

Large Growth Deformations of Thin Tissue using Solid-Shells

Danny Huang, Ian Stavness

Abstract—Simulating large scale expansion of thin structures, such as in growing leaves, is challenging. Solid-shells have a number of potential advantages over conventional thin-shell methods, but have thus far only been investigated for small plastic deformation cases. In response, we present a new general-purpose FEM growth framework for handling a wide range of challenging growth scenarios using the solid-shell element. Solid-shells are a middle-ground between traditional volume and thin-shell elements where volumetric characteristics are retained while being treatable as a 2D manifold much like thin-shells. These elements are adaptable to accommodate the many techniques that are required for simulating large and intricate plastic deformations, including morphogen diffusion, plastic embedding, strain-aware adaptive remeshing, and collision handling. We demonstrate the capabilities of growing solid-shells in reproducing buckling, rippling, curling, and collision deformations, relevant towards animating growing leaves, flowers, and other thin structures. Solid-shells are compared side-by-side with thin-shells to examine their bending behavior and runtime performance. The experiments demonstrate that solid-shells are a viable alternative to thin-shells for simulating large and intricate growth deformations.

Index Terms—Computer Graphics, Finite element methods, Physically based modelling, Animation, Applications

1 INTRODUCTION

GROWTH is a common feature exhibited across all living organisms. Even inanimate materials may grow or shrink when subjected to heat, moisture, and other environmental conditions. However, compared to elastic deformations, the animation of growing structures has received much less attention in the physics-based simulation literature. This is likely due to the unique and difficult challenges for simulating growing materials. A primary challenge is handling the complex deformations associated with growth. These deformations are large, permanent, and for thin tissue, often involve intricate shapes and patterns. The secondary challenge is accounting for the factors that govern and influence morphology development. These factors can include the chemicals, referred to as *morphogen*, that spread (i.e. diffuse) throughout tissue and signal where growth is to occur [1]. Even the collisions that occur between adjacent growing tissues can influence their morphology [2, 3].

Early physics-based simulations used linear mass-spring elements as the tissue constitutive model [4]. Mass-springs are popular for their simplicity, but have challenges in simulating biological material accurately. The linear mass-springs utilized are limited to some variant of Hooke's law and require that the mesh configuration is kept sufficiently isotropic to avoid unintended anisotropic behavior [5]. These limitations can be overcome using the finite element method (FEM), which is capable of modelling processes that are governed by partial differential equations (PDE) over a continuous domain. FEM-based tissue can model elasticity based on 3D stress-strain relationships, while supporting diffusion and specified anisotropic behavior [6]. However, these advantages come at a price of increased computational

and modelling complexity. In order to represent thin tissue, recent FEM frameworks have incorporated thin-shell and solid-shell elements. These shell elements are specialized to handle steep bending deformations efficiently, allowing intricate deformations to be modeled. Thin-shells simplify the model as a 2D manifold with accompanying thickness assumptions (e.g. infinitesimally thin, strain-free or stress-free, and shearing-free), making them specialized for modelling thin tissue [7]. In contrast, solid-shells are derived from volume elements to retain thickness dynamics and compatibility with existing 3D material laws [8].

For a given a constitutive model, there are a number of ways in which plasticity can be modelled. Mass-springs can have adjustable spring rest lengths to induce permanent deformation [4]. For FEM simulations, a straight-forward approach is *multiplicative decomposition* where the deformation gradient is decomposed into an elastic component and multiplicative plastic offset: $\mathbf{F} = \mathbf{F}_e \mathbf{F}_p$ [9]. For larger plastic deformations where the multiplicative plastic offset can cause numerical instability, growth simulations involving membrane (i.e. thin-shell elements without bending forces) and volumetric elements use an alternative approach, referred to as *plastic embedding* by Wicke et al. [10], where the stress-free rest configuration of the FEM model is permanently deformed according to the growth strain until the strain is minimized. Plastic embedding has been successful in modelling large plastic deformations for volumetric [6], membrane [11], and thin-shell elements [12]. However, for solid-shells, incorporating plastic embedding is unexplored.

When growth is prolonged, it is necessary to adaptively remesh the model to prevent mesh degradation. In existing growth simulators, these remeshing operations are triggered according to the geometry of the mesh. Outside the growth literature, there exists strain-aware remeshers that can selectively and preemptively refine a mesh where surface

• Authors are with University of Saskatchewan.

Manuscript received April 19, 20XX; revised September 12, 20XX.

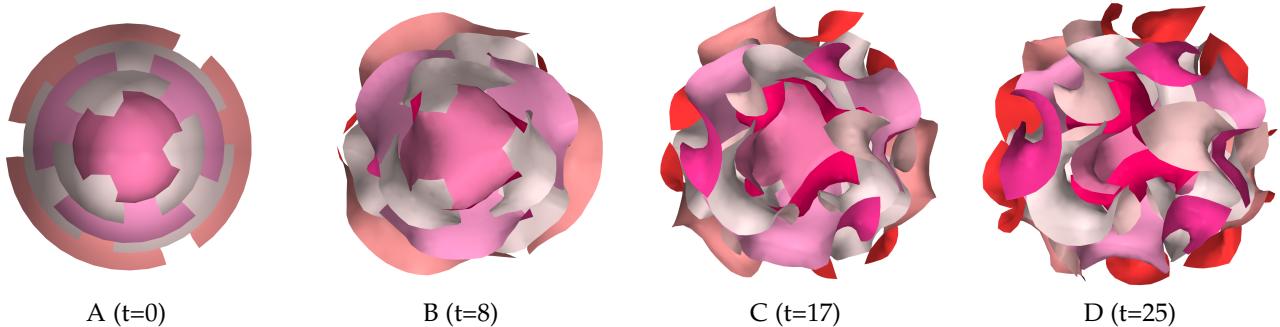


Fig. 1. Time-Lapse Growth of a Bouquet of Ripples. Starting with five thin layers composed of solid-shells, stretch-inducing morphogen is diffused along the boundary of each layer. The rippling growth deformations become large enough for collisions to occur between neighboring ripples.

buckling is imminent [13].

Along with handling complex deformations, previous works have integrated other components together to govern and influence growth. A prominent component is morphogen diffusion which models the spreading of chemicals that signal growth [4, 6, 11]. Another component that influences growth are collisions. Especially for large deformations, growing tissue can take up specific shapes due to the space-constraints imposed by neighboring tissue. Collision handling has been experimented with growth simulations involving mass-spring models [14] and FEM thin-filament models [15]. For FEM thin tissue, collision handling remains largely unaccounted for in growth applications.

Solid-shells remain undeveloped for handling the large plastic deformations for simulating intricate growth deformations. Much of the literature in growing solid-shells utilize the multiplicative decomposition approach and lack adaptive remeshing [16, 17, 18], which limit the solid-shell experiments within the scope of simulating small growth deformations. Consequently, solid-shells remain largely overlooked compared to thin-shells for handling large growth deformations. This presents an opportunity to elevate solid-shells with essential growth components and demonstrate its capabilities, including its ability to account for thickness shearing in comparison with thin-shells. We expect that this is important for capturing the physical properties of thin structures that retain some thickness and rigidity, such as plant leaves, flower petals, and other materials where the infinitesimal thickness assumption does not necessarily hold [6].

We present a general-purpose FEM framework that introduces solid-shells for simulating the growth of complex deformations. Specifically, we make two contributions:

(1) We demonstrate that solid-shells are adaptable to simulate a wide range of challenging growth scenarios involving large and intricate plastic deformations. Fundamental techniques, including plastic embedding, strain-aware adaptive remeshing, morphogen diffusion, and collision handling have been successfully integrated within the versatile solid-shell element in order to accurately simulate large-scale growth.

(2) We directly compare solid-shell and thin-shell elements undergoing plastic bending strain within the same FEM framework. Elements are compared in terms of bending behavior and runtime performance to reveal the impact of thickness shearing dynamics on growing morphology.

We perform qualitative investigations on the capabilities of the solid-shell growth framework in a number of contexts relevant to simulating the growth of leaves, fruits, and flower petals. These simulations illustrate the wide range of emergent patterns that the augmented solid-shells are able to achieve, including buckling, rippling, and curling deformations. Beyond existing FEM growth frameworks, we include demonstrations of large and intricate plastic deformations with collision handling, and the ability to generate detailed growth models starting from a coarse mesh by using the strain-aware remesher. In addition, solid-shells and thin-shells are experimentally compared to examine the added effect of thickness dynamics. Altogether, we provide experiments to demonstrate that the solid-shell are a viable alternative to thin-shells for simulating growth involving large and intricate plastic deformations of thin structures.

2 RELATED WORKS

Mass-Spring Models

The early studies of simulating growing tissue have used linear mass-spring elements. The tissue physics is approximated as a set of point-masses where neighboring points are attached using springs. The springs can have their rest lengths and angles modified to simulate plastic deformation, namely growth. Using mass-springs, researchers have been successful in simulating the growth of bulges and ripples that are similarly found in plant leaves [4, 19].

Finite Element Method

Growth can be more accurately simulated using the FEM [20]. The advantage of the FEM is that it can model PDE-governed processes over a continuous domain. 3D stress-strain relationships and diffusion are particularly modeled using the FEM and therefore can be coupled with FEM-based growth simulations. One of the pioneering works of simulating growth using the FEM was done by Kennaway et al. [6]. They have created models of the *Antirrhinum* flower by diffusing growth-inducing morphogen over tissue composed of volumetric elements.

Thin-Shell Elements

When simulating growth of thin tissues using the FEM, thin-shell elements can be used instead of volumetric elements. Thin-shells are notable for the decoupling of stretching and bending kinematics. This allows steep bends between

elements to occur without significantly impacting their stretching strain and therefore reducing the risk of element inversion. Models composed of volumetric elements can tolerate large bends if they are high enough in resolution, but this comes at a price of increased computational complexity relative to using thin-shell elements.

Growth simulations using thin-shells were successful in reproducing the behavior of thin structures subjected to growth. Experiments conducted by Chen et al. [21] were able to simulate the curling of paper and rippling of a disc by diffusing moisture into them, and were comparable to real-life experiments [21]. In the works of Vetter et al. [22], they produced fractal patterns (rippling within rippling) similar to those found in torn plastic sheets and beet leaves.

Solid-Shell Elements

Growth can alternatively be modelled using solid-shell elements. Solid-shells decouple stretching and bending by respectively mapping them to the translation and rotation of the *directors*. These directors, which can be thought as extensible normal vectors, can be described exclusively in terms of node displacement, allowing bending kinematics to be described without introducing notions of angular displacement. Consequently, solid-shells are able to (1) interface with existing linear and hyperelastic 3D material laws, (2) retain thickness compression and shearing dynamics, (3) and describe both stretching and bending in terms of stretching strain. In contrast to (3), thin-shells have a separate formulation for bending kinematics based on angular strain (Narain et al. [23]) or second fundamental form (Chen et al. [21], Vetter et al. [22]). This separation particularly complicates adapting different materials to interface with explicit rotational/curvature quantities, therefore thin-shells are usually fixed to the St. Venant-Kirchhoff material which is valid for the zero thickness limit [24].

Plasticity using Multiplicative Decomposition

Existing FEM growth simulators have their own approaches for modelling plastic deformations. A simple approach is to decompose the deformation gradient into an elastic component and multiplicative plastic offset: $\mathbf{F} = \mathbf{F}_e \mathbf{F}_p$ [9, 18, 22]. In effect, the object's permanently deformed state can be tracked via the plastic offset \mathbf{F}_p , which serves as a reference for the elastic component \mathbf{F}_e to describe the world configuration. While this allows both elasticity and plasticity to be modeled together easily, numerical instability can arise when the plastic strain associated with \mathbf{F}_p becomes too large. In particular, the magnitude of the plastic forces can reach an unsustainable threshold where the simulation cannot converge accurately without gradually reducing the simulation time step size.

Plasticity using Rest Configuration Deformation

Other works allow the rest configuration to deform according to the plastic offset \mathbf{F}_p . The advantage of this approach is that \mathbf{F}_p decrements whenever the rest configuration deforms; \mathbf{F}_p gradually becomes represented by the rest configuration. In effect, the plastic strain associated with \mathbf{F}_p is actively minimized, allowing large plastic deformations to occur without encountering numerical instability. Termed

as *plastic embedding*, Wicke et al. [10] provides a formal description of this plasticity approach for volumetric elements. Plastic embedding and its variations have been adopted for growing volumetric elements [6], membrane elements [11], and thin-shell elements [12]. For solid-shell elements, plastic embedding is largely unimplemented but could be achieved by modifying the rest configuration of its directors.

