

Second-Order Finite Elements for Deformable Surfaces

Qiqin Le
Shanghai Qi Zhi Institute
Shanghai, China
leqq@sqz.ac.cn

Yitong Deng
Dartmouth College
Hanover, United States of America
yitong.deng@dartmouth.edu

Jiamu Bu
Tsinghua University
Beijing, China
bjm21@mails.tsinghua.edu.cn

Bo Zhu
Georgia Institute of Technology
Atlanta, United States of America
Dartmouth College
Hanover, United States of America
bo.zhu@gatech.edu

Tao Du
Tsinghua University
Beijing, China
Shanghai Qi Zhi Institute
Shanghai, China
taodu@tsinghua.edu.cn



Figure 1: Lantern: Two lanterns (left and right) simulated using our second-order finite element methods on deformable surfaces coupled with elastic rods. The top row shows the intermediate frames of the lantern's dynamic motion when suppressed from above, and the bottom row shows the final frames. Copies of lanterns with white diffusive material on the right provide clearer demonstration of their geometric structures during deformation.

ABSTRACT

We present a computational framework for simulating deformable surfaces from planar rest shape with second-order triangular finite elements. Our method develops numerical schemes for discretizing stretching, shearing, and bending energies of deformable surfaces in a second-order finite-element setting. In particular, we introduce a novel discretization scheme for approximating mean curvatures on a curved triangle mesh. Our framework also integrates a virtual-node finite-element scheme that supports two-way

coupling between cut-cell rods without expensive remeshing. We compare our approach with traditional simulation methods using linear and higher-order finite elements and demonstrate its advantages in several challenging settings, such as low-resolution meshes, anisotropic triangulation, and stiff materials. Finally, we showcase several applications of our framework, including cloth simulation, mixed Origami and Kirigami, and biologically-inspired soft wing simulation.



This work is licensed under a Creative Commons Attribution International 4.0 License.

SA Conference Papers '23, December 12–15, 2023, Sydney, NSW, Australia
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0315-7/23/12.
<https://doi.org/10.1145/3610548.3618186>

CCS CONCEPTS

• **Computing methodologies** → **Physical simulation**; *Mesh geometry models*.

KEYWORDS

Higher-order finite elements, thin-shell modeling and simulation.

ACM Reference Format:

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 (SA Conference Papers '23)*, December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3610548.3618186>

1 INTRODUCTION

Linear triangles have served as a fundamental computing primitive, accommodating various cloth and shell simulations, ever since its first success in simulating large-timestep cloth a quarter of a century ago [Baraff and Witkin 1998]. By discretizing a deformed surface with planar triangles that share edges and vertices and solving elasticity using the finite element method (FEM), these triangle meshes can produce a broad range of visually appealing surface phenomena, such as cloth wrinkles and paper creases. However, linear triangles contain several inherent weaknesses. *First*, in terms of its geometric expressiveness, the planar assumption of each triangle requires a large number of elements to approximate a complex, curved surface. Bending deformation can only be expressed with the dihedral angle between triangles that share an edge. Remeshing operations, sometimes requiring highly anisotropic elements, must be performed frequently to characterize bending behaviors that are not aligned with mesh edges (e.g., see Narain et al. [2012]). *Second*, in terms of dynamics, linear triangles assume a piecewise constant deformation gradient within each element, resulting in a demanding prerequisite on the mesh quality (e.g., see Shewchuk [2002]). The simulation accuracy will be negatively affected if a mesh contains low-quality triangles. *Third*, the linear geometry and constant deformation gradient further cause kinematic locking artifacts on a triangle mesh, especially when its resolution is low, which produces unnatural bending and overly damped cloth dynamics. In short, simulations on linear triangles are highly *mesh-dependent*. High-quality meshes (in terms of both mesh resolution and element aspect ratio) are essential for accurate simulations.

In recent years, *high-order* triangles have sparked interest in computer graphics research. These methods stem from the high-order finite element analysis in mechanical engineering [Johnson 2012], by placing multiple degrees of freedom on edges or interiors of finite elements. In this way, a triangle is no longer planar but rather has its shape described by a high-order polynomial and approximated with high-order basis functions. As pointed out by Schneider et al. [2018], the simulation accuracy can be decoupled from the mesh quality when employing high-order FEM schemes. Following this thread, a natural next step is to explore the efficacy of high-order methods in simulating deformable surfaces, with a particular focus on addressing the various mesh-dependent bending, locking, and coupling artifacts in cloth or thin-shell simulation. Nonetheless, devising a high-order cloth and shell simulator to tackle these highly dynamic and nonlinear physical problems is difficult. One of the main challenges is to appropriately define the stretching, shearing, and bending energy and their discretization that fits a high-order discretization. To begin with, a curved triangle can bend due to its intrinsic non-planar shape. In addition, two curved triangles connect on a curved edge, which voids the possibility of characterizing bending with dihedral angles. To the best of our knowledge, there exists no practical algorithm that can

support cloth and shell simulations with high-order finite elements for graphics applications.

Filling such a gap, our paper proposes a second-order deformable surface simulation algorithm on curved triangle elements. At the heart of our approach is a novel elastic energy definition in conjunction with its discretion on non-planar triangles to accommodate the second-order simulation of elastic surfaces. Furthermore, we augment our simulator with a novel scheme to support second-order cut-cell finite elements, enabling native two-way coupling with elastic rods in our simulation. We evaluate our simulation on different test scenarios to demonstrate its merits in producing dynamic, smooth, and deformable surface simulations.

Our main contributions include the following:

- We present a discretization scheme for mean curvature and its induced bending energy on curved triangulation of smooth surfaces;
- We present a novel virtual-node finite element scheme to couple deformable surfaces with cut-cell curves, enabling creases and coupling with elastic rods at flexible locations;
- We validate the efficacy of the proposed framework on low-resolution, low-quality triangulation and compare it with an existing open-source finite-element implementations,
- We demonstrate the application of our method in cloth simulation, mixed Origami and Kirigami, and biologically-inspired soft-wing simulation.

2 RELATED WORK

Deformable-surface simulation. Deformable-surface simulation in computer graphics originated from the early endeavors in cloth animation [Baraff and Witkin 1998; Breen et al. 1994; Eberhardt et al. 1996; Terzopoulos et al. 1987], which model cloth as an elastic body. To counteract the numerical instability evoked by the high stiffness necessary for stretch resistance, Baraff and Witkin [1998] introduced an implicit temporal-integration scheme to enable large time steps with high stability, at the cost of significant numerical damping. To reduce such artifacts, several implicit-explicit solving schemes [Boxerman 2003; Eberhardt et al. 2000; Stern and Grinspun 2009] are developed by treating stiff and non-stiff forces differently. Recently, a line of projective dynamics methods [Bouaziz et al. 2014; Ly et al. 2020] has also been devised to solve the implicit integration problem from the variational perspective efficiently. To discretely model the deformation and bending mechanics of thin shells, a number of convenient hinge-based models have been developed [Bridson et al. 2003; Gingold et al. 2004; Grinspun et al. 2003]; we adopt the paradigm of Grinspun et al. [2003] that computes the bending energy by comparing squared mean curvatures. To date, a large gamut of thin shell phenomena have been successfully captured, including plastic folding and crumpling [Narain et al. 2013], large-scale tearing and fracture [Busaryev et al. 2013; Pfaff et al. 2014], interaction with ambient fluid [Chen et al. 2013; Sifakis et al. 2008], reaction to environmental stimuli [Chen et al. 2018], etc. In addition, the accurate and affordable simulation of thin shells has facilitated a range of inverse design tasks, such as the natural shape of shells [Ly et al. 2018], FlexMaps [Malomo et al. 2018], deployable X-shells [Panetta et al. 2019], and inflatable surfaces [Panetta et al. 2021].

