

Cosserat Rods with Projective Dynamics

Carlota Soler^{†1,2}, Tobias Martin^{‡2} and Olga Sorkine-Hornung^{§1}

¹ETH Zürich, Switzerland
²VirtaMed Research, Switzerland

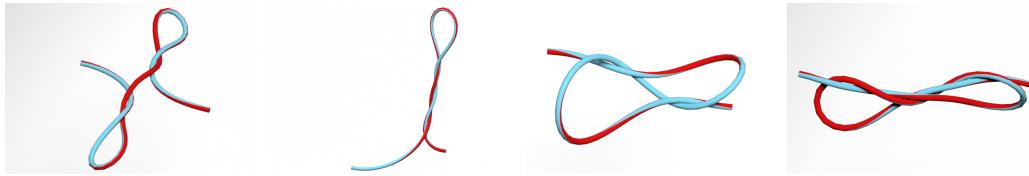


Figure 1: Snapshots of a simulated rod with self-collisions. The rod tangles with itself after applying torsional deformation on the endpoints, leading to the formation of plectonemes. Interactive displacement of the endpoints enables the formation of knots.

Abstract

We present a novel method to simulate Cosserat rods with Projective Dynamics (PD). The proposed method is both numerically robust and accurate with respect to the underlying physics, making it suitable for a variety of applications in computer graphics and related disciplines. Cosserat theory assigns an orientation frame to each point and is thus able to realistically simulate stretching and shearing effects, in addition to bending and twisting. Within the PD framework, it is possible to obtain accurate simulations given the implicit integration over time and its decoupling of the local-global solve. In the proposed method, we start from the continuous formulation of the Cosserat theory and derive the constraints for the PD solver. We extend the standard definition of PD and add body orientations as system variables. Thus, we include the preservation of angular momentum, so that twisting and bending can be accurately simulated. Our formulation allows the simulation of different bending behaviors with respect to a user-defined Young's modulus, the radius of the rod's cross-section, and material density. We show how different material specifications in our simulations converge within a few iterations to a reference solution, generated with a high-precision finite element method. Furthermore, we demonstrate mesh independence of our formulation: Refining the simulation mesh still results in the same characteristic motion, which is in contrast to previous position based methods.

CCS Concepts

• Computing methodologies → Physical simulation; Real-time simulation; Simulation by animation; Computer graphics;

1. Introduction

Simulating the physics of rods is a challenging problem. Existing methods convincingly simulate the intricate bending, twisting and stretching behavior of rods, but they usually have to trade numerical accuracy for numerical stability or vice versa. Both of these attributes, numerical stability, and physical accuracy are required for various applications: A realistic, accurate, and numerically robust rod simulation is an essential building block in virtual suturing and simulation of other deformable tubular structures.

Simulating rods solely with positions yields a simplistic and unrealistic behavior. Therefore, in order to reproduce physically plausible rod behavior, it is essential to employ a fundamental theory that models twists, as done by Cosserat theory. Cosserat theory equips points of the material with an orientation. This enables to reproduce how a rod stretches to some material and how a twist propagates along the rod when a rotational force is applied. Rod simulation is a complex problem because both positions and orientations, or local frames, need to be tracked.

Previous methods simulate stable Cosserat rods with position based dynamics (PBD) [KS16, DKWB18], but their results often drift away from reality due to the method's explicit time integration and constraint corrections. Finite element method (FEM) implementations [BWR*08, BAV*10] require small time steps to ensure stability, albeit providing accurate simulations thanks to the

[†] soler.carlota@gmail.com

[‡] tobias.martin@virtamed.com

[§] sorkine@inf.ethz.ch

internal/external force computation. The projective dynamics (PD) framework [BML^{*}14] enables accurate simulations by the implicit integration over time and the decoupling of the local-global solve steps. In this paper, we propose combining Cosserat theory with the ideas of PD, which results in a simulation that is both accurate with respect to the physical behavior (Fig. 4) of the rod and numerically robust (Fig. 13). These traits are important in the context of surgical training, where the main goal is to reproduce real surgery procedures as accurately as possible.

PD brings together the PBD constraint projection in the local step and the FEM potential computation in the global step, leading to an efficient system solving. Within the global step, PD provides the optimal solution of the system by taking all the potentials into consideration, a concept inherited from FEM. At the same time, the potentials include the constraint projections found in the local step, a concept inherited from PBD methods. Note that position based methods omit the global step by iterating over the constraints and thus lose track of the global solution, such that the rod simulation might stop following its governing equations. In contrast, PD uses an implicit integration scheme together with a local-global solve step, leading to more accurate simulations.

In this paper, we make the following contributions:

- We incorporate body orientations into the standard PD solver and thereby introduce the preservation of the angular momentum.
- We formulate Cosserat rod constraints and potentials for PD. We discretize the continuous formulation of Cosserat potentials and derive the respective constraints and potentials for the local and global steps within the PD framework.
- Similarly to FEM methods, we introduce the potential weights of the potentials in the global step with respect to material parameters or geometric properties of the rod.
- We compare our results to a FEM reference simulation and demonstrate the realism of our simulations for different elasticity values. Additionally, we show the mesh independence of our simulation when refining the simulation mesh.

2. Related work

Rod simulation is a difficult problem since both positions and orientations on the rod need to be predicted. Tracking the orientation along the rod enables to simulate twisting and torsional effects. Early theories date from around 1859, Kirchhoff [Kir59] being one of the first to devise a three-dimensional theory that replaced the 1D-body approach. His early theory and further research [Dil92] led to an explicit representation of the rod's centerline. The orientation of the rod is represented by several material frames, which enable to keep track of twisting and bending. A contemporary example of Kirchhoff rod simulation is discrete elastic rods [BWR^{*}08, BAV^{*}10], where the material frame is treated as quasi-static by assuming an inextensible rod. Later theories involving the Cosserat brothers (1909) led to the formulation of Cosserat rods [Rub13]. This theory models the rod as a space curve with two additional directions, which model material fibers in the cross-section of the rod. These fibers can stretch in length and shear relative to the normal of the cross-section and the tangent of the space curve, which allows to simulate extensible rods and hence provide a broader model compared to Kirchhoff rods.

Cosserat rods. Pai et al. [Pai02] was the first to introduce the Cosserat model to the computer graphics community with an implicit representation of the rods, aimed at simulating threads and catheters in virtual surgery procedures like laparoscopy. In this method, the centerline is expressed implicitly by an approximation of smooth curves, which makes collision detection difficult.

Explicit discretization of Cosserat rods is a more convenient approach, as the geometry of the rod can be easily reconstructed. The CORDE method [ST07] uses a deformation model for simulating dynamic elastic rods based on the Cosserat theory with continuous energies. After discretizing the rod, the energy is computed per element with finite element methods, and thus the dynamic evolution of the rod is obtained by numerical integration of the resulting Lagrange equations of motion. Although the results are shown to be physically plausible, the explicit time integration of this approach requires very small time steps and strong damping to remain stable, which is the main performance bottleneck of the simulation.

Lang et al. [LLA11] introduces a geometric model for discretizing the rod similar to the one in CORDE [ST07] and derives the equations of motion in the continuous domain by applying Lagrangian field theory. These equations are solved using the finite difference method together with standard solvers for stiff differential equations. Casati et al. [CBD13] presents an integration scheme based on power expansions, which reaches higher precision faster compared to classical numerical integrators. Their method is based on a semi-implicit time stepping scheme, which is by definition less stable than an implicit integrator, and hence the motivation to simulate rods with an implicit scheme such as PD.

Position based methods. Opposed to nodal finite element methods, Position based dynamics (PBD) [MHHR07] uses impulse based dynamics, which consists in directly manipulating positions with constraints and hence offers more control over the body, leading to an easy collision and penetration handling. Although being a stable and fast method suitable for interactive applications, the way the positional updates are formulated can lead to physically implausible results. One of PBD's limitations is that the material stiffness is dependent on the time step and iteration count, and hence it is difficult to simulate stiff materials within a few iterations.

