Rest Shape Optimization for Sag-Free Discrete Elastic Rods

Tetsuya Takahashi[1] [ID] and Christopher Batty[2] [ID]

[1]Tencent America, USA
[2]University of Waterloo, Canada

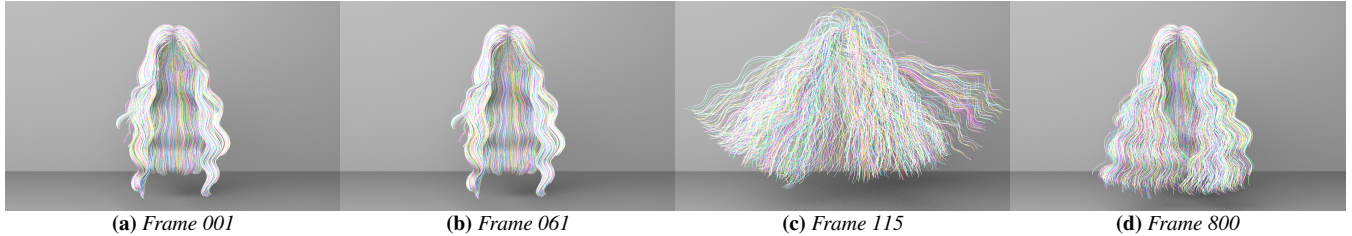**(a)** *Frame 001* **(b)** *Frame 061* **(c)** *Frame 115* **(d)** *Frame 800*

Figure 1: *Our rest shape optimizer enables sag-free simulation of discrete-elastic-rod-based hair strands. In this scene where the motions of the root vertices are driven by a rotating sphere, our method preserves the carefully designed hair style without stiffening the strands, and achieves natural hair dynamics under gravity. We use 1.9K strands, and each strand is discretized with 100 vertices. Our rest shape optimization took 11.2 s (with 7.6 Gauss-Newton iterations on average for each strand) while forward simulation (without collision handling) took 3.3 s per frame.*

Abstract

We propose a new rest shape optimization framework to achieve sag-free simulations of discrete elastic rods. To optimize rest shape parameters, we formulate a minimization problem based on the kinetic energy with a regularizer while imposing box constraints on these parameters to ensure the system's stability. Our method solves the resulting constrained minimization problem via the Gauss-Newton algorithm augmented with penalty methods. We demonstrate that the optimized rest shape parameters enable discrete elastic rods to achieve static equilibrium for a wide range of strand geometries and material parameters.

CCS Concepts
• ***Computing methodologies*** → *Physical simulation;*

1. Introduction

From human hairs to cables for electronic devices, elastic strands can be found everywhere in our daily lives, and modeling such thread-like materials is of critical importance in various domains, including computer graphics, mechanical engineering, and medical applications. The Discrete Elastic Rods (DER) formulation [BWR*08, BAV*10] is one of the most popular approaches to accurately and efficiently simulate such elastic strands and has demonstrated its effectiveness in a wide range of applications involving both forward simulations [CAR*09, DKH10, ACB*13, KTS*14, SJM17, FBGZ19, LSD*22, Dav23] and inverse problems [PTC*15, ZCT16, POT17, SMGT18, MPI*18, XKCB18, PKLI*19, NPTB22, RKP*22, CZB*24].

When modeling and designing elastic strands, a well-known problem is that such strands will immediately sag due to gravity when the simulation starts [TKA11, HWP*23], which ruins the specific strand shapes (e.g., hairstyles) that were either carefully designed by users, technical artists, and designers [IBB19, Liu22] or captured from real human hair or wigs [LLR13]. This issue occurs because the simplest way to model the strands is to focus solely on the geometry, neglecting the role of forces or dynamics; that is, the strand's initial shape is treated as its rest shape [KE22]. While one can largely mitigate this sagging effect by significantly (and artificially) increasing the material stiffness, this approach makes the elastic strands excessively stiff and sacrifices their natural dynamic response, thus negatively impacting the visual result.

To address this sagging problem in the DER framework, we propose a rest shape optimization approach that optimizes the rest shape parameters (rest length, rest curvature, and rest twist) of the input strands such that the input shape is preserved at static equilibrium under simulation (i.e., the strand shape deformed due to simulation coincides with the input shape). To find optimal rest shape parameters, we formulate a minimization problem based on the kinetic energy. Given the greater number of degrees of freedom (DOFs) for the rest shape parameters compared to the DOFs of DER strands, we incorporate a regularizer to avoid creating underdetermined systems. In addition, we impose box constraints on the rest shape parameters to prevent significant changes from their original values, ensuring the stability of the system. We solve the resulting constrained nonlinear minimization problem via the Gauss-Newton (GN) algorithm (and thus without the Hessian of the DER objectives) augmented with penalty methods and line search. Fig. 1 demonstrates the efficacy of our rest shape optimization.

2. Related Work
2.1. Elastic Rod Simulation
Due to the one-dimensional structure of thin elastic strands, it is typically difficult to accurately simulate these materials with general finite element methods (FEM) designed for volumetric structures [KE22], and specialized approaches have been investigated. Given that common strand-like materials exhibit higher stiffness in stretching modes compared to bending and twisting, early work assumed inextensibility via e.g., constraints based on Lagrange multipliers [GHF*07] with linear-time factorization and triangu-

lar solves [Bar96], a tridiagonal matrix formulation [HH13], and geometric stiffness [TNGF15, ATK17], reduced multi-body solvers [Had06, Fea16], and position-based dynamics (PBD) [MHHR07, BMM17]. To support the characteristic dynamics of elastic strands, the traditional mass-spring framework [LBOK13] was extended with altitude springs [SLF08], bending springs [IMP*13], torsion energy [SWP*23], and ghost rest shapes [HZS*24]. To simulate one-dimensional strands along with other deformables in a unified way, supporting torsional effects is critical, and therefore discretization elements that can capture twists (e.g., oriented particles [MC11] and elastons [MKB*10]) have also been presented. For highly constrained strand-like systems, further specialized approaches have been proposed with, e.g., Eulerian nodes [SJLP11, SBRBO20], long range constraints [MCMJ17], and unilateral distance constraints [MCJM18].

Since the introduction of the Cosserat theory into graphics [Pai02], it has been adapted to dynamical systems of strands [ST07] and network structures [ST09], combined with PBD using ghost particles [USS14], and further extended to support edge rotations via quaternions in the PBD framework [KS16]. The work of Kugelstadt and Schömer [KS16] has been extended to remove the ambiguity in the Darboux vector [HWP*23], to handle stiff problems via compliant constraints while utilizing the tree structures [DKWB18], to preserve the volume of elastic strands [ARM*19], and to achieve consistent dynamics [SMSH18] using projective dynamics [BML*14]. Elastic rod simulation based on the Cosserat theory [ST07] has also been extended to support rh-adaptivity [WCU*20] and a compact representation of the DOFs [ZLW*22]. A curvature-based reduced representation has also been presented by Bertails et al. [BAC*06] and extended to generate smoother dynamics via clothoids [CBD13].

While various approaches have been presented for simulating elastic strands, DER [BWR*08, BAV*10], which discretizes an elastic strand with vertex positions and edge angles in maximal coordinates, is one of the most popular approaches due to its simplicity, efficiency, and flexibility, as employed in a wide range of applications. As such, we use DER as our elastic strand simulator.

2.2. Sag-Free Simulation
Sagging of elastic materials at the onset of simulation has been a notorious and challenging problem; resolving it requires ensuring static equilibrium during the modeling, design, and fabrication of such materials. Thus, various methods have been proposed to achieve sag-free simulation. We classify these methods into the following three categories: nonlinear force solve, constrained minimization, and global-local initialization approaches.

2.2.1. Nonlinear Force Solve
Solving Newton's second law of motion for (typically nonlinear) forces and zero acceleration is equivalent to achieving zero net force and thus static equilibrium. Hadap [Had06] proposed using inverse dynamics [Fea16] to solve the nonlinear equation on the forces and thus enable sag-free simulation. While inverse dynamics works for reduced multibody systems, in general, it is not clear how to apply it for general simulation systems represented in maximal coordinates (e.g., FEM-based deformables and DER).

The nonlinear equations on the forces can be solved, e.g., with Newton-type optimizers [WWY*15, MWW18], although Iben

[IBB19] aimed to cancel gravity using spring forces via local analytical solutions within the mass-spring systems [IMP*13]. However, in practice, it is necessary to consider a global system to fully cancel gravity. A similar technique was also employed for sag-free mass-spring systems by introducing additional artificial springs (which can potentially introduce unnatural forces) [HZS*24].

Chen et al. [CZXZ14] employed an Asymptotic Numerical Method (ANM), which is a homotopy approach to solving nonlinear equations, for inverse elastic shape design with incompressible neo-Hookean materials. They demonstrated orders of magnitude faster convergence than one of the Newton-type optimizers, Levenberg–Marquardt algorithm (LMA) [NW06]. Later, Jia [Jia21] extended ANM to support more general material types and also demonstrated its efficiency for forward simulation compared to optimization-based integrators [MTGG11, GSS*15] combined with Newton's method [NW06] (although ANM can yield solutions that are local maxima, unlike those generated by the optimization-based integrators). ANM was also employed to investigate the equilibrium of elastic rods [LMR13]. While ANM can be more efficient than the Newton-type optimizers [NW06], it is not clear how to support box constraints, which are indispensable to constrain rest shape parameters within physically valid ranges and avoid introducing stability issues.

While canceling gravity forces in DER has been previously attempted by solving force equations or equivalent minimization [AX15, LSD*22], their approach lacked algorithmic and formulation specifics, such as exact objectives and DER force Jacobians with respect to the rest shape parameters. Additionally, their approach did not support box constraints which are critical to avoid significant rest shape changes and thus ensure stable simulations.

2.2.2. Constrained Minimization
Derouet-Jourdan et al. [DJBDT10] presented a method for achieving static equilibrium of elastic strands, simulated with the curvature-based discretization [BAC*06], in 2D. They formulated an objective consisting of curvatures and material parameters and then optimized these parameters via linear solves on the curvature and stiffening/lightening of materials. Derouet-Jourdan et al. [DJBDDT13] also achieved static equilibrium of curvature-based hair strands [BAC*06] in 3D under frictional contacts. Along the same lines, inverse approaches for curvature-based elastic strands have been further investigated [BDDJRL18, CBDNR20, HB21, HB23].

Twigg and Kačić-Alesić proposed optimizing rest shape parameters via force-norm minimization to achieve static equilibrium of mass-spring systems [TKA11]. Our work is most closely related to theirs, so we clarify the differences in detail in Sec. 2.2.4. Skouras et al. [STBG12] presented an optimization approach that treats the nonlinear force equations as hard constraints and solves the optimization via the augmented Lagrangian method [NW06]. Similar optimization problems under hard constraints have been proposed and efficiently addressed using the adjoint method [PTC*15, ZCT16, POT17, LCBD*18, MPI*18, PKLI*19]. While these adjoint-based approaches are efficient and can ensure local minima (not maxima), their implementation tends to be significantly more complicated because the nonlinear force constraints need to be differentiated, thus involving the Hessian of the objectives of elastic materials. Besides complexity, prior work has highlighted additional

challenges introduced by this Hessian, related to correctness, accuracy, robustness, and efficiency [PKLI*19, SWP*23]. As such, similar to the work of [TKA11], we formulate our method without explicitly encoding the Hessian into the rest shape optimization. To avoid these complications for the inversions, Choi et al. [CJSJ24] proposed using the finite difference method to approximate the gradient with respect to the curvatures while Liu [Liu22] treated a simulator as a black-box using search directions approximated with deformation gradients, although such schemes are known to be rather inefficient.

2.2.3. Global-Local Initialization

Hsu et al. [HTYW22] proposed a global-local, two-stage initialization method that first solves a global linear system to find optimal forces and then locally adjusts parameters on the rest shape, material stiffness, and internal state such that these parameters become consistent with the computed forces. Compared to the work of Twigg and Kačić-Alesić [TKA11] and our own, which need to solve a global nonlinear system iteratively, the two-stage initialization [HTYW22] requires solving a global linear system only once which reduces the computational cost significantly. This two-stage initialization was also extended to support hair simulation [HWP*23] with the elastic rod model of [KS16]. In practice, while the two-stage initialization is efficient, forces computed via the single global linear solve are not necessarily feasible with only the modified rest shape parameters of local elements. As such, one would typically need to compromise with non-zero forces failing to achieve static equilibrium or locally modified material stiffness parameters deviating from the user-desired strand dynamics, as discussed in [HTYW22, HWP*23].

In addition, it is nontrivial to apply the global-local initialization approach [HTYW22, HWP*23] to DER because its bending and twisting formulations do not satisfy the requirements of the global-local initialization. First, the bending and twisting formulations are tightly coupled in DER, and thus the local (minimal) stencil for bending does not necessarily preserve the angular momentum by itself (we confirmed this with the optimal forces generated by our method). This fact makes it difficult to define *force elements*, which are used to form the global linear system. Furthermore, assuming the use of the local stencil as a force element, the bending/twisting couples the rest length parameters with neighboring ones, transforming the (ostensibly) local steps into a global problem. As such, it is necessary to modify the global-local initialization to apply it to DER although such an approach would fail to achieve static equilibrium (see Sec. 5.6).

