ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2020
J. Bender and T. Popa
(Guest Editors)

Volume 39 (2020), Number 8

A Bending Model for Nodal Discretizations of Yarn-Level Cloth

José M. Pizana[1] Alejandro Rodríguez[1] Gabriel Cirio[1] Miguel A. Otaduy[2]

[1]Seddi Labs, Madrid, Spain
[2]Universidad Rey Juan Carlos, Madrid, Spain

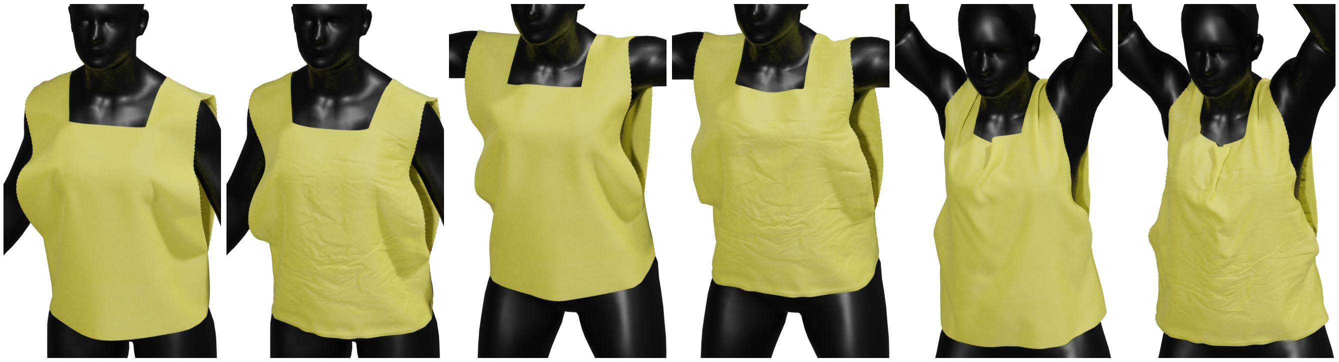

Figure 1: *Why do animated characters always look sharp in their clothes? We introduce a robust, controllable and efficient bending model to yarn-level cloth, which allows the simulation of very high-resolution wrinkles. The images compare a simulation with and without rest-shape wrinkles, which affect the formation of folds.*

Abstract

To deploy yarn-level cloth simulations in production environments, it is paramount to design very efficient implementations, which mitigate the cost of the extremely high resolution. To this end, nodal discretizations aligned with the regularity of the fabric structure provide an optimal setting for efficient GPU implementations. However, nodal discretizations complicate the design of robust and controllable bending. In this paper, we address this challenge, and propose a model of bending that is both robust and controllable, and employs only nodal degrees of freedom. We extract information of yarn and fabric orientation implicitly from the nodal degrees of freedom, with no need to augment the model explicitly. But most importantly, and unlike previous formulations that use implicit orientations, the computation of bending forces bears no overhead with respect to other nodal forces such as stretch. This is possible by tracking optimal orientations efficiently. We demonstrate the impact of our bending model in examples with controllable anisotropy, as well as ironing, wrinkling, and plasticity.

CCS Concepts

• ***Computing methodologies*** $\rightarrow$ ***Physical simulation;***

1. Introduction

Yarn-scale cloth simulation inherently captures the structural complexity of fabrics, and hence the rich nonlinearities of fabric mechanics arise in a natural manner [KJM08]. However, this richness comes at an expensive price. The simulation of yarn-scale contacts and deformation modes requires the discretization of yarns at a density equal or higher to the crossings of yarns, which can easily surpass millions on common garments. Then, production-ready yarn-level simulation code, either for animation or for fashion, must maximize computation speed and minimize memory usage.

We seek a massively parallel simulation algorithm, which leverages the regularity of common fabrics, for a highly efficient GPU deployment. We build on the Eulerian-on-Lagrangian (EoL) model of Cirio et al. [CLMMO14], which sets simulation nodes at yarn crossings and handles inter-yarn contacts as persistent constraints with yarn sliding. Their regular nodal discretization simplifies memory access and favors efficiency on a GPU implementation.

Unfortunately, nodal discretizations of yarns complicate the design of bending models. The lack of yarn orientation prevents controllable anisotropy, and curved rest shapes challenge robustness through undefined gradients and multiple stable configurations. To avoid these difficulties, yarn and rod models often carry explicit orientations [Pai02, BWR*08, ST09]. However, we wish to retain a purely nodal discretization to maximize implementation efficiency.

In this paper, we introduce a robust and controllable bending model for yarn-level cloth simulation with nodal discretizations. As discussed in Section 3, our model defines an orientation on each yarn crossing, and measures bending of a yarn segment based on its deviation from the orientation of the crossing. We can easily define non-zero rest bending, which enables effects such as ironing and wrinkling, as well as the curved rest shapes of some knit fabrics. Using the orientation of crossings, we can also model anisotropy without interference of in-plane and out-of-plane bending.

We define the orientation of a crossing as the minimizer of the bending energy of the crossing. In Section 4 we describe how to compute this orientation through a simple optimization. But, most importantly, in Section 5 we demonstrate that, in contrast to prior work that uses implicit orientations, the optimality of the orientation simplifies dramatically the computation of bending forces on yarn nodes.

We demonstrate the impact of our bending model in examples with rich yarn-scale effects, such as ironing, wrinkling, and plasticity. As shown in Figure 1, our animated clothing is not sharp and free of wrinkles. The combination of yarn-level discretization with robust and controllable bending enables realistic wrinkles.

2. Related Work

Traditionally, computer graphics research has treated fabric as a continuous elastic material, with a wide variety of bending models for particle discretizations [Pro95, CK02] and triangle discretizations [BW98, BMF03, GHDS03]. Yarn-level models, however, follow a fundamentally different approach by explicitly modeling every yarn in the fabric. Our work relies mainly on two sub-fields of computer simulation: rod simulation and yarn-level cloth simulation, that will be covered in the following subsections.

2.1. Simulation of Rods

There is no shortage of models to simulate rods. The simplest approaches use mass-spring systems [RCTI91, SLF08, IMP*13], trading accuracy for speed. Most physically grounded methods, however, describe rods using adapted frames along a curve. By discretizing the Cosserat geometry model, early approaches [Pai02, ST07] benefit from an explicit representation of the centerline using full coordinates, but need constraints to force the frames to follow the centerline. Reduced coordinate models, on the other hand, can avoid constraints while retaining a compact representation of the centerline. An example of this is the discrete elastic rod model of Bergou et al. [BWR*08, BAV*10], whose curve-angle representation enforces the oriented frame to always follow the centerline. Since, in addition, the centerline is treated explicitly, the method has become very popular in the graphics community. Other approaches use reduced coordinates [BAC*06], albeit with an implicit treatment of the centerline.