These limitations are overcome with XPBD [MMC16] by introducing a simple extension that allows simulating stiff materials within a few iterations. However, XPBD approximates an implicit integrator, resulting in a less accurate result when comparing it to a true implicit integration scheme such as PD. As an example, Fig. 2 shows the result of a study comparing rods with XPBD (using

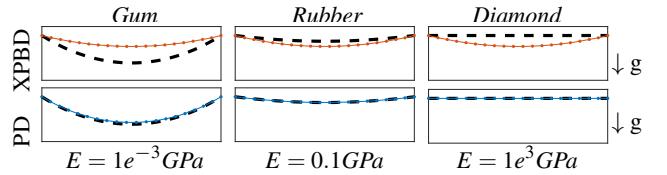


Figure 2: Deformation under gravity for different elasticities E (i.e., Young's Modulus). PD simulations converge to an Abaqus reference FEM simulation (dashed), as opposed to XPBD.

the available implementation [Nak]) and PD for both soft and stiff materials defining edge-length preservation constraints and 20 iterations. Only materials simulated with PD converge to a FEM reference solution, generated with the software *Abaqus* [HS02].

There are several examples [USS14, KS16, DKWB18] of Cosserat rods simulated within the PBD framework. The main challenge they address is how to include the rod orientations in an impulse dynamics formulation, which is meant to update solely positions. Both methods dispense with the three-component material frame defined in previous works [ST07, BWR^{*}08], so that orientations can be updated and stored similarly to positions. To replace the frame, Umetani et al. [USS14] define ghost points rotating around the rod's cross-section, which appears to introduce artifacts in rest pose. Kugelstadt et al. [KS16] define a single quaternion per segment as a particle, which rotates a given orthonormal basis defined at the origin, and is thus convenient for the update and storage within the PBD framework.

The method presented by Kugelstadt et al. [KS16] inherits the main drawback of PBD and hence the material stiffness depends on the iteration count. These inherited problems are overcome by introducing the XPBD extension [DKWB18], however, although being a significant improvement over the method presented by Kugelstadt et al. [KS16], it only allows simulating relatively stiff rods, which is due to its XPBD nature.

Projective dynamics. Projective dynamics [BML^{*}14] is a new method for implicit time integration of physical systems. The approach builds a bridge between nodal finite element methods and PBD, leading to a simple, robust, and accurate solver that supports many types of constraints.

In this paper, we formulate the Cosserat constraints using an implicit Euler integrator within the projective dynamics framework. Hence, our formulation allows the simulation of all types of rod material. As PD does not include orientations, we extend the standard formulation of the solver [BML^{*}14] and include the preservation of the angular momentum. PD combines the local step for the constraint projection and the global step for the potential optimization, leading to a more accurate (Fig. 4, 11), mesh independent (Fig. 6), and robust (Fig. 13) simulation method.

3. Cosserat theory

Cosserat rods are described by an arc-length parametrization $\mathbf{r}(s) : [0, L] \rightarrow \mathbb{R}^3$. Every point of $\mathbf{r}(s)$ is associated with a frame of orthonormal vectors $\{\mathbf{d}_1(s), \mathbf{d}_2(s), \mathbf{d}_3(s)\}$, also called directors. The

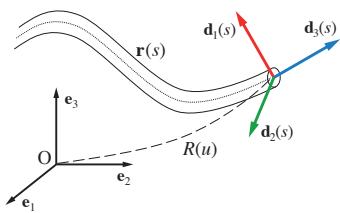


Figure 3: A Cosserat rod is defined by an arc length parametrized curve, where each point is augmented by an orientation.

cross-section of the rod is spanned by the directors $\{\mathbf{d}_1(s), \mathbf{d}_2(s)\}$. Their cross product $\mathbf{d}_3(s) = \mathbf{d}_2(s) \times \mathbf{d}_1(s)$ defines the normal of the cross-section. Each orthonormal frame, also called material frame, can be represented by a single quaternion $u(s)$. Given a fixed coordinate basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, each director is described as $\mathbf{d}_k = R(u) \mathbf{e}_k = u \mathbf{e}_k \bar{u}$, which is the quaternion rotation (denoted by $R(u)$) of the basis vector \mathbf{e}_k by quaternion u (Fig. 3). Note that we omit the parameter s for clearer notation whenever possible.

Cosserat continuous stretch and shear potential is defined by the following integral [LLA11]:

$$v_{SE} = \frac{1}{2} \int_0^L \tilde{\boldsymbol{\Gamma}}^\top C^\Gamma \tilde{\boldsymbol{\Gamma}} ds, \quad (1)$$

where the strain measure $\tilde{\boldsymbol{\Gamma}} \in \mathbb{R}^3$ is defined in material frame coordinates as

$$\tilde{\boldsymbol{\Gamma}} = R(u)^\top \partial_s \mathbf{r} - \mathbf{e}_3 \quad \text{and} \quad \boldsymbol{\Gamma} = \partial_s \mathbf{r} - \mathbf{d}_3 \quad (2)$$

is an equivalent expression to measure stretch and shear deformations. The tangent $\partial_s \mathbf{r}$ is the spatial derivative of the centerline at a given point s and \mathbf{d}_3 is the cross-section normal as defined above. The rod is subject to shear deformation if the direction of the tangent differs from the cross-section normal, $\partial_s \mathbf{r} \neq \mathbf{d}_3$. The rod is subject to stretch if the tangent is not unit length: $\|\partial_s \mathbf{r}\| \neq 1$, i.e., its length changes compared to the initial state.

Cosserat continuous bend and twist potential is defined as

$$v_{BT} = \frac{1}{2} \int_0^L \boldsymbol{\Omega}^\top C^\Omega \boldsymbol{\Omega} ds, \quad (3)$$

where $\boldsymbol{\Omega}$ denotes the material curvature vector for a given point s , which measures the rate of change in curvature [LLA11], i.e.,

$$\boldsymbol{\Omega} = R(u)^\top \partial_s R(u) \quad \text{or} \quad \boldsymbol{\Omega} = 2\bar{u} \circ \partial_s u. \quad (4)$$

The material curvature vector can also be formulated with the quaternion product (denoted by \circ) in Eq. (4), measuring the relative rotation between the material frame orientation and its spatial derivative.

The matrices

$$C^\Gamma = \begin{pmatrix} GA_1 & & \\ & GA_2 & \\ & & EA_3 \end{pmatrix} \text{ and } C^\Omega = \begin{pmatrix} EJ_1 & & \\ & EJ_2 & \\ & & GJ_3 \end{pmatrix} \quad (5)$$

encode the weight constants of the potential energies in terms of the cross-section area components A_1, A_2 and the cross-section geometrical moments of inertia J_1, J_2, J_3 [Sim85]. In the following we assume a circular cross-section, i.e., $A_1 = A_2 = A_3 = \pi r^2$ and $J_1 = J_2$. Expressing $J_1 = \iint_A x^2 d(x, y)$ in polar coordinates and substituting $d(x, y) = \tilde{r} d(\theta, \tilde{r})$ leads to the expression:

$$J_1 = J_2 = \int_0^r \int_0^{2\pi} (\tilde{r} \cos \theta)^2 \tilde{r} d\theta d\tilde{r} = \frac{\pi r^4}{4}. \quad (6)$$

Finally, J_3 corresponds to the polar moment:

$$J_3 = J_1 + J_2 = \frac{\pi r^4}{2}. \quad (7)$$

The constants $E, G > 0$ denote the Young and shear moduli of the material, respectively. $G = \frac{E}{2(1+\nu)}$, where ν is the Poisson's ratio.

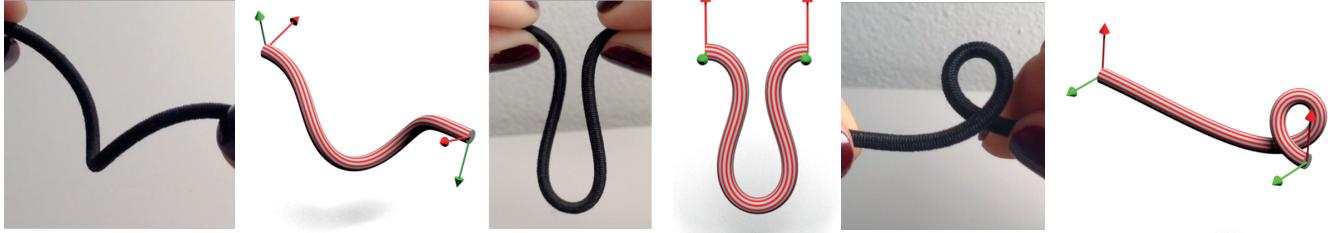


Figure 4: Comparison of our simulations to a real elastic rod using the same geometric and material parameters (defined in Table 1).

