



**UNIVERSITÉ
DE LORRAINE**

**BIBLIOTHÈQUES
UNIVERSITAIRES**

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Simulation of Intravascular Catheter Navigation Using Cosserat Rod Theory

THÈSE

présentée et soutenue publiquement le 24 juin 2025

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Radhouane Jilani

Composition du jury

<i>Président :</i>	M. Sylvain Contassot-Vivier	Professeur des universités – Université de Lorraine
<i>Rapporteurs :</i>	M. Vincent Lebastard	Maître assistant – Institut Mines-Télécom
	M. Kanty Rabenorofoa	Professeur des universités – Université de Franche-Comté
<i>Examineurs :</i>	M. Jérémie Dequidt	Professeur des universités – Université de Lille
	Mme Maud Marchal	Professeur des universités – Université de Rennes
<i>Invité :</i>	M. Stéphane Cotin	Directeur de recherche – Inria
<i>Encadrants :</i>	M. Erwan Kerrien	Chargé de recherche – Inria
	M. Pierre-Frédéric Villard	Maître de conférences – Université de Lorraine

Acknowledgments

I would like to express my gratitude to Dr. Erwan Kerrien, my thesis supervisor, and to Dr. Pierre-Frédéric Villard, my co-supervisor, for their invaluable guidance throughout my PhD. Their advice was essential to the completion of this work.

I sincerely thank Dr. Vincent Lebastard and Prof. Kanty Rabenoroso for kindly agreeing to serve as reviewers of this manuscript. I am also grateful to Prof. Jérémie Dequidt and Prof. Maud Marchal for serving as examiners, and to Prof. Stéphane Cotin for his participation as an invited member of the jury. Finally, I warmly thank Prof. Sylvain Contassot-Vivier, chair of the jury and a member of my thesis advisory committee, for his valuable insights.

Abstract

Ischemic stroke, caused by a blood clot blocking blood flow to the brain, is a leading cause of death, responsible for approximately three million fatalities worldwide each year. Timely intervention plays a key role in clinical outcomes, as the shorter the time between symptom onset and treatment, the higher the chances of recovery and the lower the risk of long-term disability or lasting effects.

The standard treatment is endovascular thrombectomy, a procedure in which a physician manually steers a slender and flexible tool called a catheter from its proximal end, typically inserted through the femoral artery, to navigate the vascular system and retrieve the blood clot from the brain. This procedure is mechanically challenging, requiring a high level of practitioner expertise. Difficulties arise both from the physical complexity of manually steering the catheter through tortuous, narrow vessels and from the variability in vascular anatomy between patients.

These challenges highlight the need for computational tools that can assist in training and preoperative planning, with the goal of reducing both planning and intervention time. This work aims to develop a robust and time-efficient numerical simulation of catheter navigation that maintains good physical fidelity. Such a simulation could support physicians by enabling them to determine in advance which vascular path to follow or which type of catheter to use. The simulation models the catheter, the steering at its proximal end, the vascular geometry, and the frictional contact between the catheter and the vessel walls.

To achieve a balance between physical fidelity and modeling complexity, the catheter is modeled using the one-dimensional geometrically exact Cosserat rod, which is well suited to capture the large deformations and rotations present in catheter navigation. Though well suited for the task, simulating catheter behavior using this model presents several numerical challenges. Solving the Cosserat rod equations in a robust and time-efficient way is nontrivial, particularly when accounting for contact and friction. The shooting method, while commonly used to solve Cosserat boundary value problems due to its high time efficiency, becomes unstable and fails to converge for soft catheters and small time steps because of the singular nature of the problem in the dynamic regime. Alternatively, the strain parameterization in the Lagrangian framework is known for its robustness and its ability to significantly reduce the number of degrees of freedom, but it can lead to stiff systems that require prohibitively small time steps for integration.

This work addresses these challenges in two complementary ways. First, a dynamic simulation based on an orthogonal collocation method with Chebyshev polynomials is proposed. We demonstrate through numerical experiments that this approach robustly addresses the inherent singularity in the boundary value problem of dynamic Cosserat rods.

In particular, we show that in every difficult scenario where the shooting method fails, increasing the polynomial degree beyond a certain threshold ensures convergence of the collocation to a correct solution. This method is extended to include frictional contact and tested for catheter navigation inside a model of a real patient’s carotid artery. While robust, this approach remains computationally intensive.

To improve efficiency, a quasi-static assumption is introduced, justified by the spatial confinement of the catheter within the vascular network. In this regime, a Lagrangian strain parameterization is used. To address the stiffness of the underlying equations, we model the blood vessel walls using implicit functions, which enable the use of implicit time integration. Thanks to the smoothness of these implicit functions, the Jacobian matrix can be derived analytically. This approach led to a 100x speed improvement over explicit integration and a 10x improvement over the collocation method.

Table of Contents

1	Introduction	1
1.1	Interventions for Ischemic Stroke	1
1.2	Simulation Objectives	2
1.3	Simulation Constraints	3
1.4	PreSPIN Project	4
1.5	Contributions	5
2	State-of-the-Art Rod Modeling	7
2.1	Rod Modeling	7
2.1.1	Three-Dimensional Continuum Mechanics Model	8
2.1.2	One-Dimensional Continuum Mechanics Models	8
2.1.3	Constant Curvature Model	10
2.1.4	Pseudo-Rigid Body Model	11
2.1.5	Mass-Spring Model	11
2.1.6	Data-Driven Models	12
2.2	Selection of a Rod Model for Catheters	13
2.3	Notational Conventions	13
2.4	Cosserat Rod Model	15
2.5	State-of-the-Art Methods for Solving Cosserat Rods	18
2.5.1	Finite Differences	18
2.5.2	Boundary Value Problem	20
2.5.3	Reduction-Based Methods	25
2.6	Discussion and Conclusion	27
3	Addressing the Singularity in the Dynamic Cosserat Rods Boundary Value Problem	29
3.1	Time Discretization	30
3.2	Chebyshev Interpolation	31
3.3	Configuration Space Reduction	33
3.4	Collocation Method	34
3.4.1	System of Nonlinear Algebraic Equations	35
3.4.2	Jacobian Matrix	35
3.5	Numerical Applications	38
3.5.1	Statics	38
3.5.2	Dynamics	42

3.6	Conclusion	46
4	Dynamic Cosserat Rods in Contact with Implicit Surfaces	49
4.1	Time Discretization	50
4.2	Collocation Method	52
4.3	Implicit Surface Modeling	53
4.4	Contact Handling	56
4.4.1	Contact Detection	56
4.4.2	Contact Response	57
4.5	Including Contact Forces into the Boundary Value Problem	58
4.6	Discretization of Contact Forces	59
4.6.1	Concentrated to Distributed Force Conversion	59
4.6.2	Space Discretization	59
4.6.3	Time Discretization	60
4.7	Numerical Applications	61
4.7.1	Elastic Longitudinal Contact	61
4.7.2	Frictional Contact with a Helical Tube	62
4.7.3	Catheter Insertion into a Carotid Artery	64
4.8	Conclusion	66
5	Quasi-Static Cosserat Rods in Contact with Implicit Surfaces	69
5.1	Lagrangian Model	70
5.1.1	Configuration Space Reduction	70
5.1.2	Balance Equation	71
5.1.3	Quasi-static Solving	72
5.2	Contact Handling	73
5.2.1	Contact Detection	73
5.2.2	Contact Response	74
5.3	Including Contacts Forces into the Balance Equation	75
5.4	Jacobian Matrix	76
5.5	Insertion and Removal Approach	78
5.6	Numerical Applications	78
5.6.1	Rod Push-Pull in a Sphere	79
5.6.2	Rod Insertion and Removal in a Helical Tube	81
5.6.3	Catheter Insertion into a Carotid Artery	83
5.7	Conclusion	84
6	Conclusion and Perspectives	87
6.1	Summary of Contributions	87
6.2	Perspectives	89
6.2.1	Estimating the Reference Curvature	90
6.2.2	Physics-Informed Neural Networks	91
	Bibliography	97

Appendices	108
A Résumé étendu en français	109
A.1 Introduction	109
A.2 État de l’art de la modélisation des poutres	111
A.3 Résolution de la singularité dans le problème aux limites des poutres de Cosserat dynamiques	113
A.4 Poutres de Cosserat dynamiques en contact avec des surfaces implicites	114
A.5 Poutres de Cosserat quasi-statiques en contact avec des surfaces implicites	116
A.6 Conclusion et perspectives	117

List of Tables

3.1	Position error, total execution time, and average execution time per loading step for the collocation method, as a function of the polynomial degree m . For comparison, the shooting method required 0.598 minutes in total, with an average of 0.119 seconds per loading step.	40
3.2	Position error, total execution time, and average execution time per loading step for the collocation method, as a function of the polynomial degree m . For comparison, the shooting method required 8.95 seconds in total, with an average of 0.0895 seconds per loading step.	41
3.3	Error $e_\Lambda(m, m + 10)$ between Chebyshev approximations of the stress field computed using polynomial degrees m and $m + 10$, over 100 loading steps. The error is averaged over time steps and integrated along the rod length. Values are reported for three representative stress components (K_3 , Γ_1 , and Γ_2) in a two-dimensional setting. The results show a rapid decay of the error, confirming the convergence of the collocation method, with negligible improvement beyond $m = 60$	42
5.1	Performance comparison of implicit analytical, implicit numerical, and explicit integration methods for various damping coefficients $\tilde{\mu}$. Time (in seconds, minutes, or hours), nfev (number of function evaluations), and njev (number of Jacobian evaluations) are reported. Explicit methods do not compute Jacobians, hence njev is omitted. Best performances (lowest time, nfev, or njev) are highlighted in bold.	82

List of Figures

1.1	Illustration comparing normal and blocked blood flow in a blood vessel [1]. The top part shows smooth and uninterrupted flow in a healthy vessel. The bottom part shows how a blood clot (thrombus) can block the vessel, reducing the amount of blood that reaches nearby tissues. This kind of blockage can damage surrounding tissue and often needs to be treated quickly using procedures like endovascular thrombectomy.	2
1.2	Stent retriever after successfully removing a blood clot [2]. The device is designed to expand and catch the clot inside a blocked blood vessel. In this image, the clot is still visible inside the metal mesh. This technique is commonly used in endovascular thrombectomy to restore blood flow caused by a blocked artery.	3
1.3	Visualization of the brain vascular network [3]. The dense and intricate vessel pathways illustrate the anatomical complexity clinicians must navigate during endovascular thrombectomy.	4
1.4	Overview of the PreSPIN project pipeline. From the patient arrival to post-operative imaging, PreSPIN focuses on the planning phase of ischemic stroke treatment using patient-specific data. It includes vessel extraction from magnetic resonance images, simulation of catheter navigation, and simulation of perfusion imaging.	5
2.1	Pseudo-rigid body model consisting of rigid links connected by revolute joints, providing a piecewise-rigid approximation of a deformable structure.	10
2.2	Discrete mass-spring-damper representation of a rod. The rod is modeled as a series of point masses m_i connected by springs with stiffness k_i and dampers with damping coefficient c_i , capturing its elastic and dissipative properties.	11
2.3	Fully connected feedforward neural network. This figure is provided to support readers unfamiliar with neural networks and illustrates the standard structure and notation of a multilayer perceptron. The input vector $x = (x_1, x_2)^T$ is propagated through a sequence of hidden layers, where each layer computes $h^{(l)} = \sigma(W^{(l)}h^{(l-1)} + b^{(l)})$. The network produces the final output $\hat{y} = \phi(W^{(L)}h^{(L-1)} + b^{(L)})$. Here, $h^{(l)}$ is the output of layer l , $W^{(l)}$ and $b^{(l)}$ are the corresponding weights and biases, σ is the hidden-layer activation function, ϕ is the output activation function, and L is the index of the output layer.	12

2.4	A Cosserat rod defined over the arc length domain $X \in [0, l]$. The centerline of the rod is depicted by the dashed black curve. At each point X , the configuration is given by a rigid body transformation $g(X) \in SE(3)$, encoding both the position of the centerline and the orientation of the cross-section.	15
2.5	Visualization of Cosserat rod deformation modes: (a) extension, (b) bending, (c) shear, and (d) torsion.	16
2.6	Lang et al. [4] introduced a rod material discretization in which positions r are defined at m nodes, whereas quaternions Q are defined at segment midpoints. The kinematics (2.3) are discrete, and the angular (K) and linear (Γ) rates of change are computed using finite differences with $r(X_i)$ and $Q(X_{i+1/2})$	18
2.7	Cubic B-spline basis functions defined over the interval $[-1, 1]$. The basis has degree 3 and consists of 7 functions, defined over the open knot vector $\{-1, -1, -1, -1, -0.5, 0, 0.5, 1, 1, 1, 1\}$. Each curve $B_i(x)$ represents a basis function associated with one control point. The functions are C^2 -continuous at interior knots.	19
2.8	The figure depicts the clamped-free (Dirichlet-Neumann) boundary conditions of a Cosserat rod subjected to distributed loads $\bar{F}$ and a concentrated wrench at the tip (F_+). At the proximal end $X = 0$, the known quantities are the pose g_0 , as well as the velocity η_0 and acceleration $\dot{\eta}_0$; at the distal end $X = l$, the stress $\Lambda(l) = F_+$ is known.	20
2.9	The figure depicts the free-free (Neumann-Neumann) boundary conditions of a Cosserat rod subjected to distributed loads $\bar{F}$ and two concentrated wrenches at the base (F_-) and the tip (F_+). The stress is specified at both ends: at the proximal end $X = 0$, the stress is prescribed as $\Lambda(0) = -F_-$; at the distal end $X = l$, the stress is given by $\Lambda(l) = F_+$	21
2.10	Solution to a clamped-free boundary value problem obtained using the shooting method. The distal boundary condition for the internal moment is set to $C(l) = C(1) = 0$. The shooting method computes the proximal value of the internal moment $C(0)$ that, after arc length integration, satisfies the boundary condition. The solution $C(0) = 31$ is found after four iterations: $C(0) = \{50, 16.5, 36.9, 31\}$	22
2.11	First five Chebyshev polynomials of the first kind $T_m(x)$ for degree $m = 0$ to 4, plotted over the interval $[-1, 1]$. These polynomials are orthogonal with respect to the weight function $w(x) = (1 - x^2)^{-1/2}$, meaning they satisfy the relation $\int_{-1}^1 T_m(x) T_n(x) w(x) dx = 0$ for $m \neq n$. Each polynomial $T_m(x)$ has m distinct zeros in $(-1, 1)$, given by $c_i = \cos\left(\frac{2i-1}{2m}\pi\right)$ for $i = 1, \dots, m$	24
2.12	First five Legendre polynomials $P_n(x)$ for degree $n = 0$ to 4, plotted over the interval $[-1, 1]$. These polynomials are orthogonal with respect to the weight function $w(x) = 1$, meaning they satisfy $\int_{-1}^1 P_m(x) P_n(x) dx = 0$ for $m \neq n$. Each polynomial $P_n(x)$ has n distinct zeros in $(-1, 1)$	25
3.1	Chebyshev basis and their roots for multiple polynomial degrees.	31
3.2	A straight rod is bent by applying a tip moment until it forms a circle.	38
3.3	A straight rod is deformed into a helix by imposing tip force and moment.	39
3.4	Rod bending with a follower tip force.	41

3.5	A tip force is applied to a straight rod for 0.5 seconds. The figures show tip position over time using $\Delta t = 0.002$ s. The collocation and shooting methods are in good agreement. However, due to the large time step, noticeable differences are observed when both methods are compared to the Lagrangian model.	43
3.6	A tip force is applied to a straight rod for 0.5 seconds. The figures show tip position over time using $\Delta t = 0.0005$ s.	44
3.7	A straight rod is initially bent with a tip force and is then released for 0.3 seconds.	45
3.8	A rubber rod is released in gravity for 0.85 seconds.	46
3.9	A rubber rod is released in gravity for 0.85 seconds. The figure shows tip position over time.	46
4.1	Left: the Cauchy profile and its derivative. Right: weighted Cauchy profile $\varrho \Psi\left(\frac{x}{\varrho}\right)$ shown with three different widths ϱ	55
4.2	Left: visualization of a surface (black) defined by the iso-level $\zeta = 0$, generated from two blobs centered at $\sigma_1 = (0, 0, 0)^T$ and $\sigma_2 = (0, 5.5, 0)^T$, with respective radii $\varrho_1 = 1$ and $\varrho_2 = 0.7$, and using a threshold $S = 0.08$. The field normals w are shown at discretized locations. Right: 3D view of the scalar field $\zeta(r)$, with a planar cross-section at $z = 0$ visualized in false colors. The field is normalized, with red indicating the most negative values. The surface (in grey) again corresponds to the iso-level $\zeta = 0$, enclosing the region influenced by the two blobs.	56
4.3	A soft rod moving at initial velocity V_0 comes into elastic longitudinal contact with a plane.	62
4.4	A soft rod moving at an initial velocity of $5 \text{ m} \cdot \text{s}^{-1}$ comes into elastic longitudinal contact with a plane and rebounds with a velocity of $-5 \text{ m} \cdot \text{s}^{-1}$. The figure plots tip velocity over tip position.	63
4.5	A rubber rod is inserted into a helical tube. Initial configuration at $t = 0$. . .	64
4.6	Snapshots at different times and varying friction coefficients show a rubber rod inserted into a helical tube.	64
4.7	A rubber rod is inserted into a helical tube. Tip position over time for various coefficients of friction μ	65
4.8	Catheter is inserted into a carotid artery model. From left to right and top to bottom, the figure shows five snapshots taken every 0.005 seconds.	66
4.9	Catheter is inserted into a carotid artery model. The figure shows tip position over time.	67
4.10	A catheter is inserted into a carotid artery model. From left to right and top to bottom, snapshots are shown at 0.25, 0.5, 0.75, and 1 s. The simulation is solved using the shooting method with damping and modified catheter material properties. The damping effect can be observed at $t = 0.5$ s, as the catheter inaccurately retains its curvature after the turn.	68
5.1	The activation function of contact forces. Collisions happen when the contact distance $\phi > 0$	74

5.2	The rod base is pushed then pulled in a sphere (black line). This figure displays the rod initial configuration.	79
5.3	The rod base is pushed then pulled in a sphere. This figure displays the rod tip position along the z-axis with respect to time for different damping coefficients $\tilde{\mu}$	80
5.4	Snapshots at different times and varying damping coefficients $\tilde{\mu}$ show a rod pushed into the surface of a sphere.	81
5.5	A rubber rod insertion in a helical tube. From left to right, and top to bottom, four snapshots taken at $t = \frac{T}{8}, \frac{T}{4}, \frac{3T}{8}$, and $\frac{T}{2}$, corresponding to $t = 375, 725, 1125$, and 1500 s, respectively.	83
5.6	A rubber rod is inserted into, then removed from, a helical tube. This figures display the evolution of the rod tip position over time. Left: $\frac{T}{\mu} = 1000$, and right: $\frac{T}{\mu} = 100$	83
5.7	A rubber rod is inserted into, then removed from, a helical tube. This figure shows the evolution of the contact force norm at the rod tip over time. . . .	84
5.8	A catheter is inserted in a carotid artery model of a patient. From left to right, four snapshots taken approximately at $t = 0.025, 0.064, 0.074$, and 0.1 s. In the second snapshot, the catheter is blocked by a small bump on the surface.	85
5.9	A catheter is inserted in a carotid artery model of a patient. The figure shows the catheter tip evolution over time and highlights the stick-slip effect. . . .	86
6.1	Comparison of the position components x , y , and z along the arc length between the shooting method (reference solution) and the PINN approximation. The PINN closely reproduces the rod configuration, showing good agreement with the reference solution across all position components.	93
6.2	Comparison of the internal moment components x , y , and z along the arc length between the shooting method (reference solution) and the PINN approximation. For the y and z components, the PINN captures the overall moment with good accuracy. In the x -component, the larger relative discrepancies observed are due to the much smaller magnitude of the moment x compared to y and z , making the error appear more significant.	94
6.3	Evolution of the total loss function during PINN training over 3000 epochs, shown on a logarithmic scale. The steady decrease confirms convergence, with the observed oscillations resulting from the learning rate schedule and optimizer behavior.	95

Variable	Description	Dim.	Unit
X	Arc length coordinate	1	m
t	Time coordinate	1	s
l	Total rod length (reference configuration)	1	m
$\bullet'$	Derivative w.r. t. X	—	—
$\dot{\bullet}$	Derivative w.r. t. t	—	—
$\widehat{\bullet}$	Maps from $\mathbb{R}^3$ to $so(3)$ or $\mathbb{R}^6$ to $se(3)$	—	—
$\bullet^\vee$	Maps from $so(3)$ to $\mathbb{R}^3$ or $se(3)$ to $\mathbb{R}^6$	—	—
$\bullet_o$	Quantity in the reference configuration	—	—
$\bullet_0$	Quantity evaluated at $X = 0$	—	—
$\bullet_i$	i -th component of a vector	—	—
$g = (R, r)$	Homogeneous transform of a section pose	4×4	$(-, m)$
R	Rotation matrix of the section	3×3	—
r	Position of the section centre	3	m
$\xi = (K^T, \Gamma^T)^T$	Space-rate twist (K : angular, Γ : linear)	6	$(m^{-1}, -)$
$\eta = (\Omega^T, V^T)^T$	Velocity twist (Ω : angular, V : linear)	6	$(s^{-1}, m/s)$
$\epsilon = \xi - \xi_o$	Strain vector	6	$(m^{-1}, -)$
$\Lambda = (C^T, N^T)^T$	Stress wrench (C : moment, N : force)	6	(Nm, N)
$F_+ = (C_+^T, N_+^T)^T$	External wrench at $X = l$	6	(Nm, N)
$F_- = (C_-^T, N_-^T)^T$	External wrench at $X = 0$	6	(Nm, N)
$\bar{F} = (\bar{C}^T, \bar{N}^T)^T$	Distributed external wrench	6	$(N, N/m)$
$f_+, f_-, \bar{f}$	Their inertial frame counterpart	6	(Nm, N)
$\bullet_c$	Related to contact response	—	—
$\mathcal{H}$	Hookean stiffness matrix	6×6	(N, Nm^2)
$\mathcal{M}$	Inertia matrix	6×6	kg m, kg/m
E	Young's modulus	1	Pa
G	Shear modulus	1	Pa
ν	Poisson's ratio	1	—
A	Cross-sectional area	1	m^2
I_x	Polar moment of area (torsion)	1	m^4
I_y, I_z	Second moments of area (bending)	1	m^4
ρ	Mass density	1	$kg\ m^{-3}$
Δt	Time step	1	s
Δt_c	Critical time step	1	s
χ	Dimensionless prefactor of the Δt_c law	1	—
a, b	Coefficients of the Newmark scheme	1	s^2, s
f_j, h_j	Predictors for $\xi, \dot{\xi}$ at step j	6	as $\xi, \dot{\xi}$
m	Polynomial degree	1	—
T_m	Chebyshev polynomial (first kind)	1	—

τ_i^*	i -th Chebyshev root in $(-1, 1)$	1	—
Φ	Vector interpolating collocation values	m	—
D_m^*	Chebyshev differentiation matrix	$m \times m$	—
ς	Linear transformation from the arc length domain to the Chebyshev polynomial domain	1	—
$\zeta(r)$	Scalar field ($\zeta = 0$ defines the surface)	1	—
$\phi(r)$	Signed distance	1	m
$w(r)$	Unit inward normal	3	—
λ	Penalty stiffness	1	N/m
$\tilde{a}(\phi)$	SiLU activation function	1	m
z	Sharpness parameter of $\tilde{a}$	1	—
$\Psi(x)$	Cauchy radial profile of a blob	1	—
$\psi(x)$	First derivative of Ψ	1	—
σ_j	Centre of the j -th blob	3	m
ϱ_j	Width of the j -th blob	1	—
S	Threshold of the implicit surface	1	—
m_b	Number of blobs in the surface model	1	—
m_c	Number of discrete contact nodes	1	—
μ	Coulomb friction coefficient	1	—
k	Total number of shape functions for the Lagrangian model	1	—
q	Vector of generalized strain coordinates	k	1
Φ	Shape function matrix	$6 \times k$	1
K_ϵ	Generalized stiffness	$k \times k$	1
D_ϵ	Generalized damping	$k \times k$	1
$\tilde{\mu}$	Dimensionless damping coefficient	1	—
J	Geometric Jacobian mapping $\dot{q} \mapsto \dot{\eta}$	$6 \times k$	1
Q_r	Generalized restoring forces	k	1
Q_e	Generalized external/contact forces	k	1

¹The unit depends on the shape functions.

Chapter 1

Introduction

1.1 Interventions for Ischemic Stroke

Ischemic stroke occurs when a blood vessel supplying the brain becomes blocked by a blood clot, thereby restricting the delivery of oxygen and nutrients to the affected region (see Fig. 1.1). Without fast treatment, this restriction can rapidly progress to irreversible brain damage, potentially resulting in long-term disability or even death. Prompt medical intervention is crucial, as treatments are most effective within the initial hours following symptom onset. Globally, ischemic stroke is responsible for approximately 3 million deaths annually, and the loss of around 63 million healthy life years due to disability [5].

Thrombolytic drugs, which dissolve clots to restore blood flow, are commonly used in early treatment. However, when medication alone is insufficient, endovascular thrombectomy provides a mechanical solution by physically removing the clot with a catheter. The catheter, a long, flexible, and steerable tube, is designed to navigate complex and narrow blood vessels while maintaining sufficient rigidity. Under real-time imaging, the practitioner manually guides the catheter from its proximal entry point, usually in the femoral artery, through the vascular network until its distal end reaches the clot. Once in place, devices such as stent retrievers (see Fig. 1.2) or aspiration catheters are deployed to extract or fragment the clot, restoring blood flow.

Catheter navigation in endovascular thrombectomy is mechanically difficult due to the complex and variable nature of the vascular network (see Fig. 1.3). Blood vessels vary in size, requiring the catheter to transition smoothly from large to smaller arteries without causing damage. Additionally, anatomical variations, such as loops and sharp curves, introduce further unpredictability. Because the catheter remains in contact with the vascular network, rotational movements applied at the proximal end do not always translate accurately to the distal tip. While real-time imaging provides guidance, interpreting it and translating it into precise hand movements requires significant practitioner expertise.

To assist physicians in preoperative planning and training, this work aims to develop a numerical simulation of catheter navigation tailored to neuroradiology procedures, including endovascular thrombectomy. Among its potential applications, this simulation can assist practitioners in selecting the appropriate catheter for a specific patient and in preplanning the optimal vascular pathway to quickly reach the blood clot.

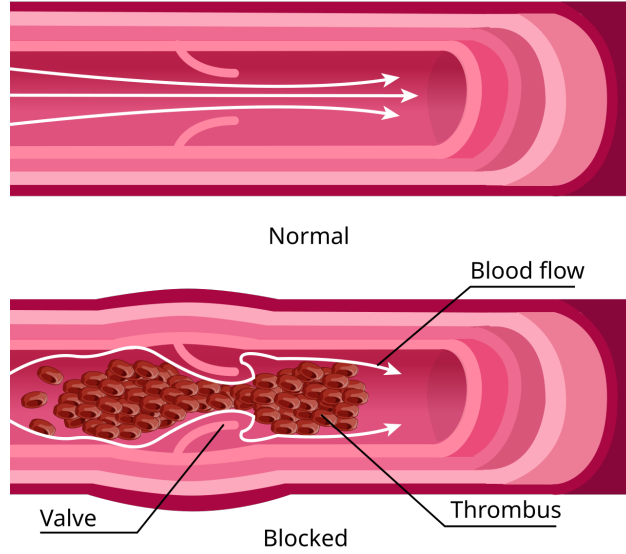


Figure 1.1: Illustration comparing normal and blocked blood flow in a blood vessel [1]. The top part shows smooth and uninterrupted flow in a healthy vessel. The bottom part shows how a blood clot (thrombus) can block the vessel, reducing the amount of blood that reaches nearby tissues. This kind of blockage can damage surrounding tissue and often needs to be treated quickly using procedures like endovascular thrombectomy.

The next section outlines the objectives of this research.

1.2 Simulation Objectives

Catheter navigation in neuroradiology procedures involves multiple physical phenomena. To define achievable objectives within a three-year research timeframe, we make the following assumptions for the simulation:

- Blood vessels are assumed to be fixed and rigid.
- External forces other than contact forces, such as those due to blood flow, are ignored.
- Intervention devices other than the catheter (e.g., blood clot aspiration or removal) are not modeled.

Instead, this work objectives are the following:

- Proposing a numerical resolution of a catheter model, treating the catheter as a passive, non-actuated, non-robotic structure.
- Accounting for both frictionless and frictional contact between blood vessel walls and the catheter.
- Capturing the physician hand manipulations by modeling the rotation and displacement of the catheter proximal end (or base).



Figure 1.2: Stent retriever after successfully removing a blood clot [2]. The device is designed to expand and catch the clot inside a blocked blood vessel. In this image, the clot is still visible inside the metal mesh. This technique is commonly used in endovascular thrombectomy to restore blood flow caused by a blocked artery.

The following section outlines the constraints that guide the development of these objectives.

1.3 Simulation Constraints

Within this medical context, the simulation is expected to be predictive. Predictive simulations can be defined as those that are accurate, real-time, and adapted to patient data [6]. Accuracy in simulation is an umbrella term that encompasses multiple interconnected aspects, including physical fidelity, numerical accuracy, and robustness. Physical fidelity refers to how precisely the simulation predicts the behavior of catheters, both in free movement and under constrained conditions, such as frictional contact with blood vessel walls. Numerical accuracy refers to how well the numerical methods, such as reduction and discretization techniques and time-stepping schemes, solve the underlying mechanical equations of the catheter model. Robustness, in turn, evaluates how consistently the simulation behaves when mechanical or numerical parameters are varied, ensuring that the simulation remains reliable across a wide range of clinical situations. A real-time simulation is one that generates results at a rate equal to or faster than the physical system it represents, allowing for interactive use during clinical planning or training. Patient-adapted simulation incorporates anatomical and physiological data specific to an individual, enabling the model to reflect case-specific conditions and support personalized medical planning.

Following the above definition of predictiveness, the reasoning behind this work is as follows. First, we treat the accuracy constraint as a hard requirement by selecting, from the beginning, a mechanically correct catheter model capable of capturing physical fidelity in free movement. Furthermore, we aim to develop numerical methods that ensure an acceptable level of numerical accuracy and robustness. Next, we address the real-time constraint as a

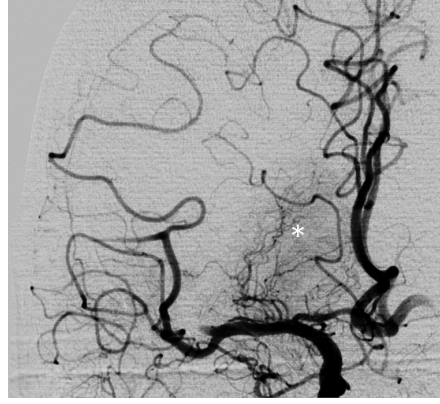


Figure 1.3: Visualization of the brain vascular network [3]. The dense and intricate vessel pathways illustrate the anatomical complexity clinicians must navigate during endovascular thrombectomy.

soft requirement by aiming for fast simulation performance. Finally, to fulfill the patient-adapted simulation constraint, we aim to validate our methods using real patient anatomical data.

The next section introduces the ANR project that framed the context of this research.

1.4 PreSPIN Project

The work presented in this dissertation is part of the ANR project PreSPIN (ANR-20-CE45-0011), which focuses on developing a predictive planning pipeline for endovascular treatment of ischemic strokes. The pipeline is built around three connected steps (see Fig. 1.4). The first step involves generating accurate and patient-specific models of the brain vascular network using three-dimensional magnetic resonance angiography [7, 8]. This imaging technique uses magnetic fields and radio waves to visualize blood vessels without needing contrast injection. The second step focuses on simulating the navigation of the catheter through these vascular structures, which is the main subject of this dissertation. The third step consists of running computational fluid dynamics simulations to predict how blood will flow through the brain after blood clot removal. This includes simulating perfusion magnetic resonance imaging, which shows how blood circulates through brain tissue. The goal is to assess the success of revascularization, which refers to restoring blood flow to parts of the brain that were previously deprived due to a clot or other blockage.

To achieve these goals, the project is structured into four work packages. Going further into details, work package 1 addresses the challenge of generating precise vascular models from noisy magnetic resonance angiography data, which often suffers from low signal quality and patient-specific anatomical variability. Work package 2, developed in this dissertation, creates efficient computational solutions for realistically simulating catheter navigation, including interactions such as contact with vessel walls and friction. Work package 3 builds models to predict blood flow dynamics at multiple scales, from large arteries down to capillary-level circulation, by simulating perfusion magnetic resonance imaging. This type of magnetic resonance imaging assesses how well blood flows through tissues, providing insight

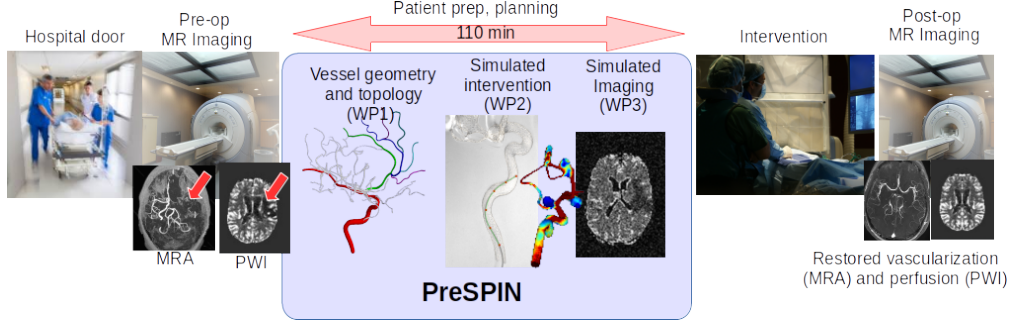


Figure 1.4: Overview of the PreSPIN project pipeline. From the patient arrival to post-operative imaging, PreSPIN focuses on the planning phase of ischemic stroke treatment using patient-specific data. It includes vessel extraction from magnetic resonance images, simulation of catheter navigation, and simulation of perfusion imaging.

into brain tissue health following stroke. Finally, work package 4 manages experimental validation, ensuring the simulation clinical relevance, data quality, and reproducibility through in vitro experiments and collaboration with clinicians.

PreSPIN integrates expertise from numerical simulation, medical imaging, and clinical practice, aiming to improve the precision and reduce the duration of preoperative planning to enhance patient outcomes in ischemic stroke interventions. This work is part of a joint research effort between academic institutions and a clinical partner. It brings together several research laboratories, including Inria Center at Université de Lorraine, CReSTIC and LMR which are affiliated with Université de Reims Champagne-Ardenne, and CREATIS which is affiliated with INSA Lyon, alongside the CHU Nancy.

1.5 Contributions

Having introduced the medical context, objectives, and constraints of this work, we now present a high-level summary of our contributions. First, to meet the constraint of physical fidelity, we review different catheter models in Chapter 2 and select the geometrically exact Cosserat rod model which balances accuracy with modeling complexity. The contributions in the following chapters can be summarized as follows.

In Chapter 3, we formulate the Cosserat rod model as a boundary value problem in space and propose a collocation method to solve it. Since the boundary value problem in the dynamic regime is known to be fundamentally singular when the catheter material is soft or the time step is small, we demonstrate through simulations that our numerical method overcomes this singularity when the number of basis functions exceeds a certain value. This contribution addresses the predictiveness constraints related numerical accuracy and robustness.

In Chapter 4, we extend this numerical technique with the ability to model proximal end displacement and rotation of the catheter which mimic the physician hand manipulation. We also incorporate contact and friction to simulate catheter navigation within tubular surfaces resembling blood vessels. These surfaces are modeled using implicit functions, which significantly accelerate contact detection and improve overall computational performance.

The catheter navigation is tested on a carotid artery model extracted from real patient data. This contribution addresses the predictiveness constraint related to patient-specific adaptation.

In Chapter 5, we adopt a quasi-static regime instead of the dynamic one and explore a Lagrangian reduction method to solve the catheter model efficiently. Since the resulting equations are stiff, we employ implicit time integration methods and derive the Jacobian matrix analytically. This analytical formulation is made possible by the differentiability of the implicit surface model. This contribution addresses the predictiveness constraint related to real-time computation.

This work has contributed to the following publications in international venues:

R. Jilani, P. -F. Villard and E. Kerrien, "An Orthogonal Collocation Method for Static and Dynamic Cosserat Rods," 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 2023, pp. 4328-4333, doi: 10.1109/IROS55552.2023.10341631.

R. Jilani, P. -F. Villard and E. Kerrien, "Solving Dynamic Cosserat Rods with Frictional Contact Using the Shooting Method and Implicit Surfaces," 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 2024, pp. 10483-10488, doi: 10.1109/IROS58592.2024.10801774.

R. Jilani, P. -F. Villard and E. Kerrien, "Angular Strain Parameterization for Solving Static Cosserat Rods," presented at the ICRA@40 Symposium, 40th Anniversary of the IEEE International Conference on Robotics and Automation, Rotterdam, Netherlands, Sep. 2024. [Online]. Available: <https://inria.hal.science/hal-04774573/>

R. Jilani, P. -F. Villard and E. Kerrien, "Quasi-Static Cosserat Rods in Contact With Implicit Surfaces," in IEEE Robotics and Automation Letters, vol. 10, no. 7, pp. 6536-6543, July 2025, doi: 10.1109/LRA.2025.3570131.

From top to bottom, the first publication corresponds to Chapter 3, the second to Chapter 4, and the fourth to Chapter 5. The work presented in this dissertation is a significantly improved version of those papers, with details provided in the respective chapters. This dissertation embraces an open-source approach. All published papers, except the first, are accompanied by publicly available source code. Each contribution chapter is supported by a corresponding open-source implementation, ensuring transparency, reproducibility, and accessibility of the research.

Finally, the manuscript concludes in Chapter 6, where we summarize our findings and discuss future perspectives.

Chapter 2

State-of-the-Art Rod Modeling

This chapter has two main goals. The first goal is to review existing approaches for modeling catheters and to select the most suitable one for our simulation objective and constraints. Recall that the objective is to model catheter navigation inside blood vessels, with constraints involving acceptable levels of accuracy and computational efficiency. The second goal is to review state-of-the-art numerical methods for solving the selected catheter model, and to detail their respective advantages and limitations. This will set the stage for a discussion at the end of the chapter, aimed at providing a clearer understanding of our contributions.

The catheter is a slender object whose radius is much smaller than its length, and can therefore be considered a rod. Similarly, blood vessels can be regarded as general tubular (tube-like) surfaces. This terminology allows us to generalize the problem and treat catheter navigation as the simulation of a rod that is displaced and rotated from its proximal end while undergoing frictional contact inside tubular surfaces.

The structure of this chapter is as follows. We begin by reviewing existing rod models in Section 2.1, which lays the groundwork for the discussion in Section 2.2 on the reasoning behind choosing the Cosserat rod theory for catheter modeling. We then introduce the notational conventions and the continuous Cosserat rod theory, in Sections 2.3 and 2.4, respectively. The Cosserat rod model is formulated using a state-of-the-art Lie group formulation, enabling a unified treatment of translational and rotational components, and thus simplifying the governing equations. This theoretical foundation supports both the following chapters and Section 2.5, where we review state-of-the-art numerical methods for solving the Cosserat rod model. Finally, in Section 2.6, we conclude with a discussion that offers deeper insight into our contributions.

2.1 Rod Modeling

Rod modeling is of interest to many research communities. Indeed, rods represent various real-world objects. To name a few, they represent catheters for the medical community [9, 10, 11], hair for the computer graphics community [12, 13, 14], underwater cables in marine engineering [15, 16, 17], and soft and continuum robots for robotics [18, 19, 20]. This dissertation is inspired by advances in the latter community, as our objectives, fast and accurate resolution of rod models, closely align with theirs. In the following, we review

existing rod models, and afterward we choose the most suitable one for our simulation objectives.

2.1.1 Three-Dimensional Continuum Mechanics Model

The most comprehensive approach to modeling the deformation of rods is through three-dimensional continuum mechanics [21, 22, 23]. In this framework, the rod is modeled as a solid volume composed of an infinite number of infinitesimally small elements. Each of these elements carries physical quantities such as mass, stress, and strain, which vary continuously throughout the rod. This includes variations within the cross-section, which is free to deform. The motion of the rod is described by a displacement field, which gives the change in position of each material point relative to its reference configuration. The deformation gradient, derived from the spatial variation of displacement, provides a measure of local stretching and shearing, and leads to strain expressions that remain valid even for large deformations.

Internal forces are derived from the stress field, which is determined by a constitutive law relating local strain to local stress. This law is often nonlinear, particularly when modeling large deformations or materials with complex behavior [24, 25]. For linearly elastic materials, this law is defined by a fourth-order elasticity tensor that describes how stiff the material is in response to deformation in different directions. The field of internal stresses contributes to the global equilibrium of the rod, which is described by the conservation of linear momentum. It states that the mass times acceleration at any point must equal the sum of all forces acting on it, including internal forces caused by stress gradients and external forces applied from outside the rod.

To solve the equations governing this system, the rod is typically discretized using the finite element method [21, 26, 27]. This involves subdividing the continuous volume into a mesh of small, finite elements like tetrahedra or hexahedra. To approximate the displacement field, values are assigned to discrete points (nodes), and shape functions are used to interpolate displacements within each element. The governing equations are reformulated in their weak form by multiplying them with shape functions and integrating over the domain. These integrals are then evaluated element-wise across the mesh and grouped into a global system of algebraic equations for numerical solution.

This modeling technique is versatile and accurate, capable of capturing all deformation modes. It does not require any simplification with respect to the rod geometry, such as assuming a uniform cross-section. Boundary conditions can be applied at any part of the rod, and the model remains valid even under complex loading or contact conditions. However, the discretized system can involve a large number of degrees of freedom, especially when high accuracy or fine spatial resolution is required. Solving the resulting equations requires high computational cost and memory, making this method more suitable for offline analysis than real-time simulations [28, 29].

2.1.2 One-Dimensional Continuum Mechanics Models

One-dimensional rod models approximate slender structures, such as catheters, as curves in space rather than full three-dimensional volumes. Physical quantities like stress, strain, and velocity are defined along this curve, reducing computational complexity while capturing

deformations such as extension, shear, bending, and torsion. Both geometrically exact and non-geometrically exact formulations adopt this one-dimensional representation, differing in how they handle geometric nonlinearities.

Non-Geometrically Exact Models

Non-geometrically exact rod models, such as classical rod theories, provide a simplified linear approximation suitable for slender structures under small deformations and rotations [30, 31]. Building on the one-dimensional framework introduced above, these models linearize both the kinematic relations and the equilibrium equations, leading to efficient formulations.

In the classical Euler-Bernoulli model, it is assumed that cross-sections remain perpendicular to the rod axis during bending [32]. This model accounts for bending but neglects shear deformation and, in some simplified forms, also ignores extension and torsion. To describe the dynamics, the balance of forces leads to a fourth-order partial differential equation for the position, which in many scenarios can be solved analytically.

The Timoshenko model improves upon this by relaxing the assumption on cross-section orientation. Cross-sections are allowed to rotate independently from the centerline tangent, enabling the model to capture shear deformation in addition to bending [33]. The governing equations form a coupled system of second-order partial differential equations for the cross-section position and orientation. These equations remain linear under small deformation and rotation assumptions and can be solved either analytically or using standard numerical methods depending on the boundary conditions.

While the reduced complexity of linear rod models offers computational advantages, their assumptions make them unsuitable for scenarios involving large deformations or rotations, like in catheter navigation.

Geometrically Exact Models

Geometrically exact rod models extend the one-dimensional rod representation to cases involving large deformations and rotations by preserving full geometric nonlinearity [34]. These models do not linearize the kinematics or equilibrium relations, making them suitable for a broader range of applications.

The Cosserat rod theory is a geometrically exact model, where each point along the rod centerline is associated with a moving orthonormal frame representing the orientation of the cross-section. The cross-section is assumed to be infinitesimal and rigid. The configuration of the rod is described by both the position and orientation of these cross-sections. Deformation is captured by how these frames evolve in space, with both translational (extension and shear) and rotational (bending and torsion) components represented. A common simplification is the Kirchhoff assumption, which states that cross-sections remain perpendicular to the centerline tangent. This eliminates shear deformation while still allowing to capture large bending and torsion. The kinematics are derived from the differential geometry of framed curves [35], and the governing equations are derived from the balance of linear and angular momentum [36] or variational principles [37].