Since we are interested in modeling bending at yarn crossings, we seek inspiration from methods that can efficiently simulate rod junctions. Cosserat Nets [ST09] use Cosserat Rods [ST07] to model networks of elastic rods linked by elastic joints. To enforce the constraints inherent to the Cosserat model, the authors resort to using penalty forces and coordinate projection. Pérez et al. [PTC*15] model rod networks using the discrete elastic rod model [BWR*08]. For each rod incident to a junction, bending and twist energies are formulated with respect to all other rods in the junction after removing any rigid transformation. This rigid transformation is computed by minimizing the deformation of all incident rods with respect to the rest pose of the junction. Using an implicit solver becomes challenging due to dependencies among incident rods and the use of junction rotations. Zehnder et al. [ZCT16] circumvented this problem by defining junction rotations as explicit degrees of freedom, in contraposition to Pérez et al. [PTC*15] implicit approach.

2.2. Yarn-Level Simulation of Cloth

The first incursion into yarn-level modeling of fabrics can be attributed to Peirce [PFF37], who proposed in 1937 a geometric model of yarns crossing in woven fabric. Since then, textile research has devoted a lot of attention to yarn-level models to better understand the behavior of fabric. Ranging from analytical yarn models [HGB69, KNK73] to continuum models of yarns [NTL98, PW00, DKBP06], multiscale models [NPS06], or simple beams and trusses [Ree03, MCM03], these approaches usually focus on small portions of fabric in controlled experiments, ignoring the computational complexity of full-size garments.

Addressing this shortcoming, the seminal work of Kaldor et al. [KJM08, KJM10] emerged as the first to be able to simulate entire knitted garments at the yarn level, from scarfs to sweaters, using inextensible rods and stiff penalty forces for resolving contacts between yarns. More recently, Leaf et al. [LWS*18] used this work to simulate knit patches at interactive rates, leveraging an efficient GPU implementation and the periodicity of boundary conditions. The model of Kaldor et al. also has been used in conjunction with the *stitch mesh* data structure to define the construction of knit garments as a tiling procedure [YKJM12].

Instead of explicitly computing yarn-yarn contacts throughout the fabric, Cirio et al. [CLMMO14, CLMO17] considered contacts as persistent in time, thus avoiding the expensive treatment of collisions while still capturing yarn-level behavior in full garments. They modeled yarns as flexible rods, and added sliding degrees of freedom at yarn crossings, allowing complex plastic effects, such as snags. Recently, Sánchez-Banderas et al. [SBRBO20] have improved the discretization of persistent contacts, extending the applicability of the model to more complex fabrics. Earlier, they also improved the damping behavior of yarn-level models, making it more controllable [SBO18].

In our work, we seek to achieve a good trade-off between the efficiency and quality of yarn-level simulation. Other works try to optimize this trade-off in different ways. Recently, Casafranca et al. [CCR*20] have developed a method to mix yarn and triangle representations on the same simulation, focusing the use of yarns only at regions of interest. Sperl et al. [SNW20], on the other hand,

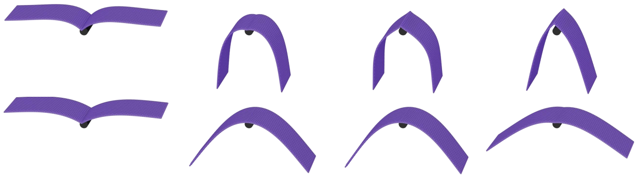

Figure 2: *Simulation of a patch of cloth with a crease, flapping under gravity. Top: using the bending model of Cirio et al. [CLMMO14], the rest shape flips when the crease flattens out. Bottom: using our bending model, the simulation is robust.*

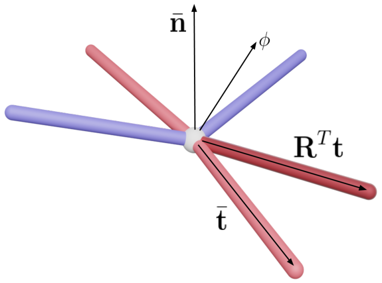

Figure 3: *A crossing of two yarns in rest shape, showing the main elements of our bending model. We characterize each yarn segment by its length vector* **t***, and the crossing by its orientation* **R***. We rotate a deformed segment to the rest shape,* $\mathbf{R}^T\,\mathbf{t}$*, and we measure the axis angle* ϕ *w.r.t. the rest-shape segment vector* $\bar{\mathbf{t}}$*. With the rest-shape normal of the crossing,* $\bar{\mathbf{n}}$*, we can separate in-plane and out-of-plane bending.*

have introduced a homogenization method that estimates thin-shell deformation models from yarn-scale simulations. Finally, Jiang et al. [JGT17] use the Material Point Method to track yarn-level details in a Lagrangian manner, while resolving collisions on an Eulerian grid.

The various yarn-level simulation methods differ in how they model bending. Kaldor et al. [KJM08] first modeled bending using the curvature of a non-oriented spline representation of yarns. They supported only straight rest shapes. Later [KJM10], they adopted the discrete elastic rod model of Bergou et al. [BWR*08, BAV*10], with non-straight rest shape but isotropic bending. Cirio et al. [CLMMO14, CLMO17] modeled bending using the angles between piecewise linear yarn segments, allowing non-zero rest angles. As discussed in the introduction, we favor a purely nodal discretization due to its convenience for efficient GPU implementation. However, with a purely nodal discretization, and due to the lack of curve orientation, the simulation of non-straight rest shapes leads to robustness problems in previous models. The bending energy of Cirio et al. is defined as $E = \frac{1}{2}\,k\,(\psi - \bar{\psi})^2$, with ψ the angle between two adjacent segments, $\bar{\psi}$ the rest angle, and k the stiffness. The bending force is $\mathbf{f} = -k\,(\psi - \bar{\psi})\,\nabla\psi$. Unfortunately, $\nabla\psi$ is undefined for $\psi = 0$. For straight yarns, this is not a problem, as the gradient is multiplied by the zero angle. But for non-straight yarns, the force is discontinuous. The yarn can pick any of the infinite stable configurations. As shown in Figure 2, this model is not robust, and a patch with a crease can flip toward a different rest shape. In contrast, our model, presented next, handles non-straight rest shapes robustly.