4. Rod discretization

To discretize the Cosserat theory, we uniformly sample the rod and obtain a piecewise linear curve with N points. Each element of the rod is defined by two points $\{\mathbf{x}_n, \mathbf{x}_{n+1}\}$ and one quaternion u_n , which represents the orientation of the material frame (Fig. 5). Hence, the sampled rod consists of $N - 1$ elements and $N - 1$ quaternions.

4.1. Discrete stretch and shear potential

The discretization of Eq. (1) is defined as

$$v_{SE} = \frac{l}{2} \sum_{n=1}^{N-1} \tilde{\boldsymbol{\Gamma}}_n^T C^T \tilde{\boldsymbol{\Gamma}}_n, \quad (8)$$

where l corresponds to the initial length of an element, assuming the polyline is uniformly sampled. The strain measure $\boldsymbol{\Gamma}_n$ is discretized as in Lang et al. [LLA11]:

$$\boldsymbol{\Gamma}_n(\mathbf{x}_n, \mathbf{x}_{n+1}, u_n) = \frac{1}{l} (\mathbf{x}_{n+1} - \mathbf{x}_n) - \text{Im}(u_n \mathbf{e}_3 \bar{u}_n), \quad (9)$$

where the tangent vector is discretized as $\partial_s \mathbf{r}_s \approx \frac{1}{l} (\mathbf{x}_{n+1} - \mathbf{x}_n)$ and only the imaginary part of the quaternion product is considered [KS16].

4.2. Discrete bend and twist potential

The discretization of Eq. (3) leads to:

$$v_{BT} = \frac{l}{2} \sum_{n=1}^{N-2} \boldsymbol{\Omega}_n^T C^Q \boldsymbol{\Omega}_n, \quad (10)$$

where $\boldsymbol{\Omega}_n$ is discretized with the finite quotient expression [LLA11] using the quaternion product, denoted by :

$$\boldsymbol{\Omega}_n(u_n, u_{n+1}) = \frac{2}{l} \text{Im}(\bar{u}_n \circ u_{n+1}). \quad (11)$$

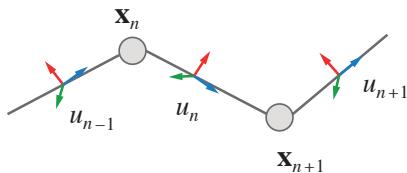


Figure 5: Rod discretization with points, elements and orientations representing the material frames.

5. Projective dynamics with angular momentum

Projective dynamics (PD) [BML*14] is a different way to express the implicit discretized equations for a nodal system by splitting the internal and external forces in the system into a local/global optimization problem. Simulating Cosserat rods with PD's standard formulation is not possible, as it solely preserves the linear momentum by updating the system's variables with linear velocities and forces. Given that Cosserat rods require keeping track of body orientations (Sec. 3), we overcome this limitation by generalizing the standard PD formulation by including the angular momentum term. With the proposed extension, the preservation of the angular momentum is a trade-off between all the constraints in the system. Note that we refer to this trade-off as the preservation of the angular momentum for simplicity. The new optimization procedure is summarized in Algorithm 1. In particular, we incorporate the rod's orientations as system variables

$$\mathbf{q} = [\mathbf{x}_1^T, \dots, \mathbf{x}_N^T, u_1, \dots, u_{N-1}]^T, \quad (12)$$

where \mathbf{q} holds both, the positions $\mathbf{x} \in \mathbb{R}^3$ and the element orientations $u \in \mathbb{R}^4$ (quaternions). Including orientations as system variables enables us to simulate rotational external forces such as torques (τ) using the body's inertia matrices (J) and angular velocities (ω) (see lines 3–4 in Algorithm 1).

As the first steps in Algorithm 1, both the linear and angular momenta ($s_v^{(t)}$ and $s_u^{(t)}$, respectively) are computed with an explicit integration scheme (lines 2–4), where t is the index of the current time step. For the angular momentum, we first compute the angular velocity $\mathbf{s}_\omega^{(t)}$ as a vector and then use it as the imaginary part of a normalized quaternion $s_\omega^{(t)}$, whose scalar coefficient is 0 [SM06]; h is the time step size. After the local/global iterative section (lines 7–10), both velocities are updated (lines 11–12) with the new system variables $\mathbf{q}^{(t+1)} = [\mathbf{x}^{(t+1)}^T, u^{(t+1)}]^T$. The angular velocity $\omega^{(t+1)}$ is derived using the temporal derivative for quaternions [Wit97, SM06] (line 12).

5.1. Local/global step

The optimization problem can be divided into a local step and a global step. In the local step (Algorithm 1, line 9), the optimization problem is solved w.r.t. a collection of constraints \mathbf{C}_i and is defined as in Bouaziz et al. [BML*14]. This local step is inherited from position based methods [MHHR07], where the positions are corrected according to certain desired constraints.

Algorithm 1 Extended projective implicit Euler solver

```

1: function SOLVER( $\mathbf{q}^{(t)}, \mathbf{v}^{(t)}, \boldsymbol{\omega}^{(t)}$ )
2:    $\mathbf{s}_x^{(t)} = \mathbf{x}^{(t)} + h\mathbf{v}^{(t)} + h^2 \mathbf{M}^{-1} \mathbf{f}_{\text{ext}}$ 
3:    $\mathbf{s}_{\boldsymbol{\omega}}^{(t)} = \boldsymbol{\omega}^{(t)} + h\mathbf{J}^{-1} [\mathbf{t} - \boldsymbol{\omega}^{(t)} \times (\mathbf{J}\boldsymbol{\omega}^{(t)})]$ 
4:    $s_u^{(t)} = u^{(t)} + \frac{1}{2}h(u^{(t)} \circ s_{\boldsymbol{\omega}}^{(t)})$ 
5:    $\mathbf{s}^{(t)} = [\mathbf{s}_x^{(t)}, s_u^{(t)}]^T$ 
6:    $\mathbf{q}^{(t+1)} = \mathbf{s}^{(t)}$ 
7: loop solverIteration times
8:   for all constraints  $i$  do
9:      $\mathbf{p}_i = \text{ProjectOnConstraintSet}(\mathbf{C}_i, \mathbf{q}^{(t+1)})$ 
10:     $\mathbf{q}^{(t+1)} = \text{SolveLinearSystem}(\mathbf{s}^{(t)}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \dots)$ 
11:     $\mathbf{v}^{(t+1)} = \frac{1}{h}(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)})$ 
12:     $\boldsymbol{\omega}^{(t+1)} = \frac{2}{h}(\bar{u}^{(t)} \circ u^{(t+1)})$ 
13: return  $\mathbf{q}^{(t+1)}, \mathbf{v}^{(t+1)}, \boldsymbol{\omega}^{(t+1)}$ 

```

In the global step (Algorithm 1, line 10), $\mathbf{q}^{(t+1)}$ is the least squares solution of the linear system

$$\left(\frac{\mathbf{M}^*}{h^2} + \sum_i w_i \mathbf{S}_i^T \mathbf{A}_i^T \mathbf{A}_i \mathbf{S}_i \right) \mathbf{q}^{(t+1)} = \frac{\mathbf{M}^*}{h^2} \mathbf{s}^{(t)} + \sum_i w_i \mathbf{S}_i^T \mathbf{A}_i^T \mathbf{B}_i \mathbf{p}_i. \quad (13)$$

This linear system consists of a sum of potentials: those preserving the linear and angular momenta, represented by $\mathbf{s}^{(t)} = [\mathbf{s}_x^{(t)}, s_u^{(t)}]^T$, and those defined per constraint with index i . The potentials defined per constraint are later derived from the Cosserat constraints (Sec. 6). The projection variables \mathbf{p}_i embed the potential defined per constraint computed in the local step. $\mathbf{A}_i, \mathbf{B}_i$ are constant matrices defined per constraint and \mathbf{S}_i is the selection matrix, which identifies the variables in \mathbf{q} involved in the constraint. w_i is the weight assigned per constraint, which we derive in Sec. 6.3. We define

$$\mathbf{M}^* = \begin{pmatrix} \mathbf{M} & \\ & \mathbf{J} \end{pmatrix}$$

as the concatenation of $\mathbf{M} \in \mathbb{R}^{3N \times 3N}$, the lumped mass matrix of the points in the polyline, and $\mathbf{J} \in \mathbb{R}^{4(N-1) \times 4(N-1)}$, the inertia matrix of the orientations in the polyline. \mathbf{J} is the concatenation of

$$\mathbf{J}_n = l\rho \text{diag}(0, J_1, J_2, J_3),$$

defined per orientation with index n . l is the distance between orientations, i.e., the length of the segment, ρ is the mass density, and J_1, J_2, J_3 are the moments of inertia (Eq. (6) and Eq. (7)). Note, the concatenation \mathbf{M}^* enables the preservation of linear and angular momenta, which is detailed in the following.

