

Solving Cosserat Rod Models via Collocation and the Magnus Expansion

Andrew L. Orekhov¹, Nabil Simaan^{1†}

Abstract—Choosing a kinematic model for a continuum robot typically involves making a tradeoff between accuracy and computational complexity. One common modeling approach is to use the Cosserat rod equations, which have been shown to be accurate for many types of continuum robots. This approach, however, still presents significant computational cost, particularly when many Cosserat rods are coupled via kinematic constraints. In this work, we propose a numerical method that combines orthogonal collocation on the local rod curvature and forward integration of the Cosserat rod kinematic equations via the Magnus expansion, allowing the equilibrium shape to be written as a product of matrix exponentials. We provide a bound on the maximum step size to guarantee convergence of the Magnus expansion for the case of Cosserat rods, compare in simulation against other approaches, and demonstrate the tradeoffs between speed and accuracy for the fourth and sixth order Magnus expansions as well as for different numbers of collocation points. Our results show that the proposed method can find accurate solutions to the Cosserat rod equations and can potentially be competitive in computation speed.

Index Terms—Continuum robots, soft robot modeling, Cosserat rod, Lie group methods

I. INTRODUCTION

Continuum and soft robot architectures have been studied for a variety of useful applications [1], [2], but their passive, continuously flexible structures make them difficult to model. There are many different kinematic and dynamic models presented in the literature (see [1–4] for reviews), but one commonly used method is to model the robot’s flexible structure as one or more Cosserat rods. This approach has been experimentally validated for concentric tube robots [5], [6], tendon-driven robots [7], [8], multi-backbone robots [9], and soft robots with fluidic [10] and tendon [11] actuation in both kinematic and dynamic studies [12].

Experimental validations have shown relatively accurate open-loop prediction of shape (position errors of 1-8% of arc length are typical), however, computing the model involves numerically solving a set of boundary value problems (BVPs) which can be computationally expensive. Although a number of works have demonstrated implementations fast enough for control in the cases of concentric tube robots [5], parallel continuum robots [13], and single-backbone tendon-driven robots [12], less accurate modeling methods

are still attractive due to their low computational cost when compared to the Cosserat rod models [8]. Furthermore, in cases where the model consists of many kinematically coupled Cosserat rods (e.g. multi-backbone robots [9], [14] and eccentric pre-curved tube robots [15]), the computational cost of the Cosserat rod models is a significant obstacle and more efficient numerical methods are still needed.

Another drawback of Cosserat rod modeling is that after the BVP has been numerically solved (typically with a shooting method), it can be difficult to compute forward/inverse Jacobians when many kinematic constraints are active. Although in many cases the partial derivatives associated with the Jacobian can be computed together with the forward integration of the Cosserat differential equations (DEs) [16], for more complicated continuum structures one typically has to resort to finite-difference estimation. A numerical method that results in an analytical expression can aid in computing these matrices and conducting other analytical analysis of the rod equilibrium shape for design and control.

This paper is motivated by these two limitations of Cosserat rod models and is a preliminary step towards addressing them. We propose a method that solves for the rod’s curvature distribution with global orthogonal collocation and uses the Magnus expansion, a Lie group integration method, to recover the shape of the rod. Solving the BVP in this way provides some computational advantages and results in a product of matrix exponentials expression for the shape, which, as shown in [17], [18], allows the Jacobian to be computed in closed-form. Although we consider the case of a single rod and additional evaluation is need for practical robotics scenarios, our results show that this method can potentially be competitive with other approaches.

Collocation, and other weighted residuals methods, have been demonstrated previously [4], [10], [19–21], but these works apply polynomial interpolation on internal wrenches or on the position and orientation instead of curvature. Curvature-based parameterizations have been used in [17], [22], but they do not combine this with collocation, and they use constant-twist deformation elements, in contrast to the global interpolation functions we use here. A polynomial curvature model was used in [23], although this work was focused on control of a planar robot and not modeling of Cosserat rods. In [24], a curvature parameterization is combined with collocation for Cosserat rod dynamics, but the Magnus expansion is not used.

Our method is most similar to recent works [18], [25] that also use polynomial approximations on curvature within a Lie-group framework. The Magnus expansion is used in

† Corresponding author

¹Department of Mechanical Engineering, Vanderbilt University, Nashville, TN 37235, USA (andrew.orekhov, nabil.simaan)@vanderbilt.edu

This work was supported by NSF award #1734461 and by Vanderbilt University funds. A. Orekhov was partly supported by the NSF Graduate Research Fellowship under #DGE-1445197.

[18] and Chebyshev polynomials are used in [25]. This paper complements and extends these works by 1) demonstrating the computational benefits of combining Chebyshev orthogonal collocation with the Magnus expansion for Cosserat rod BVPs, 2) analysing the maximum step sizes to guarantee convergence of the Magnus expansion, and 3) validating the approach with high-order polynomials against two known methods in a simulation study.

II. THE COSSERAT ROD EQUATIONS

Here we briefly review the Cosserat rod equations of static equilibrium. We assume that shear strains and extension are negligible, which has been shown to be a reasonable assumption for long, slender rods [5]. We also assume the rod is not subject to distributed loads, is not pre-curved, and has a uniform cross-section and bending stiffness. Although we consider this simpler case for brevity, the methods in this paper extend to these more general cases, which have been discussed in other work [7].

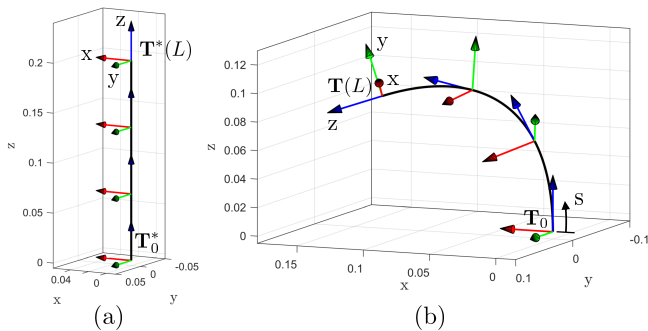


Fig. 1: Kinematic notation and frame assignment: (a) the rod in its undeflected state with reference frames $\mathbf{T}^*(s)$, and (b) the rod after undergoing a spatial deflection.

