

A Unified Analysis of Penalty-Based Collision Energies

ALVIN SHI and THEODORE KIM, Yale University, USA



Fig. 1. Cable stretching simulation using penalty-based collisions. Our analytic filter (left) proceeds smoothly, while Gauss-Newton (right) causes cables to incorrectly pop through (red arrows).

We analyze a wide class of penalty energies used for contact response through the lens of a *reduced frame*. Applying our analysis to both spring-based and barrier-based energies, we show that we can obtain closed-form, analytic eigensystems that can be used to guarantee positive semidefiniteness in implicit solvers. Our approach is both faster than direct numerical methods, and more robust than approximate methods such as Gauss-Newton. Over the course of our analysis, we investigate physical interpretations for two separate notions of length. Finally, we showcase the stability of our analysis on challenging strand, cloth, and volume scenarios with large timesteps on the order of $1/40$ s.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Collisions, Penalty Energies, Barrier Functions

ACM Reference Format:

Alvin Shi and Theodore Kim. 2023. A Unified Analysis of Penalty-Based Collision Energies. *Proc. ACM Comput. Graph. Interact. Tech.* 6, 3, Article 41 (August 2023), 19 pages. <https://doi.org/10.1145/3606934>

1 INTRODUCTION

Robust collision response is an integral part of any deformable body simulation, such as those used in feature film production, games, and biology. Perhaps the simplest approach is a penalty-based response, which applies forces to primitives that have violated some distance threshold.

In order to obtain stable motion under large timesteps, implicit solvers require the gradients of these forces [Baraff and Witkin 1998]. However, these gradients can produce indefinite systems, so

Authors' address: Alvin Shi, alvin.shi@yale.edu; Theodore Kim, Yale University, New Haven, USA, theodore.kim@yale.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2023 Copyright held by the owner/author(s).

2577-6193/2023/8-ART41

<https://doi.org/10.1145/3606934>

they are commonly filtered to be positive-semi-definite (PSD) by removing negative eigenvalues at each quadrature point using a numerical factorization [Teran et al. 2005], or by applying a Gauss-Newton approximation [Choi and Ko 2002; Kim and Eberle 2022; Zehnder et al. 2021]. A better approach is to use a fast, analytic eigendecomposition, but while these have been constructed for a wide variety of deformation energies [Kim 2020; Lin et al. 2022; Smith et al. 2019], penalty-based collisions have not undergone the same treatment. This is unfortunate, because we have found that they can provide significant gains in speed and robustness.

We perform this analysis by recasting collisions in terms of a novel *reduced frame*, \mathbf{C} . From this perspective, a unified treatment emerges that encapsulates both vertex-face and edge-edge collisions. From there, we can perform a generic eigenanalysis that yields closed-form expressions for the eigendecomposition of penalty-based force gradients. Over the course of this analysis, we present a notion of signed and unsigned length that is applicable to a variety of collision energies. We then recast several commonly-used energies in terms of these length measures.

Our final analysis is performed on general penalty energies involving unsigned and signed length. We show a variety of scenarios where applying our approach is significantly faster than computing a numerical eigendecomposition, and more robust than using approximate force gradients.

2 PREVIOUS WORK

2.1 Penalty-Based Collision Energies

2.1.1 Spring Energies. Penalty-based collision forces are a well-established method for handling contact in deformable body simulation [Baraff and Witkin 1998; Jimenez and Luciani 1993]. Object inter-penetrations are suppressed by inserting a spring between vertex-face or edge-edge pairs that cross a distance threshold. They have similarly been used for cloth at least since Baraff and Witkin [1998], though additional challenges arise when inside/outside queries become unavailable. For example, Teran et al. [2005] use quadratic energies to generate filtered force gradients for self-collisions that occur during muscle and flesh simulation.

Strand collisions can be viewed as a collection of edge-edge resolutions [Spillmann and Teschner 2007], and have additionally been used to simulate cohesion [Ward et al. 2004] in addition to contact repulsion. Many works [Chang et al. 2002; Choe et al. 2005; Selle et al. 2008] use penalty forces during hair simulation, and stiff springs to adhere joints in woven cloth [Michels et al. 2015].

Specific simulation goals have given rise to a variety of spring penalty formulations. Gast et al. [2015] introduce a linear falloff in the spring constant, yielding a cubic energy, and McAdams et al. [2011] use a control variable to interpolate between isotropic and anisotropic springs. Spring-like forces are also used to simulate clumping in wet hair, where Lin et al. [2011] use “wetness” to determine the activation distance for adhesive springs at strand joints, while Fei et al. [2017] use cross-sectional surface areas to govern the strength of inter-strand cohesion.

2.1.2 Barrier Energies. Barrier-type energies contain terms that grow to infinity under special conditions, effectively prohibiting specific configurations while maintaining a penalty-like approach [Schüller et al. 2013]. Recently, Incremental Potential Contact (IPC) has made extensive use such energies for collisions, and utilize similar per-element PSD projections [Li et al. 2020]. This underlying framework is extensible to codimensional settings featuring volumes and cloth [Li et al. 2021], rigid bodies [Lan et al. 2022], and generalized solvers [Chen et al. 2022]. Our eigenanalysis extends equally well to the vertex-face and edge-edge energies used in these IPC-style solvers.

2.2 Analytic Eigenanalysis

Several techniques have been developed for analytically computing the eigensystems of various force gradients. However, these have all been for specific cases of hyperelastic materials, not the collision energies that are the focus of this paper.

Smith et al. [2019] performed an eigenanalysis of hyperelastic deformation energies through the introduction of novel stretch-based invariants, which was later extended to membrane stretching energies [Kim 2020; Panetta 2020]. Lin et al. [2022] used a polynomial-based method on the Cauchy-Green invariants to perform an alternate analysis of ARAP-style [Sorkine and Alexa 2007] materials. All of these analyses offer broad techniques for constructing fast, analytic eigendecompositions, but to our knowledge, ours is the first to attempt a similar analysis on collision energies.

2.3 Complementary Approaches

Complementary collision response strategies have been developed to address more extreme collision scenarios, and deployed in addition to penalty-based energies. Such treatments include linear complementarity formulations [Bridson et al. 2003; Gascón et al. 2010], with careful considerations toward preventing artifacts that arise from direct vertex corrections. Eulerian grids have also been investigated for collision resolution, leveraging incompressibility conditions [McAdams et al. 2009; Teng et al. 2016] or velocity constraints [Levin et al. 2011] to maintain a penetration-free state. The success of these methods are then linked to grid resolution. Müller et al. [2015] construct volumetric “air” elements between objects that strictly prohibit negative volume, but element construction for complex scenes introduces additional difficulties.

Perhaps the closest approach to our reduced frame is in Jiang et al. [2017], where the \mathbf{R} matrix from the QR factorization is used to construct a deformation-specific frame. However, their factorization is used to construct a new elastic potential, not analyze existing ones. We investigate directions, while they generate forces.

3 A REDUCED COLLISION FRAME

In the following, we will denote matrices capital bold (\mathbf{X}), vectors with lower-case bold (\mathbf{x}), and scalars with unbolded lower-case (x). Normalized vectors are denoted with a hat ($\hat{\mathbf{x}}$). We also use the tensor unfolding operator $\text{Vec}(\cdot)$ [Golub and Van Loan 2013; Kim and Eberle 2022].

In collision processing, *length* is the most basic geometric computation, because it asks: are two primitives close to each other? We characterize this with a generic *signed length* function

$$l(\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k) = \frac{\mathbf{e}_k \cdot (\mathbf{e}_i \times \mathbf{e}_j)}{\|\mathbf{e}_i \times \mathbf{e}_j\|}. \quad (1)$$

The challenge is then to define the $\mathbf{e}_{i,j,k}$ edges appropriately. In order to arrive at a unified analysis of both vertex-face and edge-edge collision energies, we will define the edges in terms of a *projected collision frame* that contains all the important geometric information for collisions, while discarding everything else. This frame will then form the basis for our analysis.

3.1 Vertex-Face and Edge-Edge Lengths

For vertex-face collisions, we denote \mathbf{e}_0 and \mathbf{e}_1 as the two edges of the triangle face, and \mathbf{e}_2 as the vector from any vertex on the triangle to the collision vertex (Fig. 2, left). With these edges defined, $l(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ gives us the desired point-plane distance.

We observe that $\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2\}$ do not generally form an orthogonal frame. This is most visible with \mathbf{e}_2 in Fig. 2 (left), which contains components that are tangential to the triangle face. This extraneous information is not required to compute point-to-plane distance.

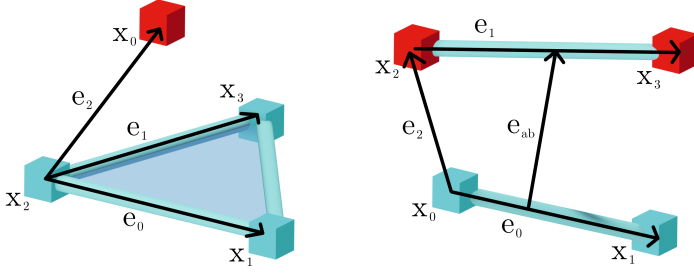


Fig. 2. *Left*: The vertex-face collisions. The \mathbf{e}_0 and \mathbf{e}_1 edges are along the triangle face, and \mathbf{e}_2 is a vector from the triangle to the collision vertex. *Right*: Edge-edge collisions. The \mathbf{e}_0 and \mathbf{e}_1 edges are being checked, and \mathbf{e}_2 is the vector pointing from the start of \mathbf{e}_0 to the start of \mathbf{e}_1 . \mathbf{e}_{ab} is the vector of least distance from \mathbf{e}_0 to \mathbf{e}_1 .

