Lifted Curls: A Model for Tightly Coiled Hair Simulation

ALVIN SHI*, HAOMIAO WU*, and JARRED PARR, Yale University, USA
A.M. DARKE, University of California, Santa Cruz, USA
THEODORE KIM, Yale University, USA

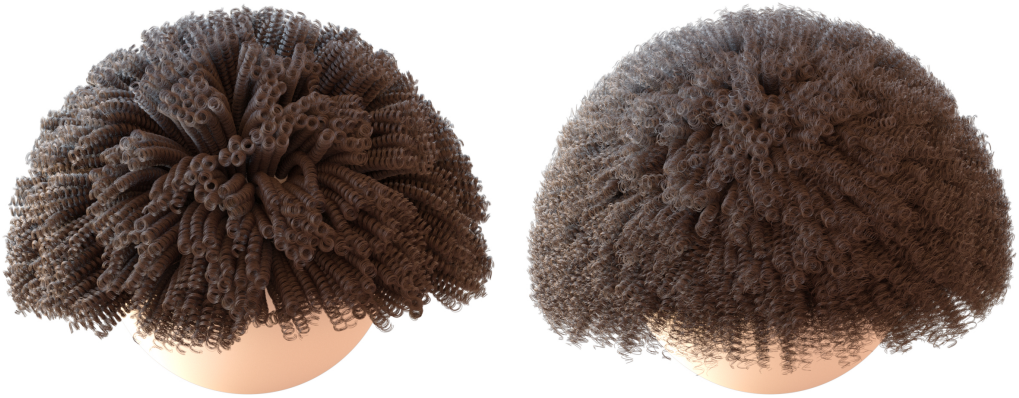


Fig. 1. We efficiently simulate tight curls on a hairball. 200K hairs generated from 2000 simulated wisps naturally form a "clumped" look (left), while 8000 simulated wisps (right) produces a more "frizzy" or "picked out" look. Both were run with $\Delta t = 1/30$ s, and respectively took 17.7 and 57.0 seconds per frame.

We present an isotropic, hyperelastic model specifically designed for the efficient simulation of tightly coiled hairs whose curl radii approach 5 mm. Our model is robust to large bends and torsions, even when they appear at the scale of the strand discretization. The terms of our model are consistently quadratic with respect to their primary variables, do not require per-edge frames or any parallel transport operators, and can efficiently take large timesteps on the order of $\sim 1/30$ of a second. Additionally, we show that it is possible to obtain fast, closed-form eigensystems for all the terms in the energy. Our eigenanalysis is sufficiently generic that it generalizes to other models. Our entirely vertex-based formulation integrates naturally with existing finite element codes, and we demonstrate its efficiency and robustness in a variety of scenarios.

ACM Reference Format:
Alvin Shi, Haomiao Wu, Jarred Parr, A.M. Darke, and Theodore Kim. 2023. Lifted Curls: A Model for Tightly Coiled Hair Simulation. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 2, Article 1 (August 2023), 19 pages. https://doi.org/10.1145/3606920

*Joint first authors

Authors' addresses: Alvin Shi, alvin.shi@yale.edu; Haomiao Wu, haomiao.wu@yale.edu; Jarred Parr, jarred.parr@yale.edu, Yale University, 51 Prospect Street, New Haven, CT, 06511-8937, USA; A.M. Darke, darke@ucsc.edu, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA, 95064, USA; Theodore Kim, theodore.kim@yale.edu, Yale University, USA.

2577-6193/2023/8-ART1
https://doi.org/10.1145/3606920

1 INTRODUCTION

We present a fast, robust, isotropic hyperelastic model specifically designed for the simulation of *tightly coiled* strands. Our goal is an efficient model that can simulate a full head of hair containing tight curls with a radius approaching ~5 millimeters (Fig. 1). Such hair is known as Types 3-4 [Walker 1997], Types V-VIII [De La Mettrie et al. 2007], kinky, coily, or Afro-textured hair.

Prior models for *curly* hair [Iben et al. 2013; Selle et al. 2008] tend to be designed for coils that are Type 2 [Walker 1997] or Types III-IV [De La Mettrie et al. 2007], which are significantly looser than the coils we examine. We are only aware of one other work in graphics that attempts a full-head simulation of hair within the "tightly coiled" regime [Bertails et al. 2006].

Cosserat-based models [Bergou et al. 2010; Pai 2002] tend to be demonstrated on straight or wavy hair, where a single hair strand is well-approximated by twenty [Gornowicz and Borac 2015] to thirty [Daviet 2020; Kaufman et al. 2014] line segments. Significantly denser discretizations, e.g. ~100 segments per hair, are needed to resolve the spatial frequencies we focus on here, but when large bends and torsions appear at scales near the strand discretization, they tend to exacerbate well-known numerical difficulties [Cirio et al. 2014; Gornowicz and Borac 2015] in existing models. The cost of increasing the strand resolution to obviate these issues becomes prohibitive.

We instead present a model specifically designed for tightly coiled hair. In particular, we robustly account for large bends and torsions near the discretization resolution by assuming that the rest state of each strand is *non-straight*: that is, the initial vertices are not co-planar. This assumption allows us to forgo the usual Cosserat approach [Pai 2002] of per-edge frames. Distinct from *twist*, which is defined with respect to edge frames, we compute *torsion* directly from vertex positions. Since hair is a thin strand with one clamped end, we have found this to be sufficient .

We refer to our model as *Lifted Curls*, because it consists of a 1D *stretching* energy, a bending term analogous to a 2D *membrane* energy, and a torsion term that recalls a 3D *volume* energy. In particular, the 3D term allows the strand to robustly resolve angle ambiguities that appear during large deformation. We target large, efficient timesteps on the order of $1/30$ seconds, such as those used in fully implicit production simulators [Kim and Eberle 2022]. These schemes require energy Hessians with all their negative eigenvalues removed. To this end, we obtain compact, closed-form eigensystems for every term in our energy, which allows the eigenvalues to be efficiently filtered. Our analysis is general, and can be used to improve the stability of previous hair models [Iben et al. 2013; Selle et al. 2008].

Our contributions are as follows:

- A fast, robust hyperelastic energy for tightly coiled hair, formulated as a 1D stretching energy, a membrane-like 2D bending energy, and a volume-like 3D torsion energy.
- Generic, closed-form eigensystems for strand-based stretching energies.
- Generic, closed-form eigensystems for unitary, angle-based isotropic bending energies.
- A closed-form 3D eigensystem for our torsion energy.

Our formulation operates solely on vertex positions, and is straightforward to incorporate into existing cloth and volume simulators.

2 RELATED WORK

Hair simulation is a classic computer graphics topic going back to at least Anjyo et al. [1992]. Most modern treatments incorporate Cosserat's theory of rods, which was first introduced to graphics by Pai [2002], and extended to include dynamics by Spillmann and Teschner [2007]. A key component of Cosserat theory is an orthonormal basis of directors defined along the strand, which allows torsion to be tracked even when the initial strand is straight.

Perhaps the most popular Cosserat-style model is Discrete Elastic Rods [Bergou et al. 2008], which formulates its directors using parallel transport, and tracks a twist-free Bishop frame. An implicit version was also proposed [Bergou et al. 2010], and has appeared in three film production systems [Gornowicz and Borac 2015; Lesser et al. 2022; Thyng et al. 2017]. This is the main model we will compare against, in particular the open source implementation from the original authors (Bergou) provided in Jawed et al. [2018]. While popular, practitioners have reported challenges with the model, including the need to smooth away singularities [Gornowicz and Borac 2015], and instabilities with BDF-2 under. a single Newton step [Eberle 2022]. We find that these problems are exacerbated in the case of tightly coiled hair, when large bends and torsions appear at the same spatial scale as the underlying discretization. Our model is designed to address these limitations.

We present an *isotropic* strand model, seemingly at odds with our goal of simulating tightly coiled hair, as curliness commonly correlates with hair containing elliptical (anisotropic) cross-sections [Bryant and Porter 2012]. However, recent studies have found that the distribution of different cell groups in a follicle is the actual dominant mechanical factor [Wortmann et al. 2020] that produces curliness (not anisotropy)', so an isotropic strand should suffice.

Position-Based Dynamics (PBD) approaches based on Cosserat theory have also been proposed. Umetani et al. [2015] used ghost points to track edge orientations, while Kugelstadt and Schömer [2016] used quaternions. Deul et al. [2018] accelerated solves on tree-structured assemblies, while Soler et al. [2018] showed how to introduce orientation into Projective Dynamics. Our contribution is orthogonal to this class of solvers, which could potentially benefit from our novel strand energy.

Spring-mass approaches also forgo the Cosserat theory, and have had success in two film production systems [Iben et al. 2013; Ward et al. 2010]. These approaches all use explicit time integration, which introduce significant timestep restrictions. Selle et al. [2008] uses implicit damping and explicit material forces, so the restrictions remain.

Volume methods have been employed to resolve dynamics in these simulations [McAdams et al. 2009], but specifically for straight hair, not the tight curls that we target. Hair Meshes [Yuksel et al. 2009] is a volumetric approach to styling hair, but is designed for, and demonstrated on, straight or wavy hair. The accompanying simulation method [Wu and Yuksel 2016] targets hair that is well-approximated by cloth, i.e. very flat, straight hair. We instead target tightly coiled hair.