Higher-order methods. High-order discretization schemes and algorithms offer a competitive alternative to their low-order counterparts in resolving physical problems at higher fidelity at a comparable computational cost [Schneider et al. 2022; Vincent and Jameson 2011]. For example, higher-order continuous Galerkin methods [Vos et al. 2010; Zhou and Lu 2005] and discontinuous Galerkin methods [Huynh et al. 2014; Jameson 2011; Wang et al. 2013] are developed to improve accuracy and efficiency in computational structural or fluid mechanics. Interested readers can refer to Johnson [2012] for a comprehensive discussion. Similarly, higher-order schemes have been bridged with the finite volume [Barth and Frederickson 1990; Harten et al. 1997] and finite difference [Shu 2003; Vasilyev 2000] paradigms. As identified by Vincent and Jameson [2011], a hindering challenge for adopting these high-order schemes is the generation of unstructured, high-order (curved) meshes. A number of works take on this challenge by directly generating meshes from CAD models [Dey et al. 1999] or by post-processing linear meshes [Ims et al. 2015]. In hybrid Lagrangian-Eulerian simulation, methods that transfer higher-order velocity modes have been shown to provide better energy and momentum conservation for both fluid [Jiang et al. 2015] and solid materials [Hu et al. 2018]. Higher-order level-set methods have also been developed to enhance mass conservation in multi-phase simulation [Moghadam et al. 2016], promote smoothness in surface construction [Bajaj et al. 2007], and preserve sub-grid structures [Nave et al. 2010].

Higher-order shells. Exploration of higher-order methods for shell simulation keeps being an important and major topic especially for structural analysis in mechanical and civil engineering. Previous works primarily target at solving (quasi-)static problems with small deformation. Many previous works [Batoz et al. 1980; Belytschko et al. 1984; Macneal 1982; Paal and Calladine 1992; Sze and Zhu 1999; Tessler and Hughes 1985; Zhongnian 1992] proposed shell elements to achieve an accurate and consistent solution with the assumption of geometric linearity. To accurately present the curved mid-surface of a shell, lots of conventional shell elements introduced rotational degrees of freedom [Bathe and Ho 1981; Batoz et al. 1980; Belytschko et al. 1984; Campello et al. 2003; Macneal 1982; Tessler and Hughes 1985; Zhongnian 1992]. This category also includes lots of existing shell algorithms in commercial software, e.g., S4 and S4R in ABAQUS [Abaqus 2011]. In computer graphics, subdivision-surface shells are of particular interest to researchers [Cirak et al. 2000]. A series of works have been performed to solve static and dynamic problems using different subdivision strategies, e.g. Catmull–Clark subdivision [Wawrzinek et al. 2011].

Previous higher-order shell elements have explored widely the methods of improving accuracy and consistency for simulating loads and displacements for (quasi-)static shells with nonzero thickness. However, scenes in graphic application usually require methods supporting stable simulation for dynamic large-scale deformation and convenient coupling with other structures, which we believe cannot be easily realized with the existing approaches.

The assumption of material or geometric linearity often leads to a visually implausible shape when simulating large deformations. Also, rotational DoFs suffer from potentially slow convergence and make crease handling considerably more expensive [Hu et al. 2021]. Finally, handling flexible coupling with creases or rods in some

of these higher-order FEMs is typically challenging. For example, Wawrzinek et al. [2011] and Cirak et al. [2000] assumed C1 continuity on thin shells, so handling creases is not straightforward. Laurent and Rio [2001] used only 9 DoFs to model a curved triangle, which would lead to incompatible boundary conditions if coupled with cut-cell creases or rods.

3 CONTINUOUS MODEL

Kinematics. We consider a deformable surface in \mathbb{R}^3 with zero thickness and a planar rest shape $\Omega \subset \mathbb{R}^2$. We use $\mathbf{X} = (u, v) \in \Omega$ to denote a material point in the rest shape. A spatial-temporal function $\mathbf{x} : \Omega \times \mathbb{R}_+ \rightarrow \mathbb{R}^3$ defines the motion of each material point $\mathbf{X}(u, v)$ at time t as $\mathbf{x}(\mathbf{X}(u, v), t)$ or $\mathbf{x}(\mathbf{X}, t)$ for brevity. We use $\mathbf{x}(t)$ to denote the surface at time t , i.e., $\mathbf{x}(t) = \mathbf{x}(\Omega, t)$. The velocity and acceleration of each material point are denoted as $\mathbf{v} = \dot{\mathbf{x}}$ and $\mathbf{a} = \ddot{\mathbf{x}}$. The Jacobian of a surface point w.r.t. its (u, v) coordinates is denoted as \mathbf{F} with its subscript indicating the surface, e.g., $\mathbf{F}_{\mathbf{x}(t)} : \Omega \rightarrow \mathbb{R}^{3 \times 2}$ returns the Jacobian of the material point $\mathbf{X}(u, v)$ at time t .

Dynamics. Our dynamics model is

$$\rho \mathbf{a}(\mathbf{X}, t) = \mathbf{f}_{\text{int}}(\mathbf{X}, t) + \mathbf{f}_{\text{ext}}(\mathbf{X}, t), \quad \forall (u, v) \in \Omega, t \geq 0, \quad (1)$$

where ρ is the material density. We use \mathbf{f}_{int} and \mathbf{f}_{ext} to denote the internal and external force densities exerted on the rest shape. We assume that \mathbf{f}_{int} consists of the stretching and shearing force \mathbf{f}_{ss} and the bending force \mathbf{f}_{bd} , both of which are conservative forces derived from the corresponding potential energy E_{ss} and E_{bd} . We complement Eqn. (1) with time-varying Dirichlet boundary conditions $\mathbf{x}(\mathbf{X}, t) = \mathbf{x}_D(\mathbf{X}, t)$ on $\Gamma_D \subseteq \partial\Omega$ where \mathbf{x}_D prescribes the kinematic motion on $\mathbf{X}(\Gamma_D)$.

Stretching and shearing energy. We follow previous works [Kim and Eberle 2020] to assume a hyperelastic material model. Therefore, the strain energy for stretching and shearing can written as $E_{\text{ss}} = \int_{\Omega} \Psi_{\text{ss}}(u, v) dudv$, where the energy density Ψ_{ss} is defined as a function of the Green strain tensor $E = 0.5(\mathbf{F}_{\mathbf{x}}^T \mathbf{F}_{\mathbf{x}} - \mathbf{I})$. In our implementation, we applied St. Venant-Kirchhoff model [Sifakis and Barbic 2012].

Bending energy. We consider the bending energy as an integral $E_{\text{bd}} = \int_{\Omega} \Psi_{\text{bd}}(u, v) dudv$ of the squared mean curvature [Grinspun et al. 2003]:

$$\Psi_{\text{bd}}(\mathbf{X}(u, v)) = k H_{\mathbf{x}(t)}^2(u, v), \quad (2)$$

where $H_{\mathbf{x}(t)} : \Omega \rightarrow \mathbb{R}$ stands for the mean curvature and k the bending stiffness.