Potentials preserving linear and angular momenta are derived with the explicit integration scheme (Algorithm 1, lines 2–4), defined both for positions and orientations. We derive the angular momentum potential similar to that of the linear momentum potential [BML*14], but using orientations and angular velocities instead of only positions and linear velocities.

A potential for preserving the angular momentum is therefore defined as follows (see also the supplement document):

$$\min_{u^{(t+1)}} \frac{1}{2h^2} \|\mathbf{J}^{\frac{1}{2}}(u^{(t+1)} - s_u^{(t)})\|_F^2, \quad (14)$$

where $s_u^{(t)} = u^{(t)} + \frac{h}{2}(u^{(t)} \circ \boldsymbol{\omega}^{(t)}) + \frac{h^2}{2}u^{(t)} \circ [\mathbf{J}^{-1}[\mathbf{t} - \boldsymbol{\omega}^{(t)} \times (\mathbf{J}\boldsymbol{\omega}^{(t)})]]$ (Algorithm 1, lines 4, 3). Note that when \mathbf{J} is multiplied by a vector (e.g., $\boldsymbol{\omega}$) instead of a quaternion (e.g., u), we take the 3×3 matrix $\mathbf{J}'_n = l\rho \text{diag}(J_1, J_2, J_3)$.

6. Projective dynamics potentials

As stated in Sec. 5, the system variables $\mathbf{q}^{(t)}$ are updated within the global step according to the momentum potentials and the potentials defined per constraint. Cosserat theory defines two potentials governing the behavior of the rod: the stretch and shear potential on the one hand and the bend and twist potential, on the other hand, discretized in Eq. (8) and Eq. (10), respectively. We now formulate the PD constraints and potentials for both these measures and describe how they are incorporated into the local and global step. In this section, we drop the time step superscript (t) for simplicity.

6.1. Stretch and shear potential

The stretch and shear constraint \mathbf{C}_{SE} minimizes Cosserat theory stretch and shear deformations measured with the stretch strain $\Gamma_n(\mathbf{x}_n, \mathbf{x}_{n+1}, u_n)$, defined per rod element with index n (Eq. (9)). The corresponding stretch and shear potential is defined per i -th constraint and minimizes the constraint \mathbf{C}_{SE} by

$$W_{SEi}(\mathbf{q}, \mathbf{p}_i) = \frac{w_{SEi}}{2} \|\mathbf{A}_i \mathbf{S}_i \mathbf{q} - \mathbf{B}_i \mathbf{p}_i\|_F^2 + \chi^{\mathbf{C}_{SE}}(\mathbf{p}_i), \quad (15)$$

where $\mathbf{S}_i \mathbf{q} = [\mathbf{x}_{n+1}, \mathbf{x}_n, u_n]^T$ are the variables involved in the constraint i , selected from \mathbf{q} with the matrix \mathbf{S}_i . $\mathbf{A}_i, \mathbf{B}_i$ are constant matrices and \mathbf{p}_i are the projection variables. The indicator function $\chi^{\mathbf{C}_{SE}}(\mathbf{p}_i)$ formalizes the requirement that \mathbf{p}_i should lie in the constraint manifold \mathbf{C}_{SE} and w_{SEi} is the potential weight (Sec. 6.3).

The minimization of Eq. (15) w.r.t. the projection variables leads to the following optimization problem in the local step:

$$\min_{\mathbf{p}_i} W_{SEi}(\mathbf{q}, \mathbf{p}_i) = \min_{\mathbf{x}_f^*, u_n^*} \|\Gamma_n\|_F^2, \quad (16)$$

which can be reformulated through the free variables $\{\mathbf{x}_f^*, u_n^*\}$. The free variable $\mathbf{x}_f^* = \frac{1}{l}(\mathbf{x}_{n+1}^* - \mathbf{x}_n^*)$ represents the element's differential positions. $\Gamma_n = \mathbf{x}_f^* - \mathbf{d}_3^*$ denotes the Cosserat stretch and shear strain measure. The free variable $\mathbf{d}_3^* = R(u_n^*)\mathbf{e}_3$ represents the normal of the rod's cross-section (Sec. 3).

Note that the rotation with u_n^* introduces a non-linear relation between the free variables in Eq. (16). Thus, formulating the minimization problem in Eq. (16) through the matrices \mathbf{A}_i and \mathbf{B}_i in Eq. (15) is not straightforward. Given that positions and orientations are independent variables, we decouple Eq. (16) into one optimization problem for the positions, i.e.,

$$\min_{\mathbf{x}_{n+1}^*, \mathbf{x}_n^*} \left\| \frac{1}{l}(\mathbf{x}_{n+1}^* - \mathbf{x}_n^*) - \mathbf{d}_3 \right\|_F^2, \quad (17)$$

and a second optimization problem for the orientations, i.e.,

$$\min_{u_n^*} \|\mathbf{x}_f - R(u_n^*)\mathbf{e}_3\|_F^2. \quad (18)$$

The solution to the minimization problem in Eq. (17) is reached when $\mathbf{x}_f^* = \mathbf{d}_3$: the vector \mathbf{x}_f^* is aligned with \mathbf{d}_3 and has unit length, i.e., the element's length is the same as in the initial configuration.

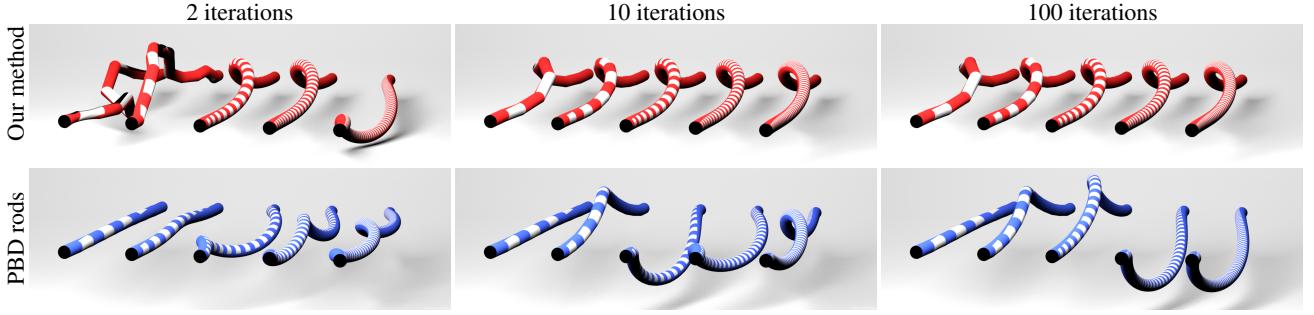


Figure 6: Comparison of twisted and compressed rods of different mesh resolutions. Rod elements are visualized by alternating colors. From 10 iterations onwards, our method converges to the same result no matter which resolution is used, which is in contrast to PBD rods [KS16].

