2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Solving Dynamic Cosserat Rods with Frictional Contact Using the Shooting Method and Implicit Surfaces

Radhouane Jilani*, Pierre-Frédéric Villard*, Erwan Kerrien*
*Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Abstract— **We propose a method for solving the strong form of the dynamic Cosserat rod clamped-free boundary value problem (BVP) under frictional contact. The shooting method is employed for BVP solving, and contact response is computed using the penalty method. The proposed method is tailored for accurate and rapid simulation of catheterization-like medical procedures. Our contact scenarios involve complex shapes and real patient geometry, modeled with implicit surfaces as blobby models. The results show that using blobby models for contact detection is orders of magnitude faster than using triangle meshes. This approach provides smoother contact forces, accelerating the convergence of the shooting method. We have made our code available as open source at https://gitlab.inria.fr/rjilani/IROS2024/.**

I. INTRODUCTION

Slender elastic tools are increasingly used in medicine due to their versatility and compliance with human anatomy. Their development leverages computer-based simulation and design optimization that require mechanical modeling [1]. The Cosserat rod model, along with its variants, has gained significant interest over the full 3D finite element method (FEM), due to its accuracy and efficiency, in particular to model continuum and soft robots [2], [3]. Accounting for environmental interactions in the Cosserat rod model has sparked little interest in the robotics community. However, these interactions are paramount in applications such as mechanical thrombectomy or cochlear implantation [4]. In such procedures, slender elastic micro-tools must be inserted into tubular structures (e.g., blood vessels) and guided to desired locations. For their simulations to be useful in pre-operative guidance, they must be accurate, real-time, and also adapted to patient data [1]. Therefore, the objective of this paper is to address frictional contact in the dynamic Cosserat rod model with the goal of developing a fast and accurate method for simulating catheterization-like procedures. Throughout this paper, when we refer to contact handling, it encompasses both contact detection and contact response. Contact detection involves determining whether a point is outside or inside the surface, calculating the distance between the point and the surface, and computing the surface normal at that point. Contact response refers to the calculation of contact forces, which includes both normal and frictional forces.

Frictional contacts have been incorporated into staggered grid discretization methods for solving Cosserat rods. These methods involve computing accelerations of discretized cross-section orientations and centerline positions with finite differences, followed by a forward integration in time. For contact response, Gazzola et al. [5] used the penalty method, which transforms the constrained problem of contacts into an unconstrained one by penalizing the interpenetration with a force that depends on the depth of collision. The penalty method is widely used for its computational efficiency, but it does not guarantee a contact-free state. Based on Lang et al.'s method [6], Tschisgale et al. [7] managed frictional contact using a constraint-based method that calculates contact response as collision impulses computed with the desired change in velocities after contact. Although this method is accurate, it requires an iterative approach for multiple contacts and is not suited for methods that do not rely on computing accelerations. Staggered grid discretization methods have linear time complexity that depends on the number of nodes, and computing node acceleration is parallelizable. However, for some applications, the discretized equations become stiff, necessitating very small time steps.

In recent years, the robotics community has mostly adopted two methods for solving Cosserat rods, namely the continuous method and the reduction-based method. The continuous method involves solving the strong form of static or dynamic boundary value problems (BVPs) either by shooting methods [8], [9] or by collocation methods [10], [11]. Recently, Wiese et al. [12] used the shooting method combined with the penalty method to solve the static BVP under frictional contact. Simulations were conducted in the context of a soft pneumatic actuator interacting with simple shapes. Their results showed good accordance with FEM. Weeger et al. [13] addressed the dynamic BVP under frictional contact using an iso-geometric collocation method by collocating positions, orientations, and their time velocities. The penalty method was used and contact forces that happen to be outside the collocation points were interpolated between the two or four nearest ones. Indeed, collocation methods compute the residual function solely at these points, which are typically limited in number for time efficiency. Consequently, contacts cannot be handled accurately for complex scenarios. One limitation of continuous methods is that they often require adjustments for modeling different robots, since their residual (objective) function changes depending on the specific design of the robot. Conversely, reduction-based methods are designed to easily model different robots, thanks to their general-purpose residual function. These methods are based on solving the weak form of the BVP, generally employing a strain-based reduction [2]. Recently, Xun et al. [14] proposed a framework for solving dynamic Cosserat rods under frictional contact that accounts for all types of

