Thin Shell Simulation with Environmental Stimuli

Nathan King
University of Waterloo
n5king@uwaterloo.ca

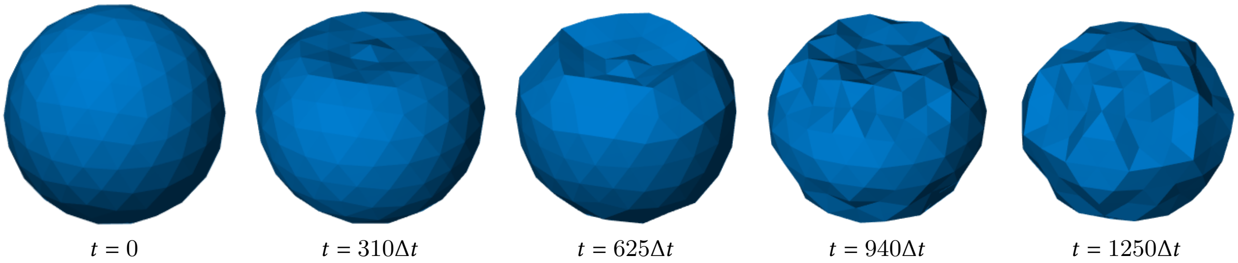

Figure 1: Contraction of a plastic sphere by a directed heat source at the top of the sphere.

ABSTRACT

This project implements a physically based thin shell model that can incorporate deformations due to environmental stimuli. The implementation is based on the model in [Chen et al. 2018]. This thin shell model captures the physics while avoiding expensive volumetric discretizations. Environmental effects induce local, dynamic changes to the rest metric of the material. Simulations of a square deforming provides an analytical experiment. A qualitative simulation of heating a plastic sphere is also given.

CCS CONCEPTS

• **Computing methodologies → Physical simulation**.

KEYWORDS

differential growth, thin shell

ACM Reference Format:
Nathan King. 2019. Thin Shell Simulation with Environmental Stimuli. In *Proceedings of CS888*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/8888888.7777777

1 INTRODUCTION

Thin shells are common objects in our physical world. Their characteristics are therefore studied in many fields, e.g., mechanics, biology, architecture, and computer graphics. Understanding how shells deform is important in mechanics for fabrication of tubes that change shape due to temperature or in biology when an almond leaf is attacked by Taphrina Deformans [Sharon and Efrati 2010]. In architecture, elastic shells are useful for self-constructing timber surfaces [Wood et al. 2016] or to construct climate responsive aperture [Correa and Menges 2017]. In computer graphics there are many applications for simulating everyday phenomena of thin shells, e.g., leaves curling up when drying out, paper stretching and curling due to spilled coffee, and plastic shrinking due to heat [Chen et al. 2018].

This project consists of implementing a physically accurate, discrete, thin shell model [Chen et al. 2018]. This thin shell model can be used to simulate deformation of shells with small thickness, while avoiding expensive volumetric discretizations. Using the Kirchhoff-Love assumption the volumetric shell is represented by just its midsurface. A triangulated mesh can then be used as a discrete version of the continuous midsurface.

The thin shell model has two main components. A discrete geometric shell model is given that supports arbitrary rest shape and physical settings (thickness and Lamé parameters). The elastic shell model is also coupled with effects from the surrounding environment such as moisture and temperature.

The next section discusses related work for modelling and simulating thin shells. The geometric and physically based thin shell model is detailed in section 3. First the continuous model is discussed followed by its representation in the discrete setting. In section 4, implementation details are given before presenting some simulation experiments. The paper concludes with a discussion and ideas for future work.

2 RELATED WORK

Environmental effects of burning a shell boundary or melting a solid have been explored in [Liu et al. 2009; Losasso et al. 2006; Melek and Keyser 2003, 2005, 2007]. Most of these works do not include elastic deformations or are non-physical. A more principle elastic model based on mass-springs is presented in [Larboulette et al. 2013]. However, update rules for the mass-springs can be difficult for non-uniform materials. A bilayer of springs is used in [Jeong et al. 2011, 2013] to allow differential growth due to moisture gradients across the thickness of leaves.

A more physically based and mathematically elegant approach involves using the Kirchhoff-Love assumption. This assumes that the shell does not shear in the transverse direction. That is, the shell volume can be represented by normal offsets of its midsurface.

CS888, Physics Based Animation, University of Waterloo

ACM ISBN 978-1-4503-1234-5/17/07.
https://doi.org/10.1145/8888888.7777777

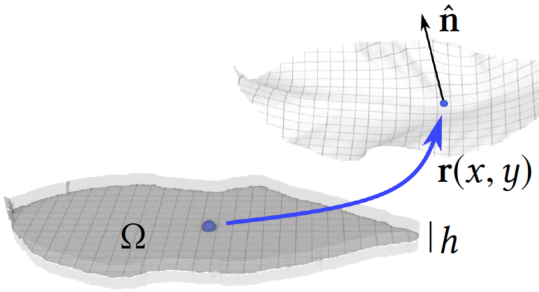

Figure 2: The volumetric slab $\Omega \times [-h/2, h/2]$ is represented by just its midsurface r, which is a map from the parameterization domain Ω to a surface in $\mathbb{R}^3$. Figure reproduced from [Chen et al. 2018].

Thus, the physics can then be detailed in terms of the fundamental forms of its midsurface [Ciarlet 2000; Weischedel 2012]. Note that the work presented in [Weischedel 2012] provides a Cosserat shell model, which has a very similar discrete geometric framework to [Chen et al. 2018]. The key property of these models is that the rest, or strain-free, state of the shell is not flat, i.e., it can protrude into $\mathbb{R}^3$. There has been much research in physics for these non-Euclidean shells [Efrati et al. 2009a,b; Goriely and Amar 2005; Kim et al. 2012; Klein et al. 2007; Sharon and Efrati 2010].