Under these assumptions, the shape of a deflected rod of length L is parameterized as a matrix function of arc-length $s \in [0, L]$ given as a homogenous transformation $\mathbf{T}(s)$:

$$\mathbf{T}(s) = \begin{pmatrix} \mathbf{R}(s) & \mathbf{p}(s) \\ 0 & 1 \end{pmatrix} \in SE(3) \quad (1)$$

where $\mathbf{p}(s) \in \mathbb{R}^3$ specifies the rod's shape in 3D and $\mathbf{R}(s) \in SO(3)$ describes the orientation of each local material frame (as shown in Fig. 1).

We assign reference frames $\mathbf{T}^*(s)$ to the unloaded rod's curve as shown in Fig. 1(a) with the z -axis of \mathbf{T}^* aligned with the rod and pointing towards its tip. Defining $\mathbf{u} = [u_x, u_y, u_z]^T$ as a vector of curvatures, we describe the motion of $\mathbf{T}(s)$ along the curve for unit-speed traversal along the arc-length by a twist vector $\boldsymbol{\xi} = [\mathbf{u}^T(s), 0, 0, 1]^T$ expressed in the moving frame. Using the wedge operator \wedge , we map \mathbf{u} to its skew-symmetric matrix $\hat{\mathbf{u}} \in so(3)$ and $\boldsymbol{\xi}$ to its $se(3)$ element, which is defined as:

$$\begin{aligned} \hat{\boldsymbol{\xi}}(s) &= \mathbf{X}(s) = \begin{pmatrix} \hat{\mathbf{u}}(s) & \mathbf{e}_3 \\ 0 & 0 \end{pmatrix} \in se(3) \\ \mathbf{X}(s) &= \mathbf{T}^{-1}(s)\mathbf{T}'(s) \end{aligned} \quad (2)$$

where $\mathbf{T}'(s)$ is the derivative of \mathbf{T} with respect to arc-length and $\mathbf{e}_3 = [0, 0, 1]^T$. We use a moving frame twist so that the rod's internal moment can be obtained directly from the local curvature:

$$\mathbf{m}(s) = \mathbf{R}(s)\mathbf{K}\mathbf{u}(s), \quad \mathbf{m}(s) \in \mathbb{R}^3 \quad (3)$$

where $\mathbf{K} = \text{diag}(EI, EI, JG)$ is the rod's arc-length normalized bending stiffness matrix. Note we have expressed $\mathbf{m}(s)$ in world frame since external wrenches are more easily expressed in world frame.

For a known tip-applied wrench, the Cosserat rod ordinary differential equations (ODEs) simplify to [7]:

$$\begin{aligned} \mathbf{T}'(s) &= \mathbf{T}(s)\mathbf{X}(s) \\ \mathbf{u}'(s) &= \mathbf{g}(\mathbf{u}(s)) = -\mathbf{K}^{-1}(\hat{\mathbf{u}}(s)\mathbf{K}\mathbf{u}(s) + \hat{\mathbf{e}}_3\mathbf{R}^T(s)\mathbf{f}_e) \end{aligned} \quad (4)$$

where $\mathbf{u}'(s)$ denotes a derivative with respect to s and \mathbf{f}_e is a force at the rod's tip expressed in world frame. The boundary conditions for a known applied tip wrench are given by:

$$\mathbf{u}(L) = \mathbf{K}^{-1}\mathbf{R}^T(L)\mathbf{m}_e \quad (5)$$

where $\mathbf{m}_e = [m_{e,x}, m_{e,y}, m_{e,z}]^T$ is the tip moment expressed in world frame. Solving the ODE's in (4) with the boundary conditions (5) provides the frames $\mathbf{T}(s)$ along the rod which gives the shape of the rod in space.

This boundary value problem (BVP) is typically solved via the shooting method. This has been successfully applied to a number of continuum robot architectures for forward/inverse kinematics [26] and forward dynamics [12]. One drawback of the shooting method is that it involves many forward integrations of $\mathbf{T}'(s)$ and $\mathbf{u}'(s)$ and therefore can be computationally expensive for Cosserat rod models with many kinematic constraints (e.g. [9], [15]). Another drawback is that, once a solution to the BVP is found, computing Jacobians requires finite difference approximation. Several works addressed this issue for some continuum robot architectures [13], [16], but this difficulty remains for architectures with many coupled Cosserat rods.

In the sections below, we describe a procedure for solving (4) and (5) that uses collocation to solve $\mathbf{u}'(s)$, which circumvents the need to numerically integrate $\mathbf{u}'(s)$, and a result from the geometric integration literature called the *Magnus expansion* to forward integrate $\mathbf{T}'(s)$. The combination of these two known techniques allows for the solution to the BVP to be expressed as a product of matrix exponentials, an analytical expression that can be used for further analytical analysis (e.g. Jacobian-based studies [17], [22]).

III. SOLVING VIA ORTHOGONAL COLLOCATION

Here we show how to solve (4) via orthogonal collocation, a direct variational approach that has been applied to a variety of DEs and BVPs [27–29]. First, we review polynomial interpolation and collocation.

Any continuous function can be approximated by obtaining the function values at a set of *interpolation nodes*, then fitting an *interpolating polynomial* to those function values. If we choose an interpolating polynomial of sufficiently

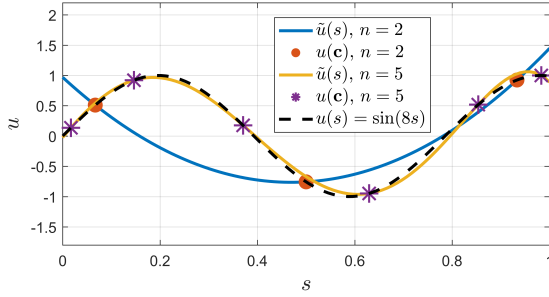


Fig. 2: Example of polynomial interpolation of a function $u(s) = \sin(8s)$ using a 2nd order and 5th order Chebyshev polynomials with interpolation nodes at the zeros of the n th order Chebyshev polynomial. Orthogonal interpolation provides rapid convergence with increasing n .

high order, it can provide a reasonable approximation of the function. Figure 2 shows an example of this, where $u(s) = \sin(8s)$ is approximated to varying degrees of fidelity by $\tilde{u}(s)$ given as a 2nd order polynomial or a 5th order polynomial. To avoid Runge's phenomenon, the interpolation nodes are chosen to be the zeros of an orthogonal polynomial. Here we choose Chebyshev polynomials but other orthogonal polynomials could also be used.

