Better Bending: Analysis, Construction and Verification of Discrete Bending Models for Kirchhoff-Love Shells

ZHEN CHEN*, Adobe, USA
ETIENNE VOUGA*, The University of Texas at Austin, USA
DANNY M. KAUFMAN*, Adobe, USA

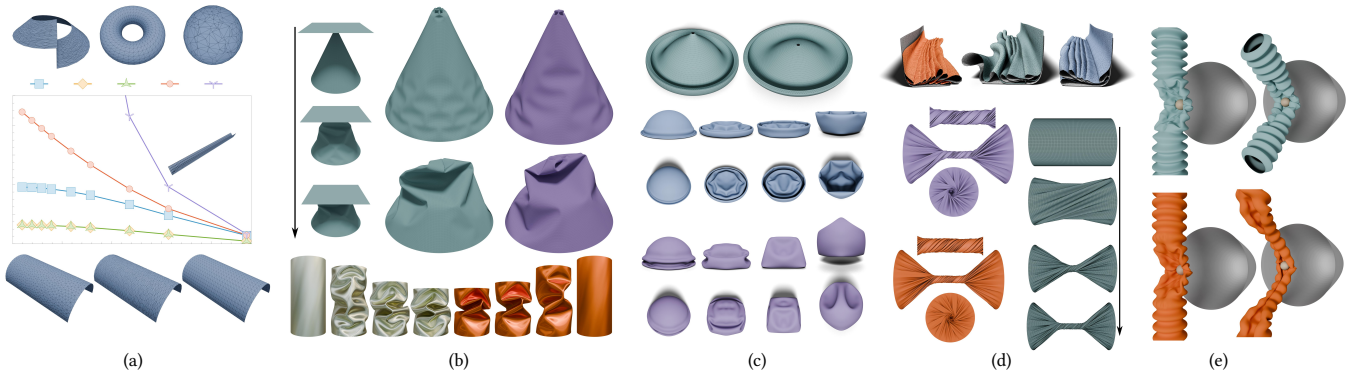


Fig. 1. We analyze the accuracy and efficiency of existing and new thin-shell bending models for simulating Kirchhoff-Love shells with (a) benchmarks that probe model convergence (under refinement across a range of midsurface geometries and meshings) and behavior at sharp bends; and a set of complex equilibrium and dynamic thin-shell modeling problems including (b) rest-curved buckling of stiff (paper and aluminum) materials; (c) high-speed, thin-wall impact and snap-through dynamics of softer rubber materials; (d) complex, contact-driven fabric wrinkling and folding; and (e) the dynamic response of structured, rest-curved materials.

While thin shells have ubiquitous applications and have been studied inside and outside computer graphics for decades, there is little consensus on how to best discretize them. We systematically study models for simulating bending of Kirchhoff-Love shells, with the goal of making practical recommendations, backed by careful numerical experiments, of when and how these models should be used. We analyze ten of the most popular discrete bending models in computer graphics for thin-shell simulation, along with new variants that we propose ourselves. We first verify all models on an analytic test benchmark to probe convergence under refinement and mesh-dependence, and then stress-test with a second benchmark that considers behavior at sharp bends. Finally, we test benchmark leaders on a practical suite of challenging large-scale equilibrium and dynamic shell modeling problems, analyzing both full solution behavior and comparative compute costs. We identify leading existing models and their tradeoffs in terms of accuracy and performance. During this analysis we also identify some issues and modeling gaps in the best-performing discrete bending models. We construct new energy model variants to address some of these gaps, as well as formulas and algorithmic tools for their practical simulation, and finally recommend best practices for modeling thin-shell bending.

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

*All authors contributed equally to this work.

Authors' Contact Information: Zhen Chen, Adobe, USA, zhenc@adobe.com; Etienne Vouga, The University of Texas at Austin, USA, evouga@utexas.edu; Danny M. Kaufman, Adobe, USA, dannykaufman@gmail.com.

ACM 1557-7368/2026/7-ART101
https://doi.org/10.1145/3811373

Additional Key Words and Phrases: Bending, Thin-Shell Models, Cloth

ACM Reference Format:
Zhen Chen, Etienne Vouga, and Danny M. Kaufman. 2026. Better Bending: Analysis, Construction and Verification of Discrete Bending Models for Kirchhoff-Love Shells. *ACM Trans. Graph.* 45, 4, Article 101 (July 2026), 31 pages. https://doi.org/10.1145/3811373

1 Introduction

Accurate simulation of elastic thin shells has been an active area of research in computer graphics for over forty years (well over fifty in computational mechanics). While volumetric elasticity is generally considered well-discretized via standard finite elements, there is little consensus on how to best discretize elastic thin shells, particularly for simulations involving significant amounts of flexure/bending. Instead, a diverse range of approaches have proliferated in the literature, growing annually with new contributions and based on a variety of different viewpoints including finite element analysis, discrete differential geometry, appeal to first geometric principles, or by analogy to flexure of 2D beams.

In this work we systematically study models for accurate simulation of thin-shell bending, with the goal of making practical recommendations of when and how these models should be used, backed by careful numerical experiments. To keep the scope of this effort tractable, we focus on thin shells satisfying the *Kirchhoff-Love* (KL) assumption that the shell does not shear transversely. KL models are widely applied across critical physical simulation applications in diverse domains spanning from engineering and robotics to fashion and computer graphics. This broad popularity

stems from the KL model's ability to predictively capture complex shell behavior despite its compact kinematics (deformation of the shell is fully determined by deformation of a 2D midsurface).

The computer graphics and mechanics literature has equipped Kirchhoff-Love shells with a range of different proposed material models and deformation energies. All of these decouple into a membrane "stretching" term, zero for isometric deformation of the shell midsurface and linear in the shell thickness, and a "bending" term that captures the remainder of the shell's deformation energy. We focus here primarily on discretizations of the bending energy contribution, since, in this setting, the membrane energy is generally well-discretized by Lagrange finite elements.

We analyze ten of the most popular discrete bending models in computer graphics for KL thin-shell simulation, along with new variants that we propose ourselves. We first verify these models on an analytic test benchmark to determine: 1) given the energy that each model proposes to discretize, does the discrete approximation converge to the analytic energy under mesh refinement; 2) how does accuracy depend on mesh resolution; and 3) how mesh-dependent is the model? We also test these models with a second benchmark that specifically evaluates discrete bending energy behavior for sharp creasing on meshes, to see how they behave in this critical regime.

We then test benchmark leaders on both a practical suite of challenging equilibrium and dynamic shell modeling problems, and a buckling validation, analyzing both full solution behavior and comparative compute costs. We identify leading existing models and their tradeoffs in terms of accuracy and computational cost.

With the above analysis, we identify modeling issues and algorithmic gaps in these best-performing discrete bending models. Based on these observations we construct new energy model variants (both bending and associated kinetic energy) to address some of these issues, algorithmic tools for their practical simulation, and recommend best practices for bending models (both prior and new) in thin-shell simulation.

1.1 Summary of Findings

Spoiler 1: The two (by far) most popular bending models used in computer graphics, Discrete Shell bending [Bridson et al. 2005; Gingold et al. 2004; Grinspun et al. 2003] and its second-order approximation via Crouzeix-Raviart elements (sometimes called Quadratic Bending (QB)) [Bergou et al. 2006; Wardetzky et al. 2007] both fail to converge to the analytic ground truth under mesh refinement *in all of our tests*. Moreover, the error is mesh-dependent. When applications require accuracy, these models should *not* be used.

Spoiler 2: Bending models based on a P1 element discretization of the biLaplacian (a variant formulation of Quadratic Bending) [Bergou et al. 2006] generally do not converge to the analytic ground truth either. However, with careful boundary treatment [Stein et al. 2018; Wang et al. 2015] they provide significant accuracy improvements over both the Crouzeix-Raviart QB *and* Discrete Shells, especially when built on intrinsic Delaunay meshes [Sharp and Crane 2020], at the cost of a denser Hessian sparsity pattern.

Spoiler 3: Three Cosserat-based bending models, the Discrete Kirchhoff Triangle (DKT) [Batoz et al. 1980], Discrete Cosserat Shells (DCS) [Weischedel 2012] and the Discrete Shape Operator (DSO) [Grinspun et al. 2006], *do* converge to the analytic ground truth under mesh refinement on all of our benchmark tests, at the cost of needing additional director degrees of freedom and increased compute time.

Spoiler 4: The DKT and DCS models suffer from a pathological decrease in bending energy whenever the dihedral angle between two neighboring triangle falls below 90 degrees. In such cases, bending forces act to *increase* the amount of bending. The DSO model does better, as its energy increases as the dihedral angle decreases, but it still converges to a finite value as the angle approaches 0. All forms of Quadratic Bending suffer from a decrease in the magnitude of force resisting further bending as the dihedral angle between two neighboring triangles falls towards 0. All of these degeneracies result in creasing artifacts in simulations, and so these models should *not* be used in practical applications where significant localized bending relative to mesh resolution can occur.

Spoiler 5: Although the DKT, DCS and DSO models were originally derived from quite different viewpoints, all three can be viewed within a common finite element framework that we cover in Section 7. We apply this framework to construct 1) a new Cosserat bending model variant, Bending-Active Cosserat (BAC), that avoids the degeneracies mentioned above in *Spoiler 4*, while still providing comparable convergence in our first benchmark; 2) mass matrices and kinetic energy terms for the director DOFs of DCS and DSO (missing from the literature), as well as BAC; and 3) algorithmic components for practical equilibrium and implicit time-integration solves of these Cosserat models.

We cover the construction of all of the above bending models in Sections 4, 5, and 7 (with analysis of hinge-based variants in Section 6), benchmark them in Sections 8 and 9, analyze their ranges of potential applicability and computational tradeoffs in complex thin-shell simulations and buckling validation in Section 10, and offer (for readers wishing to jump ahead) our final recommendations in Section 12. As they are new, we also include source code for both of our benchmarks, including implementations of all evaluated bending-model energies, for reproducibility and future testing.

2 Related Work

The study of thin shell models for simulation dates back at least forty years in computer graphics [Terzopoulos et al. 1987; Weil 1986] and goes back much further in computational mechanics. A wide array of models with overlapping constructions and sometimes uncertain properties have been developed in that time.

With sometimes differing criteria than the computational mechanics community, models developed in (and/or ported to) computer graphics have, over time, largely settled into primarily using Kirchhoff-Love (KL) shell models (see Section 3) with a discrete midsurface most often built with a linear triangle mesh. Of these, far and away the most popular are decoupled models that factor a shell's elasticity energy into a membrane (stretching) term, measured via in-plane distortion, and a bending term, measured via

a diverse range of models. We do not review these models here; instead we cover them individually and in detail in the next section.

In this work we focus exclusively on these decoupled KL models built on triangular midsurface meshes. More specifically we study the behavior of their bending energy terms, first in isolation, and then, based on these initial analyses, on their role in a full end-to-end KL shell simulation that includes membrane energy contributions.

Our study includes bending model analysis, verification benchmarking (testing discrete bending model convergence to underlying smooth models), stress benchmarking (quantitative evaluation of model robustness for practical-resolution meshes), qualitative and validation analyses (examining practical implications of these benchmark results in thin-shell simulations), and model performance (comparative timings for these same simulations). We also offer, based on these analyses, a new bending model variant and new tools for practical KL shell simulation.

Most closely aligned with our study are a pair of works in computer graphics on validation [Romero et al. 2021] and verification [Grinspun et al. 2006].

Romero et al. [2021] recently presented a complementary work benchmarking slender geometric models employed in computer graphics on a set of experimentally determined validation tests. Their work focuses on determining how well simulation models in graphics predictively match real-world experimental results, irrespective of their ability to converge. Our work instead primarily focuses on analysis and verification testing, as well as qualitative and performance analysis. We do also, however, apply Romero et al.'s thin-shell ("Lateral Buckling") validation test to better understand the bending models we study here.

Grinspun et al. [2006] propose a new discretization of the shape operator (DSO) on triangle meshes, and compare its behavior to converge under refinement to a known solution for the Squared Curvature energy on a sphere, as well as for two engineering finite-element obstacle course tests (Scordelis-Lo Roof and Pinched hemisphere) [Macneal and Harder 1985]. We discuss these latter engineering tests below. Towards analyzing computational costs of these methods Grinspun et al. also report the number of nonzeros (NNZ) per energy stencil for regular grid discretizations.

Following this excellent program, in this work we systematically test bending energy convergence across a range of bending discretizations employed in computer graphics (for both structured and unstructured meshes), including DSO, with a benchmark set of simple shell shapes (both rest-flat and rest-curved) for which exact answers for both the Squared Curvature and Koiter energies can be determined. We also similarly ask what comparative compute costs are across bending models, and so report both total NNZ counts for full bending energy Hessians on large-scale simulation meshes (giving good estimates of individual linear-solve costs) and complete wall-clock times for corresponding end-to-end shell simulation runs.

Thin Shell Benchmarking in Computational Mechanics. While a complete survey of shell benchmarking in computational mechanics is beyond our scope here, we note that a large number of nonlinear shell benchmark problems are formulated as tabulated results, e.g., in the form of load-deflection curve samples [Sze et al. 2004]. In turn, these solutions themselves are very often computed with other shell codes and, in a number of reported cases (e.g., the above-mentioned Scordelis-Lo Roof test) with somewhat murky provenance as to where reference solutions are sourced from, and which variant should be used [Krysl and Chen 2023]. We refer the interested reader to Krysl and Chen [2023] where these questions are covered and investigated in detail. They note that along with provenance and quality questions for computationally manufactured benchmarks, applicability of analytic benchmarks for shells, and so their suitability for a model, are often unclear in the literature as well. Here we focus on constructing and applying convergence testing benchmarks specifically designed for the shell bending models we study, with clearly defined analytic solutions and expected behaviors.

Thin Shell Material Models. While for the purpose of our study, we restrict our focus here to St. Venant-Kirchhoff type material models, alternate elastic material models, e.g. neo-Hookean, can be employed for KL shells [Chen et al. 2018; Wen and Barbič 2023]. Most recently Wen and Barbič [2023] develop an extensive KL material model generalization spanning a wide range of classical energy density formulations. At its root however, this generalization still requires an underlying discretization of the midsurface second-fundamental form to measure bending. Specifically Wen and Barbič [2023] employ the mid-edge average (M-e. Av.). Our analyses show that alternative discretizations should be preferred over M-e. Av., leading to easy and natural robust extensions of shell material models, beyond St. Venant-Kirchhoff, by simply swapping in some of the better-behaved discretizations of the second fundamental form we study and propose here.

3 Kirchhoff-Love Shells Background

In this section, we review basic background on thin elastic shells and establish notation. We briefly cover only essentials; more details, including proofs and derivations, can be found in textbooks on elasticity [Ciarlet 2021].

Kirchhoff-Love Shell Kinematics. Geometrically, an elastic shell is a volume that is much thinner in one of its dimensions. The configuration space of a shell is the set of embeddings $\phi(x, y, z) : \Omega \times [-h/2, h/2] \to \mathbb{R}^3$, where $\Omega \subset \mathbb{R}^2$ is a material parameter domain and the z axis parameterizes the shell's thin ("thickness") direction. The surface $\mathbf{r}(x, y) = \phi(x, y, 0)$ is the shell's *midsurface*. Kirchhoff-Love theory places additional assumptions on ϕ: that the thickness of the shell does not change during deformation, and that straight line segments normal to the xy plane in the material domain embed as straight line segments normal to the midsurface. In other words, for a Kirchhoff-Love shell of uniform thickness h,

$$\begin{aligned} \phi(x, y, z) &= \mathbf{r}(x, y) + z\hat{\mathbf{n}}(x, y), \text{ with} \\ \hat{\mathbf{n}}(x, y) &= \frac{\frac{\partial \mathbf{r}}{\partial x} \times \frac{\partial \mathbf{r}}{\partial y}}{\left\| \frac{\partial \mathbf{r}}{\partial x} \times \frac{\partial \mathbf{r}}{\partial y} \right\|}. \end{aligned} \tag{1}$$

The configuration of a Kirchhoff-Love shell is completely determined by the geometry of the midsurface $\mathbf{r}$; or equivalently, up to rigid motions, by the first and second fundamental forms of the

midsurface,

$$\mathbf{I} = \begin{bmatrix} \frac{\partial \mathbf{r}}{\partial x} \cdot \frac{\partial \mathbf{r}}{\partial x} & \frac{\partial \mathbf{r}}{\partial x} \cdot \frac{\partial \mathbf{r}}{\partial y} \\ \frac{\partial \mathbf{r}}{\partial x} \cdot \frac{\partial \mathbf{r}}{\partial y} & \frac{\partial \mathbf{r}}{\partial y} \cdot \frac{\partial \mathbf{r}}{\partial y} \end{bmatrix}, \quad \mathbf{II} = -\begin{bmatrix} \frac{\partial \hat{\mathbf{n}}}{\partial x} \cdot \frac{\partial \mathbf{r}}{\partial x} & \frac{\partial \hat{\mathbf{n}}}{\partial x} \cdot \frac{\partial \mathbf{r}}{\partial y} \\ \frac{\partial \hat{\mathbf{n}}}{\partial y} \cdot \frac{\partial \mathbf{r}}{\partial x} & \frac{\partial \hat{\mathbf{n}}}{\partial y} \cdot \frac{\partial \mathbf{r}}{\partial y} \end{bmatrix}. \tag{2}$$

The shell's (undeformed, zero-energy) *rest configuration* is often specified in terms of a rest midsurface $\bar{\mathbf{r}}(x, y)$, from which corresponding *rest* fundamental forms $\bar{\mathbf{I}}, \bar{\mathbf{II}}$ can be computed. The special case where the rest midsurface is a piece of a plane is often called a *thin plate*.

In some applications involving material changes to the shell due to plasticity, swelling, thermosetting, etc. it is more useful to directly prescribe rest fundamental forms [Chen et al. 2018] that do not correspond to any choice of $\bar{\mathbf{r}}$. Such "non-Euclidean" rest fundamental forms can encode arbitrary residual membrane or bending strains. A shell (whether Euclidean or not) with $\bar{\mathbf{II}} = \mathbf{0}$ is *rest flat*.

Deformation Energies. Many deformation energies have been proposed for Kirchhoff-Love shells, differing in what additional modeling assumptions and constitutive laws are imposed on the shell. One popular approach is to treat the shell as a linear-elastic (i.e., St. Venant-Kirchhoff) volume with Lamé parameters α and β, Taylor-expand its volumetric elastic energy in the thickness h, and keep only the dominant low-order terms. This derivation leads to the *Koiter* surface energy density [1966],

$$dE = (W_{\mathrm{M}} + W_{\mathrm{B}})\, dx\, dy, \tag{3}$$

$$W_{\mathrm{M}} = \frac{h}{2}\left(\frac{\alpha}{2}\mathrm{tr}(\boldsymbol{S}_M)^2 + \beta \mathrm{tr}(\boldsymbol{S}_M^2)\right), \quad \boldsymbol{S}_M = \bar{\mathbf{I}}^{-1}(\mathbf{I} - \bar{\mathbf{I}}), \tag{4}$$

$$W_{\mathrm{B}} = \frac{h^3}{12}\left(\frac{\alpha}{2}\mathrm{tr}(\boldsymbol{S}_B)^2 + \beta \mathrm{tr}(\boldsymbol{S}_B^2)\right), \quad \boldsymbol{S}_B = \bar{\mathbf{I}}^{-1}(\mathbf{II} - \bar{\mathbf{II}}). \tag{5}$$

The Koiter energy is composed of two terms: a membrane contribution W_{M} measuring the energy due to "stretching" of the shell, which vanishes for isometric deformations of the midsurface, and a "bending" contribution W_{B}.

Note that other deformation energies can be derived via comparable strategies, by assuming different hyperelastic materials, and/or making different choices about which terms to keep or drop in the Taylor expansion [Weischedel 2012; Wen and Barbič 2023].

Several key papers in the graphics literature [Bergou et al. 2006; Gingold et al. 2004] propose using a Squared Curvature energy,

$$W_{\mathrm{SC}} = \frac{h^3}{12}\left(\frac{\alpha}{2} + \beta\right) \mathrm{tr}\left(\bar{\mathbf{I}}^{-1}\left(\mathbf{II} - \bar{\mathbf{II}}\right)\right)^2, \tag{6}$$

to measure bending instead of W_B. This choice is motivated by the observation that *when a thin plate deforms isometrically*, $\mathbf{r}$ is a developable surface, $\mathrm{tr}\left(\bar{\mathbf{I}}^{-1}\mathbf{II}\right)$ measures the pointwise mean curvature of the surface, and $W_{\mathrm{SC}} = W_{\mathrm{B}}$. In turn, discrete mean curvature is a more straightforward quantity to estimate for a triangle mesh (from geometric principles) than the full second fundamental form.

4 Bending Models

In this section, we introduce the discrete bending models that we will analyze in detail. We give each method a short acronym for use throughout the paper. Since the different models we survey claim to discretize different shell deformation energies, a full apples-to-apples comparison across all models will not be possible. However, all models support either the Squared Curvature energy or the Koiter energy, and so we give a concise description of how each model estimates the total bending-energy contribution for either $E_{\mathrm{SC}} = \int_\Omega W_{\mathrm{SC}}$ or $E_{\mathrm{B}} = \int_\Omega W_{\mathrm{B}}$.

We give only concise formulas here, sometimes refactored slightly from how they were originally presented to harmonize notation across models. We do not motivate or derive the formulas; do not give formulas for the membrane energy contribution or for the bending derivatives and Hessian; and we do not explain how to generalize these formulas to other material models or shell deformation energies. Please see the original papers for these details.

Scope. We consider only discrete models for Kirchhoff-Love shells on triangle meshes. We consider out of scope models that allowing compression, shearing, or other deformation of the midsurface normals, e.g. solid-shells and Reissner–Mindlin shells [Lee and Bathe 2004; Lee et al. 2014; Montes Maestre et al. 2024] or micropolar models with twist DOFs [Löschner et al. 2024]; as well as nonconforming models [Gueidon and M. Chiaramonte 2025] and methods on quadrilateral meshes [Batoz and Tahar 1982], Catmull-Clark or Loop subdivision surfaces [Cirak et al. 2000; Clyde et al. 2017], or high-order surface patches.

Notation. Please see Table 1 and Figure 2 for the notation used in the following formulas.

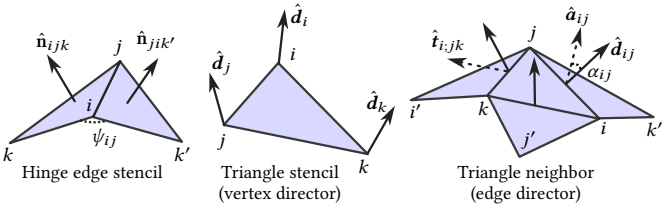


Fig. 2. Discrete quantities for bending model stencils; see Table 1.

4.1 Models Discretizing the Squared Curvature energy

All of these models use only the positions of the vertices of the mesh $\mathbf{x}_i$ as the discrete shell configuration.

4.1.1 DS(θ): Discrete Shells. A cluster of papers [Bridson et al. 2005; Gingold et al. 2004; Grinspun et al. 2003] describe a family of closely-related discrete models based on discretizing mean curvature at interior mesh edges. We take as a representative member the formulation by Grinspun et al. [2003],

$$E_{\mathrm{SC}} \approx \frac{h^3(\alpha + 2\beta)}{12} \sum_{\text{edges } ij} \frac{3\bar{\ell}_{ij}^2}{2\left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)} \left(\theta_{ij} - \bar{\theta}_{ij}\right)^2, \tag{7}$$

where the sum is over interior edges ij only.

4.1.2 DS(tan): Discrete Shells, Barrier Variant. We separately consider the variant Discrete Shells energy [Tamstorf and Grinspun 2013] with a barrier at $\theta = \pm\frac{\pi}{2}$,

$$E_{\mathrm{SC}} \approx \frac{h^3(\alpha + 2\beta)}{12} \sum_{\text{edges } ij} \frac{6\bar{\ell}_{ij}^2}{\bar{A}_{ijk} + \bar{A}_{ijk'}} \left(\tan\frac{\theta_{ij}}{2} - \tan\frac{\bar{\theta}_{ij}}{2}\right)^2. \tag{8}$$

Table 1. Notation used in the discrete bending energy formulas in this paper.

$\bullet_i$	quantities at a vertex i
$\bullet_{ij}$	quantities at an edge between vertices i, j
$\bullet_{ijk}$ $\bullet_{ijk'}$	quantities on the two faces adjacent to edge ij
$\bullet_{ij;k}$	quantities on halfedge ij of triangle ijk
$\bar{\bullet}$	quantities evaluated on the rest configuration
$\mathbf{x}_i = (x_i, y_i, z_i)$	vertex position
ℓ_{ij}	edge length $\|\mathbf{x}_j - \mathbf{x}_i\|$
$\mathbf{n}_{ijk}$	weighted face normal $\left(\mathbf{x}_j - \mathbf{x}_i\right) \times \left(\mathbf{x}_k - \mathbf{x}_i\right)$
$\hat{\mathbf{n}}_{ijk}$	face normal $\mathbf{n}_{ijk} / \|\mathbf{n}_{ijk}\|$
A_{ijk}	face area $\frac{1}{2}\|\mathbf{n}_{ijk}\|$
ψ_{ij}	dihedral angle; π for boundary edges
θ_{ij}	turning angle $\pi - \psi_{ij} = \angle\left(\hat{\mathbf{n}}_{ijk}, \hat{\mathbf{n}}_{ijk'}\right)$
c_{ijk}	cotangent of vertex j's interior angle; $c_{ijk} = \frac{(\mathbf{x}_i - \mathbf{x}_j)\cdot(\mathbf{x}_k - \mathbf{x}_j)}{\|(\mathbf{x}_i - \mathbf{x}_j)\times(\mathbf{x}_k - \mathbf{x}_j)\|}$
$\hat{\boldsymbol{t}}_{i;jk}$	outward-pointing in-plane normal to edge jk; $-\hat{\mathbf{n}}_{ijk} \times \frac{\mathbf{x}_k - \mathbf{x}_j}{\|\mathbf{x}_k - \mathbf{x}_j\|}$
$\hat{\boldsymbol{a}}_{ij}$	dihedral angle bisector at edge ij
α_{ij}	turning angle from $\hat{\boldsymbol{a}}_{ij}$ to edge director
h	shell thickness
Y, ν	Young's modulus, Poisson's ratio
α, β	Lamé parameters

Notice that the two Discrete Shell variants agree up to second order in θ at the rest configuration.