For two edges in collision, we denote \mathbf{e}_0 and \mathbf{e}_1 as the two nearby edges and \mathbf{e}_{ab} as the vector of least distance between them (Fig. 2, right). This is usually found with a geometric test [Rhodes 2001] and the signed length is computed as $l(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_{ab})$. Again, since $\{\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_{ab}\}$ is not generally orthogonal, we see that the edges contain extraneous information.

3.2 Our Reduced Frame

In order to distill the problem down to its fundamental geometry, we propose a *reduced collision frame*, which we denote $\mathbf{C} = (\mathbf{e}_{0\perp} \ \mathbf{e}_{1\perp} \ \mathbf{e}_{2\perp})$. Here, $\mathbf{e}_{0\perp}$, $\mathbf{e}_{1\perp}$, and $\mathbf{e}_{2\perp}$ are orthogonalized versions of the edges from the above collision cases. This frame is distinct from a QR factorization, because we orthogonalize the vectors but do *not* normalize them, as that would discard important geometric information. This frame closely mirrors the deformation gradient commonly used in the finite element analysis [Bonet and Wood 2008], where irrelevant translation modes are discarded from the analysis by taking the deformation derivative.

The reduced vertex-face and edge-edge collision frames can then be formulated as:

$$\mathbf{C} = \left(\begin{array}{c|c|c} \mathbf{e}_0 & \mathbf{e}_1 - \alpha\mathbf{e}_0 & \mathbf{e}_2 - \beta\mathbf{e}_0 - \gamma\mathbf{e}_1 \end{array} \right) \quad (2)$$

where α , β , and γ are modified Gram-Schmidt orthogonalization constants. These edges are then sent to l (Eqn. 1).

One immediate objection to this approach is that α , β , and γ depend on the edges, so the complexity of the derivatives could potentially explode when computing forces. This may be true for general functions, but not for the case of collisions. We will see in §4.3 that, aside from an easy-to-filter term, all potential sources of complexity when calculating gradients and Hessians of energy functions go to zero.

4 AN ANALYSIS OF COLLISION ENERGIES

We can now use our reduced collision frame to analyze signed length, and from there, generic collision energies.

4.1 Existing Analysis Methods

We begin by observing that existing analysis methods are insufficient. We tried to adapt existing deformation gradient-based methods [Kim et al. 2019], but it only yielded an analysis for *unsigned*

lengths $u = \|\mathbf{e}_{2\perp}\|$. This is insufficient because many collision energies use the sign to determine whether a vertex is approaching (positive) or already in collision (negative) with a face.

We show the results of this analysis in Appendix A. In general, they use matrix invariants along the lines of $I_3(\mathbf{C}) = \det \mathbf{C}$ and $I_5(\mathbf{C}, \mathbf{v}) = \mathbf{v}^\top \mathbf{C}^\top \mathbf{C} \mathbf{v}$. For unsigned length specifically, the cross-terms that arise from differentiating these invariants are numerous and nontrivial, and result in rank-one updates that necessitate numerical root-finding. Overall, we did not find this to be a promising path towards an analytic eigensystem.

4.2 Signed Length Analysis

Instead, we will analyze a generic collision energy that depends on the length between two primitives $\Psi(l)$, which we abbreviate Ψ_l . Our goal is to gain a detailed understanding of the derivatives of this function with respect to our reduced collision frame \mathbf{C} . In §4.3.1, we will show how to transform these generic results back to the vertex-face and edge-edge cases.

In particular, we seek the eigensystem of the Hessian of Ψ_l :

$$\frac{\partial^2 \Psi_l}{\partial \mathbf{c}^2} = \frac{\partial \Psi_l}{\partial l} \mathbf{H}_l + \frac{\partial^2 \Psi_l}{\partial l^2} \mathbf{g}_l \mathbf{g}_l^\top \quad (3)$$

where $\mathbf{c} = \text{Vec}(\mathbf{C})$, $\mathbf{H}_l = \frac{\partial^2 l}{\partial \mathbf{c}^2}$, and $\mathbf{g}_l = \frac{\partial l}{\partial \mathbf{c}}$. We will see later on that \mathbf{g}_l resides in the nullspace of \mathbf{H}_l , so the main challenge is to find the eigensystem of \mathbf{H}_l .

In contrast to the matrix invariant approach, we have found that direct inspection of the Hessian reveals its underlying structure. We directly examine \mathbf{H}_l :

$$\mathbf{H}_l = \begin{pmatrix} \frac{\partial^2 l}{\partial \mathbf{e}_{0\perp}^2} & \frac{\partial^2 l}{\partial \mathbf{e}_{1\perp} \partial \mathbf{e}_{0\perp}} & \frac{\partial^2 l}{\partial \mathbf{e}_{2\perp} \partial \mathbf{e}_{0\perp}} \\ \vdots & \frac{\partial^2 l}{\partial \mathbf{e}_{2\perp}^2} & \frac{\partial^2 l}{\partial \mathbf{e}_{2\perp} \partial \mathbf{e}_{1\perp}} \\ \ddots & \dots & \frac{\partial^2 l}{\partial \mathbf{e}_{2\perp}^2} \end{pmatrix}, \quad (4)$$

where the blocks are composed of second derivatives of l :

$$\frac{\partial^2 l}{\partial \mathbf{e}_{2\perp} \partial \mathbf{e}_{0\perp}} = -l \frac{\mathbf{e}_{2\perp} \mathbf{e}_{0\perp}^\top}{\|\mathbf{e}_{2\perp}\|^2 \|\mathbf{e}_{0\perp}\|^2} \quad \frac{\partial^2 l}{\partial \mathbf{e}_{2\perp} \partial \mathbf{e}_{1\perp}} = -l \frac{\mathbf{e}_{2\perp} \mathbf{e}_{1\perp}^\top}{\|\mathbf{e}_{2\perp}\|^2 \|\mathbf{e}_{1\perp}\|^2} \quad (5)$$

$$\frac{\partial^2 l}{\partial \mathbf{e}_{0\perp}^2} = \frac{-l}{\|\mathbf{e}_{0\perp}\|^2} (\mathbf{I} - \hat{\mathbf{e}}_{0\perp} \hat{\mathbf{e}}_{0\perp}^\top - \hat{\mathbf{e}}_{1\perp} \hat{\mathbf{e}}_{1\perp}^\top) \quad \frac{\partial^2 l}{\partial \mathbf{e}_{1\perp}^2} = \frac{-l}{\|\mathbf{e}_{1\perp}\|^2} (\mathbf{I} - \hat{\mathbf{e}}_{0\perp} \hat{\mathbf{e}}_{0\perp}^\top - \hat{\mathbf{e}}_{1\perp} \hat{\mathbf{e}}_{1\perp}^\top) \quad (6)$$

$$\frac{\partial^2 l}{\partial \mathbf{e}_{1\perp} \partial \mathbf{e}_{0\perp}} = \frac{\partial^2 l}{\partial \mathbf{e}_{2\perp}^2} = \mathbf{0}_{3 \times 3}. \quad (7)$$

A detailed derivation of these blocks is given in the supplemental materials. The matrix is symmetric, so we have fully described its entries. We observe that *all* of the individual blocks are either outer product or projection terms, which allows us to state the eigensystem of \mathbf{H}_l directly:

$$\lambda_{0,1}^l = \frac{-l}{2 \|\mathbf{e}_{1\perp}\|^2} (1 \pm f(1, 2)) \quad \mathbf{q}_{0,1}^l = \begin{pmatrix} \mathbf{0}_3 \\ \mathbf{e}_{2\perp} \\ \omega \mathbf{e}_{1\perp} \end{pmatrix} \quad (8)$$

$$\lambda_{2,3}^l = \frac{-l}{2 \|\mathbf{e}_{0\perp}\|^2} (1 \pm f(0, 2)) \quad \mathbf{q}_{2,3}^l = \begin{pmatrix} \mathbf{e}_{2\perp} \\ \mathbf{0}_3 \\ \omega \mathbf{e}_{0\perp} \end{pmatrix} \quad (9)$$

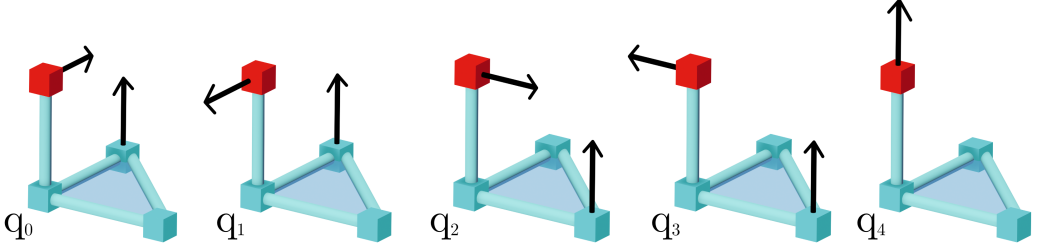


Fig. 3. Eigenvectors of $\frac{\partial^2 \Psi_l}{\partial \mathbf{c}^2}$. The \mathbf{q}_0 and \mathbf{q}_1 vectors respectively represent buckling and rotation in the direction of $\mathbf{e}_{1\perp}$, while \mathbf{q}_2 and \mathbf{q}_3 are analogous modes along $\mathbf{e}_{0\perp}$, and \mathbf{q}_4 is a purely compressive mode.

where $\mathbf{0}_3$ and $\mathbf{0}_{3 \times 3}$ respectively denote a vector-3 and 3×3 matrix of zeros, and

$$\omega = \frac{\lambda_{0,1,2,3}^l}{\left(\lambda_{0,1,2,3}^l - \frac{l}{\|\mathbf{e}_{2\perp}\|^2}\right)} \quad f(i, j) = \sqrt{1 + 4 \left(\frac{\|\mathbf{e}_{i\perp}\|}{\|\mathbf{e}_{j\perp}\|}\right)^2}. \quad (10)$$

Thus, while $\mathbf{H}_l \in \mathbb{R}^{9 \times 9}$, we can now see that it is rank-4, and contains a rank-5 nullspace.