2.2.4. Differences from [TKA11]

Twigg and Kačić-Alesić [TKA11] proposed a rest shape optimization framework for sag-free mass-spring systems. Their method formulates an objective based on the L^2 norm of forces with a related regularizer, and optimizes the rest length and rest angle under box constraints for the mass-spring networks. While this work has a goal similar to ours, there are two key differences. First, instead of mass-spring systems, we use DER (which has been validated through physical experiments to generate dynamics in accordance with theory [BWR*08, BAV*10, JNO18, RLR*21]) and therefore we optimize the rest shape parameters of the DER objectives: rest length, rest curvature, and rest twist. In addition, we also impose corresponding box constraints (based on our analysis

and numerical experiments) on the rest shape parameters to prevent significant changes in their values and thus avoid introducing stability issues. Second, instead of the L^2 norm of forces, we formulate an objective based on kinetic energy. While these are conceptually similar, our kinetic-energy-based objective can correctly reflect the mass/inertia of materials and consistently evaluate the energy contributions of each component in the system. Moreover, our kinetic-energy-based formulation has much better numerical conditioning than the force-norm-based one due to the proper consideration of the energy contributions (see Sec. 4.6), making it possible to use double-precision floating-point when solving the optimization problem. By contrast, the force-norm-based objective can be numerically ill-conditioned such that, to achieve sufficiently small error during the optimization, extended-precision would be required (such as the quad-precision adopted in [TKA11]). These features are not standard nor officially supported in C++20, for example, and significantly slow down the computations.

3. Discrete Elastic Rods Preliminaries

In preparation for the rest shape optimization approach we develop in Sec. 4, we briefly review the energy objectives in the DER formulation and their gradients, which play key roles in forward simulation [BWR*08, BAV*10]. Here, as our focus is on the static equilibrium case, we do not consider contacts, plastic deformations, or tearing, and cover details only directly relevant to our rest shape optimizer. We refer readers to the book on DER [JNO18] for the underlying principles including space/time-parallel transport and to the previous work for derivation and details of the gradient and Hessian [PKLI*19, FBGZ19]. In addition, because we consider scenarios without any contacts among elastic rods and other objects, each rod can independently be processed in both forward simulation and rest shape optimization. As such, we describe formulations for a single elastic rod below.

We consider an elastic rod discretized with N vertices (which are sufficiently well distributed without forming acute angles) whose stacked positions are denoted by $\mathbf{x} = [\mathbf{x}_0^T, \ldots, \mathbf{x}_{N-1}^T]^T \in \mathbb{R}^{3N}$ (each vertex position is three-dimensional) and $(N-1)$ connecting edges whose stacked edge angles are $\boldsymbol{\theta} = [\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{N-2}]^T \in \mathbb{R}^{N-1}$ (each edge angle is one-dimensional). The generalized positions of the rod can be defined by interleaving the vertex positions and edge angles as $\mathbf{q} = [\mathbf{x}_0^T, \boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{N-2}, \mathbf{x}_{N-1}^T]^T \in \mathbb{R}^{4N-1}$. This specific variable arrangement of $\mathbf{q}$ leads to a banded Hessian for the DER objectives with time-parallel transport (which also accommodates trivial parallel computations [KJM10]) and thus enables efficient implicit integration [BAV*10], via Cholesky/LDLT-based direct linear solves [HYW*23]. Along with the positional variables, we also define stacked velocities for the vertices as $\dot{\mathbf{x}} = [\dot{\mathbf{x}}_0^T, \ldots, \dot{\mathbf{x}}_{N-1}^T]^T \in \mathbb{R}^{3N}$, stacked angular velocities for the edges $\dot{\boldsymbol{\theta}} = [\dot{\boldsymbol{\theta}}_0, \ldots, \dot{\boldsymbol{\theta}}_{N-2}]^T \in \mathbb{R}^{N-1}$, and generalized velocities $\dot{\mathbf{q}} = [\dot{\mathbf{x}}_0^T, \dot{\boldsymbol{\theta}}_0, \ldots, \dot{\boldsymbol{\theta}}_{N-2}, \dot{\mathbf{x}}_{N-1}^T]^T \in \mathbb{R}^{4N-1}$. For simplicity, we assume perfectly circular cross-sections with a constant radius, constant density, and constant material stiffness over the strand. In addition, we initialize reference frames on each edge via space-parallel transport propagated from a randomly chosen reference frame on the first edge (i.e., any reference frame on the first edge, as long as it forms the $SO(3)$ basis with the unit tangent vector, gives identical results for the dynamics) and then compute material frames based on the reference frames

and edge angles $\boldsymbol{\theta}$ [BWR*08, BAV*10, JNO18]. While, in the previous work [BWR*08, BAV*10, JNO18], subscript and superscript indices are used for the variables defined on vertices and edges, respectively, we use only subscript indices to avoid confusion with common superscripts (e.g., exponentiation, inverse, transpose, and time index), and because we can identify whether variables are defined on a vertex or an edge from their definition.

While the dimensionality of $\mathbf{q}$ is $(4N-1)$ in 3D, we focus on an elastic rod with its root end minimally clamped as this is one of the simplest and most common settings, e.g., for hairs attached to a head (although it would be possible to support both ends clamped [HB23]). To this end, we fix $\mathbf{x}_0, \boldsymbol{\theta}_0$, and $\mathbf{x}_1$ or update them in a prescribed way (below, we use the term "fix" regardless of whether these variables are fixed in place or moved in a prescribed way) so that the elastic rod has an anchor enabling stretching/bending/twisting to work meaningfully, i.e., these variables work as a Dirichlet boundary condition to eliminate DOFs for pure rigid motions. This treatment corresponds to eliminating the DOFs for $\mathbf{x}_0, \boldsymbol{\theta}_0$, and $\mathbf{x}_1$, and thus under our minimal root-end clamping the resulting number of free DOFs is $(4N-8) = (4N-1) - (3+1+3)$.

A single forward simulation step of DER can be written as an energy minimization problem [MTGG11, GSS*15]:

$$\mathbf{q} = \arg\min_{\mathbf{q}} E_{\text{DER}}(\mathbf{q}), \tag{1}$$

where the objective of DER, $E_{\text{DER}}(\mathbf{q})$ consists of the objective terms for inertia $E_{\text{inertia}}(\mathbf{q})$, stretching $E_{\text{stretch}}(\mathbf{x})$, bending $E_{\text{bend}}(\mathbf{q})$, and twisting $E_{\text{twist}}(\mathbf{q})$, and can be defined as

$$E_{\text{DER}}(\mathbf{q}) = E_{\text{inertia}}(\mathbf{q}) + E_{\text{stretch}}(\mathbf{x}) + E_{\text{bend}}(\mathbf{q}) + E_{\text{twist}}(\mathbf{q}). \tag{2}$$

Here, we define $E_{\text{stretch}}(\mathbf{x})$ with $\mathbf{x}$ alone because stretching is independent of $\boldsymbol{\theta}$. While velocity-based damping can also be incorporated into the optimization (1) [BOFN18], it is irrelevant to our rest shape optimization since we focus on the static equilibrium case with $\dot{\mathbf{q}} = 0$. We can obtain the generalized positions for the next time step by solving the optimization problem (1), e.g., with a gradient-based optimizer (such as gradient descent or Newton's method [NW06]) using the gradient of the DER objective with respect to $\mathbf{q}$, $\nabla_{\mathbf{q}} E_{\text{DER}}(\mathbf{q})$, while handling the fixed variables as Dirichlet boundary conditions. We note that the gradient with respect to the fixed variables is not defined (or, in practice, it can be assumed 0 for convenience). In our framework, we use a single Newton iteration per simulation step for efficiency [BW98, KE22, HYW*23] (although this approach sacrifices stability with larger parameter values [SWP*23]). If one prefers avoiding the Hessian of the DER objectives (to make the entire framework Hessian-free), it would be possible to use Alternating Direction Method of Multipliers (ADMM) [NOB16] to solve (1). In the following sections, we define each objective term and its gradient.

3.1. Inertia
We define the objective for the inertia as

$$E_{\text{inertia}}(\mathbf{q}) = \frac{1}{2\Delta t^2} \|\mathbf{q} - \mathbf{q}^*\|_{\mathbf{M}}^2, \tag{3}$$

where $\mathbf{q}^*(= \mathbf{q}^t + \Delta t \dot{\mathbf{q}}^t + \Delta t^2 \mathbf{M}^{-1} \mathbf{f}_{\text{ext}})$ denotes the predicted generalized positions, $\mathbf{q}^t$ and $\dot{\mathbf{q}}^t$ denote the generalized positions and

velocities at time t, respectively, $\Delta t \ (> 0)$ time step size, $\mathbf{M} \in \mathbb{R}^{(4N-1) \times (4N-1)}$ a diagonal, generalized mass matrix [BAV*10, JNO18], and $\mathbf{f}_{\text{ext}} \in \mathbb{R}^{4N-1}$ a generalized external force vector. While sag-free simulation typically assumes gravity as the sole external force, we can nevertheless easily support other constant external forces, such as wind forces or load on specific vertices [PTC*15] or edges.

The diagonal mass matrix $\mathbf{M}$ is defined as $\text{diag}(\mathbf{M}) = [m_0, m_0, m_0, I_0, \ldots, I_{N-2}, m_{N-1}, m_{N-1}, m_{N-1}]^T$, where m_i denotes the mass for vertex i, and I_i the inertia for edge i. Specifically, $m_0 = m_1 = \infty$ (as the first two root-end vertices are fixed), and other m_i can be computed by

$$m_i = \rho \pi r^2 \left(\frac{\bar{l}_{i-1} + \bar{l}_i}{2} \right), \tag{4}$$

where ρ and r denote the mass density and the radius of the strand, respectively, $\bar{l}_i \ (> 0)$ denotes the rest length for edge i (we define $\bar{\mathbf{l}} = \left[\bar{l}_0, \ldots, \bar{l}_{N-2} \right]^T \in \mathbb{R}^{N-1}$), and we specify $\bar{l}_{-1} = \bar{l}_{N-1} = 0$ for convenience. Then, $I_0 = \infty$, and other I_i are computed by

$$I_i = \frac{1}{2} \rho \pi r^4 \bar{l}_i. \tag{5}$$

The gradient of $E_{\text{inertia}}(\mathbf{q})$ with respect to $\mathbf{q}$ (i.e., $\nabla_{\mathbf{q}} E_{\text{inertia}}(\mathbf{q}) \in \mathbb{R}^{4N-1}$) is given by

$$\nabla_{\mathbf{q}} E_{\text{inertia}}(\mathbf{q}) = \frac{\mathbf{M}}{\Delta t^2}(\mathbf{q} - \mathbf{q}^*). \tag{6}$$

3.2. Stretching
The stretching energy is defined over each edge using its associated two vertices (i.e., both ends of the edge). Thus, the objective for stretching can be decomposed and defined as

$$E_{\text{stretch}}(\mathbf{x}) = \sum_{i=1}^{N-2} E_{\text{stretch},i}(\mathbf{x}_i, \mathbf{x}_{i+1}), \tag{7}$$

and the stretching energy defined for edge i with the associated vertices i and $i+1$ is given as

$$E_{\text{stretch},i}(\mathbf{x}_i, \mathbf{x}_{i+1}) = \frac{1}{2} \left(\frac{c_{\text{stretch}} \pi r^2}{\bar{l}_i} \right) (l_i - \bar{l}_i)^2, \tag{8}$$

where c_{stretch} denotes the stiffness coefficient for stretching, and $l_i \ (= \|\mathbf{x}_{i+1} - \mathbf{x}_i\|_2)$ denotes the length of edge i. (Unlike the work of [BAV*10], which uses Young's modulus and shear modulus, we use different coefficients for each energy to individually control their strengths.) Given that the vertices $\mathbf{x}_0$ and $\mathbf{x}_1$ are fixed, there is no stretching energy contribution $E_{\text{stretch},0}(\mathbf{x}_0, \mathbf{x}_1)$ from edge 0 (the first edge at the root end).

The gradient can be computed as

$$\nabla_{\mathbf{x}_{i+1}} E_{\text{stretch},i}(\mathbf{x}_i, \mathbf{x}_{i+1}) = -\nabla_{\mathbf{x}_i} E_{\text{stretch},i}(\mathbf{x}_i, \mathbf{x}_{i+1}) \tag{9}$$

$$= c_{\text{stretch}} \pi r^2 (l_i \bar{l}_i^{-1} - 1) \mathbf{t}_i, \tag{10}$$

where $\mathbf{t}_i \ (= \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{l_i})$ denotes the unit tangent vector for edge i.