979-8-3503-7769-9/24/$31.00 ©2024 IEEE

boundary conditions and external constraints. The contact response method is based on transforming the nonlinear complementarity problem of contact [15] into a non-linear equality constraint that can be solved using standard root-finding techniques. The unknowns of this method are the generalized strain coordinates, their time derivatives, and variables that enforce the constraints, which make the method computationally not suited for fast simulations. Reduction-based methods tend to be more complex to implement compared to continuous methods that rely on the shooting method. For certain applications, continuous methods are well-adapted and offer greater computational efficiency than reduction-based methods. In this paper, we opt for the continuous model due to its accuracy, solved using the shooting method for its time efficiency. For contact response, we avoid introducing external constraints outside of the BVP. To further improve time efficiency, contact forces are computed with the penalty method and are subsequently incorporated into the model as external forces.

Our contributions are summarized as follows. The task of contact detection between complex shapes is acknowledged as a significant bottleneck in simulations. By adopting implicit surfaces to represent our objects, we show that this approach leads to more time-efficient contact detection and BVP resolution compared to using triangle meshes. Additionally, our method has been tested on a rod navigating through a model of an internal carotid artery created from real patient geometry. To the best of our knowledge, we are the first to address Cosserat rod dynamics under frictional contact and to simulate catheterization-like procedures using the shooting method, and also to use a continuous representation for both contact surfaces and the Cosserat rod. We also address the difficulty, which has never been considered in the context of the shooting method, that contact forces must be included in the BVP as external distributed forces.

In section II, we outline the continuous BVP employed in this paper. Section III describes the continuous contact handling. In section IV, we explain how to solve the BVP and incorporate contact forces. Our numerical applications are presented in section V, and we conclude in section VI.

II. COSSERAT ROD MODEL

In this paper, the dynamic BVP for clamped-free Cosserat rods is used. s and t represent the arc-length and time domains, respectively, while $(\cdot)_s$ and $(\cdot)_t$ denote their respective partial derivatives [8]:

$$\begin{pmatrix} p_s \\ R_s \\ n_s \\ m_s \\ q_s \\ \omega_s \end{pmatrix} = \begin{pmatrix} Rv \\ R\hat{u} \\ \rho AR(\hat{\omega}q + q_t) - f \\ \rho R(\hat{\omega}J\omega + J\omega_t) - \hat{p}_s n - l \\ v_t - \hat{u}q + \hat{\omega}v \\ u_t - \hat{u}\omega \end{pmatrix}, \tag{1}$$

with the following boundary conditions:

$$\begin{aligned} p(t,0) &= p_0(t), \quad & R(t,0) &= R_0(t), \\ q(t,0) &= q_0(t), \quad & \omega(t,0) &= \omega_0(t), \\ n(t,L) &= n_L(t), \quad & m(t,L) &= m_L(t). \end{aligned} \tag{2}$$

Components are introduced for first linear, then angular, degrees of freedom. p and R denote the centerline position and the rotation matrix of cross-section orientation, respectively. Their arc-length rates of change in the cross-sectional frame are captured by v and u, with q and ω as their temporal counterparts. The internal forces and moments in the inertial frame are represented by n and m, respectively, while f and l correspond to the external distributed forces and moments. Parameters L, ρ, A, and J specify the rod's pre-deformation length, density, cross-sectional area, and second mass moment of inertia. The hat operator $\hat{(\cdot)}$ maps from $\mathbb{R}^3$ to skew-symmetric matrix. v and u are computed from n and m using linear elastic constitutive law and Kelvin-Voigt type viscous damping [16].

III. CONTACT HANDLING

In this section, we present the continuous frictional contact handling between the rod's centerline p and a stationary rigid implicit surface modeled using blobby models [17]. Contact handling applied to p is justified for our application because the radii of micro-tools such as catheters and guidewires are very small compared to their length. In this section, all variables are parameterized with t and s, unless specified otherwise.