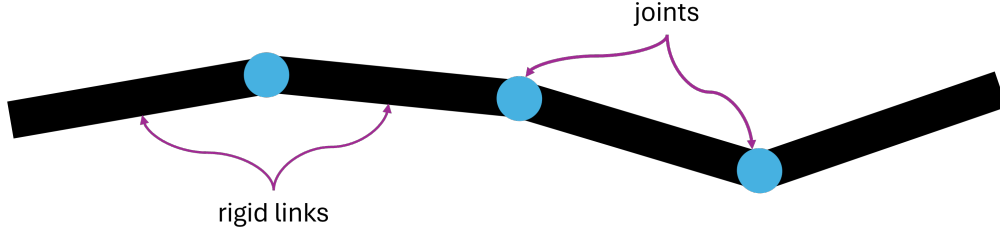


Figure 2.1: Pseudo-rigid body model consisting of rigid links connected by revolute joints, providing a piecewise-rigid approximation of a deformable structure.

Although the Cosserat rod model involves increased modeling complexity compared to linear rod models, it remains geometrically exact while being more computationally efficient than fully three-dimensional formulation [38].

2.1.3 Constant Curvature Model

Among simplified rod models, the constant curvature model is a popular choice due to its efficiency. It represents the rod as a sequence of segments, each shaped as a circular arc with constant curvature [39, 40]. This approach reduces model complexity by focusing only on bending and is particularly effective when the rod exhibits arc-like configurations.

Each segment is defined by parameters such as arc length, curvature, and in some cases, torsional twist. The pose of the rod tip can be computed by sequentially applying the rigid transformations associated with each segment, starting from the base. In many scenarios, especially when bending is constrained to one plane, the position and orientation of each arc can be obtained analytically using closed-form expressions based on trigonometric functions [41, 42].

When dynamics are involved, the time evolution of the arc parameters is often governed by Lagrangian mechanics, where they represent the generalized coordinates [43]. The kinetic and potential energies of the rod are expressed as functions of these parameters and their time derivatives. The resulting equations of motion are obtained by applying energy balance principles to derive second-order differential equations. These describe how the parameters change in response to imposed boundary conditions.

The constant curvature model is commonly used in robotics, as the actuation mechanisms inherently impose curvature constraints [44]. Furthermore, its simplicity makes it suitable for real-time simulations [45]. However, the model assumptions limit its range of applicability. It cannot easily take into account varying curvature, shear deformations, or external interactions such as contact and friction. Torsion is typically modeled in an ad hoc way, which can result in a lack of physical consistency [46]. Therefore, while constant curvature models are computationally efficient, they are best suited for applications where the rod naturally adopts configurations close to circular arcs.

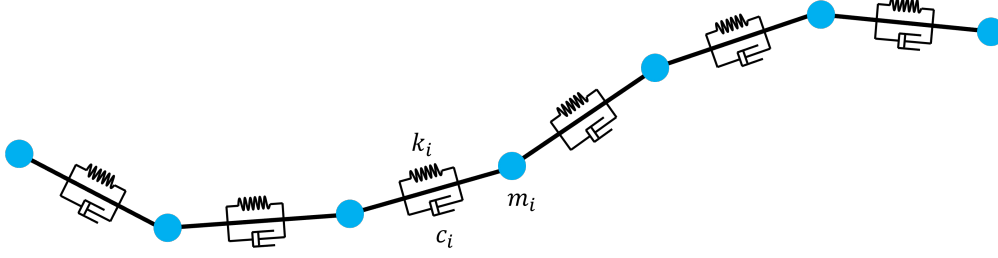


Figure 2.2: Discrete mass-spring-damper representation of a rod. The rod is modeled as a series of point masses m_i connected by springs with stiffness k_i and dampers with damping coefficient c_i , capturing its elastic and dissipative properties.

2.1.4 Pseudo-Rigid Body Model

In the pseudo-rigid body model, the rod is represented as a series of rigid links connected by rotational joints [47, 48]. This is visually illustrated in Fig. 2.1. Each link is a rigid body, and the deformation is captured by the rotation angles at the joints, which quantify the change in orientation between adjacent segments. These joints are typically modeled as torsional springs, with resistance to bending proportional to the angular displacement from their rest angles.

The system dynamics follow classical Newton-Euler rigid body mechanics. Each link experiences both translational motion of its center of mass and rotational motion around its own axis. Deformation is captured through relative joint rotations and system dynamics is determined by solving the force and moment balances across all links and joints.

A key advantage of the pseudo-rigid body model is that it builds directly on classical rigid-body mechanics, allowing the use of standard tools for dynamics and collision modeling. However, achieving accurate results often requires a large number of links, especially when the rod undergoes complex deformations. Furthermore, identifying the appropriate parameters for joint stiffness and damping is not straightforward and may require optimization techniques, as no direct mapping to the rod material properties exists [49, 50].

2.1.5 Mass-Spring Model

The mass-spring model offers a minimalistic approach to simulating rods by discretizing them into a chain of point masses connected by linear springs and, optionally, damping elements [51, 52, 53]. A visual representation of the model is shown in Fig. 2.2. Each mass represents a discrete point along the rod. The springs simulate elastic behavior by resisting changes in the distances between these points. Damping elements are introduced to model energy dissipation due to internal friction or contact.

The dynamics of the system are governed by Newton’s laws of motion. For each mass, the net force is computed based on spring elongation or compression and relative velocities for damping. These internal forces, combined with external ones, determine the accelerations that are integrated over time to update the motion. Explicit integration schemes such as Euler or Runge-Kutta methods are commonly used to simulate the system evolution.

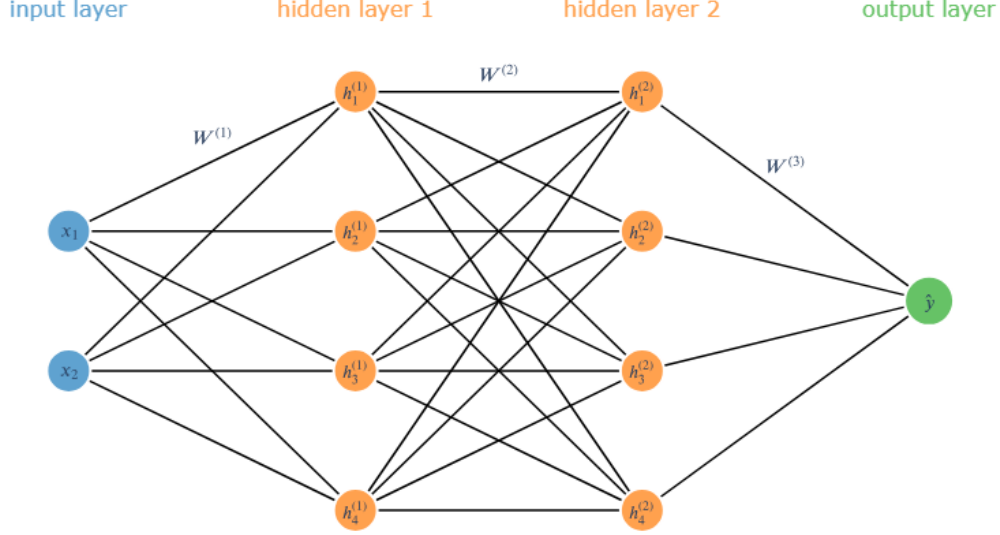


Figure 2.3: Fully connected feedforward neural network. This figure is provided to support readers unfamiliar with neural networks and illustrates the standard structure and notation of a multilayer perceptron. The input vector $x = (x_1, x_2)^T$ is propagated through a sequence of hidden layers, where each layer computes $h^{(l)} = \sigma(W^{(l)}h^{(l-1)} + b^{(l)})$. The network produces the final output $\hat{y} = \phi(W^{(L)}h^{(L-1)} + b^{(L)})$. Here, $h^{(l)}$ is the output of layer l , $W^{(l)}$ and $b^{(l)}$ are the corresponding weights and biases, σ is the hidden-layer activation function, ϕ is the output activation function, and L is the index of the output layer.

The mass-spring model is easy to implement and flexible in adapting to various geometries [54]. However, accurately capturing rod deformations requires a high number of discrete points, which increases the computational cost. Furthermore, it suffers from the same issue as the pseudo-rigid body model, where tuning the spring and damper parameters to match the rod material properties is not straightforward.

2.1.6 Data-Driven Models

Data-driven models represent a fundamentally different approach for simulating rods. Instead of relying on mechanical principles, these models learn to predict rod behavior from observed data [55, 56]. The most common framework are neural networks, which can be trained to map rod configurations to corresponding forces or future states [57]. This mapping is inferred from datasets generated through experiments or simulations.

A typical neural network consists of multiple layers, each applying a nonlinear transformation to its input (see Fig. 2.3). Given an initial configuration, such as node positions or strain values, the network outputs an estimated physical quantity of interest. The accuracy of the prediction depends on the quality and diversity of the training data, as well as the model architecture and training process.

Once trained, data-driven models can offer very fast predictions, making them attractive for real-time simulation [58]. However, building an accurate and comprehensive training dataset is challenging, especially for cases involving complex behaviors such as contact or

friction. Additionally, neural networks often fail to generalize to unseen scenarios. Their outputs may not respect physical laws such as conservation of momentum or energy, which has inspired recent work to add loss functions that enforce such constraints [59].

2.2 Selection of a Rod Model for Catheters

In our selection process, we first exclude the constant curvature model and data-driven approaches. The constant curvature model is limited by its assumption of uniform curvature, which prevents it from capturing large variations in bending and torsion observed in catheter navigation, while data-driven methods depend on extensive training data and may not generalize reliably across all scenarios. Next, the pseudo-rigid body and mass-spring models are also excluded because their reliance on parameter tuning makes it difficult to directly map these parameters to the catheter material properties, particularly under complex contact conditions. This leaves a choice between three-dimensional and one-dimensional continuum mechanics. Although the three-dimensional approach offers better accuracy and generality, its high computational cost makes it unsuitable for efficient simulation. Within one-dimensional formulations, classical linear models such as the Euler-Bernoulli or Timoshenko theories cannot adequately capture the large rotations and deformations observed in catheter navigation. In contrast, the geometrically exact Cosserat rod model maintains fully nonlinear governing equations and, compared to three-dimensional formulations, offers a balanced trade-off between accuracy and computational efficiency for a slender structure, making it the optimal choice for modeling catheters.

Before detailing the state-of-the-art numerical methods of Cosserat rods in Section 2.5, we first introduce our notational conventions and describe in details the Cosserat model in the following two sections.

2.3 Notational Conventions

Most of the notational conventions and variable names follow those of Boyer et al. [60] and Tummers et al. [37]. For any vector $x \in \mathbb{R}^n$, we denote its components by $x = (x_1, x_2, \dots, x_n)^T$, where $\bullet^T$ denotes the transpose. We use the special orthogonal group $SO(3)$ and the special Euclidean group $SE(3)$ to describe rotations and rigid-body motions, along with their respective Lie algebras $so(3)$ and $se(3)$. The special orthogonal group

$$SO(3) := \{ R \in \mathbb{R}^{3 \times 3} \mid R^T R = I_3, \det(R) = 1 \},$$

consists of all 3×3 rotation matrices. Its associated Lie algebra

$$so(3) := \{ A \in \mathbb{R}^{3 \times 3} \mid A^T = -A \},$$

contains all skew-symmetric 3×3 matrices, which correspond to infinitesimal rotations. To represent rigid body motions that include both rotation and translation, we use the special Euclidean group

$$SE(3) := \left\{ \begin{pmatrix} R & r \\ 0_{1 \times 3} & 1 \end{pmatrix} \mid R \in SO(3), r \in \mathbb{R}^3 \right\}.$$

This group consists of homogeneous transformation matrices where $R \in SO(3)$ defines the rotational part and $r \in \mathbb{R}^3$ represents the translation. The corresponding Lie algebra

$$se(3) := \left\{ \begin{pmatrix} A & v \\ 0_{1 \times 3} & 0 \end{pmatrix} \mid A \in so(3), v \in \mathbb{R}^3 \right\},$$

describes infinitesimal rigid body motions, with A governing angular velocity and v representing translational velocity.

For a six-dimensional vector $u = (\omega^T, v^T)^T = (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3)^T \in \mathbb{R}^6$, the hat operator $\hat{\bullet}$ and its inverse $\bullet^\vee$ are defined by

$$\hat{u} = \begin{pmatrix} \hat{\omega} & v \\ 0_{1 \times 3} & 0 \end{pmatrix}, \quad \hat{u}^\vee = u, \quad \text{where} \quad \hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \quad \hat{\omega}^\vee = \omega.$$

Here, $\hat{u} \in se(3)$ and $\hat{\omega} \in so(3)$. The operators $\hat{\bullet}$ and $\bullet^\vee$ are context-dependent and can refer to different mappings depending on the input. Specifically, $\hat{\bullet}$ may denote a map from $\mathbb{R}^3$ to $so(3)$ or from $\mathbb{R}^6$ to $se(3)$, while $\bullet^\vee$ can denote the inverse mappings from $so(3)$ to $\mathbb{R}^3$ or from $se(3)$ to $\mathbb{R}^6$. For a homogeneous transformation $g \in SE(3)$

$$g = \begin{pmatrix} R & r \\ 0_{1 \times 3} & 1 \end{pmatrix},$$

the adjoint representation $Ad_g : SE(3) \rightarrow \mathbb{R}^{6 \times 6}$ is expressed as:

$$Ad_g = \begin{pmatrix} R & 0_{3 \times 3} \\ \hat{r}R & R \end{pmatrix}.$$

The adjoint representation is used to transform any twist $u \in \mathbb{R}^6$ from one frame to another. The associated adjoint map $ad : \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 6}$ is given by

$$ad_u = \begin{pmatrix} \hat{\omega} & 0_{3 \times 3} \\ \hat{v} & \hat{\omega} \end{pmatrix}.$$

Furthermore, we use the vec operator, which transforms a matrix into a column vector by stacking its columns sequentially. Throughout this dissertation, we refer to the fixed spatial frame as the inertial frame and the body-attached material frame as the mobile frame. The terms proximal end and base are used interchangeably to refer to the end of the catheter or rod manipulated by the physician, while distal end and tip refer to its free end.

Now that we have introduced the necessary notational conventions, we proceed to describe the Cosserat rod model in the next section.

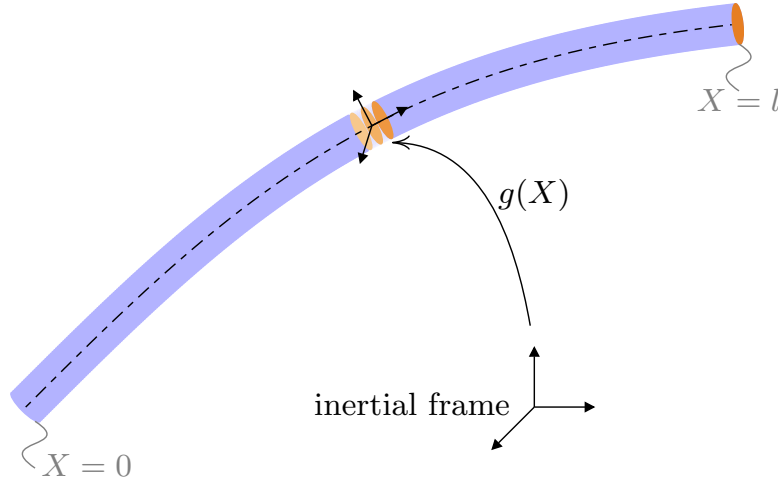


Figure 2.4: A Cosserat rod defined over the arc length domain $X \in [0, l]$. The centerline of the rod is depicted by the dashed black curve. At each point X , the configuration is given by a rigid body transformation $g(X) \in SE(3)$, encoding both the position of the centerline and the orientation of the cross-section.

2.4 Cosserat Rod Model

A Cosserat rod can be considered a continuous framed curve, i.e., each point on the curve is associated with a frame. The origin of each frame is located at the center of the cross-section, and its orientation aligns with that of the cross-section (Fig. 2.4). The first orientation vector is perpendicular to the plane of the cross-section, while the other two vectors lie within it. The rod is parameterized by the arc length $X \in [0, l]$, where l is the rod length in the reference configuration. Hence, the rod configuration space can be expressed as:

$$\mathcal{C} = \{g : X \in [0, l] \rightarrow g(X) \in SE(3)\}, \quad (2.1)$$

with g being the cross-section homogeneous transformation or pose:

$$g = \begin{pmatrix} R & r \\ 0_{1 \times 3} & 1 \end{pmatrix}.$$

The position is denoted as $r \in \mathbb{R}^3$, and the orientation is represented by a rotation matrix $R \in SO(3)$. Hence, the position and the orientation are unrestricted, giving the rod six degrees of freedom at every point. This leads to six possible deformations: bending in two directions, torsion, shear in two directions, and extension. By comparison, inextensible Kirchhoff rods experience only bending and torsion deformations. Figure 2.5 presents a three-dimensional visualization of Cosserat deformation modes. Furthermore, because the rod is reduced to its centerline, the cross-section is assumed to remain rigid and cannot deform. Consequently, the accuracy of predictions made by Cosserat rod theory improves as the object becomes more slender.

Using g and its space and time derivatives, the following twists are obtained:

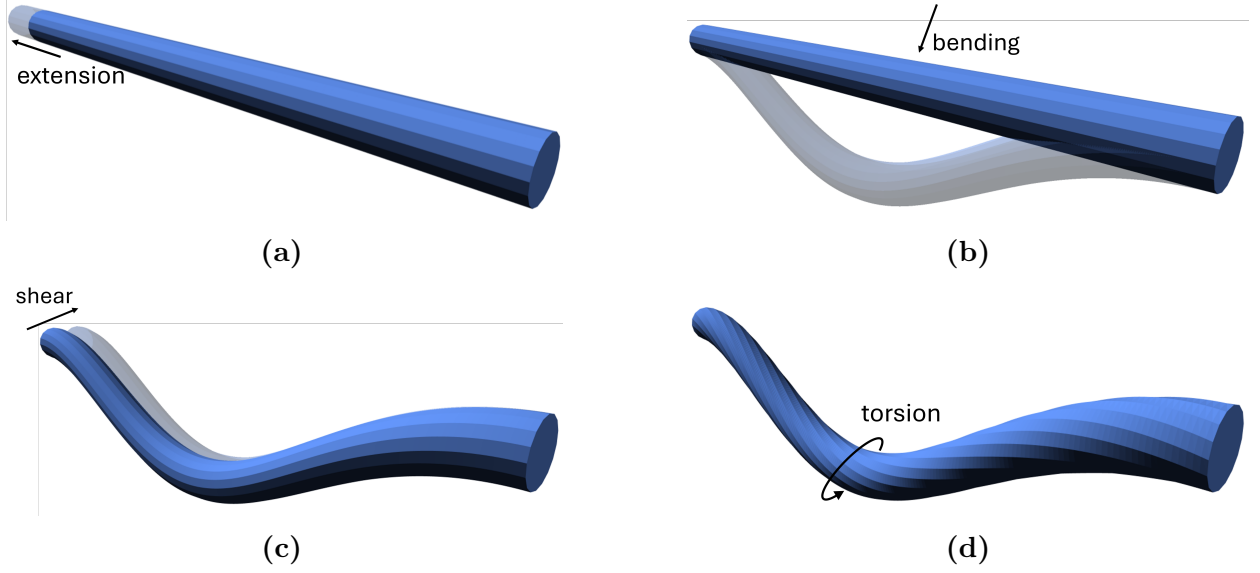


Figure 2.5: Visualization of Cosserat rod deformation modes: (a) extension, (b) bending, (c) shear, and (d) torsion.

$$\xi = (g^{-1}g')^\vee, \quad \eta = (g^{-1}\dot{g})^\vee, \quad (2.2)$$

with $\bullet' = \frac{\partial \bullet}{\partial X}$, $\dot{\bullet} = \frac{\partial \bullet}{\partial t}$, and t denotes time. $\xi = (K^T, \Gamma^T)^T$ is the space-rate twist, where K is the angular space-rate, and Γ is the linear space-rate. $\eta = (\Omega^T, V^T)^T$ is the time equivalent of ξ , it represents the velocity twist, where Ω is the angular velocity, and V is the linear velocity. Rearranging the twists (2.2) provide the standard form of Cosserat kinematics:

$$g' = g\hat{\xi}, \quad (2.3)$$

$$\dot{g} = g\hat{\eta}. \quad (2.4)$$

By abuse of terminology, ξ is sometimes referred to as the strain (as in [61]), however the strain, denoted ϵ , is the difference between ξ and its value at the reference configuration:

$$\epsilon = \xi - \xi_o, \quad (2.5)$$

with $\xi_o = (g_o^{-1}g'_o)^\vee = (K_o^T, \Gamma_o^T)^T$ is the reference configuration equivalent of ξ . For instance, if the rod is straight along the x-axis in its reference configuration, then $\xi_o = (0, 0, 0, 1, 0, 0)^T$. Assuming small strains, this work models catheter behavior as linear elastic, as they operate within the elastic range of their material. This assumption is widely used in continuum robotics [61, 37], and in some cases applied in soft robotics as well, even though such robots are often considered hyperelastic [62]. Therefore, the constitutive law is given by:

$$\Lambda = \mathcal{H}\epsilon = \mathcal{H}(\xi - \xi_o), \quad (2.6)$$

where $\Lambda = (C^T, N^T)^T$ is the stress wrench. It contains C , the internal moment, and N , the internal force. $\mathcal{H} = \text{diag}(GI_x, EI_y, EI_z, EA, GA, GA)$ is the Hookean matrix, where G is the shear modulus, E is Young's modulus, A is the cross-sectional area, and I_x , I_y , and I_z are the second moments of inertia for torsion and bending about the y - and z -axes, respectively. It is worth noting that $\mathcal{H}$ can depend on the arc length, for instance, to account for changes in cross-section diameter. The constitutive law (2.6) can be extended to include damping, such as Kelvin-Voigt type viscous damping [63, 61, 64], or robotic actuation [37]. The Kelvin-Voigt damping model introduces a linear viscous contribution to the constitutive law. Given a damping matrix $\mathcal{D} \in \mathbb{R}^{6 \times 6}$, the stress Λ is expressed as:

$$\Lambda = \mathcal{H}(\xi - \xi_o) + \mathcal{D}\dot{\xi} \quad (2.7)$$

The dynamic balance equation can be derived from Newton-Euler equations [34] or Hamilton principle [65]:

$$\mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta = \Lambda' - ad_{\xi}^T \Lambda + \bar{F}, \quad (2.8)$$

where $\dot{\eta} = (\dot{\Omega}^T, \dot{V}^T)^T$ is the acceleration twist, composed of the angular acceleration $\dot{\Omega}$ and the linear acceleration $\dot{V}$. Furthermore, the external distributed wrench $\bar{F} = (\bar{C}^T, \bar{N}^T)^T$ is composed of external distributed moment $\bar{C}$ and force $\bar{N}$. The inertia matrix is given by $\mathcal{M} = \text{diag}(\rho I_x, \rho I_y, \rho I_z, \rho A, \rho A, \rho A)$, with ρ being the rod density. Similar to $\mathcal{H}$, $\mathcal{M}$ can be arc length dependent. Finally, the static balance equation can be obtained by simply removing the inertial effects from the dynamic balance equation:

$$\Lambda' - ad_{\xi}^T \Lambda + \bar{F} = 0. \quad (2.9)$$

It is useful, in the next section, to observe that time and space differentiations commute, because t and X are independent:

$$(g')^\cdot = (\dot{g})'. \quad (2.10)$$

From this equality the evolution of velocity over arc length can be obtained [66]:

$$\eta' = -ad_{\xi}\eta + \dot{\xi}. \quad (2.11)$$

Differentiating (2.11) with respect to time allows to obtain the evolution of acceleration over arc length:

$$\dot{\eta}' = -ad_{\xi}\dot{\eta} - ad_{\xi}\eta + \ddot{\xi}. \quad (2.12)$$

To summarize, we have presented the key components of the Cosserat model, including its assumptions, kinematics, strain, stress, and balance equations. This foundation allows us to present the state-of-the-art resolution of this model, which will be provided in the next section.

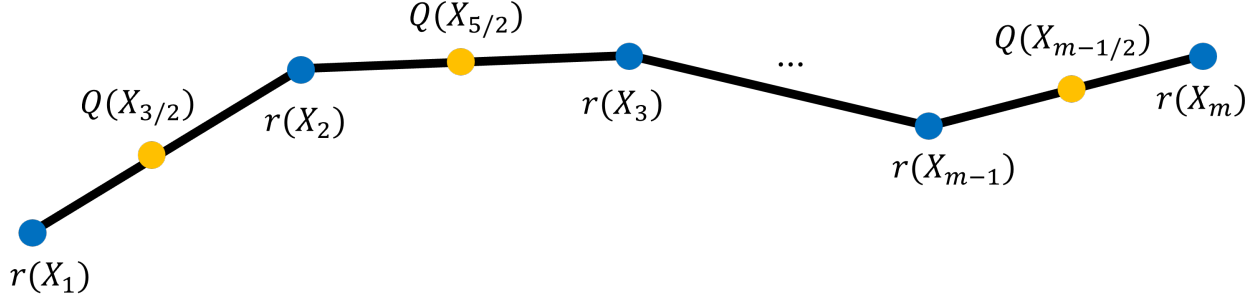


Figure 2.6: Lang et al. [4] introduced a rod material discretization in which positions r are defined at m nodes, whereas quaternions Q are defined at segment midpoints. The kinematics (2.3) are discrete, and the angular (K) and linear (Γ) rates of change are computed using finite differences with $r(X_i)$ and $Q(X_{i+1/2})$.

2.5 State-of-the-Art Methods for Solving Cosserat Rods

Since the Cosserat rod model attracts interest from various research communities, numerous numerical methods have been proposed, each with its own characteristics. In the following, we propose a classification of these methods into three categories: finite differences, boundary value problem approaches, and reduction-based methods. We describe a few representative works for each.

2.5.1 Finite Differences

Here, we describe how the Cosserat model can be solved using finite differences. This approach was employed in the discrete elastic rod model, originally proposed by Bergou et al. [67] in the computer graphics community. It later gained interest in the soft robotics community [62, 68]. For instance, Gazzola et al. [62] showed that this approach can accurately solve certain simulations in real-time. To illustrate this method, we take a representative example proposed by Lang et al. [4]. In their work, the rod configuration is discretized into segments a priori. Fig. 2.6 shows their staggered grid discretization. Let us denote the arc length of the nodes as X_i , the arc length of the segment centers as $X_{i+1/2}$, and define Q as the quaternion representation of the rotation matrix R . The authors formulated an explicit system of ordinary differential equations governing the time evolution of $r(X_i)$ and $Q(X_{i+1/2})$ in the form $\dot{z} = f(z, t)$, where $z = (r(X_i), Q(X_{i+1/2}), \dot{r}(X_i), \dot{Q}(X_{i+1/2}))$. This formulation is derived by rotating the balance equation (2.8) to the inertial frame and then isolating the accelerations. The spatial twist ξ is computed from r and Q using finite differences, the stress Λ is then obtained from ξ by inverting the constitutive law (2.6), and finally, Λ' is also computed via finite differences.

In [69, 62, 70] the discrete elastic rod model is combined with the penalty contact method to resolve collisions. In the penalty method, the contact response is determined by the interpenetration distance between the colliding objects: the greater the distance, the stronger the response. Tschisgale et al. [71] used the numerical method of Lang et al. [4] to model submerged canopies in rivers subjected to contact. Inspired by impulse-based contact models [72, 73, 74] that assume infinitesimal collision time, the contact response is computed as

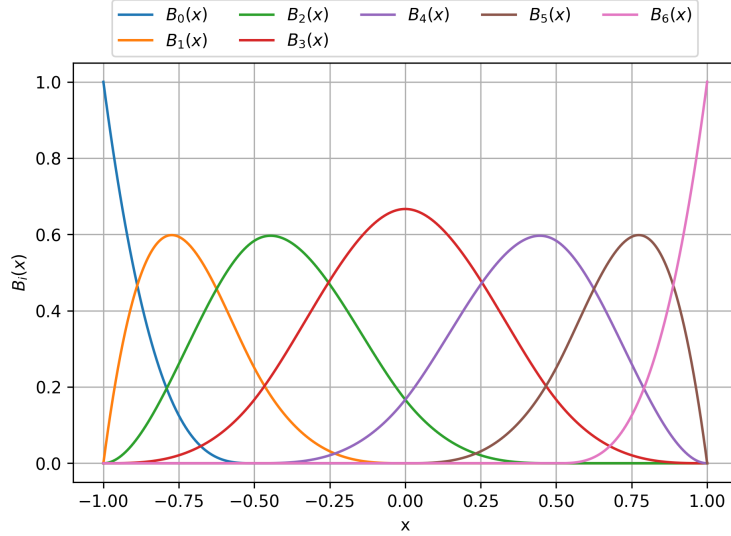


Figure 2.7: Cubic B-spline basis functions defined over the interval $[-1, 1]$. The basis has degree 3 and consists of 7 functions, defined over the open knot vector $\{-1, -1, -1, -1, -0.5, 0, 0.5, 1, 1, 1, 1\}$. Each curve $B_i(x)$ represents a basis function associated with one control point. The functions are C^2 -continuous at interior knots.

collision impulses (with units of mass times velocity), derived from the desired change in relative normal velocity after contact. Contact detection is performed for each discrete rod element and involves searching for the nearest contact point or surface; the resulting contact response is then distributed to the node accelerations.

Another representative work in the category of finite differences is that of Renda et al. [75], where they constructed an explicit system of ordinary differential equations in the form $\dot{z} = f(z, z', t)$ with $z = (g, \xi, \eta)$:

$$\begin{pmatrix} \dot{g} \\ \dot{\xi} \\ \dot{\eta} \end{pmatrix} = \begin{pmatrix} g\hat{\eta} \\ \eta' + ad_{\xi}\eta \\ \mathcal{M}^{-1} (ad_{\eta}^T \mathcal{M}\eta + \Lambda' - ad_{\xi}^T \Lambda + \bar{F}) \end{pmatrix} \quad (2.13)$$

From top to bottom, the first line represents the kinematics (2.4), the second line is a rearrangement of (2.11), and the third line is derived from the Newton-Euler balance equations (2.8). Here, Λ is computed from ξ using the constitutive law (2.6), while η' and Λ' are computed using finite differences.

The advantages of the finite difference approaches are that the evaluation of one simulation step is fast and the computation time scales linearly with the number of rod discrete elements, which enables to run certain simulations in real-time. However, the use of finite differences may lead to numerical instabilities if the time step Δt or the spatial resolution are not carefully selected. Indeed, converging to accurate solutions requires reducing the discrete element size ΔX , but this imposes a smaller time step for stability, which increases the overall computation time. Going further into details, Gazolla et al. [62]

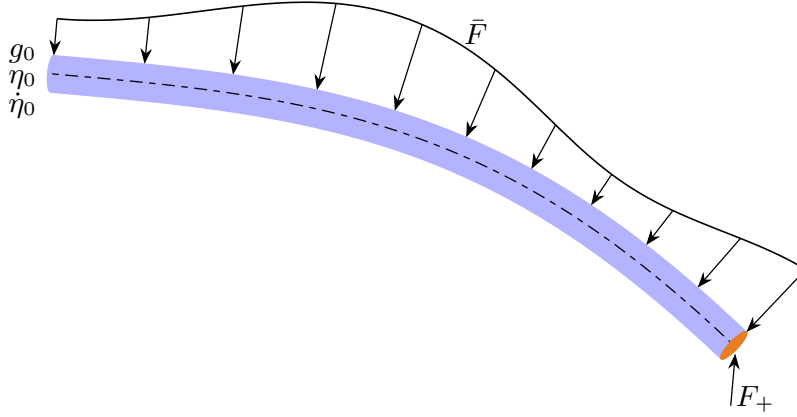


Figure 2.8: The figure depicts the clamped-free (Dirichlet-Neumann) boundary conditions of a Cosserat rod subjected to distributed loads $\bar{F}$ and a concentrated wrench at the tip (F_+). At the proximal end $X = 0$, the known quantities are the pose g_0 , as well as the velocity η_0 and acceleration $\dot{\eta}_0$; at the distal end $X = l$, the stress $\Lambda(l) = F_+$ is known.

observed a relation of the form $\Delta t = \chi \Delta X$ with $\chi \sim 10^{-2}\text{s/m}$, while the approach of Renda et al. [75] follows the Courant-Friedrichs-Lewy condition, $\Delta t \leq \frac{1}{\sqrt{E/\rho}} \Delta X$.

To avoid using finite differences, Weeger et al. [64] represented r , Q , $\dot{r}$, and $\dot{Q}$ as non-uniform rational B-splines (NURBS) and formulated an explicit system of ordinary differential equations in the form $\dot{z} = f(z, t)$ where z contains the control points of the discretized components. Fig. 2.7 shows the B-spline basis underlying the NURBS representation. This approach allows for the computation of all necessary spatial derivatives analytically using the chosen basis functions instead of finite differences approximation. Furthermore, contacts were taken into account using the penalty method. Although this approach does not rely on finite differences, it is included in this subsection because it closely aligns with the methods previously discussed.

In this category, the differential equations may become stiff in certain scenarios, for instance, when the Young's modulus is large or the rods are very thin, requiring small time steps to ensure convergence [62].

2.5.2 Boundary Value Problem

In this approach, the Cosserat rod model is formulated as a boundary value problem in the spatial domain and then solved iteratively using either the shooting method or collocation methods. Figure 2.8 shows a rod with clamped-free boundary conditions, where one end is fixed in position and orientation, and the other end is free to move. Figure 2.9 illustrates a rod with free-free boundary conditions, where both ends are unconstrained. To elaborate, we first present the boundary value problems in both static and dynamic modes. For the static case, the form simply consists in combining the kinematics (2.3) and the static balance equation (2.9)

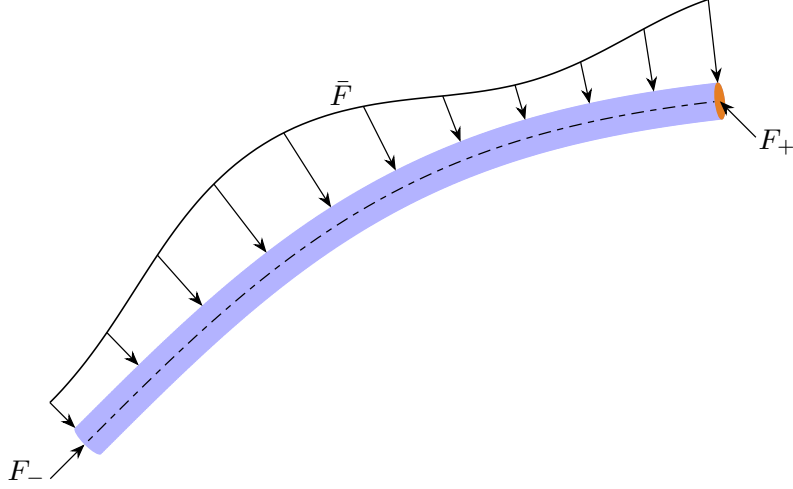


Figure 2.9: The figure depicts the free-free (Neumann-Neumann) boundary conditions of a Cosserat rod subjected to distributed loads $\bar{F}$ and two concentrated wrenches at the base (F_-) and the tip (F_+). The stress is specified at both ends: at the proximal end $X = 0$, the stress is prescribed as $\Lambda(0) = -F_-$; at the distal end $X = l$, the stress is given by $\Lambda(l) = F_+$.

$$\begin{pmatrix} g' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} g\hat{\xi} \\ ad_{\xi}^T \Lambda - \bar{F} \end{pmatrix}, \quad (2.14)$$

with clamped-free boundary conditions

$$g(0) = 1_{4 \times 4}, \quad \Lambda(l) = F_+, \quad (2.15)$$

to formulate the boundary value problem. Here, $F_+ = (C_+^T, N_+^T)^T$ is the external concentrated tip wrench, C_+ is the tip moment, and N_+ is the tip force. Free-free boundary value problem for the static mode is not defined for self-evident reasons. The boundary value problem for the dynamic mode is not as straightforward to formulate as it is for the static case. Indeed, combining the kinematics (2.3) and the dynamic balance equation (2.8)

$$\begin{pmatrix} g' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} g\hat{\xi} \\ \mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta + ad_{\xi}^T \Lambda - \bar{F} \end{pmatrix}, \quad (2.16)$$

with clamped-free boundary conditions

$$g(0) = 1_{4 \times 4}, \quad \eta(0) = 0_{6 \times 1}, \quad \dot{\eta}(0) = 0_{6 \times 1}, \quad \Lambda(l) = F_+,$$

or free-free boundary conditions

$$\Lambda(0) = -F_-, \quad \Lambda(l) = F_+,$$

with F_- being the rod proximal end equivalent of F_+ , results in an unsolvable boundary value problem. Indeed, this form has no means to compute the accelerations $\dot{\eta}$, and the

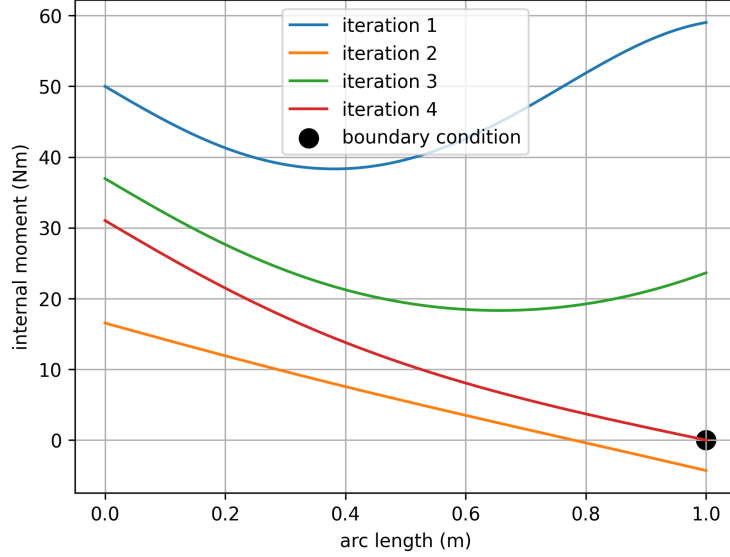


Figure 2.10: Solution to a clamped-free boundary value problem obtained using the shooting method. The distal boundary condition for the internal moment is set to $C(l) = C(1) = 0$. The shooting method computes the proximal value of the internal moment $C(0)$ that, after arc length integration, satisfies the boundary condition. The solution $C(0) = 31$ is found after four iterations: $C(0) = \{50, 16.5, 36.9, 31\}$.

velocities η . To address this, these equations are supplemented with additional equations that describe the evolution of $\dot{\eta}$ and η over arc length. Combining all the components (2.16) (2.11) (2.12)

$$\begin{pmatrix} g' \\ \eta' \\ \dot{\eta}' \\ \Lambda' \end{pmatrix} = \begin{pmatrix} g\hat{\xi} \\ -ad_{\xi}\eta + \dot{\xi} \\ -ad_{\xi}\dot{\eta} - ad_{\xi}\eta + \ddot{\xi} \\ \mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta + ad_{\xi}^T \Lambda - \bar{F} \end{pmatrix}, \quad (2.17)$$

with clamped-free boundary conditions

$$g(0) = 1_{4 \times 4}, \quad \eta(0) = 0_{6 \times 1}, \quad \dot{\eta}(0) = 0_{6 \times 1}, \quad \Lambda(l) = F_+, \quad (2.18)$$

or free-free boundary conditions

$$\Lambda(0) = -F_-, \quad \Lambda(l) = F_+, \quad (2.19)$$

provides the complete form of boundary value problem in the dynamic mode [66].

Shooting Method

These boundary value problems can be solved using the shooting method. This approach iteratively computes the unknown initial condition that, after arc length integration, satisfies the distal boundary condition. Fig. 2.10 visualizes an example of the shooting method used to compute the unknown internal moments at the rod proximal end. To illustrate this method

further, we show how clamped-free boundary value problems can be solved. For instance, by selecting the stress $\Lambda(0)$ as the unknown initial condition, as in [66], the residual vector $\mathcal{R} : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ is:

$$\begin{aligned}\mathcal{R}(\Lambda(0)) &= \Lambda(l) - F_+ \\ &= \Lambda(0) + \int_0^l \Lambda' dX - F_+\end{aligned}\tag{2.20}$$

Applied to static case (2.9) this residual becomes:

$$\mathcal{R}(\Lambda(0)) = \Lambda(0) + \int_0^l a d_\xi^T \Lambda - \bar{F} dX - F_+\tag{2.21}$$

Here, ξ is computed from Λ by inverting the constitutive law (2.6). Wiese et al. [76] used the shooting method in the static mode to model concentric tube robots in contact with simple geometrical shapes using the penalty method.

For the dynamic mode (2.8) the residual becomes:

$$\mathcal{R}(\Lambda(0)) = \Lambda(0) + \int_0^l \mathcal{M} \dot{\eta} - a d_\eta^T \mathcal{M} \eta + a d_\xi^T \Lambda - \bar{F} dX - F_+\tag{2.22}$$

Here, η and $\dot{\eta}$ are computed by integrating (2.11) and (2.12). However, $\dot{\xi}$ and $\ddot{\xi}$ in these equations remain unknown. This arises because the boundary value problem originates from a singular optimal control problem [66]. To regularize it, implicit time integration is used, allowing $\dot{\xi}$ and $\ddot{\xi}$ to be computed using known values from the current and previous time steps [66].

The shooting method is computationally efficient, as its Jacobian matrix $\frac{\partial \mathcal{R}}{\partial \Lambda(0)}$ is typically small, with a size of 6×6 when $\Lambda(0)$ is the only unknown, making it well-suited for computing certain simulations in real-time [61]. Additionally, it maintains accuracy, as the configuration space is not reduced but only discretized for numerical integration. However, the shooting method is known to be unstable for Cosserat rod dynamics. Initially, this instability was attributed to machine precision (floating-point) issues [77], but recent work by Boyer et al. [66] revealed that it comes from the fundamentally singular nature of the optimal control problem underlying the dynamic boundary value problem. Regularizing the problem using implicit time integration imposes restrictions on the time step and rod material properties. Specifically, the singularity reappears when soft rods or small time steps are used, preventing the shooting method from converging.

Collocation Method

Other solutions for solving static boundary value problems have been proposed. Orekhov et al. [78] used an orthogonal collocation method, while Weeger et al. [79] proposed an isogeometric collocation method. The collocation method approximates the solution using a finite set of basis functions, with the goal of determining their unknown parameters, such as coefficients or control points. This is achieved by solving a system of algebraic equations that ensures the derivative computed from the basis functions closely matches the derivative given by the differential equations at selected collocation points. Additionally, the system

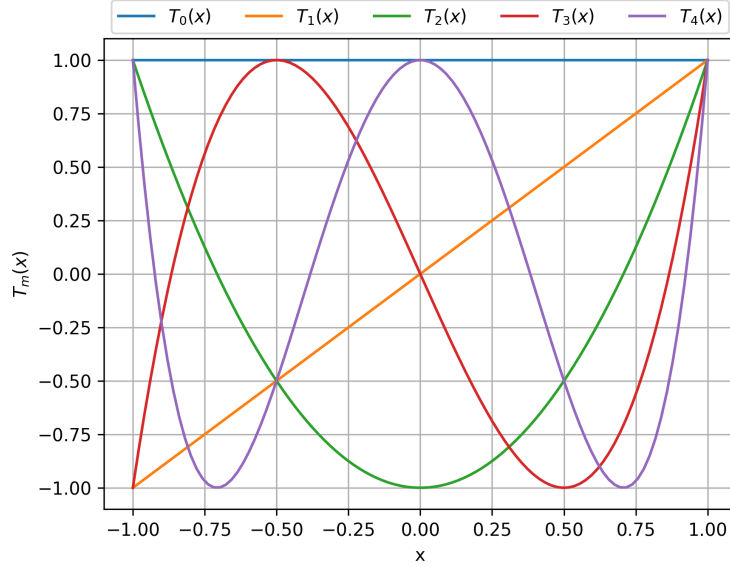


Figure 2.11: First five Chebyshev polynomials of the first kind $T_m(x)$ for degree $m = 0$ to 4, plotted over the interval $[-1, 1]$. These polynomials are orthogonal with respect to the weight function $w(x) = (1 - x^2)^{-1/2}$, meaning they satisfy the relation $\int_{-1}^1 T_m(x) T_n(x) w(x) dx = 0$ for $m \neq n$. Each polynomial $T_m(x)$ has m distinct zeros in $(-1, 1)$, given by $c_i = \cos\left(\frac{2i-1}{2m}\pi\right)$ for $i = 1, \dots, m$.

enforces boundary conditions by minimizing the difference between the computed solution at boundary points and the imposed conditions. Let us denote the collocation points as c_i . Using the Kirchhoff assumption (zero shear and extension), Orekhov et al. [78] solved the static case by setting $\{\xi(c_i)\}$ as the unknowns, representing ξ with three Chebyshev polynomials. Fig. 2.11 plots Chebyshev polynomials up to degree four.