These simple expressions are possible because of the orthogonality of our reduced frame. Without it, Eqns. 5-7 balloon in complexity, making an analytic eigensystem intractable. For more details on how we derive \mathbf{H}_l , see the supplement.

With the eigensystem (Eqns. 8 and 9) in hand, we can return to the generic problem of $\frac{\partial^2 \Psi_l}{\partial \mathbf{c}^2}$ (Eqn. 3). In that equation, the $\frac{\partial \Psi_l}{\partial l}$ term simply scales the eigenvalues of \mathbf{H}_l . Additionally,

$$\mathbf{g}_l = \frac{\partial l}{\partial \mathbf{c}} = \begin{pmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \frac{l}{\|\mathbf{e}_{2\perp}\|} \hat{\mathbf{e}}_{2\perp} \end{pmatrix}$$

is orthogonal to \mathbf{H}_l , because its sole entry $\mathbf{e}_{2\perp}$ is known to be orthogonal to the corresponding $\omega \mathbf{e}_{0\perp}$ or $\omega \mathbf{e}_{1\perp}$ terms in $\mathbf{q}_{0,1,2,3}^l$. Thus, \mathbf{g}_l must form a fifth eigenpair whose eigenvalue is $\frac{\partial^2 \Psi_l}{\partial l^2}$. The complete rank-5 eigensystem for generic collision energies is then:

$$\lambda_{0,1} = \frac{-l}{2 \|\mathbf{e}_{1\perp}\|^2} (1 \pm f(1, 2)) \frac{\partial \Psi_l}{\partial l} \quad \mathbf{q}_{0,1} = \begin{pmatrix} \mathbf{0}_3 \\ \mathbf{e}_{2\perp} \\ \omega \mathbf{e}_{1\perp} \end{pmatrix} \quad (11)$$

$$\lambda_{2,3} = \frac{-l}{2 \|\mathbf{e}_{0\perp}\|^2} (1 \pm f(0, 2)) \frac{\partial \Psi_l}{\partial l} \quad \mathbf{q}_{2,3} = \begin{pmatrix} \mathbf{e}_{2\perp} \\ \mathbf{0}_3 \\ \omega \mathbf{e}_{0\perp} \end{pmatrix} \quad (12)$$

$$\lambda_4 = \frac{\partial^2 \Psi_l}{\partial l^2} \quad \mathbf{q}_4 = \mathbf{g}_l. \quad (13)$$

4.2.1 Discussion. Physically, \mathbf{q}_0 , \mathbf{q}_2 , and \mathbf{q}_4 correspond to compression modes in the reduced frame, while \mathbf{q}_1 and \mathbf{q}_3 are rotation modes. Eigenvalue filtering can then be interpreted as disambiguating between two particular rotation and compression modes, along with filtering compression in the normal direction (Fig. 3).

4.3 Filtering in State Space

Our generic analysis can be applied to vertex-face collisions by introducing an appropriate change-of-basis. If we denote the stacked vector of vertex positions \mathbf{v}_i as $\mathbf{x} \in \mathbb{R}^{12}$, the force and stiffness densities can respectively be written using the chain rule:

$$\frac{\partial \Psi}{\partial \mathbf{x}} = \frac{\partial \mathbf{c}}{\partial \mathbf{x}}^\top \frac{\partial \Psi}{\partial \mathbf{c}} \quad (14)$$

$$\frac{\partial^2 \Psi}{\partial \mathbf{x}^2} = \frac{\partial \mathbf{c}}{\partial \mathbf{x}}^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \frac{\partial \mathbf{c}}{\partial \mathbf{x}} + \frac{\partial \Psi}{\partial \mathbf{c}} : \frac{\partial^2 \mathbf{c}}{\partial \mathbf{x}^2} \quad (15)$$

where $:$ is a tensor double contraction [Kim and Eberle 2022]. Our eigenanalysis of $\frac{\partial^2 \Psi}{\partial \mathbf{c}^2}$ now seems to be insufficient, because the double contraction term appears to add a difficult new term that must also be analyzed. However, the orthogonality properties of \mathbf{C} allow us to express everything in terms of $\frac{\partial^2 \Psi}{\partial \mathbf{c}^2}$.

By treating the orthogonalization scalars α , β , and γ as *though they were constant*, we decompose $\frac{\partial \mathbf{c}}{\partial \mathbf{x}}$ into the sum of a simple term $\left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_s$ and a remainder term $\left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_\Delta$. The gradient and Hessian can then be written in the following forms:

$$\frac{\partial \Psi}{\partial \mathbf{x}} = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_s^\top \frac{\partial \Psi}{\partial \mathbf{c}} \quad (16)$$

$$\frac{\partial^2 \Psi}{\partial \mathbf{x}^2} = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_s^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_s - \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_\Delta^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_\Delta \quad (17)$$

This transformation is shown in detail in Appendix B.1.

These expressions avoid the need to compute the unwieldy 3rd order tensor $\frac{\partial^2 \mathbf{c}}{\partial \mathbf{x}^2} \in \mathbb{R}^{9 \times 12 \times 12}$, and an analytic filtering strategy immediately becomes apparent. Equipped with the analytic eigensystem for $\frac{\partial^2 \Psi}{\partial \mathbf{c}^2}$, we can pin negative eigenmodes to zero in the first term in Eqn. 17, and pin the positive eigenmodes to zero for the second term, resulting in a PSD matrix.

4.3.1 Applying to Vertex-Face and Edge-Edge Energies. The general simplified expression for $\frac{\partial \mathbf{c}}{\partial \mathbf{x}}$ can be written as so:

$$\left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_s = \begin{pmatrix} \frac{\partial \mathbf{e}_0}{\partial \mathbf{x}}^\top \\ \frac{\partial \mathbf{e}_1}{\partial \mathbf{x}}^\top - \alpha \frac{\partial \mathbf{e}_0}{\partial \mathbf{x}}^\top \\ \frac{\partial \mathbf{e}_2}{\partial \mathbf{x}}^\top - \beta \frac{\partial \mathbf{e}_0}{\partial \mathbf{x}}^\top - \gamma \frac{\partial \mathbf{e}_1}{\partial \mathbf{x}}^\top \end{pmatrix} \quad (18)$$

$$\alpha = \frac{\mathbf{e}_0 \cdot \mathbf{e}_1}{\|\mathbf{e}_0\|^2} \quad \beta = \frac{\|\mathbf{e}_1\|^2 (\mathbf{e}_0 \cdot \mathbf{e}_2) - (\mathbf{e}_0 \cdot \mathbf{e}_1)(\mathbf{e}_1 \cdot \mathbf{e}_2)}{\|\mathbf{e}_0\|^2 \|\mathbf{e}_1\|^2 - (\mathbf{e}_0 \cdot \mathbf{e}_1)^2} \quad \gamma = \frac{\|\mathbf{e}_0\|^2 (\mathbf{e}_1 \cdot \mathbf{e}_2) - (\mathbf{e}_0 \cdot \mathbf{e}_1)(\mathbf{e}_0 \cdot \mathbf{e}_2)}{\|\mathbf{e}_0\|^2 \|\mathbf{e}_1\|^2 - (\mathbf{e}_0 \cdot \mathbf{e}_1)^2}$$

Evaluating for vertex face (vf) and edge-edge (ee) collision geometries (Fig. 2), we obtain:

$$\left(\frac{\partial \mathbf{c}_{\text{vf}}}{\partial \mathbf{x}}\right)_s = \begin{pmatrix} 0 & 1 & -1 & 0 \\ 0 & -\alpha & \alpha - 1 & 1 \\ 1 & -\beta & \beta + \gamma - 1 & -\gamma \end{pmatrix} \otimes \mathbf{I}_3 \quad \left(\frac{\partial \mathbf{c}_{\text{ee}}}{\partial \mathbf{x}}\right)_s = \begin{pmatrix} -1 & 1 & 0 & 0 \\ \alpha & -\alpha & -1 & 1 \\ \beta - 1 & -\beta & 1 + \gamma & -\gamma \end{pmatrix} \otimes \mathbf{I}_3 \quad (19)$$

Where \mathbf{I}_3 is a 3×3 identity matrix and \otimes is a Kronecker product.

In general, the remainder term is

$$\left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}}\right)_\Delta = - \begin{pmatrix} \mathbf{0}_{12}^\top \\ \mathbf{e}_0 \frac{\partial \alpha}{\partial \mathbf{x}}^\top \\ \mathbf{e}_0 \frac{\partial \beta}{\partial \mathbf{x}}^\top + \mathbf{e}_1 \frac{\partial \gamma}{\partial \mathbf{x}}^\top \end{pmatrix} \quad (20)$$

where $\mathbf{0}_{12}$ is a vector-12 of all zeros. Explicit evaluation for the vertex-face and edge-edge case are somewhat involved, so we have included their exact forms as Matlab functions. See the functions in `Projection/` to get $\frac{\partial \alpha}{\partial \mathbf{x}}$, $\frac{\partial \beta}{\partial \mathbf{x}}$, and $\frac{\partial \gamma}{\partial \mathbf{x}}$ for vertex-face and edge-edge collisions.