In a collocation method, an interpolating polynomial is used to find an approximate solution to a DE. An interpolating polynomial is chosen to approximate the unknown solution to the DE (which in our case is $\xi(s)$, the unknown twist distribution), a set of *collocation points* (similar to the interpolation nodes above) are chosen, and it is enforced that the interpolating polynomial satisfy the boundary conditions as well as the DE at the collocation points. This results in a set of algebraic equations that can be solved using standard nonlinear root-finding approaches. We now discuss how to carry out this procedure for a Cosserat rod.

In our context, we seek to find the unknown twist distribution $\xi(s)$ that satisfies the ODEs (4) and the boundary conditions (5). We choose to describe this unknown twist distribution as a set of n th order Chebyshev polynomials of the first kind, denoted as $\tilde{\xi}(s) = [\tilde{\mathbf{u}}(s), 0, 0, 1]^T$, where $\tilde{\mathbf{u}}(s) = [\tilde{u}_x(s), \tilde{u}_y(s), \tilde{u}_z(s)]^T$, $s \in [0, L]$. Note that here we use three interpolating polynomials to approximate $\xi(s)$, but for a general Cosserat rod case where shear strains are included, six interpolating polynomials would be needed. We choose the collocation points $\mathbf{c} = [c_0, \dots, c_n]^T$, $c_i \in [0, L]$, $i = 0 \dots n$ to be the Chebyshev polynomial zeros and call the function values $\tilde{\mathbf{u}}(c_i)$ the *collocation values*.

The Chebyshev polynomials of the first kind can be conveniently represented by a recurrence relation [29]:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n = 2, 3, \dots \quad (6)$$

$$T_0 = 1, \quad T_1(x) = x$$

where $T_n(x)$, $x \in [-1, 1]$ is the n th Chebyshev polynomial. To shift the Chebyshev polynomials to the domain $s \in [0, L]$ we apply the linear transformation

$$x(s) = \frac{2s - L}{L} \quad (7)$$

and evaluate $T_n(x(s))$ via (6). Henceforth, we denote $T_n(x(s))$ as simply $T_n(s)$ with (7) implied.

In a typical direct variational method, one describes the interpolating polynomial as an expansion in a basis of orthogonal polynomials, which requires computing modal coefficients from collocation values. For example, $\tilde{u}_x(s)$ may be represented by the n th order interpolating polynomial:

$$\tilde{u}_x(s) = \frac{1}{2}a_0T_0(s) + \sum_{i=1}^n a_iT_i(s) \quad (8)$$

where the modal coefficients a_i are found via:

$$a_i = \frac{2}{n+1} \sum_{k=0}^n \tilde{u}_x(c_k)T_i(c_k) \quad (9)$$

In an orthogonal collocation method, computing the modal coefficients is avoided by using a *differentiation matrix*. By taking advantage of the discrete orthogonality of Chebyshev polynomials, it has been shown that the derivatives at the collocation points can be written as a linear combination of the collocation values [29], [30]:

$$\frac{d}{ds} \begin{pmatrix} u_x(c_0) \\ \vdots \\ u_x(c_n) \end{pmatrix} = \mathbf{D}_n \begin{pmatrix} u_x(c_0) \\ \vdots \\ u_x(c_n) \end{pmatrix} \quad (10)$$

where the element in row $(i+1)$ and column $(j+1)$ of \mathbf{D}_n is given by:

$$d_{ij} = \begin{cases} \frac{1}{2} \frac{T''_{n+1}(c_i)}{T'_{n+1}(c_i)}, & i = j \\ \frac{T'_{n+1}(c_i)}{(c_i - c_j)T'_{n+1}(c_j)}, & i \neq j \end{cases} \quad (11)$$

Note that the above result requires that the collocation points be the Chebyshev zeros and that \mathbf{D}_n can be computed offline if the order of the interpolating polynomial is chosen *a priori*.

We now assemble the collocation values into a matrix \mathbf{U}_c where each column contains an interpolating polynomial:

$$\mathbf{U}_c = \begin{pmatrix} \tilde{u}_x(c_0) & \tilde{u}_y(c_0) & \tilde{u}_z(c_0) \\ \vdots & \vdots & \vdots \\ \tilde{u}_x(c_n) & \tilde{u}_y(c_n) & \tilde{u}_z(c_n) \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{u}}^T(c_0) \\ \vdots \\ \tilde{\mathbf{u}}^T(c_n) \end{pmatrix} \quad (12)$$

We want the interpolating polynomials to satisfy the boundary conditions and (4) at the collocation points. To ensure a square error residual Jacobian, we remove the last row of \mathbf{D}_n to form \mathbf{D}_{n-1} and form a matrix of error residuals:

$$\mathbf{E} = \begin{pmatrix} \mathbf{D}_{n-1} \mathbf{U}_c \\ \tilde{\mathbf{u}}^T(L) \end{pmatrix} - \begin{pmatrix} \mathbf{g}^T(\tilde{\mathbf{u}}(c_0)) \\ \vdots \\ \mathbf{g}^T(\tilde{\mathbf{u}}(c_{n-1})) \\ (\mathbf{K}^{-1} \mathbf{R}^T(L) \mathbf{m}_e)^T \end{pmatrix} \quad (13)$$

where we evaluate $\mathbf{g}(\tilde{\mathbf{u}}(c_i))$ by plugging the collocation value $\tilde{\mathbf{u}}(c_i)$ into (4). Note that $\mathbf{u}(L)$ is not one of the collocation values and must be computed from the interpolating polynomial. We show how to do this in Section V.

Equation (13) is a set of nonlinear algebraic equations we must now solve for the collocation values $\tilde{\mathbf{u}}(c_i)$, $i \in 0, \dots, n$.

This is achieved by minimizing the error in (13) using a nonlinear solver (e.g. Levenberg-Marquardt). Specifically, the error is defined as $\mathbf{e} = \text{vec}(\mathbf{E})$, where $\text{vec}(\mathbf{E})$ arranges the columns of \mathbf{E} into a vector. Once $\hat{\mathbf{u}}(c_i)$ are found, the modal coefficients and the interpolating polynomial $\hat{\mathbf{u}}$ are defined. Therefore, $\mathbf{X}(s)$ is defined in terms of the Chebyshev polynomials.