In graphics, thin shells were first used for cloth simulation with flat rest states [Baraff and Witkin 1998; Bridson et al. 2002]. The introduction of non-zero rest curvature to graphics was first done in [Grinspun et al. 2003]. An extrinsic description of the surface is however necessary to be physically robust [Grinspun et al. 2006]. The approach take in this project using [Chen et al. 2018] incorporates information about the extrinsic deformation of the shell.

3 GEOMETRIC SHELL MODEL

The thin shell model will be briefly discussed in the continuous setting first. The remainder of this section will then present the discrete model on a triangulated mesh. The energy formulation of the model is the necessary component to determine the dynamics of the shell. The overall idea is to determine the energy based on differences between the rest and current (deformed) states. The energy is differentiated to compute forces on the shell. The motion of the shell is then determined using Newton's second law.

3.1 Continuous formulation

Consider a shell $\mathcal{S} \subset \mathbb{R}^3$ that has thickness h. The shell can be represented by an embedding $\phi : \Omega \times [-h/2, h/2] \to \mathbb{R}^3$, where Ω is a planar parameter domain. The entire volume of the shell can be represented by only its midsurface $\mathrm{r} : \Omega \to \mathbb{R}^3$ using the Kirchhoff-Love assumption [Ciarlet 2000; Weischedel 2012]. The embedding of the shell can be written as

$$\phi(x, y, z) = \mathrm{r}(x, y) + z\hat{\mathrm{n}}(x, y), \tag{1}$$

where $\hat{\mathrm{n}}$ is the normal of r (see Figure 2).

The metric g on the volumetric slab $\Omega \times [-h/2, h/2]$ can be written in terms of the geometry of the midsurface r. Denote the first, second, and third fundamental forms of r as

$$\mathrm{a} = d\mathrm{r}^T d\mathrm{r}, \quad \mathrm{b} = -d\mathrm{r}^T d\hat{\mathrm{n}}, \quad \mathrm{c} = d\hat{\mathrm{n}}^T d\hat{\mathrm{n}},$$

respectively. Then the metric on the volume is

$$\mathrm{g} = \begin{bmatrix} \mathrm{a} - 2z\mathrm{b} + z^2\mathrm{c} & 0 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

It is not always possible to find an embedding into $\mathbb{R}^3$ such that the thin shell is at rest globally. For example, coffee spilled on paper will cause the paper to buckle out of plane. A new rest state is obtained by balancing between stretching and bending strains, which makes the rest state non-Euclidean. A rest metric $\bar{\mathrm{g}}$ for the volume is necessary to model any phenomena whose rest state can become non-Euclidean. For thin shells only differential in-plane swelling is important. It is therefore assumed that the rest metric is linear in the thickness direction

$$\bar{\mathrm{g}} = \begin{bmatrix} \bar{\mathrm{a}} - 2z\bar{\mathrm{b}} & 0 \\ 0 & 1 \end{bmatrix}. \tag{3}$$

Note that the special case when a shell is initially flat at rest, an embedding $\bar{\mathrm{r}}$ exists and the rest fundamental forms can be specified by the $\bar{\mathrm{a}} = \mathrm{a}^0$ and $\bar{\mathrm{b}} = \mathrm{b}^0$, where a^0 and b^0 are the shells fundamental forms at $t = 0$. Environmental effects are incorporated by inducing changes in the rest metric $\bar{\mathrm{g}}$.

Thin shells are more easily bent than they are stretched. Therefore, it is assumed that the in-plane strain of the midsurface is small, i.e., $\|\bar{\mathrm{a}}^{-1}\mathrm{a} - \mathrm{I}\|_\infty < h$. It is also assumed that the material of the shell is uniform and isotropic. The St. Venant-Kirchhoff material model with the Green strain is the simplest constitutive law that is consistent with the above assumptions [Weischedel 2012]. The elastic energy density can be approximated up to $O(h^4)$ as

$$W(x, y) = \left(\frac{h}{4}\|\bar{\mathrm{a}}^{-1}\mathrm{a} - \mathrm{I}\|_{SV}^2 + \frac{h^3}{12}\|\bar{\mathrm{a}}^{-1}(\mathrm{b} - \bar{\mathrm{b}})\|_{SV}^2\right)\sqrt{\det \bar{\mathrm{a}}}, \tag{4}$$

where $\|\cdot\|_{SV}$ is the St. Venant-Kirchhoff norm

$$\|\cdot\|_{SV}^2 = \frac{\alpha}{2}\mathrm{tr}(\cdot)^2 + \beta\mathrm{tr}(\cdot^2).$$

The material parameters α and β are dependent on Young's modulus E and Poisson's ratio ν as

$$\alpha = \frac{E\nu}{1 - \nu^2}, \qquad \beta = \frac{E}{2(1 + \nu)}.$$

3.2 Discrete formulation

Fundamental Forms and Energies. The midsurfaces of the shells are approximated with triangular meshes $M = (V, E, F)$ consisting of vertices, edges, and faces. It is assumed that the fundamental forms, $\mathrm{a}, \mathrm{b}, \bar{\mathrm{a}}, \bar{\mathrm{b}}$, are constant over the face of each triangle. A global parametrization can not be assumed since the midsurface is assume to be non-Euclidean. Therefore, each triangle is independently parameterized in its own barycentric coordinates. It is important to note that this means that the discrete fundamental forms will not agree across the edge of two adjacent faces.

Denote the vertex positions in V as $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_m$, which are a discrete representation of r. The face in F that contains x_i, x_j, x_k, is denoted f_{ijk}. The face f_{ijk} is embedded locally using the affine mapping $\mathrm{r}_{ijk} : \mathcal{T} \to \mathbb{R}^3$, which is given by

$$\mathrm{r}_{ijk}(u, v) = \mathrm{x}_i + u(\mathrm{x}_j - \mathrm{x}_i) + v(\mathrm{x}_k - \mathrm{x}_i),$$

where $\mathcal{T}$ is the unit triangle with vertices $(0,0)$, $(1,0)$, $(0,1)$. By differentiating this embedding with respect to u and v it can be shown that the first fundamental form on f_{ijk} is given by

$$\mathrm{a}_{ijk} = \begin{bmatrix} \|\mathrm{x}_j - \mathrm{x}_i\|^2 & (\mathrm{x}_j - \mathrm{x}_i)\cdot(\mathrm{x}_k - \mathrm{x}_i) \\ (\mathrm{x}_j - \mathrm{x}_i)\cdot(\mathrm{x}_k - \mathrm{x}_i) & \|\mathrm{x}_k - \mathrm{x}_i\|^2 \end{bmatrix}.$$

Notice that this 2×2 matrix encodes information about length on the triangle f_{ijk}.