A caveat to plastic embedding is that the rest configuration of the model is restricted within Euclidean space. For example, given two cube elements that are attached side-by-side (i.e. sharing four vertices), it is not possible to represent a cube that is 2x bigger than the other cube while still sharing four vertices. To get around this caveat, the non-embeddable portion of the plastic strain (i.e. residual strain) is retained and carried until it can be embedded into the rest configuration [6, 23]. Another approach is to capture both the embeddable and non-embeddable portions of plastic strain in a single reference metric tensor by using non-Euclidean plates [7, 21].

Adaptive Remeshing

Growth models should be periodically remeshed to maintain a high quality topology. This involves routinely updating the mesh such that its resolution does not degrade during growth. Otherwise, small and intricate morphologies, such as wrinkles and ripples, would be prevented from emerging. Remeshing can be carried out by subdividing edges that exceed a length threshold or local morphogen concentration threshold where the latter threshold is dynamically raised when subdivision occurs [6]. Along with subdivision, edges can be flipped to help avoid "skinny" elements that have poor aspect ratios [11]. Edges can also be collapsed to eliminate elements that become too skinny or too small [25]. In more sophisticated remeshing schemes, strain-aware thresholding is supported: strained elements are refined to anticipate potential surface buckling, and flat areas with little strain are coarsened into lower resolution elements to reduce computational complexity [13].

Adaptive remeshing offers a passive benefit of reducing the shear locking artifact. In solid-shells where bending is simulated by displacing the bottom nodes to tilt the directors, an artificial stress must be incurred in order to physically accommodate the displaced bottom nodes. This stress can be exemplified in the 2D case where a 4-node rectangle has its bottom side elongated. When using only 4-nodes, the deformed shape can only be captured as a trapezoid rather than a smooth annulus sector. This imposed trapezoid creates a shear stress because all four corners need to displace from their rest 90-degree angle to accommodate the elongated bottom side, which altogether creates a stiffer element than expected. A common solution to avoid this shear lock is to use second-order elements which contain additional nodes to bend the element like a chevron shape, allowing the corner nodes to exactly align with the expected bending curvature. An alternative option is to passively mitigate shear locking via adaptive remeshing which selectively refines the mesh where bending occurs, allowing the refined elements to more closely align with the targeted bending curvature.

TABLE 1

Distinguishing our growth framework from recent related works in the physics-based growth literature. The abbreviated terms are as follows: Mass-Spring (**S**), FEM Membranes or Volumes (**MV**), FEM Solid-Shells (**SS**), FEM Thin-Shells (**TS**), Rest Length (**L**), Multiplicative Offset (**M**), Plastic Embedding (**E**), Reference Metric Tensor (**T**), Length-Aware (**Length**), Strain-Aware (**Strain**), Diffusion via Cotangent Laplacian (**DCL**), Discrete Detection (**D**), Continuous Detection (**C**), Spring or Normal Forces (**F**), Impulses (**Imp**), Open-Source (**OS**).

Framework	Constitutive				Plasticity				Adaptive Remesh		Diffusion		Collision Handling				OS
	S	MV	SS	TS	L	M	E	T	Length	Strain	Yes	DCL	D	C	F	Imp	
Kennaway et al. [6]	X					X			X								X ¹
Vetter [15]			X		X								X		X		
Rosenkrantz and Louis-Rosenberg [14]	X				X				X		X	X	X		X		
Chen et al. [21]			X			X					X						X ²
Kierzkowski et al. [11]	X					X			X		X	X					
Gingras and Kry [12]			X			X					X	X					
Zheng et al. [18]	X			X													
Our Growth Framework	X			X		X			X	X	X	X	X	X	X	X	X ³

Diffusion

The spreading of growth-inducing morphogen can be modelled by the heat equation. The heat equation itself is a second-order PDE that is dependent on the mathematical Laplacian operator Δ with a physical interpretation of modelling the gradient of heat diffusing through a medium towards a steady state [26, Slide 35]. To solve the heat equation, FEM is commonly applied which involves analytically integrating the discrete weak form of the Laplacian and performing a second integration using a numerical time-stepping method (e.g. backward Euler). When integrating the Laplacian weak form, a mass matrix and *discrete cotangent Laplacian matrix* can be constructed [26]. These two matrices are then passed to the time-stepping method which can be solved in close to linear time due to the symmetric positive definite (SPD) property of both matrices. Altogether, this particular diffusion method has been incorporated to simulate morphogen diffusion efficiently [11]. In another work, geodesic distances are computed using the cotangent Laplacian matrix, enabling a specific growth model where the growth rate is dependent on the distance from the tissue boundary [14].

Collision Handling

When growth is prolonged, the model can potentially intersect with itself, justifying the need for collision handling. A straight-forward approach is to deploy a discrete collision detection scheme with repulsion forces [14, 15]. Alternatively, continuous collision detection (CCD) with constraint-specified impulses can be used to avoid collisions [27]. The advantage of CCD is that the time-space path of the vertices, edges, and triangles are accounted for, allowing collisions to be detected precisely and robustly. When collisions are detected, they are resolved using impulses specified by geometric constraints. The impulse approach allows energy and momentum to be conserved and does not require stiffness parameter tuning [28].

Collision impulses can be solved using iterative methods or direct methods. Direct methods, particularly LCP solvers, can simultaneously account for all the given constraints to guarantee that the impulses do not violate each other's constraints [29]. LCP solvers are also compatible with GPU-based implementations to accelerate collision handling [30].

While CCD can detect collisions exactly as they occur, the downside is that collisions become more difficult to resolve in certain cases, such as the case of tangled and pinched meshes [29]. The reasoning is that CCD algorithms aim to separate colliding objects with an infinitesimal distance between them, making secondary and repeating collisions more susceptible in the next time step unless an artificial minimum distance is enforced. To address this difficulty, previous works combine CCD with discrete collision detection [31].

Summary

We summarize the recent related works and our growth framework in Table 1. Our open-source growth framework specializes in using solid-shell elements, which are versatile to accommodate the techniques required for simulating challenging growth scenarios. The elements are equipped with plastic embedding and a strain-aware remesher for handling large and intricate plastic deformations. We further adapt the solid-shell by providing support for modelling morphogen diffusion and collision handling, which altogether govern and influence the growth process. Morphogen diffusion is modelled efficiently by leveraging on the discrete Laplacian operator. Collision handling is made robust using a combination of discrete and continuous collision detection where collisions are resolved using impulses from the LCP solver.

3 METHOD

3.1 High-Level Overview

The framework methodology is outlined in Algorithm 1 with corresponding sections highlighted in red. Growth is simulated by deforming the rest configuration of the solid-shells wherever morphogen is present. The user first specifies the initial locations of the morphogen on a FEM node basis, either through scripting or by using the provided user-interface tool. The morphogen can vary in concentration, *motion type*, and *axis type*. The motion types differ by whether plastic stretching or plastic bending is induced,

¹<https://sourceforge.net/projects/gfbox>

²<https://github.com/evouga/libshell>

³https://github.com/dannyhx/artisynth_core/tree/growth

while the axis type specifies the desired axes that plastic deformation that should occur along. By default, the axes themselves are orthogonal to the normal and plane of the elements. Morphogen is simulated to diffuse from node to node by solving the heat equation. After a single diffusion time step, a fraction of the morphogen is absorbed to compute the plastic strain. The magnitude and direction of the plastic strain are dependent on the concentration and axis type of the absorbed morphogen. Plastic bending is induced by localizing the plastic strain along the bottom virtual surface of the solid-shells, as opposed to all surfaces for plastic stretching.

From the plastic strain, plastic forces are computed. These forces are derived from the solid-shell formulation and the linear material model. The material can be substituted with another 3D material (e.g. Neo-Hookean and Mooney-Rivlin) if desired. The plastic forces are then used to deform the model's rest configuration using an implicit time integration scheme. The portion of the plastic strain that is not embedded/represented in the Euclidean rest configuration is carried over to the next time step and is regarded as the residual strain. We follow Kennaway et al. [6] in that external forces are absent during growth, allowing the world configuration to simply mirror the rest configuration. But if external forces are desired, elastic forces can be derived by either using the same 3D material that is used to compute the plastic forces or by using a fast linear corotational material model [32]. The elastic forces are passed to the same implicit time integration scheme to solve the world configuration.

Collision handling is carried out to detect and repel apart features that are nearby or colliding. Nearby features are detected in world-space using discrete collision detection and then repelled apart using impulses that are computed according to geometric constraints. Features that are colliding are detected and repelled in the same manner but using continuous collision detection instead. By default, repulsion acts on the world-space but when the world-space is set to mirror the rest-space, the rest-space is repulsed instead.

Adaptive mesh refinement is performed by bisecting, flipping, and collapsing edges. The amount of refinement or coarsening is based on the plastic strain and the following in world-space: edge lengths, curvatures, velocities. Elastic strain is also accounted as needed.

The source code of the growth framework is made public to reproduce the methods and experiments. The growing solid-shells, morphogen diffusion procedure, adaptive remesher, and collision handler are all implemented and

Algorithm 1: Growth Method

```

1 for each time step from  $k = 0$  to  $n$  do
2   Remesh (3.7)
3   Diffuse morphogen (3.3)
4   Convert some morphogen into plastic strain (3.4)
5   Calculate plastic forces from plastic strain (3.5)
6   Advance rest configuration via plastic forces (3.6)
7   Advance world configuration via elastic forces4
8   Perform collision handling (3.8)
9 end

```

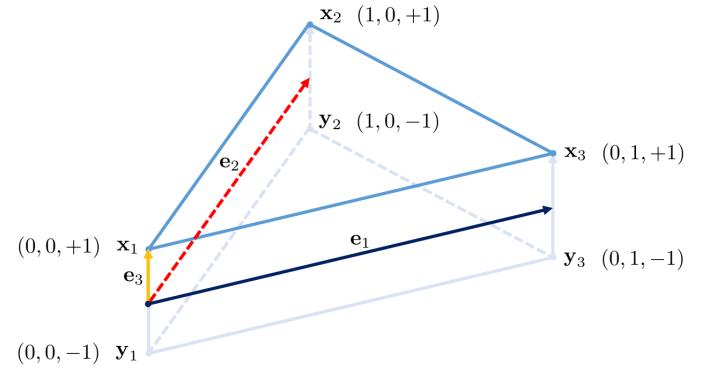


Fig. 2. Schematic Diagram of a Triangular Solid-Shell using a Front and Back Node Implementation. The element is composed of three front nodes (x_1, x_2, x_3) and three virtual back nodes (y_1, y_2, y_3). The virtual back node position can alternatively be specified as $y_a = x_a - d_a$ where d_a is the director vector that extends from the back node y_a to the front node x_a . The axes of the solid-shell domain are labelled as e_1, e_2 , and e_3 . The local coordinates (ξ_1, ξ_2, ξ_3) of each node is provided.

integrated together from the ground-up in the common Java programming language. The experiments can be reproduced by running the framework's provided scripts.

3.2 Solid-Shell Element

The surface of biological tissue can be comprised and modelled using triangular solid-shell elements. The solid-shell is a specialization of the volumetric wedge element where stretching and bending are respectively represented by the translation and rotation of the director vectors. There are two possible director implementations for the triangular solid-shell element. The first implementation uses 3 nodes that reside on the solid-shell's mid-surface where each node specifies the midpoint of its associated director. The second implementation uses 3 front surface nodes and 3 virtual back surface nodes where each front-back node pair represents a single director, as shown in Fig. 2. The front and back nodes of a given director can be imagined as encoding the director's position and orientation respectively. The 3 back nodes are virtual such that they are not visible nor exposed to external forces. We use the second implementation of the solid-shell to handle plasticity more easily and to accommodate potential volumetric element attachments. Formulations follow closely that of FEBio [33, Section 4.2]

The solid-shell is a reduced formulation of the volume element where interpolation along the thickness dimension is simplified to be linear. The solid-shell domain is spanned by 3 local axes, namely e_1, e_2 , and e_3 . The local coordinates of each node is shown in Fig. 2. Given a local coordinate point $\xi = (\xi_1, \xi_2, \xi_3)$ within the domain, it can be mapped to its corresponding global/nodal coordinates x :

$$x(\xi) = \sum_i^3 S_i(\xi) \left(\frac{1 + \xi_3}{2} x_i + \frac{1 - \xi_3}{2} y_i \right) \quad (1)$$

⁴If external forces are absent, then the world configuration does not need to be solved and can mirror the rest configuration.

where \mathbf{x}_i is the i th front node, \mathbf{y}_i is the i th back node, and S_i is the i th 2D shape function which is invariant to the local thickness coordinate ξ_3 :

$$S_1(\xi) = 1 - \xi_1 - \xi_2 \quad S_2(\xi) = \xi_1 \quad S_3(\xi) = \xi_2.$$

For solid-shells, the Jacobian $\partial\mathbf{x}/\partial\xi$ is calculated as a set of 3 covariant basis vectors:

$$\begin{aligned} \mathbf{g}_\alpha(\xi) &= \frac{\partial\mathbf{x}}{\partial\xi_\alpha} = \sum_i^3 \frac{\partial S_i}{\partial\xi_\alpha} \left(\frac{1 + \xi_3}{2} \mathbf{x}_i + \frac{1 - \xi_3}{2} \mathbf{y}_i \right) \\ \mathbf{g}_3(\xi) &= \frac{\partial\mathbf{x}}{\partial\xi_3} = \sum_i^3 \frac{1}{2} S_i(\xi) (\mathbf{x}_i - \mathbf{y}_i) \end{aligned}$$

where $\alpha \in \{1, 2\}$. By decomposing the Jacobian calculation into vectors, the gradient along the thickness (\mathbf{g}_3) can particularly be kept independent of the local thickness coordinate ξ_3 .