Higher order spatial methods such as super-helices [Bertails et al. 2006] have their own successes, especially when their quadratic complexity becomes linear using a recursive solver [Bertails 2009]. We instead target spatially linear elements, as these are standard in film production [Kim and Eberle 2022; Lesser et al. 2022; McAdams et al. 2011]. In lieu of resolving linear vs. quadratic spatial discretization mismatches and iterative vs. recursive solver mismatches, we can instead simply inject our new FEM forces into an existing solver.

3 LIFTED CURLS ENERGY

In this section, we present our *Lifted Curls* energy. The stretching and bending terms will be familiar from other models, but the torsion term will be less familiar. We will write each in a way that facilitates eigenanalysis in the next section. The overall energy is:

$$\Psi_l = \Psi_s + \Psi_b + \Psi_t \tag{1}$$

where $\Psi_{s,b,t}$ respectively refer to stretching, bending, and torsion.

Notation: We adopt the following conventions. Scalars are unbolded, lowercase letters (k), vectors are bolded lowercase letters ($\mathbf{k}$), and matrices are bolded uppercase letters ($\mathbf{K}$). The current position of a vertex i is denoted with $\mathbf{x}_i$, while the original rest-space position has an overbar, $\bar{\mathbf{x}}_i$. Edges between vertices are denoted $\mathbf{e}_i$.

3.1 Stretching Energy

We use the stretching energy common to most strand models [Bergou et al. 2008; Sánchez-Banderas et al. 2020; Sueda et al. 2011]:

$$\Psi_s = \frac{\mu_s}{2}\left(I_s - 1\right)^2, \tag{2}$$

but write it using an $I_s \in \Re$ stretching invariant, defined using the 1D deformation gradient $\mathbf{d}$. Given the current positions $\mathbf{x}_0$ and $\mathbf{x}_1$ of an edge, and their rest-state positions $\bar{\mathbf{x}}_0$ and $\bar{\mathbf{x}}_1$, the deformation gradient is:

$$\mathbf{d} = \frac{\mathbf{x}_1 - \mathbf{x}_0}{\|\bar{\mathbf{x}}_0 - \bar{\mathbf{x}}_1\|}. \tag{3}$$

The stretch invariant is then $I_s = \|\mathbf{d}\|_2 = \sqrt{\mathbf{d}^\top \mathbf{d}}$.

3.2 Bending Energy

3.2.1 Energy Formulation. We use an isotropic, angle-based bending energy commonly found in other strand models [Cirio et al. 2014; Sánchez-Banderas et al. 2020; Sperl et al. 2022; Sueda et al. 2011]:

$$\Psi_b = \frac{\mu_b}{2}\left(\theta - \theta_0\right)^2. \tag{4}$$

We define θ in terms of three consecutive vertices on a strand, $\mathbf{x}_0$, $\mathbf{x}_1$, and $\mathbf{x}_2$. The edges between them are denoted

$$\mathbf{e}_0 = \mathbf{x}_0 - \mathbf{x}_1 \qquad\qquad \mathbf{e}_1 = \mathbf{x}_2 - \mathbf{x}_1, \tag{5}$$

which we will see in §4.2.1 becomes a *cloth-like* formulation. We measure the angle between the edges as $\theta = \operatorname{acos}\left(\frac{\mathbf{e}_0^\top \mathbf{e}_1}{\|\mathbf{e}_0\|\|\mathbf{e}_1\|}\right)$, where θ_0 is the angle between the rest-state edges. This differs slightly from the usual definitions based on"turning angle" $\approx \pi - \theta$, but will facilitate analysis later. Robust-seeming $\operatorname{atan2}(y, x)$ formulations do not apply here, because y is always positive. In the absence of a third orienting vector (e.g. the hinge edge in cloth bending), the $(0, \pi)$ vs. $(-\pi, 0)$ ambiguity cannot be resolved. If acos returns $\theta = \pi/4$, it is unknown whether the angle should be $\pi/4$, or if an extreme deformation has occurred and the angle should actually be $-\pi/4$. We will resolve this issue using our torsion energy in §3.3.

3.2.2 Relationship to DER. For two reasons, we avoid tan-based energies [Bergou et al. 2008] like $\Psi_{\tan} = \frac{\mu_b}{2}\left(\tan\left(\frac{\theta}{2}\right) - \tan\left(\frac{\theta_0}{2}\right)\right)$. First, as others have observed, the singularity at $\theta = \pi$ introduces robustness issues. Cirio et al. [2014] reported undesirably large forces appearing when simulating yarn, and instead advocated for Eqn. 4. Gornowicz and Borac [2015] also found the large forces undesirable, and smoothed them with an approximate linearization (their Fig. 4). Second, tan-based energies yield a near-linear response for straight rods where $\theta_0 \approx 0$, but for tight curls where θ_0 begins to approach π, the tan asymptote starts to dominate. The force response, even near the rest state, becomes unappealingly exponential.

3.3 Torsion Energy

3.3.1 Energy Formulation. We introduce a novel torsion energy

$$\Psi_t = \frac{\mu_t}{2}\left(\tau - \tau_0\right)^2, \tag{6}$$

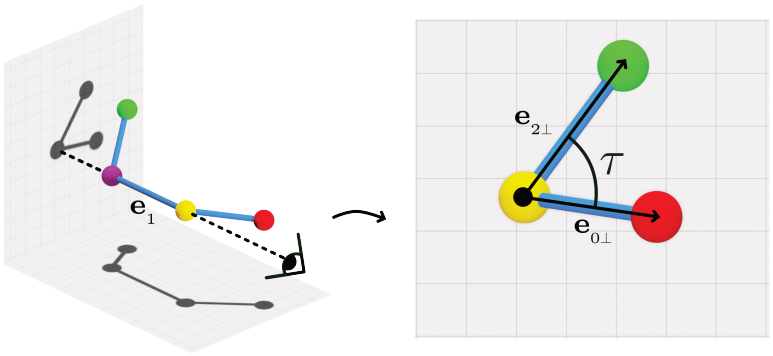


Fig. 2. Geometry of our torsion energy Ψ_t along a tetrad of vertices. We project the vertices onto the plane orthogonal to the middle edge, $\mathbf{e}_1$ (left), and obtain the projected edges $\mathbf{e}_{0\perp}$ and $\mathbf{e}_{2\perp}$ (right). The torsion angle τ is then the in-plane angle between these edges.

which is applied to each *tetrad* of vertices along a strand, and produces a *volume-like* formulation. Following Eqn. 5, the edges between the vertices are

$$\mathbf{e}_0 = \mathbf{x}_0 - \mathbf{x}_1 \qquad \mathbf{e}_1 = \mathbf{x}_2 - \mathbf{x}_1 \qquad \mathbf{e}_2 = \mathbf{x}_3 - \mathbf{x}_2.$$

The torsion τ along edge $\mathbf{e}_1$ is computed by projecting $\mathbf{e}_0$ and $\mathbf{e}_2$ onto its orthogonal plane (see Fig. 2):

$$\mathbf{e}_{0\perp} = \mathbf{e}_0 - \frac{\mathbf{e}_0^\top \mathbf{e}_1}{\|\mathbf{e}_1\|^2}\mathbf{e}_1 \qquad \mathbf{e}_{2\perp} = \mathbf{e}_2 - \frac{\mathbf{e}_2^\top \mathbf{e}_1}{\|\mathbf{e}_1\|^2}\mathbf{e}_1. \tag{7}$$

We can then assemble a deformation gradient-like matrix

$$\mathbf{W} = \left[\begin{array}{c|c|c} & & \\ \mathbf{e}_{0\perp} & \mathbf{e}_{2\perp} & \mathbf{e}_1 \\ & & \end{array}\right] \in \mathfrak{R}^{3\times 3} \tag{8}$$

where the torsion angle τ is defined as

$$\tau = -\mathcal{S}\left(\det \mathbf{W}\right) \cdot \operatorname{acos}\left(\frac{\left(\mathbf{e}_{0\perp}\right)^\top \mathbf{e}_{2\perp}}{\|\mathbf{e}_{0\perp}\|\|\mathbf{e}_{2\perp}\|}\right) \tag{9}$$

and $\mathcal{S}(x)$ is a sign function:

$$\mathcal{S}(x) = \begin{cases} -1 & \text{if } x \le 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{10}$$

Similar to bending, τ_0 denotes torsion at the rest state. The acos term resembles θ from bending, but measures the torsion induced by the two edges *orthogonal* to the bending direction. This removes a troublesome, non-physical nullspace from the system, because any triplet $\mathbf{x}_{0,1,2}$ rotated about the axis $\mathbf{e}_0$ yields the same bending energy. The addition of $\mathbf{x}_3$ and Ψ_t instead ensure that one particular rotation becomes energetically preferred.