A discretization of the second fundamental form is more involved. One would like to use the Kirchhoff-Love assumption to move the midsurface into the volume by parallel offsets. However, offsetting triangle meshes in the normal direction is not guaranteed to produce a triangle mesh. As discussed in [Chen et al. 2018] there is no perfect choice for parallelity of the mesh. Using the mid-edge normal however has the numerical advantages of having a compact stencil and being robust to triangle inversion [Grinspun et al. 2006; Weischedel 2012].

Denote e_i as the edge opposite the vertex i on face f_{ijk}, whose normal is

$$\mathrm{n}_{ijk} = \frac{(\mathrm{x}_j - \mathrm{x}_i)\times(\mathrm{x}_k - \mathrm{x}_i)}{\|(\mathrm{x}_j - \mathrm{x}_i)\times(\mathrm{x}_k - \mathrm{x}_i)\|}.$$

Then the mid-edge normal $\hat{\mathrm{n}}_i$ is given on edge e_i as

$$\hat{\mathrm{n}}_i = \begin{cases} \mathrm{n}_{ijk}, & \text{if } \mathrm{e}_i \text{ on } \partial M, \\ \text{mean of } \mathrm{n}_{ijk} \text{ of faces incident on } \mathrm{e}_i, & \text{otherwise.} \end{cases}$$

Let $\mathrm{f}^{\epsilon}_{ijk}$ be the triangle formed by offsetting all the edges of f_{ijk} by a distance ϵ in their mid-edge normal direction. Denote the first fundamental form of $\mathrm{f}^{\epsilon}_{ijk}$ as $\mathrm{a}^{\epsilon}_{ijk}$. Then, from (2), we can think of b_{ijk} as a first order correction to $\mathrm{a}^{\epsilon}_{ijk} = \mathrm{a}_{ijk} - 2\epsilon\mathrm{b}_{ijk} + O(\epsilon^2)$. This gives the definition of the discrete second fundamental form as

$$\mathrm{b}_{ijk} = \begin{bmatrix} (\hat{\mathrm{n}}_i - \hat{\mathrm{n}}_j)\cdot(\mathrm{x}_i - \mathrm{x}_j) & (\hat{\mathrm{n}}_i - \hat{\mathrm{n}}_j)\cdot(\mathrm{x}_i - \mathrm{x}_k) \\ (\hat{\mathrm{n}}_i - \hat{\mathrm{n}}_k)\cdot(\mathrm{x}_i - \mathrm{x}_j) & (\hat{\mathrm{n}}_i - \hat{\mathrm{n}}_k)\cdot(\mathrm{x}_i - \mathrm{x}_k) \end{bmatrix}.$$

Now the discrete energy can be written, analogous to the continuous formulation, as

$$E_{\text{elastic}}(\mathrm{x}) = \sum_{\mathrm{f}_{ijk}\in F}\left(\frac{h}{8}E_{\text{stretch}} + \frac{h^3}{24}E_{\text{bend}}\right)\sqrt{\det \bar{\mathrm{a}}_{ijk}},$$

where

$$E_{\text{stretch}} = \|\bar{\mathrm{a}}_{ijk}^{-1}\mathrm{a}_{ijk} - \mathrm{I}\|^2_{SV},$$
$$E_{\text{bend}} = \|\bar{\mathrm{a}}_{ijk}^{-1}(\mathrm{b}_{ijk} - \bar{\mathrm{b}}_{ijk})\|^2_{SV}.$$

A viscous damping energy is also included in the model to produce drag forces necessary to avoid unrealistic infinite elastic oscillations. The Kelvin-Voigt damping model is used, which corresponds to the following damping energy

$$E_{\text{damp}}(\mathrm{x}, \mathrm{x}^{\text{prev}}) = \frac{\eta}{E}\Delta t\sum_{\mathrm{f}_{ijk}}\left(\frac{h}{8}W_{\text{stretch}} + \frac{h^3}{24}W_{\text{bend}}\right)\sqrt{\det \bar{\mathrm{a}}_{ijk}},$$

where

$$W_{\text{stretch}} = \left\|\left(\mathrm{a}^{\text{prev}}_{ijk}\right)^{-1}\frac{\mathrm{a}_{ijk} - \mathrm{a}^{\text{prev}}_{ijk}}{\Delta t}\right\|^2_{SV},$$
$$W_{\text{bend}} = \left\|\left(\mathrm{a}^{\text{prev}}_{ijk}\right)^{-1}\frac{\mathrm{b}_{ijk} - \mathrm{b}^{\text{prev}}_{ijk}}{\Delta t}\right\|^2_{SV}.$$

Time Integration. Assume there are m vertices in our triangular mesh M. We can form a vector $\mathrm{x}\in\mathbb{R}^{3m}$ consisting of all the vertex positions $\mathrm{x}_i \in \mathbb{R}^3$. Denoting the force exerted on vertex i as $\mathrm{F}_i \in \mathbb{R}^3$, the force vector $\mathrm{F}\in\mathbb{R}^{3m}$ can be constructed by stacking F_i's in the same order as x_i's in x. Then Newton's second law can be written as

$$\Lambda\frac{d^2\mathrm{x}}{dt^2} = \mathrm{F}, \tag{5}$$

where $\Lambda\in\mathbb{R}^{3m\times 3m}$ is the diagonal mass matrix.