Replacing \mathbf{c} with \mathbf{c}_{vf} or \mathbf{c}_{ee} in Eqns. 16 and 17 yield the full gradients and Hessians.

4.4 Signed Length Energies

Once a penalty energy has been rewritten in terms of signed length l , Eqns. 11-13 can be used to obtain the analytic eigensystem of the Hessian for both the edge-edge and vertex-face case. Letting ϵ be the activation distance and k be a force constant, several common penalty energies can be rewritten into this form.

4.4.1 Quadratic Penalty. Inserting a stiff penalty spring at a small distance threshold prevents interpenetration, and can reduce the system sizes needed for constraint-based collision resolution [Bridson et al. 2002; Harmon et al. 2008]. The quadratic energy in terms of signed length l is:

$$\Psi_{\text{q}}(l) = \begin{cases} k(l - \epsilon)^2 & l < \epsilon \\ 0 & l \geq \epsilon \end{cases}. \quad (21)$$

Fully working out the analytic eigensystem, we have

$$\lambda_{0,1} = \frac{kl(\epsilon - l)}{\|\mathbf{e}_{1\perp}\|^2} (1 \pm f(1, 2)) \quad \mathbf{q}_{0,1} = \begin{pmatrix} \mathbf{0}_3 \\ \mathbf{e}_{2\perp} \\ \omega \mathbf{e}_{1\perp} \end{pmatrix} \quad (22)$$

$$\lambda_{2,3} = \frac{kl(\epsilon - l)}{\|\mathbf{e}_{0\perp}\|^2} (1 \pm f(0, 2)) \quad \mathbf{q}_{2,3} = \begin{pmatrix} \mathbf{e}_{2\perp} \\ \mathbf{0}_3 \\ \omega \mathbf{e}_{0\perp} \end{pmatrix} \quad (23)$$

$$\lambda_4 = 2k \quad \mathbf{q}_4 = \mathbf{g}_l \quad (24)$$

where ω and $f(i, j)$ are as defined in Eqn. 10. Here, we see that the fourth eigenmode corresponds precisely with simple harmonic motion in the direction of $\mathbf{e}_{2\perp}$ and is always positive.

Analyzing signs, we additionally note that $f(i, j)$ is always positive and greater than 1, meaning that one of $\lambda_{0,1}$ and $\lambda_{2,3}$ is always negative due to the multiplication by $(1 \pm f(i, j))$. As a result, one eigenmode of each pair always gets filtered. Furthermore, since $l < \epsilon$ is guaranteed, only the sign of l determines which vectors should be filtered between them.

We also make use of an analogous quadratic penalty energy using unsigned length Ψ_{uq} :

$$\Psi_{\text{uq}}(u) = \begin{cases} k(u \mp \epsilon)^2 & l < \epsilon \\ 0 & l \geq \epsilon \end{cases} \quad (25)$$

Where $+$ is chosen when interpenetrations are detected and $-$ is chosen otherwise. Eigenanalysis in this case comes from Eqns. 51 and 52, where we plug in derivatives with respect to u .

4.4.2 Adaptive Spring. Higher order versions of these energies have also been formulated. For example, Gast et al. [2015] use a cubic penalty energy to resolve collisions, modelled after a linearly varying spring constant,

$$\Psi_{\text{cubic}}(l) = \begin{cases} k(l - \epsilon)^2 \sqrt{(l - \epsilon)^2 + \delta} & l < \epsilon \\ 0 & l \geq \epsilon \end{cases} \quad (26)$$

where δ is a regularization constant to avoid singularities in the square root under differentiation.

By applying the general equations, the eigensystem is as follows:

$$\lambda_{0,1} = \frac{kl(\epsilon - l)(3(l - \epsilon)^2 + 2\delta)}{2 \|\mathbf{e}_{1\perp}\|^2 \sqrt{(l - \epsilon)^2 + \delta}} (1 \pm f(1, 2)) \quad \mathbf{q}_{0,1} = \begin{pmatrix} \mathbf{0}_3 \\ \mathbf{e}_{2\perp} \\ \omega \mathbf{e}_{1\perp} \end{pmatrix} \quad (27)$$

$$\lambda_{2,3} = \frac{kl(\epsilon - l)(3(l - \epsilon)^2 + 2\delta)}{2 \|\mathbf{e}_{0\perp}\|^2 \sqrt{(l - \epsilon)^2 + \delta}} (1 \pm f(0, 2)) \quad \mathbf{q}_{2,3} = \begin{pmatrix} \mathbf{e}_{2\perp} \\ \mathbf{0}_3 \\ \omega \mathbf{e}_{0\perp} \end{pmatrix} \quad (28)$$

$$\lambda_4 = \frac{k(6(l - \epsilon)^4 + 9\delta(l - \epsilon)^2 + 2\delta^2)}{((l - \epsilon)^2 + \delta)^{\frac{3}{2}}} \quad \mathbf{q}_4 = \mathbf{g}_l. \quad (29)$$

Looking closely at the signs, filtering between $\mathbf{q}_{0,1}$ and $\mathbf{q}_{2,3}$ once again comes down to the sign of l , while λ_4 is always positive.

4.4.3 Barrier Energies. IPC solvers use barrier energies to simulate a wide variety of collisions [Li et al. 2021]. These can be rewritten as:

$$\Psi_{\text{barrier}}(l) = \begin{cases} -(l - \epsilon)^2 \ln\left(\frac{l}{\epsilon}\right) & 0 < l < \epsilon \\ 0 & l \geq \epsilon. \end{cases} \quad (30)$$

The analytic eigensystem is then

$$\lambda_{0,1} = \frac{(\epsilon - l) \left(\epsilon - l - 2 \ln\left(\frac{l}{\epsilon}\right) \right)}{2 \|\mathbf{e}_{1\perp}\|^2} (1 \pm f(1, 2)) \quad \mathbf{q}_{0,1} = \begin{pmatrix} \mathbf{0}_3 \\ \mathbf{e}_{2\perp} \\ \omega \mathbf{e}_{1\perp} \end{pmatrix} \quad (31)$$

$$\lambda_{2,3} = \frac{(\epsilon - l) \left(\epsilon - l - 2 \ln\left(\frac{l}{\epsilon}\right) \right)}{2 \|\mathbf{e}_{0\perp}\|^2} (1 \pm f(0, 2)) \quad \mathbf{q}_{2,3} = \begin{pmatrix} \mathbf{e}_{2\perp} \\ \mathbf{0}_3 \\ \omega \mathbf{e}_{0\perp} \end{pmatrix} \quad (32)$$

$$\lambda_4 = \frac{(\epsilon - l)(\epsilon + 3l)}{l^2} - 2 \ln\left(\frac{l}{\epsilon}\right) \quad \mathbf{q}_4 = \mathbf{g}_l \quad (33)$$

Interestingly, λ_0 , λ_2 , and λ_4 are always positive while the other eigenvalues are always negative, meaning that filtering the barrier energy does not require any sign checks. However, since barrier energies are undefined when signed length is negative, they become unsuitable for solvers that are expected to deal with unavoidable interpenetration, as with the scripted intersection of kinematic objects [Baraff et al. 2003; Daviet 2020].

4.5 Hybrid Energies

McAdams et al. [2011] characterize stiff springs using an m parameter to control the anisotropy of the response, observing that higher anisotropy enables more tangential sliding.

Reformulating in terms of signed and unsigned length, we see that this tangential sliding corresponds to the difference in eigensystems between signed and unsigned length Hessians. For signed length, any tangential motion comes with a corresponding motion in one of the face vertices (Fig. 3). For unsigned length, pure tangential motion make up two of the three eigenvectors (Fig. 8).

If we characterize hybrid length as $h = l \cdot m + (1 - m) \cdot u$, with $0 \leq m \leq 1$ as the anisotropy parameter, the mixed energy becomes

$$\Psi_{\text{McAdams}}(h) = \begin{cases} k(h - \epsilon)^2 & h < \epsilon \\ 0 & h \geq \epsilon \end{cases} \quad (34)$$

and the Hessian becomes:

$$\frac{\partial^2 \Psi}{\partial \mathbf{c}^2} = \left(\frac{\partial^2 \Psi}{\partial l^2} + \frac{\partial^2 \Psi}{\partial u^2} + 2 \frac{\partial^2 \Psi}{\partial l \partial u} \right) \mathbf{g}_l \mathbf{g}_l^\top + \frac{\partial \Psi}{\partial l} \mathbf{H}_l + \frac{\partial \Psi}{\partial u} \mathbf{H}_u \quad (35)$$

where $\mathbf{H}_u = \frac{\partial^2 u}{\partial \mathbf{c}^2}$. Our analysis gives us the full eigensystem for the sum of the first two terms with Eqns. 11-13, with the slight modification of $\lambda_4 = \left(\frac{\partial^2 \Psi}{\partial l^2} + \frac{\partial^2 \Psi}{\partial u^2} + 2 \frac{\partial^2 \Psi}{\partial l \partial u} \right)$. Invariant analysis in Appendix A further reveals the eigenpairs of the final term (Eqn. 52). These terms can thus be filtered separately and then summed, yielding an analytic filtering strategy.