The residual system takes the following form, with boundary condition enforcement omitted for simplicity:

$$\begin{aligned} \mathcal{R}(\{\xi(c_i)\}) &= \tilde{\xi}' - (\mathcal{H}^{-1}\Lambda + \xi_o)' \\ &= \tilde{\xi}' - \mathcal{H}^{-1}\Lambda' \\ &= \tilde{\xi}' - \mathcal{H}^{-1}(ad_{\xi}^T \Lambda - \bar{F}) \end{aligned} \tag{2.23}$$

Here, Λ' is obtained with the balance equation (2.9), Λ is computed from the constitutive law (2.6), and $\tilde{\xi}'$ is the counterpart of ξ' calculated using the Chebyshev differentiation matrix [80]. In their work, they demonstrated that using the Magnus expansion for arc length integration of the kinematics (2.3) allows the collocation method to be competitive with the shooting method in terms of computational efficiency. Furthermore, the Magnus expansion ensures that the pose $g(X)$ remains in $SE(3)$ during arc length integration. This property is not guaranteed when using standard numerical schemes such as the explicit Euler method, where the rotation part $R(X)$ may drift from $SO(3)$. The Magnus expansion is an approximation and requires careful parameter selection to ensure convergence

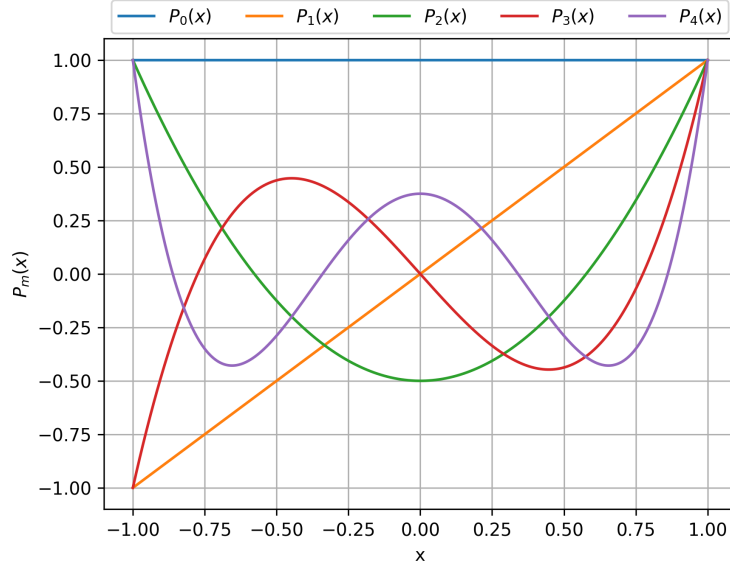


Figure 2.12: First five Legendre polynomials $P_n(x)$ for degree $n = 0$ to 4, plotted over the interval $[-1, 1]$. These polynomials are orthogonal with respect to the weight function $w(x) = 1$, meaning they satisfy $\int_{-1}^1 P_m(x)P_n(x) dx = 0$ for $m \neq n$. Each polynomial $P_n(x)$ has n distinct zeros in $(-1, 1)$.

[78]. Alternative approaches to address $SO(3)$ drift include the use of quaternions, as demonstrated in [81].

In the work of Weeger et al. [79], the position r and quaternion Q are represented using NURBS, with their collocation values as the unknowns. The residual system takes the following form, with boundary condition and unit quaternion enforcements omitted for simplicity:

$$\begin{aligned} \mathcal{R}(\{r(c_i), Q(c_i)\}) &= \tilde{\Lambda}' - \Lambda' \\ &= \tilde{\Lambda}' - ad_{\xi}^T \Lambda + \bar{F} \end{aligned} \quad (2.24)$$

Here, Λ is computed from ξ using the constitutive law (2.6), while ξ and $\tilde{\Lambda}'$ are derived from r and Q using NURBS differentiation. In [82], their work was extended to take rod to rod contact into account with the penalty method.

Compared to the shooting method, one drawback of collocation methods is that their Jacobian matrices are usually larger, as they depend on the number of basis functions, leading to greater computation time. To the best of our knowledge, no collocation method has been proposed for solving the boundary value problem of Cosserat rods in the dynamic regime.

2.5.3 Reduction-Based Methods

Reduction-based methods, which often use the Lagrangian formulation, can be categorized by their type of interpolation (nodal or modal) and the type of generalized coordinates

(absolute or relative) [38]. Nodal interpolation refers to interpolating node values, while modal interpolation represents the chosen field as a combination of global basis functions distributed along the rod arc length. Absolute coordinates are defined with respect to the inertial frame, while relative coordinates are defined with respect to the mobile frame.

Simo [83] was the first to propose a finite element method solution for Cosserat rods, referring to it as the geometrically exact finite element method (GE-FEM). This method is nodal and absolute, where the generalized coordinates are g and $\dot{g}$. It is widely considered one of the most powerful approaches for solving Cosserat rods. However, its robustness and accuracy have shown limitations [60], which inspired the development of (nodal) relative GE-FEM, such as those proposed by [84, 85, 86], where the generalized coordinates are the strains. For instance, [86] proposed a piecewise linear interpolation of the strain field. Their work was extended in [87] to take collisions into account. To obtain accurate contact response, they solved the nonlinear complementary problem of contacts [88]:

$$0 \leq \phi_n \perp F_n \geq 0 \quad (2.25)$$

This equation represents a unilateral constraint, where ϕ_n denotes the normal gap between the contacting objects and F_n the corresponding normal contact force. The condition $\perp$ implies that either the gap is strictly positive and the contact force vanishes, or the contact force is active and the gap is zero, but never both simultaneously. This guarantees a physically correct contact behavior. The unilateral constraint (2.25) is reformulated as a nonlinear equation, allowing it to be treated as an equality constraint and solved iteratively using root-finding techniques.

Boyer et al. [60] showed that modal relative reduction allows matching the accuracy of GE-FEM with fewer degrees of freedom. They used strain reduction of the form:

$$\epsilon = \Phi(X)q \quad (2.26)$$

where q represents the generalized coordinates, and Φ is a matrix of shape functions. Specifically, Legendre polynomials were used, and Fig. 2.12 displays them up to degree 4. By isolating the acceleration in the Lagrangian balance equation, they derived a minimal set of explicit ordinary differential equations governing the time evolution of q . The Lagrangian dynamic balance equation is expressed as:

$$M_\epsilon(q)\ddot{q} + Q_v(q, \dot{q}) + Q_c(q) + D_\epsilon\dot{q} + K_\epsilon q = 0 \quad (2.27)$$

Here, $M_\epsilon(q)$ is the generalized inertia matrix, $Q_v(q, \dot{q})$ represents the generalized velocity-dependent forces, and $Q_c(q)$ accounts for generalized external forces that depend on the configuration. The matrices D_ϵ and K_ϵ represent the generalized damping and generalized restoring forces, respectively. The matrices of the Lagrangian model can be computed either using the Newton-Euler algorithm [60, 37] or the geometric Jacobian matrix, which maps variations of q to variations of g [89, 90]. The Newton-Euler algorithm involves two passes of arc length integration:

- A forward pass (from $X = 0$ to $X = l$) to compute g , η , and $\dot{\eta}$ using (2.3) (2.11) (2.12)

- A backward pass (from $X = l$ to $X = 0$) to compute Λ with (2.8), which is then projected onto the shape functions to obtain the generalized forces.

The matrices in (2.27) are computed by supplying various values of q , $\dot{q}$, and $\ddot{q}$ to the algorithm. Alternatively, the geometric Jacobian approach computes the system matrices by projecting the kinematics (2.3) and the continuous balance equation (2.8) onto the strain coordinate space. In contrast to the algorithm above, this approach involves only forward integrations. It was efficiently implemented by Mathew et al. [89] and demonstrated the capability to perform certain dynamic simulations in real time. Furthermore, a special case of the relative modal approach is the piecewise constant strain formulation, as proposed by Renda et al. [91].

Although this approach allows to achieve good accuracy with few degrees of freedom, the underlying equations may become stiff, necessitating small time steps, which can degrade overall computation time.

2.6 Discussion and Conclusion

Earlier in this chapter, we reviewed different rod modeling approaches and identified the Cosserat rod theory as the most appropriate for catheters. We then provided an overview of state-of-the-art numerical methods for solving this model. All approaches presented preserve the geometric exactness of Cosserat rods and are therefore accurate, although some can achieve the same level of accuracy with fewer degrees of freedom. Furthermore, all have demonstrated the ability to perform real-time simulations: Gazzola et al. [62] for discrete elastic rods, Till et al. [61] for boundary value problems solved using the shooting method, and Mathew et al. [89] for the modal strain reduction approach. However, each approach comes with its own drawbacks. The discrete elastic rod model uses finite difference approximations, which can lead to instabilities; the shooting method has been shown to be unstable for dynamic simulations; and the Lagrangian model equations can become stiff, degrading computation time. Consequently, the current state-of-the-art solutions for Cosserat rods do not allow achieving accurate, fast, and robust simulations of complex catheter navigation.

The reasoning behind the contributions of this dissertation is as follows. We adopt a progressive modeling approach, starting with minimal assumptions and gradually increasing them to improve computation time. We begin by considering inertial effects and therefore employ the dynamic formulation to model catheter navigation. Subsequently, we neglect inertial effects and explore the use of the static formulation. From the perspective of contact modeling, this dissertation adopts the penalty method, in line with many of the previously cited works. This choice was motivated by its simplicity of integration, allowing the research effort to focus primarily on advancing numerical methods rather than introducing new contact modeling approaches.

Orekhov et al. [78] showed that solving static boundary value problems using the collocation method can be competitive with the shooting method in terms of computation time. To our knowledge, no one has solved the dynamic boundary value problem using a collocation method. Given that the boundary value problem in the dynamic regime is

fundamentally singular, a question arises as to whether such an approach can be robust. We propose a solution to the dynamic boundary value problem using an orthogonal collocation method and show through simulations that it solves the singularity problem (Chapter 3). Then, in Chapter 4, we incorporate contact modeling and apply the proposed method to simulate catheter navigation. Although realistic results were obtained, we concluded that the proposed collocation method, in its current form, is not suitable for fast simulation. To improve computational efficiency, in Chapter 5, we make a quasi-static assumption and employ the modal strain reduction of [66]. Since the problem is stiff, we use implicit functions to model the contact surfaces, which, due to their smoothness, facilitates the use of implicit methods that mitigate stiffness.

Chapter 3

Addressing the Singularity in the Dynamic Cosserat Rods Boundary Value Problem

Solving the dynamic Cosserat boundary value problem has proven to be numerically challenging. Indeed, recent work of Boyer et al. [66] showed that the boundary value problem in the dynamic mode originates from a fundamentally singular optimal control problem. To regularize it, implicit time integration is used. However, they showed that this regularization fails when the chosen time step is too low or the rod material is too soft. Going further into details, they showed that there exists a critical time step value Δt_c below which the singularity reappears and the chosen numerical solver fails to converge. They suggested that Δt_c follows the following law:

$$\Delta t_c \propto \chi l^2 \sqrt{\frac{\rho A}{EI}} \quad (3.1)$$

Recall that l denotes the rod length, ρ the density, A the cross section area, E Young's modulus, and I the second moment of inertia. Here, χ is a dimensionless prefactor that depends on imposed forces and motion, machine accuracy and the numerical solver. In particular, the value of χ increases as the imposed forces and motion become more significant, thereby requiring a larger critical time step Δt_c . In their numerical results, they found $0.5 \text{ ms} \lesssim \Delta t_c \lesssim 3 \text{ ms}$, depending on the simulation scenario. Their numerical results were obtained by using only the shooting method as numerical solver. Indeed, this method has become a standard in the field, as it is fast and easy to implement [61, 37].

On the other hand, simulating catheter navigation presents several challenges. The catheter experiences significant forces and moments at its proximal end, undergoes frictional contact along its length, and can be highly flexible. Moreover, the numerical method should be able to account for sufficiently small time steps to ensure accurate results. Consequently, these objectives are not aligned with what can be achieved using state-of-the-art solutions for the Cosserat rod boundary value problem.

Contributions

For a given simulation scenario, the only variable that can affect the critical time step below which the singularity reappears is χ . From the work of Boyer et al. [66], we know that χ depends on imposed forces and motion, machine accuracy, and the numerical solver. Consequently, for a given simulation scenario, one can only decrease the critical time step by either using a more stable numerical solver or increasing machine accuracy. In this work, we explore the first option.

To the best of our knowledge, the dynamic boundary value problem has never been solved using a collocation method. In [92], we numerically showed that, compared to the shooting method, an orthogonal collocation method is more likely to converge when the initial guess is far from the solution. Although the proposed method showed improved stability, it did not address the singularity issue. In this chapter, we further improve the method initially proposed in [92] with the objective of numerically addressing the singularity problem.

We propose a solution for the dynamic boundary value problem based on an orthogonal collocation method. The static mode will also be addressed as a subproblem. Here, we focus solely on solving the clamped-free problem, in which the rod is clamped at one end and free at the other. The free-free problem, where the rod is free at both ends, is better suited for simulating catheter navigation, will be addressed with frictional contact in the next chapter. The main contributions are summarized as follows:

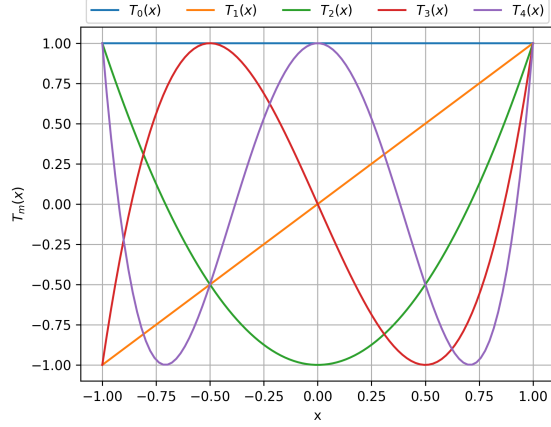
- Proposing an orthogonal collocation method for solving dynamic Cosserat rods and providing its Jacobian matrix analytically (Sections 3.3 and 3.4).
- Demonstrating the capability of the proposed method to solve the singularity problem using multiple nonlinear benchmarks from the literature and comparing it to the shooting method and the Lagrangian model (Section 3.5).
- Making the implementation open source at gitlab.inria.fr/rjilani/ch3.

This chapter is structured as follows. Section 3.1 introduces the time discretization approach, following Boyer et al. [66]. Section 3.2 describes the orthogonal polynomial interpolation used for the collocation method, as detailed in [80]. Section 3.3 proposes a configuration space reduction. Section 3.4 presents a collocation method for solving the clamped-free boundary value problem and provides its Jacobian matrix analytically. Section 3.5 presents our numerical experiments, followed by concluding remarks in Section 3.6.

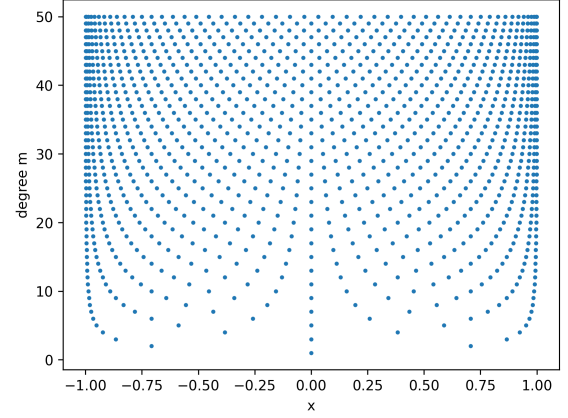
3.1 Time Discretization

The clamped-free boundary value problem (2.17), (2.18), at its original form is unsolvable because the acceleration and velocity of the space-rate twist $\ddot{\xi}$ and $\dot{\xi}$ are unknown. Indeed, this boundary value problem originates from a singular optimal control problem [66]. To regularize it, an implicit time integration is applied to $\ddot{\xi}$ and $\dot{\xi}$, allowing them to be computed using ξ and known values from prior time steps.

Let j and $j + 1$ represent two consecutive time steps, and suppose we want to move from step j to step $j + 1$, meaning that all variables at step j are known. As in Boyer et al. [66], the Newmark scheme is used:



(a) Chebyshev polynomials of degree up to 4.



(b) Chebyshev roots with respect to polynomial degree.

Figure 3.1: Chebyshev basis and their roots for multiple polynomial degrees.

$$\xi_{j+1} = a\ddot{\xi}_{j+1} + f_j, \quad \dot{\xi}_{j+1} = b\ddot{\xi}_{j+1} + h_j. \quad (3.2)$$

The space-rate twist ξ_{j+1} will be computed using a collocation method in Section 3.4. In the Newmark scheme, the coefficients a and b as well as the auxiliary variables f_j and h_j are defined by:

$$\begin{aligned} a &= \beta\Delta t^2, \quad b = \gamma\Delta t \\ f_j &= \xi_j + \Delta t\dot{\xi}_j + \Delta t^2\left(\frac{1}{2} - \beta\right)\ddot{\xi}_j \\ h_j &= \dot{\xi}_j + \Delta t(1 - \gamma)\ddot{\xi}_j \end{aligned} \quad (3.3)$$

where Δt is the time step, and β and γ are two parameters that can be adjusted to control damping. To achieve no damping, the parameters are set to $(\beta, \gamma) = (\frac{1}{4}, \frac{1}{2})$, throughout this dissertation. Furthermore, other approaches could also be used for implicit integration, such as the backward differentiation formula [61].

Now that the boundary value problem is regularized, we can introduce the collocation method to solve it. However, before doing so, the next section presents essential polynomial interpolation.

3.2 Chebyshev Interpolation

A continuous function can be approximated using a polynomial of degree $m - 1$ that matches the function values at m distinct points. By increasing the degree, the polynomial approaches the true function. A fast yet accurate way to construct this polynomial is to express it as a weighted sum of orthogonal basis functions, which allows computing its coefficients in closed form. Inspired by the work of Orekhov et al. [78] on static Cosserat rods, our work uses Chebyshev polynomials. Indeed, the Chebyshev basis is well conditioned, which ensures

numerical stability for methods like collocation [80]. This section briefly presents Chebyshev interpolation, for more details, we refer the reader to [80].

The Chebyshev polynomials of the first kind of degree m , $T_m : [-1, 1] \rightarrow [-1, 1]$, can be computed using trigonometric functions as follows:

$$T_m(x) = \cos(m \arccos(x)). \quad (3.4)$$

Or, more efficiently, using the following recurrence relation:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_m(x) &= 2xT_{m-1}(x) - T_{m-2}(x), \quad m \geq 2. \end{aligned} \quad (3.5)$$

Fig. 3.1a plots the Chebyshev polynomials up to degree 4. A continuous function $\tilde{f} : [-1, 1] \rightarrow [-1, 1]$ can be approximated using a weighted sum of Chebyshev polynomials up to degree $m - 1$ as follows:

$$\tilde{f}(x) \simeq \frac{1}{2}e_0T_0(x) + \sum_{j=1}^{m-1} e_jT_j(x), \quad (3.6)$$

where $\{e_j\}$ are the modal coefficients. They are computed efficiently using the discrete Chebyshev transform:

$$e_j = \frac{2}{m} \sum_{i=1}^m \tilde{f}(\tau_i^*) T_j(\tau_i^*), \quad (3.7)$$

where $\{\tau_i^*\}$ are the interpolation nodes. The discrete Chebyshev transform requires $\{\tau_i^*\}$ to be the roots of the Chebyshev polynomial T_m . The Chebyshev polynomial of degree m has m roots, computed as follows

$$\tau_{m+1-i}^* = \cos\left(\frac{\left(i - \frac{1}{2}\right)\pi}{m}\right), \quad i = 1, \dots, m. \quad (3.8)$$

By increasing m , the points τ_1^* and τ_m^* approach the boundaries of the interval $[-1, 1]$ but never reach them exactly. Furthermore, the roots $\{\tau_i^*\}$ are not uniformly spaced. Fig. 3.1b plots these roots for polynomial degrees up to 50. Apart from being essential for the discrete Chebyshev transform, the use of $\{\tau_i^*\}$ as interpolation nodes reduces the Runge phenomenon. For instance, this phenomenon occurs when oscillations appear near the boundaries due to equally spaced interpolation nodes. Moreover, the derivatives of $\tilde{f}$ at τ_i^* can be approximated efficiently using a linear combination of these values:

$$\begin{pmatrix} \frac{d\tilde{f}(\tau_1^*)}{dx} \\ \frac{d\tilde{f}(\tau_2^*)}{dx} \\ \vdots \\ \frac{d\tilde{f}(\tau_m^*)}{dx} \end{pmatrix} \simeq D_m^{*T} \begin{pmatrix} \tilde{f}(\tau_1^*) \\ \tilde{f}(\tau_2^*) \\ \vdots \\ \tilde{f}(\tau_m^*) \end{pmatrix} \quad (3.9)$$

Here, $D_m^* \in \mathbb{R}^{m \times m}$ is the Chebyshev differentiation matrix. This matrix is constant, its elements at row i and column j are computed using first and second derivatives of Chebyshev polynomials:

$$D_m^*(i, j) = \begin{cases} \frac{1}{2} \frac{T_{m,2}(\tau_i^*)}{T_{m,1}(\tau_i^*)} & \text{if } i = j, \\ \frac{1}{\tau_j^* - \tau_i^*} \frac{T_{m,1}(\tau_j^*)}{T_{m,1}(\tau_i^*)} & \text{if } i \neq j. \end{cases} \quad (3.10)$$

The derivatives $T_{m,1} = \frac{dT_m}{dx}$ and $T_{m,2} = \frac{d^2T_m}{dx^2}$ are computed in closed form and can be found in [80].

Now that we have established the theoretical foundation for Chebyshev polynomial interpolation, we have all the necessary components to introduce the configuration space reduction used for the collocation method.

3.3 Configuration Space Reduction

Notice that if the stress $\Lambda(X)$ is known, one can integrate the kinematics (2.3) ($g' = g\hat{\xi}$) using the initial condition $g(0) = 1_{4 \times 4}$, with ξ computed from Λ by inverting the constitutive law

$$\xi = \mathcal{H}^{-1}\Lambda + \xi_o \quad (3.11)$$

to obtain the cross-section poses $g(X)$. Therefore, $(g(0), \Lambda)$ completely define the rod configuration. Consequently, The configuration space can be expressed as

$$\mathcal{C} = SE(3) \times \{\Lambda : X \in [0, l] \rightarrow \Lambda(X) \in \mathbb{R}^6\}. \quad (3.12)$$

We propose to approximate the six functions of Λ using Chebyshev polynomials as explained in the last section. Polynomials up to a degree $m - 1$ are used, with exactly m interpolation nodes. Furthermore, the Chebyshev polynomials must be shifted from their original domain $[-1, 1]$ to the arc length domain $[0, l]$. To simplify notations and avoid extrapolation, the Chebyshev polynomials are shifted using the linear transformation $\varsigma : [0, l] \rightarrow [\tau_1^*, \tau_m^*]$. Consequently, the first and last interpolation nodes in the arc length domain correspond to the base and tip of the rod, respectively. In other words, the domain of definition of Chebyshev polynomials, $[-1, 1]$, is mapped through ς^{-1} to an extended arc length domain $[\tilde{x}, \tilde{y}]$, with $\tilde{x} < 0 < l < \tilde{y}$, where only the portion $[0, l]$ is actually used. Notice that $\tilde{x}$ and $\tilde{y}$ depend on the chosen polynomial degree.

Let $\tau_i = \varsigma^{-1}(\tau_i^*)$, $i = 1, \dots, m$, be the interpolation nodes in the arc length domain, where $\tau_1 = 0$ and $\tau_m = l$. The weighted sums (3.6) and (3.7) can be expressed in a matrix form, taking into account the mapping of Chebyshev polynomials domain and the interpolation of the six stress functions:

$$\Lambda(X) \simeq \frac{2}{m} (\Lambda(\tau_1), \Lambda(\tau_2), \dots, \Lambda(\tau_m)) \begin{pmatrix} T_0(\tau_1^*), T_1(\tau_1^*), \dots, T_{m-1}(\tau_1^*) \\ T_0(\tau_2^*), T_1(\tau_2^*), \dots, T_{m-1}(\tau_2^*) \\ \vdots \\ T_0(\tau_m^*), T_1(\tau_m^*), \dots, T_{m-1}(\tau_m^*) \end{pmatrix} \begin{pmatrix} \frac{1}{2}T_0(\varsigma(X)) \\ T_1(\varsigma(X)) \\ \vdots \\ T_{m-1}(\varsigma(X)) \end{pmatrix} \quad (3.13)$$

Let $\Lambda_m \in \mathbb{R}^{6 \times m}$ be the stress at the interpolation nodes

$$\Lambda_m = (\Lambda(\tau_1), \Lambda(\tau_2), \dots, \Lambda(\tau_m)) \quad (3.14)$$

and the vector $\Phi(X) \in \mathbb{R}^m$ be

$$\Phi(X) = \frac{2}{m} \begin{pmatrix} T_0(\tau_1^*), T_1(\tau_1^*), \dots, T_{m-1}(\tau_1^*) \\ T_0(\tau_2^*), T_1(\tau_2^*), \dots, T_{m-1}(\tau_2^*) \\ \vdots \\ T_0(\tau_m^*), T_1(\tau_m^*), \dots, T_{m-1}(\tau_m^*) \end{pmatrix} \begin{pmatrix} \frac{1}{2}T_0(\varsigma(X)) \\ T_1(\varsigma(X)) \\ \vdots \\ T_{m-1}(\varsigma(X)) \end{pmatrix} \quad (3.15)$$

then (3.13) becomes:

$$\Lambda(X) \simeq \Lambda_m \Phi(X) \quad (3.16)$$

Consequently, $g(X)$ can be obtained with $(g(0), \Lambda_m)$ and therefore the configuration space is reduced to

$$\mathcal{C} = SE(3) \times \mathbb{R}^{6 \times m}. \quad (3.17)$$

Finally, the approximation of Λ arc length derivative at interpolation nodes $\{\tau_i\}$ using Chebyshev polynomials is expressed as:

$$\Lambda'_m \simeq \Lambda_m D_m \quad (3.18)$$

Here, $D_m \in \mathbb{R}^{m \times m}$ is the Chebyshev differentiation matrix mapped to arc length domain

$$D_m(i, j) = \begin{cases} \frac{1}{2} \frac{T_{m,2}(\tau_i^*)}{T_{m,1}(\tau_i^*)} & \text{if } i = j, \\ \frac{1}{\tau_j - \tau_i} \frac{T_{m,1}(\tau_j^*)}{T_{m,1}(\tau_i^*)} & \text{if } i \neq j. \end{cases} \quad (3.19)$$

In the next section, the configuration space reduction presented here is used to solve the boundary value problem with a collocation method.

3.4 Collocation Method

In this section, we propose to solve the clamped-free boundary value problem (2.17) (2.18) using a collocation method. The differential equation we want to solve is the continuous balance equation (2.8), the explicitly enforced boundary condition is $\Lambda(l) = F_+$, the basis functions are the Chebyshev polynomials, and the collocation points are the shifted

Chebyshev roots $\{\tau_i\}$ (interpolation nodes). Finally, the unknown parameters are the collocation values Λ_m .

3.4.1 System of Nonlinear Algebraic Equations

We formulate a system of nonlinear algebraic equations, $\mathcal{R}_c : \mathbb{R}^{6 \times m} \rightarrow \mathbb{R}^{m \times 6}$, which allows us to solve for the unknown collocation values Λ_m :

$$\begin{aligned} \mathcal{R}_c(\Lambda_m) &= \begin{pmatrix} \Lambda'(\tau_1) \\ \Lambda'(\tau_2) \\ \vdots \\ \Lambda'(\tau_{m-1}) \\ \Lambda(\tau_m) \end{pmatrix} - \begin{pmatrix} \tilde{\Lambda}'(\tau_1) \\ \tilde{\Lambda}'(\tau_2) \\ \vdots \\ \tilde{\Lambda}'(\tau_{m-1}) \\ F_+ \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta + ad_{\xi}^T \Lambda - \bar{F}|_{X=\tau_1} \\ \mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta + ad_{\xi}^T \Lambda - \bar{F}|_{X=\tau_2} \\ \vdots \\ \mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta + ad_{\xi}^T \Lambda - \bar{F}|_{X=\tau_{m-1}} \\ \Lambda(\tau_m) \end{pmatrix} - \begin{pmatrix} \tilde{\Lambda}'(\tau_1) \\ \tilde{\Lambda}'(\tau_2) \\ \vdots \\ \tilde{\Lambda}'(\tau_{m-1}) \\ F_+ \end{pmatrix} \end{aligned} \quad (3.20)$$

Here, $\tilde{\Lambda}'(\tau_i)$ is the counterpart of $\Lambda'(\tau_i)$, computed using the Chebyshev differentiation matrix (3.18), $\xi(\tau_i)$ is computed by inverting the constitutive law (3.11), $\eta(\tau_i)$ and $\dot{\eta}(\tau_i)$ are computed by integrating their differential equations (2.17), where $\dot{\xi}(X)$ and $\ddot{\xi}(X)$ are computed using implicit integration (3.2). If $\bar{F}(\tau_i)$ or F_+ are originally expressed in the inertial frame, then $g(\tau_i)$ is required to rotate them to the mobile frame. We compute $g(\tau_i)$ by integrating its differential equation (2.17). While integrating g' , η' , and $\dot{\eta}'$, values of Λ at points other than the collocation points τ_i are required. They are calculated by Chebyshev interpolation (3.16). In the static case, the same method is used to solve the boundary value problem (2.14), (2.15), but without computing the velocities and accelerations η and $\dot{\eta}$. Finally, the steps for assembling the residual vector are detailed in Algorithm 1.

The system (3.20) is solved for Λ_m by using either least square minimization or root finding techniques. This iterative computation requires an initial guess of Λ_m . In statics, the initial guess is set to zero for the first loading step and the solution from the previous step for subsequent ones. In dynamics, it starts with the static solution (with which the simulation begins) for the first time step and uses the previous time step solution for the rest.

3.4.2 Jacobian Matrix

The iterative computation of Λ_m requires the Jacobian $J_c = \frac{\partial \text{vec}(\mathcal{R}_c)}{\partial \text{vec}(\Lambda_m)}$, where the operator vec transforms a matrix into a column vector. We denote the differentiation $\bullet^\dagger = \frac{\partial \bullet}{\partial \Lambda_k(\tau_i)}$, where $k \in \{1, 2, \dots, 6\}$ represents the six components of Λ , i.e., $(\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4, \Lambda_5, \Lambda_6)^T$. In the following, we will derive all necessary components of $\mathcal{R}_c^\dagger$ that permits to construct the Jacobian J_c column by column.

Algorithm 1 Residual Vector Computation

- Knowns:* $\{\Lambda(\tau_1), \Lambda(\tau_2), \dots, \Lambda(\tau_m)\}$, D_m , $\mathcal{H}$, $\mathcal{M}$, ξ_o , $\bar{F}$, F_+ , h_j , f_j , a , b
- 1: Define $\Lambda_m = (\Lambda(\tau_1), \Lambda(\tau_2), \dots, \Lambda(\tau_m))$ (3.14)
 - 2: Compute $\xi(X)$ using $\xi = \mathcal{H}^{-1}\Lambda + \xi_o$ (2.6), where $\Lambda(X) = \Lambda_m\Phi(X)$ (3.16)
 - 3: Compute $g(X)$ by integrating $g' = g\hat{\xi}$ (2.3) with $g(0) = 1_{4 \times 4}$
 - 4: Compute $\eta(X)$ by integrating $\eta' = -ad_\xi\eta + \dot{\xi}$ (2.11) with $\eta(0) = 0_{6 \times 1}$, using $\dot{\xi} = \dot{\xi}_{j+1} = b\ddot{\xi}_{j+1} + h_j$ (3.2)
 - 5: Compute $\dot{\eta}(X)$ by integrating $\dot{\eta}' = -ad_\xi\dot{\eta} - ad_\xi\eta + \ddot{\xi}$ (2.12) with $\dot{\eta}(0) = 0_{6 \times 1}$, using $\ddot{\xi} = \ddot{\xi}_{j+1} = \frac{1}{a}(\xi_{j+1} - f_j)$ (3.2)
 - 6: Compute $\{\tilde{\Lambda}'(\tau_k)\}$, $k = 1 \dots m-1$, using $\mathcal{M}\dot{\eta} - ad_\eta^T\mathcal{M}\eta + ad_\xi^T\Lambda - \bar{F}|_{X=\tau_k}$ (2.8)
 - 7: Compute $\{\tilde{\Lambda}'(\tau_k)\}$, $k = 1 \dots m-1$, using Λ_mD_m (3.18)
 - 8: Return $\left((\Lambda'(\tau_1) - \tilde{\Lambda}'(\tau_1))^T, \dots, (\Lambda'(\tau_{m-1}) - \tilde{\Lambda}'(\tau_{m-1}))^T, (\Lambda(\tau_m) - F_+)^T\right)^T$ (3.20)
-

$$\begin{aligned}
\mathcal{R}_c^\dagger(\Lambda_m) &= \begin{pmatrix} \Lambda'^\dagger(\tau_1) \\ \Lambda'^\dagger(\tau_2) \\ \vdots \\ \Lambda'^\dagger(\tau_{m-1}) \\ \Lambda'^\dagger(\tau_m) \end{pmatrix} - \begin{pmatrix} \tilde{\Lambda}'^\dagger(\tau_1) \\ \tilde{\Lambda}'^\dagger(\tau_2) \\ \vdots \\ \tilde{\Lambda}'^\dagger(\tau_{m-1}) \\ F_+^\dagger \end{pmatrix} \\
&= \begin{pmatrix} \mathcal{M}\dot{\eta}^\dagger - ad_{\eta^\dagger}^T\mathcal{M}\eta - ad_\eta^T\mathcal{M}\dot{\eta}^\dagger + ad_{\xi^\dagger}^T\Lambda + ad_\xi^T\Lambda^\dagger - \bar{F}^\dagger|_{X=\tau_1} \\ \mathcal{M}\dot{\eta}^\dagger - ad_{\eta^\dagger}^T\mathcal{M}\eta - ad_\eta^T\mathcal{M}\dot{\eta}^\dagger + ad_{\xi^\dagger}^T\Lambda + ad_\xi^T\Lambda^\dagger - \bar{F}^\dagger|_{X=\tau_2} \\ \vdots \\ \mathcal{M}\dot{\eta}^\dagger - ad_{\eta^\dagger}^T\mathcal{M}\eta - ad_\eta^T\mathcal{M}\dot{\eta}^\dagger + ad_{\xi^\dagger}^T\Lambda + ad_\xi^T\Lambda^\dagger - \bar{F}^\dagger|_{X=\tau_{m-1}} \\ \Lambda^\dagger(\tau_m) \end{pmatrix} - \begin{pmatrix} \tilde{\Lambda}'^\dagger(\tau_1) \\ \tilde{\Lambda}'^\dagger(\tau_2) \\ \vdots \\ \tilde{\Lambda}'^\dagger(\tau_{m-1}) \\ F_+^\dagger \end{pmatrix} \quad (3.21)
\end{aligned}$$

To compute $\mathcal{R}_c^\dagger$, observe first that $\Lambda(\tau_i)$ has no influence on the arc length X . Consequently, differentiation commute, i.e., $(\bullet')^\dagger = (\bullet^\dagger)'$. This property applies to g :

$$(g')^\dagger = (g^\dagger)' \quad (3.22)$$

Let us define the twist $\varphi = (g^{-1}g^\dagger)^\vee$. For instance, it is a counterpart to $\eta = (g^{-1}\dot{g})^\vee$, obtained by replacing $\dot{g}$ with $g^\dagger$. Then similar to obtaining (2.11) from (2.10), we can derive from (3.22) the following differential equation:

$$\varphi' = -ad_\xi\varphi + \xi^\dagger, \quad (3.23)$$

where differentiation of (3.11) yields

$$\xi^\dagger(X) = \mathcal{H}^{-1}\Lambda^\dagger(X), \quad (3.24)$$

and the derivative of (3.16) is

$$\Lambda^\dagger(X) = \Lambda_m^\dagger \Phi(X), \quad (3.25)$$

where $\Lambda_m^\dagger$ is obtained from:

$$\frac{\partial \Lambda(\tau_i)}{\partial \Lambda_k(\tau_j)} = \begin{cases} 0_{6 \times 1} & \text{if } \tau_i \neq \tau_j, \\ \delta_k & \text{if } \tau_i = \tau_j, \end{cases} \quad j = 1, 2, \dots, m. \quad (3.26)$$

Here, δ_k is a vector with all entries equal to 0 except for the k -th one. $g^\dagger = (R^\dagger, r^\dagger)$ is required if F_+ or $\bar{F}(\tau_i)$ are computed from their inertial frame counterpart $f_+ = (c_+^T, n_+^T)^T$, and $\bar{f}(\tau_i) = (\bar{c}(\tau_i)^T, \bar{n}(\tau_i)^T)^T$:

$$F_+ = \begin{pmatrix} R(l)^T & 0 \\ 0 & R(l)^T \end{pmatrix} f_+, \quad \bar{F}(\tau_i) = \begin{pmatrix} R(\tau_i)^T & 0 \\ 0 & R(\tau_i)^T \end{pmatrix} \bar{f}(\tau_i) \quad (3.27)$$

because their differentiation yields:

$$F_+^\dagger = \begin{pmatrix} R^\dagger(l)^T & 0 \\ 0 & R^\dagger(l)^T \end{pmatrix} f_+, \quad \bar{F}^\dagger(\tau_i) = \begin{pmatrix} R^\dagger(\tau_i)^T & 0 \\ 0 & R^\dagger(\tau_i)^T \end{pmatrix} \bar{f}(\tau_i). \quad (3.28)$$

We assume that the imposed wrenches f_+ and $\bar{f}$ are constant within each time step, and thus independent of the stress field and of Λ_m . $g^\dagger(X)$ is obtained with $g^\dagger = g\hat{\varphi}$, where φ is computed by integrating (3.23) using the initial condition $\varphi(0) = 0_{6 \times 1}$. Now we shall compute $\eta^\dagger(\tau_i)$ and $\dot{\eta}^\dagger(\tau_i)$. Recall that differentiation commute $(\bullet')^\dagger = (\bullet^\dagger)'$, then differentiating their arc length differential equations (2.17) with respect to $\Lambda_k(\tau_i)$ yields:

$$\begin{aligned} (\eta^\dagger)' &= -ad_{\xi^\dagger} \eta - ad_\xi \eta^\dagger + \dot{\xi}^\dagger \\ (\dot{\eta}^\dagger)' &= -ad_{\xi^\dagger} \dot{\eta} - ad_\xi \dot{\eta}^\dagger - ad_{\dot{\xi}^\dagger} \eta - ad_{\dot{\xi}} \eta^\dagger + \ddot{\xi}^\dagger \end{aligned} \quad (3.29)$$

and differentiating (3.2) gives:

$$\ddot{\xi}^\dagger = \frac{1}{a} \xi^\dagger, \quad \dot{\xi}^\dagger = b \ddot{\xi}^\dagger. \quad (3.30)$$

As a result, $\eta^\dagger$ and $\dot{\eta}^\dagger$ can be computed at any X by arc length integration with the initial conditions $\eta^\dagger(0) = 0_{6 \times 1}$ and $\dot{\eta}^\dagger(0) = 0_{6 \times 1}$. Finally, $\tilde{\Lambda}^\dagger(\tau_i)$ are nothing but $\tilde{\Lambda}_m^\dagger$, which in turn is computed by differentiating (3.18):

$$\tilde{\Lambda}_m^\dagger = \Lambda_m^\dagger D_m \quad (3.31)$$

The Jacobian matrix for the static mode is obtained similarly, but without considering the computation of η , $\dot{\eta}$, $\eta^\dagger$, and $\dot{\eta}^\dagger$.

Now that the collocation method has been completely described, in the next section, we compare it to two state-of-the-art methods across multiple numerical benchmarks.

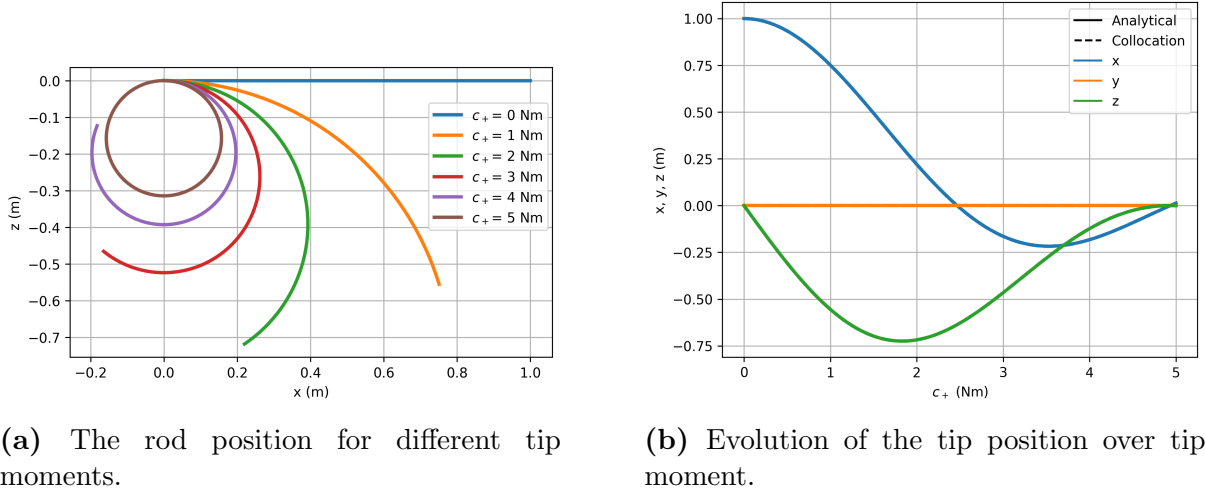


Figure 3.2: A straight rod is bent by applying a tip moment until it forms a circle.

3.5 Numerical Applications

In this section, we assess the proposed method on classical static and dynamic nonlinear benchmarks and compare it against an analytical solution, and the well-validated shooting method [66] and Lagrangian model [60]. As the implementations of these methods were not available, we implemented them as detailed in [66, 60]. For reproducibility, both implementations were made open-source on gitlab.inria.fr/rjilani/ch3 and gitlab.inria.fr/rjilani/lagrangian_model. All implemented methods (including ours) were coded in Python. We used SciPy’s Runge-Kutta 4(5) for arc length integration. For least square minimization of the residuals, we used SciPy’s *hybr* function, which implements a modified version of the hybrid Powell method. Although the Levenberg-Marquardt algorithm is the preferred choice in soft and continuum robotics, our experiments suggest that the hybrid method offers the same convergence guarantees while being faster. Numerical integrations and the root-finding technique were performed with SciPy’s built-in default tolerances. The simulations were performed on a computer equipped with an Intel Xeon W 2245 CPU running at 3.90 GHz.

3.5.1 Statics

Circular Bending using Tip Moment

The aim of the first test was to assess the proposed method against an analytical solution. We reproduced the test conducted in [60, 93, 94, 95]: a straight rod is bent using a tip moment until it forms a circle (Fig. 3.2a). The rod has a length of $l = 1$ m, a circular cross-section with a radius of $R_b = 0.01$ m, and a Young’s modulus of $E = 10^8$ Pa. The tip moment c_{2+} is incrementally increased from 0 to 5 Nm. Recall that for a vector $x \in \mathbb{R}^n$, we denote its components by $x = (x_1, x_2, \dots, x_n)^T$. We used polynomial degree $m = 1$ because the solution (stress) is linear in this scenario. The results were obtained with 100 loading steps. Fig. 3.2b shows the evolution of the tip position with respect to c_+ . As expected, the

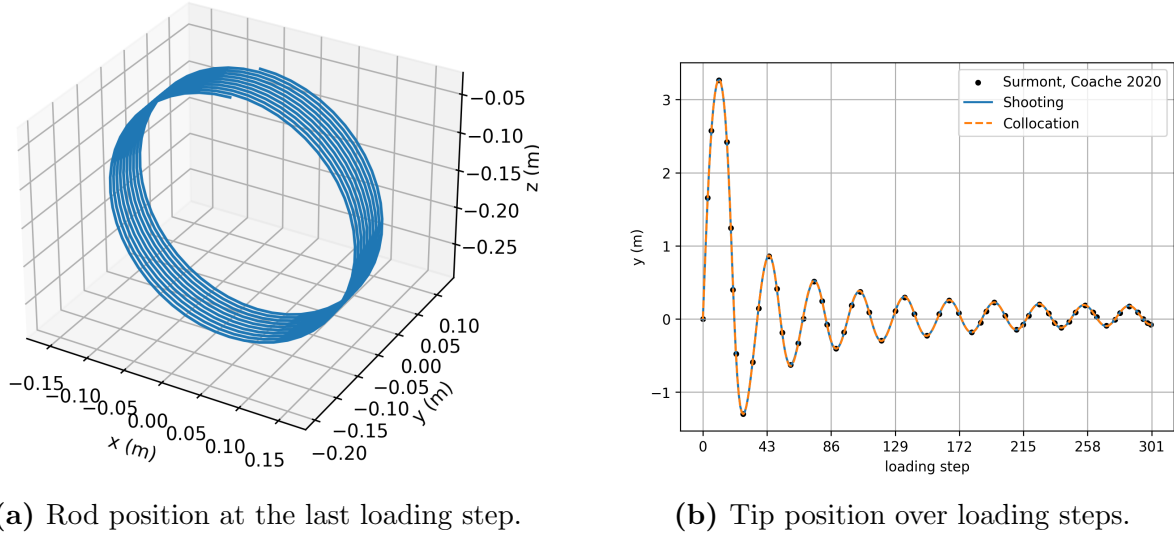


Figure 3.3: A straight rod is deformed into a helix by imposing tip force and moment.

results from the collocation method agree well with the analytical solution. Following [60], we computed the position error as follows:

$$e_{r1} = \max_{(X,t) \in [0,l] \times [0,T]} \frac{100}{l} |r_{\text{cm}} - r_{\text{analytical}}|(X, t) \quad (3.32)$$

where r_{cm} and $r_{\text{analytical}}$ are the rod position of the collocation method and the analytical solution, respectively. In the static mode, t represents a fictitious time corresponding to loading steps, while T denotes the total duration. Dividing by l is valid because the rod experiences deformations on the order of its length. We obtained an error $e_{r1} = 0.0539\%$. Finally, the total execution time was 2.91 seconds, giving an average of 0.0291 seconds per loading step.