The deformation gradient can then be calculated by multiplying the 3×3 matrix of world-space covariant basis column vectors ($\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3$) with the 3×3 matrix of rest-space contravariant basis column vectors ($\mathbf{G}^1, \mathbf{G}^2, \mathbf{G}^3$):

$$\mathbf{F} = \frac{\partial\mathbf{x}}{\partial\mathbf{X}} = \frac{\partial\mathbf{x}}{\partial\xi} \frac{\partial\xi}{\partial\mathbf{X}} = [\mathbf{g}_1 \quad \mathbf{g}_2 \quad \mathbf{g}_3] [\mathbf{G}^1 \quad \mathbf{G}^2 \quad \mathbf{G}^3]$$

where \mathbf{x} and \mathbf{X} refer to the world-space and rest-space nodal coordinates respectively. Afterwards, the deformation gradient can be passed to evaluate the strain and material stress $\boldsymbol{\sigma}$, much like for volume elements.

The basis vectors, particularly the world-space contravariants ($\mathbf{g}^1, \mathbf{g}^2, \mathbf{g}^3$), also serve a purpose of calculating the shape function gradients with respect to the nodal coordinates. The 2D shape function gradient with respect to world-space nodal coordinates are computed as

$$\text{grad } S_i = \frac{\partial S_i}{\partial\mathbf{x}} = \frac{\partial S_i}{\partial\xi} \frac{\partial\xi}{\partial\mathbf{x}} = \frac{\partial S_i}{\partial\xi_\alpha} \mathbf{g}^\alpha.$$

and extended with a linearized thickness dimension to form the 3D shape function gradient:

$$\text{grad} \left(\frac{1 + \xi_3}{2} S_i \right) = \frac{1}{2} ((1 + \xi_3) \text{grad } S_i + S_i \mathbf{g}^3) \quad (2)$$

$$\text{grad} \left(\frac{1 - \xi_3}{2} S_i \right) = \frac{1}{2} ((1 - \xi_3) \text{grad } S_i - S_i \mathbf{g}^3) \quad (3)$$

where equations 2 and 3 are used to express the 3D shape function gradient in context to the front and back nodes respectively.

The force and stiffness equations of volumes and solid-shells are mainly distinguished in terms of their 3D shape function gradient. For reference and derived from the linearization and discretization of internal virtual work [33, Section 3.3], volume elements have the familiar force \mathbf{f} and stiffness matrix \mathbf{K} equations

$$\mathbf{f}_i = \int_v \mathbf{B}_i^T \boldsymbol{\sigma} dv \quad \mathbf{K}_{ij} = \int_v \mathbf{B}_i^T \mathbf{C} \mathbf{B}_j dv$$

where i and j refer to a node pair of the element, v is the element volume, \mathbf{C} is the material elasticity tensor, and \mathbf{B} is the strain-displacement matrix formed by the shape function gradients with respect to the nodal coordinates. In solid-shells, the forces are expressed in a decoupled manner

with \mathbf{f}_i^u and \mathbf{f}_i^w denoting the force of the i th front node and i th back node:

$$\mathbf{f}_i^u = \int_v \boldsymbol{\sigma} \cdot \text{grad} \left(\frac{1 + \xi_3}{2} S_i \right) dv \quad (4)$$

$$\mathbf{f}_i^w = \int_v \boldsymbol{\sigma} \cdot \text{grad} \left(\frac{1 - \xi_3}{2} S_i \right) dv. \quad (5)$$

Continuing the notation where u and w superscripts are indicative of the front and back nodal surface, the solid-shell stiffness is expressed as

$$\mathbf{K}_{ab}^{uu} = \int_v \text{grad} \left(\frac{1 + \xi_3}{2} S_a \right) \cdot \mathbf{C} \cdot \text{grad} \left(\frac{1 + \xi_3}{2} S_b \right) dv$$

$$\mathbf{K}_{ab}^{uw} = \int_v \text{grad} \left(\frac{1 + \xi_3}{2} S_a \right) \cdot \mathbf{C} \cdot \text{grad} \left(\frac{1 - \xi_3}{2} S_b \right) dv$$

$$\mathbf{K}_{ab}^{wu} = \int_v \text{grad} \left(\frac{1 - \xi_3}{2} S_a \right) \cdot \mathbf{C} \cdot \text{grad} \left(\frac{1 + \xi_3}{2} S_b \right) dv$$

$$\mathbf{K}_{ab}^{ww} = \int_v \text{grad} \left(\frac{1 - \xi_3}{2} S_a \right) \cdot \mathbf{C} \cdot \text{grad} \left(\frac{1 - \xi_3}{2} S_b \right) dv$$

where \mathbf{K}_{ab}^{uw} , for instance, denotes the stiffness between the a th front node and b th back node.

The integral expressions in the force and stiffness equations can be numerically evaluated using the quadrature rule. Given an integral expression, its integrand is evaluated at predefined local coordinate points (referred to as *integration points*) within the element's volume. A set of 9 integration points are used for solid-shells, each comprised of 3 local coordinates and a weight (ξ_1, ξ_2, ξ_3, w) :

$$\mathbf{q}_1 = (a, b, -b, ac_1) \quad \mathbf{q}_4 = (a, a, 0, ac_2) \quad \mathbf{q}_7 = (a, a, b, ac_1)$$

$$\mathbf{q}_2 = (b, a, -b, ac_1) \quad \mathbf{q}_5 = (b, a, 0, ac_2) \quad \mathbf{q}_8 = (b, a, b, ac_1)$$

$$\mathbf{q}_3 = (a, b, -b, ac_1) \quad \mathbf{q}_6 = (a, b, 0, ac_2) \quad \mathbf{q}_9 = (a, b, b, ac_1)$$

where $a = 1/6$, $b = 2/3$, $c_1 = 5/9$, and $c_2 = 8/9$. This integration scheme is a hybrid of 3-point triangular quadrature across the membrane [34, Table 10.3] with one-dimensional 3-point quadrature across the thickness [33, Section 4.2.1.1].

Similar to thin-shells, a solid-shell model can be treated as a surface mesh (i.e. 2D manifold). The vertices, edges, and faces of the surface mesh correspond to the front nodes, front edges, and front triangles of the solid-shells. Virtual back nodes are accessed by referring to their corresponding front node. This 2D abstraction simplifies the diffusion, remeshing, and collision detection process.

3.3 Morphogen

The chemicals that initiate growth can be simulated to diffuse across a surface mesh by solving the heat equation:

$$\dot{c} = \Delta c(x, t)$$

where Δ is the Laplace operator and $c(x, t)$ is the concentration of an arbitrary substance at position x and time t . Solving this heat flow expression over some period of time would yield the change in concentration at each fixed position x .

The mass matrix \mathbf{A} and discrete cotangent Laplacian matrix \mathbf{L} that are required to solve the heat equation are simple to calculate. Suppose that a triangular mesh containing $|V|$

vertices is given, representing the front surface of the solid-shell model. The mass matrix \mathbf{A} is a $|V| \times |V|$ diagonal matrix where the i th diagonal entry contains one-third of the summed area of triangles that are incident to vertex i . The cotangent Laplacian matrix \mathbf{L} is a $|V| \times |V|$ matrix and is constructed as follows:

$$(\mathbf{L})_{ij} \approx \begin{cases} \frac{1}{2}[\cot \alpha + \cot \beta] & \text{if edge } i-j \text{ exists} \\ -\sum_n (\mathbf{L})_{in} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

where \sum_n is the summation over each neighboring vertex n that shares an edge with vertex i ; and α and β are the angles that are on opposite sides of edge $i-j$.

When given the initial concentration at each vertex i , their final concentrations after a time step of size Δt can be computed by solving for \mathbf{c}_1 in the linear system [26]:

$$(\mathbf{A} - \Delta t \mathbf{D} \mathbf{L}) \mathbf{c}_1 = \mathbf{A} \mathbf{c}_0$$

where \mathbf{D} is the diffusion coefficient and \mathbf{c}_0 is the $|V|$ -sized vector containing the initial concentration at each vertex. Solving for \mathbf{c}_1 is done using any direct sparse linear solver.

We make the assumption that the shell thickness is small enough relative to the width and length of the model such that the concentration gradient across the thickness is negligible. Consequently, diffusion is simulated across the solid-shell front surface only. However, if extra precision is needed, both front and back vertices of the solid-shell can be accounted for when simulating diffusion.

3.4 Growth Tensor

As morphogen diffuses throughout the front nodes, each front node will convert a fraction of its morphogen into a growth tensor. The growth tensor itself is a specification of the growth to occur and is formulated as a function of the morphogen and axes of growth present [6]. There are six varieties of morphogen that vary in terms of *motion type* and *axis type*. The motion type can either be *stretching* or *bending*. The direction that stretching or bending occurs is dependent on the morphogen's axis type, which are historically referred to as *PAR* (as in parallel), *PER* (as in perpendicular), or *NOR* (as in normal). By default, each element is assigned a set of orthogonal unit vectors in rest-space, namely \mathbf{d}_{PAR} , \mathbf{d}_{PER} , and \mathbf{d}_{NOR} which altogether define the 3 axes of growth that the PAR, PER, and NOR morphogen can induce growth towards respectively. For instance, if a sheet of elements is subjected to stretching PAR morphogen, the elements would elongate along their \mathbf{d}_{PAR} axes. If instead subjected to bending PAR morphogen, the elements would curl around their PAR axes, forming a cylinder sheet if \mathbf{d}_{PAR} is common across all elements. Altogether, the growth tensor encapsulates the magnitude and direction of stretching or bending that is to occur according to the six morphogen varieties present and the element's axes of growth.

Before the growth tensor is inferred from the presence of morphogen, at every time step, the \mathbf{d}_{PAR} , \mathbf{d}_{PER} , and \mathbf{d}_{NOR} growth axes of each element are prescribed. For a given element in rest-space, \mathbf{d}_{PAR} and \mathbf{d}_{PER} lie on the element's front-surface plane while \mathbf{d}_{NOR} is set to the element's normal. Assuming orthogonality, \mathbf{d}_{PER} is inferred

from the cross-product of \mathbf{d}_{PAR} and \mathbf{d}_{NOR} which leaves \mathbf{d}_{PAR} needing to be determined. One proposed mechanism suggests that cells can orient their growth according to the gradient of a substance referred to as *POL* (as in polarity) [6]. The POL gradient can be represented in a discrete matter where each node is assigned a scalar POL concentration, allowing \mathbf{d}_{PAR} to be calculated on a per-element basis. More specifically, given a triangle solid-shell element with 3 front-node positions in rest-space (\mathbf{X}_1 , \mathbf{X}_2 , \mathbf{X}_3) and 3 respective nodal POL concentrations (p_1 , p_2 , p_3), the element's \mathbf{d}_{PAR} can be calculated according its local POL gradient as follows:

$$\begin{aligned} \mathbf{X}_{21} &= \mathbf{X}_2 - \mathbf{X}_1 & p_{21} &= p_2 - p_1 \\ \mathbf{X}_{31} &= \mathbf{X}_3 - \mathbf{X}_1 & p_{31} &= p_3 - p_1 \\ \bar{\mathbf{X}} &= \begin{bmatrix} \mathbf{X}_{31} \cdot \mathbf{X}_{31} & -\mathbf{X}_{21} \cdot \mathbf{X}_{31} \\ -\mathbf{X}_{21} \cdot \mathbf{X}_{31} & \mathbf{X}_{21} \cdot \mathbf{X}_{21} \end{bmatrix} & \mathbf{p} &= \begin{bmatrix} p_{21} \\ p_{31} \end{bmatrix} \\ \mathbf{u} &= (1/\det(\bar{\mathbf{X}})) \bar{\mathbf{X}} \mathbf{p} & \mathbf{d}_{\text{PAR}} &= \mathbf{u}_1 \mathbf{X}_{21} + \mathbf{u}_2 \mathbf{X}_{31}. \end{aligned}$$

