An Eigenanalysis of Angle-Based Deformation Energies

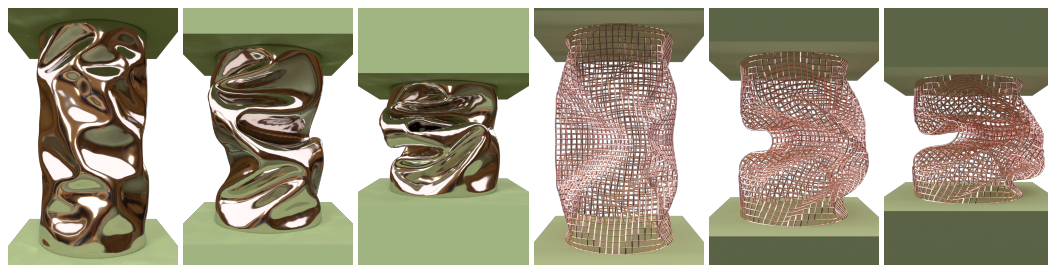HAOMIAO WU and THEODORE KIM, Yale University, USA

Fig. 1. Using our analysis, we simulate stiff rods and shells at large time steps. We crushed a metal shell and a wire net at a time step of $dt = 1/30$ s. The bending stiffness of the shell is $\mu = 1e^6$, and the Young's modulus of the wires is $E = 1e^{12}$ Pa. Our method captures detailed buckling behaviors under large bending stiffness while maintaining stability and efficiency.

Angle-based energies appear in numerous physics-based simulation models, including thin-shell bending and isotropic elastic strands. We present a generic analysis of these energies that allows us to analytically filter the negative eigenvalues of the second derivative (Hessian), which is critical for stable, implicit time integration. While these energies are usually formulated in terms of angles and positions, we propose an abstract edge stencil that succinctly parameterizes the edge deformation, and allows us to derive generic, closed-form analytical expressions for the energy eigensystems. The resultant eigenvectors have straightforward geometric interpretations. We demonstrate that our method is readily applicable to a variety of 2D and 3D angle-based elastic energies, including both cloth and strands, and is up to 7× faster than numerical eigendecomposition.

1 INTRODUCTION

Many elastic energies used for deformable objects are based on angles, which often exhibit complex nonlinear relationships to vertex positions, and incorporate geometric quantities in a reduced space. This makes the energies difficult to analyze, and in particular it becomes unclear when the Hessians associated with their force gradients become indefinite. This is unfortunate, because many implicit solvers [Baraff and Witkin 1998] require the underlying system to be semi-positive definite (SPD) in order to guarantee convergence.

Previous works have produced SPD systems by using Gauss-Newton approximation [Choi and Ko 2002] or by explicitly filtering the Hessian at each element [Teran et al. 2005]. Recent methods efficiently addressed this issue by constructing analytical eigendecompositions at each element.

Authors' address: Haomiao Wu, haomiao.wu@yale.edu; Theodore Kim, theodore.kim@yale.edu, Yale University, New Haven, Connecticut, USA, 06511-8937.

While this approach has been applied to hyperelastic volume [Kim et al. 2019; Smith et al. 2019] and membrane [Kim 2020] energies, no equivalent analysis has ever been devised for angle-based energies, in part because previous analyses rely on deformation invariants that do not generalize to angle-based energies.

In this paper, we address this challenge by devising a new low-dimensional geometric representation that allows us to analyze the eigensystems of this class of energies in closed form. Our analysis applies to a variety of different energies in both 2D and 3D. We first make use of an edge deformation matrix reminiscent of the deformation gradient for thin shells, which then allows us to derive generic, compact expressions for the energy gradient and Hessian. From there, we transform into a sparse, canonical coordinate system that decouples in-plane and out-of-plane movements. Within these reduced dimensions, we are then able to derive an analytical eigendecomposition.

The analytic expressions allow us to perform an in-depth investigation of angle-based energies. We are able to construct a geometric picture for each eigenvector, and directly examine the conditions under which indefiniteness arises. Finally, we show that our derivation can be applied to a variety of angle-based energies, including the bending energy of isotropic strands and thin shells, and results in improved speed and robustness. To summarize our contributions:

- A generic formulation for angle-based energies based on an abstract, two-edge stencil.
- An analytical eigensystem of generalized, angle-based energies.
- An intuitive, geometric interpretation of the eigenvectors.
- An application of our analysis to strand and cloth energies.

2 RELATED WORKS

2.1 Strand Models

Since at least the 1990s, computer graphics papers have used angle-based models for the simulation of strands and hair [Anjyo et al. 1992] by employing multi-body [Hadap 2006], "angular" [Bourguignon and Cani 2000] or "bending" formulations [Daldegan et al. 1993; Ward and Lin 2003]. Some works [Iben et al. 2013; Selle et al. 2008] have approximated these angular forces with linear springs, but these are known to insert non-physical tangential forces that are often undesirable. Implementing angular energies is more involved than linear springs, but by laying out generic eigensystems here, we hope to reduce this complexity for future practitioners.

Cosserat-based strand models also use angle-based bending and twisting [Pai 2002], and have been extended to include anisotropic cross-sections [Bergou et al. 2010]. In this paper, we focus on isotropic cross-sections, which for the aforementioned models reduce to half-angle tangent energies [Gingold et al. 2004]. Quadratic variants have also been useful for modelling constrained strand assemblies [Sueda et al. 2011], which in particular includes knitted cloth [Cirio et al. 2015; Sánchez-Banderas et al. 2020]. Angle-based strand formulations have also been applied in fabrication [Pérez et al. 2015] and architecture [Panetta et al. 2019a]. Finally, concurrent recent work found an analytic eigensystem for *unitary* bending energies in the course of formulating a new strand model [Wu et al. 2023], but we instead analyze the fully general case here.

2.2 Shell Models

Similar to the case of strands, some works approximate angle-based bending energies with linear springs [Bridson et al. 2002; Choi and Ko 2002; Provot et al. 1995], but these again inject tangential forces into the simulation that introduce artificial stiffness in undesired directions.

Angle-based models have also been used for cloth and shell simulation since at least the 1990s [Breen et al. 1994; Volino et al. 1995]. Many different formulations have been proposed, including quadratic [Baraff and Witkin 1998; Grinspun et al. 2003], cos-based [Bridson et al. 2003], sin-based

[Wardetzky et al. 2007], and tan-based [Gingold et al. 2004] models. Deriving the derivatives of the bending angle, especially the Hessian required by implicit integrators, can be cumbersome, so Tamstorf and Grinspun [2013] provide explicit expressions as well as architectural recommendations for efficient performance. In their supplement, they provide some intuition regarding sources of indefiniteness, and describe the rank-4 nullspace of the Ψ Hessian. Our analysis confirms their nullspace, but goes on uncover closed-form expressions for all the non-trivial eigenpairs.

3 ANGLE-BASED EIGENANALYSIS

In the following, we will denote matrices with bold uppercase ($\mathbf{A}$), vectors with bold lowercase ($\mathbf{a}$), and scalars with unbolded lowercase (a). We use vec($\cdot$) to denote tensor vectorization [Golub and Van Loan 2013], and adopt the convention that a vectorized matrix is denoted with a lowercase version of the same symbol, e.g. vec($\mathbf{A}$) = $\mathbf{a}$.

3.1 Abstract Edge Formulation

To begin, we assume that the energy is computed based on angles over some discrete object, and that each angle is formed by two abstract edges, $\mathbf{e}_0$ and $\mathbf{e}_1$ (Fig. 2). This can correspond to two adjacent edges in a strand, or two edges along one "flap" of a shell. Once our generic analysis is complete, we will show how to apply our results to these specific domains in §4.

We define an angle-based elastic energy $\Psi(\theta)$, where θ is the angle between the two edge vectors:

$$\theta = \mathcal{S} \cdot \text{acos}\left(\frac{\mathbf{e}_0^\top \mathbf{e}_1}{\|\mathbf{e}_0\|\|\mathbf{e}_1\|}\right) \tag{1}$$

The range of θ should be $[-\pi, \pi]$ to fully describe all rotational configurations, but the range of acos is $[0, \pi]$ and the sign of θ is ambiguous. To resolve this ambiguity, domain-specific information such as the direction of the bending edge is needed. We abstract this auxiliary geometric information into a sign function $\mathcal{S}$. Under differentiation, $\mathcal{S}$ becomes a constant function, aside from a point singularity at 0 that can be addressed using a simple numerical guard. Thus, we can treat $\mathcal{S}$ as a constant that does not participate in the ensuing analysis.

Without loss of generality, we define a deformation gradient for our edge stencils:

$$\mathbf{F} = \begin{bmatrix} \mathbf{e}_0 \mid \mathbf{e}_1 \end{bmatrix}. \tag{2}$$

Like the deformation gradient, $\mathbf{F}$ fully describes the scaling and rotation of the stencil related to θ, but subtracts off rigid body translation modes.