5 RESULTS

5.1 Matlab Verification

To ensure that our analyses are correct, we have provided Matlab implementations of key formulas, along with numerical tests. `Verify_General_Length.m` verifies the eigensystems for \mathbf{H}_l (Eqns. 8 and 9) and \mathbf{H}_u (Eqns. 51 and 52) by generating random reduced frames and comparing the analytic outer-product decomposition with finite differences.

We also provide Hessian eigensystems for Ψ_{lq} and Ψ_{uq} with the same numerical test, along with numerical verification for Eqns. 16 and 17 for each case of signed/unsigned length and vertex-face/edge-edge collision (see `Verify_[lq/uq]_Collision_[vf/ee].m`).

5.2 Test Scenes

5.2.1 Cable Stretching. Beginning with an implicit strand solver [Bergou et al. 2010], we pull two collections of cables against each other by moving their endpoints in opposite directions and resolve collisions with various penalty energies. We used an implicit solver [Baraff and Witkin 1998] with a large time step of $\Delta t = 1/30$. To better view the unadorned stability behavior of the collision energies, no line search was used.

We initialized the cables with a Young's modulus $E = 10^4$ Pa and a Poisson's ratio $\nu = 0.36$. All quadratic penalties had the same spring constant $k = 10$ N/m and distance envelope $\epsilon = 0.09$ m. We set a gravitational force to induce an acceleration of 9.81 m/s². To prevent excessive oscillations, we also incorporated Rayleigh damping with a factor $\beta = 0.006$.

We see that Gauss-Newton approximations of the Hessian discards important information. When equipped with Ψ_{uq} , the jittering in Gauss-Newton causes cables under tension to unrealistically pop through (Fig. 1). Analytically clamping the same scene correctly resolves the contact.

Stretching the cables again with Ψ_{lq} , Gauss-Newton persistently contains jittering artifacts. The jittering diminishes when using our approach, but still remains present. However, these artifacts arise from the collision energy itself, not the filtering strategy. When we use a hybrid energy along the lines of Cheng et al. [2022], which falls back to Ψ_{uq} when the closest point on the edge corresponds to an endpoint, we obtain less jittery results. Simply passing in the unfiltered Hessian with the same scene parameters results in an explosion immediately after the first few contacts.

The energies exhibited qualitative differences that correspond to their formulations. Unsigned penalty forces that do not explicitly cancel off tangential sliding were "stickier" than their signed counterparts (Fig. 4).

5.2.2 Fabric Stretching. We ran a test similar to the cable example with an implicit cloth solver [Baraff and Witkin 1998], $\Delta t = 1/40$, and no line search, pulling two pieces of cloth against each other (Fig. 5). We initialized the fabrics with an ARAP membrane stretching energy [Kim et al. 2019] with $\mu_{\text{stretch}} = 10$ and a dihedral bending energy with $\mu_{\text{bend}} = 0.01$. All quadratic penalties

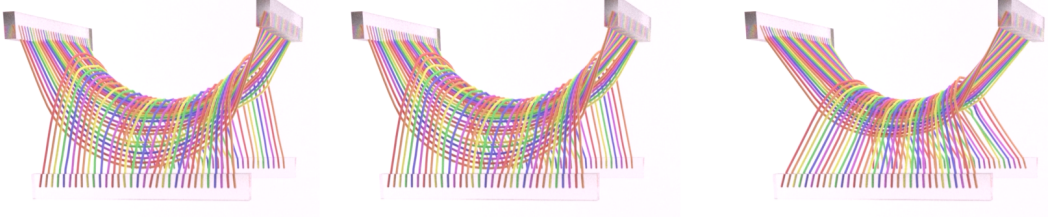


Fig. 4. Frame 45 of the cable stretching simulation with varying collision energies. Unsigned length (left) and hybrid energies (middle) exhibit "stickier" behavior, while the signed length spring penalty (right) is quicker to clump towards the middle.

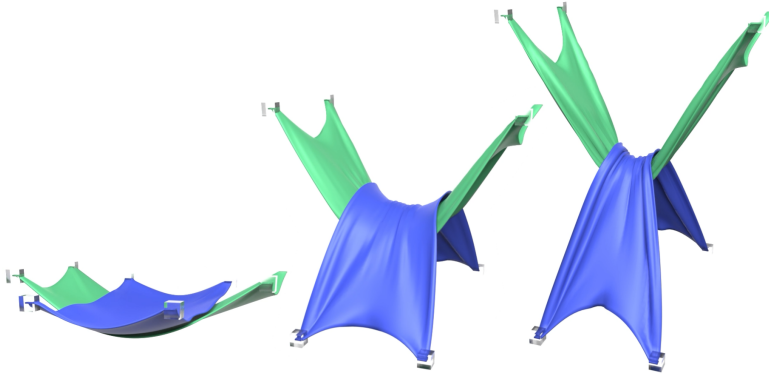


Fig. 5. Fabric stretching simulation using Ψ_{lq} for vertex-face, and Ψ_{uq} for edge-edge collisions.

were set with a spring constant $k = 4000$ N/m and a distance envelope $\epsilon = 0.02$ m. Gravity was set to be 1 m/s^2 . To prevent excessive oscillations, we used Rayleigh damping with $\beta = 1$.

Picking between Ψ_{lq} or Ψ_{uq} for vertex-face and edge-edge collisions results, we simulated two cloth pulling tests. As seen in the video, Gauss-Newton approximation results in jittery cloth across the board, while our analytic clamping smoothly settles.

5.2.3 Hammock Drop. We evaluated the qualitative behavior of a variety of energy combinations in a more complex collision scenario (Fig. 6) where several patches of cloth are dropped onto a hammock and then slide off. The differences in "stickiness" between energies is visible in the way that the cloth bunches and the rates at which different patches fall away. The simulation parameters were the same as §5.2.2, except that Rayleigh damping was set to zero.

5.2.4 Volumetric Meshes. We showcase the necessity of negative eigenvalue filtering in a volumetric example, where we lower a set of volumetric rods onto a collection of thicker rods with a time step $\Delta t = 1/40$ s and no line search.

We used a Stable Neo-Hookean distortion energy [Smith et al. 2018] with $E = 10^4$ Pa and $\nu = 0.45$. We set gravity to 1 m/s^2 as well. For collision response, we used Ψ_{lq} and Ψ_{uq} for the vertex-face and edge-edge collision penalties, respectively. Both were initialized with a spring constant $k = 1200$ N/m and a distance envelope $\epsilon = 0.02$ m. Rayleigh damping was at $\beta = 0.25$.

Disabling eigenvalue filtering in Ψ_{lq} and Ψ_{uq} results in indefinite systems that lead to obvious artifacts around frame 170 and a subsequent explosion (Fig. 7). All the other unfiltered scenes exhibited similar behavior.

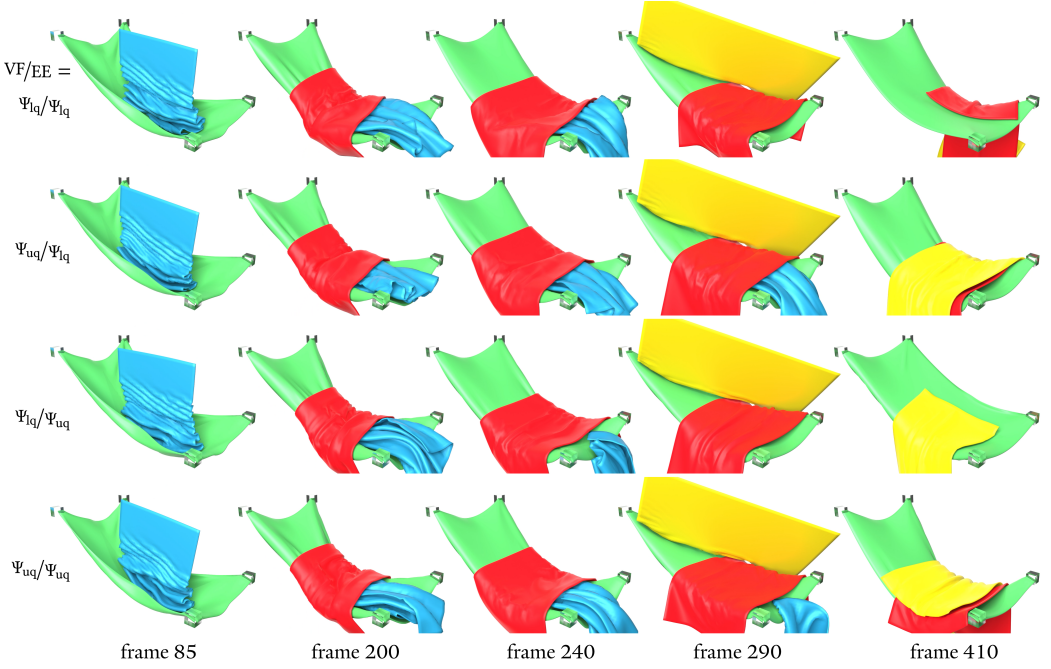


Fig. 6. Dropping pieces of cloth on a hammock with different collision energies yields different behaviors. The specific vertex-face (VF) and edge-edge (EE) energies used are listed along the left. From left to right, the “stickiness” that the collision energy gives the red cloth determines whether the blue cloth falls away at frame 240. Similarly, the stickiness determines whether the red cloth remains by frame 410 (right). We analytically filtered all of the collision energies, and ran the simulations at $\Delta t = 1/40$ s

These observations are in line with previous work, which stress the necessity of eigenvalue filtering for PSD-projection not only for collisions [Li et al. 2020], but also several other deformation energies [Kim et al. 2019; Lin et al. 2022; Smith et al. 2019; Teran et al. 2005].