4 NUMERICAL DISCRETIZATION

4.1 Variational Form

We discretize Eqn. (1) with the backward Euler time integration, leading to the following energy minimization problem [Kane et al. 2000]:

$$\min_{\mathbf{x}^{n+1}} \int_{\Omega} \frac{\rho}{2h^2} \|\mathbf{x}^{n+1} - \mathbf{y}^n\|^2 dudv + E_{\text{ss}}[\mathbf{x}^{n+1}] + E_{\text{bd}}[\mathbf{x}^{n+1}] \quad (3)$$

subject to the Dirichlet boundary conditions. Here, the superscript n indexes the time step with h its step size. $\mathbf{y}^n = \mathbf{x}^n + h\mathbf{v}^n + h^2\rho^{-1}\mathbf{f}_{\text{ext}}^n$ is

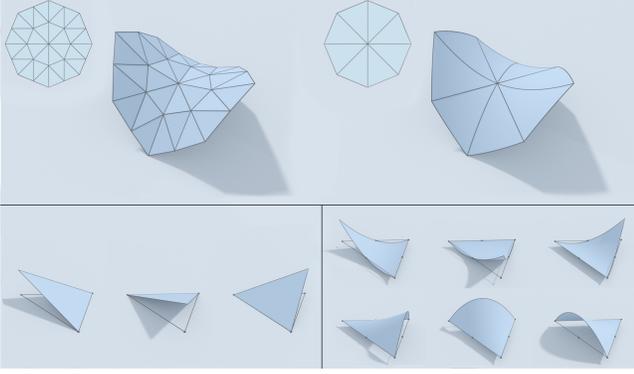


Figure 2: Top: Approximating a saddle surface with first-order (left) and second-order (right) finite elements whose triangulation is shown in the insets. Bottom: The basis function landscape (blue surfaces) and its location (black dots) of a single first-order or second-order element.

a constant during optimization. Solving \mathbf{x}^{n+1} defines the surface at the end of the n -th time step with its velocity $\mathbf{v}^{n+1} = h^{-1}(\mathbf{x}^{n+1} - \mathbf{x}^n)$.

4.2 Higher-Order Finite Elements

We use the finite element method to discretize Eqn. (3). We assume Ω is a polygonal domain and partition it into triangles $\{T_j\}$. These triangles together contain N basis functions $\phi_i : \Omega \rightarrow [0, 1]$ centered at nodes $(u_1, v_1), (u_2, v_2), \dots, (u_N, v_N) \in \Omega$. Depending on the order of finite elements, these nodes may include triangle vertices, edge midpoints, or triangle centers. The basis functions can be linear, quadratic, or higher-order polynomials on each triangle (Fig. 2) and can be precomputed. For any function f on Ω , we define \hat{f} , an approximation to f , as the sum of all basis functions ϕ_i weighted by predefined nodal values f_i , i.e.,

$$f(u, v) \approx \hat{f}(u, v) = \sum_i f_i \phi_i(u, v), \quad \forall (u, v) \in \Omega, \quad (4)$$

and it is common to use $f_i = f(u_i, v_i)$ whenever possible. \hat{f} is a continuous function on Ω , allowing us to approximate f at any $(u, v) \in \Omega$ with finite degrees of freedom f_1, f_2, \dots, f_N (Fig. 2). Note that a triangle becomes curved in a second-order finite element. The integral of a function f can be computed by quadrature rules:

$$I_\Omega[f] = \int_\Omega f dudv \approx \sum_{(\mathbf{q}_k, w_k)} w_k \hat{f}(\mathbf{q}_k), \quad (5)$$

where $\{\mathbf{q}_k \in \Omega\}$ are the quadrature points and w_k their weights. We discretize \mathbf{X} and \mathbf{x} at triangle nodes to obtain $\mathbf{X}_i, \mathbf{x}_i \in \mathbb{R}^{3 \times N}$, on which we use the second-order finite element scheme to discretize the energies in Eqn. (3). The main challenge is to discretize the bending energy due to curved triangles, which we detail below.

4.3 Energy Discretization

Previous works [Baraff and Witkin 1998; Gingold et al. 2004; Grinspun et al. 2003] have studied discrete bending energy on triangular surfaces extensively and formulated it as a function of dihedral angles. Generalizing this idea to second-order elements is non-trivial

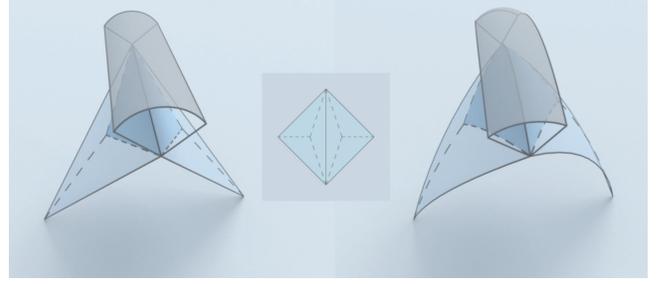


Figure 3: A visual explanation of the discrete mean curvature measure (Def. 4.1) on a surface discretized with two planar (left) and curved (right) triangles. The triangulation in the (u, v) parameter space is shown in the middle inset, and the dashed lines further subdivide each triangle to distribute average edge curvature measure. A Borel set (not shown) in \mathbb{R}^3 intersects the surface onto two of such subdivided triangles (darker blue) and fully encloses the edge of interest, resulting in the area of the gray surface to be counted toward the discrete mean curvature measure.

for two reasons: the discretized surface now consists of curved triangles with nonzero curvatures, and these triangles now meet at quadratic edges with varying dihedral angles. To resolve them in a principled way, we recall the discrete mean curvature measure from Cohen-Steiner and Morvan [2003] and generalize it to curved triangular surfaces:

Definition 4.1 (Discrete mean curvature measure). Consider a discretized surface Π built from a second-order finite element partition $\{T_j\}$ of its (u, v) -parametrization in Ω . For every (Borel) set $B \subset \mathbb{R}^3$, we define the discrete mean curvature measure $\alpha_\Pi(B)$ as

$$\alpha_\Pi(B) = \sum_{\{T_j\}} \int_{B \cap \Pi(T_j)} H_\Pi(\mathbf{p}) d\mathbf{p} + \sum_{\{e_{ij}\}} A(B \cap \Pi(e_{ij})), \quad (6)$$

where $H_\Pi(\mathbf{p})$ is the mean curvature of Π at a surface point \mathbf{p} , e_{ij} is the edge shared by the triangles T_i and T_j , and $A(B \cap \{\Pi(e_{ij})\})$ computes the signed area swept by the arc spanned by the outward unit normals of $\Pi(T_i)$ and $\Pi(T_j)$ along the edge $B \cap \{\Pi(e_{ij})\}$. The sign of A is positive if e_{ij} is convex and negative if concave.

Fig. 3 shows a geometric interpretation of the second term in $\alpha_\Pi(B)$, referred to as the “tube formula” in Cohen-Steiner and Morvan [2003]. The definition states that the contribution of an edge to the mean curvature equals the (incomplete) signed surface area of the tube. Our $\alpha_\Pi(B)$ is a natural extension of the discrete mean curvature measure in first-order elements, in which case the first term in $\alpha_\Pi(B)$ vanishes, and the second term becomes the dihedral angle times the edge length (Fig. 3 left).