Therefore, this optimization problem minimizes the stretch deformation, i.e., the length preservation of the element. The solution to Eq. (18) is attained when \mathbf{d}_3^* is aligned with \mathbf{x}_f . Hence, the optimal solution of the free variable is $u_n^* = u_n \circ \partial u_n$, where the orientation u_n is rotated by ∂u_n , ∂u_n being the differential rotation between the vectors \mathbf{d}_3 and \mathbf{x}_f . This optimization problem minimizes the shear deformation, i.e., the rotational difference between the cross-section normal \mathbf{d}_3 , and the tangent of the element, \mathbf{x}_f (Eq. (9)).

In PD, the projection variables are introduced in the local and global step through the matrices $\mathbf{A}_i, \mathbf{B}_i$, as formulated in Eq. (15) and Eq. (13). For this potential, these matrices are defined as

$$\mathbf{A}_i = \begin{bmatrix} \frac{1}{l}\mathbf{I}_3 & -\frac{1}{l}\mathbf{I}_3 & \mathbf{O}_{3,4} \\ \mathbf{O}_{4,3} & \mathbf{O}_{4,3} & \mathbf{I}_4 \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} \mathbf{I}_3 & \mathbf{O}_{3,4} \\ \mathbf{O}_{4,3} & \mathbf{I}_4 \end{bmatrix}, \quad \mathbf{p}_i = \begin{bmatrix} \mathbf{d}_3 \\ u_n^* \end{bmatrix}.$$

These matrices have two rows, one per each of the minimization problems formulated in Eq. (17) and Eq. (18); \mathbf{p}_i embeds the projection variables derived in the local solve; \mathbf{I}_k denotes a $k \times k$ identity matrix, and $\mathbf{O}_{k,m}$ denotes a $k \times m$ zero matrix.

6.2. Bend and twist potential

The bend and twist constraint \mathbf{C}_{BT} minimizes Cosserat theory bend and twist deformations measured with the twist strain $\Omega_n(u_n, u_{n+1})$, defined per rod element with index n (Eq. (11)). The corresponding bend and twist potential is defined per i -th constraint and minimizes the constraint \mathbf{C}_{BT} by

$$W_{BTi}(\mathbf{q}, \mathbf{p}_i) = \frac{w_{BTi}}{2} \|\mathbf{A}_i \mathbf{S}_i \mathbf{q} - \mathbf{B}_i \mathbf{p}_i\|_F^2 + \chi^{\mathbf{C}_{BT}}(\mathbf{p}_i), \quad (19)$$

where $\mathbf{S}_i \mathbf{q} = [u_n, u_{n+1}]^\top$ are the variables involved in the constraint, the adjacent quaternions, which are selected from \mathbf{q} with the selection matrix \mathbf{S}_i and \mathbf{p}_i are the projection variables.

The minimization of Eq. (19) w.r.t. the projection variables leads to the following optimization problem in the local step:

$$\min_{\mathbf{p}_i} W_{BTi} = \min_{u_n^*, u_{n+1}^*} \|\Omega_n\|_F^2, \quad (20)$$

which can be reformulated through the free variables $\{u_n^*, u_{n+1}^*\}$. Ω_n denotes the curvature vector, i.e., the relative curvature between adjacent quaternions. The solution to the minimization problem is reached when the relative curvature Ω_n between the adjacent ori-

entations is 0. The optimal solution to Eq. (20) is derived as:

$$u_n^* = u_n \circ \frac{\Omega_n}{2} \quad \text{and} \quad u_{n+1}^* = u_{n+1} \circ \frac{\bar{\Omega}_n}{2}, \quad (21)$$

where \circ denotes a quaternion product. The current orientations u_n and u_{n+1} are rotated with halfway of the curvature vector and its conjugate, respectively. With this solution, the resulting orientations u_n^* and u_{n+1}^* have the same direction, minimizing the relative curvature $\Omega_n = 0$ between them.

In this formulation, the curvature vector is defined as $\Omega_n = Im(\bar{u}_n \circ u_{n+1})$. This expression is not scaled by $\frac{2}{l}$, as opposed to Eq. (11), given that scaling a minimization problem by a scalar leads to the same result. Instead, the potential in Eq. (19) is scaled by w_{BTi} , further discussed in Sec. 6.3.

In PD, the projection variables are introduced in the local and global step through the matrices $\mathbf{A}_i, \mathbf{B}_i$, as formulated in Eq. (19) and Eq. (13). For this potential, these matrices are defined as $\mathbf{A}_i = \mathbf{B}_i = \mathbf{I}_8$, where \mathbf{I}_k is a $k \times k$ identity matrix.

6.3. Potential weight formulation

The potential formulations in the discretized Cosserat theory (v_{SE}, v_{BT}) in Eq. (8) and Eq. (10) are defined by the product of the strain measures (Γ_n, Ω_n) with certain weight matrices (C^Γ, C^Ω). The weight matrices depend on some material parameters such as the Young's modulus E or the radius r of the rod. Additionally, the discrete strain measures are scaled by the length of the segment l .

In the following, we formulate PD's potential weights such that the potentials are equivalent to the ones formulated in Cosserat theory. This enables us to compare our simulations to a reference solution generated with finite differences, which is parametrized with variables such as E, r or the mass density (Sec. 7.2.3).

For the stretch and shear potential, the weight w_{SEi} in Eq. (15) is formulated as:

$$w_{SEi} = EA_3 l, \quad (22)$$

where $A_3 = \pi r^2$ is the area of the cross-section. In this formulation, we assume that the three components on the weight matrix C^Γ are scaled by the constant E , i.e., Young's modulus, as opposed to the formulation in Eq. (5), where some of the components of the matrix

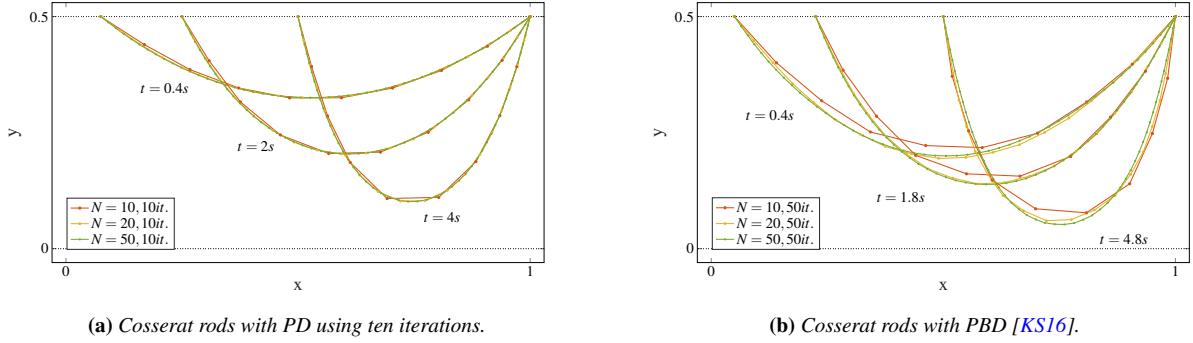


Figure 7: Motion simulations of a hanging rod, under the gravitational force and for different mesh resolutions. There are three simulation snapshots for each simulation, where the right endpoint is fixed, and the left endpoint is moved in parallel to the x -axis. The number of iterations in the PBD method are adjusted such that it has the same computation time as the PD simulation. Although PBD presents similar motion for the different resolutions, PD converges faster to a mesh independent solution.

are instead scaled by the shear modulus G . With this assumption, we neglect the Poisson ratio ν in this weight.

The reason for assuming a uniform scaling is that in PD, potentials are defined as the Frobenius norm of a certain deformation (Eq. (15)). In our potentials, the deformations are vectors. Its Frobenius norm is a scalar, and hence the weight w_{SEi} in Eq. (15) needs to be a scalar as well.

Our formulation of the weight is additionally scaled by the constant l , given that the expression of the discretized potential is also proportional to this constant (Eq. (8)).

For the bend and twist potential, the weight w_{BTi} in Eq. (19) is formulated as:

$$w_{BTi} = \frac{4GJ_3}{l}, \quad (23)$$

where $J_3 = \frac{\pi r^4}{2}$ is the expression of the polar moment of inertia (Eq. (6)). In this formulation we again assume a constant scaling, by the shear modulus $G = \frac{E}{2(1+\nu)}$, being ν the Poisson ratio.