The $\mathcal{S}\left(\det \mathbf{W}\right)$ resolves the $(0, \pi)$ vs. $(-\pi, 0)$ ambiguity. Four non-co-planar vertices form a tetrahedron, so negative volume becomes well-defined. Similar to how $\det \mathbf{F}$ is used to detect when a volumetric element has inverted [Smith et al. 2018], an negative $\det \mathbf{W}$ unambiguously signals that extreme deformation has occurred, and that the strand has torqued outside the $(0, \pi)$ range. Instead of using tan-based barriers [Bergou et al. 2008] to prohibit this ambiguity, our energy is

well-defined over $(-\pi, \pi)$. Our approach is closely related to atan2(y, x), as $\det \mathbf{W}$ corresponds to the triple product for y.

3.3.2 Relationship to DER. Our torsion force can disappear if the simulation coerces the strand into a line (i.e. $\|\mathbf{e}_{0\perp}\| = \|\mathbf{e}_{2\perp}\| = 0$). However, this is consistent with DER [Bergou et al. 2008], where the twisting force is defined in terms of the binormal. This also disappears when neighboring segments become parallel, as the cross-product of parallel segments is a zero vector. In this configuration, both models assume that stretching and bending forces will dominate.

While our energy assumes a non-straight *rest state*, it handles straight *deformed states* robustly. As can be seen in §4.3 of the supplement to Panetta et al. [2019], this more difficult for DER-like parallel transport-based methods, because differentiating the parallel transport operator produces a tan term which becomes singular as the two edges approach each other, and also introduces matrix asymmetries that must be approximated. In contrast, differentiating our energy yields a well-behaved force function. We will show in §5.1.4 that this allows us to take timesteps that are orders of magnitude larger than DER.

3.3.3 Physical Interpretation. Our core assumption that the initial configuration is *non-straight* ensures that torsion becomes visible to the vertex positions, because it guarantees that the initial τ_0 is always well-defined. Since we are simulating thin strands of hair, we can assume only one end of the strand is fixed, and that the relative cross-sectional area is small. We can justify discarding the material frames because as cross-sectional inertia approaches zero, so does the relative twist angle (see §5 in [Bergou et al. 2008]). Twist becomes almost entirely characterized by torsion, so our energy captures the predominant force. These assumptions might complicate the simulation of other phenomena such as plectonemes or torque transducers, but this is acceptable since our focus is on hair. We will see in §5.1 that these assumptions considerably improve numerical robustness.

Outside of graphics, torsion has been defined as the angle between the normals of two consecutive triangles [Banchoff 1982]. However, this measure is not orthogonal to the bending direction, and generates overlapping forces similar to the parasitic phenomena observed in spring-mass approximations. Our formulation instead produces torsion forces that are, by construction, orthogonal to bending.

4 EIGENANALYSIS OF LIFTED CURLS

Large, efficient timesteps require the matrix of force gradients to be positive semi-definite. One well-known method for ensuring this is to filter the per-element eigenvalues [Teran et al. 2005], but if this is done numerically, its running time can rival that of the PCG solve. See §5 in our supplemental document for detailed timings.

We instead follow the analytic approach of Smith et al. [2019] and Lin et al. [2022]. We show that it is possible to construct the analytic eigendecomposition of each term in our energy and efficiently filter the energy Hessians. Our generic analyses can also be applied to other energies.

4.1 Stretching Eigenanalysis

We start with the stretching energy (Eqn. 2) by applying a set of transformations that reveal its eigenstructure. Our analysis will show that a simple strategy in fact corresponds to the exact analytic filter: *under compression, fall back to Gauss-Newton.*

4.1.1 Stretching Eigensystem. The force and force gradients along an edge are computed as

$$\mathbf{f} = -l\frac{\partial \Psi_s}{\partial \mathbf{x}} \qquad\qquad \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = -l\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2}, \tag{11}$$

where $\mathbf{f}$ is the force along the edge, $\mathbf{x} \in \Re^6$ is a vector of the two vertex positions, and l is the length of the original edge. We want the analytic eigensystem of $\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2} \in \Re^{6\times 6}$.

However, $\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2}$ contains a rank-3 nullspace that corresponds to rigid translation. In order to isolate the non-null components, we rewrite the Hessian terms of the deformation gradient $\mathbf{d}$ (Eqn. 3):

$$\frac{\partial^2 \Psi_s}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{d}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} \frac{\partial \mathbf{d}}{\partial \mathbf{x}} \tag{12}$$

where $\frac{\partial \mathbf{d}}{\partial \mathbf{x}} = \begin{bmatrix} -1 & 1 \end{bmatrix} \otimes \mathbf{I}_{3\times 3}$, the $\otimes$ denotes a Kronecker product, and $\mathbf{I}_{3\times 3}$ is an identity matrix. Without loss of generality, we can now focus on obtaining the analytic eigensystem of $\frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} \in \Re^{3\times 3}$.

We next expand in terms of the stretch invariant I_s from §3.1

$$\frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} = \frac{\partial^2 \Psi_s}{\partial I_s^2} \frac{\partial I_s}{\partial \mathbf{d}} \frac{\partial I_s}{\partial \mathbf{d}}^T + \frac{\partial \Psi_s}{\partial I_s} \frac{\partial^2 I_s}{\partial \mathbf{d}^2} \tag{13}$$

and apply the identities $\frac{\partial I_s}{\partial \mathbf{d}} = \frac{\mathbf{d}}{I_s}$ and

$$\frac{\partial^2 I_s}{\partial \mathbf{d}^2} = \frac{1}{I_s} \frac{\partial \mathbf{d}}{\partial \mathbf{d}} + \mathbf{d} \frac{\partial I_s^{-1}}{\partial \mathbf{d}} = \frac{1}{I_s} \left(\mathbf{I}_{3\times 3} - \frac{1}{I_s^2} \mathbf{d}\mathbf{d}^\top \right), \tag{14}$$

to arrive at the final generic expression:

$$\frac{\partial^2 \Psi_s}{\partial \mathbf{d}^2} = \left(\frac{\partial^2 \Psi_s}{\partial I_s^2} - \frac{1}{I_s} \frac{\partial \Psi_s}{\partial I_s} \right) \left(\frac{\mathbf{d}}{I_s} \right) \left(\frac{\mathbf{d}}{I_s} \right)^\top + \frac{1}{I} \frac{\partial \Psi_s}{\partial I_s} \mathbf{I}_{3\times 3}. \tag{15}$$

Close examination of this equation reveals the underlying eigensystem. The blue term is a scaled identity, so its eigenvectors are arbitrary, and all three eigenvalues are $\frac{1}{I} \frac{\partial \Psi_s}{\partial I_s}$. The red term is an outer product, so its eigenvector is $\mathbf{d}/I_s$, which is a normalized version of $\mathbf{d}$. The red coefficient subtracts the $\frac{1}{I} \frac{\partial \Psi_s}{\partial I_s}$ eigenvalue in the $\mathbf{d}/I_s$ direction away from the blue term, and replaces it with $\frac{\partial^2 \Psi_s}{\partial I_s^2}$. Therefore, the generic eigenpairs $(\lambda_i, \mathbf{q}_i)$ must be:

$$\lambda_0 = \frac{\partial^2 \Psi_s}{\partial I_s^2} \qquad \mathbf{q}_0 = \frac{\mathbf{d}}{I_s} \tag{16}$$

$$\lambda_1 = \frac{1}{I_s} \frac{\partial \Psi_s}{\partial I_s} \qquad \mathbf{q}_1 = \mathbf{d}_\perp \tag{17}$$

$$\lambda_2 = \frac{1}{I_s} \frac{\partial \Psi_s}{\partial I_s} \qquad \mathbf{q}_2 = \mathbf{d}_\times. \tag{18}$$

The last two eigenpairs span an arbitrary subspace, so we use $\mathbf{d}_\perp$ and $\mathbf{d}_\times$ to denote two vectors orthogonal to $\mathbf{q}_0 = \mathbf{d}/I_s$ (See Fig. 3).

Similar to Smith et al. [2019], our generic analysis applies to *any* energy written in terms of I_s. The eigenvectors remain the same regardless of the energy, and only the eigenvalue expressions change. For our specific Ψ_s (Eqn. 2), the analytic eigenvalues are:

$$\lambda_0 = \mu_s \qquad \lambda_{1,2} = \mu_s \left(1 - \frac{1}{I_s} \right). \tag{19}$$

The first eigenvalue is always positive, and the second two immediately suggest a filtering strategy.

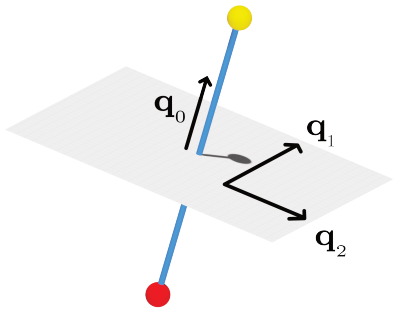


Fig. 3. Eigenvectors of strand stretching energies. The $\mathbf{q}_0$ vector is unique and its eigenvalue is always positive-definite. Non-unique eigenvectors $\mathbf{q}_1$ and $\mathbf{q}_2$ span the orthogonal plane and become indefinite under compression. *All* strand stretching energies conform to this eigenstructure.