3. Formulation of the Bending Model

In this section, we present our novel bending model for yarn-level cloth models. The section starts with definitions of the relevant variables, and it follows with the definitions of bending strains and the resulting energies.

3.1. Definitions

We model yarns as deformable rods, discretized by a sequence of nodes. We group all node positions of a yarn-level discretization in a large vector **x**. One important assumption of our model is that nodes are placed at yarn crossings, as bending is mostly concentrated at contacts, and yarns can be considered in tension and straight between crossings. We ignore the twist deformation of yarns, as done by others before [KJM08, CLMMO14], since the tight packing of yarns prevents any noticeable twist. In our work, we build on the Eulerian-on-Lagrangian method of Cirio et al. [CLMMO14] for contact handling, but our method is not limited to Eulerian-on-Lagrangian discretizations.

For each yarn segment incident on a crossing, let us define a vector **t** joining the crossing with the next crossing. This is the difference vector between adjacent yarn nodes, as depicted in Figure 3. We denote rest-shape quantities with an overbar, e.g., $\bar{\mathbf{t}}$ denotes the rest-shape yarn segment. We also characterize a yarn crossing by its orientation **R** (a rotation matrix), with the normal **n** oriented perpendicular to the fabric surface. The estimation of the orientation is discussed in Section 4. We initialize the rest-shape normal $\bar{\mathbf{n}}$ by fitting a plane to the incident yarn segments $\{\bar{\mathbf{t}}_i\}$. In most fabrics, the rest shape is flat and the normal is trivially defined. However, we use non-flat rest shapes to model ironing, plastic wrinkles, or high-resolution bending of knit fabrics such as rib knits.

3.2. Bending of Yarn Segments

As noted earlier, our yarn bending model is inspired by the model of rod networks of Pérez et al. [PTC*15], but we extend their concepts to yarn crossings. The main insight of their model is that yarn bending is captured in the deviation from the overall orientation of a connection (a crossing in our case). However, we depart in how to account for anisotropy (see later in this subsection) and how to estimate orientations (See Section 4).

Given a deformed yarn configuration, we estimate the overall orientation **R** of each yarn crossing. Based on this orientation, one could define the rotated rest segments $\mathbf{R}\,\bar{\mathbf{t}}$, and characterize bending of the corresponding yarn by the axis angle between the rotated rest segment and the current segment **t**. In practice, we choose instead to rotate the current segment to the rest shape of the crossing, $\mathbf{R}^T\,\mathbf{t}$, and measure the axis-angle vector ϕ with the rest segment:

$$\phi = \text{axis\_angle}\left(\bar{\mathbf{t}}, \mathbf{R}^T\,\mathbf{t}\right). \qquad (1)$$

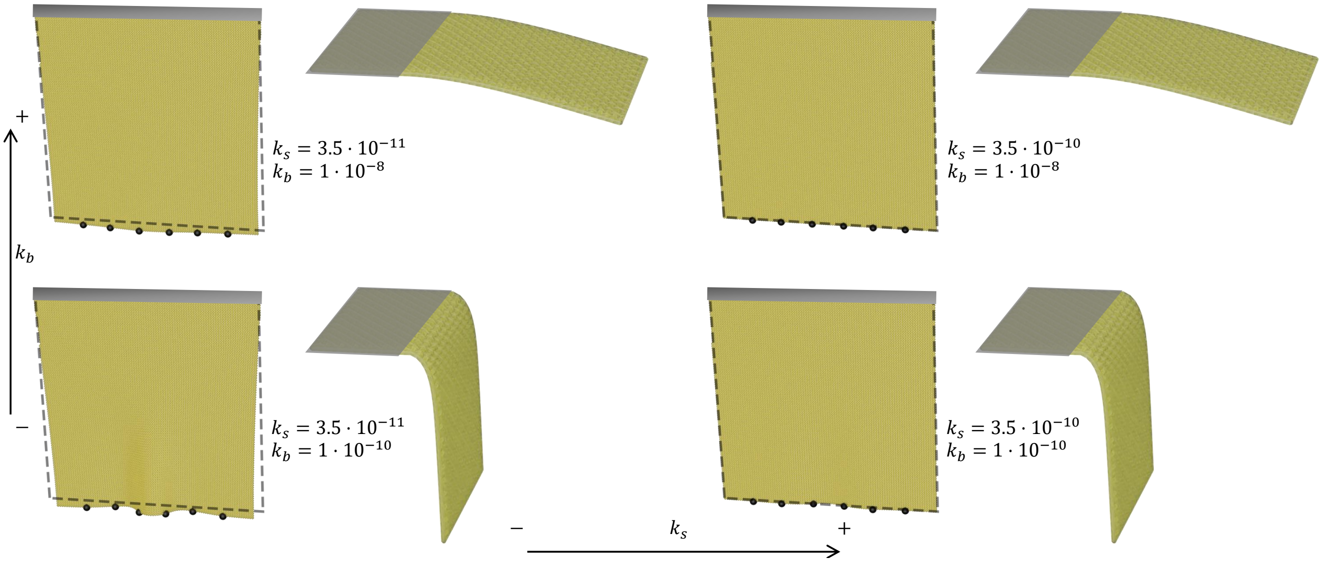

Figure 4: *Control of bending anisotropy with our model. We simulate two hanging cloths (from an edge, cut at 45 degrees, and from a strip) with a matrix of values of in-plane stiffness (k_s, fabric shear stiffness) and out-of-plane stiffness (k_b, fabric bending stiffness). As evidenced in the images and the accompanying video, the stiffness of the two deformation modes can be controlled independently. For better viewing the shear deformation when the cloth hangs from an edge, we draw the contour of the undeformed shape with a dotted line.*

The exact expression of the axis-angle vector is given in the Appendix.

The axis angle ϕ can capture bending of the yarn around another yarn (e.g., bending of a warp yarn around a weft yarn), but it can also capture shear between crossing yarns. Different bending directions lead to different intensity of contact forces between the crossing yarns, and this suggests the need for an anisotropic bending model. We use the orientation of the axis angle ϕ w.r.t. the rest-shape normal $\bar{\mathbf{t}}$ of the crossing to distinguish in-plane yarn bending (i.e., fabric shear) vs. out-of-plane yarn bending (i.e., fabric bending).

We define in-plane bending strain of a yarn segment as the component of the axis angle in the direction of the normal of the crossing:

$$\varepsilon_s = \frac{1}{L} \bar{\mathbf{n}}^T \phi. \tag{2}$$

Similarly, we define out-of-plane bending strain of a yarn segment as the component of the axis angle in the null space of the normal of the crossing:

$$\varepsilon_b = \frac{1}{L} \left(\mathbf{I} - \bar{\mathbf{n}} \bar{\mathbf{n}}^T \right) \phi. \tag{3}$$

In both cases, the strain definition includes a division by the segment length L, as this yields a metric of discrete curvature. This metric will allow the definition of strain energy densities with discretization-independent stiffness values, which can be integrated along the yarn segment.