Once \mathbf{d}_{PAR} , \mathbf{d}_{PER} , and \mathbf{d}_{NOR} have been determined and normalized into unit vectors, the *element growth frame* \mathbf{Z} can be created, which encapsulates the 3 growth axes as column vectors in a single 3×3 matrix:

$$\mathbf{Z} = [\mathbf{d}_{\text{PAR}} \quad \mathbf{d}_{\text{PER}} \quad \mathbf{d}_{\text{NOR}}]$$

Complementing the growth frame \mathbf{Z} is the *element growth tensor* \mathbf{E} which represents the magnitude of growth. Assuming morphogens of stretching type for now as an example, let \mathbf{c}_a refer to the $|V|$ -sized vector of nodal morphogen concentrations where a is the morphogen axis type and $|V|$ is the number of front nodes of the solid-shell model. As diffusion occurs, at every time step, each node p subtracts a fixed percentage (i.e. $s = 0.1\%$) from its current morphogen concentration $(\mathbf{c}_a)_p$. The amount of stretching morphogen subtracted is recorded on an element basis: each element has a 3×3 element growth tensor \mathbf{E} where \mathbf{E}_{na} refers to the amount of stretching morphogen of axis type a that is subtracted (i.e. absorbed) from the element's n th front node in the current time step, altogether expressed as $\mathbf{E}_{na} = s(\mathbf{c}_a)_p$ where p is the global index of the element's n th front node. Afterwards, the element growth tensor \mathbf{E} is rotated to align with the element's growth frame \mathbf{Z} , forming \mathbf{E}' :

$$\mathbf{E}' = \mathbf{Z} \mathbf{E} \mathbf{Z}^T.$$

The rotated element growth tensor \mathbf{E}' of assumed stretching morphogen is then used to derive the *integration growth tensor* \mathbf{G} . The intention is to transform a given rotated element growth tensor such that the number of rows corresponds to the number of integration points instead of the number of front nodes. More specifically, the amount of morphogen absorbed in each integration point is to be inferred from the element's front nodes. This transformation can be done using an extrapolation matrix. Specifically, we use an $r \times q$ matrix \mathbf{Q} where r and q are respectively the number of front-nodes and integration points of the element. \mathbf{Q}_{ni} is equal to the n th front node's 2D shape function $S_n(\xi)$ that is evaluated at the local coordinates ξ of the i th integration point. 2D shape functions are invariant to depth (i.e. 3rd local coordinate); integration points that have the same 1st

and 2nd local coordinates will receive the same extrapolated morphogen absorbed.

This extrapolation matrix will only change when given an rotated element growth tensor that is derived from bending morphogen instead of the assumed stretching morphogen. The only difference is that when bending morphogen is being extrapolated, the extrapolation matrix \mathbf{Q} is set to zero except for the first 3 columns. In effect, this conveniently allows only the 3 integration points that reside along the virtual bottom surface to receive the extrapolated morphogen. That way, the virtual bottom nodes, which handle bending dynamics, will subsequently be pushed apart, causing upward bending to occur. It is worth noting that negative concentrations can be introduced to shrink the virtual bottom surface instead and induce downward bending.

Regardless how the extrapolation matrix \mathbf{Q} is set, it is multiplied with the given rotated element growth tensor \mathbf{E}' to yield the integration growth tensor \mathbf{G} :

$$\mathbf{G}^\top = \mathbf{E}'^\top \mathbf{Q}.$$

The $q \times 3$ integration growth tensor \mathbf{G} specifies the magnitude and direction of growth to occur at each integration point of the element. Specifically, the i th row of the integration growth tensor \mathbf{G} is regarded as the 3 diagonal entries of the i th integration point's 3×3 *plastic strain* tensor $\epsilon_{i,\text{plastic}}$ with off-diagonal entries set to zero:

$$(\epsilon_{i,\text{plastic}})_{ab} = \begin{cases} \mathbf{G}_{ia} & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Altogether, the growth tensor formulation can be viewed as taking the given diffused morphogen and POL state and mapping it to corresponding plastic strain tensors. This procedure has no assumption on the time step size, mesh topology, and other simulation parameters; the tensors only reflect the amount and direction of growth that is specified to occur. Nevertheless, the smoothness of diffusion, which the growth tensors depend on, and the resulting forces and velocities of the growth tensors can vary depending on these simulation parameters. Therefore, with any FEM simulation, the mesh resolution and time step should be kept at a high enough precision in order converge into an accurate morphology.

3.5 Plastic Forces

In order to induce plastic deformations, the *plastic forces*, which are forces associated with the plastic strain, need to be derived. The Cauchy stress tensor is first computed using the plastic strain:

$$\boldsymbol{\sigma}_{i,\text{plastic}} = \mathbf{C} \cdot \epsilon_{i,\text{plastic}}$$

where \mathbf{C} is the linear material elasticity tensor. The plastic forces are then formulated exactly as the elastic forces of the solid-shell (equations 4 and 5) except that the stress tensor $\boldsymbol{\sigma}$ is substituted with $\boldsymbol{\sigma}_{i,\text{plastic}}$.

3.6 Plastic Embedding

The key idea to plastic embedding is that the rest configuration of the model is allowed to update according to the plastic strain. Wicke et al. updates the rest configuration by searching for the rest configuration that minimizes the plastic strain energy. This search is handled as an optimization problem using a simple quasi-Newton method. While this optimization scheme can minimize the plastic strain, there are trade-offs where the plastic strain is assumed small enough to be minimized in one solve, and that factors, such as masses and material stiffness, are assumed negligible. To retrain these factors, we adhere to the backward Euler time stepping scheme to advance the rest configuration by solving the velocities \mathbf{v}^{k+1} that corresponds to the forces of the plastic strain [35]:

$$\left(\mathbf{M} - \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{v}} - (\Delta t)^2 \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right) \mathbf{v}^{k+1} = \mathbf{M}\mathbf{v}^k - \Delta t \frac{\partial \mathbf{f}}{\partial \mathbf{v}}\mathbf{v}^k + \Delta t \mathbf{f}^k \quad (7)$$

where k is the current time step; \mathbf{M} is the mass matrix; derivatives $\partial \mathbf{f} / \partial \mathbf{v}$ and $\partial \mathbf{f} / \partial \mathbf{x}$ correspond to the damping matrix and global stiffness matrix; and \mathbf{f} are the plastic forces present in the front and back nodes.

The solved velocities \mathbf{v}^{k+1} are then used to displace the rest configuration from time step k to $k+1$:

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \Delta t \mathbf{v}^{k+1}$$

where \mathbf{X} is the front and back node positions of the rest configuration. By advancing the rest configuration, the stress-free shape becomes modified permanently, thus simulating plastic deformation.

When the rest configuration advances at each time step, the plastic strain tensor at each integration point decrements accordingly to reflect the remaining amount of plastic deformation to occur. Using the division operator (/) to denote that the left operand is multiplied with the inverse of the right operand, the decrement procedure is formulated as

$$\epsilon_{i,\text{plastic}}^{k+1} = (\epsilon_{i,\text{plastic}}^k + \mathbf{I}) / \hat{\mathbf{R}}_i - \mathbf{I}. \quad (8)$$

$\epsilon_{i,\text{plastic}}^k$ is the plastic strain tensor at the current time step k , specifying the expected amount of plastic deformation to occur at the i th integration point. \mathbf{I} is the 3×3 identity matrix. $\hat{\mathbf{R}}_i$ is the rest-space *incremental deformation gradient* [36], which is the deformation gradient \mathbf{F} of the updated rest configuration at time step $k+1$ with respect to the previous time step k , computed at the i th integration point:

$$\mathbf{R} = \mathbf{F}^{k+1} / \mathbf{F}^k \quad (9)$$

In equation 8, the hat symbol $\hat{\cdot}$ on \mathbf{R} indicates that the symmetrical factor of \mathbf{R} is to be used; we are only interested in the non-rotation changes of the element between the two time steps [37]. The symmetrical factor is obtained using right-polar decomposition. The resulting $\epsilon_{i,\text{plastic}}^{k+1}$ is the remaining (i.e. residual) plastic strain that will be carried over to the next time step $k+1$. When carried over, this residual plastic strain is additively incremented with the next incoming plastic strain (equation 6).

Afterwards, the masses of the individual solid-shell elements are updated to sync with the grown rest configuration. The masses are directly related to the density and rest volume of the solid-shells.

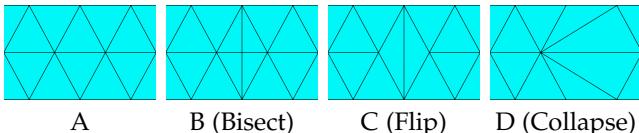


Fig. 3. Remeshing Operations. Using the center edge in (A) as a starting reference, the effect of the three operations on the edge are shown in (B), (C), and (D) respectively. The bisect operation (B) splits the center edge while creating a new middle vertex that is connected to all four of its adjacent vertices. The flip operation (C) replaces the center edge with a new edge that is connected to the other two adjacent vertices. The collapse operation (D) collapses one of the center edge's vertices (right vertex in this case) into the other center edge vertex (left vertex in this case). If the mesh symmetry needs to be preserved, the collapse operation can be performed by collapsing the two vertices of the center edge together into a new midpoint vertex.

We emphasize that the plastic embedding procedure does not require distinguishing between stretching and bending. This is due to the design of the solid-shell where stretching and bending can both commonly be described in terms of stretching strain.

3.7 Adaptive Remeshing

Periodically during growth simulation, the model is remeshed to refine curved, dynamic, and compressed surfaces with enough resolution to permit surface buckling formation and retain geometric detail. At the same time, flat, idle, and relaxed surfaces are coarsen to save computation time. Our implementation is adapted from Narain et al. [13].

In the beginning of each remesh iteration, the *sizing field* of each element is measured [13, Section 4]. The sizing field captures 3 metrics that judge the amount of refinement needed at the element. The 3 metrics are curvature M_{crv} , velocity gradient M_{vel} , and buckling anticipation M_{buc} . The curvature and velocity gradient metrics are respectively dependent on the surface normals and velocities in world-space. For growth, the buckling anticipation metric is modified to be dependent on elastic and plastic strain instead of the compressive membrane strain ϵ that is meant for paper simulation. Consequently, ϵ is substituted with

$$\epsilon = -\text{abs}(\hat{\epsilon}_{\text{elastic}} \hat{\epsilon}_{\text{plastic}}).$$

Because the buckling tendency metric is calculated on an element basis, the elastic strain $\epsilon_{\text{elastic}}$ and plastic strain $\epsilon_{\text{plastic}}$ are calculated into their element-based forms by averaging their strain across the integration points of the element. We discard the rotational factor, as indicated by the hat symbol $\hat{\cdot}$, of the elastic and plastic strain in order to multiply the strains together in a common rotation-free space and that we are only concerned about the magnitude of strain present. The $\text{abs}(\cdot)$ function returns the same tensor but each element of the tensor is replaced with its absolute value. The reasoning for taking the element-wise absolute is to make the multiplied strain invariant to either compression and stretching — both morphogen-induced stretching and resultant compression are indicators of imminent buckling. A negation is added at the end in order to comply to how the metric is originally formulated: a negative strain tensor would indicate compression, which in turn, indicate needed refinement.

The metrics are summed together to generate a sizing field for each element. The sizing fields are adjusted to target a specific mesh resolution. Targeting a given mesh resolution is done by factoring the sizing field matrices into eigenvectors and eigenvalues, clamping the eigenvalues to conform to a specific minimum and maximum edge length in world-space, and then reassembling the sizing field matrices. These matrices are then interpolated and transformed to yield a scalar *size* for each edge in order to simplify the remesher logic.

At this point, the surface mesh can be remeshed. The mesh is first refined by bisecting edges that have a size greater than 1. Afterwards, edges that can be collapsed without introducing new edges of size greater than 1 are iteratively searched and collapsed until no such edge remains. When refining and collapsing edges, flippable edges are also searched and flipped in order to improve the aspect ratio of the faces. The edge bisect, collapse, and flip operators are visualized in Fig. 3 with pseudocode presented in [23, Algorithm 1].

Among these three remeshing operators, only the edge split requires accounting for the presence of morphogen and POL. Whenever an edge split occurs, the new node's morphogen and POL is determined from the average of its parent nodes. Edge flips and collapses require no such update because no node is created nor displaced. We follow the assumption that the morphogen and POL gradient will be smooth enough such that morphogen and POL mass will be preserved if a node is removed during an edge collapse.