A. Contact detection

Following [17], the contact surface is represented by the 0-level set of a blobby model defined by:

$$\zeta(P) = T - \sum_{b=1}^{M} \alpha_b \psi\left(\frac{|P - \sigma_b|}{\alpha_b}\right), \forall P \in \mathbb{R}^3 \tag{3}$$

where the constants are: T, representing a threshold; M, denoting the number of blobs; and for each blob, σ_b and α_b indicate its center and width, respectively. $\psi(x) = \left(1 + x^2/5\right)^{-2}$ is the Cauchy profile. A point p is considered in contact if $\zeta(p) \geq 0$. An approximate but direct and fast computation of the signed distance between a point p and the surface is given by Taubin's formula [18]:

$$\phi = \frac{\zeta}{|\nabla\zeta|} \tag{4}$$

where both ζ and its gradient $\nabla\zeta$ are calculated in closed form. The normal direction, denoted by $w_\perp$, required for the computation of contact forces, is computed as follows:

$$w_\perp = -\frac{\nabla\zeta}{|\nabla\zeta|} \tag{5}$$

The minus sign makes $w_\perp$ point inside the surface.

B. Contact response

The contact force is computed using the penalty method. The normal contact force magnitude $\lambda_\perp$ is determined as follows [19]:

$$\lambda_\perp = \begin{cases} 0, & \phi \leq 0, \\ \frac{k}{2\gamma}\phi^2, & 0 < \phi \leq \gamma, \\ k\left(\phi - \frac{\gamma}{2}\right), & \gamma < \phi. \end{cases} \tag{6}$$

where the constant k represents the penalty stiffness, and the constant γ provides a smooth transition between contact and non-contact states, improving numerical stability. In the context of the penalty method, Taubin's approximation (4) is acceptable since k is empirically adapted for every simulation. Thus, the normal contact force is:

$$\bar{f}_\perp = \lambda_\perp w_\perp \tag{7}$$

Friction forces are determined using the Coulomb model, which limits their magnitude:

$$\lambda_\parallel \leq \mu \lambda_\perp \tag{8}$$

where the constant μ is the friction coefficient. The direction and magnitude of the friction force depend on whether the friction is sliding or sticking, based on relative tangential velocity $p_{t\parallel}$. For sliding friction, the force magnitude is maximal ($\lambda_\parallel = \mu\lambda_\perp$) and opposes $p_{t\parallel}$. For sticking friction, the force can take any value inside the friction cone (8). We chose it to have the maximal magnitude and to counteract the direction of other tangential external forces $\bar{f}_{o\parallel}$, if present [5]. Thus, the friction force:

$$\bar{f}_\parallel = \begin{cases} -\mu\lambda_\perp \dfrac{p_{t\parallel}}{|p_{t\parallel}|}, & |p_{t\parallel}| > \epsilon, \\ -\mu\lambda_\perp \dfrac{\bar{f}_{o\parallel}}{|\bar{f}_{o\parallel}|}, & |p_{t\parallel}| \leq \epsilon, \end{cases} \tag{9}$$

where the constant ϵ serves as a small threshold for reasons related to numerical precision. $p_{t\parallel}$ is computed as follows:

$$\begin{gathered} w_{\parallel 1} = p_s \times w_\perp \\ w_{\parallel 2} = w_\perp \times w_{\parallel 1} \\ p_{t\parallel} = (p_t \cdot w_{\parallel 1}) w_{\parallel 1} + (p_t \cdot w_{\parallel 2}) w_{\parallel 2} \end{gathered}$$

$\bar{f}_{o\parallel}$ is computed in the same way. Finally, the contact force is:

$$\bar{f}_c = \bar{f}_\perp + \bar{f}_\parallel \tag{10}$$

No moments are produced from contact forces since they are applied at p.