We now return to discretizing the bending energy with the discrete mean curvature measure. We aim to provide a discretization scheme that can be computed with quadrature on elements. To achieve this, it is sufficient to explain how to obtain an estimate of $H_{\hat{\mathbf{x}}^{n+1}}$ at each quadrature point \mathbf{q}_k , where the hat symbol indicates the surface is discretized with FEM in Eqn. (4). We first subdivide each T_j into three triangles T_j^k using the center of T_j , where k denotes the triangle edge index. Next, for each edge e_{ij} , we consider a

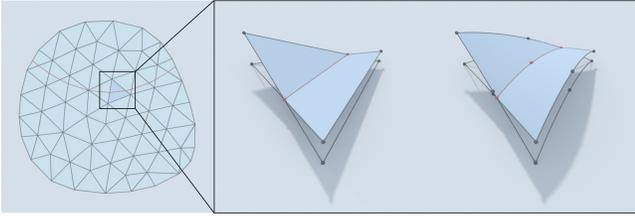


Figure 4: Left: a sample triangulation of a domain Ω with a crease (red curve) splitting multiple triangular elements. Middle and right: One such triangle is split into two linear (middle) and curved (right) triangles accompanied by extra virtual nodes (black dots, respectively). The crease moves passively with the deformation from triangles.

Borel set B_{ij} whose intersection with $\hat{\mathbf{x}}^{n+1}$ is exactly $\hat{\mathbf{x}}^{n+1}(T_i^{k(e_{ij})})$ and $\hat{\mathbf{x}}^{n+1}(T_j^{k(e_{ij})})$ (Fig. 3). Eq.6 suggests the following mean curvature estimate for any $(u, v) \in T_i^{k(e_{ij})} \cup T_j^{k(e_{ij})}$:

$$\bar{H}_{\hat{\mathbf{x}}^{n+1}}(u, v) = H_{\hat{\mathbf{x}}^{n+1}}(u, v) + \frac{A(B_{ij} \cap \hat{\mathbf{x}}^{n+1}(e_{ij}))}{|\hat{\mathbf{x}}^{n+1}(T_i^{k(e_{ij})}) \cup \hat{\mathbf{x}}^{n+1}(T_j^{k(e_{ij})})|}, \quad (7)$$

where the denominator computes the surface area of the two deformed subdivided triangles. The intuition is that $\{B_{ij} \cap \hat{\mathbf{x}}^{n+1}\}$ constitutes a partition of the surface $\hat{\mathbf{x}}^{n+1}$. Therefore, if we uniformly distribute the edge contribution A onto every point in the two adjacent triangles, integrating \bar{H} on the whole surface $\hat{\mathbf{x}}^{n+1}$ equals $\alpha_{\hat{\mathbf{x}}^{n+1}}(\mathbb{R}^3)$ as desired. We detail our implementation of the bending energy in the supplementary material.

5 COUPLING

We now extend our numerical model in Sec. 4 to couple surfaces with embedded curves with potential applications in simulating creases, wrinkles, or elastic rods. We skip the analysis on the continuous model and present our results in the discretized form. For simplicity, we will illustrate the basic idea using one curve on the planar rest surface.

Geometric representation. We recall the domain Ω with a triangulation $\{T_i\}$ and consider a curve embedded in Ω with both ends on its boundary. We discretize the curve by a series of line segments $\{l_i\}$ connecting the intersections of the curve with triangle edges, where the subscript i is the index of the triangle that contains l_i (Fig. 4). Each l_i now cuts T_i into two pieces T_i^l and T_i^r that can deform in their own ways as long as they remain coincident on l_i , which can also stretch or bend along with the surface deformation.

Cut-cell creases. An immediate solution here is to sub-divide every quadrilateral T_i^l and T_i^r into two triangles and apply Sec. 4 to the new triangulation. However, this conceptually simple method suffers from the complexity of adaptive remeshing. Instead, we ground our method on the virtual node algorithm [Bedrossian et al. 2010] to avoid the expensive remeshing of Ω . Consider a specific l_i that splits T_i into T_i^l and T_i^r . Each of them obtains copies of nodes in T_i which control its deformation (Fig. 4 middle and right).

To ensure T_i^l and T_i^r remain coincident on l_i after deformation, it is sufficient to check they agree on the positions of a finite number

of nodes in l_i . For example, when T_i is a first-order finite element, l_i remains a line regardless of the deformation, whose location can be fully determined by checking its two endpoints. Noting that the (u, v) -coordinates of these endpoints are known beforehand, we conclude from Eqn. (4) that their deformed positions are linear on the degrees of freedom at virtual nodes. Therefore, such constraints are linear, which forms a large linear equality constraint $\mathbf{Ax} = \mathbf{b}$ in Eqn. 3. Here, the notation has flattened \mathbf{x} into a column vector, and \mathbf{A} is sparse and typically has much fewer rows than columns. From counting the rows and columns of \mathbf{A} , we notice that this combination of virtual node schemes with second-order instead of first-order finite elements is not arbitrary. Interestingly, coupling it with first-order elements leads to a loss of degrees of freedom on the curve, which we detail in the supplementary material.

In summary, the full numerical method in Sec. 4 remains valid but requires three modifications: First, the quadratures need to be sampled on T_i^l and T_i^r instead of on the full T_i . Second, we need to add extra bending energy along l_i . Finally, the numerical optimization now includes slightly more linear equality constraints, which requires us to solve a symmetric indefinite KKT system at each Newton iteration [Boyd et al. 2004].

Dynamic curves. Handling dynamic curves, e.g., elastic rods with their own density and elasticity, requires small algorithmic updates on the method stated before. First, we add conventional spring energy between neighboring vertices on a rod to our previously defined potential energy. Second, we add to E_{bd} the bending energy from the curve introduced by Bergou et al. [2008], which is defined by integrating the squared curve curvature along the curve.

6 RESULTS

6.1 Implementation Details

We implement our method in C++ on a personal workstation with moderate hardware specifications (12 CPU cores and 16 GB memory). We implemented Newton's method with the standard line search and Hessian definiteness fix techniques [Nocedal and Wright 2006] and used SuiteSparse [Davis 2023] to solve sparse linear systems. We used the geometry library from Sharp et al. [2019] for geometry processing like triangulation and subdivision. Contact and collision were handled with our implementation of IPC [Li et al. 2020]. Simulating one timestep in our largest experiment (80K second-order elements with 480K degrees of freedom) took 30 seconds to finish. The total time for running each experiment ranges from several minutes to at most two hours.

6.2 Evaluation

We compare our approach with first-order elements and analyze their differences in handling low-quality triangulation, kinematic locking, and computational cost. Additionally, we compare our implementation with other higher-order methods implemented in OOFEM, an open-source finite element implementation.