The matrix Λ has the mass λ_i of each vertex on its diagonal, i.e.,

$$\operatorname{diag}(\Lambda) = (\lambda_1, \lambda_1, \lambda_1, \ldots, \lambda_m, \lambda_m, \lambda_m).$$

The mass at vertex i is computed as one third the mass of each triangle in its one-ring

$$\lambda_i = \sum_{\mathrm{f}\sim\mathrm{x}_i}\frac{\rho h}{3}\frac{\sqrt{\det\bar{a}_{\mathrm{f}}}}{2},$$

where ρ is denotes shell density and $\sqrt{\det\bar{a}_{\mathrm{f}}}/2$ is the area of face f.

Evolving the model in time is done using the implicit Euler scheme. Denote the position and velocity at time $t^i = t^0 + i\Delta t$ as x^i and v^i, respectively. Then, after writing (5) as a system of first order equations, time stepping with implicit Euler is gives

$$\begin{aligned} \mathrm{x}^{i+1} &= \mathrm{x}^i + \Delta t\mathrm{v}^{i+1}, \\ \Lambda\frac{\mathrm{v}^{i+1} - \mathrm{v}^i}{\Delta t} &= \mathrm{F}\left(\mathrm{x}^{i+1}, \mathrm{x}^i\right). \end{aligned} \tag{6}$$

The force is given by

$$\mathrm{F}(\mathrm{x}^{i+1}, \mathrm{x}^i) = \mathrm{F}_{\text{ext}} - \nabla E_{\text{elastic}}(\mathrm{x}^{i+1}) - \nabla E_{\text{damp}}(\mathrm{x}^{i+1}, \mathrm{x}^i),$$

but no external forces were used in this project.

The gradients of E_{elastic} and E_{damp} are needed to compute the force vector F. Furthermore, the nonlinear system of equations (6) is solved using Newton's method, which requires the Jacobian of F. First and second derivatives of the energies are unpleasant to derive, especially the bending term [Chen et al. 2018]. Therefore, the code for the energy derivatives was obtained from the first author of [Chen et al. 2018]. To verify the code was incorporated correctly the gradient of E_{stretch} was derived and is given in Appendix A.

Newton's method is implemented to solve the nonlinear equation in terms of $\delta\mathrm{v} = \mathrm{v}^{i+1} - \mathrm{v}^i$. Substituting the first equation into the second in (6) gives

$$\Lambda\frac{\delta\mathrm{v}}{\Delta t} = \mathrm{F}\left(\mathrm{x}^i + \Delta t\mathrm{v}^{i+1}, \mathrm{x}^i\right). \tag{7}$$

Note we can also write $\mathrm{x}^{i+1} = \mathrm{x}^i + \Delta t\mathrm{v}^i + \Delta t(\mathrm{v}^{i+1} - \mathrm{v}^i) = \tilde{\mathrm{x}} + \Delta t\delta\mathrm{v}$, where $\tilde{\mathrm{x}} = \mathrm{x}^i + \Delta t\mathrm{v}^i$ is an initial guess for the new position using explicit Euler. Rearranging (7) and substituting $\mathrm{x}^{i+1} = \tilde{\mathrm{x}} + \Delta t \delta \mathrm{v}$, gives the nonlinear equation that is solved via Newton's method

$$\sigma(\delta \mathrm{v}) \equiv \Lambda \delta \mathrm{v} - \Delta t \mathrm{F}(\tilde{\mathrm{x}} + \Delta t \delta \mathrm{v}, \mathrm{v}^i) = 0.$$

Environmental stimuli. Environmentally induced changes to the shell are incorporated by updating the rest fundamental forms $\bar{\mathrm{a}}, \bar{\mathrm{b}}$. Setting and updating the rest state depends on the model chosen for growth/shrinkage of the shell. The main models for the rest metric in [Chen et al. 2018] are no growth, isotropic growth, linear swelling, and piecewise constant swelling.

The two models used for this project are no growth and isotropic growth. The no growth model is that of a classic shell. The rest state is fixed as the initial fundamental forms a^0 and b^0, i.e., $\bar{\mathrm{a}} = \mathrm{a}^0$ and $\bar{\mathrm{b}} = \mathrm{b}^0$ for all time. In the isotropic growth model it is assumed that the shell deforms uniformly throughout the thickness. Therefore, we set $\bar{\mathrm{a}} = e^{2s_{ijk}} \mathrm{a}^0$ and $\bar{\mathrm{b}} = \mathrm{b}^0$, where s_{ijk} is the conformal factor for the amount of growth or shrinkage.

Due to time constraints, incorporating moisture from the surrounding environment was not accomplished in this project. The linear and piecewise constant swelling models in [Chen et al. 2018] are designed to handle moisture (or heat). The amount of additional moisture in the shell is modelled using a diffusion equation with source terms on the boundary of the shell. This additional moisture computed throughout the volume alters the rest state. For phenomena that add mass to the shell (e.g., coffee soaking into paper) the mass matrix Λ must also be updated.

4 RESULTS

4.1 Implementation Details

The triangle mesh data structure was created from scratch. It is based on the idea of a half-edge data structure, but is implemented as a face-based structure [OpenMesh 2019]. A triangle mesh is loaded from vertex and face data in a given .obj file format. The data structure defines directed edges between vertices on a face, while ensuring that all faces in the mesh have a consistent orientation (clockwise or counter-clockwise). Consistent orientation is achieved by having two edges between each vertex in opposing directions (except on the boundary). The geometry from the .obj file is checked to ensure the orientation of the faces is consistent and fixes any inconsistencies.

Positions x and velocities v are stored per vertex. Face normals n_{ijk}, edges e_i, and fundamental forms (a, b $\bar{\mathrm{a}}$, $\bar{\mathrm{b}}$) are stored per face. Mid-edge normals $\hat{\mathrm{n}}_{\mathrm{e}_i}$ and a pointer to the opposite face are stored per edge. A mapping from edge to incident faces is needed to compute the mid-edge normal.