4.1.3 CH(ES): Corotational Hinge, Edge-based Shell Model. Several new models were proposed in concurrent work [Liang 2025] based on approximating discrete curvature at mesh edges or faces by normal displacements in a corotated frame. The simplest of these, the edge-based ES model, leads to the bending energy

$$E_{\mathrm{SC}} \approx \frac{h^3(\alpha + 2\beta)}{12} \sum_{\text{edges } ij} \frac{\bar{A}_{ijk} + \bar{A}_{ijk'}}{2} (\kappa_{ij} - \bar{\kappa}_{ij})^2 \tag{9}$$

$$\kappa_{ij} = \frac{\bar{\ell}_{ij}^2}{\bar{A}_{ijk} + \bar{A}_{ijk'}} \left(\frac{\hat{\mathbf{n}}_{ij} \cdot (\mathbf{x}_k - \mathbf{x}_i)}{2\bar{A}_{ijk}} + \frac{\hat{\mathbf{n}}_{ij} \cdot (\mathbf{x}_{k'} - \mathbf{x}_i)}{2\bar{A}_{ijk'}} \right) \tag{10}$$

$$\bar{\kappa}_{ij} = \frac{\bar{\ell}_{ij}^2}{\bar{A}_{ijk} + \bar{A}_{ijk'}} \left(\frac{\hat{\bar{\mathbf{n}}}_{ij} \cdot (\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_i)}{2\bar{A}_{ijk}} + \frac{\hat{\bar{\mathbf{n}}}_{ij} \cdot (\bar{\mathbf{x}}_{k'} - \bar{\mathbf{x}}_i)}{2\bar{A}_{ijk'}} \right), \tag{11}$$

where the sum is over interior edges and the edge normal $\hat{\mathbf{n}}_{ij}$ is the dihedral angle bisector,

$$\hat{\mathbf{n}}_{ij} = \frac{\hat{\mathbf{n}}_{ijk} + \hat{\mathbf{n}}_{ijk'}}{\left\|\hat{\mathbf{n}}_{ijk} + \hat{\mathbf{n}}_{ijk'}\right\|}, \tag{12}$$

and likewise for the rest edge normal.

4.1.4 CH(FS): Corotational Hinge, Face-based Shell Model. A second model proposed by Liang [2025] (their "FS" model) is based on a finite-volumes discretization of curvature averaged over each triangle. The corresponding bending energy, after some simplification of the formulas as originally presented, is

$$E_{\mathrm{SC}} \approx \frac{h^3(\alpha + 2\beta)}{24} \sum_{\text{faces } ijk} \bar{A}_{ijk} \left(\kappa_{ijk} - \bar{\kappa}_{ijk}\right)^T \boldsymbol{D} \left(\kappa_{ijk} - \bar{\kappa}_{ijk}\right), \tag{13}$$

for inner product

$$\boldsymbol{D} = \begin{bmatrix} 1 & \frac{\left[(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_j)\cdot(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_k)\right]^2}{\bar{\ell}_{jk}^2 \bar{\ell}_{ki}^2} & \frac{\left[(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_j)\cdot(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i)\right]^2}{\bar{\ell}_{jk}^2 \bar{\ell}_{ij}^2} \\ \frac{\left[(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_k)\cdot(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_j)\right]^2}{\bar{\ell}_{ki}^2 \bar{\ell}_{jk}^2} & 1 & \frac{\left[(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_k)\cdot(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i)\right]^2}{\bar{\ell}_{ki}^2 \bar{\ell}_{ij}^2} \\ \frac{\left[(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i)\cdot(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_j)\right]^2}{\bar{\ell}_{ij}^2 \bar{\ell}_{jk}^2} & \frac{\left[(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i)\cdot(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_k)\right]^2}{\bar{\ell}_{ij}^2 \bar{\ell}_{ki}^2} & 1 \end{bmatrix}$$

and discrete curvatures

$$\kappa_{ijk} = \begin{bmatrix} \frac{\bar{\ell}_{jk}^2 \hat{\mathbf{n}}_{ijk} \cdot (\mathbf{x}_{i'} - \mathbf{x}_i)}{2\bar{A}_{jki'}\left(\bar{A}_{ijk} + \bar{A}_{jki'}\right)} \\ \frac{\bar{\ell}_{ki}^2 \hat{\mathbf{n}}_{ijk} \cdot \left(\mathbf{x}_{j'} - \mathbf{x}_j\right)}{2\bar{A}_{kij'}\left(\bar{A}_{ijk} + \bar{A}_{kij'}\right)} \\ \frac{\bar{\ell}_{ij}^2 \hat{\mathbf{n}}_{ijk} \cdot \left(\mathbf{x}_{k'} - \mathbf{x}_k\right)}{2\bar{A}_{ijk'}\left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)} \end{bmatrix} \quad \bar{\kappa}_{ijk} = \begin{bmatrix} \frac{\bar{\ell}_{jk}^2 \hat{\bar{\mathbf{n}}}_{ijk} \cdot (\bar{\mathbf{x}}_{i'} - \bar{\mathbf{x}}_i)}{2\bar{A}_{jki'}\left(\bar{A}_{ijk} + \bar{A}_{jki'}\right)} \\ \frac{\bar{\ell}_{ki}^2 \hat{\bar{\mathbf{n}}}_{ijk} \cdot \left(\bar{\mathbf{x}}_{j'} - \bar{\mathbf{x}}_j\right)}{2\bar{A}_{kij'}\left(\bar{A}_{ijk} + \bar{A}_{kij'}\right)} \\ \frac{\bar{\ell}_{ij}^2 \hat{\bar{\mathbf{n}}}_{ijk} \cdot \left(\bar{\mathbf{x}}_{k'} - \bar{\mathbf{x}}_k\right)}{2\bar{A}_{ijk'}\left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)} \end{bmatrix}. \tag{14}$$

The first entry of κ (and $\bar{\kappa}$) is set to zero instead if edge jk is a boundary edge; and similarly for the other entries.

For both the above CH(ES) and CH(FS) energies we have simplified their presentation from the original [Liang 2025]. This includes dropping the inclusion of extraneous vanishing terms. We also note that Liang [2025] propose several additional variants that we did not benchmark, including several plate-specific versions of the above models as well as a "smoothed" face-based model that estimates curvatures by fitting a quadratic surface to each triangle and its four neighbors.

4.1.5 QB(CR): Quadratic Bending. The most popular quadratic (i.e., constant-Hessian) bending model is the Crouzeix-Raviart formulation of Quadratic Bending proposed by Bergou et al. [2006],

$$E_{\mathrm{SC}} \approx \frac{h^3(\alpha + 2\beta)}{12} \sum_{\text{edges } ij} \frac{3}{2\left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)} \|\boldsymbol{v}_{ij}\|^2 \tag{15}$$

$$\boldsymbol{v}_{ij} = \left(\bar{c}_{ijk} + \bar{c}_{ijk'}\right) \mathbf{x}_i + \left(\bar{c}_{jik} + \bar{c}_{jik'}\right) \mathbf{x}_j \tag{16}$$

$$- \left(\bar{c}_{ijk} + \bar{c}_{jik}\right) \mathbf{x}_k - \left(\bar{c}_{ijk'} + \bar{c}_{jik'}\right) \mathbf{x}_{k'}, \tag{17}$$

where again the sum is only over interior edges ij. The QB(CR) model fundamentally assumes that the shell is rest-flat. In this regime, it was shown by Bergou et al. [2006] and Wardetzky et al. [2007] that this energy is the second-order approximation of DS(θ) at $\theta = 0$.

4.1.6 DS(2nd): Second Order Discrete Shells. Recent work [Le et al. 2023] proposed an extension of Discrete Shells to plates discretized as piecewise-quadratic patches. Given P2 second-order basis functions $\boldsymbol{b}(s,t)$ in triangle barycentric coordinates (see Table 3), the DS(2nd) model introduces additional position degrees of freedom $\mathbf{x}_{ij}$ at midedge nodes, and defines a quadratic midsurface embedding function

$$\mathbf{r}_{ijk}(s,t) = \mathbf{x}_i b_1(s,t) + \mathbf{x}_j b_2(s,t) + \mathbf{x}_k b_3(s,t) \tag{18}$$

$$+ \mathbf{x}_{jk} b_4(s,t) + \mathbf{x}_{ki} b_5(s,t) + \mathbf{x}_{ij} b_6(s,t) \tag{19}$$

on each triangle of Ω. Since $\mathbf{r}$ is twice-differentiable, the Squared Curvature energy density can be computed at any point directly from Equation (6). In practice, the second fundamental form should

be computed using the equivalent but simpler-to-implement form

$$\mathbf{II}(s,t) = \begin{bmatrix} \frac{\partial^2 \mathbf{r}}{\partial s^2} \cdot \hat{\mathbf{n}} & \frac{\partial^2 \mathbf{r}}{\partial s \partial t} \cdot \hat{\mathbf{n}} \\ \frac{\partial^2 \mathbf{r}}{\partial s \partial t} \cdot \hat{\mathbf{n}} & \frac{\partial^2 \mathbf{r}}{\partial t^2} \cdot \hat{\mathbf{n}} \end{bmatrix}. \tag{20}$$

The integrated bending energy over the faces is then estimated by numerical quadrature,

$$E_{\text{faces}} = \frac{h^3}{12}\left(\frac{\alpha}{2} + \beta\right) \sum_{\text{faces } ijk} \sum_{n=1}^{N_F} \omega_n \text{tr}\left(\bar{\mathbf{I}}^{-1}\mathbf{II}(s_n,t_n)\right)^2, \tag{21}$$

for N_F quadrature weights ω_n and points (s_n, t_n); the rest first fundamental form is discretized as in the constant-strain triangle,

$$\bar{\mathbf{I}} = \begin{bmatrix} (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i)\cdot(\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i) & (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i)\cdot(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_i) \\ (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i)\cdot(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_i) & (\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_i)\cdot(\bar{\mathbf{x}}_k - \bar{\mathbf{x}}_i) \end{bmatrix}, \tag{22}$$

taking advantage of each face of the plate being a flat triangle in the rest state.

The piecewise-quadratic midsurface $\mathbf{r}$ has tangent discontinuities at the patch edges, which contribute bending energy as well. The DS(2nd) model computes this contribution using a Discrete-Shells-like formula integrated along each interior edge:

$$E_{\text{edges}} = \frac{h^3(\alpha + 2\beta)}{12} \sum_{\text{edges } ij} \frac{3\bar{\ell}_{ij}^2}{2\left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)} \sum_{n=1}^{N_E} \omega_n \theta_{ij}(s_n)^2, \tag{23}$$

$$\theta_{ij}(s) = \angle\left[\hat{\mathbf{n}}_{ijk}(s,0), \hat{\mathbf{n}}_{ik'j}(0,s)\right]. \tag{24}$$

Le et al. [2023] recommend $N_F = 3$ and $N_E = 2$, using quadrature points and weights:

i	s_i	t_i	ω_i
1	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{3}$
2	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{1}{3}$
3	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{3}$

i	s_i	ω_i
1	$\frac{1}{2} - \frac{1}{2\sqrt{3}}$	$\frac{1}{2}$
2	$\frac{1}{2} + \frac{1}{2\sqrt{3}}$	$\frac{1}{2}$.

The final bending energy estimate is then $E_{\text{SC}} \approx E_{\text{faces}} + E_{\text{edges}}$.

4.2 Models Discretizing the Koiter energy

4.2.1 DKT: Discrete Kirchhoff Triangle. The DKT element was originally proposed in computational engineering [Batoz et al. 1980] (and subsequently applied [Furuta et al. 2010; Li et al. 2017; Zheng et al. 2021] or tested [Umetani et al. 2011] in computer graphics) in non-corotational form in terms of angles of rotation of a planar rest triangle. We rewrite the energy here using the same notation as the other models.

The discrete configuration space of a shell discretized using DKT consists of mesh vertex positions as well as *unit directors* $\hat{\boldsymbol{d}}_i$ at vertices. The key idea of the DKT element is to interpolate the vertex directors into the triangle using *second*-order elements, where the value of the directors at the midedge nodes are chosen so that the director field satisfies the Kirchhoff-Love condition along the boundary of a displaced surface interpolating the triangle corners.

Using again the P2 basis functions (see Table 3), the DKT discretization of the second fundamental form at barycentric coordinate (s,t) is:

$$\mathbf{II}(s,t) = \sum_{n=0}^{6} \begin{bmatrix} \frac{\partial \boldsymbol{b}_n}{\partial s}(\mathbf{x}_j - \mathbf{x}_i)\cdot\boldsymbol{p}_n & \mathbf{v}\cdot\boldsymbol{p}_n \\ \mathbf{v}\cdot\boldsymbol{p}_n & \frac{\partial \boldsymbol{b}_n}{\partial t}(\mathbf{x}_k - \mathbf{x}_i)\cdot\boldsymbol{p}_n \end{bmatrix} \tag{25}$$

$$\mathbf{v} = \frac{1}{2}\left(\frac{\partial \boldsymbol{b}_n}{\partial t}(\mathbf{x}_j - \mathbf{x}_i) + \frac{\partial \boldsymbol{b}_n}{\partial s}(\mathbf{x}_k - \mathbf{x}_i)\right), \tag{26}$$

where

$$\boldsymbol{p} = \begin{bmatrix} \hat{\boldsymbol{d}}_i \\ \hat{\boldsymbol{d}}_j \\ \hat{\boldsymbol{d}}_k \\ \hat{\boldsymbol{t}}_{i;jk}\cdot\frac{\hat{\boldsymbol{d}}_j+\hat{\boldsymbol{d}}_k}{2}\hat{\boldsymbol{t}}_{i;jk} - (\mathbf{x}_k - \mathbf{x}_j)\cdot\frac{\hat{\boldsymbol{d}}_j+\hat{\boldsymbol{d}}_k}{4\ell_{jk}^2}(\mathbf{x}_k - \mathbf{x}_j) \\ \hat{\boldsymbol{t}}_{j;ki}\cdot\frac{\hat{\boldsymbol{d}}_k+\hat{\boldsymbol{d}}_i}{2}\hat{\boldsymbol{t}}_{j;ki} - (\mathbf{x}_i - \mathbf{x}_k)\cdot\frac{\hat{\boldsymbol{d}}_k+\hat{\boldsymbol{d}}_i}{4\ell_{ki}^2}(\mathbf{x}_i - \mathbf{x}_k) \\ \hat{\boldsymbol{t}}_{k;ij}\cdot\frac{\hat{\boldsymbol{d}}_j+\hat{\boldsymbol{d}}_k}{2}\hat{\boldsymbol{t}}_{k;ij} - (\mathbf{x}_j - \mathbf{x}_i)\cdot\frac{\hat{\boldsymbol{d}}_i+\hat{\boldsymbol{d}}_j}{4\ell_{ij}^2}(\mathbf{x}_j - \mathbf{x}_i) \end{bmatrix}, \tag{27}$$

with analogous expressions for $\bar{\mathbf{II}}(s,t)$. Equation (22) is used to calculate the rest first fundamental form. Finally, the Koiter energy density is integrated over each triangle using three-point quadrature (same as in Section 4.1.6 above for points and weights), which is exact since Equation 5 is quadratic in the second fundamental form entries [Batoz et al. 1980] and gives

$$E_{\text{B}} \approx \sum_{\text{faces } ijk} \frac{h^3 \bar{A}_{ijk}}{12} \sum_{n=1}^{3} \omega_n \left(\frac{\alpha}{2}\text{tr}(\boldsymbol{S}_n)^2 + \beta\text{tr}\left(\boldsymbol{S}_n^2\right)\right) \tag{28}$$

$$\boldsymbol{S}_n = \bar{\mathbf{I}}^{-1}\left(\mathbf{II}(s_n,t_n) - \bar{\mathbf{II}}(s_n,t_n)\right). \tag{29}$$

4.2.2 DCS: Discrete Cosserat Shell. The Discrete Cosserat Shell model [Weischedel 2012], like the Discrete Kirchhoff Triangle, introduces extra discrete unit director DOFs. However, the DCS directors $\hat{\boldsymbol{d}}_{ij}$ are at *edge midpoints* and are constrained to remain perpendicular to edge ij; these midedge directors can therefore be represented by a single angle α_{ij} per edge, indicating the amount to rotate the dihedral angle bisector $\hat{\boldsymbol{a}}_{ij}$ about edge ij to form $\hat{\boldsymbol{d}}_{ij}$. (We note in passing that the DCS model also supports an optional, additional shear one-form; we do not consider this DOF or its energy contributions since we focus exclusively on Kirchhoff-Love shells.)

The DCS second fundamental form is constant per triangle and is given by

$$\mathbf{II}_{ijk} = \begin{bmatrix} \mathbf{II}_{jk;i} + \mathbf{II}_{ki;j} & \mathbf{II}_{jk;i} \\ \mathbf{II}_{jk;i} & \mathbf{II}_{jk;i} + \mathbf{II}_{ij;k} \end{bmatrix}, \tag{30}$$

$$\mathbf{II}_{jk;i} = \frac{4A_{ijk}}{\ell_{jk}}\sin\left(\frac{\theta_{jk}}{2} + \epsilon_{jk}\alpha_{jk}\right), \tag{31}$$

where $\epsilon_{jk} \in \{\pm 1\}$ is an arbitrary constant edge orientation satisfying $\epsilon_{jk} = -\epsilon_{kj}$. The discrete bending energy is then

$$E_{\text{B}} \approx \frac{h^3}{12} \sum_{\text{faces } ijk} \bar{A}_{ijk}\left(\frac{\alpha}{2}\text{tr}\left(\boldsymbol{S}_{ijk}\right)^2 + \beta\text{tr}\left(\boldsymbol{S}_{ijk}^2\right)\right) \tag{32}$$

$$\boldsymbol{S}_{ijk} = \bar{\mathbf{I}}_{ijk}^{-1}(\mathbf{II}_{ijk} - \bar{\mathbf{II}}_{ijk}), \tag{33}$$

where the discrete first fundamental form is defined using Equation 22.

4.2.3 DSO: Discrete Shape Operator. Grinspun et al. [2006] arrived at a similar model to DCS from the perspective of discretizing the shape operator on a triangle and its neighbors. They propose the discrete second fundamental form entries

$$\mathbf{II}_{jk;i} = \frac{4A_{ijk}}{\ell_{jk}}\left(\frac{\theta_{jk}}{2} + \epsilon_{jk}\alpha_{jk}\right), \tag{34}$$

from which the second fundamental form matrix and total bending contribution is computed using the same Equations (30) and (32) as in DCS. Note that DCS and DSO agree up to second order in θ and α for a rest-flat shell near its rest configuration.

4.2.4 M-e. Av.: Midedge Average DCS Variant. The edge directors in the DCS and DSO models significantly increase the number of discrete degrees of freedom in a physical simulation, and moreover it is not obvious nor covered how these additional DOFs should contribute to the mass matrix or inertia terms when simulating dynamics. To avoid these complications, Chen et al. [2018] propose expressing $\hat{\boldsymbol{d}}_{ij}$ as a function of the mesh vertex positions alone. Specifically, they recommend setting each director to the area-weighted average of the two neighboring face normals,

$$\hat{\boldsymbol{d}}_{ij} = \begin{cases} \hat{\mathbf{n}}_{ijk}, & \text{ij is a boundary edge} \\ \left(\mathbf{n}_{ijk} + \mathbf{n}_{ijk'}\right) / \left\|\mathbf{n}_{ijk} + \mathbf{n}_{ijk'}\right\|, & \text{otherwise.} \end{cases} \tag{35}$$

The second fundamental form entries are then

$$\mathbf{II}_{jk;i} = \left(\mathbf{x}_j + \mathbf{x}_k - 2\mathbf{x}_i\right) \cdot \hat{\boldsymbol{d}}_{jk}, \tag{36}$$

from which the second fundamental form matrix and total bending contribution is again computed using Equations (30) and (32).

5 Vertex-Based Quadratic Bending Models

The Crouzeix-Raviart form of Quadratic Bending described in Section 4.1.5 is the most popular quadratic model used in practice, and is the form recommended by Bergou et al. [2006] in their original paper. However, the same paper also mentions the possible alternative of a Lagrange form of the Squared Curvature energy that merits consideration:

$$E_{\text{SC}} \propto \frac{1}{2}\boldsymbol{x}^T\left(I_{3\times 3} \otimes L^T M^{-1} L\right)\boldsymbol{x}, \tag{37}$$

where $\boldsymbol{x} = \begin{bmatrix} x_1 & y_1 & z_1 & x_2 & \cdots \end{bmatrix}^T$ is the concatenated vertex position vector, L is the (Neumann) cotangent Laplacian constructed on the discrete rest configuration of the midsurface, and M is a mass matrix; in our experiments we use the lumped barycentric mass matrix recommended by Bergou et al. [2006],

$$M_{ii} = \sum_{\text{faces } ijk} \frac{1}{3}\bar{A}_{ijk}, \tag{38}$$

where the sum is over faces sharing vertex i.

Compared to the Crouzeix-Raviart formula, this Lagrange form has two main weaknesses:

- it does not automatically handle midsurfaces with boundary correctly: at the boundary, the gradient of Equation (37) exerts a spurious contractive membrane force; and
- its (constant) Hessian is significantly denser than that of the Crouzeix-Raviart form[1].

[1]The QB(PL) Hessian is still sparser than the (nonconstant) Hessian for DS(θ).

Subsequent work [Stein et al. 2018; Wang et al. 2015] carefully derived the missing boundary contributions to the Squared Curvature energy in terms of a discrete normal derivative operator. We test this QB(PL) variant of Quadratic Bending:

$$E_{\text{SC}} \approx \frac{h^3}{24}\left(\alpha + 2\beta\right)\boldsymbol{x}^T\left(I_{3\times 3} \otimes \left(L + N\right)^T M^{-1}\left(L + N\right)\right)\boldsymbol{x} \tag{39}$$

$$N = \frac{1}{2}\sum_{\text{bdries } jk}\left(\boldsymbol{e}_j + \boldsymbol{e}_k\right)\left[\bar{c}_{ijk}\left(\boldsymbol{e}_k - \boldsymbol{e}_i\right)^T + \bar{c}_{jki}\left(\boldsymbol{e}_j - \boldsymbol{e}_i\right)^T\right], \tag{40}$$

where the sum is over boundary edges jk that share a face with vertex i, and the $\boldsymbol{e}_i$ are Euclidean basis vectors.

We will see in the experiments of Section 8 that the QB(PL) model is far more accurate than QB(CR) on *all* of our test cases.

5.0.1 QB(int): Intrinsic Quadratic Bending Variant. The cotangent Laplacian is well-known to behave poorly on non-Delaunay triangle meshes. We propose and test one final variant of Quadratic Bending where we simply replace L with the intrinsic Laplacian [Sharp and Crane 2020] L_{int} constructed from the rest mesh edge lengths, and N with its analogue computed on the intrinsic triangulation:

$$E_{\text{SC}} \approx \frac{h^3}{24}\left(\alpha + 2\beta\right)\boldsymbol{x}^T\left(I_{3\times 3} \otimes \left(L_{\text{int}} + N_{\text{int}}\right)^T M^{-1}\left(L_{\text{int}} + N_{\text{int}}\right)\right)\boldsymbol{x}.$$

Note that for problems where the rest midsurface mesh is (intrinsic) Delaunay, the QB(PL) and QB(int) models are identical. When the rest configuration is not Delaunay, the QB(int) variant provides the accuracy benefits of a Delaunay mesh without the complications of actually remeshing the midsurface.

6 Analysis of Hinge Based Models

As we will see in Section 8, models that compute bending energy based on a two-triangle stencil of vertex positions (DS(θ), DS(tan), QB(CR)) *always* fail to converge to the analytic energy under refinement: even for rest-flat plates discretized with a regular mesh that is favorably aligned to the bending direction. We are not the first to observe this phenomenon, which has been reported by Grinspun et al. [2006] and others [Feng et al. 2022; Liang 2025].

Consider for example a rectangular plate of dimension $\pi R \times H$ that is rolled in its width direction into a half-cylinder of radius R (see Section 8.2). The mean curvature of the half-cylinder is $\frac{1}{R}$, and so the exact Squared Curvature energy is

$$E_{\text{SC}} = \left(\frac{\alpha}{2} + \beta\right)\frac{\pi h^3 H}{12R}. \tag{41}$$

When this plate is meshed using a regular, axis-aligned rectangular grid (with each rectangular face then diagonalized to form two triangles), the Squared Curvature energy reported by both Discrete Shells variants converges to exactly $3E_{\text{SC}}$ under refinement (see Table 4). For irregular meshes, the Discrete Shells energies overestimate E_{SC} by varying, mesh-dependent multiplicative factors. When using an unfavorable mesh (such as in the 4× Anisotropic Cylinder test case; see Section 8.2), the error can be as high as 100×!