Helical Bending using Tip Force and Moment

Arc length integration of rotation matrices using the kinematics (2.3) does not guarantee that they remain within the $SO(3)$ group. The aim of this test was to verify that this drift is avoided. We reproduced the test conducted in [84, 95, 96, 97, 98, 99]: a straight rod is deformed into a helix by imposing a tip force and moment (Fig. 3.3a). The rod length is $l = 10$ m, and the stiffness matrix is $\mathcal{H} = \text{diag}(10^2, 10^2, 10^2, 10^4, 10^4, 10^4)$. Furthermore, tip force n_{2+} and moment c_{2+} are incrementally and simultaneously increased from 0 to 50 N and from 0 to 200π Nm, respectively. We used polynomial degree $m = 10$ and 300 loading steps. Fig. 3.3b shows the tip position over the loading steps. The results were compared to our implementation of the shooting method of Boyer et al. [66], as well as the shooting method of Surmont and Coache [95], which uses the modified Rodrigues parameters to represent cross-section orientations. Indeed, in [95], it was shown that this parameterization effectively prevents $SO(3)$ drift. The results of the collocation method is in good agreement

Polynomial degree m	e_{r2} (%)	Avg. time (s)	Total time (min)
10	0.116	0.163	0.817
20	0.0920	0.677	3.38
30	0.0261	1.42	7.12
40	0.0116	3.18	15.9
50	0.0116	8.78	43.9

Table 3.1: Position error, total execution time, and average execution time per loading step for the collocation method, as a function of the polynomial degree m . For comparison, the shooting method required 0.598 minutes in total, with an average of 0.119 seconds per loading step.

with the shooting methods. This is due to the high-order integration scheme used (Runge-Kutta 4(5)), which limits the drift from $SO(3)$ even in complex configurations. In contrast, standard Euler integration is insufficient to ensure convergence.

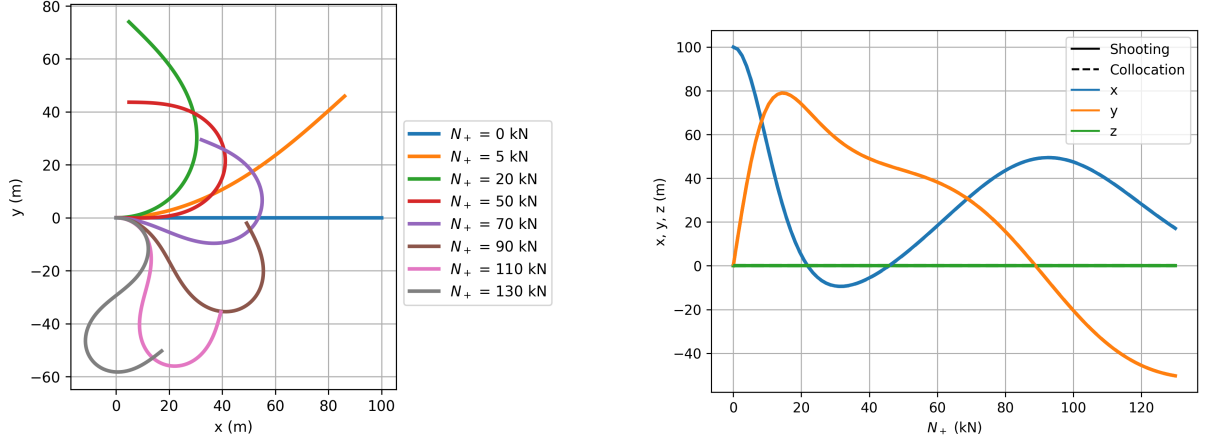
Furthermore, we increased the polynomial degree m from 10 to 50 to observe the evolution of the position error between the proposed method and the shooting method of [66]. The error was computed as follows:

$$e_{r2} = \max_{(X,t) \in [0,l] \times [0,T]} \frac{100}{l} |r_{cm} - r_{sm}|(X, t) \quad (3.33)$$

where r_{sm} is the rod position of the shooting method. Table 3.1 reports e_{r2} with respect to polynomial degree m . We note that the error remains low ($e_{r2} \leq 0.116\%$) and stabilizes for $m \geq 40$. Moreover, the shooting method total execution time was 0.598 minutes, with an average of 0.119 seconds per loading step. Table 3.1 presents the total and average per loading step computational times of the collocation method for different polynomial degrees. The shooting method is faster than the collocation method because its Jacobian matrix has smaller dimensions (6×6) compared to that of the collocation method ($6m \times 6m$). However, for a polynomial degree of $m = 10$, the collocation method yields a sufficiently low error, and its computational time is comparable to that of the shooting method.

Bending using a Follower Tip Force

The aim of this test was to assess the proposed method under a follower tip force, i.e., a force expressed in the mobile frame (Fig. 3.4a). We reproduced the test conducted in [60, 93, 94]. The rod has a length of $l = 100$ m, a circular cross-section with a radius of $R_b = 0.57$ m, and a Young's modulus of $E = 4.015 \times 10^8$ Pa. The follower tip force N_{2+} is incrementally increased from 0 to 130 kN. We used 100 loading steps with polynomial degree m ranging from 10 to 80. Fig. 3.4b shows the tip position as a function of the tip force for the collocation method with polynomial degree $m = 30$, and the shooting method. The results are in good agreement. The shooting method took 8.95 seconds, with an average of 0.0895 seconds per loading step. The collocation method took 6.85 seconds, with an average of 0.0685 seconds per loading step. The collocation method is faster in this scenario because no integration is required to solve the boundary value problem as the tip force (N_{2+}) is expressed in the mobile frame.



(a) The rod position for different tip forces.

(b) Evolution of the tip position over tip force.

Figure 3.4: Rod bending with a follower tip force.

Polynomial degree m	e_{r2} (%)	Avg. time (s)	Total time (s)
10	19.2	0.0388	3.88
20	0.578	0.0491	4.91
30	0.0682	0.0685	6.85
40	0.0672	0.0892	8.92
50	0.0672	0.114	11.4
60	0.0672	0.148	14.8
70	0.0672	0.203	20.3
80	0.0672	0.329	32.9

Table 3.2: Position error, total execution time, and average execution time per loading step for the collocation method, as a function of the polynomial degree m . For comparison, the shooting method required 8.95 seconds in total, with an average of 0.0895 seconds per loading step.

Furthermore, we computed the position error e_{r2} between the collocation and shooting methods with (3.33). Table 3.2 presents e_{r2} and execution time for different polynomial degrees. We observe that the error remains relatively low starting from $m = 20$ and stabilizes for $m \geq 30$. Moreover, as m increases, the growth in execution time is slower compared to the previous numerical application, as no arc length integration is performed here.

Inspired by [60], to evaluate the convergence of the Chebyshev approximation, we define an error metric $e_{\Lambda}(m, m + 10)$ that compares the stress fields obtained with polynomial degrees spaced by 10. This error measures the squared difference between solutions computed with degrees m and $m + 10$, integrated along the rod and averaged over all loading steps. It is defined as:

Error $\downarrow \setminus$ Component $\rightarrow$	K_3	Γ_1	Γ_2
$e_\Lambda(10, 20)$	2.39×10^{12}	7.73×10^9	5.53×10^9
$e_\Lambda(20, 30)$	5.06×10^8	1.36×10^6	2.33×10^6
$e_\Lambda(30, 40)$	14400	39.3	75.3
$e_\Lambda(40, 50)$	0.327	0.001	0.00189
$e_\Lambda(50, 60)$	5.10×10^{-6}	1.67×10^{-8}	2.96×10^{-8}
$e_\Lambda(60, 70)$	3.96×10^{-9}	3.66×10^{-13}	2.06×10^{-11}
$e_\Lambda(70, 80)$	2.09×10^{-9}	1.86×10^{-13}	1.08×10^{-11}

Table 3.3: Error $e_\Lambda(m, m + 10)$ between Chebyshev approximations of the stress field computed using polynomial degrees m and $m + 10$, over 100 loading steps. The error is averaged over time steps and integrated along the rod length. Values are reported for three representative stress components (K_3 , Γ_1 , and Γ_2) in a two-dimensional setting. The results show a rapid decay of the error, confirming the convergence of the collocation method, with negligible improvement beyond $m = 60$.

$$e_\Lambda(m, m + 10) = \frac{1}{n_{\text{steps}}} \sum_1^{n_{\text{steps}}} \int_0^l \left(\Lambda_j^{(m)}(X, t) - \Lambda_j^{(m+10)}(X, t) \right)^2 dX, \quad t \in [0, T] \quad (3.34)$$

Here, $n_{\text{steps}} = 100$ is the total number of loading steps, Λ_j is the j -th component of the stress Λ , and $\Lambda^{(m)}$ indicates that the stress is computed using a polynomial degree m . Table 3.3 presents e_Λ for polynomial degrees ranging from 10 to 80. Only three stress components (K_3 , Γ_1 , and Γ_2) are shown, as this is a two-dimensional experiment. The error decreases as m increases, and convergence is achieved for $m \geq 60$.

3.5.2 Dynamics

In this subsection, we use the Lagrangian model to compare with the collocation method, as comparison with the shooting method may not be feasible. Each time the Lagrangian model was used, it was parameterized to maximize accuracy. Specifically, for all numerical applications, we employed 8 shape function modes to capture the required deformations. Furthermore, we will report the maximum allowed arc length integration step for the Runge-Kutta 4(5) method, denoted as $\max_{dX}$. Indeed, the Runge-Kutta 4(5) may employ excessively large integration steps, which can delay the convergence of the numerical methods.

Cantilever Beam

The aim of this test was to compare the proposed method with the shooting method and the Lagrangian model. We begin with a scenario involving a stiff rod and small deformations. We reproduced the test conducted in [100, 101, 102]. The rod is initially straight at $t = 0$, then, for $t > 0$, a constant tip force $n_{3+} = -10$ N is applied for 0.5 seconds. The rod has a length of $l = 1$ m, a square cross-section with side $S_b = 0.01$ m, a Young's modulus of $E = 210 \times 10^9$

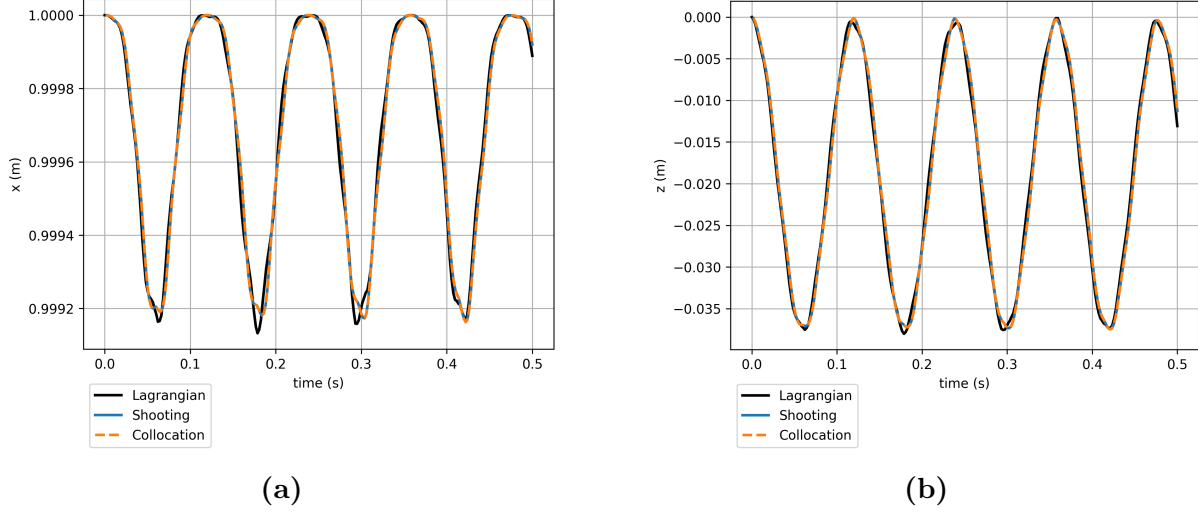


Figure 3.5: A tip force is applied to a straight rod for 0.5 seconds. The figures show tip position over time using $\Delta t = 0.002$ s. The collocation and shooting methods are in good agreement. However, due to the large time step, noticeable differences are observed when both methods are compared to the Lagrangian model.

Pa, a density of $\rho = 7800$ kg/m³, and a Poisson's ratio of $\nu = 0.2$. First, we chose a time step $\Delta t = 0.002$ s, along with a polynomial degree $m = 30$ and maximum allowed arc length integration step $\max_{dX} = \frac{l}{30}$ m. Fig. 3.5 shows the tip position over time. The results show good agreement between the shooting and collocation methods. However, compared to the Lagrangian model, noticeable differences are observed. To investigate the cause of these differences, we decreased the time step to $\Delta t = 0.0005$ s. The shooting method failed to converge, even after reducing $\max_{dX}$ and trying a different root-finding solver, suggesting that its critical time step is in the range $0.5 \text{ ms} \lesssim \Delta t_c < 2 \text{ ms}$. The collocation method initially failed as well but successfully converged when we increased the polynomial degree m to 70 and set $\max_{dX} = \frac{l}{m}$ m.

Fig. 3.6 shows the tip position over time. The Lagrangian and collocation methods are in good agreement. This test successfully confirmed that the difference was indeed caused by the large time step initially used. For the first test ($\Delta t = 0.002$ ms), the shooting method took 187 seconds, with an average of 0.749 seconds per time step, and the collocation method took 319 seconds, with an average of 1.27 seconds per time step. For the second test ($\Delta t = 0.0005$ ms), the collocation method took 4.42 hours to complete, with an average of 15.9 seconds per time step. The increase in average time is due to the increase in polynomial degree from $m = 30$ to $m = 70$, which in turn increases the Jacobian size and consequently the number of integrations. In contrast, the simulation with the Lagrangian model took 1.91 hours. This discrepancy in computation time is primarily explained by the fact that the Lagrangian model requires fewer shape function modes (here, 8) and thus fewer degrees of freedom to achieve good accuracy.

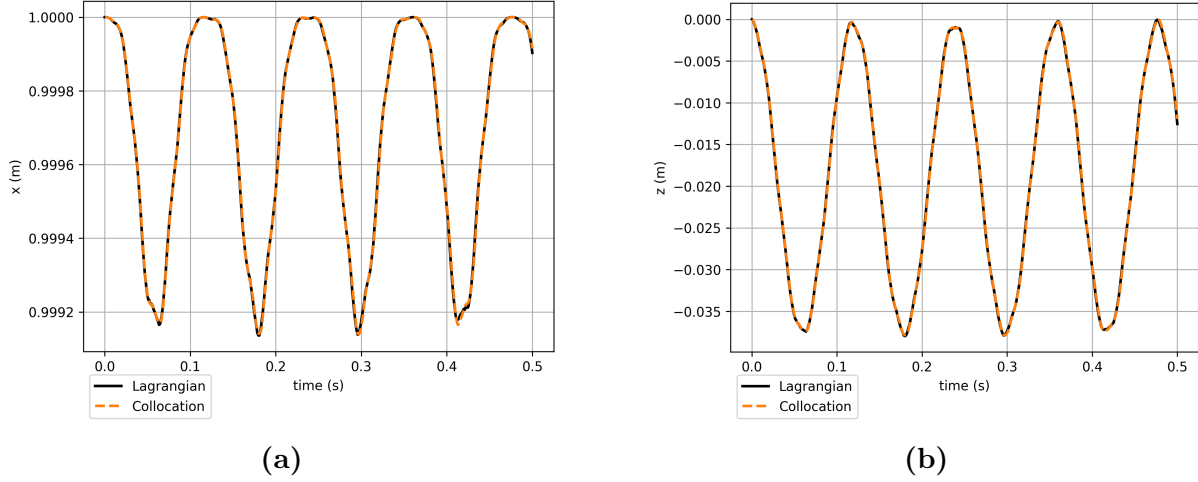


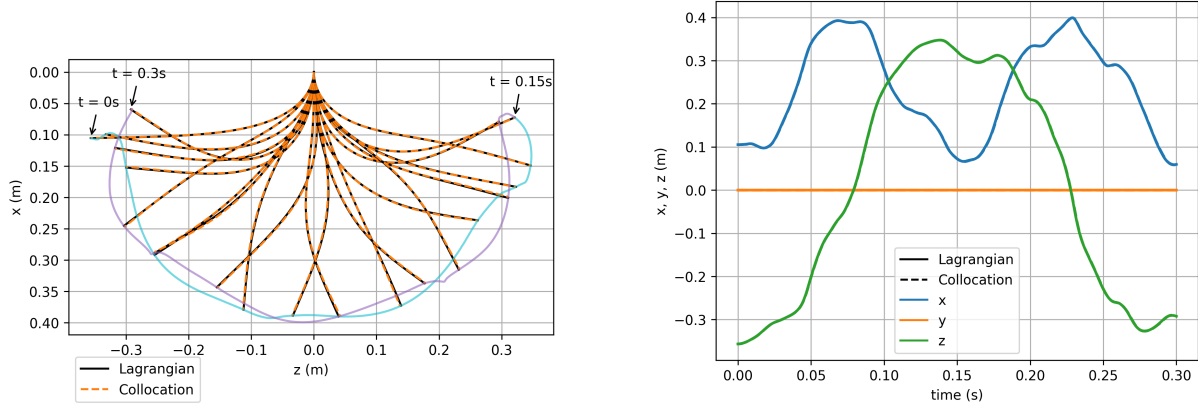
Figure 3.6: A tip force is applied to a straight rod for 0.5 seconds. The figures show tip position over time using $\Delta t = 0.0005$ s.

Hypothesis

The last numerical application unveiled a new hypothesis suggesting that increasing the polynomial degree m reduces the critical time step of the collocation method. To explore this further, recall that the critical time step is governed by the following law $\Delta t_c \propto \chi l^2 \sqrt{\frac{\rho A}{EI}}$ [66]. The hypothesis suggests that increasing the polynomial degree m decreases χ , enabling the use of smaller time steps. This hypothesis will be tested in the following two numerical applications by increasing $l^2 \sqrt{\frac{\rho A}{EI}}$, reducing Δt , and verifying that there exists a value of m beyond which the proposed method converges. To account for different rod materials, the next two numerical applications use varying mechanical parameters. This may seem inconsistent when comparing results, however, the value of $l^2 \sqrt{\frac{\rho A}{EI}}$ provides an objective measure of simulation difficulty. In the last numerical application, we had: $l^2 \sqrt{\frac{\rho A}{EI}} = 0.0667$ s.

Rod Bent and Released

Compared to the first test, this numerical application involves a softer rod and a greater magnitude of deformation. Inspired by [66], a straight vertical rod is initially bent towards the left with a tip force of $n_{3+} = -5$ N and then released for 0.3 seconds (Fig. 3.7a). The rod has a length of $l = 0.4$ m, a square cross-section with a side length of $S_b = 0.002$ m, a Young's modulus of $E = 207 \times 10^8$ Pa, a Poisson's ratio of $\nu = 0.2$, and a density of $\rho = 8000$ kg/m³. This results in $l^2 \sqrt{\frac{\rho A}{EI}} = 0.172$ s, representing an increase of more than two times compared to the previous numerical application. To match the accuracy of the Lagrangian model, we set $\Delta t = 2.5 \times 10^{-4}$ s. The shooting method failed to converge, whereas the collocation method required a minimum of $m = 100$ and $\max_d X = \frac{l}{m}$ m for successful convergence.



(a) Rod position every 0.015 seconds. Cyan and purple lines show tip evolution over time, for the first half, and second half of the simulation, respectively.

(b) Tip position over time.

Figure 3.7: A straight rod is initially bent with a tip force and is then released for 0.3 seconds.

Fig. 3.7a displays the rod position every 0.015 seconds, and Fig. 3.7b shows the tip position over time. We observe good agreement between the collocation and Lagrangian methods. We computed the position error as:

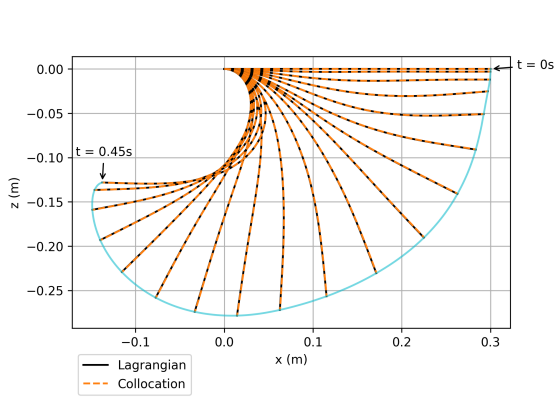
$$e_{r3} = \max_{(X,t) \in [0,l] \times [0,T]} \frac{100}{l} |r_{\text{cm}} - r_{\text{Lagrangian}}|(X, t) \quad (3.35)$$

Here, $r_{\text{Lagrangian}}$ is the rod position of the Lagrangian model. We obtained $e_{r3} = 0.857\%$. Thus, our hypothesis is successfully verified for the second time. The simulation took 22 minutes to complete with the Lagrangian model, and 3.96 hours with an average of 11.9 seconds per time step for the collocation method.

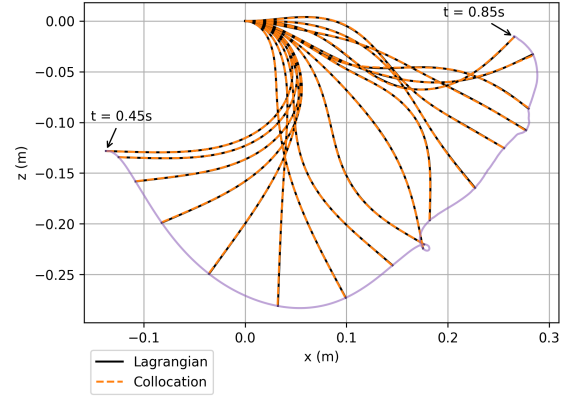
Rubber Rod Released in Gravity

The objective was to test our method on a rod made of a very soft material in a scenario with non-negligible shear deformations. Inspired by the swinging pendulum test [4, 64, 101], a clamped rubber rod is released under gravity for 0.85 seconds (Fig. 3.8). The rod has a length of $l = 0.3$ m, a square cross-section with a side length of $S_b = 0.005$ m, a Young's modulus of $E = 5 \times 10^6$ Pa, a Poisson's ratio of $\nu = 0.5$, and a density of $\rho = 1100$ kg/m³. This results in $l^2 \sqrt{\frac{\rho A}{EI}} = 0.924$ s, representing an increase of more than five times compared to the previous numerical application. As before, we chose $\Delta t = 2.5 \times 10^{-4}$ s to match the accuracy of the Lagrangian model. The shooting method failed to converge, while the collocation method required approximately $m = 200$ for successful convergence, with $\max_{dX} = \frac{l}{m}$ m.

Fig. 3.9 shows the tip position over time, and Fig. 3.8 shows the rod position every 0.025 seconds. The collocation and Lagrangian methods are in good agreement. Furthermore, we obtained an error $e_{r3} = 0.574\%$. Thus, our proposed hypothesis is verified for the third time.



(a) Rod position every 0.025 seconds for the first half of the simulation.



(b) Rod position every 0.025 seconds for the second half of the simulation.

Figure 3.8: A rubber rod is released in gravity for 0.85 seconds.

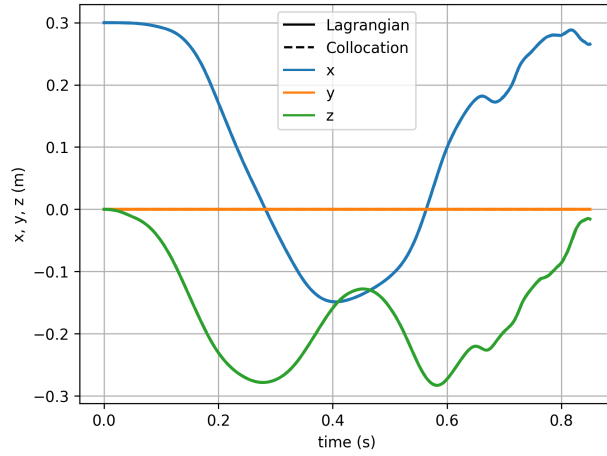


Figure 3.9: A rubber rod is released in gravity for 0.85 seconds. The figure shows tip position over time.

The simulation took 5.1 hours to complete with the Lagrangian model, and 14.7 hours with an average of 15.6 seconds per time step for the collocation method.

3.6 Conclusion

It is well known that solving the dynamic boundary value problem of the Cosserat rod model is numerically challenging. In particular, using small time steps was infeasible. This phenomenon was initially attributed to machine accuracy issues [61]. However, recent work [66] established a connection between optimal control theory and the dynamic Cosserat model, and showed that the optimal control problem underlying the dynamic boundary value problem of Cosserat rods is inherently singular. To regularize it, implicit time integration

must be used. However, the singularity manifests again when the rod is soft or when the time step falls below a critical value.

The objective of this chapter was to address the singularity problem. As most results reporting singularity came from the shooting method [61, 66], we proposed an orthogonal collocation method based on Chebyshev polynomials. A basis known to be well-conditioned [80]. The collocation solves for the strong form of the dynamic balance equation by computing the stress. Moreover, we provided its Jacobian matrix analytically.

We tested the proposed method in six nonlinear static and dynamic applications from the literature. In the static mode, when compared to an analytical solution, we obtained a position error of 0.05 %. When compared to the shooting method, we obtained a position error below 0.5 %. Furthermore, computation times between the shooting and collocation methods were comparable, with the collocation method being faster when the test required solving the problem only in the mobile frame, as it could avoid arc length integration. In the dynamic mode, the initial goal was to explore whether the collocation method could surpass what could be achieved using the shooting method. In tests where the rod was rigid, experienced small deformations, and had a large time step, both the shooting and collocation methods converged without much difficulty, and the results were in good agreement. As expected, however, when the rod was soft, experienced large deformations, and a low time step was used, the shooting method did not converge. Indeed, small changes in its initial condition at the proximal end resulted in large errors at the distal end, making its Jacobian matrix ill-conditioned. On the other hand, the collocation method also did not converge when a low polynomial degree was used. Then, we discovered that when the polynomial degree was increased above a certain value, the proposed method always converged. To validate the results, we compared them against the Lagrangian model and obtained good agreement for all three dynamic tests, and a position error lower than 0.9 %. This discovery would not have been possible without the provided Jacobian matrix. It allowed arc length integral computations to be grouped within a single function, improving time efficiency and enabling the use of high-degree polynomials (up to 200) with an execution time of 15 seconds per time step. Furthermore, the experiments showed that the Lagrangian model achieves good accuracy while being faster than the proposed method, as it requires fewer degrees of freedom.

We conclude that the proposed method successfully addressed the singularity in the boundary value problem of the dynamic Cosserat rod model. Consequently, the boundary value problem solving could be applied to a broader range of applications, including soft and medical robotics. On the other hand, our original goal was to develop a numerically accurate, robust, and fast solution for Cosserat rods. While we achieved good numerical accuracy and robustness, speed remains a challenge, and like many existing methods the proposed one, in its current state, is not suitable for fast (complex) simulations.

Future work includes addressing the question of why increasing the polynomial degree transforms the ill-conditioned problem into a well-conditioned one. Furthermore, although computing the Jacobian matrix has improved time efficiency, it still accounts for two-thirds of the total computation time. More than 90 % of the Jacobian computation time is spent on arc length integration. Indeed, for a polynomial degree $m - 1$, there are $m \times 3 \times 6$ independent integrations of φ' , $(\eta^\dagger)'$, and $(\dot{\eta}^\dagger)'$, which could benefit from CPU or GPU parallelization.

The study reported in this chapter did not take contact into account. In the next chapter, we will use the proposed method to solve the dynamics of Cosserat rods undergoing frictional contact, with the goal of advancing towards a simulation tailored for catheter navigation.

Chapter 4

Dynamic Cosserat Rods in Contact with Implicit Surfaces

In the last chapter, we proposed a collocation method for solving the clamped-free boundary value problem of dynamic Cosserat rods. We also demonstrated through numerical experiments that the proposed method effectively addresses the singularity issue. As a result of this improved stability, it can now be applied to challenging simulations. In this chapter, we use it to simulate dynamic Cosserat rods undergoing frictional contact with tubular surfaces, aiming to develop a method specifically tailored for the simulation of catheter navigation. In [103], we proposed a solution using the shooting method, where the singularity problem was overcome with damping. In this chapter, the proposed collocation method eliminates the need for damping.

Contributions

In catheter navigation, medical tools are manually manipulated by the practitioner at their proximal end through displacement and rotation. Consequently, the clamped-free boundary value problem, which models (partially) fixed rods, is not directly suitable for our application. Therefore, we first extend the collocation method proposed in the previous chapter to solve free-free boundary value problems, where the rod is free at both ends, and forces and moments applied at the proximal end can mimic the displacement and rotation of the medical tool performed by the practitioner. Moreover, incorporating contact forces into the Cosserat rod boundary value problem is not straightforward. This complexity arises because external forces are included in the boundary value problem as distributed forces, whereas penalty contact forces are concentrated (line) forces. For instance, Wiese et al. [76] did not address this issue when solving the static boundary value problem for Cosserat rods in contact. In this work, we propose an approach to effectively convert between these two types of forces. Furthermore, a key challenge is how to represent the tubular surfaces that model human vasculature. Indeed, the choice of representation affects contact detection, which is known to be computationally expensive. To address this, we model our contact surfaces using implicit functions, which enable efficient evaluations.

Our contributions can be summarized as follows:

- Developing a collocation method that solves the dynamic free-free boundary value problem for Cosserat rods, providing its Jacobian matrix and validating it against an analytical solution.
- Modeling contact surfaces using implicit functions, making contact detection execution time negligible.
- Testing our method in a real catheter navigation setup that includes a model of a carotid artery and actual catheter material properties.
- Proposing a method tailored for the simulation of catheter navigation in the dynamic regime.
- Making the implementation open-source at gitlab.inria.fr/rjilani/ch4.

This chapter is organized as follows. Section 4.1 describes the time discretization as formulated in [66]. Section 4.2 extends the collocation method to free-free boundary value problems. Section 4.3 details the implicit surface modeling process. Section 4.4 discusses contact handling, including detection and response. Sections 4.5 and 4.6 describe the inclusion of contact forces in the boundary value problem. Section 4.7 presents three numerical applications. Finally, Section 4.8 provides the concluding remarks.

4.1 Time Discretization

In this section, we describe the time discretization required to solve the free-free boundary value problem (2.17), (2.19). We adopt the discretization approach introduced by Boyer et al. [66] and used for solving Cosserat rods boundary value problems via the shooting method. First, the time discretization of space-rate twists time derivatives $\dot{\xi}$ and $\dot{\xi}$, originally described for clamped-free problems and presented in Section 3.1, also applies here. Therefore, we omit their derivation in this section and refer to (3.2) and (3.3). Furthermore, in the free-free case, additional unknowns arise, requiring specific treatment through time discretization. Indeed, the free-free boundary value problem (2.17), (2.19) is underdetermined, as it provides only 2×6 boundary conditions, namely the stresses $\Lambda(0) = -F_-$ and $\Lambda(l) = F_+$ (2.19), to solve 4×6 differential equations of the cross-section pose g' , velocity η' , acceleration $\dot{\eta}'$, and stress Λ' (2.17). In particular, the initial conditions $g(0)$, $\eta(0)$, and $\dot{\eta}(0)$, which are necessary for integrating g' , η' , and $\dot{\eta}'$, are unknown. The goal of this section is to compute these initial conditions using time discretization. Before proceeding, we first define some variables to facilitate the use of $g(0)$, $\eta(0)$, and $\dot{\eta}(0)$.

$$\begin{aligned} g_0 &= (R_0, r_0) = (R(0), r(0)) = g(0) \\ \eta_0 &= (\Omega_0^T, V_0^T)^T = (\Omega(0)^T, V(0)^T)^T = \eta(0) \\ \dot{\eta}_0 &= (\dot{\Omega}_0^T, \dot{V}_0^T)^T = (\dot{\Omega}(0)^T, \dot{V}(0)^T)^T = \dot{\eta}(0) \end{aligned} \tag{4.1}$$

Since linear velocity V_0 and linear acceleration $\dot{V}_0$ can be expressed in terms of components in the inertial frame, consequently, η_0 and $\dot{\eta}_0$ can be written as:

$$\eta_0 = \begin{pmatrix} \Omega_0 \\ V_0 \end{pmatrix} = \begin{pmatrix} \Omega_0 \\ R_0^T \dot{r}_0 \end{pmatrix} \tag{4.2}$$

$$\dot{\eta}_0 = \begin{pmatrix} \dot{\Omega}_0 \\ \dot{V}_0 \end{pmatrix} = \begin{pmatrix} \dot{\Omega}_0 \\ R_0^T \ddot{r}_0 + (R_0^T \dot{r}_0) \times \Omega_0 \end{pmatrix} \quad (4.3)$$

The inertial frame linear velocity $\dot{r}_0$ and linear acceleration $\ddot{r}_0$ required to compute V_0 and $\dot{V}_0$, can be obtained from the position r_0 via implicit integration. This process is analogous to computing $\dot{\xi}$ and $\ddot{\xi}$ from ξ (as described in Section 3.1). Let j and $j+1$ represent two consecutive time steps, and suppose we want to move from step j to step $j+1$, meaning that all variables at step j are known. Given $r_{0,j+1}$, the time derivatives $\dot{r}_{0,j+1}$ and $\ddot{r}_{0,j+1}$ can be determined using an implicit integration:

$$r_{0,j+1} = a \ddot{r}_{0,j+1} + p_j, \quad \dot{r}_{0,j+1} = b \ddot{r}_{0,j+1} + v_j \quad (4.4)$$

Here, p_j and v_j include known variables at step j computed using the Newmark scheme and correspond to f_j and h_j in (3.3), respectively, with ξ replaced by r . Recall that the coefficients a and b depend on the time step Δt and are also computed using (3.3). Next, the rotation matrix R_0 required for V_0 and $\dot{V}_0$ is obtained from a rotation vector $\Theta_0 \in \mathbb{R}^3$ as follows:

$$R_{0,j+1} = R_{0,j} \exp(\hat{\Theta}_{0,j+1}) \quad (4.5)$$

where $\exp$ denotes the exponential map of $SO(3)$. Consequently, at time step $j+1$, g_0 is given by:

$$g_{0,j+1} = \begin{pmatrix} R_{0,j+1} & r_{0,j+1} \\ 0_{1 \times 3} & 1 \end{pmatrix} = \begin{pmatrix} R_{0,j} \exp(\hat{\Theta}_{0,j+1}) & r_{0,j+1} \\ 0_{1 \times 3} & 1 \end{pmatrix} \quad (4.6)$$

Similarly to $\dot{r}_0$ and $\ddot{r}_0$, the angular velocity Ω_0 and acceleration $\dot{\Omega}_0$, required to calculate η_0 and $\dot{\eta}_0$, are computed from Θ_0 using implicit time integration:

$$\Theta_{0,j+1} = a \dot{\Omega}_{0,j+1} + k_j, \quad \Omega_{0,j+1} = b \dot{\Omega}_{0,j+1} + l_j \quad (4.7)$$

where k_j and l_j include known variable at step j and are computed using the $SO(3)$ extension of the Newmark scheme:

$$\begin{aligned} k_j &= \Delta t \Omega_{0,j} + \Delta t^2 \left(\frac{1}{2} - \beta \right) \dot{\Omega}_{0,j} \\ l_j &= \Omega_{0,j} + \Delta t (1 - \gamma) \dot{\Omega}_{0,j} \end{aligned} \quad (4.8)$$

β and γ are set to $\frac{1}{4}$ and $\frac{1}{2}$, respectively, to avoid damping (Section 3.1).

In this section, we derived the computation of g_0 , η_0 , and $\dot{\eta}_0$, enabling the integration of their respective differential equations. However, their calculation depends on Θ_0 and r_0 , which remain unknown. In the next section, we will compute these values using a collocation method.

4.2 Collocation Method

In the last chapter, we proposed a collocation method to solve the clamped-free boundary value problem (2.17) (2.18). In this section we extend this method for free-free boundary value problems (2.17) (2.19) which are tailored for catheter navigation. Firstly, the same configuration space reduction detailed in Section 3.3 is used. Secondly, the collocation method presented in Section 3.4 remains largely similar in the free-free problem, with minor differences detailed in this section. As seen in the last section, the underdetermination of the free-free boundary value problem is addressed by computing g_0 , η_0 , and $\dot{\eta}_0$ with implicit time discretization that uses the rotation vector Θ_0 and the position r_0 . As a result, new unknowns arise which are Θ_0 and r_0 . However, in the free-free problem, one of the unknowns from the clamped-free problem, $\Lambda(\tau_1)$, either becomes directly known or can be computed from Θ_0 and r_0 . Recall that $\tau_1 = 0$ corresponds to the collocation point at the proximal end. Specifically, the proximal boundary condition $\Lambda(0) = -F_-$ allows us to determine $\Lambda(\tau_1)$ as $\Lambda(\tau_1) = \Lambda(0)$. Consequently, if F_- is initially expressed in the mobile frame, then:

$$\Lambda(\tau_1) = -F_- \quad (4.9)$$

Otherwise, if F_- is originally expressed in the inertial frame with f_- , then:

$$\Lambda(\tau_1) = - \begin{pmatrix} R_0^T & 0 \\ 0 & R_0^T \end{pmatrix} f_- \quad (4.10)$$

where $g_0 = (R_0, r_0)$ is computed from Θ_0 and r_0 (4.6). Let us define

$$\nu_0 = (\Theta_0^T, r_0^T)^T \quad (4.11)$$

In Section (3.4), the clamped-free problem was solved by calculating $\Lambda(X)$ through iterative computation of stress collocation values Λ_m (3.14). Similarly, the objective of this section is to solve the free-free problem by computing $\Lambda(X)$. The difference here is that the unknowns are no longer Λ_m but rather ν_0 and $\Lambda(\tau_j)$ with $j = \{2 \dots m\}$, where m is the total number of collocation points. Let us define a matrix that contains all unknowns as:

$$\Sigma = (\nu_0, \Lambda(\tau_2), \Lambda(\tau_3), \dots, \Lambda(\tau_m)) \quad (4.12)$$

One can compute $\Sigma \in \mathbb{R}^{6 \times m}$ iteratively using the system of nonlinear algebraic equations of the collocation method $\mathcal{R}_c$ (3.20) with its input changed from Λ_m to Σ . Now, one needs to compute the Jacobian matrix of the free-free problem, defined as $J_f = \frac{\partial \text{vec}(\mathcal{R}_c)}{\partial \text{vec}(\Sigma)}$, where the vec operator transform a matrix into a column vector by ordering its columns sequentially. Since the only difference between Λ_m and Σ is the replacement of $\Lambda(\tau_1)$ by ν_0 , then J_f and the Jacobian matrix of clamped-free problem J_c differ only in their first six columns. To compute these columns, let $\bullet^\dagger = \frac{\partial \bullet}{\partial \nu_{0,k}}$, with k being the index of the components of $\nu_0 = (\nu_{0,1}, \nu_{0,2}, \nu_{0,3}, \nu_{0,4}, \nu_{0,5}, \nu_{0,6})^T$. By substituting the differentiation with respect to $\Lambda_k(\tau_i)$ ($\bullet^\dagger$) with $\bullet^\ddagger$, the equations presented in Section 3.4 for calculating $\mathcal{R}_c^\dagger$ also apply to $\mathcal{R}_c^\ddagger$, with two exceptions. The first, is the calculation of $\Lambda^\ddagger(\tau_i)$. If F_- is directly expressed in the mobile frame then

$$\Lambda^\dagger(\tau_i) = 0_{6 \times 1}, i \in \{1, 2, \dots, m\} \quad (4.13)$$

Otherwise, if F_- is expressed from its inertial frame counterpart f_- , then

$$\Lambda(\tau_1)^\dagger = - \begin{pmatrix} R_0^{\dagger T} & 0 \\ 0 & R_0^{\dagger T} \end{pmatrix} f_- \quad (4.14)$$

and

$$\Lambda(\tau_i)^\dagger = 0_{6 \times 1}, i \in \{2, 3, \dots, m\} \quad (4.15)$$

We assume that the imposed tip wrenches f_- and F_- are constant within each time step, and thus independent of ν_0 . We define the twist $\varkappa = (g^{-1}g^\dagger)^\vee$, which is the counterpart of $\varphi = (g^{-1}g^\dagger)^\vee$, obtained by replacing $g^\dagger$ with $g^\ddagger$. $R_0^\dagger$ is computed from $g_0^\dagger = (R_0^\dagger, r_0^\dagger)$, where $g_0^\dagger = g_0 \widehat{\varkappa}(0)$. Similarly to (3.23), using the identity $(g')^\dagger = (g^\dagger)'$, we obtain the differential equation of $\varkappa$ with respect to arc length:

$$\varkappa' = -ad_\xi \varkappa + \xi^\dagger \quad (4.16)$$

The initial conditions $\varkappa(0)$, $\eta_0^\dagger$, and $\dot{\eta}_0^\dagger$ needed for integrating $\varkappa'$, $(\eta^\dagger)'$, and $(\dot{\eta}^\dagger)'$ (4.16) (3.29) are computed using the k -th column of P , $\frac{\partial \eta_0}{\partial \nu_0}$, and $\frac{\partial \dot{\eta}_0}{\partial \nu_0}$, respectively, as given by [66]:

$$P = \begin{pmatrix} T(\Theta_0) & 0_{3 \times 3} \\ 0_{3 \times 3} & R_0^T \end{pmatrix} \quad (4.17)$$

$$\frac{\partial \eta_0}{\partial \nu_0} = \begin{pmatrix} \frac{b}{a} 1_{3 \times 3} & 0_{3 \times 3} \\ \widehat{V}_0 T(\Theta_0) & \frac{b}{a} R_0^T \end{pmatrix} \quad (4.18)$$

$$\frac{\partial \dot{\eta}_0}{\partial \nu_0} = \begin{pmatrix} \frac{1}{a} 1_{3 \times 3} & 0_{3 \times 3} \\ ((\dot{V}_0 + \Omega_0 \times V_0)^\wedge - \widehat{\Omega}_0 \widehat{V}_0) T(\Theta_0) + \frac{b}{a} \widehat{V}_0 & (\frac{1}{a} 1_{3 \times 3} - \frac{b}{a} \widehat{\Omega}_0) R_0^T \end{pmatrix} \quad (4.19)$$

Here, $T(\Theta)$ is the derivative of the exponential map of $SO(3)$.

Now, all the necessary components for solving free-free problems are in place. In the following sections, we will see how the tubular surfaces are modeled, how contact is handled and incorporate it into the Cosserat rod boundary value problem.

4.3 Implicit Surface Modeling

The objective of this section is to present a method for surface modeling that enables fast yet accurate contact detection. Contact detection involves determining, for a given point in space, whether it is inside or outside the surface, computing its normal direction and its signed distance with respect to the nearest point on the surface. Contact detection is known to be computationally expensive in physics based simulations.