Pérez et al. [PTC*15] built their rod network model on top of the rod model of Bergou et al. [BWR*08]. In their case, anisotropy is determined by the shape of the rod cross-section, and captures different bending resistances in the normal and binormal directions of the rod. This is different from our notion of anisotropy, which is determined by the orientation of the crossing, not the yarn itself.

3.3. Bending Energy of a Yarn Crossing

Based on the in-plane (i.e., shear) and out-of-plane (i.e., bending) strains in (2) and (3) respectively, and with in-plane stiffness k_s and out-of-plane stiffness k_b, we define the total bending energy of a yarn segment as

$$E_{segment} = \frac{1}{2} L k_s \varepsilon_s^2 + \frac{1}{2} L k_b \varepsilon_b^T \varepsilon_b. \tag{4}$$

The energy can be written in a compact way by defining an anisotropic discrete stiffness matrix $\mathbf{K} = \frac{1}{L} k_b \mathbf{I} + \frac{1}{L} (k_s - k_b) \bar{\mathbf{n}} \bar{\mathbf{n}}^T$:

$$E_{segment} = \frac{1}{2} \phi^T \mathbf{K} \phi. \tag{5}$$

In practice, and to maximize efficiency, we orient the rest configuration such that $\bar{\mathbf{n}} = (0,1,0)^T$ is aligned with the Y axis. Then, the stiffness matrix is diagonal, $\mathbf{K} = \frac{1}{L} \begin{pmatrix} k_b & 0 & 0 \\ 0 & k_s & 0 \\ 0 & 0 & k_b \end{pmatrix}$.

Adding together the energies of all segments incident in a crossing, we can define the bending energy of a yarn crossing:

$$E_{crossing} = \sum_i E_{segment,i}. \tag{6}$$

As shown in Figure 4, we succeed to control independently the stiffness of in-plane and out-of-plane deformations. We simulate hanging patches of woven cloth in two different setups: (a) cut at 45 degrees and hanging from an edge with additional weights, to produce fabric shear deformation, and (b) cut at 0 degrees and hanging from a strip under its own weight, to produce fabric bending deformation. The shear deformation is dominated by the in-plane stiffness k_s, while the bending deformation is dominated by the out-of-plane stiffness k_b, thus validating correct handling of anisotropy in our model.

4. Orientation of a Yarn Crossing

The orientation $\mathbf{R}$ of a yarn crossing is a key ingredient of our bending model. In the bending model of Pérez et al. [PTC*15], the orientation is defined geometrically, following a shape matching procedure [MHTG05]. This approach decouples the definition of the orientation from the force model.

We propose, in contrast, to define the orientation as the optimizer of the bending energy of the yarn crossing (6). At first sight, this choice increments the computational complexity, since the bending energy is more complex than the shape matching energy. However, as discussed in detail in Section 5, our choice simplifies dramatically the computation of bending forces. The savings in force computation largely outweigh the overhead in the estimation of orientations.

On every simulation step, after each node position update, we recompute the optimal crossing orientations. To this end, we follow a Gauss-Newton scheme with line-search. We express an incremental rotation with axis-angle vector θ, which yields a linear approximation $(\mathbf{I} + \text{skew}(\theta))\,\mathbf{R}$ to the rotation, with $\text{skew}(\theta) = \begin{pmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{pmatrix}$. Using this linear approximation, we iterate until convergence the optimality condition of the bending energy with respect to the orientation:

$$\frac{\partial E_{crossing}}{\partial \theta} = 0. \tag{7}$$

By substituting (5), (6) and the linear approximation of ϕ with respect to θ, the optimality condition (7) translates into:

$$\left(\sum_i \frac{\partial \phi_i}{\partial \theta}^T \mathbf{K}_i \frac{\partial \phi_i}{\partial \theta} \right) \theta = -\sum_i \frac{\partial \phi_i}{\partial \theta}^T \mathbf{K}_i \, \phi_i. \tag{8}$$

With this equation, we solve for θ, update the rotation $\mathbf{R}$, validate that the energy of the crossing decreases (and apply line search otherwise), and execute a new iteration of (8). The derivation of the Jacobian of the bending axis angle ϕ is discussed in the Appendix.

An alternative to our Gauss-Newton approach would have been to first compute a linear transformation and then find the closest rotation through polar decomposition (or novel alternatives [MBCM16]). However, the computation of the linear transformation is not well conditioned if the rest configuration is flat, which requires special treatment. Our Gauss-Newton approach does not suffer from this problem and converges fast under the typical temporal coherence of dynamic simulations.

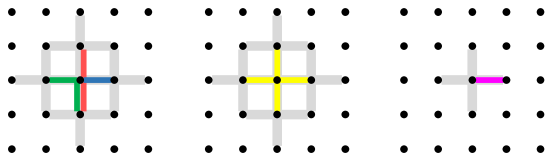

Figure 5: *Computation stencils for various force models, shown on a typical woven fabric with nodes at yarn crossings. The colors indicate the stencils of individual forces; the gray patterns indicate the stencils of the force Jacobians, i.e., their sparsity patterns. Left: stretch (blue), bending (red) and shear (green) stencils for the force models of Cirio et al. [CLMMO14]. Middle: bending/shear stencil (yellow) with crossing orientations computed based on shape matching [PTC*15]. Right: our bending/shear stencil (magenta).*

5. Bending Forces

For each yarn crossing, we must compute bending forces resulting from (6). This requires the evaluation of the full gradient of the energy of the crossing w.r.t. nodal positions $\mathbf{x}$. These affect the energy of the crossing in two ways: directly through the change of axis angles of the incident yarn segments, and indirectly through the change of the optimal crossing orientation. Therefore, the full gradient should be computed taking into account also the implicit change of the crossing orientation θ:

$$\mathbf{f} = -\frac{\partial E_{crossing}}{\partial \mathbf{x}}^T - \frac{\partial \theta}{\partial \mathbf{x}}^T \overset{=0}{\cancel{\frac{\partial E_{crossing}}{\partial \theta}^T}}. \tag{9}$$

The Jacobian of the change of orientation, $\frac{\partial \theta}{\partial \mathbf{x}}$, can be obtained by applying the implicit function theorem to the optimality condition of the orientation (8). A similar procedure was followed by Pérez et al. [PTC*15] and, while doable, it carries a notable computational cost. Moreover, the evaluation of bending forces on a yarn node requires visiting the 2-ring of nodes, due to mutual influence on optimal crossing orientations. This large computational stencil is problematic as we seek a highly efficient GPU implementation.