4.1.2 Filtering Strategy and Discussion. The $\lambda_{1,2}$ eigenvalues only go negative under edge compression ($I_s < 1$), which corresponds to bifurcation under buckling. In this case, filtering the negative eigenvalues is equivalent to discarding the blue term from Eqn. 15, as well as $\frac{1}{I_s}\frac{\partial \Psi}{\partial I_s}$ from the red term.[1] All that remains is the outer product, $\frac{\partial^2 \Psi}{\partial \mathbf{d}^2} \approx \frac{\partial^2 \Psi}{\partial I_s^2}\left(\frac{\mathbf{d}}{I_s}\right)\left(\frac{\mathbf{d}}{I_s}\right)^\top$, which corresponds to the commonly used Gauss-Newton approximation [Choi and Ko 2002; Zehnder et al. 2021] for the Hessian. Our analysis shows that this filtering strategy is in fact exact: *for compressed edges, just compute the outer product.*

Our analysis generalizes to higher-order energies, such as St. Venant-Kirchhoff-like stretching

$$\Psi_4 = \mu_s (I_s - 1)^4, \tag{20}$$

where the eigenvalues become:

$$\lambda_0 = 3\mu_s (I_s - 1)^2 \qquad \lambda_{1,2} = \mu_s \frac{(I_s - 1)^3}{I_s}. \tag{21}$$

The indefiniteness condition ($I_s < 1$) remains the same, so again the Gauss-Newton strategy is an exact filter.

4.2 Bending Eigenanalysis

We next perform an eigenanalysis of our bending energy, Eqn. 4. As with stretching, we will first project off irrelevant nullspaces, but we will then see that the main challenge is analyzing the Hessian of θ, the angle between the two strand edges.

4.2.1 Cloth-like Formulation. We start by observing that since the bending energy involves three vertices, it can be thought of as a triangle of cloth. We can then rewrite Eqn. 4 into a more cloth-like form by arranging its two edges into a matrix:

$$\mathbf{E} = \left[\begin{array}{c|c} \mathbf{e}_0 & \mathbf{e}_1 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{x}_0 - \mathbf{x}_1 & \mathbf{x}_2 - \mathbf{x}_1 \end{array}\right] \in \mathfrak{R}^{3\times 2}. \tag{22}$$

This resembles the deformation gradient for cloth [Kim 2020], but no material space pullback is needed, as it is already encoded in θ_0. By introducing the standard basis directions $\mathbf{u} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and

[1] While we are filtering the *Hessian* to maintain robustness, buckling is still present in the *forces* of the overall system.

$\mathbf{v} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, the angle formula can be rewritten as:

$$\theta = \operatorname{acos}\left(\frac{\mathbf{u}^\top \mathbf{E}^\top \mathbf{E}\mathbf{v}}{\|\mathbf{E}\mathbf{u}\| \cdot \|\mathbf{E}\mathbf{v}\|}\right) \tag{23}$$

We can *vectorize* $\mathbf{E}$ using the vec $(\cdot)$ operator [Golub and Van Loan 2013; Kim and Eberle 2022], and adopt the convention of a vectorized matrix becoming the lowercase version of the original, e.g.

$$\mathbf{e} = \operatorname{vec}(\mathbf{E}) = \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \end{bmatrix}. \tag{24}$$

Similar to Eqn. 12, we then use $\mathbf{e} = \operatorname{vec}(\mathbf{E})$ to write

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} \frac{\partial \mathbf{e}}{\partial \mathbf{x}} \in \mathfrak{R}^{9\times 9}. \tag{25}$$

We can now directly analyze $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} \in \mathfrak{R}^{6\times 6}$ while ignoring the rigid translation nullspace. Similar to Eqn. 13, we rewrite $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2}$ in terms of a scalar invariant, but this time use θ in lieu of I_s:

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} = \frac{\partial^2 \Psi_b}{\partial \theta^2} \frac{\partial \theta}{\partial \mathbf{e}} \frac{\partial \theta}{\partial \mathbf{e}}^\top + \frac{\partial \Psi_b}{\partial \theta} \frac{\partial^2 \theta}{\partial \mathbf{e}^2}. \tag{26}$$

The red term is an outer product of $\frac{\partial \theta}{\partial \mathbf{e}}$, so its eigensystem is known. The main challenge is determining the eigensystem of $\frac{\partial^2 \theta}{\partial \mathbf{e}^2}$.

4.2.2 Eigensystem of θ. The analysis becomes simpler when the edges are normalized[2], so we form the *tangent* matrix

$$\mathbf{T} = \left[\begin{array}{c|c} \frac{\mathbf{e}_0}{\|\mathbf{e}_0\|} & \frac{\mathbf{e}_1}{\|\mathbf{e}_1\|} \end{array}\right] = \left[\begin{array}{c|c} \mathbf{t}_0 & \mathbf{t}_1 \end{array}\right] \tag{27}$$

and denote its vectorization $\mathbf{t} = \operatorname{vec}(\mathbf{T})$. Using the normalization

$$\mathbf{N} = \begin{bmatrix} \frac{1}{\|\mathbf{e}_0\|} & 0 \\ 0 & \frac{1}{\|\mathbf{e}_1\|} \end{bmatrix} \otimes \mathbf{I}_{3\times 3}, \tag{28}$$

we can then write:

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2} = \mathbf{N}^\top \left(\frac{\partial^2 \Psi_b}{\partial \theta^2} \frac{\partial \theta}{\partial \mathbf{t}} \frac{\partial \theta}{\partial \mathbf{t}}^\top + \frac{\partial \Psi_b}{\partial \theta} \frac{\partial^2 \theta}{\partial \mathbf{t}^2}\right) \mathbf{N}. \tag{29}$$

The challenge is now to find the eigensystem of $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$.

Fortunately, we have now transformed the Hessian into a form that has a *very* simple structure. It can be written in terms of the tangents $\mathbf{t}_0$ and $\mathbf{t}_1$, the binormal $\mathbf{b} = \mathbf{t}_0 \times \mathbf{t}_1$, and their cross products $\mathbf{b}_0 = \mathbf{t}_0 \times \frac{\mathbf{b}}{\|\mathbf{b}\|}$ and $\mathbf{b}_1 = \mathbf{t}_1 \times \frac{\mathbf{b}}{\|\mathbf{b}\|}$. The complete analytic eigenpairs $(\lambda_i^\theta, \mathbf{q}_i^\theta)$ of $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$ are:

$$\lambda_0^\theta = \frac{\cos\theta - 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_0^\theta = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ \mathbf{b} \end{bmatrix} \tag{30}$$

$$\lambda_1^\theta = \frac{\cos\theta + 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_1^\theta = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \tag{31}$$

$$\lambda_2^\theta = -1 \qquad \mathbf{q}_2^\theta = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{t}_0 + \mathbf{b}_0 \\ \mathbf{0}_3 \end{bmatrix} \tag{32}$$

[2]The general case of non-normalized edges introduces complexities that are extensively analyzed in Wu and Kim [2023].

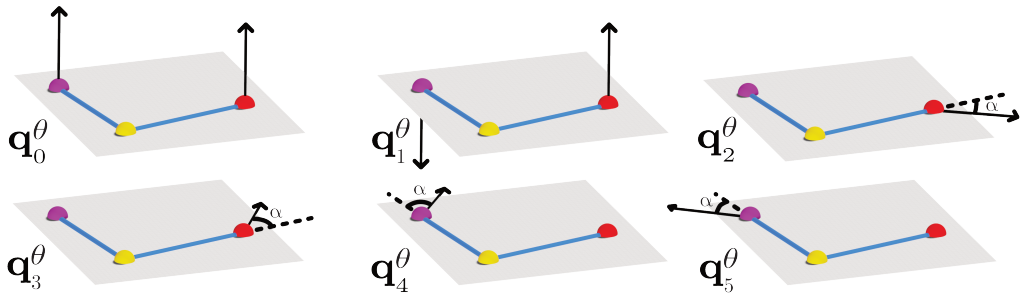


Fig. 4. The six eigenvectors of $\frac{\partial^2\theta}{\partial \mathbf{t}^2}$. The $\mathbf{q}_{0,1}^\theta$ vectors capture out-of-plane motions, while $\mathbf{q}_{2,3}^\theta$ represent in-plane stretches along the $\mathbf{t}_0$ yellow-red edge. Each is an in-plane $\alpha = \frac{\pi}{4}$ rotation of that edge. The $\mathbf{q}_{4,5}^\theta$ vectors are analogous rotations of the $\mathbf{t}_1$ yellow-purple edge.

$$\lambda_3^\theta = 1 \qquad \mathbf{q}_3^\theta = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{t}_0 - \mathbf{b}_0 \\ \mathbf{0}_3\end{bmatrix} \tag{33}$$

$$\lambda_4^\theta = -1 \qquad \mathbf{q}_4^\theta = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}_3 \\ \mathbf{t}_1 + \mathbf{b}_1\end{bmatrix} \tag{34}$$

$$\lambda_5^\theta = 1 \qquad \mathbf{q}_5^\theta = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{0}_3 \\ \mathbf{t}_1 - \mathbf{b}_1\end{bmatrix} \tag{35}$$

where $\mathbf{0}_3 \in \Re^3$ denotes a 3-vector of zeros. A Matlab/Octave script that verifies these expressions is provided in the supplemental materials. These eigenvectors can be interpreted relative to the plane spanned by $\mathbf{t}_0$ and $\mathbf{t}_1$ (see Fig. 4). The $\mathbf{q}_{0,1}^\theta$ vectors span the directions *orthogonal* to this plane, the $\mathbf{q}_{2,3}^\theta$ vectors are $\pm\frac{\pi}{4}$ rotations of $\mathbf{t}_0$ within that plane which form a local planar basis about $\mathbf{t}_0$, and $\mathbf{q}_{4,5}^\theta$ form an analogous planar basis about $\mathbf{t}_1$.