It was showed in Kerrien et al. [104] that a blood vessel surface can be accurately represented using implicit functions constructed from a point-set skeleton. Inspired by their

work, we propose using implicit functions to model our surfaces. The implicit function should satisfy several constraints:

- To have different signs for points inside and outside the surface.
- To be continuous to prevent abrupt changes in distance calculation.
- To be differentiable to allow for normal direction calculation anywhere in space.
- To be computationally efficient to evaluate.

These constraints can be met using the blobby model [104, 105, 106], which represents the surface as the zero-level set of a function $\zeta : \mathbb{R}^3 \rightarrow \mathbb{R}$, defined as a weighted sum of radial scalar fields centered at specific points $\sigma_j \in \mathbb{R}^3$. These points correspond to a point set representing the desired surface, for example, they may form the point-set skeleton. Hence, ζ is expressed as:

$$\zeta(r) = S - \sum_{j=1}^{m_b} \alpha_j \Psi^*(\vartheta_j \tilde{d}(r, \sigma_j)) \quad (4.20)$$

where m_b is the number of points, and S is the iso-surface threshold. The function $\tilde{d} : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$ defines the distance between r and a skeleton point σ_j , thereby shaping the scalar field. The function Ψ^* is a radial profile that must decay rapidly and monotonically to zero as $\tilde{d}$ increases. This ensures locality: distant points σ_j have minimal influence on one another. Additionally, α_j and ϑ_j serve as scaling parameters. Furthermore, ζ is negative when r is inside the surface and positive when it is outside. Following [104], we define $\tilde{d}$ as the Euclidean distance. Consequently, a single scalar field represents an implicit sphere, or "blob". Hence, ζ becomes:

$$\zeta(r) = S - \sum_{j=1}^{m_b} \alpha_j \Psi^*\left(\frac{|r - \sigma_j|}{\varrho_j}\right) \quad (4.21)$$

where $\vartheta = 1/\varrho$, with ϱ representing the width of the blob, related to its radius. We use the Cauchy profile:

$$\Psi^*(x) = \frac{1}{\left(\frac{x^2}{5} + 1\right)^2} \quad (4.22)$$

Normalized such that $\frac{d^2 \Psi^*(1)}{dx^2} = 0$. Notice that $\Psi^*(0) = 1$, and $\lim_{x \rightarrow \infty} \Psi^*(x) = 0$ which ensures locality (see Fig. 4.1). Differentiating ζ with respect to r yields

$$\nabla \zeta(r) = - \sum_{j=1}^{m_b} \frac{\alpha_j}{\varrho_j} \psi^*\left(\frac{|r - \sigma_j|}{\varrho_j}\right) \frac{r - \sigma_j}{|r - \sigma_j|} \quad (4.23)$$

with

$$\psi^*(x) = \frac{d\Psi^*(x)}{dx} = -\frac{100x}{(x^2 + 5)^3} \quad (4.24)$$

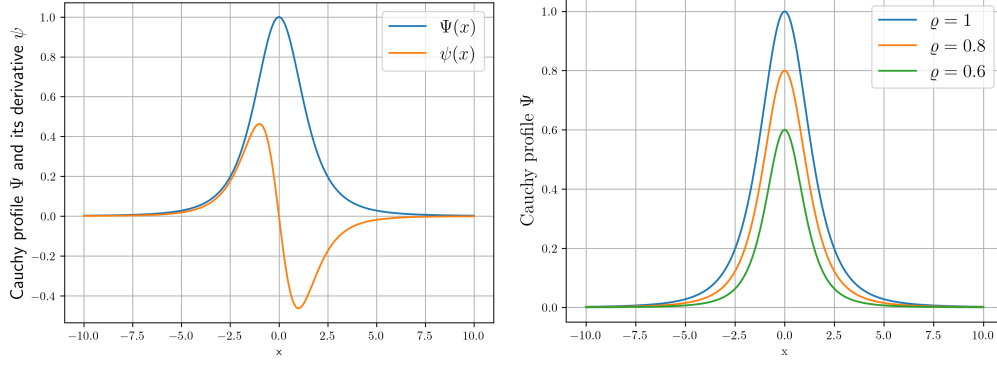


Figure 4.1: Left: the Cauchy profile and its derivative. Right: weighted Cauchy profile $\varrho \Psi\left(\frac{x}{\varrho}\right)$ shown with three different widths ϱ .

The current formulation of $\nabla\zeta$ introduces a singularity when $r = \sigma_j$. To circumvent this, one can square Ψ^* input, introducing $\Psi(x^2) = \Psi^*(x)$:

$$\zeta(r) = S - \sum_{j=1}^{m_b} \alpha_j \Psi\left(\frac{|r - \sigma_j|^2}{\varrho_j^2}\right) \quad (4.25)$$

Denoting $\psi = \frac{d\Psi}{dx}$, this revised formulation results in

$$\nabla\zeta(r) = -2 \sum_{j=1}^{m_b} \frac{\alpha_j}{\varrho_j^2} \psi\left(\frac{|r - \sigma_j|^2}{\varrho_j^2}\right) (r - \sigma_j) \quad (4.26)$$

which effectively eliminates the singularity and improves computational efficiency by avoiding square root calculations. Furthermore, the radial profile Ψ^* is adjusted to take into account the squaring of its input

$$\Psi(x) = \frac{1}{\left(\frac{x}{5} + 1\right)^2} \quad (4.27)$$

Consequently, its derivative becomes

$$\psi(x) = -\frac{50}{(x + 5)^3}. \quad (4.28)$$

As in [104], we define $\alpha = \varrho$ so that $\zeta(r)$ better relates to the distance from the surface. Additionally, this parameterization avoids redundancy, improving the convergence of optimization techniques used for surface modeling [107]. The final version of ζ and its gradient $\nabla\zeta$ are expressed as follows

$$\zeta(r) = S - \sum_{j=1}^{m_b} \varrho_j \Psi\left(\frac{|r - \sigma_j|^2}{\varrho_j^2}\right) \quad (4.29)$$

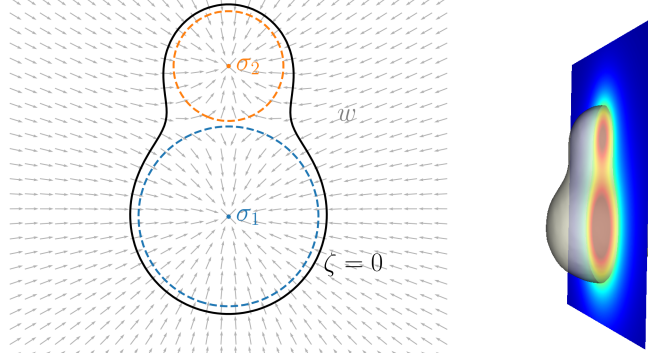


Figure 4.2: Left: visualization of a surface (black) defined by the iso-level $\zeta = 0$, generated from two blobs centered at $\sigma_1 = (0, 0, 0)^T$ and $\sigma_2 = (0, 5.5, 0)^T$, with respective radii $\varrho_1 = 1$ and $\varrho_2 = 0.7$, and using a threshold $S = 0.08$. The field normals w are shown at discretized locations. Right: 3D view of the scalar field $\zeta(r)$, with a planar cross-section at $z = 0$ visualized in false colors. The field is normalized, with red indicating the most negative values. The surface (in grey) again corresponds to the iso-level $\zeta = 0$, enclosing the region influenced by the two blobs.

$$\nabla \zeta(r) = - \sum_{j=1}^{m_b} 2 \psi \left(\frac{|r - \sigma_j|^2}{\varrho_j^2} \right) \frac{r - \sigma_j}{\varrho_j} \quad (4.30)$$

Fig. 4.1 shows the Cauchy profile with its derivative. Fig. 4.2 shows a surface created using two blobs centered at $\sigma_1 = (0, 0, 0)^T$ and $\sigma_2 = (0, 5.5, 0)^T$, with radii $\varrho_1 = 1$ and $\varrho_2 = 0.7$, respectively, and a threshold $S = 0.08$.

We have presented the implicit modeling approach used to represent our tubular surfaces. This theoretical foundation enables its application to contact handling, which will be discussed in the next section.

4.4 Contact Handling

In this section, we describe contact handling in its continuous form, which includes both contact detection and contact response. Contact response refers to the computation of contact forces necessary to reduce interpenetration. In this work, we make the following assumptions. First, contact handling is performed between the surface and the cross-sectional centerline of the rod, r . This assumption is valid for slender micro-tools used in neuroradiology procedures due to their small radii. Furthermore, the surface is considered rigid and fixed. Finally, we do not account for the effects of blood flow.

4.4.1 Contact Detection

Using the implicit function ζ (4.29), one can compute the normal direction $w(r) \in \mathbb{R}^3$ at any point $r \in \mathbb{R}^3$ using the gradient.

$$w(r) = -\frac{\nabla\zeta(r)}{|\nabla\zeta(r)|} \quad (4.31)$$

The negative sign makes the normal points toward to surface when r is outside of it (see Fig. 4.2). To estimate the signed distance from a point r to the surface $\zeta = 0$, we proceed as follows: starting from the point r , we move along the direction of the surface normal until we reach a location where the implicit function evaluates to zero. Although the exact normal direction may change along this path, we approximate it by keeping the initial normal $w(r)$ fixed. The goal is to find how far we must move along this direction for the function value to vanish. The length of that step gives an estimate of the distance to the surface. This leads to the following expression for the signed distance:

$$\phi(r) = \text{sign}(\zeta) |r + ow - r| = \text{sign}(\zeta)|o| = o \quad (4.32)$$

Here, $o \in \mathbb{R}$, and the function $\text{sign}(\zeta)$ returns 1 if ζ is positive and -1 if it is negative. In this work, we suppose that r is in contact if it is outside the surface ($\zeta > 0$). To compute the offset o , we define a residual function $\mathcal{R}_\phi : \mathbb{R} \rightarrow \mathbb{R}$ that evaluates the field at the displaced point. The objective is to find the value of o such that this residual becomes zero, meaning that the displaced point lies exactly on the surface.

$$\mathcal{R}_\phi(o) = \zeta(r + ow) \quad (4.33)$$

Since w is treated as constant during this process, the result is an approximation that remains valid when r is sufficiently close to the surface. Finally, the derivative $\frac{d\mathcal{R}_\phi}{do}$ needed for the iterative computation of o is calculated in closed form as follows:

$$\frac{d\mathcal{R}_\phi}{do} = \frac{d\zeta(r + ow)}{do} = -\sum_{j=1}^{m_b} \frac{2}{\varrho_j} \psi\left(\frac{|r + ow - \sigma_j|^2}{\varrho_j^2}\right) [(r + ow - \sigma_j) \cdot w] \quad (4.34)$$

Now that the essential elements of contact detection have been discussed, the next subsection will focus on their application in computing the contact force response.

4.4.2 Contact Response

We compute the normal contact force $n_\perp(X) \in \mathbb{R}^3$ using a penalty method as follows

$$n_\perp = \lambda \max(0, \phi) w \quad (4.35)$$

where $\lambda \in \mathbb{R}$ is a chosen penalty stiffness. No moment is created from $n_\perp$ since the contact force is applied to the centerline r . Furthermore, we adopt the Coulomb friction model, where the tangential frictional force $n_\parallel(X) \in \mathbb{R}^3$ is proportional to the normal force

$$|n_\parallel| \leq \mu |n_\perp| \quad (4.36)$$

with $\mu \in \mathbb{R}$ being the friction coefficient. The direction and magnitude of the friction force depend on whether the friction is sliding or sticking, based on the relative tangential velocity $\dot{r}_\parallel$. For sliding friction, the force magnitude reaches its maximum ($|n_\parallel| = \mu |n_\perp|$) and

opposes $\dot{r}_{\parallel}$. In the case of sticking friction, the force can take any value within the friction cone (4.36) [108]. For simplicity, in this work, we ignore the sticking mode and only account for the sliding one.

$$n_{\parallel} = \begin{cases} -\mu |n_{\perp}| \frac{\dot{r}_{\parallel}}{|\dot{r}_{\parallel}|} & \text{if } |\dot{r}_{\parallel}| \neq 0, \\ 0_{3 \times 1} & \text{otherwise.} \end{cases} \quad (4.37)$$

Here, $\dot{r}_{\parallel}$ is simply computed as

$$\dot{r}_{\parallel} = \dot{r} - (\dot{r} \cdot w)w \quad (4.38)$$

Summing the normal and tangential contact forces yields the total contact force, n_c :

$$n_c = n_{\perp} + n_{\parallel} \quad (4.39)$$

Finally, expressing the contact response as a wrench in the inertial frame with no moment

$$f_c = \begin{pmatrix} 0_{3 \times 1} \\ n_c \end{pmatrix} \quad (4.40)$$

With a slight abuse of terminology, we will refer to f_c as the contact force rather than the contact wrench, since the contact moment is zero. In this section, we presented a contact handling approach that enables the computation of the frictional contact force f_c . In the next section, we detail how f_c can be incorporated into the boundary value problem of Cosserat rods.

4.5 Including Contact Forces into the Boundary Value Problem

The objective here is to incorporate the contact force $f_c(X)$ into the free-free boundary value problem (2.17) and (2.19). First, note that the contact force f_c is a concentrated force (measured in Newtons) acting at a specific point on the rod. Consequently, if it acts on the rod proximal ($X = 0$) or distal ($X = l$) ends, it is directly assigned to the boundary conditions (2.19) after a frame transformation.

$$\begin{aligned} \Lambda(0) &= -F_- = - \begin{pmatrix} R(0)^T & 0 \\ 0 & R(0)^T \end{pmatrix} f_- = - \begin{pmatrix} R(0)^T & 0 \\ 0 & R(0)^T \end{pmatrix} (f_c(0) + f_e(0)) \\ \Lambda(l) &= F_+ = \begin{pmatrix} R(l)^T & 0 \\ 0 & R(l)^T \end{pmatrix} f_+ = \begin{pmatrix} R(l)^T & 0 \\ 0 & R(l)^T \end{pmatrix} (f_c(l) + f_e(l)) \end{aligned} \quad (4.41)$$

Here, $f_e \in \mathbb{R}^6$ represents other external concentrated wrenches. In the context of simulating catheter navigation, $f_e(0) = (c_e(0)^T, n_e(0)^T)^T$ or its mobile frame counterpart $F_e(0) = (C_e(0)^T, N_e(0)^T)^T$ is used to displace and rotate the rod proximal end, mimicking practitioner hand manipulation. Furthermore, if the contact force f_c acts at points other than

the boundaries, it is incorporated into the continuous balance equation (2.8) as a distributed contact force $\bar{f}_c$ (measured in Newtons per meter) after a frame transformation.

$$\bar{F} = \begin{pmatrix} R^T & 0 \\ 0 & R^T \end{pmatrix} \bar{f} = \begin{pmatrix} R^T & 0 \\ 0 & R^T \end{pmatrix} (\bar{f}_c + \bar{f}_e) \quad (4.42)$$

Here, $\bar{f}_e$ represents other distributed wrenches, such as the drag induced by blood flow along the catheter surface and the effect of gravity acting on the device.

Having established how contact forces are incorporated into the boundary value problem, the next section focuses on providing a time and space discretization of such contacts, along with an approach for computing $\bar{f}_c$.

4.6 Discretization of Contact Forces

4.6.1 Concentrated to Distributed Force Conversion

Now, we consider the computation of the distributed contact force $\bar{f}_c$ from the concentrated contact force f_c . According to Antman [34], distributed forces are defined as forces per unit reference (unstressed) length. Now, let us consider a finite segment of the rod, $[s_0, s_1] \subset (0, l)$. If the distributed contact force along this segment, $\bar{f}_c : [s_0, s_1] \rightarrow \mathbb{R}^6$, is assumed to be constant, then it can be computed by averaging the concentrated contact forces f_c acting on the segment over its reference length:

$$\bar{f}_c(s) = \frac{1}{s_1 - s_0} \sum_{j=1}^{m_s} f_c(s_j) \quad (4.43)$$

where $f_c(s_j)$, $\{s_j\} \subset [s_0, s_1]$ represent m_s concentrated contact forces acting on the rod segment $[s_0, s_1]$.

4.6.2 Space Discretization

In this work we discretize the rod into equidistant segments and assume that each segment has a single concentrated force applied at its center. Additionally, both the base and the tip of the rod experience one concentrated force each. Translating this into equations, we discretize the rod into $m_c - 1$ equidistant segments using m_c nodes, denoted as X_j , as follows:

$$X_j = \frac{j-1}{m_c-1}l, \quad j = \{1, \dots, m_c\} \quad (4.44)$$

Here, $X_1 = \tau_1 = 0$ and $X_{m_c} = \tau_{m_c} = l$. First, the contact forces at the first and last nodes, $f_c(X_1)$ and $f_c(X_{m_c})$, are used to compute the boundary conditions in (4.41). Second, $\bar{f}_c$ is computed from f_c as follows:

$$\bar{f}_c(X) = \begin{cases} 0, & X = 0 \text{ or } X = l, \\ \frac{1}{\Delta X} f_c(X_j + \frac{\Delta X}{2}), & \begin{cases} X \in (X_j, X_{j+1}), & \text{if } j = 1, \\ X \in [X_j, X_{j+1}), & \text{if } j = 2, \dots, m_c - 1. \end{cases} \end{cases} \quad (4.45)$$

where ΔX is the reference length of the segments. Furthermore, $\bar{f}_c(X_1)$ and $\bar{f}_c(X_{m_c})$ are set to zero, as the concentrated forces take over at the boundaries. Finally, to solve the boundary value problem using the collocation method with the system (3.20), one needs the concentrated forces at the collocation points, $\bar{f}_c(\tau_i)$. These values are directly retrieved from (4.45).

With the spatial discretization of contact forces defined, the next subsection addresses their time discretization.

4.6.3 Time Discretization

Recall that time discretization is necessary to solve the boundary value problem (Section 3.1), consequently, continuous-time contact handling cannot be performed. Furthermore, to accurately handle contact in the context of a penalty method, the rod constrained dynamics are computed in two stages. The constrained state refers to the condition where the rod is in collision with the surface and subjected to contact forces, while the unconstrained state is when no contact forces are applied, even if the rod is in collision with the surface.

Consider two consecutive time steps, t and $t + dt$, and suppose that the constrained rod dynamics for step t have already been computed, and we now proceed to step $t + dt$. First, the boundary value problem is solved at $t + dt$ as if no contact forces were present, i.e., $\bar{f}_c(X) = 0$, $f_c(0) = 0$, and $f_c(l) = 0$ (unconstrained state). Then, the contact forces are computed based on this state. Finally, the boundary value problem is re-solved for step $t + dt$, incorporating the computed contact forces (constrained state). Reflecting on this approach, the unconstrained solution allows us to obtain the rod position r after one time step, considering only the restoring, inertial, and external forces acting on it. This then permits the computation of the correct signed distance ϕ and normal direction w needed to reduce potential interpenetration with the surface. Furthermore, at each time step, the initial guesses for both the constrained and unconstrained boundary value problems are set to the solutions from the previous step for their respective problems. At first, this may seem counterintuitive, as one might think that the unconstrained solution at time t would provide a better initial guess for the constrained one at t . However, empirically, we found that this is not the case and that the proposed approach yields better initial guesses, thereby reducing the number of system evaluations (3.20), which in turn decreases computation time. Finally, the simulation loop of the proposed method is summarized in Algorithm 2.

Algorithm 2 High-Level Simulation Loop

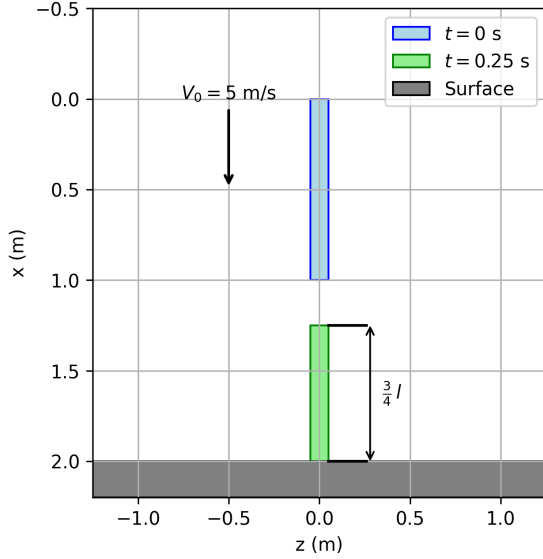
```
1: Initialize to a static solution
2: Initialize proximal boundary condition
3: for each simulation step do
4:   Time integrate the time derivatives of the space-rate twists  $\dot{\xi}$  and  $\ddot{\xi}$  (3.2)
5:   Set contact force  $f_c = 0$ 
6:   Solve the unconstrained boundary value problem (Section 4.2)
7:   Compute contact force  $f_c$  (4.40)
8:   Update boundary conditions (4.41)
9:   Update distributed forces (4.42)
10:  Solve the constrained boundary value problem (Section 4.2)
11: end for
```

4.7 Numerical Applications

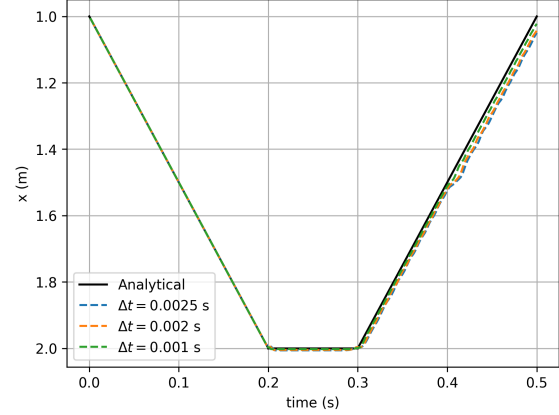
The objectives of the numerical tests are as follows. First, to verify the accuracy of free-free boundary value problem solving with respect to an analytical solution. Second, to evaluate frictional contact using a synthetic surface created with a blobby model. Lastly, to test our method in a real-world setup, including a model of a carotid artery and actual catheter material properties. The implementation was coded in Python. We used SciPy’s Runge-Kutta 4(5) for arc length integration. For least square minimization of the nonlinear algebraic system of collocation $\mathcal{R}_c$ (3.20), we used SciPy’s `hybr` function. The Levenberg-Marquardt algorithm from Scipy is used for the iterative distance computation (4.33). Numerical integrations and root-finding techniques were performed with SciPy’s built-in default tolerances. Execution times are reported, with the time for solving the boundary value problem including both unconstrained and constrained solutions. Contact detection times are provided, but not those for contact response, as the latter is negligible due to its closed form expression. The simulations were performed on a computer equipped with an Intel Xeon W 2245 CPU running at 3.90 GHz.

4.7.1 Elastic Longitudinal Contact

The objective of this test is to verify the proposed method in a free-free boundary value problem against an analytical solution. Replicating [71], a soft rod moving at initial velocity V_0 comes into elastic longitudinal contact with a plane, with its tip initiating the collision (Fig. 4.3a). Upon impact, the rod compresses along its length, storing its kinetic energy as elastic energy. It then regains its original shape and rebounds, moving away at a velocity of $-V_0$. The rod length is $l = 1$ m, Young’s modulus $E = 4 \times 10^5$ Pa, cross-sectional area $A = 0.1$ m², and density $\rho = 1000$ kg/m³. The initial velocity is $V_0 = 5$ m/s. The plane is positioned 1 m away from the tip of the rod. From the initial velocity, we determine that the first contact occurs at $t = 0.2$ s. According to the analytical solution [109], the total duration of contact is 0.1 s, with the first half corresponding to compression and the second half to decompression. Additionally, the maximum compression reaches $\frac{l}{4}$. The magnitude of the tip force is [71]:



(a) Scenario configuration.



(b) Tip position over time.

Figure 4.3: A soft rod moving at initial velocity V_0 comes into elastic longitudinal contact with a plane.

$$|f_+| = \frac{EA V_0}{\sqrt{E/\rho}} = 10000 \text{ N} \quad (4.46)$$

We used Chebyshev polynomial degree $m = 300$, maximum allowed arc length integration step $\max_{\Delta x} = \frac{l}{2m}$ m, and tried three time steps $\Delta t = \{0.0025, 0.002, 0.001\}$ s to verify convergence. Fig. 4.3b shows the evolution of the tip position over time. It is in good agreement with the theoretical solution, and the difference reduces as the time step decreases. Fig. 4.4 shows the evolution of tip velocity over tip position. As in [71], a wave with a small amplitude appears after the rod rebounds from the plane. We observe that its frequency increases and its amplitude decreases as the time step decreases, which implies a numerical effect caused by time integration. The simulations took 5.76 hours, 7.25 hours and 13.4 hours for $\Delta t = \{0.0025, 0.002, 0.001\}$ s scenarios, respectively. Consequently, the average per time step is 1.69 minutes.

4.7.2 Frictional Contact with a Helical Tube

This test aims to evaluate the proposed method in a frictional contact scenario using a synthetic surface created with a blobby model. Inspired by [87], a rubber rod is inserted into a helical tube using a constant follower force (i.e., expressed in the mobile frame) applied at the proximal end (Fig. 4.5). The surface is modeled with a blobby model consisting of $m_b = 45$ blobs, with width $\varrho_i = 2 \times 10^{-3}$, threshold $S = 2 \times 10^{-5}$, and blob centers σ_i representing the discrete centerline of the helical tube. The tube radius is 0.016 m, and the helix radius is 0.03 m. The rod has a length of $l = 0.35$ m, a circular cross-section with a radius of $R_b = 5 \times 10^{-3}$ m, a Young's modulus of $E = 5 \times 10^6$ Pa, a shear modulus

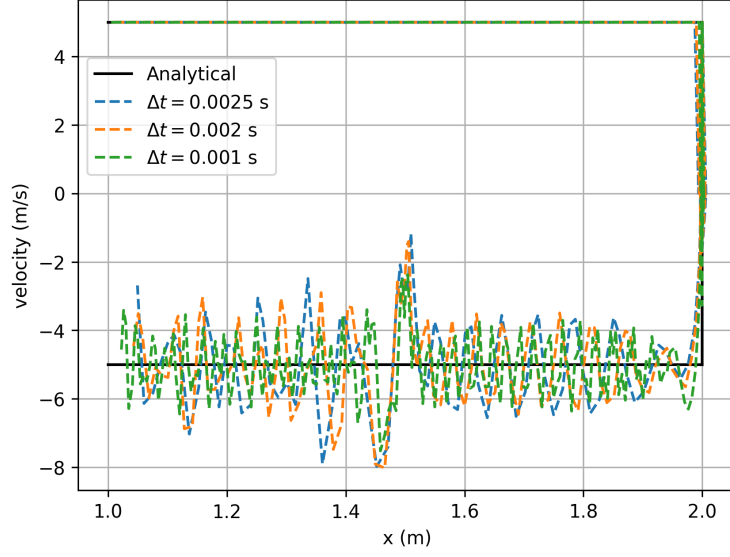


Figure 4.4: A soft rod moving at an initial velocity of $5 \text{ m} \cdot \text{s}^{-1}$ comes into elastic longitudinal contact with a plane and rebounds with a velocity of $-5 \text{ m} \cdot \text{s}^{-1}$. The figure plots tip velocity over tip position.

of $G = \frac{E}{2(1+\nu)}$ Pa, a Poisson's ratio of $\nu = 0.5$, and a density of $\rho = 1100 \text{ kg/m}^3$. The rod is pushed using a constant follower force of $N_e(0) = (1, 0, 0)^T \text{ N}$. The contact handling parameters were as follows: penalty stiffness $\lambda = 6000 \text{ N/m}$ and number of contact nodes $m_c = 100$. Three coefficients of friction were tested: $\mu = \{0, 0.1, 0.2\}$. The numerical parameters were: Chebyshev polynomial degree $m = 500$ for $\mu = 0$ test, and $m = 400$ for $\mu = \{0.1, 0.2\}$ tests, maximum allowed arc length integration step $\max_{\Delta X} = \frac{l}{2m} \text{ m}$, time step $\Delta t = 0.0005 \text{ s}$, and a total simulated time of 0.2125 seconds. The total time is chosen based on the time required for $\mu = 0$ rod tip to reach the end of the tube. For the $\mu = 0$ test, the polynomial degree m was increased by 100 compared to the $\mu = \{0.1, 0.2\}$ tests. This is because the $\mu = 0$ case involves faster dynamics, therefore, a larger m is needed to stabilize the collocation method, as explained in the previous chapter.

Fig. 4.6 shows snapshots of the simulation at different times and varying friction coefficients. The rod is correctly confined within the surface. Furthermore, by increasing the coefficient of friction the rod sliding motion is significantly reduced, and by the end of the simulation, only the rod with friction coefficient $\mu = 0$ managed to reach the end of the tube, while the other two remained inside the helix. This is also illustrated in Fig. 4.7, which shows the tip x-axis position over time, corresponding to the longitudinal axis of the helical tube. As a result, we can conclude that the contact and friction effects are effectively included in our method.

The $\mu = 0$ scenario took 43.7 hours to complete, with an average of 6.16 minutes per time step. In contrast, the $\mu = \{0.1, 0.2\}$ scenarios took 29.8 hours, with an average of 4.2 minutes per time step. This difference is explained by the fact that the $\mu = 0$ case uses a polynomial degree 100 higher than the $\mu = \{0.1, 0.2\}$ cases. Furthermore, all scenarios spent a total of 37.4 seconds on contact detection.

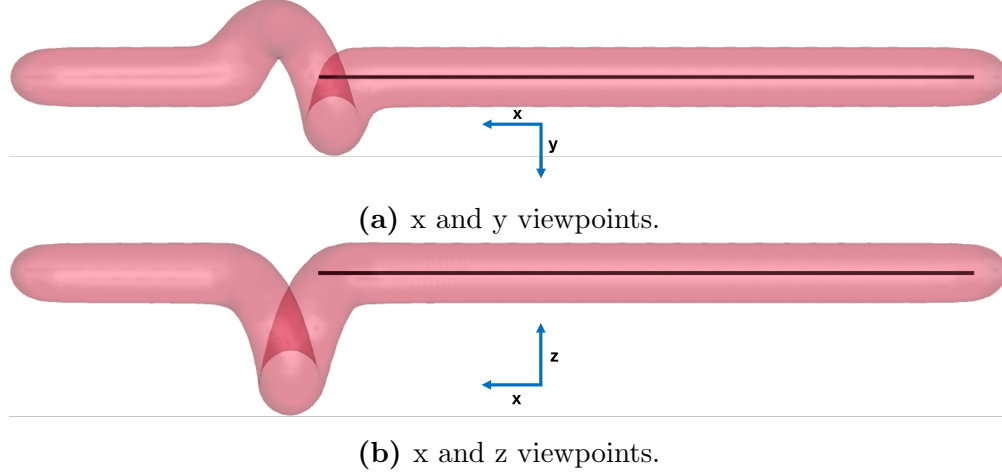


Figure 4.5: A rubber rod is inserted into a helical tube. Initial configuration at $t = 0$.

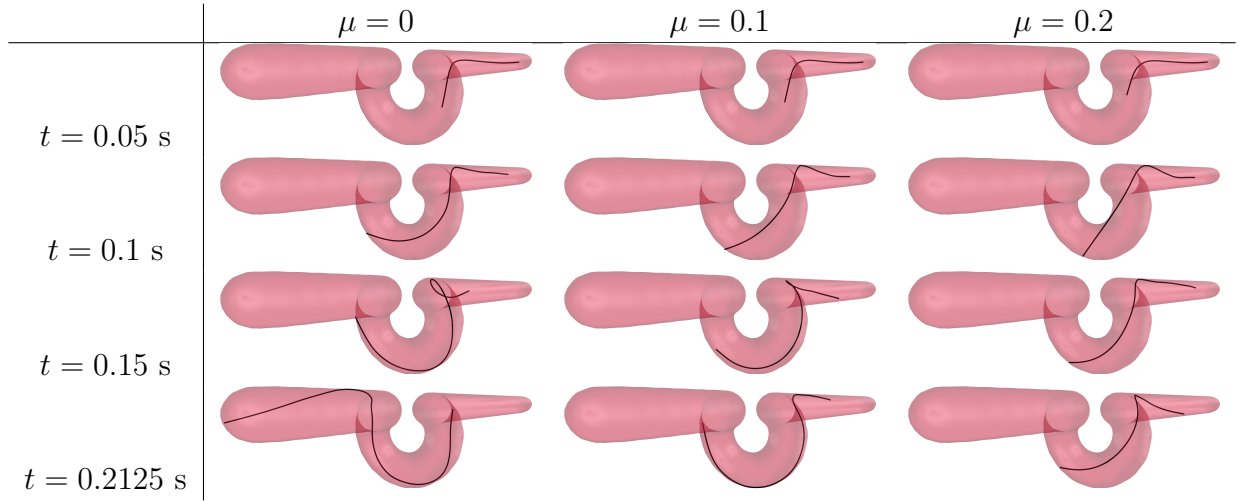


Figure 4.6: Snapshots at different times and varying friction coefficients show a rubber rod inserted into a helical tube.

4.7.3 Catheter Insertion into a Carotid Artery

The aim of this test is to evaluate our method using a real catheter navigation setup, which includes actual catheter material properties and frictional contact with a carotid artery model (Fig. 4.8). The carotid artery model is created from 3D rotational angiography data of a real patient [104]. This test is challenging because the catheter is relatively rigid, and the surface is narrow with a sharp turn, requiring considerable force to maneuver the catheter through. Hence, the catheter is inserted into the surface with constant follower force at its base of $N_e(0) = (0.01, 0, 0)^T$ N. The surface is created with a blobby model containing $m_b = 2573$ blobs. Specifically, the function ζ was constructed from a point-set skeleton representing the surface of the carotid artery model, with the blob parameters iteratively computed to minimize the distance between its zero level set and the input points. The catheter material properties are: the length $l = 0.065$ m, cross-section radius $R_b = 0.00028$ m, Young's modulus

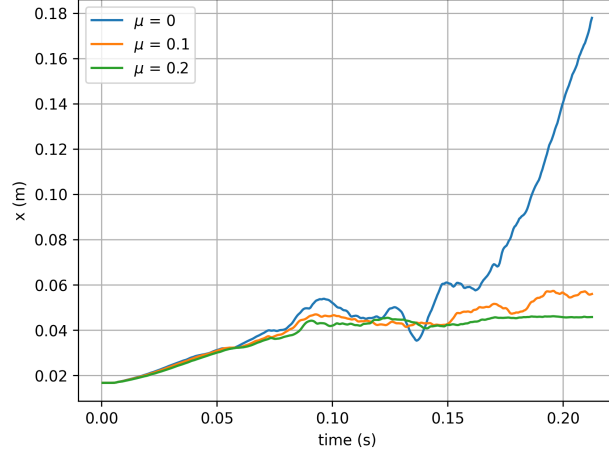


Figure 4.7: A rubber rod is inserted into a helical tube. Tip position over time for various coefficients of friction μ .

$E = 300 \times 10^6$ Pa, Poisson's ratio $\nu = 0.49$, shear modulus $G = \frac{E}{2(1+\nu)}$ Pa, and density $\rho = 1998 \text{ kg/m}^3$. Mechanical characterization was conducted to determine the catheter material properties, as this information is often not provided by the manufacturer. Contact handling parameters were: penalty stiffness $\lambda = 300 \text{ N/m}$, friction coefficient $\mu = 0.05$, and number of contact nodes $m_c = 100$. Numerical parameters were: Chebyshev polynomial degree $m = 300$, maximum allowed arc length integration step $\max_{\Delta X} = \frac{l}{m} \text{ m}$, and time step $\Delta t = 5 \times 10^{-5} \text{ s}$. For the catheter to reach the end of the surface, the total simulated time is set to 0.025 s . It is relatively low because the pushing force magnitude is large (0.01 N).

Fig. 4.8 shows six snapshots of the simulation, and Fig. 4.9 plots the tip position over time. We observe that the rod is successfully inserted and properly contained within the surface. The simulation required 12.4 hours to solve the boundary value problem, with an average of 1.49 minutes per time step. Additionally, contact detection took a total of 24.6 minutes, averaging 2.95 seconds per time step, which is relatively fast for such a complex surface.

Qualitative Comparison with the Shooting Method

Our work in [103] tested a similar simulation scenario using the shooting method. However, to address the singularity issue (described in detail in Chapter 3), damping was introduced in the constitutive equation:

$$\Lambda = \mathcal{H}(\xi - \xi_o) + \mathcal{D}\dot{\xi} \quad (4.47)$$

Recall that $\mathcal{H}$ is the stiffness matrix and $\xi - \xi_o$ represents the strain. Furthermore, $\mathcal{D}$ is a chosen damping matrix. This damping follows the Kelvin-Voigt viscous damping model [63]. Additionally, the catheter was softened by decreasing Young's modulus to $E = 5.5 \times 10^5 \text{ Pa}$, allowing the rod to pass more easily through the carotid turn. This may seem arbitrary, as the singularity is more likely to occur when the rod is soft. However, the used time step was large ($\Delta t = 0.005 \text{ s}$), and fast dynamics due to catheter stiffness and large contact forces also

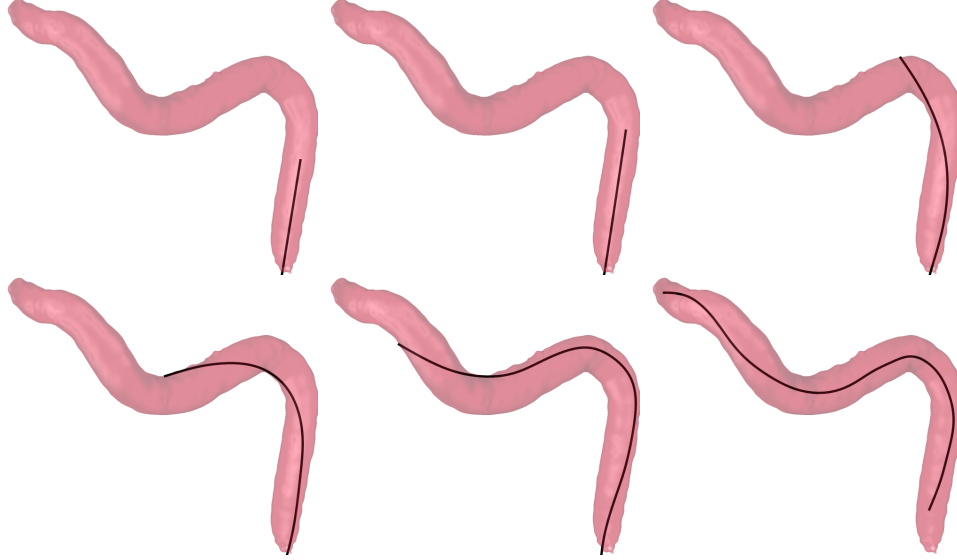


Figure 4.8: Catheter is inserted into a carotid artery model. From left to right and top to bottom, the figure shows five snapshots taken every 0.005 seconds.

destabilize the shooting method. Fig. 4.10 shows four snapshots of the simulation, where the damping effect can be observed: the catheter inaccurately retains its curvature after the turn. Moreover, the clamped-free boundary value problem was used, and the insertion was performed by displacing the fixed proximal end pose through changing of the boundary condition (2.18). Additionally, the proximal velocity and acceleration were set to zero. It is as if the catheter was fixed at one end while the surface moved toward it. Consequently, using the clamped-free problem does not produce the same catheter deformation as the free-free problem. Finally, this simulation, using the shooting method, took 38.6 seconds to complete.

4.8 Conclusion

The objective of this chapter was to propose a method tailored for simulating catheter navigation procedures. The tubular contact surfaces (such as blood vessels) were modeled as zero-level sets of implicit functions. The catheter was modeled as a Cosserat rod by solving the free-free boundary value problem in dynamic mode, with its Jacobian matrix provided analytically. The concentrated contact forces were computed using the penalty method, and we proposed an approach to incorporate them into the continuous balance equation by converting them into distributed forces. The method was first compared to an analytical solution to assess its accuracy and showed good agreement. Second, the frictional contact was verified against synthetic surfaces. Third, we tested our method in a realistic catheter navigation setup, which included a model of a carotid artery as well as actual catheter material properties. Furthermore, the method was qualitatively compared to the shooting method.

The method was successfully tested in challenging contact scenarios involving complex surface geometry and fast dynamics. This further confirms that the singularity problem explained in the previous chapter was indeed effectively addressed. Additionally, the

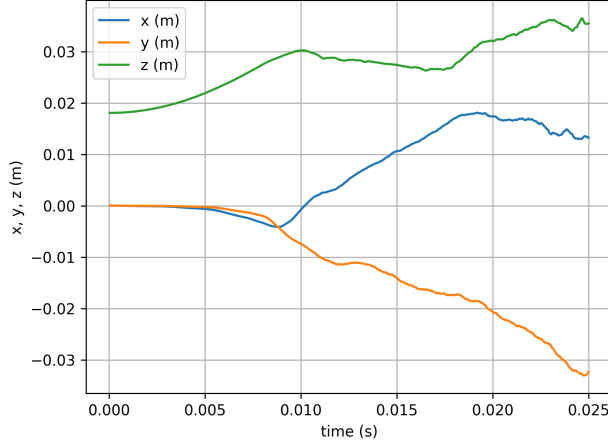


Figure 4.9: Catheter is inserted into a carotid artery model. The figure shows tip position over time.

Chebyshev polynomial degree was increased to $m = 500$ (300 more than numerical tests from Chapter 3), and the time step was set to a record low of 5×10^{-5} s (compared to $\Delta t = 2.5 \times 10^{-4}$ s in Chapter 3). Modeling the surface using implicit functions proved to be a good choice, as contact detection time was negligible compared to Cosserat rod dynamics solving. Moreover, implicit modeling is memory-efficient: the complex carotid geometry was represented using 2573 blobs, each expressed with only four floating-point values. Furthermore, compared to our solution proposed in [103] using the shooting method, the work of this chapter uses actual catheter material properties and eliminates the need for damping.

However, this work lacks validation against real data. Before proceeding with such validation, the penalty method for contact response should be replaced with a more accurate approach. Indeed, the penalty method transforms the hard constraints of contact into soft constraints by allowing small interpenetrations that generate restoring forces. In this approach, sliding friction is straightforward to include by applying a force opposite to the tangential velocity, with a magnitude proportional to the normal force. In contrast, modeling the sticking mode requires handling cases where the tangential velocity is zero and the friction force can take any value within the friction cone. This introduces an inequality condition that is not easily handled by the penalty approach and would require a complementarity-based model for accurate solution. Furthermore, the high polynomial degree was primarily used to avoid singularity issues rather than to improve accuracy. This means that reducing the degree leads to method instability, as explained in the previous chapter. Increasing the polynomial degree expands the dimensions of the Jacobian matrix, which in turn increases the number of arc length integrations. As a result, without parallelization and significant hardware improvements, using this method for fast simulations is not feasible. In the next chapter, we will explore whether ignoring inertia effects with a quasi-static mode could provide a faster alternative.

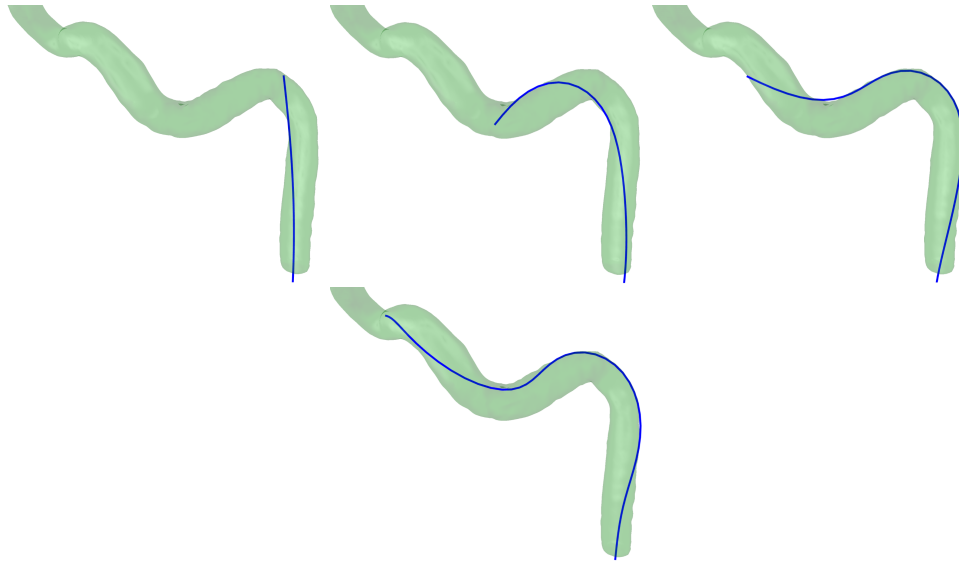


Figure 4.10: A catheter is inserted into a carotid artery model. From left to right and top to bottom, snapshots are shown at 0.25, 0.5, 0.75, and 1 s. The simulation is solved using the shooting method with damping and modified catheter material properties. The damping effect can be observed at $t = 0.5$ s, as the catheter inaccurately retains its curvature after the turn.