Fortunately, and as highlighted in (9), one can avoid altogether the Jacobian of the change of orientation. Since the orientations of crossings are optimal under the same bending energy, the gradient $\frac{\partial E_{crossing}}{\partial \theta}$ naturally cancels out. Bending forces can be computed by evaluating solely the direct gradient w.r.t. nodal positions. We have tested computing the orientation following a shape matching procedure, as Pérez et al. [PTC*15], together with ignoring the Jacobian of the change of orientation. As shown in the accompanying video, this simplification yields wrong forces that produce a complete mismatch in the simulation.

The simplification of the formulation has also a subtle implication, which is however important for an efficient GPU implementation. The bending force can be computed separately for each yarn segment, and in this way the force computation kernel only needs to access the two incident nodes:

$$\mathbf{f}_i = -\frac{\partial E_{segment,i}}{\partial \mathbf{x}}^T = -\frac{\partial \phi_i}{\partial \mathbf{x}}^T \mathbf{K}_i \, \phi_i. \tag{10}$$

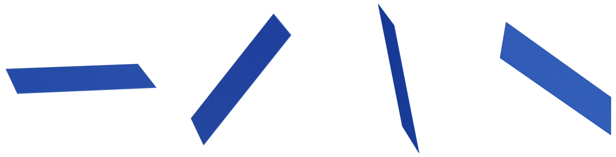

Figure 6: *Preservation of angular momentum of a rotating patch of cloth, with no gravity. With our implicit definition of optimal rotations, this patch satisfies the "tennis racket theorem", as expected. Please watch the accompanying video.*

The derivation of the Jacobian of the bending axis angle ϕ is discussed in the Appendix.

For implicit integration, we approximate the force Jacobian as

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}} = -\frac{\partial \phi_i}{\partial \mathbf{x}}^T \mathbf{K}_i \frac{\partial \phi_i}{\partial \mathbf{x}}. \tag{11}$$

An accurate expression would require the Hessian of the bending axis angle as well as Jacobians w.r.t. the crossing orientation. However, in practice we found the expression above to provide a good search direction in the Newton solve of implicit integration. As shown in Figure 5, our force model yields both force and Jacobian stencils that are considerably more compact. Combining the compactness of the bending force stencil with the approximation of its Jacobian, we manage to reduce the number of non-zero blocks per node from 13 in the work of Cirio et al. [CLMMO14] to just 5.

We have tested that, with the implicit definition of optimal rotations, our bending model preserves angular momentum correctly. Figure 6 shows several snapshots of the animation of a rotating patch of cloth, with no gravity. It satisfies the "tennis racket theorem", i.e., the rotation around the intermediate principal axis of the inertia tensor is not stable. Please watch the accompanying video. After one second of simulation the angular momentum decreases by just 8%, due to numerical damping of the integration method.

6. Results

We have implemented the proposed bending model in a massively parallel GPU simulator, following the design of Cirio et al. [CLMMO14, CLMO17]. This section validates and showcases the different properties of the model. We refer the reader to the accompanying video for the animations discussed in this section. The section concludes with a summary of performance.

6.1. Control of Bending Properties

As discussed already in Section 3, our bending model allows the independent control of the fabric's rest shape and in-plane and out-of-plane bending properties. For the simulations shown in Figure 4, we have used as reference material parameters estimated from a patch of linen fabric. The material has the following properties. For both weft and warp: 2.3 yarns per mm, with a radius of 0.17 mm, and 40 mg/m. The stretch stiffness is 1.35 in weft and 29.5 in warp. For shear and bending, we have used the same reference stiffness values in weft and warp, $k_b = 10^{-8}$ and $k_s = 3.5 \cdot 10^{-10}$. In the hanging simulations, we compare the results with experimental values of $k_b = 10^{-10}$ and $k_s = 3.5 \cdot 10^{-11}$. To test the bending behavior, we hang from a strip a small patch of 30 mm ×30 mm, with 2 200 simulation nodes, cut at 0 degrees. To test the shear behavior, we hang from an edge a larger patch of 83 mm ×83 mm, with 36 358 simulation nodes, cut at 45 degrees. We used a larger patch plus extra weights to induce more evident shear deformations.

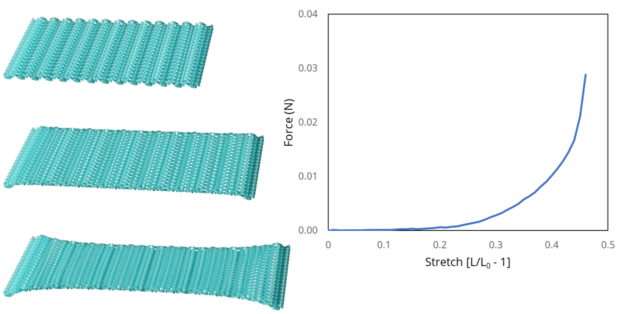

Figure 7: *As a 2x2 rib knit fabric is stretched (left, top to bottom), first its undulations are flattened, then loops are straightened, and finally yarns resist stretch. This complex geometric deformation renders a nonlinear force response to the overall stretch (right). We achieve this complex behavior thanks to the controllability of our bending model, setting non-flat rest shapes and anisotropic stiffness values.*

Our model also allows accurate simulation of knitted fabrics. Figure 7 showcases the complex geometric behavior and nonlinear macroscopic force response of a 2x2 rib knit patch during a stretching deformation. In the first regime, the patch offers small resistance while the rib undulations are flattened. Then the loop shapes are deformed by the straightening of the yarns along the stretch direction, adding more resistance to the motion. Finally yarns are fully straight, leading to a stiff response caused by the stretch stiffness of the yarns. To simulate this complex nonlinear behavior, we set a non-planar rest shape following the rib undulations, and we assign a lower value to the out-of-plane stiffness than the in-plane stiffness, to favor flattening of undulations prior to straightening of loops. In the model of Cirio et al. [CLMO17], the forces in the rib undulations were modeled by adding complex *wrapping* forces, which can turn problematic under strong stretch, as the loops deform.