3.3. Bending
The bending energy is defined on an inner vertex (excluding the first and last vertices) together with its two neighboring edge angles and vertices, i.e., with $(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}) \in \mathbb{R}^{11}$. Thus,

the objective for bending can be decomposed and defined as

$$E_{\text{bend}}(\mathbf{q}) = \sum_{i=1}^{N-2} E_{\text{bend},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}). \quad (11)$$

We define the bending energy on vertex i with the four-dimensional curvature $\boldsymbol{\kappa}_i \in \mathbb{R}^4$ ($\boldsymbol{\kappa}_i = [\boldsymbol{\kappa}_{i,0}, \boldsymbol{\kappa}_{i,1}, \boldsymbol{\kappa}_{i,2}, \boldsymbol{\kappa}_{i,3}]^T$, where two 2D curvatures ($\boldsymbol{\kappa}_{i,0}$, $\boldsymbol{\kappa}_{i,1}$) and ($\boldsymbol{\kappa}_{i,2}$, $\boldsymbol{\kappa}_{i,3}$) denote the curvature binormals evaluated on the material frames of edges $i-1$ and i, respectively, and $\boldsymbol{\kappa} = [\boldsymbol{\kappa}_1^T, \dots, \boldsymbol{\kappa}_{N-2}^T]^T \in \mathbb{R}^{4N-8}$) and its corresponding rest curvature $\bar{\boldsymbol{\kappa}}_i \in \mathbb{R}^4$ in 3D [BWR*08, FBGZ19] as

$$E_{\text{bend},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}) = \frac{1}{2} \left(\frac{c_{\text{bend}}\pi r^4}{4(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)} \right) \|\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i\|_2^2, \quad (12)$$

where c_{bend} denotes the stiffness coefficient for bending energy. Here, we do not use the bending model with the two-dimensional curvature (and rest curvature) via the averaged material frames from [BAV*10] because their model would fail to correctly evaluate the bending of strands when they are bent or twisted due to the non-unit and non-orthogonal, averaged material frames [GB15, PKLI*19]. In addition, the bending model (12) [BWR*08, FBGZ19] is slightly simpler than the one presented by [PKLI*19] (which modifies the rest length term) and avoids introducing additional complexity when differentiating (12) with respect to the rest length (see Sec. 4.1.3).

The 11-dimensional gradient of (12) can be computed by

$$\nabla E_{\text{bend},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}) = \left(\frac{c_{\text{bend}}\pi r^4}{4(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)} \right) \mathbf{J}_{\text{curv},i}^T (\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i), \quad (13)$$

where $\mathbf{J}_{\text{curv},i} \in \mathbb{R}^{4 \times 11}$ denotes the Jacobian of $\boldsymbol{\kappa}_i$ with respect to the variables $(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1})$, and $\mathbf{J}_{\text{curv},i}^T (\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i) \in \mathbb{R}^{11}$ in (13) can be rewritten with the gradient of $\boldsymbol{\kappa}_i$'s jth entry, $\boldsymbol{\kappa}_{i,j}$ (i.e., $\nabla \boldsymbol{\kappa}_{i,j} \in \mathbb{R}^{11}$) as

$$\mathbf{J}_{\text{curv},i}^T (\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i) = \sum_{j=0}^{3} (\boldsymbol{\kappa}_{i,j} - \bar{\boldsymbol{\kappa}}_{i,j}) \nabla \boldsymbol{\kappa}_{i,j}. \quad (14)$$

3.4. Twisting
Similar to the bending case, the objective for twisting can be defined with $(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}) \in \mathbb{R}^{11}$ as

$$E_{\text{twist}}(\mathbf{q}) = \sum_{i=1}^{N-2} E_{\text{twist},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}), \quad (15)$$

where the twisting energy defined for vertex i is given with twist $\mathbf{m}_i$ ($\mathbf{m} = [\mathbf{m}_1, \dots, \mathbf{m}_{N-2}]^T \in \mathbb{R}^{N-2}$), which takes the reference twist into account [BAV*10, KJM10, JNO18, PKLI*19], and its corresponding rest twist $\bar{\mathbf{m}}_i$ by

$$E_{\text{twist},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}) = \frac{1}{2} \left(\frac{c_{\text{twist}}\pi r^4}{(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)} \right) (\mathbf{m}_i - \bar{\mathbf{m}}_i)^2, \quad (16)$$

where c_{twist} denotes the stiffness coefficient for the twisting energy.

The 11-dimensional gradient of (16) can be computed as

$$\nabla E_{\text{twist},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1}) = \left(\frac{c_{\text{twist}}\pi r^4}{(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)} \right) (\mathbf{m}_i - \bar{\mathbf{m}}_i) \nabla \mathbf{m}_i. \quad (17)$$

4. Rest Shape Optimization
We define the rest shape parameter $\bar{\mathbf{s}} = [\bar{\mathbf{l}}_1, \dots, \bar{\mathbf{l}}_{N-2}, \bar{\boldsymbol{\kappa}}^T, \bar{\mathbf{m}}^T]^T \in \mathbb{R}^{6N-12}$ since $(6N-12) = (N-2) + 4(N-2) + (N-2)$. We aim to optimize $\bar{\mathbf{s}}$ based on minimization of the kinetic energy with a regularizer subject to box constraints. Specifically, we formulate the rest shape optimization as

$$\bar{\mathbf{s}} = \underset{\bar{\mathbf{s}}_{\min} \leq \bar{\mathbf{s}} \leq \bar{\mathbf{s}}_{\max}}{\arg\min} \tilde{F}(\bar{\mathbf{s}}), \quad \tilde{F}(\bar{\mathbf{s}}) = \frac{1}{\Delta t^2} \left(F_{\text{kin}}(\bar{\mathbf{s}}) + F_{\text{reg}}(\bar{\mathbf{s}}) \right), \quad (18)$$

where $\bar{\mathbf{s}}_{\min} \in \mathbb{R}^{6N-12}$ and $\bar{\mathbf{s}}_{\max} \in \mathbb{R}^{6N-12}$ denote the lower and upper bounds imposed on $\bar{\mathbf{s}}$, respectively, and $F_{\text{kin}}(\bar{\mathbf{s}})$ and $F_{\text{reg}}(\bar{\mathbf{s}})$ denote the kinetic energy objective and a regularizer, respectively. Here, we divide $F_{\text{kin}}(\bar{\mathbf{s}})$ and $F_{\text{reg}}(\bar{\mathbf{s}})$ by Δt^2, anticipating that it will cancel with the Δt^2 in (20) and (29), and noting that this scaling by a constant value has no effect on the rest shape optimization (18). In addition, note that $\bar{\mathbf{s}}$ excludes $\bar{\mathbf{l}}_0$ because the first two root-end vertices are fixed, and thus the edge between them should also be fixed. As such, we do not define the Jacobian with respect to $\bar{\mathbf{l}}_0$ (similar to the fixed variables in Sec. 3), and thus $\bar{\mathbf{l}}_0$ is not modified by the rest shape optimization (18).

While one could consider optimizing $\tilde{F}$ in (18) with respect to $\mathbf{q}$ given its fewer DOFs than those for $\bar{\mathbf{s}}$, and then initializing $\bar{\mathbf{s}}$ (which is eventually used to compute DER objectives and forces) with the optimized $\mathbf{q}$, this approach makes it difficult to control the system energy (e.g., to avoid stability problems) via box constraints. $\mathbf{q}$ is nonlinearly related to multiple components of $\bar{\mathbf{s}}$, and thus it is not straightforward to specify box constraints on $\mathbf{q}$ that enforce, e.g., $|\bar{\boldsymbol{\kappa}}_{i,0}| < 1$ to regulate bending energy (12). Additionally, computing the Jacobian of the generalized forces with respect to $\mathbf{q}$ (i.e., Hessian) is much more complicated compared to those with respect to $\bar{\mathbf{s}}$ (see Sec. 4.1.2, Sec. 4.1.3, and Sec. 4.1.4).

4.1. Generalized Kinetic Energy Objective
Considering a single forward timestep, solving (1) determines the generalized force $\mathbf{f}_{\text{DER}}$ with optimal (end of step) generalized positions $\mathbf{q}$, i.e., $\mathbf{f}_{\text{DER}} = -\nabla_{\mathbf{q}} E_{\text{DER}}(\mathbf{q})$. The corresponding optimal (end of step) generalized velocity $\dot{\mathbf{q}}$ is given by $\dot{\mathbf{q}} = \dot{\mathbf{q}}^t + \Delta t \mathbf{M}^{-1} \mathbf{f}_{\text{DER}}$. Thus, the corresponding generalized kinetic energy (summation of the translational kinetic energy for vertices and rotational kinetic energy for edges) can be computed by

$$F_{\text{kin}}(\bar{\mathbf{s}}) = \frac{1}{2} \|\dot{\mathbf{q}}\|_{\mathbf{M}}^2. \quad (19)$$

Here, we can assume $\dot{\mathbf{q}}^t = 0$ for the static equilibrium case, and therefore the kinetic energy objective (19) can be rewritten as

$$F_{\text{kin}}(\bar{\mathbf{s}}) = \frac{\Delta t^2}{2} \|\mathbf{f}_{\text{DER}}\|_{\mathbf{M}^{-1}}^2. \quad (20)$$

Considering the Jacobian $\mathbf{J}_{\text{DER}}(= \frac{\partial \mathbf{f}_{\text{DER}}}{\partial \bar{\mathbf{s}}}) \in \mathbb{R}^{(4N-8) \times (6N-12)}$ (while treating $\mathbf{M}$ as constant because it is aimed to evaluate the

norm in the kinetic energy metric at the initial setting), the gradient of (20) can be computed as

$$\nabla F_{\text{kin}}(\bar{\mathbf{s}}) = \Delta t^2 \mathbf{J}_{\text{DER}}^T \mathbf{M}^{-1} \mathbf{f}_{\text{DER}}, \tag{21}$$

and the Hessian can be approximated in the Gauss-Newton style (ignoring the third-order tensors) by

$$\nabla^2 F_{\text{kin}}(\bar{\mathbf{s}}) \approx \Delta t^2 \mathbf{J}_{\text{DER}}^T \mathbf{M}^{-1} \mathbf{J}_{\text{DER}}. \tag{22}$$

Below, we explicitly give the Jacobian for each component.

4.1.1. Inertia

In the static equilibrium case, we can assume $\mathbf{q} = \mathbf{q}^t$ and $\dot{\mathbf{q}}^t = 0$. As such, fully canceling $\mathbf{M}$ (which depends on $\bar{\mathbf{l}}$) in (6) given the predicted position $\mathbf{q}^*$ in (3), the inertia force $\mathbf{f}_{\text{inertia}} = -\nabla_{\mathbf{q}} E_{\text{inertia}}(\mathbf{q})$ can be defined as

$$\mathbf{f}_{\text{inertia}} = \mathbf{f}_{\text{ext}}. \tag{23}$$

This result indicates that the inertia force is independent of $\bar{\mathbf{l}}$, and thus the Jacobian $\frac{\partial \mathbf{f}_{\text{inertia}}}{\partial \bar{\mathbf{s}}} = 0$. In addition, (23) is also independent of Δt, and thus the actual value of Δt has no effect on the rest shape optimization (18). In the rest shape optimization for static equilibrium, a larger norm of the generalized forces, $\|\mathbf{f}_{\text{inertia}}\|_2$, typically requires more significant rest shape changes.

4.1.2. Stretching

The stretching force of edge i on vertex $i + 1$ is defined as $\mathbf{f}_{\text{stretch},i,i+1} = -\nabla_{\mathbf{x}_{i+1}} E_{\text{stretch},i}(\mathbf{x}_i, \mathbf{x}_{i+1})$ (10), and $\mathbf{f}_{\text{stretch},i,i}$ can similarly be defined. Given the dependence of $\mathbf{f}_{\text{stretch},i,i+1}$ and $\mathbf{f}_{\text{stretch},i,i}$ on $\bar{\mathbf{l}}_i$, we can compute the Jacobian by

$$\frac{\partial \mathbf{f}_{\text{stretch},i,i+1}}{\partial \bar{\mathbf{l}}_i} = -\frac{\partial \mathbf{f}_{\text{stretch},i,i}}{\partial \bar{\mathbf{l}}_i} = c_{\text{stretch}} \pi r^2 \mathbf{l}_i \bar{\mathbf{l}}_i^{-2} \mathbf{t}_i. \tag{24}$$

4.1.3. Bending

We define the bending force according to (13) by $\mathbf{f}_{\text{bend},i} = -\nabla E_{\text{bend},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1})$. Given its dependence on $\bar{\mathbf{l}}_{i-1}, \bar{\mathbf{l}}_i$, and $\bar{\boldsymbol{\kappa}}_i$, the Jacobian can be computed by

$$\frac{\partial \mathbf{f}_{\text{bend},i}}{\partial \bar{\mathbf{l}}_{i-1}} = \frac{\partial \mathbf{f}_{\text{bend},i}}{\partial \bar{\mathbf{l}}_i} = \left(\frac{c_{\text{bend}} \pi r^4}{4(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)^2} \right) \mathbf{J}_{\text{curv},i}^T (\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i), \tag{25}$$

and

$$\frac{\partial \mathbf{f}_{\text{bend},i}}{\partial \bar{\boldsymbol{\kappa}}_i} = \left(\frac{c_{\text{bend}} \pi r^4}{4(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)} \right) \mathbf{J}_{\text{curv},i}^T. \tag{26}$$