Chapter 5

Quasi-Static Cosserat Rods in Contact with Implicit Surfaces

In the last chapter, we used a collocation method to solve the dynamic boundary value problem of Cosserat rods undergoing contact. Although the results were accurate and realistic, the computation time was high and not suitable for fast simulations. The objective of this chapter is to explore ways to improve execution time.

Firstly, in simulations of catheter navigation, the rod is confined within narrow tubular surfaces, where its insertion velocity is relatively low and deformation is primarily induced by contact. Therefore, we can assume that inertial effects are negligible and use the static model. The evolution of the rod configuration can be considered as a sequence of static equilibria, which we refer to here as the quasi-static mode.

Secondly, as we saw in Chapter 3, the Lagrangian model combined with the strain parameterization proposed by Boyer et al. [60] achieved a similar level of accuracy to our proposed collocation method, while requiring fewer degrees of freedom and being computationally faster. Therefore, we propose to base our quasi-static approach on the Lagrangian model. The static mode can be computed either by solving for the unknown generalized coordinates in the balance equation using root-finding techniques, as proposed in [37], or by introducing artificial damping into the system as in [60]. The latter approach yields an explicit first-order differential equations in terms of coordinate velocities, which can be integrated over time to directly obtain the solution, thus removing the need for initial guesses near the convergence point compared to the root-finding approach. In [60], these equations were integrated using explicit solvers.

Contributions

In this chapter, we base our approach on that of Boyer et al. [60] to solve the quasi-static mode of Cosserat rods. Their ordinary differential equation governing the quasi-static evolution may become stiff in complex contact scenarios, requiring explicit solvers to take excessively small time steps, which degrades time efficiency. In contrast to methods based on root-finding, an implicit solver tailored for stiff equations can be used here. Thus, we use implicit solvers and provide the problem Jacobian analytically. The analytical derivation of the Jacobian is made possible thanks to the use of implicit functions that model our contact

surfaces. To the best of our knowledge, we are the first to introduce contact handling in quasi-static Lagrangian model.

Our contributions can be summarized as follows:

- Proposing a method for quasi-static solving of Cosserat rods in contact.
- Using an implicit method and providing its Jacobian matrix analytically.
- Improving computation time by up to two orders of magnitude compared to the use explicit solvers.
- Testing our method in a real catheter navigation setup that includes a model of a carotid artery and actual catheter material properties.
- Making the implementation open-source at gitlab.inria.fr/rjilani/RAL2024.

This chapter is structured as follows. Section 5.1 provides a brief overview of the Lagrangian model, including strain reduction and balance equations, as presented in [60, 37]. Section 5.2 introduces the continuous contact handling approach, while Section 5.3 details the integration of contact forces into the Lagrangian model. Section 5.4 derives the Jacobian matrix analytically for implicit solvers. Section 5.5 presents our approach for modeling catheter insertion and rotation. Finally, Section 5.6 tests the proposed method through three numerical applications, followed by concluding remarks in Section 5.7.

5.1 Lagrangian Model

In this section, we provide a brief overview of the Lagrangian model of static Cosserat rods that uses strain reduction. First, we introduce the reduction of Cosserat configuration space. Next, we present the Lagrangian balance equation and describe how the involved matrices are computed in their continuous form. Finally, we explain how this balance equation is solved. For more details, we refer the reader to [60, 37].

5.1.1 Configuration Space Reduction

Here, we describe the strain parameterization reduction. First, let us recall the Cosserat kinematics:

$$g' = g\hat{\xi}. \quad (\text{recall 2.3})$$

g is the cross-section pose, ξ is the space-rate twist, and $\bullet'$ denotes differentiation with respect to the arc length X . Notice that if $\xi(X)$ is known, one can integrate the kinematics (2.3) with the initial condition $g(0) = 1_{4 \times 4}$ to obtain the rod configuration. As a result, $(g(0), \xi)$ completely defines the rod configuration. Therefore, one can define the Cosserat configuration space as

$$\mathcal{C} = SE(3) \times \{\xi : X \in [0, l] \rightarrow \xi(X) \in \mathbb{R}^6\}. \quad (5.1)$$

Next, recall that the strain is given by ϵ :

$$\epsilon = \xi - \xi_o. \quad (\text{recall } 2.5)$$

Here, ξ_o denotes the reference configuration of ξ . Let $q \in \mathbb{R}^k$ represent the generalized strain coordinates. The parameter k is given by $k = \sum_{j=1}^6 k_j$, where k_j denotes the number of shape functions for the six possible deformations of Cosserat rods: two bending modes, torsion, two shear modes, and extension. Therefore, the strain can be reduced to

$$\epsilon = \Phi(X)q, \quad (5.2)$$

where $\Phi \in \mathbb{R}^{6 \times k}$ is the matrix of the chosen shape functions. Consequently, the space-rate twist ξ becomes

$$\xi = \Phi(X)q + \xi_o. \quad (5.3)$$

As a result, $(g(0), q)$ completely defines the configuration of the rod. The configuration space of Cosserat rods is reduced to

$$\mathcal{C} = SE(3) \times \mathbb{R}^k. \quad (5.4)$$

Finally, this strain reduction can be used to describe the Lagrangian balance equation, which will be presented in the next subsection.

5.1.2 Balance Equation

By applying the principle of virtual work, one can derive the Lagrangian balance equation of forces:

$$Q_r + Q_e = 0 \quad (5.5)$$

where $Q_r \in \mathbb{R}^k$ is the vector of generalized restoring forces, and $Q_e \in \mathbb{R}^k$ is the vector of generalized external forces. Assuming linear elasticity, the generalized restoring forces are expressed as:

$$Q_r = K_\epsilon q \quad (5.6)$$

where $K_\epsilon \in \mathbb{R}^{k \times k}$ is the generalized stiffness matrix, given by:

$$K_\epsilon = \int_0^l \Phi^T \mathcal{H} \Phi dX. \quad (5.7)$$

Here, l represents the unstressed length of the rod, and $\mathcal{H}$ is the Hookean matrix. Furthermore, there are two ways to compute the generalized external forces. The first approach involves projecting the stress Λ onto the shape functions:

$$Q_e = - \int_0^l \Phi^T \Lambda dX. \quad (5.8)$$

Here, Λ is computed as follows. First, forward integrate (from $X = 0$ to $X = l$) the kinematics (2.3) $g' = g\widehat{\xi}$ with the initial condition $g(0) = 1_{4 \times 4}$, where ξ computed from q using strain reduction (5.3). This allows to compute the tip wrench F_+ from its inertial frame counterpart f_+ using transformation by $\begin{pmatrix} R^T & 0 \\ 0 & R^T \end{pmatrix}$. Second, use the initial condition $\Lambda(l) = F_+$ to backward integrate (from $X = l$ to $X = 0$) the continuous balance equation:

$$\Lambda' = ad_\xi^T \Lambda - \bar{F}. \quad (\text{recall 2.9})$$

Here, $\bar{F}$ represents the distributed wrench.

Alternatively, to compute Q_e , one can use the Jacobian matrix J , which maps any variation of q to a corresponding variation of g . For instance, $\dot{g} = g\widehat{\eta} = g(\widehat{J\dot{q}})$, with $\dot{\bullet} = \frac{\partial \bullet}{\partial t}$, where in the quasi-static case, t denotes the fictitious time and η represents the velocity. Consequently, the generalized external forces are given by:

$$Q_e = - \int_0^l J^T \bar{F} dX - J(l)^T F_+. \quad (5.9)$$

Using the identity $ad_\xi = Ad_g^{-1} Ad_g'$, along with the arc length differential equation of η

$$\eta' = -ad_\xi \eta + \xi \quad (\text{recall 2.11})$$

one can show that J is given by:

$$J(X) = Ad_g^{-1} \int_0^X Ad_g \Phi dY. \quad (5.10)$$

where Ad_g being the adjoint map:

$$Ad_g = \begin{pmatrix} R & 0_{3 \times 3} \\ \widehat{r}R & R \end{pmatrix} \quad (5.11)$$

Here, r denotes the centerline position, and R denotes the cross-section rotation matrix. g is computed by arc length integration of the kinematics (2.3) $g' = g\widehat{\xi}$ with the initial condition $g(0) = 1_{4 \times 4}$, where ξ is calculated from q using strain reduction (5.3). The selection between (5.8) and (5.9) for computing the generalized external forces Q_e will be discussed in Section 5.3. In the next subsection, we will see how to solve the balance equation by computing the unknown generalized coordinates q .

5.1.3 Quasi-static Solving

The balance equation can be solved by constructing a residual vector with q as input, as shown in [37]. In this work, we avoid root-finding techniques to eliminate the need for initial guesses close to the solution for convergence. Boyer et al. [60] demonstrated that introducing artificial damping allows the balance equation (5.5) to be reformulated as an explicit first-order ordinary differential equation governing the quasi-static evolution of the generalized coordinates q :

$$\dot{q} = -D_\epsilon^{-1}(Q_e + Q_r). \quad (5.12)$$

Here, $\dot{q}$ represents the partial derivative of q with respect to the fictitious time t , which corresponds to the quasi-static evolution. Moreover, $D_\epsilon \in \mathbb{R}^{k \times k}$ is the matrix of generalized damping, which can be computed from the stiffness matrix K_ϵ as:

$$D_\epsilon = \tilde{\mu} K_\epsilon, \quad (5.13)$$

where $\tilde{\mu} \in \mathbb{R}$ is a damping coefficient.

In this section, we have presented the theoretical foundation of the Lagrangian model. In the next section, we will describe how contact handling is performed.

5.2 Contact Handling

Similar to the previous chapter, we use the implicit function ζ to model our contact surfaces (4.29). We also make the same assumptions as in the previous chapter: contact handling is performed between the surface and the cross-sectional centerline of the rod, r . Furthermore, the surface is considered rigid and fixed. Finally, the effects of blood flow are not taken into account.

5.2.1 Contact Detection

Similar to the previous chapter, the vector $w(r) \in \mathbb{R}^3$ used to project a point onto the surface, with the penalty method, is computed via the gradient of the implicit function ζ :

$$w(r) = -\frac{\nabla \zeta(r)}{|\nabla \zeta(r)|} \quad (\text{recall 4.31})$$

By a slight abuse of terminology, w is referred to as the normal vector. Recall that $\zeta(r)$ is negative when r is inside the surface and positive when it is outside of it. The minus sign in the equation above makes the normal to point inside the surface when the position r is outside. Furthermore, the signed distance between r and the surface is no longer computed iteratively as done Chapter 4. Here, a faster approach is required because time is not priorly discretized, leading to more evaluations of the distance function. Hence, we propose to use Taubin's approximation of the signed distance function [110]:

$$\phi(r) = \frac{\zeta(r)}{|\nabla \zeta(r)|} \quad (5.14)$$

In contrast to the iterative distance computation described in the previous chapter (4.32) (4.33), this approximation is less accurate. Indeed, its error increases rapidly when moving away from the surface, potentially introducing additional stiffness into the equations. Nevertheless, this stiffness is addressed through the use of implicit solvers.

We now have the essential components needed to compute the contact response, which will be described in the next subsection.

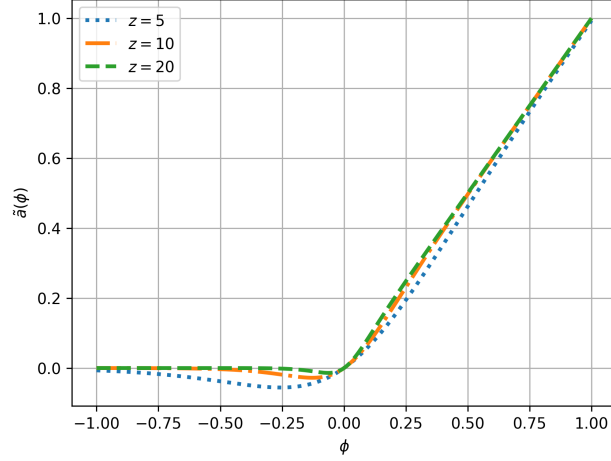


Figure 5.1: The activation function of contact forces. Collisions happen when the contact distance $\phi > 0$.

5.2.2 Contact Response

We use a penalty method to compute the contact force acting on a specific arc length point, $n_c(X) \in \mathbb{R}^3$:

$$n_c = \lambda \tilde{a}(\phi) w \quad (5.15)$$

where $\lambda \in \mathbb{R}$ is the stiffness parameter, and $\tilde{a}$ is the activation function of contact. The role of $\tilde{a}$ is to apply contact forces only when a collision is present ($\phi > 0$). In the previous chapter, the activation function was defined as the ReLU function $\max(\phi, 0)$. However, in this work, a differentiable function is required to compute the Jacobian matrix. Therefore, we propose to use the SiLU function, which is a differentiable approximation of the ReLU function:

$$\tilde{a} = \frac{\phi}{(1 + e^{-z\phi})}, \quad \frac{\partial \tilde{a}}{\partial \phi} = \frac{1}{1 + e^{-z\phi}} + \frac{z\phi e^{-z\phi}}{(1 + e^{-z\phi})^2} \quad (5.16)$$

where z is a parameter that controls the smoothness of the transition between negative and positive values of $\tilde{a}$ (see Fig. 5.1). In other words, increasing z makes $\tilde{a}$ converge to the standard activation function of the penalty method, $\max(0, \phi)$. When ϕ is negative and close to zero, $\tilde{a}(\phi)$ becomes negative, causing the rod to be attracted to the surface which is an undesirable effect. However, when z is sufficiently large, this attraction becomes negligible. Furthermore, the friction forces are not considered here and are left for future work. Indeed, incorporating a differentiable friction model is not straightforward, as discontinuities may arise between slip and stick friction. Finally, the contact force n_c does not produce a moment, as it acts on the centerline of the rod. As a result, the contact wrench (in the inertial frame) is expressed as:

$$f_c = \begin{pmatrix} 0_{3 \times 1} \\ n_c \end{pmatrix} \quad (5.17)$$

Having described how continuous contact handling is performed, the next section presents its integration into the balance equation.

5.3 Including Contacts Forces into the Balance Equation

In this section, we describe how contact wrenches f_c are incorporated into the balance equation (5.5). With a slight abuse of terminology, we will refer to f_c as the contact force rather than the contact wrench, since the contact moment is zero. First, the contact forces are included in the vector of generalized external forces Q_e . In Section 5.1, we presented two approaches for computing Q_e .

The first approach involves projecting the stress Λ onto the shape functions (5.8). Here, Λ is computed through two passes of integration: a forward pass for the kinematics and a backward pass for the continuous balance equation. The initial condition for the backward integration is obtained from the output of the forward integration, meaning they cannot be performed simultaneously. Furthermore, since integration of the continuous balance equation (2.9) is performed, the concentrated contact force f_c (in Newtons) is added to the distributed force $\bar{F}$ (in Newtons per meter), requiring a conversion as described in Chapter 4.

The second approach involves computing Q_e using the Jacobian J (5.9) (5.10). At first, this approach may seem inefficient since the integral for Q_e ($\int_0^l J^T \bar{F} dX$) requires $J(X)$, which itself contains an integral ($\int_0^X Ad_g \Phi dY$). However, because integration is a cumulative operation, computing $J(l)$ once is sufficient to retrieve any $J(X)$, with X known beforehand. Furthermore, the integrals of Q_e ($\int_0^l J^T \bar{F} dX$) and J ($\int_0^l Ad_g \Phi dY$) are forward integrations that can be performed simultaneously with the integration of the kinematics ($g' = g\hat{\xi}$) used to compute $g(X)$. Additionally, this approach eliminates the need to convert the concentrated contact force f_c into a distributed force. To explain further, let

$$F_c = \begin{pmatrix} R^T & 0 \\ 0 & R^T \end{pmatrix} f_c \quad (5.18)$$

be the counterpart of f_c in the mobile frame. Assuming that F_c act on m_c equidistant points along the rod, denoted as X_j :

$$X_j = \frac{j-1}{m_c-1}l, \quad j = \{1, \dots, m_c\}. \quad (5.19)$$

Here, $X_1 = 0$ and $X_{m_c} = l$ correspond to the rod proximal and distal ends, respectively. Then, the generalized external forces (5.9) become:

$$Q_e = - \sum_{j=1}^{m_c} J(X_j)^T F_c(X_j). \quad (5.20)$$

Consequently, we adopt the Jacobian-based approach. Putting all the pieces together, the explicit ordinary differential equation of q is expressed as:

Algorithm 3 Computation of $\dot{q}$ from q

Knowns: $q, K_\epsilon, \xi_o, \tilde{\mu}, m_b, S, \{\sigma_i\}, \{\varrho_i\}, m_c, \lambda, \{X_j\}$

- 1: Compute g by integrating the kinematics $g' = g(\Phi q + \xi_o)$ (2.3) (5.3)
 - 2: $(r(X_j), R(X_j)) = g(X_j)$
 - 3: $\zeta(r(X_j)) = S - \sum_{i=1}^{m_b} \varrho_i \Psi\left(\frac{|r(X_j) - \sigma_i|^2}{\varrho_i^2}\right)$ (4.29)
 - 4: $\nabla \zeta(r(X_j)) = - \sum_{i=1}^{m_b} \frac{2}{\varrho_i} \psi\left(\frac{|r(X_j) - \sigma_i|^2}{\varrho_i^2}\right) (r(X_j) - \sigma_i)$ (4.30)
 - 5: $\phi(r(X_j)) = \frac{\zeta(r(X_j))}{|\nabla \zeta(r(X_j))|}$ (5.14)
 - 6: $w(r(X_j)) = -\frac{\nabla \zeta(r(X_j))}{|\nabla \zeta(r(X_j))|}$ (4.31)
 - 7: $n_c(X_j) = \lambda \tilde{a}(\phi(r(X_j))) w(r(X_j))$ (5.15)
 - 8: $F_c(X_j) = (0_{1 \times 3}, (R(X_j)^T n_c(X_j))^T)^T$ (5.18) (5.17)
 - 9: Compute $\int_0^l \text{Ad}_{g(Y)} \Phi(Y) dY$ and store intermediate values $G(X_j) = \int_0^{X_j} \text{Ad}_{g(Y)} \Phi(Y) dY$
 - 10: $J(X_j) = \text{Ad}_{g(X_j)}^{-1} G(X_j)$ (5.10)
 - 11: $\dot{q} = -(\tilde{\mu} K_\epsilon)^{-1} \left(- \sum_{j=1}^{m_c} J(X_j)^T F_c(X_j) + K_\epsilon q \right)$ (5.21)
-

$$\dot{q} = -(\tilde{\mu} K_\epsilon)^{-1} \left(- \sum_{j=1}^{m_c} J(X_j)^T F_c(X_j) + K_\epsilon q \right) \quad (5.21)$$

Finally, Algorithm 3 retraces in details how $\dot{q}$ is computed from q . Furthermore, (5.21) may become stiff in complex contact scenarios, causing explicit solvers to require excessively small time steps. To address this issue, we use implicit solvers. In the next section, we analytically derive the required Jacobian matrix $\frac{\partial \dot{q}}{\partial q}$.

5.4 Jacobian Matrix

We use the notation $\bullet^\# = \frac{\partial \bullet}{\partial q_j}$ to represent the partial derivative with respect to the j -th generalized strain coordinates. Differentiating $\dot{q}$ (5.12) with respect to q_j yields:

$$\dot{q}^\# = -D_\epsilon^{-1} (Q_e^\# + K_{\epsilon_j}) \quad (5.22)$$

with K_{ϵ_j} being the j -th column of K_ϵ . Taking the derivative of Q_e (5.20) gives

$$Q_e^\# = - \sum_{j=1}^{m_c} J^{T\#}(X_j) F_c(X_j) - J^T(X_j) F_c^\#(X_j) \quad (5.23)$$

Applying differentiation to J (5.10) via the product rule and Leibniz integral rule yields:

$$J^\#(X) = \text{Ad}_g^{-1\#} \int_0^X \text{Ad}_g \Phi dY + \text{Ad}_g^{-1} \int_0^X \text{Ad}_g^\# \Phi dY \quad (5.24)$$

where the derivative of the adjoint map (5.11) is

$$Ad_g^\# = \begin{pmatrix} R^\# & 0_{3 \times 3} \\ \hat{r}^\# R + \hat{r} R^\# & R^\# \end{pmatrix}. \quad (5.25)$$

Computing the derivative of F_c (5.18) results in

$$F_c^\# = \begin{pmatrix} 0_{3 \times 1} \\ R^{T^\#} n_c + R^T n_c^\# \end{pmatrix} \quad (5.26)$$

with differentiation of n_c (5.15) yields

$$n_c^\# = \lambda(\tilde{a}^\# w + \tilde{a} w^\#). \quad (5.27)$$

Using the chain rule, we obtain

$$\tilde{a}^\# = \frac{\partial \tilde{a}}{\partial \phi} \frac{\partial \phi}{\partial r} r^\# \quad (5.28)$$

$$w^\# = \frac{\partial w}{\partial r} r^\# \quad (5.29)$$

The last two equations show that the distance and normal functions should be differentiable in 3D space. Therefore, explicit surfaces such as triangle meshes are less suitable for implicitly solving $\dot{q}$ (5.21). Using the quotient rule and simplifying, we obtain

$$\frac{\partial \phi}{\partial r} = \left(\phi \frac{\nabla^2 \zeta}{|\nabla \zeta|} - I \right) w \quad (5.30)$$

$$\frac{\partial w}{\partial r} = (w w^T - I) \frac{\nabla^2 \zeta}{|\nabla \zeta|} \quad (5.31)$$

where $I \in \mathbb{R}^{3 \times 3}$ is the identity matrix, and $\nabla^2 \zeta$ is the Hessian of ζ computed in closed form by differentiating (4.29) with respect to r twice:

$$\nabla^2 \zeta(r) = - \sum_{j=1}^{m_b} \left[\frac{4}{\varrho_j^3} \frac{d\psi}{dx} \left(\frac{|r - \sigma_j|^2}{\varrho_j^2} \right) (r - \sigma_j)(r - \sigma_j)^T + \frac{2}{\varrho_j} \psi \left(\frac{|r - \sigma_j|^2}{\varrho_j^2} \right) \right]. \quad (5.32)$$

where ψ (4.28) and $\frac{d\psi}{dx}$ are the first and second derivatives of the Cauchy profile Ψ (4.27), respectively.

$$\frac{d\psi(x)}{dx} = \frac{150}{(x + 5)^4} \quad (5.33)$$

Finally, $r^\#$ and $R^\#$ are computed using $g^\#$. Since q_j does not depend X , their differentiation commutes.

$$(g')^\# = (g^\#)' \quad (5.34)$$

Let the twist $\varpi = (g^{-1}g^\sharp)^\vee$. Using (5.34), one can obtain the derivative of ϖ with respect to X :

$$\varpi' = -ad_\xi \varpi + \xi^\sharp = -ad_\xi \varpi + \Phi_j. \quad (5.35)$$

Here, Φ_j denotes the j -th column of Φ . The computation of ϖ by integrating (5.35) using the initial condition $\varpi(0) = 0_{6 \times 1}$ allows to calculate $g^\sharp$.

Finally, the described derivation is used to construct each j -th column of the Jacobian matrix that allows for implicit time integration.

5.5 Insertion and Removal Approach

Up to this point, we have described how to obtain the quasi-static evolution of a clamped-free Cosserat rod undergoing contact. However, in catheter navigation procedures, the catheter is rotated and displaced at the proximal end. We propose modeling this by modifying the initial conditions of the previously described initial value problems. We introduce a time-dependent variable $s(t) \in [0, l]$. The arc length interval of the inserted portion becomes $[s, l]$, while the portion in $[0, s]$ is visually hidden and excluded from contact handling. The initial condition of the kinematics (2.3) becomes $g(s)$ instead of $g(0)$, where $r(s) = 0_{3 \times 1}$ is fixed and $R(s)$ can vary to allow rotation. Additionally, the initial condition of ϖ (5.35) becomes $\varpi(s) = 0_{6 \times 1}$ instead of $\varpi(0) = 0_{6 \times 1}$. At each time step t , if $s(t + \Delta t) < s(t)$, insertion is performed; conversely, if $s(t + \Delta t) > s(t)$, removal is performed. The integrals of (5.10) and (5.24) now start from s instead of 0, and X_j consists of m_c equidistant arc length points in $[s, l]$:

$$X_j = s + \frac{j-1}{m_c-1}(l-s), \quad j \in \{1, \dots, m_c\}. \quad (5.36)$$

Note that our continuous approach to contact handling enables the modification of contact discretization nodes X_j at each time step.

5.6 Numerical Applications

We tested our method through three numerical applications, each with increasing surface complexity. The objectives were as follows: to understand the effects of damping and total simulation time, to assess the insertion/removal approach, to evaluate the method using real patient geometry, and to test it across different rod material (steel, rubber, and catheter). Following [60, 37], we used Legendre polynomials for the shape functions Φ , with the number of shape functions set to $k_j = 8$. The parameter z of the SiLU activation function $\tilde{a}$ (5.16) was chosen empirically to prevent attraction to the surface. The code was implemented in Python, using the SciPy library for numerical integrations. For arc length integration, the Runge-Kutta 4(5) method was used. For time integration, the backward differentiation formula was used in the first two numerical applications, while the implicit Runge-Kutta method of order 5 (Radau) was employed in the last application due to its more complex contact scenario. The numerical integrations were performed with SciPy's built-in default

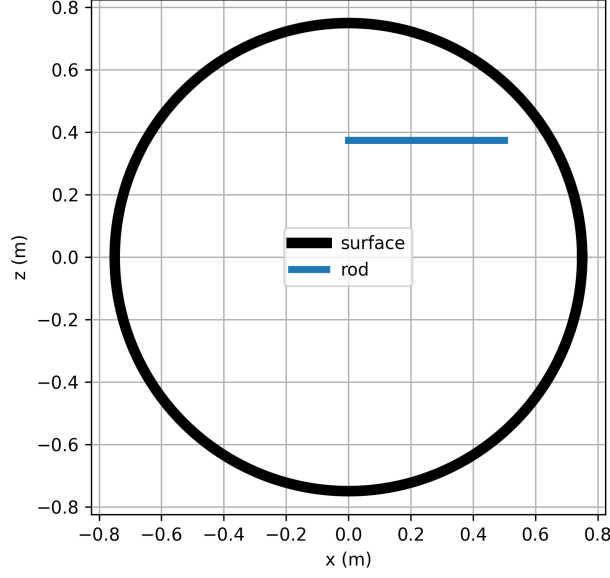


Figure 5.2: The rod base is pushed then pulled in a sphere (black line). This figure displays the rod initial configuration.

tolerances. We compared the execution time of time integration using an implicit solver versus an explicit solver, the latter employing the Runge-Kutta 4(5) method. The simulations were performed on a computer equipped with an Intel Xeon W-2245 CPU running at 3.90 GHz. Finally, the implementation is made open-source on gitlab.inria.fr/rjilani/RAL2024.

5.6.1 Rod Push-Pull in a Sphere

This numerical application aims to understand the effects of the damping coefficient $\tilde{\mu}$ and the total time T on the simulations. A straight steel rod is placed inside a sphere; its base is pushed along the x-axis toward the sphere interior surface and then pulled back to its original location. Fig. 5.2 shows the initial configuration of this scenario. The mechanical parameters of the rod are: the length $l = 0.5$ m, the cross-section radius $R_b = 0.001$ m, Young's modulus $E = 200 \times 10^9$ Pa, and the shear modulus $G = 80 \times 10^9$ Pa. Contact handling parameters are: the number of contact nodes $m_c = 100$, the penalty stiffness $\lambda = 10^4$ N/m, and the activation function parameter $z = 0.5 \times 10^6$. The sphere is centered at $(0, 0, 0)^T$, and its radius $R_s = 1.5l$. The sphere was created with a blobby model containing $m_b = 1$ blob, with center $\sigma = (0, 0, 0)^T$, width $\varrho = R_s$, and threshold $S = \frac{R_s}{(\frac{1}{5}+1)^2}$. The pushing phase occurs during the first half of the simulation, while the pulling phase occurs during the second half. The path of the rod base is $r(0) = (x(t), 0, \frac{R_s}{2})^T$ with:

$$x(t) = \begin{cases} 2.5l \frac{t}{T} & \frac{t}{T} \leq 0.5 \\ 2.5l (1 - \frac{t}{T}) & \frac{t}{T} > 0.5. \end{cases} \quad (5.37)$$

Multiple simulations were conducted. In the first part, we held $T = 1$ s constant while varying $\tilde{\mu} = \{1, 0.1, 0.01, 0.001\}$. Fig. 5.3 displays the evolution over time of the rod tip along

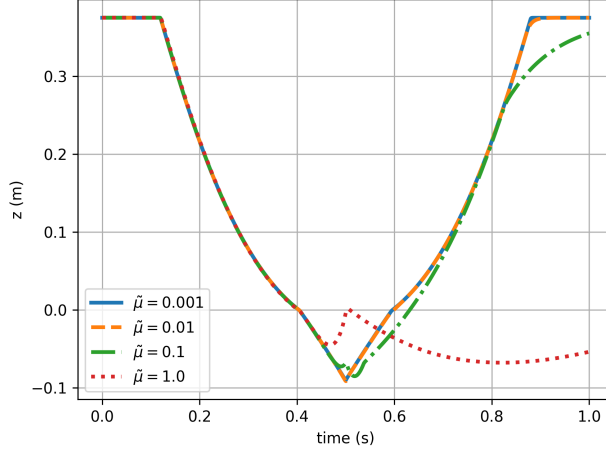


Figure 5.3: The rod base is pushed then pulled in a sphere. This figure displays the rod tip position along the z -axis with respect to time for different damping coefficients $\tilde{\mu}$.

the z -axis for different damping coefficients. The first point of contact is near $t = 0.1$ s. All simulations show nearly identical results until differences start to appear around $t = 0.45$ s. Since a quasi-static mode is used, and the contact scenario is simple, the results of pushing and pulling phases should be symmetrical (on both sides of $t = 0.5$ s). As expected, by decreasing $\tilde{\mu}$, the simulations converge to an accurate (symmetrical) solution. Fig. 5.4 shows different snapshots of the simulation with varying damping coefficients $\tilde{\mu} = \{1, 0.1, 0.01\}$. Note that the results for $\tilde{\mu} = 0.001$ are not shown, as they are visually similar to those for $\tilde{\mu} = 0.01$. We observe that $\tilde{\mu} = 0.01$ produces a realistic interaction with the sphere (particularly at $\frac{t}{T} = 0.5$), and by the end of the simulation, the rod correctly returns to its initial configuration. In contrast, $\tilde{\mu} = 1$ does not return to the initial configuration, and its high damping induces an unrealistic collision at $\frac{t}{T} = 0.5$. Table 5.1 presents the computation time, the number of evaluations of $\dot{q}$ (5.21) (nfev), and the number of evaluations of the Jacobian $\frac{\partial \dot{q}}{\partial q}$ for different values of $\tilde{\mu}$, performed using implicit integration with both analytical and numerical (finite difference) Jacobians, as well as explicit integration. The results show that our proposed analytical formulation is the most efficient across all metrics. Furthermore, as the damping decreases, the proposed method shows a slower rate of increase in computation time, nfev, and njev compared to the explicit and numerically implicit integration methods.

In the second part of our test, we fixed $\tilde{\mu} = 1$ and changed $T = \{1, 10, 100, 1000\}$ s. By comparing the results to those obtained when T was constant and $\tilde{\mu}$ varied, we noticed that the simulations with the same ratio $\frac{T}{\tilde{\mu}}$ produced the same results (apart from negligible differences caused by numerical errors). Furthermore, different choices of T or $\tilde{\mu}$ affect the numerical conditioning. In the following, they were chosen empirically to ensure convergence. We can conclude that larger damping causes strains induced by contact to persist longer. In contrast, reducing the damping allows the rod to return to its initial configuration more quickly. To summarize, although damping is inherently artificial, when it is sufficiently low or the total time is sufficiently long, our method yields accurate quasi-static solutions with contact.

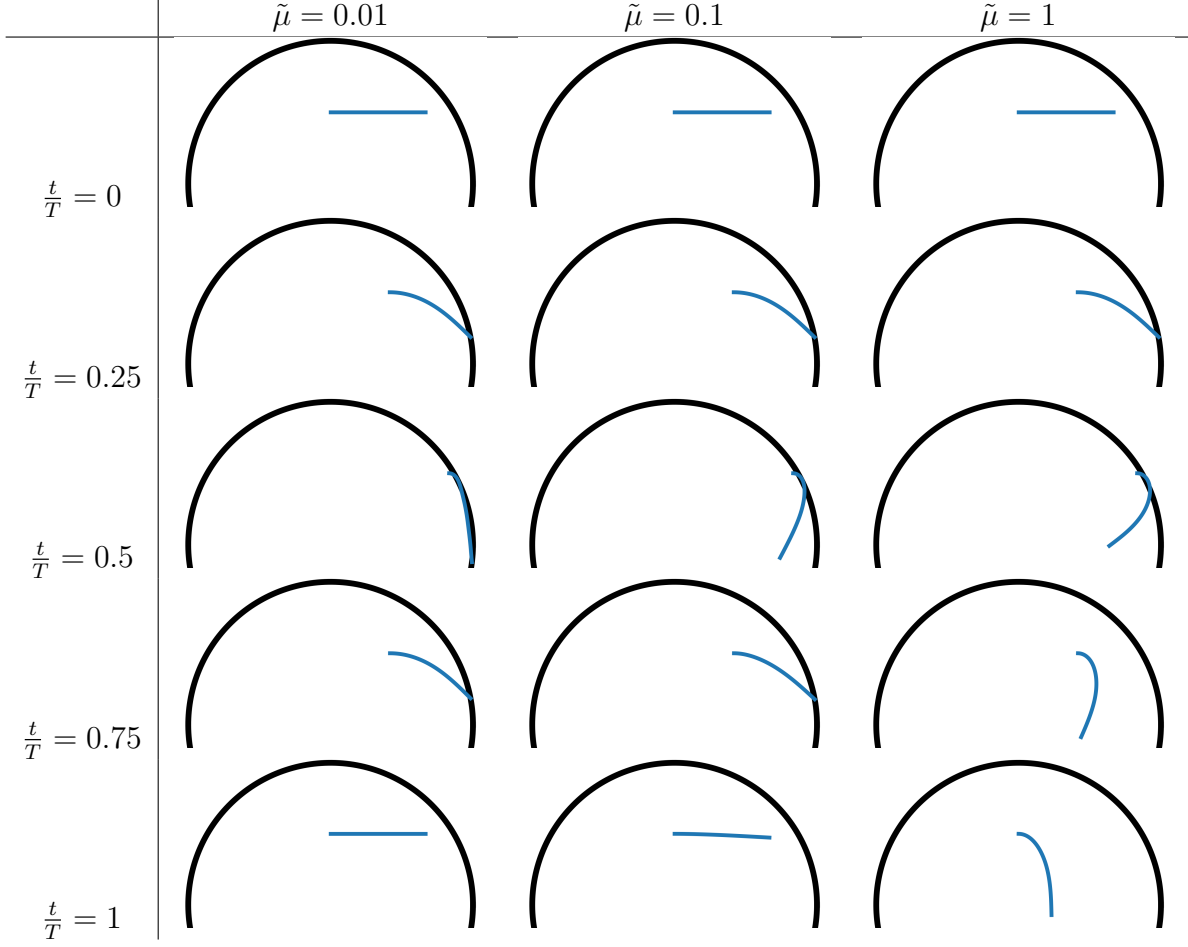


Figure 5.4: Snapshots at different times and varying damping coefficients $\tilde{\mu}$ show a rod pushed into the surface of a sphere.

5.6.2 Rod Insertion and Removal in a Helical Tube

The aim of this numerical application is to assess the correctness of the proposed insertion and removal approach. A straight rubber rod is inserted into a tube containing a helix and is then removed (Fig. 5.5). The surface is the same as the one used in the previous chapter (Fig. 4.5). It was modeled using a blobby model containing $m_b = 45$ blobs, with width $\varrho_i = 2 \times 10^{-3}$, threshold $S = 2 \times 10^{-5}$, and blob centers σ_i representing the discrete centerline of the helical tube. The tube radius is 0.016 m and the helix radius is 0.03 m. The mechanical parameters of the rod are: the length $l = 0.7$ m, the cross-section radius $R_b = 5 \times 10^{-3}$ m, Young's modulus $E = 5 \times 10^6$ Pa, Poisson's ratio $\nu = 0.5$, and the shear modulus $G = \frac{E}{2(1+\nu)}$. The numerical parameters are: the total time $T = 3000$ s and the damping $\tilde{\mu} = 3$, leading to $\frac{T}{\tilde{\mu}} = 1000$. Contact handling parameters are: the number of contact nodes $m_c = 100$, the stiffness $\lambda = 1000$ N/m, the activation function parameter $z = 0.7 \times 10^7$. The insertion and removal parameter $s(t)$ evolves as follows:

$\tilde{\mu}$	Method	Time	nfev	njev
1	implicit analytical	7 s	510	26
	implicit numerical	14 s	480	26
	explicit	37 s	6.3×10^3	-
0.1	implicit analytical	13 s	700	53
	implicit numerical	40 s	680	54
	explicit	2 min	2.0×10^4	-
0.01	implicit analytical	25 s	1.1×10^3	100
	implicit numerical	2 min	1.7×10^3	160
	explicit	26 min	2.5×10^5	-
0.001	implicit analytical	1 min	4.5×10^3	380
	implicit numerical	13 min	1.4×10^4	1.3×10^3
	explicit	4 h	2.6×10^6	-

Table 5.1: Performance comparison of implicit analytical, implicit numerical, and explicit integration methods for various damping coefficients $\tilde{\mu}$. Time (in seconds, minutes, or hours), nfev (number of function evaluations), and njev (number of Jacobian evaluations) are reported. Explicit methods do not compute Jacobians, hence njev is omitted. Best performances (lowest time, nfev, or njev) are highlighted in bold.

$$s(t) = \begin{cases} 2l(0.5 - \frac{t}{T}) & \frac{t}{T} \leq 0.5 \\ 2l(\frac{t}{T} - 0.5) & \frac{t}{T} > 0.5. \end{cases} \quad (5.38)$$

This corresponds to complete insertion during the first half of the simulation and complete removal during the second half. Fig. 5.5 shows four snapshots of the first half of the simulation (the insertion phase). The removal phase is not shown in the snapshots because it is visually similar to the insertion phase. The insertion and removal are performed correctly and the rod is properly contained within the surface. Fig. 5.6 (left) shows the evolution of the rod tip over time. The first contact occurs around $t = 750$ s, and the last one around $t = 2250$ s. The results of the insertion and removal phases are nearly symmetrical. Fig. 5.7 shows the norm of the contact force at the rod tip. As expected, the contact forces during the insertion and removal phases are also nearly symmetrical. This simulation took 8 minutes to complete, with $\text{nfev} = 16,405$ and $\text{njev} = 1,728$, which is relatively fast for such a contact scenario. The explicit solver required prohibitively large computation time when using a ratio of $\frac{T}{\tilde{\mu}} = 1000$. Thus, we reduced the total simulation time to $T = 300$ and set $\tilde{\mu} = 3$ to increase damping, resulting in a ratio of $\frac{T}{\tilde{\mu}} = 100$. Fig. 5.6 (right) shows the evolution of the rod tip position over time, comparing the explicit and implicit solvers. We conclude that the results obtained with both solvers are in good agreement. Furthermore, when comparing Fig. 5.6 left ($\frac{T}{\tilde{\mu}} = 1000$) and right ($\frac{T}{\tilde{\mu}} = 100$), we observe, as expected, that the increased damping results in a smoother solution. This effect is particularly noticeable at $t = 225$ s, corresponding to the final contact between the surface and the rod tip. The explicit solver required 9 hours to complete, with $\text{nfev} = 4,738,478$, whereas the implicit solver finished in 169 seconds, with $\text{nfev} = 6,998$ and $\text{njev} = 777$.

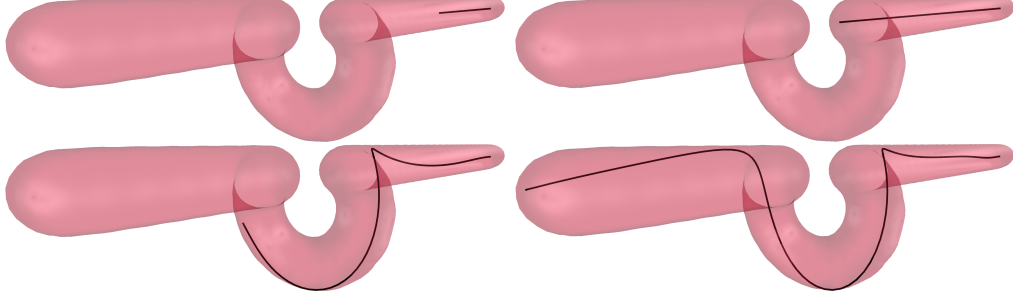


Figure 5.5: A rubber rod insertion in a helical tube. From left to right, and top to bottom, four snapshots taken at $t = \frac{T}{8}, \frac{T}{4}, \frac{3T}{8},$ and $\frac{T}{2}$, corresponding to $t = 375, 725, 1125,$ and 1500 s, respectively.

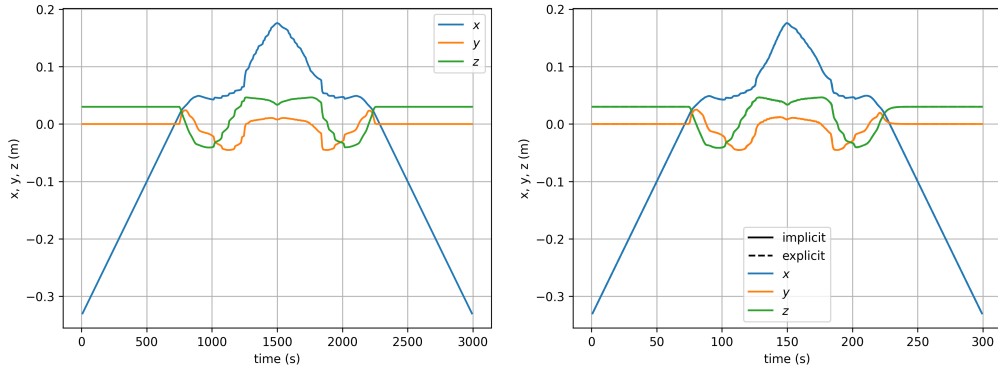


Figure 5.6: A rubber rod is inserted into, then removed from, a helical tube. This figures display the evolution of the rod tip position over time. Left: $\frac{T}{\mu} = 1000$, and right: $\frac{T}{\mu} = 100$.

In summary, this numerical application shows the computational speed of the proposed method, while verifying the correctness of our insertion and removal approach.

5.6.3 Catheter Insertion into a Carotid Artery

This numerical application tests our method in a simulation involving complex, noisy real patient geometry. A straight catheter was inserted into a model of a carotid artery constructed from a patient 3D rotational angiography data [104]. The surface is the same as the one used in the previous chapter. It was modeled using a blobby model containing $M = 2573$ blobs. The catheter mechanical parameters are: the length $l = 0.065$ m, the cross-section radius $R_b = 2.8 \times 10^{-4}$ m, Young's modulus $E = 300 \times 10^6$ Pa, Poisson's ratio $\nu = 0.49$, and the shear modulus $G = \frac{E}{2(1+\nu)}$ Pa. Contact handling parameters are: the number of contact nodes $m_c = 50$, the activation function parameter $z = 0.065 \times 10^7$, and the stiffness $\lambda = 1000$ N/m. The insertion is performed linearly using $s(t) = l \left(1 - \frac{t}{T}\right)$. The numerical parameters are: the total time $T = 0.1$ s and the damping $\tilde{\mu} = 0.001$. Here, $\frac{T}{\mu} = 100$, which is smaller than in the previous numerical example, as greater damping was required for stability due to the increased difficulty of the simulation.

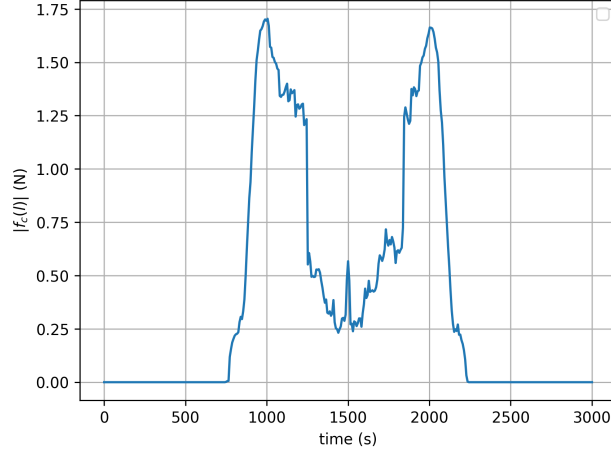


Figure 5.7: A rubber rod is inserted into, then removed from, a helical tube. This figure shows the evolution of the contact force norm at the rod tip over time.