Computing $\mathbf{g}(\hat{\mathbf{u}}(c_i))$ in (13) requires integrating $\mathbf{T}'(s)$ to find $\mathbf{R}(s)$ at the collocation points. Examples of methods to do this include explicit Runge-Kutta methods, quaternion integration [31], and a variety of geometric integration methods [32], [33]. Although any of these approaches could be combined with collocation, here we take the geometric integration approach. We observe that $\mathbf{T}(s)$ is an element of the Lie group $SE(3)$, $\mathbf{X}(s)$ is an element of the Lie algebra $se(3)$, and that given a set of collocation values $\hat{\mathbf{u}}(c_i)$ (which would be guessed at each iteration of a nonlinear solver), the twist distribution $\mathbf{X}(s)$ is known via the interpolating polynomial. Therefore, $\mathbf{T}'(s)$ in (4) represents a linear DE on a Lie group, a class of problems that has received extensive study in other work [33]. We use a known result from the Lie group integration literature called the *Magnus expansion* to forward integrate $\mathbf{T}'(s)$. Note that a similar Lie algebra expansion was given in [34], which could also be used here in place of the Magnus expansion.

The benefits of using the Magnus expansion are 1) the integration is done on the Lie algebra $se(3)$ and mapped to $SE(3)$ via an exponential mapping, so $\mathbf{T}(s)$ is guaranteed to stay on $SE(3)$, 2) the quadrature method given in [33] for the Magnus expansion is numerically efficient and allows for closed-form gradients to be computed, and 3) the resulting solution to the BVP is given as a product of matrix exponentials that can be used for further analytical study (e.g. computing Jacobians). In the next section, we provide the main results on the Magnus expansion without proof and refer the reader to [33] for additional details.

IV. THE MAGNUS EXPANSION

It was shown in [33] that for sufficiently small s the solution to $\mathbf{T}'(s) = \mathbf{X}(s)\mathbf{T}(s)$ can be expressed as a matrix exponential of a twist $\boldsymbol{\psi} = [\boldsymbol{\psi}_u, \boldsymbol{\psi}_v]^T \in \mathbb{R}^6$:

$$\begin{aligned} \mathbf{T}(s) &= \mathbf{T}_0 e^{\boldsymbol{\Psi}(s)} \\ \boldsymbol{\Psi}(s) &\triangleq \hat{\boldsymbol{\psi}}(s) = \begin{pmatrix} \hat{\boldsymbol{\psi}}_u(s) & \boldsymbol{\psi}_v(s) \\ 0 & 0 \end{pmatrix} \in se(3) \end{aligned} \quad (14)$$

A short proof in [33] shows that the twist matrix $\boldsymbol{\Psi}(s)$ satisfies the differential equation

$$\boldsymbol{\Psi}'(s) = \text{dexp}_{-\boldsymbol{\Psi}(s)}^{-1}(\mathbf{X}(s)) \quad \boldsymbol{\Psi}(0) = \mathbf{0} \quad (15)$$

where the dexp operator is defined as:

$$\text{dexp}_{\boldsymbol{\Psi}}^{-1} = \sum_{i=0}^{\infty} \frac{B_i}{i!} \text{ad}_{\boldsymbol{\Psi}}^i \quad (16)$$

where B_i are the Bernoulli numbers and $\text{ad}_{\boldsymbol{\Psi}}$ denotes the 6×6 adjoint representation of an element in $se(3)$ [35]:

$$\text{ad}_{\boldsymbol{\Psi}} = \begin{pmatrix} \hat{\boldsymbol{\psi}}_u(s) & 0 \\ \hat{\boldsymbol{\psi}}_v(s) & \hat{\boldsymbol{\psi}}_u(s) \end{pmatrix} \quad (17)$$

To compute $\text{dexp}_{-\boldsymbol{\Psi}}^{-1}$, one can either truncate the infinite series or, for special cases, derive a closed-form expression as done in [35] for $se(3)$. The DE in (15) can then be numerically integrated with a standard Runge-Kutta method to find $\boldsymbol{\Psi}(s)$. To reduce the cost of numerically integrating, we take an alternative approach that avoids computing (16). In [36], Magnus solved (15) via Picard iteration, leading to a solution for $\boldsymbol{\Psi}(s)$ written as an infinite series of terms consisting of integrals of commutators. In [33], order analysis showed which terms can be dropped for a given order, resulting in the following fourth order Magnus expansion:

$$\begin{aligned} \boldsymbol{\Psi}^{[4]}(s) &= \int_0^s \mathbf{X}(\eta) d\eta \\ &+ \frac{1}{2} \int_0^s \left[\int_0^{\eta_1} \mathbf{X}(\eta_2) d\eta_2, \mathbf{X}(\eta_1) \right] d\eta_1 \end{aligned} \quad (18)$$

where the matrix commutator is given by $[\mathbf{X}_1, \mathbf{X}_2] = \mathbf{X}_1\mathbf{X}_2 - \mathbf{X}_2\mathbf{X}_1$. We do not replicate it here, but [33] also provides the sixth order expansion with 7 terms and up to 4 integrals per term.

At first glance, the fourth and sixth order Magnus expansions seem expensive to compute, however, it was shown in [33] that both expansions can be efficiently computed with Gaussian quadrature. Gaussian quadrature is an approach for approximating definite integrals via interpolating polynomials and leads to expressing the integral as a weighted sum of the function values, which we call *quadrature values*, evaluated at the *quadrature points*, which are chosen to be the zeros of an orthogonal polynomial.

In addition to showing how to compute the expansion via Gaussian quadrature, it was shown in [33] that skew symmetry of the commutators allows many of the terms in the quadrature to be combined. It was also shown via order analysis that if the quadrature points are chosen to be symmetric about $\frac{1}{2}h$, where h is width of interval between two adjacent collocation points, many of the terms in the quadrature can be dropped for a given order. We first provide here the steps to compute the quadrature on the interval $[0, h]$ (as given in [33]) then describe how this can be combined with the collocation method above.