4.2.3 Bending Eigensystem. With the eigensystem of $\frac{\partial^2\theta}{\partial \mathbf{t}^2}$ in hand, we must now determine the eigensystem of

$$\frac{\partial^2\Psi_b}{\partial \mathbf{t}^2} = \frac{\partial^2\Psi_b}{\partial\theta^2}\frac{\partial\theta}{\partial \mathbf{t}}\frac{\partial\theta}{\partial \mathbf{t}}^\top + \frac{\partial\Psi_b}{\partial\theta}\frac{\partial^2\theta}{\partial \mathbf{t}^2}. \tag{36}$$

Then we can compute $\frac{\partial^2\Psi_b}{\partial \mathbf{e}^2} = \mathbf{N}^\top\frac{\partial^2\Psi_b}{\partial \mathbf{t}^2}\mathbf{N}$ using Eqn. 29. Since $\mathbf{N}$ is a diagonal matrix of positive scalars, filtering the eigenvalues of $\frac{\partial^2\Psi_b}{\partial \mathbf{t}^2}$ is sufficient to guarantee that $\frac{\partial^2\Psi_b}{\partial \mathbf{e}^2}$ is positive-semidefinite (PSD). Applying Eqn. 25 then ensures that $\frac{\partial^2\Psi_b}{\partial \mathbf{x}^2}$ is PSD.

Eqn. 36 is a rank-one update to $\frac{\partial^2\theta}{\partial \mathbf{t}^2}$, so we can use the approach of Kim [2020] and apply the Bunch-Nielsen-Sorensen (BNS) formulae [Bunch et al. 1978] to compute how the $\frac{\partial\theta}{\partial \mathbf{t}}\frac{\partial\theta}{\partial \mathbf{t}}^\top$ update shifts the eigensystem of $\frac{\partial^2\theta}{\partial \mathbf{t}^2}$. While BNS is a numerical approach, our transformed system is now sufficiently simple that it can be applied analytically.

Applying BNS to an $\Re^{6\times6}$ matrix generally involves solving a 6th order polynomial, but we make two key observations. First, the update vector is orthogonal to $\mathbf{q}_0^\theta$ and $\mathbf{q}_1^\theta$. Second, the remaining eigenvalues $\lambda_{2\ldots5}^\theta$ are all ±1. Together, these *reduce the 6th order polynomial to a quadratic*, and allow the rank-one updated eigensystem to be written down in closed form. The complete derivation is

in the supplemental materials, and the final eigensystem is

$$\lambda_0 = \frac{\partial \Psi}{\partial \theta} \frac{\cos\theta - 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_0 = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ \mathbf{b} \end{bmatrix} \tag{37}$$

$$\lambda_1 = \frac{\partial \Psi}{\partial \theta} \frac{\cos\theta + 1}{\|\mathbf{b}\|} \qquad \mathbf{q}_1 = \frac{1}{\sqrt{2}\|\mathbf{b}\|} \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \tag{38}$$

$$\lambda_2 = \frac{\partial^2 \Psi}{\partial \theta^2} + \sqrt{\frac{\partial^2 \Psi}{\partial \theta^2}^2 + \frac{\partial \Psi}{\partial \theta}^2} \qquad \mathbf{q}_2 = \begin{bmatrix} \frac{\partial \Psi}{\partial \theta}\mathbf{t}_0 - \lambda_2 \mathbf{b}_0 \\ \frac{\partial \Psi}{\partial \theta}\mathbf{t}_1 + \lambda_2 \mathbf{b}_1 \end{bmatrix} \tag{39}$$

$$\lambda_3 = \frac{\partial^2 \Psi}{\partial \theta^2} - \sqrt{\frac{\partial^2 \Psi}{\partial \theta^2}^2 + \frac{\partial \Psi}{\partial \theta}^2} \qquad \mathbf{q}_3 = \begin{bmatrix} \frac{\partial \Psi}{\partial \theta}\mathbf{t}_0 - \lambda_3 \mathbf{b}_0 \\ \frac{\partial \Psi}{\partial \theta}\mathbf{t}_1 + \lambda_3 \mathbf{b}_1 \end{bmatrix} \tag{40}$$

$$\lambda_4 = \frac{\partial \Psi}{\partial \theta} \qquad \mathbf{q}_4 = \begin{bmatrix} \mathbf{t}_0 + \mathbf{b}_0 \\ -\mathbf{t}_1 + \mathbf{b}_1 \end{bmatrix} \tag{41}$$

$$\lambda_5 = -\frac{\partial \Psi}{\partial \theta} \qquad \mathbf{q}_5 = \begin{bmatrix} \mathbf{t}_0 - \mathbf{b}_0 \\ -\mathbf{t}_1 - \mathbf{b}_1 \end{bmatrix} \tag{42}$$

where the last four eigenvectors are written in unnormalized form. As in the stretching case, the specific expressions for the eigenvalues change, but the form of the eigenvectors remains the same, regardless of the energy. The first two again correspond to out-of-plane motion, the middle two are now in-plane oscillations, and the last two parameterize in-plane translations. The specific eigenvalues for our bending energy Ψ_b (Eqn. 4) are:

$$\lambda_0 = \mu_b \left(\theta - \theta_0\right) \frac{\cos\theta - 1}{\|\mathbf{b}\|} \qquad \lambda_1 = \mu_b \left(\theta - \theta_0\right) \frac{\cos\theta + 1}{\|\mathbf{b}\|} \tag{43}$$

$$\lambda_2 = \mu_b \left(1 + \sqrt{1 + (\theta - \theta_0)^2}\right) \qquad \lambda_3 = \mu_b \left(1 - \sqrt{1 + (\theta - \theta_0)^2}\right) \tag{44}$$

$$\lambda_4 = \mu_b(\theta - \theta_0) \qquad \lambda_5 = -\mu_b(\theta - \theta_0). \tag{45}$$

As with $\frac{\partial^2 \theta}{\partial \mathbf{t}^2}$, Matlab/Octave scripts are provided in the supplemental materials that verify the correctness of these expressions.

4.3 Torsion Eigenanalysis

The torsion energy deals fundamentally with the angle between two vectors, so we can reuse components of our bending eigenanalysis. We start with the vectorization $\mathbf{w} = \mathrm{vec}\,(\mathbf{W})$ of Eqn. 8, and construct a torsion Hessian analogous to Eqns. 12 and 25 $\frac{\partial^2 \Psi_t}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} + \left(\frac{\partial \Psi_t}{\partial \mathbf{w}}\right)^\top : \frac{\partial^2 \mathbf{w}}{\partial \mathbf{x}^2}$, but the double-contraction is equivalent to $-\left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta$, which we describe further in the supplement. This then yields:

$$\frac{\partial^2 \Psi_t}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} - \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta. \tag{46}$$

Similar to bending, we can now directly examine $\frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2}$, where rank-3 translation nullspace has been removed. A slight wrinkle appears because $\frac{\partial \mathbf{w}}{\partial \mathbf{x}}$ seemingly contains new, non-trivial entries, but we show in the supplement that force contributions result to zero while the Hessian gains an easy-to-handle second term. Without loss of generality, we can use the expression:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{x}} = \begin{bmatrix} 1 & \mathbf{e}_0^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 - 1 & -\mathbf{e}_0^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 & 0 \\ 0 & \mathbf{e}_2^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 & -\mathbf{e}_2^\top \mathbf{e}_1 / \|\mathbf{e}_1\|^2 - 1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix} \otimes \mathbf{I}_3. \tag{47}$$

While $\frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2}$ appears to be rank-9, the third column of $\mathbf{W}$ only participates in the $\mathcal{S}(\det \mathbf{W})$ term of τ. This becomes a constant under differentiation, except for a Dirac delta at zero which we can be filtered [Kim et al. 2019]. Thus, we can deflate away another rank-3 subspace to obtain a torsion-specific edge matrix:

$$\mathbf{E}_t = \mathbf{W} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{48}$$

Using the vectorization $\mathbf{e}_t = \mathrm{vec}(\mathbf{E}_t)$, the first term in the Hessian now becomes:

$$\frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}} = \frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial \mathbf{e}_t}{\partial \mathbf{w}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{e}_t^2} \frac{\partial \mathbf{e}_t}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \mathbf{x}}. \tag{49}$$