5.3 Performance

We initialized 10^7 random vertex arrangements and compared the performance of our analytic approach with the numerical eigensolver solver in Eigen [Guennebaud et al. 2010] and Gauss-Newton approximation. Our technique is $4.5\times - 7.2\times$ faster than the numerical eigensolver, and requires only 8% – 70% more computation than the single outer product for Gauss-Newton. As shown in Fig. 1, this modest overhead significantly reduces jittering and pop-through.

6 CONCLUSIONS AND FUTURE WORK

We have presented an eigenanalysis of energies used for vertex-face and edge-edge collisions in implicit solvers. By using analytic methods to filter negative eigenvalues, we obtain faster matrix assembly times and more robust simulations. The analytic eigenpairs represent distinct collision modes, and clarify how filtering certain eigenvalues correspond to discounting specific responses. We could extend our analysis to vertex-vertex and vertex-edge, mollified energies in near-parallel, and other degenerate cases, allowing for analytic eigensystems in more hybridized penalty energies. Potential-based frictional contact [Li et al. 2020] and damping [Brown et al. 2018]

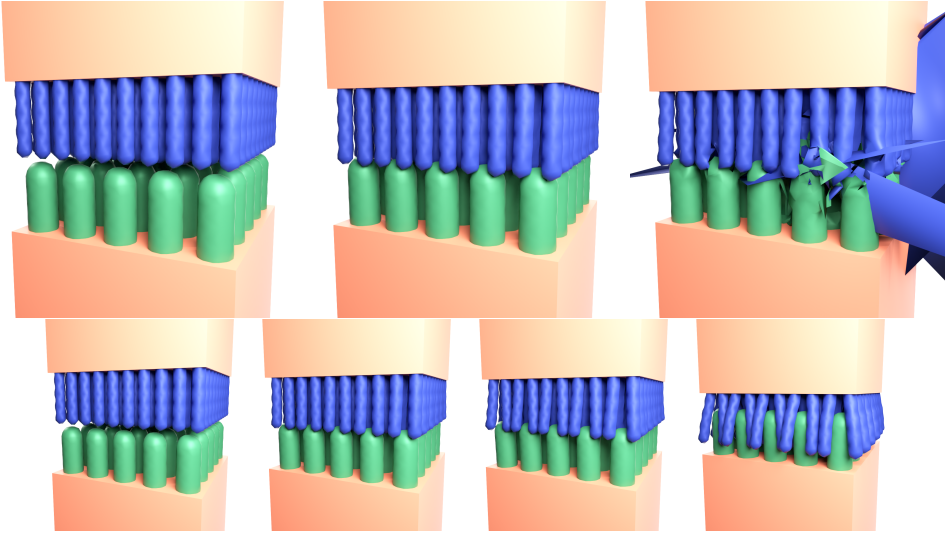


Fig. 7. Selected frames of a volumetric object collision scene using Ψ_{lq} for vertex-face, and Ψ_{uq} for edge-edge collisions. Unfiltered simulation explodes at frame 171 (top), while filtered simulation compresses further to frame 400 (bottom).

Table 1. Hessian filtering timings for 10^7 randomized vertex positions. Most energies showed a $\sim 6\times$ speedup, and were competitive with the faster but less stable Gauss-Newton (G-N) strategy. All timings are in seconds.

Energy	Numeric	Ours	G-N	Numeric/Ours
Vertex-face springs				
Signed Length	88.48s	12.31s	11.12s	7.19×
Unsigned Length	83.68s	18.55s	10.93s	4.51×
Edge-edge springs				
Signed Length	84.30s	13.01s	12.04s	6.48×
Unsigned Length	86.45s	11.98s	8.78s	7.22×
Hybrid	85.16s	13.52s	12.29s	6.30×

are other interesting directions, since the energies we analyzed only penalize interpenetration, not relative sliding motion.

ACKNOWLEDGMENTS

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

REFERENCES

- 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>
- David Baraff, Andrew Witkin, and Michael Kass. 2003. Untangling cloth. In *ACM SIGGRAPH 2003 Papers (SIGGRAPH '03)*. Association for Computing Machinery, New York, NY, USA, 862–870. <https://doi.org/10.1145/1201775.882357>

- Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Transactions on Graphics* 29, 4 (July 2010), 116:1–116:10. <https://doi.org/10.1145/1778765.1778853>
- Javier Bonet and Richard D. Wood. 2008. *Nonlinear Continuum Mechanics for Finite Element Analysis* (2 ed.). Cambridge University Press. <https://doi.org/10.1017/CBO9780511755446>
- Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02)*. Association for Computing Machinery, New York, NY, USA, 594–603. <https://doi.org/10.1145/566570.566623>
- 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 (SCA '03)*. Eurographics Association, Goslar, DEU, 28–36.
- George E. Brown, Matthew Overby, Zahra Forootaninia, and Rahul Narain. 2018. Accurate Dissipative Forces in Optimization Integrators. *ACM Trans. Graph.* 37, 6, Article 282 (dec 2018), 14 pages.
- James R. Bunch, Christopher P. Nielsen, and Danny C. Sorensen. 1978. Rank-one modification of the symmetric eigenproblem. *Numer. Math.* 31, 1 (March 1978), 31–48. <https://doi.org/10.1007/BF01396012>
- Johnny T. Chang, Jingyi Jin, and Yizhou Yu. 2002. A practical model for hair mutual interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '02)*. Association for Computing Machinery, New York, NY, USA, 73–80. <https://doi.org/10.1145/545261.545273>
- Yunuo Chen, Minchen Li, Lei Lan, Hao Su, Yin Yang, and Chenfanfu Jiang. 2022. A unified newton barrier method for multibody dynamics. *ACM Transactions on Graphics* 41, 4 (July 2022), 66:1–66:14. <https://doi.org/10.1145/3528223.3530076>
- Yuxuan Cheng, John D. Treado, Benjamin F. Lonial, Piotr Haddas, Eric R. Weeks, Mark D. Shattuck, and Corey S. O'Hern. 2022. Hopper flows of deformable particles. *Soft Matter* 18, 42 (Nov. 2022), 8071–8086. <https://doi.org/10.1039/D2SM01079H>
Publisher: The Royal Society of Chemistry.
- Byoungwon Choe, Min Gyu Choi, and Hyeong-Seok Ko. 2005. Simulating complex hair with robust collision handling. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '05)*. Association for Computing Machinery, New York, NY, USA, 153–160. <https://doi.org/10.1145/1073368.1073389>
- Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but responsive cloth. *ACM Trans. Graph.* 21, 3 (2002), 604–611.
- Gilles Daviet. 2020. Simple and scalable frictional contacts for thin nodal objects. *ACM Transactions on Graphics* 39, 4 (Aug. 2020). <https://doi.org/10.1145/3386569.3392439>
- Yun (Raymond) Fei, Henrique Teles Maia, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2017. A multi-scale model for simulating liquid-hair interactions. *ACM Transactions on Graphics* 36, 4 (July 2017), 56:1–56:17. <https://doi.org/10.1145/3072959.3073630>
- Jorge Gascón, Javier S. Zurdo, and Miguel A. Otaduy. 2010. Constraint-based simulation of adhesive contact. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10)*. Eurographics Association, Goslar, DEU, 39–44.
- Theodore F Gast, Craig Schroeder, Alexey Stomakhin, Chenfanfu Jiang, and Joseph M Teran. 2015. Optimization integrator for large time steps. *IEEE transactions on visualization and computer graphics* 21, 10 (2015), 1103–1115.
- Gene H Golub and Charles F Van Loan. 2013. *Matrix computations*. JHU press.
- Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2008. Robust treatment of simultaneous collisions. *ACM Transactions on Graphics* 27, 3 (Aug. 2008), 1–4. <https://doi.org/10.1145/1360612.1360622>
- Chenfanfu Jiang, Theodore Gast, and Joseph Teran. 2017. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14.
- Stéphane Jimenez and Annie Luciani. 1993. Animation of Interacting Objects with Collisions and Prolonged Contacts. In *Modeling in Computer Graphics (IFIP Series on Computer Graphics)*, Bianca Falcidieno and Tosiya L. Kunii (Eds.). Springer, Berlin, Heidelberg, 129–141. https://doi.org/10.1007/978-3-642-78114-8_8
- Theodore Kim. 2020. A finite element formulation of Baraff-Witkin cloth. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '20)*. Eurographics Association, Goslar, DEU, 1–9. <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 Transactions on Graphics* 38, 4 (July 2019), 69:1–69:15. <https://doi.org/10.1145/3306346.3323014>
- Theodore Kim and David Eberle. 2022. Dynamic deformables: implementation and production practicalities (now with code!). In *ACM SIGGRAPH 2022 Courses (SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, 1–259. <https://doi.org/10.1145/3532720.3535628>
- Lei Lan, Danny M. Kaufman, Minchen Li, Chenfanfu Jiang, and Yin Yang. 2022. Affine body dynamics: fast, stable and intersection-free simulation of stiff materials. *ACM Transactions on Graphics* 41, 4 (July 2022), 67:1–67:14. <https://doi.org/10.1145/3528223.3530064>
- David IW Levin, Joshua Litven, Garrett L Jones, Shinjiro Sueda, and Dinesh K Pai. 2011. Eulerian solid simulation with contact. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 1–10.

- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental potential contact: intersection-and inversion-free, large-deformation dynamics. *ACM Transactions on Graphics* 39, 4 (Aug. 2020), 49:49:1–49:49:20. <https://doi.org/10.1145/3386569.3392425>
- Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional incremental potential contact. *ACM Transactions on Graphics* 40, 4 (July 2021), 170:1–170:24. <https://doi.org/10.1145/3450626.3459767>
- Huancheng Lin, Floyd M. Chitalu, and Taku Komura. 2022. Isotropic ARAP Energy Using Cauchy-Green Invariants. *ACM Transactions on Graphics* 41, 6 (Nov. 2022), 275:1–275:14. <https://doi.org/10.1145/3550454.3555507>
- Wei Chin Lin, Wei-Kai Liao, and Chao-Hua Lee. 2011. Simulating and rendering wet hair. In *SIGGRAPH Asia 2011 Posters (SA '11)*. Association for Computing Machinery, New York, NY, USA, 1. <https://doi.org/10.1145/2073304.2073348>
- Aleka McAdams, Andrew Selle, Kelly Ward, Eftychios Sifakis, and Joseph Teran. 2009. Detail preserving continuum simulation of straight hair. *ACM Transactions on Graphics* 28, 3 (July 2009), 62:1–62:6. <https://doi.org/10.1145/1531326.1531368>
- Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient elasticity for character skinning with contact and collisions. In *ACM SIGGRAPH 2011 papers*. ACM, Vancouver British Columbia Canada, 1–12. <https://doi.org/10.1145/1964921.1964932>
- Dominik L. Michels, J. Paul T. Mueller, and Gerrit A. Sobottka. 2015. A physically based approach to the accurate simulation of stiff fibers and stiff fiber meshes. *Computers & Graphics* 53 (Dec. 2015), 136–146. <https://doi.org/10.1016/j.cag.2015.10.001>
- Matthias Müller, Nuttapon Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air meshes for robust collision handling. *ACM Transactions on Graphics* 34, 4 (July 2015), 133:1–133:9. <https://doi.org/10.1145/2766907>
- Julian Panetta. 2020. Analytic eigensystems for isotropic membrane energies. *arXiv preprint arXiv:2008.10698* (2020).
- Graham Rhodes. 2001. Fast, robust intersection of 3D line segments. *Game Programming Gems 2* (2001), 191–204.
- Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. 2013. Locally injective mappings. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 125–135.
- Andrew Selle, Michael Lentine, and Ronald Fedkiw. 2008. A mass spring model for hair simulation. *ACM Transactions on Graphics* 27, 3 (Aug. 2008), 1–11. <https://doi.org/10.1145/1360612.1360663>
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Transactions on Graphics* 37, 2 (March 2018), 12:1–12:15. <https://doi.org/10.1145/3180491>
- Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Transactions on Graphics* 38, 1 (Feb. 2019), 3:1–3:15. <https://doi.org/10.1145/3241041>
- Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, Vol. 4. 109–116.
- J. Spillmann and M. Teschner. 2007. 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 (SCA '07)*. Eurographics Association, Goslar, DEU, 63–72.
- Yun Teng, David IW Levin, and Theodore Kim. 2016. Eulerian solid-fluid coupling. *ACM Trans. Graph.* 35, 6 (2016), 1–8.
- Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust quasistatic finite elements and flesh simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '05)*. Association for Computing Machinery, New York, NY, USA, 181–190. <https://doi.org/10.1145/1073368.1073394>
- Kelly Ward, Nico Galoppo, and Ming C Lin. 2004. Modeling hair influenced by water and styling products. In *International Conference on Computer Animation and Social Agents (CASA)*, Vol. 3.
- J. Zehnder, S. Coros, and B. Thomaszewski. 2021. SGN: Sparse Gauss-Newton for Accelerated Sensitivity Analysis. *ACM Trans. Graph.* 41, 1 (2021), 1–10.

A INVARIANT-BASED ANALYSIS

The unsigned length u and signed length l of a point translated $\mathbf{e}_{2\perp}$ from the plane whose normal is in the direction of $\mathbf{e}_{0\perp} \times \mathbf{e}_{1\perp}$ are:

$$l = \frac{\mathbf{e}_{2\perp} \cdot (\mathbf{e}_{0\perp} \times \mathbf{e}_{1\perp})}{\|\mathbf{e}_{0\perp} \times \mathbf{e}_{1\perp}\|} \quad (36)$$

$$u = \|\mathbf{e}_{2\perp}\| \quad (37)$$

A.1 Invariant Formulation

The reduced frame \mathbf{C} can be viewed as a deformation gradient for collisions. From this view, we can rewrite signed and unsigned length, using the anisotropic invariants from Kim et al. [2019]:

$$l = \frac{I_3}{\sqrt{I_5(\hat{\mathbf{x}})I_5(\hat{\mathbf{y}}) - I_6(\hat{\mathbf{x}}, \hat{\mathbf{y}})^2}} \quad (38)$$

$$u = \sqrt{I_5(\hat{\mathbf{z}})} \quad (39)$$

where $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ form the canonical orthonormal basis, and

$$I_3 = \det(\mathbf{C}) \quad I_5(\mathbf{v}) = \|\mathbf{C}\mathbf{v}\|^2 \quad I_6(\mathbf{u}, \mathbf{v}) = (\mathbf{C}\mathbf{u}) \cdot (\mathbf{C}\mathbf{v}) \quad (40)$$

A.2 Unsigned Length Energies Are Special Cases

The eigenanalysis from Kim et al. [2019] can be applied to any energy of the form $\Psi_u(I_5(\mathbf{v}))$ where \mathbf{v} is kept constant. Energies that are functions of only u fit these requirements, since $\Psi_u(u) = \Psi_u(\sqrt{I_5(\hat{\mathbf{z}})})$. The eigenpairs of $\frac{\partial^2 \Psi_u}{\partial \mathbf{c}^2}$ are then:

$$\lambda_0 = 2 \left(\frac{\partial \Psi_u}{\partial I_5} + 2I_5 \frac{\partial^2 \Psi_u}{\partial I_5^2} \right) \quad \mathbf{v}_0 = \frac{1}{\sqrt{I_5}} \text{Vec}(\mathbf{C}\hat{\mathbf{z}}\hat{\mathbf{z}}^\top) \quad (41)$$

$$\lambda_{1,2} = 2 \frac{\partial \Psi_u}{\partial I_5} \quad \mathbf{v}_{1,2} \in \text{Span}(\mathbf{v}_{T_1}, \mathbf{v}_{T_2}) \quad (42)$$

For \mathbf{v}_{T_1} and \mathbf{v}_{T_2} , we use the singular value decomposition $\mathbf{C} = \mathbf{U}\Sigma\mathbf{V}^\top$ and

$$\mathbf{T}_x = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \quad \mathbf{T}_y = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \mathbf{T}_z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (43)$$

to obtain

$$\mathbf{v}_{T_1} = \text{Vec}(\mathbf{U}\mathbf{T}_x\Sigma\mathbf{V}^\top\hat{\mathbf{z}}\hat{\mathbf{z}}^\top) \quad (44)$$

$$\mathbf{v}_{T_2} = \text{Vec}((\sigma_y\hat{a}_y\mathbf{U}\mathbf{T}_z - \sigma_z\hat{a}_z\mathbf{U}\mathbf{T}_y)\Sigma\mathbf{V}^\top\hat{\mathbf{z}}\hat{\mathbf{z}}^\top) \quad (45)$$

where $\sigma_{y,z}$ are singular values from Σ , and $\hat{a}_{y,z}$ are entries of $\mathbf{V}^\top\hat{\mathbf{z}}$.

Rewriting in terms of u and \mathbf{C} reveals further structure. Observing that $u = \sqrt{I_5}$ and $\frac{\partial}{\partial I_5} = \frac{1}{2u} \frac{\partial}{\partial u}$, we have

$$\lambda_0 = 2 \left(\frac{\partial \Psi_u}{\partial I_5} + 2I_5 \frac{\partial^2 \Psi_u}{\partial I_5^2} \right) \quad (46)$$

$$= \frac{1}{u} \frac{\partial \Psi_u}{\partial u} + 2u \frac{\partial}{\partial u} \left(\frac{1}{2u} \frac{\partial \Psi_u}{\partial u} \right) \quad (47)$$

$$= \frac{1}{u} \frac{\partial \Psi_u}{\partial u} + 2u \left(\frac{-1}{2u^2} \frac{\partial \Psi_u}{\partial u} + \frac{1}{2u} \frac{\partial^2 \Psi_u}{\partial u^2} \right) = \frac{\partial^2 \Psi_u}{\partial u^2} \quad (48)$$

$\lambda_{1,2}$ simplify similarly. As for \mathbf{v}_{T_1, T_2} , the reduced frame consists of orthogonal vectors, so we take \mathbf{V} as the identity, with \mathbf{U} and Σ as

$$\mathbf{U} = (\hat{\mathbf{e}}_{0\perp} \quad \hat{\mathbf{e}}_{1\perp} \quad \hat{\mathbf{e}}_{2\perp}) \quad (49)$$

$$\Sigma = \begin{pmatrix} \|\mathbf{e}_{0\perp}\| & 0 & 0 \\ 0 & \|\mathbf{e}_{1\perp}\| & 0 \\ 0 & 0 & \|\mathbf{e}_{2\perp}\| \end{pmatrix} \quad (50)$$

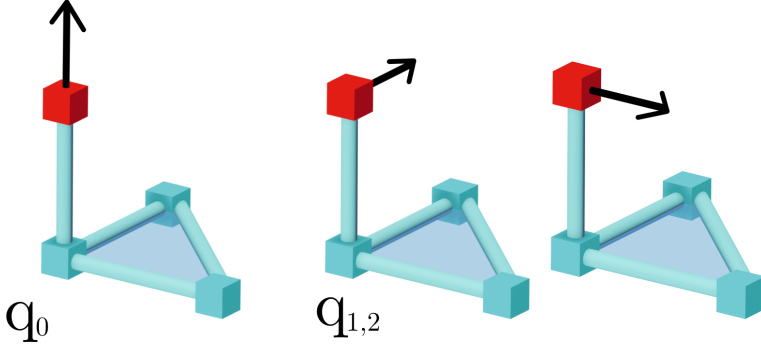


Fig. 8. Eigenvectors of Ψ_u : \mathbf{q}_0 represents compression, while $\mathbf{q}_{1,2}$ represent sliding in the parallel plane.