Eigen [Guennebaud et al. 2010] is used for matrix math throughout the project. The Newton gradient $\nabla \sigma = \Lambda - \Delta t^2 \nabla \mathrm{F}$ is symmetric and almost always positive-definite. Therefore, a sparse Cholesky decomposition is used for Newton's method. In [Chen et al. 2018], the authors encountered cases when the matrix was not positive-definite. In these cases they regularized $\nabla \sigma$. Regularization was not needed for the examples in this project.

A line search method was used in [Chen et al. 2018] to allow larger time steps. No line search was implemented with Newton's method in this project and a step size of $\Delta t = 5 \times 10^{-6}$ was used for all examples. Time steps of size $\Delta t = 10^{-4}$ were achieved in [Chen et al. 2018], which would speed up our simulations here. A line search was not implemented to simplify implementation and ensure everything worked properly first. Then, due to time constraints I was not able to implement it. Another optimization used in [Chen et al. 2018] is an inexact Hessian computation. Again, this was not used for this project to ensure correctness of the implementation (but is implemented and could be easily turned on).

Thickness h	0.1 mm	Viscosity η	5×10^{-13} Pa s
Young's Mod. E	2×10^9 Pa	Poisson Ratio ν	0.3
Density ρ	$250 \mathrm{kg/m}^3$		

Table 1: Physical parameters used for all numerical experiments in this project. These values are reasonable for ordinary paper [Chen et al. 2018].

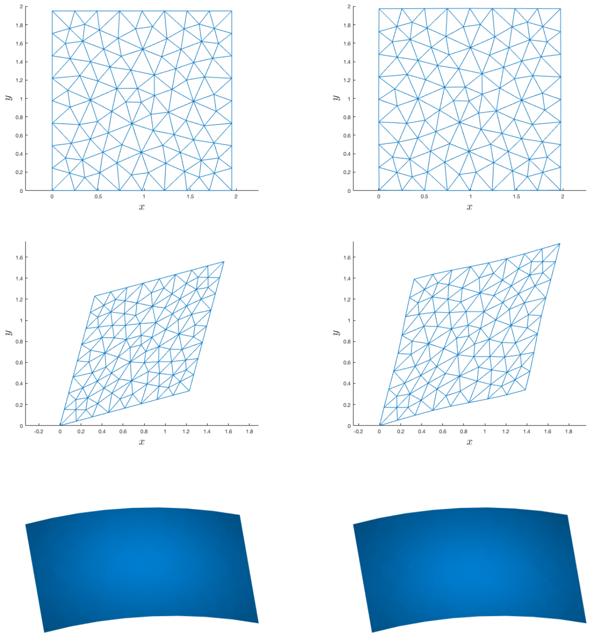

Figure 3: Deformation of the unit square using rest fundamental forms for an enlarged square (top), rhombus (middle), and cylindrical patch (bottom). The initial state (left) close to equilibrium is compared after 1000 times steps (right) with $\Delta t = 5 \times 10^{-6}$.

The physical parameters used throughout are those given in [Chen et al. 2018] and included here in Table 1. Note also that explicit Euler was implemented as a sanity check for the Newton's method implementation. It was noticed that explicit Euler needed times steps around an order of magnitude smaller than implicit Euler to remain stable.

4.2 Experiments

$\bar{\mathrm{a}}^{xy}$	$\bar{\mathrm{b}}^{xy}$	Shape
4I	0	Enlarged Square
$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$	0	Rhombus
I	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	Cylindrical Patch

Table 2: Rest fundamental forms for the deforming square.

Deforming square. An analytical benchmark of a deforming square is used to check that the implementation is correct. The parametrization domain Ω is set to the unit square. The square is represented by a triangulated mesh of 108 vertices and 184 faces. The no growth model is used for the rest fundamental forms, i.e., $\bar{\mathrm{a}} = \mathrm{a}^0$ and $\bar{\mathrm{b}} = \mathrm{b}^0$ for all t. It is expected that the square deform to equilibrate as the surface with the specified initial forms a^0 and b^0. The three shapes used in this project are an enlarged square, a rhombus, and a cylindrical patch (see Table 2).

The rest fundamental forms are specified in Table 2 with respect to the xy-plane because the shells initial configuration is flat. Therefore, we need to map the fundamental forms $\bar{\mathrm{a}}^{xy}$, $\bar{\mathrm{b}}^{xy}$ to the fundamental forms $\bar{\mathrm{a}}_{ijk}$, $\bar{\mathrm{b}}_{ijk}$ in the barycentric coordinates of each f_{ijk}. The transformation matrix that maps vectors in barycentric coordinates to $\mathbb{R}^2$ is

$$T = \left[\begin{array}{cc} \mathrm{x}_j^0 - \mathrm{x}_i^0 & \mathrm{x}_k^0 - \mathrm{x}_i^0. \end{array} \right]$$

Note that the transformation matrix is just $T = d\mathrm{r}$, which we used to construct $\mathrm{a} = d\mathrm{r}^T d\mathrm{r}$. In the initial flat configuration, however, the z-coordinate is a constant (say $z = 0$). Therefore, the transformation matrix $T \in \mathbb{R}^{2\times 2}$ is constructed with only the x and y components of x_i. The fundamental forms in barycentric coordinates are then

$$\bar{\mathrm{a}} = T^T \bar{\mathrm{a}}^{xy} T, \qquad \bar{\mathrm{b}} = T^T \bar{\mathrm{b}}^{xy} T.$$

The simulations take around 10 seconds per time step. The total simulation times are therefore on the order of hours due to the small times steps ($\Delta t = 5\times 10^{-6}$) and because growth/swelling phenomena take place at relatively long time scales [Chen et al. 2018]. Therefore, to decrease the computational time, the initial shape of the unit square was taken close to equilibrium. The exact initial shapes and how they were perturbed is discussed in Appendix B.