The edge matrix $\mathbf{E}_t$ is structurally identical to the bending edge matrix $\mathbf{E}$ (Eqn. 22). All our prior analysis of $\frac{\partial^2 \Psi_b}{\partial \mathbf{e}^2}$ (Eqn. 26) can be directly reapplied. The tangent-space eigenvalues for Ψ_t are then:

$$\lambda_0 = \mu_t (\tau - \tau_0) \frac{\cos \tau - 1}{\|\mathbf{b}\|} \qquad \lambda_1 = \mu_t (\tau - \tau_0) \frac{\cos \tau + 1}{\|\mathbf{b}\|} \tag{50}$$

$$\lambda_2 = \mu_t \left(1 + \sqrt{1 + (\tau - \tau_0)^2}\right) \qquad \lambda_3 = \mu_t \left(1 - \sqrt{1 + (\tau - \tau_0)^2}\right) \tag{51}$$

$$\lambda_4 = \mu_t (\tau - \tau_0) \qquad \lambda_5 = -\mu_t (\tau - \tau_0). \tag{52}$$

The full $\frac{\partial^2 \Psi_t}{\partial \mathbf{e}_t^2}$ is reconstructed using Eqns. 28 and 29, and $\frac{\partial \mathbf{w}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \frac{\partial \mathbf{w}}{\partial \mathbf{x}}$ obtained with Eqn. 49. PSD-ness is preserved throughout because both transformations left- and right-multiply using the same rectangular matrix. Even if a matrix contains negative singular values, they self-cancel during the second multiply. Thus, the same filtering process can be applied to $-\left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta^\top \frac{\partial^2 \Psi_t}{\partial \mathbf{w}^2} \left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta$ from Eqn. 46, albeit with the *positive* eigenvalues filtered, due to the leading negative. Further details on $\left(\frac{\partial \mathbf{w}}{\partial \mathbf{x}}\right)_\Delta$ are provided in the supplement.

The eigenanalysis of our energy is now complete.

5 IMPLEMENTATION AND RESULTS

For linear system assembly and solution, we used Eigen [Guennebaud et al. 2010], but wrote custom routines to accelerate sparse matrix assembly. We used Eigen's conjugate gradient solver, but similar to previous works [Daviet 2020; Fei et al. 2017; Lesser et al. 2022] added a per-strand block-diagonal preconditioner. We found a full Cholesky decomposition of each strand block to be highly effective, as the small block sizes ($\sim 300 \times 300$) allowed factorizations and back-solves to parallelize cleanly. This accelerated solves by between 6× and 1130×, bringing solver times to parity with system assembly and collision processing (Table 3). Detailed timing breakdowns are §5 of our supplementary document. All linear system solves were run to $\epsilon = 1 \times 10^{-8}$ on a 4-core, 4.6 GHz Intel Xeon W-2225 with 128 GB memory.

We used position-based BDF-1 with 3 Newton iterations for time integration. Kinematic collisions were handled with constraint projection [Baraff and Witkin 1998; Tamstorf et al. 2015], and all other collisions were resolved using penalty forces. Material parameters were set using Young's modulus (E) and Poisson's ratio (ν). The strand constants were set as:

$$\mu_s = E\pi r^2 \qquad \mu_b = \frac{E}{4}\pi r^2 \qquad \mu_t = \frac{E}{4(1+\nu)}\pi r^4. \tag{53}$$

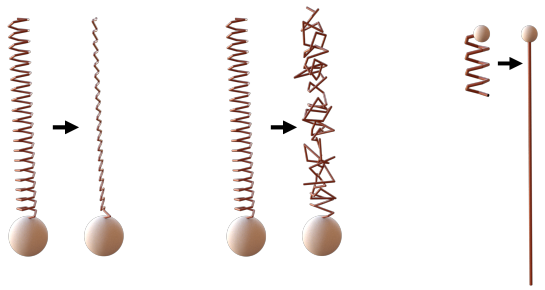
Table 1. We compress a coil (Fig. 5a) and simulate at $\Delta t = 1/30$ using Discrete Elastic Rods (DER) and Lifted Curls (LC). Multiple Hessian filtering strategies were applied to DER, including no filtering, Gauss-Newton for bend/twist/stretch, Gauss-Newton for bend/twist but full filtering for stretch, and full filtering for bend/twist/stretch. Past 40% compression, all DER filtering strategies failed. Our LC energy always recovers, even at 99.999% compression.

Curl Compression	Discrete Elastic Rods (DER)				Lifted Curls (Ours)
	Unfiltered	Gauss-Newton (GN)	GN + Stretch Filter	Filtered	
30%	unstable	unstable	stable	stable	stable
40%	unstable	unstable	stable	unstable	stable
60%	unstable	unstable	unstable	unstable	stable
99.999%	unstable	unstable	unstable	unstable	stable

5.1 Results

5.1.1 Kinematic Tests. We ran kinematic displacement tests comparing our Lifted Curls (LC) energy to the implicit implementation of Discrete Elastic Rods (DER) [Bergou et al. 2010] provided by the original authors (Bergou) [Jawed et al. 2018]. Unless stated otherwise, the tests were performed on a coil with 100 vertices, spaced 6 mm apart, with 1 mm radius strands forming curls of 5 mm radius. Collisions and gravity are deactivated. All simulations used $\Delta t = 1/30$ s, 3 Newton iterations, $E = 3.9e9$ g/cm s^2, $\nu = 0.48$ and $\rho = 1.3$ g/cm^3.

To facilitate comparison, we tested multiple Hessian filtering strategies for DER. These included *Unfiltered* (the original code), *Gauss-Newton* where only the outer product terms were used, *GN + Stretch Filter* where bending and twisting were approximated using their outer product terms but the stretching term was fully filtered, and *Filtered* where all Hessian terms were filtered.



(a) Compression Test (b) Jitter Test (c) Straighten Test

Fig. 5. Kinematic displacement tests comparing the stability of our model and Discrete Elastic Rods.

5.1.2 Compression Test. In Table 1 and Fig. 5a, we compressed a coil by 30% along the x-axis (inset, left). LC quickly recovered, as well as the *GN + Stretch Filter* and *Filtered* versions of DER (see Table 1 and video). Under 40% compression, LC recovered while only the *GN + Stretch Filter* strategy succeeded. Beginning at 60% compression, all DER strategies failed. For this case, the *GN + Stretch Filter* strategy succeeded if timestep was set to $\Delta t = 1/3{,}000$, a **100**× difference in stability. The *Filtered* strategy succeeded at $\Delta t = 1/300{,}000$, a **10,000**× difference in stability. We ran compression tests up to 99.999%, and our LC energy always recovered with $\Delta t = 1/30$.

We found that DER implementations commonly set the maximum number of Newton iterations to anywhere from 50 [Jawed et al. 2018] to 10,000 [Fei et al. 2019]. If we view Newton iterations as

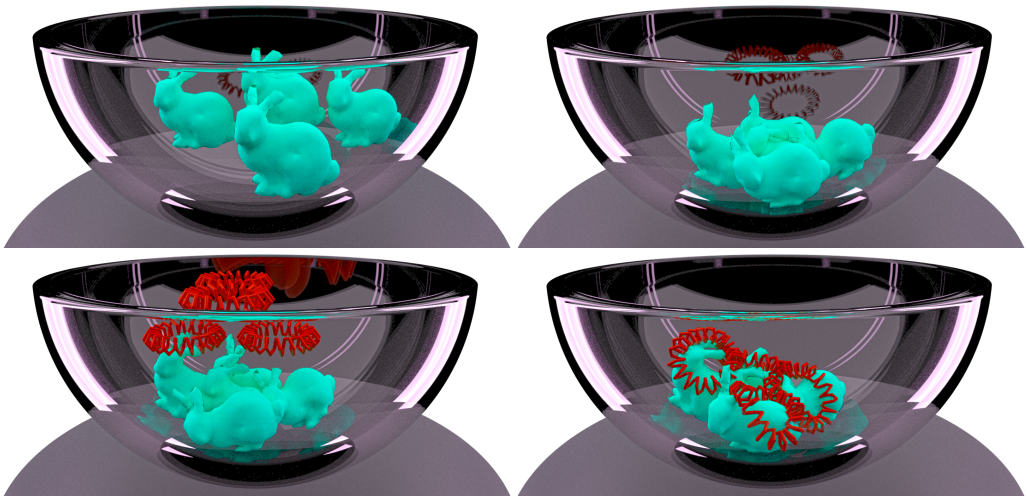


Fig. 6. Our strand model integrates cleanly with existing simulators. We drop five volumetric bunnies into a bowl, followed by four spiral hair ties. The hair ties are simulated using our Lifted Curls energy, and entangle realistically with the bunny ears.

approximate substeps in time, the numbers roughly align with the stability behavior we observed. However, these Newton iteration counts are outside our acceptable performance range.

We hypothesize that *GN + Stretch Filter* yielded slightly better stability because most of the stiffness lives in the stretching term, so it must be carefully filtered. In contrast, if bending and twisting are far from equilibrium, the outer product yields a more conservative descent direction, so convergence becomes slower but more stable.

5.1.3 Jitter Test. In Table 2 and Fig. 5b, we kinematically jittered each vertex of the coil by varying amplitudes. When each vertex was jittered by 1 mm, both LC and DER quickly recovered (see video). For larger jitters (2 mm) all DER filtering strategies diverged. We tested up to a jitter of 5 mm, where LC quickly recovered. The *GN + Stretch Filter* DER strategy again only stabilized after setting $\Delta t = 1/3{,}000$, while the *Filtered* strategy performed slightly worse than in previous tests and needed a timestep of $\Delta t = 1/3{,}000{,}000$.