This error factor can be understood in terms of the Jensen gap in discrete mean curvature estimates at the mesh edges [Solomon 2025]. Consider measuring the normal curvature $\kappa(\hat{\boldsymbol{t}})$ at a fixed point $\boldsymbol{x}$ on the analytic midsurface and in a direction $\hat{\boldsymbol{t}}$ tangent to the midsurface chosen uniformly at random. The expected normal

curvature is one-half of the mean curvature; in the case of the half-cylinder, $\langle \kappa \rangle = \frac{1}{2R}$.

Now consider a triangle mesh approximating the midsurface. The discrete curvature κ_{ij} as measured by Discrete Shells at each two-triangle hinge ij samples the normal curvature of the midsurface at that edge, in the direction perpendicular to the hinge edge. Both Gingold et al. [2004] and Grinspun et al. [2003] cite Cohen-Steiner and Morvan [2003] for a proof that in the limit of "reasonable" (Delaunay) refinement, the integrated discrete normal curvature converges to one-half of the integrated mean curvature of the underlying smooth midsurface; in other words, assuming equal-sized triangles, $\langle \kappa_{ij} \rangle \to \frac{1}{2R}$. But it does *not* follow that $\langle \kappa_{ij}^2 \rangle \to \frac{1}{4R^2}$. Indeed, for the regular discretization of the half-cylinder, two-thirds of hinge stencils have $\kappa_{ij} = 0$ (those corresponding horizontal and diagonal edges on the 2D rest plate) and one-third have $\kappa_{ij} = \frac{3}{2R}$ (those corresponding to vertical edges); hence $\langle \kappa_{ij}^2 \rangle = \frac{1}{3}\frac{9}{4R^2} = 3\frac{1}{4R^2}$, explaining the multiplicative factor of 3 error in discrete Squared Curvature energy. For irregular meshes, the factor varies depending on the precise shape of the distribution of the κ_{ij}.

Notice that there are *two* sources of variance in κ_{ij} that lead to Discrete Shells overestimating the Squared Curvature energy:

- variance in the normal curvature at a point as a function of direction $\hat{\boldsymbol{t}}$; this variance comes from the geometry of the underlying midsurface and is independent of how the surface is discretized;
- discretization error when estimating normal curvature using a two-triangle hinge.

Variance due to discretization error can be minimized (but not eliminated) by using an anisotropic mesh that is favorably aligned with the curvature direction; see the sphere test case in Section 8.5 for an example where Discrete Shells fails to converge under refinement even for an isotropic mesh approximating a surface of constant normal curvature.

Behavior of QB(PL) and QB(int). Any discrete bending model that seeks to estimate Squared Curvature energy by first approximating mean curvature and then squaring will suffer from a similar energy bias. Notice for instance that in the experiments in Section 8, the QB(PL) model fails to converge to the analytic Squared Curvature energy on *all* experiments involving irregular meshes. The overall higher accuracies we observe in Section 8 of QB(PL) and QB(int) when compared to the Discrete Shells models can be attributed to (a) elimination of normal curvature variance as a source of error, since QB(PL) estimates mean curvature directly; and (b) reduced variance in the discretization error of mean curvature, due to the larger stencil used by QB(PL).

7 Discrete Director Cosserat KL Shells

Several models covered in Section 4 can be viewed as members of a general family of mixed-element Cosserat models. In this section, we present this more general framework, taxonomize existing models within it, and propose a new variant, the Bending-Active Cosserat (BAC) model, that is better-behaved under larger deflections. We then present a general recipe for computing mass matrices for members of this family of models (not previously available), and fill in some additional technical gaps in the existing literature on how to robustly solve models like DCS, DSO and BAC when simulating statics and dynamics.

Table 2. Models that fit into the general framework we present in Section 7.1. For each ingredient we list the type of element used to discretize that quantity, as described in Table 3. We also summarize how each model discretizes the Kirchhoff-Love constraints.

Method	$\mathbf{r}$	$\mathbf{d}$	Kirchhoff-Love Constraints
DKT	P1	P2	$\|\mathbf{d}_i\| = 1$; $\int_{\text{edges } ij} \mathbf{d} \cdot (\mathbf{x}_j - \mathbf{x}_i) = 0$
DCS	P1	CR1	$\|\mathbf{d}_{ij}\| = 1$; $\mathbf{d}_{ij} \cdot (\mathbf{x}_j - \mathbf{x}_i) = 0$
BAC	P1	CRH1	$\mathbf{d}_{ij;k} \cdot \hat{\mathbf{n}}_{ijk} = 1$; $\mathbf{d}_{ij;k} \cdot (\mathbf{x}_j - \mathbf{x}_i) = 0$; $\mathbf{d}_{ij;k} \parallel \mathbf{d}_{ji;k'}$

7.1 A General Family of Mixed-Element Models

For shells satisfying the Kirchhoff-Love assumptions, the deformation of the shell volume is completely determined by the deformed midsurface $\mathbf{r}$. However, as seen in the DKT, DSO, and DCS models, when discretizing Kirchhoff-Love shells it can be useful to start with a more general formulation that augments midsurface position with an *arbitrary* director field. The Kirchhoff-Love constraints can then be enforced on the (midsurface, director) combination in a weak sense, or just at select tying points. This gives a rich design space of discretizations to consider.

Given the parameter domain Ω of a shell, we propose a general family of discrete Cosserat models with the following ingredients:

- midsurface positions $\mathbf{r} : \Omega \to \mathbb{R}^3$,
- director field $\mathbf{d} : \Omega \to \mathbb{R}^3$, and
- a set of *Kirchhoff-Love constraints* coupling the previous ingredients by discretizing the conditions that the director field is unit and normal to the midsurface.

In terms of these ingredients, the shell's volumetric embedding is given by

$$\phi(x, y, z) = \mathbf{r}(x, y) + z\mathbf{d}(x, y). \tag{42}$$

A feature of decomposing the shell degrees of freedom into separate midsurface position and director fields is that each ingredient can be discretized via a *different* finite element. Indeed, in Table 2 we list how each of the DKT and DCS models can be realized as a member of this family of models for some choice of finite element discretization (see Table 3) of each ingredient and corresponding KL constraints.

The DSO model then, in turn, derives from this framework as a second-order approximation (for rest-flat shells) of DCS with respect to director angle at the rest configuration.

Koiter Energy Approximation. Starting from the volumetric deformation function $\phi(x, y, z)$ in Equation (42) and assuming a transverse-isotropic linear material with Young's modulus Y and Poisson's ratio ν, one can derive a discrete elastic energy density by following the same recipe as when deriving the Koiter energy: Taylor-expand the volumetric elastic energy to third order in h and keep the most

Table 3. Table of finite elements [Scroggs et al. 2026] used by the various methods that we survey. Basis functions are listed for one triangle only, in that triangle's barycentric coordinates (s, t).

Element	Conforming?	Node	Basis Function
P1	Yes	$\mathbf{x}_i$	$1-s-t$
		$\mathbf{x}_j$	s
		$\mathbf{x}_k$	t
P2	Yes	$\mathbf{x}_i$	$2(1-s-t)\left(\frac{1}{2}-s-t\right)$
		$\mathbf{x}_j$	$s(2s-1)$
		$\mathbf{x}_k$	$t(2t-1)$
		$\mathbf{x}_{jk}$	$4st$
		$\mathbf{x}_{ik}$	$4t(1-s-t)$
		$\mathbf{x}_{ij}$	$4s(1-s-t)$
CR1 CRH1	No	$\mathbf{x}_{jk}$	$2s+2t-1$
		$\mathbf{x}_{ik}$	$1-2t$
		$\mathbf{x}_{ij}$	$1-2s$

dominant terms. Doing so yields a shell energy density very close to the Koiter energy,

$$dE = (W_{\mathrm{M}} + W_{\mathrm{B}} + W_{\mathrm{KL}})\, dx\, dy, \tag{43}$$

$$W_{\mathrm{M}} = \frac{h}{2}\left(\frac{\alpha}{2}\mathrm{tr}(S_M)^2 + \beta\mathrm{tr}(S_M^2)\right), \quad S_M = \bar{\mathbf{I}}^{-1}(\mathbf{I}-\bar{\mathbf{I}}), \tag{44}$$

$$W_{\mathrm{B}} = \frac{h^3}{12}\left(\frac{\alpha}{2}\mathrm{tr}(S_B)^2 + \beta\mathrm{tr}(S_B^2)\right), \quad S_B = \bar{\mathbf{I}}^{-1}(\mathbf{II}-\bar{\mathbf{II}}), \tag{45}$$

$$W_{\mathrm{KL}} = \frac{Yh}{4}\left(\|\mathbf{d}\|^2-1\right)^2 + \frac{\beta h}{2}\mathbf{d}^T\mathrm{d}\mathbf{r}\bar{\mathbf{I}}^{-1}\mathrm{d}\mathbf{r}^T\mathbf{d}, \tag{46}$$

where $\alpha = Y\nu/(1-\nu^2)$ and $\beta = Y/(2(1+\nu))$, $\mathrm{d}\mathbf{r}$ is the midsurface Jacobian, $\mathbf{I} = \mathrm{d}\mathbf{r}^T\mathrm{d}\mathbf{r}$ is the midsurface first fundamental form, and the suggestively-named $\mathbf{II}$ is the matrix

$$\mathbf{II} = -\begin{bmatrix} \frac{\partial \mathbf{r}}{\partial x}\cdot\frac{\partial \mathbf{d}}{\partial x} & \frac{1}{2}\left(\frac{\partial \mathbf{r}}{\partial x}\cdot\frac{\partial \mathbf{d}}{\partial y}+\frac{\partial \mathbf{r}}{\partial y}\cdot\frac{\partial \mathbf{d}}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial \mathbf{r}}{\partial x}\cdot\frac{\partial \mathbf{d}}{\partial y}+\frac{\partial \mathbf{r}}{\partial y}\cdot\frac{\partial \mathbf{d}}{\partial x}\right) & \frac{\partial \mathbf{r}}{\partial y}\cdot\frac{\partial \mathbf{d}}{\partial y} \end{bmatrix}, \tag{47}$$

with the corresponding rest quantities $\bar{\mathbf{I}}$ and $\bar{\mathbf{II}}$ defined analogously. W_{KL} is a correction term that vanishes *if* the Kirchhoff-Love conditions hold exactly: $\|\mathbf{d}\| = 1$ and $\mathrm{d}\mathbf{r}^T\mathbf{d} = 0$. Dropping W_{KL}, under the theory that each model's discrete Kirchhoff-Love constraints ensure that these conditions are approximately true, leaves W_B as the bending energy density. For DKT and DCS, the bending energies listed in Section 4 are *exactly* this W_B integrated over Ω.

When the Kirchhoff-Love conditions hold, the matrix $\mathbf{II}$ can be viewed as a discretization of the midsurface second fundamental form, similar to the discrete shape operator proposed by Grinspun et al. [2006] and underlying the DSO bending model. While we focus here on analyzing its use in computing the Koiter energy, it might be more broadly useful for discretizing other energies in simulation and geometry processing depending on surface curvature.

There is no guarantee that any particular choice of finite element discretization and discrete Kirchhoff-Love constraint leads to a bending model that is consistent with Kirchhoff-Love shell theory in the limit of refinement. The behavior of each combination must be analyzed on a case-by-case basis. DKT is known [Batoz et al. 1980] to be consistent, and our experiments in Section 8 strongly suggest that DCS, DSO, and BAC (described below) are all as well.

Implications for Large-Bend Behavior. As summarized in Table 2 the DKT and DCS models impose KL constant-thickness constraints at nodes (respectively vertex and midedge). Since these constraints are pointwise, shell thickness effectively varies (quadratically and linearly) via interpolation (P2 and CR1) over triangles, allowing each triangle's thickness to attenuate near boundaries. In practice, as we will see next in Sections 9 and 10, this supports spurious low-energy evaluations in configurations with large local bending.

BAC: Bending-Active Cosserat. Given the choice of discretizing $\mathbf{r}$ with P1 elements, a natural idea for a discrete Kirchhoff-Love constraint is then to require that the discrete shell have constant thickness h over each triangle: in other words, that $\mathbf{d}\cdot\hat{\mathbf{n}}_{ijk} = 1$ on each face ijk. However, one obvious difficulty with such a constraint is that it is incompatible with the director field $\mathbf{d}$ being continuous across triangle edges or vertices.

The idea can be salvaged by discretizing $\mathbf{d}$ using discontinuous *halfedge Crouziex-Raviart elements* (which we call CRH1): the basis functions of this element are supported only on single triangles, where they are equal to the ordinary CR1 basis functions; they linearly interpolate quantities $\mathbf{d}_{ij;k}$, $\mathbf{d}_{jk;i}$, and $\mathbf{d}_{ki;j}$ on triangle ijk's halfedges into the triangle.

In addition to the constant discrete thickness constraint $\mathbf{d}_{ij;k}\cdot\hat{\mathbf{n}}_{ijk} = 1$, we borrow the edge orthogonality constraint from the DCS model: $\mathbf{d}_{ij;k}\cdot\left(\mathbf{x}_j-\mathbf{x}_i\right) = 0$. We also require that, as in the DCS model, the *direction* of the director field on two neighboring triangles agrees at the midpoint of their common edge (even if the magnitude does not): $\mathbf{d}_{ij;k} \parallel \mathbf{d}_{ji;k'}$.

These discrete KL constraints conveniently reduce (see Appendix A) the BAC model's DOF from two directors per midedge to a *single* angle α_{jk} per edge (just as for DCS and DSO) that encodes the midedge's common director *orientation*. With this construction, the discrete second fundamental form in Equation (47) reduces to Equation (30) with entries

$$\mathbf{II}_{jk;i} = \frac{4A_{ijk}}{\ell_{jk}}\tan\left(\frac{\theta_{jk}}{2}+\epsilon_{jk}\alpha_{jk}\right). \tag{48}$$

The BAC approximation to the Koiter bending energy is then given by Equation (32). This energy is constant per triangle stencil so that one-point quadratures suffice. Directors are parameterized and solved for (see below) by midedge angle scalar DOF, with KL constraints enforced implicitly.

Based on the form of Equation (48), the BAC model can be viewed as the discrete Cosserat analogue of the DS(tan) variant of Discrete Shells. And indeed, just like the DS(tan) model, the BAC bending energy diverges as $\theta \to 0$ at any hinge; see the experiments in Section 9 for the full analysis. In practice, we observe that this proposed BAC model yields smoother wrinkle patterns with fewer creasing artifacts than other models like DCS and DSO, particularly in simulations involving energetic contact. (See for instance the cloth drape in Figure 19.)

7.2 Discrete Director Kinetic Energy

Given the Kirchhoff–Love shell representation in Equation (42), the material velocity field is

$$\dot{\phi}(x, y, z) = \dot{\mathbf{r}}(x, y) + z\,\dot{\mathbf{d}}(x, y). \tag{49}$$

The total kinetic energy then reads

$$\begin{aligned} E_{\text{kinetic}} &= \frac{\rho}{2} \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{\Omega} \|\dot{\phi}\|^2 \,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \\ &= \frac{\rho h}{2} \int_{\Omega} \|\dot{\mathbf{r}}\|^2 \,\mathrm{d}x\,\mathrm{d}y + \frac{\rho h^3}{24} \int_{\Omega} \|\dot{\mathbf{d}}\|^2 \,\mathrm{d}x\,\mathrm{d}y. \end{aligned} \tag{50}$$

The first term above corresponds to the standard translational kinetic energy associated with the midsurface positions and is often [Bergou et al. 2006] discretized using a lumped mass matrix. The second term accounts for the kinetic energy associated with the director field. In Appendix B we construct the corresponding discretized kinetic energies for the second term for the **DCS** and **BAC** models respectively as

$$\begin{aligned} E^{\mathbf{DCS}}_{\text{kinetic}} &= \sum_{\text{edges } ij} \frac{\rho h^3 \left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)}{72}\, \dot{\zeta}_{ij}^2, \\ E^{\mathbf{BAC}}_{\text{kinetic}} &= \sum_{\text{edges } ij} \frac{\rho h^3 \left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)}{72} \frac{\dot{\zeta}_{ij}^2}{\cos^4 \zeta_{ij}}, \\ \zeta_{ij} &= \frac{\theta_{ij}}{2} + \epsilon_{ij}\alpha_{ij}. \end{aligned} \tag{51}$$

For thin shells, where the thickness h is small, the inertial contribution associated with $\dot{\theta}_{ij}$ scales as $O(h^3)$ and is therefore two orders of magnitude smaller than the leading translational inertia given by the first term in Equation (50). Consequently, we drop all inertial terms involving $\dot{\theta}_{ij}$. In addition, we treat the factor $\cos^4 \zeta_{ij}$ as a constant evaluated at the rest configuration, i.e., $\cos^4 \bar{\zeta}_{ij}$, which yields the final discrete kinetic energy expressions

$$\begin{aligned} E^{\mathbf{DCS}}_{\text{kinetic}} &= \sum_{\text{edges } ij} \frac{\rho h^3 \left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)}{72}\, \dot{\alpha}_{ij}^2, \\ E^{\mathbf{BAC}}_{\text{kinetic}} &= \sum_{\text{edges } ij} \frac{\rho h^3 \left(\bar{A}_{ijk} + \bar{A}_{ijk'}\right)}{72} \frac{\dot{\alpha}_{ij}^2}{\cos^4 \bar{\zeta}_{ij}}. \end{aligned} \tag{52}$$

As DSO is a second-order approximation of DCS we use the same kinetic energy $E^{\mathbf{DCS}}_{\text{kinetic}}$ for this model.

7.3 Solving Discrete-Director KL Shell Problems

With the kinetic energies derived, we have all the ingredients to timestep dynamic simulations using the DCS, DSO and BAC models. For robust and efficient implicit timestep solving we next cover a few algorithmic components necessary for practical equilibrium and implicit time integration solves of these discrete Cosserat models.

Termination Criteria. When it comes to efficient Newton-based solves of physical systems it's critical to have a termination criteria in meaningful *and* consistent physical units. This enables defining stopping accuracy tolerances to be calibrated to application. Given a timestepping energy, E_t (often referred to as an incremental potential), we wish to determine convergence by measuring the Newton decrement, $p = (\nabla^2 E_t)^{-1} \nabla E_t$. If a physical system's configuration is defined solely in terms positional DOFs (as for e.g., QB(PL)) then p is naturally in units of distance and so convergence can easily be measured in terms of relative displacement progress or velocity depending on application. However, in the context of DCS, BAC and DSO, units of configuration are mixed in angles (α) and midsurface position ($\mathbf{x}$) and so have different physical units and characteristic scales. To regain a termination measure in terms of displacement (or correspondingly velocity) we compute a distance based measurement of its shell configuration, given a configurational displacement, which we construct jointly from vertex positions and edge directors for each triangle on the midsurface.

Specifically, given a triangle ijk with vertex positions $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$, and edge angles $\{\alpha_{jk}, \alpha_{ik}, \alpha_{ij}\}$, we first compute the associated edge directors $\mathbf{d}_{jk}$ ($\mathbf{d}_{jk;i}$ for BAC) using Equation (65) (Equation (66) for BAC) in Appendix B. The offset edge midpoint $\mathbf{x}^{\text{off}}_{jk}$ is then

$$\mathbf{x}^{\text{off}}_{jk} = \frac{\mathbf{x}_j + \mathbf{x}_k}{2} + \frac{h}{2}\,\mathbf{d}_{jk}, \tag{53}$$

with CR interpolation giving vertex offset positions

$$\mathbf{x}^{\text{off}}_i = \mathbf{x}^{\text{off}}_{ik} + \mathbf{x}^{\text{off}}_{ij} - \mathbf{x}^{\text{off}}_{jk}. \tag{54}$$

With this construction, $\mathbf{x}^{\text{off}}_{jk}$ remains the midpoint of the edge connecting $\mathbf{x}^{\text{off}}_j$ and $\mathbf{x}^{\text{off}}_k$.

During each inner solver iteration, we evaluate $\mathbf{x}^{\text{off}}_i$ for every triangle both before and after the update. The solver is terminated once the maximum change in the offset vertex positions, normalized by the time step size (velocity based measurement), falls below the prescribed tolerance. A geometric interpretation of this offset construction and the resulting termination criterion is covered in Appendix C.

Line-Search Filtering for BAC. The BAC model implicitly constrains the quantity $\theta_{jk}/2 + \epsilon_{jk}\alpha_{jk}$ to lie within $(-\pi/2, \pi/2)$ with the energy naturally diverging as it approaches these limits. Line-search with this energy then requires careful treatment to ensure evaluations are not applied outside these bounds. To do so, we adopt the conservative bisection-based, line-search filtering strategy of Li et al. [2021], originally applied for strain-limiting barriers. To do so, we first find an initially feasible stepsize for any other barrier-based energies applied, such as contact barriers and strain limiters. We then iteratively halve our search-direction step length if an angle-range violation is detected at the current length. We repeat until both energy decrease and angle admissibility are obtained.

8 Energy Convergence Benchmark

One gold standard for evaluating a bending model is its accuracy: whether the discrete model converges to the smooth model as the resolution increases. In this first benchmark, we perform this test by prescribing the rest and deformed midsurface geometry for simple shell shapes and comparing the energy reported by the discrete bending model to the exact answer, for both structured and unstructured meshes spanning four orders of magnitude of mesh resolution. Table 4 summarizes our findings: we report the energy computed using each model for the mesh of highest resolution for each test case, and label whether in each case the model qualitatively converges

Table 4. Bending energy reported by each method that we tested across convergence benchmarks; all results in Joules. This table lists results for each energy evaluated on the finest-resolution mesh. See Section 8 for convergence analysis of each method on each example and following Figures for behavior under mesh refinement. The DS(θ) and DS(tan) models have been combined into one column since they give identical results up to the precision reported in the table. Convergence results are summarized here with color-coding: green methods show evidence of convergence under refinement; red methods do not converge under refinement; blue results are unclear (see text for details). Results in *italics* are "off-label" uses of a method outside their domain of claimed applicability.

		Squared Curvature energy							**Koiter energy**					
Experiment	**Variant**	Analytic	QB(CR)	QB(PL)	QB(int)	DS(θ) DS(tan)	CH(ES)	CH(FS)	Analytic	DKT	M-e. Av.	DSO	DCS	BAC
Cylinder	Regular	$5.40 \cdot 10^{-2}$	$1.61 \cdot 10^{-1}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$1.61 \cdot 10^{-1}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.40 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$
	Irregular	$5.40 \cdot 10^{-2}$	$1.89 \cdot 10^{-1}$	$5.51 \cdot 10^{-2}$	$5.51 \cdot 10^{-2}$	$1.89 \cdot 10^{-1}$	$6.31 \cdot 10^{-2}$	$6.06 \cdot 10^{-2}$	$5.40 \cdot 10^{-2}$	$5.40 \cdot 10^{-2}$	$6.74 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$
	4× Aniso.	$5.40 \cdot 10^{-2}$	7.75	$3.23 \cdot 10^{-1}$	$6.01 \cdot 10^{-2}$	7.76	2.59	$8.18 \cdot 10^{-1}$	$5.40 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$8.70 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$	$5.39 \cdot 10^{-2}$
	Rest Curved	0	*$1.89 \cdot 10^{-1}$*	*$5.51 \cdot 10^{-2}$*	*$5.51 \cdot 10^{-2}$*	$1.72 \cdot 10^{-1}$	$8.40 \cdot 10^{-2}$	$1.14 \cdot 10^{-1}$	$1.40 \cdot 10^{-1}$	$1.40 \cdot 10^{-1}$	$1.59 \cdot 10^{-1}$	$1.40 \cdot 10^{-1}$	$1.40 \cdot 10^{-1}$	$1.40 \cdot 10^{-1}$
Sphere	Irregular	$2.30 \cdot 10^{-1}$	$3.26 \cdot 10^{-1}$	$2.46 \cdot 10^{-1}$	$2.46 \cdot 10^{-1}$	$3.26 \cdot 10^{-1}$	$1.09 \cdot 10^{-1}$	$1.51 \cdot 10^{-1}$	$1.50 \cdot 10^{-1}$	$1.50 \cdot 10^{-1}$	$4.42 \cdot 10^{-1}$	$1.50 \cdot 10^{-1}$	$1.50 \cdot 10^{-1}$	$1.50 \cdot 10^{-1}$
Half-Torus	Regular	$2.09 \cdot 10^{-1}$	$6.25 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$6.25 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.09 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$
	Irregular	$2.09 \cdot 10^{-1}$	$8.96 \cdot 10^{-1}$	$2.13 \cdot 10^{-1}$	$2.13 \cdot 10^{-1}$	$8.96 \cdot 10^{-1}$	$2.99 \cdot 10^{-1}$	$2.49 \cdot 10^{-1}$	$2.09 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.76 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$	$2.08 \cdot 10^{-1}$
Helix	Regular	$9.88 \cdot 10^{-3}$	$2.96 \cdot 10^{-2}$	$9.87 \cdot 10^{-3}$	$9.87 \cdot 10^{-3}$	$2.96 \cdot 10^{-2}$	$9.87 \cdot 10^{-3}$	$9.87 \cdot 10^{-3}$	$9.88 \cdot 10^{-3}$	$9.87 \cdot 10^{-3}$	$9.87 \cdot 10^{-3}$	$9.86 \cdot 10^{-3}$	$9.86 \cdot 10^{-3}$	$9.86 \cdot 10^{-3}$
	Irregular	$9.88 \cdot 10^{-3}$	7.03	$2.70 \cdot 10^{-1}$	$1.25 \cdot 10^{-2}$	7.03	2.34	$7.25 \cdot 10^{-1}$	$9.88 \cdot 10^{-3}$	$9.87 \cdot 10^{-3}$	$7.56 \cdot 10^{-1}$	$9.93 \cdot 10^{-3}$	$9.93 \cdot 10^{-3}$	$9.93 \cdot 10^{-3}$

(see below) to the analytic energy under refinement. We follow this with plots for models' behavior across resolution, with reference images of coarser-resolution meshes, for each test case. Source code for reproducing all experiments is available in the supplemental materials.