As is seen in Figure 3 the shapes stay relatively the same and are moving towards equilibrium. It can not be seen from these still images, but the accompanying videos show that the shapes oscillate bigger and smaller than the equilibrium state. In Figure 3, the enlarged square has moved from having lengths of 1.95 to closer to 2.0, but is still not there. Moreover, the rhombus has moved but its sides are bent slightly.

Figure 4 shows the elastic energy of all three configurations. It can be see that it oscillates towards equilibrium and convergence is relatively slow. Time steps of $\Delta t = 10^{-4}$ using a line search method would allow the simulation to skip every 20 times steps. Note also that the elastic energy of the cylindrical patch is much smaller than for the enlarged square and rhombus. This is expected because the bending energy is $O(h^3)$ and the cylindrical patch is only bending (no $O(h)$ stretching).

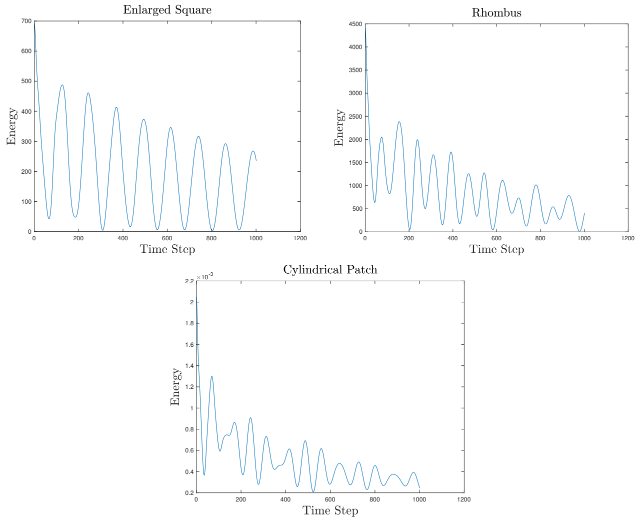

Figure 4: Elastic energy of the unit square while deforming to an enlarged square, rhombus, and cylindrical patch.

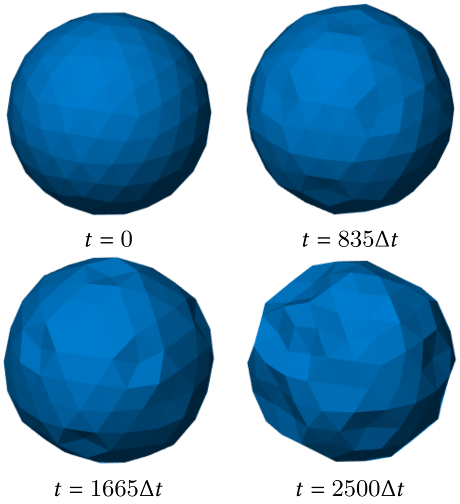

Figure 5: Contraction of a plastic sphere by applying uniform heat to the whole sphere.

Contracting sphere. In this simulation a plastic sphere is exposed to heat. To model the shrinking of plastic due to heat the isotropic growth model is used with a negative s_{ijk}. The unit sphere is approximated with 162 vertices and 320 faces.

The conformal factor s_{ijk} was specified somewhat manually since the heat was not modelled using diffusion, as discussed in section 3.2. For the first example we specify a constant value for s_{ijk} for all faces such that $\bar{\mathrm{a}} = 0.9\mathrm{a}^0$. Figure 5 shows the sphere shrinking and buckling due to the heat. Note that the time scale for the heat to buckle the plastic is obviously incorrect since we are not physically modelling the heat. However, the shape at the final time looks like a plausible outcome.

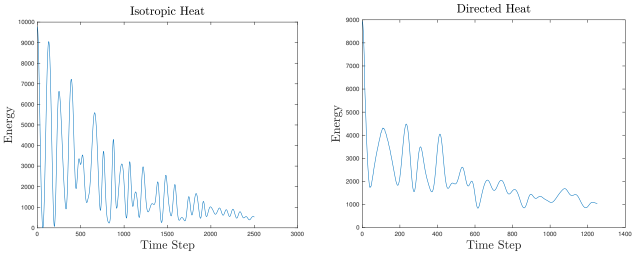

Figure 6: Elastic energy of the unit plastic sphere deforming due to heat.

It is unlikely that plastic would be heated completely uniformly around its shell. Therefore, a simulation using a directed heat source is constructed as follows. The s_{ijk} values were changed such that $\bar{\mathrm{a}} = 0.75\mathrm{a}^0$ for only faces having vertices with $z > 0.75$. Then the s_{ijk} for other faces were updated by averaging s_{ijk} from three adjacent faces with the current s_{ijk} to fake heat diffusion. Plastic objects usually only melt when close to the heat source. Therefore, the bottom of the sphere ($z \leq 0$) is fixed with $s_{ijk} = 0$ so that the bottom keeps $\bar{\mathrm{a}} = \mathrm{a}^0$.

Figure 1 shows the sphere at different times as it shrinks due to the directed heat source. Once again, the contraction and buckling behaviour seems plausible for this triangulated sphere of 320 faces. The energy for both uniform and directed heat sources is given in Figure 6, which expectedly decrease as t increase.

The included video of the simulation of the plastic spheres shrinking due to heat represent the dynamics better than the selected images included here. It is important to remember that the spheres oscillate towards equilibrium due to such small time steps (see Figure 6). Therefore, the spheres will not just contract, but will grow slighting and then shrink further. These oscillations would not be seen if the heat was modelled at the correct time scales instead of at scales of 5×10^{-6} seconds (but it would take too long to run).

5 CONCLUSION AND FUTURE WORK

Incorporating environmental effects using a physically accurate model is an important remaining piece. Moisture (temperature) changes are computed using a finite element method for diffusion on triangular prisms in [Chen et al. 2018]. Once the diffusion equation is solved on the volume, the moisture values on the top and bottom boundaries of the shell, m^+ and m^-, are used to update the rest fundamental forms $\bar{\mathrm{a}}$, $\bar{\mathrm{b}}$.