Deformation gradients are usually defined as the linear map component of an affine transformation [Bonet et al. 2021], which maps the rest state to current state. For example, the deformation gradient for membrane stretching can be computed by multiplying the current edge matrix with the inverse of the material space coordinate matrix [Kim 2020]. In the case of a 1D edge stencil, the material coordinates become constant scaling factors that will self-cancel in Eqn. 3, so we can omit them here. Therefore, we can assemble the deformation gradient of our abstract stencil using only the current state edges. We can write the derivatives of $\Psi(\theta)$ in terms of $\mathbf{F}$

$$\begin{cases} \frac{\partial \Psi}{\partial \mathbf{x}} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right) \frac{\partial \Psi}{\partial \mathbf{F}} \\ \frac{\partial^2 \Psi}{\partial \mathbf{x}^2} = \left(\frac{\partial \Psi}{\partial \mathbf{F}}\right)^\top \frac{\partial^2 \mathbf{F}}{\partial \mathbf{x}^2} + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^\top \frac{\partial^2 \Psi}{\partial \mathbf{F}^2} \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right) \end{cases} \tag{3}$$

where the Jacobian $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ takes on different forms for strands and shells. We will later show in §4 that $\frac{\partial^2 \mathbf{F}}{\partial \mathbf{x}^2}$ either resolves to zero or is of low rank. Thus, analyzing the Hessian $\frac{\partial^2 \Psi}{\partial \mathbf{F}^2}$ becomes the most significant task.
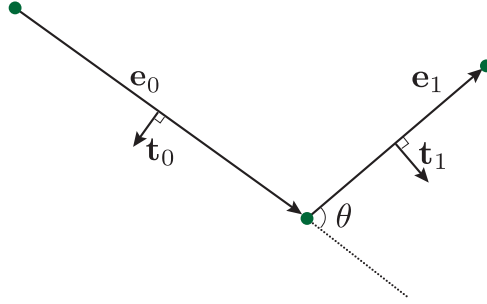
Fig. 2. The two-edge stencil that forms θ. The unit orthogonal vectors $\mathbf{t}_{0\perp}$ and $\mathbf{t}_{1\perp}$ together with the edges form a local canonical coordinate system.

3.2 2D Eigenanalysis

We start with the 2D case of $\mathbf{e}_0, \mathbf{e}_1 \in \mathbb{R}^2$ and $\mathbf{F} \in \mathbb{R}^2$. The results of this 2D analysis will then serve as a building block for 3D. We assume the form of $\Psi(\theta)$ can be arbitrary, so we first write the derivatives of the intermediate variable θ.

In $\mathbb{R}^2$, the sign of θ can be assigned with respect to a canonical z-axis. In particular, we can define $\mathcal{S} = \mathrm{sgn}(e_{00}e_{11} - e_{01}e_{10})$ where e_{ij} is the jth component of $\mathbf{e}_i$. To avoid dealing with higher order notations, we will write the analysis in terms of the vectorized deformation gradient $\mathbf{f} = \mathrm{vec}(\mathbf{F})$. The gradient and Hessian of θ in terms of the deformation gradient can be written as the blocks

$$\frac{\partial \theta}{\partial \mathbf{e}_0} = \mathcal{S} \cdot \frac{\mathbf{t}_{0\perp}}{\|\mathbf{e}_0\|} = \mathbf{g}_0 \qquad\qquad \frac{\partial \theta}{\partial \mathbf{e}_1} = -\mathcal{S} \cdot \frac{\mathbf{t}_{1\perp}}{\|\mathbf{e}_1\|} = \mathbf{g}_1 \tag{4}$$

$$\frac{\partial^2 \theta}{\partial \mathbf{e}_0^2} = \mathcal{S} \frac{\mathbf{C} - 2\mathbf{t}_{0\perp}\mathbf{t}_0^\top}{\|\mathbf{e}_0\|^2} = \mathbf{D}_0 \qquad\qquad \frac{\partial^2 \theta}{\partial \mathbf{e}_1^2} = \mathcal{S} \frac{-\mathbf{C} + 2\mathbf{t}_{1\perp}\mathbf{t}_1^\top}{\|\mathbf{e}_1\|^2} = \mathbf{D}_1 \tag{5}$$

$$\frac{\partial \theta}{\partial \mathbf{f}} = \begin{bmatrix} \mathbf{g}_0^\top & \mathbf{g}_1^\top \end{bmatrix} \qquad\qquad \frac{\partial^2 \theta}{\partial \mathbf{f}^2} = \begin{bmatrix} \mathbf{D}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_1 \end{bmatrix} \tag{6}$$

where $\mathbf{t}_i = \mathbf{e}_i/\|\mathbf{e}_i\|$ is a normalized version of each edge and $\mathbf{C} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ is a $\pi/2$ clockwise rotation (Fig. 2). The $\perp$ subscript is added to a vector whenever it is multiplied by $\mathbf{C}$. e.g. $\mathbf{C}\mathbf{t}_0 = \mathbf{t}_{0\perp}$. The gradient and Hessian of $\Psi(\theta)$ can thus be written:

$$g = \frac{\partial \Psi}{\partial \theta} \qquad h = \frac{\partial^2 \Psi}{\partial \theta^2} \qquad \frac{\partial \Psi}{\partial \mathbf{f}} = g \begin{bmatrix} \mathbf{g}_0^\top & \mathbf{g}_1^\top \end{bmatrix}^\top \tag{7}$$

$$\frac{\partial^2 \Psi}{\partial \mathbf{f}^2} = \mathbf{H}_2 = h \begin{bmatrix} \mathbf{g}_0\mathbf{g}_0^\top & \mathbf{g}_0\mathbf{g}_1^\top \\ \mathbf{g}_1\mathbf{g}_0^\top & \mathbf{g}_1\mathbf{g}_1^\top \end{bmatrix} + g \begin{bmatrix} \mathbf{D}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_1 \end{bmatrix} \tag{8}$$

where the subscript on $\mathbf{H}_2$ indicates that it is for the 2D case. To reveal the eigenstructure of $\mathbf{H}_2 \in \mathbb{R}^{4\times 4}$, we observe it has properties relative to the edges and their orthogonal vectors:

$$\frac{\partial^2 \Psi}{\partial \mathbf{f}^2} \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{0} \end{bmatrix} = g\left(-\frac{\mathcal{S}}{\|\mathbf{e}_0\|^2}\right) \begin{bmatrix} \mathbf{e}_{0\perp} \\ \mathbf{0} \end{bmatrix} \tag{9}$$

$$\frac{\partial^2 \Psi}{\partial \mathbf{f}^2} \begin{bmatrix} \mathbf{e}_{0\perp} \\ \mathbf{0} \end{bmatrix} = h \begin{bmatrix} \frac{\mathbf{e}_{0\perp}}{\|\mathbf{e}_0\|^2} \\ -\frac{\mathbf{e}_{1\perp}}{\|\mathbf{e}_1\|^2} \end{bmatrix} + g\left(-\frac{\mathcal{S}}{\|\mathbf{e}_0\|^2}\right) \begin{bmatrix} \mathbf{e}_0 \\ \mathbf{0} \end{bmatrix} \tag{10}$$

$$\frac{\partial^2 \Psi}{\partial \mathbf{f}^2} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_1 \end{bmatrix} = g\left(\frac{\mathcal{S}}{\|\mathbf{e}_1\|^2}\right) \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_{1\perp} \end{bmatrix} \tag{11}$$

$$\frac{\partial^2 \Psi}{\partial \mathbf{f}^2} \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_{1\perp} \end{bmatrix} = h \begin{bmatrix} -\frac{\mathbf{e}_{0\perp}}{\|\mathbf{e}_0\|^2} \\ \frac{\mathbf{e}_{1\perp}}{\|\mathbf{e}_1\|^2} \end{bmatrix} + g\left(\frac{\mathcal{S}}{\|\mathbf{e}_1\|^2}\right) \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_1 \end{bmatrix}. \tag{12}$$

This leads us to formulate a change-of-basis based on these vectors

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{e}_0 & \mathbf{e}_{0\perp} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{e}_1 & \mathbf{e}_{1\perp} \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{13}$$

that then yields a sparsified version of $\mathbf{H}_2$

$$\mathbf{A}_2^{-1} \mathbf{H}_2 \mathbf{A}_2 = \begin{bmatrix} 0 & m_0 & 0 & 0 \\ m_0 & l_0 & 0 & -l_0 \\ 0 & 0 & 0 & m_1 \\ 0 & -l_1 & m_1 & l_1 \end{bmatrix} \tag{14}$$

where

$$m_0 = g\left(-\frac{\mathcal{S}}{\|\mathbf{e}_0\|^2}\right), \qquad m_1 = g\left(\frac{\mathcal{S}}{\|\mathbf{e}_1\|^2}\right), \qquad l_0 = \frac{h}{\|\mathbf{e}_0\|^2}, \qquad l_1 = \frac{h}{\|\mathbf{e}_1\|^2}. \tag{15}$$

The eigensystem of a 4×4 matrix can be obtained analytically by solving for the roots of the 4th order characteristic function. Eqn. 14 shows that $\mathbf{H}_2$ can be sparsified, which then yields simpler

root expressions. The eigensystem of $\mathbf{A}_2^{-1}\mathbf{H}_2\mathbf{A}_2$ is then given by:

$$\mathbf{u}_0 = \begin{bmatrix} w\left(\frac{(\gamma-1)^2}{\gamma+1} - (\gamma+1)r - (\gamma-1)R_-\right) \\ -\frac{\gamma}{2}R_-^2 + 2w^2 + \frac{1}{2}\left(\gamma-1-(\gamma+1)r\right)R_- \\ 4\frac{w}{\gamma+1} \\ 1-r-R_- \end{bmatrix} \tag{16}$$

$$\mathbf{u}_1 = \begin{bmatrix} w\left(\frac{(\gamma-1)^2}{\gamma+1} - (\gamma+1)r + (\gamma-1)R_-\right) \\ -\frac{\gamma}{2}R_-^2 + 2w^2 - \frac{1}{2}\left(\gamma-1-(\gamma+1)r\right)R_- \\ 4\frac{w}{\gamma+1} \\ 1-r+R_- \end{bmatrix} \tag{17}$$

$$\mathbf{u}_2 = \begin{bmatrix} w\left(\frac{(\gamma-1)^2}{\gamma+1} + (\gamma+1)r - (\gamma-1)R_+\right) \\ -\frac{\gamma}{2}R_+^2 + 2w^2 + \frac{1}{2}\left(\gamma-1+(\gamma+1)r\right)R_+ \\ 4\frac{w}{\gamma+1} \\ 1+r-R_+ \end{bmatrix} \tag{18}$$

$$\mathbf{u}_3 = \begin{bmatrix} w\left(\frac{(\gamma-1)^2}{\gamma+1} + (\gamma+1)r + (\gamma-1)R_+\right) \\ -\frac{\gamma}{2}R_+^2 + 2w^2 - \frac{1}{2}\left(\gamma-1+(\gamma+1)r\right)R_+ \\ 4\frac{w}{\gamma+1} \\ 1+r+R_+ \end{bmatrix} \tag{19}$$

$$\lambda_0 = \frac{(l_0+l_1)(1-r-R_-)}{4} \qquad\qquad \lambda_1 = \frac{(l_0+l_1)(1-r+R_-)}{4} \tag{20}$$

$$\lambda_2 = \frac{(l_0+l_1)(1+r-R_+)}{4} \qquad\qquad \lambda_3 = \frac{(l_0+l_1)(1+r+R_+)}{4} \tag{21}$$

where:

$$w = \frac{Sg}{h} \qquad\qquad \gamma = \frac{\|\mathbf{e}_1\|^2}{\|\mathbf{e}_0\|^2} \qquad\qquad r = \sqrt{4w^2\left(\frac{\gamma-1}{\gamma+1}\right)^2 + 1} \tag{22}$$

$$R_- = \sqrt{2\left(2w^2+1-r\right)} \qquad\qquad R_+ = \sqrt{2\left(2w^2+1+r\right)} \tag{23}$$