8.1 Methodology

Choice of Energy Density. For all experiments in this section, we use $h = 10^{-4}$ m, with Young's modulus $E = 100$ GPa and Poisson's ratio $\nu = 0.3$, giving $\alpha = \frac{E\nu}{1-\nu^2}$, and $\beta = \frac{E}{2(1+\nu)}$. As covered above in Sections 3 and 4 we evaluate each model using either the Squared Curvature energy or Koiter energy as appropriate.

Computing Energy. For models that depend only on vertex position, calculating elastic energy is a straightforward evaluation of the formulas in Section 4.

Models involving additional director degrees of freedom require more care. For these methods, we hold vertex positions fixed and then solve for the values of the extra director DOFs that minimize the total bending energy of the shell. We minimize energy using Newton's method and terminate when the energy gradient norm is below 10^{-5}. For test cases that are rest-curved, we also need values for the director DOFs in their rest configuration. We solve for these by minimizing the elastic energy of the rest surface with respect to the rest directors, treating $\bar{\mathbf{II}}$ as zero.

In all of our test cases in this benchmark, the deformed configuration of the shell is an isometric deformation of the rest state. However, due to discretization error, triangles on the deformed midsurface mesh are not always exactly congruent to their counterparts on the rest mesh. This error vanishes in the limit of mesh refinement. Note that in this benchmark we solely consider bending energy models. For discrete shell models that separately report a stretching and bending energy, we do not include stretching contributions, but do note that stretching would contribute (a small amount of) additional elastic energy due to this discretization error.

Mesh Generation. We procedurally generate the meshes used for each test case; see below for details. In all cases, we parameterize the mesh by (approximate) triangle area A, which allows us to probe the behavior of each shell model under spatial refinement by varying A. We test $A = 4^k \cdot 10^{-8}$ square meters, for $k = 0, \ldots, 4$.

Evaluating Convergence. For each test case and each model that we benchmark, we plot energy error as a function of mesh vertex count V on a log-log plot. We declare that a method converges under refinement if there is an obvious pattern of error ϵ decreasing with mesh resolution according to some power law $\epsilon \propto |V|^p$. This determination is not an exact science: error will saturate away from 0 due to the Newton tolerance and limits on floating-point precision, and for the experiments involving irregular meshes, error might vary for different triangulations of the same resolution. We include the raw convergence plots for each test case so that the reader can form their own judgments.

8.2 Cylinder

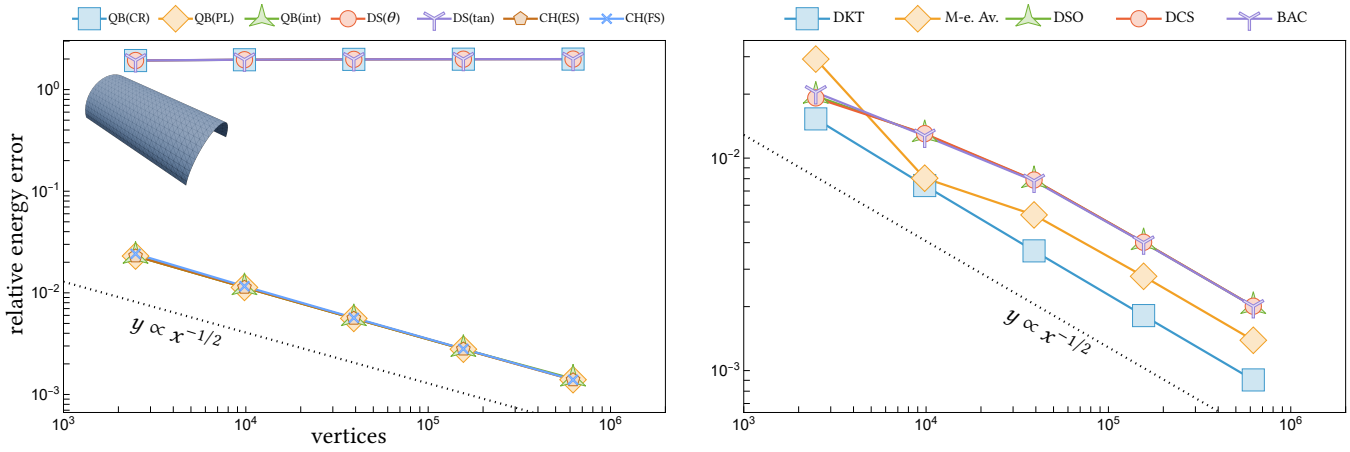
We roll a rectangular plate in its width direction into a half-cylinder to test the shell models on this simplest non-trivial curved configuration. More specifically, we choose the rectangle $\Omega = [0, \pi R] \times [0, H]$ as the parameter domain of the shell midsurface, and set this rectangle also as the midsurface rest configuration. The deformed midsurface is

$$\mathbf{r}(x, y) = \left(R \cos \frac{x}{R}, R \sin \frac{x}{R}, y\right).$$

We use $R = 0.0325$ and $H = 0.122$ m. The exact bending energies of the cylinder are

$$E_{\mathrm{SC}} = E_{\mathrm{B}} = \left(\frac{\alpha}{2} + \beta\right) \frac{\pi h^3 H}{12R}. \tag{55}$$

We test three different strategies for meshing the plate, given a target triangle area A:



Cylinder, Regular, Squared Curvature energy

Cylinder, Regular, Koiter energy

Fig. 3. Log-log plot of relative energy error of each bending model as a function of mesh resolution, for the Regular Cylinder test case. Models on the left discretize the Squared Curvature energy; on the right, the Koiter energy. All methods except for QB(CR), DS(θ), and DS(tan) converge experimentally under this mesh refinement to their respective analytic energies, with error proportional to $1/\sqrt{V}$ for meshes with V vertices.

Regular. We divide Ω into a regular grid of $n \times m$ rectangles, where

$$n = \max\left(1, \left\lfloor \frac{\pi R}{\ell} \right\rfloor\right), m = \max\left(1, \left\lfloor \frac{H}{\ell} \right\rfloor\right)$$

for target edge length $\ell = \sqrt{2A}$. We diagonalize each rectangle (choosing the same diagonal for each rectangle) to form a triangle mesh.

Irregular. We split the boundary of Ω into n and m equal-length line segments in the width and height direction, respectively, then Delaunay-triangulate the interior with target triangle area A. For this and all other experiments involving Delaunay triangulation of 2D domains, we use Shewchuk's Triangle tool [1996] with `qDY` command-line arguments to generate a conforming Delaunay triangulation with no angles smaller than 20 degrees and no Steiner points on the domain boundary.

4× *Anisotropic.* To test the behavior of discrete shell models on anisotropic meshes, we triangulate a domain $\tilde{\Omega} = [0, \pi R] \times [0, 4H]$ using the exact same methodology as in the Irregular experiment, but with target triangle area $4A$. We then uniformly scale the y coordinates of the resulting vertices by one-quarter to yield an anisotropic triangulation of Ω.

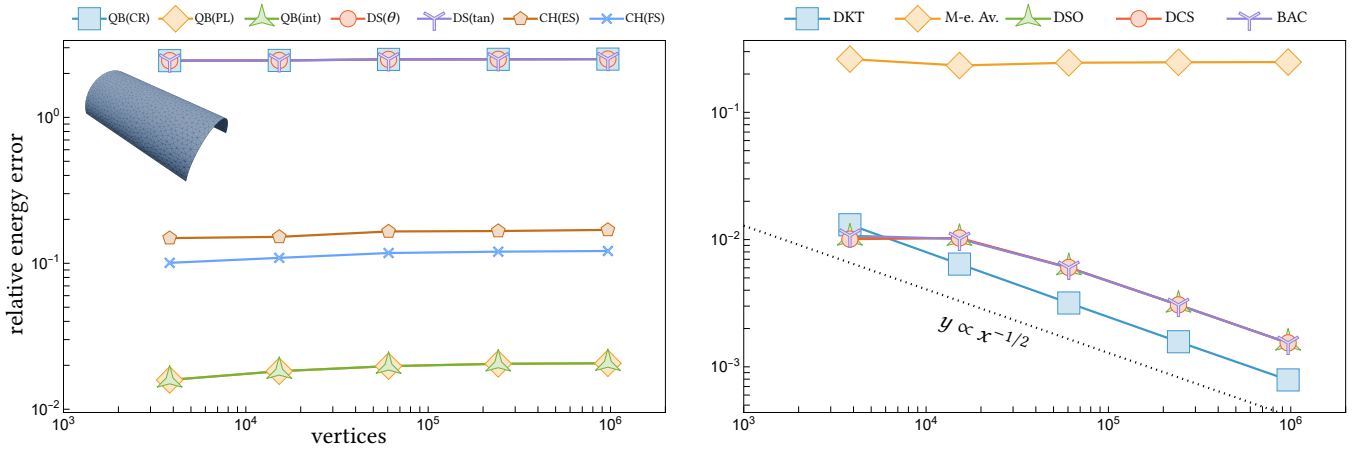
Results. Please see Figures 3, 4, and 5 for convergence plots and mesh visualizations for these test cases. Our main findings were already spoiled in Section 6: none of the Discrete Shells models converge under refinement on *any* of these test cases (not even Regular Cylinder). QB(CR) also fails to converge on all test cases, and the other Quadratic Bending variants only converge under refinement on the Regular test case (but have substantially lower energy error than QB(CR) across the board). CH(ES) and CH(FS) only converge on the Regular test as well. These results are explained by variance in the discrete mean curvature estimated by these methods, as discussed in Section 6.

8.3 Behavior of DS(2nd)

We made a best effort to include DS(2nd) [Le et al. 2023] in this benchmark as well, using a similar methodology as for the Cosserat models: we pin the positions of the vertex nodes, and use Newton's method to optimize for the mid-edge node positions so as to minimize the discrete energy, covered in Section 4.1.6. On the Regular Cylinder test case, the DS(2nd) model predicts $E_{\text{SC}} \approx 1.22 \cdot 10^{-3}$ J, well below the theoretical value of about 5.4×10^{-2} Joules. Moreover, we encountered several practical and theoretical difficulties in benchmarking DS(2nd). We review these below.

Under-integration of Face Contribution. The face contribution E_{faces} to the discrete bending energy (see Equation (21)) involves integrating an energy density over each patch that is a rational function of the barycentric coordinates (due to the normalization of the surface normal vector in the second fundamental form; the overall energy density has a degree-four numerator over a quadratic denominator). Unfortunately, unlike in the DKT model where the energy density is polynomial, there is no choice of quadrature scheme for which numerical quadrature is exact. For the three-point scheme recommended by Le et al. [2023], we observe that the faces tend to shape themselves so that curvature is concentrated at the patch edges (away from the quadrature points) leading to Equation (21) severely underestimating the true Squared Curvature energy contributed by the patch. We did not systematically test alternate higher-order quadrature schemes, but did try a few ad-hoc experiments with seven- and thirteen-point quadrature that include points on the triangle boundary—unfortunately these did not fix the problem, as curvature shifted to wherever the quadrature points didn't cover.

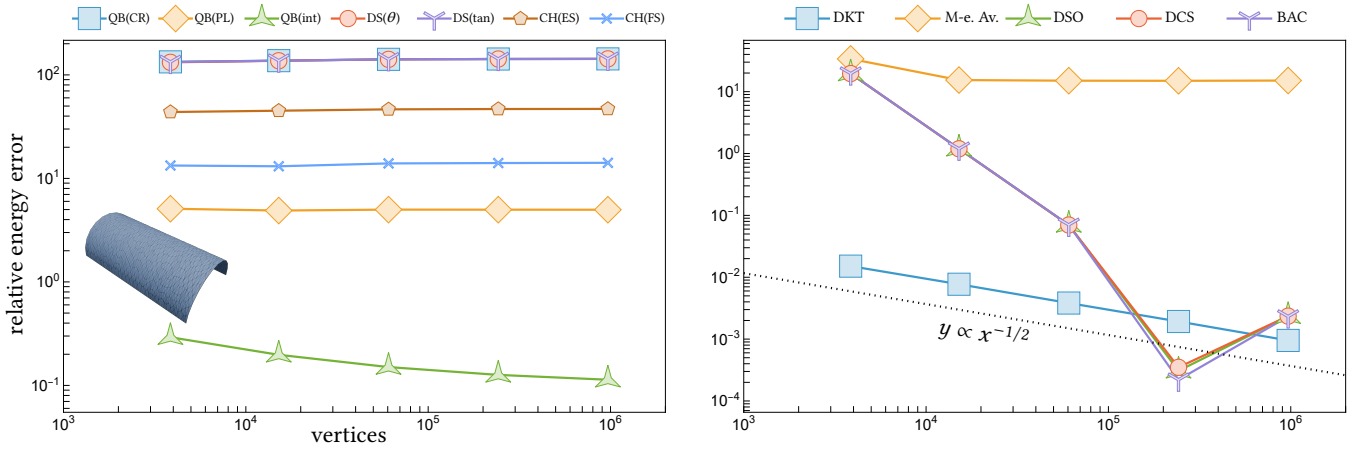
Numerical Instability at Edge Quadrature Points. We observed a related numerical issue when computing the edge contribution to bending energy (Equation (24)): the surface normals $\hat{\mathbf{n}}(s, t)$ required by this term become numerically unstable if curvature concentrates



Cylinder, Irregular, Squared Curvature energy

Cylinder, Irregular, Koiter energy

Fig. 4. Log-log plot of relative energy error of each bending model as a function of mesh resolution, for the Irregular Cylinder test case. Only DKT, DSO, DCS, and BAC converge under mesh refinement.



Cylinder, 4× Anisotropic, Squared Curvature energy

Cylinder, 4× Anisotropic, Koiter energy

Fig. 5. Log-log plot of relative energy error of each bending model as a function of mesh resolution, for the 4× Anisotropic Cylinder test case. Most methods start with substantially higher error at low mesh resolution here (except, surprisingly, DKT) but the methods that converge under this refinement in the Irregular test case do so here as well. From the data it is not impossible that QB(int) converges, very slowly, as well; though this possibility seems unlikely given that the method does not converge under refinement for isotropic irregular meshing.

at the patch edges and the surface parameterization $\mathbf{r}(s,t)$ degenerates there. As a workaround to avoid this instability, we tried adding a barrier potential to the optimization that prevents $\left\|\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial \mathbf{r}}{\partial t}\right\|^2$ from approaching zero at the edge quadrature points.

We expect that a membrane energy term could replace this barrier in a practical simulation; however, the membrane contribution would need to be carefully designed to sample the stretching of the surface near the E_{edges} quadrature points.

Geodesic Curvature of Patch Boundaries. Le et al. [2023] elegantly extend the "tube formula" of Cohen-Steiner and Morvan [2003] to second-order patches, and derive an edge contribution to the Squared Curvature energy based on the surface area swept out by the surface normals as they rotate from one patch to its neighbor along their shared boundary. Equation (24) as an approximation of this edge contribution works well when patch boundaries are close to straight lines; however, it does not consider the geodesic curvature of the patch boundaries, which also increases the surface area of Cohen-Steiner and Morvan [2003]'s "tube." As a consequence, when optimizing the midedge node positions to minimize discrete curvature, the midsurface takes on a scalloped configuration that exploits this missing bending contribution. See Figure 6 for a representative example.

Behavior on Regular Cylinder. Figure 6 shows the minimizer of discrete Squared Curvature energy for the DS(2nd) model on a coarse

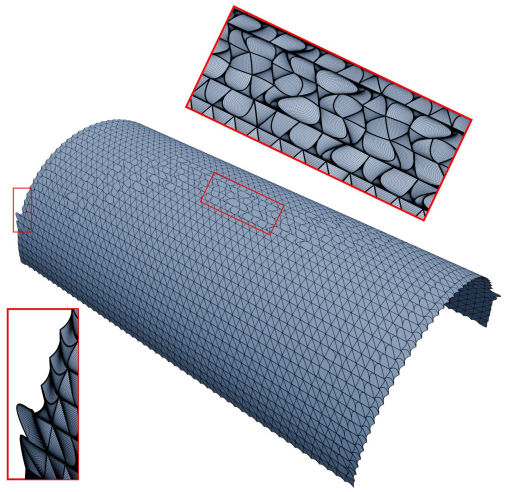


Fig. 6. We minimize the bending energy reported by the DS(2nd) model on the Regular Cylinder test case with respect to midedge nodal position, pinning the vertex positions in place. Due to under-integration of the face bending contribution, and since geodesic curvature does not contribute to the edge bending contribution, the surface patches contort into "scalloped" configurations with curvature concentrated at patch boundaries (See the zoomed-in regions). The discrete bending energy of the cylinder is consequently over an order of magnitude lower than the analytic Squared Curvature energy.

instance of the Regular Cylinder test case; we fix vertex positions and optimize midedge positions, as described above. The midsurface mesh illustrates the scalloping of the midsurface to both increase geodesic curvature of the patch boundaries as well as concentrate curvature there to exploit under-integration in Equation (21).

We conclude that additional research is needed addressing the above issues before DS(2nd) can be meaningfully benchmarked; we exclude it from the remaining test cases.

8.4 Rest Curved Cylinder

For our final cylinder test case, we prescribe a rest state for the cylinder midsurface that is *curved* in the direction perpendicular to the bending direction in the deformed state:

$$\bar{\mathbf{r}}(x, y) = \left(R\cos\frac{y}{R}, R\sin\frac{y}{R}, x\right).$$

We mesh the parameter domain of this cylinder identically to the Irregular case above.

Since the rest and deformed cylinder have identical mean curvature, the Squared Curvature energy vanishes for this test case. The exact Koiter energy is

$$E_{\mathrm{B}} = \frac{(\alpha+\beta)\pi h^3 H}{6R}. \tag{56}$$

See Figure 7 for convergence plots. Both DS(θ) and DS(tan) converge to spurious bending energies much closer to the energy of the rest-flat cylinder than to zero. All discrete models implementing the Koiter energy converge to the Koiter ground truth under refinement, except for M-e. Av. (which struggles to converge under refinement on all test cases involving irregular meshes).

8.5 Sphere

Our next test case is a spherical shell of radius 0.03 meters. We set the sphere to be rest-flat (so that the sphere has residual bending stress, like an orange peel, but no membrane stress). Note that it's impossible to develop the sphere into the plane in a way that relaxes the bending energy without stretching the shell (even if the sphere were sliced open); nevertheless, we can measure the bending energy of the sphere in its current configuration. The exact energies for the sphere are:

$$E_{\mathrm{SC}} = \frac{2}{3}(\alpha+2\beta)\pi h^3, \quad E_{\mathrm{B}} = \frac{2}{3}(\alpha+\beta)\pi h^3. \tag{57}$$

To mesh the sphere, we uniformly sample its surface and then compute the convex hull of the sampled points; the resulting polyhedron (which almost certainly only has triangular faces) is intrinsic Delaunay [Brown 1979]. We plot the relative energy error as a function of mesh resolution in Figure 8. Methods behave similarly on this test case as they do on the irregular cylinder: only DKT, DSO, DCS, and BAC converge under refinement. Interesting, DCS predicts the true energy to within 10^{-5} relative error even on meshes with only a few thousand vertices. We are not sure what explains DCS's remarkable performance on this test case (only).

8.6 Half-Torus

We slice a torus in half through its outer equator (like a bagel) and set the resulting half-torus shell to be rest-flat. We use the rectangle $\Omega = [0, 2\pi R]\times[0, \pi r]$ as the parameter domain of the shell midsurface, for torus outer and inner radii $R = 0.03$ and $r = 0.015$ meters. The deformed midsurface is given by

$$\mathbf{r}(x, y) = (R + r\cos y\cos x, R + r\cos y\sin x, r\sin y).$$

The exact energies for the torus are

$$E_{\mathrm{SC}} = E_{\mathrm{B}} = \frac{\pi^2 R^2(\alpha+2\beta)}{12r\sqrt{R^2-r^2}}. \tag{58}$$

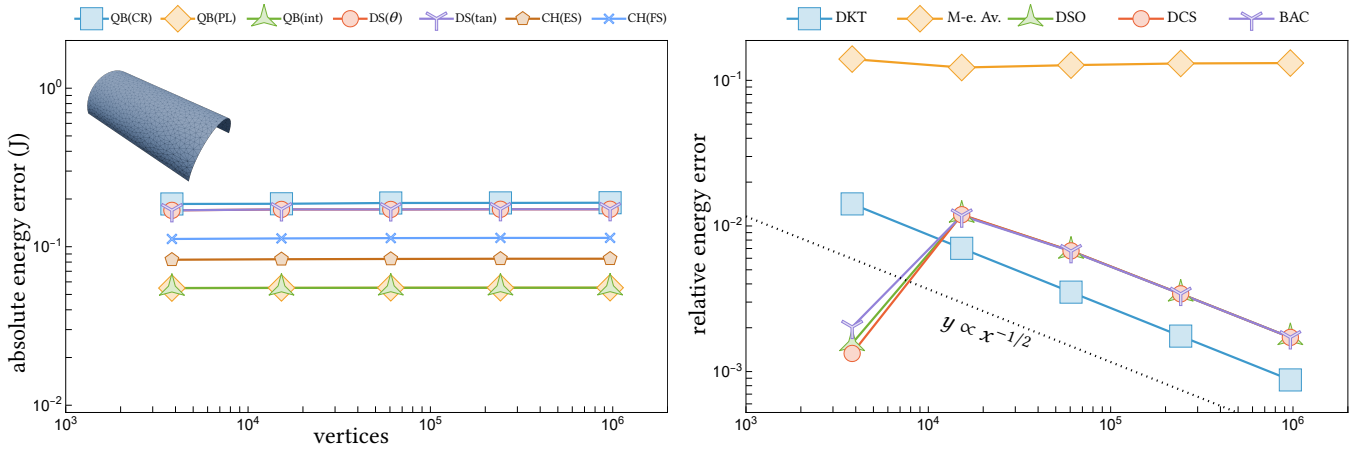
Note that the *total* Squared Curvature energy is the same as the total Koiter energy; this coincidence is due to the symmetry of the half-torus and the Gauss-Bonnet theorem; the energy *densities* differ at most points.

We test two different meshings of the half-torus, Regular and Irregular. We triangulate Ω using the exact same methodology as for the Regular and Irregular Cylinder test cases above, then map each vertex of the triangulation to its image under $\mathbf{r}$ to mesh the half-torus midsurface. As in the Sphere test case, we set the half-torus to be rest flat.

See Figures 9 and 10 for convergence plots. This half-torus, unlike the half-cylinder or sphere, has curvature that varies along the surface. Moreover, the half-torus mesh is not intrinsic Delaunay. Nevertheless, almost all models behave similarly on this example as they do on the Regular and Irregular Cylinder.

8.7 Helix Tangent Developable

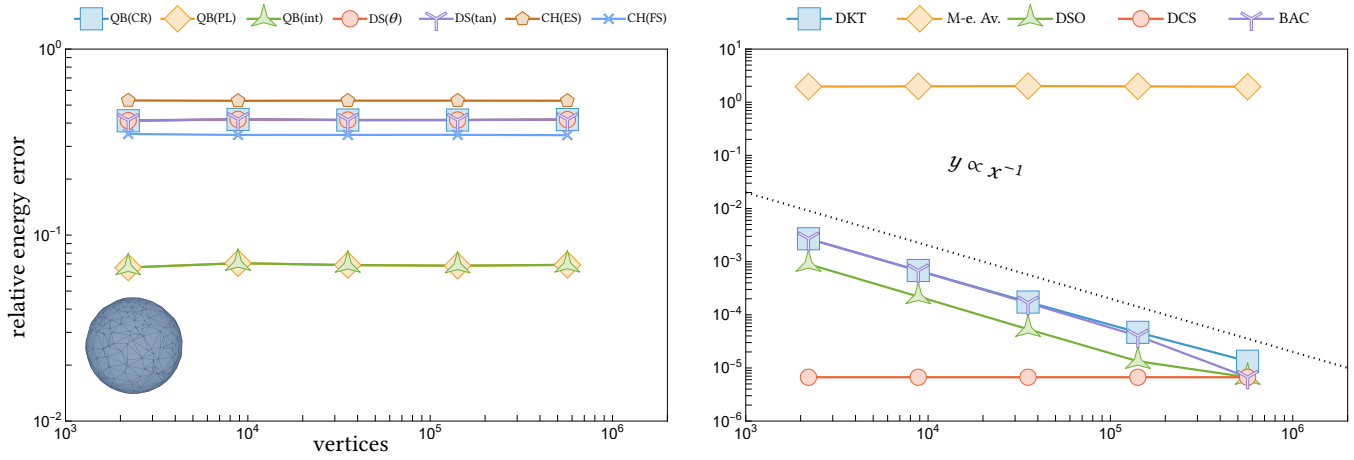
We include a second developable test case, beyond the half-cylinder: a tangent developable surface whose rulings are rays along the



Cylinder, Rest Curved, Squared Curvature energy

Cylinder, Rest Curved, Koiter energy

Fig. 7. Log-log plot of energy error of each bending model as a function of mesh resolution, for the Rest Curved Cylinder test case. Since the analytic Squared Curvature energy is zero for this test case, we report absolute rather than relative error for models that discretize it. We include the quadratic bending models for completeness, but note that there is no expectation that "off-label" use of these models will give reasonable energy estimates since they fundamentally assume a flat rest state. The discrete shell models also fail to converge to zero energy, however. Regarding the Koiter energy models: many are surprisingly accurate for the coarsest mesh in this test case. These models overestimate the exact energy for very coarse meshes, then underestimate at medium resolution, before converging to the exact energy from below. The coarsest data point is at a "lucky" spot near where the discrete energy crosses the true value.