Low-quality meshing. For the finite element method, the choice of basis functions and the mesh quality are two critical factors that jointly affect its numerical performance [Johnson 2012]. Because quadratic basis functions are more expressive than linear ones, we

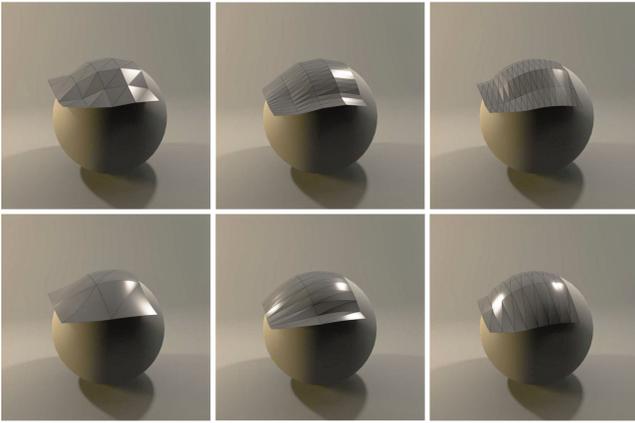


Figure 5: Cloth falling on a sphere. Top row are obtained by first-order elements with respective resolutions of 4×4 , 4×16 , and 16×4 (left to right). Bottom row are obtained by second-order elements with respective resolutions of 2×2 , 2×8 , and 8×2 , matching the number of DoFs of its counterpart.

speculate that our approach is more tolerant of low-quality meshes than traditional first-order methods. An immediate implication is that we anticipate second-order elements to generate similar-quality simulation results as first-order elements but with a coarser triangulation. To validate this hypothesis, we consider a motivating example of a piece of cloth falling on a static, rigid sphere. We deliberately used an extremely coarse triangulation and visualized the simulation results from first-order elements (32 triangles) and second-order elements (8 triangles) in Fig. 5. The planar triangles from first-order elements essentially quantized the underlying smooth motion, and noticeable artifacts from the piecewise-linear shapes are evident under such a low resolution. On the other hand, the curved triangles captured the motion more convincingly and exhibited a smoother surface, thanks to the expressiveness of their quadratic basis. This experiment indicates that our method can be a promising approach when high-resolution meshes are unaffordable. We include more experiments about the performance of our finite element methods on low-quality triangulations in the supplementary material.

Kinematic locking. Strong stretching coefficients rigidify first-order triangular elements due to their planar nature and results in the *kinematic locking* issue, i.e., a multi-facet surface that can only bend along prescribed edges. On the contrary, our model uses bendable curved elements that alleviate the strong coupling between stretching and bending. We provide results of two experiments to illustrate this property. In each experiment, the mesh resolutions were chosen so that linear and quadratic elements created a similar number of nonzeros in the energy Hessian, resulting in a similar computational time in total (Table. 1).

In the first experiment, we simulated a scene *Saddle* with four groups of test cases (Fig. 6). Each group compares the quadratic elements on a square with linear ones. The resolutions of quadratic

and linear cases are $10 \times 10 \times 2$ and $30 \times 30 \times 2$ triangles. We vary the stretching stiffness (Young’s modulus) across groups. We fix the locations of a pair of diagonal corners and simulate the motion of the square under gravity with an expectation of a saddle surface. The results are shown in Fig. 6. We adjust the bending stiffness such that when the diagonal edges are aligned with the bending direction, the quadratic and linear cases produce similar results. However, when the diagonal edges are orthogonal to the bending direction, the linear elements clearly show resistance to bending. This result shows the bending ability of linear elements is dependent on the triangulation of the surface. In contrast, the quadratic elements are better able to bend despite the same direction of triangulation. This observation suggests that our second-order approach remedies the kinematic locking issue.

In the second experiment, we simulated a scene *Curtain* with two groups of comparisons at different resolutions. Each group simulated a swinging curtain using linear and quadratic elements. Fig. 7 shows that under a similar computational time budget, our method produced a smoother surface at each frame that contrasts the polygonal artifacts in the linear elements caused by kinematic locking. The result suggests that our method is practical for alleviating kinematic locking without performance degradation.

Scene	<i>Saddle</i>		<i>Curtain</i>			
	Q-10	L-30	Q-30	L-90	Q-60	L-180
Element type						
DoFs	1323	2883	11163	24843	43923	98283
NNZ	88K	106K	803K	949K	3227K	3793K
Time (s)	0.029	0.038	0.768	0.820	3.480	3.513

Table 1: Computational cost of the linear (“L”) and quadratic (“Q”) elements in Fig. 6 *Saddle* and Fig. 7 *Curtain*. “NNZ” stands for the number of nonzeros in the resultant energy Hessian. The number after “Q-” and “L-” indicates the mesh resolution.

Comparison with other higher-order elements. As mentioned in the related works, lots of previous higher-order shell elements assumed geometric linearity, which often leads to a visually implausible shape with large deformation. To illustrate this, we compare our method with triangular shell elements in OOFEM [Patzák 2012], an open-source finite element implementation. We compare with three linear elements `tr_shell02`, `tr_shell01` and `RerShell` which are not suitable for large deformation due to their assumption on geometric linearity and their lack of coupling between stretching and bending. Given a plate square mesh ($80 \times 80 \times 2$ triangles) on the XY-plane, we fix one boundary to have it swing under gravity. All three types of elements showed artifacts with large deformation (Fig. 8 and supplementary material).

For nonlinear shell elements, rotational DoFs often leads to convergence issues [Hu et al. 2021]. We experimented `Tr2Shell17`, the only triangular element in OOFEM that supports large deformation, which showed slow convergence in a cantilever beam test with moderate loads. On a $5 \times 5 \times 2$ (50 elements, 6 nodes per element, 7 DoFs per node) shell, with 0.03/0.04/0.044 dimensionless loads

per node on free edge, it requires 25/84/586 iterations to converge with Newton's method and failed to converge with larger loads. In comparison, our method converged within 5 iterations in a similar problem.

6.3 Applications

We present seven examples in cloth simulation, mixed Origami and Kirigami structures, and their coupling with elastic rods.

Large curtain: This example simulates a curtain on $200 \times 200 \times 2$ triangular elements (Fig. 9). The curtain swings under gravity while five evenly-spaced rings on top of the curtain move along a rod. The simulation shows an elastic, highly vibrant curtain with detailed wrinkles.

Lanterns: This example simulates two lanterns (Fig. 1) with a cylindrical mesh on a resolution of $200 \times 200 \times 2$ triangles. There are 20 rods traversing the cylinder and no edge bending energy on edges where rods reside. The bottom of the cylinder is set to be fixed, and the top moves downward to squeeze the whole structure. With different stretching stiffness, the cylinders deform to different shapes: a smooth ellipsoid and a shape with curved creases.

Bat: This example simulates a bat wing on a mesh with 3600 elements. Elastic rods are inserted in the mesh by virtual nodes to simulate arms and fingers (Fig. 12). External periodic forces are applied at joints to simulate flying movements of the bat.

Tori: This example simulates a piece of cloth with 459 elements falling on two tori from above (Fig. 10). The cloth eventually rests on both tori and shows a smooth surface capturing the underlying tori shapes with the small number of curved triangles.

Origami and Kirigami: We present three more examples that simulate the dynamics of mixed Origami and Kirigami structures. Fig. 14 shows a paper-art work whose structure consists of an array of slanted paper tapes with Origami creases at linearly interpolated locations. The Kirigami design is simulated with only 572 triangles, yet it manifests smooth, large-scale elastic behaviors with visually appealing deformation and vibration. Fig. 13 shows a second Kirigami example on a 32×32 square mesh under gravity. Finally, we simulated another Kirigami design of a paper discretized into 600 triangular elements (Fig. 11). Folding the paper along its major crease produces a stair structure resulting from the spatially varying crease location at each paper tape. One of the four boundaries of the paper is fixed to a static rod so that when spinning the paper dynamically, it automatically folds to generate the final Kirigami.