The eigenvectors of $\mathbf{H}_2$ in the original coordinate system can then be retrieved by re-applying $\mathbf{A}_2$

$$\begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \end{bmatrix} = \mathbf{A}_2 \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}, \tag{24}$$

where $\mathbf{q}_*$ are written in unnormalized form. This particular change-of-basis does not alter the spectrum of the eigendecomposition, so the eigenvalue expressions remain the same.

These expressions may look onerous at first, but most of the complexity comes from the edge length scaling factor γ, and they reduce to much simpler forms when $\|\mathbf{e}_0\| = \|\mathbf{e}_1\|$:

$$\mathbf{u}_0 = \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \end{bmatrix} \qquad \mathbf{u}_1 = \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad \mathbf{u}_2 = \begin{bmatrix} 1 \\ \frac{-1+\sqrt{w^2+1}}{w} \\ 1 \\ \frac{1-\sqrt{w^2+1}}{w} \end{bmatrix} \qquad \mathbf{u}_3 = \begin{bmatrix} 1 \\ \frac{-1-\sqrt{w^2+1}}{w} \\ 1 \\ \frac{1+\sqrt{w^2+1}}{w} \end{bmatrix}$$

$$\lambda_0 = -hw/\|\mathbf{e}_0\|^2 \qquad\qquad\qquad \lambda_1 = hw/\|\mathbf{e}_0\|^2$$

$$\lambda_2 = h(1-\sqrt{w^2+1})/\|\mathbf{e}_0\|^2 \qquad\qquad \lambda_3 = h(1+\sqrt{w^2+1})/\|\mathbf{e}_0\|^2$$

For simplicity, in the above expressions we assume that $w > 0$ and removed some scaling factors. More importantly, we will see in §3.4 that they have clear geometric interpretations.

3.3 3D Eigenanalysis

The 3D case is similar to 2D, except that the osculating plane is not static. To track its orientation, we define the binormal $\mathbf{b} = \mathbf{e}_0 \times \mathbf{e}_1$ and its normalized form $\mathbf{t}_b = \mathbf{b}/\|\mathbf{b}\|$. The in-plane orthogonal edge vectors are thus defined as $\mathbf{t}_{i\perp} = \mathbf{t}_i \times \mathbf{t}_b$ for $i \in \{0, 1\}$, where $\mathbf{t}_i$ again denotes a normalized $\mathbf{e}_i$. The gradient and Hessian with respect to the vectorized deformation gradient $\text{vec}(\mathbf{F}) = \mathbf{f} \in \mathbb{R}^6$ are:

$$\frac{\partial \Psi}{\partial \mathbf{f}} = g \begin{bmatrix} \mathbf{g}_0^\top & \mathbf{g}_1^\top \end{bmatrix}^\top \qquad \frac{\partial^2 \Psi}{\partial \mathbf{f}^2} = \mathbf{H} = h \begin{bmatrix} \mathbf{g}_0 \mathbf{g}_0^\top & \mathbf{g}_0 \mathbf{g}_1^\top \\ \mathbf{g}_1 \mathbf{g}_0^\top & \mathbf{g}_1 \mathbf{g}_1^\top \end{bmatrix} + \mathcal{S}g \begin{bmatrix} \mathbf{D}_{00} & \mathbf{D}_{01} \\ \mathbf{D}_{10} & \mathbf{D}_{11} \end{bmatrix} \tag{25}$$

$$\mathbf{D}_{00} = \frac{\mathbf{T}_b - 2\mathbf{t}_{0\perp} \mathbf{t}_0^\top}{\|\mathbf{e}_0\|^2} - \frac{\mathbf{T}_0 \mathbf{T}_1 + \mathbf{t}_{0\perp} \mathbf{t}_{1\perp}^\top}{\|\mathbf{e}_0\|^2 \|\mathbf{t}_0 \times \mathbf{t}_1\|} \qquad \mathbf{D}_{11} = \frac{-\mathbf{T}_b + 2\mathbf{t}_{1\perp} \mathbf{t}_1^\top}{\|\mathbf{e}_1\|^2} - \frac{\mathbf{T}_1 \mathbf{T}_0 + \mathbf{t}_{1\perp} \mathbf{t}_{0\perp}^\top}{\|\mathbf{e}_1\|^2 \|\mathbf{t}_0 \times \mathbf{t}_1\|} \tag{26}$$

$$\mathbf{D}_{01} = \frac{\mathbf{T}_0 \mathbf{T}_0 + \mathbf{t}_{0\perp} \mathbf{t}_{0\perp}^\top}{\|\mathbf{e}_0 \times \mathbf{e}_1\|} \qquad \mathbf{D}_{10} = \frac{\mathbf{T}_1 \mathbf{T}_1 + \mathbf{t}_{1\perp} \mathbf{t}_{1\perp}^\top}{\|\mathbf{e}_0 \times \mathbf{e}_1\|} \tag{27}$$

where $g, h, \mathbf{g}_0, \mathbf{g}_1$ are defined the same as Eqns. 4 and 7. $\mathbf{T}_i \in \mathbb{R}^{3 \times 3}$ where $i \in \{0, 1, b\}$ is defined as the Levi-Civita tensor $\mathcal{E}_{ijk}$ right-multiplied by the vector $\mathbf{t}_i$ such that $\forall a \in \mathbb{R}^3, a \times \mathbf{t}_i = \mathbf{T}_i a$. Relative to Eqns. 5 and 8, the 3D Hessian now includes terms that arise from differentiating $\mathbf{b}$.

Similar to the 2D case, we can multiply the edges and their orthogonal vectors onto $\mathbf{H}$ to get their projections onto the osculating plane. Interestingly, $\mathbf{H}$ displays the exact same characteristics as $\mathbf{H}_2$ in Eqn. 9-12. If we denote the 3D version of $\mathbf{A}_2$ as $\mathbf{A} \in \mathbb{R}^{6 \times 4}$ and defined it analogously to Eqn. 13, we obtain $\mathbf{HA} = \mathbf{AA}_2^\top \mathbf{H}_2 \mathbf{A}_2$.

This implies that $\mathbf{H}$ is block-diagonal in the 2D coordinate system and that the rank-4 in-plane dynamics is linearly independent from the rank-2 out-of-plane dynamics. Thus, we can write the first four eigenvectors of $\mathbf{H}$ in terms of the osculating plane

$$\begin{bmatrix} \mathbf{q}_0 & \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{u}_0 & \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix} \tag{28}$$

where $\mathbf{q}_i \in \mathbb{R}^6$.

The remaining two eigenvectors live in the complementary orthogonal space spanned by the basis of $\mathbf{A}$. If we define $\mathbf{q} = \begin{bmatrix} \alpha \mathbf{t}_b^\top & \mathbf{t}_b^\top \end{bmatrix}^\top$, where α is an arbitrary scalar, then $\mathbf{q}$ is an eigenvector of $\mathbf{H}$ if and only if for some scalar λ, the following holds:

$$\mathbf{Hq} = \frac{g}{\sin \theta} \begin{bmatrix} \left(\frac{\alpha \cos \theta}{\|\mathbf{e}_0\|^2} - \frac{1}{\|\mathbf{e}_0\| \|\mathbf{e}_1\|} \right) \mathbf{t}_b \\ \left(\frac{\cos \theta}{\|\mathbf{e}_1\|^2} - \frac{\alpha}{\|\mathbf{e}_0\| \|\mathbf{e}_1\|} \right) \mathbf{t}_b \end{bmatrix} = \lambda \begin{bmatrix} \alpha \mathbf{t}_b \\ \mathbf{t}_b \end{bmatrix} \tag{29}$$

Solving for α and λ, we obtain the last two eigenpairs:

$$\mathbf{q}_4 = \frac{1}{\sqrt{\alpha_0^2 + 1}} \begin{bmatrix} \alpha_0 \mathbf{t}_b \\ \mathbf{t}_b \end{bmatrix} \qquad \lambda_4 = \frac{g}{\sin \theta} \left(\frac{\cos \theta}{\|\mathbf{e}_1\|^2} - \frac{\alpha}{\|\mathbf{e}_0\| \|\mathbf{e}_1\|} \right) \tag{30}$$

$$\mathbf{q}_5 = \frac{1}{\sqrt{\alpha_0^2 + 1}} \begin{bmatrix} \mathbf{t}_b \\ -\alpha_0 \mathbf{t}_b \end{bmatrix} \qquad \lambda_5 = \frac{g}{\sin \theta} \left(\frac{\cos \theta}{\|\mathbf{e}_1\|^2} + \frac{1}{\alpha \|\mathbf{e}_0\| \|\mathbf{e}_1\|} \right) \tag{31}$$

where

$$\alpha_0 = \frac{-\beta + \sqrt{\beta^2 + 4}}{2} \qquad \beta = \left(\frac{\|\mathbf{e}_1\|}{\|\mathbf{e}_0\|} - \frac{\|\mathbf{e}_0\|}{\|\mathbf{e}_1\|} \right) \cos \theta \tag{32}$$

Similarly, the complexity here is also dominated by the scaling.