Sphere, Squared Curvature energy

Sphere, Koiter energy

Fig. 8. Log-log plot of energy error of each bending model as a function of mesh resolution, for the Sphere test case. All models that discretize the Koiter energy, except for M-e. Av., converge under refinement; interestingly, they do so more quickly on this test case (with error proportional to $|V|^{-1}$ for meshes with V vertices) than most others. Note that the error of the DCS model saturates due to the accuracy tolerance on the Newton solver. None of the models discretizing Squared Curvature energy show evidence of converging under mesh refinement, though QB(PL) (and QB(int)) predict the analytic energy to within 10% error even on coarse meshes.

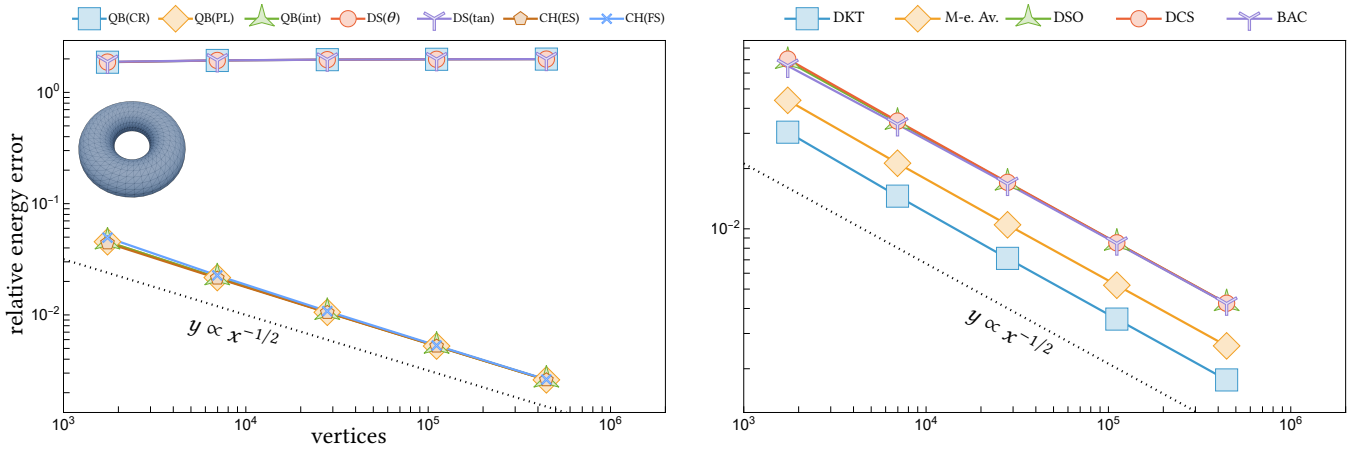
tangent direction of a 3D helix. Given a rectangular parameter domain $\Omega = [0, 2\pi R] \times [0, H]$, the deformed midsurface embedding is given by

$$\mathbf{r}(x, y) = \left(R\cos x - (H + y)\sin x, R\sin x + (H + y)\cos x, \frac{P(Rx+y)}{2\pi R}\right)$$

for P the height of one period of the helix. We use $R = 0.02$, $H = 0.1$, and $P = 0.1$ meters.

This midsurface can be isometrically developed into the plane; the midsurface rest configuration is given by

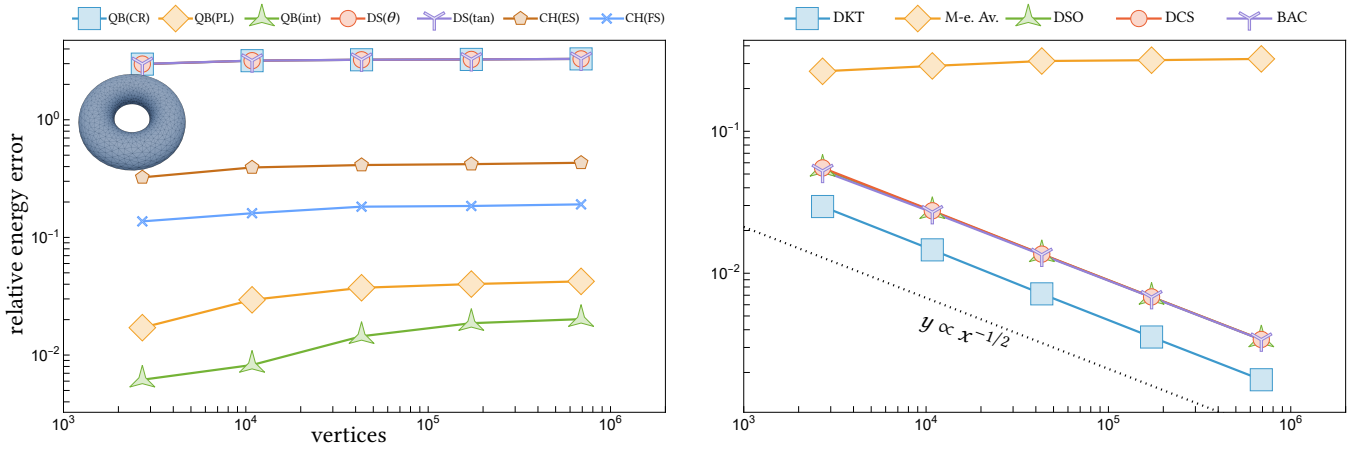
$$\bar{\mathbf{r}}(x, y) = \left(\frac{\alpha^2}{R}\cos\frac{Rx}{\alpha} - \frac{H+y}{R}\alpha\sin\frac{Rx}{\alpha}, \frac{\alpha^2}{R}\sin\frac{Rx}{\alpha} + \frac{H+y}{R}\alpha\cos\frac{Rx}{\alpha}\right)$$



Half-Torus, Regular, Squared Curvature energy

Half-Torus, Regular, Koiter energy

Fig. 9. Log-log plot of energy error of each bending model as a function of mesh resolution, for the Regular Half-Torus test case. Methods behave similarly on this test case as they do on the Regular Cylinder.



Half-Torus, Irregular, Squared Curvature energy

Half-Torus, Irregular, Koiter energy

Fig. 10. Log-log plot of energy error of each bending model as a function of mesh resolution, for the Irregular Half-Torus test case. Interestingly, QB(PL) and QB(int) not only fail to converge under refinement; but the error in the energy predicted by these models actually *increases* with mesh resolution.

where $\alpha = \sqrt{R^2 + \frac{P^2}{4\pi^2}}$. The exact energies of this tangent developable are

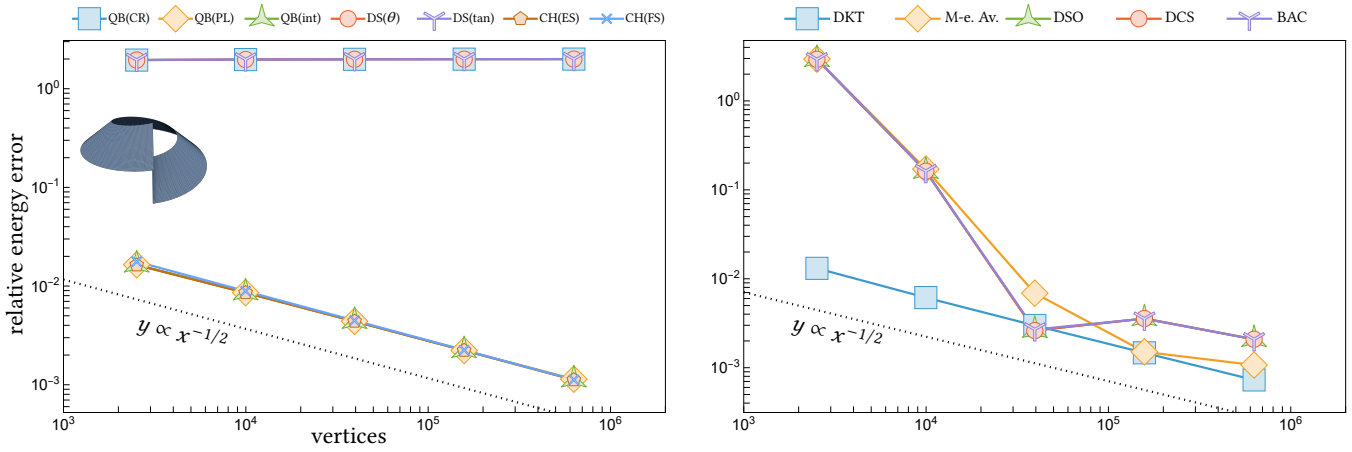
$$E_{\mathrm{SC}} = E_{\mathrm{B}} = \frac{h^3 P^2 (\alpha + 2\beta) \log 2}{24\sqrt{P^2 + 4\pi^2 R^2}}. \tag{59}$$

We test both a Regular and Irregular mesh. As with the Half-Torus test case, we first mesh the parameter domain Ω (using the exact same methodology as for the Cylinder test case) and then compute a deformed and rest mesh by mapping each vertex to its image under $\mathbf{r}$ and $\bar{\mathbf{r}}$ respectively. Note that neither mesh is Delaunay.

Convergence plots are shown in Figures 11 and 12 and largely mirror the results on the half-torus. Note that in this test case, as well as in the 4× Anisotropic Cylinder which also had a non-Delaunay rest-flat mesh, the QB(int) variant predicts the Squared Curvature energy with about 5× less error than QB(PL). Given that QB(int) requires only a small additional one-time cost over QB(PL), it should be used in all cases where the rest mesh is suspected to be non-Delaunay.

9 Creased Vee Benchmark

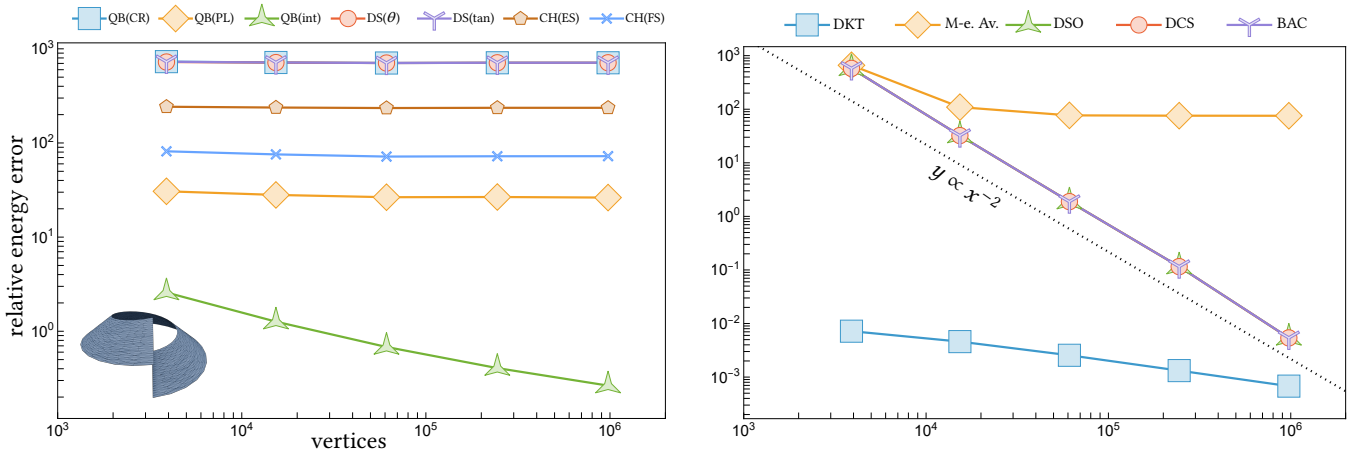
The previous benchmark focused on convergence under mesh refinement. By construction, those experiments probe the behavior of the discrete bending models in situations where the geometric features of the underlying midsurface are well-resolved by the triangulation. In practical simulation applications, and especially in scenarios involving crumpling, folding, or creasing, aliasing of the midsurface is common, where only a few triangles represent significant amounts of local curvature.



Helix, Regular, Squared Curvature energy

Helix, Regular, Koiter energy

Fig. 11. Log-log plot of energy error of each bending model as a function of mesh resolution, for the Regular Helix test case. Methods behave similarly on this test case as they do on all other examples with regular meshes.



Helix, Irregular, Squared Curvature energy

Helix, Irregular, Koiter energy

Fig. 12. Log-log plot of energy error of each model as a function of mesh resolution, for the Irregular Helix test case. QB(PL) and QB(int), if they converge at all, do so very slowly. As usual, all of the Koiter models converge under refinement except for M-e. Av.; DKT has lowest energy error but the other models converge surprisingly quickly (with error proportional to $1/|V|^2$ for meshes with $|V|$ vertices.)

To better understand behavior in these sharp bending regimes we next analyze models with a Creased Vee testing benchmark. We test how gracefully each discrete model degrades when simulating large flexure using a mesh that is coarse relative to the local curvature. To that end, we crease a long rectangular strip: we take the region $\Omega = [0, W] \times [0, H]$ as the parameter domain and rest midsurface configuration of the strip, for $W = 0.01$ and $H = 0.1$ meters. We crease the strip along the centerline of its long (H) direction so that the dihedral angle at the crease is ϕ: the deformed midsurface is then given by

$$\mathbf{r}(x, y) = \left(\frac{W}{2} + \left(x - \frac{W}{2} \right) \sin \frac{\phi}{2}, \quad y, \quad \left| x - \frac{W}{2} \right| \cos \frac{\phi}{2} \right). \tag{60}$$

We divide Ω into a regular grid of $n \times m$ rectangles, where

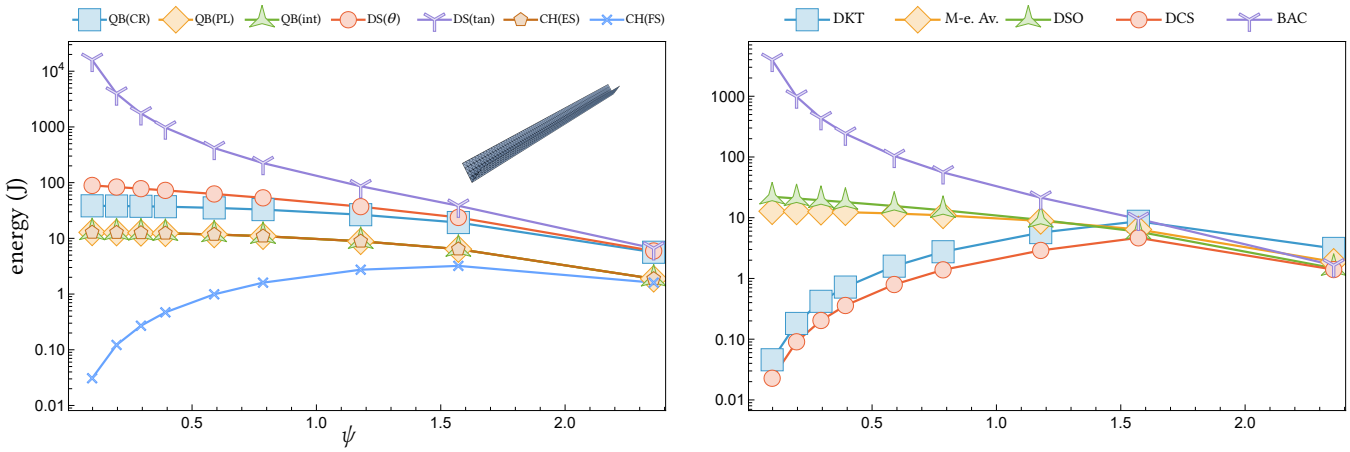
$$n = 2 \max \left(1, \left\lfloor \frac{W}{2\ell} \right\rfloor \right), m = \max \left(1, \left\lfloor \frac{H}{\ell} \right\rfloor \right),$$

for target edge length $\ell = \sqrt{2A}$; note that this construction guarantees a line of mesh edges along the crease. We measure the bending energy reported by each discrete bending model for varying values of ϕ, using the same methodology as in the Regular Cylinder experiment in Section 8.2. Table 5 lists these energies for our smallest value of A.

Convergence Under Refinement. The exact Squared Curvature energy and Koiter energy of the Creased Vee is unbounded. (If we mollify the crease to be an arc of a cylinder of radius r, the total

Table 5. Bending energy reported by each method that we tested on the Creased Vee benchmark; all results in Joules. See Section 9 for methodology and analysis. The ground truth Squared Curvature energy and Koiter energy (for all values of ϕ) is infinity.

	Squared Curvature energy							Koiter energy				
Crease Angle ϕ	QB(CR)	QB(PL)	QB(int)	DS(θ)	DS(tan)	CH(ES)	CH(FS)	DKT	M-e. Av.	DSO	DCS	BAC
$\frac{3\pi}{4}$	5.63	1.88	1.88	5.93	6.60	1.88	1.60	3.12	1.88	1.47	1.40	1.64
$\frac{\pi}{2}$	$1.92 \cdot 10^1$	6.41	6.41	$2.37 \cdot 10^1$	$3.85 \cdot 10^1$	6.41	3.21	8.74	6.41	5.90	4.66	9.56
$\frac{\pi}{4}$	$3.28 \cdot 10^1$	$1.09 \cdot 10^1$	$1.09 \cdot 10^1$	$5.34 \cdot 10^1$	$2.24 \cdot 10^2$	$1.09 \cdot 10^1$	1.60	2.75	$1.09 \cdot 10^1$	$1.33 \cdot 10^1$	1.38	$5.57 \cdot 10^1$
$\frac{\pi}{8}$	$3.70 \cdot 10^1$	$1.23 \cdot 10^1$	$1.23 \cdot 10^1$	$7.27 \cdot 10^1$	$9.72 \cdot 10^2$	$1.23 \cdot 10^1$	$4.69 \cdot 10^{-1}$	$7.21 \cdot 10^{-1}$	$1.23 \cdot 10^1$	$1.81 \cdot 10^1$	$3.59 \cdot 10^{-1}$	$2.42 \cdot 10^2$
$\frac{\pi}{16}$	$3.81 \cdot 10^1$	$1.27 \cdot 10^1$	$1.27 \cdot 10^1$	$8.34 \cdot 10^1$	$3.96 \cdot 10^3$	$1.27 \cdot 10^1$	$1.22 \cdot 10^{-1}$	$1.82 \cdot 10^{-1}$	$1.27 \cdot 10^1$	$2.07 \cdot 10^1$	$9.06 \cdot 10^{-2}$	$9.85 \cdot 10^2$
$\frac{\pi}{32}$	$3.84 \cdot 10^1$	$1.28 \cdot 10^1$	$1.28 \cdot 10^1$	$8.91 \cdot 10^1$	$1.59 \cdot 10^4$	$1.28 \cdot 10^1$	$3.08 \cdot 10^{-2}$	$4.58 \cdot 10^{-2}$	$1.28 \cdot 10^1$	$2.21 \cdot 10^1$	$2.27 \cdot 10^{-2}$	$3.96 \cdot 10^3$



Creased Vee, Squared Curvature energy

Creased Vee, Koiter energy

Fig. 13. Log plot of the energy each model as a function of the Creased Vee's dihedral angle ψ (with $\psi = 0$ when the hinge is clamped completely shut). All models use a regular mesh with 50 268 vertices (In the inset, we show a regular mesh with 801 vertices to illustrate the mesh structure). Models exhibit one of three qualitative behaviors: DS(tan) and BAC diverge as $\psi \to 0$; DKT, DCS, and CH(FS) report energies that decrease after the critical angle $\psi = \pi/2$; and for the other models, the energy increases monotonically as ψ decreases, but converges to a finite value as $\psi \to 0$.

bending energy is given by an energy density of magnitude $O\left(\frac{1}{r^2}\right)$ integrated over a curved region of area $O(r)$.) Although not the main point of this experiment, the approximate bending energy predicted by every model we test does indeed diverge to infinity under mesh refinement when ϕ is held fixed (see supplemental materials for raw data).

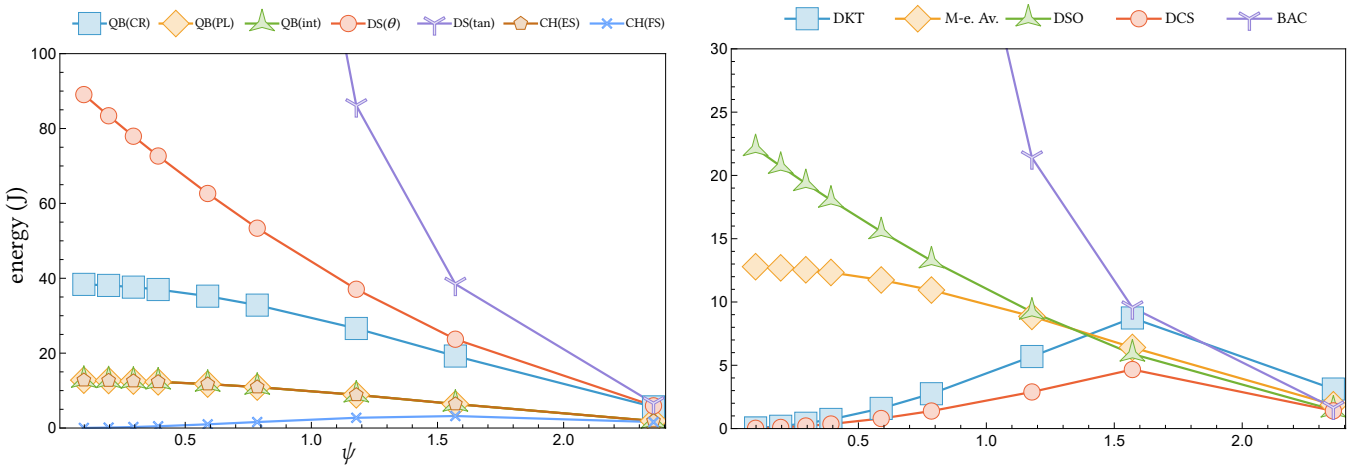
Behavior as $\phi \to 0$. Figures 13 and 14 show log and linear plots of the bending energy as $\phi \to 0$, for a *fixed* mesh resolution. From these results, each discrete models can be placed into one of four groups based on their qualitative behavior:

I *Energy decreases as $\phi \to 0$* (DKT, DCS, CH(FS)): for these three methods, bending energy starts to *decrease* once the crease dihedral angle goes below the critical value $\phi = \frac{\pi}{2}$. In a simulation, this phenomenon manifests as a spurious "magnetic" force that seeks to *further* increase the bending of any discrete hinge that bends past ninety degrees.

II *Energy increases to finite value; force decreases* (QB(CR), QB(PL), QB(int), CH(ES), M-e. Av.): for all Quadratic Bending models, as well as CH(ES) and M-e. Av., bending energy continues to increase as $\phi \to 0$, but at a rate which decreases past an inflection point around $\frac{\pi}{2}$ (see the linear plot in Figure 14). In practice, this derivative decrease implies that the force resisting bending of a discrete hinge *weakens* if the hinge bends past the inflection point, resulting in spurious, sudden collapses of the hinge, for example as applied load increases.

III *Energy increases to finite value; force increases* (DS(θ), DSO): for these models, bending energy and force continue to increase as the dihedral angle increases. However, since the bending energy remains bounded, sufficiently strong forces or applied loads can fully collapse a discrete hinge past its flat-folded configuration where $\phi = 0$.

IV *Energy diverges* (DS(tan), BAC): the bending energy (and force) predicted by the final two models diverge as $\phi \to 0$. The discrete



Creased Vee, Squared Curvature energy

Creased Vee, Koiter energy

Fig. 14. Linear plot of the energy each model as a function of the Creased Vee's dihedral angle ψ. Notice that the derivative of bending energy (and therefore bending force) decreases as $\psi \to 0$ for the M-e. Av. model, as well as for CH(ES) and all forms of Quadratic Bending.

bending energy of these models thus serves as a barrier guaranteeing[2] that a discrete hinge never bends past its flat-folded configuration, no matter the applied force.

In summary, methods in groups I–II *should not be used* if there is any risk of a discrete hinge bending past ninety degrees at any point during the simulation. For methods in group III, bending force alone cannot prevent a discrete hinge from collapsing to or bending past its flat-folded configuration. Correspondingly, as we will see in the next section, they can fail, e.g. under self-contact, to prevent degeneration and so generate artifacts in the midsurface mesh. More generally, please see the next Section 10 for evaluation of practical shell simulation cases where artifacts in the simulation can be sourced from the above-described limitations of models in groups I–III to large local bending.

10 Thin-Shell Simulation Evaluation

So far we have focused on the behavior of bending models with an outlook for their role in KL-shell modeling. In this section we now consider the practical behavior of bending models in the full context of their application, in combination with membrane energy contributions in complex thin-shell simulations and buckling validation.