7 CONCLUSIONS

This paper presented a novel second-order deformable-surface simulation algorithm. The framework discretizes a deformable surface using finite elements on curved triangles, which overcomes the prolonged limitations of linear element methods in tackling problems of anisotropic bending, locking, and low-resolution dynamics. We compared our approach with other state-of-the-art finite element methods and demonstrated its efficacy in terms of simulation details and smoothness. Our method facilitates simulations on low-resolution or low-quality meshes. This feature opens up

new possibilities to incorporate our method into the various interactive design, visualization, and geometry processing pipelines. It provides high-fidelity simulation results based on imperfect mesh data. Thanks to the second-order nature of our discretization, our method is particularly suited for producing authentic simulations on Origami and Kirigami designs, which were challenging for traditional thin-shell methods due to their mixed creases, wrinkles, low-resolution meshes, and anisotropic mesh elements.

ACKNOWLEDGMENTS

Tao Du would like to thank Tsinghua University and Shanghai Qi Zhi Institute for supporting this research. Bo Zhu and Yitong Deng acknowledges the funding supports from NSF IIS-2106733.

REFERENCES

- G Abaqus. 2011. Abaqus 6.11. *Dassault Systemes Simulia Corporation, Providence, RI, USA* (2011).
- Chandrajit L. Bajaj, Guoliang Xu, and Qin Zhang. 2007. Smooth surface constructions via a higher-order level-set method. In *CAD/Graphics*. 27.
- David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. 43–54.
- Timothy Barth and Paul Frederickson. 1990. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. In *28th aerospace sciences meeting*. 13.
- Klaus-Jürgen Bathe and Lee-Wing Ho. 1981. A simple and effective element for analysis of general shell structures. *Computers & Structures* 13, 5-6 (1981), 673–681.
- Jean-Louis Batoz, Klaus-Jürgen Bathe, and Lee-Wing Ho. 1980. A study of three-node triangular plate bending elements. *International journal for numerical methods in engineering* 15, 12 (1980), 1771–1812.
- Jacob Bedrossian, James H. Von Brecht, Siwei Zhu, Eftychios Sifakis, and Joseph M Teran. 2010. A second order virtual node method for elliptic problems with interfaces and irregular domains. *J. Comput. Phys.* 229, 18 (2010), 6405–6426.
- T. Belytschko, H. Stolarski, and N. Carpenter. 1984. A C0 triangular plate element with one-point quadrature. *Internat. J. Numer. Methods Engrg.* 20, 5 (1984), 787–802.
- Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete Elastic Rods. *ACM Transactions on Graphics (TOG)* 27, 3 (aug 2008), 63:1–63:12.
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)* 33, 4 (2014), 1–11.
- Eddy Boxerman. 2003. *Speeding up cloth simulation*. Ph. D. Dissertation. University of British Columbia.
- Stephen Boyd, Stephen P Boyd, and Lieven Vandenbergh. 2004. *Convex optimization*. Cambridge university press.
- David E. Breen, Donald H. House, and Michael J. Wozny. 1994. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. 365–372.
- R. Bridson, S. Marino, and R. Fedkiw. 2003. Simulation of Clothing with Folds and Wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, California) (SCA '03). Eurographics Association, Goslar, DEU, 28–36.
- Oleksiy Busaryev, Tamal K. Dey, and Huamin Wang. 2013. Adaptive fracture simulation of multi-layered thin plates. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–6.
- EMB Campello, PM Pimenta, and P Wriggers. 2003. A triangular finite shell element based on a fully nonlinear shell formulation. *Computational mechanics* 31, 6 (2003), 505–518.
- Hsiao-Yu Chen, Arnav Sastry, Wim M. van Rees, and Etienne Vouga. 2018. Physical simulation of environmentally induced thin shell deformation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.
- Zhili Chen, Renguo Feng, and Huamin Wang. 2013. Modeling friction and air effects between cloth and deformable bodies. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–8.
- 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 Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted Delaunay Triangulations and Normal Cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*. 312–321.
- Tim Davis. 2023. SuiteSparse: A Suite of Sparse matrix packages. <https://github.com/DrTimothyAldenDavis/SuiteSparse>.