Plugging these into \mathbf{v}_{T_1} and \mathbf{v}_{T_2} yields simpler expressions for the eigenpairs of unsigned length energies:

$$\lambda_0 = \frac{\partial^2 \Psi_u}{\partial u^2} \quad \mathbf{v}_0 = \mathbf{g}_u = \begin{pmatrix} 0 \\ 0 \\ \hat{\mathbf{e}}_{2\perp} \end{pmatrix} \quad (51)$$

$$\lambda_{1,2} = \frac{1}{u} \frac{\partial \Psi_u}{\partial u} \quad \mathbf{v}_{1,2} \in \text{Span} \left\{ \begin{pmatrix} 0 \\ 0 \\ \mathbf{e}_{0\perp} \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ \mathbf{e}_{1\perp} \end{pmatrix} \right\} \quad (52)$$

Physically, \mathbf{v}_0 represents spring compression along the normal, while $\mathbf{v}_{1,2}$ represent buckling in the orthogonal directions (Fig. 8).

In practice, we only apply Eqns. 51-52 for stiff springs using unsigned length in the vertex-face collision case, while we use Eqns. 41-42 for the edge-edge case. This is because a simpler non-orthogonal frame unique to the edge-edge case exists:

$$\mathbf{C}_{\text{unsigned ee}} = \left(\mathbf{e}_0 \mid \mathbf{e}_1 \mid \mathbf{e}_{ab} \right) \quad (53)$$

where $\mathbf{e}_{ab} = \mathbf{e}_2 - a\mathbf{e}_0 + b\mathbf{e}_1$ is the vector of least distance between \mathbf{e}_0 and \mathbf{e}_1 , and a and b are clamped and taken as constant, matching Kim and Eberle [2022]’s implementation. Using this frame, $\frac{\partial^2 \mathbf{c}}{\partial \mathbf{x}^2}$ is zero, circumventing the need for any extra term considerations.

A.3 Invariant Analysis Fails for Signed Length

Under invariant analysis, penalty functions of signed length $\Psi(l)$ admit derivatives with respect to I_3 , I_5 , and I_6 . For energy Hessians, this yields the expression

$$\frac{\partial^2 \Psi}{\partial \mathbf{c}^2} = \sum_{N \in \mathcal{S}} \frac{\partial \Psi}{\partial N} \mathbf{H}_N + \sum_{N, M \in \mathcal{S}} \frac{\partial^2 \Psi}{\partial M \partial N} \mathbf{g}_M \mathbf{g}_N^T \quad (54)$$

where $\mathcal{S} = \{I_3, I_5(\hat{\mathbf{x}}), I_5(\hat{\mathbf{y}}), I_6\}$, $\mathbf{H}_N = \frac{\partial^2 N}{\partial \mathbf{c}^2}$, and $\mathbf{g}_M = \frac{\partial N}{\partial \mathbf{c}}$. Prior analysis [Kim et al. 2019] assumed that $\frac{\partial^2 \Psi}{\partial M \partial N} = 0$ when $M \neq N$. This does not hold for signed length, and every nonzero mixed term makes a rank-two contribution to overall system.

In some cases, the Bunch-Nielsen-Sorensen formulas [Bunch et al. 1978] can yield analytically solvable polynomials. In practice, many rank-two mixed terms appear, and repeatedly applying rank-one updates results in intractably high-degree polynomials. Taking derivatives with respect to l , as we do in §4, yields a clearer path.

B CORRECTIVE TERM SIMPLIFICATIONS

B.1 Gradient Simplifications

Going from Eqn. 14 to 16 involves breaking down $\frac{\partial \mathbf{c}}{\partial \mathbf{x}}$ into a simplified $(\frac{\partial \mathbf{c}}{\partial \mathbf{x}})_s$ and remainder $(\frac{\partial \mathbf{c}}{\partial \mathbf{x}})_\Delta$ component and observing that $\frac{\partial \Psi}{\partial \mathbf{c}}$ lies in the nullspace of $(\frac{\partial \mathbf{c}}{\partial \mathbf{x}})_\Delta$.

$$\frac{\partial \Psi}{\partial \mathbf{x}} = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)^\top \frac{\partial \Psi}{\partial \mathbf{c}} = \left[\left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s^\top + \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta^\top \right] \frac{\partial \Psi}{\partial \mathbf{c}} = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s^\top \frac{\partial \Psi}{\partial \mathbf{c}} \quad (55)$$

To see how $\frac{\partial \Psi}{\partial \mathbf{c}}$ lies in the nullspace, recall from Eqn. 20 that

$$\left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta^\top = - \left(\mathbf{0}_3^\top \quad \frac{\partial \alpha}{\partial \mathbf{x}} \mathbf{e}_0^\top \quad \frac{\partial \beta}{\partial \mathbf{x}} \mathbf{e}_0^\top + \frac{\partial \gamma}{\partial \mathbf{x}} \mathbf{e}_1^\top \right) \quad (56)$$

while the chain rule yields:

$$\frac{\partial \Psi}{\partial \mathbf{c}} = \frac{\partial \Psi}{\partial l} \frac{\partial l}{\partial \mathbf{c}} = \frac{\partial \Psi}{\partial l} \begin{pmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \frac{l}{\|\mathbf{e}_{2\perp}\|} \hat{\mathbf{e}}_{2\perp} \end{pmatrix}. \quad (57)$$

Since $\mathbf{e}_{2\perp}$ is orthogonal to both \mathbf{e}_0 and \mathbf{e}_1 , $(\frac{\partial \mathbf{c}}{\partial \mathbf{x}})_\Delta^\top \frac{\partial \Psi}{\partial \mathbf{c}}$ equals zero and contributes nothing to the overall gradient.

B.2 Energy Hessian Corrections

To go from Eqn. 15 to 17, we start with:

$$\frac{\partial^2 \Psi}{\partial \mathbf{x}^2} = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \frac{\partial \mathbf{c}}{\partial \mathbf{x}} + \left(\frac{\partial \Psi}{\partial \mathbf{c}} \right)^\top : \frac{\partial^2 \mathbf{c}}{\partial \mathbf{x}^2} \quad (58)$$

where $:$ is a double contraction. Writing out the Hessian in terms of the simplified and remainder terms yields

$$\frac{\partial^2 \Psi}{\partial \mathbf{x}^2} = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s + \mathbf{A}_1 + \mathbf{A}_2 \quad (59)$$

$$\mathbf{A}_1 = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta \quad (60)$$

$$\mathbf{A}_2 = \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s + \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s^\top \frac{\partial^2 \Psi}{\partial \mathbf{c}^2} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta + \frac{\partial \Psi}{\partial \mathbf{c}} : \frac{\partial^2 \mathbf{c}}{\partial \mathbf{x}^2}$$

Using Eqn. 3 to substitute $\frac{\partial^2 \Psi}{\partial \mathbf{c}^2}$ and noting that $(\frac{\partial \mathbf{c}}{\partial \mathbf{x}})_\Delta^\top \mathbf{g}_l = \mathbf{0}_{12}$ further reduces \mathbf{A}_1 and \mathbf{A}_2 :

$$\mathbf{A}_1 = \frac{\partial \Psi}{\partial l} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta^\top \mathbf{H}_l \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta \quad (61)$$

$$\mathbf{A}_2 = \frac{\partial \Psi}{\partial l} \left(\left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta^\top \mathbf{H}_l \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s + \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_s^\top \mathbf{H}_l \left(\frac{\partial \mathbf{c}}{\partial \mathbf{x}} \right)_\Delta + \mathbf{g}_l : \frac{\partial^2 \mathbf{c}}{\partial \mathbf{x}^2} \right) \quad (62)$$

For one final simplification, we have $\mathbf{A}_1 + \mathbf{A}_2 = -\mathbf{A}_1$, which allows us to avoid computing the fourth-order tensor in \mathbf{A}_2 . For signed length energies it turns out that \mathbf{A}_1 is zero, so directly clamping the

negative eigenvalues in the first term of Eqn. 17 is enough to ensure PSD-ness. For unsigned length, \mathbf{H}_u and \mathbf{g}_u take the place of \mathbf{H}_l and \mathbf{g}_l , resulting in a nonzero \mathbf{A}_1 . The filtering strategy discussed in the main paper must then be invoked.

It is not immediately obvious why $\mathbf{A}_1 + \mathbf{A}_2 = -\mathbf{A}_1$ for every case of signed/unsigned length and vertex-face/edge-edge frame, though it essentially is a result of orthogonality in the underlying reduced frame. To validate this step, we have implemented Matlab functions that randomize an initial vertex configuration and independently calculate the left and right side of the equality to check for correctness (see `Simplification_[l/u]_[vf/ee].m`).