IV. BOUNDARY VALUE PROBLEM RESOLUTION

The BVP (1,2) with contact forces is solved for every time step t_i using the shooting method where the only unknowns are $n(t_i, 0)$ and $m(t_i, 0)$. For time efficiency, explicit Euler is used for arc-length integration over N points, denoted by $\{s_j = \frac{jL}{N-1}\}$, $j = 0, ..., N-1$. Due to the use of a low-order integration method, quaternion integration of orientations is used to prevent drifts from the $SO(3)$ space [20]. Time is discretized implicitly using the backward differentiation formula 2 (BDF2), with $q_t(t_i, s_j), \omega_t(t_i, s_j), v_t(t_i, s_j)$, and $u_t(t_i, s_j)$ computed at each time step based on known values from actual and previous time steps [8]. $p_0(t_i)$ and $R_0(t_i)$ are user inputs, while $q_0(t_i)$ and $\omega_0(t_i)$ are approximated as follows:

$$\begin{aligned} q_0(t_i) &= R_0(t_i)^{\mathrm{T}} \frac{p_0(t_i + \delta t) - p_0(t_i)}{\delta t} \\ \omega_0(t_i) &= \frac{\left(\log(R_0(t_i)^{\mathrm{T}} R_0(t_i + \delta t))\right)^\vee}{\delta t} \end{aligned} \tag{11}$$

where δt is the time-step size and $\vee$ is the inverse of the hat operator mapping. Contact handling is applied to points $\{s_\tau\}$, $\tau = 1, ..., N-1$, since a clamped-free BVP is used. This choice is computationally justified for our target application by the fact that a significant portion of the rod remains in contact throughout the simulation. This approach also maintains acceptable levels of accuracy for vessel-like shapes, which are typically smooth and continuous. $\bar{f}_c(t_i, s_\kappa)$ are added to the BVP as external distributed contact forces $f_c(t_i, s_\kappa)$, with $\kappa = 1, ..., N-2$. However, $\{\bar{f}_c(t_i, s_\kappa)\}_\kappa$ field is not differentiable, because its components act as point-wise Dirac distributions [19]. Instead, we propose the following approximation:

$$f_c(t_i, s_\kappa) = \frac{\bar{f}_c(t_i, s_\kappa)}{\delta s} \tag{12}$$

where δs is the step size of arc-length integration. Thus, the external distributed forces are:

$$f(t_i, s_\kappa) = f_c(t_i, s_\kappa) + f_o(t_i, s_\kappa) \tag{13}$$

where f_o are the other external distributed forces, such as the gravity force. Finally, the tip contact force is prescribed as a boundary condition:

$$n_L(t_i) = \bar{f}_c(t_i, L) + \bar{f}_o(t_i, L) \tag{14}$$

where $\bar{f}_o(t_i, L)$ are other external forces applied at the tip. Algorithm 1 presents our simulation loop. The initial static solution is computed to calculate BDF2 components and the initial guess for the first time step. The static BVP is simply obtained by removing all velocities and accelerations from (1,2). The unconstrained BVP (i.e., without contact forces)

Algorithm 1 Simulation loop

```
Solve the static BVP
for each time step $t_i$ do
    Advance in time (update BDF2 components)
    Update $p_0(t_i), R_0(t_i)$
    Compute $q_0(t_i), \omega_0(t_i)$ using (11)
    Solve the dynamic BVP without contact forces (1,2)
    Compute $f(t_i, s_\kappa)$ (13)
    Compute $n_L(t_i)$ (14)
    Solve the dynamic BVP with contact forces (1,2)
end for
```

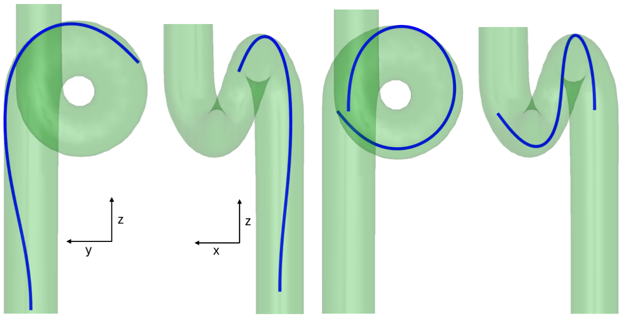

Fig. 1. Helical insertion: the rod is inserted into a helical tube. From left to right, the first two images are captured after 0.5 s, and the last two after 1 s of the simulation. In this figure, contact detection is performed using a blobby model.