- Sailkat Dey, Robert M. O'bara, and Mark S. Shephard. 1999. Curvilinear Mesh Generation in 3D. In *IMR*. Citeseer, 407–417.
- Bernhard Eberhardt, Olaf Eitzmuß, and Michael Hauth. 2000. Implicit-explicit schemes for fast animation with particle systems. In *Computer Animation and Simulation 2000*. Springer, 137–151.
- Bernhard Eberhardt, Andreas Weber, and Wolfgang Strasser. 1996. A fast, flexible, particle-system model for cloth draping. *IEEE Computer Graphics and Applications* 16, 5 (1996), 52–59.
- Yotam Gingold, Adrian Secord, Jefferson Y Han, Eitan Grinspun, and Denis Zorin. 2004. A discrete model for inelastic deformation of thin shells. In *ACM SIGGRAPH/Eurographics symposium on computer animation*. Citeseer.
- 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*. Citeseer, 62–67.
- Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R. Chakravarthy. 1997. *Uniformly high order accurate essentially non-oscillatory schemes, III*. Springer.
- Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. 2018. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.
- Y. C. Hu, Y. X. Zhou, K. W. Kwok, and K. Y. Sze. 2021. Simulating flexible origami structures by finite element method. *International Journal of Mechanics and Materials in Design* (2021), 1–29.
- H. T. Huynh, Zhi J. Wang, and Peter E. Vincent. 2014. High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids. *Computers & fluids* 98 (2014), 209–220.
- Jeremy Ims, Zhaowen Duan, and Zhi J. Wang. 2015. meshCurve: an automated low-order to high-order mesh generator. In *22nd AIAA computational fluid dynamics conference*. 2293.
- Antony Jameson. 2011. Advances in bringing high-order methods to practical applications in computational fluid dynamics. In *20th AIAA Computational Fluid Dynamics Conference*. 3226.
- Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–10.
- Claes Johnson. 2012. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Courier Corporation.
- Couro Kane, Jerrold E Marsden, Michael Ortiz, and Matthew West. 2000. Variational Integrators and the Newmark Algorithm for Conservative and Dissipative Mechanical Systems. *Internat. J. Numer. Methods Engrg.* 49, 10 (2000), 1295–1325.
- Theodore Kim and David Eberle. 2020. Dynamic deformables: implementation and production practicalities. In *ACM SIGGRAPH 2020 Courses*. 1–182.
- H. Laurent and G. Rio. 2001. Formulation of a thin shell finite element with continuity C 0 and convected material frame notion. *Computational Mechanics* 27, 3 (2001), 218–232.
- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Transactions on Graphics (TOG)* 39, 4, Article 49 (2020).
- Mickaël Ly, Romain Casati, Florence Bertails-Descoubes, Mélina Skouras, and Laurence Boissieux. 2018. Inverse elastic shell design with contact and friction. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–16.
- Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. 2020. Projective dynamics with dry frictional contact. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 57–1.
- Richard H. Macneal. 1982. Derivation of element stiffness matrices by assumed strain distributions. *Nuclear Engineering and Design* 70, 1 (1982), 3–12.
- Luigi Malomo, Jesús Pérez, Emmanuel Iarussi, Nico Pietroni, Eder Miguel, Paolo Cignoni, and Bernd Bickel. 2018. FlexMaps: computational design of flat flexible shells for shaping 3D objects. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14.
- Amin Mahmoudi Moghadam, Mehdi Shafieefar, and Roozbeh Panahi. 2016. Development of a high-order level set method: Compact conservative level set (CCLS). *Computers & Fluids* 129 (2016), 79–90.
- Rahul Narain, Tobias Pfaff, and James F O'Brien. 2013. Folding and crumpling adaptive sheets. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–8.
- Rahul Narain, Armin Samii, and James F. O'brien. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)* 31, 6 (2012), 1–10.
- Jean-Christophe Nave, Rodolfo Ruben Rosales, and Benjamin Seibold. 2010. A gradient-augmented level set method with an optimally local, coherent advection scheme. *J. Comput. Phys.* 229, 10 (2010), 3802–3827.
- Jorge Nocedal and Stephen Wright. 2006. *Numerical Optimization*. Springer Science & Business Media.
- Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. 2021. Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.
- Julian Panetta, Mina Konaković-Luković, Florin Isvoranu, Etienne Bouleau, and Mark Pauly. 2019. X-shells: A new class of deployable beam structures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–15.
- Bořek Patzák. 2012. OOFEM—an object-oriented simulation tool for advanced modeling of materials and structures. *Acta Polytechnica* 52, 6 (2012).
- Tobias Pfaff, Rahul Narain, Juan Miguel De Joya, and James F. O'Brien. 2014. Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–9.
- R. Phaal and CR. Calladine. 1992. A simple class of finite elements for plate and shell problems. II: An element for thin shells, with only translational degrees of freedom. *Internat. J. Numer. Methods Engrg.* 35, 5 (1992), 979–996.
- Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. 2018. Decoupling simulation accuracy from mesh quality. *ACM transactions on graphics* (2018).
- Teseo Schneider, Yixin Hu, Xifeng Gao, Jérémie Dumas, Denis Zorin, and Daniele Panozzo. 2022. A large-scale comparison of tetrahedral and hexahedral elements for solving elliptic PDEs with the finite element method. *ACM Transactions on Graphics (TOG)* 41, 3 (2022), 1–14.
- Nicholas Sharp, Keenan Crane, et al. 2019. geometry-central. www.geometry-central.net.
- Jonathan Shewchuk. 2002. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley* 73 (2002), 137.
- Chi-Wang Shu. 2003. High-order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD. *International Journal of Computational Fluid Dynamics* 17, 2 (2003), 107–118.
- Eftychios Sifakis and Jernej Barbic. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses* (Los Angeles, California) (SIGGRAPH '12). Association for Computing Machinery, New York, NY, USA, Article 20, 50 pages.
- Eftychios Sifakis, Sebastian Marino, and Joseph Teran. 2008. Globally coupled collision handling using volume preserving Impulses.. In *SCA*. 147–153.
- Ari Stern and Eitan Grinspun. 2009. Implicit-explicit variational integration of highly oscillatory problems. *Multiscale Modeling & Simulation* 7, 4 (2009), 1779–1794.
- K. Y. Sze and Dan Zhu. 1999. A quadratic assumed natural strain curved triangular shell element. *Computer methods in applied mechanics and engineering* 174, 1-2 (1999), 57–71.
- 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.
- Alexander Tessler and Thomas JR Hughes. 1985. A three-node Mindlin plate element with improved transverse shear. *Computer Methods in Applied Mechanics and Engineering* 50, 1 (1985), 71–101.
- Oleg V. Vasilyev. 2000. High order finite difference schemes on non-uniform meshes with good conservation properties. *J. Comput. Phys.* 157, 2 (2000), 746–761.
- Peter E. Vincent and Antony Jameson. 2011. Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists. *Mathematical Modelling of Natural Phenomena* 6, 3 (2011), 97–140.
- Peter E. J. Vos, Spencer J. Sherwin, and Robert M. Kirby. 2010. From h to p efficiently: Implementing finite and spectral/hp element methods to achieve optimal performance for low-and high-order discretisations. *J. Comput. Phys.* 229, 13 (2010), 5161–5181.
- Zhijian J. Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T. Huynh, et al. 2013. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids* 72, 8 (2013), 811–845.
- Anna Wawrzinek, Klaus Hildebrandt, and Konrad Polthier. 2011. Koiter's thin shells on catmull-clark limit surfaces.. In *VMV*. 113–120.
- Xu Zhongnian. 1992. A thick–thin triangular plate element. *Internat. J. Numer. Methods Engrg.* 33, 5 (1992), 963–973.
- Xianlian Zhou and Jia Lu. 2005. NURBS-based Galerkin method and application to skeletal muscle modeling. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*. 71–78.

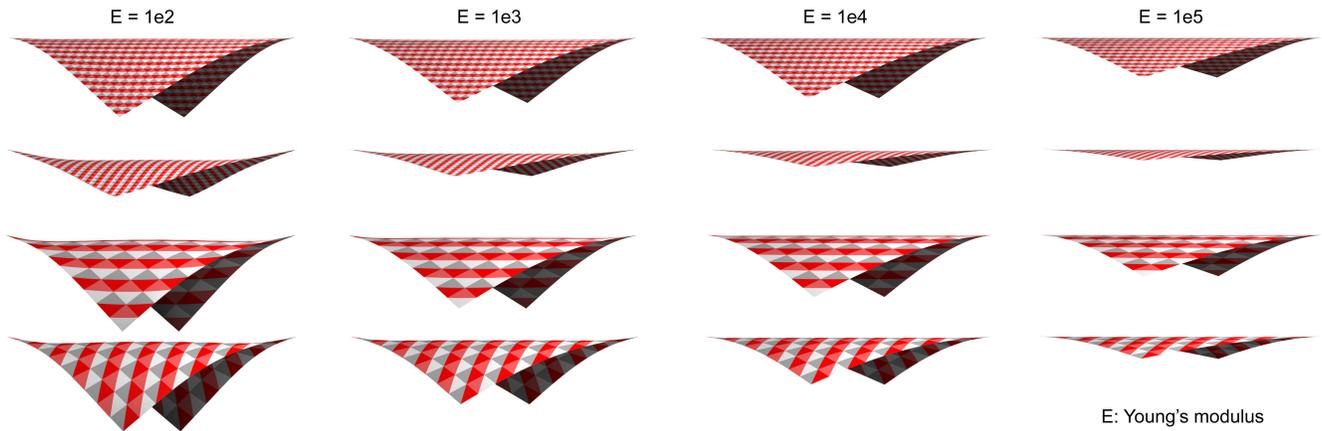


Figure 6: *Saddle*: This figure demonstrated the ability of quadratic elements to alleviate kinematic locking with various young’s modulus (E). The first two rows are linear elements with diagonal edges aligned with and orthogonal to the bending direction. The next two rows are quadratic elements with such edges. The linear elements are on a resolution of $30 \times 30 \times 2$ triangles, and quadratic elements are of $10 \times 10 \times 2$ triangles. The triangulations of the surfaces are visualized by red/white/grey triangles.