This concludes our eigenanalysis of $\Psi(\theta)$ for our abstract stencil. Compared to previous position-based formulations [Tamstorf and Grinspun 2013], our deformation gradient-based formulation and the subsequent transformation into a canonical coordinate system reveals the analytic structure of the Hessian. As our analysis was performed generically on θ, it can be applied to any isotropic angle-based energy.

3.4 Eigenmode Interpretation

The Hessian can be viewed as the Jacobian $\mathbf{H} = \frac{\partial \mathbf{g}}{\partial \mathbf{f}}$, where $\mathbf{g} = \begin{bmatrix} \mathbf{g}_0^\top & \mathbf{g}_1^\top \end{bmatrix}^\top$ is the vector of gradients, and essentially maps the differential $d\mathbf{f} = \begin{bmatrix} d\mathbf{e}_0^\top & d\mathbf{e}_1^\top \end{bmatrix}^\top$ to $d\mathbf{g}$. The eigenvectors then represents directions along which the change in gradient aligns with the motion, i.e. $d\mathbf{g} = \lambda d\mathbf{f}$. We refer to these motions as eigenmodes.

By examining the signs of the entries in the eigenvectors, we can view the geometry of each eigenmodes when $w > 0$ in Figure 3. The top two rows illustrate the in-plane motion, where the first row shows a rotation of the stencil ($\mathbf{q}_0$) and a scaling of the edge lengths ($\mathbf{q}_1$). The second row shows a compression ($\mathbf{q}_2$) and expansion ($\mathbf{q}_3$) of the hinge angle. The last row shows out-of-plane rotations about an axis in the osculating plane corresponding to $\mathbf{q}_4$ and $\mathbf{q}_5$.
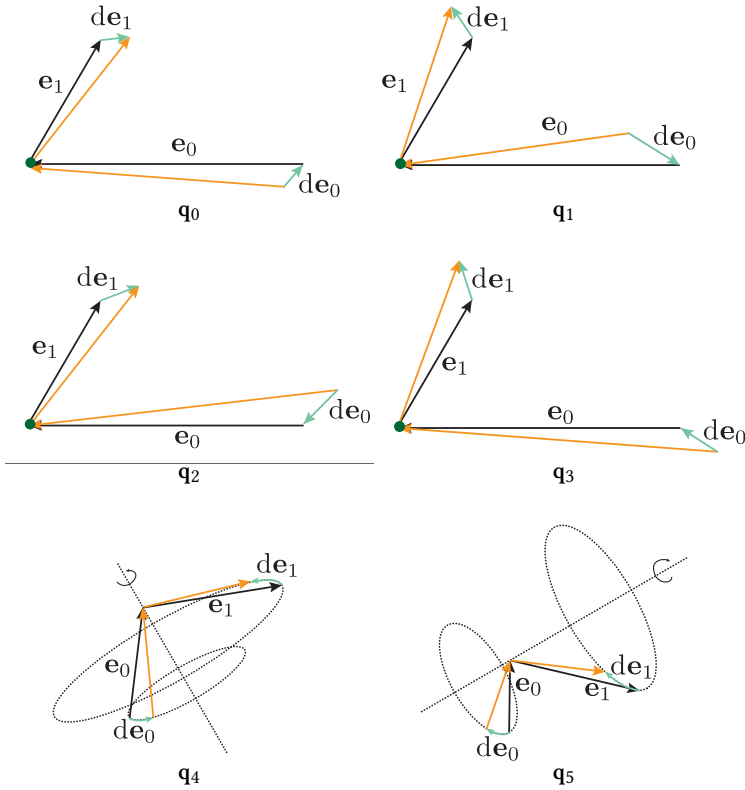

Fig. 3. Geometry of the eigenvectors. The black vectors are current edges, light green are differentials corresponding to the eigenvectors, and orange the edges $\mathbf{e}_* + d\mathbf{e}_*$ after deforming along the eigenmode. The first four represents in-plane motion, and the last row describes out-of-plane rotation.

With geometric interpretations of the eigenvectors established, we can now investigate the signs of the eigenvalues. We first examine $\lambda_{0,1,2,3}$ from Eqns. 20 and 21 which depend on $r \in [0, \sqrt{1 + 4w^2})$. By examining the monotonicity of the eigenvalues in terms of r, we can see that the signs of the eigenvalues depend on the sign of h. When $h > 0$, λ_0 and λ_2 are non-positive and λ_1 and λ_3 are non-negative. When $h < 0$ the signs switch. Thus, $\mathbf{H}$ always contains at least a rank-2 negative-semi-definite subspace.

Taking the indefinite mode $\mathbf{q}_2$ as an example, when the stencil is compressed, there should be increased force to resist the compression. Since the force is scaled with the reciprocal of the edge length, and the edges are stretched, it will be attenuated due to the scaling. Thus, the change in forces will encourage this kind of motion and make the stencil deviate further from the rest state, causing instability.

For the out-of-plane eigenpairs, the signs of λ_4 and λ_5 are determined by $\frac{g}{\sin \theta}$. If $\frac{g}{\sin \theta} < 0$, then λ_4 is negative and λ_5 is positive, while when $\frac{g}{\sin \theta} > 0$, the signs switch. This means that only one of λ_4 and λ_5 can be negative at a time, depending on the value of θ. Thus, the out-of-plane subspace of $\mathbf{H}$ always contains at least one negative eigenvalue, which corresponds to a small torque inducing a sudden out-of-plane rotation.

Using the above analysis, we can assemble an SPD system by ignoring all negative eigenpairs and assembling the Hessian using the outer products of the positive eigenpairs: $\mathbf{H}_{\text{SPD}} = \sum_{\lambda_i > 0} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top$.

4 APPLICATION TO RODS AND SHELLS

We now apply our eigenanalysis to different discrete meshes and elastic energies. Our analysis assumes that the energy model being examined depends on an angle, which is computed with respect to two vectors. In order to apply our analysis to a specific energy, we must define a deformation gradient (Eqn. 2) such that an angle is computed using its columns (Eqn. 1).

More specifically, we need to assign our abstract variables $\mathbf{e}_0$, $\mathbf{t}_1$, $\Psi(\theta)$ (including its derived quantities g and h) and $\mathcal{S}$ to corresponding variables in a concrete discrete model, such that $\mathbf{e}_0$, $\mathbf{e}_1$ and θ satisfy Eqn. 1. To integrate our analysis into an actual solver, we will then need the derivatives of the deformation gradient with respect to vertex positions.
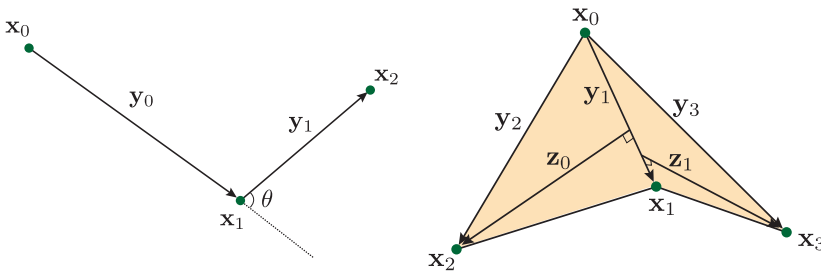

Fig. 4. Stencils for isotropic strand bending (left) and shell bending (right).

4.1 Elastic Rod Bending Energies

The most straightforward application of our analysis is to the bending energy of elastic rods. This energy involves a stencil with three consecutive vertices (Fig. 4, left) that already closely corresponds to our abstract stencil. In the isotropic case of the Discrete Elastic Rods model [Bergou

et al. 2010], the curvature-based energy can be written as an angle-based energy:

$$\Psi_{\text{DER}}(\theta) = \frac{\mu}{\bar{l}}\left(\tan\frac{\theta}{2} - \tan\frac{\theta_r}{2}\right)^2 \quad g = \frac{\mu}{\bar{l}}\cdot\frac{\tan\left(\frac{\theta}{2}\right) - \tan\left(\frac{\theta_r}{2}\right)}{\cos^2\left(\frac{\theta}{2}\right)} \quad h = \frac{\mu}{\bar{l}}\cdot\frac{2 - \cos\theta - \tan\left(\frac{\theta_r}{2}\right)\sin\theta}{2\cos^4\left(\frac{\theta}{2}\right)}.$$

(33)

Other models are instead quadratic in the bending angle [Cirio et al. 2014],

$$\Psi_{\text{bend}} = \frac{\mu}{2}(\theta - \theta_r)^2 \qquad\qquad g = \mu(\theta - \theta_r) \qquad\qquad h = \mu \qquad (34)$$

where μ denotes the stiffness and $\bar{l}$ is the average rest edge length.

For these models, we can compute $\theta = \mathcal{S}\cdot\text{acos}\left(\frac{\mathbf{y}_0^\top \mathbf{y}_1}{\|\mathbf{y}_0\|\|\mathbf{y}_1\|}\right)$. In general, $\mathcal{S}$ cannot be determined independent of the twisting, so for simplicity we set $\mathcal{S} = 1$. We assign the abstract edges to $\mathbf{e}_0 = \mathbf{y}_0, \mathbf{e}_1 = \mathbf{y}_1$ from Fig. 4 for this particular model, which yields the deformation gradient:

$$\mathbf{F} = \begin{bmatrix} \mathbf{y}_0 \mid \mathbf{y}_1 \end{bmatrix}. \qquad (35)$$

Then we have the Jacobian of the vectorized $\mathbf{F}$:

$$\frac{\partial\mathbf{f}}{\partial\mathbf{x}} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_3 & \mathbf{I}_3 \end{bmatrix} \in \mathbb{R}^{6\times9} \qquad (36)$$

where $\mathbf{I}_3$ is the 3×3 identity matrix and $\mathbf{x} = \begin{bmatrix} \mathbf{x}_0^\top & \mathbf{x}_1^\top & \mathbf{x}_2^\top \end{bmatrix}^\top$.

