,\Delta s) - X^{\mathfrak{h}}(s + \Delta s) - X^{\mathfrak{h}}(s - \Delta s) + X^{\mathfrak{h}}(s),$$

$$b = X^{\mathfrak{h}}(s - \Delta s) - X^{\mathfrak{h}}(s) - a_0\ ,$$

where $\mu \in [0, 1]$ can define any intermediate location between s and $s + \Delta s$. A simpler alternative is linear interpolation between two nodes which can be computed as

$$X^{\mathfrak{h}}(s + \mu\,\Delta s) = X^{\mathfrak{h}}(s) + \big(X^{\mathfrak{h}}(s + \Delta s) - X^{\mathfrak{h}}(s)\big)\ .$$

The proposed RK integrators and interpolation methods are only valid for $\mathfrak{se}(3)$ and special consideration is needed to extend these methods to the SE(3).

Remark 2:

1) The equations (8) and (9) are coupled and must be evaluated with an RK scheme simultaneously.
2) Implementation of cubic interpolation assumes $X^{\mathfrak{h}}(x) = X^{\mathfrak{h}}(0)$ when $x < 0$ or $X^{\mathfrak{h}}(x) = X^{\mathfrak{h}}(1)$ when $x > 1$.

Explicit RK methods have been extended for application on Lie Groups as Runge-Kutta-Munthe-Kaas methods. They adopt the same weights a, b and c as their corresponding RK methods applied to the Lie algebra. Though, the group elements $g \in \mathrm{SE}(3)$ are propagated as

$$g(s + \Delta s) = \exp(v) \cdot g(s) \ ,$$
$$v = \Delta s \sum_{i=1}^{N} b_i \, k_i \ ,$$
$$k_i = \mathrm{dexp}_{u_i}^{-1} \ \Phi\big(s + c_i \, \Delta s, \ \exp(u_i) \cdot g(s)\big) \ ,$$
$$u_i = \Delta s \sum_{j=1}^{i-1} a_{ij} \, k_j \ .$$

The dexp^{-1} is often approximated using a truncated series with the Bernoulli numbers for a generic Lie Group as presented in [27], [30]. This RKMK scheme is applied to numerically solve (7) and (8)-(9) are solved using the presented RK scheme.

V. Boundary Condition Solver

The single degree of freedom joint is attached at $s = 0$ and is identified with a constant joint twist $\xi \in \mathfrak{se}(3)^{\vee}$ and a time varying joint parameter $\theta(t) \in \mathbb{S}$. The boundary conditions on the Cosserat rod, based on a given joint motion and applied load at the End Effector (EE), are

$$\eta(t, 0) = \dot{\theta}(t) \, \xi \ ,$$
$$f_{BC}(t) := \mathbf{K}^{-1}\big(F_{ext} - \mathbf{B}\dot{f}(t, 1) + \mathbf{K} f_*(1)\big) \ .$$

These BCs are applied on different boundaries of the system, so a BC solver is needed to move them all to $s = 0$ before the equations can be numerically evaluated.

The shooting method is implemented as a BC solver to iteratively solve for the unknown states at $s = 0$ to enforce the specified BCs, at a fixed time step t. This numerically computes the Jacobian to calculate the variation of $f(t, 1)$ due to changes in estimates of $f(t, 0)$ by spatially integrating the ODEs in (7)-(9). This Jacobian is then used to iterate $f(t, 0)$ and minimize

$$\varepsilon = f_{BC}(t) - f(t, 1) \ ,$$

referred to as the residual, within a user specified tolerance. The iterated state $f(t, 0)$ is updated by a non-linear optimization scheme, e.g., the Levenberg-Marquardt (LM) algorithm [14], or the Trust Region Dog-Leg Method (TRDL) [32]. The "fsolve" function in Matlab can be easily implemented to code the shooting method, as summarized in Fig. 2.