4.1.4. Twisting

Similar to the bending case, we define the twisting force according to (17) as $\mathbf{f}_{\text{twist},i} = -\nabla E_{\text{twist},i}(\mathbf{x}_{i-1}, \boldsymbol{\theta}_{i-1}, \mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{x}_{i+1})$. Then, given its dependence on $\bar{\mathbf{l}}_{i-1}, \bar{\mathbf{l}}_i$, and $\bar{\mathbf{m}}_i$, the Jacobian is computed as

$$\frac{\partial \mathbf{f}_{\text{twist},i}}{\partial \bar{\mathbf{l}}_{i-1}} = \frac{\partial \mathbf{f}_{\text{twist},i}}{\partial \bar{\mathbf{l}}_i} = \frac{c_{\text{twist}} \pi r^4}{(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)^2} (\mathbf{m}_i - \bar{\mathbf{m}}_i) \nabla \mathbf{m}_i, \tag{27}$$

and

$$\frac{\partial \mathbf{f}_{\text{twist},i}}{\partial \bar{\mathbf{m}}_i} = \frac{c_{\text{twist}} \pi r^4}{(\bar{\mathbf{l}}_{i-1} + \bar{\mathbf{l}}_i)} \nabla \mathbf{m}_i. \tag{28}$$

4.2. Regularizer for Rest Shape Parameters

Given that the number of DOFs for $\bar{\mathbf{s}}$ is larger than for $\mathbf{q}$, the system is typically underdetermined, i.e., there are multiple sets of $\bar{\mathbf{s}}$ that achieve $F_{\text{kin}}(\bar{\mathbf{s}}) = 0$. Among such parameter sets, it is preferable to choose $\bar{\mathbf{s}}$ that minimizes deviation from the initial rest shape parameters $\bar{\mathbf{s}}_{\text{initial}}$ to avoid introducing stability problems. To this end, we penalize changes in the space of $\bar{\mathbf{s}}$ and define the regularizer objective as

$$F_{\text{reg}}(\bar{\mathbf{s}}) = \frac{\Delta t^2}{2} \alpha \|\bar{\mathbf{s}} - \bar{\mathbf{s}}_{\text{initial}}\|_2^2, \tag{29}$$

where α denotes a positive regularizer coefficient (we typically set $\alpha = 10^{-5}$). Although the given system may already be overdetermined (having a unique solution) depending on the initial strand geometry, even in such cases, using the regularizer generally has a positive effect in mitigating overfitting. The previous work [TKA11] proposed using a regularizer based on the force norm, but we found it to be unnecessary because the regularizer in our framework is used to ensure that the system is overdetermined, while the box constraints can restrict the changes in the rest shape parameters in a more controlled way (see Sec. 4.3).

The gradient of (29) is defined as

$$\nabla F_{\text{reg}}(\bar{\mathbf{s}}) = \Delta t^2 \alpha (\bar{\mathbf{s}} - \bar{\mathbf{s}}_{\text{initial}}), \tag{30}$$

and the Hessian is given as

$$\nabla^2 F_{\text{reg}}(\bar{\mathbf{s}}) = \Delta t^2 \alpha \mathbf{I}, \tag{31}$$

where $\mathbf{I}$ denotes the Identity matrix.

4.3. Box Constraints for Rest Shape Parameters

To ensure that the rest shape parameters are within the physically meaningful range and to avoid introducing stability issues into the system, we impose box constraints on these parameters: $\bar{\mathbf{l}}_{\min} \leq \bar{\mathbf{l}} \leq \bar{\mathbf{l}}_{\max}$, $\bar{\boldsymbol{\kappa}}_{\min} \leq \bar{\boldsymbol{\kappa}} \leq \bar{\boldsymbol{\kappa}}_{\max}$, and $\bar{\mathbf{m}}_{\min} \leq \bar{\mathbf{m}} \leq \bar{\mathbf{m}}_{\max}$.

Given the initial rest length $\bar{\mathbf{l}}_{\text{initial}}$ (> 0), we define the lower and upper bounds as

$$\bar{\mathbf{l}}_{\min} = a_{\min} \bar{\mathbf{l}}_{\text{initial}}, \quad \bar{\mathbf{l}}_{\max} = a_{\max} \bar{\mathbf{l}}_{\text{initial}}, \tag{32}$$

where $a_{\min}$ and $a_{\max}$ are scalars, and we use $a_{\min} = 0.1$ and $a_{\max} = 1.1$ (which are determined through our experiments).

Considering the initial rest curvature $\bar{\boldsymbol{\kappa}}_{\text{initial}}$ and rest twist $\bar{\mathbf{m}}_{\text{initial}}$ (where both of them can be 0 in contrast to $\bar{\mathbf{l}}_{\text{initial}}$), we define their lower and upper bounds as

$$\bar{\boldsymbol{\kappa}}_{\min} = \bar{\boldsymbol{\kappa}}_{\text{initial}} - \mathbf{b}, \quad \bar{\boldsymbol{\kappa}}_{\max} = \bar{\boldsymbol{\kappa}}_{\text{initial}} + \mathbf{b}, \tag{33}$$

$$\bar{\mathbf{m}}_{\min} = \bar{\mathbf{m}}_{\text{initial}} - \mathbf{d}, \quad \bar{\mathbf{m}}_{\max} = \bar{\mathbf{m}}_{\text{initial}} + \mathbf{d}, \tag{34}$$

where column vectors $\mathbf{b} \in \mathbb{R}^{4N-8}$ and $\mathbf{d} \in \mathbb{R}^{N-2}$ denote the permitted deviation from $\bar{\boldsymbol{\kappa}}_{\text{initial}}$ and $\bar{\mathbf{m}}_{\text{initial}}$, respectively.

Given the turning angle ϕ_i at vertex i [BWR*08], we can derive the relation between the norm of the curvature and turning angle as $\frac{1}{2} \|\boldsymbol{\kappa}_i\|_2^2 = \left(2 \tan \frac{\phi_i}{2} \right)^2$. Thus, if we have the change of the turning angle $\Delta\phi$ at $\phi_i = 0$ with the corresponding change of the curvature $\Delta\kappa$ (assuming $\Delta\kappa = \Delta\boldsymbol{\kappa}_{i,0} = \Delta\boldsymbol{\kappa}_{i,1} = \Delta\boldsymbol{\kappa}_{i,2} = \Delta\boldsymbol{\kappa}_{i,3}$ for simplicity), we get $\frac{1}{2}(4\Delta\kappa)^2 = \left(2 \tan \frac{\Delta\phi}{2} \right)^2$, and thus $|\Delta\kappa| = \sqrt{2} |\tan \frac{\Delta\phi}{2}|$. This

relation indicates that, e.g., allowing a maximal change of the turning angle of $\frac{\pi}{2}$ (i.e., $\Delta\phi = \frac{\pi}{2}$) leads to a change of the curvature of $\Delta\kappa = \sqrt{2}$. Based on this analysis for bending, we define $\mathbf{b}$ as

$$\mathbf{b} = \Delta\kappa\mathbf{e_b} = \sqrt{2}\mathbf{e_b}, \quad (35)$$

where $\mathbf{e_b}$ denotes a vector of all ones with the same length as $\mathbf{b}$. For twisting, we permit a maximal change of twist of $\Delta m = \frac{\pi}{8}$, and thus we define $\mathbf{d}$ as

$$\mathbf{d} = \Delta m\mathbf{e_d} = \frac{\pi}{8}\mathbf{e_d}, \quad (36)$$

where $\mathbf{e_d}$ denotes a vector of all ones with the same length as $\mathbf{d}$. Since we chose $\Delta\kappa$ in (35) and Δm in (36) empirically based on our analysis and numerical experiments with reasonable material parameters (e.g., $c_{\text{stretch}} = 10^8\ \text{kg}/(\text{m}\cdot\text{s}^2)$, $c_{\text{bend}} = 10^8\ \text{kg}/(\text{m}\cdot\text{s}^2)$, and $c_{\text{twist}} = 10^8\ \text{kg}/(\text{m}\cdot\text{s}^2)$), it would be necessary to adjust these values for much softer materials (Sec. 5.7).

4.4. Gauss-Newton Solver with Penalty Method

Given the least-squares-style nonlinear objectives arising in our problem, we aim to solve the optimization via the GN method [NW06]. While enforcing box constraints within Newton-type optimizers is possible [TB21], it requires accounting for the box constraints when solving inner problems, which leads to an additional computational cost compared to unconstrained linear solves. Considering that our problems are relatively stiff (albeit yielding sparse systems), instead of handling box constraints in the inner problems, we prefer to convert the box-constrained minimization into an unconstrained one via the penalty method [NW06], thus ensuring unconstrained inner linear systems for greater efficiency.

Specifically, we reformulate the box-constrained minimization (18) into the following unconstrained minimization by converting the box constraints into a penalty objective $F_{\text{box}}(\bar{\mathbf{s}})$ [NW06]:

$$\bar{\mathbf{s}} = \arg\min_{\bar{\mathbf{s}}} F(\bar{\mathbf{s}}), \ F(\bar{\mathbf{s}}) = \frac{1}{\Delta t^2}\left(F_{\text{kin}}(\bar{\mathbf{s}}) + F_{\text{reg}}(\bar{\mathbf{s}}) + F_{\text{box}}(\bar{\mathbf{s}})\right), \quad (37)$$

where we define $F_{\text{box}}(\bar{\mathbf{s}})$ as

$$F_{\text{box}}(\bar{\mathbf{s}}) = \frac{\Delta t^2}{2}\beta\left(\|\max(\bar{\mathbf{s}} - \bar{\mathbf{s}}_{\max}, 0)\|_2^2 + \|\max(\bar{\mathbf{s}}_{\min} - \bar{\mathbf{s}}, 0)\|_2^2\right), \quad (38)$$

with β denoting a penalty parameter (we set $\beta = 10^6$). Note that, similar to the original rest shape optimization (18), Δt in (37) cancels completely with (20), (29), (38), and therefore Δt has no effect on achieving static equilibrium via rest shape optimization.

The gradient of (38) is given by

$$\nabla F_{\text{box}}(\bar{\mathbf{s}}) = \Delta t^2\beta\left(\max(\bar{\mathbf{s}} - \bar{\mathbf{s}}_{\max}, 0) + \max(\bar{\mathbf{s}}_{\min} - \bar{\mathbf{s}}, 0)\right), \quad (39)$$

and the Hessian is computed by

$$\nabla^2 F_{\text{box}}(\bar{\mathbf{s}}) = \Delta t^2\beta\left(\mathbf{H}_{\max} + \mathbf{H}_{\min}\right), \quad (40)$$

where $\mathbf{H}_{\max} \in \mathbb{R}^{(6N-12)\times(6N-12)}$ is a diagonal matrix, with diagonal elements given by $\text{diag}(\mathbf{H}_{\max}) = [H(\bar{\mathbf{s}}_0 - \bar{\mathbf{s}}_{\max,0}), \ldots, H(\bar{\mathbf{s}}_{6N-11} - \bar{\mathbf{s}}_{\max,6N-11})]^T$ using the Heaviside step function $H(\cdot)$ with $H(0) = 0$. We define $\mathbf{H}_{\min}$ analogously.

Algorithm 1 shows our rest shape optimization algorithm with the GN solver. To accelerate its convergence via warm starting and

to reduce the possibility of falling into suboptimal local minima, we initialize $\bar{\mathbf{s}}$ from the initial $\mathbf{l}$, κ, and $\mathbf{m}$. The inner linear system is guaranteed to be symmetric positive definite (SPD) due to the GN-style Hessian approximation and the regularizer. Given that the system is relatively stiff yet sparse, we solve it using a sparse Cholesky-based direct solver. To ensure a decrease of the objective, we employ a back-tracking line search [NW06]. In practice, as we enforce the box constraints via the penalty, the rest shape parameters $\bar{\mathbf{s}}$ can slightly violate their bounds. Given these relatively small violations, if necessary, we can project $\bar{\mathbf{s}}$ back into the valid range in each iteration (although this treatment was not needed in any of our examples). We terminate the GN iterations when $\|\Delta\bar{\mathbf{s}}\|_2$ becomes smaller than a threshold ϵ ($= 10^{-5}$). In addition, to ensure that the GN solver ultimately terminates, we halt the iteration if it exceeds 500 iterations or the back-tracking line search fails.