The remesher is modified to resample the plastic strain after remeshing. In other words, before and after remeshing, the plastic strain across the model should be close to invariant in terms of its magnitude, direction, and locality. Resampling is carried out using a nearest neighbor interpolation scheme. First, integration points of the pre-remeshed model are saved. For a given integration point of the newly-remeshed model, the 6 integration points of the pre-remeshed model that are nearest are searched by traversing across neighboring triangles. Once the 6 points are found, their associated plastic strain matrices undergo an inverse distance weighting procedure to yield the plastic strain tensor $\epsilon_{\text{plastic},i}$ for the given integration point i of the new mesh:

$$\epsilon_{\text{plastic},i} = \frac{1}{\sum_j \text{dist}(i,j)^{-1}} \sum_j \text{dist}(i,j)^{-1} \epsilon_{\text{plastic},j}$$

where j denotes one of the 6 integration points of the pre-remeshed model that are closest to integration point i of the newly-remeshed model, and $\text{dist}(i,j)$ is the distance between integration point i and j in world-space. Because plastic strain is minimized due to plastic embedding, numerical errors associated with repeated resampling are minimized as well.

3.8 Collision Handling

To prevent the growing models from intersecting, the simulations are coupled with the discrete and continuous collision handling scheme adopted from Bridson et al. [31]. Letting *feature* denote either a vertex, edge, or triangle of a mesh, nearby feature pairs in world-space are found using

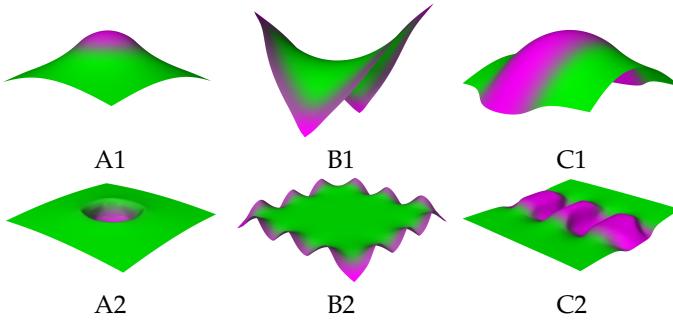


Fig. 4. Basic Grown Shapes. The bulge (A1), saddle (B1), and arc ridge (C1) are produced by respectively placing isotropic stretching morphogen (denoted in purple) in the center, boundary, and middle strip of the sheet. When the bending stiffness is lowered, the steep crater (A2), ripples (B2), and grooves (C2) are respectively produced instead.

axis-aligned bounding-boxes (AABB) at a single time step. Instead of repelling the detected pairs using spring forces, we use impulses determined by the LCP solver that is provided by the FEM framework [35]. The LCP solver is fed using unilateral constraints that are calculated based on the detected front features of the solid-shell model that are nearby. The unilateral constraints are adept in accounting for negative *penetration tolerance* where a minimum distance is respected between nearby elements, preventing interpenetration artifacts with the virtual back layer. When applying the solved impulses to the front nodes, the virtual back nodes also receive the same impulses. These impulses act in world-space unless the rest configuration is only simulated in which impulses act in rest-space instead. Collision handling in rest-space is only valid when 100% plastic yielding is assumed (i.e. collisions permanently deform the model).

Subsequently, for the feature pairs that are actually colliding, they are found using AABBs, but the search is performed continuously such that the space-time path of each feature between the previous and current time step is accounted in order to detect collisions reliably. Afterwards, the colliding pairs are resolved using impulses. Any new (i.e. secondary) collisions that occur after applying the impulses are repeatedly detected and resolved until all collisions are eliminated.

We adopt the approach where unilateral constraints of the colliding features are grouped into *impact zones*, which are resolved one at a time [38]. This approach is reserved as a “fail-safe” whenever repeated attempts of resolving collisions using impulses would always lead into secondary collisions.

Velocity correction is performed by constraining the subsequent backward Euler solve (equation 7) using the same unilateral constraints that were used to resolve the nearby feature pairs and primary (i.e. non-secondary) collision pairs. To account for remeshing, the unilateral constraints that are used to constrain the backward Euler solve are interpolated on a node basis whenever remeshing occurs.

4 RESULTS AND DISCUSSION

4.1 Basic Shapes

Before growing elaborate models, we simulate the growth of basic shapes to verify that the buckling patterns are

consistent with previously published simulation results. These basic shapes are shown in Fig. 4. The resulting variety in shapes are attributed to the different combinations of bending stiffness and locality of the isotropic (PAR and PER) stretching morphogen applied. The starting sheet has a linear POL gradient such that all elements have their d_{PAR} axes initialized to be equal to each other and parallel to the sheet’s width.

The shapes in the top row of Fig. 4 were grown using a relatively higher bending stiffness material than those in the bottom row. Material with high bending stiffness has the tendency of producing shapes and patterns with lower curvature. A prime example is the bulge shape (A1), which is produced by applying morphogen in the center of a sheet with such material. The bulge shape produced is comparable with existing experiments [4, Fig. 7.3] [6, Fig. 3]. When morphogen is applied on the sheet perimeter, a saddle (B1) is produced [4, Fig. 6.3a]. The arc ridge (C1), although an uncommon shape to compare, can be seen as a variation of the bulge experiment (A1) where the bulge is instead elongated along the strip of morphogen, forming a ridge.

When the bending stiffness is lowered, the shapes and patterns exhibit higher curvature. In other words, the energy associated with bending becomes lower, permitting bends to occur more freely in the minimal energy shape. This can be observed in the ripples (B2) and grooves (C2) shapes, which are consistent with previous experiments [4, Fig. 6.3b and 6.6]. The crater shape (A2) has subtleties that are consistent with the effects of lower bending energy: bends along the ridges are steeper and small faint buckled protrusions appear along its sides.

4.2 Ripple Cascade Demo

Growing tissue can develop fractal patterns along their edges. The fractal patterns that appear in the kale leaf (*Brassica oleracea*) are a notable example and have been modelled using physics-based approaches. Vetter et al. observed that when a flat sheet composed of thin-shells is subjected to anisotropic plastic strain along one of its edge, the edge can buckle into a cascade of ripples [22, Fig. 11].

We can perform the same ripple cascade experiment using our augmented solid-shells. The augmented solid-shells particularly builds upon Vetter et al.’s experiment by introducing plastic embedding and adaptive remeshing rather than the multiplicative decomposition approach and use of pre-refined meshes. Fig. 5 provides a time-lapse of the ripple cascade experiment using augmented solid-shells. In this experiment, anisotropic (PAR) stretching morphogen is applied along one side of the solid-shell sheet. A linear POL gradient is also applied in this experiment: the d_{PAR} axis of every element is initialized to be equal to each other and parallel to the sheet width. The sheet has a length-thickness ratio of 2000:1, Young’s Modulus of 10^4 , and started with 256 elements. When enough strain accumulates, buckling occurs to form the initial set of ripples (B). As morphogen continues to be supplied, (C) highlights the early formation of secondary ripples that occur on the initial set of ripples, which later become more prominent in (D).

One feature of our ripple cascade experiment compared to Vetter et al.’s experiment is the formation of larger ripples.

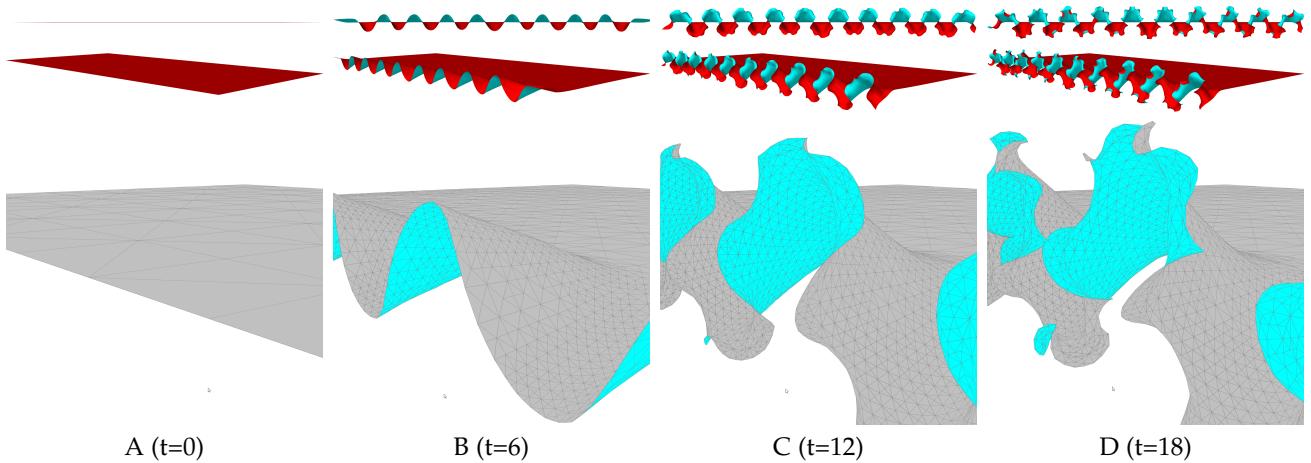


Fig. 5. Time-Lapse Growth of a Ripple Cascade. The front view (top row), oblique view (middle row), and a close-up view of the shape's mesh topology (bottom row) are shown. The ripple cascade is generated by diffusing anisotropic stretching morphogen along the front side of the sheet.

The larger ripples are attributed to plastic embedding and the adaptive remesher. Plastic embedding actively transfers the plastic strain into the rest configuration, essentially minimizing the plastic strain in order to prevent large and unstable plastic forces. Alongside, the adaptive remesher helps ensure that the aspect ratio of each element remains in balance during growth, reducing the risk of ill-conditioned elements. The balanced elements are particularly noticeable in the bottom row of Fig. 5.D despite prolonged growth.

The inclusion of adaptive remeshing has an additional advantage of conducting growth experiments without needing to preemptively predict the location and magnitude of mesh refinement needed. Vetter et al. relied on the use of *a posteriori* refined meshes [22, Fig.10], which can be time-consuming to construct. With adaptive remeshing, highly-detailed patterns can still emerge from a coarse mesh that automatically refines itself where needed.

4.3 Delicate Wrinkle Pattern Demo

There is recent research on simulating the formation of delicate wrinkle patterns on thin tissue. Simulating these patterns requires elements that can handle steep bending deformations and prolonged growth. Consequently, thin-shells with deformable rest configurations have become a common choice. Particularly, hinge-based thin-shells with plastic embedding can be used to experimentally generate such patterns [12]. Another work used non-Euclidean plates to generate delicate wrinkle patterns by gradually scaling the reference metric tensor of each triangle [39].

The formation of delicate wrinkles can also be achieved using solid-shells with plastic embedding. In contrast to existing approaches for creating wrinkles, we include a remeshing component in our augmented solid-shells. The remesher can adaptively refine the mesh where wrinkling is anticipated. The experiment is shown in Fig. 6, which displays the time-lapse growth of a pattern that resembles a reaction-diffusion pattern. The pattern was generated by applying diffusible isotropic stretching morphogen (from $t = 0$ to $t = 2$) in the center of the sheet. Likewise in the previous experiments, a linear POL gradient is applied to the entire sheet. The bending stiffness of the sheet was set

to a lower extreme (length-thickness ratio of 10000:1 and Young's Modulus of 100) to accommodate the formation of delicate wrinkles. In Fig. 6, the sheet (A) is initially flat with a uniform mesh resolution of only 200 elements. After 1 second (B), the sheet remains flat but the remesher anticipates buckling by preemptively increasing the mesh resolution in the center. After 2 seconds (C), buckling occurs, forming the initial wrinkles. The mesh resolution near the center further increases in resolution to respect the curvature metric and increasing strain. At 3 seconds (D), the mesh resolution adapts again to accommodate the more propagated and detailed wrinkle pattern. At the same time, the boundary of the mesh is hardly refined, as shown in the bottom-left of (D). Overall, this experiment showcases the ability of the augmented solid-shells in handling a growth scenario that would normally be simulated using thin-shells with deformable rest configurations.

4.4 Fruit-like Shape Demo

Beyond delicate wrinkles and ripple cascades, mechanical stresses can contribute to the formation of features at macroscopic scale. In particular, the overall shapes found in many fruits and vegetables have been suggested to be the result of stress-induced buckling [16]. This suggestion can be supported using simulation, as shown in Fig. 7 where a fruit-like model is grown by applying PAR and PER stretching morphogen uniformly across a starting regular icosahedron sphere composed of solid-shells with a Young's Modulus set to 10^6 . The solid-shells are set to be very thick (thickness to sphere radius ratio of 1:5) to model the fleshy interior that resides behind a fruit's surface.