In this case, it becomes clear that $\frac{\partial^2\mathbf{F}}{\partial\mathbf{x}^2}$ in Eqn. 3 resolves to zero. Thus, projecting $\frac{\partial^2\Psi}{\partial\mathbf{F}^2}$ to be PSD is sufficient to guarantee that $\frac{\partial^2\Psi}{\partial\mathbf{x}^2}$ is also PSD. With $\mathbf{e}_0, \mathbf{e}_1, g, h,$ and $\mathcal{S}$ all assigned, we can directly apply the eigenanalysis from §3 to rod bending energies. Specifically, we need to assign these values and vectors to the change-of-basis matrix from Eqn. 13, the coefficients w and γ from Eqn. 22, and the out-of-plane rotation eigenpairs from Eqns. 30-32. After obtaining the $\mathbf{F}$-based eigensystem, plugging the Jacobian of the deformation gradient into Eqn. 3 will generate forces and force gradients for any FEM-based solver.

4.2 Shell Bending Energies

Many shell models define their bending energy using a hinge angle, and our abstract stencil can compactly characterize many of the different $\Psi_{\text{cloth}}(\theta)$ proposed in previous papers. Here we list the derivatives of some common bending energy models with respect to θ.

The quadratic Discrete Shells model [Grinspun et al. 2003] becomes:

$$\Psi_{\text{quad}} = \frac{\mu}{2}(\theta - \theta_r)^2 \qquad\qquad g = \mu(\theta - \theta_r) \qquad\qquad h = \mu. \qquad (37)$$

The cos-based model from [Bridson et al. 2003] resolves to:

$$\Psi_{\cos} = \mu\left(\cos\left(\frac{\theta}{2}\right) - b\theta\right) \qquad g = -\mu\left(\frac{1}{2}\sin\left(\frac{\theta}{2}\right) + b\right) \qquad h = -\frac{\mu}{4}\cos\left(\frac{\theta}{2}\right). \qquad (38)$$

The sin-based energy from [Wardetzky et al. 2007] becomes:

$$\Psi_{\sin} = \mu\left(\sin\left(\frac{\theta}{2}\right)\right)^2 \qquad\qquad g = \frac{\mu}{2}\sin\theta \qquad\qquad h = \frac{\mu}{2}\cos\theta \qquad (39)$$

where the hinge angle θ is defined as the bending angle formed by two neighboring triangles with a shared "hinge" edge. The right side of Fig. 4 shows a hinge stencil on a triangle mesh surface,

where the three edges characterizing the stencil are

$$\mathbf{y}_i = \mathbf{x}_i - \mathbf{x}_0 \qquad\qquad \mathbf{t}_i = \frac{\mathbf{y}_i}{\|\mathbf{y}_i\|} \qquad\qquad i \in \{1, 2, 3\}. \tag{40}$$

In other works, the hinge angle is computed using the normal vectors of the two triangles (see e.g. [Tamstorf and Grinspun 2013]). However, to more closely mirror our abstract stencil, we compute the hinge angle as $\theta = \mathcal{S}\left(\pi - \mathrm{acos}\left(\frac{\mathbf{z}_0 \cdot \mathbf{z}_1}{\|\mathbf{z}_0\|\|\mathbf{z}_1\|}\right)\right)$, where

$$\mathbf{z}_0 = \mathbf{y}_2 - \mathbf{t}_1 \mathbf{t}_1^\top \mathbf{y}_2 \qquad\qquad \mathbf{z}_1 = \mathbf{y}_3 - \mathbf{t}_1 \mathbf{t}_1^\top \mathbf{y}_3 \tag{41}$$

are versions of $\mathbf{y}_2$ and $\mathbf{y}_3$ projected to the orthogonal plane of $\mathbf{y}_1$. This is equivalent to the normal-based definition, but also captures the stencil scaling, which plays a significant role under differentiation. Thus, we assign the abstract edge vectors as $\mathbf{e}_0 = \mathbf{z}_0$ and $\mathbf{e}_1 = \mathbf{z}_1$, and the deformation gradient then becomes

$$\mathbf{F} = \begin{bmatrix} \mathbf{z}_0 & \mathbf{z}_1 \end{bmatrix}. \tag{42}$$

The unit binormal can then be defined as $\mathbf{t}_b = \frac{\mathbf{z}_0 \times \mathbf{z}_1}{\|\mathbf{z}_0 \times \mathbf{z}_1\|}$, which is parallel to the hinge edge $\mathbf{y}_1$. We can then assign $\mathcal{S}$ to a comparison of the binormal vector to the hinge vector: $\mathcal{S} = \mathbf{t}_b \cdot \mathbf{t}_1$.

At this point, we have already assigned $\mathbf{e}_0, \mathbf{e}_1, \mathcal{S}, g$ and h so that we have everything we need to assemble the eigensystem of the Hessian. We can substitute $\mathbf{t}_0$ and $\mathbf{t}_1$ from Eqn. 4 with $\tau_0 = \mathbf{z}_0/\|\mathbf{z}_0\|$ and $\tau_1 = \mathbf{z}_1/\|\mathbf{z}_1\|$ to get the $\mathbf{F}$-based gradient. Now we have everything we need to assemble a positive definite $\mathbf{F}$-based Hessian based on the analysis in §3. To obtain the position-based derivatives in Eqn. 3, we will need the Jacobian of the deformation gradient:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \begin{bmatrix} -\eta_2 - \sigma & \eta_2 & \sigma & \mathbf{0} \\ -\eta_3 - \sigma & \eta_3 & \mathbf{0} & \sigma \end{bmatrix} \in \mathbb{R}^{6 \times 12} \tag{43}$$

where

$$\sigma = \mathbf{I}_3 - \mathbf{t}_1 \mathbf{t}_1^\top \qquad \eta_2 = -\frac{1}{\|\mathbf{y}_1\|}\left((\mathbf{t}_1^\top \mathbf{y}_2)\sigma + \mathbf{t}_1 \mathbf{z}_0^\top\right) \qquad \eta_3 = -\frac{1}{\|\mathbf{y}_1\|}\left((\mathbf{t}_1^\top \mathbf{y}_3)\sigma + \mathbf{t}_1 \mathbf{z}_1^\top\right). \tag{44}$$

Because our novel definition of deformation gradient is based on projected vectors, we need to be particularly careful with the chain rule in Eqn. 3. Unlike many FEM-based elastic energy models, the first term of Eqn. 3 is nonzero. Instead, it becomes:

$$\left(\frac{\partial \Psi}{\partial \mathbf{F}}\right)^\top \frac{\partial^2 \mathbf{F}}{\partial \mathbf{x}^2} = -g\mathcal{S}\left[\frac{\tau_0^\top}{\|\mathbf{z}_0\|} \quad -\frac{\tau_1^\top}{\|\mathbf{z}_1\|}\right]^\top \frac{\partial^2 \mathbf{F}}{\partial \mathbf{x}^2} = -g\mathcal{S}\mathbf{W} \tag{45}$$

where $\mathbf{W} \in \mathbb{R}^{12}$. To guarantee the positive-definiteness of the position-based Hessian, we need to efficiently filter $\mathbf{W}$, whose full analytic expression we give in Appendix A. Since $\mathbf{W}$ is a rank-4 matrix composed of outer products, its null space is easy to analyze. We can then find a basis in the orthogonal complement of the null space that can further sparsify $\mathbf{W}$

$$\mathbf{A_W} = \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{bmatrix} \in \mathbb{R}^{12 \times 4} \tag{46}$$

We will present the definitions of $\mathbf{p}_i$ and show that they encode motions that complement the 6-dimensional $\mathbf{F}$-based eigensystem in Appendix A. We can write $\mathbf{W}$ in terms of the new basis:

$$\mathbf{A_W}^{-1}\mathbf{W}\mathbf{A_W} = \begin{bmatrix} -\kappa_0 & 0 & d_{00} & d_{01} \\ 0 & \kappa_0 & d_{10} & d_{01} \\ \kappa_1 & 0 & 0 & 0 \\ 0 & \kappa_1 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \tag{47}$$

where

$$\kappa_0 = -\frac{4\tau_0^\top \tau_{1\perp}}{\|\mathbf{y}_1\|^2} \qquad\qquad \kappa_1 = \frac{2\tau_0^\top \tau_{1\perp}}{\|\mathbf{y}_1\|} \qquad (48)$$

$$d_{00} = \frac{\|\mathbf{p}_2\|^2}{\|\mathbf{p}_0\|^2} \qquad d_{01} = \frac{\mathbf{p}_2^\top \mathbf{p}_3}{\|\mathbf{p}_1\|^2} \qquad d_{10} = \frac{\mathbf{p}_2^\top \mathbf{p}_3}{\|\mathbf{p}_0\|^2} \qquad d_{01} = \frac{\|\mathbf{p}_3\|^2}{\|\mathbf{p}_1\|^2} \qquad (49)$$

Now the eigenvalues of $\mathbf{A_W^{-1}WA_W}$ can be computed by solving a depressed quartic equation analytically, which has existing, mature, and efficient implementations[Khashin 2014]. For an eigenvalue λ, the corresponding unnormalized eigenvector can be computed by:

$$\mathbf{q} = \mathbf{p}_0 + \rho\mathbf{p}_1 + \nu\mathbf{p}_2 + \rho\nu\mathbf{p}_3 \qquad (50)$$

where

$$\nu = \frac{\kappa_1}{\lambda} \qquad\qquad \rho = \frac{1}{\nu^2 d_{10}} - \frac{d_{00}}{d_{10}} + \frac{\kappa_0}{\kappa_1 d_{10}\nu}. \qquad (51)$$

Our experiments have shown that two out of the four eigenvalues of $\mathbf{W}$ are always non-negative. Therefore, to perform semi-positive definite projection on $\mathbf{W}$, we find the two non-negative eigenvalues and assemble their corresponding outer products.