With our model, it is easy to control separately the draping and stretch behavior of a knit fabric, which are dominated by out-of-plane and in-plane bending stiffness, respectively. This degree of control is difficult with previous yarn-level models, either derived from the work of Kaldor et al. [KJM08], where the inter-yarn contact stiffness plays a major role, or derived from the work of Cirio et al. [CLMMO14], which cannot effectively separate the various bending modes. Figure 8 shows different drape behaviors by varying in-plane and out-plane bending stiffness of a knit 2x2 rib patch. The patch has 34 408 simulation nodes, with a loop density of 435 loops/m in the course direction and 555 loops/m in the wale direction. The yarns have a radius of 1 mm and a mass density of 200 mg/m. We have used a stretch stiffness of 10, and the in-plane and out-of-plane bending stiffness values are shown in the figure.

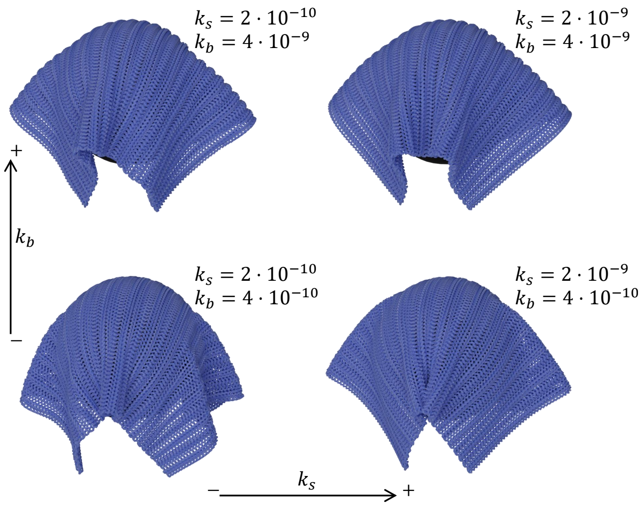

Figure 8: *A 2x2 rib knit patch draped on a sphere, simulated with four combinations of in-plane stiffness (k_s) and out-of-plane stiffness (k_b). With our model, one can control the drape behavior of a knit fabric (dominated by out-of-plane bending) independently from its stretch behavior (dominated by in-plane bending).*

6.2. Rest-Shape Editing

Our model allows the robust handling of non-zero rest angles, which are critical to represent several interesting effects present in cloth and garments, such as wrinkles, ironing or seams. The following scenarios showcase the modeling and robust handling of these effects using our model.

One can introduce ironing creases easily by defining a non-zero rest angle along a line of the fabric. As a preprocess, we edit the local segment vectors of the crossings traversed by the crease, and we estimate the rest orientation of the crossings accordingly. As shown in Figure 9, the ironing lines are clearly preserved during the simulation. As shown in the accompanying video, they are also simulated robustly in the presence of collisions.

By modifying the rest angles dynamically, one can also model plasticity effects. In Figure 10 a piece of fabric is first compressed inside a shrinking sphere. When the maximum compression is reached, we modify the rest angles to match the current deformation. Later, we pin the fabric at two corners and let it hang, and the plastic deformations are clearly visible. While we did not explore it to date, physics-based plastic effects could be added to our model by modifying the rest angles according to a yield criterion.

Another convenient approach to apply wrinkles, easy to introduce in an artistic production environment, is the use height maps that can be directly applied to modify the planar rest shape of any piece of fabric. We use this approach to edit wrinkle maps and custom printed designs, as shown in Figure 11 Note that the wrinkles affect the mechanical behavior, introducing a bias direction to bending deformations, and they flatten or bend further during the simulation. These effects cannot be achieved by simply adding a height map as a postprocess.

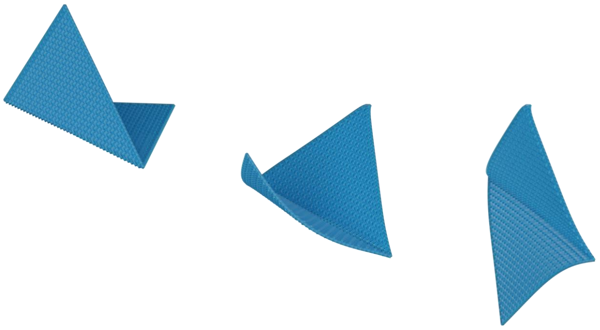

Figure 9: *A square piece of fabric ironed on one diagonal with a rest angle of 90 degrees. The crease is preserved during the simulation, and it affects the overall mechanical behavior as expected.*

Finally, Figure 1 shows our bending model on a full garment. The model behaves well under dynamic simulations with fast motion and collisions. Similar to the example in Figure 11, we add a wrinkle map to the garment. The wrinkles are preserved during the full simulation and affect the mechanical behavior of the fabric material. This example uses a plain weave fabric with the following settings, for both warp and weft yarns: 746 yarns per meter, with a radius of 0.26 mm, and 90 mg/m. The stretch stiffness is 10. For shear and bending, we have used stiffness values of $k_b = 3.5 \cdot 10^{-9}$ and $k_s = 8 \cdot 10^{-10}$. The garment totals 229 214 simulation nodes.

6.3. Performance

We have executed our simulations on an AMD Ryzen 7 3700X CPU with 32 GB of RAM, and a NVIDIA GeForce GTX 1080 Ti GPU with 11 GB of RAM. For the shirt in Figure 1, we have used adaptive time stepping. We set a reference time step of 1 ms, which is maintained during 95% of the simulation. In the remaining 5%, the time step goes down to 0.5 or 0.25 ms. The small time step is required by the high stiffness of stretch forces together with the presence of small yarn segments at seams. The cost of a simulation step is of 5.8 seconds on average. The complete simulation lasts 8 seconds, which amounts to a total computation time of roughly 14 hours. We have also compared performance with the bending and shear model of Cirio et al. [CLMMO14], as well as the use of shape matching for the estimation of rotations [PTC*15]. With respect to the model of Cirio et al., our approach incurs in an overhead of 85%, albeit with the improved properties discussed in the paper. Despite the reduction in the computational stencil discussed in Section 5, our approach requires the extra cost of estimating crossing orientations. With respect to the use of shape matching, we achieve a performance gain of 2.7×.

7. Conclusions and Future Work

We have presented a novel bending model for yarn-level cloth simulation, which features controllability and robustness properties that were missing in previous nodal discretizations. As shown in the examples, this novel model allows the simulation of wrinkling effects that were difficult before.