To integrate between collocation point c_i and c_{i+1} , we choose the quadrature interpolating polynomials to be the Legendre polynomials shifted to $[0, 1]$ and choose ν quadrature points $\mathbf{t} = [t_1, \dots, t_\nu]^T \in [0, 1]$ to be the zeros of the Legendre polynomials since they are symmetric about $\frac{1}{2}$. We then form the quadrature values:

$$\mathbf{X}_k = h\mathbf{X}(c_i + t_k h), \quad k = 1, 2, \dots, \nu \quad (19)$$

Note that an order four quadrature requires $\nu \geq 2$ and an order six quadrature requires $\nu \geq 3$. In [33], [37], a change of basis is carried out to take advantage of the symmetry of the Magnus expansion. The change of basis is done by finding the solution of the following Vandermonde system:

$$\sum_{i=1}^{\nu} \left(t_k - \frac{1}{2}\right)^{i-1} \mathbf{Y}_i = \mathbf{X}_k \quad (20)$$

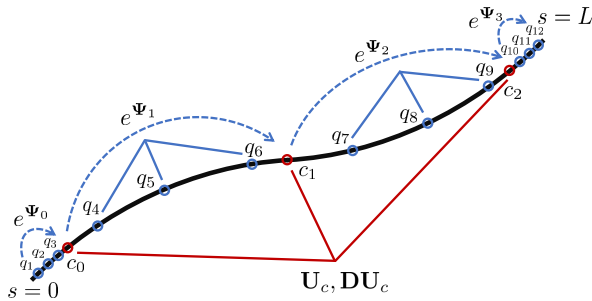


Fig. 3: Collocation points and quadrature points for the case of $n = 2$ and $\nu = 3$. Using orthogonal polynomials and their zeros results in quadrature values being linearly related to the collocation values.

where $\mathbf{Y}_i \in se(3)$ are solved for by inversion:

$$\mathbf{V}_{ij} = (t_i - \frac{1}{2})^{j-1}, \quad \mathbf{Y}_i = \sum_{j=1}^{\nu} (\mathbf{V}^{-1})_{ij} \mathbf{X}_j \quad (21)$$

This leads to a quadrature rule for an order four expansion:

$$\Psi^{[4]}(h) = \mathbf{Y}_1 + \frac{1}{12}[\mathbf{Y}_1, \mathbf{Y}_2] \quad (22)$$

and the following quadrature rule for a sixth order expansion:

$$\begin{aligned} \Psi^{[6]} = & \mathbf{Y}_1 + \frac{1}{12}\mathbf{Y}_3 + \frac{1}{12}[\mathbf{Y}_1, \mathbf{Y}_2] \\ & - \frac{1}{240}[\mathbf{Y}_2, \mathbf{Y}_3] + \frac{1}{360}[\mathbf{Y}_1, [\mathbf{Y}_1, \mathbf{Y}_3]] \\ & - \frac{1}{240}[\mathbf{Y}_2, [\mathbf{Y}_1, \mathbf{Y}_2]] - \frac{1}{720}[\mathbf{Y}_1, [\mathbf{Y}_1, [\mathbf{Y}_1, \mathbf{Y}_2]]] \end{aligned} \quad (23)$$

Note that in [33] the expansion is given for the form of $\mathbf{T}'(s) = \mathbf{X}(s)\mathbf{T}(s)$, which corresponds to twists expressed with respect to the world frame, but since we use the body twist in (4) there is a difference in sign for some terms.

We propose finding the poses at the collocation points $\mathbf{T}(c_i)$ using a Magnus expansion step between each collocation point. We assign either $\nu = 2$ or $\nu = 3$ quadrature points (for either fourth or sixth order expansions, respectively) between each pair of collocation points as well as between 0 and c_0 and between c_n and L . This leads to a total of $m = \nu(n+2)$ quadrature points along the length of the rod. Fig. 3 shows an example of this for $\nu = 3$ and $n = 2$. We then compute a Magnus expansion between each collocation point, starting from $s = 0$ and stepping to $s = L$. The frame at each collocation point therefore given by a product of exponentials where Ψ_i is expressed in frame $\mathbf{T}(c_i)$:

$$\mathbf{T}(c_k) = \mathbf{T}_0 \prod_{i=0}^k e^{\Psi_i} \quad (24)$$

Although not detailed here, we note that since the shape of the rod is given as a product of exponentials, closed-form gradients of the residual vector \mathbf{e} can be found which facilitate faster solutions to (13). Furthermore, as shown in [17], [22] Jacobians useful for studying the kinematics of continuum manipulators can be found via (24).

V. DETERMINING CURVATURE AT QUADRATURE POINTS

Evaluating Ψ_i according to the quadrature formula above ((22) or (23)) requires knowing the values at the quadrature points, $\tilde{\mathbf{u}}(q_k)$ where $q_k = c_i + t_k h$, $k = 1 \dots \nu$. Given the collocation values $\tilde{\mathbf{u}}(c_i)$ (which are guessed at each iteration of a nonlinear solver), we show here that the function values of the interpolating polynomial at the quadrature points can be found directly as a linear combination of the collocation values. First, we define a matrix \mathbf{U}_q that contains quadrature values, similar to \mathbf{U}_c :

$$\mathbf{U}_q = \begin{pmatrix} \mathbf{u}^T(q_1) \\ \vdots \\ \mathbf{u}^T(q_m) \end{pmatrix} \quad (25)$$

We then use (8) together with (9) to find a matrix relating the collocation values and the quadrature values:

$$\mathbf{U}_q = \mathbf{A}\mathbf{B}\mathbf{U}_c \quad (26)$$

where the matrix $\mathbf{B} \in \mathbb{R}^{(n+1) \times (n+1)}$ transforms the collocation values into modal coefficients which are then transformed by $\mathbf{A} \in \mathbb{R}^{m \times (n+1)}$ into quadrature values:

$$\mathbf{A} = \begin{pmatrix} \frac{1}{2}T_0(q_1) & T_1(q_1) & \dots & T_n(q_1) \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{2}T_0(q_m) & T_1(q_m) & \dots & T_n(q_m) \end{pmatrix} \quad (27)$$

$$\mathbf{B} = \frac{2}{n+1} \begin{pmatrix} T_0(c_0) & \dots & T_0(c_n) \\ \vdots & \vdots & \vdots \\ T_n(c_0) & \dots & T_n(c_n) \end{pmatrix} \quad (28)$$

This same procedure is also used to find $\mathbf{u}(L)$ which is necessary to compute the error residual for the boundary condition in (13). Note that both \mathbf{A} and \mathbf{B} can be computed offline as long as the collocation and quadrature points are chosen beforehand.