5.1.4 Straightening Test. To show that our energy only requires the *rest state* to be non-planar and handles straight *deformed states* robustly, we kinematically straightened a coil of 20 vertices (Fig. 5c). Gravity was enabled for this example. LC recovered under $\Delta t = 1/30$, while all DER filtering strategies failed (see video). Consistent with prior tests, the *GN + Stretch Filter* strategy only succeeded at $\Delta t = 1/3{,}000$, while *Filtered* succeeded at $\Delta t = 1/300{,}000$.

5.1.5 Strands, Shells, and Volumes. Past systems have unified volumes, shells, and strands [Chang et al. 2019; Martin et al. 2010], but we found integrating our LC energy into an existing FEM solver was straightforward. If linear FEM forces are already represented as couplings between the vertices of simplicies, our approach simply maps additional strain energies onto existing infrastructure. Fig. 6 shows a preliminary integration of our strand model with an existing volume solver. The simulator already handled edge-edge contacts, so strand collisions just become another instance of this contact case.

Table 2. We jitter the vertices of a coil (Fig. 5b) and simulate at $\Delta t = 1/30$ using Discrete Elastic Rods (DER) and Lifted Curls (LC). Multiple Hessian filtering strategies were applied to DER (see Table 1 and §5.1.2). When jitter amplitudes exceed 1 mm, all DER filtering strategies fail. Our LC energy recovers, even more for much larger 5 mm jitters.

Jitter Amplitude	Discrete Elastic Rods (DER)				Lifted Curls (Ours)
	Unfiltered	Gauss-Newton (GN)	GN + Stretch Filter	Filtered	
0.5 mm	unstable	unstable	stable	stable	stable
1 mm	unstable	unstable	stable	unstable	stable
2 mm	unstable	unstable	unstable	unstable	stable
5 mm	unstable	unstable	unstable	unstable	stable

5.1.6 Hairball Test. Similar to previous works, we place hairs along the top of a 15 cm radius sphere. Previous works modelled straight or wavy hairs with thirty [Daviet 2020; Kaufman et al. 2014], fifty [Michels et al. 2015], or sixty [Han et al. 2019] segments, but we found that at least 100 vertices are needed to adequately sample tight curls. We generated hairs using the parameters from §5.1.1, consistent with the measurements from De La Mettrie et al. [2007].

We adopted the *wisp* (a.k.a. clump) approximation common in film production [Butts et al. 2018; Thyng et al. 2017], where each simulated strand is a proxy for multiple hairs [Bertails et al. 2003; Choe et al. 2005; Ward and Lin 2003]. Tightly coiled hair commonly forms such wisps (see inset photo), and we linearly decreased the wisp radius approaching the tip to maintain consistency with real-world references. We uniformly distributed 200K hairs between the wisps, recomputed mass and μ_* in Eqn. 53 at each vertex according to the wisp radius, and set $\Delta t = 1/30$.

DER exploded with this Δt, so we only show LC results in Fig. 1 and 7. We simulated $E = 6e^8$ g/cm s^2 for 2000 wisps to account for voids between hairs, making it softer than the $E = 1e^9 - 1e^{11}$ g/cm s^2 range of single-strand stiffnesses from the literature [Bertails et al. 2005; Daviet 2020]. Similarly, we reduced density by 97% to $\rho = 0.033$ g/cm^3. Based on sample measurements, the 2000 wisp simulations used a base wisp radius of 3.54 mm, and tip radius of 0.71 mm. For the 4000 and 8000 wisp simulations, we proportionally reduced the radii while progressively doubling E, as hairs cluster along the centerline. This yields $E = 4e^{10}$ g/cm s^2 for 128K wisps (i.e. the resolution of individual hairs), which is within previously reported ranges.

The appearance of tightly coiled hair is qualitatively different from the straight of wavy hair that usually appears in computer graphics, so we provide photo references in Fig. 8. We compare the “clumped” and ”picked out” looks from our simulations against the tightly coiled hair from the photos of Leal Alexander [2023]. For the clumped look, our results qualitatively match the cylindrical curls in the photo. The diffuse appearance of the picked out look is also reflected in our results. Both photos feature hair from the the same model, so the visual differences arise from coil coherence.

5.2 Limitations and Future Work

We have presented a robust model for tightly coiled hair. Our energy is consistently quadratic in its primary variable, making it analogous to ARAP [Sorkine and Alexa 2007] or co-rotational [Etzmuß et al. 2003] models for shells and volumes. Similar to those models, ours yields a linearly compliant look, so increasing non-linearity while retaining robustness is future work.

Fig.	Wisps	DoFs	Collision Handling	System Assembly	PCG	Total Time
1,7	2000	806K	35.4%	32.3%	32.4%	17.7s
7	4000	1.61M	35.9%	33.6%	30.5%	27.6s
1,7	8000	3.22M	34.9%	33.9%	31.2%	57.0s

Table 3. Timings for $\Delta t = 1/30$ timesteps of BDF-1 with 3 Newton iterations. Collision Handling includes detection and response, and System Assembly is time spent in matrix assembly, minus collisions.

Our energy assumes isotopic, non-straight hair, so an anisotropic, quadratic model for straight hair is still future work. While our model was designed for curly hair, we have found that it is also stable for straight hair ($\theta_0 \approx \pi - 1e^{-8}$ and $\tau_0 \approx 1e^{-8}$). Consistent with other works [Sánchez-Banderas et al. 2020], the main source of instability is when edges become very short and the stretching energy becomes ill-conditioned.

The scaling behavior of tightly coiled strands under continuous collision detection (CCD) [Kaufman et al. 2014] and frictional contact [Daviet 2020] is also worth further investigation. In lieu of our wisp approach, Eulerian-on-Lagrangian methods [Sueda et al. 2011] could instead be used to resolve highly correlated curls. Grooming tightly coiled hair to send to our simulator also remains challenging [Ogunseitan 2022; Patrick et al. 2004]. Finally, exploring the "looks" of hair that are possible, beyond the "clumped" and "frizzy" looks in Fig. 1, is an interesting direction for future investigation.

ACKNOWLEDGMENTS

We thank the reviewers for their comments. This work was supported by Adobe, the Bungie Foundation, Teng and Han Family Fund, and National Science Foundation (IIS-2132280). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Leal Alexander. 2023. *Instagram.* https://www.instagram.com/curlygallal/

Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. 1992. A simple method for extracting the natural beauty of hair. In *Proc. of SIGGRAPH*. 111–120.

TF Banchoff. 1982. Global geometry of polygons. III. Frenet frames and theorems of Jacobi and Milnor for space polygons. *Rad Jugoslav. Akad. Znan. Umjet* 396 (1982), 101–108.

David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proc. of SIGGRAPH*. 43–54.

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Trans. Graph.* 29, 4 (2010), 1–10.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. *ACM Trans. Graph.* 27, 3 (2008), 1–12.

F. Bertails. 2009. Linear time super-helices. In *Computer graphics forum*, Vol. 28. Wiley Online Library, 417–426.

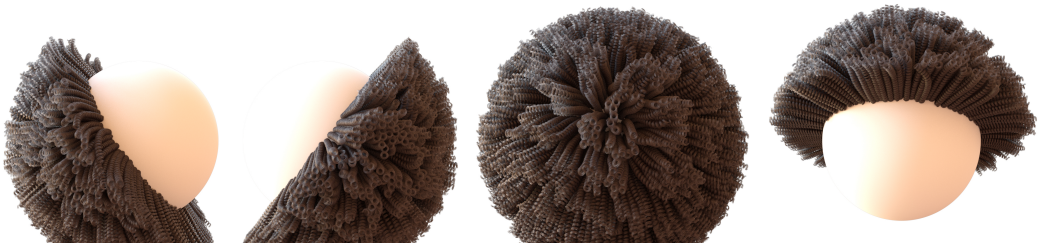
F. Bertails, B. Audoly, MP Cani, B. Querleux, F. Leroy, and JL Lévêque. 2006. Super-helices for predicting the dynamics of natural hair. *ACM Trans. Graph.* 25, 3 (2006), 1180–1187.

F. Bertails, B. Audoly, B. Querleux, F. Leroy, JL Lévêque, and MP Cani. 2005. Predicting natural hair shapes by solving the statics of flexible rods. In *Eurographics short papers*. Eurographics.

F. Bertails, TY Kim, MP Cani, and U. Neumann. 2003. Adaptive wisp tree-a multiresolution control structure for simulating dynamic clustering in hair motion. In *Proc. of Symposium on Computer Animation*.

H. Bryant and CE Porter. 2012. Hair Ethnicity. In *Practical Modern Hair Science*, T. Evans and RR Wickett (Eds.). Allured Business Media, Chapter 6, 193–222.

James R Bunch, Christopher P Nielsen, and Danny C Sorensen. 1978. Rank-one modification of the symmetric eigenproblem. *Numer. Math.* 31, 1 (1978), 31–48.



(a) 2000 wisps, 17.7 seconds per frame

(b) 4000 wisps, 27.6 seconds per frame

(c) 8000 wisps, 57.0 seconds per frame

Fig. 7. **Left to right:** hairball rotations to the left, right, towards the viewer, and away from the viewer. **Top to bottom:** 2000 wisps naturally form a "clumped" look. Maintaining total hairs at 200K, while increasing wisps to 4000 and 8000, gives progressively more diffuse, "picked out" looks. Images are high-res; please zoom in to see details.