In contrast to previous demos, d_{PAR} of each element is fixed to the global rest-space z-axis: regardless of a given element's orientation, the element is specified to stretch along the global z-axis. In other words, for elements whose normals are parallel to the global z-axis, their thickness will elongate when subjected to PAR morphogen. In turn, PER stretching morphogen will have no effect because the cross product of the fixed d_{PAR} and element's normal will be zero. For elements whose normals are perpendicular to the global z-axis, d_{PAR} and d_{PER} of these elements will

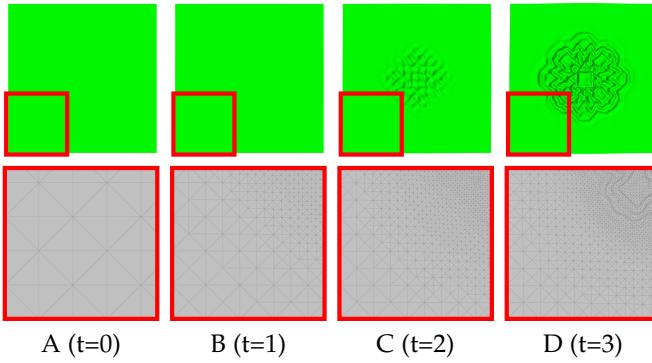


Fig. 6. Time-Lapse Growth of a Delicate Wrinkle Pattern. Each rendered image (top row) is accompanied by a close-up view of its mesh topology (bottom row). The pattern is generated by supplying diffusible isotropic stretching morphogen to the center of a very thin and flexible sheet.

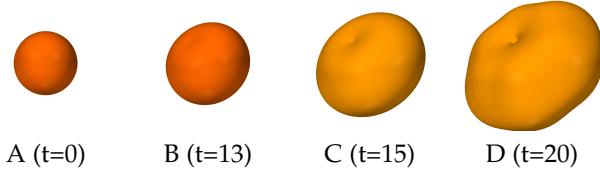


Fig. 7. Time-Lapse growth of a Fruit-like Shape. A constant PAR and PER stretching morphogen concentration is applied on a regular icosahedron where d_{PAR} is held constant to the global rest-space z-axis.

simply be orthogonal and tangent to their surface planes. This fixed d_{PAR} scheme creates a dual growth rule where top and bottom elements of the sphere are biased to elongate their thickness while elements near the sphere's horizontal circumference are biased to isotropically stretch along their planes.

Across the growth time-lapse in Fig. 7, inflation occurs predominantly along the horizontal circumference with the two z-axis polar ends developing navels, altogether forming a biconcave ellipsoid. The biconcave ellipsoid is known to coincide with the minimum energy state of a sphere growing towards a disc [40]. For a given polar end in Fig. 7.D, there is a particularity where the center steep navel is nested within a shallower concavity, likely attributed to the dual combination of the increased thickness and lack of planar strain to resist the surrounding planar stress. Overall, the experiment showcases the solid-shell's ability to accommodate considerable thickness while being capable to alter both its planar and normal topology.

4.5 Curling Sheet Demo

There are existing approaches on simulating plastic bending. A recent approach is to use an extension of non-

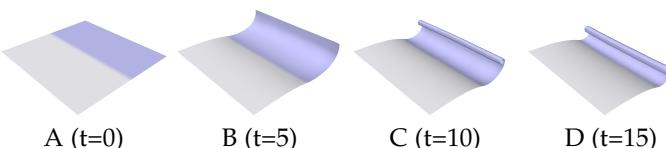


Fig. 8. Time-Lapse Simulation of Plastic Bending. Morphogen is placed on one-half of the sheet to anisotropically shrink and expand the front and virtual back surface respectively of the solid-shells, producing an exaggerated curling effect.

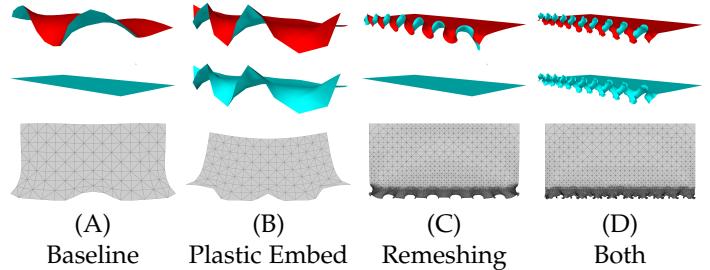


Fig. 9. Ablation of Plastic Embedding and Remeshing. Experiments include: (A) neither plastic embedding nor remeshing, (B) plastic embedding alone, (C) remeshing alone, and (D) both plastic embedding and remeshing present. For each experiment, the world-space shape (red), rest-space shape (cyan), and rest-space topology (grey) of the sheet is shown. Growth is induced by applying anisotropic stretching morphogen along one side of the sheet.

Euclidean plates that supports adjustable rest curvatures. These extended non-Euclidean plates are capable of simulating the curling of paper [21]. For solid-shells, a bilayer of solid-shells can be used where the bottom layer is only subjected to stretching growth [18]. By growing only the bottom layer, authors were able to make a flat star-like structure curl upwards into a sphere-like structure.

In this experiment, we demonstrate large plastic bending using a single layer of solid-shells by leveraging on our bending morphogen. Fig. 8 provides a time-lapse of the experiment. The starting square sheet (A) has a length-thickness ratio of 100:1, a Young's modulus of 10^5 , and a linear POL gradient. On one-half of the sheet, anisotropic (PER) bending morphogen is placed. To produce a more exaggerated curling effect, the same half of the sheet has morphogen that induces anisotropic (PER) shrinking. The initial effect of the morphogen is shown in (B) where one side of the sheet begins curling. As morphogen continues to be absorbed, (C) and (D) shows the curling curvature becoming more pronounced, causing the sheet to roll up into itself. To prevent the curling sheet from intersecting itself, collision handling is enabled in the simulation.

While thin-shells are known for handling plastic bending, this experiments reveals that solid-shells can also be used. In comparison to the curling paper experiments done using non-Euclidean plates [21], we demonstrate curling curvature of similar magnitude with the addition of adaptive remeshing and collision handling working alongside. The experiment also demonstrates that it is possible to simulate plastic bending using a single layer of solid-shells.

4.6 Remesher and Plastic Embedding Ablation

Ablation tests are performed to observe the effect of plastic embedding and adaptive remeshing on the formation of detailed shapes. In each ablation test, morphogen is applied to one side of the sheet, identical to how the ripple cascade experiment was set up. However, the ablation tests attempt to simulate growth with plastic embedding and remeshing both disabled, or only one of the two enabled, or both enabled. Tests without plastic embedding fallback to the multiplicative decomposition approach to model plasticity. The four ablation tests are shown in Fig. 9, which showcases the resulting world-space shape (red), rest-space shape (cyan), and rest-space topology (grey) for each test.

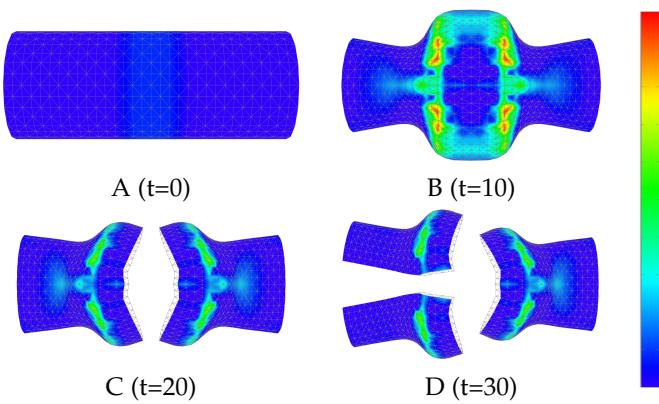


Fig. 10. Build-Up and Release of Residual Stress. (A) depicts the starting cylinder model that is subjected to growth for 10 seconds (B). (C) is the model after being vertically cut in half and allowed to relax for 10 seconds. (D) shows the left half of the model after being horizontally cut in half and relaxed for 10 seconds. The colormap denotes the magnitude of residual stress from low (blue) to high (red).

The morphology difference between the baseline (A) and plastic embedded (B) reveals a stiffening effect that accumulated stress can exhibit. Particularly in artery tissue, stress that cannot be relieved (i.e. residual stress) is a suggested mechanism that optimizes the tissue's load-bearing shape [41]. This load-bearing shape is reflected in the baseline (A) where stress is strictly accumulated, forming much fewer ripples. In other tissue, including plant tissue, there exists positive feedback mechanisms where stress is actively absorbed [6]. The absorption is demonstrated by plastic embedding (B) where stress actively dissipates into the rest-space to form a surface with less tension. When under less tension, the surface becomes more malleable to buckle with increased ripple frequency compared to the baseline (A).

If remeshing is enabled (C) instead of plastic embedding (B), the arcs occur at even greater frequency. This more complex morphology indicates the importance of maintaining a high resolution mesh, which can otherwise prevent the formation of smaller and more detailed shapes. When both plastic embedding and remeshing are enabled (D), the effect of both are combined, allowing arcs to occur within arcs, forming the elaborate ripple cascade.

4.7 Tissue Incision and Residual Stress Demo

While plastic embedding allows stress to actively dissipate, there can be stress that cannot be relieved because the grown rest configuration cannot be represented in Euclidean space. This residual stress is often a side-effect that accompanies large growth deformations. In this experiment, we demonstrate residual stress by showcasing its build-up and release, as shown in Fig. 10. The cylinder model (A) is first grown in order to accumulate residual stress (B). The grown model is then vertically incised in half (C). By incising the model, the area along the incision becomes less physically constrained and therefore permitted to expand into a lower energy shape. The release of residual stress is evident by the new shape (C) that the model takes on after incision. When the second incision is performed (D), the model again takes on a new shape. The figure provides a color-map of the residual stress, which dissipates after

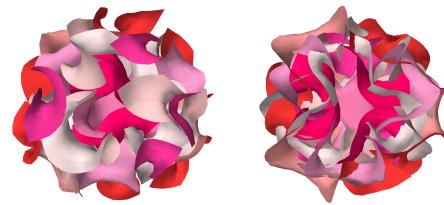


Fig. 11. Morphology Influence of Collision Handling. The left-side shows the same ripple bouquet model from Fig. 1.D, while the right-side shows the same model but without collision handling.

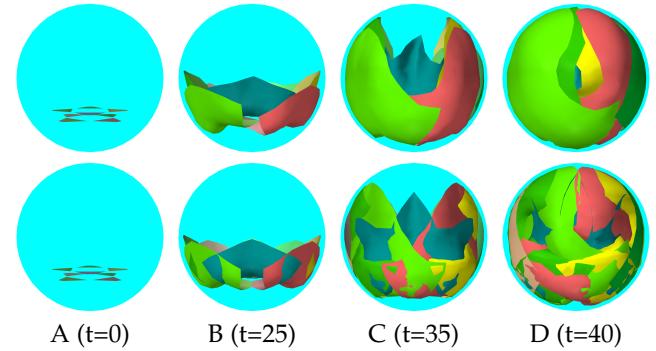


Fig. 12. Time-Lapse Growth of Thin Leaflets within a Confining Sphere. Collision handling between individual leaflets enabled is shown (top row) and contrasted against collision handling disabled (bottom row).

each incision. Altogether, the experiment highlights that the growing solid-shells can capture residual stress dynamics that accompany large plastic deformations.

4.8 Ripple Bouquet: Collision Handling Demo 1

We depart from existing FEM growth scenarios in the literature to now simulate large plastic deformations with collision handling. In this experiment, a collision stress test is set up where multiple layers of ripples are grown and forced to collide with each other. The time-lapse growth of this model is shown in Fig. 1. The model is set with a Young's modulus of 10^7 , and radius to thickness ratio of 2172:1. The model is grown by applying diffusible PAR and PER stretching morphogen along the boundary of each layer. d_{PAR} is fixed to the global z-axis, giving a bias for the elements to expand upwards regardless of their orientation. In the first 8 seconds of growth, the starting model (A) transforms into (B), which has small simple ripples developed. When the secondary ripples emerge in (C), collisions begin to occur. Eventually, the model uptakes a form where the developed ripples exist in a penetration-free state (D). Without collision handling, penetrations become severe artifacts during simulation, as shown in Fig. 11. Overall, this experiment highlights the importance of collision handling on morphology development involving large plastic deformations.

4.9 Confined Growth: Collision Handling Demo 2

When a growth model is confined to grow within a fixed volume, the model can deform into the matching shape of the fixed volume. A simulation of this scenario is depicted in Fig. 12. The starting model is composed of 8 leaflets that are distributed circularly in two layers (top A). The bottom

innermost nodes are permanently pinned in-place throughout the simulation. Growth is enabled by supplying the outermost nodes with a constant PAR and PER stretching morphogen concentration that diffuses throughout. Young's modulus was set to 10^5 with a sphere radius to leaflet thickness ratio of 2000:1. Likewise in the previous experiment, d_{PAR} is held constant to the global z-axis.