VI. CONVERGENCE OF THE MAGNUS EXPANSION

One important note is that the infinite Magnus expansion is not guaranteed to converge. In this section, we use known sufficient conditions for convergence of the Magnus expansion to provide the maximum step sizes that will guarantee convergence. These bounds are not necessary conditions, so we are able to violate the bounds in our simulation results, but we discuss convergence here since it cannot in general be guaranteed. For real matrices, the expansion converges in the 2-norm provided that [38]:

$$\int_0^h \|\mathbf{X}(\xi)\|_2 d\xi < \pi \quad (29)$$

Here we relate this bound on convergence to the case of the Cosserat rod and provide a bound on the maximum step size as a function of the maximum curvature. Recall that the Euclidian 2-norm is bounded by the Frobenius norm:

$$\|\mathbf{X}\|_2 \leq \|\mathbf{X}\|_F = \sqrt{\text{trace}(\mathbf{X}^T \mathbf{X})} \quad (30)$$

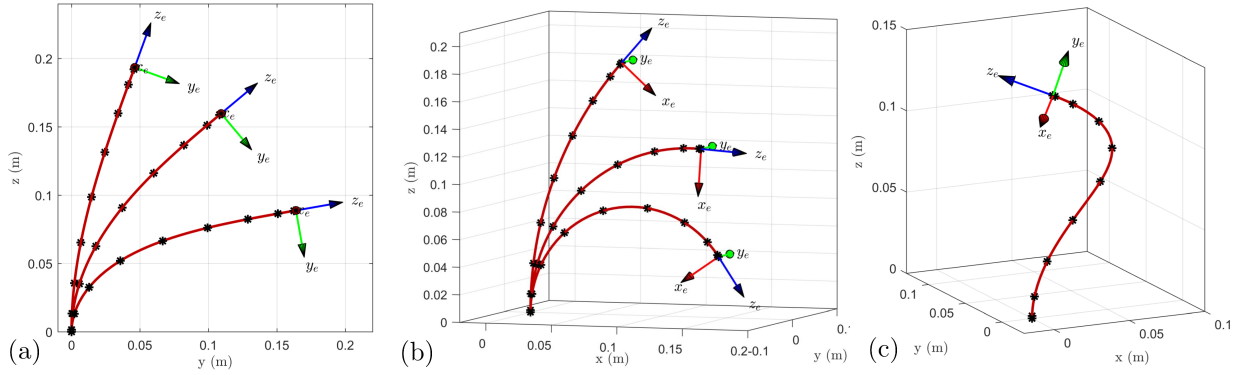


Fig. 4: Examples comparing the shooting method and our collocation approach for (a) force loads (which we also validate with elliptic integrals) (b) moment loads which result in constant-curvature, and (c) combined force and moment.

Assume the curvatures $\mathbf{u}(s)$ are bounded by a known scalar, i.e. $u_x \leq u_y \leq u_z \leq \beta$. We can then compute the Frobenius norm explicitly and provide the following bound:

$$\|\mathbf{X}\|_F = \sqrt{2u_x^2 + 2u_y^2 + 2u_z^2 + 1} \leq \sqrt{6\beta^2 + 1} \quad (31)$$

We then integrate (31) to find a conservative bound for integration step h to guarantee convergence:

$$\int_0^h \|\mathbf{X}(\xi)\|_2 d\xi \leq \int_0^h \|\mathbf{X}(\xi)\|_F d\xi < \pi \quad (32)$$

$$h_{max} < \frac{\pi}{\sqrt{6\beta^2 + 1}} \quad (33)$$

Given a particular task and continuum robot architecture, the bound β might be known beforehand through a simulation study. Another option is to choose β by considering the strain limits of the rod material. Consider as an example a superelastic nickel-titanium (Nitinol) rod, which can accept a strain of 5% with minimal loss of superelasticity due to cyclic fatigue [39]. The maximum bending strain is $\epsilon = ur$, where u is the rod's curvature and r is the rod's radius. Assuming the continuum robot is designed to avoid violating strain limits, we have $u_x \leq u_y \leq u_z \leq \epsilon/r = \beta$.

TABLE I: Maximum Step Size for Gauranteed Convergence with 6% Bending Strain

r (mm)	β	h_{max} (mm)
1	50	25.65
2	25	51.29
3	16.67	76.92
4	12.5	102.54

TABLE II: Step Sizes Used in Simulations (2 mm OD rod)

n	2	4	6	8	10
h_{max} (mm)	86.60	58.78	43.38	34.20	28.17

Table I shows the maximum step size that will guarantee convergence of the Magnus expansion under the assumption $\epsilon < 5\%$. For larger rod diameters the step sizes are not restrictive, and for smaller rod diameters the particular task can potentially be taken into account to determine a more

suitable bound β and allow larger step sizes. Online checks may be necessary to check for convergence of the expansion in cases where larger step sizes are used. In our simulations, we used the step sizes in Table II without observing issues.

VII. SIMULATION RESULTS

In this section, we will compare the accuracy of the fourth and sixth-order Magnus expansions with different numbers of collocation points. In all examples we use a shooting method as a ground-truth to compare our results against, since the shooting method has been validated in previous experimental studies [8], [9], [13]. We integrated (4) with a Runge-Kutta solver and a Levenberg-Marquardt algorithm (*ode45()* and *fsolve()* in MATLAB) to satisfy the boundary conditions with a termination tolerance of 10^{-9} . We used units of meters and radians. In the planar examples with in-plane forces, we also compared against the elliptic integral solution given in [40] and experimentally validated in [41], [42]. For all examples we simulated the case of a solid Nickel-titanium (Nitinol) rod with a 2 mm diameter and a 200 mm length.

We first consider the planar case with an in-plane load. We solved the elliptic integral equations in [40] with tip angles of 20° , 50° , and 80° , giving the position of the tip and resultant tip forces of $\mathbf{f}_e = [0, 1.04, 0.104]^T$ N, $\mathbf{f}_e = [0, 3.63, 0.362]^T$ N, and $\mathbf{f}_e = [0, 18.9, 1.89]^T$ N. We then used these forces in the shooting method and our collocation method (with a sixth order Magnus expansion and $n = 10$), both of which are shown in Fig. 4(a). All three methods showed agreement within 0.006 mm (0.003% of arc-length).

Next we consider a planar case with an in-plane moment load, as shown in Fig. 4(b). In this case, our collocation method returns the predicted constant-curvature shape, and for all three cases shown in Fig. 4(b) the shooting and collocation method (with a sixth order Magnus expansion and $n = 10$) showed agreement within 3.98e-6 mm and rotation error within machine precision.