5 DISCUSSION AND RESULTS

Our analysis is applicable to many angle-based energies and can be integrated into an FEM solver. We have included in our supplemental materials Matlab/Octave implementations that verify the correctness of our derivation and expressions using finite difference tests. We show a variety of elastic rod and thin shell simulations to demonstrate the robustness of our method across different configurations, and under large time steps.

5.1 Simulating Wire Nets

5.1.1 Implementation. To test the performance of our model on strands, we simulated a stiff net of crossing wires, such as those used in architectural [Panetta et al. 2019b] applications. We inserted a bending term between each pair of intersecting rod segments, resulting in six bending stencils for each internal vertex in a checkerboard net (Fig. 5).
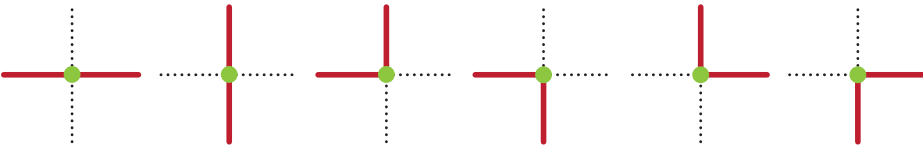

Fig. 5. The six bending elements at an inner vertex of a checkerboard net. Each pair of red edges denote a bending stencil.

We applied the stretching energy from Discrete Viscous Threads (DVT) [Bergou et al. 2010], but omitted the twisting energy because torsion is minimal in such constrained configurations. For bending, we tried both the quadratic model from Eqn. 34 and the tangent-based model from Eqn. 33. We use a backward Euler [Baraff and Witkin 1998] integrator with a large time step of $dt = 1/30$ s, and solved the global system using a Cholesky solver. We converted E to μ using the formula $\mu = \frac{E\pi}{4}r^4$ where $r = 0.1$ cm.

(a) Quadratic Bending
$E = 1e^{11}$ Pa

(b) Quadratic Bending
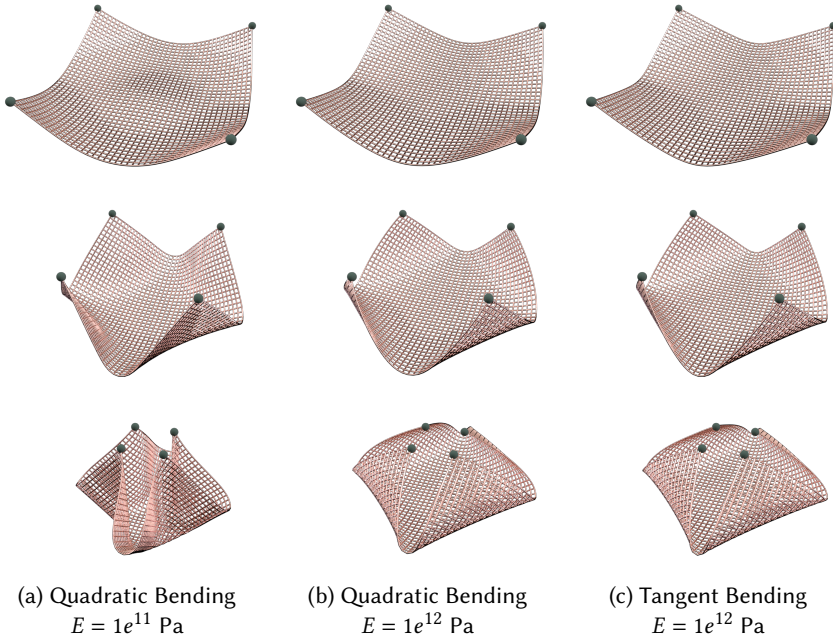$E = 1e^{12}$ Pa

(c) Tangent Bending
$E = 1e^{12}$ Pa

Fig. 6. A wire net with four pinned corners is simulated using different Young's moduli. The rows are at frames 50, 200 and 400. Our method is stable for various E in the quadratic model (Eqn. 34), as well as the tangent model (Eqn. 33).

(a) $E = 1e^{10}$ Pa, frame 5

(b) $E = 1e^{11}$ Pa, frame 12
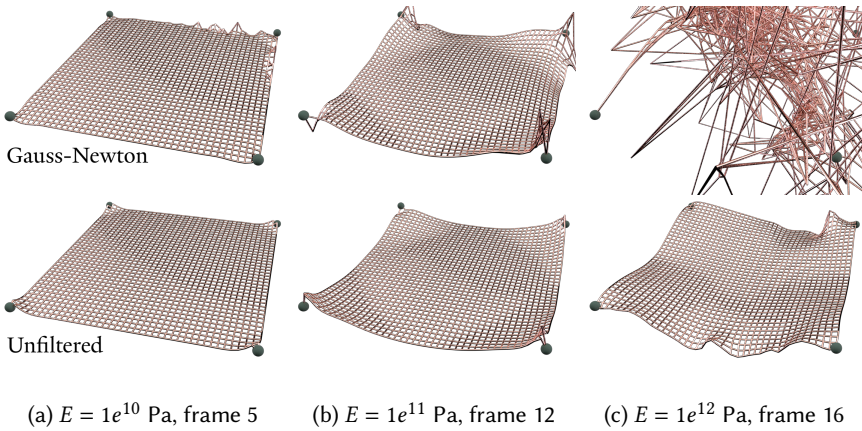
(c) $E = 1e^{12}$ Pa, frame 16

Fig. 7. Selected frames before explosion from Gauss-Newton (upper) and unfiltered (lower) simulations with different stiffnesses.

5.1.2 Results. We test the robustness of our model under three different stiffnesses: $E = 1e^{10}$ Pa, $E = 1e^{11}$ Pa, and $E = 1e^{12}$ Pa. We compared our filtered approach to an unfiltered method and a Gauss-Newton approximation [Choi and Ko 2002] that only includes the leading outer product term of the Hessian (Fig. 6).

(a) Frame 61   (b) Frame 100   (c) Frame 300   (d) Frame 400
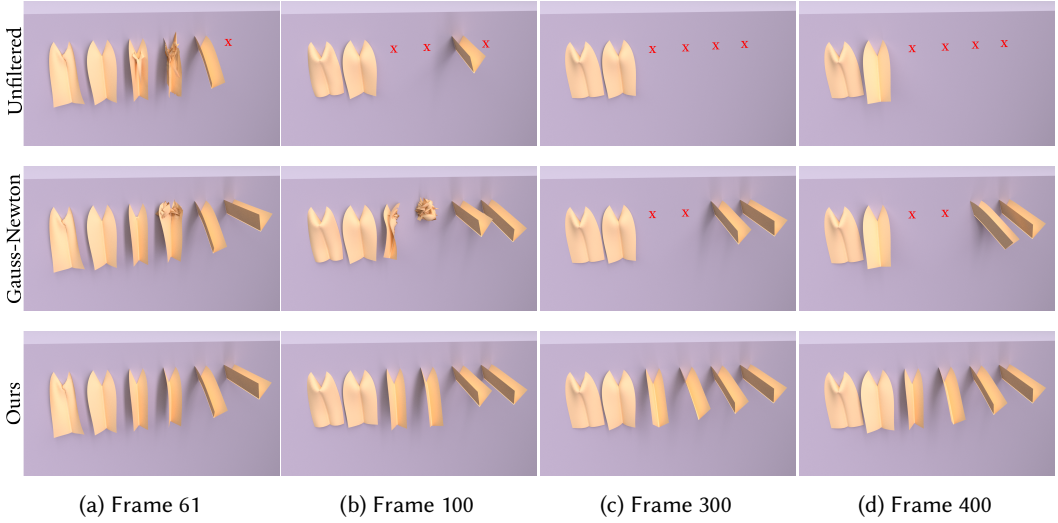
Fig. 8. Cloth with bending stiffnesses of $\mu = 1, 10, 1e^2, 1e^3, 1e^4$ and $1e^5$. All are simulated using a quadratic hinge energy [Grinspun et al. 2003]. Unfiltered and Gauss-Newton approximations explode under a variety of scenarios, which are denoted with an ×. Our filtered Hessians produce smooth results in all cases.

As can be seen in the video, at $E = 1e^{10}$ Pa, the unfiltered, Gauss-Newton and filtered (ours) approaches are all relatively stable. The Gauss-Newton approach contains some jittering at the beginning (Fig. 7a), but stabilizes in later frames.

Under larger stiffness of $E = 1e^{11}$ Pa and $E = 1e^{12}$ Pa, bending forces start to dominate and both unfiltered and Gauss-Newton explode immediately after the simulation starts. Fig. 7 shows the frames right before explosion. For both methods, the instability emerges at the corners and boundaries, and propagates to the whole model very quickly. We also found that Gauss-Newton is generally less stable than the unfiltered method and explodes faster. In contrast, our method is totally stable and settles to an equilibrium for both the quadratic and tangent-based energies. Our method efficiently simulates stiff strand bending, even with large time steps.

5.2 Simulating Shells

5.2.1 Implementation. For shells, we constructed a row of V-shaped cloth strips with varying stiffness, and set the rest angle of the V to $\pi/4$. One end of the cloth is attached to a wall, and collisions with the wall are handled with Baraff-Witkin filtering [Baraff and Witkin 1998]. We use quadratic bending (Eqn. 37), and ARAP-type membrane stretching [Kim 2020]. The scene is simulated with backward Euler with a time step of $dt = 1/30$. For the linear system solver, we used both preconditioned conjugate gradients (PCG) and Cholesky method.