Fig. 5.8 shows four snapshots of the simulation. The catheter is successfully inserted and properly contained within the surface. The catheter experiences two blockages, which occur due to small bumps on the surface, causing an effect similar to stick-slip friction. Fig. 5.9 shows the position of the catheter tip over time, highlighting the stick-slip effect. This simulation took 44 minutes to complete, with $n_{\text{fev}} = 33,433$ and $n_{\text{jev}} = 1,536$. The computation time of the explicit solver is not available, as the simulation exceeded a practical time limit (i.e., several days) without completing. To summarize, this numerical application highlights the applicability of the proposed method to complex contact scenarios.

5.7 Conclusion

The objective of this chapter was to propose a faster alternative for the dynamic Cosserat rod model. To achieve this, we introduced a quasi-static assumption and adopted the damped Lagrangian model, which allows the quasi-static evolution of the rod to be obtained by integrating a minimal set of first-order explicit ordinary differential equations. However, these equations can become stiff in complex contact scenarios, forcing explicit solvers to take excessively small steps. To address this issue, we proposed an implicit method and derived its Jacobian matrix analytically. This analytical derivation was made possible by using implicit functions to model the contact surfaces. Additionally, contact handling was discretized in arc length while remaining continuous in time.

In the numerical tests, we first verified that when damping is sufficiently low or the total simulation time is long enough, the solutions converged accurately with negligible damping effects. Secondly, we showed that implicit solvers reduced computation time by up to two orders of magnitude compared to explicit solvers. Finally, we assessed the applicability of the proposed method in a challenging contact scenario involving real patient geometry and actual catheter materials.

Compared to the dynamic solution using the collocation method presented in Chapter 4, the proposed method successfully reduced computation time from 44 hours to 8 minutes

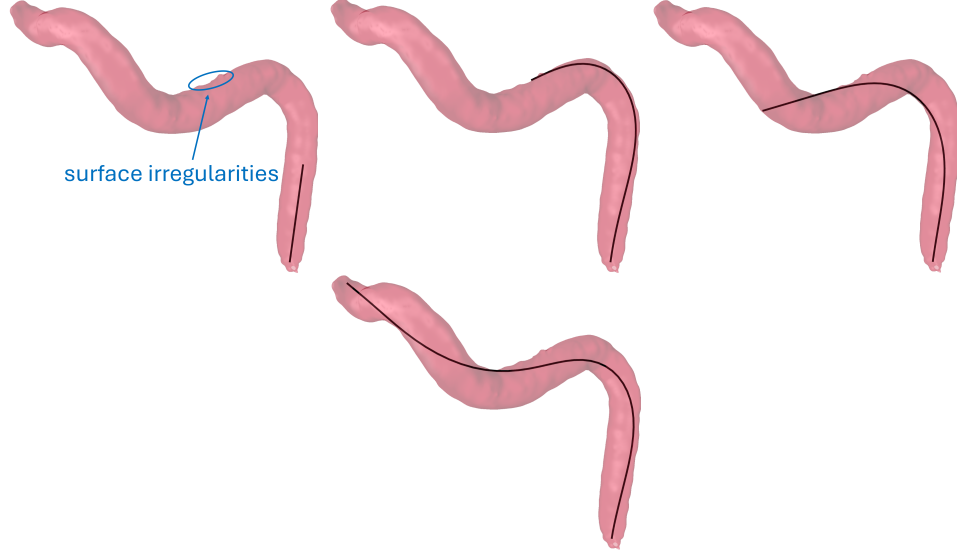


Figure 5.8: A catheter is inserted in a carotid artery model of a patient. From left to right, four snapshots taken approximately at $t = 0.025, 0.064, 0.074$, and 0.1 s. In the second snapshot, the catheter is blocked by a small bump on the surface.

for the helical tube scenario (Section 4.7.2), and from 12 hours to 44 minutes for the carotid artery scenario (Section 4.7.3). Although the time gain is substantial, achieving accurate real-time simulation of catheter navigation procedures remains out of reach. To further improve computation time, future work may include arc length integration using the Magnus expansion and a collocation method, along with GPU or CPU parallelization of the Jacobian matrix $(\frac{\partial \dot{q}}{\partial q})$ columns computation. Computation time might also benefit from the application of the Kirchhoff assumptions, which simplify the problem to bending and torsion only, effectively removing the stiff degrees of freedom associated with shear and extension. In this work, friction was neglected. Indeed, incorporating a differentiable friction model required for the implicit method is non-trivial, as discontinuities may arise between sticking and slipping behavior. This limitation will be addressed in future work. Moreover, adopting the static model necessitates the use of clamped-free boundary conditions. In contrast, the free-free boundary conditions used in Chapter 4 lead to different rod deformation results. Future work will involve comparing the clamped-free and free-free approaches and validating them against experimental data.

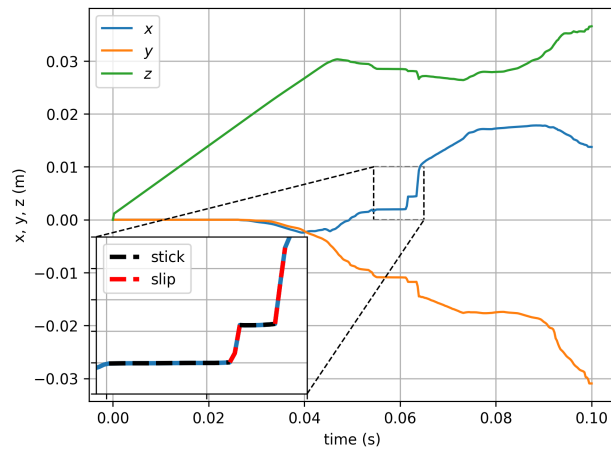


Figure 5.9: A catheter is inserted in a carotid artery model of a patient. The figure shows the catheter tip evolution over time and highlights the stick-slip effect.

Chapter 6

Conclusion and Perspectives

6.1 Summary of Contributions

The primary objective of this dissertation has been to develop robust, computationally efficient, and physically accurate numerical simulations for catheter navigation, specifically addressing clinical challenges associated with endovascular procedures such as thrombectomy for ischemic stroke treatment. The complexity and precision required for manually steering flexible catheters through tortuous and varied patient-specific vascular anatomies highlight the necessity for sophisticated computational tools. Central to this research is the one-dimensional geometrically exact Cosserat rod model, chosen for its ability to accurately capture large deformations, rotations, and intricate interactions encountered during catheter navigation. However, the numerical resolution of this model has been constrained by significant numerical instabilities, computational inefficiencies, and the intricate modeling required for contact and frictional interactions.

To address these challenges, this dissertation contributes to both theoretical frameworks and computational methodologies across multiple interconnected stages. Initially, the development of a novel dynamic numerical method using orthogonal collocation with Chebyshev polynomials was introduced to robustly overcome singularities intrinsic to boundary value problems of dynamic Cosserat rods. This numerical method, accompanied by an analytical derivation of the Jacobian matrix, significantly improved the robustness, particularly compared to traditional approaches like the shooting method. Numerical validations demonstrated the superiority of the approach, particularly in challenging scenarios involving soft rod materials, large deformations, and very small time steps. Through comprehensive testing across multiple nonlinear static and dynamic benchmark cases from the literature, positional errors were less than 0.05% deviation from analytical solutions in static tests and below 0.9% when compared to established dynamic methods.

Further analysis into the computational aspects of the orthogonal collocation method revealed important insights. By increasing polynomial degrees, ill-conditioned numerical problems transitioned to well-conditioned ones, improving solver convergence and stability. It was also highlighted that higher polynomial degrees, although improving robustness, increased computational times considerably, caused by the increase of Jacobian dimensions. Detailed profiling indicated that approximately two-thirds of the computational time was

dedicated to Jacobian matrix computations, predominantly in arc length integrations. These findings identified clear opportunities for significant computational acceleration through parallelization.

Expanding upon the orthogonal collocation framework, this dissertation made additional advancements by developing a new formulation to address free-free boundary conditions, beyond the initially investigated clamped-free boundary condition scenarios, which represents an important extension for catheter steering. A method for integrating concentrated penalty-based contact forces as distributed loads within the continuous Cosserat rod equations was proposed, addressing critical gaps in prior literature. Furthermore, implicit surface modeling was employed to improve computational efficiency in collision detection processes, significantly reducing computational time and memory usage. Numerical validations employing realistic catheter navigation scenarios, including patient carotid artery geometries and realistic catheter material properties, demonstrated not only the practical applicability of the proposed method but also its computational robustness at significantly high polynomial degrees (up to 500), allowing accurate simulations even under very small time steps without the previously required artificial numerical damping.

Despite these improvements, the computational time associated with dynamic simulations still posed practical constraints. Recognizing the limitations inherent to such scenarios, especially in clinical environments requiring fast simulations, this research introduced a quasi-static approach justified by the minimal inertial effects observed during typical slow catheter insertion procedures. This quasi-static formulation leveraged a damped Lagrangian strain parameterization framework, significantly reducing degrees of freedom. Although efficient in terms of degrees of freedom, the resulting equations were stiff, necessitating prohibitively small time steps for the time integration of generalized coordinates. To address this issue, we employed implicit time integration. This solution was enabled by the smoothness of implicit surface modeling, which also facilitated the analytical derivation of the Jacobian matrix. Performance comparisons demonstrated significant computational improvement, with reductions in computation time of up to two orders of magnitude compared to explicit solvers, and one order of magnitude faster than the collocation method in dynamic regimes.

A fundamental principle guiding this research has been transparency and reproducibility. Therefore, all implementations developed throughout this dissertation have been made publicly accessible via an open-source repository.

The contributions made in this dissertation advance the state-of-the-art in the dynamic and quasi-static simulation of Cosserat rods, specifically tailored to catheter navigation. However, several limitations remain that define the scope and clinical relevance of this work. Despite these efficiency gains, the simulations do not yet achieve real-time performance, which limits their immediate applicability in clinical procedures. Additionally, the simulations focus solely on catheter behavior, excluding other essential devices such as stent retrievers, guidewires, and introducers. This narrows the ability to fully represent actual endovascular interventions. While the methods were tested numerically using patient-specific vascular geometries, the results have not been directly confirmed through experimental or clinical studies, leaving the predictive accuracy under real-world conditions uncertain. The model also omits the deformable response of blood vessels to catheter interactions and changes in internal pressure, potentially missing critical biomechanical feedback mechanisms present in practice. Furthermore, the effects of blood flow on catheter navigation and

frictional behavior are entirely neglected, simplifying the model but overlooking important fluid-structure interactions that could influence catheter dynamics.

Another simplification concerns the initial geometry of the catheter, which is modeled as straight, whereas real catheters are stored in a coiled configuration that can lead to residual plastic deformations, which may persist when the catheter is deployed. Finally, the penalty-based method used to simulate frictional contact, though computationally efficient, may inadequately capture the complex, nonlinear interactions occurring during real catheter-vessel contacts.

These limitations offer insights that help define the boundaries of this research and guide the future directions explored in the following perspectives section.

6.2 Perspectives

A key limitation of this work lies in computational efficiency, primarily due to the intensive evaluation of the numerical integrations required for each column of the Jacobian matrix during orthogonal collocation. Several strategies could help address this challenge. Parallelization techniques applied to the computation of the Jacobian columns, including the use of multicore processors or GPU acceleration, offer promising avenues to reduce simulation runtimes. Another important direction involves gaining a deeper understanding of how increasing the polynomial degree in the collocation method transforms an ill-conditioned dynamic boundary value problem into a well-conditioned one. Such insights could guide the development of approaches that employ lower polynomial degrees to reduce the Jacobian matrix's dimension, thereby improving computational efficiency. Additionally, simplifying the modeling assumptions without compromising accuracy can further reduce computational demands. For example, adopting Kirchhoff's assumption, which neglects the stiff degrees of freedom associated with shear deformation and extensibility, provides a means to reduce complexity while preserving essential physical fidelity.

Beyond computational considerations, refining the frictional contact model remains an important objective. This study employed a penalty-based method for its simplicity. However, this approach does not fully capture the complex nonlinear interactions between catheters and vascular walls. Future research should explore complementarity-based formulations, such as nonlinear complementarity problems, to achieve a more accurate representation of contact mechanics, though these methods may substantially increase computation time. Another promising area is the development of inverse problem formulations that enable physicians to define objectives at the distal end, such as target positions or orientations, and compute the required proximal displacements and rotations to achieve these outcomes.

Among these various perspectives, the two subsequent subsections focus respectively on the estimation of the reference curvature for the catheter and on the application of machine learning methods to solve the Cosserat rod model. We have carried out preliminary studies of both topics, but they still require more in-depth analysis. For instance, physics-informed neural networks (PINNs) have recently emerged as an alternative for solving physical models [59], but to the best of our knowledge, no effective PINN-based formulation has yet been developed for Cosserat rods. The preliminary work presented here aims to identify the key

scientific challenges that need to be addressed to enable the successful application of this approach.

6.2.1 Estimating the Reference Curvature

Throughout this dissertation, the catheter’s reference configuration has been assumed to be perfectly straight. In practice, however, catheters are often packaged in a coiled configuration to accommodate their considerable lengths. In addition, many catheters are intentionally pre-curved at the distal end to facilitate passage through vascular bifurcations. As a result of material plasticity and design features, the actual reference configuration deviates substantially from a straight shape. This initial curvature influences the catheter’s mechanical behavior, particularly its deformation and interactions with vessel walls during navigation within the vascular network. Accurately accounting for this reference configuration is therefore essential to ensure the physical fidelity of numerical simulations.

The influence of the reference configuration is explicitly accounted for in the strain definition:

$$\epsilon = \xi - \xi_o \quad (\text{recall 2.5})$$

Here, $\xi = (K^T, \Gamma^T)^T$ denotes the space-rate twist, where K represents the angular component and Γ the linear component, while $\xi_o = (K_o^T, \Gamma_o^T)^T$ corresponds to its reference configuration counterpart. Throughout this work, we simplified the study by assuming a straight reference configuration, yielding $\xi_o = (0, 0, 0, 1, 0, 0)^T$. An essential direction for future research involves developing methods to accurately compute the actual reference twist ξ_o .

To address this, we propose an initial method for estimating the intrinsic curvature K_o . This formulation assumes a planar reference configuration and applies Kirchhoff’s assumption, which simplifies the model by neglecting shear and extensional effects through the condition $\Gamma = \Gamma_o = (1, 0, 0)^T$. The first step of the envisioned method consists of an experimental pipeline designed to reconstruct the catheter’s configuration under gravitational loading. A segment of the catheter intended for numerical simulation is physically clamped, and its centerline r is reconstructed using multi-view stereo vision techniques [111]. The second step of this procedure is to determine the intrinsic curvature K_o using the centerline r . Starting from the linear elastic constitutive relation that links the internal moment c (expressed in the inertial frame) to the curvature, we have:

$$c = R\mathcal{H}_a(K - K_o), \quad (6.1)$$

where R is the rotation matrix representing the cross-sectional orientation, and $\mathcal{H}_a$ is the bending and torsional stiffness matrix. Rearranging this expression allows us to isolate K_o :

$$K_o = K - \mathcal{H}_a^{-1}R^T c. \quad (6.2)$$

Determining K_o thus requires the curvature K , the internal moment c , and the orientation R . Given the reconstructed centerline r , the orientation R can be computed using the method of parallel transport. This method generates an orthonormal frame along the curve

by smoothly propagating an initial orientation in a way that minimizes rotation around the tangent direction [35]. The curvature K can subsequently be derived from R by approximating its spatial derivative through finite differences:

$$K = (R^T R')^\vee, \quad (6.3)$$

To compute the internal moment c , we consider the static equilibrium of the catheter under gravity, described by the following differential equations:

$$\begin{aligned} n' &= -\rho A \mathcal{G}, \\ c' &= -r' \times n = -(R\Gamma) \times n = -d_1 \times n, \end{aligned} \quad (6.4)$$

where n is the internal force, ρ the material density, A the cross-sectional area, $\mathcal{G}$ the gravitational acceleration vector, and d_1 the first column of the rotation matrix R , which corresponds to r' under Kirchhoff's assumption $\Gamma = (1, 0, 0)^T$. These equations are integrated backward from the free end ($X = l$), with boundary conditions $n(l) = 0$ and $c(l) = 0$, to obtain $c(X)$.

Tested on synthetic data, this approach provided a good approximation of K_o . However, a critical extension of this work is to relax the current simplifying assumptions, enabling the computation of K_o corresponding to arbitrary three-dimensional reference configurations and validating the method against experimental data. The main challenge lies in the limitations of the parallel transport technique: while it provides a unique and consistent orientation along curves in two dimensions, in three dimensions it may yield frames that do not capture the true rotational field of the catheter's cross sections. Addressing this limitation will be a key focus of future work.

6.2.2 Physics-Informed Neural Networks

This subsection focuses on applying PINNs framework to the static mode of Cosserat rods, which serves as a preliminary step toward the goal of extending the approach to dynamic scenarios. In the following, we briefly introduce the formulation of the static Cosserat rod boundary value problem in the inertial frame (in contrast to the mobile frame formulation presented in (2.14) (2.15)). We then propose a solution based on PINNs and demonstrate its applicability through a numerical example. The subsection concludes with a discussion on the advantages, limitations, and future research directions of this approach.

Boundary Value Problem

We consider the static clamped-free boundary value problem of a Cosserat rod formulated in the inertial frame, as described in [61]. The governing equations describe the evolution of the rod's position r , orientation R (as a rotation matrix), internal force n , and internal moment c along the arc length $X \in [0, l]$, where l is the length of the rod:

$$\begin{pmatrix} r' \\ R' \\ n' \\ c' \end{pmatrix} = \begin{pmatrix} R\Gamma \\ R\hat{K} \\ -\bar{n} \\ -r' \times n - \bar{c} \end{pmatrix} \quad (6.5)$$

The rod is subjected to distributed force $\bar{n}$ and distributed moment $\bar{c}$. These equations are complemented by the following boundary conditions, representing a clamped base at $X = 0$ and a free end at $X = l$:

$$r(0) = 0_{3 \times 1}, \quad R(0) = 1_{3 \times 3}, \quad n(l) = n_+, \quad c(l) = c_+ \quad (6.6)$$

The strain measures of the rod are given by the deviations $\Gamma - \Gamma_0$ and $K - K_0$, corresponding to the linear and angular space rates relative to their reference configurations. These strains are related to the internal force n and internal moment c through linear elastic constitutive laws:

$$\begin{aligned} \Gamma &= \Gamma_o + \mathcal{H}_l^{-1} R^T n \\ K &= K_o + \mathcal{H}_a^{-1} R^T c \end{aligned} \quad (6.7)$$

In these relations, $\mathcal{H}_l$ and $\mathcal{H}_a$ are the shear-extension and bending-torsion stiffness matrices, respectively.

Solution via Physics-Informed Neural Networks

To solve the boundary value problem (6.5) (6.6) using PINNs, we define the neural network $\mathcal{N}_\theta$, parameterized by weights and biases θ , which maps the rod's arc length $X \in [0, l]$ to the corresponding internal moment $c_\theta(X)$:

$$\mathcal{N}_\theta : X \mapsto c_\theta(X). \quad (6.8)$$

Similar to collocation methods, PINNs provide a continuous functional approximation and enforce the equilibrium equations and boundary conditions at a finite set of collocation points $\{s_i\}_{i=1}^m$ distributed along the rod's arc length. We assume that the first collocation point s_1 coincides with the clamped end $X = 0$, and the last collocation point s_m with the free end $X = l$. In contrast to collocation methods, which directly compute unknown quantities at collocation points, the PINN approach optimizes the network parameters θ through a physics-informed loss function. The total loss function $\mathcal{L}(\theta)$ guiding the optimization is constructed as a weighted combination of boundary condition enforcement and equilibrium equation residuals:

$$\mathcal{L}(\theta) = \alpha \mathcal{L}_{\text{BC}}(\theta) + \beta \mathcal{L}_{\text{ODE}}(\theta), \quad (6.9)$$

with α and β serving as scalar factors to adjust the contribution of each component. The boundary loss term $\mathcal{L}_{\text{BC}}(\theta)$ ensures compliance with the prescribed boundary condition for the internal moment at the rod's free end $X = l$ (6.6). Specifically, the predicted moment $c_\theta(l)$ is constrained to match the given boundary moment c_+ :

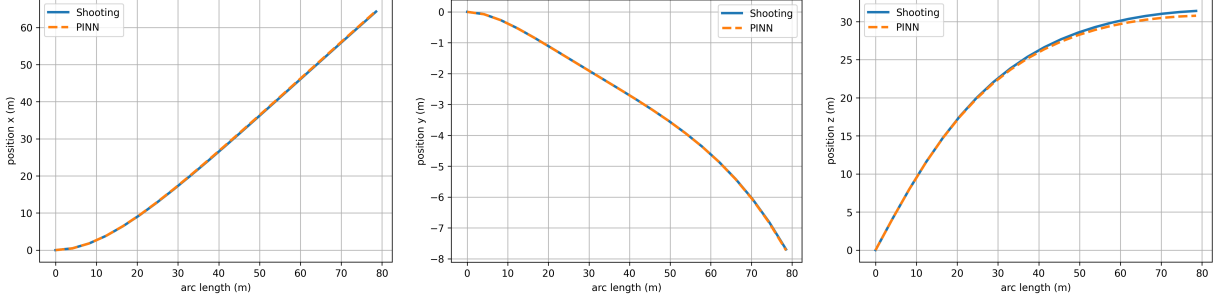


Figure 6.1: Comparison of the position components x , y , and z along the arc length between the shooting method (reference solution) and the PINN approximation. The PINN closely reproduces the rod configuration, showing good agreement with the reference solution across all position components.

$$\mathcal{L}_{\text{BC}}(\theta) = |c_\theta(l) - c_+|^2. \quad (6.10)$$

The physics-informed term $\mathcal{L}_{\text{ODE}}(\theta)$ enforces the equilibrium equations at each collocation point s_i , excluding the boundary point at the free end. At these points, the spatial derivative $c'_\theta(s_i)$, computed through automatic differentiation, is compared against the derivative prescribed by the rod's moment equilibrium equation. The automatic differentiation is enabled by the network using the arc length X as its input. The resulting discrepancy constitutes the residual of the ordinary differential equation, contributing to the loss function:

$$\mathcal{L}_{\text{ODE}}(\theta) = \frac{1}{m-1} \sum_{i=1}^{m-1} |c'_\theta(s_i) - c'(s_i)|^2, \quad (6.11)$$

where the analytical derivative $c'(s_i)$ comes from the moment equilibrium (6.5):

$$c' = -r' \times n - \bar{c}.$$

To evaluate this equilibrium derivative, the internal force n and rotation matrix R must be computed. The internal force n is determined through backward integration from the boundary condition at the rod's free end $X = l$, namely $n(l) = n_+$. The rotation matrix R is obtained by forward integration starting from the clamped base $X = 0$ with $R(0) = 1_{3 \times 3}$. The angular space rate K required for this integration is reconstructed from the constitutive relation (6.7), using the neural network's prediction c_θ .

The optimal network parameters θ^* are found by minimizing the total loss function using gradient-based optimization:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta), \quad (6.12)$$

with gradients $\nabla_{\theta} \mathcal{L}(\theta)$ computed via automatic differentiation techniques.

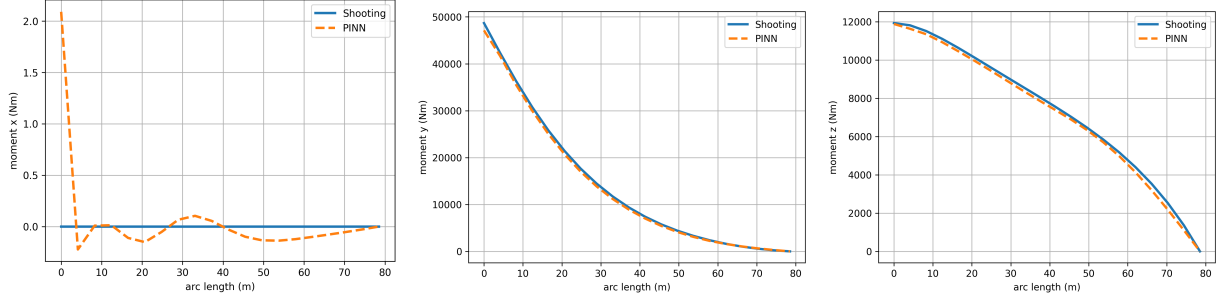


Figure 6.2: Comparison of the internal moment components x , y , and z along the arc length between the shooting method (reference solution) and the PINN approximation. For the y and z components, the PINN captures the overall moment with good accuracy. In the x -component, the larger relative discrepancies observed are due to the much smaller magnitude of the moment x compared to y and z , making the error appear more significant.

Numerical Application

This numerical example considers large displacements and coupled deformations, representing a benchmark problem extensively investigated in the literature [93, 79, 95]. The setup involves a curved beam, shaped as one-eighth of a circle with a radius of 100 m, lying in the xz -plane. An out-of-plane tip force of magnitude $|n_+| = 1550$ N is applied at the beam's free end along the y -axis. The beam features a square cross-section with sides of unit length (1 m). Material properties are defined by a Young's modulus of $E = 1 \times 10^7$ Pa, a shear modulus of $G = E/2$, and a cross-sectional area of $A = 1$ m².

The neural network employed consists of a single hidden layer containing 200,000 neurons, using the hyperbolic tangent activation function. The network is implemented in `PyTorch` and trained using the Adam optimizer with an initial learning rate of 0.01, decayed progressively by an exponential learning rate scheduler with a decay factor $\gamma = 0.999$. Training is performed over 3000 epochs using full-batch gradient descent, with all 20 collocation points uniformly distributed along the rod's arc length. The total loss function balances the contributions from the boundary condition and the equilibrium equation with weighting factors of $\alpha = 1$ and $\beta = 100$, respectively.

To evaluate the performance of the proposed PINN method, its predictions are compared against the shooting method, which is considered here as the reference solution. The shooting method solves the boundary value problem in 0.251 seconds, whereas training the PINN requires approximately 220 seconds. Fig. 6.1 and Fig. 6.2 present the comparison of the position and moment components, respectively, along the rod's arc length. As shown in Fig. 6.1, the PINN reproduces the configuration of the rod with reasonable agreement across all position components, with minor discrepancies observed. In contrast, the moment components (Fig. 6.2) exhibit more noticeable differences. While the PINN captures the general trends for the y and z components of the internal moment, larger relative discrepancies are observed in the x -component. This behavior is attributed to the much smaller magnitude of the moment x compared to y and z , which amplifies the relative error despite small absolute differences. The convergence behavior of the PINN during training is illustrated in Fig. 6.3, where the total loss function decreases over 3000 epochs,

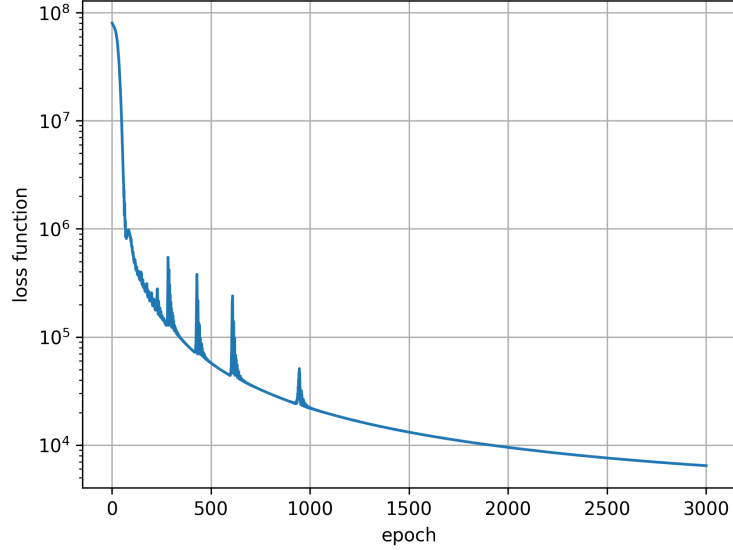


Figure 6.3: Evolution of the total loss function during PINN training over 3000 epochs, shown on a logarithmic scale. The steady decrease confirms convergence, with the observed oscillations resulting from the learning rate schedule and optimizer behavior.

with oscillations resulting from the learning rate decay and optimizer updates. These results indicate that the PINN can approximate the solution of the Cosserat rod boundary value problem with a good level of accuracy, though at a higher computational cost relative to the shooting method.

Discussion

The application of PINNs to the static boundary value problem of Cosserat rods demonstrates that this approach can approximate rod configurations with an acceptable level of accuracy when compared to established numerical methods, such as the shooting method. The continuous, mesh-free representation of the solution, combined with the inherent parallelism of neural networks, provides a flexible framework capable of approximating the rod behavior. Once trained, the network offers fast inference times, which could be advantageous in scenarios where repeated evaluations of a learned solution are required.

However, several limitations were identified in this implementation. The training phase, in particular, remains computationally expensive relative to classical solvers. For the benchmark problem considered, the PINN required approximately three minutes of training time, considerably longer than the shooting method's execution. Furthermore, the convergence of the PINN solution is sensitive to various hyperparameters, including the learning rate schedule, loss term weights, and the distribution of collocation points. Adjusting these parameters remains a largely empirical process, and achieving consistent convergence across different problem settings is not guaranteed. Additionally, the trained network is specific to a single scenario and changes in boundary conditions or distributed forces necessitate retraining.

The current formulation also requires numerical integration of the rotational kinematics to evaluate residuals in the loss function. This integration reflects the flexibility of the approach, allowing neural networks to be combined with classical numerical methods to ensure that physical constraints are properly enforced. However, this reliance also partially offsets the mesh-free nature of the PINN. Additionally, the need for numerical integration contributes to the overall computational cost, significantly increasing the training time. Furthermore, in this study, a wide but shallow network provided better performance than deeper alternatives. The underlying reasons for this behavior are not yet fully understood, indicating the need for further investigation into the relationship between network depth and convergence.

These observations suggest several directions for future research. Reducing or removing the reliance on numerical integration, potentially by directly predicting orientations while ensuring consistency with the $SO(3)$ group, could simplify the framework and improve training efficiency. Exploring systematic strategies for hyperparameter selection, including adaptive loss weighting and automated tuning methods, may improve the robustness and generality of the approach. Finally, extending the framework to dynamic problems will further assess the applicability of PINNs to more complex physical systems.

Bibliography

- [1] Hariadhi, “Blood clot.” https://en.wikipedia.org/wiki/File:Blood_clot_svg_hariadhi.svg, July 2024. Licensed under CC BY-SA 4.0. Uploaded to Wikipedia on 22 July 2024. ix, 2
- [2] P. Papanagiotou and G. Ntaios, “Endovascular thrombectomy in acute ischemic stroke,” *Circulation: Cardiovascular Interventions*, vol. 11, no. 1, p. e005362, 2018. ix, 3
- [3] J. Kaesmacher, C. Maegerlein, M. Kaesmacher, C. Zimmer, H. Poppert, B. Friedrich, T. Boeckh-Behrens, and J. F. Kleine, “Thrombus migration in the middle cerebral artery: Incidence, imaging signs, and impact on success of endovascular thrombectomy,” *Journal of the American Heart Association*, vol. 6, no. 2, p. e005149, 2017. ix, 4
- [4] 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. x, 18, 45
- [5] V. L. Feigin, M. Brainin, B. Norrving, S. Martins, R. L. Sacco, W. Hacke, M. Fisher, J. Pandian, and P. Lindsay, “World stroke organization (wso): global stroke fact sheet 2022,” *International Journal of Stroke*, vol. 17, no. 1, pp. 18–29, 2022. 1, 109
- [6] C. Duriez, *Real-time haptic simulation of medical procedures involving deformations and device-tissue interactions*. Accreditation to supervise research, Université des Sciences et Technologie de Lille - Lille I, Feb. 2013. 3, 109
- [7] P. Rougé, N. Passat, and O. Merveille, “Topology aware multitask cascaded u-net for cerebrovascular segmentation,” *PLOS ONE*, vol. 19, pp. 1–20, 12 2024. 4, 110
- [8] P. Rougé, P.-H. Conze, N. Passat, and O. Merveille, “Guidelines for cerebrovascular segmentation: Managing imperfect annotations in the context of semi-supervised learning,” *Computerized Medical Imaging and Graphics*, vol. 119, p. 102474, 2025. 4, 110
- [9] M. Khoshnam, M. Azizian, and R. V. Patel, “Modeling of a steerable catheter based on beam theory,” in *2012 IEEE International Conference on Robotics and Automation*, pp. 4681–4686, 2012. 7
- [10] Zhangchaonan, S. Guo, B. Gao, and F. Jia, “The beam theory-based collision force detection of the catheter operating system,” in *2016 IEEE International Conference on Mechatronics and Automation*, pp. 701–706, 2016. 7
- [11] T. Couture and J. Szewczyk, “Design and experimental validation of an active catheter for endovascular navigation,” *Journal of Medical Devices*, vol. 12, p. 011003, 11 2017. 7
- [12] A. Derouet-Jourdan, F. Bertails-Descoubes, G. Daviet, and J. Thollot, “Inverse dynamic hair modeling with frictional contact,” *ACM Trans. Graph.*, vol. 32, Nov. 2013. 7

- [13] G. Daviet, “Interactive hair simulation on the gpu using admm,” in *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH ’23, (New York, NY, USA), Association for Computing Machinery, 2023. [7](#)
- [14] J. Spillmann and M. Teschner, “Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects,” in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’07, (Goslar, DEU), pp. 63–72, Eurographics Association, 2007. [7](#)
- [15] Y. Sun and J. W. Leonard, “Dynamics of ocean cables with local low-tension regions,” *Ocean Engineering*, vol. 25, no. 6, pp. 443–463, 1998. [7](#)
- [16] B. Buckham, M. Nahon, M. Seto, X. Zhao, and C. Lambert, “Dynamics and control of a towed underwater vehicle system, part i: model development,” *Ocean Engineering*, vol. 30, no. 4, pp. 453–470, 2003. [7](#)
- [17] S. Chucheepsakul and N. Srinil, “Free vibrations of three-dimensional extensible marine cables with specified top tension via a variational method,” *Ocean Engineering*, vol. 29, no. 9, pp. 1067–1096, 2002. [7](#)
- [18] J. Ma, Z. Han, L. Yang, G. Min, Z. Liu, and W. He, “Dynamics modeling of a soft arm under the cosserat theory,” in *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 87–90, 2021. [7](#)
- [19] S. Grazioso, G. Di 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. PMID: 30481112. [7](#)
- [20] A. A. Alqumsan, S. Khoo, and M. Norton, “Robust control of continuum robots using cosserat rod theory,” *Mechanism and Machine Theory*, vol. 131, pp. 48–61, 2019. [7](#)
- [21] K. Nachbagauer, P. Gruber, and J. Gerstmayr, “Structural and continuum mechanics approaches for a 3d shear deformable ancf beam finite element: Application to static and linearized dynamic examples,” *Journal of Computational and Nonlinear Dynamics*, vol. 8, p. 021004, 07 2012. [8](#)
- [22] K. Yoon, Y. Lee, and P.-S. Lee, “A continuum mechanics based 3-d beam finite element with warping displacements and its modeling capabilities,” *Structural Engineering and Mechanics*, vol. 43, no. 4, pp. 411–437, 2012. [8](#)
- [23] G. Wei, P. Lardeur, and F. Druesne, “A new solid-beam approach based on first or higher-order beam theories for finite element analysis of thin to thick structures,” *Finite Elements in Analysis and Design*, vol. 200, p. 103655, 2022. [8](#)
- [24] J. Frischkorn and S. Reese, “A solid-beam finite element and non-linear constitutive modelling,” *Computer Methods in Applied Mechanics and Engineering*, vol. 265, pp. 195–212, 2013. [8](#)

- [25] F. Sadeghi, R. Ansari, and A. Basti, “3d finite element analysis of micromorphic hyperelastic structures considering finite deformations: Two-point formulation,” *International Journal of Non-Linear Mechanics*, vol. 174, p. 105054, 2025. [8](#)
- [26] K. Yoon and P.-S. Lee, “Nonlinear performance of continuum mechanics based beam elements focusing on large twisting behaviors,” *Computer Methods in Applied Mechanics and Engineering*, vol. 281, pp. 106–130, 2014. [8](#)
- [27] Y. Adagolodjo, F. Renda, and C. Duriez, “Coupling numerical deformable models in global and reduced coordinates for the simulation of the direct and the inverse kinematics of soft robots,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3910–3917, 2021. [8](#)
- [28] S. Minera, M. Patni, E. Carrera, M. Petrolo, P. Weaver, and A. Pirrera, “Three-dimensional stress analysis for beam-like structures using serendipity lagrange shape functions,” *International Journal of Solids and Structures*, vol. 141-142, pp. 279–296, 2018. [8](#), [111](#)
- [29] M. Lezgy-Nazargah, P. Vidal, and O. Polit, “A quasi-3d finite element model for the analysis of thin-walled beams under axial-flexural-torsional loads,” *Thin-Walled Structures*, vol. 164, p. 107811, 2021. [8](#)
- [30] C. M. Wang, “Timoshenko beam-bending solutions in terms of euler-bernoulli solutions,” *Journal of Engineering Mechanics*, vol. 121, no. 6, pp. 763–765, 1995. [9](#)
- [31] A. M. Ahmed and A. M. Rifai, “Euler-bernoulli and timoshenko beam theories analytical and numerical comprehensive revision,” *European Journal of Engineering and Technology Research*, vol. 6, no. 7, pp. 20–32, 2021. [9](#)
- [32] O. A. Bauchau and J. I. Craig, *Euler-Bernoulli beam theory*, pp. 173–221. Dordrecht: Springer Netherlands, 2009. [9](#)
- [33] J. Hutchinson, “Shear coefficients for timoshenko beam theory,” *J. Appl. Mech.*, vol. 68, no. 1, pp. 87–92, 2001. [9](#)
- [34] S. S. Antman, *Problems in nonlinear elasticity*. Springer, 2005. [9](#), [17](#), [59](#)
- [35] J. Linn, “Discrete cosserat rod kinematics constructed on the basis of the difference geometry of framed curves—part i: discrete cosserat curves on a staggered grid,” *Journal of Elasticity*, vol. 139, no. 2, pp. 177–236, 2020. [9](#), [91](#), [119](#)
- [36] D. C. Rucker, B. A. Jones, and R. J. Webster III, “A geometrically exact model for externally loaded concentric-tube continuum robots,” *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 769–780, 2010. [9](#)
- [37] M. Tummers, V. Lebastard, F. Boyer, J. Troccaz, B. Rosa, and M. T. Chikhaoui, “Cosserat rod modeling of continuum robots from newtonian and lagrangian perspectives,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2360–2378, 2023. [9](#), [13](#), [16](#), [17](#), [26](#), [29](#), [69](#), [70](#), [72](#), [78](#)

- [38] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, “Soft robots modeling: A structured overview,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1728–1748, 2023. [10](#), [26](#)
- [39] R. J. Webster, J. M. Romano, and N. J. Cowan, “Mechanics of precurved-tube continuum robots,” *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 67–78, 2009. [10](#), [111](#)
- [40] A. Chawla, C. Frazelle, and I. Walker, “A comparison of constant curvature forward kinematics for multisection continuum manipulators,” in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pp. 217–223, 2018. [10](#)
- [41] S. Neppalli, M. A. Csencsits, B. A. Jones, and I. D. W. and, “Closed-form inverse kinematics for continuum manipulators,” *Advanced Robotics*, vol. 23, no. 15, pp. 2077–2091, 2009. [10](#)
- [42] A. Garriga-Casanovas and F. Rodriguez y Baena, “Kinematics of continuum robots with constant curvature bending and extension capabilities,” *Journal of Mechanisms and Robotics*, vol. 11, p. 011010, 11 2018. [10](#)
- [43] C. Della Santina, A. Bicchi, and D. Rus, “On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1001–1008, 2020. [10](#)
- [44] R. J. WebsterIII 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, 2010. [10](#)
- [45] A. Sayadi, R. Cecere, and A. Hooshier, “Finite arc method: Fast-solution extended piecewise constant curvature model of soft robots with large variable curvature deformations,” *Robotics Reports*, vol. 2, no. 1, pp. 49–64, 2024. [10](#)
- [46] 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, 2010. [10](#)
- [47] M. Khoshnam and R. V. Patel, “A pseudo-rigid-body 3r model for a steerable ablation catheter,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 4427–4432, 2013. [11](#)
- [48] V. K. Venkiteswaran, J. Sikorski, and S. Misra, “Shape and contact force estimation of continuum manipulators using pseudo rigid body models,” *Mechanism and Machine Theory*, vol. 139, pp. 34–45, 2019. [11](#)
- [49] T. Greigarn, T. Liu, and M. C. Çavuşoğlu, “Parameter optimization of pseudo-rigid-body models of mri-actuated catheters,” in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5112–5115, 2016. [11](#), [111](#)

- [50] D. Lin, N. Jiao, Z. Wang, and L. Liu, “Calibration of the pseudo-rigid-body model for a multi-section magnetic catheter robot,” in *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 526–530, 2022. [11](#)
- [51] F. Wang, L. Duratti, E. Samur, U. Spaelter, and H. Bleuler, “A computer-based real-time simulation of interventional radiology,” in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1742–1745, 2007. [11](#)
- [52] J. Jung, R. S. Penning, and M. R. Z. and, “A modeling approach for robotic catheters: effects of nonlinear internal device friction,” *Advanced Robotics*, vol. 28, no. 8, pp. 557–572, 2014. [11](#)
- [53] W. Zhang, J. Guo, S. Guo, and Q. Fu, “Catheter modeling and tip force analysis based on mass spring model,” in *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 774–779, 2022. [11](#)
- [54] T. Cui, A. Song, and J. Wu, “Simulation of a mass-spring model for global deformation,” *Frontiers of Electrical and Electronic Engineering in China*, vol. 4, pp. 78–82, 2009. [12](#)
- [55] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, “Learning closed loop kinematic controllers for continuum manipulators in unstructured environments,” *Soft Robotics*, vol. 4, no. 3, pp. 285–296, 2017. [12](#)
- [56] Z. Chen, F. Renda, A. Le Gall, L. Mocellin, M. Bernabei, T. Dangel, G. Ciuti, M. Cianchetti, and C. Stefanini, “Data-driven methods applied to soft robot modeling and control: A review,” *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 2241–2256, 2025. [12](#)
- [57] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, “Relational inductive biases, deep learning, and graph networks,” *arXiv preprint arXiv:1806.01261*, 2018. [12](#)
- [58] A. Tariverdi, V. K. Venkiteswaran, M. Richter, O. J. Elle, J. Tørresen, K. Mathiassen, S. Misra, and Ø. G. Martinsen, “A recurrent neural-network-based real-time dynamic model for soft continuum manipulators,” *Frontiers in Robotics and AI*, vol. 8, p. 631303, 2021. [12](#)
- [59] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. [13](#), [89](#), [111](#), [119](#)
- [60] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, “Dynamics of continuum and soft robots: A strain parameterization based approach,” *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 847–863, 2021. [13](#), [26](#), [38](#), [39](#), [40](#), [41](#), [69](#), [70](#), [72](#), [78](#), [112](#), [116](#)