We study test-case simulation examples for both dynamics and statics, both rest-curved and plate, across diverse materials (see Table 6), ranging from thin aluminum to formed silicone rubber to cotton fabrics, in applications with large deformations, incremental loading, high-speed motion and collisions, frictional contact, and confinements via both applied boundary conditions and kinematically driven colliders. For consistency, in all examples, we pair all bending models with *the same* membrane energy, discretized via the P1-element as covered in Section 4 and Equation 22.

[2]Assuming an implementation of dynamics or statics solver with appropriate line-search filtering preventing discrete tunneling past the folded configuration; see Section 7.3.

All models are simulated in a common test-harness implementation on a 48 GB MacBook Pro M3. We apply the IPC model [Li et al. 2020] for contact processing, a Projected Newton solver [Li et al. 2021] using the Apple Accelerate framework for per-iterate sparse Cholesky linear solves [Apple Inc. 2023], and Intel Threading Building Blocks (TBB) for CPU parallelization [Intel Corporation 2023].

Material	Y (Pa)	ν	ρ (kg/m^3)	h (mm)
Cotton Fabric	10^6	0.3	500	0.1–1
Aluminum (Can)	7×10^{10}	0.33	2710	0.4
Silicone Rubber	$3 - 5 \times 10^7$	0.4-0.47	1000	2-10
Latex Rubber	10^6	0.4	500	1
Paper	3×10^9	0.38	800	0.1

Table 6. Summary of materials used in examples, with Youngs modulus (Y), Poisson ratio (ν), material density (ρ), and thickness (h).

Until now we have quantitatively verified bending energy convergence and evaluated bending energy growth in creasing. Here we next primarily qualitatively analyze the behavior of the leading Discrete-Director KL models (DSO, DCS, and BAC) and QB(PL). Notice that, among all examples in the following test cases, only the mesh in the Paper Cone Crush (Figure 16) is not intrinsically Delaunay. In all other cases, QB(PL) coincides with QB(int). At the same time, although already cautioned by its failure in both prior benchmarks we also include DS(θ), in our analysis here as a baseline for comparison, given its long-standing familiarity in computer graphics.

10.1 Test-Cases

Sphere Drape. We begin with a standard graphics benchmark of a fabric draped on a sphere. We use a 1m square cotton fabric material ($h = 1mm$), discretized with a 160k triangle, Delaunay mesh (See

Figure 18), with C-IPC [Li et al. 2021] strain-limiting augmenting the membrane model. We timestep the dynamics ($\delta t = 0.02s$) of the cloth dropped from a height of 20cm onto the sphere and let the simulation come to a rest. See Figure 18 and our supplemental video.

Ball Crash. To test model behaviors with both high-speed impact and rest-curvature we fire (initial velocity $v_0 = 6m/s$) a structured a 10 cm radius latex rubber material hollow ball at a wall. We discretize with a structured icosphere mesh, 6 subdivisions with 82k triangles, and timestep with $\delta t = 0.0005s$ to capture the rapidly changing phases of the sphere's initial compressive impact, collapse, restoration and rebound. See Figure 20 and our Supplemental video.

Pinned Cloth Drop. In this example we test a slower-speed dynamic simulation exercising folding, buckling and compression. Here a 1m square cloth, cotton fabric material with a thickness of $h = 1mm$, is discretized with a 62k triangle Delaunay mesh (See Figure 15), with C-IPC [Li et al. 2021] strain-limiting augmenting the membrane model. The cloth is oriented vertically with gravity. We pin the cloth's bottom two corners, place a floor 10 cm under it, and simulate the material's dynamics time-stepped at $\delta t = 0.02s$ The cloth collapses under its weight, buckling into a chain of self-colliding rounded folds that land on the floor, roll out and form transverse waves before coming to a final rest. See Figures 15 and 19.

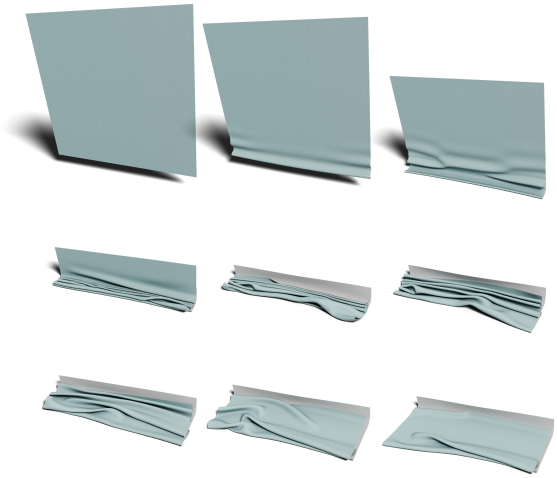


Fig. 15. **Pinned Cloth Drop.** (Sequence ordered top to bottom, left to right.) To test dynamic folding, buckling and compression, we simulate a cloth (here shown with BAC), pinned on its bottom two corners. The cloth initially collapses under its own weight. When it hits the ground it buckles into a chain of self colliding rounded folds that roll out with transverse waves before coming to a rest.

Can Crush. For classic buckling testing with rest-curvature, we compress a 10cm tall open-cylinder thin aluminum "can", discretized with 174k triangle Delaunay mesh (See Figure 21), by fixing it's bottom boundary and incrementally moving the cylinder's top boundary down over successive quasistatic steps, reaching a final crushed state.

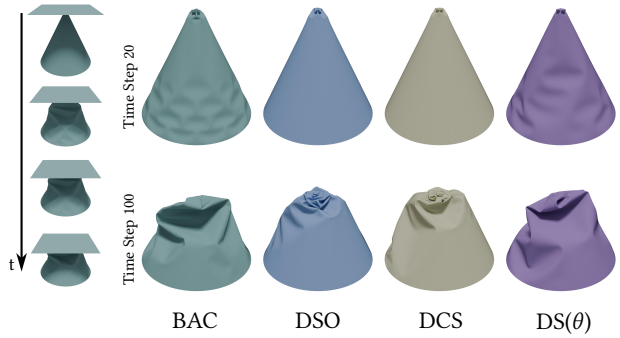


Fig. 16. **Cone Crush.** Crushing a paper cone with kinematically scripted collision plane generates distinctly different crumpling and buckling behaviors across bending models. With their weaker penalties on sharp bends, DSO and DCS form significantly more localized crumpling, afforded by sharp creasing at early stages. In turn, they do not generate the initial symmetric buckling patterns generated by BAC which distributes the initial load without the same tight bending. As the contact plane presses down they end with similarly qualitatively different wrinkle configurations, still characterized by a higher concentration of sharp folding near the apex for DSO and DCS. In contrast, DS(θ) produces wrinkle patterns similar to those of BAC, although with reduced structure during the initial buckling deformation.

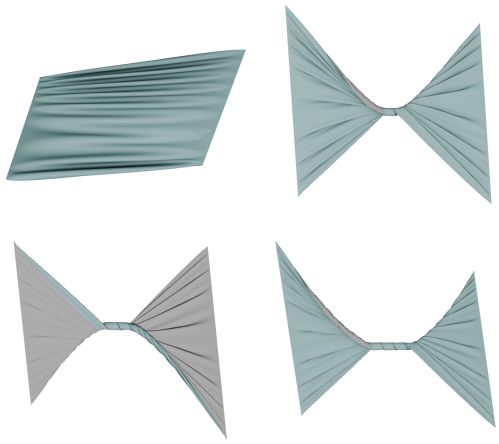


Fig. 17. **Twisted Cloth Sheet.** Tight twisting of a cotton cloth sheet, simulated here with the BAC model.

Paper Cone Crush. To exercise nonstandard buckling patterns we crush a (rest-flat) paper cone (5cm radius, 10cm tall) via frictional contact with a kinematically driven flat plane. We discretize the cone with a 161k triangle, symmetrically structured mesh (See Figure 16) and quasistatically solve for each new configuration as the plane pushes downwards.

Twisted Cloth Cylinder. We next take a 10cm-wide (3.2cm radius) cotton fabric ($h = 0.1$m) cylinder, discretized with a 89k triangle, Delaunay regular mesh and twist it into central braid by rotating

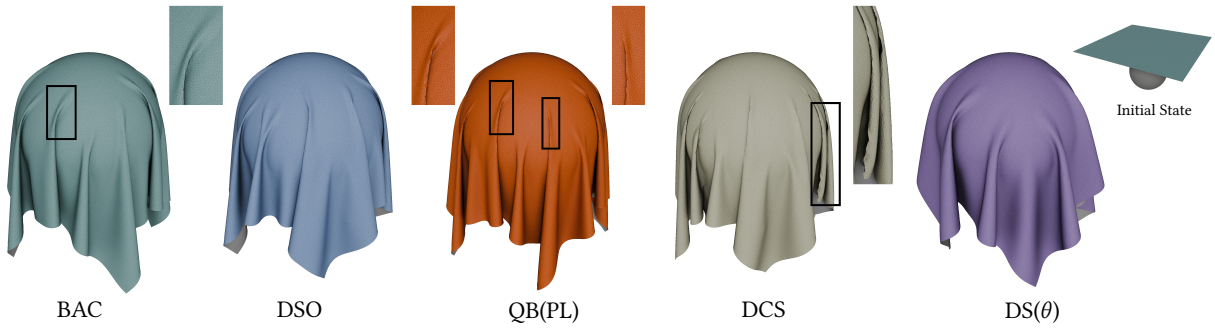


Fig. 18. **Sphere Drape.** As their respective penalties on sharp bending decreases, DSO, QB(PL), and DCS produce progressively sharper folds, accompanied by corresponding locking and aliased wrinkle patterns. In this example, without large forcing of folds, DSO obtains similar folding to BAC, while QB(PL) demonstrates noticeable artifacts, and DCS, which imposes the weakest penalty, generates severely distorted wrinkle patterns. In contrast, DS(θ) yields a different effective material behavior with smoother and noticeably lower frequency wrinkle patterns.

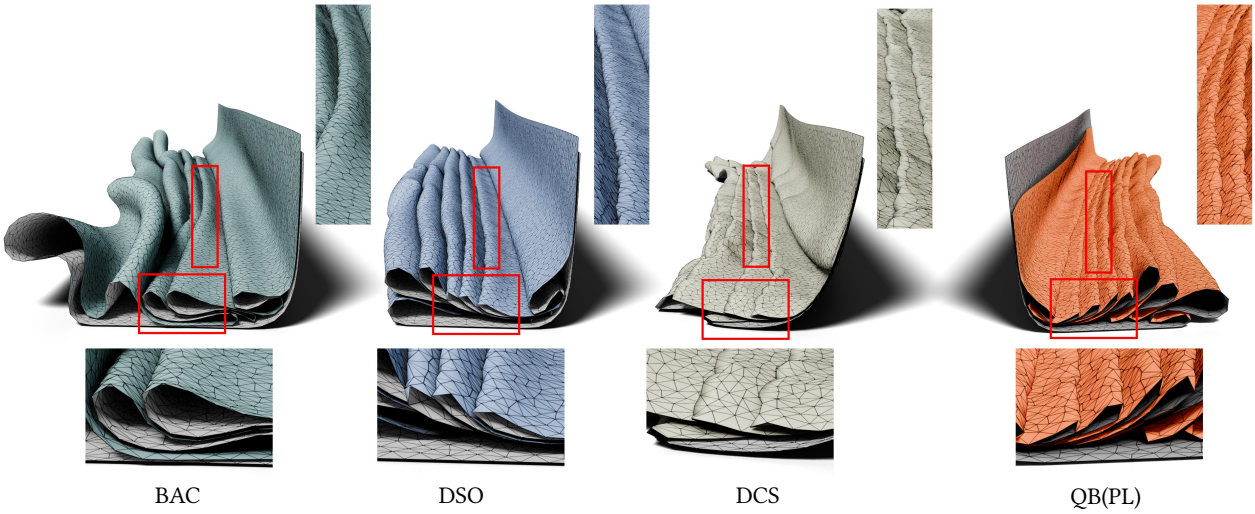


Fig. 19. **Pinned Cloth Drop.** See figure 15 for a description of this simulation sequence. We focus here on a single timestep soon after the fabric's initial impact with the ground. Here again the "impact" of each method's respective penalty on sharp bending is clear: DCS, with weakest penalty folds into entirely flattened wrinkles and soon after locks subsequent dynamics (please see our supplemental video), stronger but finite penalties in DSO and QB(PL) (which buckles rightwards) do not fully lock but still introduce sharp creasing artifacts. In contrast, BAC, with its "built-in" barrier on bending, generates rounded folds and smoother draping.

both ends. Timestepping the dynamics at $\delta t = 0.02$s, twisting first introduces wrinkled stripes, then folding, and finally tightly packed braiding (See Figure 23).

Twisted Cloth Sheet. To introduce tighter braiding, we next twist a flat sheet of the same cotton material as in the above Twisted Cylinder, discretized with a 103k triangle, Delaunay mesh (See Figure 17).

Popper. Here we simulate a classic "popper toy" [Lapp 2008]: a spherical rubber cap that is first slowly pushed inside out (everted) while holding its inner rim (a small inward radial inset at the cap's end) fixed. The cap should bend slightly at the boundary in this inside-out state. The popper is then placed on a flat surface. The expected behavior is a (relatively) slower initial unbuckling or snap-through phase towards the cap's original shape, transitioning to a rapid final impact with the ground that sends the popper flying upwards in its original rest shape. (Toy poppers include a small hole at the top of the popper's cap, as we do here.) In an initial quasistatic phase we simulate pushing a 16k triangle, symmetrically structured popper mesh (10cm radius, 72 degree cap angle) with a silicone rubber ($Y = 10^7$Pa, $h = 10$mm, $\nu = 0.47$) material, inside out. We fix its inner rim and incrementally script the pushing of the popper's geometry through 70% of its original height. We then place the popper on a flat collision surface (floor) and switch to a high-speed dynamic simulation ($\delta = 0.0001s$) to capture the snap-through phase. See Figure 24 and the supplemental video.

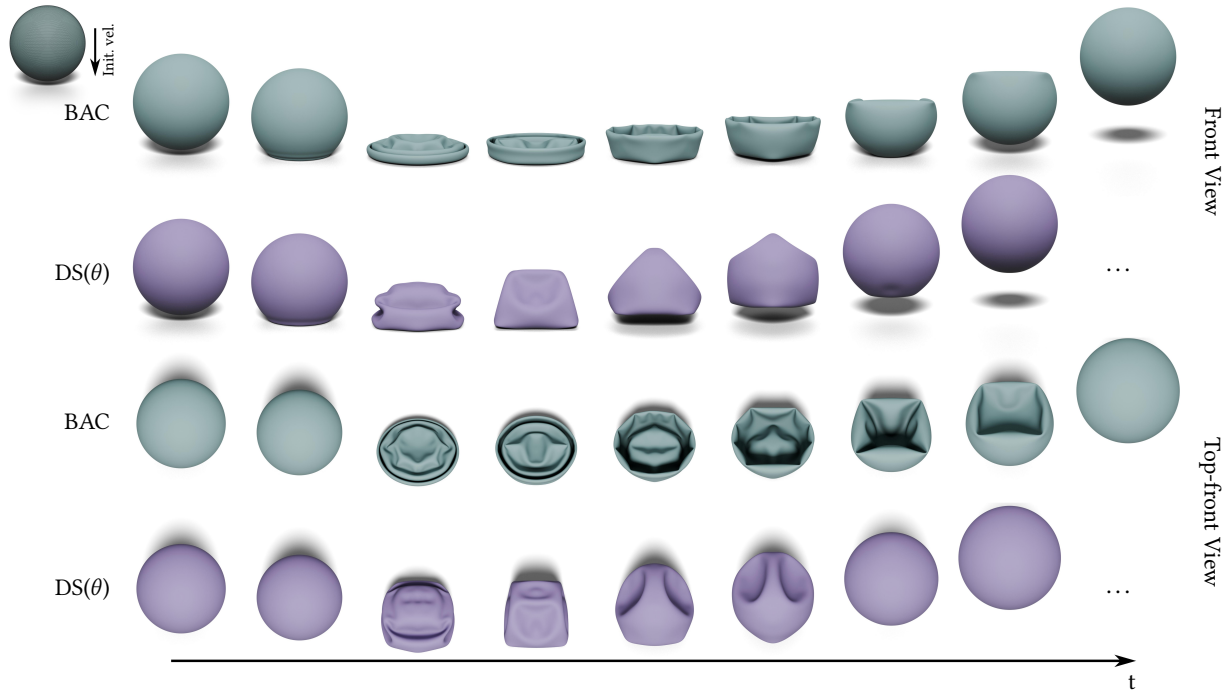


Fig. 20. **Ball Crash.** A thin-shell rubber ball thrown against a wall with a high initial velocity undergoes large initial deformation followed by rebound. Here we show two viewpoints of timesteps generated by BAC and DS(θ) (see also Figure 27, Appendix D). We highlight two differences (please see our video for entire sequence comparisons). First, DS(θ) exhibits qualitatively different material behavior, forming large, squared, blocky patterns without inner collapsing details as generated by BAC (and likewise DCS and DSO). Second, it (again consistent with different effective material behavior) rebounds significantly faster to regain its rest shape.

Corrugated Collision. We simulate a challenging high-speed dynamic scenario with complex collision response: a small heavy rubber ball (radius 3.2cm) is fired at high speed ($v^0 = 30$m/s) into a 80cm length of corrugated silicone rubber hose (average radius 8cm, $Y = 3 \times 10^7$Pa, $h = 2$mm, $\nu = 0.4$). which drives both into a conical "basket" obstacle (fixed collision object). The hose is discretized with a 106k triangle, symmetrically structured mesh (see the inset in Figure 25). On impact the initial collision forms a complex impact pattern in the hose, and then drives the hose into the basket before compressing, rebounding and recovering. Dynamics are timestepped at $\delta t = 5 \times 10^{-4}$s to capture the rapid high-frequency collision phase and the subsequent rebound. See Figure 25.

10.2 Qualitative Analysis

Quadratic Bending (QB(PL)). In studying our thin shell simulation results (please see our supplemental videos) we begin by observing that QB(PL) qualitatively captures quite similar behavior as BAC on a range of challenging rest-flat configurations, such as Pinned Cloth Drop and Sphere Drape. However, even here we see qualitative differences in that the QB(PL) model (similarly to DSO, see below) results in generally sharper wrinkles and folding, (see e.g., the zoomed-in region of Sphere Drape), so that solutions can require greater mesh resolution than BAC. When it comes to rest-curved and curvature dominated behavior ("off-brand" for QB(PL)) we see mixed results. In the Twisted Cylinder (Figure 23), QB(PL) generates a wrinkling pattern sequence quite similar to BAC, although again with sharper fold creases. However, for the other examples, for instance the Can Crush example, the buckling transition is different than BAC's, skipping completely the first symmetric bulging phase, while post-buckle, the initial nucleation is less well defined and the final crushed shape again exhibits sharper creases. We also notice that even if we consider QB(PL) in an "on-brand", rest-flat version of the Can Crush (see Figure 22) the behavior of QB(PL) is still markedly different than the BAC model in these conditions as well. A worse failure, in a rest-curved dynamics scenario, is in the Popper example. Here, when pushed inside out during its initial quasistatic loading phase, the QB(PL)-simulated popper unrolls completely, and so consequently fails in its later dynamic simulation phase to unbuckle and snap-through. Similarly, in the Corrugated Collision we see, at the very beginning of the simulation, that the QB(PL)-simulated hose first unnaturally elongates with the bending model unable to capture the rest-curved-induced geometric stiffness of the hose. After the initial impact occurs, the hose is driven into a sharply creased configuration aligned with the axial direction. Unlike BAC and DS(θ), QB(PL) is then unable to recover its rest shape and is instead maintains both its long creasing and looser, less-structured configuration. On the other hand, unlike DS(θ) (see below), for the high-speed Ball Crash example (again rest-curved dynamics), while certainly exhibiting less detail, the QB(PL) solution

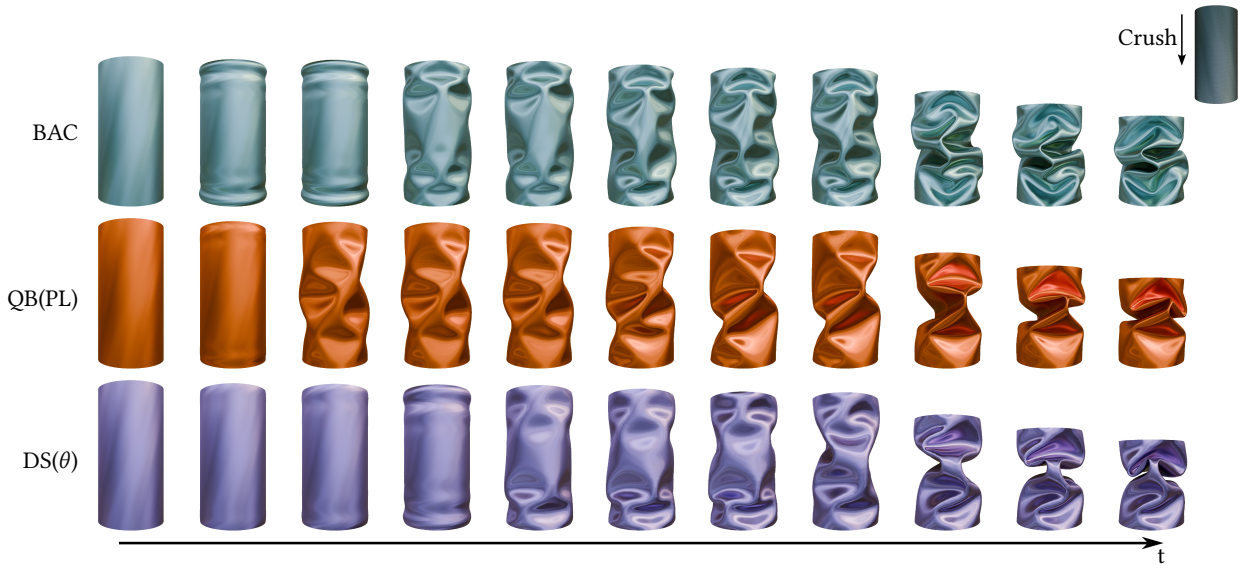


Fig. 21. **Can Crush.** We compress a metallic a thin wall aluminum with incremental loading. Here both BAC and DS(θ) exhibit a clear transition phase before symmetry breaking and ultimately produce finer buckling patterns compared to QB(PL). Note that BAC and DS(θ) are simulated with nonzero rest curvature, whereas QB(PL) by construction assumes a rest-flat configuration. In Figure 22, we further perform an "on-brand" comparison for QB(PL) with BAC using a rest-flat shell. In turn, both BAC and DS(θ) exhibit distinctly different buckling responses: BAC breaks into an earlier transition phase with a structured symmetric wrinkle patterns, while DS(θ) breaks symmetry from bottom to top, leading to more asymmetric buckling and final crumpling behavior. For reference, Figure 28 (Appendix D) presents the same simulation with the DS(tan) model; consistent with its similar discretization to DS(θ), it obtains closely matching behavior.

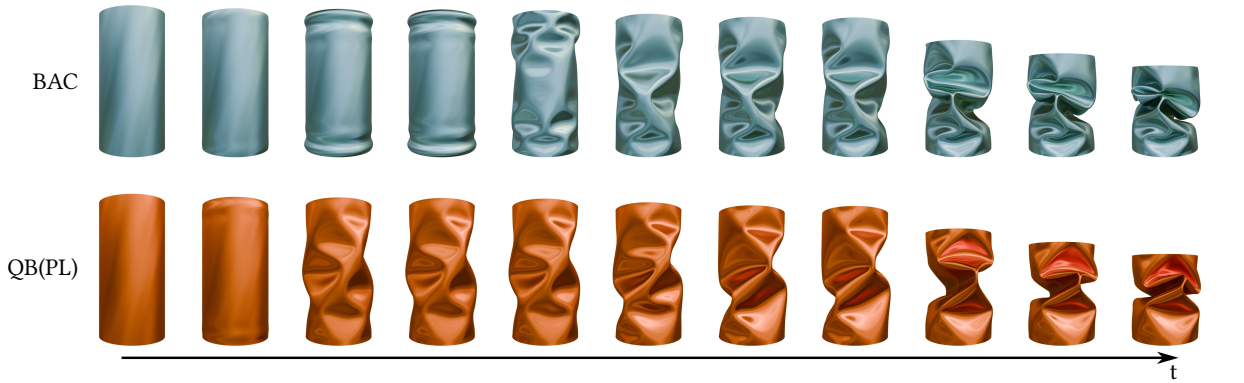


Fig. 22. **Can crush *rest-flat*.** We repeat the Can Crush experiment from Figure 21, using a rest-flat configuration for BAC by setting the rest second fundamental form to zero. In this "on-brand" case for QB(PL), BAC now produces wrinkle patterns closer to those of QB(PL). However, consistent with the rest-curved case, BAC still exhibits an initial transition phase, missed by QB(PL), and subsequently breaks symmetry gradually, buckling from the top and bottom simultaneously.

is qualitatively similar to the Discrete-Director KL models' (DSO, DCS, BAC) solutions.