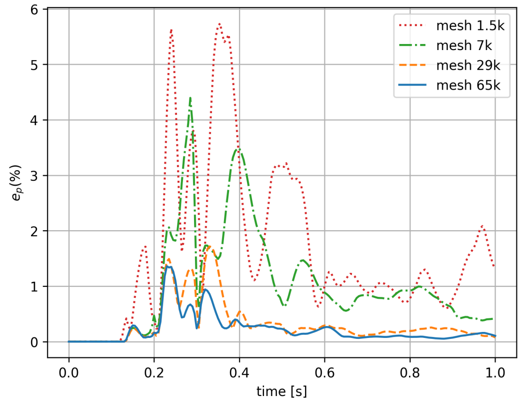

Fig. 2. Helical insertion: end position error (%) with respect to time between blobby model and triangle mesh simulations. Increasing the number of triangles in a mesh decreases the error between the two simulations.

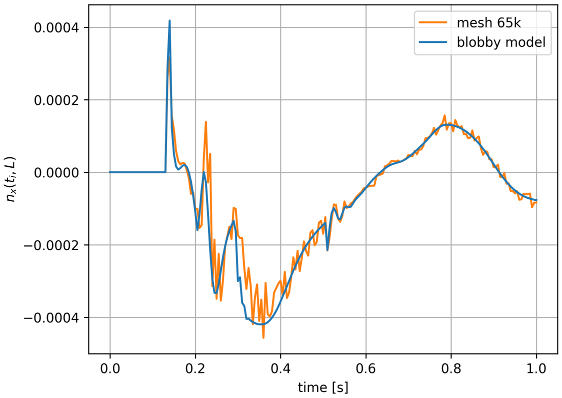

Fig. 3. Helical insertion: end point internal force along the x-axis with respect to time for blobby model and 65k triangle mesh simulations. The blobby model provides smoother contact forces.

is solved first to compute the correct penalty forces to be applied so as to decrease interpenetration. In the context of the shooting method, the choice of the initial guesses is important for the method to converge. At each time step, the initial guesses for both constrained and unconstrained BVPs are set to the solutions from the previous step for their respective problems.

V. NUMERICAL APPLICATIONS

We evaluated our method through three numerical applications. Our objective was to assess the method's ability to manage complex contact scenarios, validate it against real patient geometry, and compare the effects of blobby model and triangle mesh contact detection on simulations. The code was implemented in Python, with scipy's hybr method chosen for root finding. The termination tolerance was set to 10^{-8}. Simulations were conducted on a computer equipped with an i9-10900K CPU at 3.70 GHz. Execution time is calculated as the average across all time steps. For BVP solving, execution time covers both constrained and unconstrained solutions. Contact response took a negligible time compared to other execution times and is therefore not reported. For every numerical application, $\delta t = 0.005$ s, and $\gamma = 5 \cdot 10^{-6}$ m. B_{se} represents the damping matrix for shear and extension, while B_{bt} corresponds to the damping matrix for bending and torsion.

A. Helical insertion

The first numerical application has two objectives: first, to evaluate our method for managing multiple simultaneous frictionless contacts, and second, to compare simulations employing either blobby models or triangle meshes for contact detection. An initially straight rod is inserted with constant displacements into a helical tube over a span of 1 s (Fig. 1). $p_0(t_i)$ follows the centerline of the tube, and $R_0(t_i) = I$, the identity matrix. The rod parameters are: length $L = 0.26$ m, radius $r = 0.0003$ m, Young's modulus $E = 5.5 \cdot 10^7$ Pa, Poisson's ratio $\nu = 0.45$, $B_{bt} = 5 \cdot 10^{-7} I$, and $B_{se} = 5 \cdot 10^{-8} I$. The number of points for arc-length integration, N, is 125. Contact parameters are: $k = 0.32$ N/m, and $\mu = 0$. The helix shape radius is 0.03 m, and the tube radius is 0.016 m. The experiment is inspired by Xun et al. [14], although we have increased the length of the rod by 10 times, the ratio between the helix radius and L is preserved.