5.2.2 Results. We again compare our method against unfiltered and Gauss-Newton strategies. Fig. 8 shows frames from all three methods using PCG, where a red cross appears when the simulation exploded. In the first row of unfiltered Hessians, we see that the simulation is unstable under high bending stiffnesses. As the strips droop under gravity, the bending energy increases, and local instabilities quickly cause the simulation to explode.

The second row shows the results using Gauss-Newton. Unlike the unfiltered case, Gauss-Newton is able to keep the system semi-positive definite under large stiffnesses, as can be seen in the rightmost $\mu = 1e^5$ strips. However, the middle strips show that Gauss-Newton is unstable

(a) Frame 61   (b) Frame 100   (c) Frame 349   (d) Frame 400
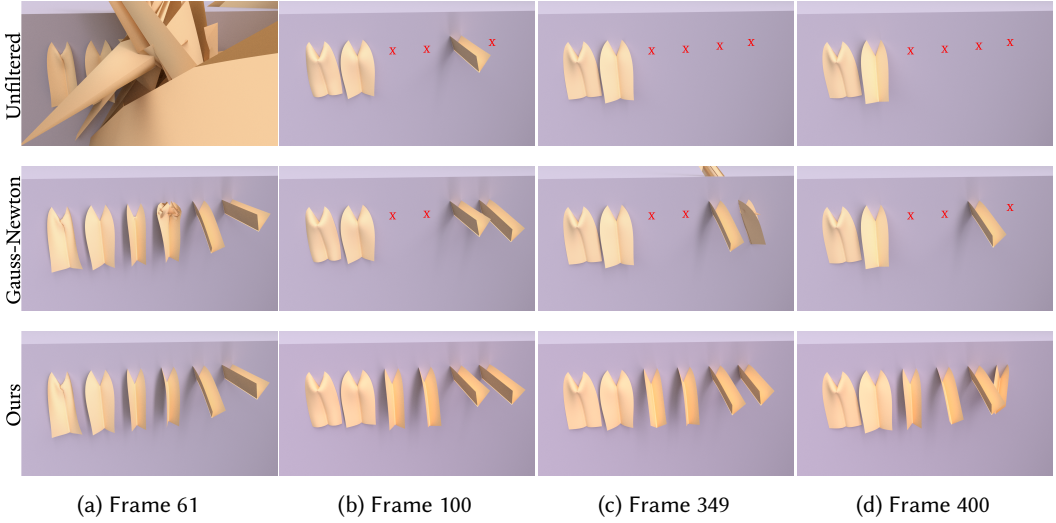
Fig. 9. Comparison of the three approaches with bending stiffnesses of $\mu = 1, 10, 1e^2, 1e^3, 1e^4$ and $1e^5$ using the same setting as Fig. 8 except that the linear system is solved using the Cholesky method.

	$\mu = 1$	10	$1e^2$	$1e^3$	$1e^4$	$1e^5$	Avg.
Baseline	547	547	558	547	546	553	550
Ours	622	620	629	636	628	625	627
Numerical	3829	3849	3922	3931	4110	4112	3959

Table 1. Shell scene (Fig. 8) timings. The three methods being compared are the unfiltered baseline, our method, and a brute-force numerical eigendecomposition. All timings are in milliseconds. The last columns shows average timing across all stiffnesses.

under mid-range stiffnesses that undergo larger deformations. Some strips explodes when they encounter a kinematic collision with the wall. The second strip from the right does not explode, but drifts to one side and non-physically breaks the symmetry of the system.

Our method is shown in the last row of Fig. 8, and is consistently stabler than both unfiltered and Gauss-Newton. Our filtered Hessian is able to fully describe the significant eigenmodes of the system while discarding unstable modes. Fig. 9 shows the simulation results using a Cholesky solver. Cholesky is usually more sensitive to instabilities when handling large linear systems, so both the Gauss-Newton and unfiltered methods explode much earlier than PCG. The second row shows that with Gauss-Newton, the rightmost strip with largest stiffness drifts to one side and hits the wall, eventually exploding. This is because Gauss-Newton is not able to adjust to sudden local displacements caused by kinematic collisions. The last row shows that our method also generates asymmetric motion, which we hypothesize is due to numerical noise pushing the simulation out of an energetic saddle point. Regardless, our method remains stable.

5.3 Crushed Cylinders

To test the robustness of our method on more complicated examples, we crushed stiff cylinders composed of both thin shells and wire nets (Fig. 1). We attach the top and bottom of the cylinder to two cubes that then crush the geometry. We simulated the thin shell using a bending stiffness of

	$E = 1e^{12}$	$1e^{11}$	$1e^{10}$	Avg.
Baseline	1764	1775	1752	1764
Ours	1898	1866	1879	1881
Numerical	12689	12920	13203	12937

Table 2. Net scene (Fig. 6) timings. All timings are in milliseconds and the Young's moduli are in Pascals. Methods are the same as in Table 1.

$\mu = 1e^6$, and the wire cylinder under the Young's modulus of $E = 1e^{12}$ Pa. Other settings are the same as §5.1 and §5.2. Non-trivial buckling patterns form and evolve during the simulation, and the dynamics remains stable even under sudden, large deformations. In the video, we show that compared to our method, both unfiltered and Gauss-Newton failed to produce the wrinkles.

5.4 Timing and Efficiency

We measure efficiency of our method by using unfiltered Hessian construction time as a baseline, and comparing it to our method, as well as a brute-force numerical call to SelfAdjointEigenSolver from the Eigen library [Guennebaud et al. 2010]. In Tables 1 and 2 we list the total time spent, in milliseconds, of computing the bending energy Hessian over 400 simulation frames. All the tests were implemented in C++ using the Eigen library, and were run on an Apple M1 Pro chip with 10-core CPU and 32 GB memory.

Our method is 6× to 7× faster than the brute-force numerical call. Compared to the unfiltered baseline, our method adds a negligible 14% to the shell scene and 7% to the net scene. Considering the robustness under large time step that our method adds, this trade-off is acceptable.

6 CONCLUSIONS AND FUTURE WORK

We present the first clean eigenanalysis of angle-based deformation energies based on an abstract stencil. Our analysis is physically interpretable and reveals the intrinsic geometric structure behind the energy Hessian. We provide a semi-quantitative interpretation of the eigenmodes of angle-based energies, and a fast scheme for assembling a semi-positive definite Hessian per stencil. We apply our method to elastic rod and shell simulation and have achieved consistent improvements to stability compared to popular existing methods, while also maintaining efficiency.

Per-element SPD projection can be suboptimal, because the finite element approximation can be sensitive to the discretization scheme. In some extreme cases involving large velocities, fixing the indefiniteness locally cannot prevent the instability from propagating to the global modes. Whether there is a projection scheme that can account for global low-frequency dynamics is future work. Even for a local per-element scenario, a fully analytical eigensystem is not guaranteed to exist for energy types not covered in this paper, such as anisotropic rod bending.

A general eigendecomposition that is applicable to this type of angle-based energy can have other potential applications, such as exerting controls and constraints over the simulation system in a physically valid way. Thus, generalizing our ideas to broader applications remains future work.

ACKNOWLEDGMENTS

A FULL EXPRESSIONS OF THE SHELL EIGENANALYSIS

We present the full expression of the first term in Eqn. 3. This is used to derive the eigensystem of $\mathbf{W}$ in §4.2. We also implemented this for experiments. First, we give the building blocks based on the block-wise formulation in Eqn. 43:

$$\tau_{0\perp}^\top \left(\frac{\partial \eta_2}{\partial \mathbf{x}}\right) = \frac{1}{\|\mathbf{y}_1\|^2} \left(\begin{bmatrix} \phi_2 \tau_{0\perp}^\top & -\phi_2 \tau_{0\perp}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} - \tau_{0\perp} \begin{bmatrix} -\phi_2^\top - \mathbf{y}_1^\top & \phi_2^\top & \mathbf{y}_1^\top & \mathbf{0} \end{bmatrix}\right) \in \mathbb{R}^{3\times12}$$

$$\tau_{1\perp}^\top \left(\frac{\partial \eta_3}{\partial \mathbf{x}}\right) = \frac{1}{\|\mathbf{y}_1\|^2} \left(\begin{bmatrix} \phi_3 \tau_{1\perp}^\top & -\phi_3 \tau_{1\perp}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} - \tau_{1\perp} \begin{bmatrix} -\phi_3^\top - \mathbf{y}_1^\top & \phi_3^\top & \mathbf{y}_1^\top & \mathbf{0} \end{bmatrix}\right) \in \mathbb{R}^{3\times12}$$

$$\tau_{0\perp}^\top \left(\frac{\partial \sigma}{\partial \mathbf{x}}\right) = \frac{1}{\|\mathbf{y}_1\|^2} \begin{bmatrix} \mathbf{y}_1 \tau_{0\perp}^\top & -\mathbf{y}_1 \tau_{0\perp}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3\times12}$$

$$\tau_{1\perp}^\top \left(\frac{\partial \sigma}{\partial \mathbf{x}}\right) = \frac{1}{\|\mathbf{y}_1\|^2} \begin{bmatrix} \mathbf{y}_1 \tau_{1\perp}^\top & -\mathbf{y}_1 \tau_{1\perp}^\top & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3\times12}$$

where

$$\phi_2 = \mathbf{y}_2 - 2\mathbf{t}_1 \mathbf{t}_1^\top \mathbf{y}_2 \qquad\qquad \phi_3 = \mathbf{y}_3 - 2\mathbf{t}_1 \mathbf{t}_1^\top \mathbf{y}_3 \tag{52}$$

Then the full expression of $\mathbf{W} \in \mathbb{R}^{12\times12}$ is:

$$\mathbf{W} = \begin{bmatrix} \frac{\tau_{0\perp}^\top}{\|\mathbf{z}_0\|}\left(-\frac{\partial \eta_2}{\partial \mathbf{x}} - \frac{\partial \sigma}{\partial \mathbf{x}}\right) + \frac{\tau_{1\perp}^\top}{\|\mathbf{z}_1\|}\left(-\frac{\partial \eta_3}{\partial \mathbf{x}} - \frac{\partial \sigma}{\partial \mathbf{x}}\right) \\ \frac{\tau_{0\perp}^\top}{\|\mathbf{z}_0\|}\left(\frac{\partial \eta_2}{\partial \mathbf{x}}\right) - \frac{\tau_{1\perp}^\top}{\|\mathbf{z}_1\|}\left(\frac{\partial \eta_3}{\partial \mathbf{x}}\right) \\ \frac{\tau_{0\perp}^\top}{\|\mathbf{z}_0\|}\left(\frac{\partial \sigma}{\partial \mathbf{x}}\right) \\ -\frac{\tau_{1\perp}^\top}{\|\mathbf{z}_0\|}\left(\frac{\partial \sigma}{\partial \mathbf{x}}\right) \end{bmatrix} \tag{53}$$

We can see that $\mathbf{W}$ is sparse and each block is an outer product. This property enables fast matrix assembly, and leads us to analyze the structure of the null space and row space.