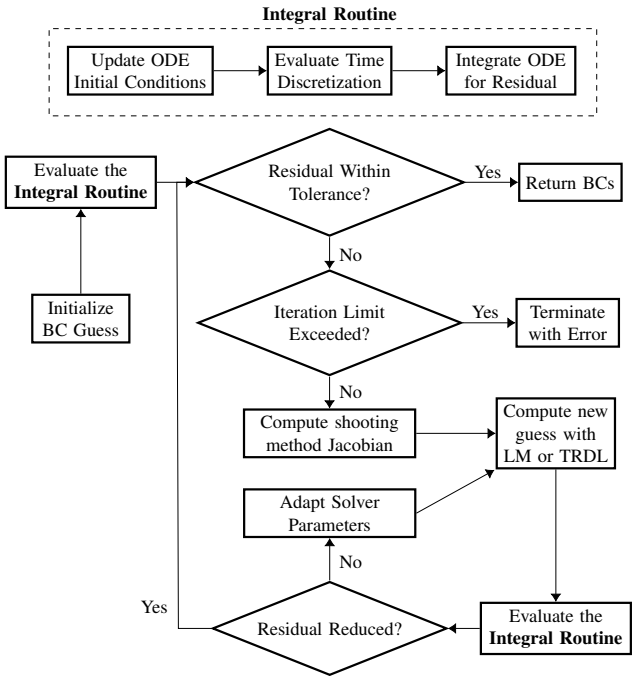

Fig. 2. Shooting method solver routine using Matlab function "fsolve" to determine the BCs at a particular time step

For each iterated guess in the BC solver we must update the semi-discretizations in the integral routine with the current guess for $f(t, 0)$ (see Fig. 2), since they depend on the current states of the system. When the dynamically compatible $f(t, 1)$ is determined, the system of ODEs is spatially integrated to obtain the states of the rod at the current time. To propagate to the next time step, we update the history of the states $f^{\mathfrak{h}}(s)$ and $\eta^{\mathfrak{h}}(s)$ based on the current solution and repeat the BC solver algorithm with the updated BCs. The dynamically compatible solution for $f(t, 0)$ is used as the initial guess for $f(t + \Delta t, 0)$. This BC solver approach has been implemented to simulate dynamic Cosserat rods in real-time [19].

VI. Simulation & Discussion

The proposed method expresses the equations of motion in the local coordinate frame for a dynamic Cosserat rod as a system of PDEs on the Lie group SE(3). The numerical solution, using FDM, was implemented in Matlab to study the computational efficiency of different discretizations in the context of alternatives to our solution. The dynamic response of a revolute joint actuated visco-elastic flexible body with the material and geometric properties are noted in Table I, assuming a circular CS.

The constant mass, stiffness and damping matrices are calculated using these material and geometric properties as

$$\mathbf{M} = \begin{bmatrix} \rho A \, \mathbb{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & J_{II} \end{bmatrix}, \ \mathbf{K} = \begin{bmatrix} K_{uu} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & K_{\omega\omega} \end{bmatrix},$$
$$\mathbf{B} = \begin{bmatrix} B_{uu} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mu \, J_{II} \end{bmatrix},$$

TABLE I

SIMULATION PARAMETERS

Variable	Value	Description
L	$0.4\ m$	Free Length
f_*	$[1, \mathbf{0}_{1\times5}]^T$	Free Strain
A	$3.14 \times 10^{-4}\ m^2$	CS Area
I	$7.854 \times 10^{-9}\ m^4$	CS Inertia
ρ	$4.5 \times 10^{3}\ kg/m^3$	Density
μ	$2 \times 10^{6}\ N/sm^2$	Viscosity
E	$8 \times 10^{3}\ N/m^2$	Young's Modulus
G	$3.85 \times 10^{7}\ N/m^2$	Shear Modulus
Tol	1×10^{-9}	BC Solver Tolerance

where $J_{II} = \text{diag}(2\,I, I, I)$, $K_{uu} = A\ \text{diag}(E, G, G)$, $K_{\omega\omega} = J_{II}\ \text{diag}(G, E, E)$ and $B_{uu} = \mu\, A\ \text{diag}(3, 1, 1)$.

A. Computational Efficiency of Discretization Schemes

The joint actuated flexible body is simulated under different combinations of the following discretizations: BDF-1, BDF-2, Euler, RK2 and RK4. A reference trajectory, adopting BDF-2 and RK4, was simulated and the EE position for the system response is shown in Figure 3 and the simulation parameters are noted in Table II.

TABLE II

SIMULATION PARAMETERS FOR TEST CASE I

Variable	Value	Variable	Value
$\ddot{\theta}_{act}$	$30\ rad/s^2$	t_{act}	$0.5\ s$
F_{EE}	$\mathbf{0}_{6\times1}$	Δt	$0.005\ s$
F_{dst}	$\mathbf{0}_{6\times1}$	Δs	$0.004\ m$

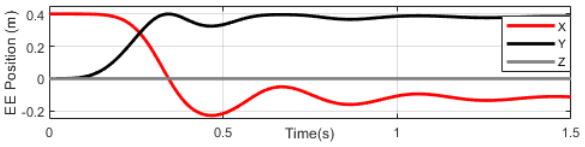

Fig. 3. Reference EE position for the simulation

The system starts from rest with $\eta(0, s) = \mathbf{0}_{6\times1}$ and under no applied loads so $f(0, s) = f_*$. It is then subject to a constant joint acceleration of $\ddot{\theta}_{act}$ followed by a constant joint deceleration of $\ddot{\theta}_{act}$ in t_{act} time. The simulation remains at a constant joint position after t_{act} and captures the damped vibrating response of the EE. The EE position for the response is saved as $X_{i,Ref}$ at time step t_i and will be used to compare to the solutions using a coarser mesh.