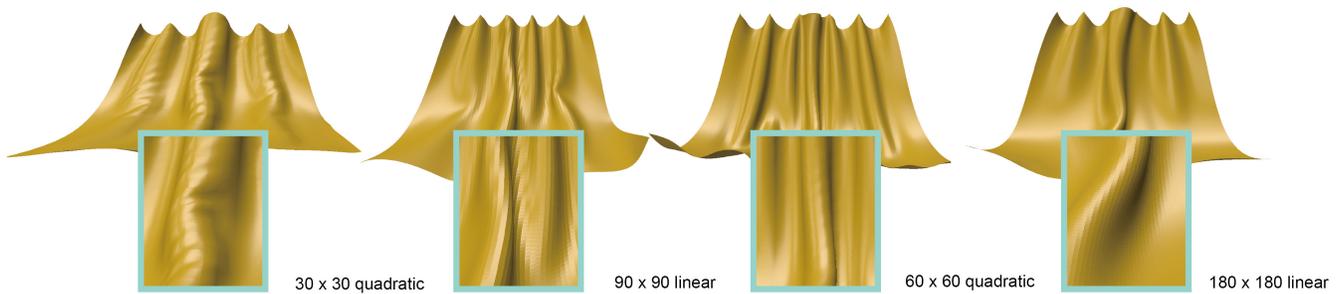


Figure 7: *Curtain*: Quadratic elements at 30×30 (60×60) resolution are compared with linear elements at 90×90 (180×180) resolution. The curtain swings under gravity with 6 evenly-spaced positions on the top moving horizontally. The close-ups of location where locking is apparent are shown in blue boxes.

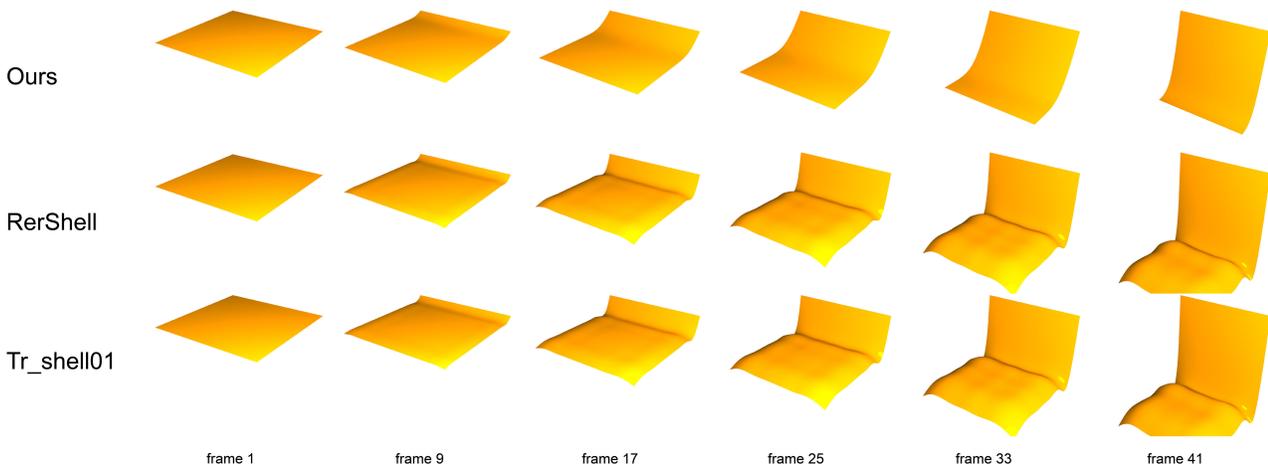


Figure 8: *Comparison with other higher-order elements*: Comparison of our method (top) with RerShell (middle) and tr_shell01 (bottom). A square swings under gravity with one side fixed. Only our method presents reasonable result. The other two elements deform incorrectly at the bottom of the surface.



Figure 9: *Large curtain*: The left figure shows an intermediate frame from simulating the curtain with $200 \times 200 \times 2$ triangles with textures. The right figure displays the same frame without texture to showcase the rich wrinkles.



Figure 10: *Tori*: A piece of cloth simulated with 459 elements is in contact with two solid tori (top-right inset) and gently slips.

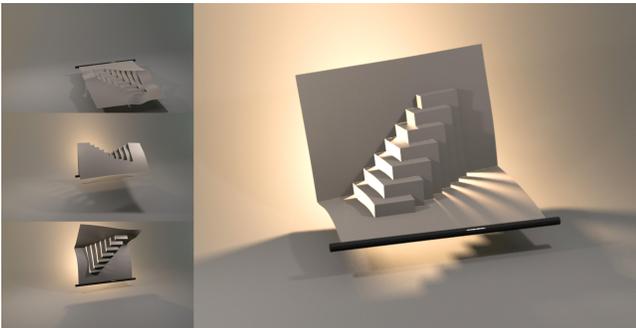


Figure 11: *Stairs*: A Kirigami design simulated with 600 elements. Starting from a planer rest shape (top left), the paper was pulled by its creases with non-zero rest angles (middle and bottom left) to form a stair-like kirigami (right).

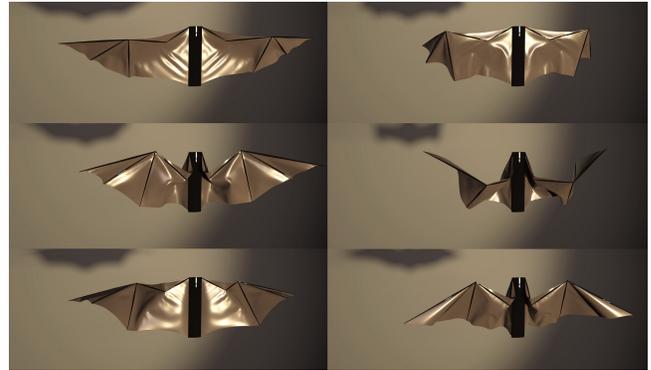


Figure 12: *Bat*: A bat wing composed of elastic rods and shells. The motion is controlled by rods representing "arms". See the video for more details.

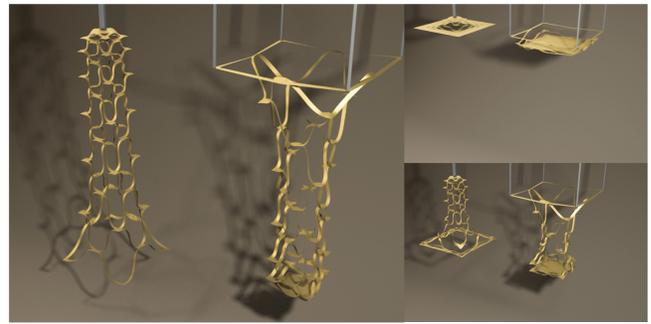


Figure 13: *Kirigami*: A Kirigami design simulated with 32×32 elements. Starting from a planer rest shape (top right), the shell drooped under gravity.



Figure 14: *Paper art*: A paper-art example assembled by ten paper strips with creases is simulated using 572 elements. The left ends of the strips are fixed, and the right ends move leftward to force the strips to pop upward. The right rows are middle frames of the simulation.