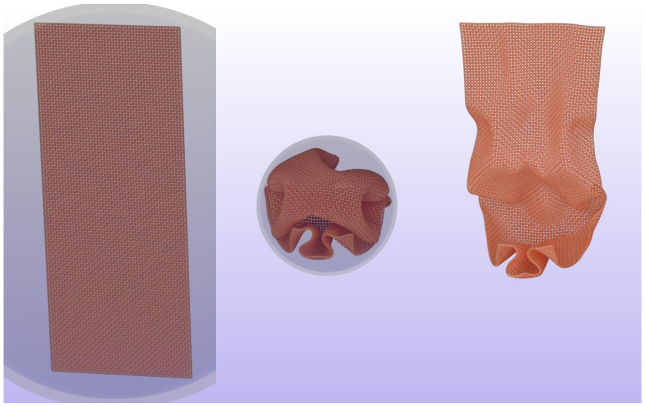

Figure 10: *A hanging piece of cloth (left) is crumpled by a shrinking sphere (middle). Then, we modify the rest angles of our bending model according to this shape, to produce a plasticity effect. As we remove the sphere and let the cloth hang, the deformed shape is largely affected by the crumpling state (right).*

Our model suffers some limitations that motivate lines for future work. While it supports anisotropy, the differences in bending stiffness are limited to in-plane vs. out-of-plane behavior. Since out-of-plane bending captures the effect of yarns bending around each other, it appears relevant to make this stiffness asymmetric as well. Similarly, we use a constant stiffness for all bending angles, but in reality creases exhibit a highly nonlinear stiffness. We have observed that prescribed wrinkles can momentarily flatten out in our simulations, but this effect is not present in reality. Altogether, it appears interesting to make bending forces nonlinear w.r.t. the bending angle, with the stiffness a function of the rest angle. In connection with this extension, stiffness functions and rest angles could be estimated from deformation examples of real fabrics.

In the proposed bending model, forces are a function of the orientation of a crossing and the orientation of each particular yarn segment. Effectively, this approach projects the full dimensionality of the bending behavior to a six-dimensional space. This projection involves inevitable simplifications w.r.t. the full setting. In addition, as acknowledged in the paper, we ignore yarn twist. However, one could try to estimate yarn twist from the relative orientations of adjacent crossings.

In some examples, we have applied plasticity simply by changing the rest angles of yarn segments. However, plasticity and friction could be modeled by applying yield and Coulomb models to the axis angle, respectively. These plasticity and friction effects would complement the sliding coordinates of the yarn-scale model.

To conclude, we pay attention to numerical approximations adopted in our solvers. Both in the computation of rotations and in the implicit integration solve, we discard second derivatives of the axis angle. We found this approximation to be effective in dynamic simulations, thanks to the high temporal coherence. However, this approximation may not be suitable for static simulations that try to take larger steps.

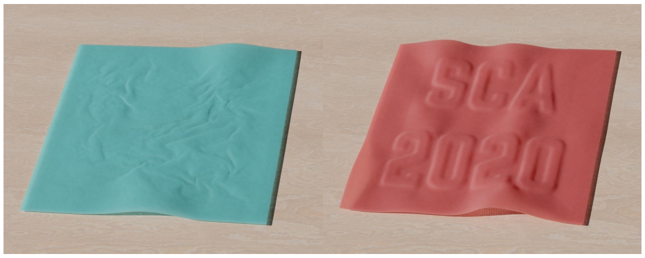

Figure 11: *Using height maps, one can define non-flat rest shapes. In these examples, we use height maps to create wrinkle maps (left) and custom printed designs (right). Note how the rest-shape wrinkles affect the mechanical behavior of the fabric, which cannot be achieved by simply adding wrinkles as a postprocess.*

Acknowledgments. The authors would like to thank Jesús Pérez for many discussions, Juanjo Casafranca for help with the rendering setup, and Igor Santesteban for the character simulation.This work was funded in part by the Spanish Ministry of Science (Industrial doctorate fellowship DIN2019-010912 and grants PTQ-17-09154, PTQ-17-09156, RTI2018-098694-B-I00 VizLearning) and by the ERC (ERC-2017-CoG-772738 TouchDesign).

References

[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

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

[BMF03] Bridson R., Marino S., Fedkiw R.: Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2003), SCA '03, Eurographics Association, p. 28–36. 2

[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. 2

[BWR*08] Bergou M., Wardetzky M., Robinson S., Audoly B., Grinspun E.: Discrete Elastic Rods. *ACM Transactions on Graphics (SIGGRAPH) 27*, 3 (2008), 63:1–63:12. 2, 3, 4

[CCR*20] Casafranca J. J., Cirio G., Rodríguez A., Miguel E., Otaduy M. A.: Mixing yarns and triangles in cloth simulation. *Computer Graphics Forum 39*, 2 (2020). 2

[CK02] Choi K.-J., Ko H.-S.: Stable but responsive cloth. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2002), SIGGRAPH '02, Association for Computing Machinery, p. 604–611. 2

[CLMMO14] Cirio G., Lopez-Moreno J., Miraut D., Otaduy M. A.: Yarn-level simulation of woven cloth. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH Asia) 33*, 6 (2014). 1, 2, 3, 5, 6, 7

[CLMO17] Cirio G., Lopez-Moreno J., Otaduy M. A.: Yarn-level cloth simulation with sliding persistent contacts. *IEEE Transactions on Visualization and Computer Graphics 23*, 2 (2017), 1152–1162. 2, 3, 6

[DKBP06] Duan Y., Keefe M., Bogetti T. A., Powers B.: Finite element modeling of transverse impact on a ballistic fabric. *International Journal of Mechanical Sciences 48*, 1 (2006), 33–43. 2

[GHDS03] Grinspun E., Hirani A. N., Desbrun M., Schröder

P.: Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Goslar, DEU, 2003), SCA '03, Eurographics Association, p. 62–67. 2

[HGB69] HEARLE J. W. S., GROSBERG P., BACKER S.: *Structural Mechanics of Fibers, Yarns, and Fabrics*, vol. 1. John Wiley & Sons Inc, New York, 1969. 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

[JGT17] JIANG C., GAST T., TERAN J.: Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM TOG 36*, 4 (2017). 3

[KJM08] KALDOR J. M., JAMES D. L., MARSCHNER S.: Simulating knitted cloth at the yarn level. *ACM Trans. Graph. 27*, 3 (2008), 65:1–65:9. 1, 2, 3, 6

[KJM10] KALDOR J. M., JAMES D. L., MARSCHNER S.: Efficient yarn-based cloth with adaptive contact linearization. *ACM Transactions on Graphics 29*, 4 (July 2010), 105:1–105:10. 2, 3

[KNK73] KAWABATA S., NIWA M., KAWAI H.: The finite-deformation theory of plain-weave fabrics part i: The biaxial-deformation theory. *Journal of the Textile Institute 64*, 1 (1973), 21–46. 2