The strain measure Ω_n is scaled by $\frac{2}{l}$ (Eq. (11)). The potential \mathbf{v}_{BTi} in Eq. (10) is defined by the product $l\Omega_n^\top C^\Omega \Omega_n$, which leads to the formulated weight $w_{BTi} = l^2(GJ_3)\frac{2}{l}$, and therefore the simplified expression in Eq. (23).

Note that the formulation of w_{BTi} is divided by l , as opposed to the formulation of w_{SEi} . In later sections, we demonstrate that the formulation of these potential weights ensures mesh independence in our simulations within a few iterations (Sec. 7). Experiments show that modifying the weights affects the convergence rate of the solver (see supplement document). Given that with already 2 to 10 iterations the system gives similar results to finite element methods (Sec. 7), we consider that it is worth using these weights.

7. Mesh independence and convergence study

In this section, we compare some of our simulations to previous work in rod simulation. We implemented Cosserat Rods in

PBD [KS16] following the available implementation [Ben] and apply constraint weights as defined in [KS16]. The concurrent publication by Deul et al. [DKWB18] combines a previous publication on Cosserat rods with PBD [KS16] together with XPBD [MMC16]. Once the official implementation is released, it would be interesting to compare the behavior of their stiff rods with our method. Further previous methods [CBD13, BWR^{*}08, BAV^{*}10] are compared with implementations provided by the authors. Table 1 summarizes the simulation parameters of our simulations.

7.1. Mesh independence in motion

Fig. 8 and Fig. 7 show time snapshots of rods simulated at different resolutions. Both scenarios initialize the rod along the x -axis, with two endpoints attached (Fig. 7) and only one attachment (Fig. 8). Both scenarios have a gravitational force and the left endpoint in the former scenario shown in Fig. 7 is displaced gradually. PD simulations (Fig. 7a and Fig. 8a) indicate the same characteristic motion for different resolutions by only applying ten iterations.

The same setup is implemented for the PBD method [KS16]. The number of iterations used in PBD is chosen such that both simulations are executed in the same computation time. Note, Fig. 7b and Fig. 8b indicate the number of iterations used for PBD. Note, in PBD, the number of iterations affects the material properties, i.e., the higher the number of iterations, the stiffer the material gets. Thus, when using PBD, we use a constant number of iterations when comparing different resolutions. Although Fig. 7b shows rods with similar behavior, the rod dynamics in Fig. 8b are not equivalent. Within this test, we demonstrate how PD converges faster, in terms of computation time, to a mesh independent solution. Please refer to the accompanying video for further details.

7.2. Static analysis and convergence to FEM

In this section, we compare PD simulations to a reference solution in the rest state. As common in the math literature (e.g., [LLA11]), we realize this reference solution by performing a high-resolution FEM simulation using the software Abaqus [HS02], where the rod

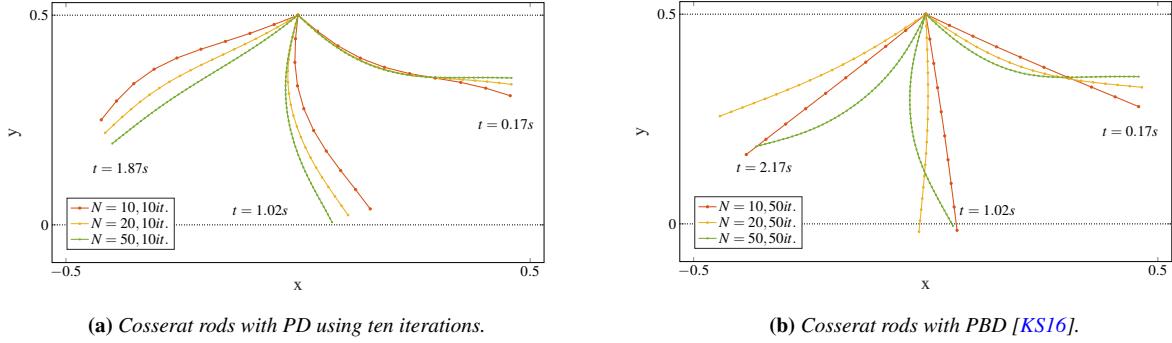


Figure 8: Motion simulations of a hanging rod with one endpoint attached, under the gravitational force and for different mesh resolutions. The number of iterations in PBD is chosen, such that PBD and PD execute the simulation in the same computation time. It can be seen that rods simulated with PD have an equivalent motion, as opposed to the ones simulated with PBD.

is discretized with B21 Timoshenko beams. A Cosserat rod can be considered as the geometrically nonlinear generalization of a Timoshenko rod, allowing to model extension and shearing apart from bending and twist, as opposed to Kirchhoff rods. All the simulations in *Abaqus* are implemented with the explicit procedure using 990 elements. The equations of motion are integrated using the explicit central difference integration rule (*Abaqus* 6.14 Theory Guide, Section 2.4.5). The time step is defined automatically by the software according to the stability conditions of each simulation.

Since the equations of motion used in our model are solved differently than the ones used in the reference solution, we can only provide comparisons of simulations in static equilibrium, which is reached when the rods do not undergo further motion. Comparing exact motion is beyond the scope of this section.

Simulations with a small Young's modulus and a small radius result in a highly elastic behavior. Such elastic materials present high frequency vibrations in the *Abaqus* simulations, which we damp with the bulk viscosity parameter (we use Linear bulk viscosity parameter = 0.07, Quadratic bulk viscosity parameter = 1.3). Additionally, we damp the motion with the α dampening coefficient in the material properties (we use $\alpha = 0.8$). Thus, damping is an issue when comparing to a reference solution. The damping models used in both methods are different and therefore an exact position correspondence in both simulations is difficult to achieve. For this reason, in order to compute the convergence of our simulation in respect to a reference solution (Fig. 10), the rod's potential is used as an error measure, instead of taking the positional difference.

7.2.1. Stretch and bending constraints

Fig. 9 shows the simulations at static equilibrium for different mesh resolutions and for a different number of iterations. Increasing the number of points on the polyline converges to the FEM simulation. As observed in Fig. 9, the solution converges already for 21 points, and proves to be independent on the number of iterations. This experiment is the final state of the simulation in Fig. 7. After initializing the rod along the x -axis with two endpoints attached and gravitational force, one of the endpoints is gradually displaced.

7.2.2. Twist constraint

For this simulation, the rod is initialized along the x -axis with two attached endpoints $x_1 = 0$ and $x_N = L$. The endpoint orientations are additionally fixed. The material frame u_1 is fixed to $u_1 = [1, 0, 0, 0]^\top$, the unit quaternion. The material frame u_{N-1} is initialized with the unit quaternion and gradually rotated to realize a twist of 360° . Figures 10a, 10b show how the simulations at the rest state converge towards the *Abaqus* solution ($N = 991$) with an increase in the number of samples.

Fig. 10c shows the error to the reference simulation of the experiment in Figs 10a, 10b. The error measure is the difference of the total bending potential W_{BT} (see Eq. (10)) computed for each resolution towards the one obtained in the FEM simulation. For each constraint defined along the rod, a scalar $W_{BT,i}$ is computed. The total bending potential is the sum of all these values. The convergence error, in log scale, has a linear trend with slope 2. The error w.r.t. the reference solution reduces quadratically when increasing the degrees of freedom by a factor of 10.

The remaining plots in Fig. 10c illustrate the accumulated potential per constraint $W_{BT,i}$ along the rod, starting from the endpoint $x_1 = 0$ until the endpoint $x_N = 1$. All the plots include the reference value of the accumulated W_{BT} along the rod, simulated with

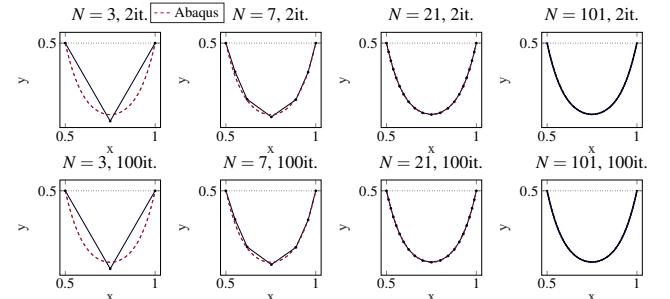


Figure 9: Rods simulated with our method converge to the solution generated by *Abaqus* for different number of iterations.