Remark 3: The ODE integration on a Lie group is implemented using a RKMK scheme with corresponding weights.

Using this computed solution as the reference, various discretizations are studied using a larger time step $\Delta t = .01\ s$. All simulations and computations were done in Matlab on an Intel i5-8400 2.8GHz CPU and 12 GB of RAM. The accumulated tip displacement error between the studied and reference solution in time is computed as

$$e = \sum_{i=1}^{t_f/\Delta t} \left\| X_i - X_{i,Ref} \right\|,$$

where X_i is the tip position under the studied discretization at time step t_i. To compare its performance, the spatial step size is iterated to determine the most computationally efficient solution that yields an accumulated error of $e < 1.21\,m$ (corresponding to an error of about 2% of L at each time step). The corresponding spatial step size, accumulated error and computation run-time are tabulated in Table III.

TABLE III

FOR $\Delta t = .01$ AND 2% ACCURACY WITH $e < 1.21$

Semi-Dsc	ODE Int	Δs (m)	e	Run-time (s)
BDF-1	Euler	0.04	*1.34*	10.42
BDF-1	RK2	0.044	*1.35*	14.99
BDF-1	RK4	0.057	*1.38*	20.04
BDF-2	Euler	0.033	0.72	12.75
BDF-2	RK2	0.036	1.16	19.38
BDF-2	RK4	0.050	1.09	25.30

Remark 4:

1) For the Δt assumed, all the solutions using BDF-1 cannot be computed within the desired accuracy.
2) All of these solutions used cubic interpolation as it produced lower errors than linear interpolation with only a marginal increase to computational expense.

BDF-2 retains accurate solutions over larger Δt without a significant compromise to computational efficiency. Higher order ODE integrators introduce significant computational expense but are able to retain accurate solutions over larger step sizes using the geometric integration. For solutions prioritizing accuracy, BDF-2 should be used to semi-discretize the equations and RK4 integration to solve the resulting ODEs. For solutions prioritizing computational efficiency, BDF-2 should still be used to semi-discretize the equations but Euler integration to resolve the resulting ODEs. Even the most efficient solutions as presented in the table are not suitable for real-time applications. Further software optimizations are needed to adjust the temporal step size, BC solver tolerance, limits on EE velocity, etc. before application to real-time systems.

B. Geometric Integrator vs. Vector-Space Integrator

A case study was conducted to compare the proposed model with the solution presented by Till in [19]. Both methods adopt the same discretizations, semi-discretize with BDF-2 and resolve the resulting ODEs with RK4. The differences between the two methods are the integration scheme and the coordinate frame in which the dynamics is expressed. Our solution uses the exponential map to directly integrate the matrices (i.e. assuming curved elements) over SE(3), whereas [19] assumes quaternion parameterization and integrates the solution in coordinates (i.e. assuming linear elements). We also apply the RKMK-4 on Lie groups, an extension of the RK4 for vector spaces. Further, the proposed method is derived in the local body coordinate frame, while [19] is derived in the global coordinate frame.

A planar cantilever rod with the properties adopted from Table I is simulated from rest and subject to a wrench $F_{EE} = [0, 0, 0, 0, 0, 5\,N.m]^T$ at the EE, plotted in Figure 4.

The motion of the EE is computed using a temporal step size of $\Delta t = 0.0025\,s$ with a coarse mesh of 2 spatial elements, $\Delta s = 0.2\,m$, using both models and a refined mesh of 10 spatial elements, $\Delta s = 0.04\,m$, using Till's model. A reference trajectory is plotted assuming refined spatial and temporal step sizes of $\Delta s = 0.004\,m$ and $\Delta t = 0.001\,s$ which indicate convergence to the same solution for both models under the refined mesh.