Algorithm 1 Rest Shape Optimization with Gauss-Newton

1: $k = 0$
2: Initialize the generalized positions $\mathbf{q}$, density ρ, radius r, and material stiffness coefficients $c_{\text{stretch}}, c_{\text{bend}}$, and c_{twist}
3: Initialize the unit tangent vector $\mathbf{t}$ and length $\mathbf{l}$, and set $\bar{\mathbf{l}} = \mathbf{l}$
4: Initialize the generalized mass matrix $\mathbf{M}$
5: Initialize the reference frames and update material frames
6: Compute the curvature κ and twist $\mathbf{m}$, and $\bar{\kappa} = \kappa$ and $\bar{\mathbf{m}} = \mathbf{m}$
7: Set $\bar{\mathbf{l}}_{\text{initial}} = \bar{\mathbf{l}}$, $\bar{\kappa}_{\text{initial}} = \bar{\kappa}$, and $\bar{\mathbf{m}}_{\text{initial}} = \bar{\mathbf{m}}$
8: Initialize $\bar{\mathbf{l}}_{\min}, \bar{\mathbf{l}}_{\max}$ with (32), $\bar{\kappa}_{\min}, \bar{\kappa}_{\max}$ with (33), and $\bar{\mathbf{m}}_{\min}, \bar{\mathbf{m}}_{\max}$ with (34)
9: **do**
10:     Compute gradient $\nabla F(\bar{\mathbf{s}}^k)$ with (21), (30), and (39)
11:     Approximate Hessian $\nabla^2 F(\bar{\mathbf{s}}^k)$ with (22), (31), and (40)
12:     Compute $\Delta\bar{\mathbf{s}}^{k+1}$ by solving $(\nabla^2 F(\bar{\mathbf{s}}^k))\Delta\bar{\mathbf{s}}^{k+1} = -\nabla F(\bar{\mathbf{s}}^k)$ with a sparse Cholesky-based direct solver
13:     Compute the step length γ using back-tracking line search with (20), (29), and (38)
14:     Update the rest shape parameters by $\bar{\mathbf{s}}^{k+1} = \bar{\mathbf{s}}^k + \gamma\Delta\bar{\mathbf{s}}^{k+1}$
15:     Optionally, clamp $\bar{\mathbf{s}}^{k+1}$ into the valid range
16:     $k = k + 1$
17: **while** $\|\Delta\bar{\mathbf{s}}^{k+1}\|_2 > \epsilon$

4.5. Simplification with Zero Rest Twist

While capturing the twisting effects of elastic strands is essential for visual fidelity, it is relatively rare to exactly specify initial twist or edge angles when designing a strand geometry, e.g., for hair assets, because edge angles have almost no effect on the visualization result at the static state, and capturing and specifying edge angle information is challenging [LLR13, HMLL15]. In addition, some elastic rod simulators do not necessarily support edge angles [SLF08, USS14, SWP*23] and it is generally difficult for humans to intuit how the edge angles influence the strand dynamics for simulators that do consider the edge angles [BWR*08, BAV*10, KS16, SMSH18]. As a result, it is common to specify only vertex positions and initialize edge angles θ to zero. Given such zero edge angles, which also yield zero initial twist $\mathbf{m}$, setting the rest twist $\bar{\mathbf{m}}$ to zero completely eliminates the twist energy (16) and its gradient (17). As such, we can potentially exclude $\bar{\mathbf{m}}$ from $\bar{\mathbf{s}}$, leading to an effective DOF count of $(5N-10)$ for $\bar{\mathbf{s}}$ and thus reducing the computational cost for rest shape optimization.

In practice, rest shape optimization without the rest twist still

works well (i.e., achieves a stable static equilibrium) if the bending of the strands introduces only a limited amount of twisting given the tight coupling of bending and twisting in the DER formulation. Such cases do occur, so we will evaluate and compare the rest shape optimization with and without rest twist in terms of performance and quality (see Sec. 5.5 and Sec. 5.6).

4.6. Discussion of Norm Formulations

Eliding unnecessary coefficients for a simpler analysis, the objective based on the L^2 norm of the generalized force $\mathbf{f}$ is given by $\|\mathbf{f}\|_2^2$ [TKA11] whereas our kinetic-energy-based objective is $\|\mathbf{f}\|_{\mathbf{M}^{-1}}^2$ due to (20). As an example, consider the inertia force (23) due to generalized gravity $\mathbf{g}$ since the inertia is the unique trigger for non-zero elastic forces (which are expected to cancel the inertia) in our framework. In this case, $\mathbf{f} = \mathbf{Mg}$, and thus we get $\mathbf{g}^T \mathbf{M}^2 \mathbf{g}$ for the force-norm-based objective versus $\mathbf{g}^T \mathbf{Mg}$ for ours. As such, when mass/inertia of the strands is too small or large, squaring $\mathbf{M}$ can negatively impact the numerical conditioning of the system. In practice, Gauss-Newton optimization on $\|\mathbf{f}\|_2^2$ through $(\mathbf{J}^T \mathbf{J})\Delta \bar{\mathbf{s}} = -\mathbf{J}^T \mathbf{f}$ (where $\mathbf{J}$ denotes the Jacobian of $\mathbf{f}$ with respect to $\bar{\mathbf{s}}$) fails to reach sufficiently small $\|\mathbf{f}\|_2^2$ because $\mathbf{J}^T \mathbf{J}$ has entries significantly different in their magnitude and thus is numerically ill-conditioned, neglecting some force components. Consequently, extended precision would be required for the optimization to consistently succeed [TKA11]. Additionally, it is worth noting that our formulation properly accounts for the energy contributions from each component (unlike simple scaling and dimension analysis approaches that merely mitigate numerical precision problems) and is essential for DER as it involves variables in different dimensions (vertex positions and edge angles).

5. Results and Discussions

We implemented our method in C++20 with double-precision floating-point for scalars. When multiple strands are involved, we parallelize forward simulation and inverse problems with OpenMP, processing each strand in parallel. We executed the examples on a desktop machine with an Intel Core i7-9700 (8 cores) with 16GB RAM. We use 60 frames per second with a single simulation step except for Figure 1 and Figure 11, where we used five steps. We include the rest twist in the optimization unless otherwise mentioned. We set $\rho = 10^3$ kg/m^3, $r = 10^{-3}$ m, $c_{\text{stretch}} = 10^8$ kg/(m·s^2), $c_{\text{bend}} = 10^8$ kg/(m·s^2), and $c_{\text{twist}} = 10^8$ kg/(m·s^2) unless otherwise specified. For visualization purposes, we enlarge the vertices and edges. The first two black vertices and cyan edge indicate the "fixed" vertices and edge, respectively, while the red vertices and blue and white edges represent movable vertices and edges with active DOFs, respectively (except for Figures 1, 10, and 11). In addition, we render the material frames at the centers of the corresponding edges as green and magenta lines whose lengths are the same as the rest lengths of the edges.

5.1. Single Vertical Strand

To illustrate the effectiveness of our rest shape optimization, we first experiment with a single vertical elastic strand (whose length is 1m), uniformly discretized with 20 vertices, as shown in Figure 2. With three different stretch coefficients c_{stretch} ($5 \times 10^3, 5 \times 10^4$, and 5×10^5), we compare simulation results with the following three cases:

(a) $c_{\text{stretch}} = 5 \times 10^3$ **(b)** $c_{\text{stretch}} = 5 \times 10^4$ **(c)** $c_{\text{stretch}} = 5 \times 10^5$
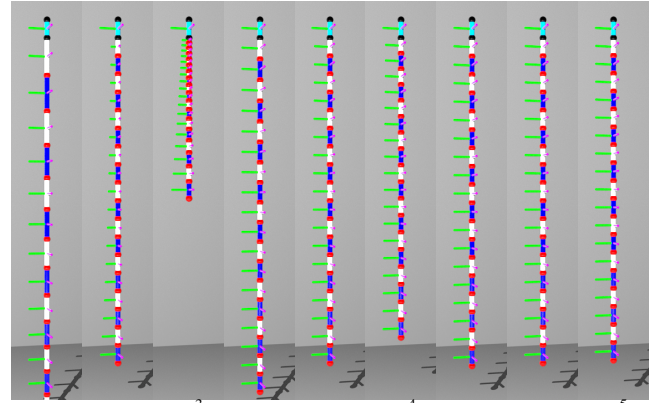
Figure 2: *Evaluation with a single vertical strand. For each c_{stretch} trio: (left) simulation results with naive initialization, (middle) with rest shape optimization, and (right) with rest shape optimization and zero gravity during simulation. Our rest shape optimizer modifies the rest length more significantly for softer materials to completely cancel gravity and thus achieve static equilibrium.*
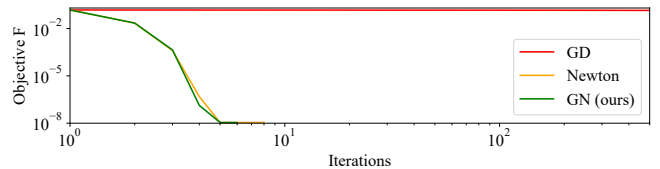

Figure 3: *Convergence profile for GD, Newton, and GN (ours). Both Newton and GN quickly converge while GD requires many more iterations.*

1. Naive initialization: initialization without our rest shape optimization;
2. With rest shape optimization;
3. With rest shape optimization, and zero gravity during simulation.

Due to the vertical arrangement of the strands, our method modifies the rest length only to counteract gravity without changing the rest curvature or rest twist, even though both are included in the rest shape optimization. Without the rest shape optimization the strand sags due to gravity, whereas our method enables the strand to achieve static equilibrium under all three different stiffness settings, modifying the rest length more significantly for softer materials to completely cancel the gravity force. Disabling gravity after the rest shape optimization reveals the modified rest lengths, which are shorter than the original ones.

5.2. Nonlinear Optimizer Comparisons

To justify our choice for the nonlinear optimizer, we again experiment with the vertical strand scenario (similar to Fig. 2), using 500 vertices and $c_{\text{stretch}} = 10^4$. We compare the following schemes:

1. GD: gradient descent;
2. Newton: Newton's method;
3. GN (ours): Gauss-Newton method.

For Newton's method, we compute the exact Hessian via the third-order tensors for (37) with the stretching force (since bending and twisting will not be introduced in this setting) and ensure an SPD system via Hessian projection per edge. Figure 3 shows a convergence profile of the objective over optimization iterations.
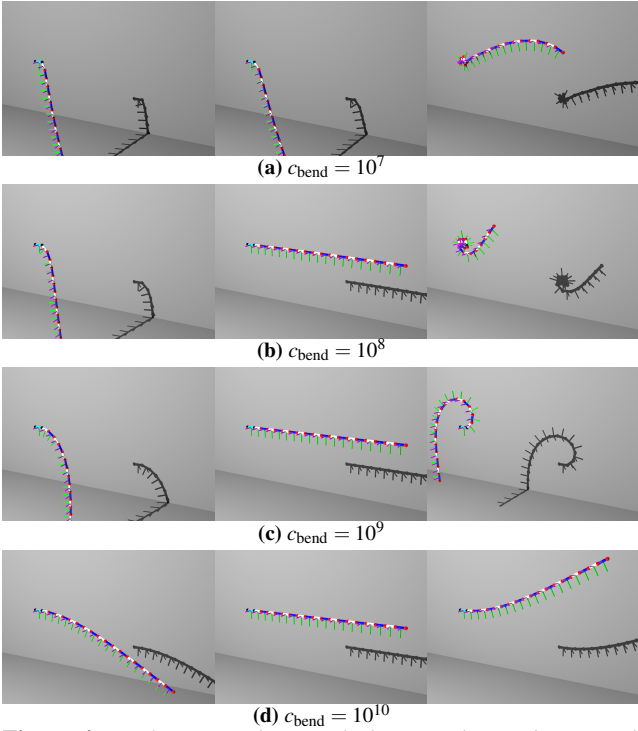
(a) $c_{bend} = 10^7$

(b) $c_{bend} = 10^8$

(c) $c_{bend} = 10^9$

(d) $c_{bend} = 10^{10}$

Figure 4: *Evaluation with a single horizontal strand. For each c_{bend} trio: (left) simulation results with naive initialization, (middle) with rest shape optimization, and (right) with rest shape optimization and zero gravity during simulation. Our optimizer modifies the rest curvature more significantly for softer strands to achieve static equilibrium (b), (c), and (d) while it fails for too soft material (a).*

The back-tracking line search leads GD to require tiny steps due to the stiff system, and GD fails to sufficiently decrease the objective within 500 iterations. Due to the one-dimensional strand structure, a sparse Cholesky solver with matrix reordering can solve the system (linear solve is required only for Newton-type optimizers, but not for GD) with an almost linear cost and is sufficiently fast (per-iteration cost for GN is less than $2\times$ of GD), and thus GN is much faster (GD took 10.7 s with 500 iterations whereas GN took 0.11 s with 6 iterations). Given GN's low per-iteration cost, it is likely to outperform quasi-Newton methods. With Newton's method, we do not observe a particular benefit on convergence over GN while computing the exact Hessian introduces an additional cost and significant implementation complexity associated with the third-order tensors. These observations make GN most suitable as our optimizer.

5.3. Single Horizontal Strand

Next, we test with a horizontal strand uniformly discretized with 20 vertices, as shown in Figure 4, and used $\Delta\boldsymbol{\kappa} = 10$ to allow larger curvature changes. With four different bending coefficients c_{bend} ($10^7, 10^8, 10^9,$ and 10^{10}), we compare the three cases:

1. Naive initialization;
2. With rest shape optimization;
3. With rest shape optimization, and zero gravity during simulation.
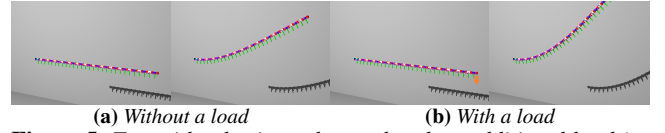
(a) *Without a load*   **(b)** *With a load*

Figure 5: *Test with a horizontal strand and an additional load (as illustrated with an orange arrow) using our rest shape optimization. For each case, we show the simulation with gravity (left) and without gravity (right). To counteract the additional load, our rest shape optimizer modifies the rest curvature more significantly.*
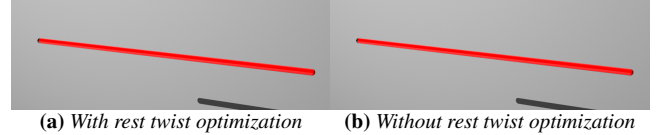

(a) *With rest twist optimization*   **(b)** *Without rest twist optimization*

Figure 6: *Stress test with a horizontal strand discretized with $1,000$ vertices. Both approaches successfully achieve static equilibrium.*

Due to the horizontal arrangement of the strands, our rest shape optimizer modifies the rest curvatures only (but not rest lengths or rest twists). While we can achieve static equilibrium with the three stiffer strands, with our optimizer modifying the rest curvature more significantly for the softer materials, the softest strand fails because it requires much larger rest curvature changes (which are not permitted by the box constraints). While increasing the stiffness is one way to reduce the sagging at equilibrium, this approach can change the strand dynamics and our perception of the material.