This model has proven to be difficult numerically since it only allows time steps less than 10^{-4}. It would be interesting to discretize the first and second fundamental forms in terms of a closest point representation of the surface [Ruuth and Merriman 2008]. Discretization using the closest point representation would result a numerical scheme that uses standard finite difference methods on Cartesian grids, while respecting the shell geometry. Using finite differences on Cartesian grids would be more easily optimized due to its regular structure.

This project aimed at implementing a physically-based thin-shell deformation model that responded to environmental stimuli. It followed closely the work in [Chen et al. 2018] for the discrete elastic energy and time evolution. From the example of a plastic sphere, it was demonstrated that environmental stimuli can deform the shell through changing the rest fundamental forms $\bar{\mathrm{a}}$ and $\bar{\mathrm{b}}$.

ACKNOWLEDGMENTS

The code for the energy and its derivatives was obtained from the first author of [Chen et al. 2018]. I have included the file she provided me (ShellEnergy.cpp) with my code. This should make it easier to know which parts are hers instead of looking through my code (although her code is only in my DeformableThinShells.cpp).

A wire-framed mesh viewer was created using OpenGL by modifying code from the OpenGL tutorial at http://www.opengl-tutorial.org/. However, the figures in this report were created using MATLAB.

I first created the triangulated square in MATLAB using the built in Delaunay triangulation function. This seemed to produce triangles that were not the best shapes near the boundaries. Therefore, I instead used a triangulated square given with an assignment in CS 778. For the sphere I also needed a triangulation will well shaped triangles. I therefore used the MATLAB code spheretri by Peter Gagarinov https://github.com/pgagarinov/spheretri.

REFERENCES

David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 43–54.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *ACM Transactions on Graphics (ToG)*, Vol. 21. ACM, 594–603.

Hsiao-Yu Chen, Arnav Sastry, Wim M van Rees, and Etienne Vouga. 2018. Physical simulation of environmentally induced thin shell deformation. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 146.

Philippe G Ciarlet. 2000. Mathematical Elasticity, Vol III, Theory of Shells. (2000).

David Correa and Achim Menges. 2017. Fused Filament Fabrication for Multi-Kinematic-State climate-responsice aperature.

Efi Efrati, Eran Sharon, and Raz Kupferman. 2009a. Buckling transition and boundary layer in non-Euclidean plates. *Physical Review E* 80, 1 (2009), 016602.

Efi Efrati, Eran Sharon, and Raz Kupferman. 2009b. Elastic theory of unconstrained non-Euclidean plates. *Journal of the Mechanics and Physics of Solids* 57, 4 (2009), 762–775.

Alain Goriely and Martine Ben Amar. 2005. Differential growth and instability in elastic shells. *Physical review letters* 94, 19 (2005), 198103.

Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. 2006. Computing discrete shape operators on general meshes. In *Computer Graphics Forum*, Vol. 25. Wiley Online Library, 547–556.

Eitan Grinspun, Anil N Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 62–67.

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

SoHyeon Jeong, Tae-hyeong Kim, and Chang-Hun Kim. 2011. Shrinkage, wrinkling and ablation of burning cloth and paper. *The Visual Computer* 27, 6-8 (2011), 417.

SoHyeon Jeong, Si-Hyung Park, and Chang-Hun Kim. 2013. Simulation of morphology changes in drying leaves. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 204–215.

Jungwook Kim, James A Hanna, Myunghwan Byun, Christian D Santangelo, and Ryan C Hayward. 2012. Designing responsive buckled surfaces by halftone gel lithography. *Science* 335, 6073 (2012), 1201–1205.

Yael Klein, Efi Efrati, and Eran Sharon. 2007. Shaping of elastic sheets by prescription of non-Euclidean metrics. *Science* 315, 5815 (2007), 1116–1120.

Caroline Larboulette, Pablo Quesada, and Olivier Dumas. 2013. Burning paper: Simulation at the fiber's level. In *Proceedings of Motion on Games*. ACM, 47–52.

Shiguang Liu, Qiguang Liu, Tai An, Jizhou Sun, and Qunsheng Peng. 2009. Physically based simulation of thin-shell objects' burning. *The Visual Computer* 25, 5-7 (2009), 687–696.

Frank Losasso, Geoffrey Irving, Eran Guendelman, and Ronald Fedkiw. 2006. Melting and burning solids into liquids and gases. *IEEE Transactions on Visualization and*

Computer Graphics 12, 3 (2006), 343–352.

Zeki Melek and John Keyser. 2003. Interactive simulation of burning objects. In *11th Pacific Conference onComputer Graphics and Applications, 2003. Proceedings.* IEEE, 462–466.

Zeki Melek and John Keyser. 2005. Multi-representation interaction for physically based modeling. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*. ACM, 187–196.

Zeki Melek and John Keyser. 2007. Driving object deformations from internal physical processes. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*. ACM, 51–59.

OpenMesh. 2019. The Halfedge Data Structure. https://www.openmesh.org/media/Documentations/OpenMesh-6.1-Documentation/a00016.html

Steven J Ruuth and Barry Merriman. 2008. A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.* 227, 3 (2008), 1943–1961.

Eran Sharon and Efi Efrati. 2010. The mechanics of non-Euclidean plates. *Soft Matter* 6, 22 (2010), 5693–5704.

Clarisse Weischedel. 2012. A discrete geometric view on shear-deformable shell models. (2012).

Dylan Marx Wood, David Correa, Oliver David Krieg, and Achim Menges. 2016. Material computation—4D timber construction: Towards building-scale hygroscopic actuated, self-constructing timber surfaces. *International Journal of Architectural Computing* 14, 1 (2016), 49–62.

A GRADIENT OF STRETCHING ENERGY

The energy for the stretching term is given by

$$E_{\text{stretch}}(\mathrm{x}) = \sum_{\mathrm{f}_{ijk}\in F} \left(\frac{h}{8}\|\bar{\mathrm{a}}_{ijk}^{-1}\mathrm{a}_{ijk} - \mathrm{I}\|_{SV}^{2}\right)\sqrt{\det \bar{\mathrm{a}}_{ijk}}.$$

The gradient of $\nabla E_{\text{elastic}} \in \mathbb{R}^{3m}$ is a vector in the $3m$-dimensional space of all the position coordinates. For simplicity, consider differentiating just with respect to one component x.