Discrete Shell (DS(θ)). Unlike QB(PL), DS(θ) captures distinctly different material behaviors from all other bending models, across all examples, irrespective of whether rest-curved or flat. In both the Sphere Drape (Figure 18) and Twisted Cloth Cylinder (Figure 17) we see that DS(θ) obtains a significantly smoother cloth material, with lower-frequency fold patterns than all other models. In the Pinned Cloth Drop (see the supplemental video) we see this even more exaggerated with fewer folds appearing and less dynamics, in contrast to both the Discrete-Director KL (DSO, DCS, BAC) and QB(PL) bending models. DS(θ) also significantly changes function in modeling rest-curved dynamics and incremental loading as well. In the Ball Crash, unlike the Discrete-Director KL (DSO, DCS, BAC) and QB(PL) bending models, DS(θ) generates a very different set of blocky, squared-off shapes throughout the compressive, restorative, and free-flight phases of the high-speed collision (See Figure 20 and our supplemental video). Similarly in the stiffer thin-aluminum Can

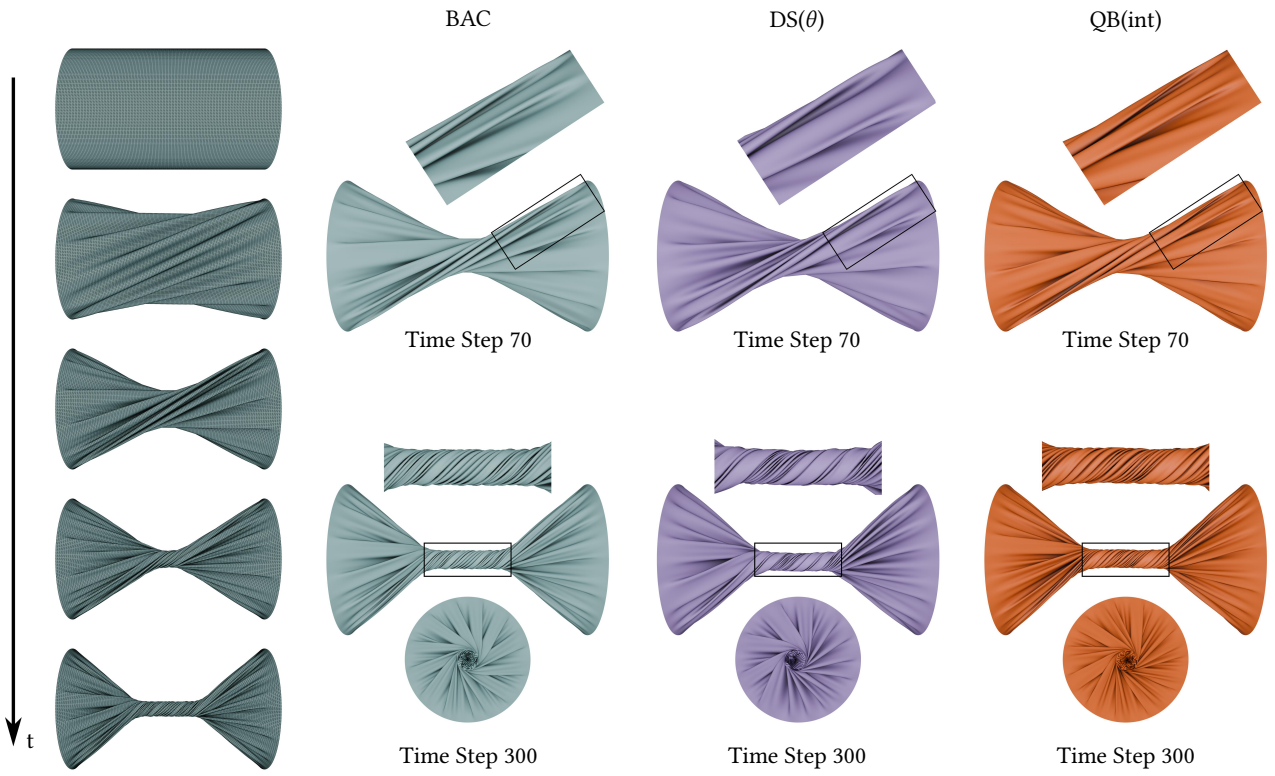


Fig. 23. **Twisted Cloth Cylinder.** We simulate the twisting of a cloth using BAC (*left*), DS(θ) (*middle*), and QB(int) (*right*). In the leftmost column, we show the full simulation sequence of BAC with a wireframe overlay. The three columns on the right visualize two representative time steps obtained across bending models. Here we note that (as might be expected) QB(int) produces generally sharper wrinkles and folds (illustrated in the zoomed-in region in the top row). In contrast, DS(θ) again yields noticeably coarser wrinkles (see the zoomed-in region in the bottom row), with an effectively "thicker" material-type behavior. From the side view, both BAC and QB(int) exhibit closely similar folding patterns, which differ substantially from those produced by DS(θ).

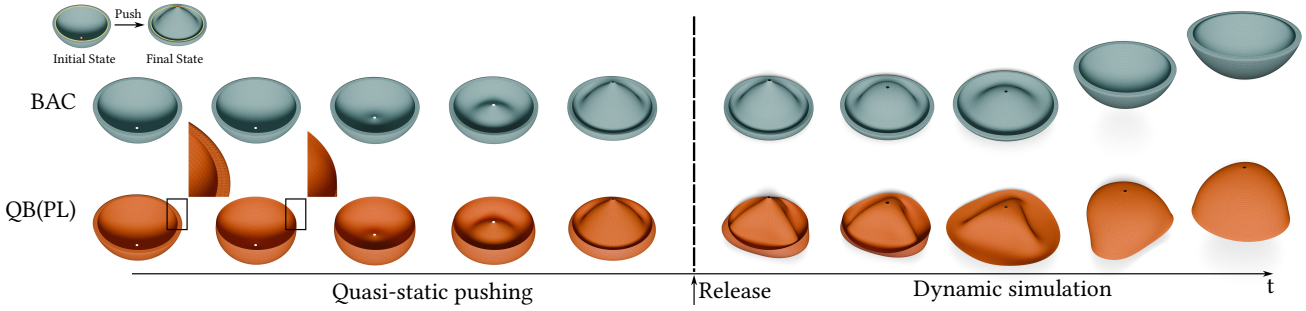


Fig. 24. **Popper.** We quasistatically simulate pushing a a rest-curved popper toy inside our and then dynamically simulate its snap-through "popping" jump at high-speed. During the quasi-static phase, we fix the inner ring of the popper (outer yellow points on the inset) and push the middle (inner yellow points on the inset) upward. We note that QB(PL) immediately collapses and unrolls the shell in the first quasi-static frame. After pushing inside out, we release the all prescribed boundary conditions, place the popper on a flat ground plane and step the dynamics. Here BAC correctly captures the resulting dynamics: the popper first snaps back and then bounces into the air from the resulting impact. In contrast, due to its initial collapse during the quasi-static phase, QB(PL) unsurprisingly fails to snap-through and instead remains in collapsed while bouncing upward.

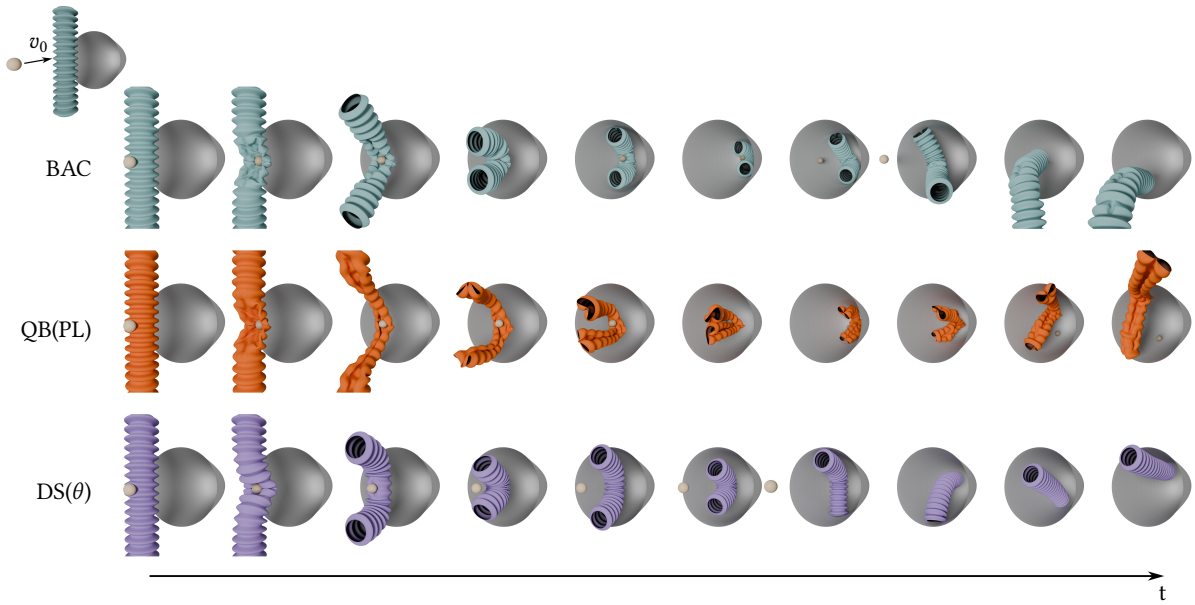


Fig. 25. **Corrugated Crash.** We simulate a small heavy rubber ball, launched with a high speed initial velocity of $30m/s$, crashing into a corrugated rubber hose backed by a conical "basket" obstacle. Because the rest configuration is not flat QB(PL) generates an initial transient elongation occurs at the beginning of the simulation to accommodate the geometry (see the supplemental video for details). Upon impact, both BAC and QB(PL) exhibit large localized deformations in the collision region, whereas DS(θ) produces smaller deformations due to its comparatively smoother and stiffer bending response. As the ball continues to crash into and push the rubber material, DS(θ) remains noticeably stiffer than the other models, resulting in little local deformation, faster recovery and a shorter travel distance. At the same time, QB(PL) does not, as expected, recover it rest shape, remaining in a globally buckled configuration with significant residual deformation.

Crush the DS(θ) model is distinctly different. It transitions from compression into a first symmetric bulging phase later than BAC and, post-buckle, its initial nucleation, rather than evenly distributed (as in both the Discrete-Director KL (DSO, DCS, BAC) and QB(PL) models) is biased longitudinally, with deeper and sharper cells towards the bottom of the can. As the loading continues, at midway through the process the buckling pattern jumps to create a skinny fold at the waist of the can that persists and grows into a sharp-necked bend in the final, crushed shape (See Figure 21). At the same time, unlike QB(PL), DS(θ) is able to able capture rest-curvature so that, in the Popper example, it is able capture both a reasonable loaded shape and a snap-through in the dynamic phase (see supplemental video).

Discrete-Director KL Models (DSO, DCS, BAC). Aligning with our convergence benchmark in Section 8, we find that DSO, DCS, and BAC generate quite close solutions (with very similar compute costs, see Section 10.4), for a range of cases. In the extreme, for the Can Crush we see that buckling solutions, for all three models, closely match across *all* quasistatic steps of the incremental loading process (see supplemental video). Similarly, in the high-speed Ball Crash (Figure 20), DSO, DCS, and BAC all generate similar bulk deformations with small variations on the indentation patterns generated during the compression and restoration (again see supplemental video). At the same time, we do observe differences across DSO, DCS, and BAC, with corresponding decreasing quality aligning with each model's behavior in our above Creasing Vee Benchmark study. These differences can lead to qualitatively distinct wrinkling behavior. For example, in the Cone Crush, DCS and DSO initially crumple into more localized and sharper folds near the apex, whereas BAC resists the sharpest bending in the starting crumpling of the tip, and so generates an initially more global and ordered buckling response (Figure 16). *When* there is sharp material folding and bending with respect to mesh resolution we observe artifacts in DCS and DSO. This is highlighted in the Pin Cloth Drop where folds of DCS and DSO exhibit sharp creasing artifacts, and correspondingly flattened folds in contrast to BAC, especially when folds of the cloth impact the ground (see Figures 19 and 15, and our supplemental video). We note as well that while these issues are clear in DSO (with similar behavior in QB(PL)) they are, as might be expected from our analysis in Section 9 above, significantly worse for DCS. Similarly, in the gentler contact of the Sphere Drape we see that while DCS still exhibits significant (albeit less severe) bending artifacts in tight folds, both DSO and BAC well-resolve wrinkles at this resolution.

10.3 Lateral Buckling Test [Romero et al. 2021]

We performed the lateral buckling experiment recently proposed by Romero et al. [2021] for validating thin-shell codes. We repeat the tests of DCS and DS(θ) from the original paper, and add our own results for the BAC, QB(PL), and QB(CR) models.

Methodology. We largely follow the "recipe" in Romero et al. [2021]'s supplemental materials. We use $\Omega = [0, w] \times [0, L]$ as

Example	Mesh Stat.			Sim. Para.		BAC			DSO			DCS			DS(θ)			QB(PL)		
	#V	#F	#E	δt	#TS	Time	#Iter	NNZ^b	Time	#Iter	NNZ^b	Time	#Iter	NNZ^b	Time	#Iter	NNZ^b	Time	#Iter	NNZ^b
Sphere Drape (Fig. 18)	80k	160k	241k	$2 \cdot 10^{-2}$	300	7h4m	9030	26.7M	6h55m	8859	26.7M	7h32m	8461	26.7M	2h32m	9725	9.4M	2h52m	8042	4.7M
Ball Crash (Fig. 20)	41k	82k	123k	$5 \cdot 10^{-4}$	1000	3h16m	6538	13.5M	3h3m	6685	13.5M	4h36m	9691	13.5M	30m21s	3897	4.8M	2h55m	6143	2.3M
Pinned Cloth Drop (Fig. 19)	31k	62k	93k	$2 \cdot 10^{-2}$	300	3h1m	9583	10.3M	3h37m	11077	10.3M	2h19m	6961	10.3M	58m	8175	3.6M	2h2m	12205	2.0M
Can Crush (Fig. 21)	87k	174k	262k	$4 \cdot 10^{-2}$	100	1h13m	1224	28.9M	1h39m	1660	28.9M	1h30m	1486	28.9M	34m	2025	10.2M	52m	2217	5.0M
Paper Cone Crush (Fig. 16)	81k	161k	241k	$4 \cdot 10^{-2}$	100	35m2s	739	26.5M	47m28s	1028	26.5M	46m40s	1019	26.5M	11m12s	828	9.4M	16m8s	939	4.6M
Twisted Cloth Cylinder (Fig. 23)	45k	89k	134k	$2 \cdot 10^{-2}$	300	1h48m	2572	14.7M	1h49m	2571	14.7M	1h47m	2571	14.7M	41m27s	2644	5.23M	45m6s	2307	2.5M
Twisted Cloth Sheet (Fig. 17)	52k	103k	156k	$2 \cdot 10^{-2}$	500	1h6m	1872	17.2M	56m50s	1541	17.2M	57m24s	1533	17.2M	-	-	-	-	-	-
Popper[1] (Fig. 24)	9.1k	18k	27k	$4 \cdot 10^{-2}$	36	1m23s	404	3.0M	1m27s	411	3.0M	1m25s	419	3.0M	35.8s	490	1.1M	14.0s	161	0.52M
Popper[2] (Fig. 24)	9.1k	18k	27k	$1 \cdot 10^{-4}$	464	12m14s	3097	3.0M	10m32s	2615	3.0M	11m44s	2990	3.0M	3m1s	1982	1.1M	2m33s	1725	0.52M
Corrugated Collision (Fig. 25)	53k	106k	160k	$5 \cdot 10^{-4}$	2000	16h29m	38408	17.5M	-	-	-	-	-	-	1h28m	11307	6.2M	3h51m	21223	3.0M

Table 7. **Timing statistics for the examples shown in the paper.** Columns 2–4 report the number of vertices, faces, and edges of each mesh. (Recall that DS(θ) and QB(PL) solely have midsurface DOF at mesh vertices, while BAC, DCS, and DSO additionally have a scalar director DOF for each edge.) Columns 5–6 list the time-step size (δt) and the total number of time steps (TS) used in each simulation. For each bending model, we report the total simulation time, the total number of Newton iterations accumulated *over all time steps*, and the number of nonzeros (NNZ) in the bending Hessian, which is the only Hessian contribution that differs across models. The marker "–" indicates that a simulation was not performed for the corresponding example and bending model. The superscripts in the Popper examples distinguish different phases: "1" denotes the quasi-static pushing phase, while "2" denotes the subsequent dynamic simulation phase. On average, BAC, DCS, and DSO exhibit similar runtime behavior, as they share the same bending Hessian structure. Compared to DS(θ) and QB(PL), these director-based discretizations have approximately three times as many nonzeros in the bending Hessian, resulting in an average slowdown of about 4× and 3× respectively. The QB(PL) model has the smallest number of nonzeros; although its discrete stencil (vertex one-ring neighborhoods) is larger than the edge-based stencil with four vertices, its Hessian is block-diagonal due to the absence of cross-coordinate coupling terms. However, improved sparsity alone does not guarantee faster runtimes than DS(θ), which we attribute to the stronger (but less accurate) bending induced by the discrete hinge formulation, typically producing fewer wrinkles and enabling easier and so lower iteration-count solver convergence.

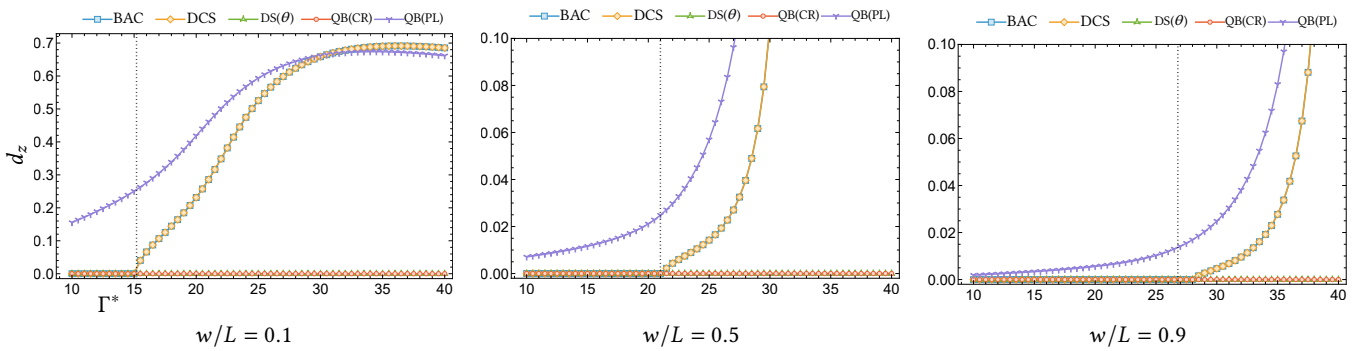


Fig. 26. **Lateral Buckling Validation.** Lateral displacement d_z as a function of normalized gravity Γ^*, for three different width-to-length ratios 0.1, 0.5, and 0.9. The black vertical line is the value of the critical Γ^* predicted by the master curve approximated by Romero et al. [2021]. The BAC and DCS models predict the onset of lateral buckling to good accuracy; QB(CR) and DS(θ) predict no buckling at all for $\Gamma^* \leq 40$ (consistent with similar behavior observed by Romero et al. [2021] for DS(θ)), whereas QB(PL) buckles for all applied loads.

the parameter domain and rest configuration of the shell midsurface, and apply gravity in the $+x$ direction. We mesh the shell using the same methodology as for Irregular test case in Section 8.2, using $n = \lfloor 100w/L \rfloor$ and $m = 100$ and target area $A = \frac{wL}{2nm}$, which corresponds to the Res++ mesh resolution recommended by Romero et al. [2021]. Our methodology differs in a few ways from that proposed in the original recipe:

- *clamping*: we could not find a detailed description of how the shell should be clamped in the numerical experiment. Clamping only the vertices at the mesh boundary $y = 0$ is insufficient, since doing so allows the shell to rigidly swivel about the x-axis at no energy cost, polluting the measure of lateral displacement. We clamp all vertices with $y < 3L/200$.

- *initialization*: instead of the tilting initialization procedure proposed by Romero et al. [2021], we instead apply a lateral displacement δz uniformly sampled in the range $[0, 10^{-4}]$ to the initial positions of all non-clamped vertices of the shell. This displacement was ample for breaking symmetry while being easier to implement.
- *termination*: Romero et al. [2021] do not prescribe a specific termination condition for the experiment. We terminate our static solver when the force residual magnitude decreases below 10^{-8} units, or when the Newton solver stalls with Newton decrement below 10^{-12}.

Results. We plot the maximum lateral displacement of the shell at termination as a function of scaled applied load Γ^* for three representative width-to-length ratios $w/L = 0.1, 0.5, 0.9$ in Figure 26. Raw data for other values of w/L and source code for reproducing the experiment are provided in the supplemental materials.

We compare the onset of buckling to the master curve,

$$\Gamma_c^* \approx 18.2(1-\nu)\sqrt{1+\nu} + 14.5\frac{w}{L}, \tag{61}$$

reported by Romero et al. [2021] (black line in the figures). When using the BAC model, buckling occurs at a critical load closely matching the master curve. (For $w/L = 0.9$, buckling occurs at slightly larger Γ^*; however, Equation (61) is a linear approximation of the true master curve, which from Figure 4 in Romero et al. [2021] curves slightly concave up. We believe this approximation error explains the slight discrepancy, especially since we see the same behavior for the DCS model, which agrees closely with the true master curve in Romero et al. [2021]'s own experiments. When using the QB(CR) and DS(θ) models the shell does not buckle, even at $\Gamma^* = 40$, which is consistent with similar behavior Romero et al. [2021] report for the DS(θ) model. Interestingly, QB(PL) shows the opposite behavior: the plate remains buckled even as load decreases to $\Gamma^* = 10$. We do not have good intuition explaining this behavior, but we note that the pre-buckled plate has significant membrane strain, violating the QB(PL) modeling assumption of isometric deformation.

10.4 Bending Model Performance

QB(PL) has a per-iterate cost numerical advantage over DS(θ) in terms of a fixed and sparser Hessian. However, we see that in timing, DS(θ) is generally faster (up to 6x) than QB(PL) – although not always. We see the primary source for these faster compute times reflected, see Table 7, in the larger iteration counts for QB(PL) timestep solves over DS(θ) for the same simulation examples. We hypothesize that these different iteration counts could be due to these two models simulating effectively different material behaviors (see analysis in the last section), and so very different configurations per timestep solve, with (as discussed above) QB(PL) generally resolving significantly more wrinkling details, per example, for the same material settings (see e.g., Popper and Pin Cloth Drop). This somewhat surprising outcome warrants further future investigation. We also observe that QB(PL) provides significant qualitative benefits over DS(θ) for flat-rest simulations along with its somewhat improved convergence behavior, with the possibility to potentially further speed QB(PL) solutions by taking advantage of its fixed bending Hessian, independent of changing membrane contributions.

In keeping with their identical sparsity, across DSO, DCS and BAC simulations we generally observe very close compute times, with differences largely explained by small fluctuations in iteration counts (to Newton-solver convergence). Mild exceptions are the Pin Drape where DCS model is 55% faster than DSO, and 30% faster than BAC; and the Ball Crash where DSO and BAC are respectively 50% and 40% faster than DCS. These differences again appear largely explained by iteration count differences with respect to different simulation outcomes, as highlighted in the Pin Drape example where both the DCS and DSO models give different results with sharp-creased artifacts in contrast to BAC (see our Section 10.2).

In comparison to QB(PL) we see that, BAC is up to 2x slower than QB(PL) in rest-flat configurations (where QB(PL) is clearly preferable over DS(θ)), while for rest-curved configurations where it's challenging to decide between QB(PL) and DS(θ), DS(θ) is up to 11.2x faster than BAC. Notably, the largest speedup occurs in the Corrugated Collision example, where DS(θ) generates significantly less wrinkling and local deformation (see, for example, the initial impact in the first two columns of Figure 25) than BAC and QB(PL). In this case, the BAC model requires approximately 3.4x more solver iterations overall. The other significant speedup (6.5x) is in the Ball Crash example where DS(θ) again generates very different simulation behavior (Figure 20), while QB(PL) simulates similar (see supplemental video) behavior to BAC, and requires significantly more iterations to do so, and so only obtains a 12% speedup over BAC.

11 Discussion

Our convergence analysis in Section 8 only considers purely isometric bending. Many shell behaviors in practice, including those we consider in Section 10, certainly involve some degree of coupling across stretching and bending. A careful study of shell model convergence in contexts where such coupling is dominant is clearly an important future direction.

At the same time, given its ubiquity in computer graphics, we have used the common P1 discretization of stretching energy for all of our evaluations in Section 10. Perhaps higher-order discretizations of the membrane model might offer better value. Similarly our constructions in Section 7 suggests a potential roadmap to combined higher-order discretization of both midsurface stretching and bending. While the efficiency tradeoffs of increased resolution vs. higher-order elements is not always clear, this likewise seems like a promising avenue to explore.

Our study focuses solely on the Koiter energy and Squared Curvature energy, both of which are typically derived under an isotropic material assumption. As covered above, Wen and Barbič [2023] propose a generalized Kirchhoff–Love formulation that supports a diversity of material models. Of note, these include orthotropic constitutive laws. At the same time, their framework remains grounded in the use of discrete second fundamental forms, and currently employs M-e. Av. for its discretization. Based on our analysis of M-e. Av., investigating potential convergence and robustness improvements by swapping in alternate discrete second fundamental forms, via e.g., BAC, seem both straightforward to try and likely to improve accuracy of the model.

Our work has focused mostly on model *verification*, with validation limited to the lateral buckling experiment of Romero et al. [2021] and to visual plausibility of our large-scale simulations in Section 10. Several papers in the mechanics and condensed matter literature report interesting experiments with buckling and crumpling of thin shells that could serve as valuable validation test cases, including spherical confinement of elastic plates [Hure et al. 2012] and indentation of a hemisphere by a falling plane [Vaziri 2009].