5.4. External Load

We also introduce an additional external load (which is a force $10\times$ larger than gravity) exactly on the tail-end vertex of the horizontal strand, as shown in Figure 5. While static equilibrium can be achieved even with the load, our method modifies the rest curvature further to cancel the force due to the load.

5.5. Stress Test

To demonstrate the capabilities of our method with high-resolution strands, we experiment with a horizontal strand discretized with $1,000$ vertices, as shown in Figure 6 (notably, hair strands are more commonly discretized with up to 100 vertices [SWP*23]). Given the horizontal structure of the strand, bending does not introduce twisting, and thus we can safely exclude the rest twist from the rest shape optimization. Our method achieved static equilibrium in both cases, i.e., including and excluding rest twist for the rest shape optimization. The computation times were 0.087 s and 0.072 s which are proportional to the number of DOFs, $(6N - 12)$ and $(5N - 10)$, respectively.
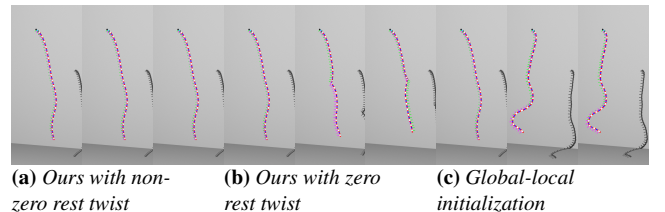

(a) *Ours with non-zero rest twist*   **(b)** *Ours with zero rest twist*   **(c)** *Global-local initialization*

Figure 7: *An experiment with a single hair-like strand. Enforcing zero rest twist (b) leads to a relatively unstable static equilibrium initially, causing flipping that yields a kink. Global-local initialization of DER (c) fails to achieve static equilibrium.*

5.6. Single Hair-Like Strand

Next, we experiment with a single hair-like strand (which can introduce non-negligible torsion effects, unlike vertical and horizontal strands), as shown in Figure 7, to evaluate our method including/excluding rest twist in the optimization, along with the global-local initialization [HTYW22]. Specifically, we compare the following schemes:

1. Ours with non-zero rest twist: we include rest twist in the optimization, which generally leads to non-zero rest twist;
2. Ours with zero rest twist: we exclude rest twist from the optimization, setting rest twist to zero;
3. Global-local initialization: [HTYW22].

Since directly applying the global-local initialization [HTYW22] to DER is not possible, as discussed in Sec. 2.2.3, we use the following two simplifications. First, we use the minimal stencil of the bending and twisting as a unified single force element to form a global linear system, assuming that the force element is linear/angular momentum preserving. Second, we first determine rest length from the stretch force without considering the bending and twisting forces to avoid transforming the local step into a global problem assuming relatively small changes in the rest length due to the large c_{stretch}. Then, we optimize the rest curvature and rest twist using the rest length and local-element forces (computed by the global linear solve) in a least-squares way.

The strand with non-zero rest twist achieves static equilibrium and retains the same shape. By contrast, while the strand with zero rest twist can achieve static equilibrium at first, it is relatively unstable and flips in its middle part to reach a more stable static equilibrium. This is because the bending forces need to cancel any torques by themselves (as the twist forces are zero by construction at first) leading to significant changes in the rest curvatures and thus the unstable static equilibrium. Due to the necessary simplifications, the global-local initialization [HTYW22] failed to achieve static equilibrium (although the global-local initialization, as described above, was able to achieve static equilibrium for perfectly vertical or horizontal strands in our experiments).

5.7. Box Constraint Evaluation

To evaluate the effectiveness of the box constraints, we experiment with the hair-like strand, as shown in Figure 8. We use two different sets of stiffness parameters ($c_{\text{bend}} = c_{\text{twist}} = 2.5 \times 10^6$ and $c_{\text{bend}} = c_{\text{twist}} = 2.5 \times 10^7$) and employ $\Delta\kappa = 0.6$ and $\Delta\kappa = 1.0$ in (35), respectively, given the very soft materials. We compare the following schemes:

1. Naive initialization;
2. No box constraints: initialized with our rest shape optimization using no box constraints;
3. With box constraints.

Without using the box constraints, the rest shape parameters can be modified significantly, leading to a relatively unstable static equilibrium and thus failing to retain the original strand shape. In particular, for softer materials, the rest shape change can be excessive, introducing stability issues in the forward simulation. By contrast, the use of box constraints can avoid introducing stability problems and achieve stable strand dynamics and states (although the strand needs to compromise static equilibrium), preserving the original shape better compared to the naive initialization.
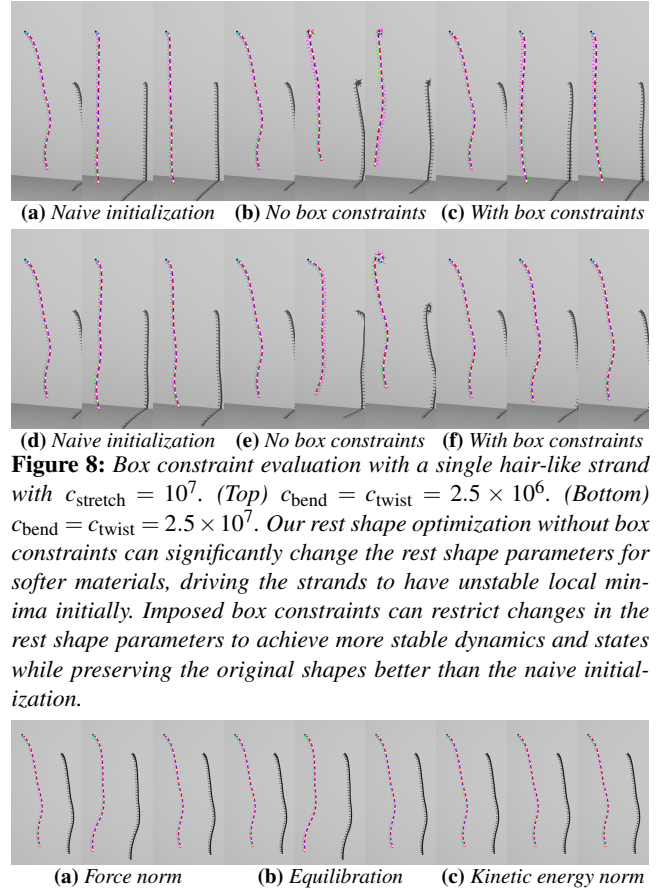

(a) *Naive initialization*   **(b)** *No box constraints*   **(c)** *With box constraints*

(d) *Naive initialization*   **(e)** *No box constraints*   **(f)** *With box constraints*

Figure 8: *Box constraint evaluation with a single hair-like strand with $c_{\text{stretch}} = 10^7$. (Top) $c_{\text{bend}} = c_{\text{twist}} = 2.5 \times 10^6$. (Bottom) $c_{\text{bend}} = c_{\text{twist}} = 2.5 \times 10^7$. Our rest shape optimization without box constraints can significantly change the rest shape parameters for softer materials, driving the strands to have unstable local minima initially. Imposed box constraints can restrict changes in the rest shape parameters to achieve more stable dynamics and states while preserving the original shapes better than the naive initialization.*

(a) *Force norm*   **(b)** *Equilibration*   **(c)** *Kinetic energy norm*

Figure 9: *Experiment with a hair-like strand. Optimization with the force norm and equilibration approach fails to achieve static equilibrium whereas our method succeeds.*

5.8. Evaluation of Norm Formulations

To demonstrate the effectiveness of our norm formulation, we experiment with the hair-like strand, as shown in Figure 9. We compare the following schemes:

1. Force norm: force-norm formulation $\|\mathbf{f}\|_2^2$ [TKA11] which forms the Hessian via $\mathbf{J}^T\mathbf{J}$;
2. Equilibration: force-norm formulation [TKA11] with symmetric equilibration [Bun71] to improve numerical conditioning of the system via $\mathbf{D}\mathbf{J}^T\mathbf{J}\mathbf{D}$ ($\mathbf{D}$: diagonal scaling matrix);
3. Kinetic energy norm: our kinetic-energy-based norm $\|\mathbf{f}\|_{\mathbf{M}^{-1}}^2$ which forms the Hessian via $\mathbf{J}^T\mathbf{M}^{-1}\mathbf{J}$.

We ignore the constant scaling of objectives for readability. Table 1 summarizes the result of the experiment. Optimization with the

Table 1: *Results at solver convergence for the experiment in Figure 9. $\min(\mathbf{A})$ and $\max(\mathbf{A})$ denote the minimum and maximum magnitude of non-zeros in the system matrix $\mathbf{A}$, respectively, σ their ratio ($\max(\mathbf{A})/\min(\mathbf{A})$), $\|\mathbf{f}\|_2$ and $\|\mathbf{f}\|_{\mathbf{M}^{-1}}$ the norm of corresponding objectives.*

Schemes	$\min(\mathbf{A})$	$\max(\mathbf{A})$	σ	$\|\mathbf{f}\|_2$	$\|\mathbf{f}\|_{\mathbf{M}^{-1}}$
Force norm	6.0×10^{-9}	8.4×10^8	1.4×10^{17}	3.1×10^{-4}	1.2×10^1
Equilibration	1.9×10^{-14}	1.0×10^0	5.4×10^{13}	3.1×10^{-4}	1.2×10^1
Kinetic energy norm	1.4×10^{-2}	2.7×10^{13}	1.9×10^{15}	4.3×10^{-8}	6.3×10^{-6}

(a) *Naive initialization*
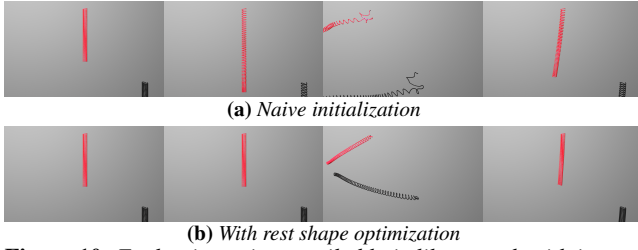

(b) *With rest shape optimization*

Figure 10: *Evaluation using a coiled hair-like strand, with its root vertex perturbed. The naive initialization fails to achieve static equilibrium and suffers from the stability issue, whereas the rest shape optimization enables sag-free and stable simulation.*

force norm fails to achieve static equilibrium because the system has entries significantly different in magnitude and thus neglects some force components. While equilibrating the matrix improved the numerical conditioning of the system with smaller variations in the matrix entries (i.e., smaller σ), it did not help to further reduce the objective values within the limited numerical precision (giving a result almost identical to the one with the force norm), because it neglects the energy contributions from each force component. By contrast, our kinetic-energy-based objective properly considers the energy contributions in the system and sufficiently reduces both the objectives $\|\mathbf{f}\|_2^2$ and $\|\mathbf{f}\|_{\mathbf{M}^{-1}}^2$, thus achieving static equilibrium.

5.9. Evaluation with Coiled Hair-Like Strand

To demonstrate that our method can handle curly strands, we experiment with a coiled hair-like strand, as shown in Figure 10, using $c_{\text{stretch}} = c_{\text{bend}} = c_{\text{twist}} = 10^9$. By simultaneously optimizing the rest length, rest curvature, and rest twist, our method achieves static equilibrium. Additionally, when the root vertices are perturbed, the optimized rest shape prevents too much strand deformation and thus stabilizes the simulation in this example, whereas the strand with the naive initialization exhibits unnatural oscillations.

5.10. Evaluation with Complex Strand Geometry

To evaluate the effectiveness of our method in more complex scenarios, we experiment with hair strand data released publicly by Hu et al. [HMLL15]. We use $c_{\text{stretch}} = 3.0 \times 10^8, c_{\text{bend}} = 3.0 \times 10^8$, and $c_{\text{twist}} = 3.0 \times 10^8$. Figure 11 compares our method with the naive initialization. While the hair strands sag due to the gravity with the naive initialization, the strands with our method retain the original hair style. When the root vertices are rotated in a prescribed way, both approaches generate natural and comparable hair motions. After the root vertices are stopped at the same position as at the start, hair strands with our approach retain the original hair shape, unlike those with the naive initialization. Figure 1 shows hair simulations using our method with another complex hair style.

In Figure 12, we use the same setting as Figure 11, but with thinner strands (strand radius $r = 10^{-4}$). In this case, rest shape optimization fails to retain the original hair shape (while the sagging is mitigated compared to naive initialization) because the strands cannot exert sufficiently strong forces to support themselves. Additionally, significant rest shape changes can lead to unstable strand configurations, causing unnatural hair lifting.