When growth begins, the leaflets expand outwards towards the wall of the sphere. Eventually, the leaflets collide against the sphere, causing growth to be deflected along the inner surface of the sphere (top B-C). With enough time, the growth model engulfs the available space and altogether becomes spherical-like itself (top D). The bottom row of Fig. 12 shows the scenario where collision handling between leaflets is disabled. Artifacts appear due to the penetration between leaflets. Because the inner leaflets no longer press the outer leaflets against the sphere surface, the final model (bottom D) appears less uniform in curvature.

Confined growth can arise during tissue development. Particularly in plants, there are studies suggesting that volume constraints can influence the possible shapes of leaves and flower petals [2, 3]. This demo aims to demonstrate the potential of using augmented solid-shells towards studying the effects of confinement on growth.

4.10 Solid-Shell and Thin-Shell Comparison

Thin-shells and solid-shells have a fundamental difference in terms of their bending implementations. Narain's hinge-based (HB) thin-shells, known to support non-zero rest angles, have bending forces that are a function of the dihedral angles between adjacent elements, and have been demonstrated to be a reliable basis for applications involving plasticity, remeshing, and collision handling [13] [12]. In contrast, bending kinematics of solid-shells are derived much from volumetric elements in that the element can shear across its thickness to induce bending. For discrete geometric thin-shells, bending forces are derived from bending strains that are measured in terms of the second fundamental form (II). To extend its bending kinematics, these thin-shells can feature a degree of freedom at each mesh edge to soften the Kirchhoff assumption, arising the shear-deformable Discrete Cosserat (DC) thin-shell where the inextensible director is no longer fixed to the surface normal [42]. To observe the difference between these shell types, we simulate a bending experiment using solid-shells, HB thin-shells, and DC thin-shells. Volumetric elements are simulated alongside to serve as a ground-truth reference for thickness shearing.

The bending experiment consists of a square sheet whose rest state is permanently altered to match a tight cylinder shape. The starting 1-by-1 square sheet of 20000 elements has a length-thickness ratio of 200:1 and a Young's modulus of 10^4 . The targetted cylinder shape is essentially the same 1-by-1 square sheet but rolled into a cylinder until its circumference is one-tenth of the square sheet's length. In DC thin-shells, the elemental rest II is initialized using the same elemental II of the targetted cylinder mesh. For HB thin-shells, the elemental plastic bending strain \mathbf{Y} is specified as a 3×3 matrix that is zeroed except for \mathbf{Y}_{22} set to 2π . This matrix has an effect of offsetting the dihedral "rest

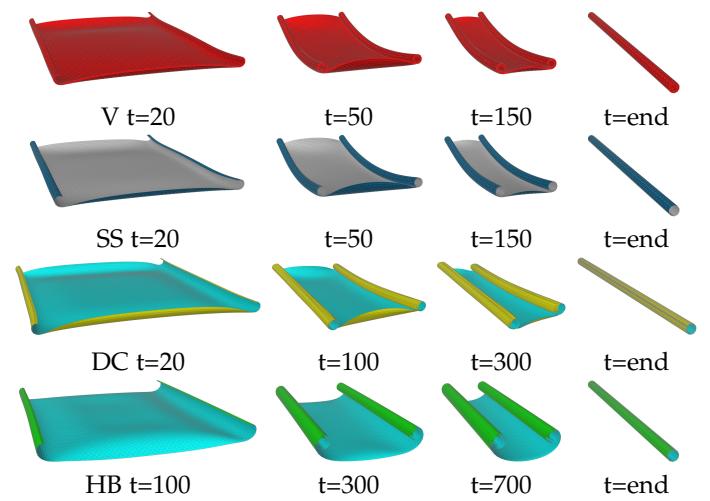


Fig. 13. Volumes (V), Solid-Shells (SS), Discrete Cosserat (DC) Thin-Shells, and Hinge-Based (HB) Thin-Shells undergoing Severe Plastic Bending. For a given element type, a starting square sheet has its rest state altered to target a tight cylinder. $t = \text{end}$ corresponds to time step t where the model energy (i.e. sum of the nodal norm-squared velocities) reaches below 10^{-3} .

angles" of the edges such that the tight cylinder is formed. For volumes and solid-shells, the rest positions of the front and back nodes are altered to coincide with the targetted cylinder mesh.

In the interest of isolating the differences between the element types as much as possible, external processes namely diffusion, plastic embedding, remeshing, and collision handling have been disabled. In addition, for better comparison with the volume element, the number of integration points of the solid-shell element has been reduced to match the 6 integration points of the volume element. Lastly, simulation is halted when the model at hand reaches an energy threshold below 10^{-3} ; energy is calculated as the sum of the nodes' norm-squared velocities.

The transition from a square sheet into a tight cylinder shape is displayed in Fig. 13 for the 4 element types. For HB thin-shells (4th row of Fig. 13), the sheet curls tightly from both sides, forming a pair of rolls, and eventually consolidating into a single tight cylinder. Compared to HB thin-shells, solid-shells (2nd row) also exhibit the formation of two rolls and an eventual cylinder but with the addition of secondary deformations that are similar to the reference volume simulation: (1) auxiliary curling that is perpendicular to the pair of rolls and (2) arcing of the pair of rolls at $t = 100$. These secondary deformations are hypothesized to correspond to local energy minimas that the sheet can temporarily converge into as it traverses down the energy landscape. Specifically, when volumes and solid-shells are subjected to high bending strain, the center of the sheet can temporarily experience compression along the bottom surface due to the strong thickness shearing that is emanating from the two rolls. In turn, the compression squeezes the sheet into stretching towards the perpendicular direction in order to reduce the overall energy, leading to the formation of curls that protrude perpendicularly, which in turn, stresses the two rolls to arc in the same upwards direction. For DC thin-shells (3rd row), both auxiliary curling and roll arcing are also visible to an extent. The roll

TABLE 2

Performance Timings for Volume, Solid-Shell, Hinge-Based (HB) Thin-Shell, and Discrete Cosserat (DC) Thin-Shell elements. Volumes and Solid-Shells are implemented as 6-node elements with 6 integration points each, while HB and DC Thin-Shells respectively follow Narain and LibShell's 3-node implementation. The profiled simulation entails a square sheet of elements undergoing plastic bending strain, detailed in Section 4.10. The "Force and Stiffness Computation" (Build), "Implicit Time Integration" (Solve), and their sum are profiled for each time step and averaged across 300 time steps in milliseconds on an Intel i7-7700K and 16GB RAM machine.

Element	1250 Elements			5000 Elements			20000 Elements		
	Build	Solve	Sum	Build	Solve	Sum	Build	Solve	Sum
Volume	21.1	3.9	25.0	83.0	19.9	102.9	277.9	87.6	365.5
Solid-Shell	20.7	3.8	24.5	84.0	19.2	103.2	273.0	78.5	351.5
HB Thin-Shell	19.1	3.1	22.2	79.7	11.2	90.9	244.0	51.1	295.1
DC Thin-Shell	131.5	5.1	136.6	442.3	28.3	470.6	1632.1	146.0	1778.1

arching is particularly more faint at $t = 100$ and $t = 300$, suggesting a weaker thickness shearing being emanated from the two rolls. Weaker thickness shearing energies allude to a possible explanation for why the DC and HB thin-shells have a slower rate of curling while converging into the final cylinder shape, exposing a consequence of when thickness is inextensible and virtualized in these thin-shells. Nevertheless, the results do show that DC thin-shells are capable of exhibiting secondary deformations that are induced by thickness dynamics.

Compared to the other element types, the DC thin-shells exhibit two noticeable asymmetric behavior. At $t = 100$, asymmetry can be observed in the pair of rolls where the right roll is visibly more arched. In addition, at $t = 300$, the rolls do not necessarily spiral inwards within themselves symmetrically. These two asymmetric behavior are suggested to arise from the increased numerical error when the element *relative thickness* is large. For DC thin-shells, its reduction of the 3D energy elastic model into a simpler 2D model relies on the assumption that the element thickness-to-length ratio is small (i.e. substantially below 1.0) [42]. At a given thickness, this assumption has an implication where the numerical error increases when the mesh is increasingly refined [42, Fig. 5.2]. The amplified bending experiment particularly accentuates the violation of this assumption where the sheet length is significantly high resolution such that the element thickness-to-length ratio is close to 1.0. Consequently, DC thin-shells can exhibit unexpected dynamics when using high resolution meshes under high stress.

We also provide a direct comparison of the computational complexity of the solid-shell against the other element types. The runtime differences are shown in Table 2, which summarizes the profiled runtime metrics collected during the bending experiment. For a given element type and mesh resolution, the "Force and Stiffness Computation" and "Implicit Time Integration" are timed in each time step and averaged across 300 time steps.

The runtime performance between volumes and solid-shells do not largely differ as both require generating a mass matrix of the same size and sparsity. The force and stiffness computation runtime of solid-shells only deviated between -1% to 2% relative to using volumes. For the implicit time integration solve, we observe a 2% to 12% improvement in runtime using solid-shells, although we suspect that this consistent improvement is contributed more so from

TABLE 3

Performance Timings of Growth Simulations. "# Ele" and "# Con" are respectively the number of elements and collision constraints present at the end of the simulation. Simulations were all performed on an Intel i7-7700K and 16GB RAM machine.

Model	Fig	Frames	# Ele	# Con	Elapsed
Curling Sheet	8.D	1500	1k	368	7 min
Delicate Wrinkles	6.D	300	30k	-	8 min
Fruit	7.D	2000	32k	-	46 min
Ripple Cascade	5.D	2000	51k	-	80 min
Ripple Bouquet	1.D	2500	11k	552	3.6 hrs
Confined Growth	12.D	4000	8k	3276	3.7 hrs

memory layout and caching factors as we store nodal and director information contiguously for solid-shells.

We observe that 6-node volumes and solid-shells do not necessarily incur a steep performance penalty relative to using 3-node HB thin-shells during growth. Much of the runtime is largely dominated by the slower force and stiffness computation, which accounts for 76% to 84% of the total runtime across all elements. When the runtime metrics are summed, thin-shells exhibit a 13% to 26% improvement over volumes and 10% to 19% improvement over solid-shells.

The alternative 3-node DC thin-shell is found to be the most computationally expensive. Relative to solid-shells, DC thin-shells require 456% to 558% more time to simulate a single time step. When profiled more finely, it is revealed that the majority of the calculation runtime stems from computing the Hessian of the bending energy, which accounts for 68% to 70% of the force and stiffness calculation runtime alone. Chen et al. also observes the bending energy Hessian to be expensive. In their 3000 element annuli and 19000 element pasta curling experiments, timings respectively averaged to 8 seconds and 23 seconds per time step on an Intel Xeon E3-1270 and 16GB RAM machine, however, these timings can be reduced by a proposed approximation to the exact bending Hessian [21].

Recalling that 3-node DC thin-shells particularly include a degree of freedom at each mesh edge to soften the Kirchhoff assumption, its stiffness matrix is sized B-by-B where B is equal to the number of nodal and edge degrees of freedoms. Therefore, these thin-shells have similar solves times with 6-node elements within the same order of magnitude. We emphasize that the higher solve times observed in the DC thin-shells implemented compared to 6-node elements is suggested to be due to memory layout as the stiffness matrix, implemented as sparse block matrix, is made more finely-grained in terms of block count to easily account for the new node-edge and edge-edge stiffness relationships that accompany the usual node-node relationships.

5 LIMITATIONS AND FUTURE WORK

The framework carries limitations that accompany finite element simulations, particularly regarding the element inversion risk and collision handling performance. Element inversion can occur when the build-up of plastic strain exceeds the rate of plastic embedding, often due to excessive morphogen being applied within a short amount of time. Consequently, it is necessary to grow at a slow enough rate

such that the rest configuration is given enough iterations to converge into a shape that represents the plastic strain [12].

Collision handling is a known bottleneck in physics-based simulation [31, 38]. This can be observed in Table 3 which reveals that the Ripple Bouquet and Confined Growth required the most amount of time to simulate. Much of the computational complexity of collision handling is attributed from contact points typically becoming persistent once they form due to plasticity. Persistent contact points cause an accumulation of unilateral constraints that the LCP solver must account for. Consequently, the collision resolution process can slow down substantially, which was also observed in another work that simulated crumpled paper using plasticity and unilateral constraints [13, Table 1].

There are approaches to address the collision handling performance. One technique is to treat contact constraints as bilateral (i.e. equality) constraints during each simulation step, with separation applied as required between steps, which effectively distributes the LCP solve across the entire simulation to improve simulation efficiency [35]. The LCP solve itself can be sped-up by GPU parallelization [30]. Alternatively, a different paradigm could be taken, namely Incremental Potential Contact where CCD is combined with a smooth barrier potential and a variational time-stepped solver to handle collisions efficiently within a given error tolerance [43].