Finally, we note that there has been growing interest in computer graphics in shell models, including solid shells and Reissner-Mindlin models, not restricted to the Kirchhoff–Love assumption. Motivated by applications like deep drawing [Montes Maestre et al. 2024] that involve significant shear of the shell cross-section, it would be valuable to reconsider and study the same questions we address here for these important more general families of shells.

12 Recommendations

We close with our recommendations. When accurate and robust simulation is the goal, the Bending-Active Cosserat (BAC) model is currently the best option among the models we study here. BAC provides comparable convergence, and similar or better performance (both in Hessian NNZ count and total simulation timing) to the other Discrete Director KL models (the Discrete Shape Operator, Discrete Cosserat Shell, and Discrete Kirchhoff Triangle), and it does not suffer from their degenerate behavior in underresolved regions of high curvature.

When simulating rest-flat shells with efficiency in mind, the intrinsic Lagrange form of the Quadratic Bending model (QB(int)) (equivalently QB(PL) when the rest midsurface mesh is intrinsic Delaunay) is not a bad option, balancing accuracy with performance. In these cases, e.g., for cloth simulation, it is clearly preferable over the Discrete Shell (DS(θ)) model, but does not match the BAC model's accuracy. The fact that QB(int) is a simple quadratic function of the deformed midsurface vertex positions is also attractive in applications like machine learning where an easily-differentiable, qualitative "smoothness loss" is all that's needed.

Finally, when simulating rest-curved shapes in applications that do not require accuracy, consistency with real-world material parameters, nor mesh independence, DS(θ) remains a performant alternative. We do not recommend QB(int) for rest-curved shells as it does not generally capture (with some notable exceptions) qualitatively-reasonable behavior for such shells.

References

Apple Inc. 2023. Accelerate Framework. https://developer.apple.com/documentation/accelerate. Accessed: 2026-01.

Jean-Louis Batoz, Klaus-JÜRgen Bathe, and Lee-Wing Ho. 1980. A study of three-node triangular plate bending elements. *Internat. J. Numer. Methods Engrg.* 15, 12 (1980), 1771–1812. doi:10.1002/nme.1620151205

Jean-Louis Batoz and Mabrouk Ben Tahar. 1982. Evaluation of a new quadrilateral thin plate bending element. *Internat. J. Numer. Methods Engrg.* 18, 11 (1982), 1655–1677. doi:10.1002/nme.1620181106

Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A quadratic bending model for inextensible surfaces. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing* (Cagliari, Sardinia, Italy) *(SGP '06)*. Eurographics Association, Goslar, DEU, 227–230.

R. Bridson, S. Marino, and R. Fedkiw. 2005. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) *(SIGGRAPH '05)*. Association for Computing Machinery, New York, NY, USA, 3–es. doi:10.1145/1198555.1198573

Kevin Quentin Brown. 1979. *Geometric transforms for fast geometric algorithms.* Ph. D. Dissertation. USA. AAI8012772.

Hsiao-Yu Chen, Arnav Sastry, Wim M. van Rees, and Etienne Vouga. 2018. Physical simulation of environmentally induced thin shell deformation. *ACM Trans. Graph.* 37, 4, Article 146 (July 2018), 13 pages. doi:10.1145/3197517.3201395

Philippe G. Ciarlet. 2021. *Mathematical Elasticity: Three-Dimensional Elasticity.* Society for Industrial and Applied Mathematics, Philadelphia, PA. doi:10.1137/1.9781611976786

Fehmi Cirak, Michael Ortiz, and Peter Schröder. 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Internat. J. Numer. Methods Engrg.* 47, 12 (2000), 2039–2072.

David Clyde, Joseph Teran, and Rasmus Tamstorf. 2017. Modeling and data-driven parameter estimation for woven fabrics. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Los Angeles, California) *(SCA '17)*. Association for Computing Machinery, New York, NY, USA, Article 17, 11 pages. doi:10.1145/3099564.3099577

David Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted delaunay triangulations and normal cycle. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry* (San Diego, California, USA) *(SCG '03)*. Association for Computing Machinery, New York, NY, USA, 312–321. doi:10.1145/777792.777839

Xudong Feng, Wenchao Huang, Weiwei Xu, and Huamin Wang. 2022. Learning-Based Bending Stiffness Parameter Estimation by a Drape Tester. *ACM Trans. Graph. (SIGGRAPH Asia)* 41, 6, Article 221 (nov 2022), 16 pages.

Yohsuke Furuta, Nobuyuki Umetani, Jun Mitani, Takeo Igarashi, and Yukio Fukui. 2010. A Film Balloon Design System Integrated with Shell Element Simulation. In *Eurographics 2010 - Short Papers*, H. P. A. Lensch and S. Seipel (Eds.). The Eurographics Association. doi:10.2312/egsh.20101041

Yotam Gingold, Adrian Secord, Jefferson Y. Han, Eitan Grinspun, and Denis Zorin. 2004. A Discrete Model for Inelastic Deformation of Thin Shells. In *Poster at ACM/Eurographics Symposium on Computer Animation (SCA)*. Grenoble, France. https://cims.nyu.edu/gcl/papers/secord2004sds.pdf Poster.

Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. 2006. Computing discrete shape operators on general meshes. *Computer Graphics Forum* 25, 3 (2006), 547–556. doi:10.1111/j.1467-8659.2006.00974.x

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.

Elias Gueidon and Maurizio M. Chiaramonte. 2025. A Nonconforming Formulation of Cloth. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–11.

Jérémy Hure, Benoît Roman, and José Bico. 2012. Stamping and Wrinkling of Elastic Plates. *Phys. Rev. Lett.* 109 (Aug 2012), 054302. Issue 5. doi:10.1103/PhysRevLett.109.054302

Intel Corporation. 2023. Intel Threading Building Blocks. https://www.threadingbuildingblocks.org/. Accessed: 2026-01.

Warner Tjardus Koiter. 1966. On the nonlinear theory of thin elastic shells. *Proc. Koninkl. Ned. Akad. van Wetenschappen, Series B* 69 (1966), 1–54.

Petr Krysl and Jiun-Shyan Chen. 2023. Benchmarking computational shell models. *Archives of Computational Methods in Engineering* 30, 1 (2023), 301–315.

David R Lapp. 2008. Exploring 'extreme' physics with an inexpensive plastic toy popper. *Physics Education* 43, 5 (sep 2008), 492.

Qiqin Le, Yitong Deng, Jiamu Bu, Bo Zhu, and Tao Du. 2023. Second-Order Finite Elements for Deformable Surfaces. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 113, 10 pages. doi:10.1145/3610548.3618186

Phill-Seung Lee and Klaus-Jürgen Bathe. 2004. Development of MITC isotropic triangular shell finite elements. *Computers & Structures* 82, 11 (2004), 945–962. doi:10.1016/j.compstruc.2004.02.004

Youngyu Lee, Phill-Seung Lee, and Klaus-Jürgen Bathe. 2014. The MITC3+ shell element and its performance. *Computers & Structures* 138 (2014), 12–23. doi:10.1016/j.compstruc.2014.02.005

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 Trans. Graph. (SIGGRAPH)* 39, 4, Article 49 (2020).

Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. 2021. Codimensional Incremental Potential Contact. *ACM Trans. Graph. (SIGGRAPH)* 40, 4, Article 170 (2021).

Wei Li, Anzong Zheng, Lihua You, Xiaosong Yang, Jianjun Zhang, and Ligang Liu. 2017. Rib-reinforced Shell Structure. *Computer Graphics Forum* 36, 7 (2017), 15–27. doi:10.1111/cgf.13268

Qixin Liang. 2025. Corotational Hinge-based Thin Plates/Shells. *Computer Graphics Forum* 44, 2 (2025).

F. Löschner, J. A. Fernández-Fernández, S. R. Jeske, and J. Bender. 2024. Curved Three-Director Cosserat Shells with Strong Coupling. *Computer Graphics Forum* 43, 8 (2024), e15183. doi:10.1111/cgf.15183

Richard H Macneal and Robert L Harder. 1985. A proposed standard set of problems to test finite element accuracy. *Finite elements in analysis and design* 1, 1 (1985), 3–20.

Juan Sebastian Montes Maestre, Stelian Coros, and Bernhard Thomaszewski. 2024. Q3T Prisms: A Linear-Quadratic Solid Shell Element for Elastoplastic Surfaces. In *SIGGRAPH Asia 2024 Conference Papers* (Tokyo, Japan) *(SA '24)*. Association for Computing Machinery, New York, NY, USA, Article 63, 9 pages. doi:10.1145/3680528.3687697

Victor Romero, Mickaël Ly, Abdullah Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes. 2021. Physical validation of simulators in computer graphics: a new framework dedicated to slender elastic structures and frictional contact. *ACM Trans. Graph.* 40, 4, Article 66 (July 2021), 19 pages. doi:10.1145/3450626.3459931

Matthew W. Scroggs, Pablo D. Brubeck, Joseph P. Dean, Jørgen S. Dokken, India Marsden, Nuno Nobre, et al. 2020–2026. DefElement: an encyclopedia of finite element definitions. https://defelement.org. [Online; accessed 20-January-2026].

Nicholas Sharp and Keenan Crane. 2020. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum* (2020). doi:10.1111/cgf.14069

Jonathan Richard Shewchuk. 1996. *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. Springer-Verlag, 203–222.

Justin Solomon. 2025. Private communication.

Oded Stein, Eitan Grinspun, Max Wardetzky, and Alec Jacobson. 2018. Natural Boundary Conditions for Smoothing in Geometry Processing. *ACM Trans. Graph.* 37, 2, Article 23 (May 2018), 13 pages. doi:10.1145/3186564

KY Sze, XH Liu, and SH Lo. 2004. Popular benchmark problems for geometric nonlinear analysis of shells. *Finite elements in analysis and design* 40, 11 (2004), 1551–1569.

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

Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 205–214.

Nobuyuki Umetani, Danny M. Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. In *ACM SIGGRAPH 2011 Papers* (Vancouver, British Columbia, Canada) *(SIGGRAPH '11)*. Association for Computing Machinery, New York, NY, USA, Article 90, 12 pages. doi:10.1145/1964921.1964985

Ashkan Vaziri. 2009. Mechanics of highly deformed elastic shells. *Thin-Walled Structures* 47, 6 (2009), 692–700. doi:10.1016/j.tws.2008.11.009

Yu Wang, Alec Jacobson, Jernej Barbič, and Ladislav Kavan. 2015. Linear subspace design for real-time shape deformation. *ACM Trans. Graph.* 34, 4, Article 57 (July 2015), 11 pages. doi:10.1145/2766952

Max Wardetzky, Miklós Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. 2007. Discrete quadratic curvature energies. *Computer Aided Geometric Design* 24, 8 (2007), 499–518. doi:10.1016/j.cagd.2007.07.006 Discrete Differential Geometry.

Jerry Weil. 1986. The synthesis of cloth objects. *ACM Siggraph Computer Graphics* 20, 4 (1986), 49–54.

Clarisse Ablavi Weischedel. 2012. *A discrete geometric view on shear-deformable shell models*. Ph.D. Dissertation. Georg-August-Universität Göttingen. doi:10.53846/goediss-2453 Advisor: Prof. Dr. Max Wardetzky; Referees: Prof. Dr. Max Wardetzky and Prof. Dr. Bernd Simeon.

Jiahao Wen and Jernej Barbič. 2023. Kirchhoff-Love Shells with Arbitrary Hyperelastic Materials. *ACM Trans. Graph.* 42, 6, Article 174 (Dec. 2023), 15 pages. doi:10.1145/3618405

Anzong Zheng, Zaiping Zhu, Shaojun Bian, Jian Chang, Habibollah Haron, Andres Iglesias, Lihua You, and Jianjun Zhang. 2021. Optimized Unidirectional and Bidirectional Stiffened Objects for Minimum Material Consumption of 3D Printing. *Mathematics* 9, 21 (2021). doi:10.3390/math9212835

A BAC Model Kinematics

For each triangle $ijk = \{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$ we equip midedges ij with a free director $\mathbf{d}_{ij;k} \in \mathbb{R}^3$ (see Table 1 for notation). We apply P1 elements for vertex positions and CRH1 elements for for BAC edge directors (CR1 for DCS).

Parameterizing with barycentric coordinates (s, t) the interpolated director field within this triangle is then

$$\mathbf{d}(s,t) = (2s + 2t - 1)\,\mathbf{d}_{jk;i} + (1 - 2t)\,\mathbf{d}_{ik;j} + (1 - 2s)\,\mathbf{d}_{ij;k}. \tag{62}$$

Applying the BAC/DCS edge orthogonality constraint we have $\mathbf{d}_{jk;i} \cdot (\mathbf{x}_j - \mathbf{x}_k) = 0$, so that the director $\mathbf{d}_{jk;i}$ lies in the plane spanned by $\{\hat{\mathbf{n}}_{ijk}, \hat{\mathbf{t}}_{i;jk}\}$. Let ζ_{jk} give the rotation angle from the face normal $\hat{\mathbf{n}}_{ijk}$ to $\mathbf{d}_{jk;i}$, so that

$$\zeta_{jk} = \frac{\theta_{jk}}{2} + \epsilon_{jk}\alpha_{jk}. \tag{63}$$

Then $\mathbf{d}_{jk;i}$ can be expressed as

$$\mathbf{d}_{jk;i} = \eta_{jk;i}\big(\cos\zeta_{jk}\,\hat{\mathbf{n}}_{ijk} + \sin\zeta_{jk}\,\hat{\mathbf{t}}_{i;jk}\big), \tag{64}$$

where $\eta_{jk;i}$ gives director $\mathbf{d}_{jk;i}$'s length.

DCS applies the pointwise KL constraint $\|\mathbf{d}_{jk;i}\| = 1$ which requires $\eta_{jk;i} = 1$, giving us

$$\mathbf{d}^{\mathrm{DCS}}_{jk;i} = \cos\zeta_{jk}\,\hat{\mathbf{n}}_{ijk} + \sin\zeta_{jk}\,\hat{\mathbf{t}}_{i;jk}. \tag{65}$$

BAC instead applies the per-triangle KL constraint $\mathbf{d}_{jk;i} \cdot \hat{\mathbf{n}}_{ijk} = 1$, which yields $\eta_{jk;i} = 1/\cos\zeta_{jk}$, so that

$$\mathbf{d}^{\mathrm{BAC}}_{jk;i} = \hat{\mathbf{n}}_{ijk} + \tan\zeta_{jk}\,\hat{\mathbf{t}}_{i;jk}. \tag{66}$$

B Kinetic Energy Derivation

Here we provide derivation of the kinetic energy expressions in Equation (52), as referenced in Section 7. Following Appendix A above, we consider the same triangle stencil $ijk = \{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$.

The interpolated director field time derivative for BAC and DCS is then

$$\dot{\mathbf{d}} = (2s + 2t - 1)\,\dot{\mathbf{d}}_{jk;i} + (1 - 2t)\,\dot{\mathbf{d}}_{ik;j} + (1 - 2s)\,\dot{\mathbf{d}}_{ij;k}. \tag{67}$$

We then have

$$\int_\Omega \|\dot{\mathbf{d}}\|^2 \mathrm{d}x\mathrm{d}y = \sum_{ijk} \frac{A_{ijk}}{6} \sum_i \|\dot{\mathbf{d}}_{i;jk}\|^2, \tag{68}$$

where cross terms vanish in integration.

Taking time derivatives of Equations (65) and (66) above then respectively yield

$$\begin{aligned} \dot{\mathbf{d}}^{\mathrm{DCS}}_{jk;i} &= \dot{\zeta}_{jk}\big(-\sin\zeta_{jk}\,\hat{\mathbf{n}}_{ijk} + \cos\zeta_{jk}\,\hat{\mathbf{t}}_{i;jk}\big) + \sum_i O(\|\dot{\mathbf{x}}_i\|), \\ \dot{\mathbf{d}}^{\mathrm{BAC}}_{jk;i} &= \frac{\dot{\zeta}_{jk}}{\cos^2\zeta_{jk}}\,\hat{\mathbf{t}}_{i;jk} + \sum_i O(\|\dot{\mathbf{x}}_i\|). \end{aligned} \tag{69}$$

We assume that vertex velocity contributions $\dot{\mathbf{x}}_i$ are small relative to the remaining terms, as they are already penalized by the first kinetic energy term in Equation (50), which scales linearly with the shell thickness. Neglecting these higher-order contributions, we obtain the approximations

$$\|\dot{\mathbf{d}}^{\mathrm{DCS}}_{jk;i}\|^2 \approx \|\dot{\zeta}_{jk}\|^2, \qquad \|\dot{\mathbf{d}}^{\mathrm{BAC}}_{jk;i}\|^2 \approx \frac{1}{\cos^4\zeta_{jk}}\,\|\dot{\zeta}_{jk}\|^2. \tag{70}$$

C Geometric Interpretation of the Termination Condition

In Section 7.3, we introduced a termination condition specifically designed for the discrete director Cosserat Kirchhoff–Love shell formulation. In this section, we provide geometric and mathematical intuition for the particular offset construction underlying this criterion.

Recall that the offset vertex position is defined as

$$\begin{aligned} \mathbf{x}^{\mathrm{off}}_i &= \mathbf{x}^{\mathrm{off}}_{ik} + \mathbf{x}^{\mathrm{off}}_{ij} - \mathbf{x}^{\mathrm{off}}_{jk} \\ &= \mathbf{x}_{ik} + \mathbf{x}_{ij} - \mathbf{x}_{jk} + \frac{h}{2}\big(\mathbf{d}_{ik} + \mathbf{d}_{ij} - \mathbf{d}_{jk}\big) \\ &= \mathbf{x}_i + \frac{h}{2}\big(\mathbf{d}_{ik} + \mathbf{d}_{ij} - \mathbf{d}_{jk}\big). \end{aligned} \tag{71}$$

Notice that the offset faces $(\mathbf{x}^{\mathrm{off}}_i, \mathbf{x}^{\mathrm{off}}_j, \mathbf{x}^{\mathrm{off}}_k)$ form a discontinous triangle soup: neighboring triangles do not agree on offset positions of shared vertices or edges. Nevertheless, locally, each triangle is a meaningful approximation of the upper envelope $\mathbf{r}^{\mathrm{off}}(x, y) = \phi(x, y, h/2)$ of the shell: using the offset vertex positions, we define the discrete first fundamental form of the envelope as

$$\mathbf{I}^{\mathrm{off}} = \begin{bmatrix} (\mathbf{x}^{\mathrm{off}}_j - \mathbf{x}^{\mathrm{off}}_i)\cdot(\mathbf{x}^{\mathrm{off}}_j - \mathbf{x}^{\mathrm{off}}_i) & (\mathbf{x}^{\mathrm{off}}_j - \mathbf{x}^{\mathrm{off}}_i)\cdot(\mathbf{x}^{\mathrm{off}}_k - \mathbf{x}^{\mathrm{off}}_i) \\ (\mathbf{x}^{\mathrm{off}}_j - \mathbf{x}^{\mathrm{off}}_i)\cdot(\mathbf{x}^{\mathrm{off}}_k - \mathbf{x}^{\mathrm{off}}_i) & (\mathbf{x}^{\mathrm{off}}_k - \mathbf{x}^{\mathrm{off}}_i)\cdot(\mathbf{x}^{\mathrm{off}}_k - \mathbf{x}^{\mathrm{off}}_i) \end{bmatrix},$$

where

$$\mathbf{I}^{\mathrm{off}} = \mathbf{I} + h\,\mathrm{I\!I} + O(h^2),$$

with $\mathbf{I}$ the discrete first fundamental form of the midsurface, and $\mathrm{I\!I}$ is defined in Equation (47).

This expansion demonstrates that, up to second order in the shell thickness h, the metric induced by the discrete offset configuration agrees with the first fundamental form of $\mathbf{r}^{\mathrm{off}}$. Consequently, we can measure convergence in terms of the displacement of the offset vertices as a physically and geometrically consistent coupled measure of combined director and midsurface position DOFs for our termination criterion.

D Additional Results for DS(tan)

Here we revisit the Ball Crash and Can Crush simulation tests solved with the DS(tan) bending model. With a comparable discretization to DS(θ), we see in Figures 27 and 28 that DS(tan) produces closely tracking results.

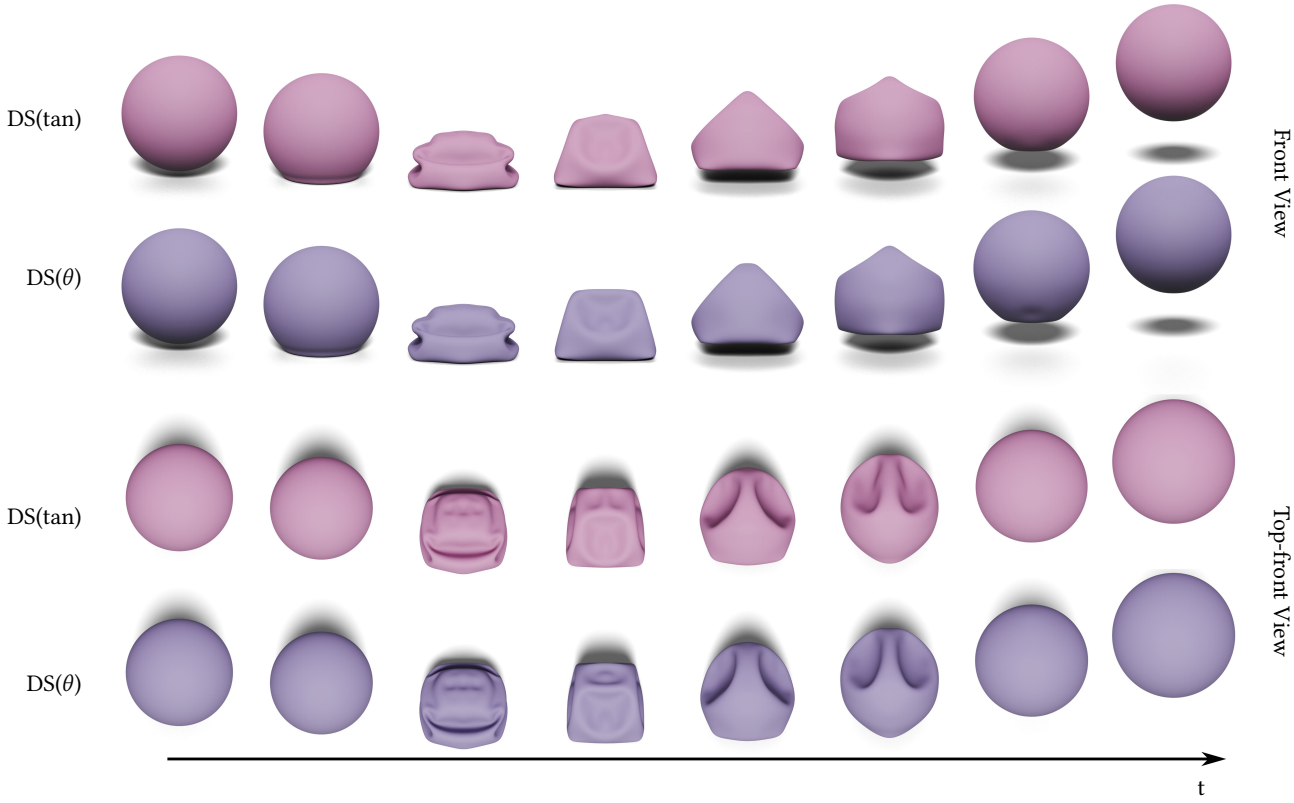


Fig. 27. **Ball Crash.** We compare Ball Crash experiment results (see Figure 20) for the DS(θ) and DS(tan) hinge models. Simulations exhibit same global behavior across trajectories with small, fine-scale localized differences.

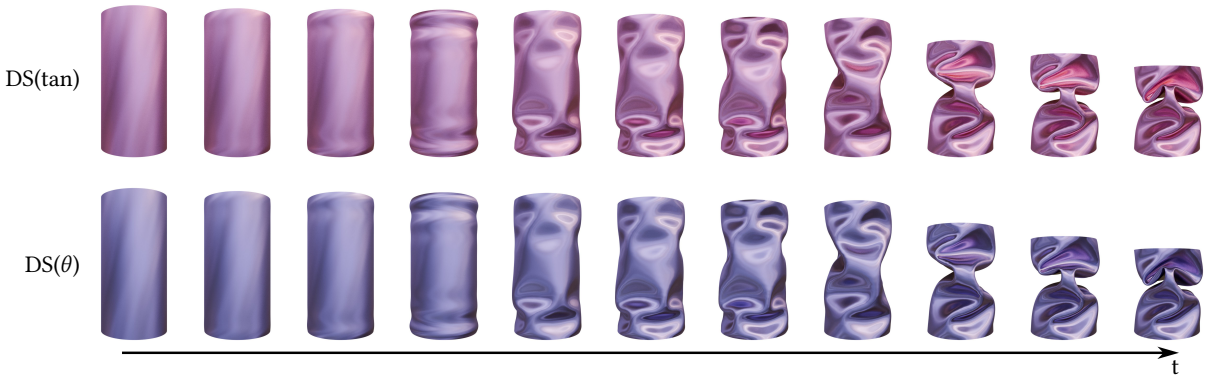


Fig. 28. **Can Crush.** We revisit the Can Crush experiment from Figure 21, comparing results obtained with the DS(θ) and DS(tan) discrete hinge models. The two models simulate closely matching behavior over the entire compression sequence.