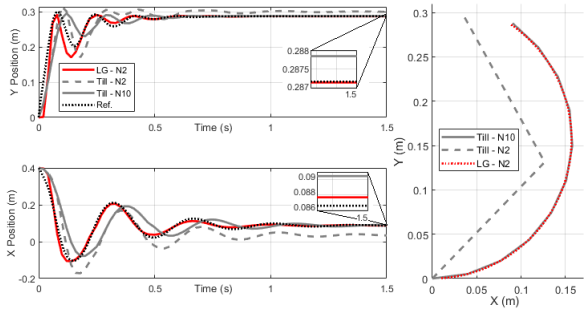

Fig. 4. EE position and a snapshot of configuration after the simulation.

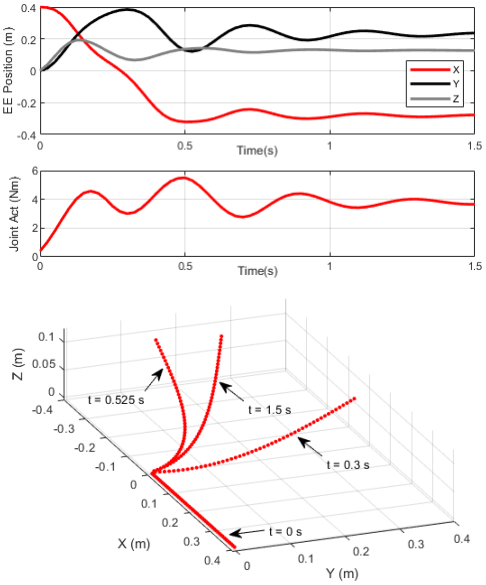

Fig. 5. Dynamic response of the simulated 3D Cosserat rod.

The proposed formulation implementing geometric integrators on the Lie group is more accurate than the solutions implemented in [19], even for a spatially refined mesh. For the large deformations in this simulation, a large number of linear elements are needed to capture the response accurately. This was illustrated in Figure 4 which shows the deformed rod using the different models at $t = 1.5\,s$, and the deficiency of linear spatial elements is visually evident.

Till's model (which has been implemented in real-time after software optimization) with 10 elements and our solution with 2 elements took $3.685\ s$ and $3.977\ s$ respectively to compute this simulation. It was reported that the FDM solution was able to be refined up to $0.35\ s$ in [19], so following similar refinement our proposed solution has the demonstrated potential for real-time application.

C. Complex Loading Case Simulation

The model can capture joint actuation, applied distributed or discrete wrenches for the 3D response of a Cosserat rod. This simulation highlights all of these conditions to demonstrate the completeness of our model. The same flexible body is simulated for the joint trajectory and the applied wrenches given in Table IV, both expressed in the local coordinate system.

TABLE IV
SIMULATION PARAMETERS FOR TEST CASE III

Variable	Value	Variable	Value
$\ddot{\theta}_{act}$	$30\ rad/s^2$	t_{act}	$0.5\ s$
F_{EE}	$[0,0,5,0,0,0]^T\ N$	Δt	$0.025\ s$
F_{dst}	$[0,-50,0,0,0,0]^T\ N/m$	Δs	$0.04\ s$

The control torque needed to follow the trajectory, as well as the motion of the EE, as well as a few snapshots of the response for the initial configuration, peaks of oscillations, and the final position, are illustrated in Figure 5.

Remark 5: The formulations in this paper expressed in the local frame need to consider the transformation from the local frame to the inertial frame for wrenches applied on the global frame, e.g. gravity, but not for wrenches applied on the local frame, e.g. distributed pressure channels. The inverse is true for formulations in global coordinates.

VII. CONCLUSION

A joint-actuated dynamic Cosserat rod was modeled on the SE(3) and simulated using the FDM. RKMK schemes were applied to geometrically integrate the PDEs and complete a parameterization-free solution. Our model was compared to an experimentally validated model to demonstrate improved accuracy at comparable computational efficiency, highlighting its advantages and potential for real-time implementations. The software library developed for the various integration schemes is available at: `https://github.com/HSamei/CosseratRod-RKMK`. The proposed framework will be extended to solve the dynamics of actuated serial-link rigid-flexible manipulators with multi-degree-of-freedom joints that has been recently studied by the authors. The developed software extends the library with higher-order RKMK geometric integrators.

The future directions of this research include: (i) software optimization for real-time applications, (ii) extension of the proposed model to include moving-base and multi-degree-of-freedom joints, and (iii) investigating the proposed method in model-based control of soft robots.

ACKNOWLEDGMENT

This work is partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada (DGECR-2019-00085) and the Canada Research Chair Program.

REFERENCES

[1] G. Chirikjian, "Theory and applications of hyper-redundant robotic manipulators," Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 1992.

[2] G. Chirikjian and J. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 343–354, 1994.