The collocation method also agrees with the shooting method in cases with combined forces and moments, as shown in Fig. 4(c). In the next study, we compare the shooting and collocation methods for combinations of forces/moments and compare different numbers of collocation

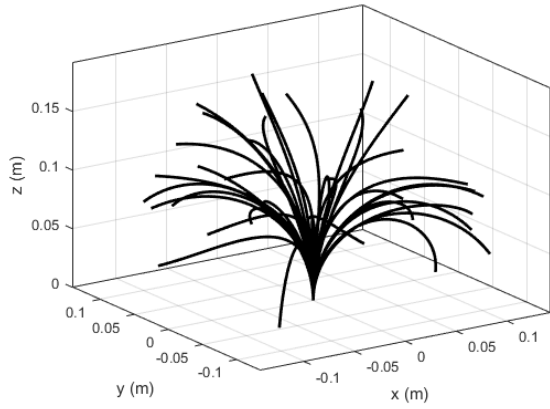


Fig. 5: Samples from the set of 729 rod shapes used when comparing our approach to the shooting method.

tion points. We simulated combinations of ± 1 N tip forces and ± 0.5 Nm moments applied to the tip of the rod (a total of 729 applied wrench cases). The solution speed of any BVP solver is affected by how close initial guesses are to the solution, so for each case, we begin the simulation with the rod in its straight configuration and linearly interpolate from a zero wrench to the final wrench in 3 steps. This is to more realistically simulate a scenario where the model is being continuously solved (e.g. kinematic control) and the previous solution is used as the initial guess for the solver. With the interpolated wrenches the total number of applied wrench cases is 2,187, samples of which are shown in Fig. 5. We solved for the equilibrium shape of the rod for each case with different numbers of collocation points, $n \in [2, 4, 6, 8, 10]$. We also solved each case using the fourth-order Magnus expansion ($\nu = 2$) and the sixth-order Magnus expansion ($\nu = 3$). The tip pose found with each collocation method was compared to the tip pose found with the shooting method using the following metrics:

$$e_p = \frac{\|\mathbf{p}_c(L) - \mathbf{p}_s(L)\|}{L} \times 100$$

$$e_r = \cos^{-1} \left(\frac{\text{trace}(\mathbf{R}_s(L)\mathbf{R}_c^T(L)) - 1}{2} \right) \quad (34)$$

where we report the position error as percent of total arc-length, $\mathbf{p}_c(L), \mathbf{R}_c(L)$ are the tip position/rotation found using a collocation method, and $\mathbf{p}_s(L), \mathbf{R}_s(L)$ are the tip poses found using the shooting method.

Tables III and IV show the results of the simulations. The difference between the shooting method and our method rapidly converges to zero with increasing n . Both the fourth-order and sixth-order expansions provided rapid convergence with increasing n , and for $n \leq 6$ are comparable. However, for $n \geq 8$ sixth-order Magnus steps did a better job reducing the e_p and e_r to below 0.005%. This implies that for small n , the primary source of the error is in the interpolation error in collocation (and not the Magnus stepping), while for larger n the error in the Magnus steps becomes more important. Any method with $n \geq 6$ was able to calculate the tip position with agreement with shooting of less than 0.15% of arc length.

We also report in Tables III and IV the average speed at

which solutions were found across the set of 2,187 shapes, running MATLAB 2019b on an i7-4770 3.4 GHz CPU. The fourth order and sixth order Magnus expansions were comparable in terms of speed, since the primary source of computation effort is not in evaluating the Magnus expansion but in computing the gradient of the error residual in (13), which despite being available in closed-form, becomes increasingly expensive as n increases. When n is not too large, the gradient is cheap to compute which provides quick solutions when combined with the efficient integration of $\mathbf{T}'(s)$ via the Magnus expansion.

TABLE III: Fourth Order Magnus Tip Error as a Function of Collocation Polynomial Order ($L = 200$ mm)

	Pos. e_p (%)		Rot. e_r (deg)		Speed (Hz)
	Avg.	Max.	Avg.	Max.	
$n = 2$	2.97	28.0	4.28	36.3	179.6
$n = 4$	0.141	2.15	0.235	3.78	112.1
$n = 6$	0.00573	0.147	0.00889	0.183	71.6
$n = 8$	0.00122	0.0173	0.00453	0.0571	46.3
$n = 10$	5.46e-4	0.00707	0.00448	0.0543	33.1

TABLE IV: Sixth Order Magnus Tip Error as a Function of Collocation Polynomial Order ($L = 200$ mm)

	Pos. e_p (%)		Rot. e_r (deg)		Speed (Hz)
	Avg.	Max.	Avg.	Max.	
$n = 2$	3.00	28.1	4.29	36.5	176.8
$n = 4$	0.140	2.26	0.234	3.79	106.2
$n = 6$	0.00467	0.115	0.00889	0.193	68.8
$n = 8$	1.95e-4	0.00493	0.00450	0.0553	42.5
$n = 10$	2.66e-5	0.00140	0.00448	0.0542	32.4

Our shooting method implementation in MATLAB solved the BVPs at an average rate of 17.6 Hz (with residual gradients estimated via finite differences). Our collocation implementation was faster than the shooting method in all examples, but we would like to stress that a fair comparison between the shooting and collocation is difficult since either method could be improved with more specialized implementations in pre-compiled code and by providing good initial guesses (see [5], [13] for shooting method implementations with >1 kHz). The main takeaway from these results is that our method can be competitive in terms of computation speed, and due to the availability of closed-form expressions, may improve speed in cases where lower-order polynomials are sufficient. More importantly, the Magnus expansion leading to closed-form direct kinematics offers fast evaluation of instantaneous direct kinematics Jacobians.

VIII. CONCLUSIONS

In this paper, we presented a numerical approach to solving the Cosserat rod BVP that combines orthogonal collocation and the Magnus expansion which, when solved, results in a closed-form product of matrix exponentials equation. We have discussed the convergence of the Magnus expansion for the case of Cosserat rods and showed in simulation that both the fourth order and sixth order Magnus expansions provide accurate solutions to the BVP and that a small

number of collocation points can provide reasonably accurate results. Although our results are preliminary, we believe this method has the potential to provide reduced computational cost when solving Cosserat rod models. Future work will study Jacobian-based analysis of the resulting product of exponentials and implementations for robotic applications.