Fig. 8. **Left pair:** Comparison to real-world "clumped" look. **Right pair:** Comparison to real-world "picked out" look. Images are from Alexander [2023].

A. Butts, B. Porter, D. Van Gelder, M. Hessler, V. Krishna, and G. Monheit. 2018. Engineering full-fidelity hair for Incredibles 2. In *ACM SIGGRAPH Talks*. 1–2.

J. Chang, F. Da, E. Grinspun, and C. Batty. 2019. A Unified Simplicial Model for Mixed-Dimensional and Non-Manifold Deformable Elastic Objects. *Proc. of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–18.

Byoungwon Choe, Min Gyu Choi, and Hyeong-Seok Ko. 2005. Simulating complex hair with robust collision handling. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 153–160.

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but responsive cloth. *ACM Trans. Graph.* 21, 3 (2002), 604–611.

Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A Otaduy. 2014. Yarn-level simulation of woven cloth. *ACM Trans. Graph.* 33, 6 (2014), 1–11.

Gilles Daviet. 2020. Simple and scalable frictional contacts for thin nodal objects. *ACM Trans. Graph.* 39, 4 (2020), 61–1.

R. De La Mettrie, D. Saint-Léger, G. Loussouarn, A. Garcel, C. Porter, and A. Langaney. 2007. Shape variability and classification of human hair: a worldwide approach. *Human biology* 79, 3 (2007), 265–281.

C. Deul, T. Kugelstadt, M. Weiler, and J. Bender. 2018. Direct position-based solver for stiff rods. In *Computer Graphics Forum*, Vol. 37. 313–324.

D. Eberle. 2022. Personal Communication.

Olaf Etzmuß, Michael Keckeisen, and Wolfgang Straßer. 2003. A fast finite element solution for cloth modelling. In *Proc. of Pacific Graphics*. 244–251.

Y. Fei, C. Batty, E. Grinspun, and C. Zheng. 2019. A multi-scale model for coupling strands with shear-dependent liquid. *ACM Trans. Graph.* 38, 6 (2019), 1–20.

Y. Fei, H. Maia, C. Batty, C. Zheng, and E. Grinspun. 2017. A multi-scale model for simulating liquid-hair interactions. *ACM Trans. Graph.* 36, 4 (2017), 1–17.

Gene H Golub and Charles F Van Loan. 2013. *Matrix computations*. JHU press.

G. Gornowicz and S. Borac. 2015. Efficient and stable approach to elasticity and collisions for hair animation. In *Proc. of the Symp. on Digital Production*. 41–49.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

X. Han, T. Gast, Q. Guo, S. Wang, C. Jiang, and J. Teran. 2019. A hybrid material point method for frictional contact with diverse materials. *Proc. of the ACM on Computer Graphics and Interactive Techniques* 2, 2 (2019), 1–24.

Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin. 2013. Artistic simulation of curly hair. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 63–71.

M Khalid Jawed, Alyssa Novelia, and Oliver M O'Reilly. 2018. *A primer on the kinematics of discrete elastic rods*. Springer.

D. Kaufman, R. Tamstorf, B. Smith, J. Aubry, and E. Grinspun. 2014. Adaptive nonlinearity for collisions in complex rod assemblies. *ACM Trans. Graph.* 33, 4 (2014), 1–12.

T. Kim. 2020. A finite element formulation of Baraff-Witkin cloth. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 1–9.

T. Kim, F. De Goes, and H. Iben. 2019. Anisotropic elasticity for inversion-safety and element rehabilitation. *ACM Trans. Graph.* 38, 4 (2019), 1–15.

T. Kim and D. Eberle. 2022. Dynamic deformables: implementation and production practicalities (now with code!). In *ACM SIGGRAPH Courses*. 1–259.

T. Kugelstadt and E. Schömer. 2016. Position and orientation based Cosserat rods.. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 169–178.

S. Lesser, A. Stomakhin, G. Daviet, J. Wretborn, J. Edholm, N. Lee, E. Schweickart, X. Zhai, S. Flynn, and A. Moffat. 2022. Loki: a unified multiphysics simulation framework for production. *ACM Trans. Graph.* 41, 4 (2022), 1–20.

Huancheng Lin, Floyd M Chitalu, and Taku Komura. 2022. Isotropic ARAP energy using Cauchy-Green invariants. *ACM Trans. Graph.* 41, 6 (2022), 1–14.

S. Martin, P. Kaufmann, M. Botsch, E. Grinspun, and M. Gross. 2010. Unified simulation of elastic rods, shells, and solids. *ACM Trans. Graph.* 29, 4 (2010), 1–10.

A. McAdams, A. Selle, K. Ward, E. Sifakis, and J. Teran. 2009. Detail preserving continuum simulation of straight hair. *ACM Trans. Graph.* 28, 3 (2009), 1–6.

A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. 2011. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.* 30, 4 (2011), 1–12.

D. Michels, P. Mueller, and G. Sobottka. 2015. A physically based approach to the accurate simulation of stiff fibers and stiff fiber meshes. *Computers & Graphics* 53 (2015), 136–146.

Sofya Ogunseitan. 2022. Space Rangers with Cornrows: Methods for Modeling Braids and Curls in Pixar's Groom Pipeline. In *ACM SIGGRAPH Talks*. 1–2.

Dinesh K Pai. 2002. Strands: Interactive simulation of thin solids using cosserat models. In *Computer graphics forum*, Vol. 21. Wiley Online Library, 347–352.

Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Trans. Graph.* 38, 4 (2019), 1–15.

Deborah Patrick, Shaun Bangay, and Adele Lobb. 2004. Modelling and rendering techniques for african hairstyles. In *Proc. of Afrigraph*. 115–124.

RM Sánchez-Banderas, A. Rodríguez, H. Barreiro, and MA Otaduy. 2020. Robust eulerian-on-lagrangian rods. *ACM Trans. Graph.* 39, 4 (2020), 59–1.

Andrew Selle, Michael Lentine, and Ronald Fedkiw. 2008. A mass spring model for hair simulation. *ACM Trans. Graph.* 27, 3 (2008), 1–11.

B. Smith, F. De Goes, and T. Kim. 2018. Stable neo-hookean flesh simulation. *ACM Trans. Graph.* 37, 2 (2018), 1–15.

B. Smith, F. De Goes, and T. Kim. 2019. Analytic eigensystems for isotropic distortion energies. *ACM Trans. Graph.* 38, 1 (2019), 1–15.

Carlota Soler, Tobias Martin, and Olga Sorkine-Hornung. 2018. Cosserat rods with projective dynamics. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 137–147.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Proc. of Symposium on Geometry processing*, Vol. 4. 109–116.

G. Sperl, RM Sánchez-Banderas, M. Li, C. Wojtan, and MA Otaduy. 2022. Estimation of yarn-level simulation models for production fabrics. *ACM Trans. Graph.* 41, 4 (2022), 1–15.

Jonas Spillmann and Matthias Teschner. 2007. CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer animation*. 63–72.

S. Sueda, GL Jones, D. Levin, and D. Pai. 2011. Large-scale dynamic simulation of highly constrained strands. *ACM Trans. Graph.* 30, 4 (2011), 1–10.

R. Tamstorf, T. Jones, and S. McCormick. 2015. Smoothed aggregation multigrid for cloth simulation. *ACM Trans. Graph.* 34, 6 (2015), 1–13.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 181–190.

M. Thyng, C. Evart, T. Jones, and A. McAdams. 2017. The art and technology of hair simulation in Disney's Moana. In *ACM SIGGRAPH Talks*. 1–2.

N. Umetani, R. Schmidt, and J. Stam. 2015. Position-based elastic rods. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. Comp. Anim.* 21–30.

Andre Walker. 1997. *Andre Talks Hair.* Simon & Schuster.

K. Ward and MC Lin. 2003. Adaptive grouping and subdivision for simulating hair dynamics. In *Proc. of Pacific Graphics*. IEEE, 234–243.

K. Ward, M. Simmons, A. Milne, H. Yosumi, and X. Zhao. 2010. Simulating Rapunzel's hair in Disney's Tangled. In *ACM SIGGRAPH Talks*. 1–1.

FJ Wortmann, G Wortmann, and T Sripho. 2020. Why is hair curly?—Deductions from the structure and the biomechanics of the mature hair shaft. *Experimental dermatology* 29, 3 (2020), 366–372.

H. Wu and T. Kim. 2023. An Eigenanalysis of Angle-Based Deformation Energies. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

Kui Wu and Cem Yuksel. 2016. Real-Time Hair Mesh Simulation. In *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (Redmond, Washington). 59–64.

Cem Yuksel, Scott Schaefer, and John Keyser. 2009. Hair Meshes. *ACM Trans. Graph.* 28, 5 (2009).

J. Zehnder, S. Coros, and B. Thomaszewski. 2021. SGN: Sparse Gauss-Newton for Accelerated Sensitivity Analysis. *ACM Trans. Graph.* 41, 1 (2021), 1–10.