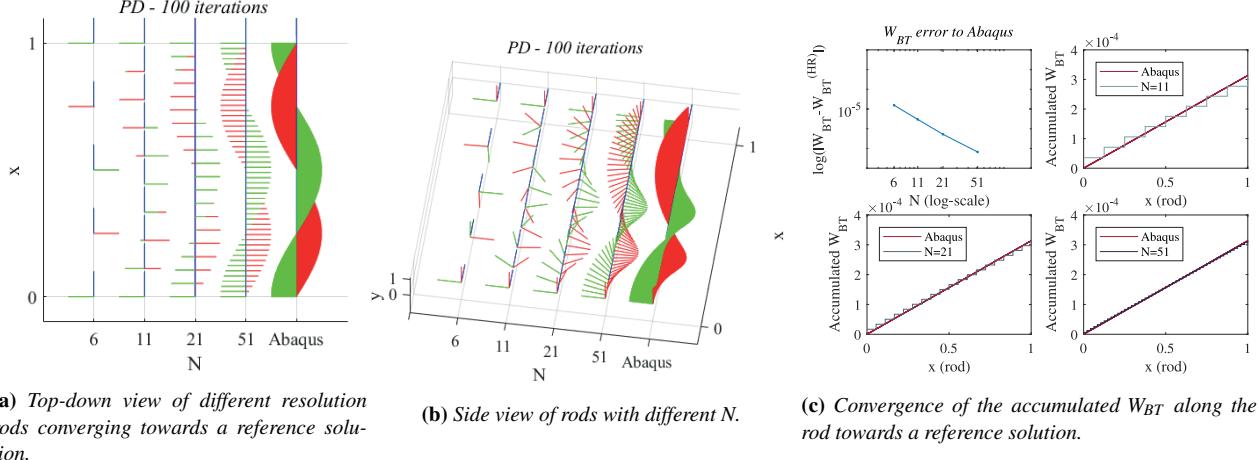


Figure 10: Convergence analysis to a FEM reference solution (Abaqus) of a 360° twist applied on the endpoint orientation u_{N-1} .

Abaqus. Fig. 10c shows how the potential of the endpoint x_N converges to the reference solution when refining the simulation mesh.

7.2.3. Bending behavior comparison

Fig. 11 shows similar bending behavior between our simulations and the ones obtained with Abaqus for different radii and Young's modulus E . For this simulation, the rod is initialized along the x -axis with two attached endpoints at $x_1 = 0, x_N = L$, under the effect of gravitational force. The endpoint x_1 is displaced until $x_1 = L$.

7.3. Out-of-plane curl effect comparison

This simulation validates both constraints at the same time, as opposed to the separate analysis from previous sections. The rod is initialized along the x -axis, with both endpoints $x_1 = 0$ and $x_N = L$ attached. Twists of 120° and -120° are applied gradually to the orientation frames u_1 and u_{N-1} , respectively. After some frames, the rod is slightly compressed by displacing both endpoints gradually towards the center of the rod. This results in an out-of-plane

curl (Fig. 6), a common feature in rod simulation [vdHT00]. The same experiment is executed with $\{2, 10, 100\}$ iterations and different mesh resolutions $N = \{10, 20, 50, 100, 200\}$. As opposed to the results obtained with the [KS16] method, the curls obtained with PD are mesh independent after 10 iterations (Fig. 6).

7.3.1. Previous work comparison

To compare to further previous work, we initialize the rod along the x -axis and attach the left endpoint. With the effect of gravity, we let the simulation run until there is no apparent motion. Fig. 12 shows the rest pose of the simulation for rods simulated with Super Space Clothoids (SSC) [CBD13], (implicit) Discrete Elastic Rods (DER) [BWR*08, BAV*10] and the reference FEM simulation generated with Abaqus. The implicit Euler integration nature of PD introduces some artificial damping. Although an exact correspondence to these methods is difficult to achieve, our method shows converging bending properties.

7.3.2. Self-collisions

Our formulation includes the detection and response to self-collisions, adapted from the edge-edge distance constraint implemented with PBD [Ben] and formulated for projective dynamics. Fig. 1 shows plectoneme formation after torsional deformation. Interactively displacing the rod endpoints enables knot formation.

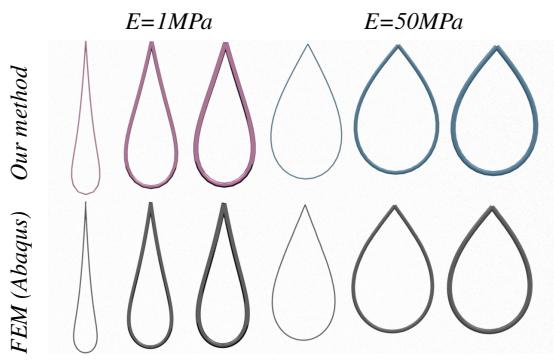


Figure 11: Bending behavior comparison to a FEM reference solution (Abaqus) for different geometric and material parameters. Different colors on the rods denote different Young's Modulus E .

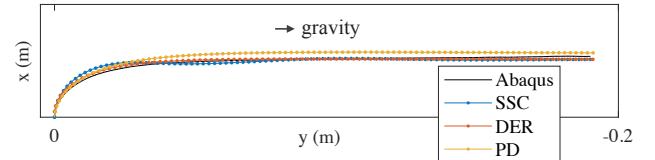


Figure 12: Final state of a hanging rod under the gravitational force solved for different methods.

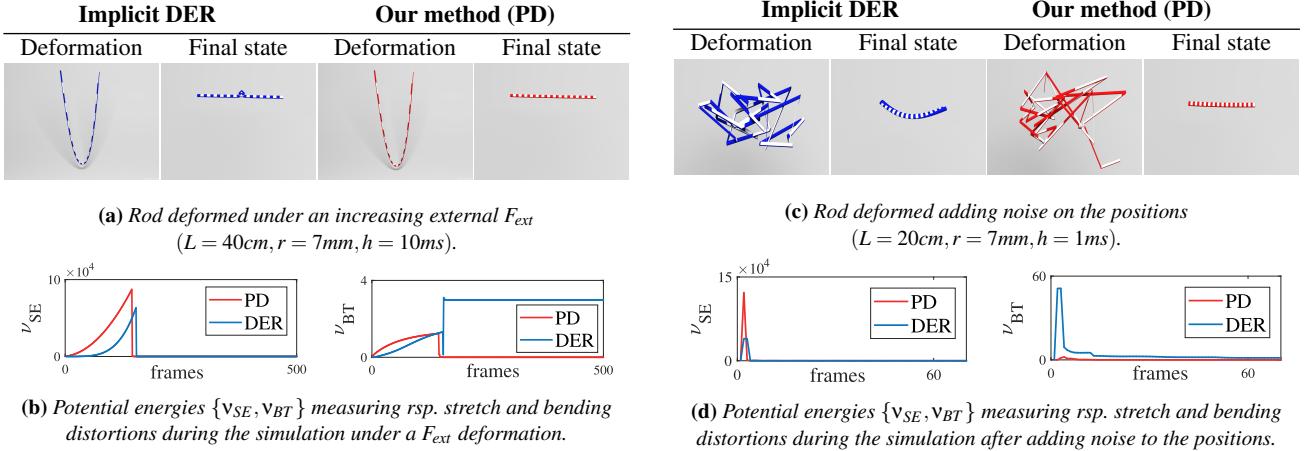


Figure 13: Robustness comparison for different distortions between (implicit) Discrete Elastic Rods (DER) and our method (PD).

7.3.3. Robustness

Fig. 13 compares the robustness of our method with (implicit) Discrete Elastic Rods (DER) [BWR^{*}08, BAV^{*}10] for different deformations. Furthermore, it evaluates how the methods recover to the initial state, which is a rod initialized over the x -axis. Fig. 13b shows how the stretching deformation v_{SE} increases when the external force is gradually stronger. Once the external force is set to zero, the rods simulated with PD recover to its initial state, leading to an undeformed rod ($\{v_{SE}, v_{BT}\} = 0$). In contrast, the rods simulated with (implicit) DER remain deformed ($v_{BT} \neq 0$). Adding noise to the initialized positions (Fig. 13c) proves the recovery capabilities of the method after a large deformation. Fig. 13c shows how the rods simulated with implicit DER remain twisted at the end of the simulation, as opposed to the PD simulation (Fig. 13d).