[LWS*18] LEAF J., WU R., SCHWEICKART E., JAMES D. L., MARSCHNER S.: Interactive design of periodic yarn-level cloth patterns. *ACM Trans. Graph. 37*, 6 (2018). 2

[MBCM16] MÜLLER M., BENDER J., CHENTANEZ N., MACKLIN M.: A robust method to extract the rotational part of deformations. In *Proceedings of the 9th International Conference on Motion in Games* (New York, NY, USA, 2016), MIG '16, ACM, pp. 55–60. 5

[MCM03] MCGLOCKTON M. A., COX B. N., MCMEEKING R. M.: A binary model of textile composites: III high failure strain and work of fracture in 3D weaves. *Journal of the Mechanics and Physics of Solids 51*, 8 (2003), 1573–1600. 2

[MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. *ACM Trans. Graph. 24*, 3 (2005), 471–478. 5

[NPS06] NADLER B., PAPADOPOULOS P., STEIGMANN D. J.: Multiscale constitutive modeling and numerical simulation of fabric material. *International Journal of Solids and Structures 43*, 2 (2006), 206 – 221. 2

[NTL98] NG S.-P., TSE P.-C., LAU K.-J.: Numerical and experimental determination of in-plane elastic properties of 2/2 twill weave fabric composites. *Composites Part B: Engineering 29*, 6 (1998), 735–744. 2

[Pai02] PAI D. K.: Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum 21* (2002). 2

[PFF37] PEIRCE F. T., F.INST.P., F.T.I.: The geometry of cloth structure. *Journal of the Textile Institute Transactions 28*, 3 (1937), T45–T96. 2

[Pro95] PROVOT X.: Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Proceedings of Graphics Interface '95* (Toronto, Ontario, Canada, 1995), GI '95, Canadian Human-Computer Communications Society, pp. 147–154. 2

[PTC*15] PEREZ J., THOMASZEWSKI B., COROS S., BICKEL B., CANABAL J. A., SUMNER R., OTADUY M. A.: Design and fabrication of flexible rod meshes. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH) 34*, 4 (2015). 2, 3, 4, 5, 7

[PW00] PAGE J., WANG J.: Prediction of shear force and an analysis of yarn slippage for a plain-weave carbon fabric in a bias extension state. *Composites Science and Technology 60*, 7 (2000), 977 – 986. 2

[RCTI91] ROSENBLUM R. E., CARLSON W. E., TRIPP III E.: Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation 2*, 4 (1991), 141–148. 2

[Ree03] REESE S.: Anisotropic elastoplastic material behavior in fabric structures. In *IUTAM Symposium on Computational Mechanics of Solid Materials at Large Strains*, Miehe C., (Ed.), no. 108 in Solid Mechanics and Its Applications. Springer Netherlands, 2003, pp. 201–210. 2

[SBO18] SÁNCHEZ-BANDERAS R. M., OTADUY M. A.: Strain rate dissipation for elastic deformations. *Computer Graphics Forum 37*, 8 (2018), 161–170. 2

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

[SLF08] SELLE A., LENTINE M., FEDKIW R.: A mass spring model for hair simulation. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, Association for Computing Machinery. 2

[SNW20] SPERL G., NARAIN R., WOJTAN C.: Homogenized yarn-level cloth. *ACM Transactions on Graphics (TOG) 39*, 4 (2020). 2

[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* (Goslar, DEU, 2007), SCA '07, Eurographics Association, p. 63–72. 2

[ST09] SPILLMANN J., TESCHNER M.: Cosserat nets. *IEEE transactions on visualization and computer graphics 15* (05 2009), 325–38. 2

[YKJM12] YUKSEL C., KALDOR J. M., JAMES D. L., MARSCHNER S.: Stitch meshes for modeling knitted clothing with yarn-level detail. *ACM Trans. Graph. 31*, 4 (2012), 37:1–37:12. 2

[ZCT16] ZEHNDER J., COROS S., THOMASZEWSKI B.: Designing structurally-sound ornamental curve networks. *ACM Transactions on Graphics 35* (2016), 1–10. 2

Appendix A: Derivatives of Axis Angle

Given two vectors $\bar{\mathbf{t}}$ and $\mathbf{R}^T\,\mathbf{t}$, one can compute the axis-angle vector representation (1) of their smallest relative rotation using their cross and dot products, c and d respectively:

$$\begin{aligned}\phi &= \arctan\left(\frac{\|c\|}{d}\right)\frac{c}{\|c\|}, \\ &\text{with } c = \bar{\mathbf{t}}\times\left(\mathbf{R}^T\mathbf{t}\right) \;\text{ and }\; d = \bar{\mathbf{t}}^T\mathbf{R}^T\mathbf{t}.\end{aligned} \tag{12}$$

Our bending model requires the computation of Jacobians of this axis angle w.r.t. an incremental change θ of the rotation $\mathbf{R}$, and w.r.t. the nodal positions $\mathbf{x}$. We can express a generic differential of the axis angle as

$$\delta\phi = \left(\left(\frac{d\,c}{d^2+c^T c}-\phi\right)\frac{c^T}{c^T c}+\frac{\phi^T c}{c^T c}\mathbf{I}\right)\delta c - \frac{c}{d^2+c^T c}\,\delta d. \tag{13}$$

This differential is not robust for small axis angles ($\lim_{\phi\to 0}$). In that case, we substitute in (12) the approximation $\lim_{\phi\to 0}\frac{\tan\phi}{\phi} = 1$, which yields:

$$\lim_{\phi\to 0}\phi = \frac{c}{d}, \qquad \lim_{\phi\to 0}\delta\phi = \frac{1}{d}\left(\delta c - \phi\,\delta d\right). \tag{14}$$

The necessary Jacobians of the cross and dot products can be computed as

$$\frac{\partial c}{\partial\theta} = \operatorname{skew}(\bar{\mathbf{t}})\,\mathbf{R}^T\operatorname{skew}(\mathbf{t}), \qquad \frac{\partial c}{\partial\mathbf{x}} = \operatorname{skew}(\bar{\mathbf{t}})\,\mathbf{R}^T\,\frac{\partial\mathbf{t}}{\partial\mathbf{x}}, \tag{15}$$

$$\frac{\partial d}{\partial\theta} = \bar{\mathbf{t}}^T\,\mathbf{R}^T\operatorname{skew}(\mathbf{t}), \qquad \frac{\partial d}{\partial\mathbf{x}} = \bar{\mathbf{t}}^T\,\mathbf{R}^T\,\frac{\partial\mathbf{t}}{\partial\mathbf{x}}. \tag{16}$$