1) Blobby model: First, contact detection was performed with a blobby model made of $M = 286$ blobs. Fig. 1 depicts the rod after 0.5 s and 1 s of the simulation. It shows that our method can handle multiple out-of-plane contacts. Note that with the penalty method, a small interpenetration is acceptable, and our rod correctly follows the contact surface. The execution time for BVP solving was 0.374 s, and for contact detection, it was 0.011 s, totaling 0.385 s.

2) Comparison between blobby models and triangle meshes: Second, we compared the previous simulation with simulations that use triangle meshes for contact detection. The meshes were generated from the blobby model with the marching cubes algorithm. Cube sizes of $L/25$, $L/50$, $L/100$, and $L/150$ were used, resulting in approximately 1.5k, 7k, 29k, and 65k triangles, respectively. Contact detection for triangle meshes leverages a bounding volume hierarchy, with ray casting used to determine if a point lies inside or outside the mesh. For a fair comparison, we

TABLE I

HELICAL INSERTION: EXECUTION TIME AND NUMBER OF EVALUATIONS OF THE OBJECTIVE FUNCTION ξ

Contact surface	Avg. contact detection time (s)	ξ
Blobby model	0.011	4773
Mesh 1.5k	0.104	5070
Mesh 7k	0.337	4963
Mesh 29k	1.05	4816
Mesh 65k	2.16	4797

first verify that triangle mesh simulations converge to the blobby mesh simulation. Fig. 2 displays the error in end positions for mesh simulations compared to the simulation with the blobby model. The error was computed as: $e_p = \dfrac{|\bar{p}(t_i, L) - \tilde{p}(t_i, L)|}{L} \times 100$, where $\bar{p}$ and $\tilde{p}$ represent the rod's centerline position in respectively the blobby model and the mesh simulations. As expected, increasing the number of triangles entails a decrease in the error. Fig. 3 shows the boundary condition $n_x(t_i, L)$ (equal to $\bar{f}_{cx}(t_i, L)$ within the solver's tolerance) for both blobby model and 65k mesh simulations. We observe that the contact forces in the blobby model simulation are smoother than those produced in mesh simulations, even when compared to the largest one. This is due to the smooth normal function in 3D space provided by the blobby model. This smoothness affects the number of evaluations of the residual function, denoted by ξ. Table I shows ξ for both blobby model and mesh simulations. It reveals that the smoother the surface, the fewer evaluations are needed for convergence. This is attributed to the fact that surface smoothness impacts the normal directions, hence the contact force, and significant changes in this force between two time steps necessitate more evaluations by the BVP solver, given that its initial guesses rely on the solution from the previous step. Consequently, we can conclude that the blobby model approach to contact detection speeds up the convergence of the BVP solver. Table I also shows the average contact detection execution time for blobby model and mesh simulations. Blobby model contact detection is nine times faster than the 1.5k mesh simulation and nearly 200 times faster than the 65k mesh simulation. We also observe that starting from the 29k mesh simulation, the contact detection execution time exceeds that of BVP solving, making our method overall several times faster by using blobby models. Finally, a triangle mesh involves a trade-off between accuracy and execution time, whereas the blobby model offers good accuracy with minimal execution time. Furthermore, blobby models enable modeling complex surfaces with fewer parameters, thereby reducing memory footprint.

B. Cantilever rod in frictional contact

The objective of this numerical application is to evaluate the influence of friction on our method. Inspired by Weeger et al. [13], an initially straight rod, clamped at one end, falls due to the effect of gravity, while a sphere is positioned near its end and below it. The experiment was conducted with different values of μ. Fig. 4 depicts the rod after 1 s of the simulation for each value. The clamping parameters are: $p_0(t_i) = 0$ and $R_0(t_i) = I$. μ takes the following values: $0, 0.1, 0.2, 0.3$ and 0.4. Gravity acts along the $+x$-axis. The rod's parameters are: $L = 1$ m, $r = 0.005$ m, $E = 5 \cdot 10^7$ Pa, $\nu = 0.5$, $\rho = 1100$ kg/m^3, $B_{bt} = 5 \cdot 10^{-3} I$, and $B_{se} = 5 \cdot 10^{-4} I$. The sphere is created with its implicit equation. Its normal points outward, its radius is $\beta = 0.01$ m, and its center is set to $[\beta, 0, 0.9]^{\mathrm{T}}$. Contact response parameters are: $k = 500$ N/m, and $\epsilon = 0.001$ m/s. The number of points for arc-length integration is $N = 200$. The execution time for solving the BVP was 0.554 s, based on the average from the five simulations. Execution time for contact detection is not reported, as it is negligible. As expected, the results in Fig. 4 demonstrate that with an increasing μ value, the rod remains in contact with the sphere for a longer duration, indicating that the friction effect is successfully incorporated into our method.