REFERENCES

- [1] 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.
- [2] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [3] R. J. Webster and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.
- [4] S. H. Sadati, S. E. Naghibi, A. Shiva, B. Michael, L. Renson, M. Howard, C. D. Rucker, K. Althoefer, T. Nanayakkara, S. Zschaler, C. Bergeles, H. Hauser, and I. D. Walker, "TMTDYN: A Matlab package for modeling and control of hybrid rigid–continuum robots based on discretized lumped systems and reduced-order models," *The International Journal of Robotics Research*, Jan. 2020.
- [5] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 209–225, 2009.
- [6] D. C. Rucker, B. A. Jones, and R. J. W. III, "A Geometrically Exact Model for Externally Loaded Concentric-Tube Continuum Robots," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 769–780, Oct. 2010.
- [7] D. C. Rucker and R. J. Webster, "Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1033–1044, Dec. 2011.
- [8] M. T. Chikhaoui, S. Lilje, S. Kleinschmidt, and J. Burgner-Kahrs, "Comparison of Modeling Approaches for a Tendon Actuated Continuum Robot With Three Extensible Segments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 989–996, Apr. 2019.
- [9] A. L. Orekhov, V. A. Aloï, and D. C. Rucker, "Modeling parallel continuum robots with general intermediate constraints," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 6142–6149.
- [10] S. M. H. Sadati, S. E. Naghibi, I. D. Walker, K. Althoefer, and T. Nanayakkara, "Control Space Reduction and Real-Time Accurate Modeling of Continuum Manipulators Using Ritz and Ritz-Galerkin Methods," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 328–335, Jan. 2018.
- [11] F. Renda, M. Cianchetti, H. Abidi, J. Dias, and L. Seneviratne, "Screw-Based Modeling of Soft Manipulators with Tendon and Fluidic Actuation," *Journal of Mechanisms and Robotics*, Apr. 2017.
- [12] J. Till, V. Aloï, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 723–746, 2019.
- [13] 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," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5067–5074.
- [14] N. Simaan, "Snake-like units using flexible backbones and actuation redundancy for enhanced miniaturization," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3012–3017.
- [15] J. Wang, J. Ha, and P. E. Dupont, "Steering a multi-armed robotic sheath using eccentric precurved tubes," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9834–9840.
- [16] D. C. Rucker and R. J. Webster, "Computing jacobians and compliance matrices for externally loaded continuum robots," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 945–950.
- [17] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete Cosserat Approach for Multisection Soft Manipulator Dynamics," *IEEE Transactions on Robotics*, pp. 1–16, 2018.
- [18] F. Renda, C. Armanini, V. Lebastard, F. Candelier, and F. Boyer, "A Geometric Variable-Strain Approach for Static Modeling of Soft Manipulators With Tendon and Fluidic Actuation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4006–4013, Jul. 2020.
- [19] O. Weeger, S.-K. Yeung, and M. L. Dunn, "Isogeometric collocation methods for cosserat rods and rod structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 100–122, 2017.
- [20] Y. Liu, M. Wang, and L. Meng, "Spectral Collocation Method in the Large Deformation Analysis of Flexible Beam," p. 8, 2018.
- [21] P. Khaneh Masjedi and H. R. Ovesy, "Chebyshev collocation method for static intrinsic equations of geometrically exact beams," *International Journal of Solids and Structures*, vol. 54, pp. 183–191, Feb. 2015.
- [22] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A Geometrically Exact Model for Soft Continuum Robots: The Finite Element Deformation Space Formulation," *Soft Robotics*, Nov. 2018.
- [23] C. D. Santina and D. Rus, "Control Oriented Modeling of Soft Robots: The Polynomial Curvature Case," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 290–298, Apr. 2020, conference Name: IEEE Robotics and Automation Letters.
- [24] P. Češarek, M. Saje, and D. Zupan, "Dynamics of flexible beams: Finite-element formulation based on interpolation of strain measures," *Finite Elements in Analysis and Design*, vol. 72, pp. 47–63, 2013.
- [25] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: a strain parametrization based approach," *Preprint, submitted to the IEEE Transactions on Robotics*, Oct 2019, <https://hal.archives-ouvertes.fr/hal-02318617>.
- [26] C. B. Black, J. Till, and D. C. Rucker, "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 29–47, 2017.
- [27] L. C. Young, "Orthogonal collocation revisited," *Computer Methods in Applied Mechanics and Engineering*, vol. 345, pp. 1033–1076, 2019.
- [28] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods: fundamentals in single domains*. Springer Science & Business Media, 2007.
- [29] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*. Chapman and Hall/CRC, 2002.
- [30] J. Villadsen and M. L. Michelsen, *Solution of differential equation models by polynomial approximation*. Prentice-Hall Englewood Cliffs, NJ, 1978, vol. 7.
- [31] C. Rucker, "Integrating rotations using nonunit quaternions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2979–2986, 2018.
- [32] J. Park and W.-K. Chung, "Geometric integration on euclidean group with application to articulated multibody systems," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 850–863, 2005.
- [33] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie-group methods," *Acta numerica*, vol. 9, pp. 215–365, 2000.
- [34] A. Müller, "Approximation of finite rigid body motions from velocity fields," *ZAMM - Journal of Applied Mathematics and Mechanics*, vol. 90, no. 6, pp. 514–521, Jun. 2010.
- [35] J. M. Selig, *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004.
- [36] W. Magnus, "On the exponential solution of differential equations for a linear operator," *Communications on pure and applied mathematics*, vol. 7, no. 4, pp. 649–673, 1954.
- [37] H. Munthe-Kaas and B. Owren, "Computations in a free lie algebra," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 357, no. 1754, pp. 957–981, 1999.
- [38] P. C. Moan and J. Niesen, "Convergence of the magnus series," *Foundations of Computational Mathematics*, vol. 8, no. 3, pp. 291–301, 2008.
- [39] M. G. de Azevedo Bahia, R. F. Dias, and V. T. L. Buono, "The influence of high amplitude cyclic straining on the behaviour of superelastic niti," *International journal of fatigue*, vol. 28, no. 9, pp. 1087–1091, 2006.
- [40] L. L. Howell and A. Midha, "Parametric deflection approximations for end-loaded, large-deflection beams in compliant mechanisms," 1995.
- [41] K. Xu and N. Simaan, "Analytic formulation for kinematics, statics, and shape restoration of multibackbone continuum robots via elliptic integrals," *Journal of Mechanisms and Robotics*, vol. 2, no. 1, 2010.
- [42] W. Wei and N. Simaan, "Modeling, Force Sensing, and Control of Flexible Cannulas for Microstent Delivery," *Journal of Dynamic Systems, Measurement, and Control*, vol. 134, no. 4, pp. 041004–041004–12, Apr. 2012.