[3] R. W. III and B. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[4] A. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *International Journal of Robotics Research*, vol. 35, no. 7, pp. 840–869, 2016.

[5] D. Rucker, R. W. III, G. Chirikjian, and N. Cowan, "Equilibrium conformations of concentric-tube continuum robots," *International Journal of Robotics Research*, vol. 29, no. 10, pp. 1263–1280, 2010.

[6] A. Marchese, K. Komorowski, C. Onal, and D. Rus, "Design and control of a soft and continuously deformable 2d robotic manipulation system," *IEEE Conference on Robotics and Automation*, pp. 2189–2196, 2014.

[7] C. Santina, R. Katzschmann, A. Bicchi, and D. Rus, "Dynamics of soft robots interacting with the environment," *IEEE International Conference on Soft Robots*, pp. 46–53, 2018.

[8] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of bionic handling assistant," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 935–949, 2014.

[9] G. Runge, M. Wiese, L. Gunther, and A. Raatz, "A framework for kinematic modeling of soft material robots combining finite element analysis and piecewise constant curvature kinematics," *International Conference on Control, Automation and Robotics*, pp. 7–14, 2017.

[10] M. Gazzola, L. Dudte, A. McCormick, and L. Mahadevan, "Forward and inverse problems in the mechanics of soft filaments," *Royal Science Open Society*, 2018.

[11] N. Naughton, J. Sun, A. Tekinalp, G. Chowdhary, and M. Gazzola, "Elastica: A compliant mechanics environment for soft robotic control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3589–3396, 2021.

[12] D. Trivedi, A. Lofti, and C. Rahn, "Geometrically exact model for soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 773–780, 2008.

[13] J. Till, "On the statics, dynamics and stability of continuum robots: Model formulation and efficient computational schemes," Ph.D. dissertation, University of Tennessee, Knoxville, TN, 2019.

[14] J. Till, C. Bryson, S. Chung, A. Orekhov, and C. Rucker, "Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators," *IEEE International Conference on Robotics and Automation*, pp. 5067–5074, 2015.

[15] S. Grazioso, "Geometric soft robotics: A finite element approach," Ph.D. dissertation, University of Naples Federico, Naples, Italy, 2017.

[16] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: A strain parameterization based approach," *IEEE Transaction on Robotics*, vol. 37, pp. 847–863, 2021.

[17] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multi-section soft manipulator dynamics," *IEEE Transaction on Robotics*, vol. 34, pp. 1518–1533, 2018.

[18] S. Grazioso, G. Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft Robotics*, vol. 6, no. 6, pp. 790–811, 2019.

[19] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *International Journal of Robotics Research*, vol. 38, no. 6, pp. 723–746, 2019.

[20] H. Lang, J. Linn, and M. Arnold, "Multi-body dynamics simulation of geometrically exact cosserat rods," *Multibody System Dynamics*, vol. 25, pp. 285–312, 03 2011.

[21] R. Murray, Z. Lie, S. Sastry, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[22] A. Tariverdi, V. Venkiteswaran, O. Martinsen, O. Elle, J. Torrensen, and S. Misra, "Dynamic modeling of soft continuum manpulators using lie group variational integration," *PLOS ONE*, 2020.

[23] F. Lagilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," *IEEE International Conferenc on Robotics and Automation*, pp. 2550–2555, 2015.

[24] C. Grossmann, H. Roos, and M. Stynes, *Numerical Treatment of Partial Differential Equations*. Berlin, Germany: Springer, 2007.

[25] A. Iserles, *A First Course in Numerical Analysis of Differential Equation*. Cambridge, UK: Cambridge University Press, 2009.

[26] H. Munthe-Kaas, "Runge-kutta methods on lie groups," *BIT Numerical Mathematics*, vol. 38, 1998.

[27] ——, "High order runge-kutta methods on manifolds," *Applied Numerical Mathematics*, vol. 29, 1999.

[28] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie-group methods," *Acta Numerica*, vol. 9, 2000.

[29] E. Celledoni, H. Marthinsen, and B. Owren, "An introduction to lie group integrators - basics, new developments and applications," *Journal of Computational Physics*, vol. 257, 2014.

[30] E. Jakobsen, "Obtaining numerical solutions to ordinary differential equations on differentiable manifolds with lie group integrators," *Norweigian University of Science and Technology*, 2021.

[31] R. Chhabra and M. Emami, "A generalized exponential formula for forward and differential kinematics of open-chain multibody systems," *Mechanism and Machine Theory*, vol. 73, pp. 61–75, 2014.

[32] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer, 2006.