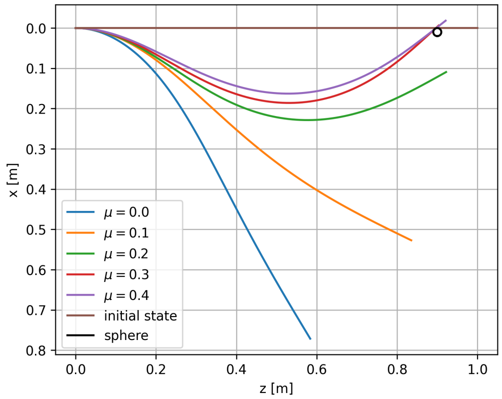

Fig. 4. Cantilever rod in frictional contact: the position of the rod is shown after 1 s of simulation for different friction coefficients, μ. The higher the value of μ, the longer the rod remains in contact with the sphere.

C. Carotid artery insertion

The objective of the last numerical application is to test our method against real patient geometry in order to verify its ability to simulate catheterization-like procedures. A rod is inserted with constant displacement into a model of an internal carotid artery for 1 s (Fig. 5). This model was created from 3D rotational angiography data of a real patient using the algorithm from [17]. The surface is modeled as a centerline where each point on the centerline is associated with a blobby model. This results in a model consisting of a tree of 70 blobby models, with each model having an average of 35 blobs. As a consequence, a required preliminary step to managing contact for a position $p(t_i, s_\tau)$ consists in selecting the adequate blobby model. It was chosen to be the one associated with the closest point to $p(t_i, s_\tau)$ on the surface centerline. The rod parameters are: $L = 0.08$ m, $r = 0.0003$ m, $E = 5.5 \cdot 10^5$ Pa, $\nu = 0.45$, $B_{bt} = 5 \cdot 10^{-10} I$, $B_{se} = 5 \cdot 10^{-11} I$, and $q_0(t_i) = 0$. For arc-length integration, $N = 50$. Contact parameters are: $k = 0.1$ N/m, $\mu = 0.05$,

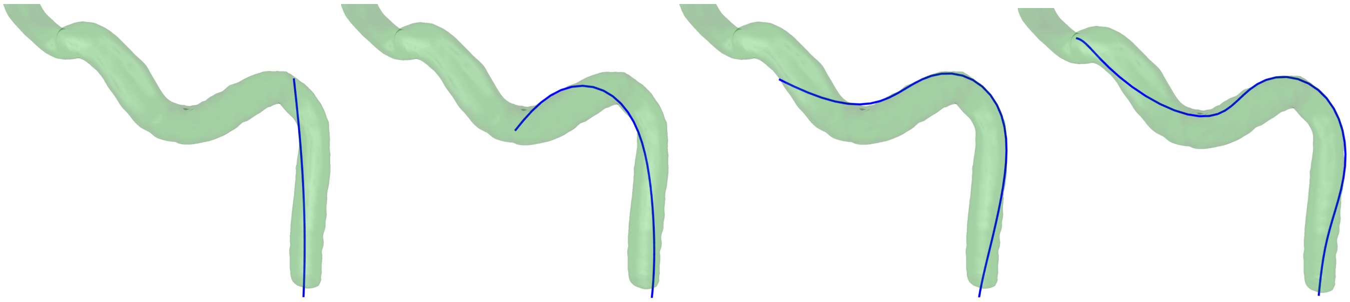

Fig. 5. Carotid artery insertion: from left to right, the simulation at 0.25, 0.5, 0.75, and 1 s. At 0.75 and 1 s, we observe a buckling effect due to friction.

and $\epsilon = 8 \cdot 10^{-5}$ m/s. Fig. 5 shows the simulation for four different time steps. We can observe that the insertion has been successfully achieved. Additionally, a buckling effect is observed due to friction. The execution time for solving the BVP was 0.181 s, and for contact detection, it was 0.0177 s. A total of 0.193 s per time step, which is relatively fast.