6. Conclusions and Future Work

We have proposed our rest shape optimizer to achieve sag-free DER simulation and evaluated its efficacy in various examples. In the following, we discuss tradeoffs inherent to our approach and promising research directions for future work.

6.1. Static Equilibrium at Local Maxima and Minima

Our formulation based on the kinetic energy is akin to the force-norm minimization [TKA11] (which is also related to the nonlinear force solves with inverse dynamics [Had06, Fea16], ANM [CZXZ14, Jia21], and global-local approaches [HTYW22, HWP*23]) and is designed to find a static equilibrium where forces are zero. While the achieved static equilibrium is at a local minimum for the rest shape optimization (37), it can be at a local maximum for the optimization on forward simulation (1) [DJBDT10, HTYW22]. As such, perturbations on the generalized positions can relatively easily break the static equilibrium; however, this case is rare for strands with one-sided clamping since such strands typically do not experience compression and settle at a local minimum for (1) via the rest shape optimization (37). If one needs to guarantee a static equilibrium at a local minimum with respect to both generalized positions and rest shape parameters, it seems promising to use local material stiffening [DJBDT10, HTYW22] and the adjoint-based approaches [PTC*15].

6.2. Material Stiffness Parameter Optimization

While our rest shape optimizer was able to find rest shape parameters that achieve the desired static equilibrium in various settings with reasonable material stiffness parameters, in general, it is not guaranteed that such parameters exist, e.g., when the given system is overdetermined or such parameters are outside of the box constraints (see Figures 4 and 8). In addition, when the rest shape is significantly modified to achieve the static equilibrium (even under the box constraints, which are designed to avoid introducing stability problems), the strand shape may not return to the original configuration after the strand moves dynamically because it can get stuck in different local extrema (1). In these cases, increasing the material stiffness (or equivalently increasing the strand radius) can reduce the required rest shape changes and can be helpful to establish a stabler static equilibrium (albeit compromising user-desired elastic dynamics) [DJBDT10, TKA11, HTYW22, HWP*23].

6.3. Optimization Solvers

We designed our optimization solvers by seeking a balance between performance and simplicity, but it is possible that other alternatives could potentially perform better. Although the penalty method was quite satisfactory in our examples (since small violations of box constraints or clamping were permissible), other constrained optimization solvers (e.g., augmented Lagrangian method (ALM) [NW06, TB21], interior point method (IPM) [TB24], and active-set method [DS05, TB23]) could be employed to more accurately satisfy the box constraints. Other nonlinear solvers (e.g., nonlinear conjugate gradient, quasi-Newton, and LMA [NW06]) are also available, and it would be worth comparing their performance with Gauss-Newton. While we arranged the rest shape parameters in their current order (rest length, rest curvature, rest twist) so that the rest twist can easily be added and removed from the rest shape optimization, by interleaving these parameters (as we did for the generalized positions), the inner linear systems for Newton-type optimizers would possess a banded structure that enables more efficient direct solves without reordering of the system(s).

6.4. Toward More General Inverse Problems

In our framework, for simplicity and efficiency, we assumed a strand with constant radius, density, and material stiffness. It would
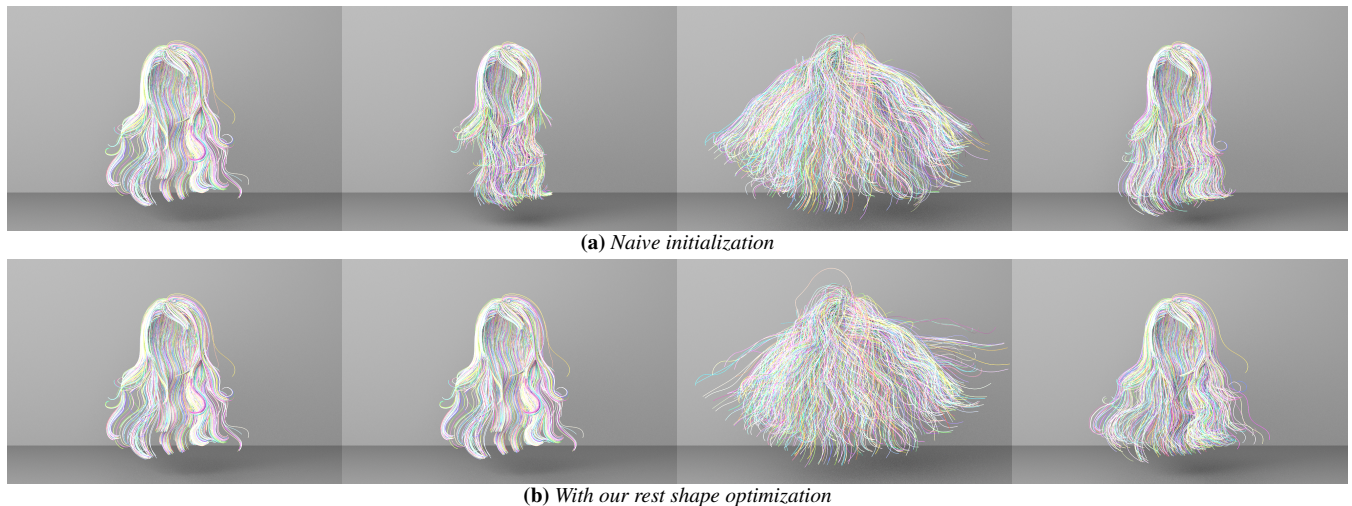
(a) *Naive initialization*

(b) *With our rest shape optimization*

Figure 11: *Hair simulation with hair assets. Hair strands sag with the naive initialization while our method enables hair strands to retain their original shapes, exhibit natural motions due to the prescribed motion of the root vertices, and then return toward the original shape.*

(a) *Initial condition*　　**(b)** *Naive initialization*　　**(c)** *With optimization*

Figure 12: *Experiment with thinner strands. With naive initialization, strands sag significantly. With rest shape optimization, strands still sag (though sagging is mitigated) and exhibit unnatural lifting.*

be worthwhile removing this assumption to support more general anisotropic and inhomogeneous strands [HB21, HB23]. Moreover, our framework should be able to extend to support two-dimensional shells and three-dimensional volumetric structures, along with more rigorous analysis for the box constraints imposed on the rest shape parameters. In addition, although we also assumed no contacts among strands or other objects, this approach can lead to larger rest shape changes than necessary in cases where the elastic strands could be supported by each other or other objects when the simulation starts [DJBDDT13, LCBD*18, HTYW22, HWP*23]. Thus, a promising extension would be to take frictional contacts into account during the rest shape optimization.

Acknowledgements

References

[ACB*13]　Audoly B., Clauvelin N., Brun P.-T., Bergou M., Grinspun E.: A discrete geometric approach for simulating the dynamics of thin viscous threads. *Journal of Computational Physics 253* (2013), 18–49. 1

[ARM*19]　Angles B., Rebain D., Macklin M., Wyvill B., Barthe L., Lewis J., Von Der Pahlen J., Izadi S., Valentin J., Bouaziz S., Tagliasacchi A.: Viper: Volume invariant position-based elastic rods. *Proc. ACM Comput. Graph. Interact. Tech. 2*, 2 (2019). 2

[ATK17]　Andrews S., Teichmann M., Kry P. G.: Geometric stiffness for real-time constrained multibody dynamics. *Comput. Graph. Forum 36*, 2 (may 2017), 235–246. 2

[AX15]　Aubry J.-M., Xian X.: Fast implicit simulation of flexible trees. In *Mathematical Progress in Expressive Image Synthesis II* (2015), Ochiai H., Anjyo K., (Eds.), Springer Japan, pp. 47–61. 2

[BAC*06]　Bertails F., Audoly B., Cani M.-P., Querleux B., Leroy F., Lévêque J.-L.: Super-helices for predicting the dynamics of natural hair. *ACM Trans. Graph. 25*, 3 (2006), 1180–1187. 2

[Bar96]　Baraff D.: Linear-time dynamics using lagrange multipliers. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), pp. 137–146. 2

[BAV*10]　Bergou M., Audoly B., Vouga E., Wardetzky M., Grinspun E.: Discrete viscous threads. *ACM Transactions on Graphics 29*, 4 (2010), 116:1–116:10. 1, 2, 3, 4, 5, 7

[BDDJRL18]　Bertails-Descoubes F., Derouet-Jourdan A., Romero V., Lazarus A.: Inverse design of an isotropic suspended Kirchhoff rod: theoretical and numerical results on the uniqueness of the natural shape. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474*, 2212 (Apr. 2018), 1–26. 2

[BML*14]　Bouaziz S., Martin S., Liu T., Kavan L., Pauly M.: Projective dynamics: Fusing constraint projections for fast simulation. *ACM Trans. Graph. 33*, 4 (2014), 154:1–154:11. 2

[BMM17]　Bender J., Müller M., Macklin M.: A survey on position based dynamics, 2017. In *Proceedings of the European Association for Computer Graphics: Tutorials* (2017), EG '17. 2

[BOFN18]　Brown G. E., Overby M., Forootaninia Z., Narain R.: Accurate dissipative forces in optimization integrators. *ACM Trans. Graph. 37*, 6 (dec 2018). 4

[Bun71]　Bunch J. R.: Equilibration of symmetric matrices in the max-norm. *J. ACM 18*, 4 (Oct. 1971), 566–572. 10

[BW98]　Baraff D., Witkin A.: Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, Association for Computing Machinery, p. 43–54. 4

[BWR*08]　Bergou M., Wardetzky M., Robinson S., Audoly B., Grinspun E.: Discrete elastic rods. In *ACM SIGGRAPH 2008 Papers*

(New York, NY, USA, 2008), SIGGRAPH '08, Association for Computing Machinery. 1, 2, 3, 4, 5, 6, 7

[CAR*09] CHENTANEZ N., ALTEROVITZ R., RITCHIE D., CHO L., HAUSER K. K., GOLDBERG K., SHEWCHUK J. R., O'BRIEN J. F.: Interactive simulation of surgical needle insertion and steering. *ACM Trans. Graph. 28*, 3 (jul 2009). 1

[CBD13] CASATI R., BERTAILS-DESCOUBES F.: Super space clothoids. *ACM Trans. Graph. 32*, 4 (jul 2013). 2

[CBDNR20] CHARRONDIÈRE R., BERTAILS-DESCOUBES F., NEUKIRCH S., ROMERO V.: Numerical modeling of inextensible elastic ribbons with curvature-based elements. *Computer Methods in Applied Mechanics and Engineering 364* (2020), 112922. 2

[CJSJ24] CHOI A., JING R., SABELHAUS A. P., JAWED M. K.: Dismech: A discrete differential geometry-based physical simulator for soft robots and structures. *IEEE Robotics and Automation Letters 9*, 4 (2024), 3483–3490. 3

[CZB*24] CHEN Y., ZHANG Y., BREI Z., ZHANG T., CHEN Y., WU J., VASUDEVAN R.: Differentiable discrete elastic rods for real-time modeling of deformable linear objects, 2024. arXiv:2406.05931. 1

[CZXZ14] CHEN X., ZHENG C., XU W., ZHOU K.: An asymptotic numerical method for inverse elastic shape design. *ACM Trans. Graph. 33*, 4 (jul 2014). 2, 11

[Dav23] DAVIET G.: Interactive hair simulation on the gpu using admm. In *ACM SIGGRAPH 2023 Conference Proceedings* (2023). 1

[DJBDDT13] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., DAVIET G., THOLLOT J.: Inverse dynamic hair modeling with frictional contact. *ACM Trans. Graph. 32*, 6 (2013), 159:1–159:10. 2, 12

[DJBDT10] DEROUET-JOURDAN A., BERTAILS-DESCOUBES F., THOLLOT J.: Stable inverse dynamic curves. *ACM Trans. Graph. 29*, 6 (dec 2010). 2, 11

[DKH10] DIVERDI S., KRISHNASWAMY A., HADAP S.: Industrial-strength painting with a virtual bristle brush. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2010), VRST '10, Association for Computing Machinery, p. 119–126. 1

[DKWB18] DEUL C., KUGELSTADT T., WEILER M., BENDER J.: Direct position-based solver for stiff rods. *Computer Graphics Forum 37*, 6 (2018), 313–324. 2

[DS05] DOSTAL Z., SCHOBERL J.: Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Computational Optimization and Applications 30*, 1 (2005), 23–43. 11

[FBGZ19] FEI Y. R., BATTY C., GRINSPUN E., ZHENG C.: A multi-scale model for coupling strands with shear-dependent liquid. *ACM Trans. Graph. 38*, 6 (Nov. 2019). 1, 3, 5

[Fea16] FEATHERSTONE R.: *Rigid Body Dynamics Algorithms*. 2016. 2, 11

[GB15] GORNOWICZ G., BORAC S.: Efficient and stable approach to elasticity and collisions for hair animation. In *Proceedings of the 2015 Symposium on Digital Production* (New York, NY, USA, 2015), DigiPro '15, Association for Computing Machinery, p. 41–49. 5