7.4. Limitations

The computation time for the hanging rods simulation (Fig. 8) with PD ($1\text{ms}/\text{frame}$ for 10 points and $5.7\text{ms}/\text{frame}$ for 50 points) is close to PBD methods ($0.6\text{ms}/\text{frame}$ for 10 points and $5\text{ms}/\text{frame}$ for 50 points), both with 10 iterations. However, for other simulation scenarios, PBD is currently faster. This is due to the way we set up the linear system in the global step, using non-optimized multiplication of sparse matrices (using the library *Eigen*), which is not used in the PBD simulations. Additionally, the linear system in the global step is currently solved using Cholesky decomposition. The left-hand side matrix of the linear system can be prefactored when the constraints remain constant during the simulation [BML^{*}14]. However, in our simulations, we refactorize the left-hand side matrix every time we displace the attachment constraints, which introduces a significant overhead. We implement the attachments as Dirichlet boundary conditions. After every update, the value of the constrained variables is subtracted from the left-hand side matrix, followed by the matrix refactorization.

We are currently working on improving the computation time of our method by using a different solver, which can be executed in parallel [Wan15, FTP16]. With the next version of our solver, we expect a similar performance to other position-based methods.

8. Conclusions and future work

In this paper, we present a projective dynamics (PD) solver, which allows the preservation of angular momentum. This formulation enables the simulation of primitives from Cosserat theory, which assigns an orientation to each point in a material and thus requires the simulation of rotations. We further discretize the continuous potentials from Cosserat theory applied to rods, and define corresponding PD potentials for the global step, in addition to constraints for the local step. Additionally, we formulate the potential weights in terms of material parameters such as radius and Young's modulus, which are key to provide realistic and controllable simulations. We show, that a PD rod converges, under refinement, to the characteristic behavior of a high-precision finite element analysis. Further, we provide examples, which verify mesh independence of our proposed formulation. This property enables several avenues for future work. For instance, a multi-grid rod representation could speed up the rod simulation. Alternatively, local refinements could be applied to regions of interests which require a higher resolution, such as regions where knots are being tied. By comparing our simulations to a FEM reference solution, we demonstrate the capability of our formulation to simulate different elasticity values in a realistic manner. We additionally demonstrate correspondence with reality by comparing to a real elastic rod (Fig. 4).

Fig.	7a, 9	8a	10	11	6	4	12
r	3	8	3	$\{2, 6, 8\}$	3	1.5	0.2
ρ	1.3	4.3	1.3	1.3	1.3	1.5	3
E	1	1	1	$\{1, 50\}$	500	100	100
L	1	0.5	1	1	0.2	0.7	0.2
h	1	1	1	1	1	10	1
g_y	9.861	9.861	0	9.861	0	9.861	9.861

Table 1: Simulation parameters for our experiments. Here, r is radius (mm), ρ is mass density (g/m^3), E is Young's modulus (MPa), L is length (m), h is time step (ms) and $-g_y$ is the gravity (m/s^2).

References

- [BAV*10] BERGOU M., AUDOLY B., VOUGA E., WARDETZKY M., GRINSPUN E.: Discrete Viscous Threads. *ACM Trans. Graph. (Siggraph)* (2010). [1](#), [2](#), [7](#), [9](#), [10](#)
- [Ben] BENDER J.: PositionBasedDynamics Online Implementation. <https://github.com/InteractiveComputerGraphics/PositionBasedDynamics/>. Accessed: 2017-09-17. [7](#), [9](#)
- [BML*14] BOUAZIZ S., MARTIN S., LIU T., KAVAN L., PAULY M.: Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph. (Siggraph)* (2014). [2](#), [3](#), [4](#), [5](#), [10](#)
- [BWR*08] BERGOU M., WARDETZKY M., ROBINSON S., AUDOLY B., GRINSPUN E.: Discrete Elastic Rods. *ACM Trans. Graph. (Siggraph)* (2008). [1](#), [2](#), [3](#), [7](#), [9](#), [10](#)
- [CBD13] CASATI R., BERTAILS-DESCOUBES F.: Super Space Clothoids. *ACM Trans. Graph. (Siggraph)* (2013). [2](#), [7](#)
- [Dil92] DILL E. H.: Kirchhoff's Theory of Rods. *Archive for History of Exact Sciences* (1992). [2](#)
- [DKWB18] DEUL C., KUGELSTADT T., WEILER M., BENDER J.: Direct Position-Based Solver for Stiff Rods. *Computer Graphics Forum (Eurographics)* (2018). [1](#), [3](#), [7](#)
- [FTP16] FRATARCANELI M., TIBALDO V., PELLACINI F.: Vivace: A practical gauss-seidel method for stable soft body dynamics. *ACM Trans. Graph. (Siggraph)* (2016). [10](#)
- [HS02] HIBBITT K., SORENSEN: *ABAQUS/CAE User's Manual*. Hibbit, Karlsson & Sorensen, Incorporated, 2002. [3](#), [7](#)
- [Kir59] KIRCHHOFF G.: Über das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes. *Journal für die Reine und Angewandte Mathematik* (1859). [2](#)
- [KS16] KUGELSTADT T., SCHOEMER E.: Position and Orientation Based Cosserat Rods. *Eurographics/SIGGRAPH Symposium on Computer Animation* (2016). [1](#), [3](#), [4](#), [6](#), [7](#), [8](#), [9](#)
- [LLA11] LANG H., LINN J., ARNOLD M.: Multi-Body Dynamics Simulation of Geometrically Exact Cosserat rods. *Multibody System Dynamics* (2011). [2](#), [3](#), [4](#), [7](#)
- [MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position Based Dynamics. *Journal of Visual Communication and Image Representation* (2007). [2](#), [4](#)
- [MMC16] MACKLIN M., MÜLLER M., CHENTANEZ N.: XPBD: Position-based Simulation of Compliant Constrained Dynamics. *Proceedings of ACM Motion in Games* (2016). [2](#), [7](#)
- [Nak] NAKAGAWA N.: XPBD Online Implementation. <https://github.com/nobuo-nakagawa/xpbd/>. Accessed: 2018-03-28. [3](#)
- [Pai02] PAI D. K.: STRANDS: Interactive Simulation of Thin Solids using Cosserat Models. *Computer Graphics Forum* (2002). [2](#)
- [Rub13] RUBIN M. B.: *Cosserat Theories: Shells, Rods and Points*. Springer Science & Business Media, 2013. [2](#)
- [Sim85] SIMO J.: A Finite Strain Beam Formulation. The Three-dimensional Dynamic Problem. Part I. *Computer Methods in Applied Mechanics and Engineering* (1985). [3](#)
- [SM06] SCHWAB A. L., MEIJARD J.: How to Draw Euler Angles and Utilize Euler Parameters. *Proceedings of IDETC/CIE* (2006). [4](#)
- [ST07] SPILLMANN J., TESCHNER M.: CORDE: Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects. *Eurographics/SIGGRAPH Symposium on Computer Animation* (2007). [2](#), [3](#)
- [USS14] UMETANI N., SCHMIDT R., STAM J.: Position-based Elastic Rods. *Eurographics/SIGGRAPH Symposium on Computer Animation* (2014). [3](#)
- [vdHT00] VAN DER HEIJDEN G., THOMPSON J.: Helical and Localised Buckling in Twisted Rods: A Unified Analysis of the Symmetric Case. *Nonlinear Dynamics* (2000). [9](#)
- [Wan15] WANG H.: A Chebyshev Semi-iterative Approach for Accelerating Projective and Position-based Dynamics. *ACM Trans. Graph. (Siggraph)* (2015). [10](#)
- [Wit97] WITKIN A.: Physically Based Modeling: Principles and Practice – Rigid Body Dynamics, Lecture Notes I. *Computer Graphics* (1997). [4](#)