VI. CONCLUSIONS

We propose a fast and accurate method for solving Cosserat rod dynamic BVP under frictional contact, designed for catheterization-like procedures. The BVP is solved using the shooting method and the penalty method is used for contact response. Our surfaces are modeled with implicit functions using blobby models. We compare the impact of using blobby models and triangle meshes, for contact detection, on simulations. Our method was tested against complex vessel-like shapes and real patient geometry. The results show that our method successfully handles complex frictional contact scenarios that involve multiple out-of-plane contact forces. Contact detection using blobby models is orders of magnitude faster than that with triangle meshes, making our method overall multiple times faster. Furthermore, the blobby model approach provides smooth contact forces which makes the shooting method converge more quickly compared to the triangle mesh approach. Moreover, the presented method can be used for continuum and soft robots after minor adjustments (e.g., adding contact moments). In the near future, we plan to add blobby model deformation and validate our method against experimental data.

ACKNOWLEDGMENT

This work was supported by the French Agence Nationale de la Recherche under Grant ANR-20-CE45-0011.

REFERENCES

[1] C. Duriez, *Real-time haptic simulation of medical procedures involving deformations and device-tissue interactions.* habilitation thesis, Université des Sciences et Technologie de Lille - Lille I, Feb. 2013.

[2] 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, pp. 847–863, June 2021.

[3] D. C. Rucker and R. J. Webster III, “Statics and dynamics of continuum robots with general tendon routing and external loading,” *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1033–1044, 2011.

[4] G. Dogangil, B. L. Davies, and F. Rodriguez y Baena, “A review of medical robotics for minimally invasive soft tissue surgery,” *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 224, pp. 653–679, May 2010.

[5] 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, p. 171628, June 2018.

[6] H. Lang, J. Linn, and M. Arnold, “Multi-body dynamics simulation of geometrically exact Cosserat rods,” *Multibody System Dynamics*, vol. 25, pp. 285–312, Mar. 2011.

[7] S. Tschisgale, L. Thiry, and J. Fröhlich, “A constraint-based collision model for Cosserat rods,” *Archive of Applied Mechanics*, vol. 89, pp. 167–193, Feb. 2019.

[8] J. Till, V. Aloi, and C. Rucker, “Real-time dynamics of soft and continuum robots based on Cosserat rod models,” *The International Journal of Robotics Research*, vol. 38, pp. 723–746, May 2019.

[9] 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, pp. 1544–1562, Apr. 2023.

[10] 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, Oct. 2020.

[11] 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, Oct. 2023.

[12] 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.

[13] 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, Jan. 2018.

[14] L. Xun, G. Zheng, and A. Kruszewski, “Cosserat-rod based dynamic modeling of soft slender robot interacting with environment,” *arXiv preprint arXiv:2307.06261*, July 2023.

[15] C. Duriez, C. Andriot, and A. Kheddar, “Signorini's contact model for deformable objects in haptic simulations,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3232–3237, 2004.

[16] H. Lang, S. Leyendecker, and J. Linn, *Numerical experiments for viscoelastic Cosserat rods with Kelvin-Voigt damping*. Fraunhofer Inst. für Techno-und Wirtschaftsmathematik, ITWM, 2013.

[17] 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, p. 685, Jan. 2017.

[18] 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 and Machine Intelligence*, vol. 13, pp. 1115–1138, Nov. 1991.

[19] O. Weeger, B. Narayanan, L. De Lorenzis, J. Kiendl, and M. L. Dunn, “An isogeometric collocation method for frictionless contact of Cosserat rods,” *Computer Methods in Applied Mechanics and Engineering*, vol. 321, pp. 361–382, July 2017.

[20] C. Rucker, “Integrating Rotations Using Nonunit Quaternions,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 2979–2986, Oct. 2018.