Next, by investigating the structure of $\mathbf{W}$, we found four vectors in the row space that has an interesting property:

$$\mathbf{p}_1^\top \mathbf{W} \mathbf{p}_0 = \mathbf{p}_0^\top \mathbf{W} \mathbf{p}_1 = 0 \qquad\qquad \mathbf{p}_2^\top \mathbf{W} \mathbf{p}_2 = \mathbf{p}_3^\top \mathbf{W} \mathbf{p}_3 = \mathbf{p}_2^\top \mathbf{W} \mathbf{p}_3 = 0 \tag{54}$$

$$\mathbf{W} \mathbf{p}_0 = -\kappa_0 \mathbf{p}_0 + \kappa_1 \mathbf{p}_2 \qquad\qquad \mathbf{W} \mathbf{p}_1 = \kappa_0 \mathbf{p}_1 + \kappa_1 \mathbf{p}_3 \tag{55}$$

where $\mathbf{p}_0$ and $\mathbf{p}_1$ are defined as

$$\mathbf{p}_0 = \begin{bmatrix} \tau_0 + \tau_1 \\ -\tau_0 - \tau_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \qquad\qquad \mathbf{p}_1 = \begin{bmatrix} \tau_0 - \tau_1 \\ -\tau_0 + \tau_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{56}$$

and computing $\mathbf{p}_2$ and $\mathbf{p}_3$ using Eqn. 55 yields:

$$\mathbf{p}_2 = \begin{bmatrix} (-a_1 - a_0 + c_1 + c_0)\mathbf{t}_1 \\ (a_1 + a_0)\mathbf{t}_1 \\ -c_0 \mathbf{t}_1 \\ -c_1 \mathbf{t}_1 \end{bmatrix} \qquad\qquad \mathbf{p}_3 = \begin{bmatrix} (-a_1 + a_0 - c_1 + c_0)\mathbf{t}_1 \\ (a_1 - a_0)\mathbf{t}_1 \\ c_0 \mathbf{t}_1 \\ -c_1 \mathbf{t}_1 \end{bmatrix}$$

$$a_0 = \frac{\mathbf{t}_1^\top \mathbf{y}_2}{\|\mathbf{y}_1\|\|\mathbf{z}_0\|} \qquad a_1 = \frac{\mathbf{t}_1^\top \mathbf{y}_3}{\|\mathbf{y}_1\|\|\mathbf{z}_1\|} \qquad c_0 = \frac{1}{\|\mathbf{z}_0\|} \qquad c_1 = \frac{1}{\|\mathbf{z}_1\|} \tag{57}$$

where $\mathbf{p}_2$ and $\mathbf{p}_3$ are displacements along the hinge direction $\mathbf{t}_1$ for each of the four vertices. The projected edges $\mathbf{z}_0$ and $\mathbf{z}_1$ is invariant in this case and these modes $\mathbf{p}_2$ and $\mathbf{p}_3$ are not captured in

the $\mathbf{F}$-based Hessian. The $\mathbf{p}_0$ and $\mathbf{p}_1$ directions represent a rotation of the hinge edge $\mathbf{y}_1$, and they are supplementary to the rotation modes of the $\mathbf{F}$-based Hessian $\mathbf{q}_4$ and $\mathbf{q}_5$. After further exploiting Eqn. 54 and Eqn. 55 using the symmetry of $\mathbf{W}$, we are able to construct the matrix in Eqn. 47 in the chosen basis.

Because assembling the 12 by 12 filtered $\mathbf{W}$ directly from two outer products of 12-dimensional vectors is relatively lightweight compared to the $\mathbf{F}$-based term, the shell-based variation of our analysis remains efficient as shown in §5.4.

REFERENCES

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

David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 43–54. https://doi.org/10.1145/280814.280821

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. In *ACM SIGGRAPH 2010 Papers* (Los Angeles, California) *(SIGGRAPH '10)*. Association for Computing Machinery, New York, NY, USA, Article 116, 10 pages. https://doi.org/10.1145/1833349.1778853

Javier Bonet, Antonio J Gil, and Richard D Wood. 2021. *Nonlinear solid mechanics for finite element analysis: dynamics*. Cambridge University Press.

David Bourguignon and Marie-Paule Cani. 2000. Controlling anisotropy in mass-spring systems. In *Proceedings of the Eurographics Workshop on Computer Animation and Simulation 2000*. Springer, 113–123.

David E Breen, Donald H House, and Michael J Wozny. 1994. Predicting the drape of woven cloth using interacting particles. In *Proceedings of SIGGRAPH*. 365–372.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of SIGGRAPH*. 594–603.

R. Bridson, S. Marino, and R. Fedkiw. 2003. Simulation of Clothing with Folds and Wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 28–36.

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph.* 21, 3 (jul 2002), 604–611. https://doi.org/10.1145/566654.566624

Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. 2014. Yarn-Level Simulation of Woven Cloth. 33, 6, Article 207 (nov 2014), 11 pages. https://doi.org/10.1145/2661229.2661279

Gabriel Cirio, Jorge Lopez-Moreno, and Miguel A Otaduy. 2015. Efficient simulation of knitted cloth using persistent contacts. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 55–61.

Agnes Daldegan, Nadia Magnenat Thalmann, Tsuneya Kurihara, and Daniel Thalmann. 1993. An integrated system for modeling, animating and rendering hair. In *Computer Graphics Forum*, Vol. 12. Wiley Online Library, 211–221.

Yotam Gingold, Adrian Secord, Jefferson Y Han, Eitan Grinspun, and Denis Zorin. 2004. A discrete model for inelastic deformation of thin shells. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Citeseer.

Gene H Golub and Charles F Van Loan. 2013. *Matrix Computations*. Johns Hopkins University Press.

Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete Shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) *(SCA '03)*. Eurographics Association, Goslar, DEU, 62–67.

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

Sunil Hadap. 2006. Oriented strands: dynamics of stiff multi-body system. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer animation*. 91–100.

Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin. 2013. Artistic simulation of curly hair. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 63–71.

Sergei Khashin. 2014. *Solution of cubic and quartic equations C++*. http://math.ivanovo.ac.ru/dalgebra/Khashin/poly/index.html

Theodore Kim. 2020. A Finite Element Formulation of Baraff-Witkin Cloth. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Virtual Event, Canada) *(SCA '20)*. Eurographics Association, Goslar, DEU, Article 16, 9 pages. https://doi.org/10.1111/cgf.14111

Theodore Kim, Fernando De Goes, and Hayley Iben. 2019. Anisotropic Elasticity for Inversion-Safety and Element Rehabilitation. *ACM Trans. Graph.* 38, 4, Article 69 (jul 2019), 15 pages. https://doi.org/10.1145/3306346.3323014

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. 2019a. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.

J. Panetta, M. Konaković-Luković, F. Isvoranu, E. Bouleau, and M. Pauly. 2019b. X-Shells: A New Class of Deployable Beam Structures. *ACM Trans. Graph.* 38, 4, Article 83 (jul 2019), 15 pages. https://doi.org/10.1145/3306346.3323040

Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A Canabal, Robert Sumner, and Miguel A Otaduy. 2015. Design and fabrication of flexible rod meshes. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–12.

Xavier Provot et al. 1995. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface.* Canadian Information Processing Society, 147–147.

Rosa M Sánchez-Banderas, Alejandro Rodríguez, Héctor Barreiro, and Miguel A Otaduy. 2020. Robust eulerian-on-lagrangian rods. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 59–1.

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

Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Trans. Graph.* 38, 1, Article 3 (feb 2019), 15 pages. https://doi.org/10.1145/3241041

Shinjiro Sueda, Garrett L. Jones, David I. W. Levin, and Dinesh K. Pai. 2011. Large-Scale Dynamic Simulation of Highly Constrained Strands. *ACM Trans. Graph.* 30, 4, Article 39 (jul 2011), 10 pages. https://doi.org/10.1145/2010324.1964934

Rasmus Tamstorf and Eitan Grinspun. 2013. Discrete bending forces and their Jacobians. *Graphical Models* 75, 6 (2013), 362–370. https://doi.org/10.1016/j.gmod.2013.07.001

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '05).* Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/1073368.1073394

Pascal Volino, Martin Courchesne, and Nadia Magnenat Thalmann. 1995. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of SIGGRAPH.* 137–144.

Kelly Ward and Ming Lin. 2003. Adaptive Grouping and Subdivision for Simulating Hair Dynamics. In *Proceedings of Pacific Graphics.* 234–243.

Max Wardetzky, Miklós Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. 2007. Discrete quadratic curvature energies. *Computer Aided Geometric Design* 24, 8-9 (2007), 499–518.

Haomiao Wu, Alvin Shi, Jarred Parr, A.M. Darke, and Theodore Kim. 2023. Lifted Curls: A Model for Tightly Coiled Hair Simulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, California). Association for Computing Machinery.