While solid-shells serve as an intermediary between thin-shells and volumes, the performance of solid-shells are much closer to volumes. Both element types utilize a 6-node configuration and the same material model interface, which results in a similar computational complexity. Consequently, the solid-shell is more so distinguished from its volume counterpart in terms of its added thin-shell physical characteristics, namely the 2D manifold interface and decoupled stretching and bending kinematics.

6 CONCLUSION

Solid-shells have not previously been used for simulating large growth deformations and have potential to provide a more physically realistic representation of thin tissues, as compared to thin-shells, by retaining volumetric characteristics. In response, we have developed a new growth framework to investigate the potential of simulating large growth using solid-shells. Solid-shells offer shell-like decoupling of stretching and bending to handle large bends, while accounting for thickness compression, shearing, and compatibility with existing 3D material laws. Both stretching and bending are captured by the director vectors, allowing both kinematics to share a common plastic procedure, namely plastic embedding. Plastic embedding enables large plastic deformations to occur in solid-shells by deforming the stress-free rest configuration according to the growth/plastic strain. Along with plastic embedding, solid-shells are highly adaptable to incorporate the many other techniques required for simulating challenging growth structures. Strain-aware adaptive remeshing can be adapted to automatically refine the mesh where needed to maintain a stable and high quality mesh topology. Alongside, the large and intricate plastic deformations can be governed and influenced by adapting efficient morphogen

diffusion, and a discrete and continuous constraint-based collision handler.

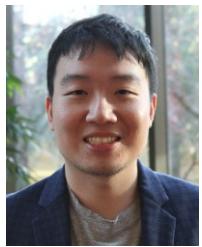
The growth experiments presented in this paper demonstrate a wide range of growth scenarios that the framework can simulate. Solid-shells are adaptable to handle the growth of large fractal patterns, fine delicate wrinkles, and large plastic bending in thin tissue. The solid-shells are also adept in the growth of thick tissue along both its plane and normal, as shown in the formation of a fruit-like shape. In contrast to similar experiments in the literature, our experiments incorporate strain-aware adaptive remeshing to permit detailed growth starting from a coarse mesh. We also showcase novel experiments of simulating large plastic deformations with collision handling of growing FEM tissue. Solid-shells were also directly compared with thin-shells to highlight their performance and the added effect of thickness shearing dynamics. Altogether, the experiments conducted demonstrate that the augmented solid-shell formulation is a viable finite element for physics-based simulation of large and intricate plastic deformations of thin structures compared to thin-shells.

REFERENCES

- [1] A. M. Turing, "The chemical basis of morphogenesis," *Bulletin of mathematical biology*, vol. 52, no. 1-2, pp. 153–197, 1990.
- [2] E. Couturier, S. C. Du Pont, and S. Douady, "The filling law: a general framework for leaf folding and its consequences on leaf shape diversity," *Journal of theoretical biology*, vol. 289, pp. 47–64, 2011.
- [3] S. Takeda, A. Iwasaki, N. Matsumoto, T. Uemura, K. Tatematsu, and K. Okada, "Physical interaction of floral organs controls petal morphogenesis in arabidopsis," *Plant physiology*, vol. 161, no. 3, p. 1242, 2013.
- [4] M. J. Matthews, "Physically based simulation of growing surfaces," 2002.
- [5] T. Omori, T. Ishikawa, D. Barthès-Biesel, A.-V. Salsac, J. Walter, Y. Imai, and T. Yamaguchi, "Comparison between spring network models and continuum constitutive laws: Application to the large deformation of a capsule in shear flow," *Physical Review E*, vol. 83, no. 4, p. 041918, 2011.
- [6] R. Kennaway, E. Coen, A. Green, and A. Bangham, "Generation of diverse biological forms through combinatorial interactions between tissue polarity and growth," *PLoS computational biology*, vol. 7, no. 6, p. e1002071, 2011.
- [7] E. Efrati, E. Sharon, and R. Kupferman, "Elastic theory of unconstrained non-euclidean plates," *Journal of the Mechanics and Physics of Solids*, vol. 57, no. 4, pp. 762–775, 2009.
- [8] E. N. Dvorkin and K.-J. Bathe, "A continuum mechanics based four-node shell element for general non-linear analysis," *Engineering computations*, 1984.
- [9] A. Goriely and M. Amar, "On the definition and modeling of incremental, cumulative, and continuous growth laws in morphoelasticity," *Biomechanics and Modeling in Mechanobiology*, vol. 6, no. 5, pp. 289–296, 2007.
- [10] M. Wicke, D. Ritchie, B. M. Klingner, S. Burke, J. R. Shewchuk, and J. F. O'Brien, "Dynamic local remeshing

- for elastoplastic simulation," in *ACM Transactions on graphics (TOG)*, vol. 29, no. 4. ACM, 2010, p. 49.
- [11] D. Kierzkowski, A. Runions, F. Vuolo, S. Strauss, R. Lymouridou, A.-L. Routier-Kierzkowska, D. Wilson-Sánchez, H. Jenke, C. Galinha, G. Mosca *et al.*, "A growth-based framework for leaf shape development and diversity," *Cell*, vol. 177, no. 6, pp. 1405–1418, 2019.
- [12] C. Gingras and P. G. Kry, "Procedural modelling with reaction diffusion and growth of thin shells," in *Proceedings of the 45th Graphics Interface Conference on Proceedings of Graphics Interface 2019*. Canadian Human-Computer Communications Society, 2019, pp. 1–7.
- [13] R. Narain, T. Pfaff, and J. F. O'Brien, "Folding and crumpling adaptive sheets," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 51, 2013.
- [14] J. Rosenkrantz and J. Louis-Rosenberg, "Nervous systems." (2018). <https://n-e-r-v-o-u-s.com>, 2018.
- [15] R. Vetter, "Growth, interaction and packing of thin objects," Ph.D. dissertation, ETH Zurich, 2015.
- [16] J. Yin, Z. Cao, C. Li, I. Sheinman, and X. Chen, "Stress-driven buckling patterns in spheroidal core/shell structures," *Proceedings of the National Academy of Sciences*, vol. 105, no. 49, pp. 19 132–19 135, 2008.
- [17] A. M. Zöllner, M. A. Holland, K. S. Honda, A. K. Gosain, and E. Kuhl, "Growth on demand: reviewing the mechanobiology of stretched skin," *Journal of the mechanical behavior of biomedical materials*, vol. 28, pp. 495–509, 2013.
- [18] Y. Zheng, J. Wang, H. Ye, Y. Liu, and H. Zhang, "A solid-shell based finite element model for thin-walled soft structures with a growing mass," *International Journal of Solids and Structures*, vol. 163, pp. 87–101, 2019.
- [19] P. Prusinkiewicz and P. Barbier de Reuille, "Constraints of space in plant development," *Journal of experimental botany*, vol. 61, no. 8, pp. 2117–2129, 2010.
- [20] D. Ambrosi, G. A. Ateshian, E. M. Arruda, S. Cowin, J. Dumais, A. Goriely, G. A. Holzapfel, J. D. Humphrey, R. Kemkemer, E. Kuhl *et al.*, "Perspectives on biological growth and remodeling," *Journal of the Mechanics and Physics of Solids*, vol. 59, no. 4, pp. 863–883, 2011.
- [21] H.-Y. Chen, A. Sastry, W. M. van Rees, and E. Vouga, "Physical simulation of environmentally induced thin shell deformation," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, p. 146, 2018.
- [22] R. Vetter, N. Stoop, T. Jenni, F. K. Wittel, and H. J. Herrmann, "Subdivision shell elements with anisotropic growth," *International journal for numerical methods in engineering*, vol. 95, no. 9, pp. 791–810, 2013.
- [23] R. Narain, A. Samii, and J. F. O'Brien, "Adaptive anisotropic remeshing for cloth simulation," *ACM transactions on graphics (TOG)*, vol. 31, no. 6, p. 152, 2012.
- [24] P. Betsch, F. Gruttmann, and E. Stein, "A 4-node finite shell element for the implementation of general hyperelastic 3d-elasticity at finite strains," *Computer Methods in Applied Mechanics and Engineering*, vol. 130, no. 1-2, pp. 57–79, 1996.
- [25] S. Dong, Y. Yan, L. Tang, J. Meng, and Y. Jiang, "Simulation of 3d tumor cell growth using nonlinear finite element method," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 19, no. 8, pp. 807–818, 2016.
- [26] J. Solomon, K. Crane, and E. Vouga, "Laplace-beltrami: The swiss army knife of geometry processing," in *Symposium on Geometry Processing graduate school (Cardiff, UK, 2014)*, vol. 2, 2014.
- [27] A. Owens, M. Cieslak, J. Hart, R. Classen-Bockhoff, and P. Prusinkiewicz, "Modeling dense inflorescences," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 136, 2016.
- [28] X. Tang, A. Paluszny, and R. Zimmerman, "Energy conservative property of impulse-based methods for collision resolution," *International Journal for Numerical Methods in Engineering*, vol. 95, no. 6, pp. 529–540, 2013.
- [29] M. A. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross, "Implicit contact handling for deformable objects," in *Computer Graphics Forum*, vol. 28, no. 2. Wiley Online Library, 2009, pp. 559–568.
- [30] P. Kipfer, "LCP algorithms for collision detection using CUDA," *GPU Gems*, vol. 3, pp. 723–740, 2007.
- [31] R. Bridson, R. Fedkiw, and J. Anderson, "Robust treatment of collisions, contact and friction for cloth animation," in *ACM Transactions on Graphics (ToG)*, vol. 21, no. 3. ACM, 2002, pp. 594–603.
- [32] M. Müller and M. Gross, "Interactive virtual materials," in *Proceedings of Graphics Interface 2004*. Canadian Human-Computer Communications Society, 2004.
- [33] S. Maas, D. Rawlins, J. Weiss, and G. Ateshian, "Febio theory manual v2.6," 2014.
- [34] J. E. Akin, *Finite element analysis with error estimators: An introduction to the FEM and adaptive error analysis for engineering students*. Elsevier, 2005.
- [35] J. Lloyd, I. Stavness, and S. Fels, "Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation," in *Soft Tissue Biomech Modeling for Computer Assisted Surgery*. Springer, 2012, pp. 355–394.
- [36] R. F. Valente, R. A. De Sousa, and R. N. Jorge, "An enhanced strain 3d element for large deformation elastoplastic thin-shell applications," *Computational Mechanics*, vol. 34, no. 1, pp. 38–52, 2004.
- [37] G. Irving, J. Teran, and R. Fedkiw, "Invertible finite elements for robust simulation of large deformation," in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2004, pp. 131–140.
- [38] D. Harmon, E. Vouga, R. Tamstorf, and E. Grinspun, "Robust treatment of simultaneous collisions," *ACM Transactions on Graphics (TOG)*, 2008.
- [39] D. Matoz-Fernandez, F. A. Davidson, N. R. Stanley-Wall, and R. Sknepnek, "Wrinkle patterns in active viscoelastic thin sheets," *Physical Review Research*, vol. 2, no. 1, p. 013165, 2020.
- [40] P. B. Canham, "The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell," *Journal of theoretical biology*, vol. 26, no. 1, pp. 61–81, 1970.
- [41] L. A. Taber and J. D. Humphrey, "Stress-modulated growth, residual stress, and vascular heterogeneity," *Journal of biomechanical engineering*, vol. 123, no. 6, pp. 528–535, 2001.
- [42] C. Weischedel, "A discrete geometric view on shear-deformable shell models," 2012.
- [43] M. Li, Z. Ferguson, T. Schneider, T. Langlois, D. Zorin,

D. Panozzo, C. Jiang, and D. M. Kaufman, "Incremental potential contact: Intersection-and inversion-free, large-deformation dynamics," *ACM transactions on graphics*, 2020.



Danny Huang received his B.Sc. and M.Sc. in Computer Science from the University of Saskatchewan. He is currently working as a research assistant at the Global Institute for Food Security in Saskatchewan, Canada. His research interests include computer animation and deep learning.



Ian Stavness is an Associate Professor in the Department of Computer Science at the University of Saskatchewan. He directs the Biological Imaging & Graphics lab focused on 3D computational modeling, image analysis and deep learning for biological and biomedical applications. Dr. Stavness completed his Postdoctoral Fellowship in 2012 at Stanford University in the NIH Center for Biomedical Computation. He received his PhD and MSc in Electrical and Computer Engineering from the University of British Columbia, and bachelor degrees in Electrical Engineering and Computer Science from the University of Saskatchewan.