- [61] 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. [16](#), [17](#), [23](#), [27](#), [29](#), [31](#), [46](#), [47](#), [91](#), [112](#)
- [62] M. Gazzola, L. H. Dudte, A. G. McCormick, and L. Mahadevan, “Forward and inverse problems in the mechanics of soft filaments,” *Royal Society Open Science*, vol. 5, no. 6, p. 171628, 2018. [16](#), [18](#), [19](#), [20](#), [27](#), [112](#)
- [63] J. Linn, H. Lang, and A. Tuganov, “Geometrically exact cosserat rods with kelvin-voigt type viscous damping,” *Mechanical Sciences*, vol. 4, no. 1, pp. 79–96, 2013. [17](#), [65](#)
- [64] O. Weeger, B. Narayanan, and M. L. Dunn, “Isogeometric collocation for nonlinear dynamic analysis of cosserat rods with frictional contact,” *Nonlinear Dynamics*, vol. 91, pp. 1213–1227, 2018. [17](#), [20](#), [45](#)
- [65] F. Boyer and F. Renda, “Poincaré’s equations for cosserat media: application to shells,” *Journal of Nonlinear Science*, vol. 27, no. 1, pp. 1–44, 2017. [17](#)
- [66] F. Boyer, V. Lebastard, F. Candelier, F. Renda, and M. Alamir, “Statics and dynamics of continuum robots based on cosserat rods and optimal control theories,” *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1544–1562, 2023. [17](#), [22](#), [23](#), [28](#), [29](#), [30](#), [38](#), [39](#), [40](#), [44](#), [46](#), [47](#), [50](#), [53](#), [112](#), [113](#), [114](#)
- [67] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, “Discrete elastic rods,” in *ACM SIGGRAPH 2008 Papers*, SIGGRAPH ’08, (New York, NY, USA), Association for Computing Machinery, 2008. [18](#), [112](#)
- [68] A. Choi, R. Jing, A. P. Sabelhaus, and M. K. Jawed, “Dis Mech: A discrete differential geometry-based physical simulator for soft robots and structures,” *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3483–3490, 2024. [18](#)
- [69] J. Spillmann and M. Teschner, “Cosserat nets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 2, pp. 325–338, 2009. [18](#)
- [70] D. Tong, A. Choi, J. Joo, and M. K. Jawed, “A fully implicit method for robust frictional contact handling in elastic rods,” *Extreme Mechanics Letters*, vol. 58, p. 101924, 2023. [18](#)
- [71] S. Tschisgale, L. Thiry, and J. Fröhlich, “A constraint-based collision model for cosserat rods,” *Archive of Applied Mechanics*, vol. 89, pp. 167–193, 2019. [18](#), [61](#), [62](#)
- [72] B. Mirtich and J. Canny, “Impulse-based simulation of rigid bodies,” *Association for Computing Machinery*, pp. 181–ff., 1995. [18](#)
- [73] E. Guendelman, R. Bridson, and R. Fedkiw, “Nonconvex rigid bodies with stacking,” *Association for Computing Machinery*, vol. 22, pp. 871–878, July 2003. [18](#)

- [74] J. Bender and A. Schmitt, “Constraint-based collision and contact handling using impulses,” in *Proceedings of the 19th international conference on computer animation and social agents*, pp. 3–11, 2006. [18](#)
- [75] 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, 2014. [19](#), [20](#)
- [76] M. Wiese, R. Berthold, M. Wangenheim, and A. Raatz, “Describing and analyzing mechanical contact for continuum robots using a shooting-based cosserat rod implementation,” *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1668–1675, 2024. [23](#), [49](#)
- [77] J. Till and D. C. Rucker, “Elastic rod dynamics: Validation of a real-time implicit approach,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3013–3019, 2017. [23](#)
- [78] A. L. Orekhov and N. Simaan, “Solving cosserat rod models via collocation and the magnus expansion,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8653–8660, 2020. [23](#), [24](#), [25](#), [27](#), [31](#), [112](#)
- [79] 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. Special Issue on Isogeometric Analysis: Progress and Challenges. [23](#), [25](#), [94](#), [112](#)
- [80] J. C. Mason and D. C. Handscomb, *Chebyshev polynomials*. Chapman and Hall/CRC, 2002. [24](#), [30](#), [32](#), [33](#), [47](#), [113](#)
- [81] C. Rucker, “Integrating rotations using nonunit quaternions,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2979–2986, 2018. [25](#)
- [82] O. Weeger, B. Narayanan, L. De Lorenzis, J. Kiendl, and M. Dunn, “An isogeometric collocation method for frictionless contact of cosserat rods,” *Computer Methods in Applied Mechanics and Engineering*, vol. 321, pp. 361–382, 2017. [25](#)
- [83] J. Simo, “On a fully three-dimensional finite-strain viscoelastic damage model: Formulation and computational aspects,” *Computer Methods in Applied Mechanics and Engineering*, vol. 60, no. 2, pp. 153–173, 1987. [26](#)
- [84] D. Zupan and M. Saje, “Finite-element formulation of geometrically exact three-dimensional beam theories based on interpolation of strain measures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 49, pp. 5209–5248, 2003. [26](#), [39](#)
- [85] 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. [26](#)

- [86] H. Li, L. Xun, and G. Zheng, “Piecewise linear strain cosserat model for soft slender manipulator,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2342–2359, 2023. [26](#)
- [87] L. Xun, G. Zheng, and A. Kruszewski, “Cosserat-rod-based dynamic modeling of soft slender robot interacting with environment,” *IEEE Transactions on Robotics*, vol. 40, pp. 2811–2830, 2024. [26](#), [62](#)
- [88] P. Wriggers and T. A. Laursen, *Computational contact mechanics*, vol. 2. Springer, 2006. [26](#)
- [89] A. T. Mathew, I. B. Hmida, C. Armanini, F. Boyer, and F. Renda, “Sorosim: A matlab toolbox for hybrid rigid-soft robots based on the geometric variable-strain approach,” *IEEE Robotics & Automation Magazine*, vol. 30, no. 3, pp. 106–122, 2023. [26](#), [27](#), [112](#)
- [90] A. T. Mathew, D. Feliu-Talegon, A. Y. Alkayas, F. Boyer, and F. Renda, “Reduced order modeling of hybrid soft-rigid robots using global, local, and state-dependent strain parameterization,” *The International Journal of Robotics Research*, vol. 44, no. 1, pp. 129–154, 2025. [26](#)
- [91] F. Renda, V. Cacucciolo, J. Dias, and L. Seneviratne, “Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5495–5502, 2016. [27](#)
- [92] R. Jilani, P.-F. Villard, and E. Kerrien, “An orthogonal collocation method for static and dynamic cosserat rods,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4328–4333, 2023. [30](#), [110](#)
- [93] J. Simo and L. Vu-Quoc, “A three-dimensional finite-strain rod model. part ii: Computational aspects,” *Computer Methods in Applied Mechanics and Engineering*, vol. 58, no. 1, pp. 79–116, 1986. [38](#), [40](#), [94](#)
- [94] F. Boyer and D. Primault, “Finite element of slender beams in finite transformations: a geometrically exact approach,” *International Journal for Numerical Methods in Engineering*, vol. 59, no. 5, pp. 669–702, 2004. [38](#), [40](#)
- [95] S. Florian and C. Damien, “Geometrically exact static 3d cosserat rods problem solved using a shooting method,” *International Journal of Non-Linear Mechanics*, vol. 119, p. 103330, 2020. [38](#), [39](#), [94](#)
- [96] A. Ibrahimbegovic, “On the choice of finite rotation parameters,” *Computer Methods in Applied Mechanics and Engineering*, vol. 149, no. 1, pp. 49–71, 1997. Containing papers presented at the Symposium on Advances in Computational Mechanics. [39](#)
- [97] J.-M. Battini and C. Pacoste, “Co-rotational beam elements with warping effects in instability problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 17, pp. 1755–1789, 2002. [39](#)

- [98] S. Ghosh and D. Roy, “A frame-invariant scheme for the geometrically exact beam using rotation vector parametrization,” *Computational Mechanics*, vol. 44, pp. 103–118, 2009. [39](#)
- [99] E. Zupan, M. Saje, and D. Zupan, “The quaternion-based three-dimensional beam theory,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 49, pp. 3944–3956, 2009. [39](#)
- [100] A. Gravouil and A. Combescure, “Multi-time-step explicit–implicit method for non-linear structural dynamics,” *International Journal for Numerical Methods in Engineering*, vol. 50, no. 1, pp. 199–225, 2001. [42](#)
- [101] E. Marino, J. Kiendl, and L. De Lorenzis, “Explicit isogeometric collocation for the dynamics of three-dimensional beams undergoing finite motions,” *Computer Methods in Applied Mechanics and Engineering*, vol. 343, pp. 530–549, 2019. [42](#), [45](#)
- [102] E. Marino, J. Kiendl, and L. De Lorenzis, “Isogeometric collocation for implicit dynamics of three-dimensional beams undergoing finite motions,” *Computer Methods in Applied Mechanics and Engineering*, vol. 356, pp. 548–570, 2019. [42](#)
- [103] R. Jilani, P.-F. Villard, and E. Kerrien, “Solving dynamic cosserat rods with frictional contact using the shooting method and implicit surfaces,” in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10483–10488, 2024. [49](#), [65](#), [67](#), [110](#), [116](#)
- [104] E. Kerrien, A. Yureidini, J. Dequidt, C. Duriez, R. Anxionnat, and S. Cotin, “Blood vessel modeling for interactive simulation of interventional neuroradiology procedures,” *Medical Image Analysis*, vol. 35, pp. 685–698, 2017. [53](#), [54](#), [55](#), [64](#), [83](#), [115](#)
- [105] J. F. Blinn, “A generalization of algebraic surface drawing,” *ACM transactions on graphics (TOG)*, vol. 1, no. 3, pp. 235–256, 1982. [54](#), [115](#)
- [106] S. Muraki, “Volumetric shape description of range data using “blobby model”,” in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’91, (New York, NY, USA), pp. 227–235, Association for Computing Machinery, 1991. [54](#)
- [107] A. Yureidini, *Robust blood vessel surface reconstruction for interactive simulations from patient data*. Theses, Université des Sciences et Technologie de Lille - Lille I, May 2014. Rédigée au 21 janvier 2014. [55](#)
- [108] S. Andrews, K. Erleben, and Z. Ferguson, “Contact and friction simulation for computer graphics,” in *ACM SIGGRAPH 2022 Courses*, SIGGRAPH ’22, (New York, NY, USA), Association for Computing Machinery, 2022. [58](#)
- [109] Y. Sankin and N. Yuganova, “Longitudinal vibrations of elastic rods of stepwise-variable cross-section colliding with a rigid obstacle,” *Journal of Applied Mathematics and Mechanics*, vol. 65, no. 3, pp. 427–433, 2001. [61](#)

- [110] G. Taubin, “Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, 1991. [73](#), [117](#)
- [111] M. M. Dalvand, S. Nahavandi, and R. D. Howe, “Fast vision-based catheter 3d reconstruction,” *Physics in Medicine & Biology*, vol. 61, no. 14, p. 5128, 2016. [90](#), [119](#)
- [112] R. Jilani, P.-F. Villard, and E. Kerrien, “Quasi-static cosserat rods in contact with implicit surfaces,” *IEEE Robotics and Automation Letters*, vol. 10, no. 7, pp. 6536–6543, 2025. [110](#)

Appendices

A Résumé étendu en français

A.1 Introduction

L'accident vasculaire cérébral (AVC) ischémique, est l'une des principales causes de décès et d'invalidité à long terme dans le monde [5]. Il résulte de l'occlusion d'un vaisseau sanguin cérébral par un thrombus, ou caillot sanguin, qui restreint de manière critique l'apport d'oxygène et nutriments aux tissus cérébraux. Bien que les médicaments thrombolytiques constituent une première ligne de défense, leur efficacité est limitée, ce qui nécessite une approche interventionnelle dans de nombreux cas. La thrombectomie endovasculaire s'est imposée comme une procédure recommandée où un cathéter, un tube long, fin et flexible, est navigué à travers le système vasculaire du patient, généralement depuis l'artère fémorale, jusqu'au site du caillot. Une fois sur place, le thrombus peut être extrait, soit par aspiration, soit grâce à des dispositifs médicaux tels qu'un "stent retriever", rétablissant ainsi le flux sanguin.

Le principal défi de cette procédure réside dans la complexité de la navigation du cathéter. Le praticien doit guider le cathéter à travers un réseau de vaisseaux sanguins complexe et tortueux, dont l'anatomie varie selon les patients. Le geste est donc effectué sous le rétro-contrôle visuel d'images d'angiographie, révélant le réseau vasculaire, et d'images fluoroscopiques permettant de suivre en temps réel le cathéter et autres dispositifs thérapeutiques. Ces images reposent cependant sur les rayons X, ce qui limite le temps de traitement (rayonnement cumulé) et rend parfois difficile leur interprétation (images en projection). Ainsi, une navigation de cathéter efficace repose sur l'expérience du praticien et sur son interprétation en temps réel de l'imagerie médicale. Pour relever ce défi, cette thèse vise à développer une simulation numérique de la navigation de cathéter pour la planification préopératoire et la formation. Un tel outil pourrait aider à sélectionner les cathéters les plus adaptés à la tortuosité du trajet et à planifier les trajectoires les plus sûres et les plus efficaces pour atteindre l'occlusion.

Les objectifs principaux sont de : (1) développer une résolution numérique robuste pour un modèle cathéter passif et non actionné ; (2) prendre en compte les interactions de contact avec et sans frottement entre le cathéter et les parois des vaisseaux ; et (3) modéliser les actions du médecin en simulant le déplacement et la rotation de l'extrémité proximale du cathéter. Nous supposons que les vaisseaux sanguins sont rigides et fixes, et nous négligeons les forces hydrodynamiques du flux sanguin, en nous concentrant exclusivement sur la mécanique du cathéter.

Ces objectifs sont poursuivis sous le principe directeur de la création d'une simulation prédictive, définie par trois piliers : la précision, la performance en temps réel et la spécificité au patient [6]. La précision, traitée dans cette thèse comme une contrainte forte, englobe la fidélité physique dans la modélisation du comportement du cathéter, la précision numérique dans la résolution des équations directrices et la robustesse. La performance en temps réel, considérée comme une contrainte souple, correspond à des vitesses de calcul permettant à la simulation de s'exécuter au moins aussi vite que le système physique qu'elle modélise. L'adaptation au patient, traitée également comme une contrainte souple, nécessite que la simulation puisse être configurée avec les données anatomiques individuelles.

Ce travail s’inscrit dans le cadre du projet ANR PreSPIN (ANR-20-CE45-0011), un effort collaboratif visant à construire une chaîne de traitement prédictive pour la planification des interventions pour l’AVC ischémique. Cette chaîne intègre trois étapes principales : premièrement, la génération de modèles vasculaires spécifiques au patient à partir de l’angiographie par résonance magnétique [7, 8] ; deuxièmement, la simulation de la navigation du cathéter, qui est le sujet central de cette thèse ; et troisièmement, des simulations de dynamique des fluides computationnelle pour prédire l’hémodynamique post-intervention. Le projet implique un consortium multidisciplinaire de laboratoires de recherche et de partenaires cliniques, garantissant que les outils développés soient à la fois innovants et cliniquement pertinents.

Dans le chapitre 2, pour garantir la fidélité physique, nous adoptons le modèle géométriquement exact de poutre de Cosserat pour modéliser le cathéter, car il offre un bon équilibre entre précision et complexité de modélisation. Les trois chapitres suivants exposent les contributions principales de cette thèse.

Premièrement, dans le chapitre 3, nous abordons les objectifs de précision numérique et de robustesse. Nous formulons le modèle de poutre de Cosserat dynamique comme un problème aux limites (PAL) spatial et introduisons une méthode de collocation orthogonale pour sa résolution. Une découverte clé est que cette approche numérique surmonte avec succès la singularité fondamentale qui apparaît dans les simulations dynamiques de poutres souples, en particulier avec de petits pas de temps, assurant ainsi une solution robuste. Une partie de ce travail a été publiée à IEEE IROS 2023 [92].

Deuxièmement, dans le chapitre 4, nous nous concentrons sur l’adaptation spécifique au patient et le réalisme clinique. La méthode de collocation est étendue pour inclure les manipulations de l’extrémité proximale qui reproduisent les gestes du médecin, ainsi que le contact avec frottement avec l’environnement. Pour modéliser efficacement la géométrie complexe des vaisseaux sanguins, nous les représentons par des surfaces implicites, une technique qui accélère considérablement la détection de contact. L’applicabilité à la navigation de cathéter est démontrée en utilisant un modèle d’artère carotide extrait de données réelles de patient. Cette contribution a été publiée à IEEE IROS 2024 [103].

Troisièmement, dans le chapitre 5, nous nous attaquons à l’objectif de performance en temps réel. Nous passons d’un régime dynamique à un régime quasi-statique et employons une méthode de réduction lagrangienne. Pour résoudre les équations raides qui en résultent, nous utilisons des schémas d’intégration temporelle implicites. L’une des contributions est la dérivation analytique de la matrice jacobienne du système, rendue possible par la différentiabilité du modèle de surface implicite des parois vasculaires, ce qui conduit à des gains de performance significatifs. Cette contribution a été publiée dans la revue IEEE RA-L [112].

La thèse se termine par le chapitre 6, qui résume nos résultats et discute des perspectives futures. Cette recherche s’engage en faveur de la science ouverte ; toutes les contributions clés sont accompagnées d’un code source accessible au public pour promouvoir la transparence et la reproductibilité.

A.2 État de l’art de la modélisation des poutres

Ce chapitre propose une revue des techniques de modélisation des poutres afin d’identifier le cadre le plus approprié pour la simulation de la navigation de cathéter, puis examine les méthodes numériques de l’état de l’art pour la résolution du modèle choisi. L’objectif est aussi d’établir un fondement théorique qui justifie les contributions présentées dans les chapitres suivants de cette thèse. Dans un premier temps, le problème de la navigation de cathéter dans les vaisseaux sanguins est généralisé à la simulation d’une poutre élastique manipulée à son extrémité proximale tout en subissant un contact avec frottement à l’intérieur de surfaces tubulaires. Dans ce cadre, la poutre représente le cathéter, tandis que les surfaces tubulaires modélisent les parois des vaisseaux sanguins.

L’examen des modèles de poutres couvre un éventail d’approches, chacune présentant des compromis distincts entre précision et coût de calcul. D’un côté, la mécanique des milieux continus tridimensionnelle offre la plus grande fidélité en modélisant la poutre comme un volume solide, capable de capturer tous les modes de déformation sans simplification géométrique. Cependant, sa discrétisation par la méthode des éléments finis aboutit à un système dont la résolution exige des calculs intensifs, le rendant inadapté pour les performances en temps réel requises dans les contextes de simulation clinique [28]. De l’autre côté, des modèles simplifiés tels que le modèle à courbure constante [39], le modèle de corps pseudo-rigide, et le modèle masse-ressort offrent une efficacité de calcul au détriment de la précision physique. Ces modèles font soit des hypothèses géométriques restrictives, comme une courbure constante par morceaux, soit reposent sur des paramètres discrétisés difficiles à relier directement aux propriétés matérielles d’un cathéter réel [49]. Les méthodes basées sur les données, souvent fondées sur des réseaux de neurones, proposent une alternative en apprenant le comportement de la poutre à partir de données, mais leur pouvoir prédictif est limité par la portée des données d’entraînement et elles peuvent échouer à généraliser ou à respecter les lois de conservation physiques [59].

Entre ces extrêmes se trouvent les modèles de la mécanique des milieux continus unidimensionnels, qui représentent la poutre comme une courbe dans l’espace. Bien que les théories linéaires classiques comme celles d’Euler-Bernoulli et de Timoshenko soient efficaces en calcul, leur applicabilité est limitée aux petites déformations et rotations, une condition non satisfaite lors de la navigation de cathéter. Par conséquent, la théorie des poutres de Cosserat est identifiée comme le choix le plus approprié. Elle conduit à des modèles qui offrent un compromis équilibré, capturant de grandes déformations non linéaires, y compris l’extension, le cisaillement, la flexion et la torsion, tout en étant plus efficaces en calcul que les modèles tridimensionnels complets. Le modèle de Cosserat représente la poutre comme une courbe continue munie d’un repère mobile, paramétrée par l’abscisse curviligne X , où la configuration en chaque point est décrite par une pose $g(X) \in SE(3)$, qui encode à la fois la position et l’orientation de la section transversale.

La cinématique de la poutre de Cosserat est décrite à l’aide d’une formulation sur les groupes de Lie, où le taux de variation spatial et la vitesse sont capturés respectivement par les torseurs de déformation $\xi = (g^{-1}g')^\vee$ et de vitesse $\eta = (g^{-1}\dot{g})^\vee$, où g' et $\dot{g}$ sont respectivement les dérivées spatiale et temporelle. En supposant une élasticité linéaire, la loi de comportement relie le torseur des contraintes Λ à la déformation $\epsilon = \xi - \xi_o$ via la matrice de raideur de Hooke $\mathcal{H}$, de sorte que $\Lambda = \mathcal{H}\epsilon$. L’indice $\bullet_o$ désigne une valeur au repos. Le

comportement du système est régi par des équations d'équilibre dérivées des principes de Newton-Euler. L'équilibre statique est décrit par l'équation différentielle :

$$\Lambda' - ad_{\xi}^T \Lambda + \bar{F} = 0$$

où Λ' est la dérivée spatiale du torseur des contraintes Λ , ad_{ξ}^T est l'opérateur co-adjoint associé au torseur des déformations ξ , et $\bar{F}$ est le torseur des forces externes distribuées. Le cas dynamique étend ceci en incorporant les termes inertiels :

$$\mathcal{M}\dot{\eta} - ad_{\eta}^T \mathcal{M}\eta = \Lambda' - ad_{\xi}^T \Lambda + \bar{F}$$

où $\mathcal{M}$ est la matrice d'inertie et η est le torseur des accélérations.

Le modèle de Cosserat étant sélectionné, le chapitre procède à l'examen de ses méthodes de résolution numérique de l'état de l'art, les classant en trois grandes familles. Premièrement, les méthodes des différences finies, telles que le modèle de poutre élastique discrète [67, 62], discrétisent la poutre et approximent les dérivées spatiales de manière algébrique. Bien que rapides en calcul par pas de temps, ces méthodes peuvent souffrir d'instabilités numériques, nécessitant de petits pas de temps couplés à la taille de la discrétisation spatiale, ce qui peut nuire aux performances globales.

Deuxièmement, les approches par PAL formulent les équations de la poutre dans le domaine spatial, en considérant des conditions aux limites telles qu'encasté-libre ou libre-libre. La méthode de tir résout ce problème en calculant itérativement les conditions initiales pour satisfaire les contraintes terminales. Bien qu'efficace dans certains cas statiques [61], son application au problème dynamique est fondamentalement instable pour les matériaux souples ou les petits pas de temps en raison d'une singularité sous-jacente dans la formulation de la commande optimale [66]. Alternativement, les méthodes de collocation approximent la solution à l'aide de fonctions de base (par exemple, les polynômes de Tchebychev [78] ou les B-splines [79]) et se sont révélées prometteuses pour les problèmes statiques [78], mais leur application au PAL dynamique singulier reste une question de recherche ouverte.

Troisièmement, un dernier groupe de méthodes, en particulier celles employant une formulation lagrangienne, réduit le modèle de Cosserat continu à un ensemble fini d'équations différentielles ordinaires. Des approches comme la méthode des éléments finis géométriquement exacte ou les techniques de réduction modale utilisant des fonctions de base comme les polynômes de Legendre se sont avérées précises et efficaces [60, 89]. Cependant, les équations résultantes sont parfois raides numériquement, en particulier pour les matériaux à fort module de Young, nécessitant de petits pas de temps qui peuvent compromettre la capacité à réaliser les calculs temps réel.

En conclusion, cette revue révèle qu'aucune méthode existante pour résoudre les poutres de Cosserat n'atteint la combinaison souhaitée de précision, de robustesse et de vitesse temps réel nécessaire à la simulation prédictive de cathéter.

A.3 Résolution de la singularité dans le problème aux limites des poutres de Cosserat dynamiques

Ce chapitre aborde un défi fondamental dans la simulation des poutres de Cosserat dynamiques : la singularité inhérente du PAL qui les régit. Comme établi par Boyer et al. [66], bien que l'intégration temporelle implicite puisse régulariser ce PAL, la singularité réapparaît lors de la simulation de poutres souples ou lors de l'utilisation de pas de temps inférieurs à une valeur critique, Δt_c . Cette limitation restreint l'applicabilité des solveurs de PAL standards, tels que la méthode de tir, en particulier dans le contexte de la navigation de cathéter, qui implique des instruments flexibles et nécessite de petits pas de temps pour atteindre un niveau acceptable de précision. L'objectif principal de ce chapitre est de développer et de valider une méthode numérique capable de surmonter cette singularité, permettant ainsi des simulations dynamiques robustes et précises sur un ensemble plus large de propriétés matérielles et de résolutions temporelles.

À cette fin, nous proposons une approche basée sur une méthode de collocation orthogonale, en substitut à la méthode de tir, couramment utilisée mais instable. L'étape initiale et centrale de notre méthode est une réduction de l'espace de configuration où le champ de contraintes internes de la poutre, $\Lambda(X)$, est approximé à l'aide d'une base de polynômes de Tchebychev, connus pour leurs propriétés de stabilité numérique [80]. Le champ de contraintes continu est ainsi représenté par un ensemble fini de paramètres inconnus, spécifiquement les valeurs de contrainte $\Lambda_m \in \mathbb{R}^{6 \times m}$ à m nœuds de collocation discrets, qui correspondent aux racines des polynômes de Tchebychev. La contrainte en tout point le long de la poutre est alors interpolée par $\Lambda(X) \simeq \Lambda_m \Phi(X)$, où $\Phi(X)$ est un vecteur des fonctions de base.

Le problème est formulé comme un système d'équations algébriques non linéaires, $\mathcal{R}_c(\Lambda_m) = 0$, à résoudre pour les valeurs de contrainte inconnues Λ_m . Le résidu, $\mathcal{R}_c$, est construit en imposant l'équation d'équilibre dynamique à chacun des $m - 1$ premiers points de collocation, garantissant que la dérivée spatiale de la contrainte calculée à partir de la dynamique de la poutre corresponde à la dérivée obtenue par l'approximation de Tchebychev. La condition aux limites distale, $\Lambda(l) = F_+$, est imposée au dernier point de collocation. Un élément essentiel de ce travail est la dérivation analytique de la matrice jacobienne complète du système résiduel. Cette jacobienne analytique est cruciale pour l'efficacité et la convergence du solveur itératif de recherche de racines, en particulier lorsque des polynômes de haut degré sont nécessaires. La procédure globale de calcul du résidu est détaillée dans l'Algorithme 1 (page 36).

La méthode de collocation proposée a été validée par une série de bancs d'essai numériques statiques et dynamiques. En régime statique, la méthode a démontré une bonne précision, avec des erreurs de position inférieures à 0.1% par rapport à une solution analytique pour la flexion circulaire et inférieures à 0.5% par rapport à la méthode de tir dans des scénarios plus complexes de flexion hélicoïdale et de force suiveuse. Ces tests ont également établi que le temps de calcul est comparable à celui de la méthode de tir pour des faibles degrés polynomiaux.

La contribution principale est démontrée en régime dynamique, où la méthode a été testée sur des scénarios de plus en plus difficiles conçus pour déclencher la singularité. Dans une simulation d'une poutre en porte-à-faux rigide, la méthode de tir n'a pas réussi à converger

lorsque le pas de temps a été réduit à $\Delta t = 0.5$ ms. La méthode de collocation a également échoué initialement, mais a convergé avec succès après avoir augmenté le degré polynomial à $m = 70$, produisant des résultats en accord avec un modèle lagrangien de référence. Cela a conduit à l’hypothèse centrale du chapitre : l’augmentation du degré polynomial m de la méthode de collocation abaisse le pas de temps critique Δt_c , surmontant ainsi la singularité numérique.

Cette hypothèse a été testée et confirmée sur des bancs d’essai plus exigeants, y compris le relâchement d’une poutre souple et courbée et un pendule en caoutchouc sous l’effet de la gravité. La méthode de tir a échoué dans tous ces scénarios. En revanche, la méthode de collocation a atteint la convergence de manière constante, à condition d’utiliser un degré polynomial suffisamment élevé ($m = 100$ et $m = 200$, respectivement), et a produit des résultats qui correspondaient étroitement au modèle lagrangien, avec des erreurs de position inférieures à 0.9%.

En conclusion, ce chapitre présente et valide avec succès une méthode de collocation orthogonale qui résout le problème de singularité dans le PAL dynamique des poutres de Cosserat. La découverte clé est que le problème, bien que mal conditionné pour les solveurs standards, peut être rendu bien conditionné en augmentant le nombre de fonctions de base dans le schéma de collocation. Cette contribution élargit le domaine d’applicabilité du PAL en robotique souple et continue. Cependant, cette robustesse a un coût de calcul ; les hauts degrés polynomiaux requis pour les problèmes difficiles entraînent des temps de simulation plus longs par rapport à des formulations alternatives comme le modèle lagrangien. Le chapitre suivant s’appuiera sur cette base robuste en y intégrant le contact avec frottement pour créer un cadre de simulation complet pour la navigation de cathéter.

A.4 Poutres de Cosserat dynamiques en contact avec des surfaces implicites

S’appuyant sur le solveur dynamique développé au chapitre précédent, ce chapitre aborde l’objectif principal de la simulation de la navigation de cathéter. La méthode de collocation est maintenant étendue pour incorporer la physique essentielle de l’interaction outil-tissu. Cela implique trois améliorations : premièrement, l’adaptation de la formulation du PAL pour modéliser des conditions aux limites libre-libre, nécessaires pour simuler la manipulation de l’extrémité proximale du cathéter par le praticien ; deuxièmement, la mise en œuvre d’une méthode efficace pour la gestion du contact avec frottement ; et troisièmement, la représentation de géométries vasculaires complexes permettant des calculs géométriques rapides, à très faible impact sur le temps de résolution.

La première contribution est l’extension de la méthode de collocation orthogonale pour résoudre le PAL libre-libre, ce qui est essentiel pour simuler la manipulation du cathéter par le praticien. Cela représente un défi, car le PAL libre-libre est sous-déterminé : il manque de conditions aux limites suffisantes pour résoudre l’état de la poutre, laissant la pose initiale $g(0)$, la vitesse $\eta(0)$ et l’accélération $\dot{\eta}(0)$ inconnues. Pour résoudre ce problème, nous adoptons la stratégie de discrétisation temporelle proposée par Boyer et al. [66]. Cette approche utilise un schéma d’intégration temporelle implicite, basé sur la méthode de Newmark, pour exprimer les conditions initiales inconnues en termes d’un ensemble plus

restreint de paramètres : un vecteur de position proximale r_0 et un vecteur de rotation Θ_0 . Ces paramètres sont ensuite intégrés dans le cadre de la collocation en tant qu'inconnues. Par conséquent, l'ensemble des variables résolues par la méthode itérative devient ces paramètres de pose proximale ainsi que les valeurs de contrainte aux nœuds de collocation intérieurs. La contrainte au premier nœud, $\Lambda(0)$, n'est plus une inconnue à résoudre, mais est prescrite par le torseur des forces externes proximales F_- . Pour maintenir l'efficacité du solveur itératif sous cette nouvelle formulation, une nouvelle dérivation de la matrice jacobienne est proposée en tenant compte du nouvel ensemble d'inconnues.

La deuxième contribution réside dans la représentation des surfaces de contact et la gestion des interactions. Pour permettre une détection de contact rapide au sein de géométries complexes comme les vaisseaux sanguins, nous modélisons les surfaces à l'aide de fonctions implicites basées sur le modèle de "blobs" [105, 104]. La surface est définie comme l'ensemble de niveau zéro d'un champ scalaire $\zeta(r)$, construit comme une somme de profils de Cauchy Ψ pondérés centrés sur un squelette de points. La forme finale est donnée par :

$$\zeta(r) = S - \sum_{j=1}^{m_b} \varrho_j \Psi \left(\frac{|r - \sigma_j|^2}{\varrho_j^2} \right)$$

où r est une position, S est un seuil, et pour chacun des m_b points du squelette σ_j , ϱ_j est un paramètre d'échelle lié au rayon local. Cette représentation est continûment différentiable, permettant le calcul efficace de la distance signée $\phi(r)$ et de la normale à la surface $w(r)$ en tout point. Les forces de contact sont calculées à l'aide d'une méthode de pénalisation, où la force normale est proportionnelle à la profondeur de pénétration, et un modèle de Coulomb est utilisé pour le frottement tangentiel.

Un défi central abordé est l'intégration de ces forces de contact concentrées dans le cadre du PAL, qui nécessite des forces distribuées. Nous proposons une méthode où la poutre est discrétisée en segments, et la force de contact concentrée calculée au centre de chaque segment est moyennée sur la longueur du segment pour produire une force distribuée constante par morceaux, $\bar{f}_c$. La boucle de simulation globale, détaillée dans l'Algorithme 2 (page 61), emploie un processus en deux étapes à chaque pas de temps : d'abord, le PAL est résolu dans un état non contraint pour prédire le mouvement de la poutre, ensuite les forces de contact sont calculées sur la base de cette prédiction, et enfin, le PAL est résolu à nouveau dans un état contraint intégrant ces forces.

La méthode a été validée par trois applications numériques. Premièrement, un scénario de contact longitudinal élastique a été utilisé pour vérifier la précision du solveur de PAL libre-libre par rapport à une solution analytique, démontrant une conservation correcte de l'énergie lors de l'impact et du rebond. Deuxièmement, un test impliquant l'insertion d'une poutre dans un tube hélicoïdal a validé avec succès le modèle de contact avec frottement, montrant que l'augmentation du coefficient de frottement entravait correctement le mouvement de la poutre.

Le test final a consisté à simuler une insertion de cathéter dans un modèle d'artère carotide, extrait de données d'imagerie clinique. Cette application a servi de démonstration et intègre tous les composants développés : le solveur dynamique libre-libre, la géométrie de surface implicite complexe et le modèle de contact avec frottement. La simulation a

reproduit la navigation du cathéter, en utilisant ses propriétés matérielles réelles, à travers le chemin vasculaire tortueux sans nécessiter l’amortissement artificiel ou l’assouplissement du matériau qui étaient nécessaires dans les travaux précédents utilisant la méthode de tir, moins stable [103].

En conclusion, ce chapitre présente une méthode robuste pour simuler la navigation dynamique de cathéter. La combinaison du solveur de collocation avec la modélisation de surface implicite et un schéma d’intégration des forces de contact permet la simulation de scénarios réalistes. Cependant, le coût de calcul élevé de la méthode, dû aux très hauts degrés polynomiaux (m jusqu’à 500) nécessaires pour assurer la stabilité dans des scénarios de contact complexes, exclut son utilisation dans des applications temps réel. Cette limitation motive l’étude du chapitre suivant, qui explore une formulation quasi-statique comme une voie potentielle pour atteindre les performances de calcul requises.

A.5 Poutres de Cosserat quasi-statiques en contact avec des surfaces implicites

Ce chapitre explore une approche quasi-statique visant à améliorer les performances de calcul sans sacrifier le comportement physique essentiel. L’hypothèse sous-jacente est que pour de nombreuses procédures de navigation de cathéter, où les vitesses d’insertion sont faibles et les déformations principalement dictées par le contact, les effets d’inertie peuvent être négligés.

Pour ce faire, nous passons de la formulation en PAL à un modèle lagrangien basé sur l’approche de paramétrisation en déformation de Boyer et al. [60]. Dans ce cadre, la configuration de la poutre est décrite par un ensemble de coordonnées de déformation généralisées q , où la déformation est donnée par $\epsilon = \Phi(X)q$. L’évolution quasi-statique de la poutre est modélisée en introduisant un amortissement artificiel, qui transforme l’équation d’équilibre statique en une équation différentielle ordinaire (EDO) du premier ordre régissant l’évolution des coordonnées généralisées sur un temps fictif t :

$$\dot{q} = -D_\epsilon^{-1}(Q_e + Q_r)$$

Ici, D_ϵ est une matrice d’amortissement généralisée, $Q_r = K_\epsilon q$ est le vecteur des forces de rappel généralisées, et Q_e représente les forces externes généralisées, qui incluent les forces provenant du contact. Bien que cette EDO puisse être résolue avec des intégrateurs explicites, nous avons identifié que les scénarios de contact complexes introduisent une raideur significative dans le système, forçant les méthodes explicites à prendre des pas de temps excessivement petits et annulant ainsi les avantages de performance de l’hypothèse quasi-statique.

La contribution centrale de ce chapitre est une méthode pour surmonter cette raideur. Nous proposons de résoudre l’EDO directrice à l’aide d’un schéma d’intégration temporelle implicite, qui est bien adapté aux problèmes raides. La viabilité de cette approche repose sur le calcul efficace de la matrice jacobienne du système, $\frac{\partial \dot{q}}{\partial q}$. Notre contribution clé est la dérivation analytique de cette jacobienne. Cela a été rendu possible en tirant parti de la nature différentiable des surfaces implicites utilisées pour modéliser la géométrie de contact, une méthodologie établie au chapitre précédent. Pour faciliter cette dérivation analytique, le modèle de contact a été affiné : la distance signée est calculée à l’aide de l’approximation

non itérative de Taubin [110], et la fonction d'activation non différentiable ReLU pour la force de pénalisation est remplacée par une approximation lisse et différentiable (la fonction SiLU).

La méthode proposée a été testée sur une série d'applications numériques conçues pour évaluer ses performances et son réalisme physique. Les premiers tests, tels qu'une poutre interagissant avec une sphère, ont été utilisés pour étudier l'effet du coefficient d'amortissement artificiel $\tilde{\mu}$. Ceux-ci ont confirmé que pour un rapport suffisamment faible entre l'amortissement et le temps total de simulation, la méthode produit des solutions quasi-statiques précises. La validation principale des performances a démontré que le solveur implicite avec la jacobienne analytique est plus efficace qu'un solveur explicite et qu'un solveur implicite avec une jacobienne numérique, réduisant le temps de calcul jusqu'à deux ordres de grandeur pour les problèmes raides numériquement.

La capacité de l'approche a été démontrée plus en détail dans des scénarios plus complexes. Une simulation d'insertion et de retrait d'une poutre dans un tube hélicoïdal a validé l'approche pour la gestion des conditions aux limites variables dans le temps et a de nouveau mis en évidence l'accélération de la méthode implicite par rapport à la méthode explicite (par exemple, 169 secondes contre 9 heures pour un test comparable). L'expérience principale a consisté à simuler une insertion de cathéter dans le même modèle d'artère carotide du chapitre 4. Ce test a confirmé l'applicabilité de la méthode à des problèmes cliniquement pertinents, capturant avec succès des comportements complexes comme un effet d'adhérence-glissement causé par de petites aspérités sur la surface du vaisseau. Plus important encore, cette approche quasi-statique a considérablement réduit le temps de calcul pour la simulation de l'artère carotide, passant de plus de 12 heures avec le modèle dynamique à 44 minutes.

En conclusion, ce chapitre présente une approche de simulation quasi-statique efficace en calcul pour la navigation de cathéter. En combinant un modèle lagrangien à réduction de déformation avec un solveur implicite et une jacobienne dérivée analytiquement grâce à des surfaces implicites différentiables, nous atténuons efficacement le problème de raideur numérique qui affecte les simulations de contact quasi-statiques. Bien que la méthode néglige le frottement et n'ait pas encore atteint des performances en temps réel, la réduction significative du temps de calcul, de plusieurs heures à quelques minutes, représente une étape essentielle vers le développement d'outils de planification prédictifs et interactifs pour les procédures endovasculaires.

A.6 Conclusion et perspectives

Cette thèse a abordé le développement de simulations numériques pour la navigation de cathéter, motivée par le besoin d'outils prédictifs dans les procédures cliniques telles que la thrombectomie endovasculaire. La recherche s'est concentrée sur le modèle de poutre de Cosserat géométriquement exact, qui capture avec précision les grandes déformations inhérentes aux instruments médicaux flexibles. Les contributions principales de ce travail résident dans la résolution des défis numériques importants associés à la résolution de ce modèle, notamment en ce qui concerne la stabilité, l'efficacité de calcul et la modélisation des interactions de contact complexes.

La première contribution majeure a été le développement d’un cadre de simulation dynamique nouveau et robuste. En employant une méthode de collocation orthogonale avec des polynômes de Tchebychev, nous avons résolu la singularité fondamentale qui affecte le PAL dynamique des poutres de Cosserat, une limitation qui avait auparavant entravé l’utilisation de solveurs standards comme la méthode de tir dans des scénarios impliquant des matériaux souples ou de petits pas de temps. La dérivation analytique de la matrice jacobienne du résidu a été déterminante pour atteindre cette stabilité. Au titre de la deuxième contribution, ce cadre a ensuite été étendu pour gérer les conditions aux limites libre-libre afin de simuler la manipulation par le praticien et a intégré des modèles de surface implicite pour une détection de contact efficace au sein de géométries vasculaires. Bien que cette approche dynamique ait démontré une bonne robustesse sans nécessiter d’amortissement numérique artificiel, son coût de calcul important l’a rendue inadaptée pour les applications où le temps est critique.

Pour relever le défi de la vitesse de calcul, la troisième contribution majeure a été le développement d’un cadre de simulation quasi-statique efficace en calcul. Cette approche, justifiée par la faible vitesse d’insertion du cathéter, s’appuie sur un modèle lagrangien amorti avec une paramétrisation en déformation pour réduire les degrés de liberté du système. La contribution essentielle a été de s’attaquer au problème de la raideur des équations, qui survient dans des scénarios de contact complexes, en utilisant un intégrateur temporel implicite. La faisabilité de cette méthode implicite a été rendue possible par la dérivation analytique de sa matrice jacobienne, une tâche facilitée par la différentiabilité des fonctions implicites utilisées pour modéliser les surfaces de contact. Cette méthode quasi-statique a permis une amélioration considérable des performances, réduisant les temps de calcul pour des simulations complexes de plusieurs heures à quelques minutes, et représentant une étape importante vers l’applicabilité clinique. Tout au long de ce travail, un engagement envers la science ouverte a été maintenu, toutes les implémentations étant rendues publiques pour assurer la reproductibilité.

Malgré ces avancées, ce travail présente plusieurs limitations qui définissent sa portée actuelle. Les simulations, bien que beaucoup plus rapides, n’atteignent pas encore les performances en temps réel nécessaires à une utilisation clinique interactive. Les modèles se concentrent exclusivement sur le cathéter, omettant d’autres composants essentiels d’une procédure endovasculaire comme les guides ou les stents retrievers. De plus, les simulations n’ont pas été validées par des données expérimentales ou cliniques, et elles font des hypothèses simplificatrices en traitant les vaisseaux sanguins comme rigides et en ignorant l’influence du flux sanguin, qui pourrait affecter la dynamique du cathéter par des interactions fluide-structure. Enfin, la mécanique du contact a été modélisée à l’aide d’une méthode de pénalisation qui, bien que pratique en calcul, est une simplification des interactions complexes et non linéaires à l’interface cathéter-vaisseau.

Ces limitations ouvrent plusieurs pistes de recherche prometteuses. Sur le plan calculatoire, les performances pourraient être améliorées par la parallélisation (par ex. accélération GPU) des calculs intensifs de la jacobienne. Sur le plan théorique, une investigation plus approfondie des raisons pour lesquelles l’augmentation du degré polynomial stabilise la méthode de collocation dynamique pourrait conduire à des solveurs plus efficaces. Deux directions futures ont été explorées dans des études préliminaires.

Une direction de travail future est d'améliorer la fidélité physique en estimant la véritable configuration de référence non rectiligne du cathéter, caractérisée par le torseur de déformation de référence ξ_o . Cette thèse ayant supposé une référence rectiligne, la modélisation précise de la courbure intrinsèque est une étape critique suivante. Une méthode hybride proposée aborde ce problème en capturant d'abord expérimentalement la ligne centrale 3D, $r(X)$, d'un cathéter suspendu sous l'effet de la gravité à l'aide de la stéréovision [111]. Cette forme reconstruite est ensuite utilisée pour résoudre numériquement la courbure de référence, K_o , en inversant la loi de comportement :

$$K_o = K - \mathcal{H}_a^{-1} R^T c$$

où la courbure actuelle K , l'orientation R , le moment interne c et la raideur $\mathcal{H}_a$ sont déterminés à partir de la ligne centrale reconstruite r . Le processus implique l'utilisation de la méthode du transport parallèle [35] pour trouver le repère d'orientation R , à partir duquel la courbure actuelle K est dérivée. Le champ de moment interne c est obtenu en intégrant les équations d'équilibre statique en partant de l'extrémité libre du cathéter.

Bien que cette approche, qui suppose actuellement une référence plane et le modèle de Kirchhoff, soit prometteuse sur des données synthétiques, son principal défi reste la limitation de la technique du transport parallèle pour déterminer avec précision la véritable orientation matérielle 3D. Les travaux futurs devront donc s'attaquer à cette limitation et lever les hypothèses simplificatrices initiales pour créer une méthode plus générale et validée expérimentalement.

Une seconde perspective explore l'utilisation des réseaux de neurones informés par la physique (PINNs) comme alternative pour résoudre le modèle de poutre de Cosserat [59]. Nous avons présenté une formulation PINN préliminaire pour résoudre le PAL statique, où un réseau de neurones apprend à prédire le champ de moment interne $c_\theta(X)$. Le réseau est entraîné en minimisant une fonction de perte qui pénalise les écarts par rapport aux conditions aux limites et aux équations différentielles sous-jacentes de l'équilibre statique, les dérivées étant calculées par différentiation automatique. Nos premiers résultats indiquent que le PINN peut approximer la solution avec une précision raisonnable. Cependant, le processus d'entraînement est actuellement de plusieurs ordres de grandeur plus lent que les solveurs classiques et est sensible au réglage des hyperparamètres. Ces défis suggèrent que, bien que les PINN offrent une approche nouvelle et sans maillage, des recherches importantes sont nécessaires pour en faire un outil compétitif et fiable pour résoudre les équations complexes régissant la robotique continue.