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3 (jul 2007), 49–es. 1

[GSS*15] GAST T. F., SCHROEDER C., STOMAKHIN A., JIANG C., TERAN J. M.: Optimization integrator for large time steps. *IEEE Transactions on Visualization and Computer Graphics 21*, 10 (2015), 1103–1115. 2, 4

[Had06] HADAP S.: Oriented strands: dynamics of stiff multi-body system. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), SCA '06, p. 91–100. 2, 11

[HB21] HAFNER C., BICKEL B.: The design space of plane elastic curves. *ACM Trans. Graph. 40*, 4 (jul 2021). 2, 12

[HB23] HAFNER C., BICKEL B.: The design space of kirchhoff rods. *ACM Trans. Graph. 42*, 5 (sep 2023). 2, 4, 12

[HH13] HAN D., HARADA T.: Tridiagonal Matrix Formulation for Inextensible Hair Strand Simulation. In *Workshop on Virtual Reality Interaction and Physical Simulation* (2013), The Eurographics Association. 2

[HMLL15] HU L., MA C., LUO L., LI H.: Single-view hair modeling using a hairstyle database. *ACM Trans. Graph. 34*, 4 (jul 2015). 7, 11, 12

[HTYW22] HSU J., TRUONG N., YUKSEL C., WU K.: A general two-stage initialization for sag-free deformable simulations. *ACM Trans. Graph. 41*, 4 (jul 2022). 3, 10, 11, 12

[HWP*23] HSU J., WANG T., PAN Z., GAO X., YUKSEL C., WU K.: Sag-free initialization for strand-based hybrid hair simulation. *ACM Trans. Graph. 42*, 4 (jul 2023). 1, 2, 3, 11, 12

[HYW*23] HUANG L., YANG F., WEI C., CHEN Y. J. E., YUAN C., GAO M.: Towards realtime: A hybrid physics-based method for hair animation on gpu. *Proc. ACM Comput. Graph. Interact. Tech. 6*, 3 (2023). 3, 4

[HZS*24] HERRERA J. A. A., ZHOU Y., SUN X., SHU Z., HE C., PIRK S., MICHELS D. L.: Digitalsalon: A fast simulator for dense hair via augmented mass-spring model. 2

[IBB19] IBEN H., BROOKS J., BOLWYN C.: Holding the shape in hair simulation. In *ACM SIGGRAPH 2019 Talks* (New York, NY, USA, 2019), SIGGRAPH '19, Association for Computing Machinery. 1, 2

[IMP*13] IBEN H., MEYER M., PETROVIC L., SOARES O., ANDERSON J., WITKIN A.: Artistic simulation of curly hair. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2013), SCA '13, Association for Computing Machinery, p. 63–71. 2

[Jia21] JIA K.: Sanm: a symbolic asymptotic numerical solver with applications in mesh deformation. *ACM Trans. Graph. 40*, 4 (2021). 2, 11

[JNO18] JAWED M., NOVELIA A., O'REILLY O.: *A Primer on the Kinematics of Discrete Elastic Rods*. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, 2018. 3, 4, 5

[KE22] KIM T., EBERLE D.: Dynamic deformables: implementation and production practicalities (now with code!). In *ACM SIGGRAPH 2022 Courses* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. 1, 4

[KJM10] KALDOR J. M., JAMES D. L., MARSCHNER S.: Efficient yarn-based cloth with adaptive contact linearization. *ACM Trans. Graph. 29*, 4 (jul 2010). 3, 5

[KS16] KUGELSTADT T., SCHÖMER E.: Position and orientation based Cosserat rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), p. 169–178. 2, 3, 7

[KTS*14] KAUFMAN D. M., TAMSTORF R., SMITH B., AUBRY J.-M., GRINSPUN E.: Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Trans. Graph. 33*, 4 (jul 2014). 1

[LBOK13] LIU T., BARGTEIL A. W., O'BRIEN J. F., KAVAN L.: Fast simulation of mass-spring systems. *ACM Trans. Graph. 32*, 6 (2013), 214:1–214:7. 2

[LCBD*18] LY M., CASATI R., BERTAILS-DESCOUBES F., SKOURAS M., BOISSIEUX L.: Inverse elastic shell design with contact and friction. *ACM Trans. Graph. 37*, 6 (dec 2018). 2, 12

[Liu22] LIU H.: Gravity preloading for maintaining hair shape using the simulator as a closed-box function. In *ACM SIGGRAPH 2022 Talks* (2022), SIGGRAPH '22. 1, 3

[LLR13] LUO L., LI H., RUSINKIEWICZ S.: Structure-aware hair capture. *ACM Trans. Graph. 32*, 4 (jul 2013). 1, 7

[LMR13] LAZARUS A., MILLER J., REIS P.: Continuation of equilibria and stability of slender elastic rods using an asymptotic numerical

method. *Journal of the Mechanics and Physics of Solids 61*, 8 (2013), 1712–1736. 2

[LSD*22] LESSER S., STOMAKHIN A., DAVIET G., WRETBORN J., EDHOLM J., LEE N.-H., SCHWEICKART E., ZHAI X., FLYNN S., MOFFAT A.: Loki: A unified multiphysics simulation framework for production. *ACM Trans. Graph. 41*, 4 (jul 2022). 1, 2

[MC11] MÜLLER M., CHENTANEZ N.: Solid simulation with oriented particles. *ACM Trans. Graph. 30*, 4 (jul 2011). 2

[MCJM18] MÜLLER M., CHENTANEZ N., JESCHKE S., MACKLIN M.: Cable joints. *Computer Graphics Forum 37*, 8 (2018), 1–10. 2

[MCMJ17] MÜLLER M., CHENTANEZ N., MACKLIN M., JESCHKE S.: Long range constraints for rigid body simulations. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (2017), SCA '17, Association for Computing Machinery. 2

[MHHR07] MÜLLER M., HEIDELBERGER B., HENNIX M., RATCLIFF J.: Position based dynamics. *Journal of Visual Communication Image Representation 18*, 2 (2007), 109–118. 2

[MKB*10] MARTIN S., KAUFMANN P., BOTSCH M., GRINSPUN E., GROSS M.: Unified simulation of elastic rods, shells, and solids. *ACM Trans. Graph. 29*, 4 (jul 2010). 2

[MPI*18] MALOMO L., PÉREZ J., IARUSSI E., PIETRONI N., MIGUEL E., CIGNONI P., BICKEL B.: Flexmaps: computational design of flat flexible shells for shaping 3d objects. *ACM Trans. Graph. 37*, 6 (2018). 1, 2

[MTGG11] MARTIN S., THOMASZEWSKI B., GRINSPUN E., GROSS M.: Example-based elastic materials. *ACM Trans. Graph. 30*, 4 (2011), 72:1–72:8. 2, 4

[MWW18] MUKHERJEE R., WU L., WANG H.: Interactive two-way shape design of elastic bodies. *Proc. ACM Comput. Graph. Interact. Tech. 1*, 1 (jul 2018). 2

[NOB16] NARAIN R., OVERBY M., BROWN G. E.: Admm ⊇ rojective dynamics: Fast simulation of general constitutive models. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2016), SCA '16, Eurographics Association, p. 21–28. 4

[NPTB22] NEVEU W., PUHACHOV I., THOMASZEWSKI B., BESSMELTSEV M.: Stability-aware simplification of curve networks. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. 1

[NW06] NOCEDAL J., WRIGHT S. J.: *Numerical Optimization*, second ed. Springer, New York, NY, USA, 2006. 2, 4, 7, 11

[Pai02] PAI D. K.: STRANDS: Interactive Simulation of Thin Solids using Cosserat Models. *Computer Graphics Forum* (2002). 2

[PKLI*19] PANETTA J., KONAKOVIĆ-LUKOVIĆ M., ISVORANU F., BOULEAU E., PAULY M.: X-shells: a new class of deployable beam structures. *ACM Trans. Graph. 38*, 4 (jul 2019). 1, 2, 3, 5

[POT17] PÉREZ J., OTADUY M. A., THOMASZEWSKI B.: Computational design and automated fabrication of kirchhoff-plateau surfaces. *ACM Trans. Graph. 36*, 4 (jul 2017). 1, 2

[PTC*15] PÉREZ J., THOMASZEWSKI B., COROS S., BICKEL B., CANABAL J. A., SUMNER R., OTADUY M. A.: Design and fabrication of flexible rod meshes. *ACM Trans. Graph. 34*, 4 (2015). 1, 2, 4, 11

[RKP*22] REN Y., KUSUPATI U., PANETTA J., ISVORANU F., PELLIS D., CHEN T., PAULY M.: Umbrella meshes: elastic mechanisms for freeform shape deployment. *ACM Trans. Graph. 41*, 4 (jul 2022). 1

[RLR*21] ROMERO V., LY M., RASHEED A. H., CHARRONDIÈRE R., LAZARUS A., NEUKIRCH S., BERTAILS-DESCOUBES F.: Physical validation of simulators in computer graphics: a new framework dedicated to slender elastic structures and frictional contact. *ACM Trans. Graph. 40*, 4 (jul 2021). 3

[SBRBO20] SÁNCHEZ-BANDERAS R. M., RODRÍGUEZ A., BARREIRO H., OTADUY M. A.: Robust eulerian-on-lagrangian rods. *ACM Trans. Graph. 39*, 4 (aug 2020). 2

[SJLP11] SUEDA S., JONES G. L., LEVIN D. I. W., PAI D. K.: Large-scale dynamic simulation of highly constrained strands. *ACM Trans. Graph. 30*, 4 (jul 2011). 2

[SJM17] SCHWEICKART E., JAMES D. L., MARSCHNER S.: Animating elastic rods with sound. *ACM Trans. Graph. 36*, 4 (jul 2017). 1

[SLF08] SELLE A., LENTINE M., FEDKIW R.: A mass spring model for hair simulation. *ACM Trans. Graph. 27*, 3 (aug 2008), 1–11. 2, 7

[SMGT18] SCHUMACHER C., MARSCHNER S., GROSS M., THOMASZEWSKI B.: Mechanical characterization of structured sheet materials. *ACM Trans. Graph. 37*, 4 (jul 2018). 1

[SMSH18] SOLER C., MARTIN T., SORKINE-HORNUNG O.: Cosserat rods with projective dynamics. *Computer Graphics Forum 37*, 8 (2018), 137–147. 2, 7

[ST07] SPILLMANN J., TESCHNER M.: 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* (2007), SCA '07, Eurographics Association, p. 63–72. 2

[ST09] SPILLMANN J., TESCHNER M.: Cosserat nets. *IEEE Transactions on Visualization and Computer Graphics 15*, 2 (2009), 325–338. 2

[STBG12] SKOURAS M., THOMASZEWSKI B., BICKEL B., GROSS M.: Computational design of rubber balloons. *Comput. Graph. Forum 31*, 2pt4 (may 2012), 835–844. 2

[SWP*23] SHI A., WU H., PARR J., DARKE A. M., KIM T.: Lifted curls: A model for tightly coiled hair simulation. *Proc. ACM Comput. Graph. Interact. Tech. 6*, 3 (aug 2023). 2, 3, 4, 7, 9

[TB21] TAKAHASHI T., BATTY C.: Frictionalmonolith: A monolithic optimization-based approach for granular flow with contact-aware rigid-body coupling. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–16. 7, 11

[TB23] TAKAHASHI T., BATTY C.: A multilevel active-set preconditioner for box-constrained pressure poisson solvers. *Proc. ACM Comput. Graph. Interact. Tech. 6*, 3 (aug 2023). 11

[TB24] TAKAHASHI T., BATTY C.: A primal-dual box-constrained qp pressure poisson solver with topology-aware geometry-inspired aggregation amg. *IEEE Transactions on Visualization and Computer Graphics* (2024), 1–12. 11

[TKA11] TWIGG C. D., KAČIĆ-ALESIĆ Z.: Optimization for sag-free simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), pp. 225–236. 1, 2, 3, 6, 8, 10, 11

[TNGF15] TOURNIER M., NESME M., GILLES B., FAURE F.: Stable constrained dynamics. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–10. 2

[USS14] UMETANI N., SCHMIDT R., STAM J.: Position-based elastic rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2014), SCA '14, p. 21–30. 2, 7

[WCU*20] WEN J., CHEN J., UMETANI N., BAO H., HUANG J.: Cosserat rod with rh-adaptive discretization. *Computer Graphics Forum 39*, 7 (2020), 143–154. 2

[WWY*15] WANG B., WU L., YIN K., ASCHER U., LIU L., HUANG H.: Deformation capture and modeling of soft objects. *ACM Trans. Graph. 34*, 4 (July 2015), 94:1–94:12. 2

[XKCB18] XU H., KNOOP E., COROS S., BÄCHER M.: Bend-it: design and fabrication of kinetic wire characters. *ACM Trans. Graph. 37*, 6 (2018). 1

[ZCT16] ZEHNDER J., COROS S., THOMASZEWSKI B.: Designing structurally-sound ornamental curve networks. *ACM Trans. Graph. 35*, 4 (jul 2016). 1, 2

[ZLW*22] ZHAO C., LIN J., WANG T., BAO H., HUANG J.: Efficient and stable simulation of inextensible cosserat rods by a compact representation. *Computer Graphics Forum 41*, 7 (2022), 567–578. 2