Grouping constants and writing out the norm gives the form

$$E = c_1 \mathrm{tr}^2\left(\bar{\mathrm{a}}^{-1}(\mathrm{a}-\bar{\mathrm{a}})\right) + c_2 \mathrm{tr}\left(\bar{\mathrm{a}}^{-1}(\mathrm{a}-\bar{\mathrm{a}})\right)^2,$$

where $c_1, c_2 \in \mathbb{R}$. For any matrix $A(x)$ we have that

$$\frac{d}{dx}\mathrm{tr}^2(A) = 2\mathrm{tr}(A)\mathrm{tr}\left(\frac{dA}{dx}\right)$$

and

$$\frac{d}{dx}\mathrm{tr}(A)^2 = \mathrm{tr}\left(A\frac{dA}{dx}\right) + \mathrm{tr}\left(\frac{dA}{dx}A\right) = 2\mathrm{tr}\left(A\frac{dA}{dx}\right),$$

using the property that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

We now just need to compute dA/dx, which in our case $A = \left(\bar{\mathrm{a}}^{-1}(\mathrm{a}-\bar{\mathrm{a}})\right)$. The rest fundamental form $\bar{\mathrm{a}}$ does not depend on the positions x, therefore we only need the derivative of a. Let $R = d\mathrm{r} = [\mathrm{x}_j - \mathrm{x}_i \mid \mathrm{x}_k - \mathrm{x}_i]$, then $\mathrm{a} = R^T R$ and

$$\frac{d\mathrm{a}}{dx} = \frac{d}{dx}\left(R^T R\right) = \left(\frac{dR}{dx}\right)^T R + R^T\left(\frac{dR}{dx}\right).$$

Simplifying using properties of the trace, it can be shown that

$$\frac{dE}{dx} = 2c_1\mathrm{tr}\left(R\bar{\mathrm{a}}^{-1}\left(\frac{dR}{dx}\right)^T\right) + 4c_2\mathrm{tr}\left(R\bar{\mathrm{a}}^{-1}A^T\left(\frac{dR}{dx}\right)^T\right).$$

This expression can be further simplified for implementation since the components of $\frac{dR}{dx}$ are just ±1.

B INITIAL CONDITIONS FOR DEFORMING SQUARE

Enlarged square. For the enlarged square $\bar{\mathrm{a}}^{xy} = 4I$ and $\bar{\mathrm{b}} = 0$. It is obvious that this shape is a square with side length 2 because the diagonal components in the definition of a are lengths squared. The initial state was taken as a square with side lengths of 1.95.

Rhombus. For the rhombus we have $\bar{\mathrm{b}} = 0$ and

$$\bar{\mathrm{a}}^{xy} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Looking at the components of a we have that

$$(\mathrm{x}_j - \mathrm{x}_i)\cdot(\mathrm{x}_j - \mathrm{x}_i) = 2,$$
$$(\mathrm{x}_j - \mathrm{x}_i)\cdot(\mathrm{x}_k - \mathrm{x}_i) = 1,$$
$$(\mathrm{x}_k - \mathrm{x}_i)\cdot(\mathrm{x}_k - \mathrm{x}_i) = 2,$$

and we need to find the vectors $\mathrm{x}_{ji} = (\mathrm{x}_j - \mathrm{x}_i)$ and $\mathrm{x}_{ki} = (\mathrm{x}_j - \mathrm{x}_i)$. Since the rhombus is 2D we can denote $\mathrm{x}_{ji} = [x_1, y_1]^T$ and $\mathrm{x}_{ki} = [x_2, y_2]^T$ giving three equations

$$x_1^2 + y_1^2 = 2,$$
$$x_1x_2 + y_1y_2 = 1,$$
$$x_2^2 + y_2^2 = 2,$$

with four unknowns. Using the symmetry of the rhombus about $y = x$ gives $x_1 = y_2$ and $y_1 = x_2$, which allows us to solve for $\mathrm{x}_{ji}, \mathrm{x}_{ki}$ as

$$\mathrm{x}_{ji} = \frac{1}{2}\begin{bmatrix} \sqrt{3}+1 \\ \sqrt{3}-1 \end{bmatrix}, \qquad \mathrm{x}_{ki} = \frac{1}{2}\begin{bmatrix} \sqrt{3}-1 \\ \sqrt{3}+1 \end{bmatrix}.$$

A transformation matrix $T = [\mathrm{x}_{ji} \mid \mathrm{x}_{ki}]$ is used to map vertices on the original mesh of the unit square to the rhombus spanned by $\{\mathrm{x}_{ji}, \mathrm{x}_{ki}\}$. The initial state was taken as 90% the size of the equilibrium rhombus using $0.9T$ for the transformation matrix.

Cylindrical patch. For the cylindrical patch we have $\bar{\mathrm{a}}^{xy} = I$ and

$$\bar{\mathrm{b}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

This means the mean curvature of the cylinder is $\kappa = 1$. A cylinder aligned along the y-axis only has curvature in the xz-plane. Viewing the cylinder down the y-axis gives a circle, whose curvature is $\kappa = 1/r$. Thus, the radius of the equilibrium cylinder is $r = 1$.

Therefore, to map the unit square in the xy-plane to the unit radius cylinder we just change the x and z coordinates by mapping unto the circle in the xz-plane. All y coordinates remain the same. The mapping is the usual mapping of $x = r\cos(\theta)$ and $z = r\sin(\theta)$ but with the angle θ carefully shifted as

$$\theta = \frac{x - 0.5}{r} - \frac{pi}{2}.$$

The initial state of the surface was a cylindrical patch of radius $r = 2$ instead of $r = 1$.